

TMS320C6655 and TMS320C6657 Fixed and Floating-Point Digital Signal Processor

1 C665x Features and Description

1.1 Features

- One (C6655) or Two (C6657) TMS320C66x™ DSP Core Subsystems (CorePacs), Each With
 - 850 MHz (C6657 only), 1.0 GHz, or 1.25 GHz C66x Fixed/Floating-Point CPU Core
 - 40 GMAC/Core for Fixed Point @ 1.25 GHz
 - 20 GFLOP/Core for Floating Point @ 1.25 GHz
- Multicore Shared Memory Controller (MSMC)
 - 1024KB MSM SRAM Memory (Shared by Two DSP C66x CorePacs for C6657)
 - Memory Protection Unit for Both MSM SRAM and DDR3_EMIF
- Multicore Navigator
 - 8192 Multipurpose Hardware Queues with Queue Manager
 - Packet-Based DMA for Zero-Overhead Transfers
- Hardware Accelerators
 - Two Viterbi Coprocessors
 - One Turbo Coprocessor Decoder
- Peripherals
 - Four Lanes of SRIO 2.1
 - 1.24/2.5/3.125/5 GBaud Operation Supported Per Lane
 - Supports Direct I/O, Message Passing
 - Supports Four 1x, Two 2x, One 4x, and Two 1x + One 2x Link Configurations
 - PCIe Gen2
 - Single Port Supporting 1 or 2 Lanes
 - Supports Up To 5 GBaud Per Lane
 - HyperLink
 - Supports Connections to Other KeyStone Architecture Devices Providing Resource Scalability
 - Supports up to 40 Gbaud
 - Gigabit Ethernet (GbE) Subsystem
 - One SGMII Port
 - Supports 10/100/1000 Mbps Operation
 - 32-Bit DDR3 Interface
 - DDR3-1333
 - 8G Byte Addressable Memory Space
 - 16-Bit EMIF
 - Universal Parallel Port
 - Two Channels of 8 bits or 16 bits Each
 - Supports SDR and DDR Transfers
 - Two UART Interfaces
 - Two Multichannel Buffered Serial Ports (McBSP)
 - I²C Interface
 - 32 GPIO Pins
 - SPI Interface
 - Semaphore Module
 - Eight 64-Bit Timers
 - Two On-Chip PLLs
- Commercial Temperature:
 - 0°C to 85°C
- Extended Temperature:
 - -40°C to 100°C
- Extended Low Temperature:
 - -55°C to 100°C

1.2 KeyStone Architecture

TI's KeyStone Multicore Architecture provides a high performance structure for integrating RISC and DSP cores with application specific coprocessors and I/O. KeyStone is the first of its kind that provides adequate internal bandwidth for nonblocking access to all processing cores, peripherals, coprocessors, and I/O. This is achieved with four main hardware elements: Multicore Navigator, TeraNet, Multicore Shared Memory Controller, and HyperLink.

Multicore Navigator is an innovative packet-based manager that controls 8192 queues. When tasks are allocated to the queues, Multicore Navigator provides hardware-accelerated dispatch that directs tasks to the appropriate available hardware. The packet-based system on a chip (SoC) uses the two Tbps capacity of the TeraNet switched central resource to move packets. The Multicore Shared Memory Controller enables processing cores to access shared memory directly without drawing from TeraNet's capacity, so packet movement cannot be blocked by memory access.



HyperLink provides a 40-Gbaud chip-level interconnect that allows SoCs to work in tandem. Its low-protocol overhead and high throughput make HyperLink an ideal interface for chip-to-chip interconnections. Working with Multicore Navigator, HyperLink dispatches tasks to tandem devices transparently and executes tasks as if they are running on local resources.

1.3 Trademarks

All trademarks are the property of their respective owners.

1.4 Device Description

The C665x DSP is a highest-performance fixed/floating-point DSP that is based on TI's KeyStone multicore architecture. Incorporating the new and innovative C66x DSP core, this device can run at a core speed of up to 1.25 GHz. For developers of a broad range of applications, such as mission critical, medical imaging, test and automation, and other applications requiring high performance, TI's C665x DSP offers up to 2.5 GHz cumulative DSP and enables a platform that is power-efficient and easy to use. In addition, it is fully backward compatible with all existing C6000 family of fixed and floating point DSPs.

TI's KeyStone architecture provides a programmable platform integrating various subsystems (C66x cores, memory subsystem, peripherals, and accelerators) and uses several innovative components and techniques to maximize intra-device and inter-device communication that allows the various DSP resources to operate efficiently and seamlessly. Central to this architecture are key components such as Multicore Navigator that allows for efficient data management between the various device components. The TeraNet is a non-blocking switch fabric enabling fast and contention-free internal data movement. The multicore shared memory controller allows access to shared and external memory directly without drawing from switch fabric capacity.

For fixed-point use, the C66x core has 4x the multiply accumulate (MAC) capability of C64x+ cores. In addition, the C66x core integrates floating point capability and the per core raw computational performance is an industry-leading 40 GMACS/core and 20 GFLOPS/core (@1.25 GHz operating frequency). It can execute 8 single precision floating point MAC operations per cycle and can perform double- and mixed-precision operations and is IEEE754 compliant. The C66x core incorporates 90 new instructions (compared to the C64x+ core) targeted for floating point and vector math oriented processing. These enhancements yield sizeable performance improvements in popular DSP kernels used in signal processing, mathematical, and image acquisition functions. The C66x core is backwards code compatible with TI's previous generation C6000 fixed and floating point DSP cores, ensuring software portability and shortened software development cycles for applications migrating to faster hardware.

The C665x DSP integrates a large amount of on-chip memory. In addition to 32KB of L1 program and data cache, there is 1024KB of dedicated memory per core that can be configured as mapped RAM or cache. The device also integrates 1024KB of Multicore Shared Memory that can be used as a shared L2 SRAM and/or shared L3 SRAM. All L2 memories incorporate error detection and error correction. For fast access to external memory, this device includes a 32-bit DDR-3 external memory interface (EMIF) running at 1333 MHz and has ECC DRAM support.

This family supports a number of high speed standard interfaces including RapidIO ver 2, PCI Express Gen2, and Gigabit Ethernet. It also includes I²C, UART, Multichannel Buffered Serial Port (McBSP), Universal Parallel Port, and a 16-bit asynchronous EMIF, along with general purpose CMOS IO. For high throughput, low latency communication between devices or with an FPGA, a 40-Gbaud full-duplex interface called HyperLink is included.

The C665x device has a complete set of development tools, which includes: an enhanced C compiler, an assembly optimizer to simplify programming and scheduling, and a Windows® debugger interface for visibility into source code execution.

1.5 Functional Block Diagram

Figure 1-1 shows the functional block diagram of the device.

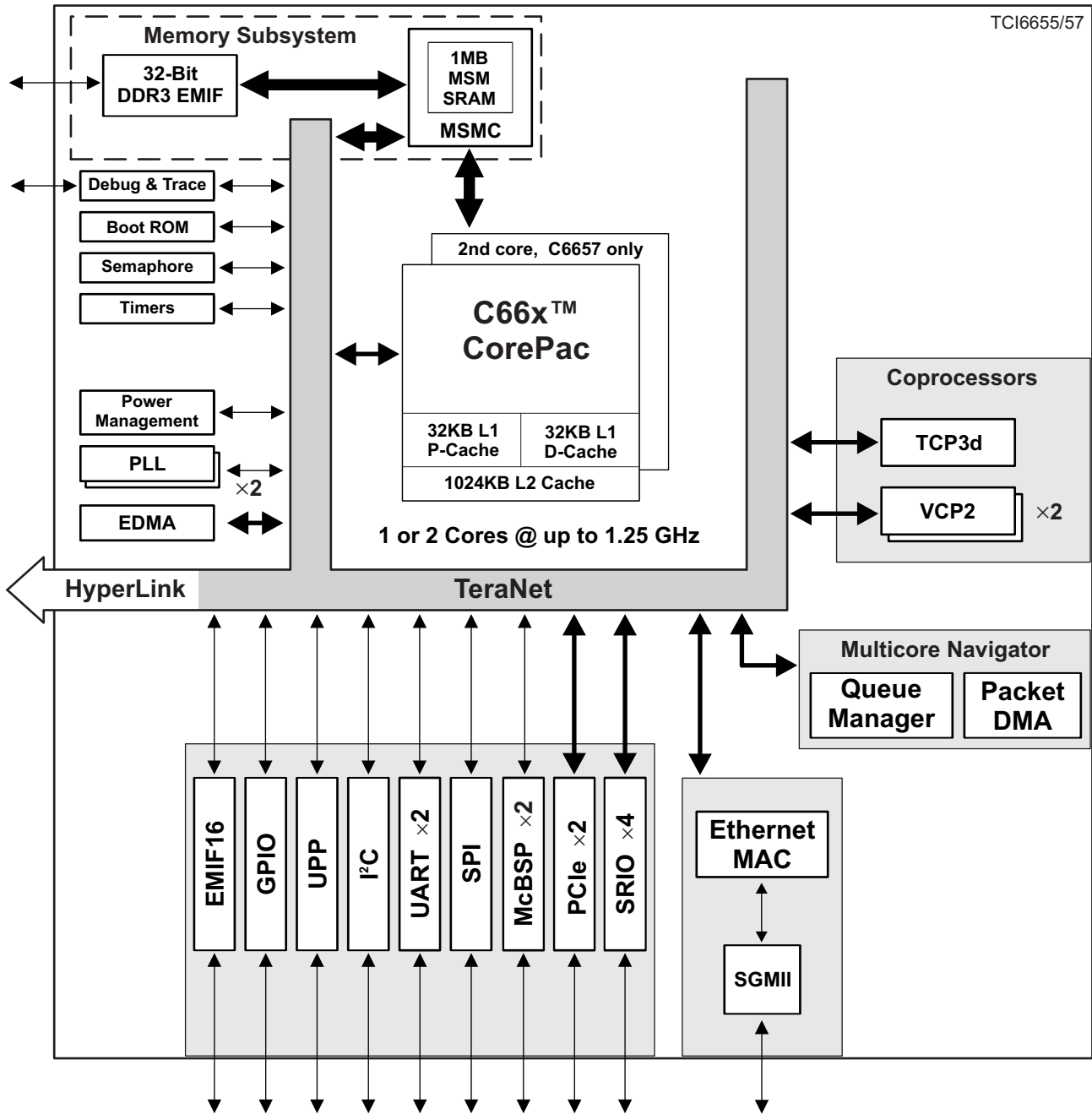


Figure 1-1. Functional Block Diagram

Table of Contents

| | | | | | |
|----------|--|----------------------------|----------|---|----------------------------|
| 1 | C665x Features and Description | 1 | 7.2 | Recommended Operating Conditions | 113 |
| 1.1 | Features | 1 | 7.3 | Electrical Characteristics | 114 |
| 1.2 | KeyStone Architecture | 1 | 7.4 | Power Supply to Peripheral I/O Mapping | 115 |
| 1.3 | Trademarks | 2 | 8 | Peripheral Information and Electrical Specifications | 116 |
| 1.4 | Device Description | 2 | 8.1 | Recommended Clock and Control Signal Transition Behavior | 116 |
| 1.5 | Functional Block Diagram | 3 | 8.2 | Power Supplies | 116 |
| 2 | Revision History | 5 | 8.3 | Power Sleep Controller (PSC) | 124 |
| 3 | Device Overview | 6 | 8.4 | Reset Controller | 128 |
| 3.1 | Device Characteristics | 6 | 8.5 | Main PLL and PLL Controller | 134 |
| 3.2 | DSP Core Description | 7 | 8.6 | DDR3 PLL | 151 |
| 3.3 | Memory Map Summary | 10 | 8.7 | Enhanced Direct Memory Access (EDMA3) Controller | 154 |
| 3.4 | Boot Sequence | 14 | 8.8 | Interrupts | 158 |
| 3.5 | Boot Modes Supported and PLL Settings | 15 | 8.9 | Memory Protection Unit (MPU) | 183 |
| 3.6 | PLL Boot Configuration Settings | 37 | 8.10 | DDR3 Memory Controller | 198 |
| 3.7 | Second-Level Bootloaders | 37 | 8.11 | I ² C Peripheral | 199 |
| 3.8 | Terminals | 38 | 8.12 | I ² C Timing Requirements | 202 |
| 3.9 | Terminal Functions | 43 | 8.13 | SPI Peripheral | 204 |
| 3.10 | Development and Support | 65 | 8.14 | HyperLink Peripheral | 207 |
| 3.11 | Related Documentation from Texas Instruments | 67 | 8.15 | UART Peripheral | 211 |
| 4 | Device Configuration | 68 | 8.16 | PCIe Peripheral | 212 |
| 4.1 | Device Configuration at Device Reset | 68 | 8.17 | EMIF16 Peripheral | 213 |
| 4.2 | Peripheral Selection After Device Reset | 69 | 8.18 | Ethernet Media Access Controller (EMAC) | 216 |
| 4.3 | Device State Control Registers | 69 | 8.19 | Management Data Input/Output (MDIO) | 222 |
| 4.4 | Pullup/Pulldown Resistors | 96 | 8.20 | Timers | 224 |
| 5 | System Interconnect | 97 | 8.21 | General-Purpose Input/Output (GPIO) | 225 |
| 5.1 | Internal Buses and Switch Fabrics | 97 | 8.22 | Semaphore2 | 226 |
| 5.2 | Switch Fabric Connections Matrix | 97 | 8.23 | Multichannel Buffered Serial Port (McBSP) | 226 |
| 5.3 | TeraNet Switch Fabric Connections | 100 | 8.24 | Universal Parallel Port (uPP) | 231 |
| 5.4 | Bus Priorities | 103 | 8.25 | Serial RapidIO (SRIO) Port | 235 |
| 6 | C66x CorePac | 105 | 8.26 | Turbo Decoder Coprocessor (TCP3d) | 236 |
| 6.1 | Memory Architecture | 106 | 8.27 | Enhanced Viterbi-Decoder Coprocessor (VCP2) | 236 |
| 6.2 | Memory Protection | 109 | 8.28 | Emulation Features and Capability | 236 |
| 6.3 | Bandwidth Management | 110 | 8.29 | Related Links | 239 |
| 6.4 | Power-Down Control | 110 | 9 | Mechanical Data | 240 |
| 6.5 | C66x CorePac Revision | 111 | 9.1 | Thermal Data | 240 |
| 6.6 | C66x CorePac Register Descriptions | 111 | 9.2 | Packaging Information | 240 |
| 7 | Device Operating Conditions | 112 | | | |
| 7.1 | Absolute Maximum Ratings | 112 | | | |

2 Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

| Changes from September 1, 2012 to April 24, 2015 | Page |
|--|---------------------|
| • Removed Security/Key Manager from Figure 1-1 | 3 |
| • Removed "Secure ROM Boot" and changed "Public ROM Boot" to "ROM Boot" in Section 3.5 | 15 |
| • Added Boot Parameter Table section | 28 |
| • Updated OUTPUT_DIVIDE default value and PLL clock formula in PLL Settings section | 37 |
| • Clarified SmartReflex pin output type | 53 |
| • Removed SECURITY information from Figure 3-31 | 66 |
| • Corrected EMAC User's Guide link in Related Documentation section | 67 |
| • Added DDR3PLLCTL1 register to Device Status Control Registers table | 71 |
| • Updated PKTDMA_PRI_ALLOC register to be CHIP_MSIC_CTL register with new bit field added | 72 |
| • Corrected SmartReflex peripheral I/O Buffer Type from LVCMOS category to Open drain | 115 |
| • Updated Power Domain 12 to be VCP2_B from VCP2. Changed all references to VCP20, and VCP2-A to VCP2_A. Changed all references to VCP21, and VCP2-B to VCP2_B. | 124 |
| • Updated "slow peripherals" in SYSCLK7 description | 136 |
| • Updated BWADJ value setting description in Main/DDR3 PLL control registers | 148 |
| • Updated all SerDes clocks to discrete frequencies in the Clock Input Timing Requirements table | 149 |
| • Corrected differential clock rise and fall time in the PLL timing table for the clock inputs that feed into the LJCB clock buffers | 149 |
| • Added note to DDR3 PLL initialization sequence | 152 |
| • Clarified table caption and first column heading | 160 |
| • Updated event "po_vp_smpsack_intr" to be Reserved in CIC2 event table | 174 |
| • Corrected MPU0 Memory Protection End Address from 0x026203FF to 0x026207FF | 183 |
| • Removed SECURITY LEVEL column from Table 8-45 | 184 |
| • Updated/Changed Bit 7 of Table 8-55 from "NS" to "Reserved" | 195 |
| • Added MPU Registers Reset Values section | 196 |
| • Updated the Min/Max values of EMIF read cycle time and write cycle time | 213 |
| • Updated Timer number description across the data manual | 224 |
| • Changed all footnote references from CORECLK to SYSCLK1 | 225 |
| • Updated the descriptions of how Semaphore module is accessible | 226 |
| • Corrected McBSP FIFO Control and Status Register address to be 0x021B6000 for McBSP0 and 0x021BA000 for McBSP1 | 227 |
| • Corrected McBSP FIFO Data Register address to be 0x22400000 for McBSP1 | 227 |
| • Updated Trace Electrical Timing tables and Timing diagrams | 237 |

3 Device Overview

3.1 Device Characteristics

Table 3-1. Characteristics of the C665x Processor

| HARDWARE FEATURES | | TMS320C6655 | TMS320C6657 |
|--|--|---|------------------|
| Peripheral | DDR3 Memory Controller (32-bit bus width) [1.5 V I/O] (clock source = DDRREFCLKNP) | 1 | |
| | DDR3 Maximum Data Rate | 1333 | |
| | EDMA3 (64 independent channels) [DSP/3 clock rate] | 1 | |
| | High-speed 1x/2x/4x Serial RapidIO Port (4 lanes) | 1 | |
| | PCIe (2 lanes) | 1 | |
| | 10/100/1000 Ethernet | 1 | |
| | Management Data Input/Output (MDIO) | 1 | |
| | HyperLink | 1 | |
| | EMIF16 | 1 | |
| | McBSP | 2 | |
| | SPI | 1 | |
| | UART | 2 | |
| | uPP | 1 | |
| | I ² C | 1 | |
| | 64-Bit Timers (configurable) (internal clock source = CPU/6 clock frequency) | 8 (each configurable as two 32-bit timers) | |
| | General-Purpose Input/Output port (GPIO) | 32 | |
| Encoder/ Decoder Coproprocessors | VCP2 (clock source = CPU/3 clock frequency) | 2 | |
| | TCP3d (clock source = CPU/2 clock frequency) | 1 | |
| On-Chip Memory | CorePac Memory | 32KB L1 Program Memory [SRAM/Cache] 32KB L1 Data Memory [SRAM/Cache] 1024KB L2 Unified Memory/Cache | |
| | ROM Memory | 128KB L3 ROM | |
| | Multicore Shared Memory | 1024KB MSM SRAM | |
| C66x CorePac Revision ID | CorePac Revision ID Register (address location: 0181 2000h) | See Section 6.5 | |
| JTAG BSDL_ID | JTAGID register (address location: 0262 0018h) | See Section 4.3.3 | |
| Frequency | MHz | 1250 (1.25GHz) | |
| | | 1000 (1.0 GHz) | |
| | | - | 850 (0.85 GHz) |
| Cycle Time | ns | 0.8 (1.25 GHz) | |
| | | 1 (1.0 GHz) | |
| | | - | 1.175 (0.85 GHz) |
| Voltage | Core (V) | SmartReflex variable supply | |
| | I/O (V) | 1.0 V, 1.5 V, and 1.8 V | |
| Process Technology | µm | 0.040 µm | |
| BGA Package | 21 mm × 21mm | 625-Pin Flip-Chip Plastic BGA (CZH or GZH) | |
| Product Status ⁽¹⁾ | Production Data (PD) | PD | PD |

(1) PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

3.2 DSP Core Description

The C66x Digital Signal Processor (DSP) extends the performance of the C64x+ and C674x DSPs through enhancements and new features. Many of the new features target increased performance for vector processing. The C64x+ and C674x DSPs support 2-way SIMD operations for 16-bit data and 4-way SIMD operations for 8-bit data. On the C66x DSP, the vector processing capability is improved by extending the width of the SIMD instructions. C66x DSPs can execute instructions that operate on 128-bit vectors. For example the QMPY32 instruction is able to perform the element-to-element multiplication between two vectors of four 32-bit data each. The C66x DSP also supports SIMD for floating-point operations. Improved vector processing capability (each instruction can process multiple data in parallel) combined with the natural instruction level parallelism of C6000 architecture (e.g execution of up to 8 instructions per cycle) results in a very high level of parallelism that can be exploited by DSP programmers through the use of TI's optimized C/C++ compiler.

The C66x DSP consists of eight functional units, two register files, and two data paths as shown in [Figure 3-1](#). The two general-purpose register files (A and B) each contain 32 32-bit registers for a total of 64 registers. The general-purpose registers can be used for data or can be data address pointers. The data types supported include packed 8-bit data, packed 16-bit data, 32-bit data, 40-bit data, and 64-bit data. Multiplies also support 128-bit data. 40-bit-long or 64-bit-long values are stored in register pairs, with the 32 LSBs of data placed in an even register and the remaining 8 or 32 MSBs in the next upper register (which is always an odd-numbered register). 128-bit data values are stored in register quadruplets, with the 32 LSBs of data placed in a register that is a multiple of 4 and the remaining 96 MSBs in the next 3 upper registers.

The eight functional units (.M1, .L1, .D1, .S1, .M2, .L2, .D2, and .S2) are each capable of executing one instruction every clock cycle. The .M functional units perform all multiply operations. The .S and .L units perform a general set of arithmetic, logical, and branch functions. The .D units primarily load data from memory to the register file and store results from the register file into memory.

Each C66x .M unit can perform one of the following fixed-point operations each clock cycle: four 32×32 bit multiplies, sixteen 16×16 bit multiplies, four 16×32 bit multiplies, four 8×8 bit multiplies, four 8×8 bit multiplies with add operations, and four 16×16 multiplies with add/subtract capabilities. There is also support for Galois field multiplication for 8-bit and 32-bit data. Many communications algorithms such as FFTs and modems require complex multiplication. Each C66x .M unit can perform one 16×16 bit complex multiply with or without rounding capabilities, two 16×16 bit complex multiplies with rounding capability, and a 32×32 bit complex multiply with rounding capability. The C66x can also perform two 16×16 bit and one 32×32 bit complex multiply instructions that multiply a complex number with a complex conjugate of another number with rounding capability. Communication signal processing also requires an extensive use of matrix operations. Each C66x .M unit is capable of multiplying a $[1 \times 2]$ complex vector by a $[2 \times 2]$ complex matrix per cycle with or without rounding capability. A version also exists allowing multiplication of the conjugate of a $[1 \times 2]$ vector with a $[2 \times 2]$ complex matrix.

Each C66x .M unit also includes IEEE floating-point multiplication operations from the C674x DSP, which includes one single-precision multiply each cycle and one double-precision multiply every 4 cycles. There is also a mixed-precision multiply that allows multiplication of a single-precision value by a double-precision value and an operation allowing multiplication of two single-precision numbers resulting in a double-precision number. The C66x DSP improves the performance over the C674x double-precision multiplies by adding a instruction allowing one double-precision multiply per cycle and also reduces the number of delay slots from 10 down to 4. Each C66x .M unit can also perform one the following floating-point operations each clock cycle: one, two, or four single-precision multiplies or a complex single-precision multiply.

The .L and .S units can now support up to 64-bit operands. This allows for new versions of many of the arithmetic, logical, and data packing instructions to allow for more parallel operations per cycle. Additional instructions were added yielding performance enhancements of the floating point addition and subtraction instructions, including the ability to perform one double precision addition or subtraction per cycle. Conversion to/from integer and single-precision values can now be done on both .L and .S units on the C66x. Also, by taking advantage of the larger operands, instructions were also added to double the number of these conversions that can be done. The .L unit also has additional instructions for logical AND and OR instructions, as well as, 90 degree or 270 degree rotation of complex numbers (up to two per cycle). Instructions have also been added that allow for the computing the conjugate of a complex number.

The MFENCE instruction is a new instruction introduced on the C66x DSP. This instruction will create a DSP stall until the completion of all the DSP-triggered memory transactions, including:

- Cache line fills
- Writes from L1D to L2 or from the CorePac to MSMC and/or other system endpoints
- Victim write backs
- Block or global coherence operations
- Cache mode changes
- Outstanding XMC prefetch requests

This is useful as a simple mechanism for programs to wait for these requests to reach their endpoint. It also provides ordering guarantees for writes arriving at a single endpoint via multiple paths, multiprocessor algorithms that depend on ordering, and manual coherence operations.

For more details on the C66x DSP and its enhancements over the C64x+ and C674x architectures, see the following documents:

- *C66x CPU and Instruction Set Reference Guide* ([SPRUGH7](#)).
- *C66x DSP Cache User's Guide* ([SPRUGY8](#)).
- *C66x CorePac User's Guide* ([SPRUGW0](#)).

[Figure 3-1](#) shows the DSP core functional units and data paths.

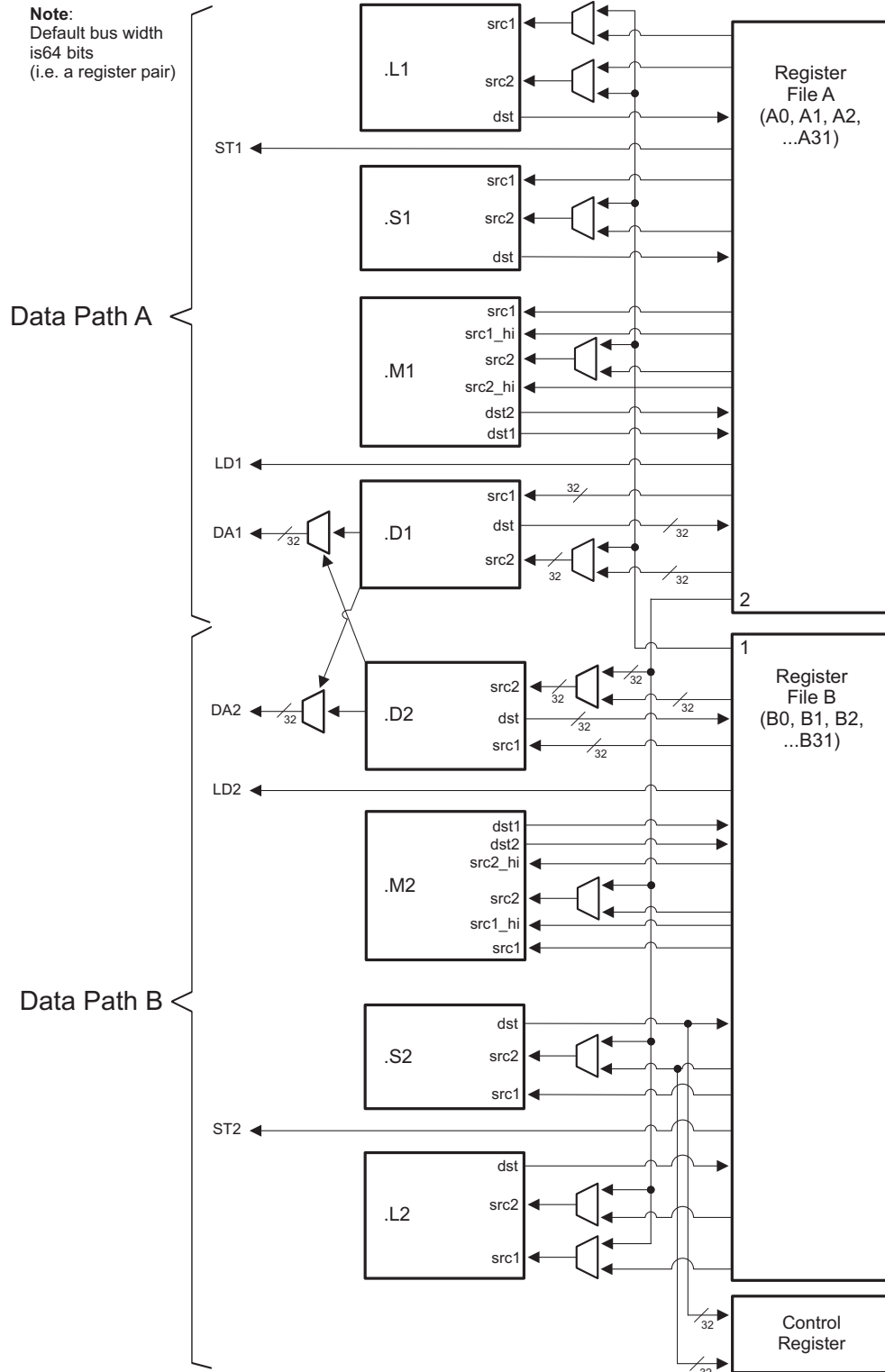


Figure 3-1. DSP Core Data Paths

3.3 Memory Map Summary

Table 3-2 shows the memory map address ranges of the C665x device.

Table 3-2. Memory Map Summary

| LOGICAL 32-BIT ADDRESS | | PHYSICAL 36-BIT ADDRESS | | BYTES | DESCRIPTION |
|------------------------|------------|-------------------------|--------------|------------|--|
| START | END | START | END | | |
| 00000000 | 007FFFFFFF | 0 00000000 | 0 007FFFFFFF | 8M | Reserved |
| 00800000 | 008FFFFFFF | 0 00800000 | 0 008FFFFFFF | 1M | Local L2 SRAM |
| 00900000 | 00DFFFFFFF | 0 00900000 | 0 00DFFFFFFF | 5M | Reserved |
| 00E00000 | 00E07FFF | 0 00E00000 | 0 00E07FFF | 32K | Local L1P SRAM |
| 00E08000 | 00EFFFFFFF | 0 00E08000 | 0 00EFFFFFFF | 1M-32K | Reserved |
| 00F00000 | 00F07FFF | 0 00F00000 | 0 00F07FFF | 32K | Local L1D SRAM |
| 00F08000 | 017FFFFFFF | 0 00F08000 | 0 017FFFFFFF | 9M-32K | Reserved |
| 01800000 | 01BFFFFFFF | 0 01800000 | 0 01BFFFFFFF | 4M | C66x CorePac Registers |
| 01C00000 | 01CFFFFFFF | 0 01C00000 | 0 01CFFFFFFF | 1M | Reserved |
| 01D00000 | 01D0007F | 0 01D00000 | 0 01D0007F | 128 | Tracer_MSMC_0 |
| 01D00080 | 01D07FFF | 0 01D00080 | 0 01D07FFF | 32K-128 | Reserved |
| 01D08000 | 01D0807F | 0 01D08000 | 0 01D0807F | 128 | Tracer_MSMC_1 |
| 01D08080 | 01D0FFFF | 0 01D08080 | 0 01D0FFFF | 32K-128 | Reserved |
| 01D10000 | 01D1007F | 0 01D10000 | 0 01D1007F | 128 | Tracer_MSMC_2 |
| 01D10080 | 01D17FFF | 0 01D10080 | 0 01D17FFF | 32K-128 | Reserved |
| 01D18000 | 01D1807F | 0 01D18000 | 0 01D1807F | 128 | Tracer_MSMC_3 |
| 01D18080 | 01D1FFFF | 0 01D18080 | 0 01D1FFFF | 32K-128 | Reserved |
| 01D20000 | 01D2007F | 0 01D20000 | 0 01D2007F | 128 | Tracer_QM_DMA |
| 01D20080 | 01D27FFF | 0 01D20080 | 0 01D27FFF | 32K-128 | Reserved |
| 01D28000 | 01D2807F | 0 01D28000 | 0 01D2807F | 128 | Tracer_DDR |
| 01D28080 | 01D2FFFF | 0 01D28080 | 0 01D2FFFF | 32K-128 | Reserved |
| 01D30000 | 01D3007F | 0 01D30000 | 0 01D3007F | 128 | Tracer_SM |
| 01D30080 | 01D37FFF | 0 01D30080 | 0 01D37FFF | 32K-128 | Reserved |
| 01D38000 | 01D3807F | 0 01D38000 | 0 01D3807F | 128 | Tracer_QM_CFG |
| 01D38080 | 01D3FFFF | 0 01D38080 | 0 01D3FFFF | 32K-128 | Reserved |
| 01D40000 | 01D4007F | 0 01D40000 | 0 01D4007F | 128 | Tracer_CFG |
| 01D40080 | 01D47FFF | 0 01D40080 | 0 01D47FFF | 32K-128 | Reserved |
| 01D48000 | 01D4807F | 0 01D48000 | 0 01D4807F | 128 | Tracer_L2_0 |
| 01D48080 | 01D4FFFF | 0 01D48080 | 0 01D4FFFF | 32K-128 | Reserved |
| 01D50000 | 01D5007F | 0 01D50000 | 0 01D5007F | 128 | Tracer_L2_1(C6657) or Reserved (C6655) |
| 01D50080 | 01D57FFF | 0 01D50080 | 0 01D57FFF | 32K-128 | Reserved |
| 01D58000 | 01D5807F | 0 01D58000 | 0 01D5807F | 128 | Tracer_TNet_6P_A |
| 01D58080 | 021B3FFF | 0 01D58080 | 0 021B3FFF | 4464K -128 | Reserved |
| 021B4000 | 021B47FF | 0 021B4000 | 0 021B47FF | 2K | McBSP0 Registers |
| 021B4800 | 021B5FFF | 0 021B4800 | 0 021B5FFF | 6K | Reserved |
| 021B6000 | 021B67FF | 0 021B6000 | 0 021B67FF | 2K | McBSP0 FIFO Registers |
| 021B6800 | 021B7FFF | 0 021B6800 | 0 021B7FFF | 6K | Reserved |
| 021B8000 | 021B87FF | 0 021B8000 | 0 021B87FF | 2K | McBSP1 Registers |
| 021B8800 | 021B9FFF | 0 021B8800 | 0 021B9FFF | 6K | Reserved |
| 021BA000 | 021BA7FF | 0 021BA000 | 0 021BA7FF | 2K | McBSP1 FIFO Registers |
| 021BA800 | 021BFFFF | 0 021BA800 | 0 021BFFFF | 22K | Reserved |
| 021C0000 | 021C03FF | 0 021C0000 | 0 021C03FF | 1K | TCP3d Registers |

Table 3-2. Memory Map Summary (continued)

| LOGICAL 32-BIT ADDRESS | | PHYSICAL 36-BIT ADDRESS | | BYTES | DESCRIPTION |
|------------------------|-----------|-------------------------|-------------|------------|---|
| START | END | START | END | | |
| 021C0400 | 021CFFFF | 0 021C0400 | 0 021CFFFF | 63K | Reserved |
| 021D0000 | 021D00FF | 0 021D0000 | 0 021D00FF | 256 | VCP2_A Registers |
| 021D0100 | 021D3FFF | 0 021D0100 | 0 021D3FFF | 16K - 256 | Reserved |
| 021D4000 | 021D40FF | 0 021D4000 | 0 021D40FF | 256 | VCP2_B Registers |
| 021D4100 | 021FFFFFF | 0 021D4100 | 0 021FFFFFF | 176K - 256 | Reserved |
| 02200000 | 0220007F | 0 02200000 | 0 0220007F | 128 | Timer0 |
| 02200080 | 0220FFFF | 0 02200080 | 0 0220FFFF | 64K-128 | Reserved |
| 02210000 | 0221007F | 0 02210000 | 0 0221007F | 128 | Reserved |
| 02210080 | 0221FFFF | 0 02210080 | 0 0221FFFF | 64K-128 | Reserved |
| 02220000 | 0222007F | 0 02220000 | 0 0222007F | 128 | Timer2 |
| 02220080 | 0222FFFF | 0 02220080 | 0 0222FFFF | 64K-128 | Reserved |
| 02230000 | 0223007F | 0 02230000 | 0 0223007F | 128 | Timer3 |
| 02230080 | 0223FFFF | 0 02230080 | 0 0223FFFF | 64K-128 | Reserved |
| 02240000 | 0224007F | 0 02240000 | 0 0224007F | 128 | Timer4 |
| 02240080 | 0224FFFF | 0 02240080 | 0 0224FFFF | 64K-128 | Reserved |
| 02250000 | 0225007F | 0 02250000 | 0 0225007F | 128 | Timer5 |
| 02250080 | 0225FFFF | 0 02250080 | 0 0225FFFF | 64K-128 | Reserved |
| 02260000 | 0226007F | 0 02260000 | 0 0226007F | 128 | Timer6 |
| 02260080 | 0226FFFF | 0 02260080 | 0 0226FFFF | 64K-128 | Reserved |
| 02270000 | 0227007F | 0 02270000 | 0 0227007F | 128 | Timer7 |
| 02270080 | 0230FFFF | 0 02270080 | 0 0230FFFF | 640K - 128 | Reserved |
| 02310000 | 023101FF | 0 02310000 | 0 023101FF | 512 | PLL Controller |
| 02310200 | 0231FFFF | 0 02310200 | 0 0231FFFF | 64K-512 | Reserved |
| 02320000 | 023200FF | 0 02320000 | 0 023200FF | 256 | GPIO |
| 02320100 | 0232FFFF | 0 02320100 | 0 0232FFFF | 64K-256 | Reserved |
| 02330000 | 023303FF | 0 02330000 | 0 023303FF | 1K | SmartReflex |
| 02330400 | 0234FFFF | 0 02330400 | 0 0234FFFF | 127K | Reserved |
| 02350000 | 02350FFF | 0 02350000 | 0 02350FFF | 4K | Power Sleep Controller (PSC) |
| 02351000 | 0235FFFF | 0 02351000 | 0 0235FFFF | 64K-4K | Reserved |
| 02360000 | 023603FF | 0 02360000 | 0 023603FF | 1K | Memory Protection Unit (MPU) 0 |
| 02360400 | 02367FFF | 0 02360400 | 0 02367FFF | 31K | Reserved |
| 02368000 | 023683FF | 0 02368000 | 0 023683FF | 1K | Memory Protection Unit (MPU) 1 |
| 02368400 | 0236FFFF | 0 02368400 | 0 0236FFFF | 31K | Reserved |
| 02370000 | 023703FF | 0 02370000 | 0 023703FF | 1K | Memory Protection Unit (MPU) 2 |
| 02370400 | 02377FFF | 0 02370400 | 0 02377FFF | 31K | Reserved |
| 02378000 | 023783FF | 0 02378000 | 0 023783FF | 1K | Memory Protection Unit (MPU) 3 |
| 02378400 | 0237FFFF | 0 02378400 | 0 0237FFFF | 31K | Reserved |
| 02380000 | 023803FF | 0 02380000 | 0 023803FF | 1K | Memory Protection Unit (MPU) 4 |
| 02380400 | 023FFFFFF | 0 02380400 | 0 023FFFFFF | 511K | Reserved |
| 02440000 | 02443FFF | 0 02440000 | 0 02443FFF | 16K | DSP trace formatter 0 |
| 02444000 | 0244FFFF | 0 02444000 | 0 0244FFFF | 48K | Reserved |
| 02450000 | 02453FFF | 0 02450000 | 0 02453FFF | 16K | DSP trace formatter 1 (C6657) or Reserved (C6655) |
| 02454000 | 02521FFF | 0 02454000 | 0 02521FFF | 824K | Reserved |
| 02522000 | 02522FFF | 0 02522000 | 0 02522FFF | 4K | Efuse |
| 02523000 | 0252FFFF | 0 02523000 | 0 0252FFFF | 52K | Reserved |
| 02530000 | 0253007F | 0 02530000 | 0 0253007F | 128 | I ² C data & control |

Table 3-2. Memory Map Summary (continued)

| LOGICAL 32-BIT ADDRESS | | PHYSICAL 36-BIT ADDRESS | | BYTES | DESCRIPTION |
|------------------------|------------|-------------------------|--------------|-----------|--|
| START | END | START | END | | |
| 02530080 | 0253FFFF | 0 02530080 | 0 0253FFFF | 64K-128 | Reserved |
| 02540000 | 0254003F | 0 02540000 | 0 0254003F | 64 | UART 0 |
| 02540400 | 0254FFFF | 0 02540400 | 0 0254FFFF | 64K-64 | Reserved |
| 02550000 | 0255003F | 0 02550000 | 0 0255003F | 64 | UART 1 |
| 02550040 | 0257FFFF | 0 02550040 | 0 0257FFFF | 192K-64 | Reserved |
| 02580000 | 02580FFF | 0 02580000 | 0 02580FFF | 4K | uPP |
| 02581000 | 025FFFFFFF | 0 02581000 | 0 025FFFFFFF | 508K | Reserved |
| 02600000 | 02601FFF | 0 02600000 | 0 02601FFF | 8K | Chip Interrupt Controller (CIC) 0 |
| 02602000 | 02603FFF | 0 02602000 | 0 02603FFF | 8K | Reserved |
| 02604000 | 02605FFF | 0 02604000 | 0 02605FFF | 8K | Chip Interrupt Controller (CIC) 1 |
| 02606000 | 02607FFF | 0 02606000 | 0 02607FFF | 8K | Reserved |
| 02608000 | 02609FFF | 0 02608000 | 0 02609FFF | 8K | Chip Interrupt Controller (CIC) 2 |
| 0260A000 | 0261FFFF | 0 0260A000 | 0 0261FFFF | 88K | Reserved |
| 02620000 | 026207FF | 0 02620000 | 0 026207FF | 2K | Chip-Level Registers |
| 02620800 | 0263FFFF | 0 02620800 | 0 0263FFFF | 126K | Reserved |
| 02640000 | 026407FF | 0 02640000 | 0 026407FF | 2K | Semaphore |
| 02640800 | 0273FFFF | 0 02640800 | 0 0273FFFF | 1022K | Reserved |
| 02740000 | 02747FFF | 0 02740000 | 0 02747FFF | 32K | EDMA Channel Controller (EDMA3CC) |
| 02748000 | 0278FFFF | 0 02748000 | 0 0278FFFF | 288K | Reserved |
| 02790000 | 027903FF | 0 02790000 | 0 027903FF | 1K | EDMA3CC Transfer Controller EDMA3TC0 |
| 02790400 | 02797FFF | 0 02790400 | 0 02797FFF | 31K | Reserved |
| 02798000 | 027983FF | 0 02798000 | 0 027983FF | 1K | EDMA3CC Transfer Controller EDMA3TC1 |
| 02798400 | 0279FFFF | 0 02798400 | 0 0279FFFF | 31K | Reserved |
| 027A0000 | 027A03FF | 0 027A0000 | 0 027A03FF | 1K | EDMA3CC Transfer Controller EDMA3TC2 |
| 027A0400 | 027A7FFF | 0 027A0400 | 0 027A7FFF | 31K | Reserved |
| 027A8000 | 027A83FF | 0 027A8000 | 0 027A83FF | 1K | EDMA3CC Transfer Controller EDMA3TC3 |
| 027A8400 | 027CFFFF | 0 027A8400 | 0 027CFFFF | 159K | Reserved |
| 027D0000 | 027D0FFF | 0 027D0000 | 0 027D0FFF | 4K | TI embedded trace buffer (TETB) - CorePac0 |
| 027D1000 | 027DFFFF | 0 027D1000 | 0 027DFFFF | 60K | Reserved |
| 027E0000 | 027E0FFF | 0 027E0000 | 0 027E0FFF | 4K | TI embedded trace buffer (TETB) - CorePac1 (C6657) or Reserved (C6655) |
| 027E1000 | 0284FFFF | 0 027E1000 | 0 0284FFFF | 444K | Reserved |
| 02850000 | 02857FFF | 0 02850000 | 0 02857FFF | 32K | TI embedded trace buffer (TETB) — system |
| 02858000 | 028FFFFFFF | 0 02858000 | 0 028FFFFFFF | 672K | Reserved |
| 02900000 | 02920FFF | 0 02900000 | 0 02920FFF | 132K | Serial RapidIO (SRIO) configuration |
| 02921000 | 029FFFFFFF | 0 02921000 | 0 029FFFFFFF | 1M-132K | Reserved |
| 02A00000 | 02AFFFFFFF | 0 02A00000 | 0 02AFFFFFFF | 1M | Queue manager subsystem configuration |
| 02B00000 | 02C07FFF | 0 02B00000 | 0 02C07FFF | 1056K | Reserved |
| 02C08000 | 02C8BFFF | 0 02C08000 | 0 02C8BFFF | 16K | EMAC subsystem configuration |
| 02C0C000 | 07FFFFFFF | 0 02C0C000 | 0 07FFFFFFF | 84M - 48K | Reserved |
| 08000000 | 0800FFFF | 0 08000000 | 0 0800FFFF | 64K | Extended memory controller (XMC) configuration |
| 08010000 | 0BBFFFFFFF | 0 08010000 | 0 0BBFFFFFFF | 60M-64K | Reserved |
| 0BC00000 | 0BCFFFFFFF | 0 0BC00000 | 0 0BCFFFFFFF | 1M | Multicore shared memory controller (MSMC) config |
| 0BD00000 | 0BFFFFFFF | 0 0BD00000 | 0 0BFFFFFFF | 3M | Reserved |
| 0C000000 | 0C1FFFFFFF | 0 0C000000 | 0 0C1FFFFFFF | 1M | Multicore shared memory (MSM) |
| 0C200000 | 107FFFFFFF | 0 0C200000 | 0 107FFFFFFF | 71 M | Reserved |

Table 3-2. Memory Map Summary (continued)

| LOGICAL 32-BIT ADDRESS | | PHYSICAL 36-BIT ADDRESS | | BYTES | DESCRIPTION |
|------------------------|-----------|-------------------------|-------------|----------|---|
| START | END | START | END | | |
| 10800000 | 108FFFFFF | 0 10800000 | 0 108FFFFFF | 1M | CorePac0 L2 SRAM |
| 10900000 | 10DFFFFFF | 0 10900000 | 0 10DFFFFFF | 5M | Reserved |
| 10E00000 | 10E07FFF | 0 10E00000 | 0 10E07FFF | 32K | CorePac0 L1P SRAM |
| 10E08000 | 10EFFFFFF | 0 10E08000 | 0 10EFFFFFF | 1M-32K | Reserved |
| 10F00000 | 10F07FFF | 0 10F00000 | 0 10F07FFF | 32K | CorePac0 L1D SRAM |
| 10F08000 | 117FFFFFF | 0 10F08000 | 0 117FFFFFF | 9M-32K | Reserved |
| 11800000 | 118FFFFFF | 0 11800000 | 0 118FFFFFF | 1M | CorePac1 L2 SRAM (C6657) or Reserved (C6655) |
| 11900000 | 11DFFFFFF | 0 11900000 | 0 11DFFFFFF | 5M | Reserved |
| 11E00000 | 11E07FFF | 0 11E00000 | 0 11E07FFF | 32K | CorePac1 L1P SRAM (C6657) or Reserved (C6655) |
| 11E08000 | 11EFFFFFF | 0 11E08000 | 0 11EFFFFFF | 1M-32K | Reserved |
| 11F00000 | 11F07FFF | 0 11F00000 | 0 11F07FFF | 32K | CorePac1 L1D SRAM (C6657) or Reserved (C6655) |
| 11F08000 | 1FFFFFFF | 0 11F08000 | 0 1FFFFFFF | 225M-32K | Reserved |
| 20000000 | 200FFFFFF | 0 20000000 | 0 200FFFFFF | 1M | System trace manager (STM) configuration |
| 20100000 | 207FFFFFF | 0 20100000 | 0 207FFFFFF | 7M | Reserved |
| 20800000 | 208FFFFFF | 0 20800000 | 0 208FFFFFF | 1M | TCP3d Data |
| 20900000 | 20AFFFFFF | 0 20900000 | 0 20AFFFFFF | 2M | Reserved |
| 20B00000 | 20B1FFFF | 0 20B00000 | 0 20B1FFFF | 128K | Boot ROM |
| 20B20000 | 20BEFFFF | 0 20B20000 | 0 20BEFFFF | 832K | Reserved |
| 20BF0000 | 20BF01FF | 0 20BF0000 | 0 20BF01FF | 512 | SPI |
| 20BF0400 | 20BFFFFF | 0 20BF0200 | 0 20BFFFFF | 64K -512 | Reserved |
| 20C00000 | 20C000FF | 0 20C00000 | 0 20C000FF | 256 | EMIF16 configuration |
| 20C00100 | 20FFFFFF | 0 20C00100 | 0 20FFFFFF | 4M - 256 | Reserved |
| 21000000 | 210001FF | 1 00000000 | 1 000001FF | 512 | DDR3 EMIF configuration |
| 21000200 | 213FFFFFF | 0 21000200 | 0 213FFFFFF | 4M-512 | Reserved |
| 21400000 | 214000FF | 0 21400000 | 0 214000FF | 256 | HyperLink config |
| 21400100 | 217FFFFFF | 0 21400100 | 0 217FFFFFF | 4M-256 | Reserved |
| 21800000 | 21807FFF | 0 21800000 | 0 21807FFF | 32K | PCIe config |
| 21808000 | 33FFFFFF | 0 21808000 | 0 33FFFFFF | 8M-32K | Reserved |
| 22000000 | 22000FFF | 0 22000000 | 0 22000FFF | 4K | McBSP0 FIFO Data |
| 22000100 | 223FFFFFF | 0 22000100 | 0 223FFFFFF | 4M-4K | Reserved |
| 22400000 | 22400FFF | 0 22400000 | 0 22400FFF | 4K | McBSP1 FIFO Data |
| 22400100 | 229FFFFFF | 0 22400100 | 0 229FFFFFF | 6M-4K | Reserved |
| 22A00000 | 22A0FFFF | 0 22A00000 | 0 22A0FFFF | 64K | VCP2_A |
| 22A01000 | 22AFFFFFF | 0 22A01000 | 0 22AFFFFFF | 1M-64K | Reserved |
| 22B00000 | 22B0FFFF | 0 22B00000 | 0 22B0FFFF | 64K | VCP2_B |
| 22B01000 | 33FFFFFF | 0 22B01000 | 0 33FFFFFF | 277M-64K | Reserved |
| 34000000 | 341FFFFFF | 0 34000000 | 0 341FFFFFF | 2M | Queue manager subsystem data |
| 34200000 | 3FFFFFFF | 0 34200000 | 0 3FFFFFFF | 190M | Reserved |
| 40000000 | 4FFFFFFF | 0 40000000 | 0 4FFFFFFF | 256M | HyperLink data |
| 50000000 | 5FFFFFFF | 0 50000000 | 0 5FFFFFFF | 256M | Reserved |
| 60000000 | 6FFFFFFF | 0 60000000 | 0 6FFFFFFF | 256M | PCIe data |

Table 3-2. Memory Map Summary (continued)

| LOGICAL 32-BIT ADDRESS | | PHYSICAL 36-BIT ADDRESS | | BYTES | DESCRIPTION |
|------------------------|----------|-------------------------|------------|-------|--|
| START | END | START | END | | |
| 70000000 | 73FFFFFF | 0 70000000 | 0 73FFFFFF | 64M | EMIF16 CE0 data space, supports NAND, NOR, or SRAM memory ⁽¹⁾ |
| 74000000 | 77FFFFFF | 0 74000000 | 0 77FFFFFF | 64M | EMIF16 CE1 data space, supports NAND, NOR, or SRAM memory ⁽¹⁾ |
| 78000000 | 7BFFFFFF | 0 78000000 | 0 7BFFFFFF | 64M | EMIF16 CE2 data space, supports NAND, NOR, or SRAM memory ⁽¹⁾ |
| 7C000000 | 7FFFFFFF | 0 7C000000 | 0 7FFFFFFF | 64M | EMIF16 CE3 data space, supports NAND, NOR or SRAM memory ⁽¹⁾ |
| 80000000 | FFFFFFFF | 8 00000000 | 8 7FFFFFFF | 2G | DDR3 EMIF data ⁽²⁾ |

(1) 32MB per chip select for 16-bit NOR and SRAM. 16MB per chip select for 8-bit NOR and SRAM. The 32MB and 16MB size restrictions do not apply to NAND.

(2) The memory map only shows the default MPAX configuration of DDR3 memory space. For the extended DDR3 memory space access (up to 8GB), please refer to the MPAX configuration details in *C66x CorePac User's Guide* and *Multicore Shared Memory Controller (MSMC) for KeyStone Devices User's Guide* in [Section 3.11](#).

3.4 Boot Sequence

The boot sequence is a process by which the DSP's internal memory is loaded with program and data sections. The DSP's internal registers are programmed with predetermined values. The boot sequence is started automatically after each power-on reset, warm reset, and system reset. A local reset to an individual C66x CorePac should not affect the state of the hardware boot controller on the device. For more details on the initiators of the resets, see [Section 8.4](#). The bootloader uses a section of the L2 SRAM (start address 0x008EFD00 and end address 0x008F FFFF) during initial booting of the device. For more details on the type of configurations stored in this reserved L2 section see the *Bootloader for the C66x DSP User's Guide* ([SPRUGY5](#)).

3.5 Boot Modes Supported and PLL Settings

The device supports several boot processes, which leverage the internal boot ROM. Most boot processes are software driven, using the BOOTMODE[2:0] device configuration inputs to determine the software configuration that must be completed. From a hardware perspective, there are two possible boot modes:

- **ROM Boot** - C66x CorePac0 is released from reset and begins executing from the L3 ROM base address. After performing the boot process (e.g., from I²C ROM, Ethernet, or RapidIO), C66x CorePac0 then begins execution from the provided boot entry point. For C6657 only, the other C66x CorePac is released from reset and begins executing an IDLE from the L3 ROM. It is then released from IDLE based on interrupts generated by C66x CorePac0. See the *Bootloader for the C66x DSP User's Guide* ([SPRUGY5](#)) for more details.

The boot process performed by the C66x CorePac0 in ROM boot is determined by the BOOTMODE[12:0] value in the DEVSTAT register. The C66x CorePac0 reads this value, and then executes the associated boot process in software. [Figure 3-2](#) shows the bits associated with BOOTMODE[12:0].

Figure 3-2. Boot Mode Pin Decoding

| | | | | | | | | | | | | |
|--|----|----|----------------------|---|---|---|---|---|---|-------------|---|---|
| 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PLL Mult I ² C /SPI Ext Dev Cfg | | | Device Configuration | | | | | | | Boot Device | | |

3.5.1 Boot Device Field

The Boot Device field BOOTMODE[2:0] defines the boot device that is chosen. [Table 3-3](#) shows the supported boot modes.

Table 3-3. Boot Mode Pins: Boot Device Values

| Bit | Field | Description |
|-----|-------------|--|
| 2-0 | Boot Device | Device boot mode <ul style="list-style-type: none"> • 0 = EMIF16 / UART / No Boot • 1 = Serial Rapid I/O • 2 = Ethernet (SGMII) • 3 = NAND • 4 = PCIe • 5 = I²C • 6 = SPI • 7 = HyperLink |

3.5.2 Device Configuration Field

The device configuration fields BOOTMODE[9:3] are used to configure the boot peripheral and, therefore, the bit definitions depend on the boot mode.

3.5.2.1 EMIF16 / UART / No Boot Device Configuration

Figure 3-3. EMIF16 / UART / No Boot Configuration Fields

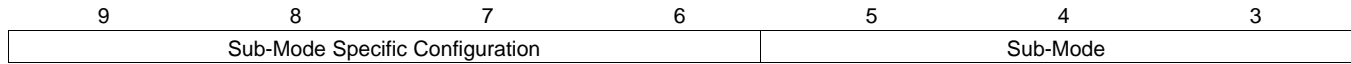


Table 3-4. EMIF16 / UART / No Boot Configuration Field Descriptions

| Bit | Field | Description |
|-----|---------------------------------|--|
| 9-6 | Sub-Mode Specific Configuration | Configures the selected sub-mode. See Section 3.5.2.1.1 , Section 3.5.2.1.2 , and Section 3.5.2.1.3 |
| 5-3 | Sub-Mode | Sub mode selection. <ul style="list-style-type: none"> • 0 = No boot • 1 = UART port 0 boot • 2 - 3 = Reserved • 4 = EMIF16 boot • 5 = UART port 1 boot • 6 - 7 = Reserved |

3.5.2.1.1 No Boot Mode

Figure 3-4. No Boot Configuration Fields



Table 3-5. No Boot Configuration Field Descriptions

| Bit | Field | Description |
|-----|----------|--|
| 9-6 | Reserved | <ul style="list-style-type: none"> Reserved |

3.5.2.1.2 UART Boot Mode

Figure 3-5. UART Boot Configuration Fields

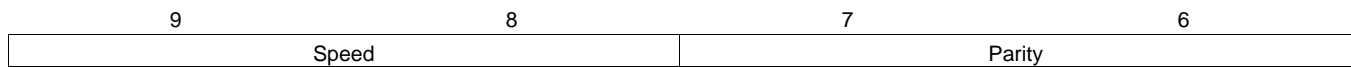


Table 3-6. UART Boot Configuration Field Descriptions

| Bit | Field | Description |
|-----|--------|--|
| 9-8 | Speed | UART interface speed. <ul style="list-style-type: none"> • 0 = 115200 baud • 1 = 38400 baud • 2 = 19200 baud • 3 = 9600 baud |
| 7-6 | Parity | UART parity used during boot. <ul style="list-style-type: none"> • 0 = None • 1 = Odd • 2 = Even • 4 = None |

3.5.2.1.3 EMIF16 Boot Mode

Figure 3-6. EMIF16 Boot Configuration Fields

| | | | |
|-------------|--------------|-------------|---|
| 9 | 8 | 7 | 6 |
| Wait Enable | Width Select | Chip Select | |

Table 3-7. EMIF16 Boot Configuration Field Descriptions

| Bit | Field | Description |
|-----|--------------|--|
| 9 | Wait Enable | Extended Wait mode for EMIF16. <ul style="list-style-type: none"> • 0 = Wait enable disabled (EMIF16 sub mode) • 1 = Wait enable enabled (EMIF16 sub mode) |
| 8 | Width Select | EMIF data width for EMIF16. <ul style="list-style-type: none"> • 0 = 8-bit wide EMIF (EMIF16 sub mode) • 1 = 16-bit wide EMIF (EMIF16 sub mode) |
| 7-6 | Chip Select | EMIF Chip Select used during EMIF 16 boot. <ul style="list-style-type: none"> • 0 = CS2 • 1 = CS3 • 2 = CS4 • 4 = CS5 |

3.5.2.2 Serial Rapid I/O Boot Device Configuration

The device ID is always set to 0xff (8-bit node IDs) or 0xffff (16 bit node IDs) at power-on reset.

Figure 3-7. Serial Rapid I/O Device Configuration Fields

| | | | | | | |
|------------|-----------|---|-----------|---|----------|---|
| 9 | 8 | 7 | 6 | 5 | 4 | 3 |
| Lane Setup | Data Rate | | Ref Clock | | Reserved | |

Table 3-8. Serial Rapid I/O Configuration Field Descriptions

| Bit | Field | Description |
|-----|------------|--|
| 9 | Lane Setup | SRIO port and lane configuration <ul style="list-style-type: none"> • 0 = Port Configured as 4 ports each 1 lane wide (4 -1x ports) • 1 = Port Configured as 2 ports 2 lanes wide (2 – 2x ports) |
| 8-7 | Data Rate | SRIO data rate configuration <ul style="list-style-type: none"> • 0 = 1.25 GBaud • 1 = 2.5 GBaud • 2 = 3.125 GBaud • 3 = 5.0 GBaud |
| 6-5 | Ref Clock | SRIO reference clock configuration <ul style="list-style-type: none"> • 0 = 156.25 MHz • 1 = 250 MHz • 2 = 312.5 MHz • 3 = Reserved |
| 4-3 | Reserved | Reserved |

In SRIO boot mode, the message mode will be enabled by default. If use of the memory reserved for received messages is required and reception of messages cannot be prevented, the master can disable the message mode by writing to the boot table and generating a boot restart.

3.5.2.3 Ethernet (SGMII) Boot Device Configuration

Figure 3-8. Ethernet (SGMII) Device Configuration Fields

| | | | | | | |
|-------------------|---|----------------|---|---|-----------|---|
| 9 | 8 | 7 | 6 | 5 | 4 | 3 |
| SerDes Clock Mult | | Ext connection | | | Device ID | |

Table 3-9. Ethernet (SGMII) Configuration Field Descriptions

| Bit | Field | Description |
|-----|-------------------|--|
| 9-8 | SerDes Clock Mult | SGMII SerDes input clock. The output frequency of the PLL must be 1.25 GBs. <ul style="list-style-type: none"> • 0 = x8 for input clock of 156.25 MHz • 1 = x5 for input clock of 250 MHz • 2 = x4 for input clock of 312.5 MHz • 3 = Reserved |
| 7-6 | Ext connection | External connection mode <ul style="list-style-type: none"> • 0 = MAC to MAC connection, master with auto negotiation • 1 = MAC to MAC connection, slave, and MAC to PHY • 2 = MAC to MAC, forced link • 3 = MAC to fiber connection |
| 5-3 | Device ID | This value can range from 0 to 7 is used in the device ID field of the Ethernet-ready frame. |

3.5.2.4 NAND Boot Device Configuration

Figure 3-9. NAND Device Configuration Fields

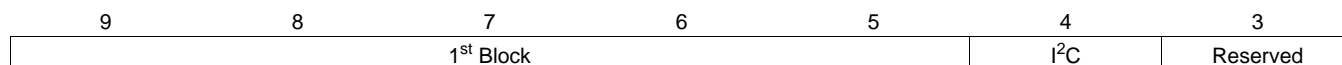


Table 3-10. NAND Configuration Field Descriptions

| Bit | Field | Description |
|-----|-----------------------|--|
| 9-5 | 1 st Block | NAND Block to be read first by the boot ROM. <ul style="list-style-type: none"> • 0 = Block 0 • ... • 31 = Block 31 |
| 4 | I ² C | NAND parameters read from I ² C EEPROM <ul style="list-style-type: none"> • 0 = Parameters are not read from I²C • 1 = Parameters are read from I²C |
| 3 | Reserved | Reserved |

3.5.2.5 PCI Boot Device Configuration

Extra device configuration is provided in the PCI bits in the DEVSTAT register.

Figure 3-10. PCI Device Configuration Fields

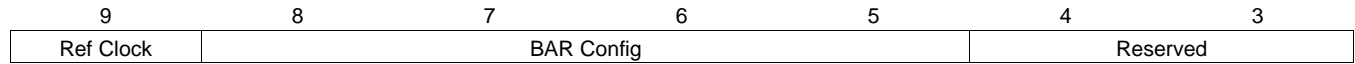


Table 3-11. PCI Device Configuration Field Descriptions

| Bit | Field | Description |
|-----|------------|---|
| 9 | Ref Clock | PCIe reference clock configuration <ul style="list-style-type: none"> • 0 = 100 MHz • 1 = 250 MHz |
| 8-5 | BAR Config | PCIe BAR registers configuration This value can range from 0 to 0xf. See Table 3-12 . |
| 4-3 | Reserved | Reserved |

Table 3-12. BAR Config / PCIe Window Sizes

| BAR CFG | BAR0 | 32-BIT ADDRESS TRANSLATION | | | | | 64-BIT ADDRESS TRANSLATION | | | | |
|---------|-----------|----------------------------|------|------|------|---------------|----------------------------|--------|--|-----|-----|
| | | BAR1 | BAR2 | BAR3 | BAR4 | BAR5 | BAR2/3 | BAR4/5 | | | |
| 0b0000 | PCIe MMRs | 32 | 32 | 32 | 32 | Clone of BAR4 | | | | | |
| 0b0001 | | 16 | 16 | 32 | 64 | | | | | | |
| 0b0010 | | 16 | 32 | 32 | 64 | | | | | | |
| 0b0011 | | 32 | 32 | 32 | 64 | | | | | | |
| 0b0100 | | 16 | 16 | 64 | 64 | | | | | | |
| 0b0101 | | 16 | 32 | 64 | 64 | | | | | | |
| 0b0110 | | 32 | 32 | 64 | 64 | | | | | | |
| 0b0111 | | 32 | 32 | 64 | 128 | | | | | | |
| 0b1000 | | 64 | 64 | 128 | 256 | | | | | | |
| 0b1001 | | 4 | 128 | 128 | 128 | | | | | | |
| 0b1010 | | 4 | 128 | 128 | 256 | | | | | | |
| 0b1011 | | 4 | 128 | 256 | 256 | | | | | | |
| 0b1100 | | | | | | | | | | 256 | 256 |
| 0b1101 | | | | | | | | | | 512 | 512 |
| 0b1110 | | | | | | 1024 | 1024 | | | | |
| 0b1111 | | | | | | 2048 | 2048 | | | | |

3.5.2.6 I²C Boot Device Configuration

3.5.2.6.1 I²C Master Mode

In master mode, the I²C device configuration uses ten bits of device configuration instead of seven as used in other boot modes. In this mode, the device will make the initial read of the I²C EEPROM while the PLL is in bypass mode. The initial read will contain the desired clock multiplier, which will be set up prior to any subsequent reads.

Figure 3-11. I²C Master Mode Device Configuration Bit Fields

| | | | | | | | | | |
|------|---------|----|-------|-----------------|---|---|---|---|---|
| 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 |
| Mode | Address | | Speed | Parameter Index | | | | | |

Table 3-13. I²C Master Mode Device Configuration Field Descriptions

| Bit | Field | Description |
|---------|-----------------|---|
| 12 | Mode | I ² C operation mode <ul style="list-style-type: none"> 0 = Master mode 1 = Passive mode (see Section 3.5.2.6.2) |
| 11 - 10 | Address | I ² C bus address configuration <ul style="list-style-type: none"> 0 = Boot from I²C EEPROM at I²C bus address 0x50 1 = Boot from I²C EEPROM at I²C bus address 0x51 2 = Boot from I²C EEPROM at I²C bus address 0x52 3 = Boot from I²C EEPROM at I²C bus address 0x53 |
| 9 | Speed | I ² C data rate configuration <ul style="list-style-type: none"> 0 = I²C slow mode. Initial data rate is SYSCLK / 5000 until PLLs and clocks are programmed 1 = I²C fast mode. Initial data rate is SYSCLK / 250 until PLLs and clocks are programmed |
| 8-3 | Parameter Index | Identifies the index of the configuration table initially read from the I2C EEPROM This value can range from 0 to 31. |

3.5.2.6.2 I²C Passive Mode

In passive mode, the device does not drive the clock, but simply acks data received on the specified address.

Figure 3-12. I²C Passive Mode Device Configuration Bit Fields

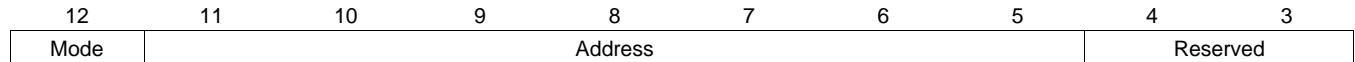


Table 3-14. I²C Passive Mode Device Configuration Field Descriptions

| Bit | Field | Description |
|--------|----------|---|
| 12 | Mode | I ² C operation mode <ul style="list-style-type: none"> • 0 = Master mode (see Section 3.5.2.6.1) • 1 = Passive mode |
| 11 - 5 | Address | I ² C bus address accepted during boot. Value may range from 0x00 to 0x7F |
| 4 - 3 | Reserved | Reserved |

3.5.2.7 SPI Boot Device Configuration

In SPI boot mode, the SPI device configuration uses ten bits of device configuration instead of seven as used in other boot modes.

Figure 3-13. SPI Device Configuration Bit Fields

| | | | | | | | | | |
|------|----|----------|------------|-------------|---|-----------------------|---|---|---|
| 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 |
| Mode | | 4, 5 Pin | Addr Width | Chip Select | | Parameter Table Index | | | |

Table 3-15. SPI Device Configuration Field Descriptions

| Bit | Field | Description |
|-------|-----------------------|---|
| 12-11 | Mode | Clk Pol / Phase <ul style="list-style-type: none"> • 0 = Data is output on the rising edge of SPICLK. Input data is latched on the falling edge. • 1 = Data is output one half-cycle before the first rising edge of SPICLK and on subsequent falling edges. Input data is latched on the rising edge of SPICLK. • 2 = Data is output on the falling edge of SPICLK. Input data is latched on the rising edge. • 3 = Data is output one half-cycle before the first falling edge of SPICLK and on subsequent rising edges. Input data is latched on the falling edge of SPICLK. |
| 10 | 4, 5 Pin | SPI operation mode configuration <ul style="list-style-type: none"> • 0 = 4-pin mode used • 1 = 5-pin mode used |
| 9 | Addr Width | SPI address width configuration <ul style="list-style-type: none"> • 0 = 16-bit address values are used • 1 = 24-bit address values are used |
| 8-7 | Chip Select | The chip select field value |
| 6-3 | Parameter Table Index | Specifies which parameter table is loaded |

3.5.2.8 HyperLink Boot Device Configuration

Figure 3-14. HyperLink Boot Device Configuration Fields

| | | | | | | |
|----------|-----------|---|-----------|---|----------|---|
| 9 | 8 | 7 | 6 | 5 | 4 | 3 |
| Reserved | Data Rate | | Ref Clock | | Reserved | |

Table 3-16. HyperLink Boot Device Configuration Field Descriptions

| Bit | Field | Description |
|-----|------------|---|
| 9 | Reserved | Reserved |
| 8-7 | Data Rate | HyperLink data rate configuration <ul style="list-style-type: none"> • 0 = 1.25 GBaud/s • 1 = 3.125 GBaud/s • 2 = 6.25 GBaud/s • 3 = Reserved |
| 6-5 | Ref Clocks | HyperLink reference clock configuration <ul style="list-style-type: none"> • 0 = 156.25 MHz • 1 = 250 MHz • 2 = 312.5 MHz • 3 = Reserved |
| 4-3 | Reserved | Reserved |

3.5.3 Boot Parameter Table

The ROM Bootloader (RBL) is guided by the boot parameter table to carry out the boot process. The boot parameter table is the most common format the RBL employs to determine the boot flow. These boot parameter tables have certain parameters common across all the boot modes, while the rest of the parameters are unique to the boot modes. The common entries in boot parameter table is shown in table below.

Table 3-17. Boot Parameter Table Common Values

| Byte Offset | Name | Description |
|-------------|-----------------|---|
| 0 | Length | The length of this table, including this length field, in bytes. |
| 2 | Checksum | Identifies the device port number to boot from, if applicable. The value 0xFFFF indicates that all ports are configured (Ethernet, SRIO). |
| 4 | Boot Mode | See Table 3-18 |
| 6 | Port Num | Identifies the device port number to boot from, if applicable. The value 0xFFFF indicates that all ports are configured (Ethernet, SRIO). |
| 8 | PLL config, MSW | PLL configuration, MSW (see Figure 5-6) |
| 10 | PLL config, LSW | PLL configuration, LSW |

Table 3-18. Boot Parameter Table Boot Mode Field

| Value | Boot Mode |
|-------|-----------------------------|
| 10 | Ethernet (boot table) |
| 20 | Rapid I/O |
| 30 | PCIe |
| 40 | I2C Master |
| 41 | I2C Slave |
| 42 | I2C Master Write |
| 50 | SPI |
| 60 | Hyperlink |
| 70 | EMIF25 |
| 80 | NAND |
| 81 | NAND I2C |
| 100 | SLEEP, no PLL configuration |
| 110 | UART |

Figure 3-15. Boot Parameter Table Boot Mode Values

| 31 | 30 | 29 | 16 | 15 | 8 | 7 | 0 |
|----------------|----|----|----------------|----|-----------------|---|------------------|
| PLL Config Ctl | | | PLL Multiplier | | PLL Pre-Divider | | PLL Post-Divider |

Table 3-19. PLL Configuration Field Description

| Field | Value | Description |
|----------------|---------|---|
| PLL Config Ctl | 0b00 | PLL is not configured |
| | 0b01 | PLL is configured only if it is currently disabled or in bypass |
| | 0b10 | PLL is configured only if it is currently disabled or in bypass |
| | 0b11 | PLL is disabled and put into bypass |
| Pre-divider | 0-255 | Input clock division. The value 0 is treated as pre-divide by 1 |
| Multiplier | 0-16383 | Multiplier. The value 0 is treated as multiply by 1 |
| Post-divider | 0-255 | PLL output division. The value 0 is treated as post divide by 1 |

3.5.3.1 Sleep / XIP Mode Parameter Table

The sleep mode parameter table has no fields in addition to the common fields described in [Section 3.5.3](#).

Table 3-20. EMIF16 XIP Parameter Table Values

| Byte Offset | Name | Descriptions |
|-------------|-------------------|---|
| 12 | Options | Figure 3-16 |
| 14 | Type | Must be set to 0 for NOR flash |
| 16 | Branch Addr, MSW | Address to branch to |
| 18 | Branch Addr, LSW | |
| 20 | CsNum | The chip select number, valid values are 2-5 |
| 22 | memWidth | The bit width of the memory, valid values are 8 or 16 |
| 24 | waitEnable | Extended wait is enabled if this value is 1, otherwise disabled |
| 26 | Async config, MSW | EMIF16 async config register value, msw |
| 28 | Async config, LSW | EMIF16 async config register value, lsw |

Figure 3-16. EMIF16 XIP Options Fields

| | | | |
|----|----------|---|-------|
| 15 | Reserved | 1 | 0 |
| | | | async |

Table 3-21. EMIF16 XIP Option Field Descriptions

| Field | Value | Description |
|-------|-------|--|
| Async | 0 | The async config register is not changed by the boot code |
| | 1 | The async config value in the boot parameter table is programmed in the async config register (EMIF timing values) |

3.5.3.2 SRIO Mode Boot Parameter Table

Table 3-22. SRIO Mode Boot Parameter Table

| Byte Offset | Name | Description |
|-------------|-----------------------|---|
| 12 | Options | See Figure 3-17 |
| 14 | Lane Setup | See Table 3-24 |
| 16 | Reserved | Reserved |
| 18 | Node ID | The node ID value to set for this device |
| 20 | SERDES ref clk | The SERDES reference clock frequency, in 1/100 MHZ. Used only if PLL setup field in options is set. |
| 22 | Link Rate | Link rate, MHz. Used only if PLL setup field in options is set. |
| 24 | PF Low | Packet forward address range, low value |
| 26 | PF high | Packet forward address range, high value |
| 28 | Promiscuous Mask | A bit is set for each lane/port that is configured as promiscuous. |
| 30 | Serdes AUX, MSW | Serdes Auxillary Register Configuration, MSW |
| 32 | Serdes AUX, LSW | Serdes Auxillary Register Configuration, LSW |
| 34 | SERDES Rx Lane 0, MSW | Serdes Rx Config, Lane 0, MSW |
| 36 | SERDES Rx Lane 0, LSW | Serdes Rx Config, Lane 0, LSW |
| 38 | SERDES Rx Lane 1, MSW | Serdes Rx Config, Lane 1, MSW |
| 40 | SERDES Rx Lane 1, LSW | Serdes Rx Config, Lane 1, LSW |
| 42 | SERDES Rx Lane 2, MSW | Serdes Rx Config, Lane 2, MSW |
| 44 | SERDES Rx Lane 2, LSW | Serdes Rx Config, Lane 2, LSW |
| 46 | SERDES Rx Lane 3, MSW | Serdes Rx Config, Lane 3, MSW |
| 48 | SERDES Rx Lane 3, LSW | Serdes Rx Config, Lane 3, LSW |

Table 3-22. SRIO Mode Boot Parameter Table (continued)

| Byte Offset | Name | Description |
|-------------|-----------------------|-------------------------------|
| 50 | SERDES Tx Lane 0, MSW | Serdes Tx Config, Lane 0, MSW |
| 52 | SERDES Tx Lane 0, LSW | Serdes Tx Config, Lane 0, LSW |
| 54 | SERDES Tx Lane 1, MSW | Serdes Tx Config, Lane 1, MSW |
| 56 | SERDES Tx Lane 1, LSW | Serdes Tx Config, Lane 1, LSW |
| 58 | SERDES Tx Lane 2, MSW | Serdes Tx Config, Lane 2, MSW |
| 60 | SERDES Tx Lane 2, LSW | Serdes Tx Config, Lane 2, LSW |
| 62 | SERDES Tx Lane 3, MSW | Serdes Tx Config, Lane 3, MSW |
| 64 | SERDES Tx Lane 3, LSW | Serdes Tx Config, Lane 3, LSW |

Figure 3-17. SRIO Boot Options

| | | | | | | |
|----------|---|--------------|--------------|---------------|---------------|-------|
| 15 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | PLL Setup | QM Bypass | Cfg Bypass | Mailbox En | Tx En |

Table 3-23. SRIO Boot Options Description

| Parameter | Value | Description |
|------------|-------|--|
| PLL Setup | 0 | Serdes Configuration registers taken without modification |
| | 1 | Multiplier and rate fields are modified based on the reference clock and link rate fields. |
| QM Bypass | 0 | Configure the QM (and cpdma) |
| | 1 | Bypass QM configuration |
| Cfg Bypass | 0 | Configure the SRIO |
| | 1 | Bypass SRIO configuration |
| Mailbox En | 0 | Mailbox mode disabled. SRIO boot is in Master mode |
| | 1 | Mailbox mode enabled. SRIO boot is in message mode (master boot still works) |
| Tx En | 0 | SRIO transmit disabled |
| | 1 | SRIO transmit enabled |

Table 3-24. SRIO Lane Setup Values

| Value | Description |
|----------|---|
| 0 | SRIO configured as four 1x ports |
| 1 | SRIO configured as 3 ports (2x, 1x, 1x) |
| 2 | SRIO configured as 3 ports (1x, 1x, 2x) |
| 3 | SRIO configured as 2 ports (2x, 2x) |
| 4 | SRIO configured as 1 4x port |
| 5-0xFFFF | Reserved |

3.5.3.3 2.5.3.3 Ethernet Mode Boot Parameter Table

The default multi-cast Ethernet mac address is the broadcast address.

Table 3-25. Ethernet Boot Parameter Table Values

| Byte Offset | Name | Description |
|-------------|----------------|--|
| 12 | Options | See Figure 3-18 |
| 14 | MAC High | The 16 MSBs of the MAC address to receive during boot |
| 16 | MAC Med | The 16 middle bits of the MAC address to receive during boot |
| 18 | MAC Low | The 16 LSBs of the MAC address to receive during boot |
| 20 | Multi MAC High | The 16 MSBs of the multi-cast MAC address to receive during boot |

Table 3-25. Ethernet Boot Parameter Table Values (continued)

| Byte Offset | Name | Description |
|-------------|------------------------|---|
| 22 | Multi MAC Med | The 16 middle bits of the multi-cast MAC address to receive during boot |
| 24 | Mulit MAC Low | The 16 LSBs of the multi-cast MAC address to receive during boot |
| 26 | Source Port | The source UDP port to accept boot packets from. A value of 0 will accept packets from any UDP port |
| 28 | Dest Port | The destination port to accept boot packets on. |
| 30 | Device ID 12 | The 1st two bytes of the device ID. This is typically a string value, and is sent in the Ethernet ready frame |
| 32 | Device ID 34 | The 2nd two bytes of the device ID. |
| 34 | Dest MAC High | The 16 MSBs of the MAC destination address used for the Ethernet ready frame. Default is broadcast. |
| 36 | Dest MAC Med | The 16 middle bits of the MAC destination address |
| 38 | DEST MAC Low | The 16 LSBs of the MAC destination address |
| 40 | Sgmii Config | See Figure 3-19 |
| 42 | Sgmii Control | The SGMII control register value (if table value not used) |
| 44 | Sgmii Adv Ability | The SGMII ADV Ability register value (if table value not used) |
| 46 | Sgmii Tx Cfg High | The 16 MSBs of the sgmii Tx config register (if table value not used) |
| 48 | Sgmii Tx Cfg Low | The 16 LSBs of the sgmii Tx config register (if table value not used) |
| 50 | Sgmii Rx Cfg High | The 16 MSBs of the sgmii Rx config register (if table value not used) |
| 52 | Sgmii Rx Cfg Low | The 16 LSBs of the sgmii Rx config register (if table value not used) |
| 54 | Sgmii Aux Cfg High | The 16 MSBs of the sgmii Aux config register (if table value not used) |
| 56 | Sgmii Aux Cfg Low | The 16 LSBs of the sgmii Aux config register (if table value not used) |
| 58 | Pkt PLL Config, MSW | The packet subsystem PLL configuration, MSW (unused in gauss) |
| 60 | Packet PLL Config, LSW | The packet subsystem PLL configuration, LSW |

Figure 3-18. Ethernet Mode Boot Parameter Options Field

| | | | | | | |
|----------|---|---|-------------|---------|----------|---|
| 15 | 7 | 6 | 5 | 4 | 3 | 0 |
| Reserved | | | Init Config | Skip Tx | Reserved | |

Table 3-26. Ethernet Options Field Descriptions

| Name | Value | Description |
|-------------|-------|--|
| Init Config | 0b00 | SERDES and SGMII are configured. |
| | 0b01 | SERDES and SGMII are NOT configured |
| | 0b10 | Reserved |
| | 0b11 | None of the Ethernet system hardware is configured. |
| Skip tx | 0 | Ethernet ready frame is sent once when the system is first ready to receive packets, and then roughly every 3 seconds until the first boot packet is accepted. |
| | 1 | Ethernet ready frame is not sent |

Figure 3-19. SGMII Config Bit Field

| | | | | | |
|----------|---|---|--------|--------|-------|
| 15 | 6 | 5 | 4 | 3 | 0 |
| Reserved | | | bypass | direct | Index |

Table 3-27. SGMII Config Field Descriptions

| Field | Value | Description |
|--------|-------|---|
| Index | 0 | Configure the SGMII as a master |
| | 1 | Configure the SGMII as a slave, or connected to a Phy |
| | 2 | Configure the SGMII as a forced link |
| | 3 | Configure the SGMII as mac to fiber |
| | 4-15 | Reserved |
| Direct | 0 | Configure the SGMII as directed in the index field |
| | 1 | Configure the SGMII using the advise ability and control fields in the boot parameter table, not based on the index field |
| Bypass | 0 | Configure the SGMII. |
| | 1 | Do not configure the SGMII. |

3.5.3.4 NAND Mode Boot Parameter Table**Table 3-28. NAND Mode Boot Parameter Table**

| Byte Offset | Name | Description |
|-------------|---------------|---|
| 12 | Options | See Figure 3-20 |
| 14 | I2cClkFreqKhz | The I2C clock frequency to use when using I2C tables |
| 16 | I2cTargetAddr | The I2C bus address of the EEPROM |
| 18 | I2cLocalAddr | The I2C bus address of the Appleton device |
| 20 | I2cDataAddr | The address on the EEPROM of the NAND configuration table |
| 22 | I2cWtoRDelay | Delay between address writes and data reads, in I2C clock periods |
| 24 | csNum | The NAND chip select region (0-3) |
| 26 | firstBlock | The first block of the boot image |

Figure 3-20. NAND Boot Parameter Option Fields

| | | | |
|----|----------|---|-----|
| 15 | Reserved | 1 | 0 |
| | | | I2C |

Table 3-29. NAND Boot Parameter Options Bit Field Descriptions

| Name | Value | Description |
|------|-------|---|
| I2C | 0 | NAND configuration is NOT read from I2C |
| | 1 | NAND configuration is read from the I2C |

3.5.3.5 PCIe Mode Boot Parameter Table

Table 3-30. PCIe Mode Boot Parameter Table

| Byte Offset | Name | Description |
|-------------|------------------------|--|
| 12 | options | PCI configuration options (see Figure 3-21) |
| 14 | Address Width | PCI address width, can be 32 or 64 |
| 16 | Serdes Frequency | Serdes frequency, in MBs. Currently only 2500 supported. |
| 18 | Reference clock | Reference clock frequency, in units of 10kHz. Valid values are 10000 (100MHz), 12500 (125MHz), 15625 (156.25MHz), 25000 (250MHz) and 31250 (312.5 MHz), although other values should work. |
| 20 | Window 1 Size | Window 1 size, in Mbytes |
| 22 | Window 2 Size | Window 2 size, in Mbytes |
| 24 | Window 3 Size | Window 3 size, in Mbytes. Valid only if address width is 32. |
| 26 | Window 4 Size | Window 4 Size, in Mbytes Valid only if address width is 32. |
| 28 | Window 5 Size | Window 5 Size. Valid only if the address width is 32. |
| 30 | Vendor ID | Vendor ID field |
| 32 | Device ID | Device ID field (0xb006 by default for Gauss) |
| 34 | Class code Rev Id, MSW | Class code/revision ID field |
| 36 | Class code Rev Id, LSW | Class code/revision ID field |
| 38 | Serdes cfg msw | PCIe serdes config word, MSW |
| 40 | Serdes cfg lsw | PCIe serdes config word, LSW |
| 42 | Serdes lane 0 cfg msw | Serdes lane config word, msw lane 0 |
| 44 | Serdes lane 0 cfg lsw | Serdes lane config word, lsw, lane 0 |
| 46 | Serdes lane 1 cfg msw | Serdes lane config word, msw, lane 1 |
| 48 | Serdes lane 1 cfg lsw | Serdes lane config word, lsw, lane 1 |

Figure 3-21. PCIe Options Bit Field



Table 3-31. PCIe Options Field Descriptions

| Field | Value | Description |
|-------------|-------|---|
| Cfg disable | 0 | PCIe peripheral is configured by the boot rom |
| | 1 | PCIe peripheral is not configured by the boot rom |
| Serdes Cfg | 0 | Serdes PLL multiplier and rate fields in the table are used directly |
| | 1 | Serdes PLL multiplier and rate fields in the serdes registers will be overwritten based on the values in the serdes frequency and reference clock parameters. |

3.5.3.6 I2C Mode Boot Parameter Table

Table 3-32. I2C Mode Boot Parameter Table

| Byte Offset | Name | Description |
|-------------|-------------------|--|
| 12 | Options | See Figure 3-22 |
| 14 | Boot Dev Addr | The I2C device address to boot from |
| 16 | Boot Dev Addr Ext | Extended boot device address, or I2C bus address (typically 0x50, 0x51) |
| 18 | Broadcast Addr | In master broadcast boot, this is the I2C address to send the boot data to |
| 20 | Local Address | The I2C address of this device. |
| 22 | Device Freq | The operating frequency of the device (MHz). Used to compute the divide down to the I2C module |
| 24 | Bus Frequency | The desired I2C data rate (kHz). |
| 26 | Next Dev Addr | The next device to boot from (used in boot config mode) |
| 28 | Next Dev Addr Ext | The extended next boot device address |
| 30 | Address Delay | The number of CPU cycles to delay between writing the address to an I2C eeprom and reading data. This allows the I2C eeprom time to load the data. |

Figure 3-22. I2C Mode Boot Options Bitfield

| | | | | |
|----|----------|---|---|------|
| 15 | Reserved | 2 | 1 | 0 |
| | | | | Mode |

Table 3-33. Register Description

| Parameter | Value | Description |
|-----------|-------|---|
| Mode | 0 | Load a boot parameter table from the I2C |
| | 1 | Load boot records from the I2C (boot tables) |
| | 2 | Load boot config records from the I2C (boot config tables) |
| | 3 | Perform a slave mode boot, listening on the local address specified in the table. |

3.5.3.7 SPI Mode Boot Parameter Table

Table 3-34. 2.5.3.7 SPI Mode Boot Parameter Table

| Byte Offset | Name | Description |
|-------------|---------------|---|
| 12 | options | See Figure 3-23 |
| 14 | Address Width | The number of bytes in the SPI device address. Can be 2 or 3 (16 or 24 bit) |
| 16 | NPin | The operational mode, 4 or 5 pin |
| 18 | Chipsel | The chip select used. Can be 0-3. |
| 20 | Mode | SPI mode, 0-3 |
| 22 | C2T Delay | SPI chip select active to transmit start delay value (0-255) |
| 24 | CPU Freq MHz | The speed of the CPU, in MHz |
| 26 | Bus Freq, MHz | The MHz portion of the SPI bus frequency. Default = 5MHz |
| 28 | Bus Freq, kHz | The kHz portion of the SPI bus frequency. Default = 0 |
| 30 | Read Addr MSW | The first address to read from, MSW (valid for 24 bit address width only) |
| 32 | Read Addr LSW | The first address to read from, LSW |
| 34 | Next chipsel | Chipsel value used after boot config table processing is complete |
| 36 | Next read MSW | The next read address, MSW after config table processing is complete |
| 38 | Next read LSW | The next read address, LSW after config table processing is complete |

The bus frequency programmed into the SPI by the boot ROM is from the table: MHz.kHz. So for a 5.1 MHz bus frequency the MHz value is 5, the kHz value is 100.

Figure 3-23. SPI Options Field Bit Map

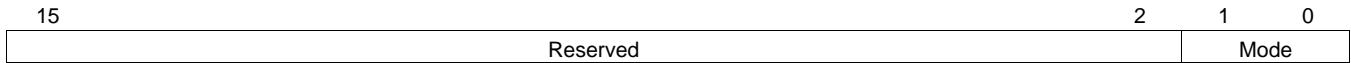


Table 3-35. SPI Options Field Description

| Parameter | Value | Description |
|-----------|-------|--|
| Mode | 0 | Load a boot parameter table from the SPI |
| | 1 | Load boot records from the SPI (boot tables) |
| | 2 | Load boot config records from the SPI (boot config tables) |
| | 3 | Reserved |

3.5.3.8 Hyperlink Mode Boot Parameter Table

Table 3-36. Hyperlink Mode Boot Parameter Table

| Byte Offset | Name | Description |
|-------------|-----------------|---------------------------------------|
| 12 | Options | See Figure 3-24 |
| 14 | N lanes | The number of lanes to configure |
| 16 | Serdes Aux, MSW | SERDES Aux register config value, MSW |
| 18 | Serdes Aux, LSW | SERDES Aux register config value, LSW |
| 20 | Rx Lane 0, MSW | SERDES Rx Lane 0 register value, MSW |
| 22 | Rx Lane 0, LSW | SERDES Rx Lane 0 register value, LSW |
| 24 | Tx Lane 0, MSW | SERDES Tx Lane 0 register value, MSW |
| 26 | Tx Lane 0, LSW | SERDES Tx Lane 0 register value, LSW |
| 28 | Rx Lane 1, MSW | SERDES Rx Lane 1 register value, MSW |
| 30 | Rx Lane 1, LSW | SERDES Rx Lane 1 register value, LSW |
| 32 | Tx Lane 1, MSW | SERDES Tx Lane 1 register value, MSW |
| 34 | Tx Lane 1, LSW | SERDES Tx Lane 1 register value, LSW |
| 36 | Rx Lane 2, MSW | SERDES Rx Lane 2 register value, MSW |
| 38 | Rx Lane 2, LSW | SERDES Rx Lane 2 register value, LSW |
| 40 | Tx Lane 2, MSW | SERDES Tx Lane 2 register value, MSW |
| 42 | Tx Lane 2, LSW | SERDES Tx Lane 2 register value, LSW |

Figure 3-24. Hyperlink Options Bit Field

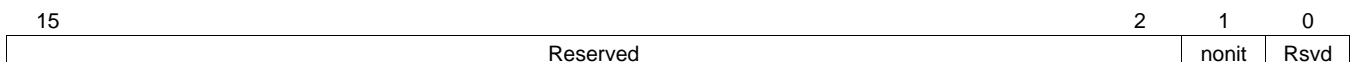


Table 3-37. Hyperlink Options Field Descriptions

| Field | Value | Description |
|-------|-------|--|
| nonit | 0 | Initialize hyperlink peripheral |
| | 1 | Do not initialize hyperlink peripheral |

3.5.3.9 UART Mode Boot Parameter Table

Table 3-38. UART Mode Boot Parameter Table

| Byte Offset | Field | Description |
|-------------|------------------|---|
| 12 | Rsvd | Reserved |
| 14 | Data Format | Only value 1, boot table format is supported |
| 16 | Protocol | Only value 0, XMODEM is supported |
| 18 | Initial Ping Cnt | Number of initial pings without reply before the boot times out |
| 20 | Max Err Count | Number of consecutive errors before the boot fails |
| 22 | Nack timeout | Timeout period waiting for an ack/nack, in milli-seconds |
| 24 | Char timeout | Timeout period between characters |
| 26 | Data bits | Number of data bits. Only the value 8 is supported |
| 28 | Parity | 0 = none, 1 = odd, 2 = even |
| 30 | Stop bits x2 | Number of stop bits x2, (2 = 1 stop bit, 4 = 2 stop bits) |
| 32 | Oversample | The over-sample factor. Only 13 and 16 are valid |
| 34 | Flow Control | Only 0, no flow control is supported. |
| 36 | Data Rate, MSW | The Baud rate, MSW |
| 38 | Data Rate, LSW | The Baud rate, LSW |
| 40 | timerRefMhz | Timer reference frequency, in MHz. In Gauss this is the frequency the device is operating at after the PLL is programmed. |

3.6 PLL Boot Configuration Settings

The PLL default settings are determined by the BOOTMODE[12:10] bits. The following table shows settings for various input clock frequencies.

Table 3-39. C66x DSP System PLL Configuration⁽¹⁾

| BOOTMODE [12:10] | INPUT CLOCK FREQ (MHz) | 850 MHz DEVICE | | | 1000 MHz DEVICE | | | 1250 MHz DEVICE | | |
|---------------------|---------------------------|----------------|------|---------|-----------------|------|---------|-----------------|------|----------|
| | | PLLD | PLLM | DSP f | PLLD | PLLM | DSP f | PLLD | PLLM | DSP f |
| 0b000 | 50.00 | 0 | 33 | 850 | 0 | 39 | 1000 | 0 | 49 | 1250 |
| 0b001 | 66.67 | 1 | 50 | 850.04 | 0 | 29 | 1000.05 | 1 | 74 | 1250.063 |
| 0b010 | 80.00 | 3 | 84 | 850 | 0 | 24 | 1000 | 3 | 124 | 1250 |
| 0b011 | 100.00 | 0 | 16 | 850 | 0 | 19 | 1000 | 0 | 24 | 1250 |
| 0b100 | 156.25 | 49 | 543 | 850 | 4 | 63 | 1000 | 0 | 15 | 1250 |
| 0b101 | 250.00 | 4 | 33 | 850 | 0 | 7 | 1000 | 0 | 9 | 1250 |
| 0b110 | 312.50 | 49 | 271 | 850 | 4 | 31 | 1000 | 0 | 7 | 1250 |
| 0b111 | 122.88 | 5 | 82 | 849.92 | 28 | 471 | 999.989 | 28 | 589 | 1249.986 |

(1) The PLL boot configuration table above may not include all the frequency values that the device supports.

OUTPUT_DIVIDE is the value of the field of SECCTL[22:19]. This will set the PLL to the maximum clock setting for the device (with OUTPUT_DIVIDE=2, by default).

- $CLK = CLKIN \times ((PLLM+1) \div ((OUTPUT_DIVIDE+1) \times (PLLD+1)))$

The Main PLL is controlled using a PLL controller and a chip-level MMR. The DDR3 PLL is controlled by chip level MMRs. For details on how to set up the PLL see [Section 8.5](#). For details on the operation of the PLL controller module, see the *Phase Locked Loop (PLL) for KeyStone Devices User's Guide (SPRUGV2)*.

3.7 Second-Level Bootloaders

Any of the boot modes can be used to download a second-level bootloader. A second-level bootloader allows for any level of customization to current boot methods as well as the definition of a completely customized boot.

3.8 Terminals

3.8.1 Package Terminals

Figure 3-25 shows the C665x CZH and GZH ball grid area (BGA) packages (bottom view).

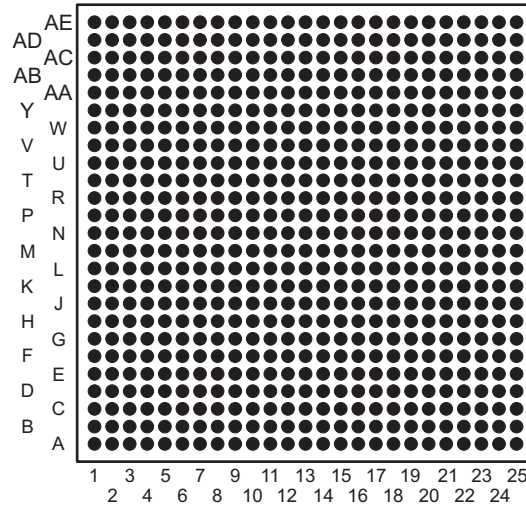


Figure 3-25. CZH/GZH 625-Pin BGA Package (Bottom View)

3.8.2 Pin Map

Figure 3-27 through Figure 3-30 show the C665x pin assignments in four quadrants (A, B, C, and D).

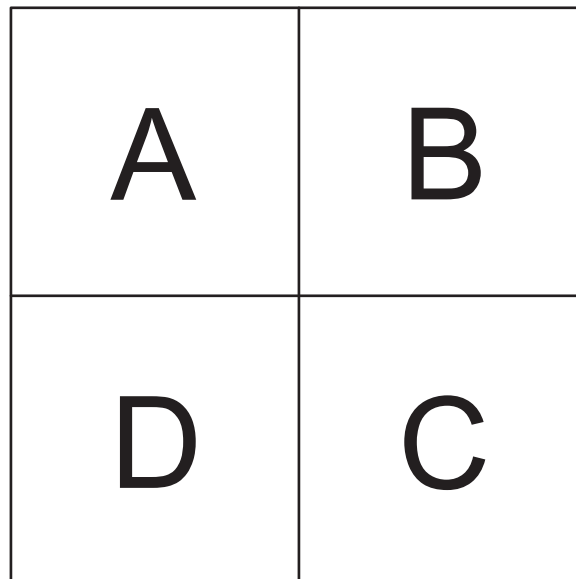


Figure 3-26. Pin Map Quadrants (Bottom View)

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|-----------|---------|------------|------------|---------|------------|---------|---------|---------|---------|----------|----------|----------|----------------|
| AE | VSS | SGMII0 RXN | SGMII0 RXP | VSS | RIORXN2 | RIORXP2 | VSS | RIORXP0 | RIORXN0 | VSS | PCIERXP0 | PCIERXN0 | VSS |
| AD | VSS | VSS | VSS | RIORXN3 | RIORXP3 | VSS | RIORXP1 | RIORXN1 | VSS | PCIERXN1 | PCIERXP1 | VSS | SRIOSGMII CLKP |
| AC | VSS | SGMII0 TXN | SGMII0 TXP | VSS | RIOTXN2 | RIOTXP2 | VSS | RIOTXP0 | RIOTXN0 | VSS | PCIETXP0 | PCIETXN0 | VSS |
| AB | EMIFD14 | VSS | RSV19 | RIOTXN3 | RIOTXP3 | VSS | RIOTXN1 | RIOTXP1 | VSS | PCIETXP1 | PCIETXN1 | VSS | SPIDOUT |
| AA | EMIFD13 | EMIFD15 | VDDR3 | VSS | VDDR4 | VSS | RSV17 | VSS | VDDR2 | VSS | RSV18 | SPISCS0 | SPICKL |
| Y | EMIFD09 | EMIFD11 | DVDD18 | RSV13 | RSV12 | VSS | VDDT2 | VSS | VDDT2 | VSS | VDDT2 | VSS | DVDD18 |
| W | EMIFD06 | EMIFD08 | VSS | EMIFD10 | EMIFD12 | DVDD18 | VSS | VDDT2 | VSS | VDDT2 | VSS | VDDT2 | VSS |
| V | EMIFD02 | EMIFD03 | EMIFD04 | EMIFD05 | EMIFD07 | VSS | DVDD18 | VSS | CVDD | VSS | CVDD | VSS | CVDD |
| U | EMIFA21 | EMIFA22 | EMIFA23 | EMIFD00 | EMIFD01 | DVDD18 | VSS | CVDD1 | VSS | CVDD | VSS | CVDD | VSS |
| T | EMIFA19 | VSS | DVDD18 | EMIFA18 | EMIFA20 | VSS | DVDD18 | VSS | CVDD1 | VSS | CVDD | VSS | CVDD |
| R | EMIFA17 | EMIFA16 | EMIFA14 | EMIFA15 | EMIFA13 | DVDD18 | VSS | VSS | VSS | CVDD | VSS | CVDD | VSS |
| P | EMIFA12 | EMIFA11 | EMIFA09 | EMIFA05 | EMIFA03 | VSS | DVDD18 | VSS | CVDD | VSS | CVDD | VSS | CVDD |
| N | EMIFA10 | EMIFA08 | DVDD18 | VSS | EMIF WAIT0 | DVDD18 | VSS | CVDD | VSS | CVDD | VSS | CVDD | VSS |

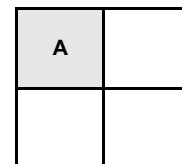


Figure 3-27. Upper Left Quadrant — A (Bottom View)

| | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | |
|--|-------------------|----------|----------|---------|-------------------------|--------------------------|--------|---------|---------|--------|---------|---------|-----------|
| | SRIOSGMII CLKN | PCIECLKN | UARTCTS1 | TDI | TMS | CORECLKN | TIMO1 | TIM1 | DX1 | FSX1 | CLKX1 | VSS | AE |
| | PCIECLKP | UARTRTS1 | VSS | TCK | CORECLKP | TDO | TIM0 | DR1 | FSR1 | CLKR1 | FSR0 | EMU16 | AD |
| | UARTRXD1 | UARTTXD1 | DVDD18 | UARTCTS | RSV04 | TIM00 | DVDD18 | CLKS1 | DX0 | CLKS0 | EMU17 | EMU13 | AC |
| | SPIDIN | UARTRXD | MDIO | UARTRTS | RSV05 | $\overline{\text{TRST}}$ | VSS | DR0 | EMU15 | DVDD18 | VSS | EMU12 | AB |
| | SPISCS1 | UARTTXD | MDCLK | SCL | SDA | SYSCLK OUT | FSX0 | CLKR0 | RSV01 | EMU14 | EMU10 | EMU11 | AA |
| | VSS | AVDDA1 | VSS | DVDD18 | $\overline{\text{POR}}$ | RSV08 | CLKX0 | EMU18 | EMU09 | EMU07 | EMU06 | EMU05 | Y |
| | DVDD18 | VSS | DVDD18 | VSS | DVDD18 | VSS | DVDD18 | GPIO14 | EMU08 | EMU03 | EMU04 | EMU02 | W |
| | VSS | CVDD | VSS | CVDD | VSS | DVDD18 | VSS | GPIO15 | GPIO13 | GPIO10 | EMU00 | EMU01 | V |
| | CVDD | VSS | CVDD | VSS | CVDD1 | VSS | DVDD18 | GPIO11 | GPIO08 | GPIO09 | GPIO05 | GPIO03 | U |
| | VSS | CVDD | VSS | CVDD1 | VSS | DVDD18 | VSS | GPIO12 | GPIO06 | GPIO04 | DVDD18 | GPIO00 | T |
| | CVDD | VSS | CVDD | VSS | CVDD | VSS | DVDD18 | GPIO07 | VSS | GPIO02 | VSS | GPIO01 | R |
| | VSS | CVDD | VSS | CVDD | VSS | CVDD | VSS | VSS | MCMTXN0 | VSS | MCMRXN0 | VSS | P |
| | CVDD | VSS | CVDD | VSS | CVDD | VSS | VDDT1 | MCMTXN1 | MCMTXP0 | VSS | MCMRXP0 | MCMRXP1 | N |

| | |
|--|----------|
| | B |
| | |

Figure 3-28. Upper Right Quadrant—B (Bottom View)

| | |
|--|----------|
| | |
| | C |

| | | | | | | | | | | | | |
|--------------|-----------|-----------|-----------|-----------|-----------|-----------|--------------|-------------|-----------|-------------|----------------|----------|
| VSS | CVDD | VSS | CVDD | VSS | VDDT1 | VDDR1 | MCM TXP1 | VSS | VSS | VSS | MCMRXN1 | M |
| CVDD | VSS | CVDD | VSS | CVDD | VSS | VDDT1 | VSS | MCMTXP2 | VSS | MCMRX3 | VSS | L |
| VSS | CVDD | VSS | CVDD1 | VSS | VDDT1 | VSS | MCMTXP3 | MCMTXN2 | VSS | MCMRXN3 | MCMRX2 | K |
| CVDD | VSS | CVDD | VSS | CVDD1 | VSS | RSV16 | MCMTXN3 | VSS | VSS | VSS | MCMRXN2 | J |
| VSS | CVDD | VSS | CVDD | VSS | DVDD18 | VSS | VSS | RSV11 | VSS | DVDD18 | VSS | H |
| DVDD15 | VSS | DVDD15 | VSS | DVDD15 | RSV0A | RSV0B | RSV15 | RSV10 | VCNTL3 | MCMTX PMDAT | MCMREF CLKOUTP | G |
| VSS | PTV15 | VSS | DVDD15 | VSS | DVDD15 | AVDDA2 | RSV14 | RSV20 | VCNTL2 | MCMTX PMCLK | MCMREF CLKOUTN | F |
| DDRODT0 | DDRA03 | DDRA02 | DDRA15 | DDRA14 | DDRA10 | DDRA09 | DVDD18 | VCNTL0 | VCNTL1 | MCMRX PMCLK | MCMTX FLCLK | E |
| DDRCAS | DVDD15 | DDRA00 | DDRBA1 | DDRA12 | DVDD15 | DDRA08 | VSS | DDRSL RATE1 | RSV21 | MCMRX PMDAT | MCMTX FLDAT | D |
| DDRCE1 | VSS | DDRA06 | DVDD15 | DDRBA0 | VSS | DDRA13 | DVDD15 | DDRSL RATE0 | RSV09 | MCMRX FLDAT | MCMCLKP | C |
| DDRCLK OUTN0 | DDRCE0 | DDRRESET | VSS | DDRA04 | DDRBA2 | DDRA11 | DDRCLK OUTN1 | DDRCLKN | RSV06 | MCMRX FLCLK | MCMCLKN | B |
| DDRCLK OUTP0 | DDRRAS | DDRCKE0 | DDRA05 | DDRA07 | DDRA01 | DDRCKE1 | DDRCLK OUTP1 | DDRCLKP | RSV07 | DVDD18 | VSS | A |
| 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | |

Figure 3-29. Lower Right Quadrant—C (Bottom View)

| | |
|---|--|
| | |
| D | |

| | | | | | | | | | | | | | |
|---|-------------|---------|---------------|-----------|------------|--------|--------|--------|--------|--------|---------|----------|--------|
| M | EMIFA07 | EMIFA06 | EMIFA01 | EMIFWAIT1 | EMIFCE3 | VSS | DVDD18 | VSS | CVDD | VSS | CVDD | VSS | CVDD |
| L | EMIFA04 | EMIFA02 | EMIFBE1 | EMIFOE | EMIFRNW | DVDD18 | VSS | CVDD | VSS | CVDD | VSS | CVDD | VSS |
| K | EMIFA00 | VSS | DVDD18 | EMIFWE | EMIFCE0 | VSS | DVDD18 | VSS | CVDD1 | VSS | CVDD | VSS | CVDD |
| J | EMIFBE0 | EMIFCE2 | RSV02 | RESETFULL | CORESEL0 | DVDD18 | VSS | CVDD1 | VSS | CVDD | VSS | CVDD | VSS |
| H | NMI | RSV03 | BOOT COMPLETE | RESET | RESET STAT | VSS | DVDD18 | VSS | CVDD | VSS | CVDD | VSS | CVDD |
| G | EMIFCE1 | HOUT | DVDD18 | LRESET | CORESEL1 | DVDD18 | VSS | DVDD15 | VSS | DVDD15 | VSS | DVDD15 | VSS |
| F | LRESET NMEN | DDR25 | VSS | DDR18 | DDRQ2 | VSS | DVDD15 | VSS | DVDD15 | VSS | DVDD15 | VSS | DVDD15 |
| E | DDRQ3 | DDR24 | DDR31 | DDR19 | DDR16 | DDR08 | DDRQ1 | DDR09 | DDR04 | DDR05 | VSS | VREFSSTL | DDRWE |
| D | DDR28 | DVDD15 | DDR29 | DVDD15 | DDR23 | DDR12 | DDR14 | DVDD15 | DDR02 | DDRQ5P | DDRCB00 | DDR0T1 | DVDD15 |
| C | DDR27 | VSS | DDR30 | VSS | DDR22 | DVDD15 | DDR13 | VSS | DDR01 | DDRQ5N | DDRCB02 | DDRQ8 | VSS |
| B | DDR26 | DDRQ3N | DDR17 | DDRQ2P | DDR21 | VSS | DDRQ1P | DDR15 | DDR03 | DVDD15 | DDR07 | DDRCB01 | DDRQ8P |
| A | VSS | DDRQ3P | DDR20 | DDRQ2N | DDR11 | DDR10 | DDRQ1N | DDRQ0 | DDR00 | VSS | DDR06 | DDRCB03 | DDRQ8N |
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |

Figure 3-30. Lower Left Quadrant—D (Bottom View)

3.9 Terminal Functions

The terminal functions table ([Table 3-41](#)) identifies the external signal names, the associated pin (ball) numbers, the pin type (I, O/Z, or I/O/Z), whether the pin has any internal pullup/pulldown resistors, and gives functional pin descriptions. This table is arranged by function. The power terminal functions table ([Table 3-42](#)) lists the various power supply pins and ground pins and gives functional pin descriptions. [Table 3-43](#) shows all pins arranged by signal name. [Table 3-44](#) shows all pins arranged by ball number.

There are 73 pins that have a secondary function as well as a primary function. The secondary function is indicated with a dagger (†). There is one pin that has a tertiary function as well as primary and secondary functions. The tertiary function is indicated with a double dagger (‡).

For more detailed information on device configuration, peripheral selection, multiplexed/shared pins, and pullup/pulldown resistors, see [Section 4.4](#).

Use the symbol definitions in [Table 3-40](#) when reading [Table 3-41](#).

Table 3-40. I/O Functional Symbol Definitions

| FUNCTIONAL SYMBOL | DEFINITION | Table 3-41 COLUMN HEADING |
|-------------------|--|--|
| IPD or IPU | Internal 100- μ A pulldown or pullup is provided for this terminal. In most systems, a 1-k Ω resistor can be used to oppose the IPD/IPU. For more detailed information on pulldown/pullup resistors and situations in which external pulldown/pullup resistors are required, see <i>Hardware Design Guide for KeyStone Devices</i> (SPRABI2). | IPD/IPU |
| A | Analog signal | Type |
| GND | Ground | Type |
| I | Input terminal | Type |
| O | Output terminal | Type |
| S | Supply voltage | Type |
| Z | Three-state terminal or high impedance | Type |

Table 3-41. Terminal Functions — Signals and Control by Function

| SIGNAL NAME | BALL NO. | TYPE | IPD/IPU | DESCRIPTION |
|--------------------------------|----------|------|---------|---|
| Boot Configuration Pins | | | | |
| LENDIAN † | T25 | IOZ | UP | Endian configuration pin (Pin shared with GPIO[0]) |
| BOOTMODE00 † | R25 | IOZ | Down | See Section 3.5 for more details (Pins shared with GPIO[1:13]) |
| BOOTMODE01 † | R23 | IOZ | Down | |
| BOOTMODE02 † | U25 | IOZ | Down | |
| BOOTMODE03 † | T23 | IOZ | Down | |
| BOOTMODE04 † | U24 | IOZ | Down | |
| BOOTMODE05 † | T22 | IOZ | Down | |
| BOOTMODE06 † | R21 | IOZ | Down | |
| BOOTMODE07 † | U22 | IOZ | Down | |
| BOOTMODE08 † | U23 | IOZ | Down | |
| BOOTMODE09 † | V23 | IOZ | Down | |
| BOOTMODE10 † | U21 | IOZ | Down | |
| BOOTMODE11 † | T21 | IOZ | Down | |
| BOOTMODE12 † | V22 | IOZ | Down | |
| PCIESSMODE0 † | W21 | IOZ | Down | PCIe Mode selection pins (Pins shared with GPIO[14:15]) |
| PCIESSMODE1 † | V21 | IOZ | Down | |
| PCIESSSEN ‡ | AD20 | I | Down | PCIe module enable (Pin shared with TIMI0 and GPIO16) |
| Clock / Reset | | | | |
| CORECLKP | AD18 | I | | Core Clock Input to main PLL. |
| CORECLKN | AE19 | I | | |
| SRIOSGMIICLKP | AD13 | I | | RapidIO/SGMII Reference Clock to drive the RapidIO and SGMII SerDes |
| SRIOSGMIICLKN | AE14 | I | | |
| DDRCLKP | A22 | I | | DDR Reference Clock Input to DDR PLL |
| DDRCLKN | B22 | I | | |
| PCIECLKP | AD14 | I | | PCIe Clock Input to drive PCIe SerDes |
| PCIECLKN | AE15 | I | | |
| MCMCLKP | C25 | I | | HyperLink Reference Clock to drive the HyperLink SerDes |
| MCMCLKN | B25 | I | | |
| AVDDA1 | Y15 | P | | SYS_CLK PLL Power Supply Pin |
| AVDDA2 | F20 | P | | DDR_CLK PLL Power Supply Pin |
| SYSCLKOUT | AA19 | OZ | Down | System Clock Output to be used as a general purpose output clock for debug purposes |
| HOUT | G2 | OZ | UP | Interrupt output pulse created by IPCGRH |
| NMI | H1 | I | UP | Non-maskable Interrupt |
| LRESET | G4 | I | UP | Warm Reset |
| LRESETNMIEN | F1 | I | UP | Enable for core selects |
| CORESEL0 | J5 | I | Down | Select for the target core for LRESET and NMI. For more details see Table 8-42 |
| CORESEL1 | G5 | I | Down | |
| RESETFULL | J4 | I | UP | Full Reset |
| RESET | H4 | I | UP | Warm Reset of non isolated portion on the IC |
| POR | Y18 | I | | Power-on Reset |
| RESETSTAT | H5 | O | UP | Reset Status Output |
| BOOTCOMPLETE | H3 | OZ | Down | Boot progress indication output |
| PTV15 | F15 | A | | PTV Compensation NMOS Reference Input. A precision resistor placed between the PTV15 pin and ground is used to closely tune the output impedance of the DDR interface drivers to 50 Ohms. Presently, the recommended value for this 1% resistor is 45.3 Ohms. |

Table 3-41. Terminal Functions — Signals and Control by Function (continued)

| SIGNAL NAME | BALL NO. | TYPE | IPD/IPU | DESCRIPTION |
|-------------|----------|------|---------|----------------------|
| DDR | | | | |
| DDRDQM0 | A8 | OZ | | DDR EMIF Data Masks |
| DDRDQM1 | E7 | OZ | | |
| DDRDQM2 | F5 | OZ | | |
| DDRDQM3 | E1 | OZ | | |
| DDRDQM8 | C12 | OZ | | |
| DDRDQS0P | D10 | IOZ | | DDR EMIF Data Strobe |
| DDRDQS0N | C10 | IOZ | | |
| DDRDQS1P | B7 | IOZ | | |
| DDRDQS1N | A7 | IOZ | | |
| DDRDQS2P | B4 | IOZ | | |
| DDRDQS2N | A4 | IOZ | | |
| DDRDQS3P | A2 | IOZ | | |
| DDRDQS3N | B2 | IOZ | | |
| DDRDQS8P | B13 | IOZ | | |
| DDRDQS8N | A13 | IOZ | | |
| DDRCB00 | D11 | IOZ | | DDR EMIF Check Bits |
| DDRCB01 | B12 | IOZ | | |
| DDRCB02 | C11 | IOZ | | |
| DDRCB03 | A12 | IOZ | | |
| DDRD00 | A9 | IOZ | | DDR EMIF Data Bus |
| DDRD01 | C9 | IOZ | | |
| DDRD02 | D9 | IOZ | | |
| DDRD03 | B9 | IOZ | | |
| DDRD04 | E9 | IOZ | | |
| DDRD05 | E10 | IOZ | | |
| DDRD06 | A11 | IOZ | | |
| DDRD07 | B11 | IOZ | | |
| DDRD08 | E6 | IOZ | | |
| DDRD09 | E8 | IOZ | | |
| DDRD10 | A6 | IOZ | | |
| DDRD11 | A5 | IOZ | | |
| DDRD12 | D6 | IOZ | | |
| DDRD13 | C7 | IOZ | | |
| DDRD14 | D7 | IOZ | | |
| DDRD15 | B8 | IOZ | | |
| DDRD16 | E5 | IOZ | | |
| DDRD17 | B3 | IOZ | | |
| DDRD18 | F4 | IOZ | | |
| DDRD19 | E4 | IOZ | | |
| DDRD20 | A3 | IOZ | | |

Table 3-41. Terminal Functions — Signals and Control by Function (continued)

| SIGNAL NAME | BALL NO. | TYPE | IPD/IPU | DESCRIPTION |
|-------------|----------|------|---------|---|
| DDR21 | B5 | IOZ | | DDR EMIF Data Bus |
| DDR22 | C5 | IOZ | | |
| DDR23 | D5 | IOZ | | |
| DDR24 | E2 | IOZ | | |
| DDR25 | F2 | IOZ | | |
| DDR26 | B1 | IOZ | | |
| DDR27 | C1 | IOZ | | |
| DDR28 | D1 | IOZ | | |
| DDR29 | D3 | IOZ | | |
| DDR30 | C3 | IOZ | | |
| DDR31 | E3 | IOZ | | DDR EMIF Data Bus |
| DDRCE0 | B15 | OZ | | DDR EMIF Chip Enables |
| DDRCE1 | C14 | OZ | | |
| DDRBA0 | C18 | OZ | | DDR EMIF Bank Address |
| DDRBA1 | D17 | OZ | | |
| DDRBA2 | B19 | OZ | | |
| DDRA00 | D16 | OZ | | DDR EMIF Address Bus |
| DDRA01 | A19 | OZ | | |
| DDRA02 | E16 | OZ | | |
| DDRA03 | E15 | OZ | | |
| DDRA04 | B18 | OZ | | |
| DDRA05 | A17 | OZ | | |
| DDRA06 | C16 | OZ | | |
| DDRA07 | A18 | OZ | | |
| DDRA08 | D20 | OZ | | |
| DDRA09 | E20 | OZ | | |
| DDRA10 | E19 | OZ | | |
| DDRA11 | B20 | OZ | | |
| DDRA12 | D18 | OZ | | |
| DDRA13 | C20 | OZ | | |
| DDRA14 | E18 | OZ | | |
| DDRA15 | E17 | OZ | | |
| DDRCAS | D14 | OZ | | DDR EMIF Column Address Strobe |
| DDRRAS | A15 | OZ | | DDR EMIF Row Address Strobe |
| DDRWE | E13 | OZ | | DDR EMIF Write Enable |
| DDRCKE0 | A16 | OZ | | DDR EMIF Clock Enable |
| DDRCKE1 | A20 | OZ | | DDR EMIF Clock Enable |
| DDRCLKOUTP0 | A14 | OZ | | DDR EMIF Output Clocks to drive SDRAMs (one clock pair per SDRAM) |
| DDRCLKOUTN0 | B14 | OZ | | |
| DDRCLKOUTP1 | A21 | OZ | | |
| DDRCLKOUTN1 | B21 | OZ | | |
| DDRODT0 | E14 | OZ | | DDR EMIF On Die Termination Outputs used to set termination on the SDRAMs |
| DDRODT1 | D12 | OZ | | DDR EMIF On Die Termination Outputs used to set termination on the SDRAMs |
| DDRRESET | B16 | OZ | | DDR Reset signal |
| DDRSLRATE0 | C22 | I | Down | DDR Slew rate control |
| DDRSLRATE1 | D22 | I | Down | |
| VREFSSTL | E12 | P | | Reference Voltage Input for SSTL15 buffers used by DDR EMIF (VDDS15 ÷ 2) |

Table 3-41. Terminal Functions — Signals and Control by Function (continued)

| SIGNAL NAME | BALL NO. | TYPE | IPD/IPU | DESCRIPTION |
|-----------------------|----------|------|---------|--|
| EMIF16 | | | | |
| EMIFR \overline{W} | L5 | OZ | UP | EMIF16 Control Signals |
| EMIFCE $\overline{0}$ | K5 | OZ | UP | |
| EMIFCE1 | G1 | OZ | UP | |
| EMIFCE $\overline{2}$ | J2 | OZ | UP | |
| EMIFCE $\overline{3}$ | M5 | OZ | UP | |
| EMIFOE | L4 | OZ | UP | |
| EMIFWE | K4 | OZ | UP | |
| EMIFBE $\overline{0}$ | J1 | OZ | UP | |
| EMIFBE $\overline{1}$ | L3 | OZ | UP | |
| EMIFWAIT0 | N5 | I | Down | |
| EMIFWAIT1 | M4 | I | Down | EMIF16 Control Signal This EMIF16 pin has a secondary function assigned to it as mentioned elsewhere in this table (see UPP). |
| EMIFA00 | K1 | OZ | Down | EMIF16 Address These EMIF16 pins have secondary functions assigned to them as mentioned elsewhere in this table (see uPP). |
| EMIFA01 | M3 | OZ | Down | |
| EMIFA02 | L2 | OZ | Down | |
| EMIFA03 | P5 | OZ | Down | |
| EMIFA04 | L1 | OZ | Down | |
| EMIFA05 | P4 | OZ | Down | |
| EMIFA06 | M2 | OZ | Down | |
| EMIFA07 | M1 | OZ | Down | |
| EMIFA08 | N2 | OZ | Down | |
| EMIFA09 | P3 | OZ | Down | |
| EMIFA10 | N1 | OZ | Down | |
| EMIFA11 | P2 | OZ | Down | |
| EMIFA12 | P1 | OZ | Down | |
| EMIFA13 | R5 | OZ | Down | |
| EMIFA14 | R3 | OZ | Down | |
| EMIFA15 | R4 | OZ | Down | |
| EMIFA16 | R2 | OZ | Down | |
| EMIFA17 | R1 | OZ | Down | |
| EMIFA18 | T4 | OZ | Down | |
| EMIFA19 | T1 | OZ | Down | |
| EMIFA20 | T5 | OZ | Down | |
| EMIFA21 | U1 | OZ | Down | |
| EMIFA22 | U2 | OZ | Down | |
| EMIFA23 | U3 | OZ | Down | |

Table 3-41. Terminal Functions — Signals and Control by Function (continued)

| SIGNAL NAME | BALL NO. | TYPE | IPD/IPU | DESCRIPTION |
|------------------|----------|------|---------|--|
| EMIFD00 | U4 | IOZ | Down | EMIF16 Data These EMIF16 pins have secondary functions assigned to them as mentioned elsewhere in this table (see uPP). |
| EMIFD01 | U5 | IOZ | Down | |
| EMIFD02 | V1 | IOZ | Down | |
| EMIFD03 | V2 | IOZ | Down | |
| EMIFD04 | V3 | IOZ | Down | |
| EMIFD05 | V4 | IOZ | Down | |
| EMIFD06 | W1 | IOZ | Down | |
| EMIFD07 | V5 | IOZ | Down | |
| EMIFD08 | W2 | IOZ | Down | |
| EMIFD09 | Y1 | IOZ | Down | |
| EMIFD10 | W4 | IOZ | Down | |
| EMIFD11 | Y2 | IOZ | Down | |
| EMIFD12 | W5 | IOZ | Down | |
| EMIFD13 | AA1 | IOZ | Down | |
| EMIFD14 | AB1 | IOZ | Down | |
| EMIFD15 | AA2 | IOZ | Down | |
| uPP | | | | |
| UPP_2TXCLK † | M4 | I | Down | uPP Transmit Reference Clock (2x Transmit Rate) This uPP pin has a primary function assigned to it as mentioned elsewhere in this table (see EMIF16). |
| UPP_CH0_CLK † | R2 | IOZ | Down | uPP Channel 0 Clock This uPP pin has a primary function assigned to it as mentioned elsewhere in this table (see EMIF16). |
| UPP_CH0_START † | R1 | IOZ | Down | uPP Channel 0 Start This uPP pin has a primary function assigned to it as mentioned elsewhere in this table (see EMIF16). |
| UPP_CH0_ENABLE † | T4 | IOZ | Down | uPP Channel 0 Enable This uPP pin has a primary function assigned to it as mentioned elsewhere in this table (see EMIF16). |
| UPP_CH0_WAIT † | T1 | IOZ | Down | uPP Channel 0 Wait This uPP pin has a primary function assigned to it as mentioned elsewhere in this table (see EMIF16). |
| UPP_CH1_CLK † | T5 | IOZ | Down | uPP Channel 1 Clock This uPP pin has a primary function assigned to it as mentioned elsewhere in this table (see EMIF16). |
| UPP_CH1_START † | U1 | IOZ | Down | uPP Channel 1 Start This uPP pin has a primary function assigned to it as mentioned elsewhere in this table (see EMIF16). |
| UPP_CH1_ENABLE † | U2 | IOZ | Down | uPP Channel 1 Enable This uPP pin has a primary function assigned to it as mentioned elsewhere in this table (see EMIF16). |
| UPP_CH1_WAIT † | U3 | IOZ | Down | uPP Channel 1 Wait This uPP pin has a primary function assigned to it as mentioned elsewhere in this table (see EMIF16). |

Table 3-41. Terminal Functions — Signals and Control by Function (continued)

| SIGNAL NAME | BALL NO. | TYPE | IPD/IPU | DESCRIPTION |
|-------------|----------|------|---------|--|
| UPPD00 † | U4 | IOZ | Down | uPP Data This uPP pin has a primary function assigned to it as mentioned elsewhere in this table (see EMIF16). |
| UPPD01 † | U5 | IOZ | Down | |
| UPPD02 † | V1 | IOZ | Down | |
| UPPD03 † | V2 | IOZ | Down | |
| UPPD04 † | V3 | IOZ | Down | |
| UPPD05 † | V4 | IOZ | Down | |
| UPPD06 † | W1 | IOZ | Down | |
| UPPD07 † | V5 | IOZ | Down | |
| UPPD08 † | W2 | IOZ | Down | |
| UPPD09 † | Y1 | IOZ | Down | |
| UPPD10 † | W4 | IOZ | Down | |
| UPPD11 † | Y2 | IOZ | Down | |
| UPPD12 † | W5 | IOZ | Down | |
| UPPD13 † | AA1 | IOZ | Down | |
| UPPD14 † | AB1 | IOZ | Down | |
| UPPD15 † | AA2 | IOZ | Down | uPP Extended Data This uPP pin has a primary function assigned to it as mentioned elsewhere in this table (see EMIF16). |
| UPPXD00 † | K1 | IOZ | Down | |
| UPPXD01 † | M3 | IOZ | Down | |
| UPPXD02 † | L2 | IOZ | Down | |
| UPPXD03 † | P5 | IOZ | Down | |
| UPPXD04 † | L1 | IOZ | Down | |
| UPPXD05 † | P4 | IOZ | Down | |
| UPPXD06 † | M2 | IOZ | Down | |
| UPPXD07 † | M1 | IOZ | Down | |
| UPPXD08 † | N2 | IOZ | Down | |
| UPPXD09 † | P3 | IOZ | Down | |
| UPPXD10 † | N1 | IOZ | Down | |
| UPPXD11 † | P2 | IOZ | Down | |
| UPPXD12 † | P1 | IOZ | Down | |
| UPPXD13 † | R5 | IOZ | Down | |
| UPPXD14 † | R3 | IOZ | Down | |
| UPPXD15 † | R4 | IOZ | Down | |

Table 3-41. Terminal Functions — Signals and Control by Function (continued)

| SIGNAL NAME | BALL NO. | TYPE | IPD/IPU | DESCRIPTION |
|--|----------|------|---------|---|
| EMU | | | | |
| EMU00 | V24 | IOZ | UP | Emulation and Trace Port |
| EMU01 | V25 | IOZ | UP | |
| EMU02 | W25 | IOZ | UP | |
| EMU03 | W23 | IOZ | UP | |
| EMU04 | W24 | IOZ | UP | |
| EMU05 | Y25 | IOZ | UP | |
| EMU06 | Y24 | IOZ | UP | |
| EMU07 | Y23 | IOZ | UP | |
| EMU08 | W22 | IOZ | UP | |
| EMU09 | Y22 | IOZ | UP | |
| EMU10 | AA24 | IOZ | UP | |
| EMU11 | AA25 | IOZ | UP | |
| EMU12 | AB25 | IOZ | UP | |
| EMU13 | AC25 | IOZ | UP | |
| EMU14 | AA23 | IOZ | UP | |
| EMU15 | AB22 | IOZ | UP | |
| EMU16 | AD25 | IOZ | UP | |
| EMU17 | AC24 | IOZ | UP | |
| EMU18 | Y21 | IOZ | UP | |
| General Purpose Input/Output (GPIO) | | | | |
| GPIO00 | T25 | IOZ | UP | General Purpose Input/Output These GPIO pins have secondary functions assigned to them as mentioned elsewhere in this table (see Boot Configuration Pins). |
| GPIO01 | R25 | IOZ | Down | |
| GPIO02 | R23 | IOZ | Down | |
| GPIO03 | U25 | IOZ | Down | |
| GPIO04 | T23 | IOZ | Down | |
| GPIO05 | U24 | IOZ | Down | |
| GPIO06 | T22 | IOZ | Down | |
| GPIO07 | R21 | IOZ | Down | |
| GPIO08 | U22 | IOZ | Down | |
| GPIO09 | U23 | IOZ | Down | |
| GPIO10 | V23 | IOZ | Down | |
| GPIO11 | U21 | IOZ | Down | |
| GPIO12 | T21 | IOZ | Down | |
| GPIO13 | V22 | IOZ | Down | |
| GPIO14 | W21 | IOZ | Down | |
| GPIO15 | V21 | IOZ | Down | |
| GPIO16 † | AD20 | IOZ | Down | General Purpose Input/Output This GPIO pin has a primary function assigned to it as mentioned elsewhere in this table (see Timer) and a tertiary function assigned to it as mentioned elsewhere in this table (see Boot Configuration Pins). |
| GPIO17 † | AE21 | IOZ | Down | General Purpose Input/Output |
| GPIO18 † | AC19 | IOZ | Down | These GPIO pins have primary functions assigned to them as mentioned elsewhere in this table (see Timer). |
| GPIO19 † | AE20 | IOZ | Down | |

Table 3-41. Terminal Functions — Signals and Control by Function (continued)

| SIGNAL NAME | BALL NO. | TYPE | IPD/IPU | DESCRIPTION |
|-----------------------|----------|------|---------|--|
| GPIO20 † | AB15 | IOZ | Down | General Purpose Input/Output These GPIO pins have primary functions assigned to them as mentioned elsewhere in this table (see UART). |
| GPIO21 † | AA15 | IOZ | Down | |
| GPIO22 † | AC17 | IOZ | Down | |
| GPIO23 † | AB17 | IOZ | Down | |
| GPIO24 † | AC14 | IOZ | Down | |
| GPIO25 † | AC15 | IOZ | Down | |
| GPIO26 † | AE16 | IOZ | Down | |
| GPIO27 † | AD15 | IOZ | Down | |
| GPIO28 † | AA12 | IOZ | Up | General Purpose Input/Output These GPIO pins have primary functions assigned to them as mentioned elsewhere in this table (see SPI). |
| GPIO29 † | AA14 | IOZ | Up | |
| GPIO30 † | AB14 | IOZ | Down | |
| GPIO31 † | AB13 | IOZ | Down | |
| HyperLink | | | | |
| MCMRXN0 | P24 | I | | Serial HyperLink Receive Data |
| MCMRXP0 | N24 | I | | |
| MCMRXN1 | M25 | I | | |
| MCMRXP1 | N25 | I | | |
| MCMRXN2 | J25 | I | | |
| MCMRXP2 | K25 | I | | |
| MCMRXN3 | K24 | I | | |
| MCMRXP3 | L24 | I | | |
| MCMTXN0 | P22 | O | | Serial HyperLink Transmit Data |
| MCMTXP0 | N22 | O | | |
| MCMTXN1 | N21 | O | | |
| MCMTXP1 | M21 | O | | |
| MCMTXN2 | K22 | O | | |
| MCMTXP2 | L22 | O | | |
| MCMTXN3 | J21 | O | | |
| MCMTXP3 | K21 | O | | |
| MCMRXFLCLK | B24 | O | Down | Serial HyperLink Sideband Signals |
| MCMRXFLDAT | C24 | O | Down | |
| MCMTXFLCLK | E25 | I | Down | |
| MCMTXFLDAT | D25 | I | Down | |
| MCMRXPCLK | E24 | I | Down | |
| MCMRXPMDAT | D24 | I | Down | |
| MCMTXPCLK | F24 | O | Down | |
| MCMTXPMDAT | G24 | O | Down | |
| MCMREFCLKOUTP | G25 | O | | HyperLink Reference clock output for daisy chain connection |
| MCMREFCLKOUTN | F25 | O | | |
| I²C | | | | |
| SCL | AA17 | IOZ | | I ² C Clock |
| SDA | AA18 | IOZ | | I ² C Data |

Table 3-41. Terminal Functions — Signals and Control by Function (continued)

| SIGNAL NAME | BALL NO. | TYPE | IPD/IPU | DESCRIPTION |
|--------------------------|----------|------|---------|---------------------------------------|
| JTAG | | | | |
| TCK | AD17 | I | Up | JTAG Clock Input |
| TDI | AE17 | I | Up | JTAG Data Input |
| TDO | AD19 | OZ | Up | JTAG Data Output |
| TMS | AE18 | I | Up | JTAG Test Mode Input |
| $\overline{\text{TRST}}$ | AB19 | I | Down | JTAG Reset |
| McBSP | | | | |
| CLKR0 | AA21 | IOZ | Down | McBSP Receive Clock |
| CLKX0 | Y20 | IOZ | Down | McBSP Transmit Clock |
| CLKS0 | AC23 | IOZ | Down | McBSP Slow Clock |
| FSR0 | AD24 | IOZ | Down | McBSP Receive Frame Sync |
| FSX0 | AA20 | IOZ | Down | McBSP Transmit Frame Sync |
| DR0 | AB21 | I | Down | McBSP Receive Data |
| DX0 | AC22 | OZ | Down | McBSP Transmit Data |
| CLKR1 | AD23 | IOZ | Down | McBSP Receive Clock |
| CLKX1 | AE24 | IOZ | Down | McBSP Transmit Clock |
| CLKS1 | AC21 | IOZ | Down | McBSP Slow Clock |
| FSR1 | AD22 | IOZ | Down | McBSP Receive Frame Sync |
| FSX1 | AE23 | IOZ | Down | McBSP Transmit Frame Sync |
| DR1 | AD21 | I | Down | McBSP Receive Data |
| DX1 | AE22 | OZ | Down | McBSP Transmit Data |
| MDIO | | | | |
| MDIO | AB16 | IOZ | Up | MDIO Data |
| MDCLK | AA16 | O | Down | MDIO Clock |
| PCIe | | | | |
| PCIERXN0 | AE12 | I | | PCIexpress Receive Data (2 links) |
| PCIERXP0 | AE11 | I | | |
| PCIERXN1 | AD10 | I | | |
| PCIERXP1 | AD11 | I | | |
| PCIETXN0 | AC12 | O | | PCIexpress Transmit Data (2 links) |
| PCIETXP0 | AC11 | O | | |
| PCIETXN1 | AB11 | O | | |
| PCIETXP1 | AB10 | O | | |
| Serial RapidIO | | | | |
| RIORXN0 | AE9 | I | | Serial RapidIO Receive Data (4 links) |
| RIORXP0 | AE8 | I | | |
| RIORXN1 | AD8 | I | | |
| RIORXP1 | AD7 | I | | |
| RIORXN2 | AE5 | I | | |
| RIORXP2 | AE6 | I | | |
| RIORXN3 | AD4 | I | | |
| RIORXP3 | AD5 | I | | |

Table 3-41. Terminal Functions — Signals and Control by Function (continued)

| SIGNAL NAME | BALL NO. | TYPE | IPD/IPU | DESCRIPTION |
|--------------------|----------|------|---------|---|
| RIOTXN0 | AC9 | O | | Serial RapidIO Receive Data (4 links) |
| RIOTXP0 | AC8 | O | | |
| RIOTXN1 | AB7 | O | | |
| RIOTXP1 | AB8 | O | | |
| RIOTXN2 | AC5 | O | | |
| RIOTXP2 | AC6 | O | | |
| RIOTXN3 | AB4 | O | | |
| RIOTXP3 | AB5 | O | | |
| SGMII | | | | |
| SGMII0RXN | AE2 | I | | Ethernet MAC SGMII Receive Data |
| SGMII0RXP | AE3 | I | | |
| SGMII0TXN | AC2 | O | | Ethernet MAC SGMII Transmit Data |
| SGMII0TXP | AC3 | O | | |
| SmartReflex | | | | |
| VCNTL0 | E22 | OZ | | Voltage Control Outputs to variable core power supply. These are open-drain output buffers. |
| VCNTL1 | E23 | OZ | | |
| VCNTL2 | F23 | OZ | | |
| VCNTL3 | G23 | OZ | | |
| SPI | | | | |
| SPISCS0 | AA12 | OZ | Up | SPI Interface Enable 0 This SPI pin has a secondary function assigned to it as mentioned elsewhere in this table (see GPIO). |
| SPISCS1 | AA14 | OZ | Up | SPI Interface Enable 1 This SPI pin has a secondary function assigned to it as mentioned elsewhere in this table (see GPIO). |
| SPICLK | AA13 | OZ | Down | SPI Clock |
| SPIDIN | AB14 | I | Down | SPI Data In This SPI pin has a secondary function assigned to it as mentioned elsewhere in this table (see GPIO). |
| SPIDOUT | AB13 | OZ | Down | SPI Data Out This SPI pin has a secondary function assigned to it as mentioned elsewhere in this table (see GPIO). |
| Timer | | | | |
| TIMIO | AD20 | I | Down | Timer Inputs |
| TIMI1 | AE21 | I | Down | This SPI pin has a secondary function assigned to it as mentioned elsewhere in this table (see GPIO). |
| TIMO0 | AC19 | OZ | Down | Timer Outputs |
| TIMO1 | AE20 | OZ | Down | These Timer pins have secondary functions assigned to them as mentioned elsewhere in this table |
| UART | | | | |
| UARTRXD | AB15 | I | Down | UART Serial Data In This SPI pin has a secondary function assigned to it as mentioned elsewhere in this table (see GPIO). |
| UARTTXD | AA15 | OZ | Down | UART Serial Data Out This SPI pin has a secondary function assigned to it as mentioned elsewhere in this table (see GPIO). |

Table 3-41. Terminal Functions — Signals and Control by Function (continued)

| SIGNAL NAME | BALL NO. | TYPE | IPD/IPU | DESCRIPTION |
|-----------------|----------|------|---------|---|
| UARTCTS | AC17 | I | Down | UART Clear To Send This SPI pin has a secondary function assigned to it as mentioned elsewhere in this table (see GPIO). |
| UARTRTS | AB17 | OZ | Down | UART Request To Send This SPI pin has a secondary function assigned to it as mentioned elsewhere in this table (see GPIO). |
| UARTRXD1 | AC14 | I | Down | UART Serial Data In This SPI pin has a secondary function assigned to it as mentioned elsewhere in this table (see GPIO). |
| UARTTXD1 | AC15 | OZ | Down | UART Serial Data Out This SPI pin has a secondary function assigned to it as mentioned elsewhere in this table (see GPIO). |
| UARTCTS1 | AE16 | I | Down | UART Clear To Send This SPI pin has a secondary function assigned to it as mentioned elsewhere in this table (see GPIO). |
| UARTRTS1 | AD15 | OZ | Down | UART Request To Send This SPI pin has a secondary function assigned to it as mentioned elsewhere in this table (see GPIO). |
| Reserved | | | | |
| RSV01 | AA22 | IOZ | Up | Reserved - pullup to DVDD18 |
| RSV02 | J3 | OZ | Down | Reserved - leave unconnected |
| RSV03 | H2 | OZ | Down | Reserved - leave unconnected |
| RSV04 | AC18 | O | | Reserved - leave unconnected |
| RSV05 | AB18 | O | | Reserved - leave unconnected |
| RSV06 | B23 | O | | Reserved - leave unconnected |
| RSV07 | A23 | O | | Reserved - leave unconnected |
| RSV08 | Y19 | OZ | Down | Reserved - leave unconnected |
| RSV09 | C23 | OZ | Down | Reserved - leave unconnected |
| RSV10 | G22 | A | | Reserved - connect to GND |
| RSV11 | H22 | A | | Reserved - leave unconnected |
| RSV12 | Y5 | A | | Reserved - leave unconnected |
| RSV13 | Y4 | A | | Reserved - leave unconnected |
| RSV14 | F21 | A | | Reserved - leave unconnected |
| RSV15 | G21 | A | | Reserved - leave unconnected |
| RSV16 | J20 | A | | Reserved - leave unconnected |
| RSV17 | AA7 | A | | Reserved - leave unconnected |
| RSV18 | AA11 | A | | Reserved - leave unconnected |
| RSV19 | AB3 | A | | Reserved - leave unconnected |
| RSV20 | F22 | IOZ | | Reserved - leave unconnected |
| RSV21 | D23 | IOZ | | Reserved - leave unconnected |
| RSV0A | G19 | A | | Reserved - leave unconnected |
| RSV0B | G20 | A | | Reserved - leave unconnected |

Table 3-42. Terminal Functions — Power and Ground

| SUPPLY | BALL NO. | VOLTS | DESCRIPTION |
|----------|--|-------------|--|
| AVDDA1 | Y15 | 1.8 | PLL Supply - CORE_PLL |
| AVDDA2 | F20 | 1.8 | PLL Supply - DDR3_PLL |
| CVDD | H9, H11, H13, H15, H17, J10, J12, J14, J16, K11, K13, K15, L8, L10, L12, L14, L16, L18, M9, M11, M13, M15, M17, N8, N10, N12, N14, N16, N18, P9, P11, P13, P15, P17, P19, R10, R12, R14, R16, R18, T11, T13, T15, U10, U12, U14, U16, V9, V11, V13, V15, V17 | 0.85 to 1.1 | SmartReflex core supply voltage |
| CVDD1 | J8, J18, K9, K17, T9, T17, U8, U18 | 1.0 | Fixed core supply voltage for memory array |
| DVDD15 | B10, C6, C17, C21, D2, D4, D8, D13, D15, D19, F7, F9, F11, F13, F17, F19, G8, G10, G12, G14, G16, G18 | 1.5 | DDR IO supply |
| DVDD18 | A24, E21, G3, G6, H7, H19, H24, J6, K3, K7, L6, M7, N3, N6, P7, R6, R20, T3, T7, T19, T24, U6, U20, V7, V19, W6, W14, W16, W18, W20, Y3, Y13, Y17, AB23, AC16, AC20 | 1.8 | IO supply |
| VDDR1 | M20 | 1.5 | HyperLink SerDes regulator supply |
| VDDR2 | AA9 | 1.5 | PCIe SerDes regulator supply |
| VDDR3 | AA3 | 1.5 | SGMII SerDes regulator supply |
| VDDR4 | AA5 | 1.5 | SRIO SerDes regulator supply |
| VDDT1 | K19, L20, M19, N20 | 1.0 | HyperLink SerDes termination supply |
| VDDT2 | W8, W10, W12, Y7, Y9, Y11 | 1.0 | SGMII/SRIO/PCIe SerDes termination supply |
| VREFSSTL | E12 | 0.75 | DDR3 reference voltage |
| VSS | A1, A10, A25, B6, B17, C2, C4, C8, C13, C15, C19, D21, E11, F3, F6, F8, F10, F12, F14, F16, F18, G7, G9, G11, G13, G15, G17, H6, H8, H10, H12, H14, H16, H18, H20, H21, H23, H25, J7, J9, J11, J13, J15, J17, J19, J22, J23, J24, K2, K6, K8, K10, K12, K14, K16, K18, K20, K23, L7, L9, L11, L13, L15, L17, L19, L21, L23, L25, M6, M8, M10, M12, M14, M16, M18, M22, M23, M24, N4, N7, N9, N11, N13, N15, N17, N19, N23, P6, P8, P10, P12, P14, P16, P18, P20, P21, P23, P25, R7, R8, R9, R11, R13, R15, R17, R19, R22, R24, T2, T6, T8, T10, T12, T14, T16, T18, T20, U7, U9, U11, U13, U15, U17, U19, V6, V8, V10, V12, V14, V16, V18, V20, W3, W7, W9, W11, W13, W15, W17, W19, Y6, Y8, Y10, Y12, Y14, Y16, AA4, AA6, AA8, AA10, AB2, AB6, AB9, AB12, AB20, AB24, AC1, AC4, AC7, AC10, AC13, AD1, AD2, AD3, AD6, AD9, AD12, AD16, AE1, AE4, AE7, AE10, AE13, AE25 | GND | Ground |

Table 3-43. Terminal Functions — By Signal Name

| SIGNAL NAME | BALL NUMBER | SIGNAL NAME | BALL NUMBER | SIGNAL NAME | BALL NUMBER | | | | |
|--------------|--|-------------|-------------|-------------|---|--------|---|--|--|
| AVDDA1 | Y15 | DDRA09 | E20 | DDRD22 | C5 | | | | |
| AVDDA2 | F20 | DDRA10 | E19 | DDRD23 | D5 | | | | |
| BOOTCOMPLETE | H3 | DDRA11 | B20 | DDRD24 | E2 | | | | |
| BOOTMODE00 † | R25 | DDRA12 | D18 | DDRD25 | F2 | | | | |
| BOOTMODE01 † | R23 | DDRA13 | C20 | DDRD26 | B1 | | | | |
| BOOTMODE02 † | U25 | DDRA14 | E18 | DDRD27 | C1 | | | | |
| BOOTMODE03 † | T23 | DDRA15 | E17 | DDRD28 | D1 | | | | |
| BOOTMODE04 † | U24 | DDRBA0 | C18 | DDRD29 | D3 | | | | |
| BOOTMODE05 † | T22 | DDRBA1 | D17 | DDRD30 | C3 | | | | |
| BOOTMODE06 † | R21 | DDRBA2 | B19 | DDRD31 | E3 | | | | |
| BOOTMODE07 † | U22 | DDRCAS | D14 | DDRDQM0 | A8 | | | | |
| BOOTMODE08 † | U23 | DDRCB00 | D11 | DDRDQM1 | E7 | | | | |
| BOOTMODE09 † | V23 | DDRCB01 | B12 | DDRDQM2 | F5 | | | | |
| BOOTMODE10 † | U21 | DDRCB02 | C11 | DDRDQM3 | E1 | | | | |
| BOOTMODE11 † | T21 | DDRCB03 | A12 | DDRDQM8 | C12 | | | | |
| BOOTMODE12 † | V22 | DDRCOE0 | B15 | DDRDQS0N | C10 | | | | |
| CLKR0 | AA21 | DDRCOE1 | C14 | DDRDQS0P | D10 | | | | |
| CLKR1 | AD23 | DDRCKE0 | A16 | DDRDQS1N | A7 | | | | |
| CLKS0 | AC23 | DDRCKE1 | A20 | DDRDQS1P | B7 | | | | |
| CLKS1 | AC21 | DDRCLKN | B22 | DDRDQS2N | A4 | | | | |
| CLKX0 | Y20 | DDRCLKOUTN0 | B14 | DDRDQS2P | B4 | | | | |
| CLKX1 | AE24 | DDRCLKOUTN1 | B21 | DDRDQS3N | B2 | | | | |
| CORECLKN | AE19 | DDRCLKOUTP0 | A14 | DDRDQS3P | A2 | | | | |
| CORECLKP | AD18 | DDRCLKOUTP1 | A21 | DDRDQS8N | A13 | | | | |
| CORESEL0 | J5 | DDRCLKP | A22 | DDRDQS8P | B13 | | | | |
| CORESEL1 | G5 | DDRD00 | A9 | DDRODT0 | E14 | | | | |
| CVDD | H9, H11, H13, H15, H17, J10, J12, J14, J16, K11, K13, K15, L8, L10, L12, L14, L16, L18, M9, M11, M13, M15, M17, N8, N10, N12, N14, N16, N18, P9, P11, P13, P15, P17, P19, R10, R12, R14, R16, R18, T11, T13, T15, U10, U12, U14, U16, V9, V11, V13, V15, V17 | DDRD01 | C9 | DDRODT1 | D12 | | | | |
| | | DDRD02 | D9 | DDRRAS | A15 | | | | |
| | | DDRD03 | B9 | DDRRESET | B16 | | | | |
| | | DDRD04 | E9 | DDRSLRATE0 | C22 | | | | |
| | | DDRD05 | E10 | DDRSLRATE1 | D22 | | | | |
| | | DDRD06 | A11 | DDRWE | E13 | | | | |
| | | DDRD07 | B11 | DR0 | AB21 | | | | |
| | | DDRD08 | E6 | DR1 | AD21 | | | | |
| | | DDRD09 | E8 | | | | | | |
| | | DDRD10 | A6 | | | | | | |
| CVDD1 | J8, J18, K9, K17, T9, T17, U8, U18 | DDRD11 | A5 | DVDD15 | B10, C6, C17, C21, D2, D4, D8, D13, D15, D19, F7, F9, F11, F13, F17, F19, G8, G10, G12, G14, G16, G18 | | | | |
| | | DDRD12 | D6 | | | | | | |
| DDRA00 | D16 | DDRD13 | C7 | | | | | | |
| DDRA01 | A19 | DDRD14 | D7 | | | DVDD18 | A24, E21, G3, G6, H7, H19, H24, J6, K3, K7, L6, M7, N3, N6, P7, R6, R20, T3, T7, T19, T24, U6, U20, V7, V19, W6, W14, W16, W18, W20, Y3, Y13, Y17, AB23, AC16, AC20 | | |
| DDRA02 | E16 | DDRD15 | B8 | | | | | | |
| DDRA03 | E15 | DDRD16 | E5 | | | | | | |
| DDRA04 | B18 | DDRD17 | B3 | | | | | | |
| DDRA05 | A17 | DDRD18 | F4 | | | | | | |
| DDRA06 | C16 | DDRD19 | E4 | | | | | | |
| DDRA07 | A18 | DDRD20 | A3 | | | | | | |
| DDRA08 | D20 | DDRD21 | B5 | | | | | | |

Table 3-43. Terminal Functions — By Signal Name (continued)

| SIGNAL NAME | BALL NUMBER | SIGNAL NAME | BALL NUMBER | SIGNAL NAME | BALL NUMBER |
|-------------|-------------|-------------|-------------|---------------|-------------|
| DX0 | AC22 | EMIFD15 | AA2 | GPIO18 † | AC19 |
| DX1 | AE22 | EMIFOE | L4 | GPIO19 † | AE20 |
| EMIFA00 | K1 | EMIFRNW | L5 | GPIO20 † | AB15 |
| EMIFA01 | M3 | EMIFWAIT0 | N5 | GPIO21 † | AA15 |
| EMIFA02 | L2 | EMIFWAIT1 | M4 | GPIO22 † | AC17 |
| EMIFA03 | P5 | EMIFWE | K4 | GPIO23 † | AB17 |
| EMIFA04 | L1 | EMU00 | V24 | GPIO24 † | AC14 |
| EMIFA05 | P4 | EMU01 | V25 | GPIO25 † | AC15 |
| EMIFA06 | M2 | EMU02 | W25 | GPIO26 † | AE16 |
| EMIFA07 | M1 | EMU03 | W23 | GPIO27 † | AD15 |
| EMIFA08 | N2 | EMU04 | W24 | GPIO28 † | AA12 |
| EMIFA09 | P3 | EMU05 | Y25 | GPIO29 † | AA14 |
| EMIFA10 | N1 | EMU06 | Y24 | GPIO30 † | AB14 |
| EMIFA11 | P2 | EMU07 | Y23 | GPIO31 † | AB13 |
| EMIFA12 | P1 | EMU08 | W22 | HOUT | G2 |
| EMIFA13 | R5 | EMU09 | Y22 | LENDIAN † | T25 |
| EMIFA14 | R3 | EMU10 | AA24 | LRESETNMIEN | F1 |
| EMIFA15 | R4 | EMU11 | AA25 | LRESET | G4 |
| EMIFA16 | R2 | EMU12 | AB25 | MCMCLKN | B25 |
| EMIFA17 | R1 | EMU13 | AC25 | MCMCLKP | C25 |
| EMIFA18 | T4 | EMU14 | AA23 | MCMREFCLKOUTN | F25 |
| EMIFA19 | T1 | EMU15 | AB22 | MCMREFCLKOUTP | G25 |
| EMIFA20 | T5 | EMU16 | AD25 | MCMRXFLCLK | B24 |
| EMIFA21 | U1 | EMU17 | AC24 | MCMRXFLDAT | C24 |
| EMIFA22 | U2 | EMU18 | Y21 | MCMRXN0 | P24 |
| EMIFA23 | U3 | FSR0 | AD24 | MCMRXN1 | M25 |
| EMIFBE0 | J1 | FSR1 | AD22 | MCMRXN2 | J25 |
| EMIFBE1 | L3 | FSX0 | AA20 | MCMRXN3 | K24 |
| EMIFCE0 | K5 | FSX1 | AE23 | MCMRXP0 | N24 |
| EMIFCE1 | G1 | GPIO00 | T25 | MCMRXP1 | N25 |
| EMIFCE2 | J2 | GPIO01 | R25 | MCMRXP2 | K25 |
| EMIFCE3 | M5 | GPIO02 | R23 | MCMRXP3 | L24 |
| EMIFD00 | U4 | GPIO03 | U25 | MCMRXPCLK | E24 |
| EMIFD01 | U5 | GPIO04 | T23 | MCMRXPMDAT | D24 |
| EMIFD02 | V1 | GPIO05 | U24 | MCMTXFLCLK | E25 |
| EMIFD03 | V2 | GPIO06 | T22 | MCMTXFLDAT | D25 |
| EMIFD04 | V3 | GPIO07 | R21 | MCMTXN0 | P22 |
| EMIFD05 | V4 | GPIO08 | U22 | MCMTXN1 | N21 |
| EMIFD06 | W1 | GPIO09 | U23 | MCMTXN2 | K22 |
| EMIFD07 | V5 | GPIO10 | V23 | MCMTXN3 | J21 |
| EMIFD08 | W2 | GPIO11 | U21 | MCMTXP0 | N22 |
| EMIFD09 | Y1 | GPIO12 | T21 | MCMTXP1 | M21 |
| EMIFD10 | W4 | GPIO13 | V22 | MCMTXP2 | L22 |
| EMIFD11 | Y2 | GPIO14 | W21 | MCMTXP3 | K21 |
| EMIFD12 | W5 | GPIO15 | V21 | MCMTXPCLK | F24 |
| EMIFD13 | AA1 | GPIO16 † | AD20 | MCMTXPMDAT | G24 |
| EMIFD14 | AB1 | GPIO17 † | AE21 | MDCLK | AA16 |

Table 3-43. Terminal Functions — By Signal Name (continued)

| SIGNAL NAME | BALL NUMBER | SIGNAL NAME | BALL NUMBER | SIGNAL NAME | BALL NUMBER |
|-------------------------------|-------------|---------------|-------------|------------------|----------------|
| MDIO | AB16 | RSV12 | Y5 | UPP_CH0_WAIT † | T1 |
| $\overline{\text{NMI}}$ | H1 | RSV13 | Y4 | UPP_CH1_CLK † | T5 |
| PCIECLKN | AE15 | RSV14 | F21 | UPP_CH1_ENABLE † | U2 |
| PCIECLKP | AD14 | RSV15 | G21 | UPP_CH1_START † | U1 |
| PCIERXN0 | AE12 | RSV16 | J20 | UPP_CH1_WAIT † | U3 |
| PCIERXN1 | AD10 | RSV17 | AA7 | UPPD00 † | U4 |
| PCIERXP0 | AE11 | RSV18 | AA11 | UPPD01 † | U5 |
| PCIERXP1 | AD11 | RSV19 | AB3 | UPPD02 † | V1 |
| PCIESSEN ‡ | AD20 | RSV20 | F22 | UPPD03 † | V2 |
| PCIETXN0 | AC12 | RSV21 | D23 | UPPD04 † | V3 |
| PCIETXN1 | AB11 | SCL | AA17 | UPPD05 † | V4 |
| PCIETXP0 | AC11 | SDA | AA18 | UPPD06 † | W1 |
| PCIETXP1 | AB10 | SGMIIORXN | AE2 | UPPD07 † | V5 |
| $\overline{\text{POR}}$ | Y18 | SGMIIORXP | AE3 | UPPD08 † | W2 |
| PTV15 | F15 | SGMIIOTXN | AC2 | UPPD09 † | Y1 |
| $\overline{\text{RESETFULL}}$ | J4 | SGMIIOTXP | AC3 | UPPD10 † | W4 |
| $\overline{\text{RESETSTAT}}$ | H5 | SPICLK | AA13 | UPPD11 † | Y2 |
| $\overline{\text{RESET}}$ | H4 | SPIDIN | AB14 | UPPD12 † | W5 |
| RIORXN0 | AE9 | SPIDOUT | AB13 | UPPD13 † | AA1 |
| RIORXN1 | AD8 | SPISCS0 | AA12 | UPPD14 † | AB1 |
| RIORXN2 | AE5 | SPISCS1 | AA14 | UPPD15 † | AA2 |
| RIORXN3 | AD4 | SRIOSGMIICLK | AE14 | UPPXD00 † | K1 |
| RIORXP0 | AE8 | SRIOSGMIICLK | AD13 | UPPXD01 † | M3 |
| RIORXP1 | AD7 | SYSCLKOUT | AA19 | UPPXD02 † | L2 |
| RIORXP2 | AE6 | TCK | AD17 | UPPXD03 † | P5 |
| RIORXP3 | AD5 | TDI | AE17 | UPPXD04 † | L1 |
| RIOTXN0 | AC9 | TDO | AD19 | UPPXD05 † | P4 |
| RIOTXN1 | AB7 | TIMIO | AD20 | UPPXD06 † | M2 |
| RIOTXN2 | AC5 | TIMI1 | AE21 | UPPXD07 † | M1 |
| RIOTXN3 | AB4 | TIMO0 | AC19 | UPPXD08 † | N2 |
| RIOTXP0 | AC8 | TIMO1 | AE20 | UPPXD09 † | P3 |
| RIOTXP1 | AB8 | TMS | AE18 | UPPXD10 † | N1 |
| RIOTXP2 | AC6 | TRST | AB19 | UPPXD11 † | P2 |
| RIOTXP3 | AB5 | UARTCTS | AC17 | UPPXD12 † | P1 |
| RSV01 | AA22 | UARTCTS1 | AE16 | UPPXD13 † | R5 |
| RSV02 | J3 | UARTRTS | AB17 | UPPXD14 † | R3 |
| RSV03 | H2 | UARTRTS1 | AD15 | UPPXD15 † | R4 |
| RSV04 | AC18 | UARTRXD | AB15 | VCNTL0 | E22 |
| RSV05 | AB18 | UARTRXD1 | AC14 | VCNTL1 | E23 |
| RSV06 | B23 | UARTTXD | AA15 | VCNTL2 | F23 |
| RSV07 | A23 | UARTTXD1 | AC15 | VCNTL3 | G23 |
| RSV08 | Y19 | UPP_2XTXCLK † | M4 | VDDR1 | M20 |
| RSV09 | C23 | UPP_CH0_CLK † | R2 | VDDR2 | AA9 |
| RSV0A | G19 | UPP_CH0_ | T4 | VDDR3 | AA3 |
| RSV0B | G20 | ENABLE † | | VDDR4 | AA5 |
| RSV10 | G22 | UPP_CH0_ | R1 | VDDT1 | K19, L20, M19, |
| RSV11 | H22 | START † | | | N20 |

Table 3-43. Terminal Functions — By Signal Name (continued)

| SIGNAL NAME | BALL NUMBER | SIGNAL NAME | BALL NUMBER | SIGNAL NAME | BALL NUMBER |
|-------------|--|-------------|-------------|-------------|-------------|
| VDDT2 | W8, W10, W12, Y7, Y9, Y11 | | | | |
| VDDT1 | N20 | | | | |
| VDDT2 | W10 | | | | |
| VDDT2 | W12 | | | | |
| VDDT2 | Y7 | | | | |
| VDDT2 | Y9 | | | | |
| VDDT2 | Y11 | | | | |
| VREFSSTL | E12 | | | | |
| VSS | A1, A10, A25, B6, B17, C2, C4, C8, C13, C15, C19, D21, E11, F3, F6, F8, F10, F12, F14, F16, F18, G7, G9, G11, G13, G15, G17, H6, H8, H10, H12, H14, H16, H18, H20, H21, H23, H25, J7, J9, J11, J13, J15, J17, J19, J22, J23, J24, K2, K6, K8, K10, K12, K14, K16, K18, K20, K23, L7, L9, L11, L13, L15, L17, L19, L21, L23, L25, M6, M8, M10, M12, M14, M16, M18, M22, M23, M24, N4, N7, N9, N11, N13, N15, N17, N19, N23, P6, P8, P10, P12, P14, P16, P18, P20, P21, P23, P25, R7, R8, R9, R11, R13, R15, R17, R19, R22, R24, T2, T6, T8, T10, T12, T14, T16, T18, T20, U7, U9, U11, U13, U15, U17, U19, V6, V8, V10, V12, V14, V16, V18, V20, W3, W7, W9, W11, W13, W15, W17, W19, Y6, Y8, Y10, Y12, Y14, Y16, AA4, AA6, AA8, AA10, AB2, AB6, AB9, AB12, AB20, AB24, AC1, AC4, AC7, AC10, AC13, AD1, AD2, AD3, AD6, AD9, AD12, AD16, AE1, AE4, AE7, AE10, AE13, AE25 | | | | |

Table 3-44. Terminal Functions — By Ball Number

| BALL NUMBER | SIGNAL NAME | BALL NUMBER | SIGNAL NAME | BALL NUMBER | SIGNAL NAME |
|-------------|------------------------------|-------------|-----------------------------|-------------|---------------------------------|
| A1 | VSS | B23 | RSV06 | D20 | DDRA08 |
| A2 | DDRDQS3P | B24 | MCMRXFLCLK | D21 | VSS |
| A3 | DDRD20 | B25 | MCMCLKN | D22 | DDRSLRATE1 |
| A4 | DDRDQS2N | C1 | DDRD27 | D23 | RSV21 |
| A5 | DDRD11 | C2 | VSS | D24 | MCMRXPMDAT |
| A6 | DDRD10 | C3 | DDRD30 | D25 | MCMTXFLDAT |
| A7 | DDRDQS1N | C4 | VSS | E1 | DDRDQM3 |
| A8 | DDRDQM0 | C5 | DDRD22 | E2 | DDRD24 |
| A9 | DDRD00 | C6 | DVDD15 | E3 | DDRD31 |
| A10 | VSS | C7 | DDRD13 | E4 | DDRD19 |
| A11 | DDRD06 | C8 | VSS | E5 | DDRD16 |
| A12 | DDRCB03 | C9 | DDRD01 | E6 | DDRD08 |
| A13 | DDRDQS8N | C10 | DDRDQS0N | E7 | DDRDQM1 |
| A14 | DDRCLKOUTP0 | C11 | DDRCB02 | E8 | DDRD09 |
| A15 | $\overline{\text{DDRRAS}}$ | C12 | DDRDQM8 | E9 | DDRD04 |
| A16 | DDRCKE0 | C13 | VSS | E10 | DDRD05 |
| A17 | DDRA05 | C14 | $\overline{\text{DDRCCE1}}$ | E11 | VSS |
| A18 | DDRA07 | C15 | VSS | E12 | VREFSSTL |
| A19 | DDRA01 | C16 | DDRA06 | E13 | $\overline{\text{DDRWE}}$ |
| A20 | DDRCKE1 | C17 | DVDD15 | E14 | DDRODT0 |
| A21 | DDRCLKOUTP1 | C18 | DDRBA0 | E15 | DDRA03 |
| A22 | DDRCLKP | C19 | VSS | E16 | DDRA02 |
| A23 | RSV07 | C20 | DDRA13 | E17 | DDRA15 |
| A24 | DVDD18 | C21 | DVDD15 | E18 | DDRA14 |
| A25 | VSS | C22 | DDRSLRATE0 | E19 | DDRA10 |
| B1 | DDRD26 | C23 | RSV09 | E20 | DDRA09 |
| B2 | DDRDQS3N | C24 | MCMRXFLDAT | E21 | DVDD18 |
| B3 | DDRD17 | C25 | MCMCLKP | E22 | VCNTL0 |
| B4 | DDRDQS2P | D1 | DDRD28 | E23 | VCNTL1 |
| B5 | DDRD21 | D2 | DVDD15 | E24 | MCMRXPMCLK |
| B6 | VSS | D3 | DDRD29 | E25 | MCMTXFLCLK |
| B7 | DDRDQS1P | D4 | DVDD15 | F1 | $\overline{\text{LRESETNMIEN}}$ |
| B8 | DDRD15 | D5 | DDRD23 | F2 | DDRD25 |
| B9 | DDRD03 | D6 | DDRD12 | F3 | VSS |
| B10 | DVDD15 | D7 | DDRD14 | F4 | DDRD18 |
| B11 | DDRD07 | D8 | DVDD15 | F5 | DDRDQM2 |
| B12 | DDRCB01 | D9 | DDRD02 | F6 | VSS |
| B13 | DDRDQS8P | D10 | DDRDQS0P | F7 | DVDD15 |
| B14 | DDRCLKOUTN0 | D11 | DDRCB00 | F8 | VSS |
| B15 | $\overline{\text{DDRCCE0}}$ | D12 | DDRODT1 | F9 | DVDD15 |
| B16 | $\overline{\text{DDRRESET}}$ | D13 | DVDD15 | F10 | VSS |
| B17 | VSS | D14 | $\overline{\text{DDRCAS}}$ | F11 | DVDD15 |
| B18 | DDRA04 | D15 | DVDD15 | F12 | VSS |
| B19 | DDRBA2 | D16 | DDRA00 | F13 | DVDD15 |
| B20 | DDRA11 | D17 | DDRBA1 | F14 | VSS |
| B21 | DDRCLKOUTN1 | D18 | DDRA12 | F15 | PTV15 |
| B22 | DDRCLKN | D19 | DVDD15 | F16 | VSS |

Table 3-44. Terminal Functions — By Ball Number (continued)

| BALL NUMBER | SIGNAL NAME | BALL NUMBER | SIGNAL NAME | BALL NUMBER | SIGNAL NAME |
|-------------|---------------|-------------|-------------|-------------|-------------|
| F17 | DVDD15 | H14 | VSS | K10 | VSS |
| F18 | VSS | H15 | CVDD | K11 | CVDD |
| F19 | DVDD15 | H16 | VSS | K12 | VSS |
| F20 | AVDDA2 | H17 | CVDD | K13 | CVDD |
| F21 | RSV14 | H18 | VSS | K14 | VSS |
| F22 | RSV20 | H19 | DVDD18 | K15 | CVDD |
| F23 | VCNTL2 | H20 | VSS | K16 | VSS |
| F24 | MCMTXPMCLK | H21 | VSS | K17 | CVDD1 |
| F25 | MCMREFCLKOUTN | H22 | RSV11 | K18 | VSS |
| G1 | EMIFCE1 | H23 | VSS | K19 | VDDT1 |
| G2 | HOUT | H24 | DVDD18 | K20 | VSS |
| G3 | DVDD18 | H25 | VSS | K21 | MCMTXP3 |
| G4 | LRESET | J1 | EMIFBE0 | K22 | MCMTXN2 |
| G5 | CORESEL1 | J2 | EMIFCE2 | K23 | VSS |
| G6 | DVDD18 | J3 | RSV02 | K24 | MCMRXN3 |
| G7 | VSS | J4 | RESETFULL | K25 | MCMRXP2 |
| G8 | DVDD15 | J5 | CORESEL0 | L1 | EMIFA04 |
| G9 | VSS | J6 | DVDD18 | L1 | UPPXD04 † |
| G10 | DVDD15 | J7 | VSS | L2 | EMIFA02 |
| G11 | VSS | J8 | CVDD1 | L2 | UPPXD02 † |
| G12 | DVDD15 | J9 | VSS | L3 | EMIFBE1 |
| G13 | VSS | J10 | CVDD | L4 | EMIFOE |
| G14 | DVDD15 | J11 | VSS | L5 | EMIFRNW |
| G15 | VSS | J12 | CVDD | L6 | DVDD18 |
| G16 | DVDD15 | J13 | VSS | L7 | VSS |
| G17 | VSS | J14 | CVDD | L8 | CVDD |
| G18 | DVDD15 | J15 | VSS | L9 | VSS |
| G19 | RSV0A | J16 | CVDD | L10 | CVDD |
| G20 | RSV0B | J17 | VSS | L11 | VSS |
| G21 | RSV15 | J18 | CVDD1 | L12 | CVDD |
| G22 | RSV10 | J19 | VSS | L13 | VSS |
| G23 | VCNTL3 | J20 | RSV16 | L14 | CVDD |
| G24 | MCMTXPMDAT | J21 | MCMTXN3 | L15 | VSS |
| G25 | MCMREFCLKOUTP | J22 | VSS | L16 | CVDD |
| H1 | NMI | J23 | VSS | L17 | VSS |
| H2 | RSV03 | J24 | VSS | L18 | CVDD |
| H3 | BOOTCOMPLETE | J25 | MCMRXN2 | L19 | VSS |
| H4 | RESET | K1 | EMIFA00 | L20 | VDDT1 |
| H5 | RESETSTAT | K1 | UPPXD00 † | L21 | VSS |
| H6 | VSS | K2 | VSS | L22 | MCMTXP2 |
| H7 | DVDD18 | K3 | DVDD18 | L23 | VSS |
| H8 | VSS | K4 | EMIFWE | L24 | MCMRXP3 |
| H9 | CVDD | K5 | EMIFCE0 | L25 | VSS |
| H10 | VSS | K6 | VSS | M1 | EMIFA07 |
| H11 | CVDD | K7 | DVDD18 | M1 | UPPXD07 † |
| H12 | VSS | K8 | VSS | M2 | EMIFA06 |
| H13 | CVDD | K9 | CVDD1 | M2 | UPPXD06 † |

Table 3-44. Terminal Functions — By Ball Number (continued)

| BALL NUMBER | SIGNAL NAME | BALL NUMBER | SIGNAL NAME | BALL NUMBER | SIGNAL NAME |
|-------------|-------------|-------------|-----------------|-------------|------------------|
| M3 | EMIFA01 | N21 | MCMTXN1 | R8 | VSS |
| M3 | UPPXD01 † | N22 | MCMTXP0 | R9 | VSS |
| M4 | EMIFWAIT1 | N23 | VSS | R10 | CVDD |
| M4 | UPP2TXCLK † | N24 | MCMRXP0 | R11 | VSS |
| M5 | EMIFCE3 | N25 | MCMRXP1 | R12 | CVDD |
| M6 | VSS | P1 | EMIFA12 | R13 | VSS |
| M7 | DVDD18 | P1 | UPPXD12 † | R14 | CVDD |
| M8 | VSS | P2 | EMIFA11 | R15 | VSS |
| M9 | CVDD | P2 | UPPXD11 † | R16 | CVDD |
| M10 | VSS | P3 | EMIFA09 | R17 | VSS |
| M11 | CVDD | P3 | UPPXD09 † | R18 | CVDD |
| M12 | VSS | P4 | EMIFA05 | R19 | VSS |
| M13 | CVDD | P4 | UPPXD05 † | R20 | DVDD18 |
| M14 | VSS | P5 | EMIFA03 | R21 | GPIO07 |
| M15 | CVDD | P5 | UPPXD03 † | R21 | BOOTMODE06 † |
| M16 | VSS | P6 | VSS | R22 | VSS |
| M17 | CVDD | P7 | DVDD18 | R23 | GPIO02 |
| M18 | VSS | P8 | VSS | R23 | BOOTMODE01 † |
| M19 | VDDT1 | P9 | CVDD | R24 | VSS |
| M20 | VDDR1 | P10 | VSS | R25 | GPIO01 |
| M21 | MCMTXP1 | P11 | CVDD | R25 | BOOTMODE00 † |
| M22 | VSS | P12 | VSS | T1 | EMIFA19 |
| M23 | VSS | P13 | CVDD | T1 | UPP_CH0_WAIT † |
| M24 | VSS | P14 | VSS | T2 | VSS |
| M25 | MCMRXN1 | P15 | CVDD | T3 | DVDD18 |
| N1 | EMIFA10 | P16 | VSS | T4 | EMIFA18 |
| N1 | UPPXD10 † | P17 | CVDD | T4 | UPP_CH0_ENABLE † |
| N2 | EMIFA08 | P18 | VSS | T5 | EMIFA20 |
| N2 | UPPXD08 † | P19 | CVDD | T5 | UPP_CH1_CLK † |
| N3 | DVDD18 | P20 | VSS | T6 | VSS |
| N4 | VSS | P21 | VSS | T7 | DVDD18 |
| N5 | EMIFWAIT0 | P22 | MCMTXN0 | T8 | VSS |
| N6 | DVDD18 | P23 | VSS | T9 | CVDD1 |
| N7 | VSS | P24 | MCMRXN0 | T10 | VSS |
| N8 | CVDD | P25 | VSS | T11 | CVDD |
| N9 | VSS | R1 | EMIFA17 | T12 | VSS |
| N10 | CVDD | R1 | UPP_CH0_START † | T13 | CVDD |
| N11 | VSS | R2 | EMIFA16 | T14 | VSS |
| N12 | CVDD | R2 | UPP_CH0_CLK † | T15 | CVDD |
| N13 | VSS | R3 | EMIFA14 | T16 | VSS |
| N14 | CVDD | R3 | UPPXD14 † | T17 | CVDD1 |
| N15 | VSS | R4 | EMIFA15 | T18 | VSS |
| N16 | CVDD | R4 | UPPXD15 † | T19 | DVDD18 |
| N17 | VSS | R5 | EMIFA13 | T20 | VSS |
| N18 | CVDD | R5 | UPPXD13 † | T21 | GPIO12 |
| N19 | VSS | R6 | DVDD18 | T21 | BOOTMODE11 † |
| N20 | VDDT1 | R7 | VSS | T22 | GPIO06 |

Table 3-44. Terminal Functions — By Ball Number (continued)

| BALL NUMBER | SIGNAL NAME | BALL NUMBER | SIGNAL NAME | BALL NUMBER | SIGNAL NAME |
|-------------|------------------|-------------|---------------|-------------|---------------|
| T22 | BOOTMODE05 † | V3 | UPPD04 † | W16 | DVDD18 |
| T23 | GPIO04 | V4 | EMIFD05 | W17 | VSS |
| T23 | BOOTMODE03 † | V4 | UPPD05 † | W18 | DVDD18 |
| T24 | DVDD18 | V5 | EMIFD07 | W19 | VSS |
| T25 | GPIO00 | V5 | UPPD07 † | W20 | DVDD18 |
| T25 | LENDIAN † | V6 | VSS | W21 | GPIO14 † |
| U1 | EMIFA21 | V7 | DVDD18 | W21 | PCIESSMODE0 † |
| U1 | UPP_CH1_START † | V8 | VSS | W22 | EMU08 |
| U2 | EMIFA22 | V9 | CVDD | W23 | EMU03 |
| U2 | UPP_CH1_ENABLE † | V10 | VSS | W24 | EMU04 |
| | | V11 | CVDD | W25 | EMU02 |
| U3 | EMIFA23 | V12 | VSS | Y1 | EMIFD09 |
| U3 | UPP_CH1_WAIT † | V13 | CVDD | Y1 | UPPD09 † |
| U4 | EMIFD00 | V14 | VSS | Y2 | EMIFD11 |
| U4 | UPPD00 † | V15 | CVDD | Y2 | UPPD11 † |
| U5 | EMIFD01 | V16 | VSS | Y3 | DVDD18 |
| U5 | UPPD01 † | V17 | CVDD | Y4 | RSV13 |
| U6 | DVDD18 | V18 | VSS | Y5 | RSV12 |
| U7 | VSS | V19 | DVDD18 | Y6 | VSS |
| U8 | CVDD1 | V20 | VSS | Y7 | VDDT2 |
| U9 | VSS | V21 | GPIO15 | Y8 | VSS |
| U10 | CVDD | V21 | PCIESSMODE1 † | Y9 | VDDT2 |
| U11 | VSS | V22 | GPIO13 | Y10 | VSS |
| U12 | CVDD | V22 | BOOTMODE12 † | Y11 | VDDT2 |
| U13 | VSS | V23 | GPIO10 | Y12 | VSS |
| U14 | CVDD | V23 | BOOTMODE09 † | Y13 | DVDD18 |
| U15 | VSS | V24 | EMU00 | Y14 | VSS |
| U16 | CVDD | V25 | EMU01 | Y15 | AVDDA1 |
| U17 | VSS | W1 | EMIFD06 | Y16 | VSS |
| U18 | CVDD1 | W1 | UPPD06 † | Y17 | DVDD18 |
| U19 | VSS | W2 | EMIFD08 | Y18 | POR |
| U20 | DVDD18 | W2 | UPPD08 † | Y19 | RSV08 |
| U21 | GPIO11 | W3 | VSS | Y20 | CLKX0 |
| U21 | BOOTMODE10 † | W4 | EMIFD10 | Y21 | EMU18 |
| U22 | GPIO08 | W4 | UPPD10 † | Y22 | EMU09 |
| U22 | BOOTMODE07 † | W5 | EMIFD12 | Y23 | EMU07 |
| U23 | GPIO09 | W5 | UPPD12 † | Y24 | EMU06 |
| U23 | BOOTMODE08 † | W6 | DVDD18 | Y25 | EMU05 |
| U24 | GPIO05 | W7 | VSS | AA1 | EMIFD13 |
| U24 | BOOTMODE04 † | W8 | VDDT2 | AA1 | UPPD13 † |
| U25 | GPIO03 | W9 | VSS | AA2 | EMIFD15 |
| U25 | BOOTMODE02 † | W10 | VDDT2 | AA2 | UPPD15 † |
| V1 | EMIFD02 | W11 | VSS | AA3 | VDDR3 |
| V1 | UPPD02 † | W12 | VDDT2 | AA4 | VSS |
| V2 | EMIFD03 | W13 | VSS | AA5 | VDDR4 |
| V2 | UPPD03 † | W14 | DVDD18 | AA6 | VSS |
| V3 | EMIFD04 | W15 | VSS | AA7 | RSV17 |

Table 3-44. Terminal Functions — By Ball Number (continued)

| BALL NUMBER | SIGNAL NAME | BALL NUMBER | SIGNAL NAME | BALL NUMBER | SIGNAL NAME |
|-------------|-------------|-------------|---------------|-------------|---------------|
| AA8 | VSS | AB22 | EMU15 | AD15 | UARTRTS1 |
| AA9 | VDDR2 | AB23 | DVDD18 | AD15 | GPIO27 † |
| AA10 | VSS | AB24 | VSS | AD16 | VSS |
| AA11 | RSV18 | AB25 | EMU12 | AD17 | TCK |
| AA12 | SPISCS0 | AC1 | VSS | AD18 | CORECLKP |
| AA12 | GPIO28 † | AC2 | SGMII0TXN | AD19 | TDO |
| AA13 | SPICLK | AC3 | SGMII0TXP | AD20 | TIMIO |
| AA14 | SPISCS1 | AC4 | VSS | AD20 | GPIO16 † |
| AA14 | GPIO29 † | AC5 | RIOTXN2 | AD20 | PCIESSEN ‡ |
| AA15 | UARTTXD | AC6 | RIOTXP2 | AD21 | DR1 |
| AA15 | GPIO21 † | AC7 | VSS | AD22 | FSR1 |
| AA16 | MDCLK | AC8 | RIOTXP0 | AD23 | CLKR1 |
| AA17 | SCL | AC9 | RIOTXN0 | AD24 | FSR0 |
| AA18 | SDA | AC10 | VSS | AD25 | EMU16 |
| AA19 | SYSCCLKOUT | AC11 | PCIETXP0 | AE1 | VSS |
| AA20 | FSX0 | AC12 | PCIETXN0 | AE2 | SGMII0RXN |
| AA21 | CLKR0 | AC13 | VSS | AE3 | SGMII0RXP |
| AA22 | RSV01 | AC14 | UARTRXD1 | AE4 | VSS |
| AA23 | EMU14 | AC14 | GPIO24 † | AE5 | RIORXN2 |
| AA24 | EMU10 | AC15 | UARTTXD1 | AE6 | RIORXP2 |
| AA25 | EMU11 | AC15 | GPIO25 † | AE7 | VSS |
| AB1 | EMIFD14 | AC16 | DVDD18 | AE8 | RIORXP0 |
| AB1 | UPPD14 † | AC17 | UARTCTS | AE9 | RIORXN0 |
| AB2 | VSS | AC17 | GPIO22 † | AE10 | VSS |
| AB3 | RSV19 | AC18 | RSV04 | AE11 | PCIERXP0 |
| AB4 | RIOTXN3 | AC19 | TIMO0 | AE12 | PCIERXN0 |
| AB5 | RIOTXP3 | AC19 | GPIO18 † | AE13 | VSS |
| AB6 | VSS | AC20 | DVDD18 | AE14 | SRIOSGMIICLKN |
| AB7 | RIOTXN1 | AC21 | CLKS1 | AE15 | PCIECLKN |
| AB8 | RIOTXP1 | AC22 | DX0 | AE16 | UARTCTS1 |
| AB9 | VSS | AC23 | CLKS0 | AE16 | GPIO26 † |
| AB10 | PCIETXP1 | AC24 | EMU17 | AE17 | TDI |
| AB11 | PCIETXN1 | AC25 | EMU13 | AE18 | TMS |
| AB12 | VSS | AD1 | VSS | AE19 | CORECLKN |
| AB13 | SPIDOUT | AD2 | VSS | AE20 | TIMO1 |
| AB13 | GPIO31 † | AD3 | VSS | AE20 | GPIO19 † |
| AB14 | SPIDIN | AD4 | RIORXN3 | AE21 | TIMI1 |
| AB14 | GPIO30 † | AD5 | RIORXP3 | AE21 | GPIO17 † |
| AB15 | UARTRXD | AD6 | VSS | AE22 | DX1 |
| AB15 | GPIO20 † | AD7 | RIORXP1 | AE23 | FSX1 |
| AB16 | MDIO | AD8 | RIORXN1 | AE24 | CLKX1 |
| AB17 | UARTRTS | AD9 | VSS | AE25 | VSS |
| AB17 | GPIO23 † | AD10 | PCIERXN1 | | |
| AB18 | RSV05 | AD11 | PCIERXP1 | | |
| AB19 | TRST | AD12 | VSS | | |
| AB20 | VSS | AD13 | SRIOSGMIICLKP | | |
| AB21 | DR0 | AD14 | PCIECLKP | | |

3.10 Development and Support

3.10.1 Development Support

In case the customer would like to develop their own features and software on the C665x device, TI offers an extensive line of development tools for the TMS320C6000™ DSP platform, including tools to evaluate the performance of the processors, generate code, develop algorithm implementations, and fully integrate and debug software and hardware modules. The tool's support documentation is electronically available within the Code Composer Studio™ Integrated Development Environment (IDE).

The following products support development of C6000™ DSP-based applications:

- **Software Development Tools:**
 - Code Composer Studio™ Integrated Development Environment (IDE), including Editor C/C++/Assembly Code Generation, and Debug plus additional development tools.
 - Scalable, Real-Time Foundation Software (DSP/BIOS™), which provides the basic run-time target software needed to support any DSP application.
- **Hardware Development Tools:**
 - Extended Development System (XDS™) Emulator (supports C6000™ DSP multiprocessor system debug)
 - EVM (Evaluation Module)

3.10.2 Device Support

3.10.2.1 Device and Development-Support Tool Nomenclature

To designate the stages in the product development cycle, TI assigns prefixes to the part numbers of all DSP devices and support tools. Each DSP commercial family member has one of three prefixes: TMX, TMP, or TMS (e.g., TMX320CMH). Texas Instruments recommends two of three possible prefix designators for its support tools: TMDX and TMDS. These prefixes represent evolutionary stages of product development from engineering prototypes (TMX/TMDX) through fully qualified production devices/tools (TMS/TMDS).

Device development evolutionary flow:

- **TMX:** Experimental device that is not necessarily representative of the final device's electrical specifications
- **TMP:** Final silicon die that conforms to the device's electrical specifications but has not completed quality and reliability verification
- **TMS:** Fully qualified production device

Support tool development evolutionary flow:

- **TMDX:** Development-support product that has not yet completed Texas Instruments internal qualification testing.
- **TMDS:** Fully qualified development-support product

TMX and TMP devices and TMDX development-support tools are shipped with the following disclaimer:

- **"Developmental product is intended for internal evaluation purposes."**

TMS devices and TMDS development-support tools have been characterized fully, and the quality and reliability of the device have been demonstrated fully. TI's standard warranty applies.

Predictions show that prototype devices (TMX or TMP) have a greater failure rate than the standard production devices. Texas Instruments recommends that these devices not be used in any production system because their expected end-use failure rate still is undefined. Only qualified production devices are to be used.

TI device nomenclature also includes a suffix with the device family name. This suffix indicates the package type (for example, CZH), the temperature range (for example, blank is the default case temperature range), and the device speed range, in Megahertz (for example, blank is 1000 MHz [1 GHz]).

For device part numbers and further ordering information for C665x in the CZH or GZH package type, see the TI website www.ti.com or contact your TI sales representative.

Figure 3-31 provides a legend for reading the complete device name for any C66x KeyStone device.

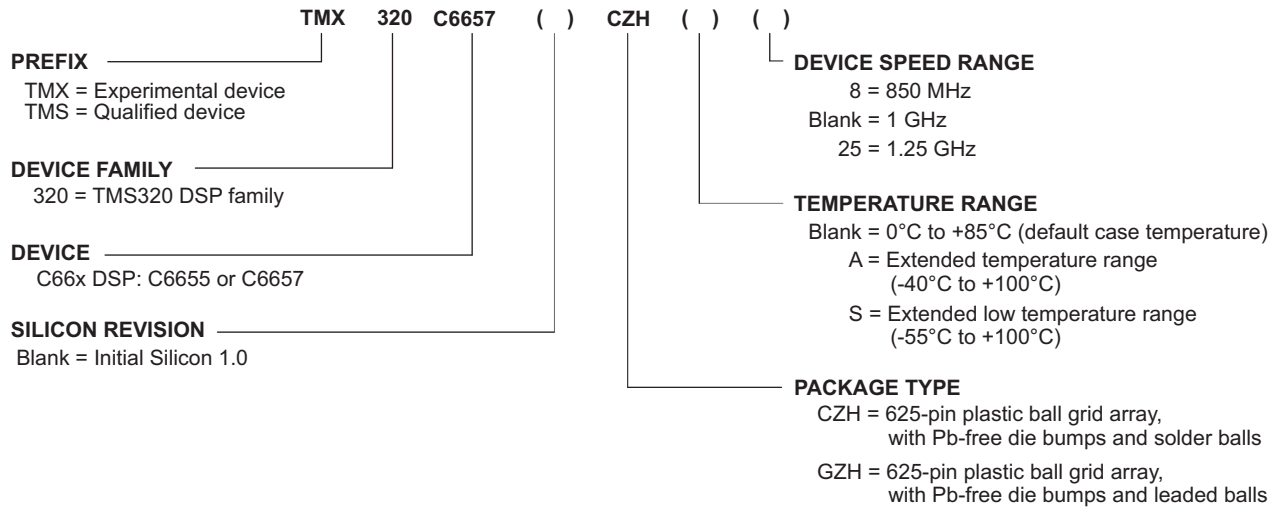


Figure 3-31. C66x DSP Device Nomenclature (including the C665x)

3.11 Related Documentation from Texas Instruments

These documents describe the C665x Fixed and Floating-Point Digital Signal Processor. Copies of these documents are available on the Internet at www.ti.com.

| | |
|---|-------------------------|
| 64-bit Timer (Timer 64) for KeyStone Devices User's Guide | SPRUGV5 |
| Bootloader for the C66x DSP User's Guide | SPRUGY5 |
| C66x CorePac User's Guide | SPRUGW0 |
| C66x CPU and Instruction Set Reference Guide | SPRUGH7 |
| C66x DSP Cache User's Guide | SPRUGY8 |
| DDR3 Design Guide for KeyStone Devices | SPRAB11 |
| DDR3 Memory Controller for KeyStone Devices User's Guide | SPRUGV8 |
| DSP Power Consumption Summary for KeyStone Devices | SPRABL4 |
| Debug and Trace for KeyStone I Devices User's Guide | SPRUGZ2 |
| Emulation and Trace Headers Technical Reference | SPRU655 |
| Enhanced Direct Memory Access 3 (EDMA3) for KeyStone Devices User's Guide | SPRUGS5 |
| External Memory Interface (EMIF16) for KeyStone Devices User's Guide | SPRUGZ3 |
| General Purpose Input/Output (GPIO) for KeyStone Devices User's Guide | SPRUGV1 |
| Ethernet Media Access Controller (EMAC) for KeyStone Devices User's Guide | SPRUHH1 |
| Hardware Design Guide for KeyStone Devices | SPRAB12 |
| HyperLink for KeyStone Devices User's Guide | SPRUGW8 |
| <i>Inter Integrated Circuit (I²C) for KeyStone Devices User's Guide</i> | SPRUGV3 |
| Chip Interrupt Controller (CIC) for KeyStone Devices User's Guide | SPRUGW4 |
| Memory Protection Unit (MPU) for KeyStone Devices User's Guide | SPRUGW5 |
| Multichannel Buffered Serial Port (McBSP) for KeyStone Devices User's Guide | |
| Multicore Navigator for KeyStone Devices User's Guide | SPRUGR9 |
| Multicore Shared Memory Controller (MSMC) for KeyStone Devices User's Guide | SPRUGW7 |
| Peripheral Component Interconnect Express (PCIe) for KeyStone Devices User's Guide | SPRUGS6 |
| Phase Locked Loop (PLL) for KeyStone Devices User's Guide | SPRUGV2 |
| Power Sleep Controller (PSC) for KeyStone Devices User's Guide | SPRUGV4 |
| Semaphore2 Hardware Module for KeyStone Devices User's Guide | SPRUGS3 |
| Serial Peripheral Interface (SPI) for KeyStone Devices User's Guide | SPRUGP2 |
| Serial RapidIO (SRIO) for KeyStone Devices User's Guide | SPRUGW1 |
| Turbo Decoder Coprocessor 3 (TCP3d) for KeyStone Devices User's Guide | SPRUGS0 |
| Universal Asynchronous Receiver/Transmitter (UART) for KeyStone Devices User's Guide | SPRUGP1 |
| Universal Parallel Port (uPP) for KeyStone Devices User's Guide | |
| Using Advanced Event Triggering to Debug Real-Time Problems in High Speed Embedded Microprocessor Systems | SPRA387 |
| Using Advanced Event Triggering to Find and Fix Intermittent Real-Time Bugs | SPRA753 |
| Using IBIS Models for Timing Analysis | SPRA839 |
| Viterbi Coprocessor (VCP2) for KeyStone Devices User's Guide | SPRUGV6 |

4 Device Configuration

On the C665x device, certain device configurations like boot mode and endianness, are selected at device power-on reset. The status of the peripherals (enabled/disabled) is determined after device power-on reset.

4.1 Device Configuration at Device Reset

Table 4-1 describes the device configuration pins. The logic level is latched at power-on reset to determine the device configuration. The logic level on the device configuration pins can be set by using external pullup/pulldown resistors or by using some control device (e.g., FPGA/CPLD) to intelligently drive these pins. When using a control device, care should be taken to ensure there is no contention on the lines when the device is out of reset. The device configuration pins are sampled during power-on reset and are driven after the reset is removed. To avoid contention, the control device must stop driving the device configuration pins of the DSP. And when driving by a control device, the control device must be fully powered and out of reset itself and driving the pins before the DSP can be taken out of reset.

Also, please note that most of the device configuration pins are shared with other function pins (LENDIAN/GPIO[0], BOOTMODE[12:0]/GPIO[13:1], PCIESSMODE[1:0]/GPIO[15:14] and PCIESSSEN/TIM10), some time must be given following the rising edge of reset in order to drive these device configuration input pins before they assume an output state (those GPIO pins should not become outputs during boot). Another caution that needs to be noted is that systems using TIM10 (pin shared with PCIESSSEN) as a clock input must assure that the clock itself is disabled from the input until after reset is released and a control device is no longer driving that input.

NOTE

If a configuration pin must be routed out from the device and it is not driven (Hi-Z state), the internal pullup/pulldown (IPU/IPD) resistor should not be relied upon. TI recommends the use of an external pullup/pulldown resistor. For more detailed information on pullup/pulldown resistors and situations in which external pullup/pulldown resistors are required, see [Section 4.4](#).

Table 4-1. C665x Device Configuration Pins

| CONFIGURATION PIN | PIN NO. | IPD/IPU ⁽¹⁾ | FUNCTIONAL DESCRIPTION |
|-----------------------------------|--|------------------------|---|
| LENDIAN ⁽¹⁾⁽²⁾ | T25 | IPU | Device endian mode (LENDIAN). <ul style="list-style-type: none"> 0 = Device operates in big endian mode 1 = Device operates in little endian mode |
| BOOTMODE[12:0] ⁽¹⁾⁽²⁾ | R25, R3, U25, T23, U24, T22, R21, U22, U23, V23, U21, T21, V22 | IPD | Method of boot. Some pins may not be used by bootloader and can be used as general purpose config pins. Refer to the <i>Bootloader for the C66x DSP User's Guide</i> (SPRUGY5) for how to determine the device enumeration ID value. |
| PCIESSMODE[1:0] ⁽¹⁾⁽²⁾ | W21, V21 | IPD | PCIe Subsystem mode selection. <ul style="list-style-type: none"> 00 = PCIe in end point mode 01 = PCIe legacy end point (support for legacy INTx) 10 = PCIe in root complex mode 11 = Reserved |
| PCIESSSEN ⁽¹⁾⁽²⁾ | AD20 | IPD | PCIe subsystem enable/disable. <ul style="list-style-type: none"> 0 = PCIe Subsystem is disabled 1 = PCIe Subsystem is enabled |

(1) Internal 100- μ A pulldown or pullup is provided for this terminal. In most systems, a 1-k Ω resistor can be used to oppose the IPD/IPU. For more detailed information on pulldown/pullup resistors and situations in which external pulldown/pullup resistors are required, see [Section 4.4](#).

(2) These signal names are the secondary functions of these pins.

4.2 Peripheral Selection After Device Reset

Several of the peripherals on the C665x are controlled by the Power Sleep Controller (PSC). By default, the PCIe, SRIO, and HyperLink are held in reset and clock-gated. The memories in these modules are also in a low-leakage sleep mode. Software is required to turn these memories on. The software enables the modules (turns on clocks and de-asserts reset) before these modules can be used.

If one of the above modules is used in the selected ROM boot mode, the ROM code will automatically enable the module.

All other modules come up enabled by default and there is no special software sequence to enable. For more detailed information on the PSC usage, see the *Power Sleep Controller (PSC) for KeyStone Devices User's Guide* ([SPRUGV4](#)).

4.3 Device State Control Registers

The C665x device has a set of registers that are used to provide the status or configure certain parts of its peripherals. These registers are shown in [Table 4-2](#).

Table 4-2. Device State Control Registers

| ADDRESS START | ADDRESS END | SIZE | FIELD | DESCRIPTION |
|---------------|-------------|------|--------------------|--|
| 0x02620000 | 0x02620007 | 8B | Reserved | |
| 0x02620008 | 0x02620017 | 16B | Reserved | |
| 0x02620018 | 0x0262001B | 4B | JTAGID | See Section 4.3.3 |
| 0x0262001C | 0x0262001F | 4B | Reserved | |
| 0x02620020 | 0x02620023 | 4B | DEVSTAT | See Section 4.3.1 |
| 0x02620024 | 0x02620037 | 20B | Reserved | |
| 0x02620038 | 0x0262003B | 4B | KICK0 | See Section 4.3.4 |
| 0x0262003C | 0x0262003F | 4B | KICK1 | |
| 0x02620040 | 0x02620043 | 4B | DSP_BOOT_ADDR0 | The boot address for C66x DSP CorePac0 |
| 0x02620044 | 0x02620047 | 4B | DSP_BOOT_ADDR1 | The boot address for C66x DSP CorePac1 (C6657) or Reserved (C6655) |
| 0x02620048 | 0x0262004B | 4B | Reserved | |
| 0x0262004C | 0x0262004F | 4B | Reserved | |
| 0x02620050 | 0x02620053 | 4B | Reserved | |
| 0x02620054 | 0x02620057 | 4B | Reserved | |
| 0x02620058 | 0x0262005B | 4B | Reserved | |
| 0x0262005C | 0x0262005F | 4B | Reserved | |
| 0x02620060 | 0x026200DF | 128B | Reserved | |
| 0x026200E0 | 0x0262010F | 48B | Reserved | |
| 0x02620110 | 0x02620117 | 8B | MACID | See Section 8.18 |
| 0x02620118 | 0x0262012F | 24B | Reserved | |
| 0x02620130 | 0x02620133 | 4B | LRSTNMIPINSTAT_CLR | See Section 4.3.6 |
| 0x02620134 | 0x02620137 | 4B | RESET_STAT_CLR | See Section 4.3.8 |
| 0x02620138 | 0x0262013B | 4B | Reserved | |
| 0x0262013C | 0x0262013F | 4B | BOOTCOMPLETE | See Section 4.3.9 |
| 0x02620140 | 0x02620143 | 4B | Reserved | |
| 0x02620144 | 0x02620147 | 4B | RESET_STAT | See Section 4.3.7 |
| 0x02620148 | 0x0262014B | 4B | LRSTNMIPINSTAT | See Section 4.3.5 |
| 0x0262014C | 0x0262014F | 4B | DEVCFG | See Section 4.3.2 |
| 0x02620150 | 0x02620153 | 4B | PWRSTATECTL | See Section 4.3.10 |
| 0x02620154 | 0x02620157 | 4B | SRIO_SERDES_STS | See Section 3.11 |

Table 4-2. Device State Control Registers (continued)

| ADDRESS START | ADDRESS END | SIZE | FIELD | DESCRIPTION |
|---------------|-------------|------|----------------------|--|
| 0x02620158 | 0x0262015B | 4B | SMGII_SERDES_STS | See Section 3.11 |
| 0x0262015C | 0x0262015F | 4B | PCIE_SERDES_STS | |
| 0x02620160 | 0x02620163 | 4B | HYPERLINK_SERDES_STS | See Section 3.11 |
| 0x02620164 | 0x02620167 | 4B | Reserved | |
| 0x02620168 | 0x0262016B | 4B | Reserved | |
| 0x0262016C | 0x0262016F | 4B | UPP_CLOCK | See Section 4.3.22 |
| 0x02620170 | 0x02620183 | 20B | Reserved | |
| 0x02620184 | 0x0262018F | 12B | Reserved | |
| 0x02620190 | 0x02620193 | 4B | Reserved | |
| 0x02620194 | 0x02620197 | 4B | Reserved | |
| 0x02620198 | 0x0262019B | 4B | Reserved | |
| 0x0262019C | 0x0262019F | 4B | Reserved | |
| 0x026201A0 | 0x026201A3 | 4B | Reserved | |
| 0x026201A4 | 0x026201A7 | 4B | Reserved | |
| 0x026201A8 | 0x026201AB | 4B | Reserved | |
| 0x026201AC | 0x026201AF | 4B | Reserved | |
| 0x026201B0 | 0x026201B3 | 4B | Reserved | |
| 0x026201B4 | 0x026201B7 | 4B | Reserved | |
| 0x026201B8 | 0x026201BB | 4B | Reserved | |
| 0x026201BC | 0x026201BF | 4B | Reserved | |
| 0x026201C0 | 0x026201C3 | 4B | Reserved | |
| 0x026201C4 | 0x026201C7 | 4B | Reserved | |
| 0x026201C8 | 0x026201CB | 4B | Reserved | |
| 0x026201CC | 0x026201CF | 4B | Reserved | |
| 0x026201D0 | 0x026201FF | 48B | Reserved | |
| 0x02620200 | 0x02620203 | 4B | NMIGR0 | See Section 4.3.11 |
| 0x02620204 | 0x02620207 | 4B | NMIGR1 | See Section 4.3.11 (C6657) or Reserved (C6655) |
| 0x02620208 | 0x0262020B | 4B | Reserved | |
| 0x0262020C | 0x0262020F | 4B | Reserved | |
| 0x02620210 | 0x02620213 | 4B | Reserved | |
| 0x02620214 | 0x02620217 | 4B | Reserved | |
| 0x02620218 | 0x0262021B | 4B | Reserved | |
| 0x0262021C | 0x0262021F | 4B | Reserved | |
| 0x02620220 | 0x0262023F | 32B | Reserved | |
| 0x02620240 | 0x02620243 | 4B | IPCGR0 | See Section 4.3.12 |
| 0x02620244 | 0x02620247 | 4B | IPCGR1 | See Section 4.3.12 (C6657) or Reserved (C6655) |
| 0x02620248 | 0x0262024B | 4B | Reserved | |
| 0x0262024C | 0x0262024F | 4B | Reserved | |
| 0x02620250 | 0x02620253 | 4B | Reserved | |
| 0x02620254 | 0x02620257 | 4B | Reserved | |
| 0x02620258 | 0x0262025B | 4B | Reserved | |
| 0x0262025C | 0x0262025F | 4B | Reserved | |
| 0x02620260 | 0x0262027B | 28B | Reserved | |
| 0x0262027C | 0x0262027F | 4B | IPCGRH | See Section 4.3.14 |
| 0x02620280 | 0x02620283 | 4B | IPCAR0 | See Section 4.3.13 |
| 0x02620284 | 0x02620287 | 4B | IPCAR1 | See Section 4.3.13 (C6657) or Reserved (C6655) |
| 0x02620288 | 0x0262028B | 4B | Reserved | |

Table 4-2. Device State Control Registers (continued)

| ADDRESS START | ADDRESS END | SIZE | FIELD | DESCRIPTION |
|---------------|-------------|------|---------------------|--|
| 0x0262028C | 0x0262028F | 4B | Reserved | |
| 0x02620290 | 0x02620293 | 4B | Reserved | |
| 0x02620294 | 0x02620297 | 4B | Reserved | |
| 0x02620298 | 0x0262029B | 4B | Reserved | |
| 0x0262029C | 0x0262029F | 4B | Reserved | |
| 0x026202A0 | 0x026202BB | 28B | Reserved | |
| 0x026202BC | 0x026202BF | 4B | IPCARH | See Section 4.3.15 |
| 0x026202C0 | 0x026202FF | 64B | Reserved | |
| 0x02620300 | 0x02620303 | 4B | TINPSEL | See Section 4.3.16 |
| 0x02620304 | 0x02620307 | 4B | TOUTPSEL | See Section 4.3.17 |
| 0x02620308 | 0x0262030B | 4B | RSTMUX0 | See Section 4.3.18 |
| 0x0262030C | 0x0262030F | 4B | RSTMUX1 | See Section 4.3.18 (C6657) or Reserved (C6655) |
| 0x02620310 | 0x02620313 | 4B | Reserved | |
| 0x02620314 | 0x02620317 | 4B | Reserved | |
| 0x02620318 | 0x0262031B | 4B | Reserved | |
| 0x0262031C | 0x0262031F | 4B | Reserved | |
| 0x02620320 | 0x02620323 | 4B | Reserved | |
| 0x02620324 | 0x02620327 | 4B | Reserved | |
| 0x02620328 | 0x0262032B | 4B | MAINPLLCTL0 | See Section 8.5 |
| 0x0262032C | 0x0262032F | 4B | MAINPLLCTL1 | |
| 0x02620330 | 0x02620333 | 4B | DDR3PLLCTL0 | See Section 8.6 |
| 0x02620334 | 0x02620337 | 4B | DDR3PLLCTL1 | |
| 0x02620338 | 0x0262033B | 4B | Reserved | |
| 0x0262033C | 0x0262033F | 4B | Reserved | |
| 0x02620340 | 0x02620343 | 4B | SGMII_SERDES_CFGPLL | See Section 3.11 |
| 0x02620344 | 0x02620347 | 4B | SGMII_SERDES_CFGRX0 | |
| 0x02620348 | 0x0262034B | 4B | SGMII_SERDES_CFGTX0 | |
| 0x0262034C | 0x0262034F | 4B | Reserved | |
| 0x02620350 | 0x02620353 | 4B | Reserved | |
| 0x02620354 | 0x02620357 | 4B | Reserved | |
| 0x02620358 | 0x0262035B | 4B | PCIE_SERDES_CFGPLL | |
| 0x0262035C | 0x0262035F | 4B | Reserved | |
| 0x02620360 | 0x02620363 | 4B | SRIO_SERDES_CFGPLL | See Section 3.11 |
| 0x02620364 | 0x02620367 | 4B | SRIO_SERDES_CFGRX0 | |
| 0x02620368 | 0x0262036B | 4B | SRIO_SERDES_CFGTX0 | |
| 0x0262036C | 0x0262036F | 4B | SRIO_SERDES_CFGRX1 | |
| 0x02620370 | 0x02620373 | 4B | SRIO_SERDES_CFGTX1 | |
| 0x02620374 | 0x02620377 | 4B | SRIO_SERDES_CFGRX2 | |
| 0x02620378 | 0x0262037B | 4B | SRIO_SERDES_CFGTX2 | |
| 0x0262037C | 0x0262037F | 4B | SRIO_SERDES_CFGRX3 | |
| 0x02620380 | 0x02620383 | 4B | SRIO_SERDES_CFGTX3 | |
| 0x02620384 | 0x02620387 | 8B | Reserved | |
| 0x02620388 | 0x026203AF | 28B | Reserved | |
| 0x026203B0 | 0x026203B3 | 4B | Reserved | |

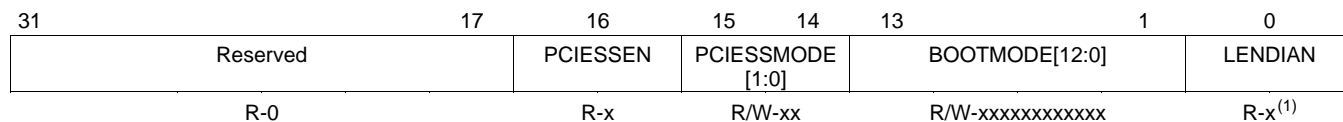
Table 4-2. Device State Control Registers (continued)

| ADDRESS START | ADDRESS END | SIZE | FIELD | DESCRIPTION |
|---------------|-------------|------|--------------------------|------------------------------------|
| 0x026203B4 | 0x026203B7 | 4B | HYPERLINK_SERDES_CFG PLL | See Section 3.11 |
| 0x026203B8 | 0x026203BB | 4B | HYPERLINK_SERDES_CFG RX0 | |
| 0x026203BC | 0x026203BF | 4B | HYPERLINK_SERDES_CFG TX0 | |
| 0x026203C0 | 0x026203C3 | 4B | HYPERLINK_SERDES_CFG RX1 | |
| 0x026203C4 | 0x026203C7 | 4B | HYPERLINK_SERDES_CFG TX1 | |
| 0x026203C8 | 0x026203CB | 4B | HYPERLINK_SERDES_CFG RX2 | |
| 0x026203CC | 0x026203CF | 4B | HYPERLINK_SERDES_CFG TX2 | |
| 0x026203D0 | 0x026203D3 | 4B | HYPERLINK_SERDES_CFG RX3 | |
| 0x026203D4 | 0x026203D7 | 4B | HYPERLINK_SERDES_CFG TX3 | |
| 0x026203D8 | 0x026203DB | 4B | Reserved | |
| 0x026203DC | 0x026203F7 | 28B | Reserved | |
| 0x026203F8 | 0x026203FB | 4B | DEVSPEED | See Section 4.3.19 |
| 0x026203FC | 0x026203FF | 4B | Reserved | |
| 0x02620400 | 0x02620403 | 4B | CHIP_MISC_CTL | See Section 5.4 |
| 0x02620404 | 0x02620467 | 100B | Reserved | |
| 0x02620468 | 0x0262057f | 280B | Reserved | |
| 0x02620580 | 0x02620583 | 4B | PIN_CONTROL_0 | See Section 4.3.20 |
| 0x02620584 | 0x02620587 | 4B | PIN_CONTROL_1 | See Section 4.3.21 |
| 0x02620588 | 0x0262058B | 4B | EMAC_UPP_PRI_ALLOC | See Section 5.4 |

4.3.1 Device Status Register

The Device Status Register depicts the device configuration selected upon a power-on reset by either the **POR** or **RESETFULL** pin. Once set, these bits will remain set until the next power-on reset. The Device Status Register is shown in [Figure 4-1](#) and described in [Table 4-3](#).

Figure 4-1. Device Status Register



Legend: R = Read only; RW = Read/Write; -n = value after reset

(1) x indicates the bootstrap value latched via the external pin

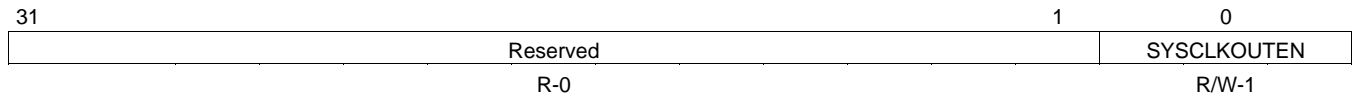
Table 4-3. Device Status Register Field Descriptions

| Bit | Field | Description |
|-------|-----------------|--|
| 31-17 | Reserved | Reserved. Read only, writes have no effect. |
| 16 | PCIESSEN | PCIe module enable <ul style="list-style-type: none"> • 0 = PCIe module disabled • 1 = PCIe module enabled |
| 15-14 | PCIESSMODE[1:0] | PCIe Mode selection pins <ul style="list-style-type: none"> • 00b = PCIe in End-point mode • 01b = PCIe in Legacy End-point mode (support for legacy INTx) • 10b = PCIe in Root complex mode • 11b = Reserved |
| 13-1 | BOOTMODE[12:0] | Determines the bootmode configured for the device. For more information on bootmode, refer to Section 3.5 and see the <i>Bootloader for the C66x DSP User's Guide</i> (SPRUGY5) |
| 0 | LENDIAN | Device Endian mode (LENDIAN) — Shows the status of whether the system is operating in Big Endian mode or Little Endian mode. <ul style="list-style-type: none"> • 0 = System is operating in Big Endian mode • 1 = System is operating in Little Endian mode |

4.3.2 Device Configuration Register

The Device Configuration Register is one-time writeable through software. The register is reset on all hard resets and is locked after the first write. The Device Configuration Register is shown in [Figure 4-2](#) and described in [Table 4-4](#).

Figure 4-2. Device Configuration Register (DEVCFG)



Legend: R = Read only; RW = Read/Write; -n = value after reset

Table 4-4. Device Configuration Register Field Descriptions

| Bit | Field | Description |
|------|-------------|--|
| 31-1 | Reserved | Reserved. Read only, writes have no effect. |
| 0 | SYSCLKOUTEN | SYSCLKOUT Enable <ul style="list-style-type: none"> • 0 = No clock output • 1 = Clock output enabled (default) |

4.3.3 JTAG ID (JTAGID) Register Description

The JTAG ID register is a read-only register that identifies to the customer the JTAG/Device ID. For the device, the JTAG ID register resides at address location 0x0262 0018. The JTAG ID Register is shown in [Figure 4-3](#) and described in [Table 4-5](#).

Figure 4-3. JTAG ID (JTAGID) Register

| | | | | | | | |
|---------|----|------------------------|----|----|----------------|---|-----|
| 31 | 28 | 27 | 12 | 11 | 1 | 0 | |
| VARIANT | | PART NUMBER | | | MANUFACTURER | | LSB |
| R-xxxxb | | R-1011 1001 0111 1010b | | | 0000 0010 111b | | R-1 |

Legend: RW = Read/Write; R = Read only; -n = value after reset

Table 4-5. JTAG ID Register Field Descriptions

| Bit | Field | Value | Description |
|-------|--------------|----------------------|-----------------------------------|
| 31-28 | VARIANT | xxxxb | Variant (4-Bit) value. |
| 27-12 | PART NUMBER | 1011 1001 0111 1010b | Part Number for boundary scan |
| 11-1 | MANUFACTURER | 0000 0010 111b | Manufacturer |
| 0 | LSB | 1b | This bit is read as a 1 for C665x |

NOTE

The value of the VARIANT and PART NUMBER fields depend on the silicon revision. See the Silicon Errata for details.

4.3.4 Kicker Mechanism (KICK0 and KICK1) Register

The Bootcfg module contains a kicker mechanism to prevent any spurious writes from changing any of the Bootcfg MMR values. When the kicker is locked (which it is initially after power on reset), none of the Bootcfg MMRs are writable (they are only readable). On the C665x, the exceptions to this are the IPC registers such as IPCGRx and IPCARx. These registers are not protected by the kicker mechanism. This mechanism requires two MMR writes to the KICK0 and KICK1 registers with exact data values before the kicker lock mechanism is un-locked. See [Table 4-2](#) for the address location. Once released, then all the Bootcfg MMRs having write permissions are writable (the read only MMRs are still read only). The first KICK0 data is 0x83e70b13. The second KICK1 data is 0x95a4f1e0. Writing any other data value to either of these kick MMRs will lock the kicker mechanism and block any writes to Bootcfg MMRs. To ensure protection of all Bootcfg MMRs, software must always re-lock the kicker mechanism after completing the MMR writes.

4.3.5 LRESETNMI PIN Status (LRSTNMIPINSTAT) Register

The LRSTNMIPINSTAT Register is created in Boot Configuration to latch the status of $\overline{\text{LRESET}}$ and $\overline{\text{NMI}}$ based on CORESEL. The LRESETNMI PIN Status Register is shown and described in the following tables.

Figure 4-4. LRESETNMI PIN Status Register (LRSTNMIPINSTAT)

| | | | | | | | |
|---------------|----|---------------|------|---------------|---|------|------|
| 31 | 18 | 17 | 16 | 15 | 2 | 1 | 0 |
| Reserved | | NMI1/Reserved | NMI0 | Reserved | | LR1 | LR0 |
| R, +0000 0000 | | R,+0 | R,+0 | R, +0000 0000 | | R,+0 | R,+0 |

Legend: R = Read only; -n = value after reset;

Table 4-6. LRESETNMI PIN Status Register (LRSTNMIPINSTAT) Field Descriptions

| Bit | Field | Description |
|-------|---------------|---|
| 31-18 | Reserved | Reserved |
| 17 | NMI1/Reserved | CorePac1 in NMI (C6657) or Reserved (C6655) |
| 16 | NMI0 | CorePac0 in NMI |
| 15-2 | Reserved | Reserved |
| 1 | LR1/Reserved | CorePac1 in Local Reset (C6657) or Reserved (C6655) |
| 0 | LR0 | CorePac0 in Local Reset |

4.3.6 LRESETNMI PIN Status Clear (LRSTNMIPINSTAT_CLR) Register

The LRSTNMIPINSTAT_CLR Register is used to clear the status of $\overline{\text{LRESET}}$ and $\overline{\text{NMI}}$ based on CORESEL. The LRESETNMI PIN Status Clear Register is shown and described in the following tables.

Figure 4-5. LRESETNMI PIN Status Clear Register (LRSTNMIPINSTAT_CLR)

| | | | | | | | |
|---------------|----|---------------|-------|---------------|---|--------------|-------|
| 31 | 18 | 17 | 16 | 15 | 2 | 1 | 0 |
| Reserved | | NMI1/Reserved | NMI0 | Reserved | | LR1/Reserved | LR0 |
| R, +0000 0000 | | WC,+0 | WC,+0 | R, +0000 0000 | | WC,+0 | WC,+0 |

Legend: R = Read only; -n = value after reset; WC = Write 1 to Clear

Table 4-7. LRESETNMI PIN Status Clear Register (LRSTNMIPINSTAT_CLR) Field Descriptions

| Bit | Field | Description |
|-------|---------------|---|
| 31-18 | Reserved | Reserved |
| 17 | NMI1/Reserved | CorePac1 in NMI Clear (C6657) or Reserved (C6655) |
| 16 | NMI0 | CorePac0 in NMI Clear |
| 15-2 | Reserved | Reserved |
| 1 | LR1/Reserved | CorePac1 in Local Reset Clear (C6657) or Reserved (C6655) |
| 0 | LR0 | CorePac0 in Local Reset Clear |

4.3.7 Reset Status (RESET_STAT) Register

The reset status register (RESET_STAT) captures the status of Local reset (LRx) for each of the cores and also the global device reset (GR). Software can use this information to take different device initialization steps, if desired.

- In case of Local reset: The LRx bits are written as 1 and GR bit is written as 0 only when the CorePac receives a local reset without receiving a global reset.
- In case of Global reset: The LRx bits are written as 0 and GR bit is written as 1 only when a global reset is asserted.

The Reset Status Register is shown and described in the following tables.

Figure 4-6. Reset Status Register (RESET_STAT)

| | | | | |
|-------|-----------------------------------|---|--------------|------|
| 31 | 30 | 2 | 1 | 0 |
| GR | Reserved | | LR1/Reserved | LR0 |
| R, +1 | R, + 000 0000 0000 0000 0000 0000 | | R,+0 | R,+0 |

Legend: R = Read only; -n = value after reset

Table 4-8. Reset Status Register (RESET_STAT) Field Descriptions

| Bit | Field | Description |
|------|--------------|--|
| 31 | GR | Global reset status <ul style="list-style-type: none"> • 0 = Device has not received a global reset. • 1 = Device received a global reset. |
| 30-2 | Reserved | Reserved. |
| 1 | LR1/Reserved | CorePac1 reset status (C6657) or Reserved (C6655) <ul style="list-style-type: none"> • 0 = CorePac1 has not received a local reset. • 1 = CorePac1 received a local reset. |
| 0 | LR0 | CorePac0 reset status <ul style="list-style-type: none"> • 0 = CorePac0 has not received a local reset. • 1 = CorePac0 received a local reset. |

4.3.8 Reset Status Clear (RESET_STAT_CLR) Register

The RESET_STAT bits can be cleared by writing 1 to the corresponding bit in the RESET_STAT_CLR register. The Reset Status Clear Register is shown and described in the following tables.

Figure 4-7. Reset Status Clear Register (RESET_STAT_CLR)

| | | | | |
|--------|------------------------------|---|--------------|-------|
| 31 | 30 | 2 | 1 | 0 |
| GR | Reserved | | LR1/Reserved | LR0 |
| RW, +0 | R, + 000 0000 0000 0000 0000 | | RW,+0 | RW,+0 |

Legend: R = Read only; RW = Read/Write; -n = value after reset

Table 4-9. Reset Status Clear Register (RESET_STAT_CLR) Field Descriptions

| Bit | Field | Description |
|------|--------------|--|
| 31 | GR | Global reset clear bit <ul style="list-style-type: none"> 0 = Writing a 0 has no effect. 1 = Writing a 1 to the GR bit clears the corresponding bit in the RESET_STAT register. |
| 30-2 | Reserved | Reserved. |
| 1 | LR1/Reserved | CorePac1 reset clear bit (C6657) or Reserved (C6655) <ul style="list-style-type: none"> 0 = Writing a 0 has no effect. 1 = Writing a 1 to the LR1 bit clears the corresponding bit in the RESET_STAT register. |
| 0 | LR0 | CorePac0 reset clear bit <ul style="list-style-type: none"> 0 = Writing a 0 has no effect. 1 = Writing a 1 to the LR0 bit clears the corresponding bit in the RESET_STAT register. |

4.3.9 Boot Complete (BOOTCOMPLETE) Register

The BOOTCOMPLETE register controls the BOOTCOMPLETE pin status. The purpose is to indicate the completion of the ROM booting process. The Boot Complete Register is shown and described in the following tables.

Figure 4-8. Boot Complete Register (BOOTCOMPLETE)

| | | | | | | |
|------------------------------------|----------|---|--------------|-------|-----|-------|
| 31 | Reserved | 2 | BC1/Reserved | 1 | BC0 | 0 |
| R, + 0000 0000 0000 0000 0000 0000 | | | | RW,+0 | | RW,+0 |

Legend: R = Read only; RW = Read/Write; -n = value after reset

Table 4-10. Boot Complete Register (BOOTCOMPLETE) Field Descriptions

| Bit | Field | Description |
|------|----------|---|
| 31-2 | Reserved | Reserved. |
| 1 | BC1 | CorePac1 boot status (C6657) or Reserved (C6655) <ul style="list-style-type: none"> 0 = CorePac1 boot NOT complete 1 = CorePac1 boot complete |
| 0 | BC0 | CorePac0 boot status <ul style="list-style-type: none"> 0 = CorePac0 boot NOT complete 1 = CorePac0 boot complete |

The BCx bit indicates the boot complete status of the corresponding core. All BCx bits will be sticky bits — that is they can be set only once by the software after device reset and they will be cleared to 0 on all device resets.

Boot ROM code will be implemented such that each core will set its corresponding BCx bit immediately before branching to the predefined location in memory.

4.3.10 Power State Control (PWRSTATECTL) Register

The PWRSTATECTL register is controlled by the software to indicate the power-saving mode. ROM code reads this register to differentiate between the various power saving modes. This register is cleared only by POR and will survive all other device resets. See the *Hardware Design Guide for KeyStone Devices* ([SPRABI2](#)) for more information. The Power State Control Register is shown in [Figure 4-9](#) and described in [Table 4-11](#).

Figure 4-9. Power State Control Register (PWRSTATECTL)

| | | | | |
|--------------------------------------|---|----------------------|-------------|---------|
| 31 | 3 | 2 | 1 | 0 |
| GENERAL_PURPOSE | | HIBERNATION _MODE | HIBERNATION | STANDBY |
| RW, +0000 0000 0000 0000 0000 0000 0 | | RW,+0 | RW,+0 | RW,+0 |

Legend: RW = Read/Write; -n = value after reset

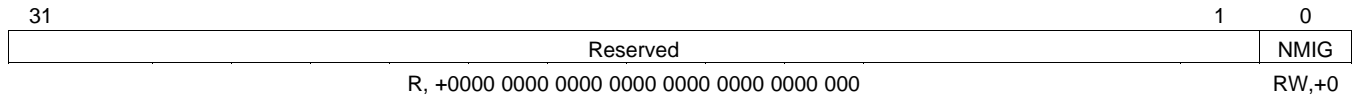
Table 4-11. Power State Control Register (PWRSTATECTL) Field Descriptions

| Bit | Field | Description |
|------|------------------|---|
| 31-3 | GENERAL_PURPOSE | Used to provide a start address for execution out of the hibernation modes. See the Bootloader for the <i>C66x DSP User's Guide</i> (SPRUGY8). |
| 2 | HIBERNATION_MODE | Indicates whether the device is in hibernation mode 1 or mode 2. <ul style="list-style-type: none"> • 0 = Hibernation mode 1 • 1 = Hibernation mode 2 |
| 1 | HIBERNATION | Indicates whether the device is in hibernation mode or not. <ul style="list-style-type: none"> • 0 = Not in hibernation mode • 1 = Hibernation mode |
| 0 | STANDBY | Indicates whether the device is in standby mode or not. <ul style="list-style-type: none"> • 0 = Not in standby mode • 1 = Standby mode |

4.3.11 NMI Event Generation to CorePac (NMIGRx) Register

NMIGRx registers are used for generating NMI events to the corresponding CorePac. The C6657 has two NMIGRx registers (NMIGR0 and NMIGR1) while the C6655 has only NMIGR0. The NMIGR0 register generates an NMI event to CorePac0, and the NMIGR1 register generates an NMI event to CorePac1. Writing a 1 to the NMIG field generates an NMI pulse. Writing a 0 has no effect and reads return 0 and have no other effect. The NMI Event Generation to CorePac Register is shown in [Figure 4-10](#) and described in [Table 4-12](#).

Figure 4-10. NMI Generation Register (NMIGRx)



Legend: RW = Read/Write; -n = value after reset

Table 4-12. NMI Generation Register (NMIGRx) Field Descriptions

| Bit | Field | Description |
|------|----------|---|
| 31-1 | Reserved | Reserved |
| 0 | NMIG | NMI pulse generation. Reads return 0 Writes: <ul style="list-style-type: none"> • 0 = No effect • 1 = Sends an NMI pulse to the corresponding CorePac — CorePac0 for NMIGR0, etc. |

4.3.12 IPC Generation (IPCGRx) Registers

IPCGRx are the IPC interrupt generation registers to facilitate inter CorePac interrupts.

The C6657 has two IPCGRx registers (IPCGR0 and IPCGR1) while the C6655 has only IPCGR0. These registers can be used by external hosts or CorePacs to generate interrupts to other CorePacs. A write of 1 to the IPCG field of the IPCGRx register will generate an interrupt pulse to CorePacx (0 ≤ x ≤ 1).

These registers also provide a *Source ID* facility by which up to 28 different sources of interrupts can be identified. Allocation of source bits to source processor and meaning is entirely based on software convention. The register field descriptions are given in the following tables. Virtually anything can be a source for these registers as this is completely controlled by software. Any master that has access to BOOTCFG module space can write to these registers. The IPC Generation Register is shown in [Figure 4-11](#) and described in [Table 4-13](#).

Figure 4-11. IPC Generation Registers (IPCGRx)

| | | | | | | | | | | | | |
|------------|------------|------------|------------|-----------------------|---|-------|-------|-------|-------|----------|---|-------|
| 31 | 30 | 29 | 28 | 27 | 8 | 7 | 6 | 5 | 4 | 3 | 1 | 0 |
| SRCS 27 | SRCS 26 | SRCS 25 | SRCS 24 | SRCS23 – SRCS4 | | SRCS3 | SRCS2 | SRCS1 | SRCS0 | Reserved | | IPCG |
| RW +0 | RW +0 | RW +0 | RW +0 | RW +0 (per bit field) | | RW +0 | RW +0 | RW +0 | RW +0 | R, +000 | | RW +0 |

Legend: R = Read only; RW = Read/Write; -n = value after reset

Table 4-13. IPC Generation Registers (IPCGRx) Field Descriptions

| Bit | Field | Description |
|------|----------|--|
| 31-4 | SRCSx | Interrupt source indication. Reads return current value of internal register bit. Writes: <ul style="list-style-type: none"> 0 = No effect 1 = Sets both SRCSx and the corresponding SRCCx. |
| 3-1 | Reserved | Reserved |
| 0 | IPCG | Inter-DSP interrupt generation. Reads return 0. Writes: <ul style="list-style-type: none"> 0 = No effect 1 = Creates an Inter-DSP interrupt. |

4.3.13 IPC Acknowledgement (IPCARx) Registers

IPCARx are the IPC interrupt-acknowledgement registers to facilitate inter-CorePac core interrupts.

The C6657 has two IPCARx registers (IPCAR0 and IPCAR1) while the C6655 has only IPCAR0. These registers also provide a *Source ID* facility by which up to 28 different sources of interrupts can be identified. Allocation of source bits to source processor and meaning is entirely based on software convention. The register field descriptions are shown in the following tables. Virtually anything can be a source for these registers as this is completely controlled by software. Any master that has access to BOOTCFG module space can write to these registers. The IPC Acknowledgement Register is shown in [Figure 4-12](#) and described in [Table 4-14](#).

Figure 4-12. IPC Acknowledgement Registers (IPCARx)

| | | | | | | | | | | | |
|------------|------------|------------|------------|-----------------------|-------|-------|-------|-------|----------|----------|---|
| 31 | 30 | 29 | 28 | 27 | 8 | 7 | 6 | 5 | 4 | 3 | 0 |
| SRCC 27 | SRCC 26 | SRCC 25 | SRCC 24 | SRCC23 – SRCC4 | SRCC3 | SRCC2 | SRCC1 | SRCC0 | Reserved | | |
| RW +0 | RW +0 | RW +0 | RW +0 | RW +0 (per bit field) | RW +0 | RW +0 | RW +0 | RW +0 | RW +0 | R, +0000 | |

Legend: R = Read only; RW = Read/Write; -n = value after reset

Table 4-14. IPC Acknowledgement Registers (IPCARx) Field Descriptions

| Bit | Field | Description |
|------|----------|--|
| 31-4 | SRCCx | Interrupt source acknowledgement. Reads return current value of internal register bit. Writes: <ul style="list-style-type: none"> • 0 = No effect • 1 = Clears both SRCCx and the corresponding SRCs |
| 3-0 | Reserved | Reserved |

4.3.14 IPC Generation Host (IPCGRH) Register

The IPCGRH register facilitates interrupts to external hosts. Operation and use of the IPCGRH register is the same as for other IPCGR registers. The interrupt output pulse created by the IPCGRH register appears on device pin HOUT.

The host interrupt output pulse should be stretched. It should be asserted for 4 bootcfg clock cycles (CPU/6) followed by a deassertion of 4 bootcfg clock cycles. Generating the pulse will result in 8 CPU/6 cycle pulse blocking window. Write to IPCGRH with IPCG bit (bit 0) set will only generate a pulse if they are beyond 8 CPU/6 cycle period. The IPC Generation Host Register is shown in [Figure 4-13](#) and described in [Table 4-15](#).

Figure 4-13. IPC Generation Registers (IPCGRH)

| | | | | | | | | | | | | |
|------------|------------|------------|------------|-----------------------|---|-------|-------|-------|-------|----------|---|-------|
| 31 | 30 | 29 | 28 | 27 | 8 | 7 | 6 | 5 | 4 | 3 | 1 | 0 |
| SRCS 27 | SRCS 26 | SRCS 25 | SRCS 24 | SRCS23 – SRCS4 | | SRCS3 | SRCS2 | SRCS1 | SRCS0 | Reserved | | IPCG |
| RW +0 | RW +0 | RW +0 | RW +0 | RW +0 (per bit field) | | RW +0 | RW +0 | RW +0 | RW +0 | R, +000 | | RW +0 |

Legend: R = Read only; RW = Read/Write; -n = value after reset

Table 4-15. IPC Generation Registers (IPCGRH) Field Descriptions

| Bit | Field | Description |
|------|----------|--|
| 31-4 | SRCSx | Interrupt source indication. Reads return current value of internal register bit. Writes: <ul style="list-style-type: none"> 0 = No effect 1 = Sets both SRCSx and the corresponding SRCCx. |
| 3-1 | Reserved | Reserved |
| 0 | IPCG | Host interrupt generation. Reads return 0. Writes: <ul style="list-style-type: none"> 0 = No effect 1 = Creates an interrupt pulse on device pin (host interrupt/event output in HOUT pin) |

4.3.15 IPC Acknowledgement Host (IPCARH) Register

IPCARH registers are provided to facilitate host DSP interrupt. Operation and use of IPCARH is the same as other IPCAR registers. The IPC Acknowledgement Host Register is shown in [Figure 4-14](#) and described in [Table 4-16](#).

Figure 4-14. IPC Acknowledgement Register (IPCARH)

| | | | | | | | | | | | |
|------------|------------|------------|------------|-----------------------|---|-------|-------|-------|-------|----------|---|
| 31 | 30 | 29 | 28 | 27 | 8 | 7 | 6 | 5 | 4 | 3 | 0 |
| SRCC 27 | SRCC 26 | SRCC 25 | SRCC 24 | SRCC23 – SRCC4 | | SRCC3 | SRCC2 | SRCC1 | SRCC0 | Reserved | |
| RW +0 | RW +0 | RW +0 | RW +0 | RW +0 (per bit field) | | RW +0 | RW +0 | RW +0 | RW +0 | R, +0000 | |

Legend: R = Read only; RW = Read/Write; -n = value after reset

Table 4-16. IPC Acknowledgement Register (IPCARH) Field Descriptions

| Bit | Field | Description |
|------|----------|---|
| 31-4 | SRCCx | Interrupt source acknowledgement. Reads return current value of internal register bit. Writes: <ul style="list-style-type: none"> • 0 = No effect • 1 = Clears both SRCCx and the corresponding SRCSx |
| 3-0 | Reserved | Reserved |

4.3.16 Timer Input Selection Register (TINPSEL)

Timer input selection is handled within the control register TINPSEL. The Timer Input Selection Register is shown in [Figure 4-15](#) and described in [Table 4-17](#).

Figure 4-15. Timer Input Selection Register (TINPSEL)

| | | | | | | | | | | | | | | | |
|-------------------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|
| Reserved | | | | | | | | | | | | | | | |
| R, +1010 1010 1010 1010 | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TINPH SEL7 | TINPL SEL7 | TINPH SEL6 | TINPL SEL6 | TINPH SEL5 | TINPL SEL5 | TINPH SEL4 | TINPL SEL4 | TINPH SEL3 | TINPL SEL3 | TINPH SEL2 | TINPL SEL2 | TINPH SEL1 | TINPL SEL1 | TINPH SEL0 | TINPL SEL0 |
| RW, +1 | RW, +0 | RW, +1 | RW, +0 | RW, +1 | RW, +0 | RW, +1 | RW, +0 | RW, +1 | RW, +0 | RW, +1 | RW, +0 | RW, +1 | RW, +0 | RW, +1 | RW, +0 |

Legend: R = Read only; RW = Read/Write; -n = value after reset

Table 4-17. Timer Input Selection Field Description (TINPSEL)

| Bit | Field | Description |
|-------|-----------|--|
| 31-16 | Reserved | <ul style="list-style-type: none"> Reserved |
| 15 | TINPHSEL7 | Input select for TIMER7 high. <ul style="list-style-type: none"> 0 = TIMI0 1 = TIMI1 |
| 14 | TINPLSEL7 | Input select for TIMER7 low. <ul style="list-style-type: none"> 0 = TIMI0 1 = TIMI1 |
| 13 | TINPHSEL6 | Input select for TIMER6 high. <ul style="list-style-type: none"> 0 = TIMI0 1 = TIMI1 |
| 12 | TINPLSEL6 | Input select for TIMER6 low. <ul style="list-style-type: none"> 0 = TIMI0 1 = TIMI1 |
| 11 | TINPHSEL5 | Input select for TIMER5 high. <ul style="list-style-type: none"> 0 = TIMI0 1 = TIMI1 |
| 10 | TINPLSEL5 | Input select for TIMER5 low. <ul style="list-style-type: none"> 0 = TIMI0 1 = TIMI1 |
| 9 | TINPHSEL4 | Input select for TIMER4 high. <ul style="list-style-type: none"> 0 = TIMI0 1 = TIMI1 |
| 8 | TINPLSEL4 | Input select for TIMER4 low. <ul style="list-style-type: none"> 0 = TIMI0 1 = TIMI1 |
| 7 | TINPHSEL3 | Input select for TIMER3 high. <ul style="list-style-type: none"> 0 = TIMI0 1 = TIMI1 |
| 6 | TINPLSEL3 | Input select for TIMER3 low. <ul style="list-style-type: none"> 0 = TIMI0 1 = TIMI1 |
| 5 | TINPHSEL2 | Input select for TIMER2 high. <ul style="list-style-type: none"> 0 = TIMI0 1 = TIMI1 |

Table 4-17. Timer Input Selection Field Description (TINPSEL) (continued)

| Bit | Field | Description |
|-----|-----------|---|
| 4 | TINPLSEL2 | Input select for TIMER2 low. <ul style="list-style-type: none">• 0 = TIMI0• 1 = TIMI1 |
| 3 | TINPHSEL1 | Input select for TIMER1 high. <ul style="list-style-type: none">• 0 = TIMI0• 1 = TIMI1 |
| 2 | TINPLSEL1 | Input select for TIMER1 low. <ul style="list-style-type: none">• 0 = TIMI0• 1 = TIMI1 |
| 1 | TINPHSEL0 | Input select for TIMER0 high. <ul style="list-style-type: none">• 0 = TIMI0• 1 = TIMI1 |
| 0 | TINPLSEL0 | Input select for TIMER0 low. <ul style="list-style-type: none">• 0 = TIMI0• 1 = TIMI1 |

4.3.17 Timer Output Selection Register (TOUTPSEL)

The timer output selection is handled within the control register TOUTSEL. The Timer Output Selection Register is shown in [Figure 4-16](#) and described in [Table 4-18](#).

Figure 4-16. Timer Output Selection Register (TOUTPSEL)

| | | | | | |
|-------------------------------------|----|-----------|---|-----------|---|
| 31 | 10 | 9 | 5 | 4 | 0 |
| Reserved | | TOUTPSEL1 | | TOUTPSEL0 | |
| R,+00000000000000000000000000000000 | | RW,+00001 | | RW,+00000 | |

Legend: R = Read only; RW = Read/Write; -n = value after reset

Table 4-18. Timer Output Selection Field Description (TOUTPSEL)

| Bit | Field | Description |
|-------|-----------|--|
| 31-10 | Reserved | Reserved |
| 9-5 | TOUTPSEL1 | Output select for TIMO1 <ul style="list-style-type: none"> • 0x0: TOUTL0 • 0x1: TOUTH0 • 0x2: TOUTL1 • 0x3: TOUTH1 • 0x4: TOUTL2 • 0x5: TOUTH2 • 0x6: TOUTL3 • 0x7: TOUTH3 • 0x8: TOUTL4 • 0x9: TOUTH4 • 0xA: TOUTL5 • 0xB: TOUTH5 • 0xC: TOUTL6 • 0xD: TOUTH6 • 0xE: TOUTL7 • 0xF: TOUTH7 • 0x10 to 0x1F: Reserved |
| 4-0 | TOUTPSEL0 | Output select for TIMO0 <ul style="list-style-type: none"> • 0x0: TOUTL0 • 0x1: TOUTH0 • 0x2: TOUTL1 • 0x3: TOUTH1 • 0x4: TOUTL2 • 0x5: TOUTH2 • 0x6: TOUTL3 • 0x7: TOUTH3 • 0x8: TOUTL4 • 0x9: TOUTH4 • 0xA: TOUTL5 • 0xB: TOUTH5 • 0xC: TOUTL6 • 0xD: TOUTH6 • 0xE: TOUTL7 • 0xF: TOUTH7 • 0x10 to 0x1F: Reserved |

4.3.18 Reset Mux (RSTMUXx) Register

The software controls the Reset Mux block through the reset multiplex registers using RSTMUX0 through RSTMUX1 for each of the two CorePacs on the C6657. The C6655 has only RSTMUX0. These registers are located in Bootcfg memory space. The Reset Mux Register is shown in [Figure 4-17](#) and described in [Table 4-19](#).

Figure 4-17. Reset Mux Register RSTMUXx

| | | | | | | | | | |
|------------------------------------|----|------------|----------|----------|---------|----------|--------|---|---|
| 31 | 10 | 9 | 8 | 7 | 5 | 4 | 3 | 1 | 0 |
| Reserved | | EVTSTATCLR | Reserved | DELAY | EVTSTAT | OMODE | LOCK | | |
| R, +0000 0000 0000 0000 0000 00 | | RC, +0 | R, +0 | RW, +100 | R, +0 | RW, +000 | RW, +0 | | |

Legend: R = Read only; RW = Read/Write; -n = value after reset; RC = Read only and write 1 to clear

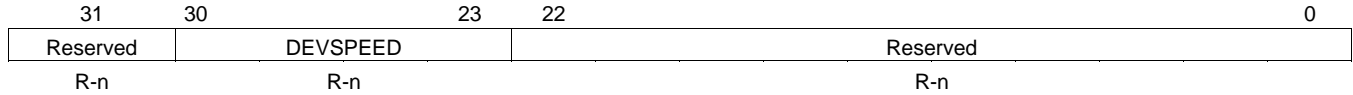
Table 4-19. Reset Mux Register Field Descriptions

| Bit | Field | Description |
|-------|------------|---|
| 31-10 | Reserved | Reserved |
| 9 | EVTSTATCLR | Clear event status <ul style="list-style-type: none"> 0 = Writing 0 has no effect 1 = Writing 1 clears the EVTSTAT bit |
| 8 | Reserved | Reserved |
| 7-5 | DELAY | Delay cycles between NMI & local reset <ul style="list-style-type: none"> 000b = 256 CPU/6 cycles delay between NMI & local reset, when OMODE = 100b 001b = 512 CPU/6 cycles delay between NMI & local reset, when OMODE=100b 010b = 1024 CPU/6 cycles delay between NMI & local reset, when OMODE=100b 011b = 2048 CPU/6 cycles delay between NMI & local reset, when OMODE=100b 100b = 4096 CPU/6 cycles delay between NMI & local reset, when OMODE=100b (Default) 101b = 8192 CPU/6 cycles delay between NMI & local reset, when OMODE=100b 110b = 16384 CPU/6 cycles delay between NMI & local reset, when OMODE=100b 111b = 32768 CPU/6 cycles delay between NMI & local reset, when OMODE=100b |
| 4 | EVTSTAT | Event status. <ul style="list-style-type: none"> 0 = No event received (Default) 1 = WD timer event received by Reset Mux block |
| 3-1 | OMODE | Timer event operation mode <ul style="list-style-type: none"> 000b = WD timer event input to the reset mux block does not cause any output event (default) 001b = Reserved 010b = WD timer event input to the reset mux block causes local reset input to CorePac 011b = WD timer event input to the reset mux block causes NMI input to CorePac 100b = WD timer event input to the reset mux block causes NMI input followed by local reset input to CorePac. Delay between NMI and local reset is set in DELAY bit field. 101b = WD timer event input to the reset mux block causes device reset to C665x 110b = Reserved 111b = Reserved |
| 0 | LOCK | Lock register fields <ul style="list-style-type: none"> 0 = Register fields are not locked (default) 1 = Register fields are locked until the next timer reset |

4.3.19 Device Speed (DEVSPEED) Register

The Device Speed Register indicates the device speed grade. The Device Speed Register is shown in [Figure 4-18](#) and described in [Table 4-20](#).

Figure 4-18. Device Speed Register (DEVSPEED)



Legend: R = Read only; RW = Read/Write; -n = value after reset

Table 4-20. Device Speed Register Field Descriptions

| Bit | Field | Description |
|-------|----------|--|
| 31 | Reserved | Reserved. Read only |
| 30-23 | DEVSPEED | Indicates the speed of the device (Read Only) <ul style="list-style-type: none"> • 1xxx xxxxb = 850 MHz • 01xx xxxxb = 1000 MHz • 001x xxxxb = 1250 MHz • 0001 xxxxb = Reserved • 0000 1xxxb = Reserved • 0000 01xxb = 1250 MHz • 0000 001xb = 1000 MHz • 0000 0001b = 850 MHz • 0000 0000b = 850 MHz |
| 22-0 | Reserved | Reserved. Read only |

4.3.20 Pin Control 0 (PIN_CONTROL_0) Register

The Pin Control 0 Register controls the pin muxing between GPIO[16:31] and TIMER / UART / SPI pins. The Pin Control 0 Register is shown in [Figure 4-19](#) and described in [Table 4-21](#).

Figure 4-19. Pin Control 0 Register (PIN_CONTROL_0)

| | | | | | | | |
|-------------------------|-------------------------|------------------------|------------------------|-------------------------|-------------------------|------------------------|------------------------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
| GPIO31_SPID OUT_MUX | GPIO30_SPIDI N_MUX | GPIO29_SPIC S1_MUX | GPIO28_SPIC S0_MUX | GPIO27_UART RTS1_MUX | GPIO26_UART CTS1_MUX | GPIO25_UART TX1_MUX | GPIO24_UART RX1_MUX |
| RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| GPIO23_UART RTS0_MUX | GPIO22_UART CTS0_MUX | GPIO21_UART TX0_MUX | GPIO20_UART RX0_MUX | GPIO19_TIMO 1_MUX | GPIO18_TIMO 0_MUX | GPIO17_TIMI1 _MUX | GPIO16_TIMIO _MUX |
| RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 |
| 15 | Reserved | | | | | | 0 |
| R-0 | | | | | | | |

Legend: R = Read only; RW = Read/Write; -n = value after reset

Table 4-21. Pin Control 0 Register Field Descriptions

| Bit | Field | Description |
|-----|---------------------|---|
| 31 | GPIO31_SPIDOUT_MUX | SPI or GPIO mux control <ul style="list-style-type: none"> 0 = SPIDOUT pin enabled 1 = GPIO31 pin enabled |
| 30 | GPIO30_SPIDIN_MUX | SPI or GPIO mux control <ul style="list-style-type: none"> 0 = SPIDIN pin enabled 1 = GPIO30 pin enabled |
| 29 | GPIO29_SPICS1_MUX | SPI or GPIO mux control <ul style="list-style-type: none"> 0 = SPICS1 pin enabled 1 = GPIO29 pin enabled |
| 28 | GPIO28_SPICS0_MUX | SPI or GPIO mux control <ul style="list-style-type: none"> 0 = SPICS0 pin enabled 1 = GPIO28 pin enabled |
| 27 | GPIO27_UARTRTS1_MUX | UART or GPIO mux control <ul style="list-style-type: none"> 0 = UARTRTS1 pin enabled 1 = GPIO27 pin enabled |
| 26 | GPIO26_UARTCTS1_MUX | UART or GPIO mux control <ul style="list-style-type: none"> 0 = UARTCTS1 pin enabled 1 = GPIO26 pin enabled |
| 25 | GPIO25_UARTTX1_MUX | UART or GPIO mux control <ul style="list-style-type: none"> 0 = UARTRTX1 pin enabled 1 = GPIO25 pin enabled |
| 24 | GPIO24_UARTRX1_MUX | UART or GPIO mux control <ul style="list-style-type: none"> 0 = UARTRX1 pin enabled 1 = GPIO24 pin enabled |
| 23 | GPIO23_UARTRTS0_MUX | UART or GPIO mux control <ul style="list-style-type: none"> 0 = UARTRTS0 pin enabled 1 = GPIO23 pin enabled |
| 22 | GPIO22_UARTCTS0_MUX | UART or GPIO mux control <ul style="list-style-type: none"> 0 = UARTCTS0 pin enabled 1 = GPIO22 pin enabled |
| 21 | GPIO21_UARTTX0_MUX | UART or GPIO mux control <ul style="list-style-type: none"> 0 = UARTRTX0 pin enabled 1 = GPIO21 pin enabled |

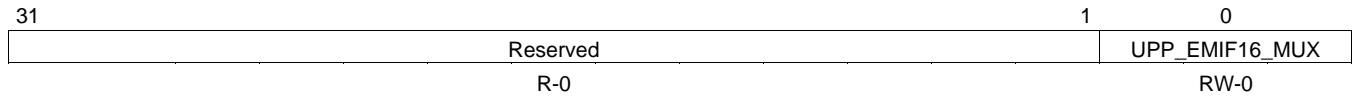
Table 4-21. Pin Control 0 Register Field Descriptions (continued)

| Bit | Field | Description |
|------|--------------------|--|
| 20 | GPIO20_UARTRX0_MUX | UART or GPIO mux control <ul style="list-style-type: none"> • 0 = UARTRX0 pin enabled • 1 = GPIO20 pin enabled |
| 19 | GPIO19_TIMO1_MUX | TIMER or GPIO mux control <ul style="list-style-type: none"> • 0 = TIMO1 pin enabled • 1 = GPIO19 pin enabled |
| 18 | GPIO18_TIMO0_MUX | TIMER or GPIO mux control <ul style="list-style-type: none"> • 0 = TIMO0 pin enabled • 1 = GPIO18 pin enabled |
| 17 | GPIO17_TIMI1_MUX | TIMER or GPIO mux control <ul style="list-style-type: none"> • 0 = TIMI1 pin enabled • 1 = GPIO17 pin enabled |
| 16 | GPIO16_TIMO0_MUX | TIMER or GPIO mux control <ul style="list-style-type: none"> • 0 = TIMO0 pin enabled • 1 = GPIO16 pin enabled |
| 15-0 | Reserved | Reserved |

4.3.21 Pin Control 1 (PIN_CONTROL_1) Register

The Pin Control 1 Register controls the pin muxing between uPP and EMIF16 pins. The Pin Control 1 Register is shown in [Figure 4-20](#) and described in [Table 4-22](#).

Figure 4-20. Pin Control 1 Register (PIN_CONTROL_1)



Legend: R = Read only; RW = Read/Write; -n = value after reset

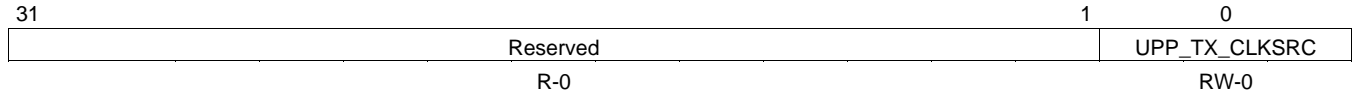
Table 4-22. Pin Control 1 Register Field Descriptions

| Bit | Field | Description |
|------|--------------|---|
| 31-1 | Reserved | Reserved |
| 0 | UPP_EMIF_MUX | uPP or EMIF16 mux control <ul style="list-style-type: none"> • 0 = EMIF16 pins enabled • 1 = uPP pins enabled |

4.3.22 uPP Clock Source (UPP_CLOCK) Register

The uPP Clock Source Register controls whether the uPP transmit clock is internally or externally sourced. The uPP Clock Source Register is shown in [Figure 4-21](#) and described in [Table 4-23](#).

Figure 4-21. uPP Clock Source Register (UPP_CLOCK)



Legend: R = Read only; RW = Read/Write; -n = value after reset

Table 4-23. uPP Clock Source Register Field Descriptions

| Bit | Field | Description |
|------|---------------|--|
| 31-1 | Reserved | Reserved |
| 0 | UPP_TX_CLKSRC | uPP clock source selection <ul style="list-style-type: none"> • 0 = from internal SYSCLK4 (CPU/3) • 1 = from external UPP_2TXCLK pin |

4.4 Pullup/Pulldown Resistors

Proper board design should ensure that input pins to the device always be at a valid logic level and not floating. This may be achieved via pullup/pulldown resistors. The device features internal pullup (IPU) and internal pulldown (IPD) resistors on most pins to eliminate the need, unless otherwise noted, for external pullup/pulldown resistors.

An external pullup/pulldown resistor needs to be used in the following situations:

- **Device Configuration Pins:** If the pin is both routed out and is not driven (in Hi-Z state), an external pullup/pulldown resistor must be used, even if the IPU/IPD matches the desired value/state.
- **Other Input Pins:** If the IPU/IPD does not match the desired value/state, use an external pullup/pulldown resistor to pull the signal to the opposite rail.

For the device configuration pins (listed in [Table 4-1](#)), if they are both routed out and are not driven (in Hi-Z state), it is strongly recommended that an external pullup/pulldown resistor be implemented. Although, internal pullup/pulldown resistors exist on these pins and they may match the desired configuration value, providing external connectivity can help ensure that valid logic levels are latched on these device configuration pins. In addition, applying external pullup/pulldown resistors on the device configuration pins adds convenience to the user in debugging and flexibility in switching operating modes.

Tips for choosing an external pullup/pulldown resistor:

- Consider the total amount of current that may pass through the pullup or pulldown resistor. Make sure to include the leakage currents of all the devices connected to the net, as well as any internal pullup or pulldown resistors.
- Decide a target value for the net. For a pulldown resistor, this should be below the lowest V_{IL} level of all inputs connected to the net. For a pullup resistor, this should be above the highest V_{IH} level of all inputs on the net. A reasonable choice would be to target the V_{OL} or V_{OH} levels for the logic family of the limiting device; which, by definition, have margin to the V_{IL} and V_{IH} levels.
- Select a pullup/pulldown resistor with the largest possible value that can still ensure that the net will reach the target pulled value when maximum current from all devices on the net is flowing through the resistor. The current to be considered includes leakage current plus, any other internal and external pullup/pulldown resistors on the net.
- For bidirectional nets, there is an additional consideration that sets a lower limit on the resistance value of the external resistor. Verify that the resistance is small enough that the weakest output buffer can drive the net to the opposite logic level (including margin).
- Remember to include tolerances when selecting the resistor value.
- For pullup resistors, also remember to include tolerances on the DV_{DD} rail.

For most systems:

- A 1-k Ω resistor can be used to oppose the IPU/IPD while meeting the above criteria. Users should confirm this resistor value is correct for their specific application.
- A 20-k Ω resistor can be used to compliment the IPU/IPD on the device configuration pins while meeting the above criteria. Users should confirm this resistor value is correct for their specific application.

For more detailed information on input current (I_i), and the low-level/high-level input voltages (V_{IL} and V_{IH}) for the C665x device, see [Section 7.3](#).

To determine which pins on the device include internal pullup/pulldown resistors, see [Table 3-41](#).

5 System Interconnect

On the C665x device, the C66x CorePacs, the EDMA3 transfer controller, and the system peripherals are interconnected through the TeraNet, which is a non-blocking switch fabric enabling fast and contention-free internal data movement. The TeraNet allows for low-latency, concurrent data transfers between master peripherals and slave peripherals. The TeraNet also allows for seamless arbitration between the system masters when accessing system slaves.

5.1 Internal Buses and Switch Fabrics

Two types of buses exist in the device: data buses and configuration buses. Some peripherals have both a data bus and a configuration bus interface, while others have only one type of interface. Further, the bus interface width and speed varies from peripheral to peripheral. Configuration buses are mainly used to access the register space of a peripheral and the data buses are used mainly for data transfers.

The C66x CorePacs, the EDMA3 traffic controller, and the various system peripherals can be classified into two categories: masters and slaves. Masters are capable of initiating read and write transfers in the system and do not rely on the EDMA3 for their data transfers. Slaves, on the other hand, rely on the masters to perform transfers to and from them. Examples of masters include the EDMA3 traffic controller, SRIO, and PCI Express. Examples of slaves include the SPI, UART, and I²C.

The masters and slaves in the device are communicating through the TeraNet (switch fabric). The device contains two switch fabrics. The data switch fabric (data TeraNet) and the configuration switch fabric (configuration TeraNet). The data TeraNet, is a high-throughput interconnect mainly used to move data across the system. The data TeraNet connects masters to slaves via data buses. The configuration TeraNet, is mainly used to access peripheral registers. The configuration TeraNet connects masters to slaves via configuration buses. Note that the data TeraNet also connects to the configuration TeraNet. For more details see [Section 5.2](#).

5.2 Switch Fabric Connections Matrix

The tables below list the master and slave end point connections.

Intersecting cells may contain one of the following:

- Y — There is a connection between this master and that slave.
- - — There is NO connection between this master and that slave.
- n — A numeric value indicates that the path between this master and that slave goes through bridge n.

Table 5-1. Switch Fabric Connection Matrix Section 1

| MASTERS | SLAVES | | | | | | | | | | | | | | | | | | | | | | | |
|---------------------------|---------------|----------------------------|-------------|----------|-----|--------|------------------|------------------|----------|-----------------|----------|----------|-----|-----------|-------|--------|-------|--------------------|----------|-------|---------|--------------|-----------|--------|
| | CorePac0_SDMA | CorePac1_SDMA (C6657 Only) | PCIe0_Slave | Boot_ROM | SPI | EMIF16 | Mcbsp0_FIFO_Data | Mcbsp1_FIFO_Data | QM_Slave | HyperLink_Slave | MSMC_SES | MSMC_SMS | STM | VCP2(0-1) | TCF3d | TETB_D | TETB0 | TETB1 (C6657 Only) | VCP2_Cfg | TCF3d | EDMA3CC | EDMA3TC(0-3) | Semaphore | QM_CFG |
| HyperLink_Master | Y | Y | Y | Y | Y | Y | 1, 4 | 1, 4 | 1 | - | Y | Y | - | Y | Y | - | - | - | 1 | 1 | 1 | 1 | 1 | 1 |
| EDMA3CC_TC0_RD | Y | Y | Y | Y | Y | Y | - | - | - | Y | Y | Y | - | Y | Y | 1 | - | - | 1 | 1 | 1 | 1 | 1 | 1 |
| EDMA3CC_TC0_WR | Y | Y | Y | - | Y | Y | - | - | - | Y | Y | Y | 1 | Y | Y | - | - | - | 1 | 1 | 1 | 1 | 1 | 1 |
| EDMA3CC_TC1_RD | Y | Y | Y | Y | Y | Y | 2, 4 | 2, 4 | - | Y | Y | Y | - | - | Y | - | 2 | 2 | - | - | 2 | 2 | - | - |
| EDMA3CC_TC1_WR | Y | Y | Y | - | Y | Y | 2, 4 | 2, 4 | - | Y | Y | Y | - | - | Y | - | - | - | - | 2 | 2 | - | - | |
| EDMA3CC_TC2_RD | Y | Y | Y | Y | Y | Y | 1, 4 | 1, 4 | - | Y | Y | Y | - | Y | Y | 1 | - | - | 1 | 1 | 1 | 1 | 1 | 1 |
| EDMA3CC_TC2_WR | Y | Y | Y | - | Y | Y | 1, 4 | 1, 4 | - | Y | Y | Y | - | Y | Y | - | - | - | 1 | 1 | 1 | 1 | 1 | 1 |
| EDMA3CC_TC3_RD | Y | Y | Y | Y | Y | Y | - | - | 2 | Y | Y | Y | - | - | - | - | - | - | - | - | 2 | 2 | - | - |
| EDMA3CC_TC3_WR | Y | Y | Y | - | Y | Y | - | - | 2 | Y | Y | Y | 2 | - | - | - | - | - | - | - | 2 | 2 | - | - |
| SRIO packet DMA | Y | Y | - | - | - | - | - | - | 1 | Y | Y | Y | - | - | - | - | - | - | - | - | - | - | - | - |
| SRIO_Master | Y | Y | - | - | Y | Y | 1, 4 | 1, 4 | 1 | Y | Y | Y | 1 | Y | Y | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| PCIe_Master | Y | Y | - | - | Y | Y | 1, 4 | 1, 4 | 1 | Y | Y | Y | 1 | Y | Y | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| EMAC | 3 | 3 | - | - | - | - | - | - | - | 3 | 3 | 3 | - | - | - | - | - | - | - | - | - | - | - | - |
| MSMC_Data_Master | Y | Y | Y | Y | Y | Y | 1, 4 | 1, 4 | 1 | Y | - | - | 1 | Y | Y | - | - | - | - | - | - | - | - | - |
| QM Packet DMA | Y | Y | - | - | - | - | - | - | 1 | Y | Y | Y | - | - | - | - | - | - | - | - | - | - | - | - |
| QM Second | Y | Y | - | Y | Y | Y | - | - | 1 | Y | Y | Y | - | - | - | - | - | - | - | - | - | - | - | - |
| DAP_Master | Y | Y | Y | Y | Y | Y | 1, 4 | 1, 4 | 1 | Y | Y | Y | 1 | Y | Y | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| CorePac0_CFG | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | Y | - | - |
| CorePac1_CFG (C6657 Only) | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | Y | Y | Y | Y | Y | Y | Y | Y | Y |
| Tracer_Master | - | - | - | - | - | - | - | - | - | - | - | - | 1 | - | - | Y | Y | Y | Y | Y | Y | Y | Y | Y |
| UPP | 3 | 3 | - | - | - | - | - | - | - | 3 | 3 | 3 | - | - | - | - | - | - | - | - | - | - | - | - |

Table 5-2. Switch Fabric Connection Matrix Section 2

| MASTERS | SLAVES | | | | | | | | | | | | | | | | | | | | | |
|---------------------------|--------|--------------------------|-------|------|------|---------|-------------|-------|----------|------|------|------|--------|------|--------------|-------------|----------------|----------------|---------------------|----------|---------|---|
| | Tracer | SRIO_CFG (C6655/57 Only) | Timer | GPIO | I2C | SEC_CTL | SEC_KEY_MGR | Efuse | Boot_CFG | PSC | PLL | CIC | MPU0-3 | MPU4 | Debug_SS_CFG | SmartReflex | UART_CFG (0-1) | McBSP_CFG(0-1) | McBSP_FIFO_CFG(0-1) | EMAC_CFG | UPP_CFG | |
| HyperLink_Master | 1 | 1 | 1, 4 | 1, 4 | 1, 4 | 1, 4 | 1, 4 | 1, 4 | 1, 4 | 1, 4 | 1, 4 | 1, 4 | 1 | 1, 4 | 1, 4 | 1, 4 | 1, 4 | 1, 4 | 1, 4 | 1, 4 | 1, 4 | 1 |
| EDMA3CC_TC0_RD | 1 | 1 | 1, 4 | 1, 4 | 1, 4 | 1, 4 | 1, 4 | - | 1, 4 | 1, 4 | 1, 4 | 1, 4 | 1 | 1, 4 | - | - | 1, 4 | 1, 4 | 1, 4 | 1, 4 | 1, 4 | 1 |
| EDMA3CC_TC0_WR | 1 | 1 | 1, 4 | 1, 4 | 1, 4 | 1, 4 | 1, 4 | - | 1, 4 | 1, 4 | 1, 4 | 1, 4 | 1 | 1, 4 | - | - | 1, 4 | 1, 4 | 1, 4 | 1, 4 | 1, 4 | 1 |
| EDMA3CC_TC1_RD | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| EDMA3CC_TC1_WR | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| EDMA3CC_TC2_RD | 1 | 1 | 1, 4 | 1, 4 | 1, 4 | 1, 4 | 1, 4 | - | 1, 4 | 1, 4 | 1, 4 | 1, 4 | 1 | 1, 4 | - | - | 1, 4 | 1, 4 | 1, 4 | 1, 4 | 1, 4 | 1 |
| EDMA3CC_TC2_WR | 1 | 1 | 1, 4 | 1, 4 | 1, 4 | 1, 4 | 1, 4 | - | 1, 4 | 1, 4 | 1, 4 | 1, 4 | 1 | 1, 4 | - | - | 1, 4 | 1, 4 | 1, 4 | 1, 4 | 1, 4 | 1 |
| EDMA3CC_TC3_RD | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| EDMA3CC_TC3_WR | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| SRIO packet DMA | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| SRIO_Master | 1 | 1 | 1, 4 | 1, 4 | 1, 4 | 1, 4 | 1, 4 | - | 1, 4 | 1, 4 | 1, 4 | 1, 4 | 1 | 1, 4 | 1, 4 | 1, 4 | 1, 4 | 1, 4 | 1, 4 | 1, 4 | 1, 4 | 1 |
| PCIe_Master | 1 | 1 | 1, 4 | 1, 4 | 1, 4 | 1, 4 | 1, 4 | - | 1, 4 | 1, 4 | 1, 4 | 1, 4 | 1 | 1, 4 | 1, 4 | 1, 4 | 1, 4 | 1, 4 | 1, 4 | 1, 4 | 1, 4 | 1 |
| EMAC | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| MSMC_Data_Master | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| QM Packet DMA | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| QM Second | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| DAP_Master | 1 | 1 | 1, 4 | 1, 4 | 1, 4 | 1, 4 | 1, 4 | 1, 4 | 1, 4 | 1, 4 | 1, 4 | 1, 4 | 1 | 1, 4 | 1, 4 | 1, 4 | 1, 4 | 1, 4 | 1, 4 | 1, 4 | 1, 4 | 1 |
| EDMA3CC | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| CorePac0_CFG | Y | Y | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | Y | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | Y |
| CorePac1_CFG (C6657 Only) | Y | Y | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | Y | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | Y |
| Tracer_Master | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| UPP | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |

5.3 TeraNet Switch Fabric Connections

The figures below show the connections between masters and slaves through various sections of the TeraNet.

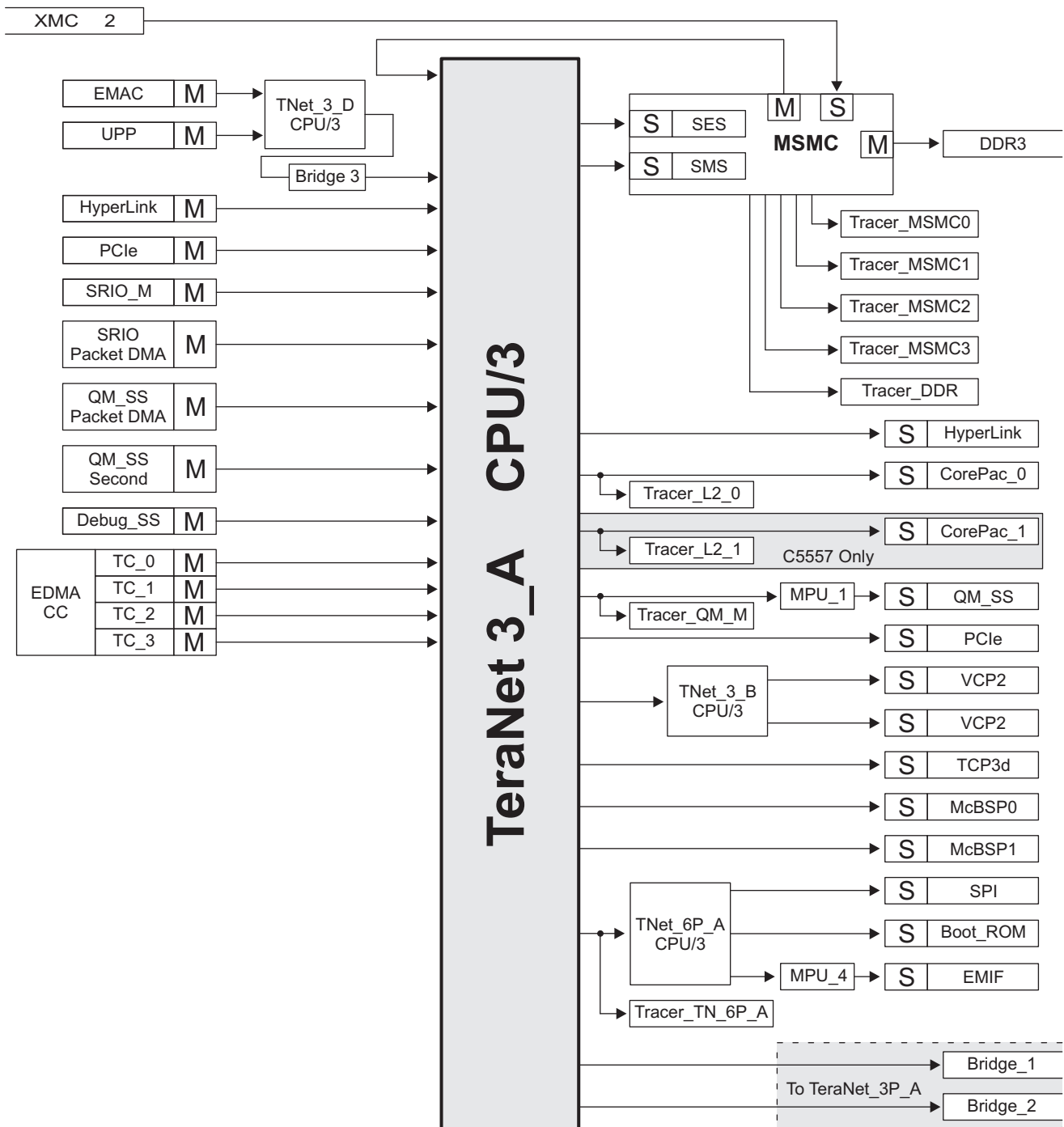


Figure 5-1. TeraNet 3A

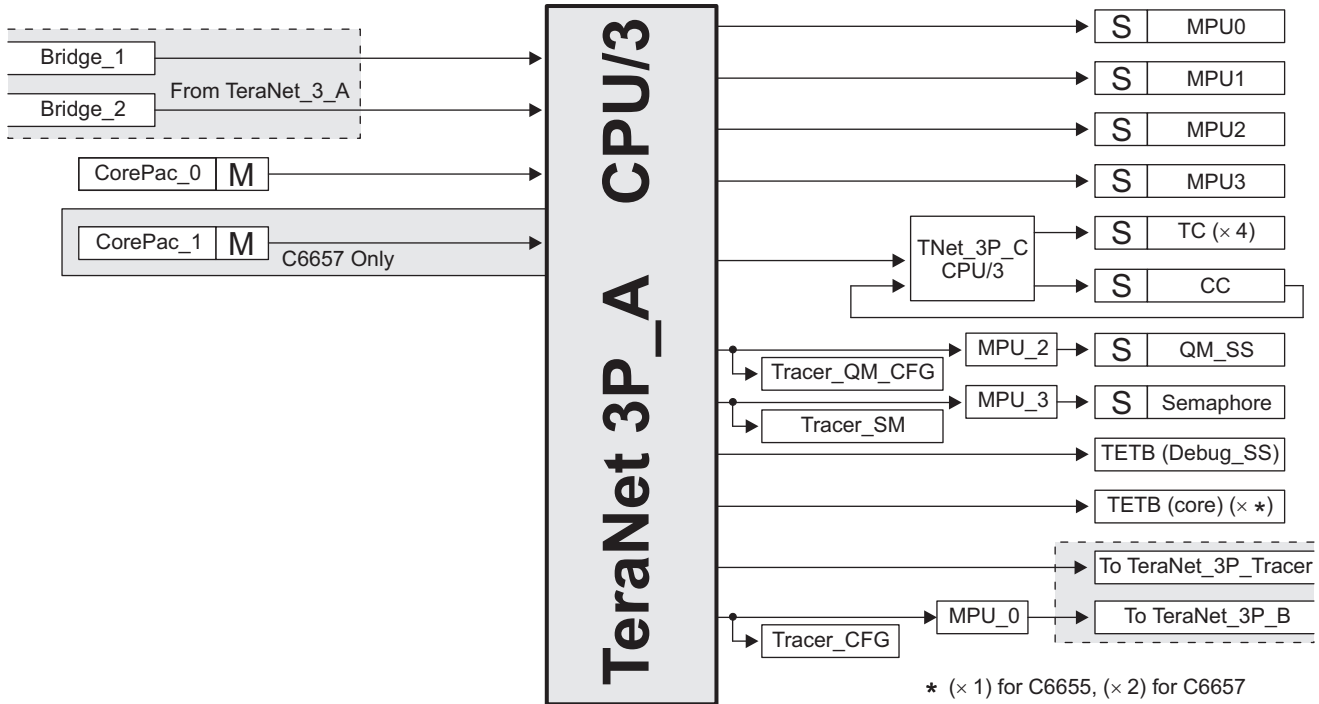


Figure 5-2. TeraNet 3P_A

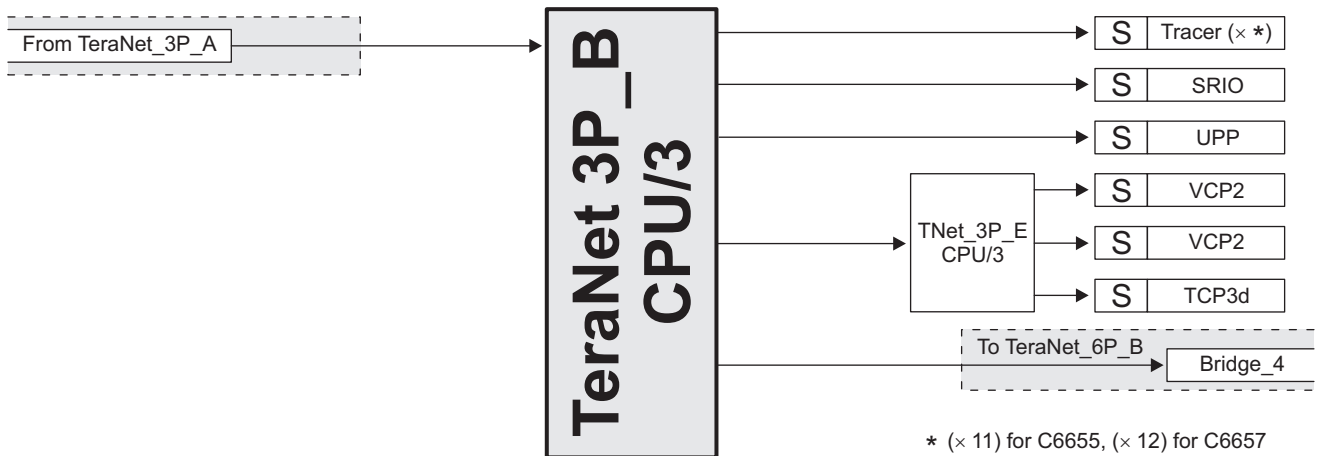


Figure 5-3. TeraNet 3P_B

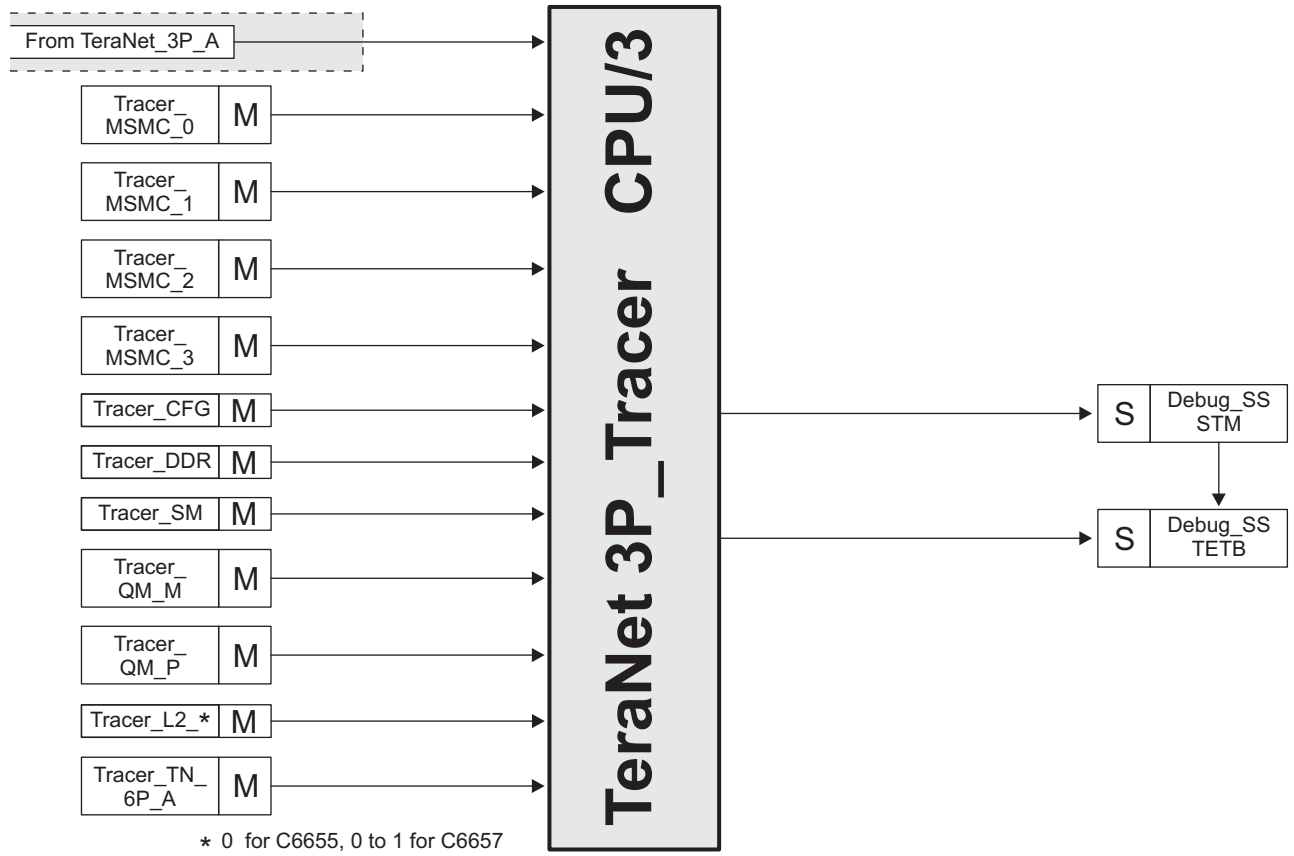


Figure 5-4. TeraNet 3P_Tracer

Table 5-3. Packet DMA Priority Allocation Register (PKTDMA_PRI_ALLOC) Field Descriptions

| Bit | Name | Description |
|------|------------|---|
| 31-3 | Reserved | Reserved |
| 2-0 | PKTDMA_PRI | Control the priority level for the transactions from packet DMA master port, which access the external linking RAM. |

5.4.2 EMAC / uPP Priority Allocation (EMAC_UPP_PRI_ALLOC) Register

The EMAC and uPP are master ports that do not have priority allocation registers inside the IP. The priority level for transaction from these master ports is described by EMAC_UPP_PRI_ALLOC register in [Figure 5-7](#) and [Table 5-4](#).

Figure 5-7. EMAC / uPP Priority Allocation Register (EMAC_UPP_PRI_ALLOC)

| 31 | 27 | 26 | 24 | 23 | 19 | 18 | 16 | 15 | 11 | 10 | 8 | 7 | 3 | 2 | 0 |
|----------|-----------|----------|----------|----------|----------|----------|----------|----------|---------|----------|---------|----------|---------|----------|---------|
| Reserved | EMAC_EPRI | Reserved | EMAC_PRI | Reserved | EMAC_PRI | Reserved | UPP_EPRI | Reserved | UPP_PRI | Reserved | UPP_PRI | Reserved | UPP_PRI | Reserved | UPP_PRI |
| R-00000 | RW-110 | R-00000 | RW-111 | R-00000 | RW-111 | R-00000 | RW-110 | R-00000 | RW-110 | R-00000 | RW-111 | R-00000 | RW-111 | R-00000 | RW-111 |

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

Table 5-4. EMAC / uPP Priority Allocation Register (EMAC_UPP_PRI_ALLOC) Field Descriptions

| Bit | Name | Description |
|-------|-----------|--|
| 31-27 | Reserved | Reserved |
| 26-24 | EMAC_EPRI | Control the maximum priority level for the transactions from EMAC master port. |
| 23-19 | Reserved | Reserved |
| 18-16 | EMAC_PRI | Control the priority level for the transactions from EMAC master port. |
| 15-11 | Reserved | Reserved |
| 10-8 | UPP_EPRI | Control the maximum priority level for the transactions from uPP master port. |
| 7-3 | Reserved | Reserved |
| 2-0 | UPP_PRI | Control the priority level for the transactions from uPP master port. |

6 C66x CorePac

The C66x CorePac consists of several components:

- The C66x DSP and associated C66x CorePac core
- Level-one and level-two memories (L1P, L1D, L2)
- Data Trace Formatter (DTF)
- Embedded Trace Buffer (ETB)
- Interrupt Controller
- Power-down controller
- External Memory Controller
- Extended Memory Controller
- A dedicated power/sleep controller (LPSC)

The C66x CorePac also provides support for memory protection, bandwidth management (for resources local to the C66x CorePac) and address extension. Figure 6-1 shows a block diagram of the C66x CorePac.

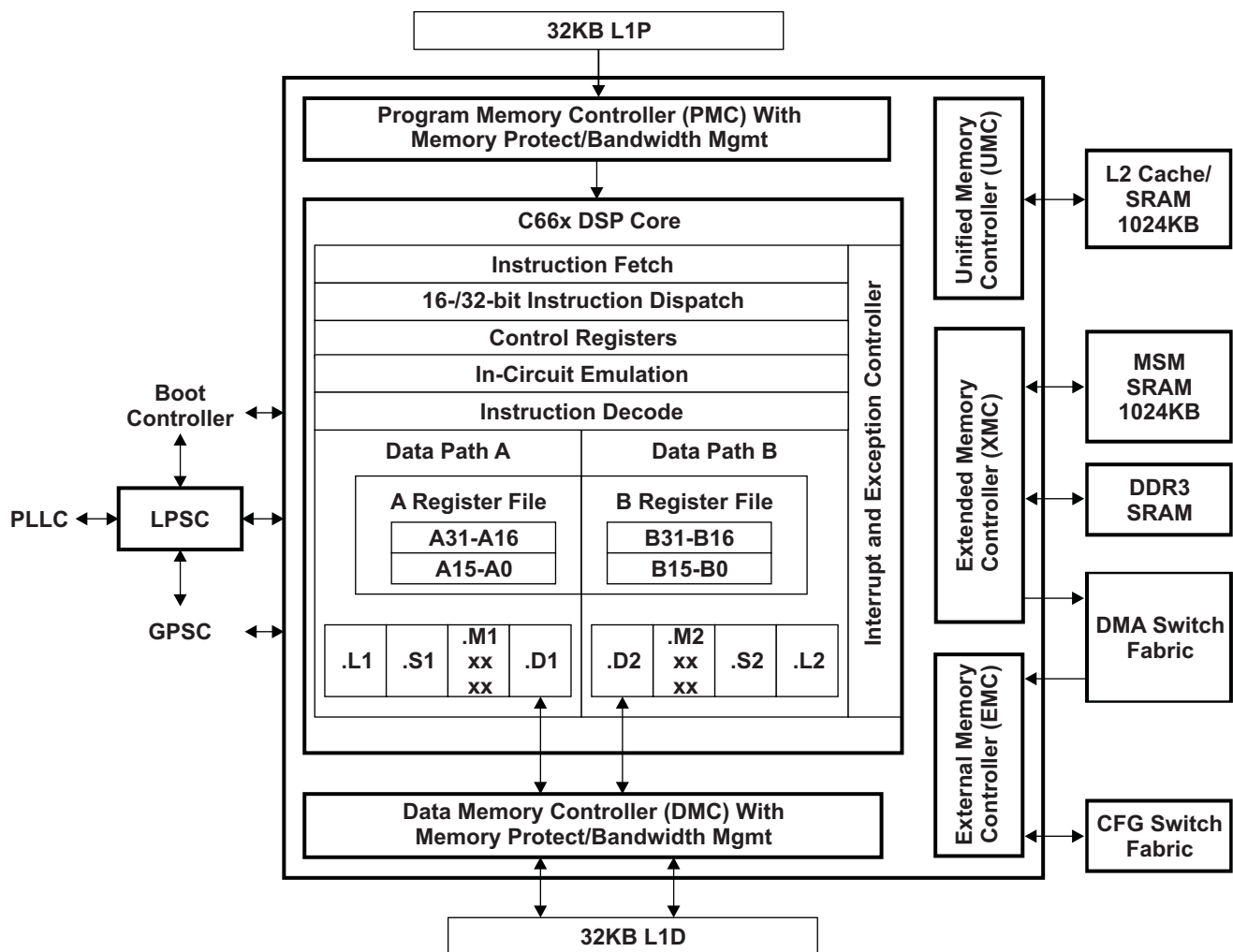


Figure 6-1. C66x CorePac Block Diagram

For more detailed information on the TMS320C66x CorePac on the C665x device, see the *C66x CorePac User's Guide* ([SPRUGW0](#)).

6.1 Memory Architecture

Each C66x CorePac of the device contains a 1024KB level-2 memory (L2), a 32KB level-1 program memory (L1P), and a 32KB level-1 data memory (L1D). The C665x devices also contain a 1024KB multicore shared memory (MSM). All memory on the C665x has a unique location in the memory map (see [Table 3-2](#)).

After device reset, L1P and L1D cache are configured as all cache, by default. The L1P and L1D cache can be reconfigured via software through the L1PMODE field of the L1P Configuration Register (L1PCFG) and the L1DMODE field of the L1D Configuration Register (L1DCFG) of the C66x CorePac. L1D is a two-way set-associative cache, while L1P is a direct-mapped cache.

The on-chip bootloader changes the reset configuration for L1P and L1D. For more information, see the *Bootloader for the C66x DSP User's Guide* ([SPRUGY5](#)).

For more information on the operation L1 and L2 caches, see the *C66x DSP Cache User's Guide* ([SPRUGY8](#)).

6.1.1 L1P Memory

The L1P memory configuration for the C665x device is as follows:

- 32K bytes with no wait states

[Figure 6-2](#) shows the available SRAM/cache configurations for L1P.

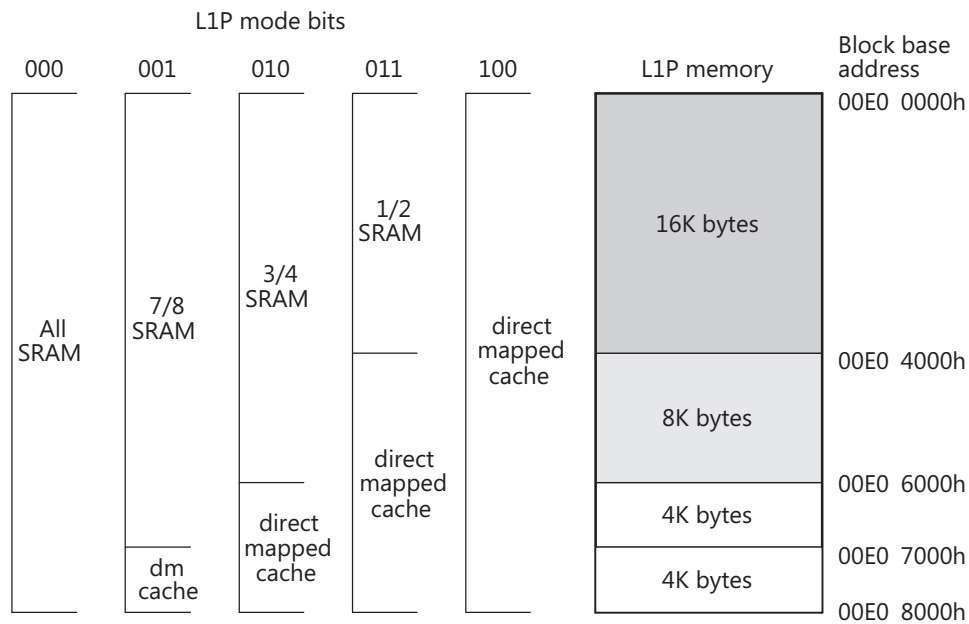


Figure 6-2. L1P Memory Configurations

6.1.2 L1D Memory

The L1D memory configuration for the C665x device is as follows:

- 32K bytes with no wait states

Figure 6-3 shows the available SRAM/cache configurations for L1D.

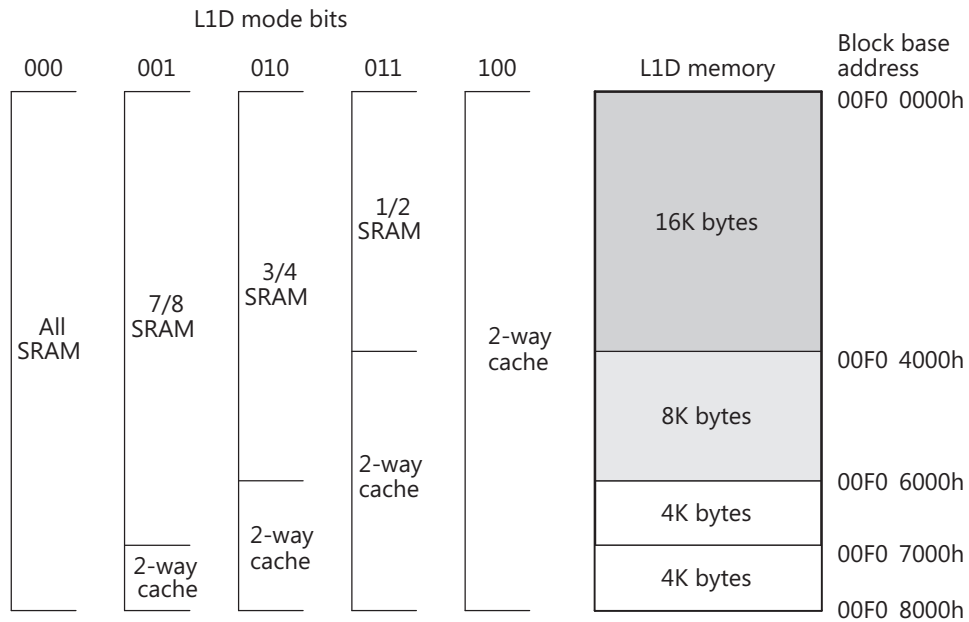


Figure 6-3. L1D Memory Configurations

6.1.3 L2 Memory

The L2 memory configuration for the C665x device is as follows:

- Total memory is 1024KB (C6655) or 2048KB (C6657)
- Each core contains 1024KB of memory
- Local starting address for each core is 0080 0000h

L2 memory can be configured as all SRAM, all 4-way set-associative cache, or a mix of the two. The amount of L2 memory that is configured as cache is controlled through the L2MODE field of the L2 Configuration Register (L2CFG) of the C66x CorePac. Figure 6-4 shows the available SRAM/cache configurations for L2. By default, L2 is configured as all SRAM after device reset.

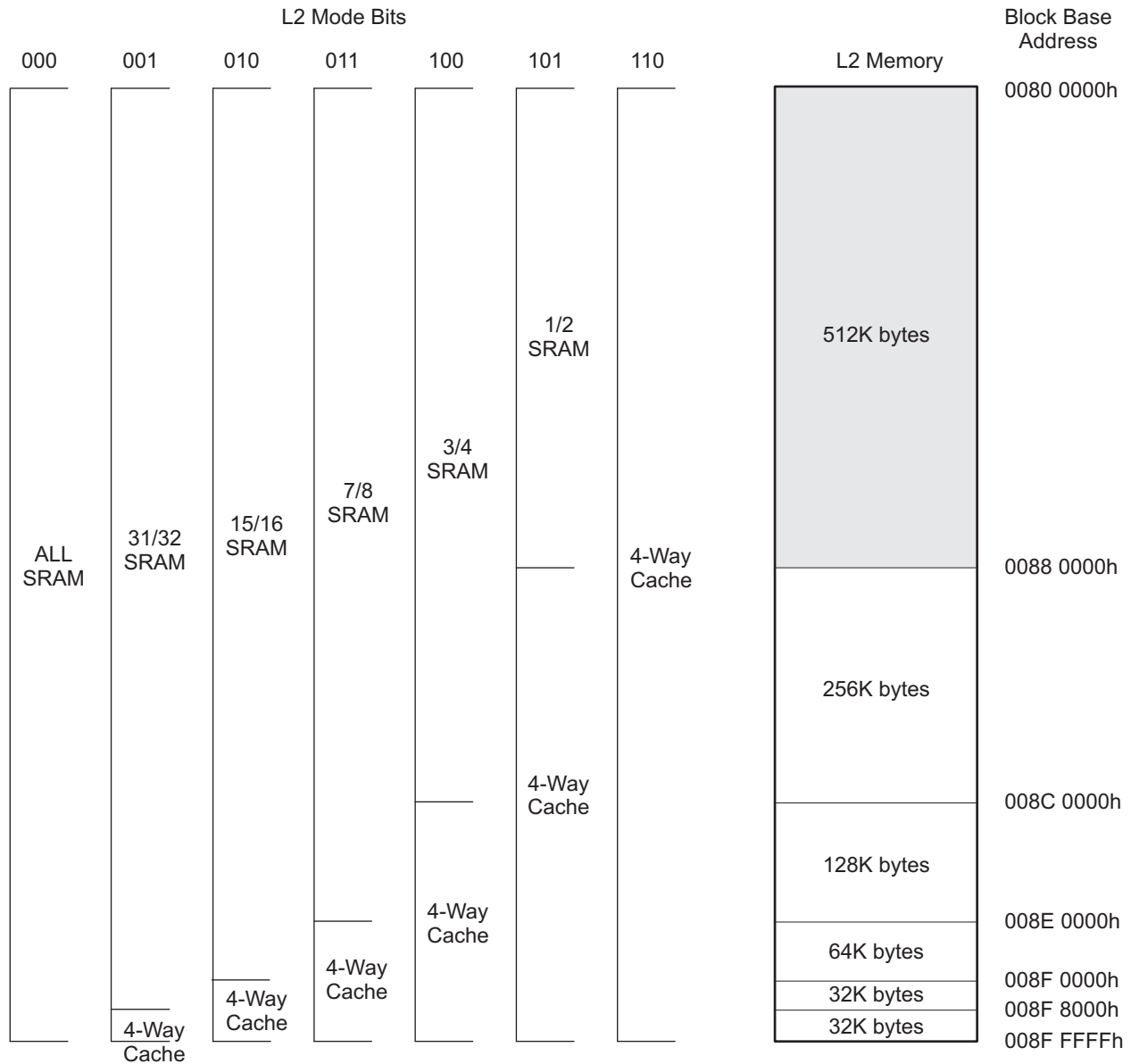


Figure 6-4. L2 Memory Configurations

Global addresses are accessible to all masters in the system. In addition, local memory can be accessed directly by the associated processor through aliased addresses, where the eight MSBs are masked to zero. The aliasing is handled within the C66x CorePac and allows for common code to be run unmodified on multiple cores. For example, address location 0x10800000 is the global base address for C66x CorePac Core 0's L2 memory. C66x CorePac Core 0 can access this location by either using 0x10800000 or 0x00800000. Any other master on the device must use 0x10800000 only. Conversely, 0x00800000 can be used by any of the cores as their own L2 base addresses.

For C66x CorePac Core 0, address 0x00800000 is equivalent to 0x10800000, and for C66x CorePac Core 1 (C6657 only) address 0x00800000 is equivalent to 0x11800000. Local addresses should be used only for shared code or data, allowing a single image to be included in memory. Any code/data targeted to a specific core, or a memory region allocated during run-time by a particular core should always use the global address only.

6.1.4 MSM SRAM

The MSM SRAM configuration for the device is as follows:

- Memory size is 1024KB
- The MSM SRAM can be configured as shared L2 and/or shared L3 memory
- Allows extension of external addresses from 2GB to up to 8GB
- Has built in memory protection features

The MSM SRAM is always configured as all SRAM. When configured as a shared L2, its contents can be cached in L1P and L1D. When configured in shared L3 mode, its contents can be cached in L2 also. For more details on external memory address extension and memory protection features, see the *Multicore Shared Memory Controller (MSMC) for KeyStone Devices User's Guide* ([SPRUGW7](#)).

6.1.5 L3 Memory

The L3 ROM on the device is 128KB. The ROM contains software used to boot the device. There is no requirement to block accesses from this portion to the ROM.

6.2 Memory Protection

Memory protection allows an operating system to define who or what is authorized to access L1D, L1P, and L2 memory. To accomplish this, the L1D, L1P, and L2 memories are divided into pages. There are 16 pages of L1P (2KB each), 16 pages of L1D (2KB each), and 32 pages of L2 (16KB each). The L1D, L1P, and L2 memory controllers in the C66x CorePac are equipped with a set of registers that specify the permissions for each memory page.

Each page may be assigned with fully orthogonal user and supervisor read, write, and execute permissions. In addition, a page may be marked as either (or both) locally accessible or globally accessible. A local access is a direct DSP access to L1D, L1P, and L2, while a global access is initiated by a DMA (either IDMA or the EDMA3) or by other system masters. Note that EDMA or IDMA transfers programmed by the DSP count as global accesses.

The DSP and each of the system masters on the device are all assigned a privilege ID. It is possible to specify whether memory pages are locally or globally accessible.

The AIDx and LOCAL bits of the memory protection page attribute registers specify the memory page protection scheme, see [Table 6-1](#).

Table 6-1. Available Memory Page Protection Schemes

| AIDx BIT | LOCAL BIT | DESCRIPTION |
|----------|-----------|--|
| 0 | 0 | No access to memory page is permitted. |
| 0 | 1 | Only direct access by DSP is permitted. |
| 1 | 0 | Only accesses by system masters and IDMA are permitted (includes EDMA and IDMA accesses initiated by the DSP). |
| 1 | 1 | All accesses permitted. |

Faults are handled by software in an interrupt (or an exception, programmable within the C66x CorePac interrupt controller) service routine. A DSP or DMA access to a page without the proper permissions will:

- Block the access — reads return 0, writes are ignored
- Capture the initiator in a status register — ID, address, and access type are stored
- Signal event to DSP interrupt controller

The software is responsible for taking corrective action to respond to the event and resetting the error status in the memory controller. For more information on memory protection for L1D, L1P, and L2, see the *C66x CorePac User's Guide* ([SPRUGW0](#)).

6.3 Bandwidth Management

When multiple requestors contend for a single C66x CorePac resource, the conflict is resolved by granting access to the highest priority requestor. The following four resources are managed by the Bandwidth Management control hardware:

- Level 1 Program (L1P) SRAM/Cache
- Level 1 Data (L1D) SRAM/Cache
- Level 2 (L2) SRAM/Cache
- Memory-mapped registers configuration bus

The priority level for operations initiated within the C66x CorePac are declared through registers in the C66x CorePac. These operations are:

- DSP-initiated transfers
- User-programmed cache coherency operations
- IDMA-initiated transfers

The priority level for operations initiated outside the C66x CorePac by system peripherals is declared through the Priority Allocation Register (PRI_ALLOC), see [Section 5.4](#) for more details. System peripherals with no fields in the PRI_ALLOC have their own registers to program their priorities.

More information on the bandwidth management features of the C66x CorePac can be found in the *C66x CorePac User's Guide* ([SPRUGW0](#)).

6.4 Power-Down Control

The C66x CorePac supports the ability to power down various parts of the C66x CorePac. The power down controller (PDC) of the C66x CorePac can be used to power down L1P, the cache control hardware, the DSP, and the entire C66x CorePac. These power-down features can be used to design systems for lower overall system power requirements.

NOTE

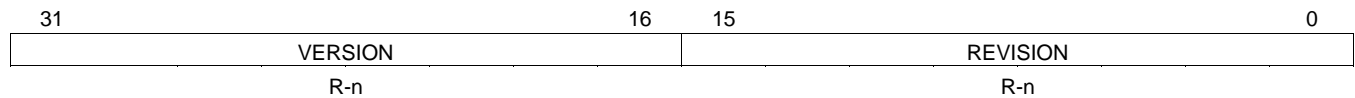
The C665x does not support power-down modes for the L2 memory at this time.

More information on the power-down features of the C66x CorePac can be found in the *C66x CorePac User's Guide* ([SPRUGW0](#)).

6.5 C66x CorePac Revision

The version and revision of the C66x CorePac can be read from the CorePac Revision ID Register (MM_REVID) located at address 0181 2000h. The MM_REVID register is shown in [Figure 6-5](#) and described in [Table 6-2](#). The C66x CorePac revision is dependent on the silicon revision being used.

Figure 6-5. CorePac Revision ID Register (MM_REVID) Address - 0181 2000h



Legend: R = Read; -n = value after reset

Table 6-2. CorePac Revision ID Register (MM_REVID) Field Descriptions

| Bit | Field | Description |
|-------|----------|---|
| 31-16 | VERSION | Version of the C66x CorePac implemented on the device. |
| 15-0 | REVISION | Revision of the C66x CorePac version implemented on the device. |

6.6 C66x CorePac Register Descriptions

See the *C66x CorePac Reference Guide* ([SPRUGW0](#)) for register offsets and definitions.

7 Device Operating Conditions

7.1 Absolute Maximum Ratings⁽¹⁾

Over Operating Case Temperature Range (Unless Otherwise Noted)

| | | |
|---|---|---|
| Supply voltage range ⁽²⁾ : | CVDD | -0.3 V to 1.3 V |
| | CVDD1 | -0.3 V to 1.3 V |
| | DVDD15 | -0.3 V to 2.45 V |
| | DVDD18 | -0.3 V to 2.45 V |
| | VREFSSTL | 0.49 × DVDD15 to 0.51 × DVDD15 |
| | VDDT1, VDDT2 | -0.3 V to 1.3 V |
| | VDDR1, VDDR2, VDDR3, VDDR4 | -0.3 V to 2.45 V |
| | AVDDA1, AVDDA2 | -0.3 V to 2.45 V |
| | VSS Ground | 0 V |
| Input voltage (V _I) range: | LVC MOS (1.8V) | -0.3 V to DVDD18+0.3 V |
| | DDR3 | -0.3 V to 2.45 V |
| | I ² C | -0.3 V to 2.45 V |
| | LVDS | -0.3 V to DVDD18+0.3 V |
| | LJCB | -0.3 V to 1.3 V |
| | SerDes | -0.3 V to CVDD1+0.3 V |
| Output voltage (V _O) range: | LVC MOS (1.8V) | -0.3 V to DVDD18+0.3 V |
| | DDR3 | -0.3 V to 2.45 V |
| | I ² C | -0.3 V to 2.45 V |
| | SerDes | -0.3 V to CVDD1+0.3 V |
| Operating case temperature range, T _C : | Commercial | 0°C to 85°C |
| | Extended | -40°C to 100°C |
| ESD stress voltage, V _{ESD} ⁽³⁾ : | HBM (human body model) ⁽⁴⁾ | ±1000 V |
| | CDM (charged device model) ⁽⁵⁾ | ±250 V |
| Overshoot/undershoot ⁽⁶⁾ | LVC MOS (1.8V) | 20% Overshoot/Undershoot for 20% of Signal Duty Cycle |
| | DDR3 | |
| | I ² C | |
| Storage temperature range, T _{stg} : | | -65°C to 150°C |

(1) Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

(2) All voltage values are with respect to V_{SS}.

(3) Electrostatic discharge (ESD) to measure device sensitivity/immunity to damage caused by electrostatic discharges into the device.

(4) Level listed above is the passing level per ANSI/ESDA/JEDEC JS-001-2010. JEDEC document JEP155 states that 500 V HBM allows safe manufacturing with a standard ESD control process, and manufacturing with less than 500 V HBM is possible if necessary precautions are taken. Pins listed as 1000 V may actually have higher performance.

(5) Level listed above is the passing level per EIA-JEDEC JESD22-C101E. JEDEC document JEP157 states that 250 V CDM allows safe manufacturing with a standard ESD control process. Pins listed as 250 V may actually have higher performance.

(6) Overshoot/Undershoot percentage relative to I/O operating values - for example the maximum overshoot value for 1.8-V LVC MOS signals is DVDD18 + 0.20 × DVDD18 and maximum undershoot value would be V_{SS} - 0.20 × DVDD18

7.2 Recommended Operating Conditions⁽¹⁾⁽²⁾

| | | MIN | NOM | MAX | UNIT | |
|---|--------------------------------------|------------------|------------------------------|---------------|----------------|-----|
| CVDD | SR Core Supply | 850MHz - Device | SRVnom ⁽³⁾ × 0.95 | 0.85-1.1 | SRVnom × 1.05 | V |
| | | 1000MHz - Device | SRVnom × 0.95 | 0.85-1.1 | SRVnom × 1.05 | |
| | | 1250MHZ - Device | SRVnom × 0.95 | 0.85-1.1 | SRVnom × 1.05 | |
| CVDD1 | Core supply voltage for memory array | 0.95 | 1 | 1.05 | V | |
| DVDD18 | 1.8-V supply I/O voltage | 1.71 | 1.8 | 1.89 | V | |
| DVDD15 | 1.5-V supply I/O voltage | 1.425 | 1.5 | 1.575 | V | |
| VREFSSTL | DDR3 reference voltage | 0.49 × DVDD15 | 0.5 × DVDD15 | 0.51 × DVDD15 | V | |
| V _{DDR_x} ⁽⁴⁾ | SerDes regulator supply | 1.425 | 1.5 | 1.575 | V | |
| V _{DDA_x} | PLL analog supply | 1.71 | 1.8 | 1.89 | V | |
| V _{DDT_x} | SerDes termination supply | 0.95 | 1 | 1.05 | V | |
| V _{SS} | Ground | 0 | 0 | 0 | V | |
| V _{IH} | High-level input voltage | LVC MOS (1.8 V) | 0.65 × DVDD18 | | | V |
| | | I ² C | 0.7 × DVDD18 | | | V |
| | | DDR3 EMIF | VREFSSTL + 0.1 | | | V |
| V _{IL} | Low-level input voltage | LVC MOS (1.8 V) | | 0.35 × DVDD18 | | V |
| | | DDR3 EMIF | -0.3 | | VREFSSTL - 0.1 | V |
| | | I ² C | | | 0.3 × DVDD18 | V |
| T _C | Operating case temperature | Commercial | 0 | | 85 | °C |
| | | Extended | -40 | | | 100 |

(1) All differential clock inputs comply with the LVDS Electrical Specification, IEEE 1596.3-1996 and all SERDES I/Os comply with the XAUI Electrical Specification, IEEE 802.3ae-2002.

(2) All SERDES I/Os comply with the XAUI Electrical Specification, IEEE 802.3ae-2002.

(3) SRVnom refers to the unique SmartReflex core supply voltage set from the factory for each individual device.

(4) Where x = 1, 2, 3, 4... to indicate all supplies of the same kind.

7.3 Electrical Characteristics

Over Recommended Ranges of Supply Voltage and Operating Case Temperature (Unless Otherwise Noted)

| PARAMETER | | TEST CONDITIONS ⁽¹⁾ | MIN | NOM | MAX | UNIT | | |
|--------------------------------|--------------------------------|---------------------------------|---|-----|--|------|----|-----|
| V _{OH} | High-level output voltage | LVC MOS (1.8 V) | DVDD18 - 0.45 | | | V | | |
| | | DDR3 | DVDD15 - 0.4 | | | | | |
| | | I ² C ⁽²⁾ | | | | | | |
| V _{OL} | Low-level output voltage | LVC MOS (1.8 V) | 0.45 | | | V | | |
| | | DDR3 | 0.4 | | | | | |
| | | I ² C | I _O = 3 mA, pulled up to 1.8 V | | | | | |
| I _I ⁽³⁾ | Input current [DC] | LVC MOS (1.8 V) | No IPD/IPU | | | μA | | |
| | | | Internal pullup | | | | -5 | 5 |
| | | | Internal pulldown | | | | 50 | 100 |
| | | I ² C | | | 0.1 × DVDD18 V < V _I < 0.9 × DVDD18 V | | | |
| I _{OH} | High-level output current [DC] | LVC MOS (1.8 V) | | | | mA | | |
| | | DDR3 | | | | | | |
| | | I ² C ⁽⁵⁾ | | | | | | |
| I _{OL} | Low-level output current [DC] | LVC MOS (1.8 V) | 6 | | | mA | | |
| | | DDR3 | 8 | | | | | |
| | | I ² C | 3 | | | | | |
| I _{OZ} ⁽⁶⁾ | Off-state output current [DC] | LVC MOS (1.8 V) | -2 | | | μA | | |
| | | DDR3 | -2 | | | | | |
| | | I ² C | -2 | | | | | |

(1) For test conditions shown as MIN, MAX, or TYP, use the appropriate value specified in the recommended operating conditions table.

(2) I²C uses open collector IOs and does not have a V_{OH} Minimum.

(3) I_I applies to input-only pins and bi-directional pins. For input-only pins, I_I indicates the input leakage current. For bi-directional pins, I_I includes input leakage current and off-state (Hi-Z) output leakage current.

(4) For RESETSTAT, max DC input current is 300 μA.

(5) I²C uses open collector IOs and does not have a I_{OH} Maximum.

(6) I_{OZ} applies to output-only pins, indicating off-state (Hi-Z) output leakage current.

7.4 Power Supply to Peripheral I/O Mapping⁽¹⁾⁽²⁾

Over Recommended Ranges of Supply Voltage and Operating Case Temperature (Unless Otherwise Noted)

| POWER SUPPLY | | I/O BUFFER TYPE | ASSOCIATED PERIPHERAL |
|--------------|--|--------------------------------|---|
| CVDD | Supply Core Voltage | LJCB | CORECLK(P N) PLL input buffers |
| | | | SRIOSGMIICLK(P N) SerDes PLL input buffers |
| | | | DDRCLK(P N) PLL input buffers |
| | | | PCIECLK(P N) SERDES PLL input buffers |
| CVDD | Supply Core Voltage | LJCB | MCMCLK(P N) SERDES PLL input buffers |
| DVDD15 | 1.5-V supply I/O voltage | DDR3 (1.5 V) | All DDR3 memory controller peripheral I/O buffers |
| DVDD18 | 1.8-V supply I/O voltage | LVCMOS (1.8 V) | All GPIO peripheral I/O buffers |
| | | | All JTAG and EMU peripheral I/O buffers |
| | | | All Timer peripheral I/O buffers |
| | | | All SPI peripheral I/O buffers |
| | | | All RESETs, NMI, Control peripheral I/O buffers |
| | | | All MDIO peripheral I/O buffers |
| | | | All UART peripheral I/O buffers |
| | | | All McBSP peripheral I/O buffers |
| | | | All EMIF16 peripheral I/O buffers |
| | | All uPP peripheral I/O buffers | |
| DVDD18 | 1.8-V supply I/O voltage | Open-drain (1.8V) | All I ² C peripheral I/O buffers |
| | | | All SmartReflex peripheral I/O buffers |
| DVDD18 | 1.8-V supply I/O voltage | LVCMOS (1.8 V) | All Hyperlink sideband peripheral I/O buffers |
| VDDT1 | Hyperlink SerDes termination and analogue front-end supply | SerDes/CML | Hyperlink SerDes CML IO buffers |
| VDDT2 | SRIO/SGMII/PCIE SerDes termination and analogue front-end supply | SerDes/CML | SRIO/SGMII/PCIE SerDes CML IO buffers |

(1) Please note that this table does not attempt to describe all functions of all power supply terminals but only those whose purpose it is to power peripheral I/O buffers and clock input buffers.

(2) Please see the *Hardware Design Guide for KeyStone Devices* ([SPRABI2](#)) for more information about individual peripheral I/O.

8 Peripheral Information and Electrical Specifications

This chapter covers the various peripherals on the C665x DSP. Peripheral-specific information, timing diagrams, electrical specifications, and register memory maps are described in this chapter.

8.1 Recommended Clock and Control Signal Transition Behavior

All clocks and control signals **must** transition between V_{IH} and V_{IL} (or between V_{IL} and V_{IH}) in a monotonic manner.

8.2 Power Supplies

The following sections describe the proper power-supply sequencing and timing needed to properly power on the C665x. The various power supply rails and their primary function is listed in [Table 8-1](#).

Table 8-1. Power Supply Rails on C665x

| NAME | PRIMARY FUNCTION | VOLTAGE | NOTES |
|----------|---|----------------|---|
| CVDD | SmartReflex core supply voltage | 0.85 V - 1.1 V | Includes core voltage for DDR3 module |
| CVDD1 | Core supply voltage for memory array | 1.0 V | Fixed supply at 1.0 V |
| VDDT1 | HyperLink SerDes termination supply | 1.0 V | Filtered version of CVDD1. Special considerations for noise. Filter is not needed if HyperLink is not in use. |
| VDDT2 | SGMII/SRIO/PCIE SerDes termination supply | 1.0 V | Filtered version of CVDD1. Special considerations for noise. Filter is not needed if SGMII/SRIO/PCIE is not in use. |
| DVDD15 | 1.5-V DDR3 IO supply | 1.5 V | |
| VDDR1 | HyperLink SerDes regulator supply | 1.5 V | Filtered version of DVDD15. Special considerations for noise. Filter is not needed if HyperLink is not in use. |
| VDDR2 | PCIE SerDes regulator supply | 1.5 V | Filtered version of DVDD15. Special considerations for noise. Filter is not needed if PCIE is not in use. |
| VDDR3 | SGMII SerDes regulator supply | 1.5 V | Filtered version of DVDD15. Special considerations for noise. Filter is not needed if SGMII is not in use. |
| VDDR4 | SRIO SerDes regulator supply | 1.5 V | Filtered version of DVDD15. Special considerations for noise. Filter is not needed if HyperLink is not in use. |
| DVDD18 | 1.8-V IO supply | 1.8V | |
| AVDDA1 | Main PLL supply | 1.8 V | Filtered version of DVDD18. Special considerations for noise. |
| AVDDA2 | DDR3 PLL supply | 1.8 V | Filtered version of DVDD18. Special considerations for noise. |
| VREFSSTL | 0.75-V DDR3 reference voltage | 0.75 V | Should track the 1.5-V supply. Use 1.5 V as source. |
| VSS | Ground | GND | |

8.2.1 Power-Supply Sequencing

This section defines the requirements for a power up sequencing from a power-on reset condition. There are two acceptable power sequences for the device. The first sequence stipulates the core voltages starting before the IO voltages as shown below.

1. CVDD
2. CVDD1, VDDT1-2
3. DVDD18, AVDDA1, AVDDA2
4. DVDD15, VDDR1-4

The second sequence provides compatibility with other TI processors with the IO voltage starting before the core voltages as shown below.

1. DVDD18, AVDDA1, AVDDA2
2. CVDD
3. CVDD1, VDDT1-2
4. DVDD15, VDDR1-4

The clock input buffers for CORECLK, DDRCLK, SRIOSGMICLK, MCMCLK, and PCIECLK use only CVDD as a supply voltage. These clock inputs are not failsafe and must be held in a high-impedance state until CVDD is at a valid voltage level. Driving these clock inputs high before CVDD is valid could cause damage to the device. Once CVDD is valid it is acceptable that the P and N legs of these CLKs may be held in a static state (either high and low or low and high) until a valid clock frequency is needed at that input. To avoid internal oscillation the clock inputs should be removed from the high impedance state shortly after CVDD is present.

If a clock input is not used it must be held in a static state. To accomplish this the N leg should be pulled to ground through a 1K ohm resistor. The P leg should be tied to CVDD to ensure it won't have any voltage present until CVDD is active. Connections to the IO cells powered by DVDD18 and DVDD15 are not failsafe and should not be driven high before these voltages are active. Driving these IO cells high before DVDD18 or DVDD15 are valid could cause damage to the device.

The device initialization is broken into two phases. The first phase consists of the time period from the activation of the first power supply until the point in which all supplies are active and at a valid voltage level. Either of the sequencing scenarios described above can be implemented during this phase. The figures below show both the core-before-IO voltage sequence and the IO-before-core voltage sequence. $\overline{\text{POR}}$ must be held low for the entire power stabilization phase.

This is followed by the device initialization phase. The rising edge of $\overline{\text{POR}}$ followed by the rising edge of $\overline{\text{RESETFULL}}$ will trigger the end of the initialization phase but both must be inactive for the initialization to complete. $\overline{\text{POR}}$ must always go inactive before $\overline{\text{RESETFULL}}$ goes inactive as described below. SYSCLK1 in the following section refers to the clock input that has been selected as the source for the main PLL and SYSCLK1 refers to the main PLL output that is used by the CorePac, see [Figure 8-7](#) for more details.

8.2.1.1 Core-Before-IO Power Sequencing

Figure 8-1 shows the power sequencing and reset control of C665x for device initialization. $\overline{\text{POR}}$ may be removed after the power has been stable for the required 100 μsec . $\overline{\text{RESETFULL}}$ must be held low for a period after the rising edge of $\overline{\text{POR}}$ but may be held low for longer periods if necessary. The configuration bits shared with the GPIO pins will be latched on the rising edge of $\overline{\text{RESETFULL}}$ and must meet the setup and hold times specified. SYSCLK1 must always be active before $\overline{\text{POR}}$ can be removed. Core-before-IO power sequencing is defined in Table 8-2.

NOTE

TI recommends a maximum of 100 ms between one power rail being valid, and the next power rail in the sequence starting to ramp.

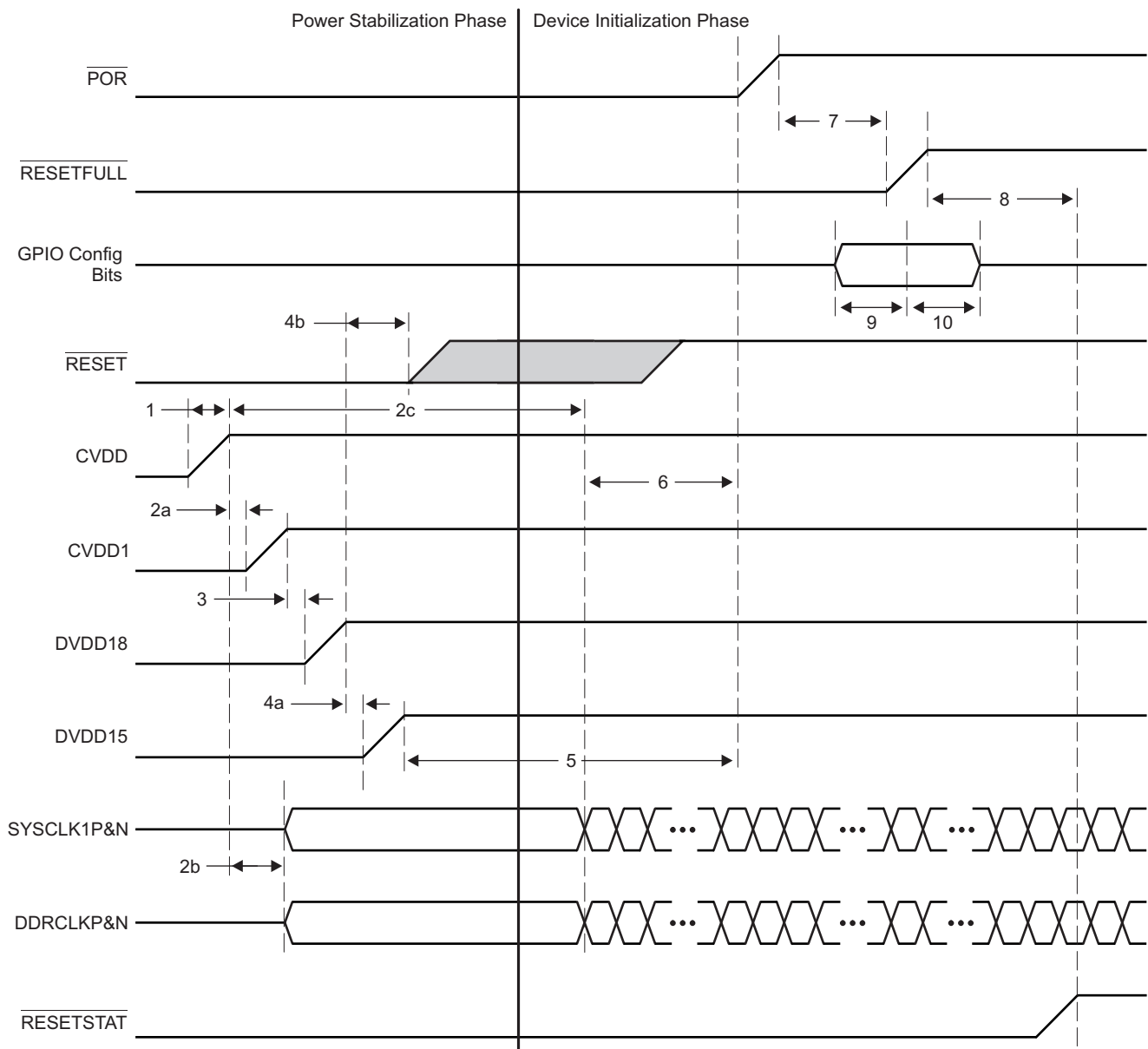


Figure 8-1. Core Before IO Power Sequencing

Table 8-2. Core Before IO Power Sequencing

| TIME | SYSTEM STATE |
|------|---|
| 1 | Begin Power Stabilization Phase <ul style="list-style-type: none"> CVDD (core AVS) ramps up. $\overline{\text{POR}}$ must be held low through the power stabilization phase. Because $\overline{\text{POR}}$ is low, all the core logic that has async reset (created from POR) is put into the reset state. |
| 2a | <ul style="list-style-type: none"> CVDD1 (core constant) ramps at the same time or shortly following CVDD. Although ramping CVDD1 and CVDD simultaneously is permitted, the voltage for CVDD1 must never exceed CVDD until after CVDD has reached a valid voltage. The purpose of ramping up the core supplies close to each other is to reduce crowbar current. CVDD1 should trail CVDD as this will ensure that the WLs in the memories are turned off and there is no current through the memory bit cells. If, however, CVDD1 (core constant) ramps up before CVDD (core AVS), then the worst-case current could be on the order of twice the specified draw of CVDD1. |
| 2b | <ul style="list-style-type: none"> Once CVDD is valid, the clock drivers should be enabled. Although the clock inputs are not necessary at this time, they should either be driven with a valid clock or be held in a static state with one leg high and one leg low. |
| 2c | <ul style="list-style-type: none"> The DDRCLK and SYCLK1 may begin to toggle anytime between when CVDD is at a valid level and the setup time before $\overline{\text{POR}}$ goes high specified by t6. |
| 3 | <ul style="list-style-type: none"> Filtered versions of 1.8 V can ramp simultaneously with DVDD18. RESETSTAT is driven low once the DVDD18 supply is available. All LVC MOS input and bidirectional pins must not be driven or pulled high until DVDD18 is present. Driving an input or bidirectional pin before DVDD18 is valid could cause damage to the device. |
| 4a | <ul style="list-style-type: none"> DVDD15 (1.5 V) supply is ramped up following DVDD18. Although ramping DVDD18 and DVDD15 simultaneously is permitted, the voltage for DVDD15 must never exceed DVDD18. |
| 4b | <ul style="list-style-type: none"> $\overline{\text{RESET}}$ may be driven high any time after DVDD18 is at a valid level. In a $\overline{\text{POR}}$-controlled boot, $\overline{\text{RESET}}$ must be high before $\overline{\text{POR}}$ is driven high. |
| 5 | <ul style="list-style-type: none"> $\overline{\text{POR}}$ must continue to remain low for at least 100 μs after power has stabilized. End Power Stabilization Phase |
| 6 | <ul style="list-style-type: none"> Device initialization requires 500 SYCLK1 periods after the Power Stabilization Phase. The maximum clock period is 33.33 nsec, so a delay of an additional 16 μs is required before a rising edge of $\overline{\text{POR}}$. The clock must be active during the entire 16 μs. |
| 7 | <ul style="list-style-type: none"> $\overline{\text{RESETFULL}}$ must be held low for at least 24 transitions of the SYCLK1 after $\overline{\text{POR}}$ has stabilized at a high level. |
| 8 | <ul style="list-style-type: none"> The rising edge of the $\overline{\text{RESETFULL}}$ will remove the reset to the efuse farm allowing the scan to begin. Once device initialization and the efuse farm scan are complete, the $\overline{\text{RESETSTAT}}$ signal is driven high. This delay will be 10000 to 50000 clock cycles. End Device Initialization Phase |
| 9 | <ul style="list-style-type: none"> GPIO configuration bits must be valid for at least 12 transitions of the SYCLK1 before the rising edge of $\overline{\text{RESETFULL}}$ |
| 10 | <ul style="list-style-type: none"> GPIO configuration bits must be held valid for at least 12 transitions of the SYCLK1 after the rising edge of $\overline{\text{RESETFULL}}$ |

8.2.1.2 IO-Before-Core Power Sequencing

The timing diagram for IO-before-core power sequencing is shown in [Figure 8-2](#) and defined in [Table 8-3](#).

NOTE

TI recommends a maximum of 100 ms between one power rail being valid, and the next power rail in the sequence starting to ramp.

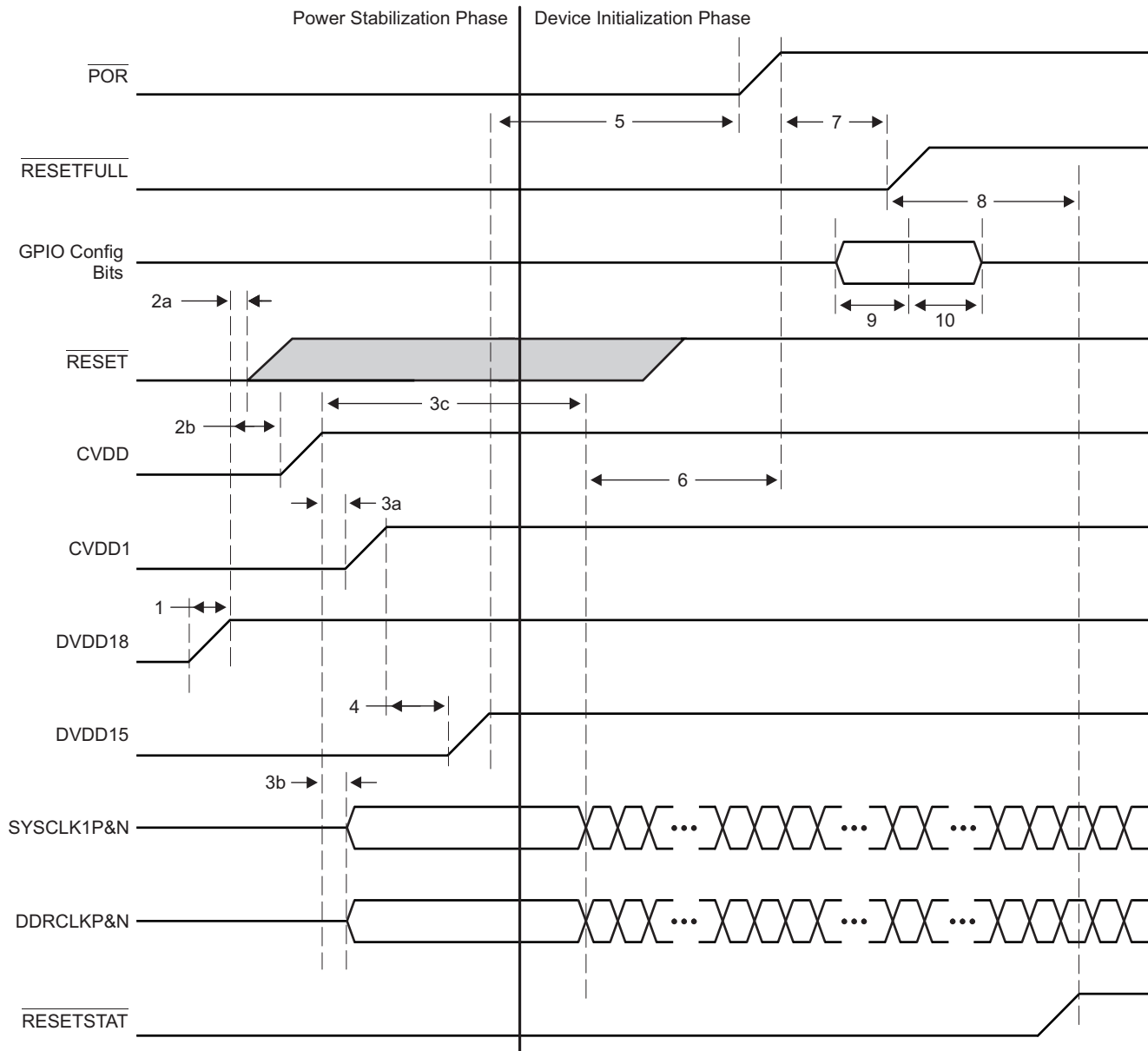


Figure 8-2. IO Before Core Power Sequencing

Table 8-3. IO Before Core Power Sequencing

| TIME | SYSTEM STATE |
|------|--|
| 1 | Begin Power Stabilization Phase <ul style="list-style-type: none"> Because $\overline{\text{POR}}$ is low, all the core logic having async reset (created from POR) are put into reset state once the core supply ramps. POR must remain low through Power Stabilization Phase. Filtered versions of 1.8 V can ramp simultaneously with DVDD18. $\overline{\text{RESETSTAT}}$ is driven low once the DVDD18 supply is available. All input and bidirectional pins must not be driven or pulled high until DVDD18 is present. Driving an input or bidirectional pin before DVDD18 could cause damage to the device. |
| 2a | <ul style="list-style-type: none"> $\overline{\text{RESET}}$ may be driven high anytime after DVDD18 is at a valid level. |
| 2b | <ul style="list-style-type: none"> CVDD (core AVS) ramps up. |
| 3a | <ul style="list-style-type: none"> CVDD1 (core constant) ramps at the same time or following CVDD. Although ramping CVDD1 and CVDD simultaneously is permitted the voltage for CVDD1 must never exceed CVDD until after CVDD has reached a valid voltage. The purpose of ramping up the core supplies close to each other is to reduce crowbar current. CVDD1 should trail CVDD as this will ensure that the WLs in the memories are turned off and there is no current through the memory bit cells. If, however, CVDD1 (core constant) ramps up before CVDD (core AVS), then the worst case current could be on the order of twice the specified draw of CVDD1. |
| 3b | <ul style="list-style-type: none"> Once CVDD is valid, the clock drivers should be enabled. Although the clock inputs are not necessary at this time, they should either be driven with a valid clock or held in a static state with one leg high and one leg low. |
| 3c | <ul style="list-style-type: none"> The DDRCLK and SYSCLK1 may begin to toggle anytime between when CVDD is at a valid level and the setup time before $\overline{\text{POR}}$ goes high specified by t6. |
| 4 | <ul style="list-style-type: none"> DVDD15 (1.5 V) supply is ramped up following CVDD1. |
| 5 | <ul style="list-style-type: none"> $\overline{\text{POR}}$ must continue to remain low for at least 100 μs after power has stabilized. End Power Stabilization Phase |
| 6 | Begin Device Initialization <ul style="list-style-type: none"> Device initialization requires 500 SYSCLK1 periods after the Power Stabilization Phase. The maximum clock period is 33.33 nsec so a delay of an additional 16 μs is required before a rising edge of $\overline{\text{POR}}$. The clock must be active during the entire 16 μs. $\overline{\text{POR}}$ must remain low. |
| 7 | <ul style="list-style-type: none"> $\overline{\text{RESETFULL}}$ is held low for at least 24 transitions of the SYSCLK1 after $\overline{\text{POR}}$ has stabilized at a high level. The rising edge of the $\overline{\text{RESETFULL}}$ will remove the reset to the efuse farm allowing the scan to begin. |
| 8 | <ul style="list-style-type: none"> Once device initialization and the efuse farm scan are complete, the $\overline{\text{RESETSTAT}}$ signal is driven high. This delay will be 10000 to 50000 clock cycles. End Device Initialization Phase |
| 9 | <ul style="list-style-type: none"> GPIO configuration bits must be valid for at least 12 transitions of the SYSCLK1 before the rising edge of $\overline{\text{RESETFULL}}$ |
| 10 | <ul style="list-style-type: none"> GPIO configuration bits must be held valid for at least 12 transitions of the SYSCLK1 after the rising edge of $\overline{\text{RESETFULL}}$ |

8.2.1.3 Prolonged Resets

Holding the device in $\overline{\text{POR}}$, $\overline{\text{RESETFULL}}$, or $\overline{\text{RESET}}$ for long periods of time will affect the long term reliability of the part. The device should not be held in a reset for times exceeding one hour and should not be held in reset for more the 5% of the time during which power is applied. Exceeding these limits will cause a gradual reduction in the reliability of the part. This can be avoided by allowing the DSP to boot and then configuring it to enter a hibernation state soon after power is applied. This will satisfy the reset requirement while limiting the power consumption of the device.

8.2.1.4 Clocking During Power Sequencing

Some of the clock inputs are required to be present for the device to initialize correctly, but behavior of many of the clocks is contingent on the state of the boot configuration pins. [Table 8-4](#) describes the clock sequencing and the conditions that affect the clock operation. Note that all clock drivers should be in a high-impedance state until CVDD is at a valid level and that all clock inputs either be active or in a static state with one leg pulled low and the other connected to CVDD.

Table 8-4. Clock Sequencing

| CLOCK | CONDITION | SEQUENCING |
|---------------|---|---|
| DDRCLK | None | Must be present 16 μ sec before $\overline{\text{POR}}$ transitions high. |
| CORECLK | None | CORECLK used to clock the core PLL. It must be present 16 μ sec before $\overline{\text{POR}}$ transitions high. |
| SRIOSGMII CLK | The SGMII port will be used. | SRIOSGMIICLK must be present 16 μ sec before $\overline{\text{POR}}$ transitions high. |
| | SGMII will not be used. SRIO will not be used. | SRIOSGMIICLK is not used and should be tied to a static state. |
| SRIOSGMII CLK | SGMII will not be used. SRIO will be used as a boot device. | SRIOSGMIICLK must be present 16 μ sec before $\overline{\text{POR}}$ transitions high. |
| | SGMII will not be used. SRIO will be used after boot. | SRIOSGMIICLK is used as a source to the SRIO SERDES PLL. It must be present before the SRIO is removed from reset and programmed. |
| PCIECLK | PCIE will be used as a boot device. | PCIECLK must be present 16 μ sec before $\overline{\text{POR}}$ transitions high. |
| | PCIE will be used after boot. | PCIECLK is used as a source to the PCIE SERDES PLL. It must be present before the PCIE is removed from reset and programmed. |
| | PCIE will not be used. | PCIECLK is not used and should be tied to a static state. |
| MCMCLK | HyperLink will be used as a boot device. | MCMCLK must be present 16 μ sec before $\overline{\text{POR}}$ transitions high. |
| | HyperLink will be used after boot. | MCMCLK is used as a source to the MCM SERDES PLL. It must be present before the HyperLink is removed from reset and programmed. |
| | HyperLink will not be used. | MCMCLK is not used and should be tied to a static state. |

8.2.2 Power-Down Sequence

The power down sequence is the exact reverse of the power-up sequence described above. The goal is to prevent a large amount of static current and to prevent overstress of the device. A power-good circuit that monitors all the supplies for the device should be used in all designs. If a catastrophic power supply failure occurs on any voltage rail, $\overline{\text{POR}}$ should transition to low to prevent over-current conditions that could possibly impact device reliability.

A system power monitoring solution is needed to shut down power to the board if a power supply fails. Long-term exposure to an environment in which one of the power supply voltages is no longer present will affect the reliability of the device. Holding the device in reset is not an acceptable solution because prolonged periods of time with an active reset can also affect long term reliability.

8.2.3 Power Supply Decoupling and Bulk Capacitors

In order to properly decouple the supply planes on the PCB from system noise, decoupling and bulk capacitors are required. Bulk capacitors are used to minimize the effects of low frequency current transients and decoupling or bypass capacitors are used to minimize higher frequency noise. For recommendations on selection of Power Supply Decoupling and Bulk capacitors see the *Hardware Design Guide for KeyStone Devices* ([SPRAB12](#)).

8.2.4 SmartReflex

Increasing the device complexity increases its power consumption and with the smaller transistor structures responsible for higher achievable clock rates and increased performance, comes an inevitable penalty, increasing the leakage currents. Leakage currents are present in any active circuit, independently of clock rates and usage scenarios. This static power consumption is mainly determined by transistor type and process technology. Higher clock rates also increase dynamic power, the power used when transistors switch. The dynamic power depends mainly on a specific usage scenario, clock rates, and I/O activity.

Texas Instruments' SmartReflex technology is used to decrease both static and dynamic power consumption while maintaining the device performance. SmartReflex in the C665x device is a feature that allows the core voltage to be optimized based on the process corner of the device. This requires a voltage regulator for each C665x device.

To guarantee maximizing performance and minimizing power consumption of the device, SmartReflex is required to be implemented whenever the C665x device is used. The voltage selection is done using 4 VCNTL pins which are used to select the output voltage of the core voltage regulator.

For information on implementation of SmartReflex see the *Power Management for KeyStone Devices* application report and the *Hardware Design Guide for KeyStone Devices* ([SPRABI2](#)).

Table 8-5. SmartReflex 4-Pin VID Interface Switching Characteristics

(see [Figure 8-3](#))

| NO. | PARAMETER | MIN | MAX | UNIT |
|-----|---|------|------------------------|------|
| 1 | td(VCNTL[2:0]-VCNTL[3]) Delay Time - VCNTL[2:0] valid after VCNTL[3] low | | 300.00 | ns |
| 2 | toh(VCNTL[3] -VCNTL[2:0]) Output Hold Time - VCNTL[2:0] valid after VCNTL[3] low | 0.07 | 172020C ⁽¹⁾ | ms |
| 3 | td(VCNTL[2:0]-VCNTL[3]) Delay Time - VCNTL[2:0] valid after VCNTL[3] high | | 300.00 | ns |
| 4 | toh(VCNTL[3] -VCNTL[2:0]) Output Hold Time - VCNTL[2:0] valid after VCNTL[3] high | 0.07 | 172020C | ms |
| 5 | VCNTL being valid to CVDD being switched to SmartReflex Voltage ⁽²⁾ | | 10 | ms |

(1) C = 1/SYSCLK1 frequency (See [Figure 8-9](#)) in ms

(2) SmartReflex voltage must be set before execution of application code

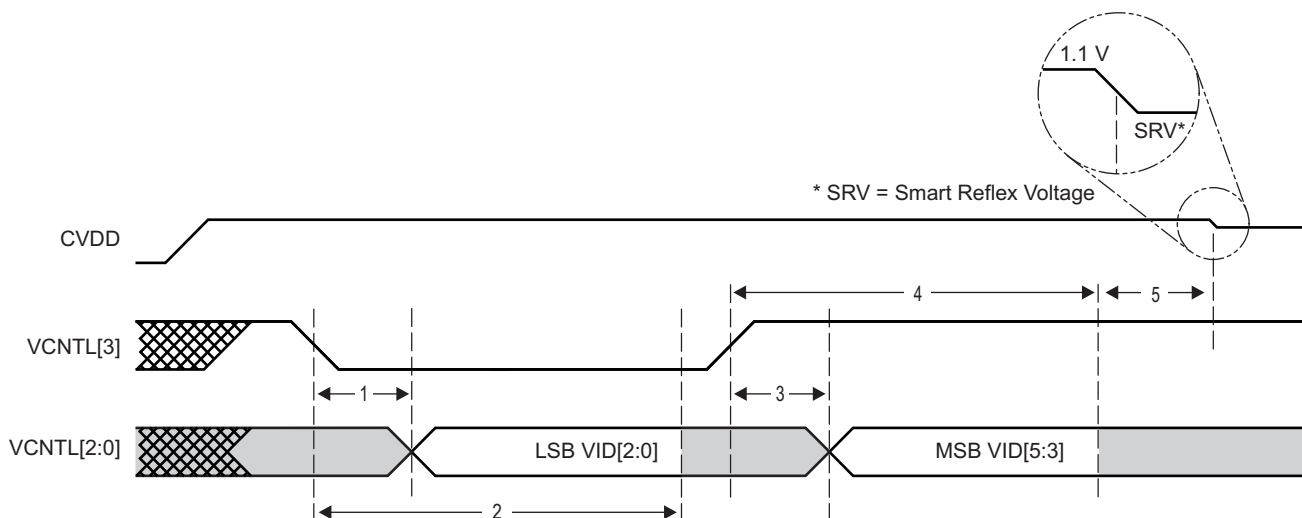


Figure 8-3. SmartReflex 4-Pin VID Interface Timing

8.3 Power Sleep Controller (PSC)

The Power Sleep Controller (PSC) controls overall device power by turning off unused power domains and gating off clocks to individual peripherals and modules. The PSC provides the user with an interface to control several important power and clock operations.

For information on the Power Sleep Controller, see the *Power Sleep Controller (PSC) for KeyStone Devices User's Guide* ([SPRUGV4](#)).

8.3.1 Power Domains

The device has several power domains that can be turned on for operation or off to minimize power dissipation. The global power/sleep controller (GPSC) is used to control the power gating of various power domains.

[Table 8-6](#) shows the C665x power domains.

Table 8-6. Power Domains

| DOMAIN | BLOCK(S) | NOTE | POWER CONNECTION |
|--------|--------------------------------------|------------------------------|---|
| 0 | Most peripheral logic | Cannot be disabled | Always on |
| 1 | Per-core TETB and System TETB | RAMs can be powered down | Software control |
| 2 | Reserved | Reserved | Reserved |
| 3 | PCIe | Logic can be powered down | Software control |
| 4 | SRIO | Logic can be powered down | Software control |
| 5 | HyperLink | Logic can be powered down | Software control |
| 6 | Reserved | Reserved | Reserved |
| 7 | MSMC RAM | MSMC RAM can be powered down | Software control |
| 8 | Reserved | Reserved | Reserved |
| 9 | Reserved | Reserved | Reserved |
| 10 | Reserved | Reserved | Reserved |
| 11 | TCP3d | RAMs can be powered down | Software control |
| 12 | VCP2_B | RAMs can be powered down | Software control |
| 13 | C66x Core 0, L1/L2 RAMs | L2 RAMs can sleep | Software control via C66x CorePac. For details, see the C66x CorePac Reference Guide. |
| 14 | C66x Core 1, L1/L2 RAMs (C6657 only) | L2 RAMs can sleep | Software control via C66x CorePac. For details, see the C66x CorePac Reference Guide. |
| 15 | Reserved | Reserved | Reserved |

8.3.2 Clock Domains

Clock gating to each logic block is managed by the local power/sleep controllers (LPSCs) of each module. For modules with a dedicated clock or multiple clocks, the LPSC communicates with the PLL controller to enable and disable that module's clock(s) at the source. For modules that share a clock with other modules, the LPSC controls the clock gating.

Table 8-7 shows the C665x clock domains.

Table 8-7. Clock Domains

| LPSC NUMBER | MODULE(S) | NOTES |
|-------------|---|-------------------------------|
| 0 | Shared LPSC for all peripherals other than those listed in this table | Always on |
| 1 | SmartReflex | Always on |
| 2 | DDR3 EMIF | Always on |
| 3 | EMAC | Software control |
| 4 | VCP2_A | Software control |
| 5 | Debug Subsystem and Tracers | Software control |
| 6 | Per-core TETB and System TETB | Software control |
| 7 | Reserved | Reserved |
| 8 | Reserved | Reserved |
| 9 | Reserved | Reserved |
| 10 | PCIe | Software control |
| 11 | SRIO | Software control |
| 12 | HyperLink | Software control |
| 13 | Reserved | Reserved |
| 14 | MSMC RAM | Software control |
| 15 | Reserved | Reserved |
| 16 | Reserved | Reserved |
| 17 | Reserved | Reserved |
| 18 | Reserved | Reserved |
| 19 | TCP3d | Software control |
| 20 | VCP2_1 | Software control |
| 21 | Reserved | Reserved |
| 22 | Reserved | Reserved |
| 23 | C66x CorePac 0 and Timer 0 | Software control |
| 24 | C66x CorePac 1 (C6657 only) and Timer 1 | Software control |
| No LPSC | Bootcfg, PSC, and PLL controller | These modules do not use LPSC |

8.3.3 PSC Register Memory Map

Table 8-8 shows the PSC Register memory map.

Table 8-8. PSC Register Memory Map

| OFFSET | REGISTER | DESCRIPTION |
|---------------|----------|---|
| 0x000 | PID | Peripheral Identification Register |
| 0x004 - 0x010 | Reserved | Reserved |
| 0x014 | VCNTLID | Voltage Control Identification Register ⁽¹⁾ |
| 0x018 - 0x11C | Reserved | Reserved |
| 0x120 | PTCMD | Power Domain Transition Command Register |
| 0x124 | Reserved | Reserved |
| 0x128 | PTSTAT | Power Domain Transition Status Register |
| 0x12C - 0x1FC | Reserved | Reserved |
| 0x200 | PDSTAT0 | Power Domain Status Register 0 (AlwaysOn) |
| 0x204 | PDSTAT1 | Power Domain Status Register 1 (Per-core TETB and System TETB) |
| 0x208 | PDSTAT2 | Power Domain Status Register 2 (Reserved) |
| 0x20C | PDSTAT3 | Power Domain Status Register 3 (PCIe) |
| 0x210 | PDSTAT4 | Power Domain Status Register 4 (SRIO) |
| 0x214 | PDSTAT5 | Power Domain Status Register 5 (Hyperlink) |
| 0x218 | PDSTAT6 | Power Domain Status Register 6 (Reserved) |
| 0x21C | PDSTAT7 | Power Domain Status Register 7 (MSMC RAM) |
| 0x220 | PDSTAT8 | Power Domain Status Register 8 (Reserved) |
| 0x224 | PDSTAT9 | Power Domain Status Register 9 (Reserved) |
| 0x228 | PDSTAT10 | Power Domain Status Register 10 (Reserved) |
| 0x22C | PDSTAT11 | Power Domain Status Register 11 (TCP3d) |
| 0x230 | PDSTAT12 | Power Domain Status Register 12 (VCP2_B) |
| 0x234 | PDSTAT13 | Power Domain Status Register 13 (C66x CorePac 0) |
| 0x238 | PDSTAT14 | Power Domain Status Register 14 (C66x CorePac 1) (C6657) or Reserved (C6655) |
| 0x23C | Reserved | Reserved |
| 0x240 - 0x2FC | Reserved | Reserved |
| 0x300 | PDCTL0 | Power Domain Control Register 0 (AlwaysOn) |
| 0x304 | PDCTL1 | Power Domain Control Register 1 (Per-core TETB and System TETB) |
| 0x308 | PDCTL2 | Power Domain Control Register 2 (Reserved) |
| 0x30C | PDCTL3 | Power Domain Control Register 3 (PCIe) |
| 0x310 | PDCTL4 | Power Domain Control Register 4 (SRIO) |
| 0x314 | PDCTL5 | Power Domain Control Register 5 (HyperLink) |
| 0x318 | PDCTL6 | Power Domain Control Register 6 (Reserved) |
| 0x31C | PDCTL7 | Power Domain Control Register 7 (MSMC RAM) |
| 0x320 | PDCTL8 | Power Domain Control Register 8 (Reserved) |
| 0x324 | PDCTL9 | Power Domain Control Register 9 (Reserved) |
| 0x328 | PDCTL10 | Power Domain Control Register 10 (Reserved) |
| 0x32C | PDCTL11 | Power Domain Control Register 11 (TCP3d) |
| 0x330 | PDCTL12 | Power Domain Control Register 12 (VCP2_B) |
| 0x334 | PDCTL13 | Power Domain Control Register 13 (C66x CorePac 0) |
| 0x338 | PDCTL14 | Power Domain Control Register 14 (C66x CorePac 1) (C6657) or Reserved (C6655) |
| 0x33C | Reserved | Reserved |
| 0x340 - 0x7FC | Reserved | Reserved |
| 0x800 | MDSTAT0 | Module Status Register 0 (Never Gated) |

(1) VCNTLID register is available for debug purpose only.

Table 8-8. PSC Register Memory Map (continued)

| OFFSET | REGISTER | DESCRIPTION |
|---------------|----------|--|
| 0x804 | MDSTAT1 | Module Status Register 1 (SmartReflex) |
| 0x808 | MDSTAT2 | Module Status Register 2 (DDR3 EMIF) |
| 0x80C | MDSTAT3 | Module Status Register 3 (EMAC) |
| 0x810 | MDSTAT4 | Module Status Register 4 (VCP2_A) |
| 0x814 | MDSTAT5 | Module Status Register 5 (Debug Subsystem and Tracers) |
| 0x818 | MDSTAT6 | Module Status Register 6 (Per-core TETB and System TETB) |
| 0x81C | MDSTAT7 | Module Status Register 7 (Reserved) |
| 0x820 | MDSTAT8 | Module Status Register 8 (Reserved) |
| 0x824 | MDSTAT9 | Module Status Register 9 (Reserved) |
| 0x828 | MDSTAT10 | Module Status Register 10 (PCIe) |
| 0x82C | MDSTAT11 | Module Status Register 11 (SRIO) |
| 0x830 | MDSTAT12 | Module Status Register 12 (HyperLink) |
| 0x834 | MDSTAT13 | Module Status Register 13 (Reserved) |
| 0x838 | MDSTAT14 | Module Status Register 14 (MSMC RAM) |
| 0x83C | MDSTAT15 | Module Status Register 15 (Reserved) |
| 0x840 | MDSTAT16 | Module Status Register 16 (Reserved) |
| 0x844 | MDSTAT17 | Module Status Register 17 (Reserved) |
| 0x848 | MDSTAT18 | Module Status Register 18 (Reserved) |
| 0x84C | MDSTAT19 | Module Status Register 19 (TCP3d) |
| 0x850 | MDSTAT20 | Module Status Register 20 (VCP2_B) |
| 0x854 | MDSTAT21 | Module Status Register 11 (Reserved) |
| 0x858 | MDSTAT22 | Module Status Register 22(Reserved) |
| 0x85C | MDSTAT23 | Module Status Register 23(C66x CorePac 0 and Timer 0) |
| 0x860 | MDSTAT24 | Module Status Register 24(C66x CorePac 1 [C6657 only] and Timer 1) |
| 0x864 - 0x9FC | Reserved | Reserved |
| 0xA00 | MDCTL0 | Module Control Register 0 (Never Gated) |
| 0xA04 | MDCTL1 | Module Control Register 1 (SmartReflex) |
| 0xA08 | MDCTL2 | Module Control Register 2 (DDR3 EMIF) |
| 0xA0C | MDCTL3 | Module Control Register 3 (EMAC) |
| 0xA10 | MDCTL4 | Module Control Register 4 (VCP2_A) |
| 0xA14 | MDCTL5 | Module Control Register 5 (Debug Subsystem and Tracers) |
| 0xA18 | MDCTL6 | Module Control Register 6 (Per-core TETB and System TETB) |
| 0xA1C | MDCTL7 | Module Control Register 7 (Reserved) |
| 0xA20 | MDCTL8 | Module Control Register 8 (Reserved) |
| 0xA24 | MDCTL9 | Module Control Register 9 (Reserved) |
| 0xA28 | MDCTL10 | Module Control Register 10 (PCIe) |
| 0xA2C | MDCTL11 | Module Control Register 11 (SRIO) |
| 0xA30 | MDCTL12 | Module Control Register 12 (HyperLink) |
| 0xA34 | MDCTL13 | Module Control Register 13 (Reserved) |
| 0xA38 | MDCTL14 | Module Control Register 14 (MSMC RAM) |
| 0xA3C | MDCTL15 | Module Control Register 15 (Reserved) |
| 0xA40 | MDCTL16 | Module Control Register 16 (Reserved) |
| 0xA44 | MDCTL17 | Module Control Register 17 (Reserved) |
| 0xA48 | MDCTL18 | Module Control Register 18 (Reserved) |
| 0xA4C | MDCTL19 | Module Control Register 19 (TCP3d) |
| 0xA50 | MDCTL20 | Module Control Register 20 (VCP2_1) |
| 0xA54 | MDCTL21 | Module Control Register 21(Reserved) |

Table 8-8. PSC Register Memory Map (continued)

| OFFSET | REGISTER | DESCRIPTION |
|---------------|----------|---|
| 0xA58 | MDCTL22 | Module Control Register 22(Reserved) |
| 0xA5C | MDCTL23 | Module Control Register 23(C66x CorePac 0 and Timer 0) |
| 0xA60 | MDCTL24 | Module Control Register 24(C66x CorePac 1 [C6657 only] and Timer 1) |
| 0xA5C - 0xFFC | Reserved | Reserved |

8.4 Reset Controller

The reset controller detects the different type of resets supported on the C665x device and manages the distribution of those resets throughout the device.

The device has several types of resets:

- Power-on reset
- Hard reset
- Soft reset
- CPU local reset

[Table 8-9](#) explains further the types of reset, the reset initiator, and the effects of each reset on the device. For more information on the effects of each reset on the PLL controllers and their clocks, see [Section 8.4.7](#).

Table 8-9. Reset Types

| RESET TYPE | INITIATOR | EFFECT ON DEVICE WHEN RESET OCCURS | RESETSTAT PIN STATUS |
|--------------------------|--|---|---|
| POR (Power On Reset) | $\overline{\text{POR}}$ pin active low $\overline{\text{RESETFULL}}$ pin active low | Total reset of the chip. Everything on the device is reset to its default state in response to this. Activates the POR signal on chip, which is used to reset test/emu logic. Boot configurations are latched. ROM boot process is initiated. | Toggles $\overline{\text{RESETSTAT}}$ pin |
| Hard reset | $\overline{\text{RESET}}$ pin active low Emulation PLLCTL register (RSCTRL) Watchdog timers | Resets everything except for test/emu logic and reset isolation modules. Emulator and reset Isolation modules stay alive during this reset. This reset is also different from POR in that the PLLCTL assumes power and clocks are stable when device reset is asserted. Boot configurations are not latched. ROM boot process is initiated. | Toggles $\overline{\text{RESETSTAT}}$ pin |
| Soft reset | $\overline{\text{RESET}}$ pin active low PLLCTL register (RSCTRL) Watchdog timers | Software can program these initiators to be hard or soft. Hard reset is the default, but can be programmed to be soft reset. Soft reset will behave like hard reset except that EMIF16 MMRs, DDR3 EMIF MMRs, sticky bits in PCIe MMRs, and external memory contents are retained. Boot configurations are not latched. ROM boot process is initiated. | Toggles $\overline{\text{RESETSTAT}}$ pin |
| C66x CorePac local reset | Software (through LPSC MMR) Watchdog timers $\overline{\text{LRESET}}$ pin | MMR bit in LPSC controls C66x CorePac local reset. Used by watchdog timers (in the event of a timeout) to reset C66x CorePac. Can also be initiated by $\overline{\text{LRESET}}$ device pin. C66x CorePac memory system and slave DMA port are still alive when C66x CorePac is in local reset. Provides a local reset of the C66x CorePac, without destroying clock alignment or memory contents. Does not initiate ROM boot process. | Does not toggle $\overline{\text{RESETSTAT}}$ pin |

8.4.1 Power-on Reset

Power-on reset is used to reset the entire device, including the test and emulation logic.

Power-on reset is initiated by the following:

1. $\overline{\text{POR}}$ pin
2. $\overline{\text{RESETFULL}}$ pin

During power-up, the $\overline{\text{POR}}$ pin must be asserted (driven low) until the power supplies have reached their normal operating conditions. A $\overline{\text{RESETFULL}}$ pin is also provided to allow the on-board host to reset the entire device including the reset isolated logic. The assumption is that the device is already powered up and hence, unlike the $\overline{\text{POR}}$ pin, the $\overline{\text{RESETFULL}}$ pin will be driven by the on-board host control instead of the power-good circuitry. For power-on reset, the Main PLL Controller comes up in bypass mode and the PLL is not enabled. Other resets do not affect the state of the PLL or the dividers in the PLL controller.

The following sequence must be followed during a power-on reset:

1. Wait for all power supplies to reach normal operating conditions while keeping the $\overline{\text{POR}}$ pin asserted (driven low). While $\overline{\text{POR}}$ is asserted, all pins except $\overline{\text{RESETSTAT}}$ will be set to high-impedance. After the $\overline{\text{POR}}$ pin is de-asserted (driven high), all Z group pins, low group pins, and high group pins are set to their reset state and will remain at their reset state until otherwise configured by their respective peripheral. All peripherals that are power managed, are disabled after a power-on reset and must be enabled through the Device State Control Registers (for more details, see [Table 4-2](#)).
2. Clocks are reset, and they are propagated throughout the device to reset any logic that was using reset synchronously. All logic is now reset and $\overline{\text{RESETSTAT}}$ will be driven low indicating that the device is in reset.
3. $\overline{\text{POR}}$ must be held active until all supplies on the board are stable then for at least an additional time for the chip-level PLLs to lock.
4. The $\overline{\text{POR}}$ pin can now be de-asserted. Reset-sampled pin values are latched at this point. The chip level PLLs are taken out of reset and begin their locking sequence, and all power-on device initialization also begins.
5. After device initialization is complete, the $\overline{\text{RESETSTAT}}$ pin is de-asserted (driven high). By this time, the DDR3 PLL has already completed its locking sequence and is outputting a valid clock. The system clocks of both PLL controllers are allowed to finish their current cycles and then paused for 10 cycles of their respective system reference clocks. After the pause, the system clocks are restarted at their default divide by settings.
6. The device is now out of reset and device execution begins as dictated by the selected boot mode.

NOTE

To most of the device, reset is de-asserted only when the $\overline{\text{POR}}$ and $\overline{\text{RESET}}$ pins are both de-asserted (driven high). Therefore, in the sequence described above, if the $\overline{\text{RESET}}$ pin is held low past the low period of the $\overline{\text{POR}}$ pin, most of the device will remain in reset. The $\overline{\text{RESET}}$ pin should not be tied together with the $\overline{\text{POR}}$ pin.

8.4.2 Hard Reset

A hard reset will reset everything on the device except the PLLs, test, emulation logic, and reset isolation modules. $\overline{\text{POR}}$ should also remain de-asserted during this time.

Hard reset is initiated by the following:

- $\overline{\text{RESET}}$ pin
- RSCTRL register in PLLCTL
- Watchdog timer
- Emulation

All the above initiators, by default, are configured to act as a hard reset. Except emulation, all the other three initiators can be configured as soft resets in the RSCFG register in PLLCTL.

The following sequence must be followed during a hard reset:

1. The $\overline{\text{RESET}}$ pin is pulled active low for a minimum of 24 input clock cycles. During this time, the $\overline{\text{RESET}}$ signal is able to propagate to all modules (except those specifically mentioned above). All I/O are Hi-Z for modules affected by $\overline{\text{RESET}}$, to prevent off-chip contention during the warm reset.
2. Once all logic is reset, $\overline{\text{RESETSTAT}}$ is driven active to denote that the device is in reset.
3. The $\overline{\text{RESET}}$ pin can now be released. A minimal device initialization begins to occur. Note that configuration pins are not re-latched and clocking is unaffected within the device.
4. After device initialization is complete, the $\overline{\text{RESETSTAT}}$ pin is de-asserted (driven high).

NOTE

The $\overline{\text{POR}}$ pin should be held inactive (high) throughout the warm reset sequence. Otherwise, if $\overline{\text{POR}}$ is activated (brought low), the minimum $\overline{\text{POR}}$ pulse width must be met. The $\overline{\text{RESET}}$ pin should not be tied together with the $\overline{\text{POR}}$ pin.

8.4.3 Soft Reset

A soft reset will behave like a hard reset except that the PCIe MMR sticky bits and DDR3 EMIF MMRs contents are retained. $\overline{\text{POR}}$ should also remain de-asserted during this time.

Soft reset is initiated by the following:

- $\overline{\text{RESET}}$ pin
- RCTRL register in PLLCTL
- Watchdog timer

All the above initiators by default are configured to act as hard reset. Except emulation, all the other three initiators can be configured as soft resets in the RSCFG register in PLLCTL.

In the case of a soft reset, the clock logic or the power control logic of the peripherals are not affected, and, therefore, the enabled/disabled state of the peripherals is not affected. On a soft reset, the DDR3 memory controller registers are not reset. In addition, the DDR3 SDRAM memory content is retained if the user places the DDR3 SDRAM in self-refresh mode before invoking the soft reset.

During a soft reset, the following happens:

1. The $\overline{\text{RESETSTAT}}$ pin goes low to indicate an internal reset is being generated. The reset is allowed to propagate through the system. Internal system clocks are not affected. PLLs also remain locked.
2. After device initialization is complete, the $\overline{\text{RESETSTAT}}$ pin is deasserted (driven high). In addition, the PLL controllers pause their system clocks for about 8 cycles.
 - At this point:
 - The state of the peripherals before the soft reset is not changed.
 - The I/O pins are controlled as dictated by the DEVSTAT register.
 - The DDR3 MMRs and PCIe MMR sticky bits retain their previous values. Only the DDR3 Memory Controller and PCIe state machines are reset by the soft reset.
 - The PLL controllers are operating in the mode prior to soft reset. System clocks are unaffected.

The boot sequence is started after the system clocks are restarted. Since the configuration pins are not latched with a system reset, the previous values, as shown in the DEVSTAT register, are used to select the boot mode.

8.4.4 Local Reset

The local reset can be used to reset a particular CorePac without resetting any other chip components.

Local reset is initiated by the following (for more details see the *Phase Locked Loop (PLL) for KeyStone Devices User's Guide* ([SPRUGV2](#)):

- $\overline{\text{LRESET}}$ pin
- Watchdog timer should cause one of the below based on the setting of the CORESEL[2:0] and RSTCFG register in the PLL controller. See [Section 8.5.2.8](#) and [Section 8.8.2](#):
 - Local Reset
 - NMI
 - NMI followed by a time delay and then a local reset for the CorePac selected
 - Hard Reset by requesting reset via PLLCTL
- LPSC MMRs (memory-mapped registers)

8.4.5 Reset Priority

If any of the above reset sources occur simultaneously, the PLLCTL processes only the highest priority reset request. The reset request priorities are as follows (high to low):

- Power-on reset
- Hard/soft reset

8.4.6 Reset Controller Register

The reset controller register is part of the PLLCTL MMRs. All C665x device-specific MMRs are covered in [Section 8.5.3](#). For more details on these registers and how to program them, see the *Phase Locked Loop (PLL) for KeyStone Devices User's Guide (SPRUGV2)*.

8.4.7 Reset Electrical Data / Timing

Table 8-10. Reset Timing Requirements⁽¹⁾

(see [Figure 8-4](#) and [Figure 8-5](#))

| NO. | | | MIN | MAX | UNIT |
|----------------------------|---------------|---|------|-----|------|
| RESETFULL Pin Reset | | | | | |
| 1 | tw(RESETFULL) | Pulse width - Pulse width $\overline{\text{RESETFULL}}$ low | 500C | | ns |
| Soft/Hard-Reset | | | | | |
| 2 | tw(RESET) | Pulse width - Pulse width $\overline{\text{RESET}}$ low | 500C | | ns |

(1) C = 1 / CORECLK(N/P) frequency in ns.

Table 8-11. Reset Switching Characteristics Over Recommended Operating Conditions⁽¹⁾

(see [Figure 8-4](#) and [Figure 8-5](#))

| NO. | PARAMETER | MIN | MAX | Unit |
|----------------------------|---|-----|--------|------|
| RESETFULL Pin Reset | | | | |
| 3 | td(RESETFULLH-RESETSTATH) Delay time - RESETSTAT high after $\overline{\text{RESETFULL}}$ high | | 50000C | ns |
| Soft/Hard Reset | | | | |
| 4 | td(RESETH-RESETSTATH) Delay time - RESETSTAT high after $\overline{\text{RESET}}$ high | | 50000C | ns |

(1) C = 1 / CORECLK(N/P) frequency in ns.

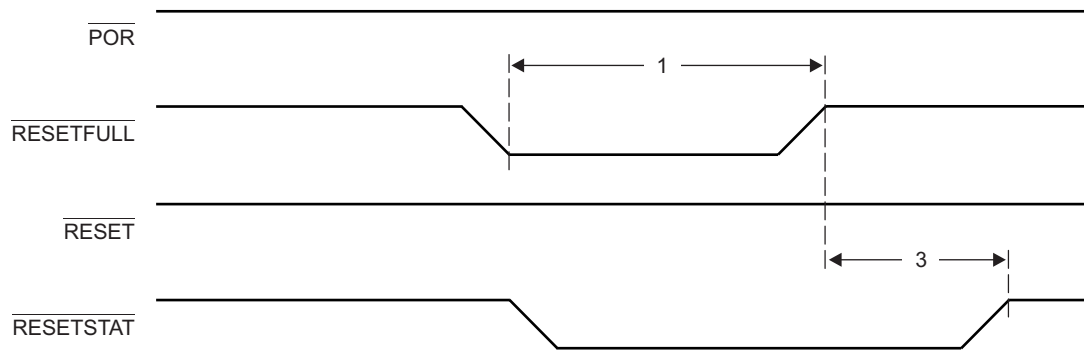


Figure 8-4. $\overline{\text{RESETFULL}}$ Reset Timing

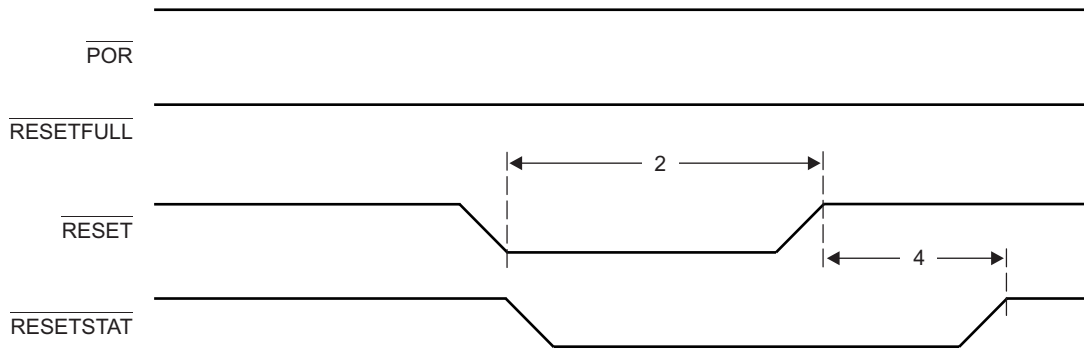


Figure 8-5. Soft/Hard-Reset Timing

Table 8-12. Boot Configuration Timing Requirements⁽¹⁾

(See Figure 8-6)

| NO. | | | MIN | MAX | UNIT |
|-----|--|---|-----|-----|------|
| 1 | $t_{\text{su}}(\text{GPIO}_{\text{On}}-\overline{\text{RESETFULL}})$ | Setup time - GPIO valid before $\overline{\text{RESETFULL}}$ asserted | 12C | | ns |
| 2 | $t_{\text{h}}(\overline{\text{RESETFULL}}-\text{GPIO}_{\text{On}})$ | Hold time - GPIO valid after $\overline{\text{RESETFULL}}$ asserted | 12C | | ns |

(1) C = 1/SYSCLK1 frequency in ns.

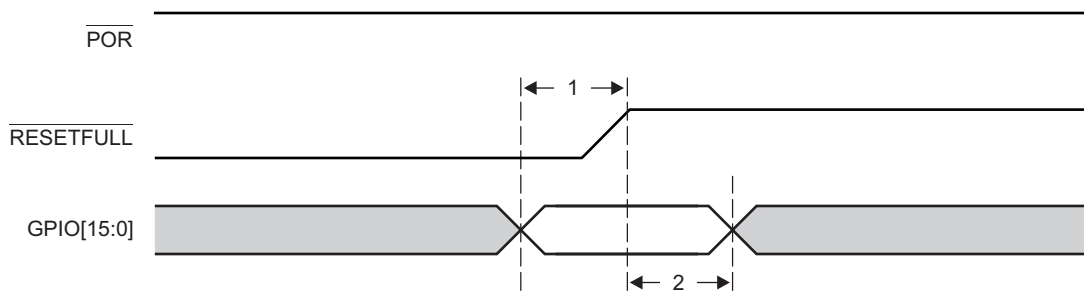


Figure 8-6. Boot Configuration Timing

8.5 Main PLL and PLL Controller

This section provides a description of the Main PLL and the PLL controller. For details on the operation of the PLL controller module, see the *Phase Locked Loop (PLL) for KeyStone Devices User's Guide (SPRUGV2)*.

The Main PLL is controlled by the standard PLL controller. The PLL controller manages the clock ratios, alignment, and gating for the system clocks to the device. [Figure 8-7](#) shows a block diagram of the main PLL and the PLL controller.

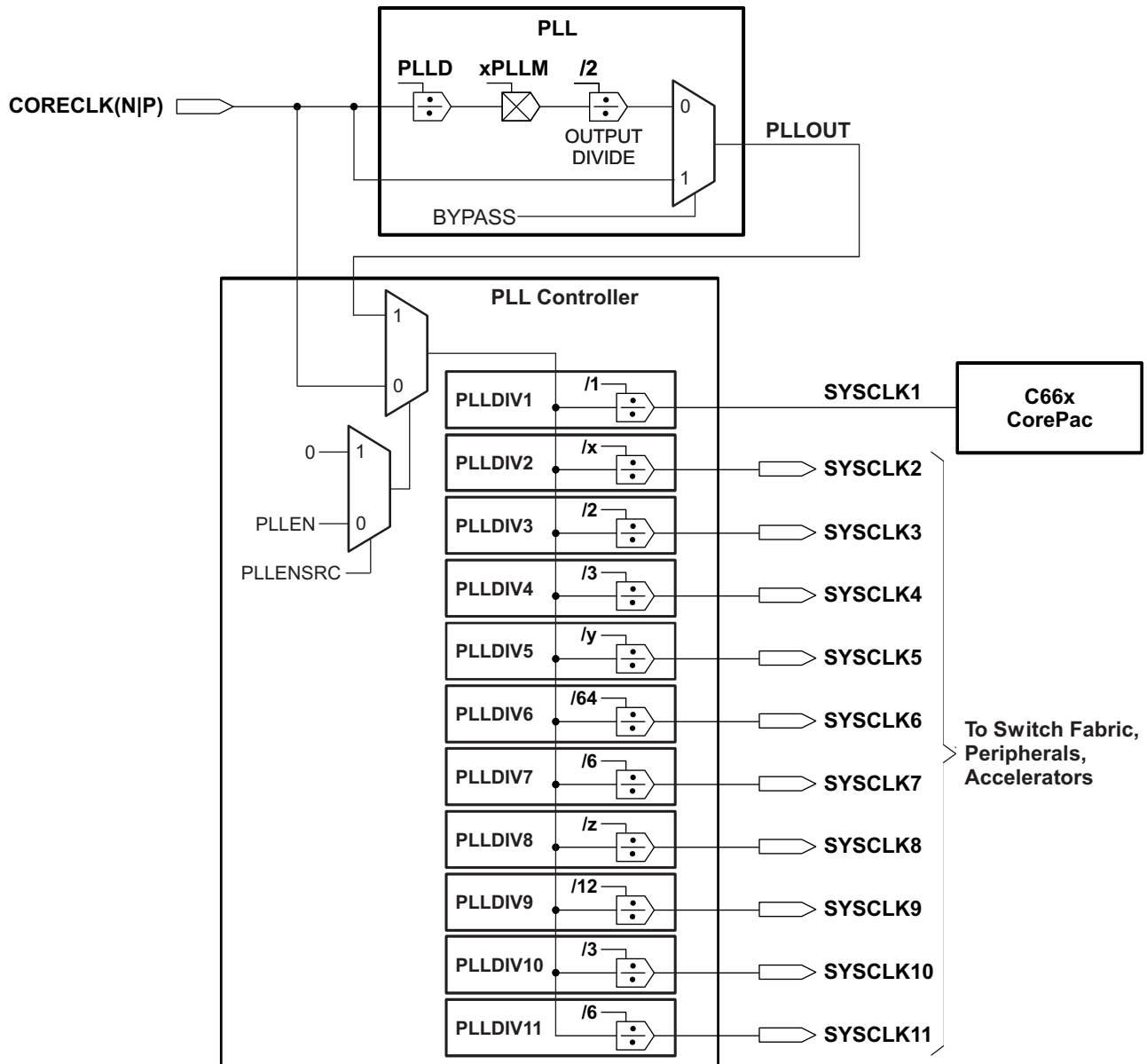


Figure 8-7. Main PLL and PLL Controller

NOTE

PLLM[5:0] bits of the multiplier are controlled by the PLLM register inside the PLL controller and PLLM[12:6] bits are controlled by the chip level MAINPLLCTL0 register. The complete 13-bit value is latched when the GO operation is initiated in the PLL controller. Only PLLDIV2, PLLDIV5, and PLLDIV8 are programmable on the C665x device. See the *Phase Locked Loop (PLL) for KeyStone Devices User's Guide* ([SPRUGV2](#)) for more details on how to program the PLL controller.

The multiplication and division ratios within the PLL and the post-division for each of the chip-level clocks are determined by a combination of this PLL and the PLL Controller. The PLL controller also controls reset propagation through the chip, clock alignment, and test points. The PLL controller monitors the PLL status and provides an output signal indicating when the PLL is locked.

Main PLL power is supplied externally via the Main PLL power-supply pin (AVDDA1). An external EMI filter circuit must be added to all PLL supplies. See the *Hardware Design Guide for KeyStone Devices* ([SPRABI2](#)) for detailed recommendations. For the best performance, TI recommends that all the PLL external components be on a single side of the board without jumpers, switches, or components other than those shown. For reduced PLL jitter, maximize the spacing between switching signal traces and the PLL external components (C1, C2, and the EMI Filter).

The minimum SYSCLK rise and fall times should also be observed. For the input clock timing requirements, see [Section 8.5.5](#).

CAUTION

The PLL controller module as described in the *Phase Locked Loop (PLL) for KeyStone Devices User's Guide* [SPRUGV2](#) includes a superset of features, some of which are not supported on the C665x device. The following sections describe the registers that are supported; it should be assumed that any registers not included in these sections is not supported by the device. Furthermore, only the bits within the registers described here are supported. Avoid writing to any reserved memory location or changing the value of reserved bits.

8.5.1 Main PLL Controller Device-Specific Information

8.5.1.1 Internal Clocks and Maximum Operating Frequencies

The Main PLL, used to drive the CorePacs, the switch fabric, and a majority of the peripheral clocks (all but the DDR3) requires a PLL controller to manage the various clock divisions, gating, and synchronization. The Main PLL's PLL controller has several SYSCLK outputs that are listed below, along with the clock description. Each SYSCLK has a corresponding divider that divides down the output clock of the PLL. Note that dividers are not programmable unless explicitly mentioned in the description below.

- **SYCLK1:** Full-rate clock for the CorePacs.
- **SYCLK2:** 1/x-rate clock for CorePac emulation. The default rate for this is 1/3. It is programmable from /1 to /32, where this clock does not violate the max of 350 MHz. The SYCLK2 can be turned off by software.
- **SYCLK3:** 1/2-rate clock used to clock the MSMC, HyperLink, and DDR EMIF.
- **SYCLK4:** 1/3-rate clock for the switch fabrics and fast peripherals. The Debug_SS and ETBs use this as well.
- **SYCLK5:** 1/y-rate clock for the system trace module only. The default rate for this is 1/5. It is configurable and the max configurable clock is 210 MHz and min configurable clock is 32 MHz. The SYCLK5 can be turned off by software.
- **SYCLK6:** 1/64-rate clock. 1/64 rate clock (emif_ptv) used to clock the PVT-compensated buffers for DDR3 EMIF.
- **SYCLK7:** 1/6-rate clock for slow peripherals (GPIO, UART, Timer, I²C, SPI, EMIF16, McBSP, etc.) and sources the SYCLKOUT output pin.
- **SYCLK8:** 1/z-rate clock. This clock is used as slow_sysclk in the system. Default is 1/64. It is programmable from /24 to /80.
- **SYCLK9:** 1/12-rate clock for SmartReflex.
- **SYCLK10:** 1/3-rate clock for SRIO only.
- **SYCLK11:** 1/6-rate clock for PSC only.

Only SYCLK2, SYCLK5, and SYCLK8 are programmable on the C665x device.

NOTE

In case any of the other programmable SYSCLKs are set slower than 1/64 rate, then SYCLK8 (SLOW_SYSCLK) needs to be programmed to either match, or be slower than, the slowest SYSCLK in the system.

8.5.1.2 Main PLL Controller Operating Modes

The Main PLL controller has two modes of operation: bypass mode and PLL mode. The mode of operation is determined by BYPASS bit of the PLL Secondary Control Register (SECCTL). In PLL mode, SYCLK1 is generated from the PLL output using the values set in PLLM and PLLD bit fields in the MAINPLLCTL0 Register. In bypass mode, PLL input is fed directly out as SYCLK1.

All hosts must hold off accesses to the DSP while the frequency of its internal clocks is changing. A mechanism must be in place such that the DSP notifies the host when the PLL configuration has completed.

8.5.1.3 Main PLL Stabilization, Lock, and Reset Times

The PLL stabilization time is the amount of time that must be allotted for the internal PLL regulators to become stable after device powerup. The PLL should not be operated until this stabilization time has elapsed.

The PLL reset time is the amount of wait time needed when resetting the PLL (writing PLLRST = 1), in order for the PLL to properly reset, before bringing the PLL out of reset (writing PLLRST = 0). For the Main PLL reset time value, see [Table 8-13](#).

The PLL lock time is the amount of time needed from when the PLL is taken out of reset (PLLRST = 1 with PLEN = 0) to when the PLL controller can be switched to PLL mode (PLEN = 1). The Main PLL lock time is given in [Table 8-13](#).

Table 8-13. Main PLL Stabilization, Lock, and Reset Times

| | MIN | TYP | MAX | UNIT |
|------------------------|------|-----|---|------|
| PLL stabilization time | 100 | | | μs |
| PLL lock time | | | $500 \times (\text{PLLD}^{(1)+1}) \times C^{(2)}$ | |
| PLL reset time | 1000 | | | ns |

(1) PLLD is the value in PLLD bit fields of MAINPLLCTL0 register

(2) C = SYSCLK1(N|P) cycle time in ns.

8.5.2 PLL Controller Memory Map

The memory map of the PLL controller is shown in [Table 8-14](#). C665x-specific PLL Controller register definitions can be found in the sections following [Table 8-14](#). For other registers in the table, see the *Phase Locked Loop (PLL) for KeyStone Devices User's Guide* ([SPRUGV2](#)).

CAUTION

Note that only registers documented here are accessible on the C665x. Other addresses in the PLL controller memory map including the reserved registers should not be modified. Furthermore, only the bits within the registers described here are supported. Avoid writing to any reserved memory location or changing the value of reserved bits. It is recommended to use read-modify-write sequence to make any changes to the valid bits in the register.

Table 8-14. PLL Controller Registers (Including Reset Controller)

| HEX ADDRESS RANGE | FIELD | REGISTER NAME |
|-----------------------|--------------------|--|
| 0231 0000 - 0231 00E3 | - | Reserved |
| 0231 00E4 | RSTYPE | Reset Type Status Register (Reset Controller) |
| 0231 00E8 | RSTCTRL | Software Reset Control Register (Reset Controller) |
| 0231 00EC | RSTCFG | Reset Configuration Register (Reset Controller) |
| 0231 00F0 | RSISO | Reset Isolation Register (Reset Controller) |
| 0231 00F0 - 0231 00FF | - | Reserved |
| 0231 0100 | PLLCTL | PLL Control Register |
| 0231 0104 | - | Reserved |
| 0231 0108 | SECCTL | PLL Secondary Control Register |
| 0231 010C | - | Reserved |
| 0231 0110 | PLLM | PLL Multiplier Control Register |
| 0231 0114 | - | Reserved |
| 0231 0118 | PLLDIV1 | Reserved |
| 0231 011C | PLLDIV2 | PLL Controller Divider 2 Register |
| 0231 0120 | PLLDIV3 | Reserved |
| 0231 0124 | - | Reserved |
| 0231 0128 | - | Reserved |
| 0231 012C - 0231 0134 | - | Reserved |
| 0231 0138 | PLLCMD | PLL Controller Command Register |
| 0231 013C | PLLSTAT | PLL Controller Status Register |
| 0231 0140 | ALNCTL | PLL Controller Clock Align Control Register |
| 0231 0144 | DCHANGE | PLLDIV Ratio Change Status Register |
| 0231 0148 | CKEN | Reserved |
| 0231 014C | CKSTAT | Reserved |
| 0231 0150 | SYSTAT | SYSCLK Status Register |
| 0231 0154 - 0231 015C | - | Reserved |
| 0231 0160 | PLLDIV4 | Reserved |
| 0231 0164 | PLLDIV5 | PLL Controller Divider 5 Register |
| 0231 0168 | PLLDIV6 | Reserved |
| 0231 016C | PLLDIV7 | Reserved |
| 0231 0170 | PLLDIV8 | PLL Controller Divider 8 Register |
| 0231 0174 - 0231 0193 | PLLDIV9 - PLLDIV16 | Reserved |
| 0231 0194 - 0231 01FF | - | Reserved |

8.5.2.1 PLL Secondary Control Register (SECCTL)

The PLL Secondary Control Register contains extra fields to control the Main PLL and is shown in [Figure 8-8](#) and described in [Table 8-15](#).

Figure 8-8. PLL Secondary Control Register (SECCTL)

| | | | | | | |
|-------------|----|--------|---------------|----|----------------------------|---|
| 31 | 24 | 23 | 22 | 19 | 18 | 0 |
| Reserved | | BYPASS | OUTPUT_DIVIDE | | Reserved | |
| R-0000 0000 | | RW-0 | RW-0001 | | RW-001 0000 0000 0000 0000 | |

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

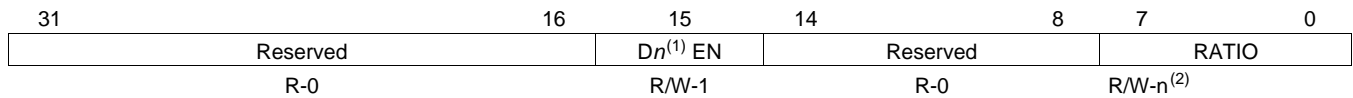
Table 8-15. PLL Secondary Control Register (SECCTL) Field Descriptions

| Bit | Field | Description |
|-------|---------------|--|
| 31-24 | Reserved | Reserved |
| 23 | BYPASS | Main PLL Bypass Enable <ul style="list-style-type: none"> • 0 = Main PLL Bypass disabled. • 1 = Main PLL Bypass enabled. |
| 22-19 | OUTPUT_DIVIDE | Output Divider ratio bits. <ul style="list-style-type: none"> • 0h = ÷1. Divide frequency by 1. • 1h = ÷2. Divide frequency by 2. • 2h - Fh = Reserved. |
| 18-0 | Reserved | Reserved |

8.5.2.2 PLL Controller Divider Register (PLLDIV2, PLLDIV5, PLLDIV8)

The PLL Controller Divider Registers (PLLDIV2, PLLDIV5, and PLLDIV8) are shown in [Figure 8-9](#) and described in [Table 8-16](#). The default values of the RATIO field on a reset for PLLDIV2, PLLDIV5, and PLLDIV8 are different and mentioned in the footnote of [Figure 8-9](#).

Figure 8-9. PLL Controller Divider Register (PLLDIVn)



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

(1) D2EN for PLLDIV2; D5EN for PLLDIV5; D8EN for PLLDIV8

(2) n=02h for PLLDIV2; n=04h for PLLDIV5; n=3Fh for PLLDIV8

Table 8-16. PLL Controller Divider Register (PLLDIVn) Field Descriptions

| Bit | Field | Description |
|-------|----------|--|
| 31-16 | Reserved | Reserved. |
| 15 | DnEN | Divider Dn enable bit. (see footnote of Figure 8-9) <ul style="list-style-type: none"> • 0 = Divider n is disabled. • 1 = No clock output. Divider n is enabled. |
| 14-8 | Reserved | Reserved. The reserved bit location is always read as 0. A value written to this field has no effect. |
| 7-0 | RATIO | Divider ratio bits. (see footnote of Figure 8-9) <ul style="list-style-type: none"> • 0h = ÷1. Divide frequency by 1. • 1h = ÷2. Divide frequency by 2. • 2h = ÷3. Divide frequency by 3. • 3h = ÷4. Divide frequency by 4. • 4h - 4Fh = ÷5 to ÷80. Divide frequency by 5 to divide frequency by 80. |

8.5.2.3 PLL Controller Clock Align Control Register (ALNCTL)

The PLL controller clock align control register (ALNCTL) is shown in [Figure 8-10](#) and described in [Table 8-17](#).

Figure 8-10. PLL Controller Clock Align Control Register (ALNCTL)

| | | | | | | | | | |
|----------|---|-------|----------|---|-------|----------|---|-------|----------|
| 31 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | ALN8 | Reserved | | ALN5 | Reserved | | ALN2 | Reserved |
| R-0 | | R/W-1 | R-0 | | R/W-1 | R-0 | | R/W-1 | R-0 |

Legend: R/W = Read/Write; R = Read only; -n = value after reset, for reset value

Table 8-17. PLL Controller Clock Align Control Register (ALNCTL) Field Descriptions

| Bit | Field | Description |
|------|----------|---|
| 31-8 | Reserved | Reserved. The reserved bit location is always read as 0. A value written to this field has no effect. |
| 7 | ALN8 | <p>SYSClK_n alignment. Do not change the default values of these fields.</p> <ul style="list-style-type: none"> 0 = Do not align SYSClK_n to other SYSClKs during GO operation. If SYS_n in DCHANGE is set, SYSClK_n switches to the new ratio immediately after the GOSET bit in PLLCMD is set. 1 = Align SYSClK_n to other SYSClKs selected in ALNCTL when the GOSET bit in PLLCMD is set and SYS_n in DCHANGE is 1. The SYSClK_n rate is set to the ratio programmed in the RATIO bit in PLLDIV_n. |
| 6-5 | Reserved | Reserved. The reserved bit location is always read as 0. A value written to this field has no effect. |
| 4 | ALN5 | <p>SYSClK_n alignment. Do not change the default values of these fields.</p> <ul style="list-style-type: none"> 0 = Do not align SYSClK_n to other SYSClKs during GO operation. If SYS_n in DCHANGE is set, SYSClK_n switches to the new ratio immediately after the GOSET bit in PLLCMD is set. 1 = Align SYSClK_n to other SYSClKs selected in ALNCTL when the GOSET bit in PLLCMD is set and SYS_n in DCHANGE is 1. The SYSClK_n rate is set to the ratio programmed in the RATIO bit in PLLDIV_n. |
| 3-2 | Reserved | Reserved. The reserved bit location is always read as 0. A value written to this field has no effect. |
| 1 | ALN2 | <p>SYSClK_n alignment. Do not change the default values of these fields.</p> <ul style="list-style-type: none"> 0 = Do not align SYSClK_n to other SYSClKs during GO operation. If SYS_n in DCHANGE is set, SYSClK_n switches to the new ratio immediately after the GOSET bit in PLLCMD is set. 1 = Align SYSClK_n to other SYSClKs selected in ALNCTL when the GOSET bit in PLLCMD is set and SYS_n in DCHANGE is 1. The SYSClK_n rate is set to the ratio programmed in the RATIO bit in PLLDIV_n. |
| 0 | Reserved | Reserved. The reserved bit location is always read as 0. A value written to this field has no effect. |

8.5.2.4 PLLDIV Divider Ratio Change Status Register (DCHANGE)

When a different ratio is written to the PLLDIV n registers, the PLLCTL flags the change in the DCHANGE Status Register. During the GO operation, the PLL controller will change only the divide ratio of the SYSCLKs with the bit set in DCHANGE. Note that the ALNCTL Register determines if that clock also needs to be aligned to other clocks. The PLLDIV divider ratio change status register is shown in [Figure 8-11](#) and described in [Table 8-18](#).

Figure 8-11. PLLDIV Divider Ratio Change Status Register (DCHANGE)

| | | | | | | | | | |
|----------|---|-------|----------|---|-------|----------|---|-------|----------|
| 31 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | SYS8 | Reserved | | SYS5 | Reserved | | SYS2 | Reserved |
| R-0 | | R/W-0 | R-0 | | R/W-0 | R-0 | | R/W-0 | R-0 |

Legend: R/W = Read/Write; R = Read only; - n = value after reset, for reset value

Table 8-18. PLLDIV Divider Ratio Change Status Register (DCHANGE) Field Descriptions

| Bit | Field | Description |
|------|----------|---|
| 31-8 | Reserved | Reserved. The reserved bit location is always read as 0. A value written to this field has no effect. |
| 7 | SYS8 | Identifies when the SYSCLK n divide ratio has been modified. <ul style="list-style-type: none"> 0 = SYSCLKn ratio has not been modified. When GOSET is set, SYSCLKn will not be affected. 1 = SYSCLKn ratio has been modified. When GOSET is set, SYSCLKn will change to the new ratio. |
| 6-5 | Reserved | Reserved. The reserved bit location is always read as 0. A value written to this field has no effect. |
| 4 | SYS5 | Identifies when the SYSCLK n divide ratio has been modified. <ul style="list-style-type: none"> 0 = SYSCLKn ratio has not been modified. When GOSET is set, SYSCLKn will not be affected. 1 = SYSCLKn ratio has been modified. When GOSET is set, SYSCLKn will change to the new ratio. |
| 3-2 | Reserved | Reserved. The reserved bit location is always read as 0. A value written to this field has no effect. |
| 1 | SYS2 | Identifies when the SYSCLK n divide ratio has been modified. <ul style="list-style-type: none"> 0 = SYSCLKn ratio has not been modified. When GOSET is set, SYSCLKn will not be affected. 1 = SYSCLKn ratio has been modified. When GOSET is set, SYSCLKn will change to the new ratio. |
| 0 | Reserved | Reserved. The reserved bit location is always read as 0. A value written to this field has no effect. |

8.5.2.5 SYSCLK Status Register (SYSTAT)

The SYSCLK Status Register (SYSTAT) shows the status of SYSCLK[11:1]. SYSTAT is shown in [Figure 8-12](#) and described in [Table 8-19](#).

Figure 8-12. SYSCLK Status Register (SYSTAT)

| | | | | | | | | | | | | |
|----------|-------------|-------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|-----|
| 31 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | SYS11 ON | SYS10 ON | SYS9ON | SYS8ON | SYS7ON | SYS6ON | SYS5ON | SYS4ON | SYS3ON | SYS2ON | SYS1ON | |
| R-n | R-1 | R-1 | R-1 | R-1 | R-1 | R-1 | R-1 | R-1 | R-1 | R-1 | R-1 | R-1 |

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

Table 8-19. SYSCLK Status Register (SYSTAT) Field Descriptions

| Bit | Field | Description |
|-------|--------------------------|--|
| 31-11 | Reserved | Reserved. The reserved bit location is always read as 0. A value written to this field has no effect. |
| 10-0 | SYS[N ⁽¹⁾]ON | SYSCLK[N] on status. <ul style="list-style-type: none"> • 0 = SYSCLK[N] is gated. • 1 = SYSCLK[N] is on. |

(1) Where N = 1, 2, 3,....N (Not all these output clocks may be used on a specific device. For more information, see the device-specific data manual)

8.5.2.6 Reset Type Status Register (RSTYPE)

The Reset Type Status (RSTYPE) Register latches the cause of the last reset. If multiple reset sources occur simultaneously, this register latches the highest priority reset source. The Reset Type Status Register is shown in [Figure 8-13](#) and described in [Table 8-20](#).

Figure 8-13. Reset Type Status Register (RSTYPE)

| | | | | | | | | | | | |
|----------|----|---------|----------|----|----------|---|----------|---|-------------|-------|-----|
| 31 | 29 | 28 | 27 | 12 | 11 | 8 | 7 | 3 | 2 | 1 | 0 |
| Reserved | | EMU-RST | Reserved | | WDRST[N] | | Reserved | | PLLCTRL RST | RESET | POR |
| R-0 | | R-0 | R-0 | | R-0 | | R-0 | | R-0 | R-0 | R-0 |

Legend: R = Read only; -n = value after reset

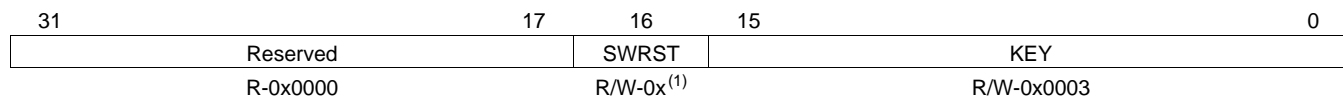
Table 8-20. Reset Type Status Register (RSTYPE) Field Descriptions

| Bit | Field | Description |
|-------|-----------|--|
| 31-29 | Reserved | Reserved. Read only. Always reads as 0. Writes have no effect. |
| 28 | EMU-RST | Reset initiated by emulation. <ul style="list-style-type: none"> 0 = Not the last reset to occur. 1 = The last reset to occur. |
| 27-12 | Reserved | Reserved. Read only. Always reads as 0. Writes have no effect. |
| 11 | WDRST3 | Reset initiated by watchdog timer[N]. <ul style="list-style-type: none"> 0 = Not the last reset to occur. 1 = The last reset to occur. |
| 10 | WDRST2 | Reset initiated by watchdog timer[N]. <ul style="list-style-type: none"> 0 = Not the last reset to occur. 1 = The last reset to occur. |
| 9 | WDRST1 | Reset initiated by watchdog timer[N]. <ul style="list-style-type: none"> 0 = Not the last reset to occur. 1 = The last reset to occur. |
| 8 | WDRST0 | Reset initiated by watchdog timer[N]. <ul style="list-style-type: none"> 0 = Not the last reset to occur. 1 = The last reset to occur. |
| 7-3 | Reserved | Reserved. Read only. Always reads as 0. Writes have no effect. |
| 2 | PLLCTRLST | Reset initiated by PLLCTL. <ul style="list-style-type: none"> 0 = Not the last reset to occur. 1 = The last reset to occur. |
| 1 | RESET | RESET reset. <ul style="list-style-type: none"> 0 = RESET was not the last reset to occur. 1 = RESET was the last reset to occur. |
| 0 | POR | Power-on reset. <ul style="list-style-type: none"> 0 = Power-on reset was not the last reset to occur. 1 = Power-on reset was the last reset to occur. |

8.5.2.7 Reset Control Register (RSTCTRL)

This register contains a key that enables writes to the MSB of this register and the RSTCFG Register. The key value is 0x5A69. A valid key will be stored as 0x000C, any other key value is invalid. When the RSTCTRL or the RSTCFG is written, the key is invalidated. Every write must be set up with a valid key. The Software Reset Control Register (RSTCTRL) is shown in [Figure 8-14](#) and described in [Table 8-21](#).

Figure 8-14. Reset Control Register (RSTCTRL)



Legend: R = Read only; -n = value after reset;

(1) Writes are conditional based on valid key.

Table 8-21. Reset Control Register (RSTCTRL) Field Descriptions

| Bit | Field | Description |
|-------|----------|---|
| 31-17 | Reserved | Reserved. |
| 16 | SWRST | Software reset <ul style="list-style-type: none"> • 0 = Reset • 1 = Not reset |
| 15-0 | KEY | Key used to enable writes to RSTCTRL and RSTCFG. |

8.5.2.8 Reset Configuration Register (RSTCFG)

This register is used to configure the type of reset initiated by $\overline{\text{RESET}}$, watchdog timer and the PLL controller's RSTCTRL Register; i.e., a hard reset or a soft reset. By default, these resets will be hard resets. The Reset Configuration Register (RSTCFG) is shown in [Figure 8-15](#) and described in [Table 8-22](#).

Figure 8-15. Reset Configuration Register (RSTCFG)

| | | | | | | | |
|----------|----|----------------------|--------------------------------|----------|---|---------------------------|---|
| 31 | 14 | 13 | 12 | 11 | 4 | 3 | 0 |
| Reserved | | PLLCTLRST TYPE | $\overline{\text{RESET}}$ TYPE | Reserved | | WDTYPE[N ⁽¹⁾] | |
| R-0 | | R/W-0 ⁽²⁾ | R/W-0 ² | R-0 | | R/W-0 ² | |

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

- (1) Where N = 1, 2, 3,...N (Not all these output may be used on a specific device. For more information, see the device-specific data manual)
- (2) Writes are conditional based on valid key. For details, see [Section 8.5.2.7](#).

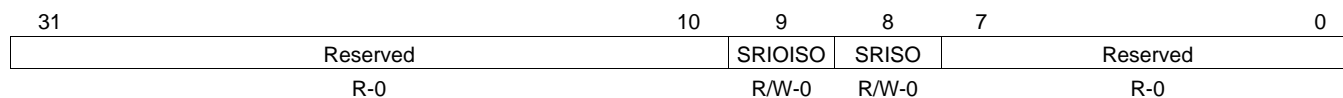
Table 8-22. Reset Configuration Register (RSTCFG) Field Descriptions

| Bit | Field | Description |
|-------|--------------------------------|--|
| 31-14 | Reserved | Reserved. |
| 13 | PLLCTLRSTTYPE | PLL controller initiates a software-driven reset of type: <ul style="list-style-type: none"> 0 = Hard reset (default) 1 = Soft reset |
| 12 | $\overline{\text{RESET}}$ TYPE | $\overline{\text{RESET}}$ initiates a reset of type: <ul style="list-style-type: none"> 0 = Hard Reset (default) 1 = Soft Reset |
| 11-4 | Reserved | Reserved. |
| 3 | WDTYPE3 | Watchdog timer [N] initiates a reset of type: <ul style="list-style-type: none"> 0 = Hard Reset (default) 1 = Soft Reset |
| 2 | WDTYPE2 | Watchdog timer [N] initiates a reset of type: <ul style="list-style-type: none"> 0 = Hard Reset (default) 1 = Soft Reset |
| 1 | WDTYPE1 | Watchdog timer [N] initiates a reset of type: <ul style="list-style-type: none"> 0 = Hard Reset (default) 1 = Soft Reset |
| 0 | WDTYPE0 | Watchdog timer [N] initiates a reset of type: <ul style="list-style-type: none"> 0 = Hard Reset (default) 1 = Soft Reset |

8.5.2.9 Reset Isolation Register (RSISO)

This register is used to select the module clocks that must maintain their clocking without pausing through non power-on reset. Setting any of these bits effectively blocks reset to all PLLCTL registers in order to maintain current values of PLL multiplier, divide ratios, and other settings. Along with setting module specific bit in RSISO, the corresponding MDCTLx[12] bit also needs to be set in PSC to reset-isolate a particular module. For more information on MDCTLx Register see the *Power Sleep Controller (PSC) for KeyStone Devices User's Guide (SPRUGV4)*. The Reset Isolation Register (RSTCTRL) is shown below.

Figure 8-16. Reset Isolation Register (RSISO)



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

Table 8-23. Reset Isolation Register (RSISO) Field Descriptions

| Bit | Field | Description |
|-------|----------|--|
| 31-10 | Reserved | Reserved. |
| 9 | SRIOISO | Isolate SRIO module <ul style="list-style-type: none"> • 0 = Not reset isolated • 1 = Reset Isolated |
| 8 | SRISO | Isolate SmartReflex <ul style="list-style-type: none"> • 0 = Not reset isolated • 1 = Reset Isolated |
| 7-0 | Reserved | Reserved. |

NOTE

The boot ROM code will enable the reset isolation for both SRIO and SmartReflex modules during boot with the Reset Isolation Register. It is up to the user application to disable.

8.5.3 Main PLL Control Register

The Main PLL uses two chip-level registers (MAINPLLCTL0 and MAINPLLCTL1) along with the PLL controller for its configuration. These MMRs exist inside the Bootcfg space. To write to these registers, software should go through an unlocking sequence using KICK0/KICK1 registers. For valid configurable values into the MAINPLLCTL0 and MAINPLLCTL1 Registers, see [Section 3.6](#). See [Section 4.3.4](#) for the address location of the registers and locking and unlocking sequences for accessing the registers. The registers are reset on $\overline{\text{POR}}$ only.

Figure 8-17. Main PLL Control Register 0 (MAINPLLCTL0)

| | | | | | | | | | |
|--------------|----|-----------|----|----|------------|----|------------|---|------------|
| 31 | 24 | 23 | 19 | 18 | 12 | 11 | 6 | 5 | 0 |
| BWADJ[7:0] | | Reserved | | | PLLM[12:6] | | Reserved | | PLLD |
| RW-0000 0101 | | RW-0000 0 | | | RW-0000000 | | RW-0000000 | | RW-0000000 |

Legend: RW = Read/Write; -n = value after reset

Table 8-24. Main PLL Control Register 0 (MAINPLLCTL0) Field Descriptions

| Bit | Field | Description |
|-------|------------|---|
| 31-24 | BWADJ[7:0] | BWADJ[11:8] and BWADJ[7:0] are located in separate registers. The combination (BWADJ[11:0]) should be programmed to a value related to PLLM[12:0] value based on the equation: $\text{BWADJ} = ((\text{PLLM}+1) \gg 1) - 1$ |
| 23-19 | Reserved | Reserved |
| 18-12 | PLLM[12:6] | A 13-bit bus that selects the values for the multiplication factor (see Note below) |
| 11-6 | Reserved | Reserved |
| 5-0 | PLLD | A 6-bit bus that selects the values for the reference divider |

Figure 8-18. Main PLL Control Register 1 (MAINPLLCTL1)

| | | | | | | | |
|-------------------------------------|---|---|---|---|-------|----------|-------------|
| 31 | 7 | 6 | 5 | 4 | 3 | 0 | |
| Reserved | | | | | ENSAT | Reserved | BWADJ[11:8] |
| RW-00000000000000000000000000000000 | | | | | RW-0 | RW-00 | RW-0000 |

Legend: RW = Read/Write; -n = value after reset

Table 8-25. Main PLL Control Register 1 (MAINPLLCTL1) Field Descriptions

| Bit | Field | Description |
|------|-------------|---|
| 31-7 | Reserved | Reserved |
| 6 | ENSAT | Needs to be set to 1 for proper operation of PLL |
| 5-4 | Reserved | Reserved |
| 3-0 | BWADJ[11:8] | BWADJ[11:8] and BWADJ[7:0] are located in separate registers. The combination (BWADJ[11:0]) should be programmed to a value related to PLLM[12:0] value based on the equation: $\text{BWADJ} = ((\text{PLLM}+1) \gg 1) - 1$ |

NOTE

PLLM[5:0] bits of the multiplier are controlled by the PLLM Register inside the PLL controller and PLLM[12:6] bits are controlled by the MAINPLLCTL0 chip-level register. The MAINPLLCTL0 Register PLLM[12:6] bits should be written just before writing to the PLLM Register PLLM[5:0] bits in the controller to have the complete 13-bit value latched when the GO operation is initiated in the PLL controller. See the *Phase Locked Loop (PLL) for KeyStone Devices User's Guide (SPRUGV2)* for the recommended programming sequence. Output divide ratio and bypass enable/disable of the Main PLL is controlled by the SECCTL Register in the PLL Controller. See the [Section 8.5.2.1](#) for more details.

8.5.4 Main PLL and PLL Controller Initialization Sequence

See the *Phase Locked Loop (PLL) for KeyStone Devices User's Guide* ([SPRUGV2](#)) for details on the initialization sequence for Main PLL and PLL Controller.

8.5.5 Main PLL Controller/SRIO/HyperLink/PCIe Clock Input Electrical Data/Timing

Table 8-26. Main PLL Controller/SRIO/HyperLink/PCIe Clock Input Timing Requirements

(see [Figure 8-19](#) and [Figure 8-20](#))

| NO. | | | MIN | MAX | UNIT |
|--------------------------|------------------------|--|------------------------|------------------------|--------|
| CORECLK[P:N] | | | | | |
| 1 | tc(CORCLKN) | Cycle time _ CORECLKN cycle time | 3.2 | 25 | ns |
| 1 | tc(CORECLKP) | Cycle time _ CORECLKP cycle time | 3.2 | 25 | ns |
| 3 | tw(CORECLKN) | Pulse width _ CORECLKN high | 0.45*tc(CORECLKN) | 0.55*tc(CORECLKN) | ns |
| 2 | tw(CORECLKN) | Pulse width _ CORECLKN low | 0.45*tc(CORECLKN) | 0.55*tc(CORECLKN) | ns |
| 2 | tw(CORECLKP) | Pulse width _ CORECLKP high | 0.45*tc(CORECLKP) | 0.55*tc(CORECLKP) | ns |
| 3 | tw(CORECLKP) | Pulse width _ CORECLKP low | 0.45*tc(CORECLKP) | 0.55*tc(CORECLKP) | ns |
| 4 | tr(CORECLK_250mv) | Transition time _ CORECLK differential rise time (250mV) | 50 | 350 | ps |
| 4 | tf(CORECLK_250mv) | Transition time _ CORECLK differential fall time (250 mV) | 50 | 350 | ps |
| 5 | tj(CORECLKN) | Jitter, peak_to_peak _ periodic CORECLKN | | 100 | ps |
| 5 | tj(CORECLKP) | Jitter, peak_to_peak _ periodic CORECLKP | | 100 | ps |
| SRIOSGMIICLK[P:N] | | | | | |
| 1 | tc(SRIOSGMIICLKN) | Cycle time _ SRIOSGMIICLKN cycle time | 3.2 or 4 or 6.4 | | ns |
| 1 | tc(SRIOSGMIICLKP) | Cycle time _ SRIOSGMIICLKP cycle time | 3.2 or 4 or 6.4 | | ns |
| 3 | tw(SRIOSGMIICLKN) | Pulse width _ SRIOSGMIICLKN high | 0.45*tc(SRIOSGMIICLKN) | 0.55*tc(SRIOSGMIICLKN) | ns |
| 2 | tw(SRIOSGMIICLKN) | Pulse width _ SRIOSGMIICLKN low | 0.45*tc(SRIOSGMIICLKN) | 0.55*tc(SRIOSGMIICLKN) | ns |
| 2 | tw(SRIOSGMIICLKP) | Pulse width _ SRIOSGMIICLKP high | 0.45*tc(SRIOSGMIICLKP) | 0.55*tc(SRIOSGMIICLKP) | ns |
| 3 | tw(SRIOSGMIICLKP) | Pulse width _ SRIOSGMIICLKP low | 0.45*tc(SRIOSGMIICLKP) | 0.55*tc(SRIOSGMIICLKP) | ns |
| 4 | tr(SRIOSGMIICLK_250mv) | Transition time _ SRIOSGMIICLK differential rise time (250 mV) | 50 | 350 | ps |
| 4 | tf(SRIOSGMIICLK_250mv) | Transition time _ SRIOSGMIICLK differential fall time (250 mV) | 50 | 350 | ps |
| 5 | tj(SRIOSGMIICLKN) | Jitter, peak_to_peak _ periodic SRIOSGMIICLKN | 4 | | ps,RMS |
| 5 | tj(SRIOSGMIICLKP) | Jitter, peak_to_peak _ periodic SRIOSGMIICLKP | | 4 | ps,RMS |
| 5 | tj(SRIOSGMIICLKN) | Jitter, peak_to_peak _ periodic SRIOSGMIICLKN (SRIO not used) | | 8 | ps,RMS |
| 5 | tj(SRIOSGMIICLKP) | Jitter, peak_to_peak _ periodic SRIOSGMIICLKP (SRIO not used) | | 8 | ps,RMS |
| HyperLinkCLK[P:N] | | | | | |
| 1 | tc(MCMCLKN) | Cycle time _ MCMCLKN cycle time | 3.2 | 6.4 | ns |
| 1 | tc(MCMCLKP) | Cycle time _ MCMCLKP cycle time | 3.2 | 6.4 | ns |
| 3 | tw(MCMCLKN) | Pulse width _ MCMCLKN high | 0.45*tc(MCMCLKN) | 0.55*tc(MCMCLKN) | ns |
| 2 | tw(MCMCLKN) | Pulse width _ MCMCLKN low | 0.45*tc(MCMCLKN) | 0.55*tc(MCMCLKN) | ns |
| 2 | tw(MCMCLKP) | Pulse width _ MCMCLKP high | 0.45*tc(MCMCLKP) | 0.55*tc(MCMCLKP) | ns |
| 3 | tw(MCMCLKP) | Pulse width _ MCMCLKP low | 0.45*tc(MCMCLKP) | 0.55*tc(MCMCLKP) | ns |

Table 8-26. Main PLL Controller/SRIO/HyperLink/PCIe Clock Input Timing Requirements (continued)

(see [Figure 8-19](#) and [Figure 8-20](#))

| NO. | | | MIN | MAX | UNIT |
|---------------------|-------------------|---|-------------------|-------------------|--------|
| 4 | tr(MCMCLK_250mv) | Transition time _ MCMCLK differential rise time (250mV) | 50 | 350 | ps |
| 4 | tf(MCMCLK_250mv) | Transition time _ MCMCLK differential fall time (250mV) | 50 | 350 | ps |
| 5 | tj(MCMCLKN) | Jitter, peak_to_peak _ periodic MCMCLKN | | 4 | ps,RMS |
| 5 | tj(MCMCLKP) | Jitter, peak_to_peak _ periodic MCMCLKP | | 4 | ps,RMS |
| PCIECLK[P:N] | | | | | |
| 1 | tc(PCIECLKN) | Cycle time _ PCIECLKN cycle time | 3.2 | 10 | ns |
| 1 | tc(PCIECLKP) | Cycle time _ PCIECLKP cycle time | 3.2 | 10 | ns |
| 3 | tw(PCIECLKN) | Pulse width _ PCIECLKN high | 0.45*tc(PCIECLKN) | 0.55*tc(PCIECLKN) | ns |
| 2 | tw(PCIECLKN) | Pulse width _ PCIECLKN low | 0.45*tc(PCIECLKN) | 0.55*tc(PCIECLKN) | ns |
| 2 | tw(PCIECLKP) | Pulse width _ PCIECLKP high | 0.45*tc(PCIECLKP) | 0.55*tc(PCIECLKP) | ns |
| 3 | tw(PCIECLKP) | Pulse width _ PCIECLKP low | 0.45*tc(PCIECLKP) | 0.55*tc(PCIECLKP) | ns |
| 4 | tr(PCIECLK_250mv) | Transition time _ PCIECLK differential rise time (250 mV) | 50 | 350 | ps |
| 4 | tf(PCIECLK_250mv) | Transition time _ PCIECLK differential fall time (250 mV) | 50 | 350 | ps |
| 5 | tj(PCIECLKN) | Jitter, peak_to_peak _ periodic PCIECLKN | | 4 | ps,RMS |
| 5 | tj(PCIECLKP) | Jitter, peak_to_peak _ periodic PCIECLKP | | 4 | ps,RMS |

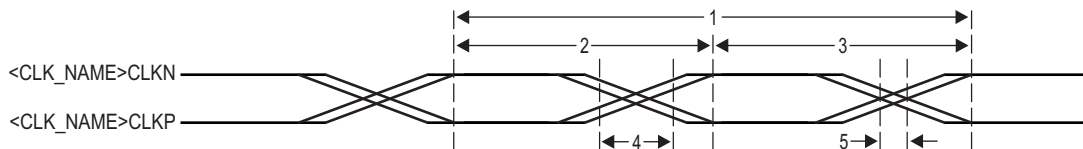


Figure 8-19. Main PLL Controller/SRIO/HyperLink/PCIe Clock Input Timing

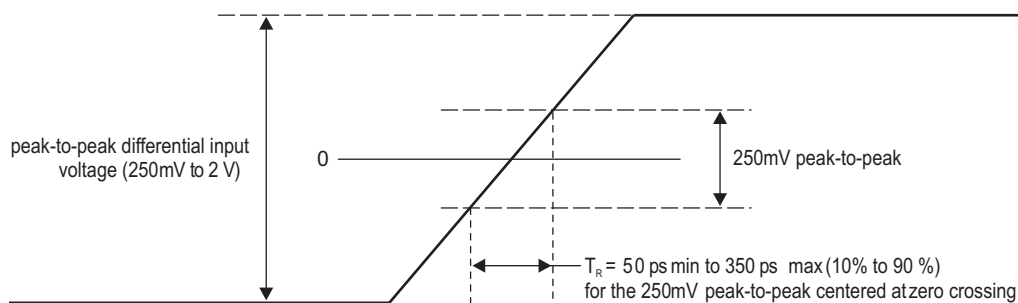


Figure 8-20. Main PLL Clock Input Transition Time

8.6 DDR3 PLL

The DDR3 PLL generates interface clocks for the DDR3 memory controller. When coming out of power-on reset, the DDR3 PLL is programmed to a valid frequency during the boot config before being enabled and used.

DDR3 PLL power is supplied externally via the Main PLL power-supply pin (AVDDA2). An external EMI filter circuit must be added to all PLL supplies. See the *Hardware Design Guide for KeyStone Devices* (SPRABI2). For the best performance, TI recommends that all the PLL external components be on a single side of the board without jumpers, switches, or components other than those shown. For reduced PLL jitter, maximize the spacing between switching signal traces and the PLL external components (C1, C2, and the EMI Filter).

Figure 8-21 shows the DDR3 PLL.

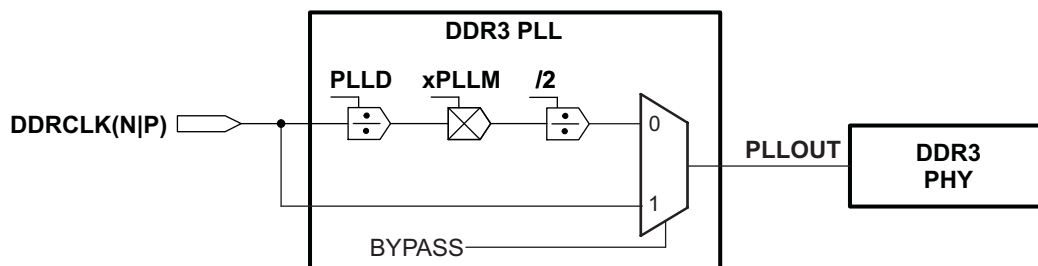


Figure 8-21. DDR3 PLL Block Diagram

8.6.1 DDR3 PLL Control Register

The DDR3 PLL, which is used to drive the DDR PHY for the EMIF, does not use a PLL controller. The DDR3 PLL can be controlled using the DDR3PLLCTL0 and DDR3PLLCTL1 Registers located in the Bootcfg module. These MMRs exist inside the Bootcfg space. To write to these registers, software should go through an un-locking sequence using the KICK0/KICK1 registers. For suggested configurable values, see Section 4.3.4 for the address location of the registers and locking and unlocking sequences for accessing the registers. This register is reset on POR only.

Figure 8-22. DDR3 PLL Control Register 0 (DDR3PLLCTL0)⁽¹⁾

| | | | | | | | | |
|---------------|----|--------|----------|------------------|----|---|------------|---|
| 31 | 24 | 23 | 22 | 19 | 18 | 6 | 5 | 0 |
| BWADJ[7:0] | | BYPASS | Reserved | PLLM | | | PLLD | |
| RW,+0000 1001 | | RW,+0 | RW,+0001 | RW,+000000010011 | | | RW,+000000 | |

Legend: RW = Read/Write; -n = value after reset

(1) This register is Reset on POR only. The regreset, reset and bgreset from PLL are all tied to a common pll0_ctrl_rst_n. The pwrdsn, regpwrdsn, bgpwrdsn are all tied to common pll0_ctrl_to_pll_pwrdsn.

Table 8-27. DDR3 PLL Control Register 0 Field Descriptions

| Bit | Field | Description |
|-------|------------|--|
| 31-24 | BWADJ[7:0] | BWADJ[11:8] and BWADJ[7:0] are located in DDR3PLLCTL0 and DDR3PLLCTL1 registers. The combination (BWADJ[11:0]) should be programmed to a value related to PLLM[12:0] value based on the equation: BWADJ = ((PLLM+1)>>1) -1 |
| 23 | BYPASS | Enable bypass mode <ul style="list-style-type: none"> 0 = Bypass disabled 1 = Bypass enabled |
| 22-19 | Reserved | Reserved |
| 18-6 | PLLM | A 13-bit bus that selects the values for the multiplication factor |
| 5-0 | PLLD | A 6-bit bus that selects the values for the reference divider |

Figure 8-23. DDR3 PLL Control Register 1 (DDR3PLLCTL1)

| | | | | | | | | | |
|-----------------------|----|-------|-----------|---|-------|----------|---|-------------|---|
| 31 | 14 | 13 | 12 | 7 | 6 | 5 | 4 | 3 | 0 |
| Reserved | | PLLST | Reserved | | ENSAT | Reserved | | BWADJ[11:8] | |
| RW-000000000000000000 | | RW-0 | RW-000000 | | RW-0 | R-0 | | RW-0000 | |

Legend: RW = Read/Write; -n = value after reset

Table 8-28. DDR3 PLL Control Register 1 Field Descriptions

| Bit | Field | Description |
|-------|-------------|---|
| 31-14 | Reserved | Reserved |
| 13 | PLLST | PLL reset bit. <ul style="list-style-type: none"> • 0 = PLL reset is released. • 1 = PLL reset is asserted. |
| 12-7 | Reserved | Reserved |
| 6 | ENSAT | Needs to be set to 1 for proper operation of the PLL |
| 5-4 | Reserved | Reserved |
| 3-0 | BWADJ[11:8] | BWADJ[11:8] and BWADJ[7:0] are located in separate registers. The combination (BWADJ[11:0]) should be programmed to a value related to PLLM[12:0] value based on the equation: $BWADJ = ((PLLM+1) \gg 1) - 1$ |

8.6.2 DDR3 PLL Device-Specific Information

As shown in [Figure 8-21](#), the output of DDR3 PLL (PLLOUT) is divided by 2 and directly fed to the DDR3 memory controller. The DDR3 PLL is affected by power-on reset. During power-on resets, the internal clocks of the DDR3 PLL are affected as described in [Section 8.4](#). The DDR3 PLL is unlocked only during the power-up sequence and is locked by the time the $\overline{RESETSTAT}$ pin goes high. It does not lose lock during any of the other resets.

8.6.3 DDR3 PLL Initialization Sequence

See the *Phase Locked Loop (PLL) for KeyStone Devices User's Guide* ([SPRUGV2](#)) for details on the initialization sequence for DDR3 PLL.

8.6.4 DDR3 PLL Input Clock Electrical Data/Timing

Table 8-29. DDR3 PLL DDRSYSCLK1(N|P) Timing Requirements

(see Figure 8-24 and Figure 8-20)

| NO | DDRCLK[P:N] | | MIN | MAX | UNIT |
|----|------------------|--|-------------------|------------------|------|
| 1 | tc(DDRCLKN) | Cycle time _ DDRCLKN cycle time | 3.2 | 25 | ns |
| 1 | tc(DDRCLKP) | Cycle time _ DDRCLKP cycle time | 3.2 | 25 | ns |
| 3 | tw(DDRCLKN) | Pulse width _ DDRCLKN high | 0.45*tc(DDRCLKN) | 0.55*tc(DDRCLKN) | ns |
| 2 | tw(DDRCLKN) | Pulse width _ DDRCLKN low | 0.45*tc(DDRCLKN) | 0.55*tc(DDRCLKN) | ns |
| 2 | tw(DDRCLKP) | Pulse width _ DDRCLKP high | 0.45*tc(DDRCLKP) | 0.55*tc(DDRCLKP) | ns |
| 3 | tw(DDRCLKP) | Pulse width _ DDRCLKP low | 0.45*tc(DDRCLKP) | 0.55*tc(DDRCLKP) | ns |
| 4 | tr(DDRCLK_250mv) | Transition time _ DDRCLK differential rise time (250 mV) | 50 | 350 | ps |
| 4 | tf(DDRCLK_250mv) | Transition time _ DDRCLK differential fall time (250 mV) | 50 | 350 | ps |
| 5 | tj(DDRCLKN) | Jitter, peak_to_peak _ periodic DDRCLKN | 0.025*tc(DDRCLKN) | | ps |
| 5 | tj(DDRCLKP) | Jitter, peak_to_peak _ periodic DDRCLKP | 0.025*tc(DDRCLKP) | | ps |

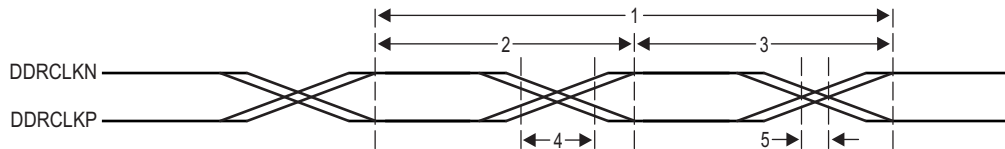


Figure 8-24. DDR3 PLL DDRCLK Timing

8.7 Enhanced Direct Memory Access (EDMA3) Controller

The primary purpose of the EDMA3 is to service user-programmed data transfers between two memory-mapped slave endpoints on the device. The EDMA3 services software-driven paging transfers (e.g., data movement between external memory and internal memory), performs sorting or subframe extraction of various data structures, services event driven peripherals, and offloads data transfers from the device CPU.

There is one EDMA Channel Controller on the C665x device: EDMA3_CC. It has four transfer controllers: TC0, TC1, TC2, and TC3. In the context of this document, TCx associated with CC is referred to as EDMA3_CC_TCx. Each of the transfer controllers has a direct connection to the switch fabric. [Section 5.2](#) lists the peripherals that can be accessed by the transfer controllers.

The EDMA3 Channel Controller includes the following features:

- Fully orthogonal transfer description
 - Three transfer dimensions:
 - Array (multiple bytes)
 - Frame (multiple arrays)
 - Block (multiple frames)
 - Single event can trigger transfer of array, frame, or entire block
 - Independent indexes on source and destination
- Flexible transfer definition:
 - Increment or FIFO transfer addressing modes
 - Linking mechanism allows for ping-pong buffering, circular buffering, and repetitive/continuous transfers, all with no CPU intervention
 - Chaining allows multiple transfers to execute with one event
- 512 PaRAM entries
 - Used to define transfer context for channels
 - Each PaRAM entry can be used as a DMA entry, QDMA entry, or link entry
- 64 DMA channels
 - Manually triggered (CPU writes to channel controller register), external event triggered, and chain triggered (completion of one transfer triggers another)
- Eight Quick DMA (QDMA) channels
 - Used for software-driven transfers
 - Triggered upon writing to a single PaRAM set entry
- Four transfer controllers and four event queues with programmable system-level priority
- Interrupt generation for transfer completion and error conditions
- Debug visibility
 - Queue watermarking/threshold allows detection of maximum usage of event queues
 - Error and status recording to facilitate debug

8.7.1 EDMA3 Device-Specific Information

The EDMA supports two addressing modes: constant addressing and increment addressing mode. Constant addressing mode is applicable to a very limited set of use cases. For most applications, increment mode must be used. On the C665x, the EDMA can use constant addressing mode only with the Enhanced Viterbi-Decoder Coprocessor (VCP) and the Enhanced Turbo Decoder Coprocessor (TCP). Constant addressing mode is not supported by any other peripheral or internal memory in the device. Note that increment mode is supported by all peripherals, including VCP and TCP. For more information on these two addressing modes, see the *Enhanced Direct Memory Access 3 (EDMA3) for KeyStone Devices User's Guide* ([SPRUGS5](#)).

For the range of memory addresses that include EDMA3 channel controller (EDMA3_CC) control registers and EDMA3 transfer controller (TC) control register, see [Table 3-2](#). For memory offsets and other details on EDMA3_CC and TC control registers entries, see the *Enhanced Direct Memory Access 3 (EDMA3) for KeyStone Devices User's Guide* ([SPRUGS5](#)).

8.7.2 EDMA3 Channel Controller Configuration

[Table 8-30](#) provides the configuration of the EDMA3 channel controller present on the device.

Table 8-30. EDMA3 Channel Controller Configuration

| DESCRIPTION | EDMA3 CC |
|--|----------|
| Number of DMA channels in Channel Controller | 64 |
| Number of QDMA channels | 8 |
| Number of interrupt channels | 64 |
| Number of PaRAM set entries | 512 |
| Number of event queues | 4 |
| Number of Transfer Controllers | 4 |
| Memory Protection Existence | Yes |
| Number of Memory Protection and Shadow Regions | 8 |

8.7.3 EDMA3 Transfer Controller Configuration

Each transfer controller on a device is designed differently based on considerations like performance requirements, system topology (like main TeraNet bus width, external memory bus width), and so on. The parameters that determine the transfer controller configurations are:

- **FIFOSIZE:** Determines the size in bytes for the data FIFO that is the temporary buffer for the in-flight data. The data FIFO is where the read return data read by the TC read controller from the source endpoint is stored and subsequently written out to the destination endpoint by the TC write controller.
- **BUSWIDTH:** The width of the read and write data buses, in bytes, for the TC read and write controller, respectively. This is typically equal to the bus width of the main TeraNet interface.
- **Default Burst Size (DBS):** The DBS is the maximum number of bytes per read/write command issued by a transfer controller.
- **DSTREGDEPTH:** This determines the number of destination FIFO register set. The number of destination FIFO register set for a transfer controller determines the maximum number of outstanding transfer requests.

All four parameters listed above are specified by the design of the device.

Table 8-31 provides the configuration of the EDMA3 transfer controller present on the device.

Table 8-31. EDMA3 Transfer Controller Configuration

| PARAMETER | EDMA3 CC | | | |
|-------------|------------|-----------|-----------|------------|
| | TC0 | TC1 | TC2 | TC3 |
| FIFOSIZE | 1024 bytes | 512 bytes | 512 bytes | 1024 bytes |
| BUSWIDTH | 16 bytes | 16 bytes | 16 bytes | 16 bytes |
| DSTREGDEPTH | 4 entries | 4 entries | 4 entries | 4 entries |
| DBS | 64 bytes | 64 bytes | 64 bytes | 64 bytes |

8.7.4 EDMA3 Channel Synchronization Events

The EDMA3 supports up to 64 DMA channels for EDMA3_CC that can be used to service system peripherals and to move data between system memories. DMA channels can be triggered by synchronization events generated by system peripherals. The following tables lists the source of the synchronization event associated with each of the EDMA3_CC DMA channels. On the C665x, the association of each synchronization event and DMA channel is fixed and cannot be reprogrammed.

For more detailed information on the EDMA3 module and how EDMA3 events are enabled, captured, processed, prioritized, linked, chained, and cleared, etc., see the *Enhanced Direct Memory Access 3 (EDMA3) for KeyStone Devices User's Guide* ([SPRUGS5](#)).

Table 8-32. EDMA3_CC Events for C665x

| EVENT NUMBER | EVENT | EVENT DESCRIPTION |
|--------------|--------------|------------------------|
| 0 | TCP3D_AREVT0 | TCP3D_A receive event0 |
| 1 | TCP3D_AREVT1 | TCP3D_A receive event1 |
| 2 | TINT2L | Timer2 interrupt low |
| 3 | TINT2H | Timer2 interrupt high |
| 4 | URXEVT | UART0 receive event |
| 5 | UTXEVT | UART0 transmit event |
| 6 | GPINT0 | GPIO interrupt |
| 7 | GPINT1 | GPIO interrupt |
| 8 | GPINT2 | GPIO Interrupt |
| 9 | GPINT3 | GPIO interrupt |
| 10 | VCPAREVT | VCP2_A receive event |
| 11 | VCPAXEVT | VCP2_A transmit event |
| 12 | VCPBREVT | VCP2_B receive event |
| 13 | VCPBXEVT | VCP2_B transmit event |
| 14 | URXEVT_B | UART1 receive event |
| 15 | UTXEVT_B | UART1 transmit event |
| 16 | SPIINT0 | SPI interrupt |
| 17 | SPIINT1 | SPI interrupt |
| 18 | SEMINT0 | Semaphore interrupt |
| 19 | SEMINT1 | Semaphore interrupt |
| 20 | SEMINT2 | Semaphore interrupt |
| 21 | SEMINT3 | Semaphore interrupt |
| 22 | TINT4L | Timer4 interrupt low |
| 23 | TINT4H | Timer4 interrupt high |
| 24 | TINT5L | Timer5 interrupt low |

Table 8-32. EDMA3_CC Events for C665x (continued)

| EVENT NUMBER | EVENT | EVENT DESCRIPTION |
|--------------|---------------|-----------------------------|
| 25 | TINT5H | Timer5 interrupt high |
| 26 | TINT6L | Timer6 interrupt low |
| 27 | TINT6H | Timer6 interrupt high |
| 28 | TINT7L | Timer7 interrupt low |
| 29 | TINT7H | Timer7 interrupt high |
| 30 | SPIXEVT | SPI transmit event |
| 31 | SPIREVT | SPI receive event |
| 32 | I2CREVET | I2C receive event |
| 33 | I2CXEVT | I2C transmit event |
| 34 | TINT3L | Timer3 interrupt low |
| 35 | TINT3H | Timer3 interrupt high |
| 36 | MCBSP0_REVT | McBSP_0 receive event |
| 37 | MCBSP0_XEVT | McBSP_0 transmit event |
| 38 | MCBSP1_REVT | McBSP_1 receive event |
| 39 | MCBSP1_XEVT | McBSP_1 transmit event |
| 40 | TETBHFULLINT | TETB half full interrupt |
| 41 | TETBHFULLINT0 | TETB half full interrupt |
| 42 | TETBHFULLINT1 | TETB half full interrupt |
| 43 | CIC1_OUT0 | Interrupt Controller output |
| 44 | CIC1_OUT1 | Interrupt Controller output |
| 45 | CIC1_OUT2 | Interrupt Controller output |
| 46 | CIC1_OUT3 | Interrupt Controller output |
| 47 | CIC1_OUT4 | Interrupt Controller output |
| 48 | CIC1_OUT5 | Interrupt Controller output |
| 49 | CIC1_OUT6 | Interrupt Controller output |
| 50 | CIC1_OUT7 | Interrupt Controller output |
| 51 | CIC1_OUT8 | Interrupt Controller output |
| 52 | CIC1_OUT9 | Interrupt Controller output |
| 53 | CIC1_OUT10 | Interrupt Controller output |
| 54 | CIC1_OUT11 | Interrupt Controller output |
| 55 | CIC1_OUT12 | Interrupt Controller output |
| 56 | CIC1_OUT13 | Interrupt Controller output |
| 57 | CIC1_OUT14 | Interrupt Controller output |
| 58 | CIC1_OUT15 | Interrupt Controller output |
| 59 | CIC1_OUT16 | Interrupt Controller output |
| 60 | CIC1_OUT17 | Interrupt Controller output |
| 61 | TETBFULLINT | TETB full interrupt |
| 62 | TETBFULLINT0 | TETB full interrupt |
| 63 | TETBFULLINT1 | TETB full interrupt |

8.8 Interrupts

8.8.1 Interrupt Sources and Interrupt Controller

The CPU interrupts on the C665x device are configured through the C66x CorePac Interrupt Controller. The interrupt controller allows for up to 128 system events to be programmed to any of the twelve CPU interrupt inputs (CPUINT4 - CPUINT15), the CPU exception input (EXCEP), or the advanced emulation logic. The 128 system events consist of both internally-generated events (within the CorePac) and chip-level events.

Additional system events are routed to each of the C66x CorePacs to provide chip-level events that are not required as CPU interrupts/exceptions to be routed to the interrupt controller as emulation events. In addition, error-class events or infrequently used events are also routed through the system event router to offload the C66x CorePac interrupt selector. This is accomplished through CIC blocks, CIC[2:0]. This is clocked using CPU/6.

The event controllers consist of simple combination logic to provide additional events to the C66x CorePacs, plus the EDMA3_CC and CIC0 provide 12 additional events as well as 8 broadcast events to the C66x CorePacs. CIC1 provides 18 additional events to EDMA3_CC, and CIC2 provides 32 additional events to HyperLink.

There are a large number of events on the chip level. The chip level CIC provides a flexible way to combine and remap those events. Multiple events can be combined to a single event through chip level CIC. However, an event can be mapped only to a single event output from the chip level CIC. The chip level CIC also allows the software to trigger system events through memory writes. The broadcast events to C66x CorePacs can be used for synchronization among multiple cores, inter-processor communication purposes, etc. For more details on the CIC features, please refer to the *Chip Interrupt Controller (CIC) for KeyStone Devices User's Guide* ([SPRUGW4](#)).

NOTE

Modules such as MPU, Tracer, and BOOT_CFG have level interrupts and an EOI handshaking interface. The EOI value is 0 for MPU, Tracer, and BOOT_CFG.

[Figure 8-25](#) shows the C665x interrupt topology.

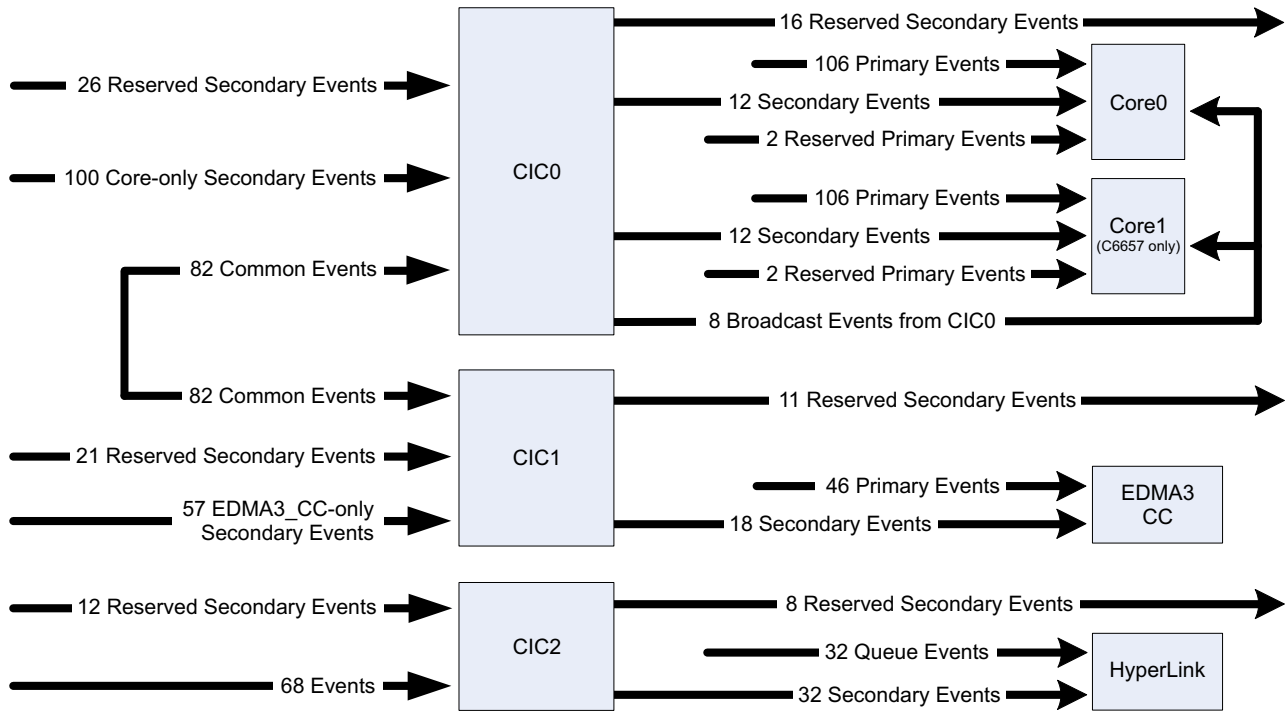


Figure 8-25. C665x Interrupt Topology

Table 8-33 shows the mapping of system events. For more information on the Interrupt Controller, see the *C66x CorePac User's Guide* ([SPRUGW0](#)).

Table 8-33. C665x System Event Inputs — C66x CorePac Primary Interrupts

| INPUT EVENT NUMBER | INTERRUPT EVENT | DESCRIPTION |
|--------------------|---|--|
| 0 | EVT0 | Event combiner 0 output |
| 1 | EVT1 | Event combiner 1 output |
| 2 | EVT2 | Event combiner 2 output |
| 3 | EVT3 | Event combiner 3 output |
| 4 | TETBHFULLINT _n ⁽¹⁾ | TETB is half full |
| 5 | TETBFULLINT _n ⁽¹⁾ | TETB is full |
| 6 | TETBACQINT _n ⁽¹⁾ | Acquisition has been completed |
| 7 | TETBOVFLINT _n ⁽¹⁾ | Overflow condition interrupt |
| 8 | TETBUNFLINT _n ⁽¹⁾ | Underflow condition interrupt |
| 9 | EMU_DTDMA | ECM interrupt for: <ul style="list-style-type: none"> 1. Host scan access 2. DTDMA transfer complete 3. AET interrupt |
| 10 | MSMC_mpf_errorn ⁽²⁾ | Memory protection fault indicators for local core |
| 11 | EMU_RTDXRX | RTDX receive complete |
| 12 | EMU_RTDXTX | RTDX transmit complete |
| 13 | IDMA0 | IDMA channel 0 interrupt |
| 14 | IDMA1 | IDMA channel 1 interrupt |
| 15 | SEMERR _n ⁽³⁾ | Semaphore error interrupt |
| 16 | SEMINT _n ⁽³⁾ | Semaphore interrupt |
| 17 | PCIEExpress_MSI_INT _n ⁽⁴⁾ | Message signaled interrupt mode |
| 18 | PCIEExpress_MSI_INT _{n+4} ⁽⁴⁾ | Message signaled interrupt mode |
| 19 | MACINT _n ⁽⁵⁾ | EMAC interrupt |
| 20 | INTDST _(n+16) ⁽⁶⁾ | SRIO Interrupt |
| 21 | INTDST _(n+20) ⁽⁷⁾ | SRIO Interrupt |
| 22 | CIC0_OUT _(0+20*n) ⁽⁸⁾ | Interrupt Controller Output |
| 23 | CIC0_OUT _(1+20*n) ⁽⁸⁾ | Interrupt Controller Output |
| 24 | CIC0_OUT _(2+20*n) ⁽⁸⁾ | Interrupt Controller Output |
| 25 | CIC0_OUT _(3+20*n) ⁽⁸⁾ | Interrupt Controller Output |
| 26 | CIC0_OUT _(4+20*n) ⁽⁸⁾ | Interrupt Controller Output |
| 27 | CIC0_OUT _(5+20*n) ⁽⁸⁾ | Interrupt Controller Output |
| 28 | CIC0_OUT _(6+20*n) ⁽⁸⁾ | Interrupt Controller Output |
| 29 | CIC0_OUT _(7+20*n) ⁽⁸⁾ | Interrupt Controller Output |
| 30 | CIC0_OUT _(8+20*n) ⁽⁸⁾ | Interrupt Controller Output |
| 31 | CIC0_OUT _(9+20*n) ⁽⁸⁾ | Interrupt Controller Output |
| 32 | QM_INT_LOW_0 | QM Interrupt for 0–31 Queues |
| 33 | QM_INT_LOW_1 | QM Interrupt for 32–63 Queues |
| 34 | QM_INT_LOW_2 | QM Interrupt for 64–95 Queues |
| 35 | QM_INT_LOW_3 | QM Interrupt for 96–127 Queues |

(1) CorePac_[n] will receive TETBHFULLINT_n, TETBFULLINT_n, TETBACQINT_n, TETBOVFLINT_n, and TETBUNFLINT_n.

(2) CorePac_[n] will receive MSMC_mpf_errorn.

(3) CorePac_[n] will receive SEMINT_n and SEMERR_n.

(4) CorePac_[n] will receive PCIEExpress_MSI_INT_n.

(5) CorePac_[n] will receive MACINT_n/MACRXINT_n/MACTXINT_n/MACTRESH_n.

(6) CorePac_[n] will receive INTDST_(n+16).

(7) CorePac_[n] will receive INTDST_(n+20).

(8) n is core number.

Table 8-33. C665x System Event Inputs — C66x CorePac Primary Interrupts (continued)

| INPUT EVENT NUMBER | INTERRUPT EVENT | DESCRIPTION |
|--------------------|---------------------------------------|---|
| 36 | QM_INT_LOW_4 | QM Interrupt for 128~159 Queues |
| 37 | QM_INT_LOW_5 | QM Interrupt for 160~191 Queues |
| 38 | QM_INT_LOW_6 | QM Interrupt for 192~223 Queues |
| 39 | QM_INT_LOW_7 | QM Interrupt for 224~255 Queues |
| 40 | QM_INT_LOW_8 | QM Interrupt for 256~287 Queues |
| 41 | QM_INT_LOW_9 | QM Interrupt for 288~319 Queues |
| 42 | QM_INT_LOW_10 | QM Interrupt for 320~351 Queues |
| 43 | QM_INT_LOW_11 | QM Interrupt for 352~383 Queues |
| 44 | QM_INT_LOW_12 | QM Interrupt for 384~415 Queues |
| 45 | QM_INT_LOW_13 | QM Interrupt for 416~447 Queues |
| 46 | QM_INT_LOW_14 | QM Interrupt for 448~479 Queues |
| 47 | QM_INT_LOW_15 | QM Interrupt for 480~511 Queues |
| 48 | QM_INT_HIGH_n ⁽⁸⁾ | QM Interrupt for Queue 704+n ⁽⁸⁾ |
| 49 | QM_INT_HIGH_(n+4) ⁽⁸⁾ | QM Interrupt for Queue 708+n ⁽⁸⁾ |
| 50 | QM_INT_HIGH_(n+8) ⁽⁸⁾ | QM Interrupt for Queue 712+n ⁽⁸⁾ |
| 51 | QM_INT_HIGH_(n+12) ⁽⁸⁾ | QM Interrupt for Queue 716+n ⁽⁸⁾ |
| 52 | QM_INT_HIGH_(n+16) ⁽⁸⁾ | QM Interrupt for Queue 720+n ⁽⁸⁾ |
| 53 | QM_INT_HIGH_(n+20) ⁽⁸⁾ | QM Interrupt for Queue 724+n ⁽⁸⁾ |
| 54 | QM_INT_HIGH_(n+24) ⁽⁸⁾ | QM Interrupt for Queue 728+n ⁽⁸⁾ |
| 55 | QM_INT_HIGH_(n+28) ⁽⁸⁾ | QM Interrupt for Queue 732+n ⁽⁸⁾ |
| 56 | CIC0_OUT40 | Interrupt Controller Output |
| 57 | CIC0_OUT41 | Interrupt Controller Output |
| 58 | CIC0_OUT42 | Interrupt Controller Output |
| 59 | CIC0_OUT43 | Interrupt Controller Output |
| 60 | CIC0_OUT44 | Interrupt Controller Output |
| 61 | CIC0_OUT45 | Interrupt Controller Output |
| 62 | CIC0_OUT46 | Interrupt Controller Output |
| 63 | CIC0_OUT47 | Interrupt Controller Output |
| 64 | TINTLn ⁽⁹⁾ | Local timer interrupt low |
| 65 | TINTHn ⁽⁹⁾ | Local timer interrupt high |
| 66 | TINT2L | Timer2 interrupt low |
| 67 | TINT2H | Timer2 interrupt high |
| 68 | TINT3L | Timer3 interrupt low |
| 69 | TINT3H | Timer3 interrupt high |
| 70 | PCIEExpress_MSI_INTn+2 ⁽⁴⁾ | Message signaled interrupt mode |
| 71 | PCIEExpress_MSI_INTn+6 ⁽⁴⁾ | Message signaled interrupt mode |
| 72 | GPINT2 | GPIO interrupt |
| 73 | GPINT3 | GPIO interrupt |
| 74 | MACINTn+2 ⁽⁵⁾ | EMAC interrupt |
| 75 | MACTXINTn+2 ⁽⁵⁾ | EMAC interrupt |
| 76 | MACTRESHn+2 ⁽⁵⁾ | EMAC interrupt |
| 77 | MACRXINTn+2 ⁽⁵⁾ | EMAC interrupt |
| 78 | GPINT4 | GPIO interrupt |
| 79 | GPINT5 | GPIO interrupt |
| 80 | GPINT6 | GPIO interrupt |
| 81 | GPINT7 | GPIO interrupt |

(9) CorePac[n] will receive TINTLn and TINTHn.

Table 8-33. C665x System Event Inputs — C66x CorePac Primary Interrupts (continued)

| INPUT EVENT NUMBER | INTERRUPT EVENT | DESCRIPTION |
|--------------------|-----------------------------------|---|
| 82 | GPINT8 | GPIO interrupt |
| 83 | GPINT9 | GPIO interrupt |
| 84 | GPINT10 | GPIO interrupt |
| 85 | GPINT11 | GPIO interrupt |
| 86 | GPINT12 | GPIO interrupt |
| 87 | GPINT13 | GPIO interrupt |
| 88 | GPINT14 | GPIO interrupt |
| 89 | GPINT15 | GPIO interrupt |
| 90 | IPC_LOCAL | Inter DSP interrupt from IPCGRn |
| 91 | GPINTn ⁽¹⁰⁾ | Local GPIO interrupt |
| 92 | CIC0_OUT(10+20*n) ⁽⁸⁾ | Interrupt Controller Output |
| 93 | CIC0_OUT(11+20*n) ⁽⁸⁾ | Interrupt Controller Output |
| 94 | MACTXINTn ⁽⁵⁾ | EMAC interrupt |
| 95 | MACTRESHn ⁽⁵⁾ | EMAC interrupt |
| 96 | INTERR | Dropped CPU interrupt event |
| 97 | EMC_IDMAERR | Invalid IDMA parameters |
| 98 | Reserved | |
| 99 | MACRXINTn ⁽⁵⁾ | EMAC interrupt |
| 100 | EFIINTA | EFI Interrupt from side A |
| 101 | EFIINTB | EFI Interrupt from side B |
| 102 | QM_INT_HIGH_(n+2) ⁽⁸⁾ | QM Interrupt for Queue 706+n ⁽⁸⁾ |
| 103 | QM_INT_HIGH_(n+6) ⁽⁸⁾ | QM Interrupt for Queue 710+n ⁽⁸⁾ |
| 104 | QM_INT_HIGH_(n+10) ⁽⁸⁾ | QM Interrupt for Queue 714+n ⁽⁸⁾ |
| 105 | QM_INT_HIGH_(n+14) ⁽⁸⁾ | QM Interrupt for Queue 718+n ⁽⁸⁾ |
| 106 | QM_INT_HIGH_(n+18) ⁽⁸⁾ | QM Interrupt for Queue 722+n ⁽⁸⁾ |
| 107 | QM_INT_HIGH_(n+22) ⁽⁸⁾ | QM Interrupt for Queue 726+n ⁽⁸⁾ |
| 108 | QM_INT_HIGH_(n+26) ⁽⁸⁾ | QM Interrupt for Queue 730+n ⁽⁸⁾ |
| 109 | QM_INT_HIGH_(n+30) ⁽⁸⁾ | QM Interrupt for Queue 734+n ⁽⁸⁾ |
| 110 | MDMAERREVT | VbusM error event |
| 111 | Reserved | |
| 112 | INTDST(n+18) ⁽¹¹⁾ | SRIO Interrupt |
| 113 | PMC_ED | Single bit error detected during DMA read |
| 114 | INTDST(n+22) ⁽¹²⁾ | SRIO Interrupt |
| 115 | EDMA3_CC_AET EVT | EDMA3 CC AET Event |
| 116 | UMC_ED1 | Corrected bit error detected |
| 117 | UMC_ED2 | Uncorrected bit error detected |
| 118 | PDC_INT | Power down sleep interrupt |
| 119 | SYS_CMPA | SYS CPU memory protection fault event |

(10) CorePac[n] will receive GPINTn.

(11) CorePac[n] will receive INTDST(n+18).

(12) CorePac[n] will receive INTDST(n+22).

Table 8-33. C665x System Event Inputs — C66x CorePac Primary Interrupts (continued)

| INPUT EVENT NUMBER | INTERRUPT EVENT | DESCRIPTION |
|--------------------|-----------------|---------------------------------------|
| 120 | PMC_CMPA | PMC CPU memory protection fault event |
| 121 | PMC_DMPA | PMC DMA memory protection fault event |
| 122 | DMC_CMPA | DMC CPU memory protection fault event |
| 123 | DMC_DMPA | DMC DMA memory protection fault event |
| 124 | UMC_CMPA | UMC CPU memory protection fault event |
| 125 | UMC_DMPA | UMC DMA memory protection fault event |
| 126 | EMC_CMPA | EMC CPU memory protection fault event |
| 127 | EMC_BUSERR | EMC bus error interrupt |

Table 8-34. CIC0 Event Inputs (Secondary Interrupts for C66x CorePacs)

| INPUT EVENT# ON CIC | SYSTEM INTERRUPT | DESCRIPTION |
|---------------------|------------------|--|
| 0 | GPINT16 | GPIO interrupt |
| 1 | GPINT17 | GPIO interrupt |
| 2 | GPINT18 | GPIO interrupt |
| 3 | GPINT19 | GPIO interrupt |
| 4 | GPINT20 | GPIO interrupt |
| 5 | GPINT21 | GPIO interrupt |
| 6 | GPINT22 | GPIO interrupt |
| 7 | GPINT23 | GPIO interrupt |
| 8 | GPINT24 | GPIO interrupt |
| 9 | GPINT25 | GPIO interrupt |
| 10 | GPINT26 | GPIO interrupt |
| 11 | GPINT27 | GPIO interrupt |
| 12 | GPINT28 | GPIO interrupt |
| 13 | GPINT29 | GPIO interrupt |
| 14 | GPINT30 | GPIO interrupt |
| 15 | GPINT31 | GPIO interrupt |
| 16 | EDMA3_CC_ERRINT | EDMA3_CC error interrupt |
| 17 | EDMA3_CC_MPINT | EDMA3_CC memory protection interrupt |
| 18 | EDMA3_TC_ERRINT0 | EDMA3_CC TC0 error interrupt |
| 19 | EDMA3_TC_ERRINT1 | EDMA3_CC TC1 error interrupt |
| 20 | EDMA3_TC_ERRINT2 | EDMA3_CC TC2 error interrupt |
| 21 | EDMA3_TC_ERRINT3 | EDMA3_CC TC3 error interrupt |
| 22 | EDMA3_CC_GINT | EDMA3_CC GINT |
| 23 | Reserved | |
| 24 | EDMA3_CC_INT0 | EDMA3_CC individual completion interrupt |
| 25 | EDMA3_CC_INT1 | EDMA3_CC individual completion interrupt |
| 26 | EDMA3_CC_INT2 | EDMA3_CC individual completion interrupt |
| 27 | EDMA3_CC_INT3 | EDMA3_CC individual completion interrupt |
| 28 | EDMA3_CC_INT4 | EDMA3_CC individual completion interrupt |
| 29 | EDMA3_CC_INT5 | EDMA3_CC individual completion interrupt |
| 30 | EDMA3_CC_INT6 | EDMA3_CC individual completion interrupt |
| 31 | EDMA3_CC_INT7 | EDMA3_CC individual completion interrupt |
| 32 | MCBSP0_RINT | McBSP0 interrupt |
| 33 | MCBSP0_XINT | McBSP0 interrupt |
| 34 | MCBSP0_REVT | McBSP0 interrupt |
| 35 | MCBSP0_XEVT | McBSP0 interrupt |
| 36 | MCBSP1_RINT | McBSP1 interrupt |
| 37 | MCBSP1_XINT | McBSP1 interrupt |
| 38 | MCBSP1_REVT | McBSP1 interrupt |
| 39 | MCBSP1_XEVT | McBSP1 interrupt |
| 40 | UARTINT_B | UART_1 interrupt |
| 41 | URXEVT_B | UART_1 interrupt |
| 42 | UTXEVT_B | UART_1 interrupt |
| 43 | Reserved | |
| 44 | Reserved | |
| 45 | Reserved | |
| 46 | Reserved | |

Table 8-34. CIC0 Event Inputs (Secondary Interrupts for C66x CorePacs) (continued)

| INPUT EVENT# ON CIC | SYSTEM INTERRUPT | DESCRIPTION |
|---------------------|--|---|
| 47 | Reserved | |
| 48 | PCIExpress_ERR_INT | Protocol error interrupt |
| 49 | PCIExpress_PM_INT | Power management interrupt |
| 50 | PCIExpress_Legacy_INTA | Legacy interrupt mode |
| 51 | PCIExpress_Legacy_INTB | Legacy interrupt mode |
| 52 | PCIExpress_Legacy_CIC | Legacy interrupt mode |
| 53 | PCIExpress_Legacy_INTD | Legacy interrupt mode |
| 54 | SPIINT0 | SPI interrupt0 |
| 55 | SPIINT1 | SPI interrupt1 |
| 56 | SPIXEVT | Transmit event |
| 57 | SPIREVT | Receive event |
| 58 | I2CINT | I ² C interrupt |
| 59 | I2CREVT | I ² C receive event |
| 60 | I2CXEVT | I ² C transmit event |
| 61 | Reserved | |
| 62 | Reserved | |
| 63 | TETBHFULLINT | TETB is half full |
| 64 | TETBFULLINT | TETB is full |
| 65 | TETBACQINT | Acquisition has been completed |
| 66 | TETBOVFLINT | Overflow condition occur |
| 67 | TETBUNFLINT | Underflow condition occur |
| 68 | SEMINT2 | Semaphore interrupt |
| 69 | SEMINT3 | Semaphore interrupt |
| 70 | SEMERR2 | Semaphore interrupt |
| 71 | SEMERR3 | Semaphore interrupt |
| 72 | Reserved | |
| 73 | Tracer_core_0_INTD | Tracer sliding time window interrupt for individual core |
| 74 | Tracer_core_1_INTD | Tracer sliding time window interrupt for individual core (C6657 only) |
| 75 | Reserved | |
| 76 | Reserved | |
| 77 | Tracer_DDR_INTD | Tracer sliding time window interrupt for DDR3 EMIF1 |
| 78 | Tracer_MSMC_0_INTD | Tracer sliding time window interrupt for MSMC SRAM bank0 |
| 79 | Tracer_MSMC_1_INTD | Tracer sliding time window interrupt for MSMC SRAM bank1 |
| 80 | Tracer_MSMC_2_INTD | Tracer sliding time window interrupt for MSMC SRAM bank2 |
| 81 | Tracer_MSMC_3_INTD | Tracer sliding time window interrupt for MSMC SRAM bank3 |
| 81 | Tracer_CFG_INTD | Tracer sliding time window interrupt for CFG0 TeraNet |
| 82 | Tracer_QM_CFG_INTD | Tracer sliding time window interrupt for QM_SS CFG |
| 84 | Tracer_QM_DMA_INTD | Tracer sliding time window interrupt for QM_SS slave |
| 85 | Tracer_SM_INTD | Tracer sliding time window interrupt for semaphore |
| 86 | PSC_ALLINT | Power/sleep controller interrupt |
| 87 | MSMC_scrub_cerror | Correctable (1-bit) soft error detected during scrub cycle |
| 88 | BOOTCFG_INTD | Chip-level MMR error register |
| 89 | po_vcon_smperr_intr | SmartReflex VolCon error status |
| 90 | MPU0_INTD (MPU0_ADDR_ERR_INT and MPU0_PROT_ERR_INT combined) | MPU0 addressing violation interrupt and protection violation interrupt. |
| 91 | Reserved | |

Table 8-34. CIC0 Event Inputs (Secondary Interrupts for C66x CorePacs) (continued)

| INPUT EVENT# ON CIC | SYSTEM INTERRUPT | DESCRIPTION |
|---------------------|--|---|
| 92 | MPU1_INTD (MPU1_ADDR_ERR_INT and MPU1_PROT_ERR_INT combined) | MPU1 addressing violation interrupt and protection violation interrupt. |
| 93 | Reserved | |
| 94 | MPU2_INTD (MPU2_ADDR_ERR_INT and MPU2_PROT_ERR_INT combined) | MPU2 addressing violation interrupt and protection violation interrupt. |
| 95 | Reserved | |
| 96 | MPU3_INTD (MPU3_ADDR_ERR_INT and MPU3_PROT_ERR_INT combined) | MPU3 addressing violation interrupt and protection violation interrupt. |
| 97 | Reserved | |
| 98 | MSMC_dedc_cerror | Correctable (1-bit) soft error detected on SRAM read |
| 99 | MSMC_dedc_nc_error | Non-correctable (2-bit) soft error detected on SRAM read |
| 100 | MSMC_scrub_nc_error | Non-correctable (2-bit) soft error detected during scrub cycle |
| 101 | Reserved | |
| 102 | MSMC_mpf_error8 | Memory protection fault indicators for each system master PrivID |
| 103 | MSMC_mpf_error9 | Memory protection fault indicators for each system master PrivID |
| 104 | MSMC_mpf_error10 | Memory protection fault indicators for each system master PrivID |
| 105 | MSMC_mpf_error11 | Memory protection fault indicators for each system master PrivID |
| 105 | MSMC_mpf_error12 | Memory protection fault indicators for each system master PrivID |
| 107 | MSMC_mpf_error13 | Memory protection fault indicators for each system master PrivID |
| 108 | MSMC_mpf_error14 | Memory protection fault indicators for each system master PrivID |
| 109 | MSMC_mpf_error15 | Memory protection fault indicators for each system master PrivID |
| 110 | DDR3_ERR | DDR3 EMIF error interrupt |
| 111 | HyperLink_int_o | HyperLink interrupt |
| 112 | INTDST0 | RapidIO interrupt |
| 113 | INTDST1 | RapidIO interrupt |
| 114 | INTDST2 | RapidIO interrupt |
| 115 | INTDST3 | RapidIO interrupt |
| 116 | INTDST4 | RapidIO interrupt |
| 117 | INTDST5 | RapidIO interrupt |
| 118 | INTDST6 | RapidIO interrupt |
| 119 | INTDST7 | RapidIO interrupt |
| 120 | INTDST8 | RapidIO interrupt |
| 121 | INTDST9 | RapidIO interrupt |
| 122 | INTDST10 | RapidIO interrupt |
| 123 | INTDST11 | RapidIO interrupt |
| 124 | INTDST12 | RapidIO interrupt |
| 125 | INTDST13 | RapidIO interrupt |
| 126 | INTDST14 | RapidIO interrupt |
| 127 | INTDST15 | RapidIO interrupt |
| 128 | Reserved | |
| 129 | Reserved | |
| 130 | po_vp_smpsack_intr | Indicating that Volt_Proc receives the r-edge at its smpsack input |
| 131 | Reserved | |
| 132 | Reserved | |
| 133 | Reserved | |
| 134 | QM_INT_PASS_TXQ_PEND_662 | Queue manager pend event |

Table 8-34. CIC0 Event Inputs (Secondary Interrupts for C66x CorePacs) (continued)

| INPUT EVENT# ON CIC | SYSTEM INTERRUPT | DESCRIPTION |
|---------------------|--|---|
| 135 | QM_INT_PASS_TXQ_PEND_663 | Queue manager pend event |
| 136 | QM_INT_PASS_TXQ_PEND_664 | Queue manager pend event |
| 137 | QM_INT_PASS_TXQ_PEND_665 | Queue manager pend event |
| 138 | QM_INT_PASS_TXQ_PEND_666 | Queue manager pend event |
| 139 | QM_INT_PASS_TXQ_PEND_667 | Queue manager pend event |
| 140 | QM_INT_PASS_TXQ_PEND_668 | Queue manager pend event |
| 141 | QM_INT_PASS_TXQ_PEND_669 | Queue manager pend event |
| 142 | QM_INT_PASS_TXQ_PEND_670 | Queue manager pend event |
| 143 | VCP0INT | VCP2_0 interrupt |
| 144 | VCP1INT | VCP2_1 interrupt |
| 145 | TINT4L | Timer4 interrupt low |
| 146 | TINT4H | Timer4 interrupt high |
| 147 | VCPAREVT | VCP2_A receive event |
| 148 | VCPAXEVT | VCP2_A transmit event |
| 149 | VCPBREVT | VCP2_B receive event |
| 150 | VCPBXEVT | VCP2_B transmit event |
| 151 | TINT5L | Timer5 interrupt low |
| 152 | TINT5H | Timer5 interrupt high |
| 153 | TINT6L | Timer6 interrupt low |
| 154 | TINT6H | Timer6 interrupt high |
| 155 | TCP_INTD | TCP3d interrupt |
| 156 | UPPINT | uPP interrupt |
| 157 | TCP_REVT0 | TCP3d interrupt |
| 158 | TCP_XEVT0 | TCP3d interrupt |
| 159 | Reserved | |
| 160 | MSMC_mpf_error2 | Memory protection fault indicators for each system master PrivID |
| 161 | MSMC_mpf_error3 | Memory protection fault indicators for each system master PrivID |
| 162 | TINT7L | Timer7 interrupt low |
| 163 | TINT7H | Timer7 interrupt high |
| 164 | UARTINT_A | UART_0 interrupt |
| 165 | URXEVT_A | UART_0 interrupt |
| 166 | UTXEVT_A | UART_0 interrupt |
| 167 | EASYNCERR | EMIF16 error interrupt |
| 168 | Tracer_EMIF16 | Tracer sliding time window interrupt for EMIF16 |
| 169 | Reserved | |
| 170 | MSMC_mpf_error4 | Memory protection fault indicators for each system master PrivID |
| 171 | MSMC_mpf_error5 | Memory protection fault indicators for each system master PrivID |
| 172 | MSMC_mpf_error6 | Memory protection fault indicators for each system master PrivID |
| 173 | MSMC_mpf_error7 | Memory protection fault indicators for each system master PrivID |
| 174 | MPU4_INTD (MPU4_ADDR_ERR_INT and MPU4_PROT_ERR_INT combined) | MPU4 addressing violation interrupt and protection violation interrupt. |
| 175 | QM_INT_PASS_TXQ_PEND_671 | Queue manager pend event |
| 176 | QM_INT_PKTDMMA_0 | QM interrupt for CDMA starvation |
| 177 | QM_INT_PKTDMMA_1 | QM interrupt for CDMA starvation |
| 178 | SRIO_INT_PKTDMMA_0 | SRIO interrupt for CDMA starvation |
| 179 | Reserved | |
| 180 | Reserved | |

Table 8-34. CIC0 Event Inputs (Secondary Interrupts for C66x CorePacs) (continued)

| INPUT EVENT# ON CIC | SYSTEM INTERRUPT | DESCRIPTION |
|---------------------|----------------------|--|
| 181 | SmartReflex_intrreq0 | SmartReflex sensor interrupt |
| 182 | SmartReflex_intrreq1 | SmartReflex sensor interrupt |
| 183 | SmartReflex_intrreq2 | SmartReflex sensor interrupt |
| 184 | SmartReflex_intrreq3 | SmartReflex sensor interrupt |
| 185 | VPNoSMPSAck | VPVOLTUPDATE has been asserted but SMPS has not been responded to in a defined time interval |
| 186 | VPEqValue | SRSINTERUPT is asserted, but the new voltage is not different from the current SMPS voltage |
| 187 | VPMaXVdd | The new voltage required is equal to or greater than MaxVdd. |
| 188 | VPMiNvdd | The new voltage required is equal to or less than MinVdd. |
| 189 | VPiNIDLE | Indicating that the FSM of voltage processor is in idle. |
| 190 | VPOPPChangeDone | Indicating that the average frequency error is within the desired limit. |
| 191 | Reserved | |
| 192 | MACINT4 | EMAC interrupt |
| 193 | MACRXINT4 | EMAC interrupt |
| 194 | MACTXINT4 | EMAC interrupt |
| 195 | MACTRESH4 | EMAC interrupt |
| 196 | MACINT5 | EMAC interrupt |
| 197 | MACRXINT5 | EMAC interrupt |
| 198 | MACTXINT5 | EMAC interrupt |
| 199 | MACTRESH5 | EMAC interrupt |
| 200 | MACINT6 | EMAC interrupt |
| 201 | MACRXINT6 | EMAC interrupt |
| 202 | MACTXINT6 | EMAC interrupt |
| 203 | MACTRESH6 | EMAC interrupt |
| 204 | MACINT7 | EMAC interrupt |
| 205 | MACRXINT7 | EMAC interrupt |
| 206 | MACTXINT7 | EMAC interrupt |
| 207 | MACTRESH7 | EMAC interrupt |

Table 8-35. CIC1 Event Inputs (Secondary Events for EDMA3_CC)

| INPUT EVENT # ON CIC | SYSTEM INTERRUPT | DESCRIPTION |
|-------------------------|--------------------|--|
| 0 | GPINT8 | GPIO interrupt |
| 1 | GPINT9 | GPIO interrupt |
| 2 | GPINT10 | GPIO interrupt |
| 3 | GPINT11 | GPIO interrupt |
| 4 | GPINT12 | GPIO interrupt |
| 5 | GPINT13 | GPIO interrupt |
| 6 | GPINT14 | GPIO interrupt |
| 7 | GPINT15 | GPIO interrupt |
| 8 | Reserved | |
| 9 | Reserved | |
| 10 | TETBACQINT | System TETB acquisition has been completed |
| 11 | Reserved | |
| 12 | Reserved | |
| 13 | TETBACQINT0 | TETB0 acquisition has been completed |
| 14 | Reserved | |
| 15 | Reserved | |
| 16 | TETBACQINT1 | TETB1 acquisition has been completed (C6657 only) |
| 17 | GPINT16 | GPIO interrupt |
| 18 | GPINT17 | GPIO interrupt |
| 19 | GPINT18 | GPIO interrupt |
| 20 | GPINT19 | GPIO interrupt |
| 21 | GPINT20 | GPIO interrupt |
| 22 | GPINT21 | GPIO interrupt |
| 23 | Reserved | |
| 24 | QM_INT_HIGH_16 | QM interrupt |
| 25 | QM_INT_HIGH_17 | QM interrupt |
| 26 | QM_INT_HIGH_18 | QM interrupt |
| 27 | QM_INT_HIGH_19 | QM interrupt |
| 28 | QM_INT_HIGH_20 | QM interrupt |
| 29 | QM_INT_HIGH_21 | QM interrupt |
| 30 | QM_INT_HIGH_22 | QM interrupt |
| 31 | QM_INT_HIGH_23 | QM interrupt |
| 32 | QM_INT_HIGH_24 | QM interrupt |
| 33 | QM_INT_HIGH_25 | QM interrupt |
| 34 | QM_INT_HIGH_26 | QM interrupt |
| 35 | QM_INT_HIGH_27 | QM interrupt |
| 36 | QM_INT_HIGH_28 | QM interrupt |
| 37 | QM_INT_HIGH_29 | QM interrupt |
| 38 | QM_INT_HIGH_30 | QM interrupt |
| 39 | QM_INT_HIGH_31 | QM interrupt |
| 40 | Reserved | |
| 41 | Reserved | |
| 42 | Reserved | |
| 43 | Reserved | |
| 44 | Reserved | |
| 45 | Tracer_core_0_INTD | Tracer sliding time window interrupt for individual core |

Table 8-35. CIC1 Event Inputs (Secondary Events for EDMA3_CC) (continued)

| INPUT EVENT # ON CIC | SYSTEM INTERRUPT | DESCRIPTION |
|----------------------|--|---|
| 46 | Tracer_core_1_INTD | Tracer sliding time window interrupt for individual core (C6657 only) |
| 47 | GPINT22 | GPIO interrupt |
| 48 | GPINT23 | GPIO interrupt |
| 49 | Tracer_DDR_INTD | Tracer sliding time window interrupt for DDR3 EMIF |
| 50 | Tracer_MSMC_0_INTD | Tracer sliding time window interrupt for MSMC SRAM bank0 |
| 51 | Tracer_MSMC_1_INTD | Tracer sliding time window interrupt for MSMC SRAM bank1 |
| 52 | Tracer_MSMC_2_INTD | Tracer sliding time window interrupt for MSMC SRAM bank2 |
| 53 | Tracer_MSMC_3_INTD | Tracer sliding time window interrupt for MSMC SRAM bank3 |
| 54 | Tracer_CFG_INTD | Tracer sliding time window interrupt for CFG0 TeraNet |
| 55 | Tracer_QM_CFG_INTD | Tracer sliding time window interrupt for QM_SS CFG |
| 56 | Tracer_QM_DMA_INTD | Tracer sliding time window interrupt for QM_SS slave port |
| 57 | Tracer_SEM_INTD | Tracer sliding time window interrupt for semaphore |
| 58 | SEMERR0 | Semaphore interrupt |
| 59 | SEMERR1 | Semaphore interrupt |
| 60 | SEMERR2 | Semaphore interrupt |
| 61 | SEMERR3 | Semaphore interrupt |
| 62 | BOOTCFG_INTD | BOOTCFG interrupt BOOTCFG_ERR and BOOTCFG_PROT |
| 63 | UPPINT | uPP interrupt |
| 64 | MPU0_INTD (MPU0_ADDR_ERR_INT and MPU0_PROT_ERR_INT combined) | MPU0 addressing violation interrupt and protection violation interrupt. |
| 65 | MSMC_scrub_cerror | Correctable (1-bit) soft error detected during scrub cycle |
| 66 | MPU1_INTD (MPU1_ADDR_ERR_INT and MPU1_PROT_ERR_INT combined) | MPU1 addressing violation interrupt and protection violation interrupt. |
| 67 | RapidIO_INT_PKTDMA_0 | RapidIO interrupt for packet DMA starvation |
| 68 | MPU2_INTD (MPU2_ADDR_ERR_INT and MPU2_PROT_ERR_INT combined) | MPU2 addressing violation interrupt and protection violation interrupt. |
| 69 | QM_INT_PKTDMA_0 | QM interrupt for packet DMA starvation |
| 70 | MPU3_INTD (MPU3_ADDR_ERR_INT and MPU3_PROT_ERR_INT combined) | MPU3 addressing violation interrupt and protection violation interrupt. |
| 71 | QM_INT_PKTDMA_1 | QM interrupt for packet DMA starvation |
| 72 | MSMC_dedc_cerror | Correctable (1-bit) soft error detected on SRAM read |
| 73 | MSMC_dedc_nc_error | Non-correctable (2-bit) soft error detected on SRAM read |
| 74 | MSMC_scrub_nc_error | Non-correctable (2-bit) soft error detected during scrub cycle |
| 75 | Reserved | |
| 76 | MSMC_mpf_error0 | Memory protection fault indicators for each system master PrivID |
| 77 | MSMC_mpf_error1 | Memory protection fault indicators for each system master PrivID |
| 78 | MSMC_mpf_error2 | Memory protection fault indicators for each system master PrivID |
| 79 | MSMC_mpf_error3 | Memory protection fault indicators for each system master PrivID |
| 80 | MSMC_mpf_error4 | Memory protection fault indicators for each system master PrivID |
| 81 | MSMC_mpf_error5 | Memory protection fault indicators for each system master PrivID |
| 82 | MSMC_mpf_error6 | Memory protection fault indicators for each system master PrivID |
| 83 | MSMC_mpf_error7 | Memory protection fault indicators for each system master PrivID |
| 84 | MSMC_mpf_error8 | Memory protection fault indicators for each system master PrivID |
| 85 | MSMC_mpf_error9 | Memory protection fault indicators for each system master PrivID |
| 86 | MSMC_mpf_error10 | Memory protection fault indicators for each system master PrivID |
| 87 | MSMC_mpf_error11 | Memory protection fault indicators for each system master PrivID |
| 88 | MSMC_mpf_error12 | Memory protection fault indicators for each system master PrivID |
| 89 | MSMC_mpf_error13 | Memory protection fault indicators for each system master PrivID |

Table 8-35. CIC1 Event Inputs (Secondary Events for EDMA3_CC) (continued)

| INPUT EVENT # ON CIC | SYSTEM INTERRUPT | DESCRIPTION |
|----------------------|--|---|
| 90 | MSMC_mpf_error14 | Memory protection fault indicators for each system master PrivID |
| 91 | MSMC_mpf_error15 | Memory protection fault indicators for each system master PrivID |
| 92 | Reserved | |
| 93 | INTDST0 | RapidIO interrupt |
| 94 | INTDST1 | RapidIO interrupt |
| 95 | INTDST2 | RapidIO interrupt |
| 96 | INTDST3 | RapidIO interrupt |
| 97 | INTDST4 | RapidIO interrupt |
| 98 | INTDST5 | RapidIO interrupt |
| 99 | INTDST6 | RapidIO interrupt |
| 100 | INTDST7 | RapidIO interrupt |
| 101 | INTDST8 | RapidIO interrupt |
| 102 | INTDST9 | RapidIO interrupt |
| 103 | INTDST10 | RapidIO interrupt |
| 104 | INTDST11 | RapidIO interrupt |
| 105 | INTDST12 | RapidIO interrupt |
| 106 | INTDST13 | RapidIO interrupt |
| 107 | INTDST14 | RapidIO interrupt |
| 108 | INTDST15 | RapidIO interrupt |
| 109 | INTDST16 | RapidIO interrupt |
| 110 | INTDST17 | RapidIO interrupt |
| 111 | INTDST18 | RapidIO interrupt |
| 112 | INTDST19 | RapidIO interrupt |
| 113 | INTDST20 | RapidIO interrupt |
| 114 | INTDST21 | RapidIO interrupt |
| 115 | INTDST22 | RapidIO interrupt |
| 116 | INTDST23 | RapidIO interrupt |
| 117 | GPINT24 | GPIO interrupt |
| 118 | GPINT25 | GPIO interrupt |
| 119 | VCPAINT | VCP2_A interrupt |
| 120 | VCPBINT | VCP2_B interrupt |
| 121 | GPINT26 | GPIO interrupt |
| 122 | GPINT27 | GPIO interrupt |
| 123 | TCP3D_INTD | Error interrupt TCP3DINT0 and TCP3DINT1 |
| 124 | GPINT28 | GPIO interrupt |
| 125 | GPINT29 | GPIO interrupt |
| 126 | GPINT30 | GPIO interrupt |
| 127 | GPINT31 | GPIO interrupt |
| 128 | GPINT4 | GPIO interrupt |
| 129 | GPINT5 | GPIO interrupt |
| 130 | GPINT6 | GPIO interrupt |
| 131 | GPINT7 | GPIO interrupt |
| 132 | Hyperlink_int_o | Hyperlink interrupt |
| 133 | Tracer_EMIF16 | Tracer sliding time window interrupt for EMIF16 |
| 134 | EASyncERR | EMIF16 error interrupt |
| 135 | MPU4_INTD (MPU4_ADDR_ERR_INT and MPU4_PROT_ERR_INT combined) | MPU4 addressing violation interrupt and protection violation interrupt. |

Table 8-35. CIC1 Event Inputs (Secondary Events for EDMA3_CC) (continued)

| INPUT EVENT # ON CIC | SYSTEM INTERRUPT | DESCRIPTION |
|-------------------------|------------------|----------------------|
| 136 | Reserved | |
| 137 | QM_INT_HIGH_0 | QM interrupt |
| 138 | QM_INT_HIGH_1 | QM interrupt |
| 139 | QM_INT_HIGH_2 | QM interrupt |
| 140 | QM_INT_HIGH_3 | QM interrupt |
| 141 | QM_INT_HIGH_4 | QM interrupt |
| 142 | QM_INT_HIGH_5 | QM interrupt |
| 143 | QM_INT_HIGH_6 | QM interrupt |
| 144 | QM_INT_HIGH_7 | QM interrupt |
| 145 | QM_INT_HIGH_8 | QM interrupt |
| 146 | QM_INT_HIGH_9 | QM interrupt |
| 147 | QM_INT_HIGH_10 | QM interrupt |
| 148 | QM_INT_HIGH_11 | QM interrupt |
| 149 | QM_INT_HIGH_12 | QM interrupt |
| 150 | QM_INT_HIGH_13 | QM interrupt |
| 151 | QM_INT_HIGH_14 | QM interrupt |
| 152 | QM_INT_HIGH_15 | QM interrupt |
| 153 | Reserved | |
| 154 | Reserved | |
| 155 | Reserved | |
| 156 | Reserved | |
| 157 | Reserved | |
| 158 | Reserved | |
| 159 | DDR3_ERR | DDR3 error interrupt |

Table 8-36. CIC2 Event Inputs (Secondary Events for HyperLink)

| INPUT EVENT # ON CIC | SYSTEM INTERRUPT | DESCRIPTION |
|----------------------|-----------------------|---|
| 0 | GPINT0 | GPIO interrupt |
| 1 | GPINT1 | GPIO interrupt |
| 2 | GPINT2 | GPIO interrupt |
| 3 | GPINT3 | GPIO interrupt |
| 4 | GPINT4 | GPIO interrupt |
| 5 | GPINT5 | GPIO interrupt |
| 6 | GPINT6 | GPIO interrupt |
| 7 | GPINT7 | GPIO interrupt |
| 8 | GPINT8 | GPIO interrupt |
| 9 | GPINT9 | GPIO interrupt |
| 10 | GPINT10 | GPIO interrupt |
| 11 | GPINT11 | GPIO interrupt |
| 12 | GPINT12 | GPIO interrupt |
| 13 | GPINT13 | GPIO interrupt |
| 14 | GPINT14 | GPIO interrupt |
| 15 | GPINT15 | GPIO interrupt |
| 16 | TETBHFULLINT | System TETB is half full |
| 17 | TETBFULLINT | System TETB is full |
| 18 | TETBACQINT | System TETB acquisition has been completed |
| 19 | TETBHFULLINT0 | TETB0 is half full |
| 20 | TETBFULLINT0 | TETB0 is full |
| 21 | TETBACQINT0 | TETB0 acquisition has been completed |
| 22 | TETBHFULLINT1 | TETB1 is half full |
| 23 | TETBFULLINT1 | TETB1 is full |
| 24 | TETBACQINT1 | TETB1 acquisition has been completed |
| 25 | GPINT16 | GPIO interrupt |
| 26 | GPINT17 | GPIO interrupt |
| 27 | GPINT18 | GPIO interrupt |
| 28 | GPINT19 | GPIO interrupt |
| 29 | GPINT20 | GPIO interrupt |
| 30 | GPINT21 | GPIO interrupt |
| 31 | Tracer_core_0_INTD | Tracer sliding time window interrupt for individual core |
| 32 | Tracer_core_1_INTD | Tracer sliding time window interrupt for individual core (C6657 only) |
| 33 | GPINT22 | GPIO interrupt |
| 34 | GPINT23 | GPIO interrupt |
| 35 | Tracer_DDR_INTD | Tracer sliding time window interrupt for DDR3 EMIF1 |
| 36 | Tracer_MSMC_0_INTD | Tracer sliding time window interrupt for MSMC SRAM bank0 |
| 37 | Tracer_MSMC_1_INTD | Tracer sliding time window interrupt for MSMC SRAM bank1 |
| 38 | Tracer_MSMC_2_INTD | Tracer sliding time window interrupt for MSMC SRAM bank2 |
| 39 | Tracer_MSMC_3_INTD | Tracer sliding time window interrupt for MSMC SRAM bank3 |
| 40 | Tracer_CFG_INTD | Tracer sliding time window interrupt for CFG0 TeraNet |
| 41 | Tracer_QM_SS_CFG_INTD | Tracer sliding time window interrupt for QM_SS CFG |
| 42 | Tracer_QM_SS_DMA_INTD | Tracer sliding time window interrupt for QM_SS slave port |
| 43 | Tracer_SEM_INTD | Tracer sliding time window interrupt for semaphore |
| 44 | Reserved | |
| 45 | GPINT24 | GPIO interrupt |

Table 8-36. CIC2 Event Inputs (Secondary Events for HyperLink) (continued)

| INPUT EVENT # ON CIC | SYSTEM INTERRUPT | DESCRIPTION |
|-------------------------|------------------|--|
| 46 | GPINT25 | GPIO interrupt |
| 47 | GPINT26 | GPIO interrupt |
| 48 | GPINT27 | GPIO interrupt |
| 49 | TINT4L | Timer64_4 interrupt low |
| 50 | TINT4H | Timer64_4 interrupt high |
| 51 | TINT5L | Timer64_5 interrupt low |
| 52 | TINT5H | timer64_5 interrupt high |
| 53 | TINT6L | Timer64_6 interrupt low |
| 54 | TINT6H | Timer64_6 interrupt high |
| 55 | TINT7L | Timer64_7 interrupt low |
| 56 | TINT7H | Timer64_7 interrupt high |
| 57 | Reserved | |
| 58 | Reserved | |
| 59 | Reserved | |
| 60 | Tracer_EMIF16 | Tracer sliding time window interrupt for TNet_6P_A |
| 61 | DDR3_ERR | DDR3 EMIF Error interrupt |
| 62 | Reserved | |
| 63 | EASYNCERR | EMIF16 error interrupt |
| 64 | GPINT28 | GPIO interrupt |
| 65 | GPINT29 | GPIO interrupt |
| 66 | GPINT30 | GPIO interrupt |
| 67 | GPINT31 | GPIO interrupt |
| 68 | TINT2L | Timer2 interrupt low |
| 69 | TINT2H | Timer2 interrupt high |
| 70 | TINT3L | Timer2 interrupt low |
| 71 | TINT3H | Timer2 interrupt high |
| 72-79 | Reserved | |

8.8.2 CIC Registers

This section includes the offsets for CIC registers. The base addresses for interrupt control registers are CIC0 - 0x0260 0000, CIC1 - 0x0260 4000, and CIC2 - 0x0260 8000.

8.8.2.1 CIC0 Register Map

Table 8-37. CIC0 Register

| ADDRESS OFFSET | REGISTER MNEMONIC | REGISTER NAME |
|----------------|---------------------------|---|
| 0x0 | REVISION_REG | Revision Register |
| 0x4 | CONTROL_REG | Control Register |
| 0xc | HOST_CONTROL_REG | Host Control Register |
| 0x10 | GLOBAL_ENABLE_HINT_REG | Global Host Int Enable Register |
| 0x20 | STATUS_SET_INDEX_REG | Status Set Index Register |
| 0x24 | STATUS_CLR_INDEX_REG | Status Clear Index Register |
| 0x28 | ENABLE_SET_INDEX_REG | Enable Set Index Register |
| 0x2c | ENABLE_CLR_INDEX_REG | Enable Clear Index Register |
| 0x34 | HINT_ENABLE_SET_INDEX_REG | Host Int Enable Set Index Register |
| 0x38 | HINT_ENABLE_CLR_INDEX_REG | Host Int Enable Clear Index Register |
| 0x200 | RAW_STATUS_REG0 | Raw Status Register 0 |
| 0x204 | RAW_STATUS_REG1 | Raw Status Register 1 |
| 0x208 | RAW_STATUS_REG2 | Raw Status Register 2 |
| 0x20c | RAW_STATUS_REG3 | Raw Status Register 3 |
| 0x210 | RAW_STATUS_REG4 | Raw Status Register 4 |
| 0x214 | RAW_STATUS_REG5 | Raw Status Register 5 |
| 0x218 | RAW_STATUS_REG6 | Raw Status Register 6 |
| 0x280 | ENA_STATUS_REG0 | Enabled Status Register 0 |
| 0x284 | ENA_STATUS_REG1 | Enabled Status Register 1 |
| 0x288 | ENA_STATUS_REG2 | Enabled Status Register 2 |
| 0x28c | ENA_STATUS_REG3 | Enabled Status Register 3 |
| 0x290 | ENA_STATUS_REG4 | Enabled Status Register 4 |
| 0x294 | ENA_STATUS_REG5 | Enabled Status Register 5 |
| 0x298 | ENA_STATUS_REG6 | Enabled Status Register 6 |
| 0x300 | ENABLE_REG0 | Enable Register 0 |
| 0x304 | ENABLE_REG1 | Enable Register 1 |
| 0x308 | ENABLE_REG2 | Enable Register 2 |
| 0x30c | ENABLE_REG3 | Enable Register 3 |
| 0x310 | ENABLE_REG4 | Enable Register 4 |
| 0x314 | ENABLE_REG5 | Enable Register 5 |
| 0x318 | ENABLE_REG6 | Enable Register 6 |
| 0x380 | ENABLE_CLR_REG0 | Enable Clear Register 0 |
| 0x384 | ENABLE_CLR_REG1 | Enable Clear Register 1 |
| 0x388 | ENABLE_CLR_REG2 | Enable Clear Register 2 |
| 0x38c | ENABLE_CLR_REG3 | Enable Clear Register 3 |
| 0x390 | ENABLE_CLR_REG4 | Enable Clear Register 4 |
| 0x394 | ENABLE_CLR_REG5 | Enable Clear Register 5 |
| 0x398 | ENABLE_CLR_REG6 | Enable Clear Register 6 |
| 0x400 | CH_MAP_REG0 | Interrupt Channel Map Register for 0 to 0+3 |
| 0x404 | CH_MAP_REG1 | Interrupt Channel Map Register for 4 to 4+3 |
| 0x408 | CH_MAP_REG2 | Interrupt Channel Map Register for 8 to 8+3 |

Table 8-37. CIC0 Register (continued)

| ADDRESS OFFSET | REGISTER MNEMONIC | REGISTER NAME |
|----------------|-------------------|---|
| 0x40c | CH_MAP_REG3 | Interrupt Channel Map Register for 12 to 12+3 |
| 0x410 | CH_MAP_REG4 | Interrupt Channel Map Register for 16 to 16+3 |
| 0x414 | CH_MAP_REG5 | Interrupt Channel Map Register for 20 to 20+3 |
| 0x418 | CH_MAP_REG6 | Interrupt Channel Map Register for 24 to 24+3 |
| 0x41c | CH_MAP_REG7 | Interrupt Channel Map Register for 28 to 28+3 |
| 0x420 | CH_MAP_REG8 | Interrupt Channel Map Register for 32 to 32+3 |
| 0x424 | CH_MAP_REG9 | Interrupt Channel Map Register for 36 to 36+3 |
| 0x428 | CH_MAP_REG10 | Interrupt Channel Map Register for 40 to 40+3 |
| 0x42c | CH_MAP_REG11 | Interrupt Channel Map Register for 44 to 44+3 |
| 0x430 | CH_MAP_REG12 | Interrupt Channel Map Register for 48 to 48+3 |
| 0x434 | CH_MAP_REG13 | Interrupt Channel Map Register for 52 to 52+3 |
| 0x438 | CH_MAP_REG14 | Interrupt Channel Map Register for 56 to 56+3 |
| 0x43c | CH_MAP_REG15 | Interrupt Channel Map Register for 60 to 60+3 |
| 0x440 | CH_MAP_REG16 | Interrupt Channel Map Register for 64 to 64+3 |
| 0x444 | CH_MAP_REG17 | Interrupt Channel Map Register for 68 to 68+3 |
| 0x448 | CH_MAP_REG18 | Interrupt Channel Map Register for 72 to 72+3 |
| 0x44c | CH_MAP_REG19 | Interrupt Channel Map Register for 76 to 76+3 |
| 0x450 | CH_MAP_REG20 | Interrupt Channel Map Register for 80 to 80+3 |
| 0x454 | CH_MAP_REG21 | Interrupt Channel Map Register for 84 to 84+3 |
| 0x458 | CH_MAP_REG22 | Interrupt Channel Map Register for 88 to 88+3 |
| 0x45c | CH_MAP_REG23 | Interrupt Channel Map Register for 92 to 92+3 |
| 0x460 | CH_MAP_REG24 | Interrupt Channel Map Register for 96 to 96+3 |
| 0x464 | CH_MAP_REG25 | Interrupt Channel Map Register for 100 to 100+3 |
| 0x468 | CH_MAP_REG26 | Interrupt Channel Map Register for 104 to 104+3 |
| 0x46c | CH_MAP_REG27 | Interrupt Channel Map Register for 108 to 108+3 |
| 0x470 | CH_MAP_REG28 | Interrupt Channel Map Register for 112 to 112+3 |
| 0x474 | CH_MAP_REG29 | Interrupt Channel Map Register for 116 to 116+3 |
| 0x478 | CH_MAP_REG30 | Interrupt Channel Map Register for 120 to 120+3 |
| 0x47c | CH_MAP_REG31 | Interrupt Channel Map Register for 124 to 124+3 |
| 0x480 | CH_MAP_REG32 | Interrupt Channel Map Register for 128 to 128+3 |
| 0x484 | CH_MAP_REG33 | Interrupt Channel Map Register for 132 to 132+3 |
| 0x488 | CH_MAP_REG34 | Interrupt Channel Map Register for 136 to 136+3 |
| 0x48c | CH_MAP_REG35 | Interrupt Channel Map Register for 140 to 140+3 |
| 0x490 | CH_MAP_REG36 | Interrupt Channel Map Register for 144 to 144+3 |
| 0x494 | CH_MAP_REG37 | Interrupt Channel Map Register for 148 to 148+3 |
| 0x498 | CH_MAP_REG38 | Interrupt Channel Map Register for 152 to 152+3 |
| 0x49c | CH_MAP_REG39 | Interrupt Channel Map Register for 156 to 156+3 |
| 0x4a0 | CH_MAP_REG40 | Interrupt Channel Map Register for 160 to 160+3 |
| 0x4a4 | CH_MAP_REG41 | Interrupt Channel Map Register for 164 to 164+3 |
| 0x4a8 | CH_MAP_REG42 | Interrupt Channel Map Register for 168 to 168+3 |
| 0x4ac | CH_MAP_REG43 | Interrupt Channel Map Register for 172 to 172+3 |
| 0x4b0 | CH_MAP_REG44 | Interrupt Channel Map Register for 176 to 176+3 |
| 0x4b4 | CH_MAP_REG45 | Interrupt Channel Map Register for 180 to 180+3 |
| 0x4b8 | CH_MAP_REG46 | Interrupt Channel Map Register for 184 to 184+3 |
| 0x4bc | CH_MAP_REG47 | Interrupt Channel Map Register for 188 to 188+3 |
| 0x4c0 | CH_MAP_REG48 | Interrupt Channel Map Register for 192 to 192+3 |
| 0x4c4 | CH_MAP_REG49 | Interrupt Channel Map Register for 196 to 196+3 |

Table 8-37. CIC0 Register (continued)

| ADDRESS OFFSET | REGISTER MNEMONIC | REGISTER NAME |
|----------------|-------------------|---|
| 0x4c8 | CH_MAP_REG50 | Interrupt Channel Map Register for 200 to 200+3 |
| 0x4cc | CH_MAP_REG51 | Interrupt Channel Map Register for 204 to 204+3 |
| 0x800 | HINT_MAP_REG0 | Host Interrupt Map Register for 0 to 0+3 |
| 0x804 | HINT_MAP_REG1 | Host Interrupt Map Register for 4 to 4+3 |
| 0x808 | HINT_MAP_REG2 | Host Interrupt Map Register for 8 to 8+3 |
| 0x80c | HINT_MAP_REG3 | Host Interrupt Map Register for 12 to 12+3 |
| 0x810 | HINT_MAP_REG4 | Host Interrupt Map Register for 16 to 16+3 |
| 0x814 | HINT_MAP_REG5 | Host Interrupt Map Register for 20 to 20+3 |
| 0x818 | HINT_MAP_REG6 | Host Interrupt Map Register for 24 to 24+3 |
| 0x81c | HINT_MAP_REG7 | Host Interrupt Map Register for 28 to 28+3 |
| 0x820 | HINT_MAP_REG8 | Host Interrupt Map Register for 32 to 32+3 |
| 0x824 | HINT_MAP_REG9 | Host Interrupt Map Register for 36 to 36+3 |
| 0x828 | HINT_MAP_REG10 | Host Interrupt Map Register for 40 to 40+3 |
| 0x82c | HINT_MAP_REG11 | Host Interrupt Map Register for 44 to 44+3 |
| 0x830 | HINT_MAP_REG12 | Host Interrupt Map Register for 48 to 48+3 |
| 0x834 | HINT_MAP_REG13 | Host Interrupt Map Register for 52 to 52+3 |
| 0x838 | HINT_MAP_REG14 | Host Interrupt Map Register for 56 to 56+3 |
| 0x83c | HINT_MAP_REG15 | Host Interrupt Map Register for 60 to 60+3 |
| 0x840 | HINT_MAP_REG16 | Host Interrupt Map Register for 64 to 64+3 |
| 0x844 | HINT_MAP_REG17 | Host Interrupt Map Register for 68 to 68+3 |
| 0x848 | HINT_MAP_REG18 | Host Interrupt Map Register for 72 to 72+3 |
| 0x84c | HINT_MAP_REG19 | Host Interrupt Map Register for 76 to 76+3 |
| 0x850 | HINT_MAP_REG20 | Host Interrupt Map Register for 80 to 80+3 |
| 0x854 | HINT_MAP_REG21 | Host Interrupt Map Register for 84 to 84+3 |
| 0x858 | HINT_MAP_REG22 | Host Interrupt Map Register for 88 to 88+3 |
| 0x860 | HINT_MAP_REG23 | Host Interrupt Map Register for 92 to 92+3 |
| 0x1500 | ENABLE_HINT_REG0 | Host Int Enable Register 0 |
| 0x1504 | ENABLE_HINT_REG1 | Host Int Enable Register 1 |
| 0x1508 | ENABLE_HINT_REG2 | Host Int Enable Register 2 |

8.8.2.2 CIC1 Register Map

Table 8-38. CIC1 Register

| ADDRESS OFFSET | REGISTER MNEMONIC | REGISTER NAME |
|----------------|---------------------------|---|
| 0x0 | REVISION_REG | Revision Register |
| 0x10 | GLOBAL_ENABLE_HINT_REG | Global Host Int Enable Register |
| 0x20 | STATUS_SET_INDEX_REG | Status Set Index Register |
| 0x24 | STATUS_CLR_INDEX_REG | Status Clear Index Register |
| 0x28 | ENABLE_SET_INDEX_REG | Enable Set Index Register |
| 0x2c | ENABLE_CLR_INDEX_REG | Enable Clear Index Register |
| 0x34 | HINT_ENABLE_SET_INDEX_REG | Host Int Enable Set Index Register |
| 0x38 | HINT_ENABLE_CLR_INDEX_REG | Host Int Enable Clear Index Register |
| 0x200 | RAW_STATUS_REG0 | Raw Status Register 0 |
| 0x204 | RAW_STATUS_REG1 | Raw Status Register 1 |
| 0x208 | RAW_STATUS_REG2 | Raw Status Register 2 |
| 0x20c | RAW_STATUS_REG3 | Raw Status Register 3 |
| 0x210 | RAW_STATUS_REG4 | Raw Status Register 4 |
| 0x280 | ENA_STATUS_REG0 | Enabled Status Register 0 |
| 0x284 | ENA_STATUS_REG1 | Enabled Status Register 1 |
| 0x288 | ENA_STATUS_REG2 | Enabled Status Register 2 |
| 0x28c | ENA_STATUS_REG3 | Enabled Status Register 3 |
| 0x290 | ENA_STATUS_REG4 | Enabled Status Register 4 |
| 0x300 | ENABLE_REG0 | Enable Register 0 |
| 0x304 | ENABLE_REG1 | Enable Register 1 |
| 0x308 | ENABLE_REG2 | Enable Register 2 |
| 0x30c | ENABLE_REG3 | Enable Register 3 |
| 0x310 | ENABLE_REG4 | Enable Register 4 |
| 0x380 | ENABLE_CLR_REG0 | Enable Clear Register 0 |
| 0x384 | ENABLE_CLR_REG1 | Enable Clear Register 1 |
| 0x388 | ENABLE_CLR_REG2 | Enable Clear Register 2 |
| 0x38c | ENABLE_CLR_REG3 | Enable Clear Register 3 |
| 0x390 | ENABLE_CLR_REG4 | Enable Clear Register 4 |
| 0x400 | CH_MAP_REG0 | Interrupt Channel Map Register for 0 to 0+3 |
| 0x404 | CH_MAP_REG1 | Interrupt Channel Map Register for 4 to 4+3 |
| 0x408 | CH_MAP_REG2 | Interrupt Channel Map Register for 8 to 8+3 |
| 0x40c | CH_MAP_REG3 | Interrupt Channel Map Register for 12 to 12+3 |
| 0x410 | CH_MAP_REG4 | Interrupt Channel Map Register for 16 to 16+3 |
| 0x414 | CH_MAP_REG5 | Interrupt Channel Map Register for 20 to 20+3 |
| 0x418 | CH_MAP_REG6 | Interrupt Channel Map Register for 24 to 24+3 |
| 0x41c | CH_MAP_REG7 | Interrupt Channel Map Register for 28 to 28+3 |
| 0x420 | CH_MAP_REG8 | Interrupt Channel Map Register for 32 to 32+3 |
| 0x424 | CH_MAP_REG9 | Interrupt Channel Map Register for 36 to 36+3 |
| 0x428 | CH_MAP_REG10 | Interrupt Channel Map Register for 40 to 40+3 |
| 0x42c | CH_MAP_REG11 | Interrupt Channel Map Register for 44 to 44+3 |
| 0x430 | CH_MAP_REG12 | Interrupt Channel Map Register for 48 to 48+3 |
| 0x434 | CH_MAP_REG13 | Interrupt Channel Map Register for 52 to 52+3 |
| 0x438 | CH_MAP_REG14 | Interrupt Channel Map Register for 56 to 56+3 |
| 0x43c | CH_MAP_REG15 | Interrupt Channel Map Register for 60 to 60+3 |
| 0x440 | CH_MAP_REG16 | Interrupt Channel Map Register for 64 to 64+3 |

Table 8-38. CIC1 Register (continued)

| ADDRESS OFFSET | REGISTER MNEMONIC | REGISTER NAME |
|----------------|-------------------|---|
| 0x444 | CH_MAP_REG17 | Interrupt Channel Map Register for 68 to 68+3 |
| 0x448 | CH_MAP_REG18 | Interrupt Channel Map Register for 72 to 72+3 |
| 0x44c | CH_MAP_REG19 | Interrupt Channel Map Register for 76 to 76+3 |
| 0x450 | CH_MAP_REG20 | Interrupt Channel Map Register for 80 to 80+3 |
| 0x454 | CH_MAP_REG21 | Interrupt Channel Map Register for 84 to 84+3 |
| 0x458 | CH_MAP_REG22 | Interrupt Channel Map Register for 88 to 88+3 |
| 0x45c | CH_MAP_REG23 | Interrupt Channel Map Register for 92 to 92+3 |
| 0x460 | CH_MAP_REG24 | Interrupt Channel Map Register for 96 to 96+3 |
| 0x464 | CH_MAP_REG25 | Interrupt Channel Map Register for 100 to 100+3 |
| 0x468 | CH_MAP_REG26 | Interrupt Channel Map Register for 104 to 104+3 |
| 0x46c | CH_MAP_REG27 | Interrupt Channel Map Register for 108 to 108+3 |
| 0x470 | CH_MAP_REG28 | Interrupt Channel Map Register for 112 to 112+3 |
| 0x474 | CH_MAP_REG29 | Interrupt Channel Map Register for 116 to 116+3 |
| 0x478 | CH_MAP_REG30 | Interrupt Channel Map Register for 120 to 120+3 |
| 0x47c | CH_MAP_REG31 | Interrupt Channel Map Register for 124 to 124+3 |
| 0x480 | CH_MAP_REG32 | Interrupt Channel Map Register for 128 to 128+3 |
| 0x484 | CH_MAP_REG33 | Interrupt Channel Map Register for 132 to 132+3 |
| 0x488 | CH_MAP_REG34 | Interrupt Channel Map Register for 136 to 136+3 |
| 0x48c | CH_MAP_REG35 | Interrupt Channel Map Register for 140 to 140+3 |
| 0x490 | CH_MAP_REG36 | Interrupt Channel Map Register for 144 to 144+3 |
| 0x494 | CH_MAP_REG37 | Interrupt Channel Map Register for 148 to 148+3 |
| 0x498 | CH_MAP_REG38 | Interrupt Channel Map Register for 152 to 152+3 |
| 0x49c | CH_MAP_REG39 | Interrupt Channel Map Register for 156 to 156+3 |
| 0x800 | HINT_MAP_REG0 | Host Interrupt Map Register for 0 to 0+3 |
| 0x804 | HINT_MAP_REG1 | Host Interrupt Map Register for 4 to 4+3 |
| 0x808 | HINT_MAP_REG2 | Host Interrupt Map Register for 8 to 8+3 |
| 0x80c | HINT_MAP_REG3 | Host Interrupt Map Register for 12 to 12+3 |
| 0x810 | HINT_MAP_REG4 | Host Interrupt Map Register for 16 to 16+3 |
| 0x814 | HINT_MAP_REG5 | Host Interrupt Map Register for 20 to 20+3 |
| 0x818 | HINT_MAP_REG6 | Host Interrupt Map Register for 24 to 24+3 |
| 0x81c | HINT_MAP_REG7 | Host Interrupt Map Register for 28 to 28+3 |
| 0x820 | HINT_MAP_REG8 | Host Interrupt Map Register for 32 to 32+3 |
| 0x824 | HINT_MAP_REG9 | Host Interrupt Map Register for 36 to 36+3 |
| 0x828 | HINT_MAP_REG10 | Host Interrupt Map Register for 40 to 40+3 |
| 0x82c | HINT_MAP_REG11 | Host Interrupt Map Register for 44 to 44+3 |
| 0x830 | HINT_MAP_REG12 | Host Interrupt Map Register for 48 to 48+3 |
| 0x834 | HINT_MAP_REG13 | Host Interrupt Map Register for 52 to 52+3 |
| 0x838 | HINT_MAP_REG14 | Host Interrupt Map Register for 56 to 56+3 |
| 0x83c | HINT_MAP_REG15 | Host Interrupt Map Register for 60 to 60+3 |
| 0x1500 | ENABLE_HINT_REG0 | Host Int Enable Register 0 |
| 0x1504 | ENABLE_HINT_REG1 | Host Int Enable Register 1 |

8.8.2.3 CIC2 Register Map

Table 8-39. CIC2 Register

| ADDRESS OFFSET | REGISTER MNEMONIC | REGISTER NAME |
|----------------|---------------------------|---|
| 0x0 | REVISION_REG | Revision Register |
| 0x10 | GLOBAL_ENABLE_HINT_REG | Global Host Int Enable Register |
| 0x20 | STATUS_SET_INDEX_REG | Status Set Index Register |
| 0x24 | STATUS_CLR_INDEX_REG | Status Clear Index Register |
| 0x28 | ENABLE_SET_INDEX_REG | Enable Set Index Register |
| 0x2c | ENABLE_CLR_INDEX_REG | Enable Clear Index Register |
| 0x34 | HINT_ENABLE_SET_INDEX_REG | Host Int Enable Set Index Register |
| 0x38 | HINT_ENABLE_CLR_INDEX_REG | Host Int Enable Clear Index Register |
| 0x200 | RAW_STATUS_REG0 | Raw Status Register 0 |
| 0x204 | RAW_STATUS_REG1 | Raw Status Register 1 |
| 0x208 | RAW_STATUS_REG2 | Raw Status Register 2 |
| 0x280 | ENA_STATUS_REG0 | Enabled Status Register 0 |
| 0x284 | ENA_STATUS_REG1 | Enabled Status Register 1 |
| 0x288 | ENA_STATUS_REG2 | Enabled Status Register 2 |
| 0x300 | ENABLE_REG0 | Enable Register 0 |
| 0x304 | ENABLE_REG1 | Enable Register 1 |
| 0x308 | ENABLE_REG2 | Enable Register 2 |
| 0x380 | ENABLE_CLR_REG0 | Enable Clear Register 0 |
| 0x384 | ENABLE_CLR_REG1 | Enable Clear Register 1 |
| 0x388 | ENABLE_CLR_REG2 | Enable Clear Register 2 |
| 0x400 | CH_MAP_REG0 | Interrupt Channel Map Register for 0 to 0+3 |
| 0x404 | CH_MAP_REG1 | Interrupt Channel Map Register for 4 to 4+3 |
| 0x408 | CH_MAP_REG2 | Interrupt Channel Map Register for 8 to 8+3 |
| 0x40c | CH_MAP_REG3 | Interrupt Channel Map Register for 12 to 12+3 |
| 0x410 | CH_MAP_REG4 | Interrupt Channel Map Register for 16 to 16+3 |
| 0x414 | CH_MAP_REG5 | Interrupt Channel Map Register for 20 to 20+3 |
| 0x418 | CH_MAP_REG6 | Interrupt Channel Map Register for 24 to 24+3 |
| 0x41c | CH_MAP_REG7 | Interrupt Channel Map Register for 28 to 28+3 |
| 0x420 | CH_MAP_REG8 | Interrupt Channel Map Register for 32 to 32+3 |
| 0x424 | CH_MAP_REG9 | Interrupt Channel Map Register for 36 to 36+3 |
| 0x428 | CH_MAP_REG10 | Interrupt Channel Map Register for 40 to 40+3 |
| 0x42c | CH_MAP_REG11 | Interrupt Channel Map Register for 44 to 44+3 |
| 0x430 | CH_MAP_REG12 | Interrupt Channel Map Register for 48 to 48+3 |
| 0x434 | CH_MAP_REG13 | Interrupt Channel Map Register for 52 to 52+3 |
| 0x438 | CH_MAP_REG14 | Interrupt Channel Map Register for 56 to 56+3 |
| 0x43c | CH_MAP_REG15 | Interrupt Channel Map Register for 60 to 60+3 |
| 0x440 | CH_MAP_REG16 | Interrupt Channel Map Register for 64 to 64+3 |
| 0x444 | CH_MAP_REG17 | Interrupt Channel Map Register for 68 to 68+3 |
| 0x448 | CH_MAP_REG18 | Interrupt Channel Map Register for 72 to 72+3 |
| 0x44c | CH_MAP_REG19 | Interrupt Channel Map Register for 76 to 76+3 |
| 0x800 | HINT_MAP_REG0 | Host Interrupt Map Register for 0 to 0+3 |
| 0x804 | HINT_MAP_REG1 | Host Interrupt Map Register for 4 to 4+3 |
| 0x808 | HINT_MAP_REG2 | Host Interrupt Map Register for 8 to 8+3 |
| 0x80c | HINT_MAP_REG3 | Host Interrupt Map Register for 12 to 12+3 |
| 0x810 | HINT_MAP_REG4 | Host Interrupt Map Register for 16 to 16+3 |

Table 8-39. CIC2 Register (continued)

| ADDRESS OFFSET | REGISTER MNEMONIC | REGISTER NAME |
|----------------|-------------------|--|
| 0x814 | HINT_MAP_REG5 | Host Interrupt Map Register for 20 to 20+3 |
| 0x818 | HINT_MAP_REG6 | Host Interrupt Map Register for 24 to 24+3 |
| 0x81c | HINT_MAP_REG7 | Host Interrupt Map Register for 28 to 28+3 |
| 0x1500 | ENABLE_HINT_REG0 | Host Int Enable Register 0 |

8.8.3 Inter-Processor Register Map

Table 8-40. IPC Generation Registers (IPCGRx)

| ADDRESS START | ADDRESS END | SIZE | REGISTER NAME | DESCRIPTION |
|---------------|-------------|------|---------------|--|
| 0x02620200 | 0x02620203 | 4B | NMIGR0 | NMI Event Generation Register for CorePac0 |
| 0x02620204 | 0x02620207 | 4B | NMIGR1 | NMI Event Generation Register for CorePac 1 (C6657 only) |
| 0x02620208 | 0x0262020B | 4B | Reserved | Reserved |
| 0x0262020C | 0x0262020F | 4B | Reserved | Reserved |
| 0x02620210 | 0x02620213 | 4B | Reserved | Reserved |
| 0x02620214 | 0x02620217 | 4B | Reserved | Reserved |
| 0x02620218 | 0x0262021B | 4B | Reserved | Reserved |
| 0x0262021C | 0x0262021F | 4B | Reserved | Reserved |
| 0x02620220 | 0x0262023F | 32B | Reserved | Reserved |
| 0x02620240 | 0x02620243 | 4B | IPCGR0 | IPC Generation Register for CorePac 0 |
| 0x02620244 | 0x02620247 | 4B | IPCGR1 | IPC Generation Register for CorePac 1 (C6657 only) |
| 0x02620248 | 0x0262024B | 4B | Reserved | Reserved |
| 0x0262024C | 0x0262024F | 4B | Reserved | Reserved |
| 0x02620250 | 0x02620253 | 4B | Reserved | Reserved |
| 0x02620254 | 0x02620257 | 4B | Reserved | Reserved |
| 0x02620258 | 0x0262025B | 4B | Reserved | Reserved |
| 0x0262025C | 0x0262025F | 4B | Reserved | Reserved |
| 0x02620260 | 0x0262027B | 28B | Reserved | Reserved |
| 0x0262027C | 0x0262027F | 4B | IPCGRH | IPC Generation Register for Host |
| 0x02620280 | 0x02620283 | 4B | IPCAR0 | IPC Acknowledgement Register for CorePac 0 |
| 0x02620284 | 0x02620287 | 4B | IPCAR1 | IPC Acknowledgement Register for CorePac 1 (C6657 only) |
| 0x02620288 | 0x0262028B | 4B | Reserved | Reserved |
| 0x0262028C | 0x0262028F | 4B | Reserved | Reserved |
| 0x02620290 | 0x02620293 | 4B | Reserved | Reserved |
| 0x02620294 | 0x02620297 | 4B | Reserved | Reserved |
| 0x02620298 | 0x0262029B | 4B | Reserved | Reserved |
| 0x0262029C | 0x0262029F | 4B | Reserved | Reserved |
| 0x026202A0 | 0x026202BB | 28B | Reserved | Reserved |
| 0x026202BC | 0x026202BF | 4B | IPCARH | IPC Acknowledgement Register for Host |

8.8.4 $\overline{\text{NMI}}$ and $\overline{\text{LRESET}}$

Non-maskable interrupts ($\overline{\text{NMI}}$) can be generated by chip-level registers and the $\overline{\text{LRESET}}$ can be generated by software writing into LPSC registers. $\overline{\text{LRESET}}$ and $\overline{\text{NMI}}$ can also be asserted by device pins or watchdog timers. One $\overline{\text{NMI}}$ pin and one $\overline{\text{LRESET}}$ pin are shared by all CorePacs on the device. The CORESEL[3:0] pins can be configured to select between the CorePacs available as shown in Table 8-41.

Table 8-41. $\overline{\text{LRESET}}$ and $\overline{\text{NMI}}$ Decoding

| CORESEL[1:0] PIN INPUT | $\overline{\text{LRESET}}$ PIN INPUT | $\overline{\text{NMI}}$ PIN INPUT | $\overline{\text{LRESETNMIEN}}$ PIN INPUT | RESET MUX BLOCK OUTPUT |
|------------------------|--------------------------------------|-----------------------------------|---|--|
| XX | X | X | 1 | No local reset or $\overline{\text{NMI}}$ assertion. |
| 00 | 0 | X | 0 | Assert local reset to CorePac 0 |
| 01 | 0 | X | 0 | Assert local reset to CorePac 1 (C6657) or Reserved (C6655) |
| 1x | 0 | X | 0 | Assert local reset to all CorePacs |
| 00 | 1 | 1 | 0 | De-assert local reset & $\overline{\text{NMI}}$ to CorePac 0 |
| 01 | 1 | 1 | 0 | De-assert local reset & $\overline{\text{NMI}}$ to CorePac 1 (C6657) or Reserved (C6655) |
| 1x | 1 | 1 | 0 | De-assert local reset & $\overline{\text{NMI}}$ to all CorePacs |
| 00 | 1 | 0 | 0 | Assert $\overline{\text{NMI}}$ to CorePac 0 |
| 01 | 1 | 0 | 0 | Assert $\overline{\text{NMI}}$ to CorePac 1 (C6657) or Reserved (C6655) |
| 1x | 1 | 0 | 0 | Assert $\overline{\text{NMI}}$ to all CorePacs |

8.8.5 External Interrupts Electrical Data/Timing

Table 8-42. NMI and Local Reset Timing Requirements⁽¹⁾

(see Figure 8-26)

| NO. | | | MIN | MAX | UNIT |
|-----|--|--|------|-----|------|
| 1 | $t_{\text{su}}(\overline{\text{LRESET}}-\overline{\text{LRESETNMIENL}})$ | Setup Time - $\overline{\text{LRESET}}$ valid before $\overline{\text{LRESETNMIEN}}$ low | 12*P | | ns |
| 1 | $t_{\text{su}}(\overline{\text{NMI}}-\overline{\text{LRESETNMIENL}})$ | Setup Time - $\overline{\text{NMI}}$ valid before $\overline{\text{LRESETNMIEN}}$ low | 12*P | | ns |
| 1 | $t_{\text{su}}(\text{CORESELn}-\overline{\text{LRESETNMIENL}})$ | Setup Time - CORESEL[2:0] valid before $\overline{\text{LRESETNMIEN}}$ low | 12*P | | ns |
| 2 | $t_{\text{h}}(\overline{\text{LRESETNMIENL}}-\overline{\text{LRESET}})$ | Hold Time - $\overline{\text{LRESET}}$ valid after $\overline{\text{LRESETNMIEN}}$ high | 12*P | | ns |
| 2 | $t_{\text{h}}(\overline{\text{LRESETNMIENL}}-\overline{\text{NMI}})$ | Hold Time - $\overline{\text{NMI}}$ valid after $\overline{\text{LRESETNMIEN}}$ high | 12*P | | ns |
| 2 | $t_{\text{h}}(\overline{\text{LRESETNMIENL}}-\text{CORESELn})$ | Hold Time - CORESEL[2:0] valid after $\overline{\text{LRESETNMIEN}}$ high | 12*P | | ns |
| 3 | $t_{\text{w}}(\overline{\text{LRESETNMIEN}})$ | Pulse Width - $\overline{\text{LRESETNMIEN}}$ low width | 12*P | | ns |

(1) P = 1/SYSCLK1 clock frequency in ns.

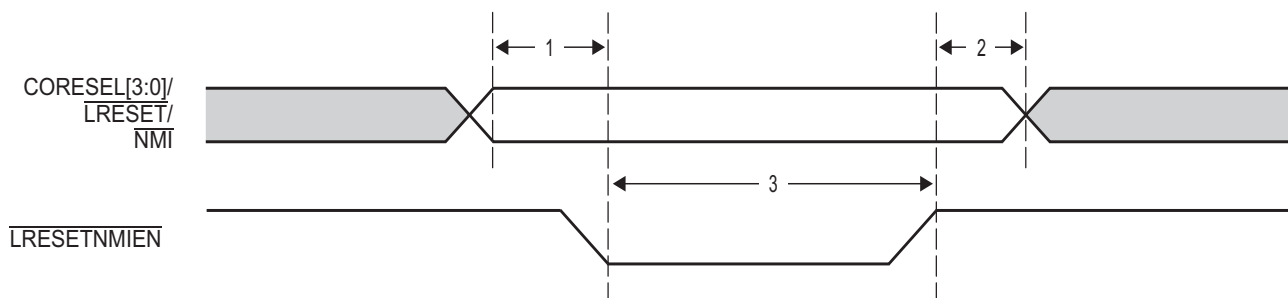


Figure 8-26. NMI and Local Reset Timing

8.9 Memory Protection Unit (MPU)

The C665x supports five MPUs:

- One MPU is used to protect main CORE/3 CFG TeraNet (CFG space of all slave devices on the TeraNet is protected by the MPU).
- Two MPUs are used for QM_SS (one for the DATA PORT port and the other is for the CFG PORT port).
- One MPU is used for Semaphore.
- One MPU is used for EMIF16

This section contains MPU register map and details of device-specific MPU registers only. For MPU features and details of generic MPU registers, see the *Memory Protection Unit (MPU) for KeyStone Devices User's Guide* ([SPRUGW5](#)).

The following tables show the configuration of each MPU and the memory regions protected by each MPU.

Table 8-43. MPU Default Configuration

| SETTING | MPU0 (MAIN CFG TERANET) | MPU1 (QM_SS DATA PORT) | MPU2 (QM_SS CFG PORT) | MPU3 (SEMAPHORE) | MPU4 (EMIF16) |
|---|-------------------------|------------------------|-----------------------|------------------|-----------------|
| Default permission | Assume allowed | Assume allowed | Assume allowed | Assume allowed | Assume allowed |
| Number of allowed IDs supported | 16 | 16 | 16 | 16 | 16 |
| Number of programmable ranges supported | 16 | 5 | 16 | 1 | 16 |
| Compare width | 1KB granularity | 1KB granularity | 1KB granularity | 1KB granularity | 1KB granularity |

Table 8-44. MPU Memory Regions

| | MEMORY PROTECTION | START ADDRESS | END ADDRESS |
|-------------|-------------------|---------------|-------------|
| MPU0 | Main CFG TeraNet | 0x01D00000 | 0x026207FF |
| MPU1 | QM_SS DATA PORT | 0x34000000 | 0x340BFFFF |
| MPU2 | QM_SS CFG PORT | 0x02A00000 | 0x02ABFFFF |
| MPU3 | Semaphore | 0x02640000 | 0x026407FF |
| MPU4 | EMIF16 | 0x70000000 | 0x7FFFFFFF |

Table 8-45 shows the privilege ID of each CORE and every mastering peripheral. Table 8-45 also shows the privilege level (supervisor vs. user), and access type (instruction read vs. data/DMA read or write) of each master on the device. In some cases, a particular setting depends on software being executed at the time of the access or the configuration of the master peripheral.

Table 8-45. Privilege ID Settings

| PRIVILEGE ID | MASTER | PRIVILEGE LEVEL | ACCESS TYPE |
|--------------|------------------------|--|-------------|
| 0 | CorePac0 | SW dependant, driven by MSMC | DMA |
| 1 | CorePac1 (C6657 only) | SW dependant, driven by MSMC | DMA |
| 2 | Reserved | | |
| 3 | Reserved | | |
| 4 | Reserved | | |
| 5 | Reserved | | |
| 6 | uPP | User | DMA |
| 7 | EMAC | User | DMA |
| 8 | QM_PKTDMA | User | DMA |
| 9 | SRIO_Packet DMA/SRIO_M | User/Driven by SRIO block, user mode and supervisor mode is determined on a per-transaction basis. Only the transaction with source ID matching the value in the SupervisorID Register is granted supervisor mode. | DMA |
| 10 | QM_second | User | DMA |
| 11 | PCIe | Supervisor | DMA |
| 12 | DAP | Driven by Debug_SS | DMA |
| 13 | HyperLink | Supervisor | DMA |
| 14 | HyperLink | Supervisor | DMA |
| 15 | HyperLink | Supervisor | DMA |

Table 8-46 shows the master ID of each CorePac and every mastering peripheral. Master IDs are used to determine allowed connections between masters and slaves. Unlike privilege IDs, which can be shared across different masters, master IDs are unique to each master.

Table 8-46. Master ID Settings⁽¹⁾

| MASTER ID | MASTER | MASTER ID | MASTER |
|-----------|--|-----------|---|
| 0 | CorePac0 | 40 - 47 | Reserved |
| 1 | CorePac1 (C6657) or Reserved (C6655) | 48 | DAP |
| 2 | Reserved | 49 | Reserved |
| 3 | Reserved | 50 | EDMA3_CC |
| 4 | Reserved | 51 | Reserved |
| 5 | Reserved | 52 | MSMC ⁽²⁾ |
| 6 | Reserved | 53 | PCIe |
| 7 | Reserved | 54 | SRIO_Master |
| 8 | CorePac0_CFG | 55 | HyperLink |
| 9 | CorePac1_CFG (C6657) or Reserved (C6655) | 56 | EMAC |
| 10 | Reserved | 57 - 87 | Reserved |
| 11 | Reserved | 88 - 91 | QM_PKTDMMA |
| 12 | Reserved | 92 - 93 | QM_Second |
| 13 | Reserved | 94 | Reserved |
| 14 | Reserved | 95 | uPP |
| 15 | Reserved | 96 - 127 | Reserved |
| 16 | Reserved | 128 | Tracer_core_0 ⁽³⁾ |
| 17 | Reserved | 129 | Tracer_core_1 (C6657) or Reserved (C6655) |
| 18 | Reserved | 130 | Reserved |
| 19 | Reserved | 131 | Reserved |
| 20 | Reserved | 132 | Reserved |
| 21 | Reserved | 133 | Reserved |
| 22 | Reserved | 134 | Reserved |
| 23 | Reserved | 135 | Reserved |
| 24 | Reserved | 136 | Tracer_MSMC0 |
| 25 | Reserved | 137 | Tracer_MSMC1 |
| 26 | Reserved | 138 | Tracer_MSMC2 |
| 27 | Reserved | 139 | Tracer_MSMC3 |
| 28 | EDMA_TC0 read | 140 | Tracer_DDR |
| 29 | EDMA_TC0 write | 141 | Tracer_SEM |
| 30 | EDMA_TC1 read | 142 | Tracer_QM_CFG |
| 31 | EDMA_TC1 write | 143 | Tracer_QM_DMA |
| 32 | EDMA_TC2 read | 144 | Tracer_CFG |
| 33 | EDMA_TC2 write | 145 | Reserved |
| 34 | EDMA_TC3 read | 146 | Reserved |
| 35 | EDMA_TC3 write | 147 | Reserved |
| 36 - 37 | Reserved | 148 | Tracer_EMIF16 |
| 38 - 39 | SRIO_PKTDMMA | | |

- (1) Some of the PKTDMA-based peripherals require multiple master IDs. QMS_PKTDMMA is assigned with 88,89,90,91, but only 88-89 are actually used. There are two master ID values are assigned for the QM_second master port, one master ID for external linking RAM and the other one for the PDSP/MCDM accesses.
- (2) The master ID for MSMC is for the transactions initiated by MSMC internally and sent to the DDR.
- (3) All Tracers are set to the same master ID and bit 7 of the master ID needs to be 1.

8.9.1 MPU Registers

This section includes the offsets for MPU registers and definitions for device specific MPU registers.

8.9.1.1 MPU Register Map

Table 8-47. MPU0 Registers

| OFFSET | NAME | DESCRIPTION |
|--------|--------------|--|
| 0h | REVID | Revision ID |
| 4h | CONFIG | Configuration |
| 10h | IRAWSTAT | Interrupt raw status/set |
| 14h | IENSTAT | Interrupt enable status/clear |
| 18h | IENSET | Interrupt enable |
| 1Ch | IENCLR | Interrupt enable clear |
| 20h | EOI | End of interrupt |
| 200h | PROG0_MPSAR | Programmable range 0, start address |
| 204h | PROG0_MPEAR | Programmable range 0, end address |
| 208h | PROG0_MPPA | Programmable range 0, memory page protection attributes |
| 210h | PROG1_MPSAR | Programmable range 1, start address |
| 214h | PROG1_MPEAR | Programmable range 1, end address |
| 218h | PROG1_MPPA | Programmable range 1, memory page protection attributes |
| 220h | PROG2_MPSAR | Programmable range 2, start address |
| 224h | PROG2_MPEAR | Programmable range 2, end address |
| 228h | PROG2_MPPA | Programmable range 2, memory page protection attributes |
| 230h | PROG3_MPSAR | Programmable range 3, start address |
| 234h | PROG3_MPEAR | Programmable range 3, end address |
| 238h | PROG3_MPPA | Programmable range 3, memory page protection attributes |
| 240h | PROG4_MPSAR | Programmable range 4, start address |
| 244h | PROG4_MPEAR | Programmable range 4, end address |
| 248h | PROG4_MPPA | Programmable range 4, memory page protection attributes |
| 250h | PROG5_MPSAR | Programmable range 5, start address |
| 254h | PROG5_MPEAR | Programmable range 5, end address |
| 258h | PROG5_MPPA | Programmable range 5, memory page protection attributes |
| 260h | PROG6_MPSAR | Programmable range 6, start address |
| 264h | PROG6_MPEAR | Programmable range 6, end address |
| 268h | PROG6_MPPA | Programmable range 6, memory page protection attributes |
| 270h | PROG7_MPSAR | Programmable range 7, start address |
| 274h | PROG7_MPEAR | Programmable range 7, end address |
| 278h | PROG7_MPPA | Programmable range 7, memory page protection attributes |
| 280h | PROG8_MPSAR | Programmable range 8, start address |
| 284h | PROG8_MPEAR | Programmable range 8, end address |
| 288h | PROG8_MPPA | Programmable range 8, memory page protection attributes |
| 290h | PROG9_MPSAR | Programmable range 9, start address |
| 294h | PROG9_MPEAR | Programmable range 9, end address |
| 298h | PROG9_MPPA | Programmable range 9, memory page protection attributes |
| 2A0h | PROG10_MPSAR | Programmable range 10, start address |
| 2A4h | PROG10_MPEAR | Programmable range 10, end address |
| 2A8h | PROG10_MPPA | Programmable range 10, memory page protection attributes |
| 2B0h | PROG11_MPSAR | Programmable range 11, start address |
| 2B4h | PROG11_MPEAR | Programmable range 11, end address |

Table 8-47. MPU0 Registers (continued)

| OFFSET | NAME | DESCRIPTION |
|--------|--------------|--|
| 2B8h | PROG11_MPPA | Programmable range 11, memory page protection attributes |
| 2C0h | PROG12_MPSAR | Programmable range 12, start address |
| 2C4h | PROG12_MPEAR | Programmable range 12, end address |
| 2C8h | PROG12_MPPA | Programmable range 12, memory page protection attributes |
| 2D0h | PROG13_MPSAR | Programmable range 13, start address |
| 2D4h | PROG13_MPEAR | Programmable range 13, end address |
| 2Dh | PROG13_MPPA | Programmable range 13, memory page protection attributes |
| 2E0h | PROG14_MPSAR | Programmable range 14, start address |
| 2E4h | PROG14_MPEAR | Programmable range 14, end address |
| 2E8h | PROG14_MPPA | Programmable range 14, memory page protection attributes |
| 2F0h | PROG15_MPSAR | Programmable range 15, start address |
| 2F4h | PROG15_MPEAR | Programmable range 15, end address |
| 2F8h | PROG15_MPPA | Programmable range 15, memory page protection attributes |
| 300h | FLTADDRR | Fault address |
| 304h | FLTSTAT | Fault status |
| 308h | FLTCLR | Fault clear |

Table 8-48. MPU1 Registers

| OFFSET | NAME | DESCRIPTION |
|--------|-------------|---|
| 0h | REVID | Revision ID |
| 4h | CONFIG | Configuration |
| 10h | IRAWSTAT | Interrupt raw status/set |
| 14h | IENSTAT | Interrupt enable status/clear |
| 18h | IENSET | Interrupt enable |
| 1Ch | IENCLR | Interrupt enable clear |
| 20h | EOI | End of interrupt |
| 200h | PROG0_MPSAR | Programmable range 0, start address |
| 204h | PROG0_MPEAR | Programmable range 0, end address |
| 208h | PROG0_MPPA | Programmable range 0, memory page protection attributes |
| 210h | PROG1_MPSAR | Programmable range 1, start address |
| 214h | PROG1_MPEAR | Programmable range 1, end address |
| 218h | PROG1_MPPA | Programmable range 1, memory page protection attributes |
| 220h | PROG2_MPSAR | Programmable range 2, start address |
| 224h | PROG2_MPEAR | Programmable range 2, end address |
| 228h | PROG2_MPPA | Programmable range 2, memory page protection attributes |
| 230h | PROG3_MPSAR | Programmable range 3, start address |
| 234h | PROG3_MPEAR | Programmable range 3, end address |
| 238h | PROG3_MPPA | Programmable range 3, memory page protection attributes |
| 240h | PROG4_MPSAR | Programmable range 4, start address |
| 244h | PROG4_MPEAR | Programmable range 4, end address |
| 248h | PROG4_MPPA | Programmable range 4, memory page protection attributes |
| 300h | FLTADDRR | Fault address |
| 304h | FLTSTAT | Fault status |
| 308h | FLTCLR | Fault clear |

Table 8-49. MPU2 Registers

| OFFSET | NAME | DESCRIPTION |
|--------|--------------|--|
| 0h | REVID | Revision ID |
| 4h | CONFIG | Configuration |
| 10h | IRAWSTAT | Interrupt raw status/set |
| 14h | IENSTAT | Interrupt enable status/clear |
| 18h | IENSET | Interrupt enable |
| 1Ch | IENCLR | Interrupt enable clear |
| 20h | EOI | End of interrupt |
| 200h | PROG0_MPSAR | Programmable range 0, start address |
| 204h | PROG0_MPEAR | Programmable range 0, end address |
| 208h | PROG0_MPPA | Programmable range 0, memory page protection attributes |
| 210h | PROG1_MPSAR | Programmable range 1, start address |
| 214h | PROG1_MPEAR | Programmable range 1, end address |
| 218h | PROG1_MPPA | Programmable range 1, memory page protection attributes |
| 220h | PROG2_MPSAR | Programmable range 2, start address |
| 224h | PROG2_MPEAR | Programmable range 2, end address |
| 228h | PROG2_MPPA | Programmable range 2, memory page protection attributes |
| 230h | PROG3_MPSAR | Programmable range 3, start address |
| 234h | PROG3_MPEAR | Programmable range 3, end address |
| 238h | PROG3_MPPA | Programmable range 3, memory page protection attributes |
| 240h | PROG4_MPSAR | Programmable range 4, start address |
| 244h | PROG4_MPEAR | Programmable range 4, end address |
| 248h | PROG4_MPPA | Programmable range 4, memory page protection attributes |
| 250h | PROG5_MPSAR | Programmable range 5, start address |
| 254h | PROG5_MPEAR | Programmable range 5, end address |
| 258h | PROG5_MPPA | Programmable range 5, memory page protection attributes |
| 260h | PROG6_MPSAR | Programmable range 6, start address |
| 264h | PROG6_MPEAR | Programmable range 6, end address |
| 268h | PROG6_MPPA | Programmable range 6, memory page protection attributes |
| 270h | PROG7_MPSAR | Programmable range 7, start address |
| 274h | PROG7_MPEAR | Programmable range 7, end address |
| 278h | PROG7_MPPA | Programmable range 7, memory page protection attributes |
| 280h | PROG8_MPSAR | Programmable range 8, start address |
| 284h | PROG8_MPEAR | Programmable range 8, end address |
| 288h | PROG8_MPPA | Programmable range 8, memory page protection attributes |
| 290h | PROG9_MPSAR | Programmable range 9, start address |
| 294h | PROG9_MPEAR | Programmable range 9, end address |
| 298h | PROG9_MPPA | Programmable range 9, memory page protection attributes |
| 2A0h | PROG10_MPSAR | Programmable range 10, start address |
| 2A4h | PROG10_MPEAR | Programmable range 10, end address |
| 2A8h | PROG10_MPPA | Programmable range 10, memory page protection attributes |
| 2B0h | PROG11_MPSAR | Programmable range 11, start address |
| 2B4h | PROG11_MPEAR | Programmable range 11, end address |
| 2B8h | PROG11_MPPA | Programmable range 11, memory page protection attributes |
| 2C0h | PROG12_MPSAR | Programmable range 12, start address |
| 2C4h | PROG12_MPEAR | Programmable range 12, end address |
| 2C8h | PROG12_MPPA | Programmable range 12, memory page protection attributes |
| 2D0h | PROG13_MPSAR | Programmable range 13, start address |

Table 8-49. MPU2 Registers (continued)

| OFFSET | NAME | DESCRIPTION |
|--------|--------------|--|
| 2D4h | PROG13_MPEAR | Programmable range 13, end address |
| 2Dh | PROG13_MPPA | Programmable range 13, memory page protection attributes |
| 2E0h | PROG14_MPSAR | Programmable range 14, start address |
| 2E4h | PROG14_MPEAR | Programmable range 14, end address |
| 2E8h | PROG14_MPPA | Programmable range 14, memory page protection attributes |
| 2F0h | PROG15_MPSAR | Programmable range 15, start address |
| 2F4h | PROG15_MPEAR | Programmable range 15, end address |
| 2F8h | PROG15_MPPA | Programmable range 15, memory page protection attributes |
| 300h | FLTADDRR | Fault address |
| 304h | FLTSTAT | Fault status |
| 308h | FLTCLR | Fault clear |

Table 8-50. MPU3 Registers

| OFFSET | NAME | DESCRIPTION |
|--------|-------------|---|
| 0h | REVID | Revision ID |
| 4h | CONFIG | Configuration |
| 10h | IRAWSTAT | Interrupt raw status/set |
| 14h | IENSTAT | Interrupt enable status/clear |
| 18h | IENSET | Interrupt enable |
| 1Ch | IENCLR | Interrupt enable clear |
| 20h | EOI | End of interrupt |
| 200h | PROG0_MPSAR | Programmable range 0, start address |
| 204h | PROG0_MPEAR | Programmable range 0, end address |
| 208h | PROG0_MPPA | Programmable range 0, memory page protection attributes |
| 300h | FLTADDRR | Fault address |
| 304h | FLTSTAT | Fault status |
| 308h | FLTCLR | Fault clear |

Table 8-51. MPU4 Registers

| OFFSET | NAME | DESCRIPTION |
|--------|--------------|--|
| 0h | REVID | Revision ID |
| 4h | CONFIG | Configuration |
| 10h | IRAWSTAT | Interrupt raw status/set |
| 14h | IENSTAT | Interrupt enable status/clear |
| 18h | IENSET | Interrupt enable |
| 1Ch | IENCLR | Interrupt enable clear |
| 20h | EOI | End of interrupt |
| 200h | PROG0_MPSAR | Programmable range 0, start address |
| 204h | PROG0_MPEAR | Programmable range 0, end address |
| 208h | PROG0_MPPA | Programmable range 0, memory page protection attributes |
| 210h | PROG1_MPSAR | Programmable range 1, start address |
| 214h | PROG1_MPEAR | Programmable range 1, end address |
| 218h | PROG1_MPPA | Programmable range 1, memory page protection attributes |
| 220h | PROG2_MPSAR | Programmable range 2, start address |
| 224h | PROG2_MPEAR | Programmable range 2, end address |
| 228h | PROG2_MPPA | Programmable range 2, memory page protection attributes |
| 230h | PROG3_MPSAR | Programmable range 3, start address |
| 234h | PROG3_MPEAR | Programmable range 3, end address |
| 238h | PROG3_MPPA | Programmable range 3, memory page protection attributes |
| 240h | PROG4_MPSAR | Programmable range 4, start address |
| 244h | PROG4_MPEAR | Programmable range 4, end address |
| 248h | PROG4_MPPA | Programmable range 4, memory page protection attributes |
| 250h | PROG5_MPSAR | Programmable range 5, start address |
| 254h | PROG5_MPEAR | Programmable range 5, end address |
| 258h | PROG5_MPPA | Programmable range 5, memory page protection attributes |
| 260h | PROG6_MPSAR | Programmable range 6, start address |
| 264h | PROG6_MPEAR | Programmable range 6, end address |
| 268h | PROG6_MPPA | Programmable range 6, memory page protection attributes |
| 270h | PROG7_MPSAR | Programmable range 7, start address |
| 274h | PROG7_MPEAR | Programmable range 7, end address |
| 278h | PROG7_MPPA | Programmable range 7, memory page protection attributes |
| 280h | PROG8_MPSAR | Programmable range 8, start address |
| 284h | PROG8_MPEAR | Programmable range 8, end address |
| 288h | PROG8_MPPA | Programmable range 8, memory page protection attributes |
| 290h | PROG9_MPSAR | Programmable range 9, start address |
| 294h | PROG9_MPEAR | Programmable range 9, end address |
| 298h | PROG9_MPPA | Programmable range 9, memory page protection attributes |
| 2A0h | PROG10_MPSAR | Programmable range 10, start address |
| 2A4h | PROG10_MPEAR | Programmable range 10, end address |
| 2A8h | PROG10_MPPA | Programmable range 10, memory page protection attributes |
| 2B0h | PROG11_MPSAR | Programmable range 11, start address |
| 2B4h | PROG11_MPEAR | Programmable range 11, end address |
| 2B8h | PROG11_MPPA | Programmable range 11, memory page protection attributes |
| 2C0h | PROG12_MPSAR | Programmable range 12, start address |
| 2C4h | PROG12_MPEAR | Programmable range 12, end address |
| 2C8h | PROG12_MPPA | Programmable range 12, memory page protection attributes |
| 2D0h | PROG13_MPSAR | Programmable range 13, start address |

Table 8-51. MPU4 Registers (continued)

| OFFSET | NAME | DESCRIPTION |
|--------|--------------|--|
| 2D4h | PROG13_MPEAR | Programmable range 13, end address |
| 2Dh | PROG13_MPPA | Programmable range 13, memory page protection attributes |
| 2E0h | PROG14_MPSAR | Programmable range 14, start address |
| 2E4h | PROG14_MPEAR | Programmable range 14, end address |
| 2E8h | PROG14_MPPA | Programmable range 14, memory page protection attributes |
| 2F0h | PROG15_MPSAR | Programmable range 15, start address |
| 2F4h | PROG15_MPEAR | Programmable range 15, end address |
| 2F8h | PROG15_MPPA | Programmable range 15, memory page protection attributes |
| 300h | FLTADDRR | Fault address |
| 304h | FLTSTAT | Fault status |
| 308h | FLTCLR | Fault clear |

8.9.1.2 Device-Specific MPU Registers

8.9.1.2.1 Configuration Register (CONFIG)

The Configuration Register (CONFIG) contains the configuration value of the MPU.

Figure 8-27. Configuration Register (CONFIG)

| | 31 | 24 | 23 | 20 | 19 | 16 | 15 | 12 | 11 | 1 | 0 |
|---------------------|------------|-----|-----------|-----|----------|------|----------|------|----------|----------------|-----|
| | ADDR_WIDTH | | NUM_FIXED | | NUM_PROG | | NUM_AIDS | | Reserved | ASSUME_ALLOWED | |
| | MPU0 | R-0 | | R-0 | | R-16 | | R-16 | | R-0 | R-1 |
| | MPU1 | R-0 | | R-0 | | R-5 | | R-16 | | R-0 | R-1 |
| Reset Values | MPU2 | R-0 | | R-0 | | R-16 | | R-16 | | R-0 | R-1 |
| | MPU3 | R-0 | | R-0 | | R-1 | | R-16 | | R-0 | R-1 |
| | MPU4 | R-0 | | R-0 | | R-16 | | R-16 | | R-0 | R-1 |

Legend: R = Read only; -n = value after reset

Table 8-52. Configuration Register (CONFIG) Field Descriptions

| Bit | Field | Description |
|---------|----------------|---|
| 31 – 24 | ADDR_WIDTH | Address alignment for range checking <ul style="list-style-type: none"> 0 = 1KB alignment 6 = 64KB alignment |
| 23 – 20 | NUM_FIXED | Number of fixed address ranges |
| 19 – 16 | NUM_PROG | Number of programmable address ranges |
| 15 – 12 | NUM_AIDS | Number of supported AIDs |
| 11 – 1 | Reserved | Reserved. These bits will always reads as 0. |
| 0 | ASSUME_ALLOWED | Assume allowed bit. When an address is not covered by any MPU protection range, this bit determines whether the transfer is assumed to be allowed or not. <ul style="list-style-type: none"> 0 = Assume disallowed 1 = Assume allowed |

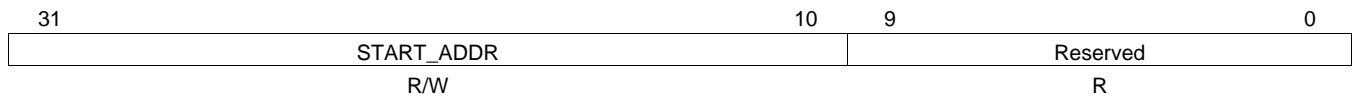
8.9.2 MPU Programmable Range Registers

8.9.2.1 Programmable Range *n* Start Address Register (PROG_{*n*}_MPSAR)

The Programmable Address Start Register holds the start address for the range. This register is writeable by a supervisor entity only.

The start address must be aligned on a page boundary. The size of the page is 1K byte. The size of the page determines the width of the address field in MPSAR and MPEAR.

Figure 8-28. Programmable Range *n* Start Address Register (PROG_{*n*}_MPSAR)



Legend: R = Read only; R/W = Read/Write

Table 8-53. Programmable Range *n* Start Address Register (PROG_{*n*}_MPSAR) Field Descriptions

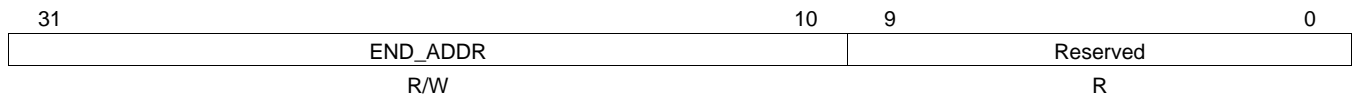
| Bit | Field | Description |
|---------|------------|---|
| 31 – 10 | START_ADDR | Start address for range <i>n</i> . |
| 9 – 0 | Reserved | Reserved and these bits always read as 0. |

8.9.2.2 Programmable Range *n* End Address Register (PROG_{*n*}MPEAR)

The Programmable Address End Register holds the end address for the range. This register is writeable by a supervisor entity only.

The end address must be aligned on a page boundary. The size of the page depends on the MPU number. The page size for MPU1 is 1K byte and for MPU2 it is 64K bytes. The size of the page determines the width of the address field in MPSAR and MPEAR

Figure 8-29. Programmable Range *n* End Address Register (PROG_{*n*}MPEAR)



Legend: R = Read only; R/W = Read/Write

Table 8-54. Programmable Range *n* End Address Register (PROG_{*n*}MPEAR) Field Descriptions

| Bit | Field | Description |
|---------|----------|--|
| 31 – 10 | END_ADDR | End address for range <i>n</i> . |
| 9 – 0 | Reserved | Reserved and these bits always read as 3FFh. |

8.9.2.3 Programmable Range *n* Memory Protection Page Attribute Register (PROG_{*n*}_MPPA)

The Programmable Address Memory Protection Page Attribute Register holds the permissions for the region. This register is writeable only by a non-debug supervisor entity.

Figure 8-30. Programmable Range *n* Memory Protection Page Attribute Register (PROG_{*n*}_MPPA)

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|----------|------|------|------|------|------|----------|--|----------|--|-------|-------|-------|-------|-------|-------|------|------|------|------|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 31 | | | | 26 | | | | 25 | | 24 | | 23 | | 22 | | 21 | | 20 | | 19 | | 18 | | 17 | | 16 | | 15 | |
| Reserved | | | | | | | | | | AID15 | AID14 | AID13 | AID12 | AID11 | AID10 | AID9 | AID8 | AID7 | AID6 | AID5 | | | | | | | | | |
| R | | | | | | | | | | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| 14 | | 13 | | 12 | | 11 | | 10 | | 9 | | 8 | | 7 | | 6 | | 5 | | 4 | | 3 | | 2 | | 1 | | 0 | |
| AID4 | AID3 | AID2 | AID1 | AID0 | AIDX | Reserved | | Reserved | | EMU | SR | SW | SX | UR | UW | UX | | | | | | | | | | | | | |
| R/W | R/W | R/W | R/W | R/W | R/W | R | | R | | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Legend: R = Read only; R/W = Read/Write

Table 8-55. Programmable Range *n* Memory Protection Page Attribute Register (PROG_{*n*}_MPPA) Field Descriptions

| Bit | Field | Description |
|---------|----------|---|
| 31 – 26 | Reserved | Reserved. These bits will always reads as 0. |
| 25 | AID15 | Controls permission check of ID = 15 <ul style="list-style-type: none"> 0 = AID is not checked for permissions 1 = AID is checked for permissions |
| 24 | AID14 | Controls permission check of ID = 14 <ul style="list-style-type: none"> 0 = AID is not checked for permissions 1 = AID is checked for permissions |
| 23 | AID13 | Controls permission check of ID = 13 <ul style="list-style-type: none"> 0 = AID is not checked for permissions 1 = AID is checked for permissions |
| 22 | AID12 | Controls permission check of ID = 12 <ul style="list-style-type: none"> 0 = AID is not checked for permissions 1 = AID is checked for permissions |
| 21 | AID11 | Controls permission check of ID = 11 <ul style="list-style-type: none"> 0 = AID is not checked for permissions 1 = AID is checked for permissions |
| 20 | AID10 | Controls permission check of ID = 10 <ul style="list-style-type: none"> 0 = AID is not checked for permissions 1 = AID is checked for permissions |
| 19 | AID9 | Controls permission check of ID = 9 <ul style="list-style-type: none"> 0 = AID is not checked for permissions 1 = AID is checked for permissions |
| 18 | AID8 | Controls permission check of ID = 8 <ul style="list-style-type: none"> 0 = AID is not checked for permissions 1 = AID is checked for permissions |
| 17 | AID7 | Controls permission check of ID = 7 <ul style="list-style-type: none"> 0 = AID is not checked for permissions 1 = AID is checked for permissions |
| 16 | AID6 | Controls permission check of ID = 6 <ul style="list-style-type: none"> 0 = AID is not checked for permissions 1 = AID is checked for permissions |
| 15 | AID5 | Controls permission check of ID = 5 <ul style="list-style-type: none"> 0 = AID is not checked for permissions 1 = AID is checked for permissions |
| 14 | AID4 | Controls permission check of ID = 4 <ul style="list-style-type: none"> 0 = AID is not checked for permissions 1 = AID is checked for permissions |

Table 8-55. Programmable Range n Memory Protection Page Attribute Register (PROG n _MPPA) Field Descriptions (continued)

| Bit | Field | Description |
|-----|----------|--|
| 13 | AID3 | Controls permission check of ID = 3 <ul style="list-style-type: none"> • 0 = AID is not checked for permissions • 1 = AID is checked for permissions |
| 12 | AID2 | Controls permission check of ID = 2 <ul style="list-style-type: none"> • 0 = AID is not checked for permissions • 1 = AID is checked for permissions |
| 11 | AID1 | Controls permission check of ID = 1 <ul style="list-style-type: none"> • 0 = AID is not checked for permissions • 1 = AID is checked for permissions |
| 10 | AID0 | Controls permission check of ID = 0 <ul style="list-style-type: none"> • 0 = AID is not checked for permissions • 1 = AID is checked for permissions |
| 9 | AIDX | Controls permission check of ID > 15 <ul style="list-style-type: none"> • 0 = AID is not checked for permissions • 1 = AID is checked for permissions |
| 8 | Reserved | Always reads as 0. |
| 7 | Reserved | Always reads as 1. |
| 6 | EMU | Emulation (debug) access permission. <ul style="list-style-type: none"> • 0 = Debug access not allowed. • 1 = Debug access allowed. |
| 5 | SR | Supervisor Read permission <ul style="list-style-type: none"> • 0 = Access not allowed. • 1 = Access allowed. |
| 4 | SW | Supervisor Write permission <ul style="list-style-type: none"> • 0 = Access not allowed. • 1 = Access allowed. |
| 3 | SX | Supervisor Execute permission <ul style="list-style-type: none"> • 0 = Access not allowed. • 1 = Access allowed. |
| 2 | UR | User Read permission <ul style="list-style-type: none"> • 0 = Access not allowed. • 1 = Access allowed |
| 1 | UW | User Write permission <ul style="list-style-type: none"> • 0 = Access not allowed. • 1 = Access allowed. |
| 0 | UX | User Execute permission <ul style="list-style-type: none"> • 0 = Access not allowed. • 1 = Access allowed. |

8.9.2.4 MPU Registers Reset Values

Table 8-56. Programmable Range *n* Registers Reset Values for MPU0

| PROGRAMMABLE RANGE | MPU0 (MAIN CFG TERANET) | | | |
|--------------------|--|--|--|-----------------------------|
| | START ADDRESS (PROG _n _MPSAR) | END ADDRESS (PROG _n _MPEAR) | MEMORY PAGE PROTECTION ATTRIBUTE (PROG _n _MPPA) | MEMORY PROTECTION |
| PROG0 | 0x01D0_0000 | 0x01D8_007F | 0x03FF_FCB6 | Tracers |
| PROG1 | 0x01F0_0000 | 0x01F7_FFFF | 0x03FF_FC80 | Reserved |
| PROG2 | 0x0200_0000 | 0x0209_FFFF | 0x03FF_FCB6 | Reserved |
| PROG3 | 0x01E0_0000 | 0x01EB_FFFF | 0x03FF_FCB6 | Reserved |
| PROG4 | 0x021C_0000 | 0x021E_0C3F | 0x03FF_FCB6 | TCP/VCP |
| PROG5 | 0x021F_0000 | 0x021F_7FFF | 0x03FF_FCB6 | Reserved |
| PROG6 | 0x0220_0000 | 0x0227_007F | 0x03FF_FCB6 | Timers |
| PROG7 | 0x0231_0000 | 0x0231_03FF | 0x03FF_FCB4 | PLL |
| PROG8 | 0x0232_0000 | 0x0232_03FF | 0x03FF_FCB4 | GPIO |
| PROG9 | 0x0233_0000 | 0x0233_03FF | 0x03FF_FCB4 | SmartReflex |
| PROG10 | 0x0235_0000 | 0x0235_0FFF | 0x03FF_FCB4 | PSC |
| PROG11 | 0x0240_0000 | 0x0245_3FFF | 0x03FF_FCB6 | DEBUG_SS, Tracer Formatters |
| PROG12 | 0x0250_0000 | 0x0252_03FF | 0x03FF_FCB4 | EFUSE |
| PROG13 | 0x0253_0000 | 0x0255_03FF | 0x03FF_FCB6 | I ² C, UART |
| PROG14 | 0x0260_0000 | 0x0260_BFFF | 0x03FF_FCB4 | CICs |
| PROG15 | 0x0262_0000 | 0x0262_07FF | 0x03FF_FCB4 | Chip-level Registers |

Table 8-57. Programmable Range *n* Registers Reset Values for MPU1

| PROGRAMMABLE RANGE | MPU1 (QM_SS DATA PORT) | | | |
|--------------------|--|--|--|------------------------------|
| | START ADDRESS (PROG _n _MPSAR) | END ADDRESS (PROG _n _MPEAR) | MEMORY PAGE PROTECTION ATTRIBUTE (PROG _n _MPPA) | MEMORY PROTECTION |
| PROG0 | 0x3400_0000 | 0x3401_FFFF | 0x03FF_FC80 | Queue Manager subsystem data |
| PROG1 | 0x3402_0000 | 0x3405_FFFF | 0x000F_FCB6 | |
| PROG2 | 0x3406_0000 | 0x3406_7FFF | 0x03FF_FCB4 | |
| PROG3 | 0x3406_8000 | 0x340B_7FFF | 0x03FF_FC80 | |
| PROG4 | 0x340B_8000 | 0x340B_FFFF | 0x03FF_FCB6 | |

Table 8-58. Programmable Range *n* Registers Reset Values for MPU2

| PROGRAMMABLE RANGE | MPU2 (QM_SS CFG PORT) | | | MEMORY PROTECTION |
|--------------------|--|--|--|---------------------------------------|
| | START ADDRESS (PROG _n _MPSAR) | END ADDRESS (PROG _n _MPEAR) | MEMORY PAGE PROTECTION ATTRIBUTE (PROG _n _MPPA) | |
| PROG0 | 0x02A0_0000 | 0x02A1_FFFF | 0x03FF_FCA4 | Queue Manager subsystem configuration |
| PROG1 | 0x02A2_0000 | 0x02A3_FFFF | 0x000F_FCB6 | |
| PROG2 | 0x02A4_0000 | 0x02A5_FFFF | 0x000F_FCB6 | |
| PROG3 | 0x02A6_0000 | 0x02A6_7FFF | 0x03FF_FCB4 | |
| PROG4 | 0x02A6_8000 | 0x02A6_8FFF | 0x03FF_FCB4 | |
| PROG5 | 0x02A6_9000 | 0x02A6_9FFF | 0x03FF_FCB4 | |
| PROG6 | 0x02A6_A000 | 0x02A6_AFFF | 0x03FF_FCB4 | |
| PROG7 | 0x02A6_B000 | 0x02A6_BFFF | 0x03FF_FCB4 | |
| PROG8 | 0x02A6_C000 | 0x02A6_DFFF | 0x03FF_FCB4 | |
| PROG9 | 0x02A6_E000 | 0x02A6_FFFF | 0x03FF_FCB4 | |
| PROG10 | 0x02A8_0000 | 0x02A8_FFFF | 0x03FF_FCA4 | |
| PROG11 | 0x02A9_0000 | 0x02A9_FFFF | 0x03FF_FCB4 | |
| PROG12 | 0x02AA_0000 | 0x02AA_7FFF | 0x03FF_FCB4 | |
| PROG13 | 0x02AA_8000 | 0x02AA_FFFF | 0x03FF_FCB4 | |
| PROG14 | 0x02AB_0000 | 0x02AB_7FFF | 0x03FF_FCB4 | |
| PROG15 | 0x02AB_8000 | 0x02AB_FFFF | 0x03FF_FCB6 | |

Table 8-59. Programmable Range *n* Registers Reset Values for MPU3

| PROGRAMMABLE RANGE | MPU3 (SEMAPHORE) | | | MEMORY PROTECTION |
|--------------------|--|--|---|-------------------|
| | START ADDRESS (PROG _n _MPSAR) | END ADDRESS (PROG _n _MPEAR) | MEMORY PAGE PROTECTION ATTRIBUTES (PROG _n _MPPA) | |
| PROG0 | 0x0264_0000 | 0x0264_07FF | 0x0003_FCB6 | Semaphore |

Table 8-60. Programmable Range *n* Registers Reset Values for MPU4

| PROGRAMMABLE RANGE | MPU4 (EMIF16) | | | MEMORY PROTECTION |
|--------------------|--|--|--|-------------------|
| | START ADDRESS (PROG _n _MPSAR) | END ADDRESS (PROG _n _MPEAR) | MEMORY PAGE PROTECTION ATTRIBUTE (PROG _n _MPPA) | |
| PROG0 | 0x7000_0000 | 0x70FF_FFFF | 0x03FF_FCB6 | EMIF16 data |
| PROG1 | 0x7100_0000 | 0x71FF_FFFF | 0x03FF_FCB6 | |
| PROG2 | 0x7200_0000 | 0x72FF_FFFF | 0x03FF_FCB6 | |
| PROG3 | 0x7300_0000 | 0x73FF_FFFF | 0x03FF_FCB6 | |
| PROG4 | 0x7400_0000 | 0x74FF_FFFF | 0x03FF_FCB6 | |
| PROG5 | 0x7500_0000 | 0x75FF_FFFF | 0x03FF_FCB6 | |
| PROG6 | 0x7600_0000 | 0x76FF_FFFF | 0x03FF_FCB6 | |
| PROG7 | 0x7700_0000 | 0x77FF_FFFF | 0x03FF_FCB6 | |
| PROG8 | 0x7800_0000 | 0x78FF_FFFF | 0x03FF_FCB6 | |
| PROG9 | 0x7900_0000 | 0x79FF_FFFF | 0x03FF_FCB6 | |
| PROG10 | 0x7A00_0000 | 0x7AFF_FFFF | 0x03FF_FCB6 | |
| PROG11 | 0x7B00_0000 | 0x7BFF_FFFF | 0x03FF_FCB6 | |
| PROG12 | 0x7C00_0000 | 0x7CFF_FFFF | 0x03FF_FCB6 | |
| PROG13 | 0x7D00_0000 | 0x7DFF_FFFF | 0x03FF_FCB6 | |
| PROG14 | 0x7E00_0000 | 0x7EFF_FFFF | 0x03FF_FCB6 | |
| PROG15 | 0x7F00_0000 | 0x7FFF_FFFF | 0x03FF_FCB6 | |

8.10 DDR3 Memory Controller

The 32-bit DDR3 Memory Controller bus of the C665x is used to interface to JEDEC-standard-compliant DDR3 SDRAM devices. The DDR3 external bus interfaces only to DDR3 SDRAM devices; it does not share the bus with any other types of peripherals.

8.10.1 DDR3 Memory Controller Device-Specific Information

The C665x includes one 32-bit wide 1.5-V DDR3 SDRAM EMIF interface. The DDR3 interface can operate at 800 Mega transfers per second (MTS), 1033 MTS, and 1333 MTS.

Due to the complicated nature of the interface, a limited number of topologies will be supported to provide a 16-bit or 32-bit interface.

The DDR3 electrical requirements are fully specified in the DDR Jedec Specification JESD79-3C. Standard DDR3 SDRAMs are available in 8- and 16-bit versions, allowing for the following bank topologies to be supported by the interface:

- 36-bit: Three 16-bit SDRAMs (including 4 bits of ECC)
- 36-bit: Five 8-bit SDRAMs (including 4 bits of ECC)
- 32-bit: Two 16-bit SDRAMs
- 32-bit: Four 8-bit SDRAMs
- 16-bit: One 16-bit SDRAM
- 16-bit: Two 8-bit SDRAM

The approach to specifying interface timing for the DDR3 memory bus is different than on other interfaces such as I²C or SPI. For these other interfaces, the device timing was specified in terms of data manual specifications and I/O buffer information specification (IBIS) models. For the DDR3 memory bus, the approach is to specify compatible DDR3 devices and provide the printed circuit board (PCB) solution and guidelines directly to the user.

A race condition may exist when certain masters write data to the DDR3 memory controller. For example, if master A passes a software message via a buffer in external memory and does not wait for an indication that the write completes, before signaling to master B that the message is ready, when master B attempts to read the software message, then the master B read may bypass the master A write and, thus, master B may read stale data and, therefore, receive an incorrect message.

Some master peripherals (e.g., EDMA3 transfer controllers with TCCMOD=0) will always wait for the write to complete before signaling an interrupt to the system, thus avoiding this race condition. For masters that do not have a hardware specification of write-read ordering, it may be necessary to specify data ordering via software.

If master A does not wait for indication that a write is complete, it must perform the following workaround:

1. Perform the required write to DDR3 memory space.
2. Perform a dummy write to the DDR3 memory controller module ID and revision register.
3. Perform a dummy read from the DDR3 memory controller module ID and revision register.
4. Indicate to master B that the data is ready to be read after completion of the read in step 3. The completion of the read in step 3 ensures that the previous write was done.

8.10.2 DDR3 Memory Controller Electrical Data/Timing

The *KeyStone DSP DDR3 Implementation Guidelines* ([SPRABI1](#)) specifies a complete DDR3 interface solution as well as a list of compatible DDR3 devices. The DDR3 electrical requirements are fully specified in the DDR3 Jedec Specification JESD79-3C. TI has performed the simulation and system characterization to ensure all DDR3 interface timings in this solution are met; therefore, no electrical data/timing information is supplied here for this interface.

NOTE

TI supports **only** designs that follow the board design guidelines outlined in the application report.

8.11 I²C Peripheral

The inter-integrated circuit (I²C) module provides an interface between DSP and other devices compliant with Philips Semiconductors Inter-IC bus (I²C bus) specification version 2.1 and connected by way of an I²C bus. External components attached to this 2-wire serial bus can transmit/receive up to 8-bit data to/from the DSP through the I²C module.

8.11.1 I²C Device-Specific Information

The C665x device includes an I²C peripheral module.

NOTE

When using the I²C module, ensure there are external pullup resistors on the SDA and SCL pins.

The I²C modules on the C665x may be used by the DSP to control local peripheral ICs (DACs, ADCs, etc.) or may be used to communicate with other controllers in a system or to implement a user interface.

The I²C port is compatible with Philips I²C specification revision 2.1 (January 2000) and supports:

- Fast mode up to 400 Kbps (no fail-safe I/O buffers)
- Noise filter to remove noise 50 ns or less
- 7-bit and 10-bit device addressing modes
- Multi-master (transmit/receive) and slave (transmit/receive) functionality
- Events: DMA, interrupt, or polling
- Slew-rate limited open-drain output buffers

[Figure 8-31](#) shows a block diagram of the I²C module.

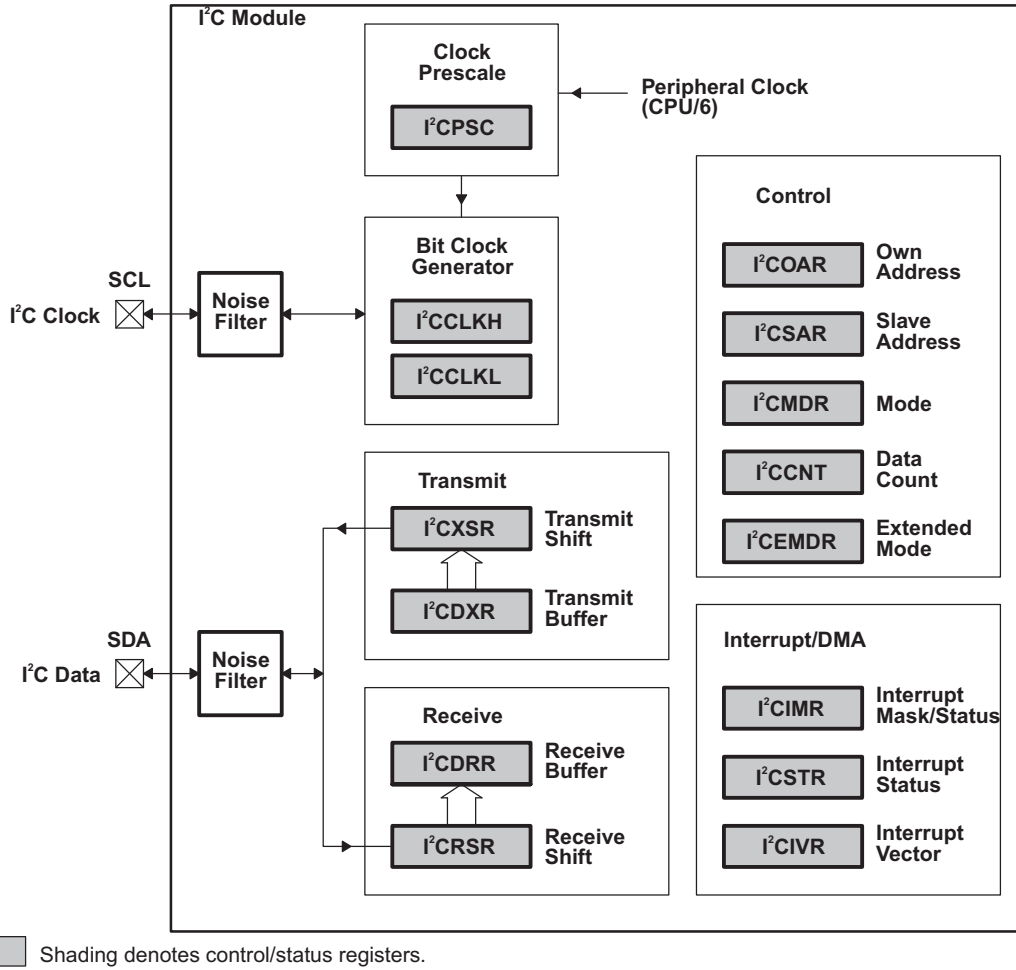


Figure 8-31. I²C Module Block Diagram

8.11.2 I²C Peripheral Register Description(s)

Table 8-61. I²C Registers

| HEX ADDRESS RANGE | REGISTER | REGISTER NAME |
|-----------------------|----------|--|
| 0253 0000 | ICOAR | I ² C Own Address Register |
| 0253 0004 | ICIMR | I ² C Interrupt Mask/Status Register |
| 0253 0008 | ICSTR | I ² C Interrupt Status Register |
| 0253 000C | ICCLKL | I ² C Clock Low-Time Divider Register |
| 0253 0010 | ICCLKH | I ² C Clock High-Time Divider Register |
| 0253 0014 | ICCNT | I ² C Data Count Register |
| 0253 0018 | ICDRR | I ² C Data Receive Register |
| 0253 001C | ICSAR | I ² C Slave Address Register |
| 0253 0020 | ICDXR | I ² C Data Transmit Register |
| 0253 0024 | ICMDR | I ² C Mode Register |
| 0253 0028 | ICIVR | I ² C Interrupt Vector Register |
| 0253 002C | ICEMDR | I ² C Extended Mode Register |
| 0253 0030 | ICPSC | I ² C Prescaler Register |
| 0253 0034 | ICPID1 | I ² C Peripheral Identification Register 1 [Value: 0x0000 0105] |
| 0253 0038 | ICPID2 | I ² C Peripheral Identification Register 2 [Value: 0x0000 0005] |
| 0253 003C - 0253 007F | - | Reserved |

8.11.3 I²C Electrical Data/Timing

8.11.3.1 Inter-Integrated Circuits (I²C) Timing

8.12 I²C Timing Requirements⁽¹⁾

(see Figure 8-32)

| NO. | | | STANDARD MODE | | FAST MODE | | UNIT S |
|-----|----------------------|---|------------------|------|------------------------------|--------------------|-----------|
| | | | MIN | MAX | MIN | MAX | |
| 1 | $t_{c(SCL)}$ | Cycle time, SCL | 10 | | 2.5 | | μ s |
| 2 | $t_{su(SCLH-SDAL)}$ | Setup time, SCL high before SDA low (for a repeated START condition) | 4.7 | | 0.6 | | μ s |
| 3 | $t_{h(SDAL-SCLL)}$ | Hold time, SCL low after SDA low (for a START and a repeated START condition) | 4 | | 0.6 | | μ s |
| 4 | $t_{w(SCLL)}$ | Pulse duration, SCL low | 4.7 | | 1.3 | | μ s |
| 5 | $t_{w(SCLH)}$ | Pulse duration, SCL high | 4 | | 0.6 | | μ s |
| 6 | $t_{su(SDAV-SCLH)}$ | Setup time, SDA valid before SCL high | 250 | | 100 ⁽²⁾ | | ns |
| 7 | $t_{h(SCLL-SDAV)}$ | Hold time, SDA valid after SCL low (For I ² C bus devices) | 0 ⁽³⁾ | 3.45 | 0 ⁽³⁾ | 0.9 ⁽⁴⁾ | μ s |
| 8 | $t_{w(SDAH)}$ | Pulse duration, SDA high between STOP and START conditions | 4.7 | | 1.3 | | μ s |
| 9 | $t_r(SDA)$ | Rise time, SDA | | 1000 | $20 + 0.1C_b$ ⁽⁵⁾ | 300 | ns |
| 10 | $t_r(SCL)$ | Rise time, SCL | | 1000 | $20 + 0.1C_b$ ⁽⁵⁾ | 300 | ns |
| 11 | $t_f(SDA)$ | Fall time, SDA | | 300 | $20 + 0.1C_b$ ⁽⁵⁾ | 300 | ns |
| 12 | $t_f(SCL)$ | Fall time, SCL | | 300 | $20 + 0.1C_b$ ⁽⁵⁾ | 300 | ns |
| 13 | $t_{su(SCLH-SDAH)}$ | Setup time, SCL high before SDA high (for STOP condition) | 4 | | 0.6 | | μ s |
| 14 | $t_w(SP)$ | Pulse duration, spike (must be suppressed) | | | 0 | 50 | ns |
| 15 | C_b ⁽⁵⁾ | Capacitive load for each bus line | | 400 | | 400 | pF |

- (1) The I²C pins SDA and SCL do not feature fail-safe I/O buffers. These pins could potentially draw current when the device is powered down.
- (2) A Fast-mode I²C-bus™ device can be used in a Standard-mode I²C-bus™ system, but the requirement $t_{su(SDA-SCLH)} \geq 250$ ns must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCL signal. If such a device does stretch the LOW period of the SCL signal, it must output the next data bit to the SDA line $t_r \text{ max} + t_{su(SDA-SCLH)} = 1000 + 250 = 1250$ ns (according to the Standard-mode I²C-Bus Specification) before the SCL line is released.
- (3) A device must internally provide a hold time of at least 300 ns for the SDA signal (referred to the V_{IHmin} of the SCL signal) to bridge the undefined region of the falling edge of SCL.
- (4) The maximum $t_{h(SDA-SCLL)}$ has only to be met if the device does not stretch the low period [$t_{w(SCLL)}$] of the SCL signal.
- (5) C_b = total capacitance of one bus line in pF. If mixed with HS-mode devices, faster fall-times are allowed.

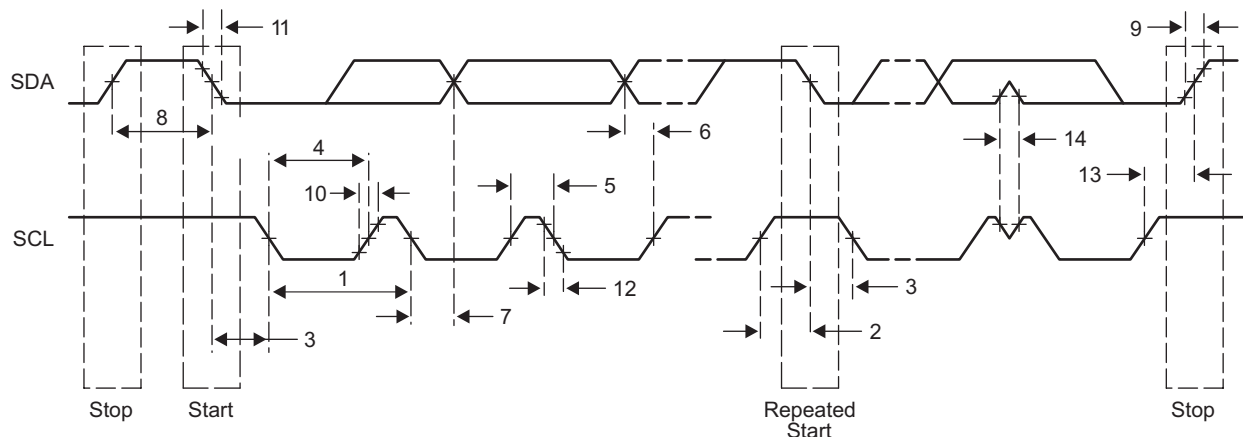


Figure 8-32. I²C Receive Timings

Table 8-62. I²C Switching Characteristics⁽¹⁾

(see Figure 8-33)

| NO. | PARAMETER | STANDARD MODE | | FAST MODE | | UNIT |
|-----|---|---------------|------|---------------------------------------|-----|------|
| | | MIN | MAX | MIN | MAX | |
| 16 | t _{c(SCL)} Cycle time, SCL | 10 | | 2.5 | | ms |
| 17 | t _{su(SCLH-SDAL)} Setup time, SCL high to SDA low (for a repeated START condition) | 4.7 | | 0.6 | | ms |
| 18 | t _{h(SDAL-SCLL)} Hold time, SDA low after SCL low (for a START and a repeated START condition) | 4 | | 0.6 | | ms |
| 19 | t _{w(SCLL)} Pulse duration, SCL low | 4.7 | | 1.3 | | ms |
| 20 | t _{w(SCLH)} Pulse duration, SCL high | 4 | | 0.6 | | ms |
| 21 | t _{d(SDAV-SDLH)} Delay time, SDA valid to SCL high | 250 | | 100 | | ns |
| 22 | t _{v(SDLL-SDAV)} Valid time, SDA valid after SCL low (For I ² C bus devices) | 0 | | 0 | 0.9 | ms |
| 23 | t _{w(SDAH)} Pulse duration, SDA high between STOP and START conditions | 4.7 | | 1.3 | | ms |
| 24 | t _{r(SDA)} Rise time, SDA | | 1000 | 20 + 0.1C _b ⁽¹⁾ | 300 | ns |
| 25 | t _{r(SCL)} Rise time, SCL | | 1000 | 20 + 0.1C _b ⁽¹⁾ | 300 | ns |
| 26 | t _{f(SDA)} Fall time, SDA | | 300 | 20 + 0.1C _b ⁽¹⁾ | 300 | ns |
| 27 | t _{f(SCL)} Fall time, SCL | | 300 | 20 + 0.1C _b ⁽¹⁾ | 300 | ns |
| 28 | t _{d(SCLH-SDAH)} Delay time, SCL high to SDA high (for STOP condition) | 4 | | 0.6 | | ms |
| 29 | C _p Capacitance for each I ² C pin | | 10 | | 10 | pF |

(1) C_b = total capacitance of one bus line in pF. If mixed with HS-mode devices, faster fall-times are allowed.

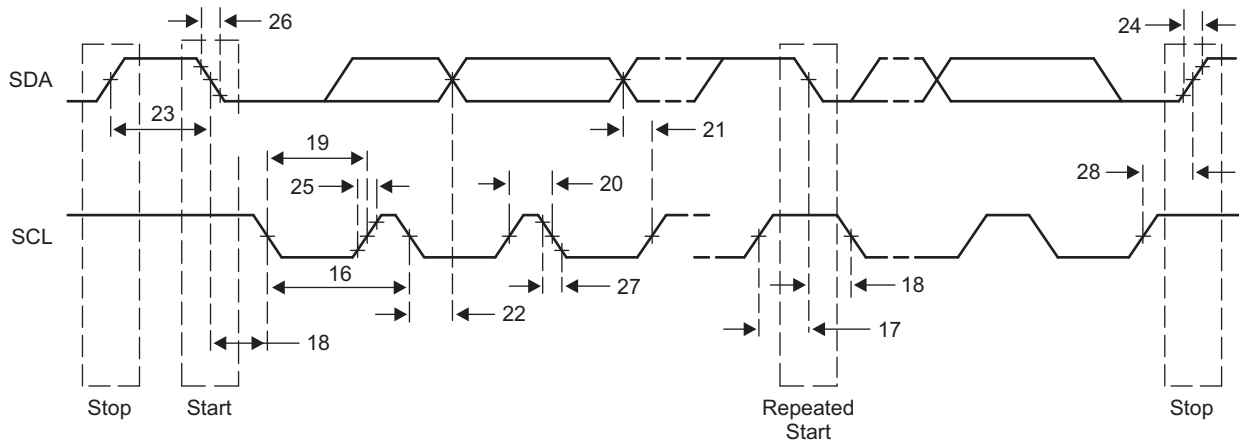


Figure 8-33. I²C Transmit Timings

8.13 SPI Peripheral

The serial peripheral interconnect (SPI) module provides an interface between the DSP and other SPI-compliant devices. The primary intent of this interface is to allow for connection to an SPI ROM for boot. The SPI module on the C665x is supported only in master mode. Additional chip-level components can also be included, such as temperature sensors or an I/O expander.

8.13.1 SPI Electrical Data/Timing

8.13.1.1 SPI Timing

Table 8-63. SPI Timing Requirements

(See [Figure 8-34](#))

| NO. | | | MIN | MAX | UNIT |
|--|--------------|--|-----|-----|------|
| Master Mode Timing Diagrams — Base Timings for 3 Pin Mode | | | | | |
| 7 | tsu(SDI-SPC) | Input Setup Time, SPIDIN valid before receive edge of SPICLK. Polarity = 0 Phase = 0 | 2 | | ns |
| 7 | tsu(SDI-SPC) | Input Setup Time, SPIDIN valid before receive edge of SPICLK. Polarity = 0 Phase = 1 | 2 | | ns |
| 7 | tsu(SDI-SPC) | Input Setup Time, SPIDIN valid before receive edge of SPICLK. Polarity = 1 Phase = 0 | 2 | | ns |
| 7 | tsu(SDI-SPC) | Input Setup Time, SPIDIN valid before receive edge of SPICLK. Polarity = 1 Phase = 1 | 2 | | ns |
| 8 | th(SPC-SDI) | Input Hold Time, SPIDIN valid after receive edge of SPICLK. Polarity = 0 Phase = 0 | 5 | | ns |
| 8 | th(SPC-SDI) | Input Hold Time, SPIDIN valid after receive edge of SPICLK. Polarity = 0 Phase = 1 | 5 | | ns |
| 8 | th(SPC-SDI) | Input Hold Time, SPIDIN valid after receive edge of SPICLK. Polarity = 1 Phase = 0 | 5 | | ns |
| 8 | th(SPC-SDI) | Input Hold Time, SPIDIN valid after receive edge of SPICLK. Polarity = 1 Phase = 1 | 5 | | ns |

Table 8-64. SPI Switching Characteristics

 (See [Figure 8-34](#) and [Figure 8-35](#))

| NO. | PARAMETER | | MIN | MAX | UNIT |
|---|--------------|---|-----------------------------------|-----------------------------------|------|
| Master Mode Timing Diagrams — Base Timings for 3 Pin Mode | | | | | |
| 1 | tc(SPC) | Cycle Time, SPICLK, All Master Modes | $3 \cdot P2^{(1)}$ | | ns |
| 2 | tw(SPCH) | Pulse Width High, SPICLK, All Master Modes | $0.5 \cdot tc - 1$ | | ns |
| 3 | tw(SPCL) | Pulse Width Low, SPICLK, All Master Modes | $0.5 \cdot tc - 1$ | | ns |
| 4 | td(SDO-SPC) | Setup (Delay), initial data bit valid on SPIDOUT to initial edge on SPICLK. Polarity = 0, Phase = 0. | | 5 | ns |
| 4 | td(SDO-SPC) | Setup (Delay), initial data bit valid on SPIDOUT to initial edge on SPICLK. Polarity = 0, Phase = 1. | | 5 | ns |
| 4 | td(SDO-SPC) | Setup (Delay), initial data bit valid on SPIDOUT to initial edge on SPICLK. Polarity = 1, Phase = 0 | | 5 | ns |
| 4 | td(SDO-SPC) | Setup (Delay), initial data bit valid on SPIDOUT to initial edge on SPICLK. Polarity = 1, Phase = 1 | | 5 | ns |
| 5 | td(SPC-SDO) | Setup (Delay), subsequent data bits valid on SPIDOUT to initial edge on SPICLK. Polarity = 0 Phase = 0 | | 2 | ns |
| 5 | td(SPC-SDO) | Setup (Delay), subsequent data bits valid on SPIDOUT to initial edge on SPICLK. Polarity = 0 Phase = 1 | | 2 | ns |
| 5 | td(SPC-SDO) | Setup (Delay), subsequent data bits valid on SPIDOUT to initial edge on SPICLK. Polarity = 1 Phase = 0 | | 2 | ns |
| 5 | td(SPC-SDO) | Setup (Delay), subsequent data bits valid on SPIDOUT to initial edge on SPICLK. Polarity = 1 Phase = 1 | | 2 | ns |
| 6 | toh(SPC-SDO) | Output hold time, SPIDOUT valid after receive edge of SPICLK except for final bit. Polarity = 0 Phase = 0 | $0.5 \cdot tc - 2$ | | ns |
| 6 | toh(SPC-SDO) | Output hold time, SPIDOUT valid after receive edge of SPICLK except for final bit. Polarity = 0 Phase = 1 | $0.5 \cdot tc - 2$ | | ns |
| 6 | toh(SPC-SDO) | Output hold time, SPIDOUT valid after receive edge of SPICLK except for final bit. Polarity = 1 Phase = 0 | $0.5 \cdot tc - 2$ | | ns |
| 6 | toh(SPC-SDO) | Output hold time, SPIDOUT valid after receive edge of SPICLK except for final bit. Polarity = 1 Phase = 1 | $0.5 \cdot tc - 2$ | | ns |
| Additional SPI Master Timings — 4 Pin Mode with Chip Select Option | | | | | |
| 19 | td(SCS-SPC) | Delay from $\overline{\text{SPISCS}}[n]$ active to first SPICLK. Polarity = 0 Phase = 0 | $2 \cdot P2 - 5$ | $2 \cdot P2 + 5$ | ns |
| 19 | td(SCS-SPC) | Delay from $\overline{\text{SPISCS}}[n]$ active to first SPICLK. Polarity = 0 Phase = 1 | $0.5 \cdot tc + (2 \cdot P2) - 5$ | $0.5 \cdot tc + (2 \cdot P2) + 5$ | ns |
| 19 | td(SCS-SPC) | Delay from $\overline{\text{SPISCS}}[n]$ active to first SPICLK. Polarity = 1 Phase = 0 | $2 \cdot P2 - 5$ | $2 \cdot P2 + 5$ | ns |
| 19 | td(SCS-SPC) | Delay from $\overline{\text{SPISCS}}[n]$ active to first SPICLK. Polarity = 1 Phase = 1 | $0.5 \cdot tc + (2 \cdot P2) - 5$ | $0.5 \cdot tc + (2 \cdot P2) + 5$ | ns |
| 20 | td(SPC-SCS) | Delay from final SPICLK edge to master deasserting $\overline{\text{SPISCS}}[n]$. Polarity = 0 Phase = 0 | $1 \cdot P2 - 5$ | $1 \cdot P2 + 5$ | ns |
| 20 | td(SPC-SCS) | Delay from final SPICLK edge to master deasserting $\overline{\text{SPISCS}}[n]$. Polarity = 0 Phase = 1 | $0.5 \cdot tc + (1 \cdot P2) - 5$ | $0.5 \cdot tc + (1 \cdot P2) + 5$ | ns |
| 20 | td(SPC-SCS) | Delay from final SPICLK edge to master deasserting $\overline{\text{SPISCS}}[n]$. Polarity = 1 Phase = 0 | $1 \cdot P2 - 5$ | $1 \cdot P2 + 5$ | ns |
| 20 | td(SPC-SCS) | Delay from final SPICLK edge to master deasserting $\overline{\text{SPISCS}}[n]$. Polarity = 1 Phase = 1 | $0.5 \cdot tc + (1 \cdot P2) - 5$ | $0.5 \cdot tc + (1 \cdot P2) + 5$ | ns |
| | tw(SCSH) | Minimum inactive time on $\overline{\text{SPISCS}}[n]$ pin between two transfers when $\overline{\text{SPISCS}}[n]$ is not held using the CSHOLD feature. | $2 \cdot P2 - 5$ | | ns |

 (1) $P2 = 1/\text{SYSCLK7}$

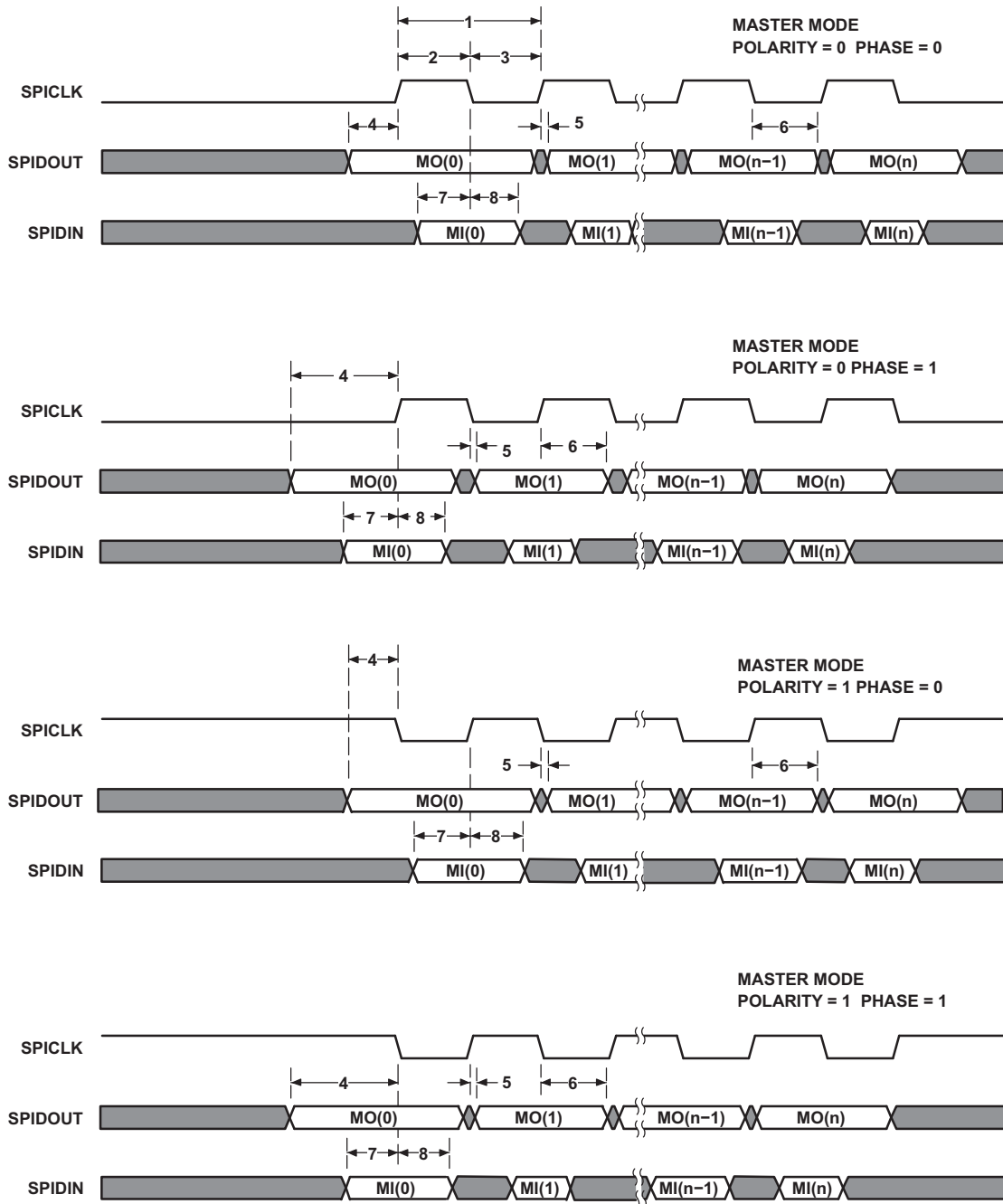


Figure 8-34. SPI Master Mode Timing Diagrams — Base Timings for 3 Pin Mode

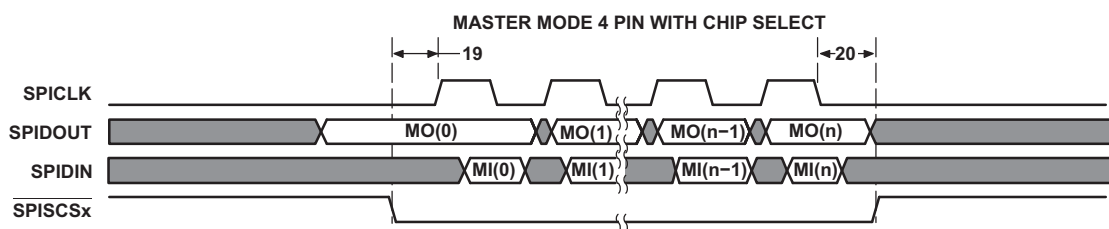


Figure 8-35. SPI Additional Timings for 4 Pin Master Mode with Chip Select Option

8.14 HyperLink Peripheral

The devices include the HyperLink bus for companion chip/die interfaces. This is a four-lane SerDes interface designed to operate at up to 10 Gbaud per lane. The supported data rates include 1.25 Gbaud, 3.125 Gbaud, 6.25 Gbaud, and 10 Gbaud. The interface is used to connect with external accelerators. The HyperLink links must be connected with DC coupling.

The interface includes the Serial Station Management Interfaces used to send power management and flow messages between devices. This consists of four LVCMOS inputs and four LVCMOS outputs configured as two 2-wire output buses and two 2-wire input buses. Each 2-wire bus includes a data signal and a clock signal.

8.14.1 HyperLink Device-Specific Interrupt Event

The HyperLink has 64 input events. Events 0 to 31 come from the chip-level interrupt controller and events 32 to 63 are from queue-pending signals from the Queue Manager to monitor some of the transmission queue status.

Table 8-65. HyperLink Events

| EVENT NUMBER | EVENT | EVENT DESCRIPTION |
|--------------|------------|-----------------------------|
| 0 | CIC2_OUT8 | Interrupt Controller output |
| 1 | CIC2_OUT9 | Interrupt Controller output |
| 2 | CIC2_OUT10 | Interrupt Controller output |
| 3 | CIC2_OUT11 | Interrupt Controller output |
| 4 | CIC2_OUT12 | Interrupt Controller output |
| 5 | CIC2_OUT13 | Interrupt Controller output |
| 6 | CIC2_OUT14 | Interrupt Controller output |
| 7 | CIC2_OUT15 | Interrupt Controller output |
| 8 | CIC2_OUT16 | Interrupt Controller output |
| 9 | CIC2_OUT17 | Interrupt Controller output |
| 10 | CIC2_OUT18 | Interrupt Controller output |
| 11 | CIC2_OUT19 | Interrupt Controller output |
| 12 | CIC2_OUT20 | Interrupt Controller output |
| 13 | CIC2_OUT21 | Interrupt Controller output |
| 14 | CIC2_OUT22 | Interrupt Controller output |
| 15 | CIC2_OUT23 | Interrupt Controller output |
| 16 | CIC2_OUT24 | Interrupt Controller output |
| 17 | CIC2_OUT25 | Interrupt Controller output |
| 18 | CIC2_OUT26 | Interrupt Controller output |
| 19 | CIC2_OUT27 | Interrupt Controller output |
| 20 | CIC2_OUT28 | Interrupt Controller output |
| 21 | CIC2_OUT29 | Interrupt Controller output |
| 22 | CIC2_OUT30 | Interrupt Controller output |
| 23 | CIC2_OUT31 | Interrupt Controller output |
| 24 | CIC2_OUT32 | Interrupt Controller output |
| 25 | CIC2_OUT33 | Interrupt Controller output |
| 26 | CIC2_OUT34 | Interrupt Controller output |
| 27 | CIC2_OUT35 | Interrupt Controller output |
| 28 | CIC2_OUT36 | Interrupt Controller output |
| 29 | CIC2_OUT37 | Interrupt Controller output |
| 30 | CIC2_OUT38 | Interrupt Controller output |
| 31 | CIC2_OUT39 | Interrupt Controller output |

Table 8-65. HyperLink Events (continued)

| EVENT NUMBER | EVENT | EVENT DESCRIPTION |
|--------------|-----------------|--------------------------|
| 32 | QM_INT_PEND_864 | Queue manager pend event |
| 33 | QM_INT_PEND_865 | Queue manager pend event |
| 34 | QM_INT_PEND_866 | Queue manager pend event |
| 35 | QM_INT_PEND_867 | Queue manager pend event |
| 36 | QM_INT_PEND_868 | Queue manager pend event |
| 37 | QM_INT_PEND_869 | Queue manager pend event |
| 38 | QM_INT_PEND_870 | Queue manager pend event |
| 39 | QM_INT_PEND_871 | Queue manager pend event |
| 40 | QM_INT_PEND_872 | Queue manager pend event |
| 41 | QM_INT_PEND_873 | Queue manager pend event |
| 42 | QM_INT_PEND_874 | Queue manager pend event |
| 43 | QM_INT_PEND_875 | Queue manager pend event |
| 44 | QM_INT_PEND_876 | Queue manager pend event |
| 45 | QM_INT_PEND_877 | Queue manager pend event |
| 46 | QM_INT_PEND_878 | Queue manager pend event |
| 47 | QM_INT_PEND_879 | Queue manager pend event |
| 48 | QM_INT_PEND_880 | Queue manager pend event |
| 49 | QM_INT_PEND_881 | Queue manager pend event |
| 50 | QM_INT_PEND_882 | Queue manager pend event |
| 51 | QM_INT_PEND_883 | Queue manager pend event |
| 52 | QM_INT_PEND_884 | Queue manager pend event |
| 53 | QM_INT_PEND_885 | Queue manager pend event |
| 54 | QM_INT_PEND_886 | Queue manager pend event |
| 55 | QM_INT_PEND_887 | Queue manager pend event |
| 56 | QM_INT_PEND_888 | Queue manager pend event |
| 57 | QM_INT_PEND_889 | Queue manager pend event |
| 58 | QM_INT_PEND_890 | Queue manager pend event |
| 59 | QM_INT_PEND_891 | Queue manager pend event |
| 60 | QM_INT_PEND_892 | Queue manager pend event |
| 61 | QM_INT_PEND_893 | Queue manager pend event |
| 62 | QM_INT_PEND_894 | Queue manager pend event |
| 63 | QM_INT_PEND_895 | Queue manager pend event |

8.14.2 HyperLink Electrical Data/Timing

The tables and figure below describe the timing requirements and switching characteristics of HyperLink peripheral.

Table 8-66. HyperLink Peripheral Timing Requirements

See [Figure 8-36](#), [Figure 8-37](#), [Figure 8-38](#)

| NO. | | | MIN | MAX | UNIT |
|---------------------|-----------------------------|--|--------|--------|------|
| FL Interface | | | | | |
| 1 | tc(MCMTXFLCLK) | Clock period - MCMTXFLCLK (C1) | 6.4 | | ns |
| 2 | tw(MCMTXFLCLKH) | High pulse width - MCMTXFLCLK | 0.4*C1 | 0.6*C1 | ns |
| 3 | tw(MCMTXFLCLKL) | Low pulse width - MCMTXFLCLK | 0.4*C1 | 0.6*C1 | ns |
| 6 | tsu(MCMTXFLDAT-MCMTXFLCLKH) | Setup time - MCMTXFLDAT valid before MCMTXFLCLK high | 1 | | ns |
| 7 | th(MCMTXFLCLKH-MCMTXFLDAT) | Hold time - MCMTXFLDAT valid after MCMTXFLCLK high | 1 | | ns |
| 6 | tsu(MCMTXFLDAT-MCMTXFLCLKL) | Setup time - MCMTXFLDAT valid before MCMTXFLCLK low | 1 | | ns |
| 7 | th(MCMTXFLCLKL-MCMTXFLDAT) | Hold time - MCMTXFLDAT valid after MCMTXFLCLK low | 1 | | ns |
| PM Interface | | | | | |
| 1 | tc(MCMRXPCLK) | Clock period - MCMRXPCLK (C3) | 6.4 | | ns |
| 2 | tw(MCMRXPCLK) | High pulse width - MCMRXPCLK | 0.4*C3 | 0.6*C3 | ns |
| 3 | tw(MCMRXPCLK) | Low pulse width - MCMRXPCLK | 0.4*C3 | 0.6*C3 | ns |
| 6 | tsu(MCMRXPMDAT-MCMRXPCLKH) | Setup time - MCMRXPMDAT valid before MCMRXPCLK high | 1 | | ns |
| 7 | th(MCMRXPCLKH-MCMRXPMDAT) | Hold time - MCMRXPMDAT valid after MCMRXPCLK high | 1 | | ns |
| 6 | tsu(MCMRXPMDAT-MCMRXPCLKL) | Setup time - MCMRXPMDAT valid before MCMRXPCLK low | 1 | | ns |
| 7 | th(MCMRXPCLKL-MCMRXPMDAT) | Hold time - MCMRXPMDAT valid after MCMRXPCLK low | 1 | | ns |

Table 8-67. HyperLink Peripheral Switching Characteristics

See [Figure 8-36](#), [Figure 8-37](#), [Figure 8-38](#)

| NO. | PARAMETER | | MIN | MAX | UNIT |
|---------------------|------------------------------|--|-------------|--------|------|
| FL Interface | | | | | |
| 1 | tc(MCMRXFLCLK) | Clock period - MCMRXFLCLK (C2) | 6.4 | | ns |
| 2 | tw(MCMRXFLCLKH) | High pulse width - MCMRXFLCLK | 0.4*C2 | 0.6*C2 | ns |
| 3 | tw(MCMRXFLCLKL) | Low pulse width - MCMRXFLCLK | 0.4*C2 | 0.6*C2 | ns |
| 4 | tosu(MCMRXFLDAT-MCMRXFLCLKH) | Setup time - MCMRXFLDAT valid before MCMRXFLCLK high | 0.25*C2-0.4 | | ns |
| 5 | toh(MCMRXFLCLKH-MCMRXFLDAT) | Hold time - MCMRXFLDAT valid after MCMRXFLCLK high | 0.25*C2-0.4 | | ns |
| 4 | tosu(MCMRXFLDAT-MCMRXFLCLKL) | Setup time - MCMRXFLDAT valid before MCMRXFLCLK low | 0.25*C2-0.4 | | ns |
| 5 | toh(MCMRXFLCLKL-MCMRXFLDAT) | Hold time - MCMRXFLDAT valid after MCMRXFLCLK low | 0.25*C2-0.4 | | ns |
| PM Interface | | | | | |
| 1 | tc(MCMTXPMCLK) | Clock period - MCMTXPMCLK (C4) | 6.4 | | ns |
| 2 | tw(MCMTXPMCLK) | High pulse width - MCMTXPMCLK | 0.4*C4 | 0.6*C4 | ns |
| 3 | tw(MCMTXPMCLK) | Low pulse width - MCMTXPMCLK | 0.4*C4 | 0.6*C4 | ns |
| 4 | tosu(MCMTXPMDAT-MCMTXPMCLKH) | Setup time - MCMTXPMDAT valid before MCMTXPMCLK high | 0.25*C4-0.4 | | ns |
| 5 | toh(MCMTXPMCLKH-MCMTXPMDAT) | Hold time - MCMTXPMDAT valid after MCMTXPMCLK high | 0.25*C4-0.4 | | ns |
| 4 | tosu(MCMTXPMDAT-MCMTXPMCLKL) | Setup time - MCMTXPMDAT valid before MCMTXPMCLK low | 0.25*C4-0.4 | | ns |
| 5 | toh(MCMTXPMCLKL-MCMTXPMDAT) | Hold time - MCMTXPMDAT valid after MCMTXPMCLK low | 0.25*C4-0.4 | | ns |

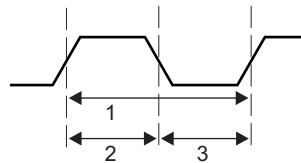
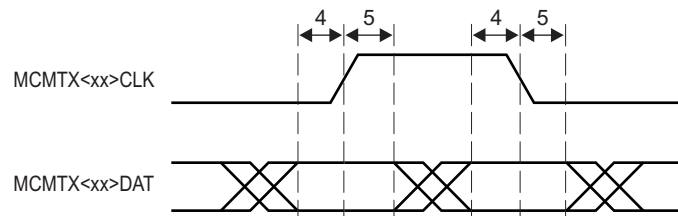
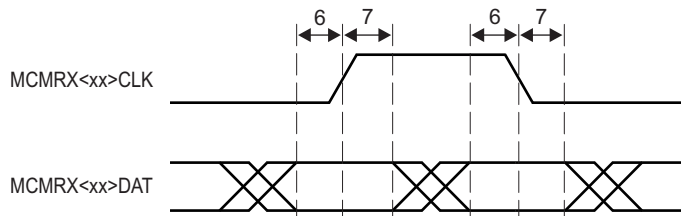


Figure 8-36. HyperLink Station Management Clock Timing



A. <xx> represents the interface that is being used: PM or FL

Figure 8-37. HyperLink Station Management Transmit Timing



A. <xx> represents the interface that is being used: PM or FL

Figure 8-38. HyperLink Station Management Receive Timing

8.15 UART Peripheral

The universal asynchronous receiver/transmitter (UART) module provides an interface between the DSP and a UART terminal interface or other UART-based peripheral. The UART is based on the industry standard TL16C550 asynchronous communications element, which, in turn, is a functional upgrade of the TL16C450. Functionally similar to the TL16C450 on power up (single character or TL16C450 mode), the UART can be placed in an alternate FIFO (TL16C550) mode. This relieves the DSP of excessive software overhead by buffering received and transmitted characters. The receiver and transmitter FIFOs store up to 16 bytes including three additional bits of error status per byte for the receiver FIFO.

The UART performs serial-to-parallel conversions on data received from a peripheral device and parallel-to-serial conversion on data received from the DSP. The DSP can read the UART status at any time. The UART includes control capability and a processor interrupt system that can be tailored to minimize software management of the communications link. For more information on UART, see the *Universal Asynchronous Receiver/Transmitter (UART) for KeyStone Devices User's Guide (SPRUGP1)*.

Table 8-68. UART Timing Requirements

(see Figure 8-39 and Figure 8-40)

| NO. | | MIN | MAX | UNIT |
|-------------------------------------|--|----------------------|-------------|------|
| Receive Timing | | | | |
| 4 | tw(RXSTART) Pulse width, receive start bit | 0.96U ⁽¹⁾ | 1.05U | ns |
| 5 | tw(RXH) Pulse width, receive data/parity bit high | 0.96U | 1.05U | ns |
| 5 | tw(RXL) Pulse width, receive data/parity bit low | 0.96U | 1.05U | ns |
| 6 | tw(RXSTOP1) Pulse width, receive stop bit 1 | 0.96U | 1.05U | ns |
| 6 | tw(RXSTOP15) Pulse width, receive stop bit 1.5 | 1.5*(0.96U) | 1.5*(1.05U) | ns |
| 6 | tw(RXSTOP2) Pulse width, receive stop bit 2 | 2*(0.96U) | 2*(1.05U) | ns |
| Autoflow Timing Requirements | | | | |
| 8 | td(CTSL-TX) Delay time, CTS asserted to START bit transmit | P ⁽²⁾ | 5P | ns |

(1) U = UART baud time = 1/programmed baud rate

(2) P = 1/SYSCLK7

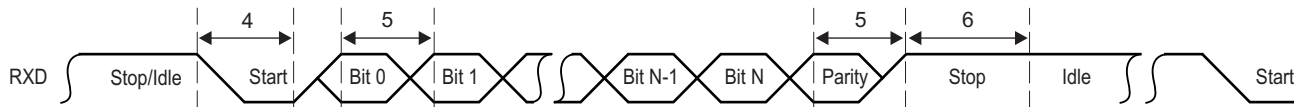


Figure 8-39. UART Receive Timing Waveform

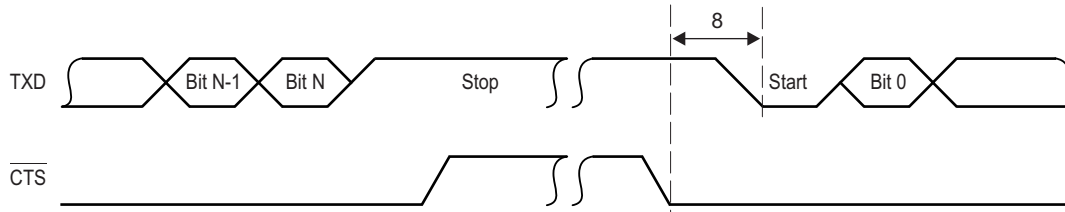


Figure 8-40. UART CTS (Clear-to-Send Input) — Autoflow Timing Waveform

Table 8-69. UART Switching Characteristics

(See Figure 8-41 and Figure 8-42)

| NO. | PARAMETER | | MIN | MAX | UNIT |
|-------------------------------------|--------------|---|-----------------|-----------------|------|
| Transmit Timing | | | | | |
| 1 | tw(TXSTART) | Pulse width, transmit start bit | $U^{(1)} - 2$ | $U + 2$ | ns |
| 2 | tw(TXH) | Pulse width, transmit data/parity bit high | $U - 2$ | $U + 2$ | ns |
| 2 | tw(TXL) | Pulse width, transmit data/parity bit low | $U - 2$ | $U + 2$ | ns |
| 3 | tw(TXSTOP1) | Pulse width, transmit stop bit 1 | $U - 2$ | $U + 2$ | ns |
| 3 | tw(TXSTOP15) | Pulse width, transmit stop bit 1.5 | $1.5 * (U - 2)$ | $1.5 * (U + 2)$ | ns |
| 3 | tw(TXSTOP2) | Pulse width, transmit stop bit 2 | $2 * (U - 2)$ | $2 * (U + 2)$ | ns |
| Autoflow Timing Requirements | | | | | |
| 7 | td(RX-RTSH) | Delay time, STOP bit received to RTS deasserted | $P^{(2)}$ | 5P | ns |

(1) U = UART baud time = $1/\text{programmed baud rate}$

(2) P = $1/\text{SYSCLK7}$

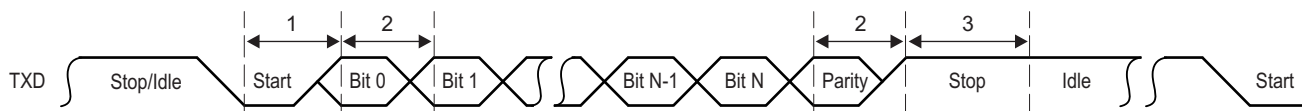


Figure 8-41. UART Transmit Timing Waveform

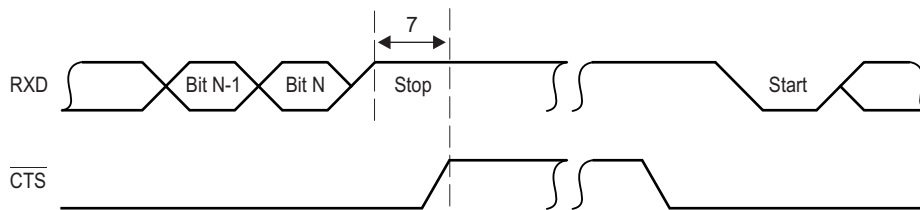


Figure 8-42. UART RTS (Request-to-Send Output) — Autoflow Timing Waveform

8.16 PCIe Peripheral

The two-lane PCI express (PCIe) module on the device provides an interface between the DSP and other PCIe-compliant devices. The PCI Express module provides low-pin-count, high-reliability, and high-speed data transfer at rates of 5.0 GBaud per lane on the serial links. For more information, see the *Peripheral Component Interconnect Express (PCIe) for KeyStone Devices User's Guide (SPRUGS6)*. The PCIe electrical requirements are fully specified in the PCI Express Base Specification Revision 2.0 of PCI-SIG. TI has performed the simulation and system characterization to ensure all PCIe interface timings in this solution are met; therefore, no electrical data/timing information is supplied here for this interface.

8.17 EMIF16 Peripheral

The EMIF16 module provides an interface between DSP and external memories such as NAND and NOR flash. For more information, see the *External Memory Interface (EMIF16) for KeyStone Devices User's Guide* ([SPRUGZ3](#)).

8.17.1 EMIF16 Electrical Data/Timing

Table 8-70. EMIF16 Asynchronous Memory Timing Requirements⁽¹⁾⁽²⁾

(see [Figure 8-43](#) and [Figure 8-44](#))

| NO. | | | MIN | MAX | UNIT |
|-----------------------|----------------------------------|---|---|---|------|
| General Timing | | | | | |
| 2 | $t_w(\text{WAIT})$ | Pulse duration, WAIT assertion and deassertion minimum time | | 2E | ns |
| 28 | $t_d(\text{WAIT-WEH})$ | Setup time, WAIT asserted before WE high | | 4E + 3 | ns |
| 14 | $t_d(\text{WAIT-OEH})$ | Setup time, WAIT asserted before OE high | | 4E + 3 | ns |
| Read Timing | | | | | |
| 3 | $t_c(\text{CEL})$ | EMIF read cycle time when ew = 0, meaning not in extended wait mode | $(\text{RS}+\text{RST}+\text{RH}+3) * \text{E}-3$ | $(\text{RS}+\text{RST}+\text{RH}+3) * \text{E}+3$ | ns |
| 3 | $t_c(\text{CEL})$ | EMIF read cycle time when ew = 1, meaning extended wait mode enabled | $(\text{RS}+\text{RST}+\text{WAIT}+\text{RH}+3) * \text{E}-3$ | $(\text{RS}+\text{RST}+\text{WAIT}+\text{RH}+3) * \text{E}+3$ | ns |
| 4 | $t_{\text{osu}}(\text{CEL-OEL})$ | Output setup time from CE low to OE low. SS = 0, not in select strobe mode | $(\text{RS}+1) * \text{E} - 3$ | $(\text{RS}+1) * \text{E} + 3$ | ns |
| 5 | $t_{\text{oh}}(\text{OEH-CEH})$ | Output hold time from OE high to CE high. SS = 0, not in select strobe mode | $(\text{RH}+1) * \text{E} - 3$ | $(\text{RH}+1) * \text{E} + 3$ | ns |
| 4 | $t_{\text{osu}}(\text{CEL-OEL})$ | Output setup time from CE low to OE low in select strobe mode, SS = 1 | $(\text{RS}+1) * \text{E} - 3$ | $(\text{RS}+1) * \text{E} + 3$ | ns |
| 5 | $t_{\text{oh}}(\text{OEH-CEH})$ | Output hold time from OE high to CE high in select strobe mode, SS = 1 | $(\text{RH}+1) * \text{E} - 3$ | $(\text{RH}+1) * \text{E} + 3$ | ns |
| 6 | $t_{\text{osu}}(\text{BAV-OEL})$ | Output setup time from BA valid to OE low | $(\text{RS}+1) * \text{E} - 3$ | $(\text{RS}+1) * \text{E} + 3$ | ns |
| 7 | $t_{\text{oh}}(\text{OEH-BAIV})$ | Output hold time from OE high to BA invalid | $(\text{RH}+1) * \text{E} - 3$ | $(\text{RH}+1) * \text{E} + 3$ | ns |
| 8 | $t_{\text{osu}}(\text{AV-OEL})$ | Output setup time from A valid to OE low | $(\text{RS}+1) * \text{E} - 3$ | $(\text{RS}+1) * \text{E} + 3$ | ns |
| 9 | $t_{\text{oh}}(\text{OEH-AIV})$ | Output hold time from OE high to A invalid | $(\text{RH}+1) * \text{E} - 3$ | $(\text{RH}+1) * \text{E} + 3$ | ns |
| 10 | $t_w(\text{OEL})$ | OE active time low, when ew = 0. Extended wait mode is disabled. | $(\text{RST}+1) * \text{E} - 3$ | $(\text{RST}+1) * \text{E} + 3$ | ns |
| 10 | $t_w(\text{OEL})$ | OE active time low, when ew = 1. Extended wait mode is enabled. | $(\text{RST}+1) * \text{E} - 3$ | $(\text{RST}+1) * \text{E} + 3$ | ns |
| 11 | $t_d(\text{WAITH-OEH})$ | Delay time from WAIT deasserted to OE# high | | 4E + 3 | ns |
| 12 | $t_{\text{su}}(\text{D-OEH})$ | Input setup time from D valid to OE high | 3 | | ns |
| 13 | $t_h(\text{OEH-D})$ | Input hold time from OE high to D invalid | 0.5 | | ns |
| Write Timing | | | | | |
| 15 | $t_c(\text{CEL})$ | EMIF write cycle time when ew = 0, meaning not in extended wait mode | $(\text{WS}+\text{WST}+\text{WH}+3) * \text{E}-3$ | $(\text{WS}+\text{WST}+\text{WH}+3) * \text{E}+3$ | ns |
| 15 | $t_c(\text{CEL})$ | EMIF write cycle time when ew = 1., meaning extended wait mode is enabled | $(\text{WS}+\text{WST}+\text{WAIT}+\text{WH}+3) * \text{E}-3$ | $(\text{WS}+\text{WST}+\text{WAIT}+\text{WH}+3) * \text{E}+3$ | ns |
| 16 | $t_{\text{osu}}(\text{CEL-WEL})$ | Output setup time from CE low to WE low. SS = 0, not in select strobe mode | $(\text{WS}+1) * \text{E} - 3$ | | ns |
| 17 | $t_{\text{oh}}(\text{WEH-CEH})$ | Output hold time from WE high to CE high. SS = 0, not in select strobe mode | $(\text{WH}+1) * \text{E} - 3$ | | ns |
| 16 | $t_{\text{osu}}(\text{CEL-WEL})$ | Output setup time from CE low to WE low in select strobe mode, SS = 1 | $(\text{WS}+1) * \text{E} - 3$ | | ns |
| 17 | $t_{\text{oh}}(\text{WEH-CEH})$ | Output hold time from WE high to CE high in select strobe mode, SS = 1 | $(\text{WH}+1) * \text{E} - 3$ | | ns |
| 18 | $t_{\text{osu}}(\text{RNW-WEL})$ | Output setup time from RNW valid to WE low | $(\text{WS}+1) * \text{E} - 3$ | | ns |
| 19 | $t_{\text{oh}}(\text{WEH-RNW})$ | Output hold time from WE high to RNW invalid | $(\text{WH}+1) * \text{E} - 3$ | | ns |
| 20 | $t_{\text{osu}}(\text{BAV-WEL})$ | Output setup time from BA valid to WE low | $(\text{WS}+1) * \text{E} - 3$ | | ns |
| 21 | $t_{\text{oh}}(\text{WEH-BAIV})$ | Output hold time from WE high to BA invalid | $(\text{WH}+1) * \text{E} - 3$ | | ns |
| 22 | $t_{\text{osu}}(\text{AV-WEL})$ | Output setup time from A valid to WE low | $(\text{WS}+1) * \text{E} - 3$ | | ns |
| 23 | $t_{\text{oh}}(\text{WEH-AIV})$ | Output hold time from WE high to A invalid | $(\text{WH}+1) * \text{E} - 3$ | | ns |
| 24 | $t_w(\text{WEL})$ | WE active time low, when ew = 0. Extended wait mode is disabled. | $(\text{WST}+1) * \text{E} - 3$ | | ns |
| 24 | $t_w(\text{WEL})$ | WE active time low, when ew = 1. Extended wait mode is enabled. | $(\text{WST}+1) * \text{E} - 3$ | | ns |
| 26 | $t_{\text{osu}}(\text{DV-WEL})$ | Output setup time from D valid to WE low | $(\text{WS}+1) * \text{E} - 3$ | | ns |
| 27 | $t_{\text{oh}}(\text{WEH-DIV})$ | Output hold time from WE high to D invalid | $(\text{WH}+1) * \text{E} - 3$ | | ns |
| 25 | $t_d(\text{WAITH-WEH})$ | Delay time from WAIT deasserted to WE# high | | 4E + 3 | ns |

(1) E = 1/SYSCLK7, RS = Read Setup, RST = Read Strobe, RH = Read Hold, WS = Write Setup, WST = Write Strobe, WH = Write Hold.

(2) WAIT = number of cycles wait is asserted between the programmed end of the strobe period and wait de-assertion.

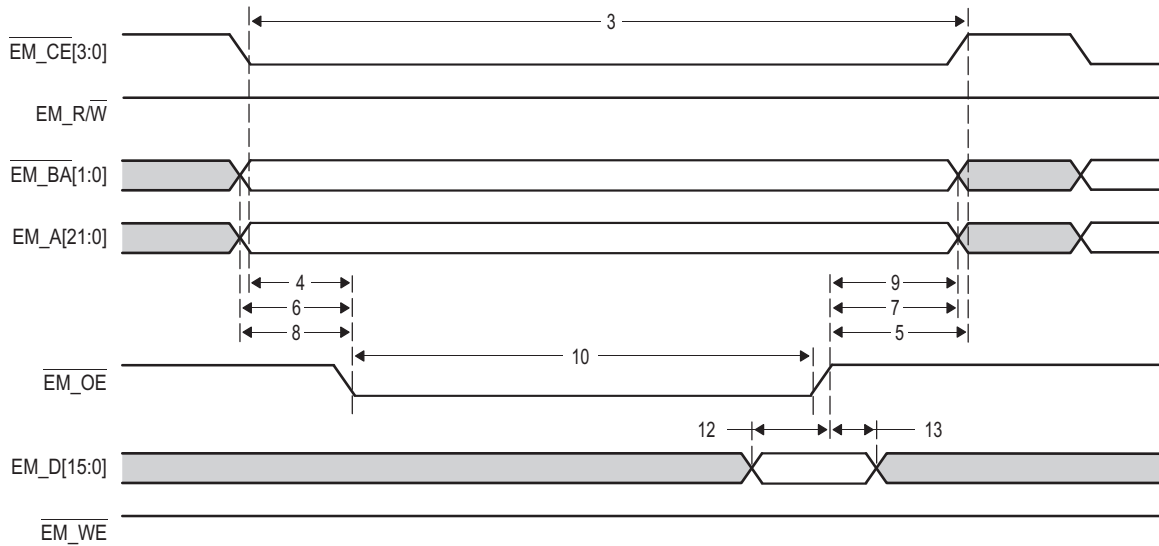


Figure 8-43. EMIF16 Asynchronous Memory Read Timing Diagram

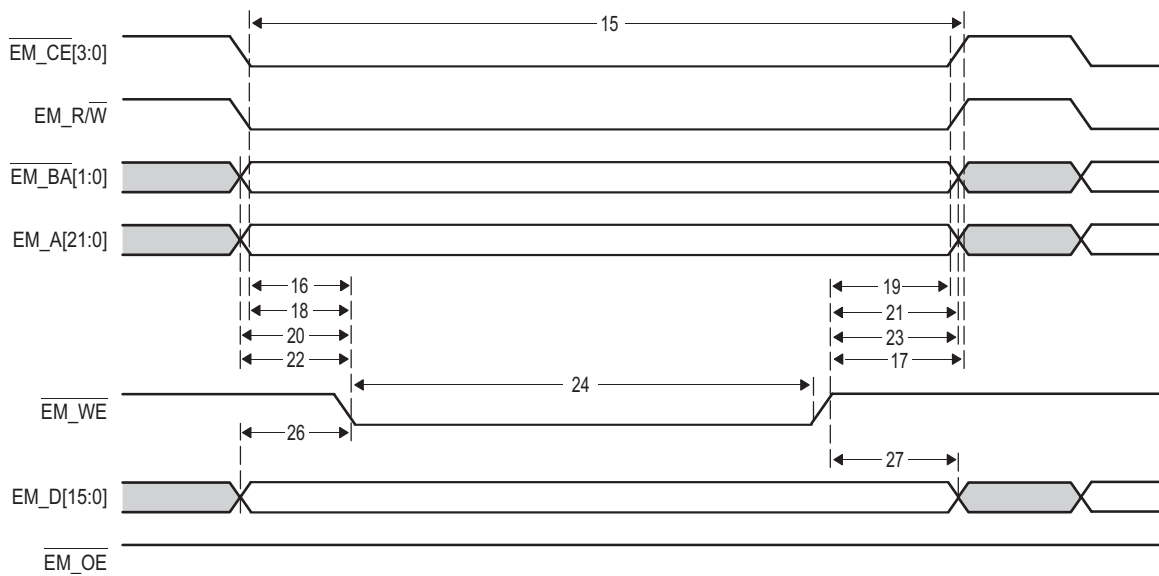


Figure 8-44. EMIF16 Asynchronous Memory Write Timing Diagram

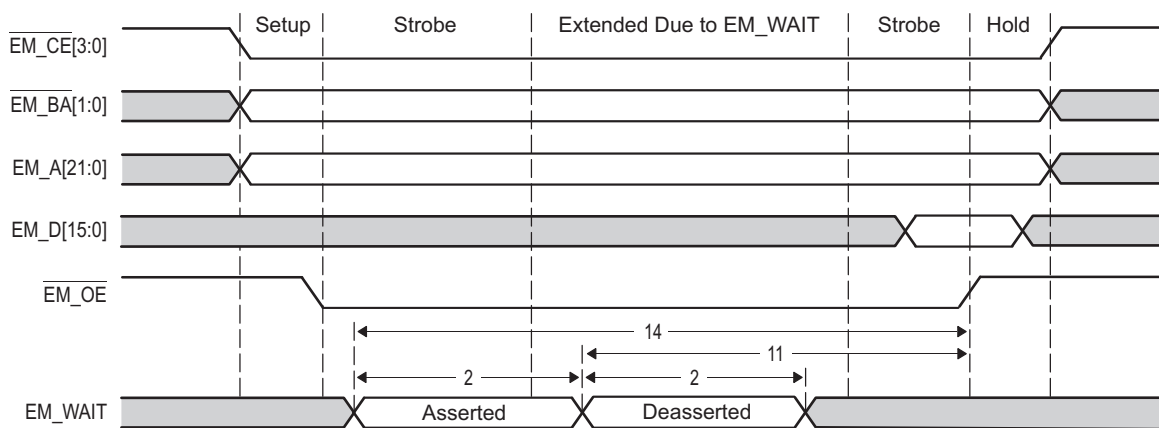


Figure 8-45. EMIF16 EM_WAIT Read Timing Diagram

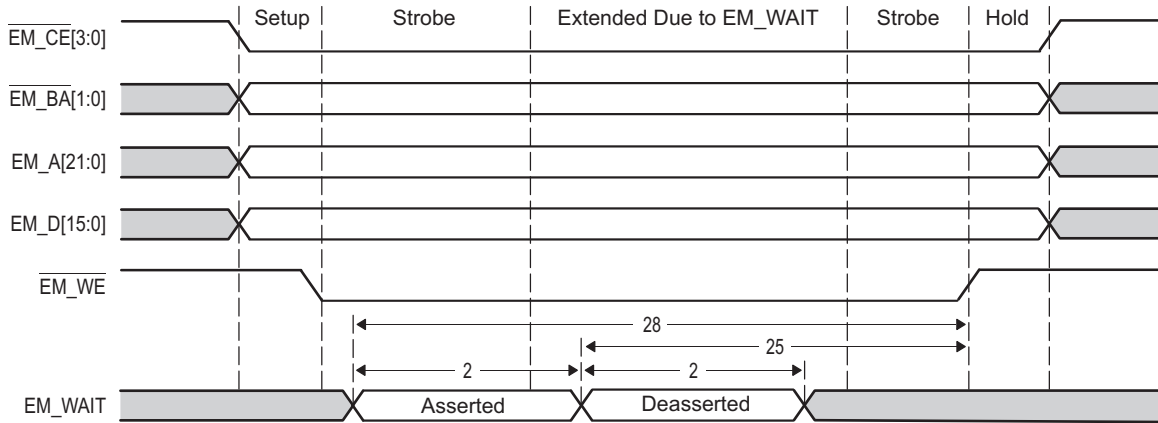


Figure 8-46. EMIF16 EM_WAIT Write Timing Diagram

8.18 Ethernet Media Access Controller (EMAC)

The Ethernet media access controller (EMAC) module provides an efficient interface between the C665x DSP core processor and the networked community. The EMAC supports 10Base-T (10 Mbps/second [Mbps]), and 100BaseTX (100 Mbps), in half- or full-duplex mode, and 1000BaseT (1000 Mbps) in full-duplex mode, with hardware flow control and quality-of-service (QoS) support.

The EMAC module conforms to the IEEE 802.3-2002 standard, describing the *Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer* specifications. The IEEE 802.3 standard has also been adopted by ISO/IEC and re-designated as ISO/IEC 8802-3:2000(E).

Deviating from this standard, the EMAC module does not use the transmit coding error signal MTXER. Instead of driving the error pin when an underflow condition occurs on a transmitted frame, the EMAC will intentionally generate an incorrect checksum by inverting the frame CRC, so that the transmitted frame will be detected as an error by the network.

The EMAC control module is the main interface between the device core processor, the MDIO module, and the EMAC module. The relationship between these three components is shown in [Figure 8-47](#). The EMAC control module contains the necessary components to allow the EMAC to make efficient use of device memory, plus it controls device interrupts. The EMAC control module incorporates 8K bytes of internal RAM to hold EMAC buffer descriptors.

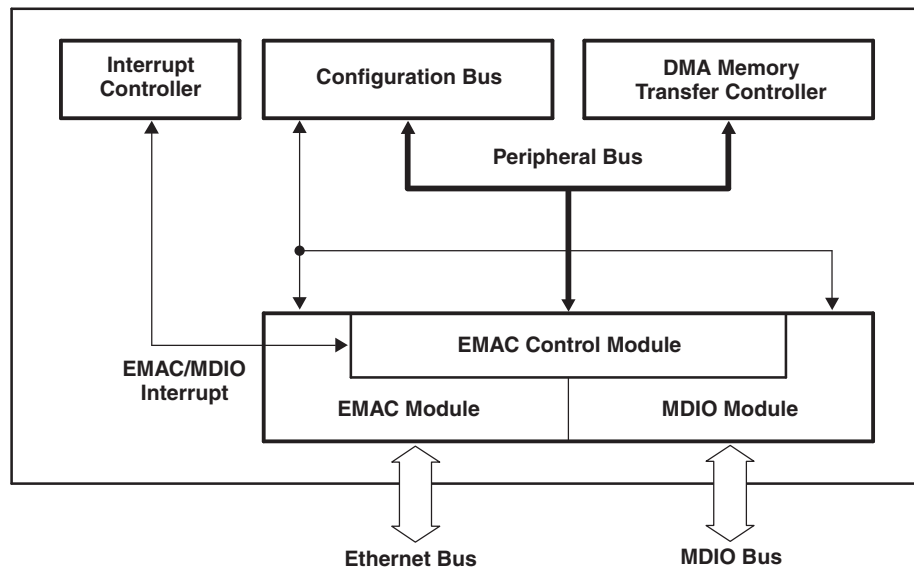


Figure 8-47. EMAC, MDIO, and EMAC Control Modules

For more detailed information on the EMAC/MDIO, see *Gigabit Ethernet (GbE) Subsystem for KeyStone Devices User's Guide* ([SPRUGV9](#)).

8.18.1 EMAC Device-Specific Information

The EMAC module on the device supports Serial Gigabit Media Independent Interface (SGMII). The SGMII interface conforms to version 1.8 of the industry standard specification.

8.18.2 EMAC Peripheral Register Description(s)

The memory maps of the EMAC are shown in [Table 8-71](#) through [Table 8-76](#).

Table 8-71. Ethernet MAC (EMAC) Control Registers

| HEX ADDRESS | ACRONYM | REGISTER NAME |
|-----------------------|-------------------|---|
| 02C0 8000 | TXIDVER | Transmit Identification and Version Register |
| 02C0 8004 | TXCONTROL | Transmit Control Register |
| 02C0 8008 | TXTEARDOWN | Transmit Teardown register |
| 02C0 800F | - | Reserved |
| 02C0 8010 | RXIDVER | Receive Identification and Version Register |
| 02C0 8014 | RXCONTROL | Receive Control Register |
| 02C0 8018 | RXTEARDOWN | Receive Teardown Register |
| 02C0 801C | - | Reserved |
| 02C0 8020 - 02C0 807C | - | Reserved |
| 02C0 8080 | TXINTSTATRAW | Transmit Interrupt Status (Unmasked) Register |
| 02C0 8084 | TXINTSTATMASKED | Transmit Interrupt Status (Masked) Register |
| 02C0 8088 | TXINTMASKSET | Transmit Interrupt Mask Set Register |
| 02C0 808C | TXINTMASKCLEAR | Transmit Interrupt Mask Clear Register |
| 02C0 8090 | MACINVECTOR | MAC Input Vector Register |
| 02C0 8094 | MACEOIVECTOR | MAC End of Interrupt Vector Register |
| 02C0 8098 - 02C0 819C | - | Reserved |
| 02C0 80A0 | RXINTSTATRAW | Receive Interrupt Status (Unmasked) Register |
| 02C0 80A4 | RXINTSTATMASKED | Receive Interrupt Status (Masked) Register |
| 02C0 80A8 | RXINTMASKSET | Receive Interrupt Mask Set Register |
| 02C0 80AC | RXINTMASKCLEAR | Receive Interrupt Mask Clear Register |
| 02C0 80B0 | MACINTSTATRAW | MAC Interrupt Status (Unmasked) Register |
| 02C0 80B4 | MACINTSTATMASKED | MAC Interrupt Status (Masked) Register |
| 02C0 80B8 | MACINTMASKSET | MAC Interrupt Mask Set Register |
| 02C0 80BC | MACINTMASKCLEAR | MAC Interrupt Mask Clear Register |
| 02C0 80C0 - 02C0 80FC | - | Reserved |
| 02C0 8100 | RXMBPENABLE | Receive Multicast/Broadcast/Promiscuous Channel Enable Register |
| 02C0 8104 | RXUNICASTSET | Receive Unicast Enable Set Register |
| 02C0 8108 | RXUNICASTCLEAR | Receive Unicast Clear Register |
| 02C0 810C | RXMAXLEN | Receive Maximum Length Register |
| 02C0 8110 | RXBUFFEROFFSET | Receive Buffer Offset Register |
| 02C0 8114 | RXFILTERLOWTHRESH | Receive Filter Low Priority Frame Threshold Register |
| 02C0 8118 - 02C0 811C | - | Reserved |
| 02C0 8120 | RX0FLOWTHRESH | Receive Channel 0 Flow Control Threshold Register |
| 02C0 8124 | RX1FLOWTHRESH | Receive Channel 1 Flow Control Threshold Register |
| 02C0 8128 | RX2FLOWTHRESH | Receive Channel 2 Flow Control Threshold Register |
| 02C0 812C | RX3FLOWTHRESH | Receive Channel 3 Flow Control Threshold Register |
| 02C0 8130 | RX4FLOWTHRESH | Receive Channel 4 Flow Control Threshold Register |
| 02C0 8134 | RX5FLOWTHRESH | Receive Channel 5 Flow Control Threshold Register |
| 02C0 8138 | RX6FLOWTHRESH | Receive Channel 6 Flow Control Threshold Register |
| 02C0 813C | RX7FLOWTHRESH | Receive Channel 7 Flow Control Threshold Register |
| 02C0 8140 | RX0FREEBUFFER | Receive Channel 0 Free Buffer Count Register |
| 02C0 8144 | RX1FREEBUFFER | Receive Channel 1 Free Buffer Count Register |
| 02C0 8148 | RX2FREEBUFFER | Receive Channel 2 Free Buffer Count Register |
| 02C0 814C | RX3FREEBUFFER | Receive Channel 3 Free Buffer Count Register |

Table 8-71. Ethernet MAC (EMAC) Control Registers (continued)

| HEX ADDRESS | ACRONYM | REGISTER NAME |
|-----------------------|---------------|--|
| 02C0 8150 | RX4FREEBUFFER | Receive Channel 4 Free Buffer Count Register |
| 02C0 8154 | RX5FREEBUFFER | Receive Channel 5 Free Buffer Count Register |
| 02C0 8158 | RX6FREEBUFFER | Receive Channel 6 Free Buffer Count Register |
| 02C0 815C | RX7FREEBUFFER | Receive Channel 7 Free Buffer Count Register |
| 02C0 8160 | MACCONTROL | MAC Control Register |
| 02C0 8164 | MACSTATUS | MAC Status Register |
| 02C0 8168 | EMCONTROL | Emulation Control Register |
| 02C0 816C | FIFOCONTROL | FIFO Control Register |
| 02C0 8170 | MACCONFIG | MAC Configuration Register |
| 02C0 8174 | SOFTRESET | Soft Reset Register |
| 02C0 81D0 | MACSRCADDRLO | MAC Source Address Low Bytes Register |
| 02C0 81D4 | MACSRCADDRHI | MAC Source Address High Bytes Register |
| 02C0 81D8 | MACHASH1 | MAC Hash Address Register 1 |
| 02C0 81DC | MACHASH2 | MAC Hash Address Register 2 |
| 02C0 81E0 | BOFFTEST | Back Off Test Register |
| 02C0 81E4 | TPACETEST | Transmit Pacing Algorithm Test Register |
| 02C0 81E8 | RXPAUSE | Receive Pause Timer Register |
| 02C0 81EC | TXPAUSE | Transmit Pause Timer Register |
| 02C0 8200 - 02C0 82FC | - | See Table 8-72 |
| 02C0 8300 - 02C0 84FC | - | Reserved |
| 02C0 8500 | MACADDRLO | MAC Address Low Bytes Register (used in Receive Address Matching) |
| 02C0 8504 | MACADDRHI | MAC Address High Bytes Register (used in Receive Address Matching) |
| 02C0 8508 | MACINDEX | MAC Index Register |
| 02C0 850C - 02C0 85FC | - | Reserved |
| 02C0 8600 | TX0HDP | Transmit Channel 0 DMA Head Descriptor Pointer Register |
| 02C0 8604 | TX1HDP | Transmit Channel 1 DMA Head Descriptor Pointer Register |
| 02C0 8608 | TX2HDP | Transmit Channel 2 DMA Head Descriptor Pointer Register |
| 02C0 860C | TX3HDP | Transmit Channel 3 DMA Head Descriptor Pointer Register |
| 02C0 8610 | TX4HDP | Transmit Channel 4 DMA Head Descriptor Pointer Register |
| 02C0 8614 | TX5HDP | Transmit Channel 5 DMA Head Descriptor Pointer Register |
| 02C0 8618 | TX6HDP | Transmit Channel 6 DMA Head Descriptor Pointer Register |
| 02C0 861C | TX7HDP | Transmit Channel 7 DMA Head Descriptor Pointer Register |
| 02C0 8620 | RX0HDP | Receive Channel 0 DMA Head Descriptor Pointer Register |
| 02C0 8624 | RX1HDP | Receive Channel 1 DMA Head Descriptor Pointer Register |
| 02C0 8628 | RX2HDP | Receive Channel 2 DMA Head Descriptor Pointer Register |
| 02C0 862C | RX3HDP | Receive Channel 3 DMA Head Descriptor Pointer Register |
| 02C0 8630 | RX4HDP | Receive Channel 4 DMA Head Descriptor Pointer Register |
| 02C0 8634 | RX5HDP | Receive Channel 5 DMA Head Descriptor Pointer Register |
| 02C0 8638 | RX6HDP | Receive Channel 6 DMA Head Descriptor Pointer Register |
| 02C0 863C | RX7HDP | Receive Channel 7 DMA Head Descriptor Pointer Register |
| 02C0 8640 | TX0CP | Transmit Channel 0 Completion Pointer (Interrupt Acknowledge) Register |
| 02C0 8644 | TX1CP | Transmit Channel 1 Completion Pointer (Interrupt Acknowledge) Register |
| 02C0 8648 | TX2CP | Transmit Channel 2 Completion Pointer (Interrupt Acknowledge) Register |
| 02C0 864C | TX3CP | Transmit Channel 3 Completion Pointer (Interrupt Acknowledge) Register |
| 02C0 8650 | TX4CP | Transmit Channel 4 Completion Pointer (Interrupt Acknowledge) Register |
| 02C0 8654 | TX5CP | Transmit Channel 5 Completion Pointer (Interrupt Acknowledge) Register |
| 02C0 8658 | TX6CP | Transmit Channel 6 Completion Pointer (Interrupt Acknowledge) Register |

Table 8-71. Ethernet MAC (EMAC) Control Registers (continued)

| HEX ADDRESS | ACRONYM | REGISTER NAME |
|-----------------------|---------|--|
| 02C0 865C | TX7CP | Transmit Channel 7 Completion Pointer (Interrupt Acknowledge) Register |
| 02C0 8660 | RX0CP | Receive Channel 0 Completion Pointer (Interrupt Acknowledge) Register |
| 02C0 8664 | RX1CP | Receive Channel 1 Completion Pointer (Interrupt Acknowledge) Register |
| 02C0 8668 | RX2CP | Receive Channel 2 Completion Pointer (Interrupt Acknowledge) Register |
| 02C0 866C | RX3CP | Receive Channel 3 Completion Pointer (Interrupt Acknowledge) Register |
| 02C0 8670 | RX4CP | Receive Channel 4 Completion Pointer (Interrupt Acknowledge) Register |
| 02C0 8674 | RX5CP | Receive Channel 5 Completion Pointer (Interrupt Acknowledge) Register |
| 02C0 8678 | RX6CP | Receive Channel 6 Completion Pointer (Interrupt Acknowledge) Register |
| 02C0 867C | RX7CP | Receive Channel 7 Completion Pointer (Interrupt Acknowledge) Register |
| 02C0 8680 - 02C0 86FC | - | Reserved |
| 02C0 8700 - 02C0 877C | - | Reserved |
| 02C0 8780 - 02C0 8FFF | - | Reserved |

Table 8-72. EMAC Statistics Registers

| HEX ADDRESS | ACRONYM | REGISTER NAME |
|-------------|-------------------|---|
| 02C0 8200 | RXGOODFRAMES | Good Receive Frames Register |
| 02C0 8204 | RXBCASTFRAMES | Broadcast Receive Frames Register (Total number of Good Broadcast Frames Receive) |
| 02C0 8208 | RXMCASTFRAMES | Multicast Receive Frames Register (Total number of Good Multicast Frames Received) |
| 02C0 820C | RXPAUSEFRAMES | Pause Receive Frames Register |
| 02C0 8210 | RXCRCERRORS | Receive CRC Errors Register (Total number of Frames Received with CRC Errors) |
| 02C0 8214 | RXALIGNCODEERRORS | Receive Alignment/Code Errors register (Total number of frames received with alignment/code errors) |
| 02C0 8218 | RXOVERSIZED | Receive Oversized Frames Register (Total number of Oversized Frames Received) |
| 02C0 821C | RXJABBER | Receive Jabber Frames Register (Total number of Jabber Frames Received) |
| 02C0 8220 | RXUNDERSIZED | Receive Undersized Frames Register (Total number of Undersized Frames Received) |
| 02C0 8224 | RXFRAGMENTS | Receive Frame Fragments Register |
| 02C0 8228 | RXFILTERED | Filtered Receive Frames Register |
| 02C0 822C | RXQOSFILTERERED | Received QOS Filtered Frames Register |
| 02C0 8230 | RXOCTETS | Receive Octet Frames Register (Total number of Received Bytes in Good Frames) |
| 02C0 8234 | TXGOODFRAMES | Good Transmit Frames Register (Total number of Good Frames Transmitted) |
| 02C0 8238 | TXBCASTFRAMES | Broadcast Transmit Frames Register |
| 02C0 823C | TXMCASTFRAMES | Multicast Transmit Frames Register |
| 02C0 8240 | TXPAUSEFRAMES | Pause Transmit Frames Register |
| 02C0 8244 | TXDEFERED | Deferred Transmit Frames Register |
| 02C0 8248 | TXCOLLISION | Transmit Collision Frames Register |
| 02C0 824C | TXSINGLECOLL | Transmit Single Collision Frames Register |
| 02C0 8250 | TXMULTICOLL | Transmit Multiple Collision Frames Register |
| 02C0 8254 | TXEXCESSIVECOLL | Transmit Excessive Collision Frames Register |
| 02C0 8258 | TXLATECOLL | Transmit Late Collision Frames Register |
| 02C0 825C | TXUNDERRUN | Transmit Under Run Error Register |
| 02C0 8260 | TXCARRIERSENSE | Transmit Carrier Sense Errors Register |
| 02C0 8264 | TXOCTETS | Transmit Octet Frames Register |
| 02C0 8268 | FRAME64 | Transmit and Receive 64 Octet Frames Register |

Table 8-72. EMAC Statistics Registers (continued)

| HEX ADDRESS | ACRONYM | REGISTER NAME |
|-----------------------|---------------|--|
| 02C0 826C | FRAME65T127 | Transmit and Receive 65 to 127 Octet Frames Register |
| 02C0 8270 | FRAME128T255 | Transmit and Receive 128 to 255 Octet Frames Register |
| 02C0 8274 | FRAME256T511 | Transmit and Receive 256 to 511 Octet Frames Register |
| 02C0 8278 | FRAME512T1023 | Transmit and Receive 512 to 1023 Octet Frames Register |
| 02C0 827C | FRAME1024TUP | Transmit and Receive 1024 to 1518 Octet Frames Register |
| 02C0 8280 | NETOCTETS | Network Octet Frames Register |
| 02C0 8284 | RXSOFOVERRUNS | Receive FIFO or DMA Start of Frame Overruns Register |
| 02C0 8288 | RXMOFOVERRUNS | Receive FIFO or DMA Middle of Frame Overruns Register |
| 02C0 828C | RXDMAOVERRUNS | Receive DMA Start of Frame and Middle of Frame Overruns Register |
| 02C0 8290 - 02C0 82FC | - | Reserved |

Table 8-73. EMAC Descriptor Memory

| HEX ADDRESS | ACRONYM | REGISTER NAME |
|-----------------------|---------|------------------------|
| 02C0 A000 - 02C0 BFFF | - | EMAC Descriptor Memory |

Table 8-74. SGMII Control Registers

| HEX ADDRESS | ACRONYM | REGISTER NAME |
|-----------------------|-------------------|--|
| 02C0 8900 | IDVER | Identification and Version register |
| 02C0 8904 | SOFT_RESET | Software Reset Register |
| 02C0 8910 | CONTROL | Control Register |
| 02C0 8914 | STATUS | Status Register |
| 02C0 8918 | MR_ADV_ABILITY | Advertised Ability Register |
| 02C0 891C | - | Reserved |
| 02C0 8920 | MR_LP_ADV_ABILITY | Link Partner Advertised Ability Register |
| 02C0 8924 - 02C0 8948 | - | Reserved |

Table 8-75. EMIC Control Registers

| HEX ADDRESS | ACRONYM | REGISTER NAME |
|-------------|-------------------|---|
| 02C0 8A00 | IDVER | Identification and Version register |
| 02C0 8A04 | SOFT_RESET | Software Reset Register |
| 02C0 8A08 | EM_CONTROL | Emulation Control Register |
| 02C0 8A0C | INT_CONTROL | Interrupt Control Register |
| 02C0 8A10 | C0_RX_THRESH_EN | Receive Threshold Interrupt Enable Register for CorePac0 |
| 02C0 8A14 | C0_RX_EN | Receive Interrupt Enable Register for CorePac0 |
| 02C0 8A18 | C0_TX_EN | Transmit Interrupt Enable Register for CorePac0 |
| 02C0 8A1C | C0_MISC_EN | Misc Interrupt Enable Register for CorePac0 |
| 02C0 8A10 | C1_RX_THRESH_EN | Receive Threshold Interrupt Enable Register for CorePac1 (C6657 only) |
| 02C0 8A14 | C1_RX_EN | Receive Interrupt Enable Register for CorePac1 (C6657 only) |
| 02C0 8A18 | C1_TX_EN | Transmit Interrupt Enable Register for CorePac1 (C6657 only) |
| 02C0 8A1C | C1_MISC_EN | Misc Interrupt Enable Register for CorePac1 (C6657 only) |
| 02C0 8A90 | C0_RX_THRESH_STAT | Receive Threshold Masked Interrupt Status Register for CorePac0 |
| 02C0 8A94 | C0_RX_STAT | Receive Interrupt Masked Interrupt Status Register for CorePac0 |
| 02C0 8A98 | C0_TX_STAT | Transmit Interrupt Masked Interrupt Status Register for CorePac0 |
| 02C0 8A9C | C0_MISC_STAT | Misc Interrupt Masked Interrupt Status Register for CorePac0 |
| 02C0 8AA0 | C1_RX_THRESH_STAT | Receive Threshold Masked Interrupt Status Register for CorePac1 (C6657 only) |
| 02C0 8AA4 | C1_RX_STAT | Receive Interrupt Masked Interrupt Status Register for CorePac1 (C6657 only) |
| 02C0 8AA8 | C1_TX_STAT | Transmit Interrupt Masked Interrupt Status Register for CorePac1 (C6657 only) |
| 02C0 8AAC | C1_MISC_STAT | Misc Interrupt Masked Interrupt Status Register for CorePac1 (C6657 only) |
| 02C0 8B10 | C0_RX_IMAX | Receive Interrupts Per Millisecond for CorePac0 |
| 02C0 8B14 | C0_TX_IMAX | Transmit Interrupts Per Millisecond for CorePac0 |
| 02C0 8B18 | C1_RX_IMAX | Receive Interrupts Per Millisecond for CorePac1 (C6657 only) |
| 02C0 8B1C | C1_TX_IMAX | Transmit Interrupts Per Millisecond for CorePac1 (C6657 only) |

8.18.3 EMAC Electrical Data/Timing (SGMII)

The *Hardware Design Guide for KeyStone Devices* ([SPRABI2](#)) specifies a complete EMAC and SGMII interface solution for the C665x as well as a list of compatible EMAC and SGMII devices. TI has performed the simulation and system characterization to ensure all EMAC and SGMII interface timings in this solution are met; therefore, no electrical data/timing information is supplied here for this interface.

NOTE

TI supports **only** designs that follow the board design guidelines outlined in the application report.

8.19 Management Data Input/Output (MDIO)

The management data input/output (MDIO) module implements the 802.3 serial management interface to interrogate and control up to 32 Ethernet PHY(s) connected to the device, using a shared two-wire bus. Application software uses the MDIO module to configure the auto-negotiation parameters of each PHY attached to the GbE switch subsystem, retrieve the negotiation results, and configure required parameters in the GbE switch subsystem module for correct operation. The module is designed to allow almost transparent operation of the MDIO interface, with very little maintenance from the core processor. For more information, see the *Gigabit Ethernet (GbE) Subsystem for KeyStone Devices User's Guide (SPRUGV9)*.

The EMAC control module is the main interface between the device core processor, the MDIO module, and the EMAC module. The relationship between these three components is shown in [Figure 8-47](#).

For more detailed information on the EMAC/MDIO, see *Gigabit Ethernet (GbE) Subsystem for KeyStone Devices User's Guide (SPRUGV9)*.

8.19.1 MDIO Peripheral Registers

The memory map of the MDIO is shown in [Table 8-76](#).

Table 8-76. MDIO Registers

| HEX ADDRESS | ACRONYM | REGISTER NAME |
|-----------------------|------------------|--|
| 02C0 8800 | VERSION | MDIO Version Register |
| 02C0 8804 | CONTROL | MDIO Control Register |
| 02C0 8808 | ALIVE | MDIO PHY Alive Status Register |
| 02C0 880C | LINK | MDIO PHY Link Status Register |
| 02C0 8810 | LINKINTRAW | MDIO link Status Change Interrupt (unmasked) Register |
| 02C0 8814 | LINKINTMASKED | MDIO link Status Change Interrupt (masked) Register |
| 02C0 8818 - 02C0 881C | - | Reserved |
| 02C0 8820 | USERINTRAW | MDIO User Command Complete Interrupt (Unmasked) Register |
| 02C0 8824 | USERINTMASKED | MDIO User Command Complete Interrupt (Masked) Register |
| 02C0 8828 | USERINTMASKSET | MDIO User Command Complete Interrupt Mask Set Register |
| 02C0 882C | USERINTMASKCLEAR | MDIO User Command Complete Interrupt Mask Clear Register |
| 02C0 8830 - 02C0 887C | - | Reserved |
| 02C0 8880 | USERACCESS0 | MDIO User Access Register 0 |
| 02C0 8884 | USERPHYSEL0 | MDIO User PHY Select Register 0 |
| 02C0 8888 | USERACCESS1 | MDIO User Access Register 1 |
| 02C0 888C | USERPHYSEL1 | MDIO User PHY Select Register 1 |
| 02C0 8890 - 02C0 8FFF | - | Reserved |

8.19.2 MDIO Timing

Table 8-77. MDIO Timing Requirements

See [Figure 8-48](#)

| NO. | | MIN | MAX | UNIT |
|-----|--|-----|-----|------|
| 1 | tc(MDCLK) Cycle time, MDCLK | 400 | | ns |
| 2 | tw(MDCLKH) Pulse duration, MDCLK high | 180 | | ns |
| 3 | tw(MDCLKL) Pulse duration, MDCLK low | 180 | | ns |
| 4 | tsu(MDIO-MDCLKH) Setup time, MDIO data input valid before MDCLK high | 10 | | ns |
| 5 | th(MDCLKH-MDIO) Hold time, MDIO data input valid after MDCLK high | 0 | | ns |
| | tt(MDCLK) Transition time, MDCLK | | 5 | ns |

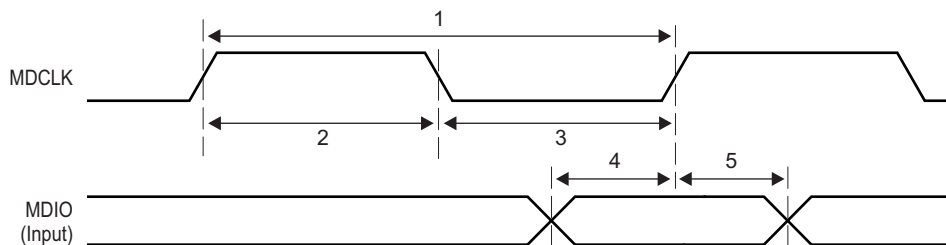


Figure 8-48. MDIO Input Timing

Table 8-78. MDIO Switching Characteristics

See [Figure 8-49](#)

| NO. | PARAMETER | MIN | MAX | UNIT |
|-----|---|-----|-----|------|
| 6 | td(MDCLKL-MDIO) Delay time, MDCLK low to MDIO data output valid | | 100 | ns |

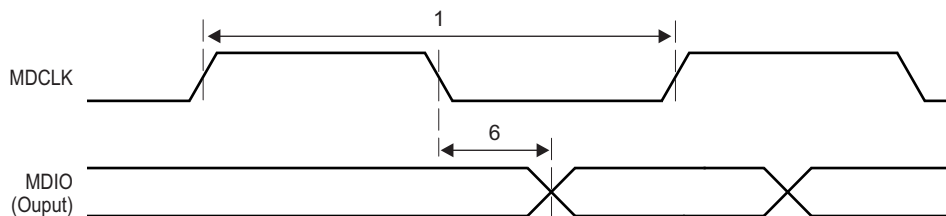


Figure 8-49. MDIO Output Timing

8.20 Timers

The timers can be used to: time events, count events, generate pulses, interrupt the CPU and send synchronization events to the EDMA3 channel controller.

8.20.1 Timers Device-Specific Information

The C665x devices have eight (C6657) or seven (C6655) 64-bit timers in total. On the C6657, Timer0 and Timer1 are dedicated to each of the two CorePacs as a watchdog timer and can also be used as general-purpose timers. Each of the other six timers can also be configured as a general-purpose timer only, with each timer programmed as a 64-bit timer or as two separate 32-bit timers. On the C6655, Timer0 is dedicated to the CorePac as a watchdog timer and can also be used as a general-purpose timer. Each of the other six timers can also be configured as a general-purpose timer only, programmed as a 64-bit timer or as two separate 32-bit timers.

When operating in 64-bit mode, the timer counts either VBUS clock cycles or input (TINPLx) pulses (rising edge) and generates an output pulse/waveform (TOUTLx) plus an internal event (TINTLx) on a software-programmable period.

When operating in 32-bit mode, the timer is split into two independent 32-bit timers. Each timer is made up of two 32-bit counters: a high counter and a low counter. The timer pins, TINPLx and TOUTLx are connected to the low counter. The timer pins, TINPHx and TOUTHx are connected to the high counter.

When operating in watchdog mode, the timer counts down to 0 and generates an event. It is a requirement that software writes to the timer before the count expires, after which the count begins again. If the count ever reaches 0, the timer event output is asserted. Reset initiated by a watchdog timer can be set by programming [Section 8.5.2.6](#) and the type of reset initiated can be set by programming [Section 8.5.2.8](#). For more information, see the *64-bit Timer (Timer 64) for KeyStone Devices User's Guide (SPRUGV5)*.

8.20.2 Timers Electrical Data/Timing

The tables and figure below describe the timing requirements and switching characteristics of Timer0 through Timer7 peripherals.

Table 8-79. Timer Input Timing Requirements⁽¹⁾

(see [Figure 8-50](#))

| NO. | PARAMETER | | MIN | MAX | UNIT |
|-----|----------------|----------------------|-----|-----|------|
| 1 | $t_{w(TINPH)}$ | Pulse duration, high | 12C | | ns |
| 2 | $t_{w(TINPL)}$ | Pulse duration, low | 12C | | ns |

(1) $C = 1 / \text{CORECLK(N|P)}$ frequency in ns.

Table 8-80. Timer Output Switching Characteristics⁽¹⁾

(see [Figure 8-50](#))

| NO. | PARAMETER | | MIN | MAX | UNIT |
|-----|----------------|----------------------|---------|-----|------|
| 3 | $t_{w(TOUTH)}$ | Pulse duration, high | 12C - 3 | | ns |
| 4 | $t_{w(TOURL)}$ | Pulse duration, low | 12C - 3 | | ns |

(1) $C = 1 / \text{CORECLK(N|P)}$ frequency in ns.

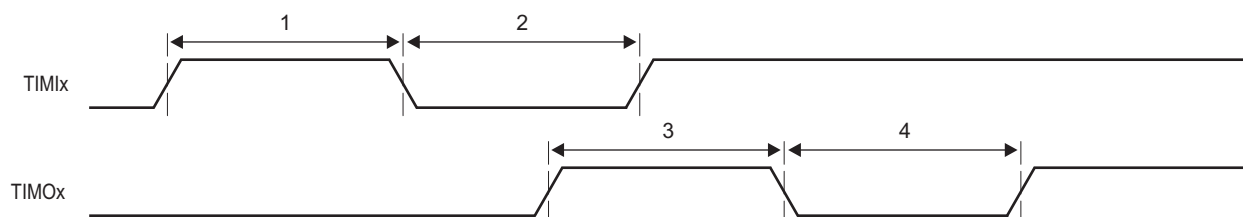


Figure 8-50. Timer Timing

8.21 General-Purpose Input/Output (GPIO)

8.21.1 GPIO Device-Specific Information

On the C665x, the GPIO peripheral pins GP[15:0] are also used to latch configuration settings. For more detailed information on device/peripheral configuration and the C665x device pin muxing, see [Section 4](#). For more information on GPIO, see the *General Purpose Input/Output (GPIO) for KeyStone Devices User's Guide* ([SPRUGV1](#)).

8.21.2 GPIO Electrical Data/Timing

Table 8-81. GPIO Input Timing Requirements

| NO. | PARAMETER | MIN | MAX | UNIT |
|-----|---|--------------------|-----|------|
| 1 | $t_{w(GPOH)}$ Pulse duration, GPOx high | 12C ⁽¹⁾ | | ns |
| 2 | $t_{w(GPOL)}$ Pulse duration, GPOx low | 12C | | ns |

(1) C = 1/SYSCLK1 frequency in ns.

Table 8-82. GPIO Output Switching Characteristics

| NO. | PARAMETER | MIN | MAX | UNIT |
|-----|---|------------------------|-----|------|
| 3 | $t_{w(GPOH)}$ Pulse duration, GPOx high | 36C ⁽¹⁾ - 8 | | ns |
| 4 | $t_{w(GPOL)}$ Pulse duration, GPOx low | 36C - 8 | | ns |

(1) C = 1/SYSCLK1 frequency in ns.

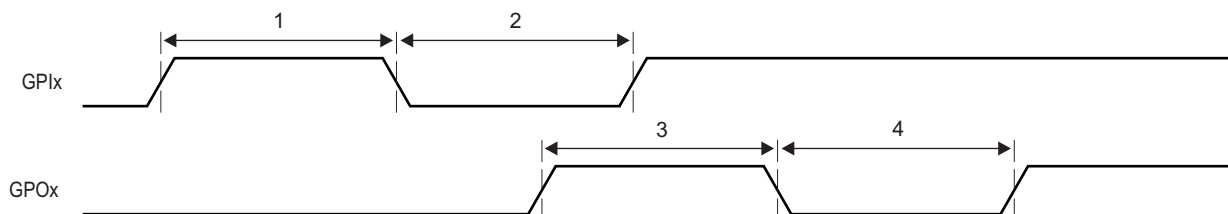


Figure 8-51. GPIO Timing

8.22 Semaphore2

The device contains an enhanced semaphore module for the management of shared resources of the DSP C66x CorePac. The semaphore enforces atomic accesses to shared chip-level resources so that the read-modify-write sequence is not broken. The semaphore module has a unique interrupt to the CorePac to identify when the core has acquired the resource.

Semaphore resources within the module are not tied to specific hardware resources. It is a software requirement to allocate semaphore resources to the hardware resource(s) to be arbitrated.

The semaphore module supports 8(C6655) or 2(C6657) masters and contains 32 semaphores to be used within the system.

The Semaphore module is accessible only by masters with privilege ID (privID) 0(C6655) or 0 to 1(C6657), which means only CorePac 0 (C6655) or 0 to 1(C6657) or the EDMA transactions initiated by CorePac 0(C6655) or 0 to 1(C6657) can access the Semaphore module.

There are two methods of accessing a semaphore resource:

- **Direct Access:** A core directly accesses a semaphore resource. If free, the semaphore will be granted. If not, the semaphore is not granted.
- **Indirect Access:** A core indirectly accesses a semaphore resource by writing it. Once it is free, an interrupt notifies the CPU that it is available.

8.23 Multichannel Buffered Serial Port (McBSP)

The McBSP provides these functions:

- Full-duplex communication
- Double-buffered data registers, which allow a continuous data stream
- Independent framing and clocking for receive and transmit
- Direct interface to industry-standard codecs, analog interface chips (AICs), and other serially connected analog-to-digital (A/D) and digital-to-analog (D/A) devices
- External shift clock or an internal, programmable frequency shift clock for data transfer
- Transmit & receive FIFO buffers allow the McBSP to operate at a higher sample rate by making it more tolerant to DMA latency

If an internal clock source is used, the CLKGDV field of the Sample Rate Generator Register (SRGR) must always be set to a value of 1 or greater.

For more information, see the *Multichannel Buffered Serial Port (McBSP) for KeyStone Devices User's Guide*.

8.23.1 McBSP Peripheral Register

Table 8-83. McBSP/FIFO Registers

| MCBSP0 BYTE ADDRESS | MCBSP1 BYTE ADDRESS | ACRONYM | REGISTER DESCRIPTION |
|--|------------------------|----------|---|
| McBSP Registers | | | |
| 0x021B 4000 | 0x021B 8000 | DRR | McBSP Data Receive Register (read-only) |
| 0x021B 4004 | 0x021B 8004 | DXR | McBSP Data Transmit Register |
| 0x021B 4008 | 0x021B 8008 | SPCR | McBSP Serial Port Control Register |
| 0x021B 400C | 0x021B 800C | RCR | McBSP Receive Control Register |
| 0x021B 4010 | 0x021B 8010 | XCR | McBSP Transmit Control Register |
| 0x021B 4014 | 0x021B 8014 | SRGR | McBSP Sample Rate Generator register |
| 0x021B 4018 | 0x021B 8018 | MCR | McBSP Multichannel Control Register |
| 0x021B 401C | 0x021B 801C | RCERE0 | McBSP Enhanced Receive Channel Enable Register 0 Partition A/B |
| 0x021B 4020 | 0x021B 8020 | XCERE0 | McBSP Enhanced Transmit Channel Enable Register 0 Partition A/B |
| 0x021B 4024 | 0x021B 8024 | PCR | McBSP Pin Control Register |
| 0x021B 4028 | 0x021B 8028 | RCERE1 | McBSP Enhanced Receive Channel Enable Register 1 Partition C/D |
| 0x021B 402C | 0x021B 802C | XCERE1 | McBSP Enhanced Transmit Channel Enable Register 1 Partition C/D |
| 0x021B 4030 | 0x021B 8030 | RCERE2 | McBSP Enhanced Receive Channel Enable Register 2 Partition E/F |
| 0x021B 4034 | 0x021B 8034 | XCERE2 | McBSP Enhanced Transmit Channel Enable Register 2 Partition E/F |
| 0x021B 4038 | 0x021B 8038 | RCERE3 | McBSP Enhanced Receive Channel Enable Register 3 Partition G/H |
| 0x021B 403C | 0x021B 803C | XCERE3 | McBSP Enhanced Transmit Channel Enable Register 3 Partition G/H |
| McBSP FIFO Control and Status Registers | | | |
| 0x021B 6000 | 0x021B A000 | BFIFOREV | BFIFO Revision Identification Register |
| 0x021B 6010 | 0x021B A010 | WFIFOCTL | Write FIFO Control Register |
| 0x021B 6014 | 0x021B A014 | WFIFOSTS | Write FIFO Status Register |
| 0x021B 6018 | 0x021B A018 | RFIFOCTL | Read FIFO Control Register |
| 0x021B 601C | 0x021B A01C | RFIFOSTS | Read FIFO Status Register |
| McBSP FIFO Data Registers | | | |
| 0x2200 0000 | 0x2240 0000 | RBUF | McBSP FIFO Receive Buffer |
| 0x2200 0000 | 0x2240 0000 | XBUF | McBSP FIFO Transmit Buffer |

8.23.2 McBSP Electrical Data/Timing

The following tables assume testing over recommended operating conditions.

8.23.2.1 McBSP Timing

Table 8-84. McBSP Timing Requirements⁽¹⁾

(see [Figure 8-52](#))

| NO. | | | | MIN | MAX | UNIT |
|-----|---------------------------|---|------------|----------------------------|-----|------|
| 2 | $t_{c(\text{CKRX})}$ | Cycle time, CLKR/X | CLKR/X ext | 2P or 20 ⁽²⁾⁽³⁾ | | ns |
| 3 | $t_{w(\text{CKRX})}$ | Pulse duration, CLKR/X high or CLKR/X low | CLKR/X ext | P-1 ⁽⁴⁾ | | ns |
| 5 | $t_{su(\text{FRH-CKRL})}$ | Setup time, external FSR high before CLKR low | CLKR int | 14 | | ns |
| | | | CLKR ext | 4 | | |
| 6 | $t_{h(\text{CKRL-FRH})}$ | Hold time, external FSR high after CLKR low | CLKR int | 6 | | ns |
| | | | CLKR ext | 3 | | |
| 7 | $t_{su(\text{DRV-CKRL})}$ | Setup time, DR valid before CLKR low | CLKR int | 14 | | ns |
| | | | CLKR ext | 4 | | |
| 8 | $t_{h(\text{CKRL-DRV})}$ | Hold time, DR valid after CLKR low | CLKR int | 3 | | ns |
| | | | CLKR ext | 3 | | |
| 10 | $t_{su(\text{FXH-CKXL})}$ | Setup time, external FSX high before CLKX low | CLKR int | 14 | | ns |
| | | | CLKR ext | 4 | | |
| 11 | $t_{h(\text{CKXL-FXH})}$ | Hold time, external FSX high after CLKX low | CLKR int | 6 | | ns |
| | | | CLKR ext | 3 | | |

- (1) CLKRP = CLKXP = FSRP = FSXP = 0. If polarity of any of the signals is inverted, then the timing references of that signal are also inverted.
- (2) P = SYSCLK7 period in ns. For example, when the SYSCLK7 clock domain is running at 166MHz, use 6ns.
- (3) Use whichever value is greater. Minimum CLKR/X cycle times must be met, even when CLKR/X is generated by an internal clock source. The minimum CLKR/X cycle times are based on internal logic speed; the maximum usable speed may be lower due to EDMA limitations and AC timing requirements
- (4) This parameter applies to the maximum McBSP frequency. Operate serial clocks (CLKR/X) in the reasonable range of 40/60 duty cycle.

Table 8-85. McBSP Switching Characteristics⁽¹⁾

(see Figure 8-52)

| NO. | PARAMETER | | MIN | MAX | UNIT | |
|-----|----------------------|---|------------|----------------------------|----------------------------|----|
| 1 | $t_{d(CKSH-CKRXH)}$ | Delay time, CLKS high to CLKR/X high for internal CLKR/X generated from CLKS input. | 1 | 14.5 | ns | |
| 2 | $t_c(CKRX)$ | Cycle time, CLKR/X | CLKR/X int | 2P or 20 ⁽²⁾⁽³⁾ | ns | |
| 3 | $t_w(CKRX)$ | Pulse duration, CLKR/X high or CLKR/X low | CLKR/X int | C - 2 ⁽⁴⁾ | C + 2 ⁽⁴⁾ ns | |
| 4 | $t_{d(CKRH-FRV)}$ | Delay time, CLKR high to internal FSR valid | CLKR int | -4 | 5.5 | ns |
| 4 | | | CLKR int | 1 | 14.5 | ns |
| 9 | $t_{d(CKXH-FXV)}$ | Delay time, CLKX high to internal FSX valid | CLKX int | -4 | 5.5 | ns |
| | | | CLKX ext | 1 | 14.5 | |
| 12 | $t_{dis(CKXH-DXHZ)}$ | Disable time, DX Hi-Z following last data bit from CLKX high | CLKX int | -4 | 7.5 | ns |
| | | | CLKX ext | 1 | 14.5 | |
| 13 | $t_{d(CKXH-DXV)}$ | Delay time, CLKX high to DX valid | CLKX int | -4 + D1 ⁽⁵⁾ | 5.5 + D2 ⁽⁵⁾ | ns |
| | | | CLKX ext | 1 + D1 ⁽⁵⁾ | 14.5 + D2 ⁽⁵⁾ | |
| 14 | $t_{d(FXH-DXV)}$ | Delay time, FSX high to DX valid applies ONLY when in data delay 0 (XDATDLY = 00b) mode | FSX int | -4 + D1 ⁽⁶⁾ | 5 + D2 ⁽⁶⁾ | ns |
| | | | FSX ext | -2 + D1 ⁽⁶⁾ | 14.5 + D2 ⁽⁶⁾ | |

(1) Minimum delay times also represent minimum output hold times.

(2) P = SYSCLK7 period in ns. For example, when the SYSCLK7 clock domain is running at 166 MHz, use 6 ns.

(3) Use whichever value is greater.

(4) C = H or L

S = sample rate generator input clock = P if CLKSM = 1 (P = SYSCLK7 period)

S = sample rate generator input clock = P_clks if CLKSM = 0 (P_clks = CLKS period)

If CLKGDV is even:

(1) H = CLKX high pulse width = (CLKGDV/2 + 1) * S

(2) L = CLKX low pulse width = (CLKGDV/2) * S

If CLKGDV is odd:

(1) H = (CLKGDV + 1)/2 * S

(2) L = (CLKGDV + 1)/2 * S

CLKGDV should be set appropriately to ensure the McBSP bit rate does not exceed the maximum limit.

(5) Extra delay from CLKX high to DX valid applies only to the first data bit of a device, if and only if DXENA = 1 in SPCR.

if DXENA = 0, then D1 = D2 = 0

if DXENA = 1, then D1 = 4P, D2 = 8P

(6) Extra delay from FSX high to DX valid applies only to the first data bit of a device, if and only if DXENA = 1 in SPCR.

if DXENA = 0, then D1 = D2 = 0

if DXENA = 1, then D1 = 4P, D2 = 8P

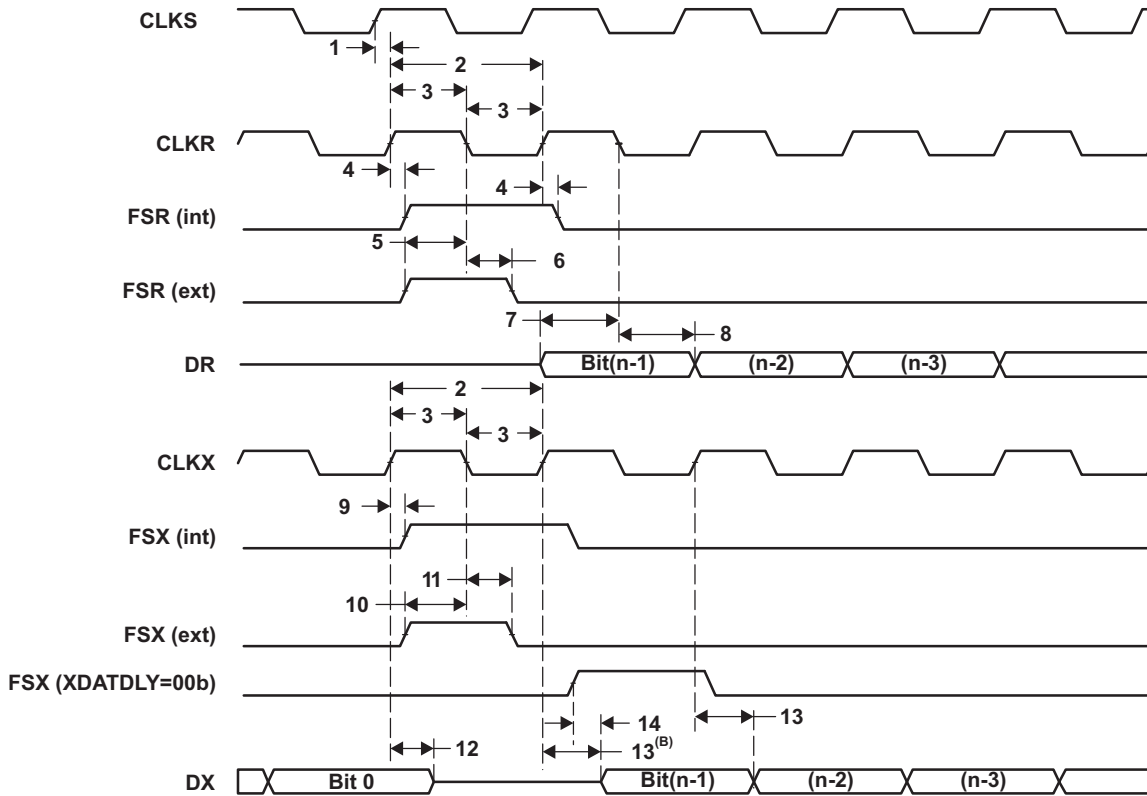


Figure 8-52. McBSP Timing

Table 8-86. McBSP Timing Requirements for FSR When GSYNC = 1

(see Figure 8-53)

| NO. | | MIN | MAX | UNIT |
|-----|--|-----|-----|------|
| 1 | $t_{su}(FRH-CKSH)$ Setup time, FSR high before CLKS high | 4 | | ns |
| 2 | $t_h(CKSH-FRH)$ Hold time, FSR high after CLKS high | 4 | | ns |

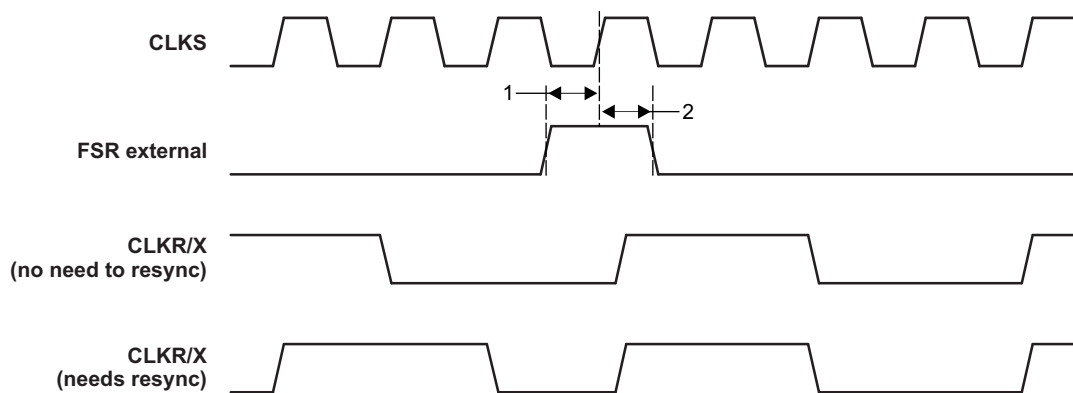


Figure 8-53. FSR Timing When GSYNC = 1

8.24 Universal Parallel Port (uPP)

The universal parallel port (uPP) peripheral is a multichannel, high-speed parallel interface with dedicated data lines and minimal control signals. It is designed to interface cleanly with high-speed analog-to-digital converters (ADCs) or digital-to-analog converters (DACs) with up to 16-bits of data width (per channel). It may also be interconnected with field-programmable gate arrays (FPGAs) or other uPP devices to achieve high-speed digital data transfer. It can operate in receive mode, transmit mode, or duplex mode, in which its individual channels operate in opposite directions.

The uPP peripheral includes an internal DMA controller to maximize throughput and minimize CPU overhead during high-speed data transmission. All uPP transactions use the internal DMA to provide data to or retrieve data from the I/O channels. The DMA controller includes two DMA channels, which typically service separate I/O channels. The uPP peripheral also supports data interleave mode, in which all DMA resources service a single I/O channel. In this mode, only one I/O channel may be used.

The features of the uPP include:

- Programmable data width per channel (from 8 bits to 16 bits inclusive)
- Programmable data justification
 - Right-justify with 0 extend
 - Right-justify with sign extend
 - Left-justify with 0 fill
- Supports multiplexing of interleaved data during SDR transmit
- Optional frame START signal with programmable polarity
- Optional data ENABLE signal with programmable polarity
- Optional synchronization WAIT signal with programmable polarity
- Single Data Rate (SDR) or Double Data Rate (DDR, interleaved) interface
 - Supports multiplexing of interleaved data during SDR transmit
 - Supports demultiplexing and multiplexing of interleaved data during DDR transfers

For more information, see the *Universal Parallel Port (uPP) for KeyStone Devices User's Guide*.

8.24.1 uPP Register Descriptions

Table 8-87. Universal Parallel Port (uPP) Registers

| BYTE ADDRESS | ACRONYM | REGISTER DESCRIPTION |
|--------------|---------|---|
| 0x0258 0000 | UPPID | uPP Peripheral Identification Register |
| 0x0258 0004 | UPPCR | uPP Peripheral Control Register |
| 0x0258 0008 | UPDLB | uPP Digital Loopback Register |
| 0x0258 0010 | UPCTL | uPP Channel Control Register |
| 0x0258 0014 | UPICR | uPP Interface Configuration Register |
| 0x0258 0018 | UPIVR | uPP Interface Idle Value Register |
| 0x0258 001C | UPTCR | uPP Threshold Configuration Register |
| 0x0258 0020 | UPISR | uPP Interrupt Raw Status Register |
| 0x0258 0024 | UPIER | uPP Interrupt Enabled Status Register |
| 0x0258 0028 | UPIES | uPP Interrupt Enable Set Register |
| 0x0258 002C | UPIEC | uPP Interrupt Enable Clear Register |
| 0x0258 0030 | UPEOI | uPP End-of-Interrupt Register |
| 0x0258 0040 | UPID0 | uPP DMA Channel I Descriptor 0 Register |
| 0x0258 0044 | UPID1 | uPP DMA Channel I Descriptor 1 Register |
| 0x0258 0048 | UPID2 | uPP DMA Channel I Descriptor 2 Register |
| 0x0258 0050 | UPIS0 | uPP DMA Channel I Status 0 Register |
| 0x0258 0054 | UPIS1 | uPP DMA Channel I Status 1 Register |
| 0x0258 0058 | UPIS2 | uPP DMA Channel I Status 2 Register |
| 0x0258 0060 | UPQD0 | uPP DMA Channel Q Descriptor 0 Register |
| 0x0258 0064 | UPQD1 | uPP DMA Channel Q Descriptor 1 Register |
| 0x0258 0068 | UPQD2 | uPP DMA Channel Q Descriptor 2 Register |
| 0x0258 0070 | UPQS0 | uPP DMA Channel Q Status 0 Register |
| 0x0258 0074 | UPQS1 | uPP DMA Channel Q Status 1 Register |
| 0x0258 0078 | UPQS2 | uPP DMA Channel Q Status 2 Register |

Table 8-88. uPP Timing Requirements

 (see [Figure 8-54](#), [Figure 8-55](#), [Figure 8-56](#), [Figure 8-57](#))

| NO. | PARAMETER | | MIN | MAX | UNIT |
|-----|-----------------------|--|----------|-------|------|
| 1 | $t_{c(INCLK)}$ | Cycle time, CH _n _CLK | SDR mode | 13.33 | ns |
| | | | DDR mode | 26.66 | |
| 2 | $t_{w(INCLKH)}$ | Pulse width, CH _n _CLK high | SDR mode | 5 | ns |
| | | | DDR mode | 10 | |
| 3 | $t_{w(INCLKL)}$ | Pulse width, CH _n _CLK low | SDR mode | 5 | ns |
| | | | DDR mode | 10 | |
| 4 | $t_{su(STV-INCLKH)}$ | Setup time, CH _n _START valid before CH _n _CLK high | 4 | | ns |
| 5 | $t_h(INCLKH-STV)$ | Hold time, CH _n _START valid after CH _n _CLK high | 0.8 | | ns |
| 6 | $t_{su(ENV-INCLKH)}$ | Setup time, CH _n _ENABLE valid before CH _n _CLK high | 4 | | ns |
| 7 | $t_h(INCLKH-ENV)$ | Hold time, CH _n _ENABLE valid after CH _n _CLK high | 0.8 | | ns |
| 8 | $t_{su(DV-INCLKH)}$ | Setup time, CH _n _DATA/XDATA valid before CH _n _CLK high | 4 | | ns |
| 9 | $t_h(INCLKH-DV)$ | Hold time, CH _n _DATA/XDATA valid after CH _n _CLK high | 0.8 | | ns |
| 10 | $t_{su(DV-INCLKL)}$ | Setup time, CH _n _DATA/XDATA valid before CH _n _CLK low | 4 | | ns |
| 11 | $t_h(INCLKL-DV)$ | Hold time, CH _n _DATA/XDATA valid after CH _n _CLK low | 0.8 | | ns |
| 19 | $t_{su(WTV-OUTCLKL)}$ | Setup time, CH _n _WAIT valid before CH _n _CLK high | 4 | | ns |
| 20 | $t_h(INCLKL-WTV)$ | Hold time, CH _n _WAIT valid after CH _n _CLK high | 0.8 | | ns |
| 21 | $t_{c(2TXCLK)}$ | Cycle time, 2xTXCLK input clock ⁽¹⁾ | 6.66 | | ns |

(1) 2xTXCLK is an alternate transmit clock source that must be at least 2 times the required uPP transmit clock rate (as it is divided down by 2 inside the uPP). 2xTXCLK has no specified skew relationship to the CH_n_CLOCK and therefore is not shown in the timing diagram.

Table 8-89. uPP Switching Characteristics

 (see [Figure 8-56](#), [Figure 8-57](#))

| NO. | PARAMETER | | MIN | MAX | UNIT |
|-----|--------------------|---|----------|-------|------|
| 12 | $t_{c(OUTCLK)}$ | Cycle time, CH _n _CLK | SDR mode | 13.33 | ns |
| | | | DDR mode | 26.66 | |
| 13 | $t_{w(OUTCLKH)}$ | Pulse width, CH _n _CLK high | SDR mode | 5 | ns |
| | | | DDR mode | 10 | |
| 14 | $t_{w(OUTCLKL)}$ | Pulse width, CH _n _CLK low | SDR mode | 5 | ns |
| | | | DDR mode | 10 | |
| 15 | $t_d(OUTCLKH-STV)$ | Delay time, CH _n _START valid after CH _n _CLK high | 1 | 11 | ns |
| 16 | $t_d(OUTCLKH-ENV)$ | Delay time, CH _n _ENABLE valid after CH _n _CLK high | 1 | 11 | ns |
| 17 | $t_d(OUTCLKH-DV)$ | Delay time, CH _n _DATA/XDATA valid after CH _n _CLK high | 1 | 11 | ns |
| 18 | $t_d(OUTCLKL-DV)$ | Delay time, CH _n _DATA/XDATA valid after CH _n _CLK low | 1 | 11 | ns |

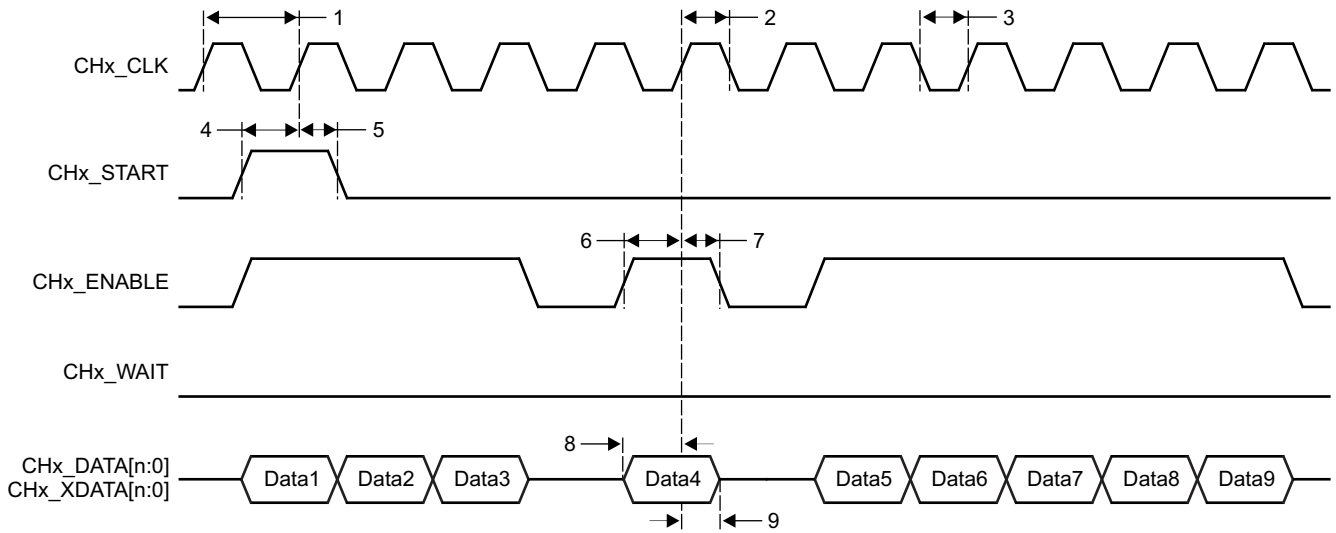


Figure 8-54. uPP Single Data Rate (SDR) Receive Timing

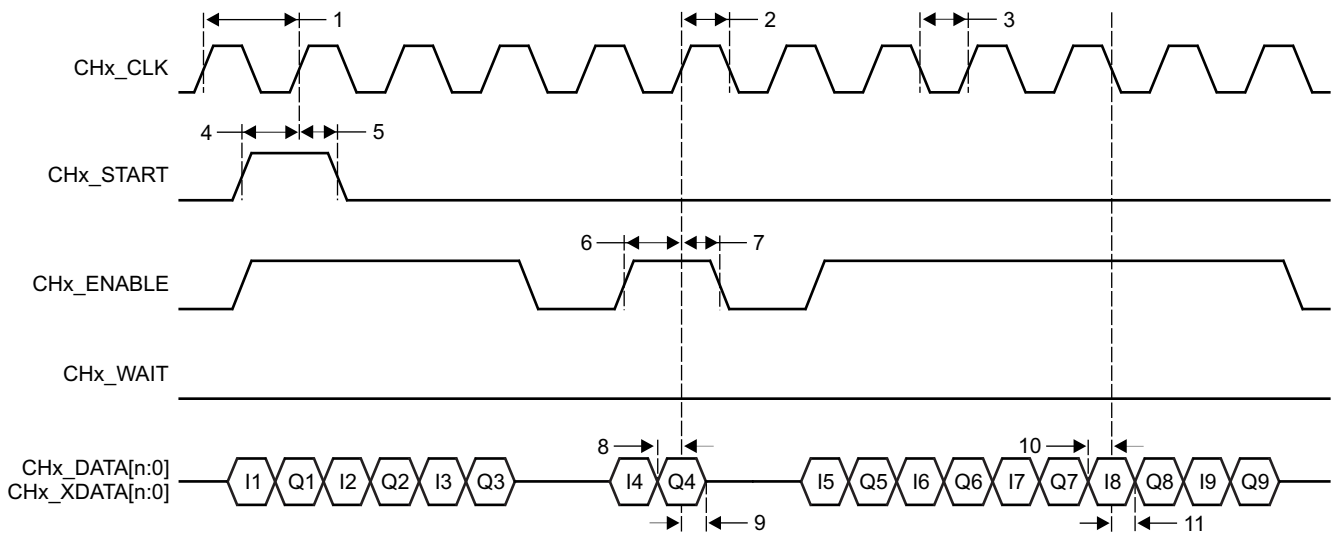


Figure 8-55. uPP Double Data Rate (DDR) Receive Timing

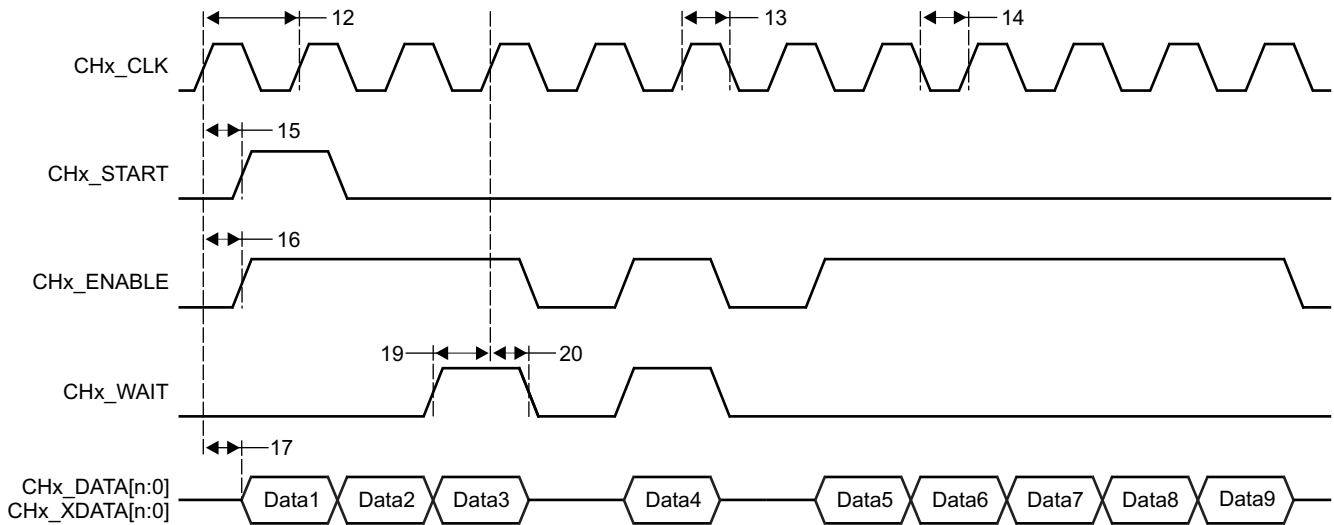


Figure 8-56. uPP Single Data Rate (SDR) Transmit Timing

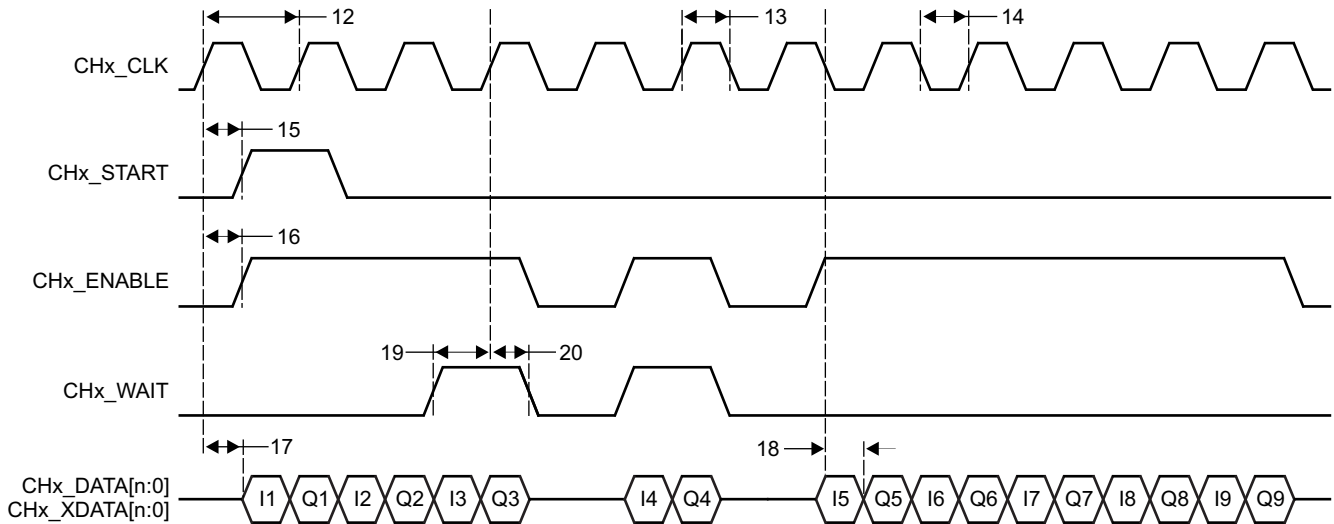


Figure 8-57. uPP Double Data Rate (DDR) Transmit Timing

8.25 Serial RapidIO (SRIO) Port

The SRIO port is a high-performance, low pin-count interconnect aimed for embedded markets. The use of the RapidIO interconnect in a baseband board design can create a homogeneous interconnect environment, providing even more connectivity and control among the components. RapidIO is based on the memory and device addressing concepts of processor buses where the transaction processing is managed completely by hardware. This enables the RapidIO interconnect to lower the system cost by providing lower latency, reduced overhead of packet data processing, and higher system bandwidth, all of which are key for wireless interfaces. For more information, see the *Serial RapidIO (SRIO) for KeyStone Devices User's Guide* in [Section 3.11](#).

8.26 Turbo Decoder Coprocessor (TCP3d)

The C6655 and C6657 have one high-performance embedded Turbo-Decoder Coprocessor (TCP3d) that significantly speeds up channel-decoding operations on-chip for WCDMA, HSPA, HSPA+, TD-SCDMA, LTE, and WiMAX. Operating at CPU clock divided-by-2, the TCP3d is capable of processing data channels at a throughput of >100 Mbps. For more information, see the *Turbo Decoder Coprocessor 3 (TCP3d) for KeyStone Devices User's Guide* in [Section 3.11](#).

8.27 Enhanced Viterbi-Decoder Coprocessor (VCP2)

The devices have two high-performance embedded Viterbi Decoder Coprocessors (VCP2) that significantly speed up channel-decoding operations on-chip. Each VCP2, operating at CPU clock divided-by-3, can decode more than 694 7.95-Kbps adaptive multi-rate (AMR) [K = 9, R = 1/3] voice channels. The VCP2 supports constraint lengths K = 5, 6, 7, 8, and 9, rates R = 3/4, 1/2, 1/3, 1/4, and 1/5, and flexible polynomials, while generating hard decisions or soft decisions. Communications between the VCP2 and the CPU are carried out through the EDMA3 controller.

The VCP2 supports:

- Unlimited frame sizes
- Code rates 3/4, 1/2, 1/3, 1/4, and 1/5
- Constraint lengths 5, 6, 7, 8, and 9
- Programmable encoder polynomials
- Programmable reliability and convergence lengths
- Hard and soft decoded decisions
- Tail and convergent modes
- Yamamoto logic
- Tail biting logic
- Various input and output FIFO lengths

For more information, see the *Viterbi Coprocessor (VCP2) for KeyStone Devices User's Guide* in [Section 3.11](#).

8.28 Emulation Features and Capability

8.28.1 Advanced Event Triggering (AET)

The C665x device supports advanced event triggering (AET). This capability can be used to debug complex problems as well as understand performance characteristics of user applications. AET provides the following capabilities:

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

For more information on AET, see the following documents in [Section 3.11](#):

- *Using Advanced Event Triggering to Find and Fix Intermittent Real-Time Bugs* ([SPRA753](#))
- *Using Advanced Event Triggering to Debug Real-Time Problems in High Speed Embedded Microprocessor Systems* ([SPRA387](#))

8.28.2 Trace

The C665x device supports trace. Trace is a debug technology that provides a detailed, historical account of application code execution, timing, and data accesses. Trace collects, compresses, and exports debug information for analysis. Trace works in real-time and does not impact the execution of the system.

For more information on board design guidelines for trace advanced emulation, see the *60-Pin Emulation Header Technical Reference*.

8.28.2.1 Trace Electrical Data/Timing

Table 8-90. DSP Trace Switching Characteristics⁽¹⁾

(see Figure 8-58)

| NO. | PARAMETER | MIN | MAX | UNIT |
|-----|---|-----|-----|------|
| 1 | $t_w(\text{DPnH})$ Pulse duration, DPn/EMUn high detected at 50% Voh | 2.4 | | ns |
| 1 | $t_w(\text{DPnH})90\%$ Pulse duration, DPn/EMUn high detected at 90% Voh | 1.5 | | ns |
| 2 | $t_w(\text{DPnL})$ Pulse duration, DPn/EMUnlow detected at 50% Voh | 2.4 | | ns |
| 2 | $t_w(\text{DPnL})10\%$ Pulse duration, DPn/EMUnlow detected at 10% Voh | 1.5 | | ns |
| 3 | $t_{sko}(\text{DPn})$ Output skew time, time delay difference between DPn/EMUnpins configured as trace | -1 | 1 | ns |
| | $t_{skp}(\text{DPn})$ Pulse skew, magnitude of difference between high-to-low (tphl) and low-to-high (tplh) propagation delays. | | 600 | ps |
| | $t_{\sigma\lambda\delta\tau_o}(\text{DPn})$ Output slew rate DPn/EMUn | 3.3 | | V/ns |

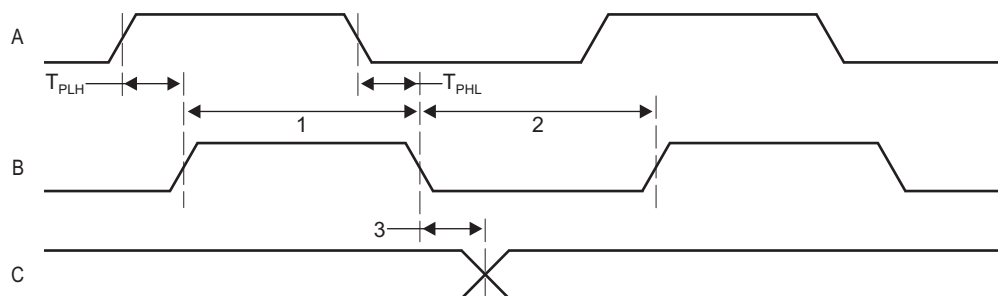
(1) Over recommended operating conditions.

Table 8-91. STM Trace Switching Characteristics⁽¹⁾

(see Figure 8-58)

| NO. | PARAMETER | MIN | MAX | UNIT |
|-----|---|-----|-----|------|
| 1 | $t_w(\text{DPnH})$ Pulse duration, DPn/EMUn high detected at 50% Voh with 60/40 duty cycle | 4 | | ns |
| 1 | $t_w(\text{DPnH})90\%$ Pulse duration, DPn/EMUn high detected at 90% Voh | 3.5 | | ns |
| 2 | $t_w(\text{DPnL})$ Pulse duration, DPn/EMUn low detected at 50% Voh with 60/40 duty cycle | 4 | | ns |
| 2 | $t_w(\text{DPnL})10\%$ Pulse duration, DPn/EMUn low detected at 10% Voh | 3.5 | | ns |
| 3 | $t_{sko}(\text{DPn})$ Output skew time, time delay difference between DPn/EMUn pins configured as trace | -1 | 1 | ns |
| | $t_{skp}(\text{DPn})$ Pulse skew, magnitude of difference between high-to-low (tphl) and low-to-high (tplh) propagation delays. | | 1 | ns |
| | $t_{\sigma\lambda\delta\tau_o}(\text{DPn})$ Output slew rate DPn/EMUn | 3.3 | | V/ns |

(1) Over recommended operating conditions.



A. EMUx represents the EMU output pin configured as the trace clock output. EMUy and EMUz represent all of the trace output data pins.

Figure 8-58. Trace Timing

8.28.3 IEEE 1149.1 JTAG

The JTAG interface is used to support boundary scan and emulation of the device. The boundary scan supported allows for an asynchronous TRST and only the 5 baseline JTAG signals (e.g., no EMU[1:0]) required for boundary scan. Most interfaces on the device follow the Boundary Scan Test Specification (IEEE1149.1), while all of the SerDes (SRIO and SGMII) support the AC-coupled net test defined in *AC-Coupled Net Test Specification* (IEEE1149.6).

It is expected that all compliant devices are connected through the same JTAG interface, in daisy-chain fashion, in accordance with the specification. The JTAG interface uses 1.8-V LVCMOS buffers, compliant with the *Power Supply Voltage and Interface Standard for Nonterminated Digital Integrated Circuit Specification* (EAI/JESD8-5).

8.28.3.1 IEEE 1149.1 JTAG Compatibility Statement

For maximum reliability, the C665x DSP includes an internal pulldown (IPD) on the TRST pin to ensure that TRST will always be asserted upon power up and the DSP's internal emulation logic will always be properly initialized when this pin is not routed out. JTAG controllers from Texas Instruments actively drive TRST high. However, some third-party JTAG controllers may not drive TRST high but expect the use of an external pullup resistor on TRST. When using this type of JTAG controller, assert TRST to initialize the DSP after powerup and externally drive TRST high before attempting any emulation or boundary scan operations.

8.28.3.2 JTAG Electrical Data/Timing

Table 8-92. JTAG Test Port Timing Requirements

(see [Figure 8-59](#))

| NO. | | | MIN | MAX | UNIT |
|-----|----------------|--|------|-----|------|
| 1 | $t_{c(TCK)}$ | Cycle time, TCK | 34 | | ns |
| 1a | $tw(TCKH)$ | Pulse duration, TCK high (40% of t_c) | 13.6 | | ns |
| 1b | $tw(TCKL)$ | Pulse duration, TCK low(40% of t_c) | 13.6 | | ns |
| 3 | $tsu(TDI-TCK)$ | input setup time, TDI valid to TCK high | 3.4 | | ns |
| 3 | $tsu(TMS-TCK)$ | input setup time, TMS valid to TCK high | 3.4 | | ns |
| 4 | $th(TCK-TDI)$ | input hold time, TDI valid from TCK high | 17 | | ns |
| 4 | $th(TCK-TMS)$ | input hold time, TMS valid from TCK high | 17 | | ns |

Table 8-93. JTAG Test Port Switching Characteristics⁽¹⁾

(see [Figure 8-59](#))

| NO. | PARAMETER | MIN | MAX | UNIT |
|-----|--------------------|-----|------|------|
| 2 | $t_{d(TCKL-TDOV)}$ | | 13.6 | ns |

(1) Over recommended operating conditions.

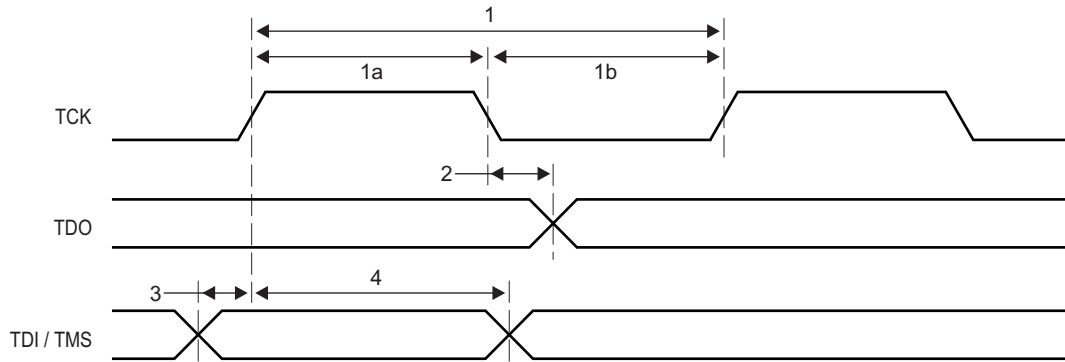


Figure 8-59. JTAG Test-Port Timing

8.29 Related Links

The table below lists quick access links. Categories include technical documents, support and community resources, tools and software, and quick access to sample or buy.

Table 8-94. Related Links

| PARTS | PRODUCT FOLDER | SAMPLE & BUY | TECHNICAL DOCUMENTS | TOOLS & SOFTWARE | SUPPORT & COMMUNITY |
|-------------|----------------------------|----------------------------|----------------------------|----------------------------|----------------------------|
| TMS320C6655 | Click here | Click here | Click here | Click here | Click here |
| TMS320C6657 | Click here | Click here | Click here | Click here | Click here |

9 Mechanical Data

9.1 Thermal Data

[Table 9-1](#) shows the thermal resistance characteristics for the PBGA - CZH/GZH mechanical package.

Table 9-1. Thermal Resistance Characteristics (PBGA Package) [CZH/GZH]

| NO. | | | °C/W |
|-----|----------------|-------------------|-------|
| 1 | $R\theta_{JC}$ | Junction-to-case | 0.284 |
| 2 | $R\theta_{JB}$ | Junction-to-board | 4.200 |

9.2 Packaging Information

The following packaging information reflects the most current released data available for the designated device(s). This data is subject to change without notice and without revision of this document.

PACKAGING INFORMATION

| Orderable Device | Status (1) | Package Type | Package Drawing | Pins | Package Qty | Eco Plan (2) | Lead/Ball Finish (6) | MSL Peak Temp (3) | Op Temp (°C) | Device Marking (4/5) | Samples |
|-------------------|---------------|--------------|--------------------|------|----------------|----------------------------|-------------------------|----------------------|--------------|--|-------------------------|
| TMS320C6655CZH | ACTIVE | FCBGA | CZH | 625 | 60 | Green (RoHS & no Sb/Br) | SNAGCU | Level-3-245C-168 HR | 0 to 85 | TMS320C6655CZH @2012 TI | Samples |
| TMS320C6655CZH25 | ACTIVE | FCBGA | CZH | 625 | 60 | Green (RoHS & no Sb/Br) | SNAGCU | Level-3-245C-168 HR | 0 to 85 | TMS320C6655CZH @2012 TI 1.25GHZ | Samples |
| TMS320C6655CZHA | ACTIVE | FCBGA | CZH | 625 | 60 | Green (RoHS & no Sb/Br) | SNAGCU | Level-3-245C-168 HR | -40 to 100 | TMS320C6655CZH @2012 TI A1GHZ | Samples |
| TMS320C6655CZHA25 | ACTIVE | FCBGA | CZH | 625 | 1 | Green (RoHS & no Sb/Br) | SNAGCU | Level-3-245C-168 HR | -40 to 100 | TMS320C6655CZH @2012 TI A1.25GHZ | Samples |
| TMS320C6655GZHA | ACTIVE | FCBGA | GZH | 625 | 60 | TBD | SNPB | Level-3-220C-168 HR | -40 to 100 | TMS320C6655GZH @2012 TI A1GHZ | Samples |
| TMS320C6655SCZH | ACTIVE | FCBGA | CZH | 625 | | Green (RoHS & no Sb/Br) | SNAGCU | Level-3-245C-168 HR | 0 to 85 | TMS320C6655SCZH @2012 TI | Samples |
| TMS320C6657CZH | ACTIVE | FCBGA | CZH | 625 | 60 | Green (RoHS & no Sb/Br) | SNAGCU | Level-3-245C-168 HR | 0 to 85 | TMS320C6657CZH @2012 TI | Samples |
| TMS320C6657CZH25 | ACTIVE | FCBGA | CZH | 625 | 60 | Green (RoHS & no Sb/Br) | SNAGCU | Level-3-245C-168 HR | 0 to 85 | TMS320C6657CZH @2012 TI 1.25GHZ | Samples |
| TMS320C6657CZH8 | ACTIVE | FCBGA | CZH | 625 | 1 | Green (RoHS & no Sb/Br) | SNAGCU | Level-3-245C-168 HR | 0 to 85 | TMS320C6657CZH @2012 TI 850MHZ | Samples |
| TMS320C6657CZHA | ACTIVE | FCBGA | CZH | 625 | 1 | Green (RoHS & no Sb/Br) | SNAGCU | Level-3-245C-168 HR | -40 to 100 | TMS320C6657CZH @2012 TI A1GHZ | Samples |
| TMS320C6657CZHA25 | ACTIVE | FCBGA | CZH | 625 | 60 | Green (RoHS & no Sb/Br) | SNAGCU | Level-3-245C-168 HR | -40 to 100 | TMS320C6657CZH @2012 TI A1.25GHZ | Samples |
| TMS320C6657GZHA | ACTIVE | FCBGA | GZH | 625 | 60 | TBD | SNPB | Level-3-220C-168 HR | -40 to 100 | TMS320C6657GZH @2012 TI A1GHZ | Samples |
| TMS320C6657SCZH | ACTIVE | FCBGA | CZH | 625 | | Green (RoHS & no Sb/Br) | SNAGCU | Level-3-245C-168 HR | 0 to 85 | TMS320C6657SCZH @2012 TI | Samples |

(1) The marketing status values are defined as follows:

ACTIVE: Product device recommended for new designs.

LIFEBUY: TI has announced that the device will be discontinued, and a lifetime-buy period is in effect.

NRND: Not recommended for new designs. Device is in production to support existing customers, but TI does not recommend using this part in a new design.

PREVIEW: Device has been announced but is not in production. Samples may or may not be available.

OBSOLETE: TI has discontinued the production of the device.

(2) Eco Plan - The planned eco-friendly classification: Pb-Free (RoHS), Pb-Free (RoHS Exempt), or Green (RoHS & no Sb/Br) - please check <http://www.ti.com/productcontent> for the latest availability information and additional product content details.

TBD: The Pb-Free/Green conversion plan has not been defined.

Pb-Free (RoHS): TI's terms "Lead-Free" or "Pb-Free" mean semiconductor products that are compatible with the current RoHS requirements for all 6 substances, including the requirement that lead not exceed 0.1% by weight in homogeneous materials. Where designed to be soldered at high temperatures, TI Pb-Free products are suitable for use in specified lead-free processes.

Pb-Free (RoHS Exempt): This component has a RoHS exemption for either 1) lead-based flip-chip solder bumps used between the die and package, or 2) lead-based die adhesive used between the die and leadframe. The component is otherwise considered Pb-Free (RoHS compatible) as defined above.

Green (RoHS & no Sb/Br): TI defines "Green" to mean Pb-Free (RoHS compatible), and free of Bromine (Br) and Antimony (Sb) based flame retardants (Br or Sb do not exceed 0.1% by weight in homogeneous material)

(3) MSL, Peak Temp. - The Moisture Sensitivity Level rating according to the JEDEC industry standard classifications, and peak solder temperature.

(4) There may be additional marking, which relates to the logo, the lot trace code information, or the environmental category on the device.

(5) Multiple Device Markings will be inside parentheses. Only one Device Marking contained in parentheses and separated by a "~" will appear on a device. If a line is indented then it is a continuation of the previous line and the two combined represent the entire Device Marking for that device.

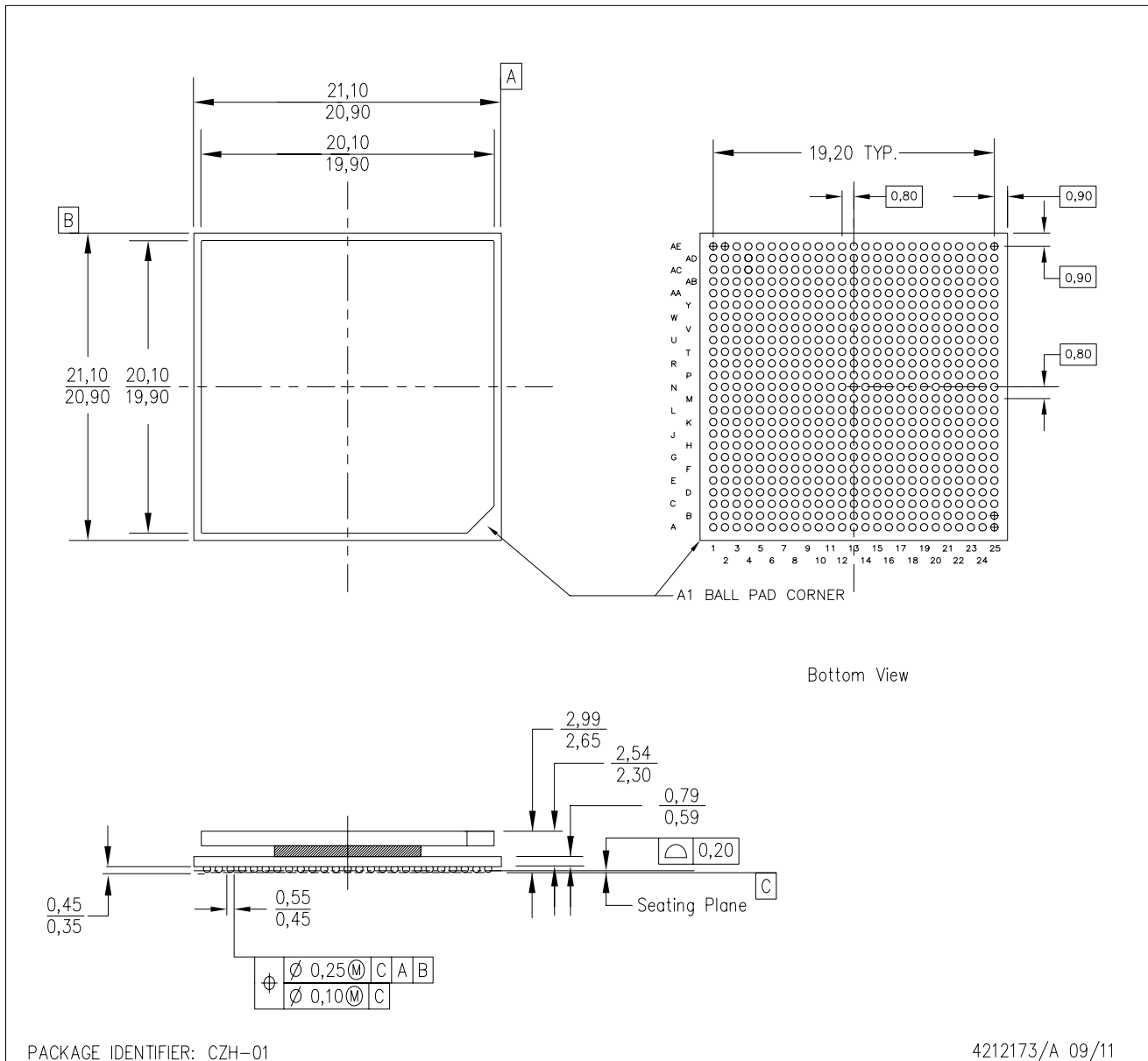
(6) Lead/Ball Finish - Orderable Devices may have multiple material finish options. Finish options are separated by a vertical ruled line. Lead/Ball Finish values may wrap to two lines if the finish value exceeds the maximum column width.

Important Information and Disclaimer: The information provided on this page represents TI's knowledge and belief as of the date that it is provided. TI bases its knowledge and belief on information provided by third parties, and makes no representation or warranty as to the accuracy of such information. Efforts are underway to better integrate information from third parties. TI has taken and continues to take reasonable steps to provide representative and accurate information but may not have conducted destructive testing or chemical analysis on incoming materials and chemicals. TI and TI suppliers consider certain information to be proprietary, and thus CAS numbers and other limited information may not be available for release.

In no event shall TI's liability arising out of such information exceed the total purchase price of the TI part(s) at issue in this document sold by TI to Customer on an annual basis.

CZH (S-PBGA-N625)

PLASTIC BALL GRID ARRAY



- NOTES:
- A. All linear dimensions are in millimeters. Dimensioning and tolerancing per ASME Y14.5M-1994.
 - B. This drawing is subject to change without notice.
 - C. Flip chip application only.
 - D. Thermally enhanced plastic package with heat slug (HSL).
 - E. Pb-free die bump and solder ball.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as "components") are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI's terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers' products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers' products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI's goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or "enhanced plastic" are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have **not** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products

| | |
|------------------------------|--|
| Audio | www.ti.com/audio |
| Amplifiers | amplifier.ti.com |
| Data Converters | dataconverter.ti.com |
| DLP® Products | www.dlp.com |
| DSP | dsp.ti.com |
| Clocks and Timers | www.ti.com/clocks |
| Interface | interface.ti.com |
| Logic | logic.ti.com |
| Power Mgmt | power.ti.com |
| Microcontrollers | microcontroller.ti.com |
| RFID | www.ti-rfid.com |
| OMAP Applications Processors | www.ti.com/omap |
| Wireless Connectivity | www.ti.com/wirelessconnectivity |

Applications

| | |
|-------------------------------|--|
| Automotive and Transportation | www.ti.com/automotive |
| Communications and Telecom | www.ti.com/communications |
| Computers and Peripherals | www.ti.com/computers |
| Consumer Electronics | www.ti.com/consumer-apps |
| Energy and Lighting | www.ti.com/energy |
| Industrial | www.ti.com/industrial |
| Medical | www.ti.com/medical |
| Security | www.ti.com/security |
| Space, Avionics and Defense | www.ti.com/space-avionics-defense |
| Video and Imaging | www.ti.com/video |

TI E2E Community

e2e.ti.com