OMAP5912

Applications Processor

Silicon Errata

SPRZ209K
December 2003 − Revised December 2007

Copyright © 2007, Texas Instruments Incorporated

Contents

1 Introduction . . . 7

1.1 Device and Development-Support Tool Nomenclature . . . 7

2 Important Notices and Information About OMAP5912 . . . 10

2.1 Useful Information Regarding TMS320C55x Assembler Diagnostic Messages . . . 10

2.1.1 ERROR Diagnostics . . . 10

2.1.2 WARNING Diagnostics . . . 10

2.1.3 REMARK Diagnostics . . . 10

3 Usage Notes . . . 12

DMA xxxx_IT_COND Bits Clear Procedure is not Correctly Documented in OMAP Reference Guide . . . 12

DSP MMU Maximum Speed on OMAP3.2 . . . 12

UART IrDA Recommendation for Clock Gating clk16x . . . 12

Activating DSP Isolation Will Increase Power Consumption on CV_{DD} Power . . . 12

ETM FIFO Overflow . . . 12

MICROWIRE Auto-Transmit Mode Definition . . . 12

Set Address Request: Minor Violation of USB 2.0 Specification . . . 13

OMAP5912 Global Clock Rules . . . 13

Dynamic Switch Locked if DSP Reset Occurs During a DSP Access . . . 13

Camera Interface Moved Between Production and Engineering OMAP5912 . . . 13

4 DSP Subsystem Advisories . . . 14

4.1 DSP ICACHE Advisories . . . 14

DSP_ICACHE_1 DSP ICACHE is Servicing a Missed Program Bus Request During a RAMSET Preload . . . 14

4.2 DSP Emulation Advisories . . . 16

DSP_EMU_1 DSP IDLE Interrupt Not Serviced When Emulator is Connected . . . 16

DSP_EMU_2 Bad RTCK Clock Signal May Cause JTAG Communication Failure between the OMAP and an Emulator . . . 17

4.3 DSP System Advisories . . . 18

DSP_SYS_1 Use Caution When Reading Following a Configuration Change on the DSP . . . 18

DSP_SYS_2 Users of (JTAG) RTDX are Advised Not to Put the DSP Domain Into Idle . . . 18

DSP_SYS_3 DSP IDLE Wakeup With Program Code in External Memory and EMIF Domain in IDLE Hangs . . . 19

DSP_SYS_4 DM Timer Incorrect Value Read During DSP Access and Value Change . . . 19

DSP_SYS_5 DSP STIO Interrupt Generation is Not Possible . . . 20

4.4 DSP Direct Memory Access (DMA) Advisories . . . 21

DSP_DMA_1 DSP DMA IDLE Prevents Transfer Completion . . . 21

DSP_DMA_2 DSP EMIF/DMA Port Hangs During EMIF Bus Error . . . 21

5 MPU Subsystem Advisories 22

5.1 System Direct Memory Access (DMA) Advisories 22

SYS_DMA_1 System DMA CSAC and CDAC Not Showing Correct Value Consistently 22
SYS_DMA_2 Cannot Get SysDMA to Complete Data Transfer Out of UART 22
SYS_DMA_3 DMA Synchronized Transfers Can Be Stalled 23

5.2 MPU Emulation Advisories 24

MPU_EMU_1 Emulation is Intrusive to ARM Idle Mode: Cannot Stop the CPU Clock With Code Composer Studio Connected 24

5.3 MPU System Advisories 25

MPU_SYS_1 STANDBY Mode Not Entered After Execution of WFI Instruction 25
MPU_SYS_2 Inefficient Fill of Instruction Prefetch Buffer 25
MPU_SYS_3 External Device Power Control is Broken 25
MPU_SYS_4 Emulation Causes OMAP Not to Enter Deep Sleep 26
MPU_SYS_5 Write Followed by Immediate Read not Supported on the Rhea Switch Peripheral . . 27
MPU_SYS_6 When Absent, the OMAP Peripheral Clock Prevents Deep Sleep Entry 27
MPU_SYS_7 MPU IRQ122−127 Can Cause Spurious Interrupts 28
MPU_SYS_8 If BVLZ is Not Configured, the Corresponding Interrupt Must be Masked 28
MPU_SYS_9 Access to the OMAP3.2 ARM_CLKM Registers 29
MPU_SYS_10 WFI Instruction Loses 2−4 Instruction Fetches 29

5.4 MPU/Watchdog Timer Advisories 31

MPU_WATCHDOG_1 MPU Watchdog Does Not Reset DPLL 31
MPU_WATCHDOG_2 The Secure Watchdog Register Reads Can Return Wrong Value Under Some Clock Configurations 32

6 Traffic Controller Subsystem Advisories 33

6.1 EMIF Slow (EMIFS) Advisories 33

EMIFS_1 Unexpected FCLK on EMIFS When Dynamic Wait State is Selected 33
EMIFS_2 Program Fetch From External Memory at Address 0x3C18 Results in Processor Reset 33

6.2 EMIF Fast (EMIFF) Advisories 34

EMIFF_1 MRS/EMRS Command Issued Twice SDRAM Interface 34
EMIFF_2 Limitation of the EMIFF Not Mentioned in Reference Guide 34
EMIFF_3 Using EMIFF POM0 Mode May Result in Data Read Corruption 35
EMIFF_4 Only 725 DLL is Supported on the EMIFF 35
EMIFF_5 The EMIFF Autoclock Feature Affects the DLL Behavior 36

6.3 Traffic Controller Advisories 37

TC_1 OMAP5912 Global Software Reset Affects Traffic Controller Frequency 37

7 OMAP5912 Peripheral Advisories . . . 39

7.1 Multimedia Card/Secure Digital (MMC/SD) Advisories . . . 39

MMC_SD_1 CMD12 Implementation in TI MMC/SDIO . . . 39
MMC_SD_2 SPI Mode Not Supported on the MMC/SD Peripheral . . . 39
MMC_SD_3 MMC1 Internal Pullups are Not Software-Controllable . . . 40
MMC_SD_4 CMD1 Implementation in TI MMC/SDIO IP May Cause Missing EOC Interrupt . . . 40

7.2 MICROWIRE Advisories . . . 41

UWIRE_1 MICROWIRE Interface RX Data Failures Possible . . . 41
UWIRE_2 Bit 14 of CSR Not Reset . . . 43

7.3 Inter-Integrated Circuit (I^2C) Advisories . . . 44

I2C_1 I^2C Random Duplication of Byte When Sending Long Messages . . . 44

7.4 Universal Serial Bus (USB) Host Advisories . . . 49

USB_HOST_1 USB Host Fails to Receive the First Packet Sent After Remote Wakeup . . . 49
USB_HOST_2 USB Device Controller Remote Wake−Up Feature May Fail . . . 49

7.5 Universal Serial Bus (USB) On-the-Go (OTG) Controller Advisories . . . 50

USB_OTG_1 Specification for TA_WAIT_BCON . . . 50
USB_OTG_2 USB OTG TA_WAIT_VRISE is Set at 200 ms . . . 50
USB_OTG_3 USB OTG Timings Do Not Comply With USB OTG Specifications Rev. 1.0a . . . 51
USB_OTG_4 USB OTG DMA Requests Control Bits Cannot Be Read Back . . . 51
USB_OTG_5 USB OTG A-Device State Machine Does Not Comply With OTG Specification . . . 52
USB_OTG_6 USB OTG Registers are Set to 0x0 After UHOST_EN Bit is Set to 0 . . . 52

7.6 TIMER32K Advisories . . . 53

TIMER32K_1 Timer32K Reload TRB Bit Does Not Work Properly . . . 53

7.7 Microprocessor Unit I/O (MPUIO) Advisories . . . 54

MPUIO_1 MPUIO Input Latch Register is Disabled During TIPB Read Access to it . . . 54
MPUIO_2 MPUIO GPIO_INT is no Longer Generated . . . 55

7.8 Pulse-Width Tone (PWT) Advisories . . . 56

PWT_1 Write Followed by Immediate Read Not Supported on the PWT Peripheral . . . 56

7.9 Camera Interface Advisories . . . 57

CAM_1 Interrupt Observability on Camera Port . . . 57
CAM_2 Peculiar Sequence Needed to Reset Camera FIFO Using the RAZ_FIFO Bit . . . 57

7.10 Universal Asynchronous Receiver/Transmitter (UART) Advisories . . . 58

UART_1 NIRQ Can Stay Low for a Few OCP Clock Cycles After the IRQ is Serviced . . . 58
UART_2 UART IrDA (MIR Mode) Sends an Additional Bit . . . 58
UART_3 When in UART Sleep, Writes to the THR Wakes up the UART but No Data is Transmitted . . . 59

UART_4 Rx Overrun and TX_FIFO_FULL Bit Not Operational When FIFO is Disabled 59
UART_5 Receive Overrun Not Reset on an Early Read of LSR 60
UART_6 Unexpected Pulse on UART.TX During Switch to SIR Mode 60
UART_7 UART3.TX3 Stuck in IrDA Mode 60
UART_8 UART Auto Baud Status Register May Contain Invalid Speeds 61
UART_9 UART Fails to Transmit XON Character in Certain Conditions 61
UART_10 UART FIFO Pointers May be Shifted Upon FIFO Enable 62

7.11 Emulation Advisories 63

EMU_1 Default TDO State During Reset is Unpredictable 63
EMU_2 Emulation Breakpoint Between Sequential 16-Bit Accesses to 32-Bit Registers 63

7.12 Specially Optimized Screen Interface (SoSSI) Advisories 64

SOSSI_1 SoSSI CPriority 64

7.13 Oscillator Advisories 65

OSC_1 32-kHz Oscillator Does Not Power Down in Deep-Sleep Mode 65

7.14 Serial Peripheral Interface (SPI) Advisories 66

SPI_1 Emulation Reads of the SPI Peripheral are Intrusive 66
SPI_2 SPI Not Able to Receive the First Data While Waking Up From Sleep Mode 66
SPI_3 SPI Peripheral Does Not Support 8-Bit Accesses 66
SPI_4 Problem With SPI Functionality After Wakeup From Deep Sleep 67
SPI_5 SPI CS Chosen Polarity Only Applies When the First SPI Transfer Begins 67

7.15 Ultra Low-Power Device (ULPD) Advisories 68

ULPD_1 Emulation Access to the ULPD IT_STATUS_REG are Instrusive 68
ULPD_2 ULPD Analog Cell Configuration are Reset After an MPU Peripheral Reset 68
IULPD_3 DLECT1.WKUP_MODE Bit Must Be 0 for Proper Operation 69
ULPD_4 System May Hang if Idle/Awake Sequence Period is Too Short 70

7.16 Multichannel Serial Interface (MCSI) Advisories 71

MCSI_1 MCSI2 DMA Requests are Not Mapped to the ARM 71
MCSI_2 Late Generation of TX DMA Request in the MCSI 71
MCSI_3 MCSI Does Not Support Multi-Master Mode 72

7.17 Multichannel Buffered Serial Port (McBSP) Advisories 73

MCBSP_1 McBSP in ABIS Mode Does Not Receive Data Properly 73
MCBSP_2 Improper Behavior in McBSP ABIS Mode Transmit 73
MCBSP_3 McBSP 3-Pin Operation Versus 4-Pin Operation Configuration 74

7.18 Dual-Mode Timers Advisories 75

DM_TIMER_1 Consecutive Captures in Dual-Mode Timer 75
DM_TIMER_2 After Wakeup, the First Access to Any Dual-Mode Timer May Fail 75
DM_TIMER_3 Dual-Mode Timer Peripheral May Not Detect Incoming Events 76
DM_TIMER_4 Specific Sequence Needed to Restart the Dual-Mode Timer When One-Shot Mode is Used and Overflow Occurs 77
DM_TIMER_5 DM Timer Continuous Accesses in Non-Posted Mode Can Be Ignored 77
DM_TIMER_6 DM Timer Events May Get Ignored 78

7.19 Liquid Crystal Display (LCD) Advisories 79

LCD_1 LCD AC Bias is not Aligned With Pixel Clock 79
LCD_2 Pullups and Pulldowns Not Supported on LCD Pins P[0:15] 80
LCD_3 The Dynamic Power-Saving Feature Changes the HSYNC and VSYNC Timings 80

7.20 HDQ/1-Wire Advisories 81

HDQ_1WIRE_1 HDQ/1-Wire Peripheral Does Not Meet HDQ Standard Requirements 81
HDQ_1WIRE_2 HDQ/1-Wire Engine Does Not Meet 1-Wire Module Standard Requirements 83

8 OMAP5912 Device/System-Level Advisories 84

8.1 System Advisories 84

SYS_1 External Pulldown is Needed on Ball Y18 of ZZG Package 84
SYS_2 UART1 and UART3 Must Be Configured to Force Idle Mode to Allow OMAP5912 to Go Into Deep Sleep 84
SYS_3 OMAP5912 Will Not Enter Deep Sleep Without OCPI_CK 85
SYS_4 OMAP5912 Gated1/Gated2 Functionality Does Not Control I/O Direction 86
SYS_5 Dynamic Voltage Scaling (DVS) Implementation Requires Special Software Sequence to Properly Enter/Exit DVS 86
SYS_6 Emulation is Broken When the DSP is in Isolation Mode 87
SYS_7 Reset Mode Limitation 87
SYS_8 Booting Mode Limitation 87
SYS_9 High Security and Emulation Mode Limitation 88
SYS_10 OMAP5912 USB Power Consumption 88
SYS_11 Peripherals That Are Not Supported on OMAP5912 88

9 Documentation Support 89

Revision History 90

1 Introduction

This document describes the silicon updates to the functional specifications for the OMAP5912, OMAP revision. Issues related to CPU operation are documented in the *TMS320C55x DSP CPU Programmer's Reference Supplement* (literature number SPRU652).

1.1 Device and Development-Support Tool Nomenclature

To designate the stages in the product development cycle, TI assigns prefixes to the part numbers of all OMAP™ devices and support tools. Each OMAP™ commercial family member has one of three prefixes: X, P, or Null (e.g., **X**OMAP-DM270MGVL-B). 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 (X/TMDX) through fully qualified production devices/tools (Null/TMDS).

Device development evolutionary flow:

X Experimental device that is not necessarily representative of the final device's electrical specifications

P Final silicon die that conforms to the device's electrical specifications but has not completed quality and reliability verification

Null 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

X and P devices and TMDX development-support tools are shipped against the following disclaimer:

"Developmental product is intended for internal evaluation purposes."

Null 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 (X or P) 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.

All trademarks are the property of their respective owners.

The device revision can be determined by the symbols marked on the top of each package. Example markings of the ZDY, GDY, and ZZG packages are shown in Figure 1 through Figure 4.

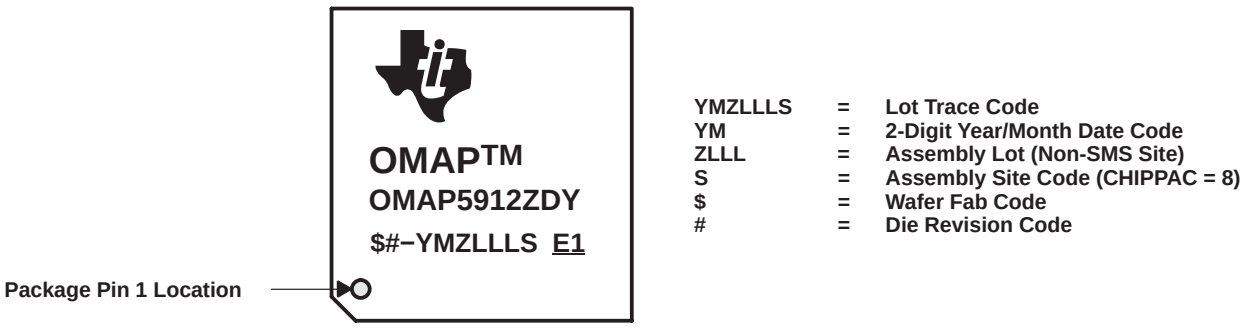

Figure 1. Example Markings for Catalog OMAP5912 ZDY Package

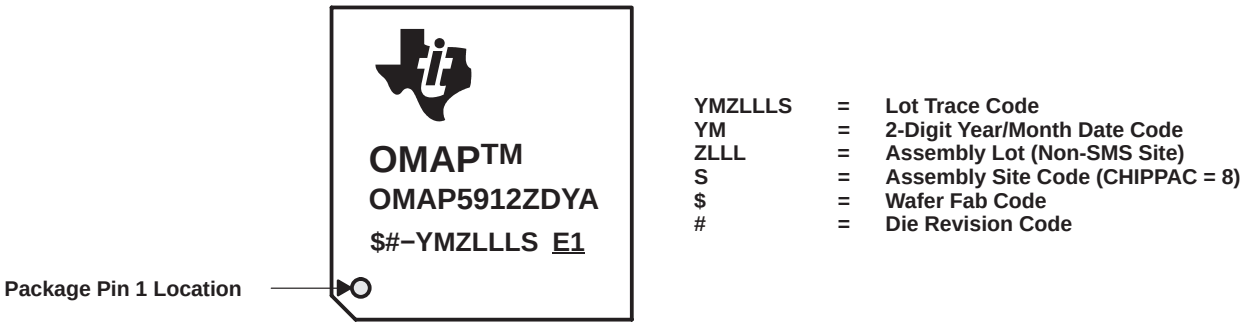

Figure 2. Example Markings for Extended Temperature OMAP5912 ZDY Package

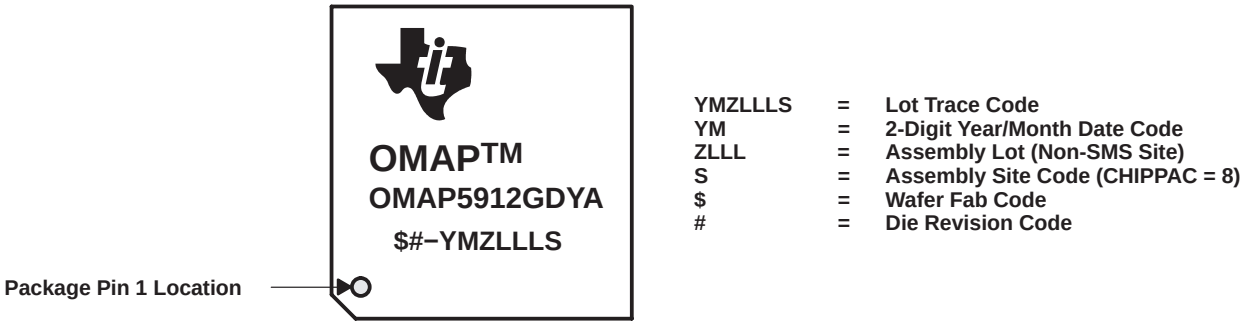

Figure 3. Example Markings for Extended Temperature OMAP5912 GDY Package

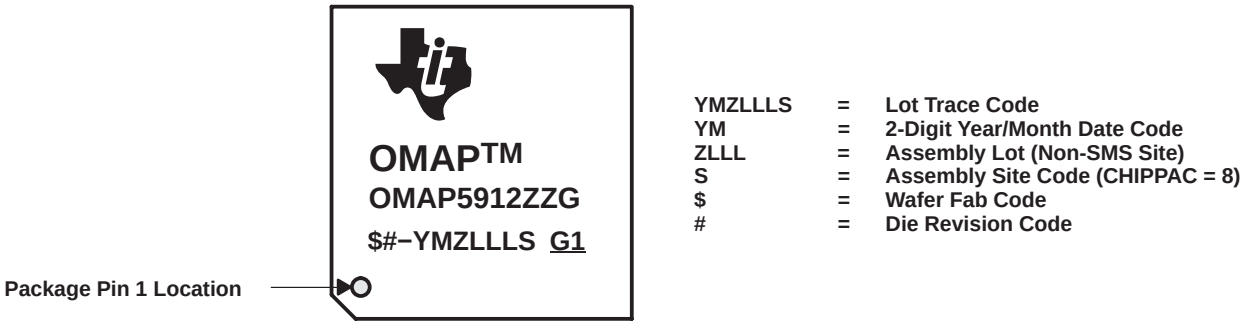

Figure 4. Example Markings for OMAP5912 ZZG Package

2 Important Notices and Information About OMAP5912

2.1 Useful Information Regarding TMS320C55x™ Assembler Diagnostic Messages

The TMS320C55x™ (C55x™) DSP assembler generates three types of diagnostic messages when it detects a potential or probable Silicon Exception.

2.1.1 ERROR Diagnostics

The assembler generates ERROR diagnostics in cases where it can fully determine that the code will cause a silicon exception to occur on hardware.

2.1.2 WARNING Diagnostics

The assembler generates WARNING diagnostics in cases where it can fully determine that the code will cause a silicon exception to occur on hardware, but which, under certain circumstances, may not be an issue for the user.

2.1.3 REMARK Diagnostics

The assembler generates REMARK diagnostics in conditions where it can fully determine that the code may cause a silicon exception to occur on hardware, but the exception itself also depends on non-visible trigger conditions that the assembler has no knowledge of, such as whether interrupts are enabled.

Since the assembler cannot determine the state of these trigger conditions, it cannot know that the exception will affect this code. Therefore, it generates a REMARK to instruct the user to examine the code and evaluate whether this is a potential silicon exception situation. (Please see the following sections for how to suppress remarks in situations where you have determined that the other trigger conditions do not exist.)

Intended Treatment of REMARK Diagnostics

The intent of generating REMARK diagnostics is to inform the user that the code could potentially cause a silicon exception and that it should be reviewed by the user side by side with the trigger conditions and a determination be made whether the code is a potential silicon exception situation.

If the code is determined to be a potential silicon exception situation, users should modify their code to prevent that exception from occurring.

If users determine that their code will not cause a silicon exception based on the trigger conditions, then the REMARK that the assembler generates can be suppressed. There are two methods of doing so; please see the “Suppressing REMARK Diagnostics” section.

Suppressing REMARK Diagnostics

Once the user determines that a silicon exception REMARK diagnostic is not appropriate for the code as written, the REMARK diagnostic can be suppressed in one of the following ways.

- REMARK directives
- REMARK command-line options

TMS320C55x and C55x are trademarks of Texas Instruments.

REMARK Directives:

The .noremark.remark directives can be used to suppress the generation of a REMARK diagnostic for particular regions of code. The .noremark directive turns off the generation of a particular REMARK diagnostic. The .remark directive re-enables the generation of a particular REMARK diagnostic.

A '.noremark ##' (where ## is the remark id) directive is placed at the beginning of the region, and a '.remark ##' directive is placed at the end of the region.

> **NOTE:** The .noremark.remark directive combination should always be placed around the entire region of code that participates in the potential silicon exception. Otherwise, spurious diagnostics may still be generated.

Additionally, the user has the option of disabling a silicon exception diagnostic for the entire file by placing just the .noremark directive at the top of the assembly file. However, this may be dangerous if, during inevitable code maintenance, the code is modified by someone not familiar with all the exception conditions. Please take great care when using the directives in this manner.

REMARK Command-Line Options:

The compiler shell (cl55) supports a command line option to suppress a particular REMARK diagnostic. The shell option −ar# (where # is the assembler's silicon exception id as described above) suppresses the named REMARK for the entire scope of all assembly files compiled with that command. Using the option −ar without a number suppresses all REMARK diagnostics.

Again, this may be dangerous if, during inevitable code maintenance, the code is modified by someone not familiar with all the silicon exception conditions. Please take great care when using the command-line REMARK options. Using the .noremark/.remark directives covering the shortest possible range of source lines is much safer.

3 Usage Notes

Usage Notes highlight and describe particular situations where the device's behavior may not match the presumed or documented behavior. This may include behaviors that affect device performance or functional correctness. These notes will be incorporated into future documentation updates for the device (such as the device-specific data sheet), and the behaviors they describe will not be altered in future silicon revisions.

DMA xxxx_IT_COND Bits Clear Procedure is not Correctly Documented in OMAP Reference Guide

The *OMAP5912 Multimedia Processor Direct Memory Access (DMA) Support Reference Guide* (literature number SPRU755) states that user must write to the DMA_LCD_CTRL register to clear the interrupt status bits. This is not true. When receiving an interrupt, software must read the DMA_LCD_CTRL and then any write on the MPU Private TIPB will clear the register.

The recommended procedure to clear the DMA xxxx_IT_COND bits in DMA_LCD_CTRL register is to:

- Read DMA_LCD_CTRL_REGISTER
- Write back to it as quickly as possible, clearing the required bits. The bits will only be cleared if a read to this register occurred previously; this decreases the interrupt loss risk.

DSP MMU Maximum Speed on OMAP3.2

The DSP MMU clock must have the same maximum speed as the Traffic Controller clock. When using synchronous scalable mode, software must program the ARM_CKCTL .DSPMMUDIV field accordingly.

UART IrDA Recommendation for Clock Gating clk16x

In the UART IrDA, clk16x is operational before the autobaud has acquired any lock status. This is power-inefficient; especially if a lock is not achieved quickly. It is recommended that the UART IrDA clk16x output be gated off until the autobaud functionality has accurately characterized the data settings, hence improving the power consumption of the peripheral.

Activating DSP Isolation Will Increase Power Consumption on CV_{DD} Power

As soon as DSP isolation is activated, INITZ of the internal memory in OMAP MPU power domain will toggle low. This will cause additional power consumption. To save power, It is recommended that the user shut down CV_{DD3} power as soon as possible after activating DSP isolation. Isolation is controlled by ULPD POWER_CTRL_REG.ISOLATION_CONTROL bit.

ETM FIFO Overflow

Frequent ETM FIFO overflows may prevent efficient tracing. For example, when running 32-bit code which contains lots of LDM/STM commands, overflows occur.

MICROWIRE® Auto-Transmit Mode Definition

In autotransmit mode, the CS_CMD and START bits of the control and status register (CSR) are not used. An internal hardware state machine detects a TXD write and automatically sets the programmed CS to its active value, and then starts the transmission. This behavior is only valid when both AUTO_TX_EN and CS_TOGGLE_TX_EN bits are 1. If AUTO_TX_EN is 1 and CS_TOGGLE_TX_EN is 0, then, the CS will not toggle automatically upon a TXD write. In this mode, CS_CMD must be set for transmission to begin when writing a TXD.

MICROWIRE is a registered trademark of National Semiconductor Corporation.

Set Address Request: Minor Violation of USB 2.0 Specification

If the host does not send an ACK in response to the device's empty DATA packet, the device will observe the proper 18-bit timeout. However, the device will be addressed at this point, and not respond to the host's retry SETUP token (using the old address).

This behavior occurs during FS20 testing and is isolated to the Status stage of a SetAddress request and will not be detected during compliance testing. Affects USB client (USB_W2FC).

In general, most OS's will retry enumeration, but they are not required to do so by specification. As a result, the host will have to reset the device.

OMAP5912 Global Clock Rules

The following table lists the maximum allowable clock speeds for the various OMAP5912 clocks.

MPU max clock	192 MHz
DSP max clock	192 MHz
TC max clock	96 MHz
DPLL1 max clock	192 MHz
Flash max clock	48 MHz
DSP MMU max clock	96 MHz
MPUPER max clock	96 MHz
DSPPER max clock	96 MHz
MPU TIPB (Public and Private) max strobe frequency	48 MHz
DSP TIPB (Public and Private) max strobe frequency	32 MHz
MPU interface max strobe frequency	32 MHz
LCD controller max clock	48 MHz

Dynamic Switch Locked if DSP Reset Occurs During a DSP Access

In OMAP5912, four GPIO modules and the 32K synchronous counter are shared between DSP and ARM. If a DSP reset occurs while the DSP is performing an access to a dynamically shared peripheral, the dynamic switch will stay in a locked state and no access to the peripheral is possible anymore. The only way to recover from this locked state is to perform an ARM peripheral reset (set ARM_RSTCT2.PER_EN bit to 0).

Camera Interface Moved Between Production and Engineering OMAP5912

On OMAP5912, the camera interface is connected to the OCP – T1/T2 bus.

On XOMAP5912 or POMAP5912, the camera interface is connected to the TIPB bus.

4 DSP Subsystem Advisories

4.1 DSP ICACHE Advisories

Advisory DSP_ICACHE_1	*DSP ICACHE is Servicing a Missed Program Bus Request During a RAMSET Preload*

Revision(s) Affected: All revisions

Details:

The two ramset banks inside the DSP ICACHE are preloaded when the corresponding ramset tag registers' content are updated through I/O access. The ramset preload uses the same line-fill mechanism as the cache miss line-fill; therefore, there is a conflict of resource when the ramset refill happens while the I-cache miss line-fill is on-going. Although there is logic to arbitrate this conflict, a functional bug can corrupt the content of Tag ram under the following corner condition:

- I-cache is turned on with ramset(s) enabled.
- The instruction(s) for updating the ramset tag register(s) are embedded with other program code in external memory.
- The external memory space where the instruction(s) for updating the ramset tag(s) are not present in the ICACHE, e.g., will generate a cache miss.

Example:

The following example shows assembly code that is likely to encounter this corner case:

```
Address       Instruction
Ext Mem       DSP code
Ext Mem       DSP code
.......
.......
Ext Mem       DSP code
Ext Mem       DSP code
Ext Mem       *port(#0x1400) = #0xc 2f   ; Configure icache
Ext Mem       bit(ST3,#14) = #1          ; Turn on icache
Ext Mem       *port(#0x1406) = #0x0800   ; Update ramset tag for bank1
Ext Mem       *port(#0x1408) = #0x0801   ; Update ramset tag for bank2
Ext Mem       DSP code
Ext Mem       DSP coI......
......
Ext Mem       DSP code
Ext Mem       DSP code
```

While the ramset tag is being updated by the I/O bus, the CPU IBQ is prefetching instructions not present in ICACHE. There are some latencies in completing I/O access to ramset tag registers, the exact time of when ramset update happens is not controllable.

DSP ICACHE is Servicing a Missed Program Bus Request During a RAMSET Preload (Continued)

Workaround:

1. I-cache must be turned on with ramset(s) enabled.
2. Branch from external memory program code to the ramset tag update code.
3. Continue polling the ramset tag valid bit to make sure the ramset preload has finished.
4. Branch back to external memory for normal program execution after the preload has finished.

Example:

```
Address       Instruction
Ext Mem       DSP code
Ext Mem       DSP code
......
......
Ext Mem       DSP code
Ext Mem       DSP code
Ext Mem       *port(#0x1400) #0xce2f           ; Configure icache
Ext Mem       bit(ST3,#14) = #1                ; Turn on icache
Ext Mem       goto Preload_ramsets

Preload_ramsets:
Int Mem       *port(#0x1406) = #0x0800         ; Update ramset tag for bank1

Scan0:
Int Mem       TC1 = bit(*port(0x1405), #15)    ; Read the Tag Valid
                     ; bit of ramset bank 0
Int Mem       nop_16
Int Mem       if (!TC1) goto Scan0             ; Continue polling if the
                          ; Tag Valid bit is not 1.
Int Mem       *port(#0x1408) = #0x0801         ; Update ramset tag for bank1

Scan1:
Int Mem       TC1 = bit(*port(0x1407), #15)    ; Read the Tag Valid
                     ; bit of ramset bank 1
Int Mem       nop_16
Int Mem       if (!TC1) goto Scan1             ; Continue polling if the
                        ; Tag Valid bit is not 1.
Int Mem       goto Back_from_ramset_preload

Back_from_ramset_preload:
Ext Mem       DSP code
Ext Mem       I Ie
......
......
Ext Mem       DSP code
Ext Mem       DSP code
```

This exception will not be fixed in future silicon revisions.

4.2 DSP Emulation Advisories

Advisory DSP_EMU_1	*DSP IDLE Interrupt Not Serviced When Emulator is Connected*

Revision(s) Affected: All revisions

Details: At the JTAG interface level, an emulator device generates the TCK signal, which can be an adaptive or fixed clock signal. Inside the OMAP16xx devices, this TCK is re−synchronized by the ARM926 in order to create the RTCK signal provided to the emulator. This RTCK signal is so used by the emulator to send and receive data to OMAP synchronously with the OMAP internal clock.
This mechanism creates some limitations for the used TCK frequency, when the ARM/MPU clock runs at low frequency (which is the case when the DPLL is not running). Indeed, if we have fast TCK (5 ¿ 12 MHz) and low MPU clock, the 2 clock frequencies are too near. So the re−synchronization machine will provide a very bad and polluted RTCK at a frequency around 1 or 2 MHz.
With such an RTCK clock, the communication between OMAP and the emulator will fail most of the time.

Workaround: Slow down the TCK at 2 MHz: it will always works, regardless the state of the DPLL (anyway, when the DPLL is not enabled, the badly generated RTCK is never higher than 3 MHz). Yet, if you want use the fast TCK capability, we recommend using the following high−level sequence:

1. Use first slow TCK (<2 MHz) to debog boot process.
2. (Once the boot process is debugged, have boot code in non−volatile memory that programs DPLL for ARM clock >>12 MHz (noting that this clock can raise up to 192 MHz).
3. Use faster TCK (5−12 MHz).

This exception will not be fixed in future silicon revisions.

Advisory DSP_EMU_2

Bad RTCK Clock Signal May Cause JTAG Communication Failure between the OMAP and an Emulator

Revision(s) Affected: All revisions

Details: A DSP IDLE interrupt is not serviced when an emulator is connected as shown in the following sequence:

1. Emulator is connected
2. RT mode is set
3. Test code is run
   - ASM code sets INTM
   - ASM code clears all pending interrupts in IFR
   - ASM code configures timers, but does not enable
   - ASM code sets ICR to 0x3F
4. Emulator halts CPU
   - PC indicates halted at IDLE instruction
   - DBGSTAT = 0x805F (IDLE, DSUSP, and FXWOK)
   - ISR = 0x3F
   - No interrupts pending in IFR
5. Single interrupt is provided
6. RT int by emulation write to DBIER0 is enabled
   - Halted background code immediately comes out of idle
   - ISR = 0x3F (interrupt not taken)

Workaround: There is no workaround.

The emulator result is different from the functional mode (which will service the ISR). This only affects the debugger mode. Therefore, do not use IDLE when the emulator is connected.

This exception will not be fixed in future silicon revisions.

4.3 DSP System Advisories

Advisory DSP_SYS_1	*Use Caution When Reading Following a Configuration Change on the DSP*

Revision(s) Affected: All revisions

Details: Within the C55x CPU, writes occur later in the pipeline than reads do. This allows reads from a later instruction to sometimes occur prior to the write of an earlier instruction. This can be a problem when the write is to a configuration register in a peripheral that affects the read. If the user intends to read memory that can be affected by the new configuration, it is recommended that the user read from the configuration register being programmed prior to memory reads.

Reads from the TIPB interface are pipeline-protected, so writes and reads of peripheral registers are not affected. Also, sequential writes to peripherals work correctly.

Workaround: After configuring a peripheral, the software should ensure that the writes have completed prior to using that configuration. This can be done with a read from the last register written or waiting 3 cycles. This configuration constraint is common to pipeline architectures.

This exception will not be fixed in future silicon revisions.

Advisory DSP_SYS_2	*Users of (JTAG) RTDX are Advised Not to Put the DSP Domain Into Idle*

Revision(s) Affected: All revisions

Details: When the DSP is idled, ALL interrupts generated external to the DSP core are latched into a register within the OMAP subsystem. In this subsystem, the logic tied to the CLEAR input signal for the latch will erroneously cause the latch to be cleared upon emulation accesses, including CMSG/DMSG/DT-DMA.

This interrupt will cause the DSP to be awakened from Idle, but once it is awakened it will not find any interrupts flagged in IFR which require servicing. Thus, the interrupt will appear to have been lost. This latch is only used when the DSP domain is in idle. Thus, this issue may be avoided if the DSP domain is never idled.

Workaround: There is no workaround.

This exception will not be fixed in future silicon revisions.

Advisory DSP_SYS_3	*DSP IDLE Wakeup With Program Code in External Memory and EMIF Domain in IDLE Hangs*

Revision(s) Affected: All revisions

Details: If the EMIF domain and DSP domain are set to IDLE, and the IDLE instruction is executed, it is not possible for the wakeup ISR to be located in EMIF. The program counter will hang since the EMIF domain will not come out of IDLE.

Workaround: Ensure that wakeup ISR is executing from internal memory.

This exception will not be fixed in future silicon revisions.

Advisory DSP_SYS_4	*DM Timer Incorrect Value Read During DSP Access and Value Change*

Revision(s) Affected: All revisions

Details: The DM timer supports OCP accesses from both the ARM® and DSP. The timer registers are 32 bits wide, so two 16-bit accesses must be performed from the DSP side in order to read/write the full data. The correct procedure is to read the 16 LSBs first, then the 16 MSBs. However, when accessing TCRR or TCAR registers, there is no hardware mechanism to ensure that the 16 MSBs of the register will not change between the LSB reading and MSB reading.

Example: DSP reads TCRR as its value is 0x0001FFFF. LSB read returns 0xFFFF. TCRR value changes before MSB read (TCRR is now 0x0002 0000). MSB read will return 0x0002 instead of 0x0001.

Workaround: A read LSB-MSB while loop can be implemented in the DSP software in order to ensure correct data is provided by reading two consecutive identical MSB values.

This exception will not be fixed in future silicon revisions.

ARM is a registered trademark of ARM Limited.

Advisory DSP_SYS_5	*DSP STIO Interrupt Generation is Not Possible*

Revision(s) Affected: All revisions

Details: The DSP EMIF is a DSP subsystem module that gives the DSP access to the shared system memory managed by the traffic controller (bus wide is 32 bits). It supports interface to a variety of external devices, including:
− Asynchronous devices (asynchronous SRAM, ROM and Flash). The external memory interface provides highly programmable timing to these interfaces.
− Synchronous burst SRAM (SBSRAM) running at either 1x or 1/2x the CPU clock rate.
− Synchronous DRAM (SDRAM) running at either 1x or 1/2x the CPU clock rate.
− Synchronous STIO interface for OMAP applications.

The interrupt for the STIO interface is linked to the DSP Level 1 interrupt handler (SINT16/ INT16: peripheral interrupt #14). The problem is that at OMAP level, this STIO interrupt input is connected by hardware to an inactive state instead coming from the EMIF module; thus it is impossible to generate a STIO interrupt.

Workaround: No existing workaround.

This exception will not be fixed in future silicon revisions.

ARM is a registered trademark of ARM Limited.

4.4 DSP Direct Memory Access (DMA) Advisories

Advisory DSP_DMA_1	*DSP DMA IDLE Prevents Transfer Completion*

Revision(s) Affected: All revisions

Details: When DSP peripherals are placed in IDLE, there is an internal hardware handshaking mechanism between the DSP and its peripherals that ensures the IDLE can occur. If a DMA transfer is occurring, this handshaking is supposed to prevent IDLEing until after the transfer is complete. The DSP DMA, however, can go into IDLE during the middle of a transfer, resulting in the transfer not completing.

Workaround: In order to enforce that all DMA transfers be complete before attempting to IDLE the DMA, the DMA status first needs to be checked. Afterwards, the DMA channels need to be disabled from which the IDLE instruction can then be safely executed.

This exception will not be fixed in future silicon revisions.

Advisory DSP_DMA_2	*DSP EMIF/DMA Port Hangs During EMIF Bus Error*

Revision(s) Affected: All revisions

Details: If the EMIF times out on an access, the DSP will get a time-out bus-error interrupt. The time-out condition may also cause a DMA interrupt. During this DMA interrupt, the DMA will not time out and go into an unknown state.

Workaround: Whenever an EMIF bus error interrupt occurs, the software needs to RESET the DMA and reschedule the transfer

This exception will not be fixed in future silicon revisions.

5 MPU Subsystem Advisories

5.1 System Direct Memory Access (DMA) Advisories

Advisory SYS_DMA_1	*System DMA CSAC and CDAC Not Showing Correct Value Consistently*

Revision(s) Affected: All revisions

Details: Due to a malfunction of the CDAC/CSAC registers, software may read 0000 or incorrect data during large transfers and may not be able to judge the progress of transfers correctly.

These registers show incorrect values of address, for interleaving enabled, synchronized transfers, when read 7 cycles after the generation of a hardware request. At that time, the registers will be read as 0000. Similarly, incorrect data is read 8 cycles after the generation of the first hardware request.

Workaround: The workaround is to synchronize hardware requests and counter reads. After giving a hardware request, read the value after 8 clock cycles.

Normally the CSAC/CDAC registers will not give a value of zero (except when the starting address is 0000 and in constant address mode). Software should read the register(s) until the 0000 disappears.

This exception will not be fixed in future silicon revisions.

Advisory SYS_DMA_2	*Cannot Get SysDMA to Complete Data Transfer Out of UART*

Revision(s) Affected: All revisions

Details: The UART FIFO can interrupt the DMA channel upon setting register bits [2:0] in the supplementary control register (SCR), even if the UART is disabled (MDR1 = 0x7). This can cause problems if the SCR[2:0] bits are set prior to setting the trigger thresholds.

Workaround: Set the DMA just before setting the mode select in the MDR1 register to avoid filling the FIFO in disable mode.

This exception will not be fixed in future silicon revisions.

Advisory SYS_DMA_3	*DMA Synchronized Transfers Can Be Stalled*

Revision(s) Affected: All revisions

Details:

When request pipelining is enabled, synchronized DMA transfers can get stalled. (In order to allow a faster response of the DMA, a new request can start while a previous one is stil under processing.) The request pipelining is a DMA feature, automatically chosen when interleaving mode is disabled (DMA_LCH_CTRL[LID]=1).

The affected transfers are:

- Synchronized element transfers
- Synchronized frame transfers
- Synchronized block transfers

This problem is independent of:

- Transfer size
- Channel priority
- Used physical channel

The consequences/symptoms of this issue depend on the DMA request initiator and on the type of synchronized transfer (element/frame/block). For example:

- Event missed in element synchronized:
  If the exception occurs during a frame of N single elements, by a chain effect, halt will be moved on the last request. Thus, the last element of the frame will not be returned to the destination and a frame interrupt will not be generated.
- Transfer stalled in Frame synchronized:
  Some peripherals generate a new request only when a full frame has been read. Consequently, if this exception occurs, transfers will be stalled, as no more requests will be issued from the peripheral.

Workaround:

This hardware issue is a corner case of the request pipelining mode and never occurs in interleaving mode. So, software must enable the interleaving mode:
DMA_LCH_CTRL[LID] = 0.

This exception will not be fixed in future silicon revisions.

5.2 MPU Emulation Advisories

Advisory MPU_EMU_1	*Emulation is Intrusive to ARM Idle Mode: Cannot Stop the CPU Clock With Code Composer Studio Connected*

Revision(s) Affected: All revisions

Details: JTAG on the ARM926EJ-S™ does not work if the functional clock is halted. However, the processor can still enter the IDLE state when an emulator is attached.

Emulation is detected based on TCK activity. When emulation is disconnected, a power up is required to switch back to the no emulation attached default state. If the emulation is connected during a power-down state, then TCK detection will trigger a wakeup without interrupt generation.

Workaround: There is no workaround.

This exception will not be fixed in future silicon revisions.

ARM926EJ-S is a trademark of ARM Limited in the EU and other countries.

5.3 MPU System Advisories

Advisory MPU_SYS_1	*STANDBY Mode Not Entered After Execution of WFI Instruction*

Revision(s) Affected: All revisions

Details: STANDBY mode is not entered after execution of WFI instruction.

Workaround: There is no workaround.

This exception will not be fixed in future silicon revisions.

Advisory MPU_SYS_2	*Inefficient Fill of Instruction Prefetch Buffer*

Revision(s) Affected: All revisions

Details: Inefficient fill of instruction prefetch buffer.

Workaround: There is no workaround.

This exception will not be fixed in future silicon revisions.

Advisory MPU_SYS_3	*External Device Power Control is Broken*

Revision(s) Affected: All revisions

Details: If an external device power control is enabled (by setting EWUPCT.REPWR_EN bit to 0), the OMAP3.2 clock and reset manager might in some conditions shut down the ARM clock even when the ARM is not in standby mode.

The following sequence will cause this exception. Note that this is only a high-level sequence given for illustration purposes: there is no way to ensure proper operation of the system if the EWUPCT.REPWR_EN bit is set to 0.

External Device Power Control is Broken (Continued)

1. WFI instruction is executed. ARM goes into idle.
2. Clock and reset manager shuts down ARM clock and begins the OMAP3.2 idle entry procedure. This implies making sure that the Traffic Controller is idle.
3. Before the clock and reset manager asserts the idle request to the traffic controller, a wake-up event occurs and wakes up the system. Clock and reset manager activates ARM and TC clock. ARM begins fetching code.
4. Since the REPWR_EN is set, the clock and reset manager shuts down the ARM clock while TC clock stays active, and begins counting up until EWUPCT.EXTPWR cycles have elapsed. Then, the ARM clock is restored. This has two effects:
   a) Shutting down ARM clock while it is active and fetching code.
   b) Asserting FLASH.RP signal while the ARM is fetching code.

Workaround: Set REPWR_EN bit inside EWUPCT register (0xFFFE CE0C) to 1. This disables the external device power control feature, and prevents the exception from occurring.

This exception will not be fixed in future silicon revisions.

Advisory MPU_SYS_4 *Emulation Causes OMAP Not to Enter Deep Sleep*

Revision(s) Affected: All revisions

Details: Rising edge on TCK causes OMAP not to enter Deep Sleep.

Workaround: Make sure that daisy-chaining other devices onto the OMAP5912 JTAG will not prevent OMAP from entering deep sleep by ensuring other JTAG device do not glitch TCK or provide a rising edge on TCK upon power up.

If a glitch propagates, either apply a power-up reset or a JTAG reset.

This exception will not be fixed in future silicon revisions.

Advisory MPU_SYS_5 — *Write Followed by Immediate Read not Supported on the Rhea Switch Peripheral*

Revision(s) Affected: All revisions

Details: A write followed by an immediate read does not work on the ARM address space defined below. If a read occurs immediately after the write to the same address, then the read will not be the data that was written. The affected address spaces are:

- Rhea Switch module: Address FFFB:C800 to End Address FFFB:CFFF

Workaround: When an immediate read is required after a write to any register in the above address space, make two consecutive writes to the same address prior to the read. Using this procedure ensures that the first write completes and the read data will be correct.

This exception will not be fixed in future silicon revisions.

Advisory MPU_SYS_6 — *When Absent, the OMAP Peripheral Clock Prevents Deep Sleep Entry*

Revision(s) Affected: All revisions

Details: If the peripheral clock is disabled, OMAP5912 cannot go into deep sleep when executing the ARM Idle instruction. If the peripheral clock is enabled prior to the execution of the ARM Idle instruction, the chip goes into deep sleep properly.

Workaround: If the ARMPER peripheral clock is enabled (via EN_PERCK of ARM_IDLECT2) prior to the execution of the ARM Idle instruction, the chip goes in deep sleep correctly.

If the ARMPER peripheral clock is disabled, make sure that the TC1 clock is disabled as well.

This exception will not be fixed in future silicon revisions.

Advisory MPU_SYS_7 — *MPU IRQ122−127 Can Cause Spurious Interrupts*

Revision(s) Affected: All revisions

Details: On the MPU, IRQ122 through IRQ127 are tied low and therefore, capable of interrupting the system if their corresponding ILR registers (from address 0xFFFE:0384 for IRQ122 to 0xFFFE:0398 for IRQ127) are programmed such as a level interrupt.

A level interrupt is configured if the $\overline{\text{SENS_EDGE}}$ bit is set to 1 (bit 1 of the ILR register).

Workaround: Make sure $\overline{\text{SENS_EDGE}}$ is cleared to 0 for IRQ122−IRQ127's IRL registers (from address 0xFFFE:0384 for IRQ122 to 0xFFFE:0398 for IRQ127).

This exception will not be fixed in future silicon revisions.

Advisory MPU_SYS_8 — *If BVLZ is Not Configured, the Corresponding Interrupt Must be Masked*

Revision(s) Affected: All revisions

Details: On OMAP5912, the UART3.CTS, UART1.DSR, and the MMC/SDIO2's MMC.DATDIR1 signals have been multiplexed with BVLZ (battery low-level external interrupt).

If BVLZ is not used, make sure that the corresponding interrupt mask is set (MPU Level 1 Interrupt Mapping, IRQ_3) and masked or controlled accordingly the RST_HOST_OUT (modem shutdown output) on ball W13 (ZZG package).

Workaround:

Case 1:
Use W19 (ZZG package) as EXT_FIQ (battery fail), use W13 as modem shutdown.

Case 2:
Use W19 as anything, provided the corresponding interrupt is masked. Use W13 as something that is not modem shutdown.

Case 3:
Use W19 as something different from EXT_FIQ, provided that the corresponding interrupt is masked. Use W13 as modem shutdown.

In Case 3, the workaround is to pull W13 high externally and configure it by software to be in mode 7 (GPIO55). This will prevent activity on W19 from propagating to W13.

Then, when software wants to assert W13, set the ULPD POWER_CTRL_REG[3] bit to 0, and reconfigure pin muxing for W13 to be modem shutdown. When software wants to deassert W13, reconfigure pin muxing to have W19 be GPIO55. Pullup should revert the line to high (inactive) state.

If using a pullup on Ball W13 is impossible, then configure the GPIO55 as an output driving 1.

This exception will not be fixed in future silicon revisions.

Advisory MPU_SYS_9	*Access to the OMAP3.2 ARM_CLKM Registers*

Revision(s) Affected: All revisions

Details: The ARM_CLKM has 32-bit width registers. Make sure that the ARM_CLKM registers are accessed with 32-bit access widths. If the ARM CLKM registers are accessed in 16 bits, the access will be correctly performed, but a TIPB abort will be generated.

Workaround: Do not access the ARM CLK registers with 16 bit access width (or mask TIPB aborts).

This exception will not be fixed in future silicon revisions.

Advisory MPU_SYS_10	*WFI Instruction Loses 2−4 Instruction Fetches*

Revision(s) Affected: All revisions

Details: When the ARM tries to wake up immediately after execution of the WFI instruction or a few cycles later, it may miss some instruction fetches. The number of instructions missed depends on the time of the wake-up interrupt. This problem does not depend on the EMIFS configuration and can occur with any configuration of the EMIFS.

The problem occurs as follows:

1. ARM goes to idle after execution WFI instruction.
2. OMAP clock and Reset module cuts clock to the ARM.
3. At the same time, Clock and Reset module asserts TCIDLE_Req to the Traffic controller.
4. TC acknowledges the idle request from clock and reset module.
5. Meanwhile, a wake-up interrupt comes to OMAP.
6. After a few synchronization cycles, the ARM clock cycles wake up.
7. In parallel, the clock and reset module synchronizes the TC idle acknowledge and cuts the clock to the TC.
8. Because the ARM clock is active and IRQ/FIQ is present, the ARM tries to fetch instructions from the EMIFS (TC).
9. This results in the following two cases:
   a) Request comes before the Clock and Reset module cutting the clock to the TC. . . . . Because the Idle acknowledge is present, the Clock and Reset module will cut the . . . . . clock to TC. [If the clock cut in the middle of request, ARM has the possibility of . . . . . missing some instructions (9a).]
   b) TC clock already cut and request from ARM waits until TC clock is active. . . . . (No problem to any instruction fetch.)

WFI Instruction Loses 2−4 Instruction Fetches (Continued)

Workaround: The software workaround is to make sure the that ARM will only execute NOPs when it wakes up, inserting NOPs after the WFI instruction, and making sure interrupts are disabled for ARM when the WFI instruction is executed. See below:

1. Disable interrupts, by setting I and F bits in ARM CPSR register. This ensures that the ARM will not jump to interrupt routine when it wakes up.
2. WFI instruction
3. NOP instructions
4. Enable interrupts by resetting I and F bits in the ARM CPSR. The number of NOPs to insert after the WFI instruction is given by the following formula:

   X = 2 + (4*DPLL_MULT)/DPLL_DIV/ARMDIV

This exception will not be fixed in future silicon revisions.

5.4 MPU/Watchdog Timer Advisories

Advisory MPU_WATCHDOG_1	*MPU Watchdog Does Not Reset DPLL*

Revision(s) Affected: All revisions

Details: If the MPU watchdog timer expires, a reset is provided to the ARMCK_CTL registers (resetting the clock divisors) but not to the DPLL.

This means that when OMAP5912 is operating at 192 MHz with SDRAM at 96 MHz (/2) and a WD timer reset occurs, the ARM will begin executing code at the reset vector with the DPLL remaining at 192 MHz and all clock divisors set at /1. See below:

- Before reset:
  - DPLL register (0xFFFE CF00) = 0x2813 (DPLL Lock mode, 16 x multiplier)
  - ARM_SYSST register (0xFFFE CE18) = 0x1000 (Synchronous scalable mode)
  - ARM_CKCTL register (0xFFFE CE00) = 0x5101 (ARMINTH, TC. PER clocks div/2; clkref = CLKGen1)
- After reset:
  - DPLL register (0xFFFE CF00) = 0x2813 − No change
  - ARM_SYSST register (0xFFFE CE18) = 0x000C (Full synchronous mode, WD + MPU reset)
  - ARM_CKCTL register (0xFFFE CE00) = 0x3000 (No div; clkref = CLKGen1)

Workaround: Use the 32k watchdog timer. It properly resets the DPLL when it expires; however, the 32k WD Timer has less precision on the timer value (it counts 30.5 µs ticks instead of ~0.1 µs ticks) and cannot cause a WD reset to occur in less than 30.5 µs.

This exception will not be fixed in future silicon revisions.

Advisory MPU_WATCHDOG_2

The Secure Watchdog Register Reads Can Return Wrong Value Under Some Clock Configurations

Revision(s) Affected: All revisions

Details: All the OMAP16xx Watchdog timers (32 KHz watchdog and secure watchdog) support only POSTED internal synchronization mode. There is no capability to change the internal synchronization scheme to NON−POSTED by software; therefore, their OCP clock must be at least 4 times faster than their functional clock (system clock or 32K clock).
In OMAP16xx clock configurations where this condition is not met (interface clock is ARMPER_CK i.e. CK_GEN1 divided by 1, 2, 4 or 8 as per ARM_CKCTL.PERDIV[1:0] setting), met stability can occur and cause erroneous value to be read from WCLR, WCRR, WLDR, WTGR and WSPR registers. This can cause problems with the secure WD, since its functional clock frequency is 12MHz/13MHz or 19.2MHz (ARMXOR_CK). The 32 KHz watchdog is not impacted, as its functional clock is 32 KHz. Interface clock frequency is always greater than 4 times the functional clock.

Workaround: Software must perform several read accesses in order to detect and discard possible erroneous values.

This exception will not be fixed in future silicon revisions.

6 Traffic Controller Subsystem Advisories

6.1 EMIF Slow (EMIFS) Advisories

Advisory EMIFS_1	*Unexpected FCLK on EMIFS When Dynamic Wait State is Selected*

Revision(s) Affected: All revisions

Details: When the EMIFS_DWS register is configured to select dynamic wait states, an unexpected FCLK is generated after an access to the CS which is configured to dynamic wait state mode.

Workaround: Avoid dynamic wait ready mode. If the dynamic wait ready mode has to be used, insert a dummy write/read cycle.

This exception will not be fixed in future silicon revisions.

Advisory EMIFS_2	*Program Fetch From External Memory at Address 0x3C18 Results in Processor Reset*

Revision(s) Affected: All revisions

Details: If the EMIFS_CONFIG.BM bit is set to 1, software must make sure no code is ever executed in the address range centered on 0x0000 3C18 ± 10 32-bit words. When the EMIFS_CONFIG.BM bit is set, this address range is located in external memory (EMIFS) connected to FLASH.CS3. Depending on the device type, executing code in the forbidden area will have different consequences:

- On emulation devices, the processor will hang up
- On production devices (general-purpose), the processor will reset (warm reset generated).

This behavior will only affect devices where the BM bit is set. Emulation devices booting externally and GP devices using Fast Boot mode have this bit set by default.

Workaround: If the EMIFS_CONFIG.BM bit is set, software has to stay away from 0x0000 3C18 ± 10 32-bit addresses (need to account for ARM926EJ-S™ pipeline architecture).

This exception will not be fixed in future silicon revisions.

ARM926EJ-S is a trademark of ARM Limited in the EU and other countries.

6.2 EMIF Fast (EMIFF) Advisories

Advisory EMIFF_1	*MRS/EMRS Command Issued Twice SDRAM Interface*

Revision(s) Affected: All revisions

Details: When the ACCESS_FACTOR1 in the RHEA CNTL register is configured for values strictly greater than 1, unpredictable behavior can occur on the SDRAM bus and multiple MRS commands are issued by the Traffic Controller.

The following registers have the above mentioned problem:

- MRS Registers (0xFFFE CC24)
- MRS New register (0xFFFE CC70)
- Manual command register (0xFFFE CC84)
- EMRS0 register (0xFFFE CC74)
- EMRS1 Register (0xFFFE CC78)
- EMRS2 Register (0xFFFE CCC8)

Workaround: Make sure that the ACCESS_FACTOR1 (bit field 7:4) in the RHEA CNTL register (0xFFFE:CA00) is set to 1 so that the strobe_1 period on the TIPB bus is equal to half the TIPB clock period.

This exception will not be fixed in future silicon revisions.

Advisory EMIFF_2	*Limitation of the EMIFF Not Mentioned in Reference Guide*

Revision(s) Affected: All revisions

Details: The following limitation of the EMIFF is not mentioned in the *OMAP5912 Multimedia Processor OMAP3.2 Subsystem Reference Guide* (literature number SPRU749):

Software must only write to EMIFF manual command register (EMIFF_CMD) at initialization or just before/after entry to deep sleep. Any write to this register at other times will result in unpredictable behavior.

Workaround: There is no workaround.

This exception will not be fixed in future silicon revisions.

Advisory EMIFF_3	*Using EMIFF POM0 Mode May Result in Data Read Corruption*

Revision(s) Affected: All revisions

Details: When using the EMIFF POM0 mode of operation, an unnecessary precharge command (PRE) may sometimes be issued to the SDRAM/DDR memory one cycle after a read (RD) command. If the PRE command occurs on the same bank as the RD command, then the RD command will be terminated too early, resulting in a data read corruption.

Workaround: There is no software workaround.

Do not use POM0 mode. HPHB and LPLB mode are not impacted by this limitation.

This exception will not be fixed in future silicon revisions.

Advisory EMIFF_4	*Only 72° DLL is Supported on the EMIFF*

Revision(s) Affected: All revisions

Details: The OMAP5912 EMIFF only supports 72° DLL.

Workaround: There is no software workaround.

Do not use 90° DLL.

This exception will not be fixed in future silicon revisions.

Advisory EMIFF_5	*The EMIFF Autoclock Feature Affects the DLL Behavior*

Revision(s) Affected: All revisions

Details: The external memories interface fast (EMIFF) module includes digitally controlled delay technology to meet the strict timing requirements for interfacing high-speed, double-data-rate memory components. This technology requires a delay-locked loop (DLL) to control several digitally-controlled delay lines (DCDLs) by providing a specific continuously updated delay value, taking into account the process, temperature, and voltage variations (PVT tracking). When the EMIFF_CONFIG[CLK] and EMIFF_SDRAM_CONFIG_2_REG[SD_AUTO_CLK] bits are set to 1 in the EMIFF registers, the DLL input clock is gated and works only when accessing the SDRAM.

The consequences of this capability are:

- A huge DLL lock time is noticed at startup (at least 40,000 TC clock cycles), depending on the refresh time of the SDRAM.
- Due to the clock wake-up time, a lesser reaction speed in PVT tracking is noted compared with DLL clock always ON.
- If the DLLLOAD happens during DLL clock OFF, a failure takes place in the delay value initialization.

Workaround: At startup, the SD_AUTO_CLK must be set to 0. The DLL is set to lock mode and provides the required DCDL (PVT tracking) value. The system must ensure that no SDRAM accesses occur during this time. Then the DLL setting is changed to unlock mode and the SD_AUTO_CLK is again set to 1.

This exception will not be fixed in future silicon revisions.

6.3 Traffic Controller Advisories

Advisory TC_1	*OMAP5912 Global Software Reset Affects Traffic Controller Frequency*

Revision(s) Affected: All revisions

Details:

At power-up reset, the traffic controller (TC) clock is the system clock (base frequency is 19.2 MHz, 12 MHz, or 13 MHz) and in fully synchronous mode (ARM clock = DSP clock = TC clock). During boot, if the user decides to do the following:

- Switch to synchronous scalable mode (ARM clock = DSP clock, TC clock = ARM clock/2) and enable the DPLL (system clock x 10 = 192-MHz clock if the base frequency is 19.2 MHz),
- Perform a global software reset by:
  - setting the SW_RST bit (bit 3 in the ARM_RSTCT1 register), *or*
  - setting the ARM RST bit (bit 0 in the ARM_RSTC1 register) and then clearing the DSP_EN bit (bit 1 in the ARM_RSTC1 register).

Then, the DPLL is still enabled and OMAP5912 is switched back automatically to fully synchronous mode (ARM_SYSST register is reset). This causes ARM clock = DSP clock = TC Clock =192 MHz. 192 MHz is too high a frequency for the TC to operate and can potentially cause incorrect data to be fetched. See Table 1.

Table 1. Reset Sources for the Clock Module and the DPLL

	EVENT							
	Cold Reset†	Warm Reset‡	Global System Reset§	ARM WD Reset	DSP WD Reset	MPU Software Reset	DSP Software Reset¶	DSP Software Reset#
MODULE								
CLKM_1	Yes	Yes	Yes	Yes	No	No	No	No
CLKM_2	Yes	Yes	Yes	Yes	No	No	No	No
CLKM_3	Yes	Yes	Yes	Yes	No	No	No	No
DPLL_1	Yes	Yes	No	No	No	No	No	No
DPLL_2	Yes	Yes	No	No	No	No	No	No

† Power-up reset or on_noff
‡ MPU_nreset, secure WD reset, or 32-kHz WD reset
§ SW_RST (Bit 3 in ARM_RSTCT1) is written to 1, or set ARMRST (Bit 0 in ARM_RSTC1) and clear DSP_EN (Bit 1 in ARM_RSTC1)
¶ DSP_EN (Bit 1 in ARM_RSTC1) is written to 0
# DSP_RST (Bit 2 in ARM_RSTC1) is written to 1

OMAP5912 Global Software Reset Affects Traffic Controller Frequency (Continued)

Workaround: Before performing a Global Software Reset, it is recommended that the DPLL be bypassed so that after the clock module has reset to fully synchronous mode, the TC frequency is equal to the OMAP5912 base frequency.

This exception will not be fixed in future silicon revisions.

7 OMAP5912 Peripheral Advisories

7.1 Multimedia Card/Secure Digital (MMC/SD) Advisories

Advisory MMC_SD_1	*CMD12 Implementation in TI MMC/SDIO*

Revision(s) Affected: All revisions

Details: The Multimedia Card System specification (Revision 3.2) states that while CMD12, the STOP_TRANSMISSION command, is issued by the Host, valid data can still be received or sent (refer to Figure 28 and Figure 31 of the MMC card specification).

The TI MMC/SDIO IP state machine will block any CMD12 command as soon as the receive FIFO is full to avoid the loss of a valid read data.

Similarly, in case of a write sequence, the TI MMC/SDIO state machine will block any CMD12 command if the transmit FIFO is empty.

So, all Data Read or Write commands **cannot** be aborted **any time** by the CMD12 command.

This behavior is applicable for both DMA and Interrupt modes.

Workaround: During a read sequence, make sure the receive FIFO is not full before issuing a CMD12 Stop command. This can be done by performing some dummy reads from the FIFO card after CMD12 is issued (MPU reads directly in the MMC FIFO register).

During a write sequence, make sure the transmit FIFO is not empty before issuing a CMD12 Stop command. This can be done by performing some dummy writes to the FIFO after CMD12 is issued (MPU writes directly in the MMC FIFO register). Be aware those dummy written data can corrupt the card data. The CMD12 command remains an abort procedure.

This exception will not be fixed in future silicon revisions.

Advisory MMC_SD_2	*SPI Mode Not Supported on the MMC/SD Peripheral*

Revision(s) Affected: All revisions

Details: SPI mode does not work on OMAP5912. Do not use the peripheral in this mode.

Workaround: There is no workaround.

This exception will not be fixed in future silicon revisions.

Advisory MMC_SD_3 — *MMC1 Internal Pullups are Not Software-Controllable*

Revision(s) Affected: All revisions

Details: When I/O multiplexing is configured for MMC1:

- MMC1.DATx and MMC1.CMD do not have any internal pulldowns
- MMC1.DATx and MMC1.CMD can be configured to have internal pullups

This is true for MMC1.DATx and MMC1.CMD, whatever ball they are propagated to. For example, MMC1.DAT2 can be on two different balls, M15 (ZZG) and W10 (ZZG). These internal pullups are controlled directly by the MMC1 peripheral hardware: there is no way to control them by software. When there is no transaction, the MMC1 peripheral enables the pullup control by default.

Workaround: In order to enable the MMC1 peripheral to control its pullups, the following steps must be performed:

1. Configure FUNC_MUX_CTRLx for MMC1.DATx or MMC1.CMD
2. Configure PU_PD_SEL and PU_PD_EN to enable a pullup on the line

This exception will not be fixed in future silicon revisions.

Advisory MMC_SD_4 — *CMD1 Implementation in TI MMC/SDIO IP May Cause Missing EOC Interrupt*

Revision(s) Affected: All revisions

Details: The MMCA systems specifications (revision 3.2) states the sequence CMD1−R3 can only be monitored with EOC and MMC_STAT[OCRB] bits. The default CMD1 implementation in the MMC/SDIO host includes the OCRB interrupt capability, which can cause a problem with the EOC interrupt capability (implemented by default with the classical ones: CTO, CCRC, CERR). At the host module level, OCRB and EOC are set almost together, and there is only 1 OCP clock cycle separating them. It is unlikely that the software will detect one, whereas the other is not already asserted. The problem is EOC interrupt can be missed.

Workaround: Masking the OCRB interrupt capability prevents the system from failing. The software must be set to disable the OCRB interrupt capability to monitor this CMD1−R3 sequence at MMC/SDIO host level.

This exception will not be fixed in future silicon revisions.

7.2 MICROWIRE Advisories

Advisory UWIRE_1	*MICROWIRE Interface RX Data Failures Possible*

Revision(s) Affected: All revisions

Details: See the table below and Figure 5 and Figure 6.

	CSx_EDGE_RD = 0 CSx_EDGE_WR = 0	CSx_EDGE_RD = 1 CSx_EDGE_WR = 1	CSx_EDGE_RD = 0 CSx_EDGE_WR = 1	CSx_EDGE_RD = 1 CSx_EDGE_WR = 0
SCLK Not Inverted	Data read on falling edge of SCLK. Data write on falling edge of SCLK. Delay of 2 SCLK cycles between last bit transmitted on falling edge and first bit captured on falling edge. No issue.	Data read on rising edge of SCLK. Data write on rising edge of SCLK. Delay of 2 SCLK cycles between last bit transmitted on rising edge and first bit captured on rising edge. No issue.	Data read on falling edge of SCLK. Data write on rising edge of SCLK. Delay of 1.5 SCLK cycles between last bit transmitted on rising edge and first bit captured on falling edge. **Issue found (see Figure 5):** One extra bit is captured. **Workaround:** Ignore Bit 0 of the receive data register (shift right one bit). Note that this workaround does not work for a 16-bit data configuration (1 bit is lost).	Data read on rising edge of SCLK. Data write on falling edge of SCLK. Delay of 2.5 SCLK cycles between last bit transmitted on falling edge and first bit captured on rising edge. **Issue found (see Figure 6):** First bit captured will start 2.5 SCLK cycles after the last bit sent instead of 1.5 SCLK as defined in the specifications. **Workaround:** Do not use this configuration. Use one of the alternate configurations available for this module.
SCLK Inverted	Data read on rising edge of SCLK. Data write on rising edge of SCLK. Delay of 2 SCLK cycles between last bit transmitted on rising edge and first bit captured on rising edge. No issue.	Data read on falling edge of SCLK. Data write on falling edge of SCLK. Delay of 2 SCLK cycles between last bit transmitted on falling edge and first bit captured on falling edge. No issue.	Data read on rising edge of SCLK. Data write on falling edge of SCLK. Delay of 1.5 SCLK cycles between last bit transmitted on falling edge and first bit captured on rising edge. **Issue found:** One extra bit is captured. **Workaround:** Ignore Bit 0 of the receive data register (shift right one bit). Note that this workaround does not work for a 16-bit data configuration (1 bit is lost).	Data read on falling edge of SCLK. Data write on rising edge of SCLK. Delay of 2.5 SCLK cycles between last bit transmitted on rising edge and first bit captured on falling edge. **Issue found:** First bit captured will start 2.5 SCLK cycles after the last bit sent instead of 1.5 SCLK as defined in the specifications. **Workaround:** Do not use this configuration.

MICROWIRE Interface RX Data Failures Possible (Continued)

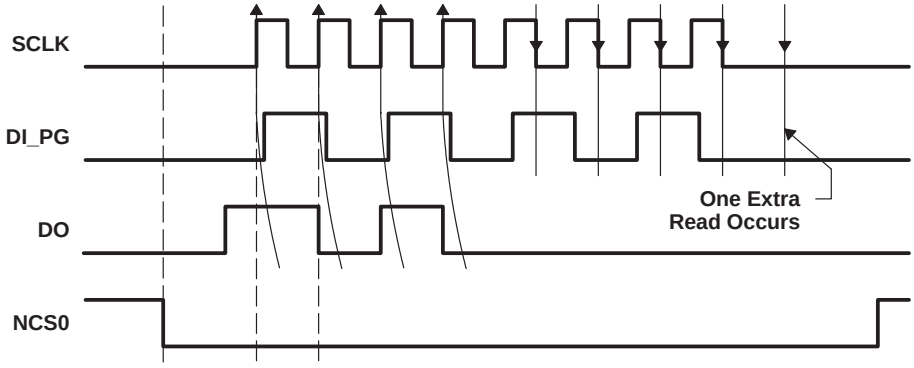

Figure 5. Write (4 Bits) on Rising/Read (4 Bits) on Falling (TX = 0xA, RX and 0x1F = 0x14)

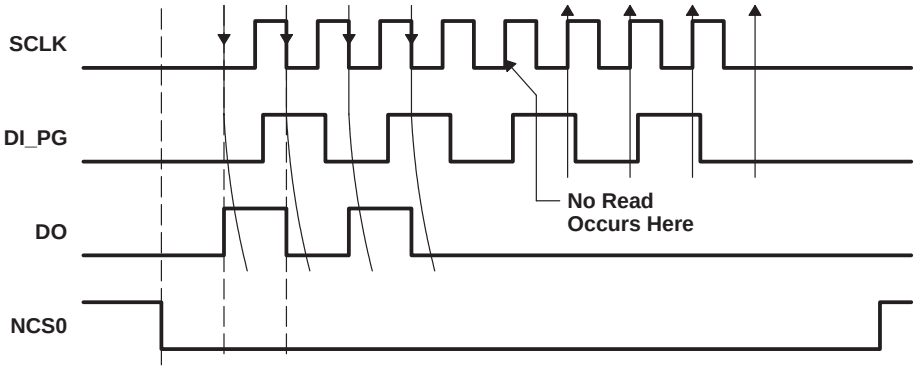

Figure 6. Write (4 Bits) on Falling/Read (4 Bits) on Rising (TX = 0xA, RX and 0xF = 0xA)

Workaround: There is no workaround.

This exception will not be fixed in future silicon revisions.

Advisory UWIRE_2	*Bit 14 of CSR Not Reset*

Revision(s) Affected: All revisions

Details: MICROWIRE® register bit CSR[14] (CSRB) is only updated/reset when MICROWIRE® internal clock is running.

Workaround: Ensure SR3[0] (CLK_EN bit) is high (i.e. MICROWIRE® internal clock is active) at least three MICROWIRE® clock cycles (FCLK) before reading CSRB bit.

For example: if SR3[2:1] = 0x3 (FCLK = ARM_XOR_CK/10), then CSRB will have a correct value of 30 ARM_XOR_CK cycles after setting CLK_EN bit.

If software does not dynamically turn on/off MICROWIRE® clock by resetting and then setting SR3.CLK_EN, then the problem only occurs the first time SR3.CLK_EN is set. Otherwise, the software needs to wait each time it sets the SR3.CLK_EN bit.

This exception will not be fixed in future silicon revisions.

MICROWIRE is a registered trademark of National Semiconductor Corporation.

7.3 Inter-Integrated Circuit (I^2C) Advisories

Advisory I2C_1	*I^2C Random Duplication of Byte When Sending Long Messages*

Revision(s) Affected: All revisions

Details: An extra byte can be transmitted by the I^2C peripheral if the I^2C FIFO is written while an I^2C transmission is in progress. For instance:

- The following byte sequences are written into the I^2C FIFO: 0x9190, 0x9392, 0x9594
- The following data sequences get sent out by the I^2C peripheral: 0x90, 0x91, 0x92, 0x92, 0x93, 0x94.

If the FIFO is empty when the write is performed, this exception does not occur.

Workaround: No workaround exists if the DMA is used.

There is a software workaround for interrupt and polling modes only: software must wait for the transmit underflow bit (I2C_STAT.XUDF) to be set before writing new data to the FIFO. This has an impact on performance as the FIFO needs to be reloaded between two transmissions.

Figure 7 and Figure 8 show the flowcharts of master transmit operation in polling and interrupt modes.

I2C Random Duplication of Byte When Sending Long Messages (Continued)

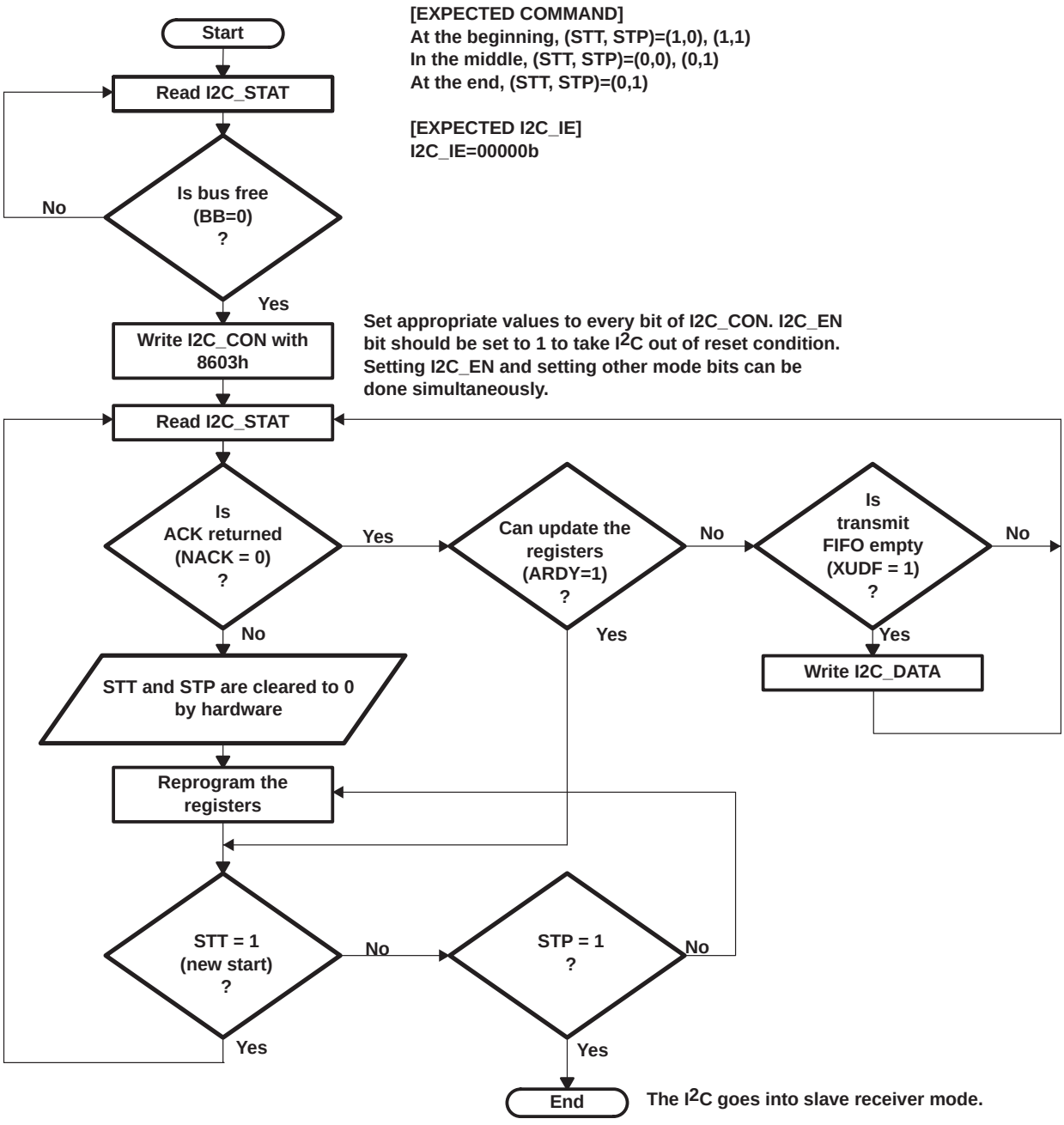

Figure 7. Master Transmitter Mode Polling Flow Diagram

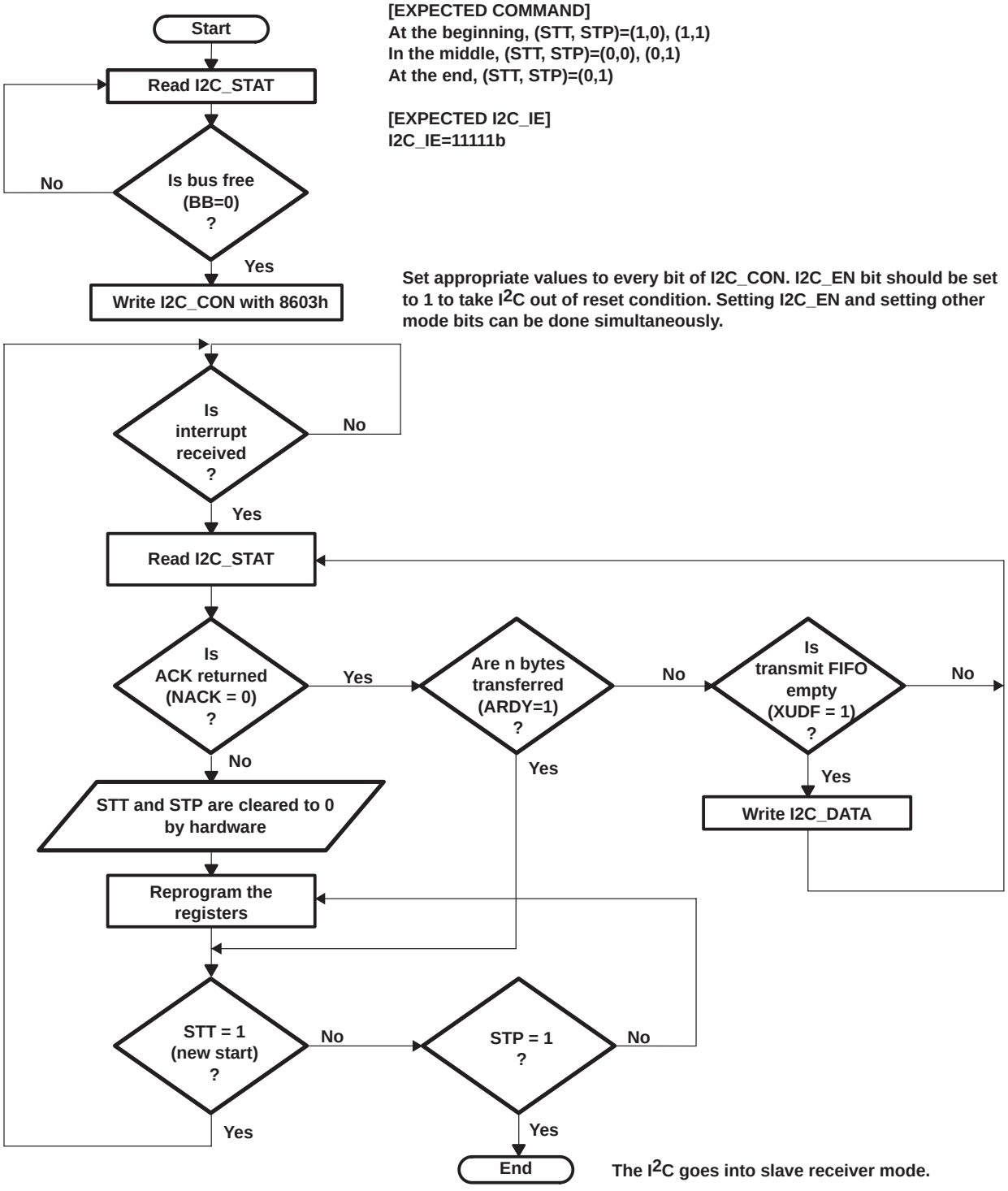

Figure 8. Master Transmitter Mode Interrupt Flow Diagram

I2C Random Duplication of Byte When Sending Long Messages (Continued)

Figure 9 and Figure 10 show the flowcharts of slave transmit/receive operation in interrupt and polling modes.

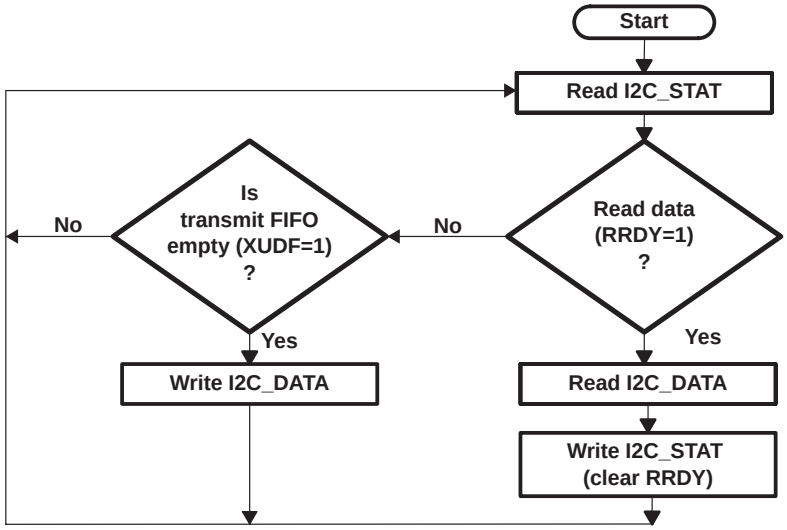

Figure 9. Slave Transmitter/Receiver Mode Polling Flow Diagram

I2C Random Duplication of Byte When Sending Long Messages (Continued)

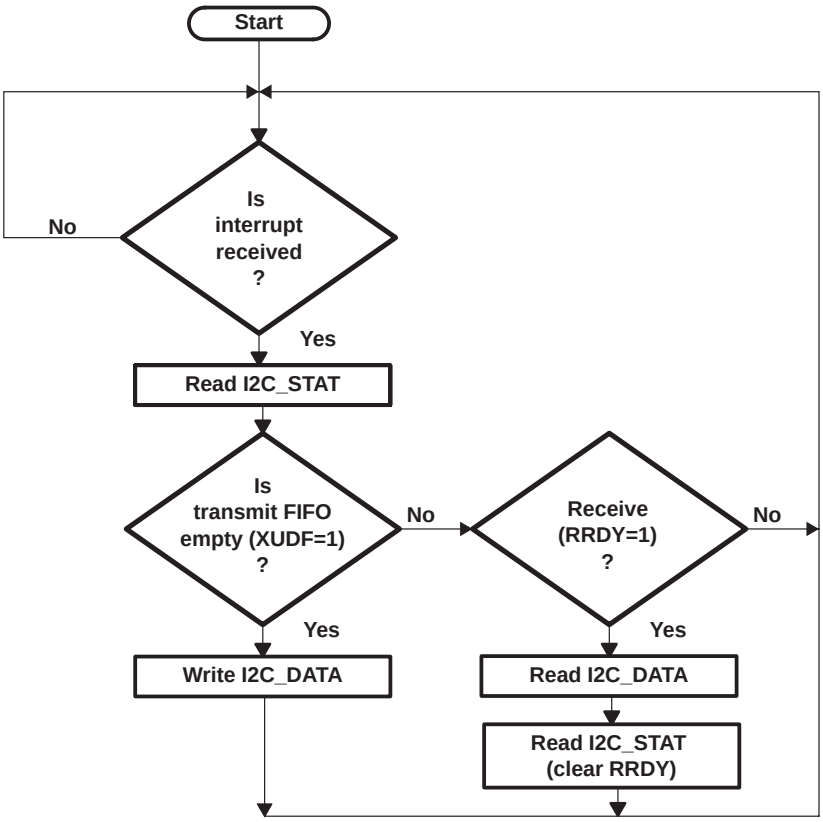

Figure 10. Slave Transmitter/Receiver Mode Interrupt Flow Diagram

This exception will not be fixed in future silicon revisions.

7.4 Universal Serial Bus (USB) Host Advisories

Advisory USB_HOST_1	*USB Host Fails to Receive the First Packet Sent After Remote Wakeup*

Revision(s) Affected: All revisions

Details: When using the HCFS (Host controller Functional State) bit field in the HC_CONTROL register to force the USB Host in Global Suspend, the internal USB Host State Machine cannot recover properly during the resume phase and the first packet sent by the device is lost.

Workaround: In order to put a given USB port in Suspend State, it is recommended to use the PSS bit field of the HcRhPortStatus instead of the HCFS bit field in the HC_CONTROL register.

This exception will not be fixed in future silicon revisions.

Advisory USB_HOST_2	*USB Device Controller Remote Wake−Up Feature May Fail*

Revision(s) Affected: All revisions

Details: The remote wake−up feature allows a USB device controller to wake a USB host from the suspend state. The USB host must be correctly configured to enable this feature. In application, the remote wake−up starts when the SYSCON2[RMT_WKP] bit is set to 1, and only after the USB host has set the DEVSTAT.R_WK_OK bit. By design, DEVSTAT.R_WK_OK will be set only if SET_FEATURE remote wake−up request has been auto−decoded; thus, this remote wake−up feature is not possible in auto−decode disabled mode (SYSCON1[AUTODEC_DIS]=1).

Workaround: There is no workaround.

This exception will not be fixed in future silicon revisions.

7.5 Universal Serial Bus (USB) On-the-Go (OTG) Controller Advisories

Advisory USB_OTG_1	*Specification for TA_WAIT_BCON*

Revision(s) Affected: All revisions

Details: According to the USB OTG Supplement to USB2.0 specification, Revision 1.0, the minimum value of TA_WAIT_BCON is 1.2 seconds. However, the USB peripheral has this fixed at 200 ms.

Workaround: A software workaround for this is to ignore the first five time-out interrupts, giving an actual time of 6*200 ms = 1.2 seconds.

This exception will not be fixed in future silicon revisions.

Advisory USB_OTG_2	*USB OTG TA_WAIT_VRISE is Set at 200 ms*

Revision(s) Affected: All revisions

Details: According to the On-The-Go Supplement to the USB 2.0 specification Revision 1.0a, wait for the BUS rise time (TA_WAIT_VRISE) should be a maximum of 100 ms. In the OMAP5912 USB OTG peripheral, the TA_WAIT_VRISE is fixed at 200 ms.

Workaround: The workaround is to emulate the time-out in software:

1. Check if VBUS is driven (OTG_CTRL.OTG_DRV_VBUS = 1). If yes, launch a timer (dual-mode timer or 32K OS timer) to wait for the desired time-out value, which should be less than 100 ms.
2. When the timer overflows, check if the VBUS is valid (OTG_CTRL.VBUSVLD bit). If OTG_CTRL.OTG_VBUSVLD is 0, then set OTG_CTRL.OTG_BUSDROP.

This exception will not be fixed in future silicon revisions.

Advisory USB_OTG_3 *USB OTG Timings Do Not Comply With USB OTG Specifications Rev. 1.0a*

Revision(s) Affected: All revisions

Details: OMAP5912 is compliant with USB OTG specification revision 1.0. The USB OTG specification revision 1.0a has significantly changed some of the protocol specific timings. The following timings do not comply with USB OTG specification 1.0a:

TIMING	VALUE IN USB OTG SPECIFICATIONS 1.0 (OMAP5912 IMPLEMENTATION)	VALUE IN USB OTG SPECIFICATIONS 1.0a
TA_BCON_LDB	No minimum (no timer implemented)	100 ms minimum
TB_AIDL_BDIS	3 ms minimum	5 ms minimum
TA_AIDL_BDIS	150 ms minimum	200 ms minimum
TA_WAIT_BCON	200 ms minimum	1 s minimum
TA_WAIT_VRISE	200 ms	100 ms maximum

Workaround: There is no workaround except for TA_WAIT_VRISE (see Advisory USB_OTG_2).

This exception will not be fixed in future silicon revisions.

Advisory USB_OTG_4 *USB OTG DMA Requests Control Bits Cannot Be Read Back*

Revision(s) Affected: All revisions

Details: Bits RX_REQ and TX_REQ in RXDMA_CFG and TXDMA_CFG registers will always read as 0. Writing to these bits will have the desired effect, but reading them will always return 0. This can cause some software problems when using read−modify−write on RXDMA_CFG and TXDMA_CFG registers.

Workaround: Do not perform read−modify−write access to RXDMA_CFG and TXDMA_CFG registers.

This exception will not be fixed in future silicon revisions.

Advisory USB_OTG_5	*USB OTG A-Device State Machine Does Not Comply With OTG Specification*

Revision(s) Affected: All revisions

Details: In the USB OTG peripheral, transitions from OTG a_suspend state to a OTG a_host state is impossible. In the USB OTG specification revision 1.0a, this transition is triggered by bus resume or active a_bus_req. This allows the A-device acting as a host to resume operation following the suspend state.

Workaround: There is no workaround.

This exception will not be fixed in future silicon revisions.

Advisory USB_OTG_6	*USB OTG Registers are Set to 0x0 After UHOST_EN Bit is Set to 0*

Revision(s) Affected: All revisions

Details: The problem occurs when:

1. Read USB OTG registers => OK
2. Read USB HHC registers => OK
3. Set UHOST_EN bit to 0 and read back USB OTG registers => OK
4. Read USB HHC registers => Register is set to 0
5. Read USB OTG registers => Register is set to 0

When the UHOST_EN bit is set to 0, any access to the USB_HHC peripheral will never be acknowledged since this bit means: disable the clock to the USB_HHC peripheral. A side effect is that: if the system can recover from the stall (Host not answering), the OTG slave bridge will stay in the Host access state. Then, any READ access to the module will be stalled.

Workaround: If the USB OTG Revision Number is equal to 0 (means that USB bridge FSM hangs), then execute following sequence in order to be able to access to USB registers:

1. Set UHOST_EN bit to 1 (without doing read−modify−write but using direct write)
2. Read USB Host revision number (or other USB Host register). This re-initializes USB bridge FSM.
3. The ARM is now able to access any USB registers.

This exception will not be fixed in future silicon revisions.

7.6 TIMER32K Advisories

Advisory TIMER32K_1	*Timer32K Reload TRB Bit Does Not Work Properly*

Revision(s) Affected: All revisions

Details: The Timer32K Reload Bit (TRB) is used to load the TCR register with the value contained in the TVR register. Once the TCR is loaded, the TRB bit should be automatically reset to '0'. The user should be able to use the TRB bit (by polling) to know when to load a new value in the TVR.

However (see Figure 11), the TRB bit is reset to zero by the falling edge of 32-kHz clock only after a maximum of 1.5 x 32-kHz clock period (or minimum 0.5 x 32-kHz clock period) after the TRB bit is set. The TCR register is loaded after a maximum of 2 x 32-kHz clock periods (or minimum of 1 x 32-kHz clock period) on the rising edge of the 32-kHz clock. After the TRB is reset, there is a one 32-kHz clock period where a new TRB setting will not be taken into account by the module. If the user programs the TVR and updates the TRB bit within this 32-kHz clock period, the new TVR will not be loaded correctly into the TCR.

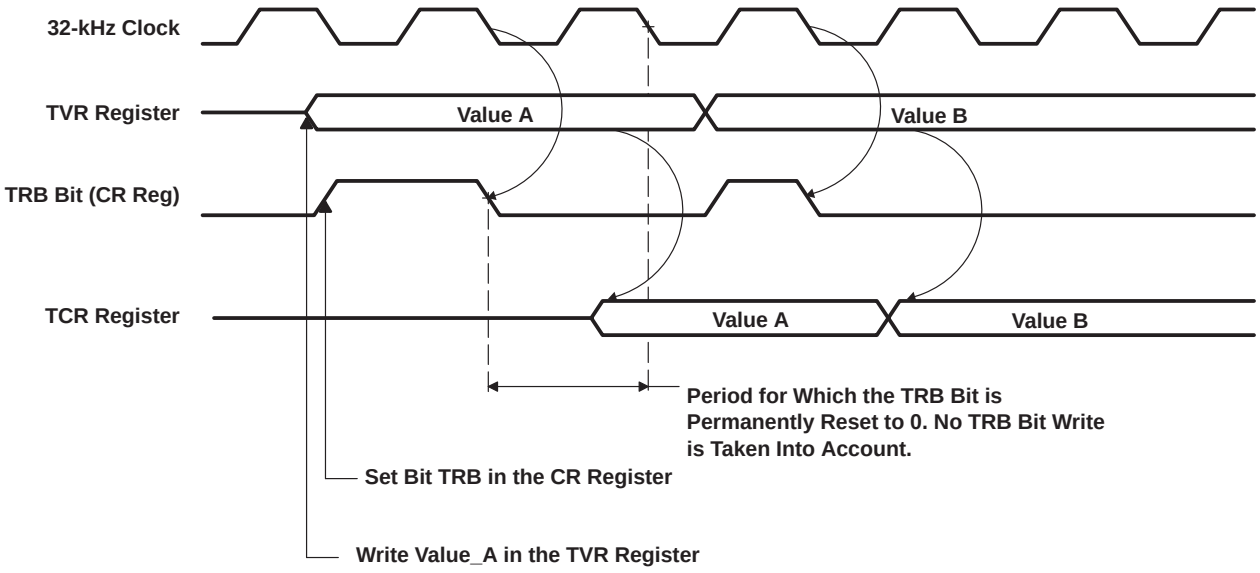

Figure 11. TRB Timing Diagram

Workaround: To load a new value into the TCR, poll the TRB bit. When the TRB is equal to '0' (signaling that the previous value has been loaded), the user has to wait for an additional one 32-kHz clock period, before loading a new value in the TVR and setting the TRB bit to '1'.

This exception will not be fixed in future silicon revisions.

7.7 Microprocessor Unit I/O (MPUIO) Advisories

Advisory MPUIO_1	*MPUIO Input Latch Register is Disabled During TIPB Read Access to it*

Revision(s) Affected: All revisions

Details: A continuous polling on the MPUIO INPUT_LATCH register does not work properly. The MPUIO_INPUT_LATCH register is implemented with a latch. This latch operates in the following manner:

- It is enabled when there is no TIPB read access to the MPUIO INPUT_LATCH register. The register is directly updated by the MPUIO inputs.
- It is disabled when there is a TIPB read access to the MPUIO INPUT_LATCH register. The register holds its last value until the latch is enabled.

As long as a MPUIO INPUT_LATCH register address is present on the TIPB bus, the corresponding MPUIO INPUT_LATCH register latch is selected and therefore disabled (MPUIO signals not selected).

Workaround: In order to avoid this during the MPUIO INPUT_LATCH register polling, a dummy TIPB access (the only condition is to do this access to another TIPB address) has to be done just after the MPUIO INPUT_LATCH register is read.

This exception will not be fixed in future silicon revisions.

Advisory MPUIO_2	*MPUIO GPIO_INT is no Longer Generated*

Revision(s) Affected: All revisions

Details: The reset of the MPUIO GPIO interrupt is done asynchronously when a GPIO_INT register read occurs. The release of this reset is done synchronously with the ARM_XOR_CK when the 32-kHz clock is high. A read of the GPIO_INT register while the 32-kHz clock is low or transitioning can result in a deadlock to the state machine that will cause all further interrupts subject to the MPUIO pin to be ignored.

Workaround: After receiving the MPUIO GPIO_INT, the MPU should observe the 32-kHz clock and then read the GPIO_INT register only just after a rising edge on the 32-kHz clock. There are different ways to detect a rising edge on the 32-kHz clock:

- Poll the timer 32k counter until there is an increment (see Figure 12, which describes the procedure)

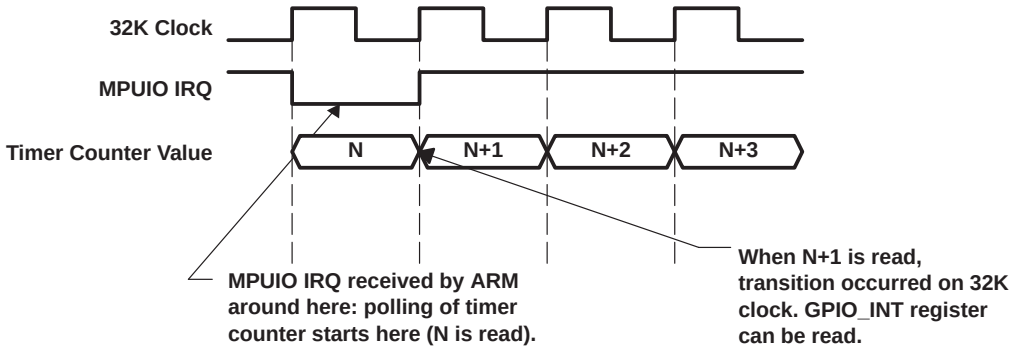

Figure 12. Polling the Timer 32k Timer

- Loop-back the CLK32K_OUT output on one available MPUIO, GPIO or KB.R input, and poll the corresponding register.

Note, that the maximum delay inserted by this workaround, before the interrupt clears, is one 32K clock period.

This exception will not be fixed in future silicon revisions.

7.8 Pulse-Width Tone (PWT) Advisories

Advisory PWT_1	*Write Followed by Immediate Read Not Supported on the PWT Peripheral*

Revision(s) Affected: All revisions

Details: A write followed by an immediate read does not work on the ARM address space defined below. If a read occurs immediately after the write to the same address, then the read will not be the data that was written. The affected address spaces are:

- PWT peripheral: Address space FFFB:6000 to End Address FFFB:67FF

Workaround: When an immediate read is required after a write to any register in the above address space, make two consecutive writes to the same address prior to the read. Using this procedure ensures that the first write completes and the read data will be correct.

This exception will not be fixed in future silicon revisions.

7.9 Camera Interface Advisories

Advisory CAM_1	*Interrupt Observability on Camera Port*

Revision(s) Affected: All revisions

Details: The active-high arm_obs_int signal, which appears on the CAM.D[4] pin when the camera interface is in observability mode, is an inverted copy of the ITR register. The CONF_ARM_INT_SEL_R bits of the FUNC_MUX_CTRL_2 register are used to select the ARM level-2 interrupt to be observed.

Workaround: There is no workaround. The interrupt observed is inactive-low and active-high.

This exception will not be fixed in future silicon revisions.

Advisory CAM_2	*Peculiar Sequence Needed to Reset Camera FIFO Using the RAZ_FIFO Bit*

Revision(s) Affected: All revisions

Details: A peculiar sequence is needed to reset the camera FIFO using RAZ_FIFO register bit in OMAP5912.

Workaround: The sequence to use is:

1. Check if LCLK and OCP clocks are enabled.
2. Assert MODE.RAZ_FIFO for a minimum of 3 CAM.LCLK clock cycles or 5 TC2_CK cycles, whichever is the longest. This will ensure proper reset of the camera input shift register and the internal FIFO pointer.
3. Read CAMDATA register. This will clear an eventual pending DMA request latched in the camera peripheral.
4. Deassert the MODE.RAZ_FIFO.
5. Wait for a minimum of 3 CAM.LCLK clock cycles or 5 TC2_CK cycles, whichever is the longest. This delay ensures the correct updating of the internal signals.

This exception will not be fixed in future silicon revisions.

7.10 Universal Asynchronous Receiver/Transmitter (UART) Advisories

Advisory UART_1	*NIRQ Can Stay Low for a Few OCP Clock Cycles After the IRQ is Serviced*

Revision(s) Affected: All revisions

Details: This advisory concerns the service of interrupts by reading from/writing to a register in the functional clock domain:

After the access to the register is finished, the NIRQ line can stay low for a few OCP clock cycles before it is deasserted. The actual delay depends on the clock ratio between the functional clock and the OCP clock. There may be potential problems with interrupts serviced twice when slower OCP clocks are used, since some IRQ controllers are programmed to be level-sensitive.

Workaround: Whenever an interrupt is processed, do the following:

- In UART mode, read the Interrupt Identification Register (IIR). If the IT_Pending is cleared, no interrupt as defined by IT_type is actually pending and the software should immediately exit the interrupt handler.
- In IrDA mode, read the Interrupt Identification Register (IIR). If the returned value is 0x00, no interrupt is actually pending and the software should immediately exit the interrupt handler.

This exception will not be fixed in future silicon revisions.

Advisory UART_2	*UART IrDA (MIR Mode) Sends an Additional Bit*

Revision(s) Affected: All revisions

Details: In MIR mode, an additional pulse is added at the end of the transmit frame after the Stop Flag. The MIR protocol makes sure that this additional bit is not interpreted as a start bit of a subsequent transmit frame (see below).

Start Flags	Frame Data	CRC-16	Stop Flag

Workaround: There is no workaround.

This exception will not be fixed in future silicon revisions.

Advisory UART_3

When in UART Sleep, Writes to the THR Wakes up the UART but No Data is Transmitted

Revision(s) Affected: All revisions

Details: When the UART has entered sleep mode, a write to the Transmit Holding Register (THR) should wake up the device and trigger a transmission of the packet written into THR. However, no data gets transmitted. If a second byte is written into THR, the FIFO starts transmitting the first data but does not transmit the second one. If a third byte is written into the THR, the FIFO sends the second data. Ultimately, the sequence is preserved and there is no loss of data.

Workaround: Software should not use UART sleep mode and should not configure UART sleep mode in the Interrupt Enable Register (IER) [bit 4 of IER]. Also, software should not use IrDA sleep mode and should not configure sleep mode in Mode Definition Register 1 (MDR1) [bit 3 of MDR1].

This exception will not be fixed in future silicon revisions.

Advisory UART_4

Rx Overrun and TX_FIFO_FULL Bit Not Operational When FIFO is Disabled

Revision(s) Affected: All revisions

Details: If the user sets the FIFO depth to 1, bit 0 (FIFO_EN) of the FIFO Control Register (FCR) is cleared to 0, and the following occurs:

- If an overrun condition occurs, it is not flagged.
- TX_FIFO_FULL [bit 0 of Supplementary Status Register (SSR)] is always 0 whether or not the FIFO (1 byte in this case) is FULL.

Workaround: Avoid overrun conditions by setting the FIFO depth to 64 (corresponding to 1 for FIFO_EN). For the case where an overrun condition is unlikely:

- In UART mode, use TX_FIFO_E mapped as bit 5 of the Line Status Register (LSR) to detect whether or not a byte is present in the FIFO.
- In IrDA mode, use THR_EMPTY mapped as bit 7 of the Line Status Register (LSR) to detect whether or not a byte is present in the FIFO.

This exception will not be fixed in future silicon revisions.

Advisory UART_5 — *Receive Overrun Not Reset on an Early Read of LSR*

Revision(s) Affected: All revisions

Details: During a receive overrun (LSR[1] = RX_OE), if a read of the Line Status Register (LSR) is done within one baud clock period after the overrun happens, the LSR is not cleared and the Overrun Interrupt remains asserted while any read of the LSR done after one baud clock period following the overrun event will clear the LSR and deassert the Overrun Interrupt.

Workaround: During the interrupt sequence, one should loop for at least one baud clock period before reading the Line Status Register (LSR).

This exception will not be fixed in future silicon revisions.

Advisory UART_6 — *Unexpected Pulse on UART.TX During Switch to SIR Mode*

Revision(s) Affected: All revisions

Details: A 1.6-µs low pulse gets transmitted on the TX pin as soon as the UART switches to SIR mode.

Workaround: When switching to SIR mode in either UART1 and UART3, the following sequence must be applied:

1. Set ACREG.PULSE_TYPE = 0
2. Set MDR1.MODE_SELECT to 001b (switch to SIR mode)
3. Wait for 1.6 µs (77 x 48 MHz clock cycles)
4. Set ACREG.PULSE_TYPE = 1

This exception will not be fixed in future silicon revisions.

Advisory UART_7 — *UART3.TX3 Stuck in IrDA Mode*

Revision(s) Affected: All revisions

Details: Once UART3 has been configured for IrDA mode, the only way to make it work in non-IrDA mode is to reset the UART3 peripheral.

Workaround: The only way to recover is to reset the peripheral (software reset by SYSC register, or global peripheral reset).

This exception will not be fixed in future silicon revisions.

Advisory UART_8 — *UART Auto Baud Status Register May Contain Invalid Speeds*

Revision(s) Affected: All revisions

Details: The UASR register is a read register used in Autobaud mode. After the reception of frame end, this register may be updated with a wrong value. This value in the UASR should be kept until the reception of ‘A/a’ (frame start). This causes the UASR to sometimes be updated with invalid information, namely the speed measured during the start bit when ‘T/t’ is received. According to the specification, speed is only valid after reception of ‘A/a’. However, the ‘A/a’ detection state machine always sets the UASR register following reception of a start bit, before detection of ‘A/a’ has been confirmed.

Workaround: There is no workaround.

This exception will not be fixed in future silicon revisions.

Advisory UART_9 — *UART Fails to Transmit XON Character in Certain Conditions*

Revision(s) Affected: All revisions

Details: In Software Flow Control mode, when the Rx FIFO reaches the HALT level, defined by TCR[3:0] (RX_FIFO_TRIG_HALT) bits, the UART transmits an XOFF character. However, if the Rx FIFO trigger level is set to the same value and the DMA is enabled, then the UARTIRDACIROCP will assert a DMA request, and the DMA will begin to empty the Rx FIFO. The Rx FIFO trigger level is determined by TLR[7:4] (RX_FIFO_TRIG_DMA), FCR[7:6] (RX_FIFO_TRIG), and SCR[7] (RX_TRIG_GRANUL1) bits. When the Rx FIFO reaches the resume level set by TCR[7:4] (RX_FIFO_TRIG_START) bits, an XON character should be transmitted. However the logic which should initiate sending the XON character gives priority over completion of sending XOFF and ignores the request to send XON.

Note that the problem could also occur when the Rx FIFO trigger level is not equal to the Rx FIFO halt level.

This can occur in any case where the Rx FIFO resume level is reached before sending of the XOFF is completed. It can also occur when the CPU, rather than the DMA, empties the FIFO due to an interrupt request.

Workaround: Avoid using the peripheral’s software flow control feature. (The software flow control feature is disabled by setting EFR[3:0] to 0000.) Instead, do one of the following:

- Use hardware flow control
- Use software to perform software flow control without hardware assist.

This exception will not be fixed in future silicon revisions.

Advisory UART_10	*UART FIFO Pointers May be Shifted Upon FIFO Enable*

Revision(s) Affected: All Revisions

Details: When the software writes FIFO_EN=1 (bit 0 of UART.FCR register), the RX/TX FIFO pointers may be spuriously incremented by one unit.

Workaround: Clear RX and TX FIFO when the FIFO has been enabled, by setting TX_FIFO_CLEAR and RX_FIFO_CLEAR bits. This operation can be done at the same time as setting FIFO_EN bit.

This exception will not be fixed in future silicon revisions.

7.11 Emulation Advisories

Advisory EMU_1	*Default TDO State During Reset is Unpredictable*

Revision(s) Affected: All revisions

Details: Some flip flops which control the enable of the TDO do not initialize properly if there is no rising edge on $\overline{\text{TRST}}$. This can be an issue if several JTAG ports are daisy-chained in a design and TDO is left floating.

Workaround: There is no workaround.

This exception will not be fixed in future silicon revisions.

Advisory EMU_2	*Emulation Breakpoint Between Sequential 16-Bit Accesses to 32-Bit Registers*

Revision(s) Affected: All revisions

Details: Getting visibility on 32-bit peripheral registers from the debugger may corrupt the on going application read sequence if the debugger access takes place between two 16-bit application reads

This emulation requirement applies to peripherals, including registers for which data width is larger than read/write access path (i.e., 32-bit register/16-bit access). Typically, read/writes have to be performed in two steps. If the read/write has to be atomic (i.e., read while counter is incrementing), a typical scheme is that the first access (i.e., MSB) triggers the capture of the LSB field into a temporary register. The second access will then read the buffer. If a breakpoint is inserted between the two accesses and the programmer wants to dump the peripheral state, there is a risk of corrupting the buffer if the same path is shared by the application flow and the emulation flow (debugger). A similar scheme applies to write accesses.

To make emulation more robust and preserve the application state, the following is recommended:

- Duplicate/Select the buffer depending on initiator (application/debugger).
- Include temporary registers (buffers) in peripheral programming model (address allocation) to support context save/restore.
- Do not set breakpoints or single step across sequential 16-bit accesses.

Workaround: Ensure no debugger access is performed to 32-bit peripheral registers between 16-bit application accesses.

This exception will not be fixed in future silicon revisions.

7.12 Specially Optimized Screen Interface (SoSSI) Advisories

Advisory SOSSI_1	*SoSSI CPriority*

Revision(s) Affected: All revisions

Details: With the Double_FIFO feature, SoSSI enables the possibility to define two data transfers with independent parameters (display data type, byte count, and reordering). If one FIFO pixel data transfer is ongoing and the Double_FIFO feature is enabled, command data transfers to another FIFO are allowed to override the pixel data transfer depending on the CPriority bit value in DIF003_INIT2 register.

If the CPriority bit is set to 1 and the command (A0 = 0) is sent during the pixel data transfer, the pixel data transfer should be aborted and the command should be sent to the display. However, due to a limitation of the SoSSI peripheral, the command will be sent but the pixel data transfer will not be aborted. The rest of the pixel data is sent after command. After the pixel data transfer is completed, the SoSSI will lock up.

If the CPriority bit is set to 0 and command (A0 = 0) is sent during the pixel data transfer, the pixel data transfer should be completed and the command should be sent to display after that. However, due to a limitation of the SoSSI, the command will be sent immediately and after which, the SoSSI will send undefined amount 0x00 data.

Workaround: Software has to ensure that the pixel data transfer in one FIFO is completed before initiating a command transfer to another FIFO.

This exception will not be fixed in future silicon revisions.

7.13 Oscillator Advisories

Advisory OSC_1	*32-kHz Oscillator Does Not Power Down in Deep-Sleep Mode*

Revision(s) Affected: All revisions

Details: The 32-kHz oscillator is always powered up regardless of the Reset_Mode value. In particular, if an external 32-kHz clock is provided to OMAP5912, the power down of the oscillator is not configured so that active circuitry in the oscillator would be cut off. This behavior is noticeable in deep-sleep mode as the oscillator current consumption is in the range of 5 µA. See Figure 13.

PWRDN	VOLTAGE ON XI	VOLTAGE ON XO	APPROXIMATE WORST-CASE CURRENT MAGNITUDE ON XI	APPROXIMATE WORST-CASE CURRENT MAGNITUDE ON XO	TOTAL CURRENT CONSUMPTION	Y
L	VSS	VSS	50 nA	5 µA	5 µA	H
L	VDD	VSS	50 nA	5 µA	5 µA	H
L	VSS	VDD	50 nA	100 µA	100 µA	L
L	VDD	VDD	50 nA	100 µA	100 µA	L
H	VSS	VSS	1 nA	1 nA	1 nA	L
H	VDD	VSS	1 nA	1 nA	2 nA	H
H	VSS	VDD	1 nA	100 µA	100 µA	L
H	VDD	VDD	1 nA	100 µA	100 µA	H

Figure 13. Oscillator Current Consumption

Workaround: Make sure that if the 32-kHz oscillator is not required, the OSC32K_PWRDN bit is set to one in the RTC Oscillator Register (RTC_OSC_REG).

This exception will not be fixed in future silicon revisions.

7.14 Serial Peripheral Interface (SPI) Advisories

Advisory SPI_1 — *Emulation Reads of the SPI Peripheral are Intrusive*

Revision(s) Affected: All revisions

Details: In the SPI peripheral, the RX_FULL flag is cleared whenever the debugger reads the receive buffer.

Workaround: There is no workaround.

This exception will not be fixed in future silicon revisions.

Advisory SPI_2 — *SPI Not Able to Receive the First Data While Waking Up From Sleep Mode*

Revision(s) Affected: All revisions

Details: While waking up from sleep mode by a data sent to the SPI (GPIO wake-up), the functional clock of the SPI is available long after the end of the data transmission, resulting in the data not being latched by the SPI.

Workaround: Resend the data after the functional clock is available at the SPI peripheral.

This exception will not be fixed in future silicon revisions.

Advisory SPI_3 — *SPI Peripheral Does Not Support 8-Bit Accesses*

Revision(s) Affected: All revisions

Details: The SPI peripheral does not support 8-bit accesses.

Workaround: If 8 bits of data must be transferred, the data must be packed in a16-bit structure.

This exception will not be fixed in future silicon revisions.

Advisory SPI_4	*Problem With SPI Functionality After Wakeup From Deep Sleep*

Revision(s) Affected: All revisions

Details: If the SPI peripheral is configured in smart idle mode, then it will prevent the system from going into deep sleep unless the following conditions are met:

- SPI must be in slave mode.
- SPI must be in receive mode.
- SPI must be in DMA mode.
- There is no pending SPI interrupt.

When all these conditions are met, the SPI will allow the system to go to sleep.

Workaround: Use force idle mode instead. Please note, that if the SPI is configured in force idle mode, the system will go into deep sleep mode. However, the peripheral can go into an unknown state when the system is awakened.

This exception will not be fixed in future silicon revisions.

Advisory SPI_5	*SPI CS Chosen Polarity Only Applies When the First SPI Transfer Begins*

Revision(s) Affected: All revisions

Details: In the SPI peripheral, the active level of SPI chip selects (SPIF.CS0, SPIF.CS1, SPIF.CS2, and SPIF.CS3) can be chosen through the SPI_SET2[9:6] register CE bits. However, the reset value of these four bits are “0000” and the inactive level of all four CS is high by default. The value set in the SPI_SET2[9:6] register bits for any one given CS is only taken into account when the first SPI transfer begins on that particular CS.

This can potentially cause bus contention on SPIF.DI, if several SPI devices with active-high CS’s are connected to the OMAP5912. As long as a transfer does not occur on each of them, their CS’s will stay high (active), regardless of the state programmed in SPI_SET2[9:6] register.

Workaround: There is no workaround.

This exception will not be fixed in future silicon revisions.

7.15 Ultra Low-Power Device (ULPD) Advisories

Advisory ULPD_1	*Emulation Access to the ULPD IT_STATUS_REG are Instrusive*

Revision(s) Affected: All revisions

Details: In the ULPD, bebugger read accesses to the IT_STATUS_REG (offset 0x14) clears status bits 0, 1, and 2 . The ULPD cannot detect the initiator of a read access, and therefore, a debugger read access impacts the register content in the same manner as in the normal application context.

Workaround: There is no workaround.

This exception will not be fixed in future silicon revisions.

Advisory ULPD_2	*ULPD Analog Cell Configuration are Reset After an MPU Peripheral Reset*

Revision(s) Affected: All revisions

Details: ULPD has the following reset scheme:

- Internal Registers are reset by ULPD power-up reset.
- ARM registers are reset by ULPD MPU Peripheral Reset.

As the ULPD is responsible for OMAP3.2 reset generation (including the MPU peripheral reset), if the MPU warm reset or any of the watchdog resets (32 kHz or secure watchdog) is active, the MPU peripheral reset is activated and the analog set-up time registers are reset. Therefore, when the FSM goes out of Deep Sleep, the set-up counters are loaded with the reset value of the analog set-up time registers instead of the optimized value previously programmed by the user, this causes wake-up latency penalty.

Note that PWR_CTRL_REG[12] (DSP isolation control) and RESET_STATUS[3:0] are not affected by this behavior.

Workaround: Software should account for the analog set-up time cell latencies.

This exception will not be fixed in future silicon revisions.

Advisory ULPD_3	*IDLECT1.WKUP_MODE Bit Must Be 0 for Proper Operation*

Revision(s) Affected: All revisions

Details: The IDLECT1.WKUP_MODE bit controls handshake between the OMAP3.2 clock and reset manager and the ULPD peripheral. When the IDLECT1.WKUP_MODE is 0, the OMAP3.2 subsystem will wait for ULPD handshake before leaving idle mode. If IDLECT1.WKUP_MODE is 1, OMAP3.2 subsystem will wake up upon reception of a wakeup event independently of the state of the ULPD. Depending on the ULPD state when the wakeup event occurs, three cases are possible:

1. No noticeable problem. The system wakes up correctly.
2. The OMAP3.2 clock is shut down while the ARM is awake (possibly in the middle of an instruction fetch). The wakeup event is caught, and corresponding interrupt routine has time to complete before the ARM clock is shut down. The system will go to sleep and resume normal operation on the next wakeup event (first wakeup event will be treated at this time). Depending on the time when OMAP3.2 clock was shut down, instruction/data corruption is possible.
3. The OMAP3.2 clock is shut down while the ARM is awake (possibly in the middle of an instruction fetch). The wakeup event is caught, but corresponding interrupt routine does not have time to complete before ARM clock is shut down. The system will resume normal operation. Depending on the time when OMAP3.2 clock was shut down, instruction/data corruption is possible.

Figure 14 summarizes briefly the conditions under which case 1, 2, and 3 may happen.

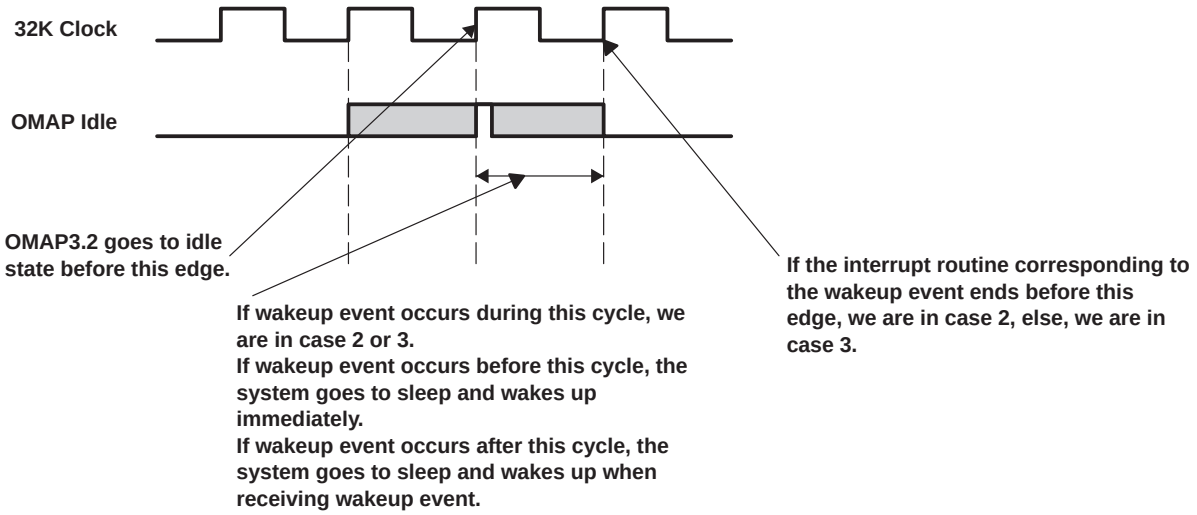

Figure 14. IDLECT1.WKUP_MODE = 1 Possible Effects

Workaround: The IDLECT1.WKUP_MODE bit should always be 0 in all OMAP5912 devices.

This exception will not be fixed in future silicon revisions.

Advisory ULPD_4	*System May Hang if Idle/Awake Sequence Period is Too Short*

Revision(s) Affected: All revisions

Details: When the system goes out of deep or big sleep, at least one 32K clock rising edge must occur before the WFI instruction is executed again. If this does not occur, the ULPD will stay in the awake state, and OMAP5912 will remain in idle mode. This will cause the system to hang. The only way to restore the system is to perform power-on reset or MPU reset.

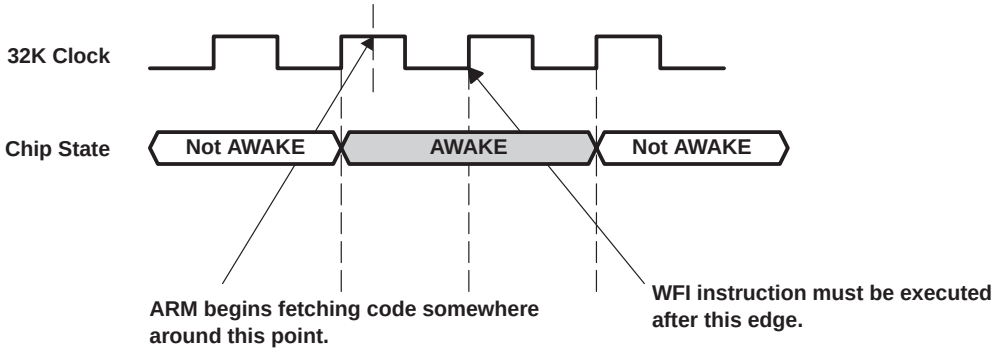

Figure 15. Minimum AWAKE Window Needed for Proper Operation

Workaround: The Minimum AWAKE window is needed for proper operation. Software needs to make sure that one 32K clock rising edge occurs between wake up of the device and next WFI instruction execution. This can be done by monitoring 32K counter value.

This exception will not be fixed in future silicon revisions.

7.16 Multichannel Serial Interface (MCSI) Advisories

Advisory MCSI_1	*MCSI2 DMA Requests are Not Mapped to the ARM*

Revision(s) Affected: All revisions

Details: MCSI2 DMA requests are only mapped on DSP side, while MCSI1 DMA requests are mapped to both the ARM and DSP sides.

Workaround: There is no workaround.

This exception will not be fixed in future silicon revisions.

Advisory MCSI_2	*Late Generation of TX DMA Request in the MCSI*

Revision(s) Affected: All revisions

Details: Multiple DSP-based DMA mode transmits and receives on the MCSI peripherals do not work properly. The MCSI peripheral is re-transmitting the last word of data that it had transmitted in the previous DMA Frame.

On the first DMA frame that the MCSI transmits, the MCSI peripheral generates a tx_dma_req just prior to asserting the frame sync signal. It then generates the proper binary stream.

On the second and subsequent DMA frames transmitted, the MCSI does not generate a tx_dma_req prior to asserting the frame sync signal and the last data from the previous DMA frame is re-transmitted by the MCSI. Then, a tx_dma_req is asserted and the character that was supposed to be first transmitted is loaded and shifted through the MCSI.

Subsequently, the first DMA request results in the transmission of the second character by the MCSI. The proper number of tx DMA requests is generated. When the last DMA request is generated, the MCSI loads the transmit buffer but does not transmit this character until the next DMA frame when the same incorrect sequence of data retrieval and transmission by the MCSI TX DMA process is repeated.

Workaround: Write the first data under DSP CPU control to the TX register before the DMA transfer. This is to ensure that the correct data is sent. Next, the DMA must be programmed to send the remaining elements 2 through N after that.

This exception will not be fixed in future silicon revisions.

Advisory MCSI_3	*MCSI Does Not Support Multi-Master Mode*

Revision(s) Affected: All revisions

Details: The high-impedance command of the MCSI data out buffer causes it to be forced to 0. It is not possible to put the MCSI data out buffer to Hi-Z state. OMAP5912 cannot support multi-master configurations.

Workaround: There is no workaround.

This exception will not be fixed in future silicon revisions.

7.17 Multichannel Buffered Serial Port (McBSP) Advisories

Advisory MCBSP_1	*McBSP in ABIS Mode Does Not Receive Data Properly*

Revision(s) Affected: All revisions

Details: In the ABIS mode, the McBSP does not receive data properly under the following conditions:

- The 16 received bits are compacted into a word.
- Only 1 enabled bit in the data stream is left.
- A new frame is about to start after the first bit is received.

When the above conditions occur, the compacted word (16 bits) previously moved into the internal buffer register is immediately overwritten by the last bit value appended with zeros. Afterwards, this value is moved to the DRR register and thus the previous value of the companded word is lost with no overflow interrupt set. Two receive interrupts are set subsequently, which is correct.

Workaround: There is no workaround. Do not use ABIS mode.

This exception will not be fixed in future silicon revisions.

Advisory MCBSP_2	*Improper Behavior in McBSP ABIS Mode Transmit*

Revision(s) Affected: All revisions

Details: In the ABIS mode-specific test for the transmit side the following problem was found:

In McBSP ABIS Transmit Mode, data sent out via DX pin is sometimes repeated depending on the mask register value (XCERA, XCERB). An internal counter inside the McBSP does not get properly incremented, resulting in bits being transmitted based on the counter value more than once.

Workaround: There is no workaround. Do not use ABIS mode.

This exception will not be fixed in future silicon revisions.

Advisory MCBSP_3	*McBSP 3-Pin Operation Versus 4-Pin Operation Configuration*

Revision(s) Affected: All revisions

Details:

Two OMAP5912 configuration bits in Module Configuration Register 0 (MOD_CONF_CTRL_0) are used for setting up McBSP3, as shown below:

- MOD_CONF_CTRL_0[28] should control the McBSP3 FSYNC output mode (McBSP3 in 3-pin or 4-pin mode)
  - 3-pin mode: FXOUT is connected to FSRIN. FSXIN is tied low.
  - 4-pin mode: FSRIN = FSXIN is connected to FSXOUT high-impedance output buffer.
- MOD_CONF_CTRL_0[19] should control the source for McBSP3 CLKS (either DSPXOR or DSPPER)

In OMAP5912, MOD_CONF_CTRL_0[28] does not work and MOD_CONF_CTRL_0[19] controls both McBSP3 modes and OCP input clock (McBSP interface clock).

- MOD_CONF_CTRL_0[19] = 0:
  - McBSP3 is in 3-pin mode
  - Interface clock: DSPXOR
- MOD_CONF_CTRL_0[19] = 1:
  - McBSP3 is in 4-pin mode
  - Interface clock: DSPPER

Note that the access time to the peripheral will change depending on 3-pin or 4-pin usage.

If McBSP3 is configured with CLKSM = 1 and CLKME = 0, the input clock of the sample rate generator (SRG) will be either the DSPXOR_CK (system clock) or DSPPER_CK (programmable) depending on MOD_CONF_CTRL_0[19]. This means the dividers in the SRG must be configured differently to achieve the same frequency in 3-pin versus 4-pin mode.

Workaround: When configuring the McBSP3 in 3-pin or 4-pin mode, make sure the correct access time to the peripheral is accounted for.

This exception will not be fixed in future silicon revisions.

7.18 Dual-Mode Timers Advisories

Advisory DM_TIMER_1	*Consecutive Captures in Dual-Mode Timer*

Revision(s) Affected: All revisions

Details: The Capture Edge Detection is not disabled if the TCAR_IT_FLAG is set. As a result, the TCAR is updated with the TCRR content whenever a new capture event occurs, even when the TCAR_IT_FLAG is still set.

Workaround: Do not allow capture to occur before the TCAR_IT_FLAG is cleared.

This exception will not be fixed in future silicon revisions.

Advisory DM_TIMER_2	*After Wakeup, the First Access to Any Dual-Mode Timer May Fail*

Revision(s) Affected: All revisions

Details: The following sequence of events may cause a fail on a dual-mode timer's first access:

- Dual-mode timer is in smart idle mode
- ARM executes WFI instruction
- Dual-mode timer enters idle mode following WFI instruction
- Peripheral clock domain shuts down
- An interrupt or wake-up event wakes the system up

In this case, the dual-mode timer will need 3 functional clock cycles + 3 ARMPER_CK clock cycles in order to be fully operational after waking up.

During this period, any access to the dual-mode timer register will result in a TIPB abort. However, any access made during this period will immediately send the dual-mode timer out of idle mode. Additionally, there is a delay of at most one functional clock period after waking up, during which a read to TCRR register will return incorrect value.

If the dual-mode timer functional clock is configured to be a 32-kHz clock, then this time can be significantly longer than the time it takes for the MPU to do an access to the dual-mode timer. In which case, if one of the first operations software performs after wakeup is an access to dual-mode timer, then the access is very likely to fail.

After Wakeup, the First Access to Any Dual-Mode Timer May Fail (Continued)

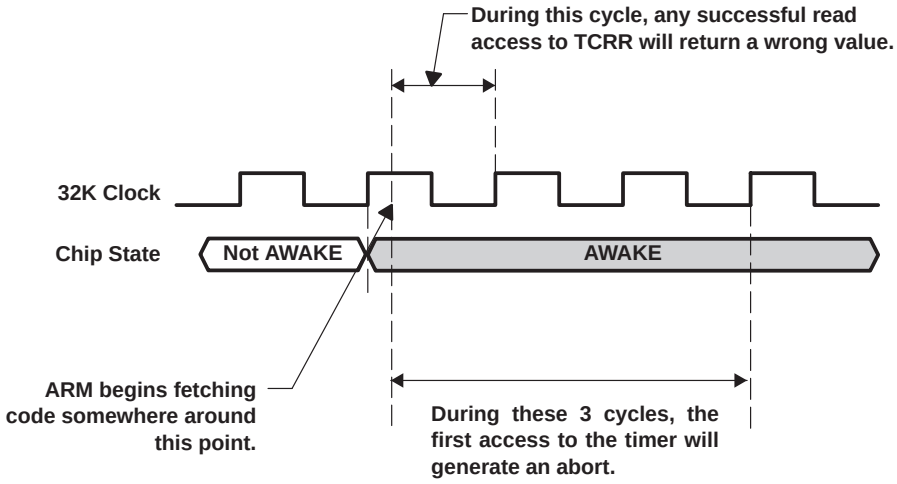

Figure 16. DM Timer Access Timings After Wake Up From Deep Sleep (Functional Clock is 32K Clock)

Workaround: There are two possible workarounds:

1. After waking up, wait for at least three functional clock periods before performing any access to dual-mode timer.
2. After waking up, wait for at least one functional clock period, then perform a dummy access to the dual-mode timer. Then, any subsequent access to the dual-mode timer will succeed.

This exception will not be fixed in future silicon revisions.

Advisory DM_TIMER_3 — *Dual-Mode Timer Peripheral May Not Detect Incoming Events*

Revision(s) Affected: All revisions

Details: Due to internal metastability issues, the dual-mode timer peripherals, when used in event-capture mode, may not detect incoming events.

Both TIMER.EVENT3 and TIMER.EVENT4 inputs of OMAP5912 are impacted.

Workaround: There is no workaround.

This exception will not be fixed in future silicon revisions.

Advisory DM_TIMER_4 — *Specific Sequence Needed to Restart the Dual-Mode Timer When One-Shot Mode is Used and Overflow Occurs*

Revision(s) Affected: All revisions

Details: When a dual-mode (DM) timer is configured in one-shot mode, and an overflow occurs, the DM timer will stop. After two timer functional clock cycles, the TCLR.ST bit will be reset. Any attempt to reload the timer counter during these two cycles will fail.

This problem occurs especially when the DM timer functional clock is configured to be a 32-kHz clock, because these two cycles are significantly long.

Workaround: When using any dual-mode timer in one-shot mode, the timer will stop once an overflow occurs.

To restart the timer once an overflow occurs, software must use the following procedure:

1. Poll TCLR.ST bit until it is cleared.
2. Reload timer counter (by writing directly to the TCRR register, or writing to the TLDR register, then to the TTGR register)
3. Restart the timer bit (by setting the TCLR.ST bit to 1).

If this sequence is not used, then the timer will be restarted with an initial value of 0x0000 0000.

This exception will not be fixed in future silicon revisions.

Advisory DM_TIMER_5 — *DM Timer Continuous Accesses in Non-Posted Mode Can Be Ignored*

Revision(s) Affected: All revisions

Details: Contiguous OCP accesses of the same type (for example: read−read or write−write) without a time gap on the same DM timer functional register can lead to un-signaled lost accesses. This occurs when the DM timer is configured in non-posted mode. The affected functional DM timer registers are:

- TCRR and TCAR for read accesses
- TCRR, TCLR, TLDR and TMAR for write accesses

Workaround: For continuous read accesses, do one of the following:

- Implement a 1-functional-clock-period delaying mechanism between each access.
- Insert an access to another functional register (for example, read[TCRR]−read[TCRR] becomes read[TCRR]−read[TCAR]−read[TCRR]).
- Reconfigure by software the DM timer (and only the DM timer) in “posted mode” (available only if the functional clock frequency is lower than 1/4 of the OCP clock frequency).

DM Timer Continuous Accesses in Non-Posted Mode Can Be Ignored (Continued)

For continuous write accesses, do one of the following:

- Implement by software a 1-functional-clock-period delaying mechanism between each access.
- Use the TWPS register, as if we were in posted mode, by polling the write pending bit until it becomes 0 before writing again.
- Insert an access to another functional register (for example, write[TCRR]−write[TCRR] becomes write[TCRR]−read[TCAR]−write[TCRR]).

This exception will not be fixed in future silicon revisions.

Advisory DM_TIMER_6	*DM Timer Events May Get Ignored*

Revision(s) Affected: All revisions

Details: A dual-mode timer can generate an overflow, match, or capture interrupts from the corresponding event (overflow/match/capture). The Timer Status Register (TISR) is used to determine which of the timer events requested an interrupt.

After the interrupt gets acknowledged by the TISR and after 3 functional clock cycles + 3 OCP clock cycles, any event of the same type other than the one initially processed, will be ignored, and the corresponding interrupt will be not generated (see Figure 17).

This problem does not occur if the event has a different type: it will be kept in the TISR.

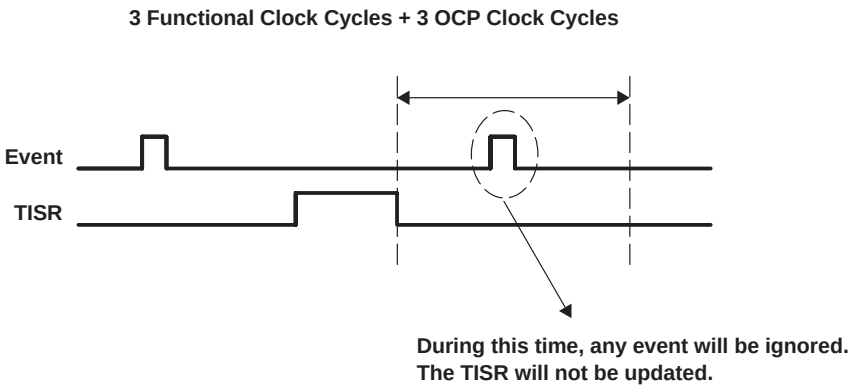

Figure 17. DM Timer Event Ignored Window

Workaround: There is no workaround.

This exception will not be fixed in future silicon revisions.

7.19 Liquid Crystal Display (LCD) Advisories

Advisory LCD_1	*LCD AC Bias is not Aligned With Pixel Clock*

Revision(s) Affected: All revisions

Details: The LCD peripheral's AC bias signal is not aligned with the pixel clock. In active TFT mode, the AC bias signal always toggles one LCD controller clock cycle (internal LCD clock) after the inactive edge. The AC bias signal will be asserted one LCD clock after the inactive edge of pixel clock, which occurs before the active edge at which first pixel data is transmitted.

The AC bias signal will be deasserted one LCD clock cycle after inactive edge, which is just after the active edge at which the last pixel data is driven.

Figure 18 shows the assertion of the AC bias signal with respect to pixel clock. Figure 19 shows the deassertion of the LCD AC bias with respect to pixel clock.

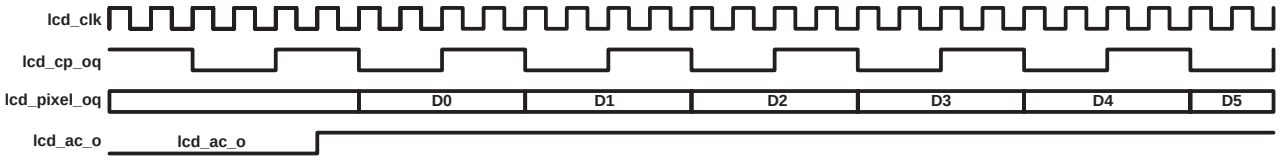

Figure 18. AC Bias Signal Assertion Waveform

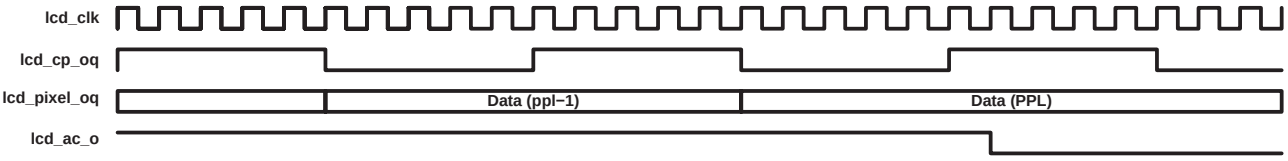

Figure 19. AC Bias Signal Deassertion Waveform

Workaround: There is no workaround.

This exception will not be fixed in future silicon revisions.

Advisory LCD_2	*Pullups and Pulldowns Not Supported on LCD Pins P[0:15]*

Revision(s) Affected: All revisions

Details: Pullups and pulldowns are not supported on LCD pins P[0:15].

Workaround: There is no workaround.

This exception will not be fixed in future silicon revisions.

Advisory LCD_3	*The Dynamic Power-Saving Feature Changes the HSYNC and VSYNC Timings*

Revision(s) Affected: All revisions

Details: The dynamic power-saving (DPS) consumption feature was added in the OMAP3.2.4 subsystem to improve power consumption. When the LCD_CTRL_REG[DPS_EN] bit is set to 1 and the LCD_CTRL_REG[LCD_EN] bit is set to 0, the LCD internal logic clock is cut. This capability affects the generation of HSYNC and VSYNC outputs. Since HSYNC and VSYNC change according to IHS or IVS bit settings only when the LCD internal clock is ON, the following consequences occur:

- If DPS_EN = 0, the HSYNC and VSYNC toggle immediately, as per the new IHS/IVS bit settings.
- If DSP_EN = 1 and LCD_EN = 0, the HSYNC and VSYNC change only after LCD_EN = 1.

Workaround: There is no workaround.

This exception will not be fixed in future silicon revisions.

7.20 HDQ/1-Wire Advisories

Advisory HDQ_1WIRE_1	*HDQ/1-Wire Peripheral Does Not Meet HDQ Standard Requirements*

Revision(s) Affected: All revisions

Details: OMAP5912 HDQ/1-Wire® peripheral timings are checked with timings from battery monitor devices like BQ26500, BQ26501, BQ2019, BQ26220 and BQ26221:

- BQ2019, BQ26500 and BQ26220 do not work with OMAP5912 (at 13 MHz and 19.2 MHz)
- BQ26501 and BQ26221 work with a workaround to meet Break timing (Tb).

Table 2. HDQ Timings Comparison

	BQ26500 (TI)		BQ26501 (TI)		BQ2019 (TI)		BQ26220 (TI)		BQ26221 (TI)		OMAP (TI)			UNIT
	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	12 MHz	13 MHz	19.2 MHz	
t_B	190		190		190		190		190		193	**178§**	**120§**	µs
t_{BR}	40		40		40		40		40		63	58	**39**	µs
t_{CYCH}	190		190		190		190		190		253	234	**157**	µs
t_{HW1}‡	*17*	50	*0.5*	50	*0.005†*	50	*0.005†*	50	*0.5*	50	**1.3¶**	**1.2¶**	**0.8¶**	µs
t_{HW0}‡	*100*	145	*86*	145	*100*	145	*100*	145	*92*	145	101	**93#**	**63‖**	µs
t_{CYCD}	190	*260*	190	*260*	190	*250*	190	*250*	190	*250*	232	214	**144**	µs
t_{DW1}	32	50	32	50	32	50	32	50	32	50	<68	<63	**<42**	µs
t_{DW0}	80	145	80	145	80	145	80	145	80	145	>68	>63	>42	µs

NOTE: **Bolded** numbers denote values out of one or more specifications.
Italicized numbers denote mismatch values between specifications.

† Any pulse greater than 5 ns will be detected as a Bit 1.
‡ t_{HWx} represents what the master transmits to the host.
§ Use 1-Wire® t_{RSTL} (477 µs minimum) for HDQ break to solve this issue.
¶ Timing (t_{HW1}) out of BQ26500 specifications.
# Timing (t_{HW0}) out of BQ26500, BQ2019, and BQ26220 specifications.
‖ Timing (t_{HW0}) out of all these specifications.

1-Wire is a registered trademark of Dallas Semiconductor Corporation.

HDQ/1-Wire Peripheral Does Not Meet HDQ Standard Requirements (Continued)

Workaround: The workaround for HDQ break time is to transmit a 1-Wire® initialization pulse, which is low for 447 µs (t_{RSTL}).

This method requires entering 1-Wire® mode to send the initialization pulse, exiting 1-Wire® mode, and transmitting the HDQ command and data bytes.

This software workaround is applicable only to BQ26501and BQ26221 devices.

This exception will not be fixed in future silicon revisions.

Advisory HDQ_1WIRE_2 — *HDQ/1-Wire Engine Does Not Meet 1-Wire Module Standard Requirements*

Revision(s) Affected: All revisions

Details: OMAP5912 HDQ/1-Wire® peripheral timings (running at 13-MHz system clock) are checked with timings from battery monitors like BQ2023, DS2761, and iButton® devices:

- None will work with OMAP5912 at 19.2-MHz system clock
- BQ2023 will work with OMAP5912 (at 12-MHz and 13-MHz system clock)
- DS2761 and iButton® devices will work with OMAP 1-Wire® with a workaround for the initialization phase (at 13-MHz system clock)

Table 3. HDQ/1-Wire Device Timings Summary

	BQ2023 (TI)		DS2761 (Maxim)		iButton®		OMAP (TI)			UNIT
	MIN	MAX	MIN	MAX	MIN	MAX	12 MHz	13 MHz	19.2 MHz	
t_{SLOT}	60	120	60	120	60	120	102	94	63	µs
t_{LOW1}	1	15	1	15	1	15	1.3	1.2	**0.8**	µs
t_{LOW0}	60	120	60	120	60	120	101	93	63	µs
t_{REC}	1		1		1		134	124	83	µs
t_{LOWR}	1	15	1				<13.3‡	<12.3‡	<8.3‡	µs
$t_{RDV}+t_{REL}$		45				60	<102	<94	<63	µs
t_{RSTL}	480†		480	960	480		484	**447§**	**302§**	µs
t_{RSTH}	*300*		*480*		*480*		484	**447¶**	**302¶**	µs
t_{PDH}	15	60	15	60	15	60	<68	<6.3	**<42#**	µs
t_{PDL}	60	240	60	240	60	240	>68 − t_{PDH}	>63 − t_{PDH}	>42 − t_{PDH}	µs

NOTE: **Bolded** numbers denote values out of one or more specifications.
Italicized numbers denote mismatch values between specifications.

† BQ2023 recognizes a reset pulse if it is greater than 418 µs.
‡ Pulse must be ≤ 15 µs to read Bit 1 or Bit 0. This will work with BQ2023.
§ Timing (t_{RSTL}) out of DS2761 and iButton® specifications.
¶ Timing (t_{RSTH}) out of DS2761 and iButton® specifications.
# Timing (t_{PDH}) out of BQ2023, DS2761, and iButton® specifications.

Workaround: The workaround for 1-Wire® reset time is using the GPIO which is multiplexed with the 1-Wire® signal (GPIO11) to increase the initialization pulse width (447 µs more than 480 µs). This software workaround is applicable for DS2761 and iButton® devices.

This exception will not be fixed in future silicon revisions.

iButton is a registered trademark of Dallas Semiconductor Corporation.

8 OMAP5912 Device/System-Level Advisories

8.1 System Advisories

Advisory SYS_1	*External Pulldown is Needed on Ball Y18 of ZZG Package*

Revision(s) Affected: All revisions

Details: For proper operation of the device, an external pulldown is required for the $\overline{\text{TRST}}$ signal on ball Y18 (ZZG package). This ensures the JTAG controller is properly kept under reset during chip operation. Omitting this pulldown can lead to undefined behavior (for example, DSP stop). There is an internal pulldown on ball Y18 but it is not enabled by default after powerup. This is why an external pulldown is mandatory.

Workaround: There is no workaround.

This exception will not be fixed in future silicon revisions.

Advisory SYS_2	*UART1 and UART3 Must Be Configured to Force Idle Mode to Allow OMAP5912 to Go Into Deep Sleep*

Revision(s) Affected: All revisions

Details: When configured in smart idle mode, the UART needs both the 48-MHz APLL clock and the OCP clock (ARMPER_CLOCK) to be active to acknowledge the idle request from OMAP3.2. However, in order to go to deep sleep, the APLL clock request must be deasserted before executing the WFI instruction.

Workaround: Configure the UART to force idle mode. To prevent any problems due to the state of the peripheral after resuming from deep sleep, the following must be done:

- Select force idle so that peripheral returns the idle acknowledge.
- Ensure data transmission and reception is stopped before entering idle/deep sleep.
- Ensure data transmission and reception is not started until out of idle or if data transmission is needed, reset the FIFO.

This exception will not be fixed in future silicon revisions.

Advisory SYS_3	*OMAP5912 Will Not Enter Deep Sleep Without OCPI_CK*

Revision(s) Affected: All revisions

Details: For OMAP3.2 to go to sleep, an internal signal called EN_CLK_API must be set to 0. This signal depends on the following conditions:

- If ARM_IDLECT2.EN_APICK is 0, then EN_CLK_API is automatically 0.
- If ARM_IDLECT2.EN_APICK is 1, another signal OCPIDLE_ACK must be set to 1, indicating that OCP devices are idle. It is necessary to make sure that Idle does not occur when an external OCPI master tries to access a DSP peripheral through the MPUI.

  For this signal to be active, the following must occur:

  - The state machine controlling OCP access to be in the correct state (Idle waiting for something to happen).
  - The EMIFF/EMIFS/OCPT1/OCPT2 need to be in Idle mode.
  - The EMIFS_CONFIG PDE bit is set to 1.
  - REPWR_EN and IDLOCPI_ARM both must be set to 1 in order to go to sleep.

Workaround: The correct procedure to enter deep sleep is:

- If the ARM_IDLECT2.EN_APICK bit is set 0, the ARM_EWUPCT.REPWR_EN and ARM_IDLECT3.IDLOCPI_ARM bits are Don’t Cares and deep sleep can be entered.
- If the ARM_IDLECT2.EN_APICK bit is set to 1, either the ARM_EWUPCT.REPWR_EN bit must be set to 0, or the ARM_EWUPCT.REPWR_EN bit set to 1 and ARM_IDLECT3.IDLOCPI_ARM bit set to 1, to enter deep sleep.
- If the EN_API_CK bit is set to 1, then the DLAPI_ARM bit needs to be set to 1 to enter deep sleep.

This exception will not be fixed in future silicon revisions.

Advisory SYS_4	*OMAP5912 Gated1/Gated2 Functionality Does Not Control I/O Direction*

Revision(s) Affected: All revisions

Details: Some OMAP5912 outputs can be forced to 0 in case of battery failure or external event (low level on GPIO9 or MPUIO3). The outputs which exhibit this feature are classified as either gated1 or gated2 outputs. Gating functionality is enabled by setting the FUNC_MUX_CTRL_0.BVLZ_MASK_OUT register bit. However, gating functionality does not control the output enable of the IO pad. As a result, if an OMAP5912 I/O is configured as an input, the gating event will not be applied to this I/O and will not turn automatically into an output driving a logical 0.

Workaround: When configuring OMAP5912 I/O, the user should make sure that the I/Os, which should be forced to 0 in case of a battery failure or event on GPIO9/MPUIO3, have been configured before as outputs.

This exception will not be fixed in future silicon revisions.

Advisory SYS_5	*Dynamic Voltage Scaling (DVS) Implementation Requires Special Software Sequence to Properly Enter/Exit DVS*

Revision(s) Affected: All revisions

Details: If OMAP5912 goes into big or deep sleep without the Dynamic Voltage Scaling (DVS) feature enabled (big or deep sleep), then next time the chip tries to go into DVS, the low_pwr signal will toggle too early (when writing POWER_CTRL_REG register bits, and before actually leaving the AWAKE state).

Workaround: The software workaround is that the MIN_MAX_REG bit must be set to 0 whenever you want to enter deep or big sleep and not DVS mode .

The simple rule is: if DVS_ENABLE is 0 or DEEP_SLEEP_TRANSITION_EN is 1 (or both), then MIN_MAX_REG must also be 0 to avoid unexpected toggling of the low_pwr signal.

This exception will not be fixed in future silicon revisions.

Advisory SYS_6 *Emulation is Broken When the DSP is in Isolation Mode*

Revision(s) Affected: All revisions

Details: DSP isolation does not route TDI to TDO to allow emulation to continue operating once the DSP is put in isolation mode. The DSP can be placed in isolation mode to save leakage current related to DSP gates by setting bit 12 of POWER_CTRL_REG in ULPD.

Workaround: Do not set the DSP in Isolation Mode while in Emulation Mode.

This exception will not be fixed in future silicon revisions.

Advisory SYS_7 *Reset Mode Limitation*

Revision(s) Affected: All revisions

Details: Reset mode 1 is not supported on OMAP5912.

Workaround: There is no workaround.

This exception will not be fixed in future silicon revisions.

Advisory SYS_8 *Booting Mode Limitation*

Revision(s) Affected: All revisions

Details: Only booting from external flash on CS3 is supported on OMAP5912. All other boot options are not available. As a result of this, MPU_BOOT is a don’t care.

GPIO13 is used to select between full and fast boot. Set GPIO13 high to program external flash on CS3 using the USB port. Set GPIO13 low to boot from external flash on CS3.

Workaround: There is no workaround.

This exception will not be fixed in future silicon revisions.

Advisory SYS_9	*High Security and Emulation Mode Limitation*

Revision(s) Affected: All revisions

Details: Only General Purpose (GP) device/mode is supported on OMAP. High Security and Emulation devices/modes are not supported.

Workaround: There is no workaround.

This exception will not be fixed in future silicon revisions.

Advisory SYS_10	*OMAP5912 USB Power Consumption*

Revision(s) Affected: All revisions

Details: On OMAP5912, leaving the USB.DP and USB.DM inputs floating creates an increased power consumption of 1 mA on the V_{DD} core supply (I_{DDQ}). This can be a problem in Deep Sleep Mode, where power conservation is important.

Workaround: Enabling of the internal pulldown reduces current by 900 μA.

Advisory SYS_11	*Peripherals That Are Not Supported on OMAP5912*

Revision(s) Affected: All revisions

Details: The following peripherals are not supported on OMAP5912: Compact Flash, Compact Camera Port (CCP), Stacked DDR, SSI, SST, SSR, STI, GDD, eFUSE, effuses, Windows tracer.

Workaround: There is no workaround.

9 Documentation Support

For device-specific data sheets and related documentation, visit the TI web site at: **http://www.ti.com**

For further information regarding the OMAP5912, please refer to:

- *OMAP5912 Applications Processor Data Manual* (literature number SPRS231)

For additional information, see the latest versions of:

- *TMS320C55x™ DSP Functional Overview* (literature number SPRU312)
- *TMS320C55x DSP CPU Reference Guide* (literature number SPRU371)
- *TMS320C55x DSP Mnemonic Instruction Set Reference Guide* (literature number SPRU374)
- *TMS320C55x DSP Algebraic Instruction Set Reference Guide* (literature number SPRU375)
- *TMS320C55x DSP CPU Programmer's Reference Supplement* (literature number SPRU652)

REVISION HISTORY

This revision history highlights the technical changes made to SPRZ209J to generate SPRZ209K.

PAGE NO(S).	ADDITIONS/CHANGES/DELETIONS
62	Added Advisory UART_10, *UART FIFO Pointers May be Shifted Upon FIFO Enable*

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI 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 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. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
RFID	www.ti-rfid.com	Telephony	www.ti.com/telephony
Low Power Wireless	www.ti.com/lpw	Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2007, Texas Instruments Incorporated