# AM335x Sitara™ Processors
# Silicon Revisions 2.1, 2.0, 1.0

# Silicon Errata

Literature Number: SPRZ360G
October 2011–Revised January 2016

# Contents

Silicon Errata
SPRZ360G – October 2011 – Revised January 2016

# AM335x Sitara™ Processors Silicon Revisions 2.1, 2.0, 1.0

This document describes the known exceptions to the functional specifications for the AM335x Sitara™ ARM® Cortex®-A8 processors. [See the *AM335x Sitara Processors* data manual (SPRS717).]

## 1 Introduction

This document describes the known exceptions to the functional specifications for the AM335x Sitara™ ARM® Cortex®-A8 processors. [See the *AM335x Sitara Processors* data manual (SPRS717).]

For additional information, see the latest version of the *AM335x Sitara Processors Technical Reference Manual* (SPRUH73).

### *1.1 AM335x Device and Development Support Tool Nomenclature*

To designate the stages in the product development cycle, TI assigns prefixes to the part numbers of all microprocessors (MPUs) and support tools. Each device has one of three prefixes: X, P, or null (no prefix) (for example, XAM3358ZCE). 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 (TMDX) through fully qualified production devices and tools (TMDS).

Device development evolutionary flow:

**X —** Experimental device that is not necessarily representative of the final device's electrical specifications and may not use production assembly flow.

**P —** Prototype device that is not necessarily the final silicon die and may not necessarily meet final electrical specifications.

**null —** Production version of the silicon die that is fully qualified.

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."

Production 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.

## 1.2 Revision Identification

The device revision can be determined by the symbols marked on the top of the package. Figure 1 provides an example of the AM335x device markings.

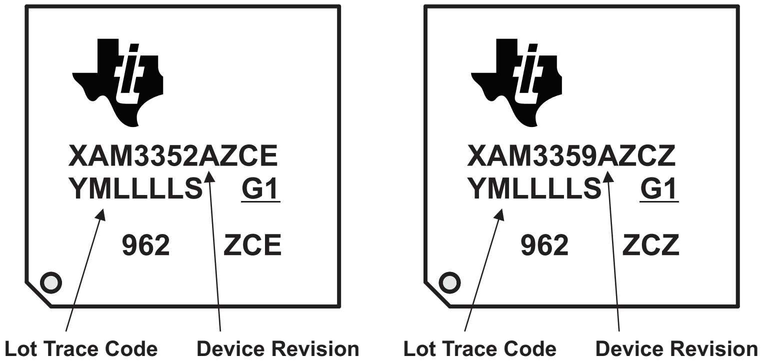

**Figure 1. Example of Device Revision Codes for the AM335x Microprocessor**

**NOTES:**

(A) Non-qualified devices are marked with the letters "X" or "P" at the beginning of the device name, while qualified devices have a "blank" at the beginning of the device name.

(B) The AM3352 and AM3359 devices shown in this device marking example are two of several valid part numbers for the AM335x family of devices.

(C) The device revision code is the device revision (A, B, and so on).

(D) YM denotes year and month.

(E) LLLL denotes Lot Trace Code.

(F) 962 is a generic family marking ID.

(G) G1 denotes green, lead-free.

(H) ZCE or ZCZ is the package designator.

(I) S denotes Assembly Site Code.

(J) On some "X" devices, the device speed may not be shown.

Silicon revision is identified by a code marked on the package. The code is of the format AM3352x or AM3358x, where "x" denotes the silicon revision. Table 1 lists the information associated with each silicon revision for each device type. For more details on device nomenclature, see the device-specific data manual.

**Table 1. Production Device Revision Codes**

| DEVICE REVISION CODE | SILICON REVISION | COMMENTS |
|---|---|---|
| (blank) | 1.0 | Silicon revision is new |
| A | 2.0 | Silicon revision 2.0 |
| B | 2.1 | Silicon revision 2.1 |

Each silicon revision uses a specific revision of TI's ARM Cortex-A8 processor. The ARM Cortex-A8 processor variant and revision can be read from the Main ID Register. The DEVREV field (bits 31-28) of the Device_ID register located at address 0x44E10600 provides a 4-bit binary value that represents the device revision. The ROM code revision can be read from address 2BFFCh. The ROM code version consists of two decimal numbers: major and minor. The major number is always 22, minor number counts ROM code version. The ROM code version is coded as hexadecimal readable values; for example, ROM version 22.02 is coded as 0000 2202h. Table 2 shows the ARM Cortex-A8 Variant and Revision, Device Revision, and ROM Code Revision values for each silicon revision of the device.

**Table 2. Silicon Revision Variables**

| SILICON REVISION | ARM CORTEX-A8 VARIANT AND REVISION | DEVICE REVISION | ROM REVISION |
|---|---|---|---|
| 1.0 | r3p2 | 0000 | 22.02 |
| 2.0 | r3p2 | 0001 | 22.03 |
| 2.1 | r3p2 | 0010 | 22.03 |

## 2 All Errata Listed With Silicon Revision Number

Advisories are numbered in the order in which they were added to this document. Some advisory numbers may be moved to the next revision and others may have been removed because the design exception was fixed or documented in the device-specific data manual or peripheral user's guide. When items are moved or deleted, the remaining numbers remain the same and are not re-sequenced.

**Table 3. All Usage Notes**

| NUMBER | TITLE | SILICON REVISION AFFECTED | | |
|---|---|---|---|---|
| | | 1.0 | 2.0 | 2.1 |
| Section 3.1.1 | LCD: Color Assignments of LCD_DATA Terminals | X | X | X |
| Section 3.1.2 | DDR3: JEDEC Compliance for Maximum Self-Refresh Command Limit | X | X | X |
| Section 3.1.3 | Boot: USB Boot ROM Code Uses Default DATAPOLARITY | X | X | X |
| Section 3.1.4 | Boot: Multiplexed Signals GPMC_WAIT0, GMII2_CRS, and RMII2_CRS_DV Cause NAND Boot Issue | X | | |
| Section 3.1.5 | Pin Multiplexing: Valid IO Sets and Restrictions | X | X | X |
| Section 3.1.6 | Boot: Multiplexed Signals GPMC_WAIT0 and GMII2_CRS Cause NAND Boot Issue | | X | X |
| Section 3.1.7 | OSC1: RTC_XTALIN Terminal Has an Internal Pull-up Resistor When OSC1 is Disabled | X | X | X |

**Table 4. All Design Exceptions to Functional Specifications**

| NUMBER | TITLE | SILICON REVISION AFFECTED | | |
|---|---|---|---|---|
| | | 1.0 | 2.0 | 2.1 |
| Advisory 1.0.1 | DDR2, DDR3, mDDR PHY: Control and Status Registers Configured for Write Only | X | X | X |
| Advisory 1.0.2 | Debug Subsystem: EMU[4:2] Signals Are Not Available by Default After Reset | X | X | X |
| Advisory 1.0.3 | Debug Subsystem: Internal Inputs Tied-off to the Wrong Value | X | X | X |
| Advisory 1.0.4 | PRU-ICSS: Clock Domain Crossing (CDC) Issue | X | | |
| Advisory 1.0.5 | RTC: 32.768-kHZ Clock is Gating Off | X | | |
| Advisory 1.0.6 | EXTINTn: Input Function of the EXTINTn Terminal is Inverted | X | | |
| Advisory 1.0.7 | Boot: Ethernet Boot ROM Code PHY Link Speed Detection | X | | |
| Advisory 1.0.8 | Boot: Ethernet Boot ROM Code Sends an Incorrect Vendor Class Identifier in BOOTP Packet | X | | |
| Advisory 1.0.9 | Ethernet Media Access Controller and Switch Subsystem: C0_TX_PEND and C0_RX_PEND Interrupts Not Connected to ARM Cortex-A8 | X | | |
| Advisory 1.0.10 | GMII_SEL Register: RGMII1_IDMODE and RGMII2_IDMODE Bits Reset to Non-supported Mode of Operation | X | | |
| Advisory 1.0.11 | USB: Attached Non-compliant USB Device that Responds to Spurious Invalid Short Packet May Lock Up Bus | X | | |
| Advisory 1.0.12 | UART: Extra Assertion of FIFO Transmit DMA Request, UARTi_DMA_TX | X | X | X |
| Advisory 1.0.13 | USB: Data May be Lost When USB Subsystem is Operating in DMA Mode and More Than One Endpoint is Transferring Data | X | | |
| Advisory 1.0.14 | GMII_SEL and CPSW Related Pad Control Registers: Context of These Registers is Lost During Transitions of PD_PER | X | | |
| Advisory 1.0.15 | ARM Cortex-A8: OPP50 Operation on MPU Domain Not Supported | X | | |

### Table 4. All Design Exceptions to Functional Specifications (continued)

| NUMBER | TITLE | SILICON REVISION AFFECTED | | |
|---|---|---|---|---|
| | | 1.0 | 2.0 | 2.1 |
| Advisory 1.0.16 | RMII: 50-MHz RMII Reference Clock Output Does Not Satisfy Clock Input Requirements of RMII Ethernet PHYs | X | X | X |
| Advisory 1.0.17 | VDDS_DDR: High-Power Consumption During DeepSleep0 | X | | |
| Advisory 1.0.18 | ROM: Ethernet Boot Code Does Not Change Default Direction of RMII1 Reference Clock When Booting from Ethernet Using RMII | X | | |
| Advisory 1.0.19 | DDR3: Fully-Automated Hardware READ and WRITE Leveling Not Supported | X | X | X |
| Advisory 1.0.20 | Boot: USB Boot ROM Code Overlapping Data in TXFIFO and RXFIFO | X | | |
| Advisory 1.0.21 | SmartReflex: Limited Support Due to Issue Described in Advisory 1.0.15 | X | | |
| Advisory 1.0.22 | EMIF: Dynamic Voltage Frequency Scaling (DVFS) is Not Supported | X | X | X |
| Advisory 1.0.23 | Ethernet Media Access Controller and Switch Subsystem: Reset Isolation Feature is Not Supported | X | X | X |
| Advisory 1.0.24 | Boot: System Boot is Not Reliable if Reset is Asserted While Operating in OPP50 | X | X | X |
| Advisory 1.0.25 | Boot: System Boot Temporarily Stalls if an Attempt to Boot from Ethernet is Not Successful | X | X | X |
| Advisory 1.0.26 | I2C: SDA and SCL Open-Drain Output Buffer Issue | X | X | |
| Advisory 1.0.27 | LCDC: LIDD DMA Mode Issue | X | X | X |
| Advisory 1.0.28 | LCDC: Raster Mode, Hardware Auto Underflow Restart Does Not Work | X | X | X |
| Advisory 1.0.29 | Latch-up Performance: Latch-up Performance Limits for Silicon Revsions 1.0 and 2.0 | X | X | |
| Advisory 1.0.30 | OSC0 and OSC1: Noise Immunity Improved When Crystal Circuit is Connected Directly to PCB Digital Ground | X | X | X |
| Advisory 1.0.31 | TSC_ADC: False Pen-up Interrupts | X | X | X |
| Advisory 1.0.32 | TSC_ADC: Terminals May be Temporarily Connected Together Through Internal Paths During Power-up Sequence | X | X | X |
| Advisory 1.0.33 | USB Host: USB Low-Speed Receive-to-Transmit Inter-Packet Delay | X | X | X |
| Advisory 1.0.34 | USB2PHY: Register Accesses After a USB Subsystem Soft Reset May Lock Up the Entire System | X | X | X |
| Advisory 1.0.35 | UART: Transactions to MDR1 Register May Cause Undesired Effect on UART Operation | X | X | X |
| Advisory 1.0.36 | EMU0 and EMU1: Terminals Must Be Pulled High Before ICEPick Samples | X | X | X |
| Advisory 1.0.37 | Watchdog Timers: Delay Needed to Read Some WDTimer Registers After Wakeup | X | X | X |
| Advisory 1.0.38 | McASP: McASP to EDMA Synchronization Level Event Can Be Lost | X | X | X |
| Advisory 1.0.39 | GPTimer: Delay Needed to Read Some GPTimer Registers After Wakeup | X | X | X |
| Advisory 1.0.41 | UART: UART0-5 Do Not Acknowledge Idle Request After DMA Has Been Enabled | X | X | X |
| Advisory 1.0.42 | DebugSS: DebugSS Does Not Acknowledge Idle Request | X | X | X |

# 3 Usage Notes and Known Design Exceptions to Functional Specifications

## *3.1 Usage Notes*

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

### 3.1.1 LCD: Color Assignments of LCD_DATA Terminals

The blue and red color assignments to the LCD data pins are reversed when operating in RGB888 (24bpp) mode compared to RGB565 (16bpp) mode. In order to correctly display RGB888 data from the SGX, or any source formatted as RGB in memory, it is necessary to connect the LCD panel as shown in Figure 2. Using the LCD Controller with this connection scheme limits the use of RGB565 mode. Any data generated for the RGB565 mode requires the red and blue color data values be swapped in order to display the correct color.

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | PIXEL_n | | | | | | | | | | | | | | | |
| B[0] | G[0] | R[0] | B[1] | G[1] | R[1] | B[2] | R[2] | B[7:3] | | | | | G[7:2] | | | | | | R[7:3] | | | | |
| | | | | | | | | 16-bit panel | | | | | | | | | | | | | | | |
| 24-bit panel | | | | | | | | | | | | | | | | | | | | | | | |

**Figure 2. RGB888 Mode LCD Controller Output Pin Mapping (LCD_DATA[23:0])**

When operating the LCD Controller in RGB565 mode the LCD panel should be connected as shown in Figure 3. Using the LCD Controller with this connection scheme limits the use of RGB888 mode. Any data generated for the RGB888 mode requires the red and blue color data values be swapped in order to display the correct color.

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | PIXEL_n | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | R[7:3] | | | | | G[7:2] | | | | | | B[7:3] | | | | |
| | | | | | | | | 16-bit panel | | | | | | | | | | | | | | | |
| 24-bit panel | | | | | | | | | | | | | | | | | | | | | | | |

**Figure 3. RGB565 Mode LCD Controller Output Pin Mapping (LCD_DATA[23:0])**

### 3.1.2 DDR3: JEDEC Compliance for Maximum Self-Refresh Command Limit

When using DDR3 EMIF Self-Refresh, it is possible to violate the maximum refresh command requirement specified in the JEDEC standard DDR3 SDRAM Specification (JESD79-3E, July 2010). This requirement states that the DDR3 EMIF controller should issue no more than 16 refresh commands within any 15.6-μs interval.

To avoid this requirement violation, when using the DDR3 EMIF and Self-Refresh (setting LP_MODE = 0x2 field in the PMCR), the SR_TIM value in the PMCR must to be programmed to a value greater than or equal to 0x9.

### 3.1.3 Boot: USB Boot ROM Code Uses Default DATAPOLARITY

The AM335x USB PHYs supports a DATAPOLARITY feature that allows the data plus (DP) and data minus (DM) data signals to be swapped. This feature was added to simplify PCB layout.

In some cases, the DP and DM data signals may need to cross over each other to connect to the respective USB connector pins. Crossing these signals on the PCB may cause signal integrity issues if not implemented properly since they must be routed as high-speed differential transmission lines. The DATAPOLARITY feature in the USB PHYs can be used resolve this issue.

The DATAPOLARITY feature is controlled by DATAPOLARITY_INV (bit 23) of the respective USB_CTRL register.

The USB boot ROM code uses the default value for DATAPOLARITY_INV when booting from USB. Therefore, the PCB must be designed to use the default DATAPOLARITY if the system must support USB boot.

### 3.1.4 Boot: Multiplexed Signals GPMC_WAIT0, GMII2_CRS, and RMII2_CRS_DV Cause NAND Boot Issue

The AM335x device multiplexes the GPMC_WAIT0, GMII2_CRS, and RMII2_CRS_DV signals on the same terminal. This causes a problem when the system must support NAND boot while an MII or RMII Ethernet PHY is connected to port 2 of the Ethernet media access controller and switch (CPSW). The GPMC_WAIT0 signal is required for NAND boot. The GMII2_CRS or RMII2_CRS_DV signal is required by the respective MII or RMII Ethernet PHY and the only pin multiplexing option for these signals is GPMC_WAIT0.

In this case, there are two sources that need to be connected to the GPMC_WAIT0 terminal. The NAND READY or BUSY output must source the GPMC_WAIT0 terminal during NAND boot and the MII CRS or RMII CRS_DV output must source the GPMC_WAIT0 terminal when the application software is using port 2 of the CPSW. Therefore, a GPIO-controlled external 2-to-1 multiplexer must be implemented in the system to select between the two sources. The GPIO selected to control the 2-to-1 multiplexer needs to have an internal or external resistor that selects the NAND READY or BUSY output as soon as power is applied and remains in that state until the application software initializes the CPSW.

The TI TS5A3157 SPDT analog switch is an example device that can be used as a 2-to-1 multiplexer. This device inserts minimum propagation delay to the signal path since it is an analog switch. The propagation delay inserted by the 2-to-1 multiplexer must be analyzed to confirm it does not cause timing violations for the respective interface.

The NAND, Ethernet PHY, AM335x VDDSHV1, AM335x VDDSHV3 (when using the ZCZ package), and 2-to-1 multiplexer IO power supply domains may need to operate at the same voltage since they share common signals.

### 3.1.5 Pin Multiplexing: Valid IO Sets and Restrictions

The AM335x device contains many peripheral interfaces. In order to reduce package size and lower overall system cost while maintaining maximum functionality, many of the AM335x terminals can multiplex up to eight signal functions. Although there are many combinations of pin multiplexing that are possible, only a certain number of sets, called IO Sets, are valid due to timing limitations. These valid IO Sets were carefully chosen to provide many possible application scenarios for the user.

Texas Instruments has developed a Windows®-based application called Pin Mux Utility that helps a system designer select the appropriate pin-multiplexing configuration for their AM335x-based product design. The Pin Mux Utility provides a way to select valid IO Sets of specific peripheral interfaces to ensure the pin multiplexing configuration selected for a design only uses valid IO Sets supported by the AM335x device.

A few IO Sets have additional restrictions not defined in the Pin Mux Utility. These additional restrictions are described below:

- MMC0, MMC1, MMC2 Interfaces
  - Only Standard (STD) and High Speed (HS) modes are supported. SDR12, SDR25, SDR50 modes as defined in SD3.0 specification are not supported.
- GEMAC_CPSW Interface
  - Operation of GEMAC_CPSW is not supported for OPP50.

### 3.1.6 Boot: Multiplexed Signals GPMC_WAIT0 and GMII2_CRS Cause NAND Boot Issue

The AM335x device multiplexes the GPMC_WAIT0 and GMII2_CRS signals on the same terminal. This causes a problem when the system must support NAND boot while an MII Ethernet PHY is connected to port 2 of the Ethernet media access controller and switch (CPSW). The GPMC_WAIT0 signal is required for NAND boot. The GMII2_CRS signal is required by the MII Ethernet PHY and the only pin multiplexing option for these signals is GPMC_WAIT0.

In this case, there are two sources that need to be connected to the GPMC_WAIT0 terminal. The NAND READY or BUSY output must source the GPMC_WAIT0 terminal during NAND boot and the MII CRS output must source the GPMC_WAIT0 terminal when the application software is using port 2 of the CPSW. Therefore, a GPIO-controlled external 2-to-1 multiplexer must be implemented in the system to select between the two sources. The GPIO selected to control the 2-to-1 multiplexer needs to have an internal or external resistor that selects the NAND READY or BUSY output as soon as power is applied and remains in that state until the application software initializes the CPSW.

The TI TS5A3157 SPDT analog switch is an example device that can be used as a 2-to-1 multiplexer. This device inserts minimum propagation delay to the signal path since it is an analog switch. The propagation delay inserted by the 2-to-1 multiplexer must be analyzed to confirm it does not cause timing violations for the respective interface.

The NAND, Ethernet PHY, AM335x VDDSHV1, AM335x VDDSHV3 (when using the ZCZ package), and 2-to-1 multiplexer IO power supply domains may need to operate at the same voltage since they share common signals.

### 3.1.7 OSC1: RTC_XTALIN Terminal Has an Internal Pull-up Resistor When OSC1 is Disabled

The RTC_XTALIN terminal has an internal pull-up resistor that is turned on when OSC1 is disabled. OSC1 is disabled by default after power is applied.

This internal pull-up resistor was not properly documented in data sheet revisions D and earlier. These early data sheet revisions recommended an external pull-down resistor to be connected to the RTC_XTALIN terminal if OSC1 was not used. The recommendation should have been to leave this terminal open-circuit when not using OSC1.

Connecting an external pull-down to the RTC_XTALIN terminal may cause unexpected leakage current.

The current recommendation is to remove any external pull-down resistor from the RTC_XTALIN terminal and leave this terminal open-circuit when not using OSC1.

## 3.2 Known Design Exceptions to Functional Specifications

The following advisories are known design exceptions to functional specifications. Advisories are numbered in the order in which they were added to this document. Some advisory numbers may be moved to the next revision and others may have been removed because the design exception was fixed or documented in the device-specific data manual or peripheral user's guide. When items are moved or deleted, the remaining numbers remain the same and are not re-sequenced.

### Advisory 1.0.1 *DDR2, DDR3, mDDR PHY: Control and Status Registers Configured for Write Only*

**Revisions Affected** 2.1, 2.0, 1.0

**Details** The DDR2, DDR3, mDDR PHY control and status registers mapped in address range 0x44e12000-0x44E123FF are configured for write-only operations, so the contents of these register cannot be read.

These registers must be configured by performing write-only operations.

**Workaround** There is no workaround for this issue.

### Advisory 1.0.2 *Debug Subsystem: EMU[4:2] Signals Are Not Available by Default After Reset*

**Revisions Affected** 2.1, 2.0, 1.0

**Details** All Debug subsystem components should remain unchanged when warm reset is asserted. For example, warm reset should not affect export of debug trace messages on the EMU[4:0] signals.

The AM335x EMU[4:2] signals can not be used to export trace messages from the Debug subsystem since AM335x does support warm reset and the EMU[4:2] signals are not assigned to pins after reset is asserted.

**Workaround** Do not assert warm reset while performing trace functions.

### Advisory 1.0.3 *Debug Subsystem: Internal Inputs Tied-off to the Wrong Value*

**Revisions Affected** 2.1, 2.0, 1.0

**Details** Internal inputs dbg_dpio_attr_dp_app_owner[4:0] and dbg_dpio_attr_dp_debug_only[4:0] to the Debug subsystem are used to report which EMU[4:0] signals can currently be used to export trace messages. These inputs were tied-off to the wrong value. The tie-off values used always indicates EMU[4:2] signals are not available and EMU[1:0] signals are available to export trace messages.

This should not cause a problem for EMU[4:2] since these signals can not be used to export trace messages for the reason explained in advisory 1.3. However, the AM335x pins used for EMU[1:0] signals may be configured as GPIO. The Debug subsystem would not know these signals are not available for exporting trace messages when these pins are configured as GPIO.

**Workaround** Do not configure the AM335x EMU[1:0] pins to operate as GPIO if you need to export trace messages.

## Advisory 1.0.4 *PRU-ICSS: Clock Domain Crossing (CDC) Issue*

**Revisions Affected** 1.0

**Details** The PRU-ICSS has a clock domain crossing issue when the MII receive multiplexer is configured to connect PR1_MII1 signals to PRU0 and PR1_MII0 signals to PRU1.

The multiplexer logic always uses the PR1_MII_MR0_CLK input to synchronize the PRU0 MII receive signals and the PR1_MII_MR1_CLK input to synchronize the PRU1 MII receive signals. This cause the wrong clock to be used when the MII receive multiplexer is configured to connect PR1_MII1 signals to PRU0 and PR1_MII0 signals to PRU1.

As a result of this issue, support for EtherCAT media redundancy is not available.

**Workaround** There is no workaround for this issue.

## Advisory 1.0.5 *RTC: 32.768-kHZ Clock is Gating Off*

**Revisions Affected** 1.0

**Details** The RTC has a clock gating issue that stops the internal 32.768-kHz clock when the VDD_CORE voltage domain drops below the recommended operating range or the PWRONRSTn input terminal is held low. This issue has the following side effects:

- The RTC counters stop incrementing when the 32.768-kHz clock is gated. This causes the RTC to lose time while the clock is gated.
- A wakeup event applied to the EXT_WAKEUP input terminal is masked if the EXT_WAKEUP_DB_EN bit in the RTC PMIC register (0x98) is set to 1 which enables the de-bounce function for the EXT_WAKEUP input. This occurs because the 32.768-kHz clock is being used to clock the de-bounce circuit.

**Workaround** Do not turn off the VDD_CORE power source or source a logic low to the PWRONRSTn input while expecting RTC to keep an accurate time.

Do not enable the de-bounce circuit on the EXT_WAKEUP input if an external wakeup event needs to be detected while the 32.768-kHz clock is gated.

## Advisory 1.0.6 *EXTINTn: Input Function of the EXTINTn Terminal is Inverted*

**Revisions Affected** 1.0

**Details** The EXTINTn input is active high.

**Workaround** Use an active high interrupt source or use an external inverter to change the polarity of any active low interrupt source.

## Advisory 1.0.7 ***Boot: Ethernet Boot ROM Code PHY Link Speed Detection***

**Revisions Affected** 1.0

**Details** The device ROM code relies on the external PHY's Control Register (Register 0), specifically bits 0.6 [Speed Selection (MSB)] and 0.13 [Speed Selection (LSB)], to determine the operating speed of the link.

If the external PHY does not update its link speed selection bits to reflect the current operating speed, the ROM code incorrectly assumes the PHY is operating at the speed indicated by the link speed selection bits and configure the device Ethernet MAC to the wrong speed. For example, if the default value of the PHY link speed selection bits indicates 100 Mbps, when the PHY is actually operating at 1 Gbps, the ROM incorrectly configures the device Ethernet MAC for 100 Mbps mode.

The IEEE 802.3 specification states: *When the Auto-Negotiation Enable (bit 0.12) is enabled, bits 0.6 and 0.13 can be read or written to, but the state of bits 0.6 and 0.13 have no effect on the link configuration, and it is not necessary for bits 0.6 and 0.13 to reflect the operating speed of the link when it is read*. While some PHYs update the link speed in these bits to reflect the current operating speed, other PHYs do not update these bits because it is not mandatory according to the specification.

**Workaround** When using Ethernet boot, an external PHY that updates the Register 0 link speed selection bits (0.6 and 0.13) to reflect the current operating speed is required.

## Advisory 1.0.8 ***Boot: Ethernet Boot ROM Code Sends an Incorrect Vendor Class Identifier in BOOTP Packet***

**Revisions Affected** 1.0

**Details** When using Ethernet boot, the device ROM code should send a BOOTP request with a unique identifier to distinguish itself from other devices on the same network. Instead, the ROM code sends the same identifier, "DM814x ROM v1.0", for all devices (DM814x, DM816x, and AM335x); hence, the download host attempting to bootstrap the devices can no longer determine which device is requesting the code to be downloaded.

Applications using the DM814x, DM816x, and AM335x devices cannot coexist in the same network if they are booted from Ethernet.

**Workaround** There is no workaround for this issue.

For some applications, it might be necessary to uniquely identify and service BOOTP packets from a client. The recommended approach to uniquely identify clients is to use the MAC address. Every device comes with a unique MAC address. A list of MAC addresses and the device type can be made available to the host in advance, so that the host can take device-specific action when it receives a BOOTP packet from a MAC address on the host's list.

## Advisory 1.0.9 *Ethernet Media Access Controller and Switch Subsystem: C0_TX_PEND and C0_RX_PEND Interrupts Not Connected to ARM Cortex-A8*

**Revisions Affected** 1.0

**Details**

The Ethernet Media Access Controller/Switch (CPSW) subsystem C0_TX_PEND and C0_RX_PEND interrupt outputs provide a single transmit interrupt that combines transmit channel interrupts TXPEND[7:0] and a single receive interrupt that combines receive channel interrupts RXPEND[7:0]. The TXPEND[0] and RXPEND[0] interrupt outputs are connected to the ARM Cortex-A8 interrupt controller (INTC) rather than the C0_TX_PEND and C0_RX_PEND interrupt outputs. This only allows channel 0 to interrupt the ARM Cortex-A8.

The C0_TX_PEND and C0_RX_PEND interrupt outputs are the only interrupt outputs that support interrupt pacing. If transmit channel interrupts 1-7, receive channel interrupts 1-7, or interrupt pacing is required, the following workaround must be implemented.

**Workarounds**

### DMTIMER Workaround

The C0_TX_PEND and C0_RX_PEND interrupts can be re-routed to the ARM Cortex-A8 through two of the AM335x timers. TIMER5 and TIMER6 can be used when configured to operate in capture mode. The time captured while operating in this mode is not relevant, since the capture event notification to the ARM Cortex-A8 represents the original causal interrupts coming from the EMAC and Switch subsystem.

The re-routed interrupts path are:

- ARM Cortex-A8: TINT5 (interrupt 93) <-- TIMER5 Capture Event <-- [Event Capture Mux: event 8] <-- EMAC and Switch: C0_RX_PEND
- ARM Cortex-A8: TINT6 (interrupt 94) <-- TIMER6 Capture Event <-- [Event Capture Mux: event 9] <-- EMAC and Switch: C0_TX_PEND

### Configuration

The following configurations are required to use timer capture module interrupts:

- TIMER5 and TIMER6 are enabled with capture mode during initialization.
  - Set bit 2 of the TIMER5 IRQENABLE set register located at 0x4804_602C to 1b.
  - Set bit 2 of the TIMER6 IRQENABLE set register located at 0x4804_802C to 1b.
- Write the value 0x908 to the TIMER_EVT_CAPT register located at 0x44E1_0FD0 to select EMAC and Switch event 8 (C0_RX_PEND) for TIMER 5 and EMAC and Switch event 9 (C0_TX_PEND) for TIMER6.
- Configure TIMER5 and TIMER6 to single-capture mode by resetting the CAPT_MODE bit of each TCLR register.
  - Reset bit 13 of the TIMER5 TCLR register located at 0x4804_6038 to 0b.
  - Reset bit 13 of the TIMER6 TCLR register located at 0x4804_8038 to 0b.
- Select rising-edge transition by setting the TCM bit of the TCLR register.
  - Set bit 8 of the TIMER5 TCLR register located at 0x4804_6038 to 1b.
  - Set bit 8 of the TIMER6 TCLR register located at 0x4804_8038 to 1b.
- Use ARM Cortex-A8 interrupt 93 for C0_RX_PEND and interrupt 94 for C0_TX_PEND instead of interrupts 41 and 42.

### Interrupt Servicing

The following is the recommended procedure for servicing interrupts. This method clears and re-enables the interrupts properly to ensure no interrupts are missed by the DMTimer edge detection logic. This procedure applies for both receive and transmit interrupts.

- Clear the timer capture interrupt by writing 1 to the TCAR_IT_FLAG bit of the respective IRQSTATUS register.

  - Clear the TIMER5 interrupt by setting bit 2 of the TIMER5 IRQSTATUS register located at 0x4804_6028 to 1b.
  - Clear the TIMER6 interrupt by setting bit 2 of the TIMER6 IRQSTATUS register located at 0x4804_8028 to 1b.
- Disable all CPSW interrupts by clearing the C0_xx_EN field in the respective C0_RX_EN/C0_TX_EN register.
- Acknowledge the interrupt by writing the appropriate RX or TX vector to the CPDMA_EOI_VECTOR register.
- Process all received or transmitted packets.
- Enable the desired CPSW interrupts in the C0_xx_EN field in the respective C0_RX_EN/C0_TX_EN register.

## Advisory 1.0.10 *GMII_SEL Register: RGMII1_IDMODE and RGMII2_IDMODE Bits Reset to Non-supported Mode of Operation*

**Revisions Affected** 1.0

**Details** The reset state of RGMII1_IDMODE (bit 4) and RGMII2_IDMODE (bit 5) in the GMII_SEL register enables internal delay mode on the transmit clock of the respective RGMII port. The AM335x device does not support internal delay mode, so RGMII1_IDMODE and RGMII2_IDMODE must be set to 1b.

**Workaround** Many RGMII Ethernet PHYs provide an internal delay mode that may be enabled to insert delays required to meet the setup and hold timing requirements of the AM335x device and attached RGMII PHY. A timing analysis is recommended before the printed circuit board (PCB) design has been completed, in case it is necessary to insert additional delays on the RGMII signals connecting the AM335x device and attached RGMII PHY.

It is necessary to insert PCB delays if the RGMII PHY being connected to the AM335x device does not support internal delay mode. A complete timing analysis is required to determine the optimum delay of each PCB signal trace.

## Advisory 1.0.11 *USB: Attached Non-compliant USB Device that Responds to Spurious Invalid Short Packet May Lock Up Bus*

**Revisions Affected** 1.0

**Details** The integrated USB PHY (analog transceiver) has a timing error that turns on its receiver too early and occasionally detects the end of its own transmit data as receive data. This causes the USB controller to transmit an invalid short packet. Normally this invalid short packet would be ignored by the attached USB device and the data transmission would continue as expected.

At least one mass storage class USB device has been found to be non-compliant to the USB specification, by responding to this packet. This non-compliant response (NACK) to the invalid short packet violates USB protocol and causes the bus to hang.

Poor signal integrity of the differential signal pair used to connect the attached USB device may contribute to this issue. Impedance discontinuities and mismatched terminations on the differential signal pair may cause reflections to propagate longer than expected, which allows the transceiver to detect these reflections of its own transmit data as receive data.

**Workaround** There is no workaround for this issue.

To prevent an unexpected response to any invalid short packets, attach only USB devices that are compliant with the USB specification.

To minimize reflections, it is also recommended that the USB DP and DM signals are routed as a 90-Ω differential pair transmission line with minimum impedance discontinuities and proper terminations.

## Advisory 1.0.12 *UART: Extra Assertion of FIFO Transmit DMA Request, UARTi_DMA_TX*

**Revisions Affected** 2.1, 2.0, 1.0

**Details** A UART transmit request with a DMA THRESHOLD default configuration of 64 bytes results in an extra DMA request assertion when the FIFO TX_FULL is switched from high to low.

**Workaround** To avoid an extra DMA request assertion, use:

TX_THRESHOLD + TRIGGER_LEVEL ≤ 63 (TX FIFO Size - 1).

## Advisory 1.0.13 *USB: Data May be Lost When USB Subsystem is Operating in DMA Mode and More Than One Endpoint is Transferring Data*

**Revisions Affected** 1.0

**Details**

Data loss may occur due to a USB data toggle synchronization error that occurs when an internal data toggle counter is erroneously reset from the DATA1 state to the DATA0 state while the USB subsystem is actively receiving data from more than one endpoint. The erroneous reset of the data toggle counter occurs because the associated logic in the USB subsystem DMA contains an error that does not support the correct data toggle update with data transfers from multiple endpoints.

If the DATA1 state is erroneously reset to the DATA0 state immediately following a USB transaction in which the PID is DATA0, the transmitter and receiver become de-synchronized. This data toggle synchronization error causes the receiver, per the USB specification, to silently discard the non-synchronized packet, which causes the packet and any data contain therein to be lost.

**NOTE:** For more information related to the definition of DATA0 and DATA1 PIDs and functional requirements of data toggle synchronization, see sections 8.4.4 and 8.6 of the *Universal Serial Bus Specification Revision 2.0*.

**Workarounds**

- Operating in USB host mode - The workaround involves detecting and correcting the data toggle mismatch by software after receiving each USB packet. In order to implement this workaround, the CPPI4.1 DMA must be configured to operate in transparent mode; generic RNDIS mode cannot be used. Software must save the previous data toggle value then compare the current data toggle value and the saved value to detect a data toggle mismatch. If a synchronization error is detected, it must be corrected by simultaneously writing 1b to the data toggle write enable and data toggle bits in the respective RxCSR registers.
- Operating in USB device mode - There is no workaround for this mode of operation.

## Advisory 1.0.14 *GMII_SEL and CPSW Related Pad Control Registers: Context of These Registers is Lost During Transitions of PD_PER*

**Revisions Affected** 1.0

**Details** The GMII_SEL and CPSW related pad control registers listed below are reset to their default state during transitions of peripheral power domain (PD_PER) while the ISO_CONTROL bit in the RESET_ISO register is set to its default value (0b). This occurs when the AM335x device enters or exits DeepSleep0.

| REGISTER NAME | REGISTER ADDRESS |
|---|---|
| GMII_SEL | 0x44E1_0650 |
| CONF_GPMC_A0 | 0x44E1_0840 |
| CONF_GPMC_A1 | 0x44E1_0844 |
| CONF_GPMC_A2 | 0x44E1_0848 |
| CONF_GPMC_A3 | 0x44E1_084C |
| CONF_GPMC_A4 | 0x44E1_0850 |
| CONF_GPMC_A5 | 0x44E1_0854 |
| CONF_GPMC_A6 | 0x44E1_0858 |
| CONF_GPMC_A7 | 0x44E1_085C |
| CONF_GPMC_A8 | 0x44E1_0860 |
| CONF_GPMC_A9 | 0x44E1_0864 |
| CONF_GPMC_A10 | 0x44E1_0868 |
| CONF_GPMC_A11 | 0x44E1_086C |
| CONF_GPMC_WAIT0 | 0x44E1_0870 |
| CONF_GPMC_WPN | 0x44E1_0874 |
| CONF_GPMC_BEN1 | 0x44E1_0878 |
| CONF_MII1_COL | 0x44E1_0908 |
| CONF_MII1_CRS | 0x44E1_090C |
| CONF_MII1_RX_ER | 0x44E1_0910 |
| CONF_MII1_TX_EN | 0x44E1_0914 |
| CONF_MII1_RX_DV | 0x44E1_0918 |
| CONF_MII1_TXD3 | 0x44E1_091C |
| CONF_MII1_TXD2 | 0x44E1_0920 |
| CONF_MII1_TXD1 | 0x44E1_0924 |
| CONF_MII1_TXD0 | 0x44E1_0928 |
| CONF_MII1_TX_CLK | 0x44E1_092C |
| CONF_MII1_RX_CLK | 0x44E1_0930 |
| CONF_MII1_RXD3 | 0x44E1_0934 |
| CONF_MII1_RXD2 | 0x44E1_0938 |
| CONF_MII1_RXD1 | 0x44E1_093C |
| CONF_MII1_RXD0 | 0x44E1_0940 |
| CONF_RMII1_REF_CLK | 0x44E1_0944 |
| CONF_MDIO | 0x44E1_0948 |
| CONF_MDC | 0x44E1_094C |

**Workaround** Re-initialize these registers after exiting DeepSleep0.

## Advisory 1.0.15 — *ARM Cortex-A8: OPP50 Operation on MPU Domain Not Supported*

**Revisions Affected** 1.0

**Details** Reliability tests have shown that a logic cell used in the ARM Cortex-A8 exhibits weakness during low-voltage operation as defined by OPP50. This eliminates support for operating the ARM Cortex-A8 at the lower voltage defined by OPP50. Therefore, the minimum voltage limit for the ARM Cortex-A8 power terminals (VDD_MPU on the ZCZ package and VDD_CORE on the ZCE package) is the minimum voltage limit defined by OPP100.

**Workaround** To minimize power consumption, the ARM Cortex-A8 may be operated at the lower frequencies defined by OPP50, but the respective power terminal (VDD_MPU for ZCZ package and VDD_CORE for ZCE package) must be operated as defined by OPP100. Table 5 below provides register settings for operating the MPU PLL at 275 MHz for each supported input clock frequency.

**Table 5. MPU PLL**

| CLK_M_OSC (MHz) | N CM_CLKSEL_DPLL_MPU[6:0] | REFCLK (MHz) | M CM_CLKSEL_DPLL_MPU[18:8] | M2 CM_DIV_M2_DPLL_MPU[4:0] | CLKOUT (MHz) |
|---|---|---|---|---|---|
| 19.2 | 95 | 0.2 | 1375 | 1 | 275 |
| 24 | 23 | 1 | 275 | 1 | 275 |
| 25 | 24 | 1 | 275 | 1 | 275 |
| 26 | 25 | 1 | 275 | 1 | 275 |

## Advisory 1.0.16 — *RMII: 50-MHz RMII Reference Clock Output Does Not Satisfy Clock Input Requirements of RMII Ethernet PHYs*

**Revisions Affected** 2.1, 2.0, 1.0

**Details** The 50-MHz RMII reference clock output is sourced from the ADPLLS CORE PLL which is not a low-jitter clock source. Therefore, the clock jitter of this output is greater than the input requirements for most RMII Ethernet PHYs.

**Workaround** Configure the respective RMII reference clock to input mode and use an external low-jitter LVCMOS clock source or RMII Ethernet PHY with a clock output to source the RMII reference clock.

RMII1_REFCLK can be configured to input mode by setting bit 6 of the GMII_SEL register to 1b. RMII2_REFCLK can be configured to input mode by setting bit 7 of the GMII_SEL register to 1b.

## Advisory 1.0.17 *VDDS_DDR: High-Power Consumption During DeepSleep0*

**Revisions Affected** 1.0

**Details** The REG_PHY_ENABLE_DYNAMIC PWRDN bit in the DDR_PHY_CTRL_1 register provides control for powering down the SSTL and HSTL input buffers to achieve lower power consumption from the VDDS_DDR power source. This register is reset to its default value during DeepSleep0 which prevents powering down the DDR SSTL and HSTL input buffers which causes higher power consumption during DeepSleep0.

**Workaround** Input buffers associated with the bi-directional DDR terminals can be configured to operate as SSTL and HSTL or LVCMOS inputs. Low-power consumption can be achieved during DeepSleep0 by configuring the input buffers to operate in LVCMOS mode and enabling internal pull-downs on each of the bi-directional DDR terminals before entering DeepSleep0. Power consumption from the VDDS_DDR power source of this solution is similar to powering down the DDR SSTL and HSTL input buffers.

The DDR PHY can be configured to operate in LVCMOS mode by setting the MDDR_SEL bit in the DDR_IO_CTRL register to 1b.

The internal pull-downs are configured by writing 0011_1111_1111_0000_0000_00pp_pppp_pppp, where p = previous binary value, to the DDR_DATA0_IOCTRL and DDR_DATA1_IOCTRL registers.

## Advisory 1.0.18 *ROM: Ethernet Boot Code Does Not Change Default Direction of RMII1 Reference Clock When Booting from Ethernet Using RMII*

**Revisions Affected** 1.0

**Details** The default direction of the RMII1 reference clock is output mode. This mode of operation is not supported, as described in Advisory 1.0.16.

When the SYSBOOT[7:6] boot mode inputs are set to 01b, which selects RMII mode, the RMII1_REF_CLK terminal is configured to operate in the non-supported output mode as soon as the ROM code changes the MUXMODE bits of the CONF_RMII1_REF_CLK register from the default state of GPIO0_29 (111b) to RMII1_REFCLK (000b).

This causes contention on the RMII1 reference clock signal if the ROM code attempts to boot from RMII since the only RMII mode of operation supported requires the RMII1_REF_CLK terminal to be driven by an external 50-MHz RMII reference clock source. Therefore, the ROM code must never be configured such that it attempts to boot from RMII.

**Workaround** Use MII if Ethernet boot is required.

## Advisory 1.0.19 *DDR3: Fully-Automated Hardware READ and WRITE Leveling Not Supported*

**Revisions Affected** 2.1, 2.0, 1.0

**Details** DDR3-based systems use a "fly-by" layout routing scheme where the address, clock, and control signals are connected to multiple memory devices using a daisy-chain topology, as opposed to DDR2-based systems which connect multiple devices using a balanced T-topology. The "fly-by" routing scheme introduces skew in the arrival time of the DDR signals to each memory device. DDR3 memories and DDR3 memory controller provide hardware assisted training that optimizes timing for each data byte lane. This is commonly referred to as READ and WRITE leveling. The objective of the READ and WRITE leveling is to obtain correct values of the DLL ratios to compensate for the skew and is done automatically during the initialization process.

The DDR3 controller does not reliably arrive at the optimal DLL ratios during the automatic training process. Therefore, the automated hardware READ and WRITE leveling is not supported.

**Workaround** Use the software-leveling procedure outlined below to obtain optimal DLL ratios that compensate READ and WRITE timing:

1. Disable automated hardware READ and WRITE leveling by setting the REG_RDWRLVL_EN bit in the RDWR_LVL_RMP_CTRL register to 0b.
2. Configure all EMIF4D registers, including AC timing values, as required for the attached DDR3 memory device.
3. Determine the initial seed DLL ratio values to be used in the software-leveling algorithm. These values are based on board trace lengths of DDR_CK(n) and DDR_DQS(n).
4. Run the software-leveling algorithm with the initial seed DLL ratio values. The algorithm iterates several times to find the optimum values for the given configuration.
5. The software-leveling algorithm determines the optimum values for the following registers in the DDR controller. Use the optimum values obtained from the program when initializing the DDR controller.
   - DATA_PHY_RD_DQS_SLAVE_RATIO
   - DATA_PHY_FIFO_WE_SLAVE_RATIO
   - DATA_PHY_WR_DQS_SLAVE_RATIO
   - DATA_PHY_WR_DATA_SLAVE_RATIO

This procedure is only required once for a given combination of DDR3 memory devices, DDR3 operating frequency, and printed circuit board layout. If there are any changes to memory devices, operating frequency, or printed circuit board layout, the procedure outlined above must be re-run.

## Advisory 1.0.20 *Boot: USB Boot ROM Code Overlapping Data in TXFIFO and RXFIFO*

**Revisions Affected** 1.0

**Details** The USB boot ROM code is overlapping data in the TXFIFO and RXFIFO which leads to data corruption. This data corruption causes a data abort and prevents USB boot from working.

**Workaround** There is no workaround for this issue.

## Advisory 1.0.21 *SmartReflex: Limited Support Due to Issue Described in Advisory 1.0.15*

**Revisions Affected** 1.0

**Details** SmartReflex is not supported on the VDD_MPU power domain of the ZCZ package when configured for OPP100 because SmartReflex may reduce the VDD_MPU voltage to OPP50 levels. Advisory 1.0.15 describes why operating the ARM Cortex-A8 at the lower voltage defined by OPP50 is not supported. For more details related to this issue, see Advisory 1.0.15.

SmartReflex is not supported on the ZCE package.

SmartReflex is supported on the ZCZ package for the following power domain and OPP combinations:

- VDD_CORE configured for OPP100.
- VDD_MPU configured for any valid OPP greater than OPP100.

**Workaround** There is no workaround for this issue.

## Advisory 1.0.22 *EMIF: Dynamic Voltage Frequency Scaling (DVFS) is Not Supported*

**Revisions Affected** 2.1, 2.0, 1.0

**Details** The L3 Interconnect and EMIF internal hookup does not allow changing the EMIF operating frequency while OCP transactions are pending. This prevents dynamic switching from OPP100 to OPP50 or from OPP50 to OPP100. The application software must initialize a static OPP50 or OPP100 configuration before enabling EMIF.

**Workaround** There is no workaround for this issue.

## Advisory 1.0.23 *Ethernet Media Access Controller and Switch Subsystem: Reset Isolation Feature is Not Supported*

**Revisions Affected** 2.1, 2.0, 1.0

**Details** The Ethernet Media Access Controller and Switch (CPSW) subsystem may lock up if the Reset Isolation feature is enabled when a warm reset is applied while the host port is transmitting data. Since most warm reset sources can be asynchronous events, this lock-up condition can only be prevented by not enabling this feature.

The Reset Isolation feature is not enabled by default.

**Workaround** There is no workaround for this issue.

### Advisory 1.0.24 *Boot: System Boot is Not Reliable if Reset is Asserted While Operating in OPP50*

**Revisions Affected** 2.1, 2.0, 1.0

**Details** The system attempts to boot using the ARM (A8), L3, L4, and respective DDR clock frequencies defined by OPP100 when a reset is asserted. The system may fail to boot if the system is operating with reduced VDD_MPU and VDD_CORE power supply voltages as defined by OPP50 when reset is asserted. This issue occurs because the device is being operated at OPP100 clock frequencies with OPP50 supply voltages. This unsupported operating condition potentially over-clocks the logic which was only timing closed to operate at OPP50 clock frequencies with OPP50 supply voltages.

There are three basic reset sources, the PWRONRSTn terminal, the WARMRSTn terminal, and the internal watchdog timer, that need to be considered when designing a product that supports OPP50.

It is important to return VDD_MPU and VDD_CORE power supplies to OPP100 defined voltages before any of these resets sources are asserted.

**Workarounds** Only source the PWRONRSTn terminal from a power management circuit that always returns VDD_MPU and VDD_CORE power supplies to OPP100 defined voltages before asserting PWRONRSTn.

There are two possible workarounds that can be applied to the other two reset sources. The first workaround provides the lowest power consumption option but eliminates the watchdog timer and WARMRSTn terminal functions. The second workaround retains the watchdog timer and WARMRSTn terminal functions, but causes the device to consume higher power.

- Disable the watchdog timer and do not assert the WARMRSTn terminal while the VDD_MPU and VDD_CORE power supply voltages are less than those defined by OPP100.
- Retain the VDD_MPU and VDD_CORE power supply voltages defined by OPP100 while operating the ARM (A8), L3, L4, and the respective DDR clocks at the reduced frequencies defined by OPP50.

## Advisory 1.0.25 *Boot: System Boot Temporarily Stalls if an Attempt to Boot from Ethernet is Not Successful*

**Revisions Affected** 2.1, 2.0, 1.0

**Details** The system is delayed for up to 4.5 minutes before continuing to the next boot device if an attempt to boot from Ethernet fails for any reason.

This delay is likely to cause an undesirable user experience when the boot sequence attempts Ethernet boot before booting from the primary boot device. For example, it may be desirable to select SYSBOOT [4:0] = 00111b for a product that would normally skip over Ethernet boot which is the first boot device in this sequence and boots from MMC0 which is the second boot device in this sequence. This example boot sequence provides an option to boot from Ethernet as a way to break into the boot sequence to update MMC0 boot code by simply connecting it to an Ethernet host capable of booting the product for the purpose of updating MMC0 boot code. However, this boot sequence attempts to boot from Ethernet first which would insert an undesirable long delay when booting from MMC0.

**Workaround** There is no device-level workaround, but it may be possible to develop a system-level solution that mitigates the effect of this issue when an occasional Ethernet boot is required. In the example described above, the system could be designed with the boot sequence defined by SYSBOOT [4:0] = 11001b which attempts to boot from MMC0 before Ethernet. This does not provide a default method to break into the boot sequence with Ethernet boot, but the product could be designed with a switch that temporarily disables MMC0 boot as a way to invoke Ethernet boot. The switch would only be used to temporarily disable MMC0 while the product boots from Ethernet. Once the product has booted from Ethernet, the switch would be returned to its normal operating mode before the MMC0 boot code is updated.

## Advisory 1.0.26 *I2C: SDA and SCL Open-Drain Output Buffer Issue*

**Revisions Affected** 2.0, 1.0

**Details**

The I2Cx_SDA and I2Cx_SCL outputs are implemented with push-pull 3-state output buffers rather than open-drain output buffers as required by I2C. While it is possible for the push-pull 3-state output buffers to behave as open-drain outputs, an internal timing skew issue causes the outputs to drive a logic-high for a duration of (0–5 ns) before the outputs are disabled. The unexpected high-level pulse will only occur when the SCL or SDA outputs transition from a driven low state to a high-impedance state and there is sufficient internal timing skew on the respective I2C output.

This short high-level pulse injects energy in the I2C signals traces, which causes the I2C signals to sustain a period of ringing as a result of multiple transmission line reflections. This ringing should not cause an issue on the SDA signal because it only occurs at times when SDA is expected to be changing logic levels and the ringing will have time to damp before data is latched by the receiving device. The ringing may have enough amplitude to cross the SCL input buffer switching threshold several times during the first few nanoseconds of this ringing period, which may cause clock glitches. This ringing should not cause a problem if the amplitude is damped within the first 50 ns because I2C devices are required to filter their SCL inputs to remove clock glitches. Therefore, it is important to design the PCB signal traces to limit the duration of the ringing to less than 50 ns. One possible solution is to insert series termination resistors near the AM335x SCL and SDA terminals to attenuate transmission line reflections.

This issue may also cause the SDA output to be in contention with the slave SDA output for the duration of the unexpected high-level pulse when the slave begins its ACK cycle. This occurs because the slave may already be driving SDA low before the unexpected high-level pulse occurs. The glitch that occurs on SDA as a result of this short period of contention does not cause any I2C protocol issue but the peak current applies unwanted stress to both I2C devices and potentially increases power supply noise. Therefore, a series termination resistor located near the respective AM335x SDA terminal is required to limit the current during the short period of contention.

A similar contention problem can occur on SCL when connected to I2C slave devices that support clock stretching. This occurs because the slave is driving SCL low before the unexpected high-level pulse occurs. The glitch that occurs on SCL as a result of this short period of contention does not cause any I2C protocol issue because I2C devices are required to apply a glitch filter to their SCL inputs. However, the peak current applies unwanted stress to both I2C devices and potentially increases power supply noise. Therefore, a series termination resistor located near the respective AM335x SCL terminal is required to limit the current during the short period of contention.

If another master is connected, the unexpected high-level pulses on the SCL and SDA outputs can cause contention during clock synchronization and arbitration. The series termination resistors described above will also limit the contention current in this use case without creating any I2C protocol issue.

**Workaround**

Insert series termination resistors on the SCL and SDA signals and locate them near the AM335x SCL and SDA terminals. The SCL and SDA pullup resistors should also be located near the AM335x SCL and SDA terminals. The placement of the series termination resistor and pullup resistor should be connected as shown in Figure 4.

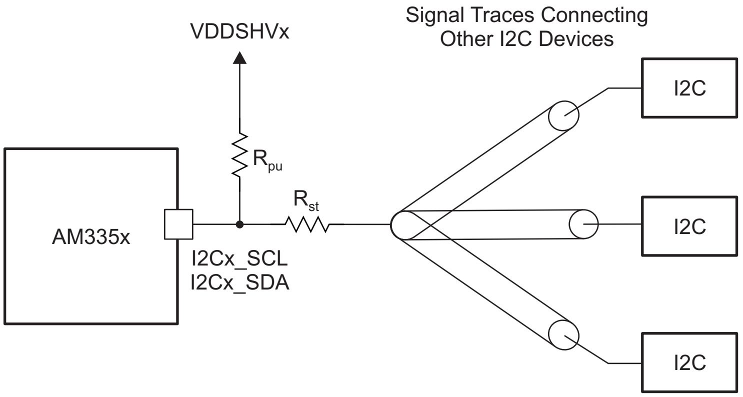

**Figure 4. Placement of Series Termination Resistor and Pullup Resistor**

Table 6 provides series termination and pullup resistor value recommendations. The I2C signal level and respective VDDSHVx power supply voltage is shown in the first column. Two resistor value combination options are provided for each voltage. One option supports a maximum high-level input current of 200 uA to all attached I2C devices, while the other option supports a maximum high-level input current of 100 uA to all attached I2C devices.

**Table 6. Recommended Values for Series Termination Resistor and Pullup Resistor**

| I2C SIGNAL LEVEL AND RESPECTIVE VDDSHVx POWER SUPPLY (V) | SERIES TERMINATION RESISTOR (Ω) | PULLUP RESISTOR (Ω) | NOTES |
|---|---|---|---|
| 1.8 | 60 | 1500 | Maximum high-level input current up to 200 µA |
| 1.8 | 75 | 3300 | Maximum high-level input current up to 100 µA |
| 3.3 | 60 | 3300 | Maximum high-level input current up to 200 µA |
| 3.3 | 75 | 6600 | Maximum high-level input current up to 100 µA |

## Advisory 1.0.27 *LCDC: LIDD DMA Mode Issue*

**Revisions Affected** 2.1, 2.0, 1.0

**Details** After a DMA transfer is complete, the LIDD_DMA_EN bit in the LIDD_CTRL register is driven low and the read/write pointers of the Asynchronous FIFO are designed to reset. However, only the write pointer gets reset and, therefore, the FIFO sequence is corrupt. This could cause flickering or tearing of images being displayed on an LCD.

**Workaround** After each DMA transfer, wait for Done interrupt by polling the DONE_RAW_SET bit in the IRQSTATUS_RAW register, and then write 1 to the DMA_RST and LIDD_RST bits in the CLKC_RESET register to perform software reset of the L3 and LIDD clock domains. This returns the write and read FIFO pointers to their default values and allows for proper FIFO operation.

This workaround has a side effect that needs to be considered. The LCDC drives the LCD control signals to their active state when software reset is asserted if the active states of the LCD panel inputs are opposite the LCDC default states. If this is the case, it may be necessary to add a hardware isolation circuit to the LCD panel chip select signal that disconnects the LCDC output from the LCD panel input before the software reset is asserted. The LCDC output could be reconnected to the LCD panel input after LCDC has been re-initialized. An example isolation circuit may be a single channel bus FET switch inserted in the chip select signal path with the enable being controlled by a GPIO. The LCD panel input would need a pull-up or pull-down resistor to force an inactive state while the switch is open.

## Advisory 1.0.28 LCDC: Raster Mode, Hardware Auto Underflow Restart Does Not Work

**Revisions Affected** 2.1, 2.0, 1.0

**Details** The hardware restart of the LCDC during a FIFO underflow condition does not work. Setting the AUTO_UFLOW_RESTART bit in the LCDC control register has no effect.

**Workaround** This functionality can be implemented using the software reset method outlined in the Disable and Software Reset Sequence section of the *AM335x Sitara Processors Technical Reference Manual* (SPRUH73).

Below are the summarized steps to perform a software reset:

1. Write 0 to the LCDEN bit in the RASTER_CTRL register.
2. Wait for Done interrupt by polling the DONE_RAW_SET bit in the IRQSTATUS_RAW register.
3. Write 1 to the MAIN_RST, or DMA_RST and CORE_RST bits in the CLKC_RESET register.
4. Wait several clock cycles.
5. Write 0 to the MAIN_RST, or DMA_RST and CORE_RST bits in the CLKC_RESET register.
6. Write 1 to the LCDEN bit in the RASTER_CTRL register.

## Advisory 1.0.29 Latch-up Performance: Latch-up Performance Limits for Silicon Revsions 1.0 and 2.0

**Revisions Affected** 2.0, 1.0

**Details** Latch-up performance was improved in silicon revision 2.1 devices and the new limits have been updated in revision F and later of the *AM335x Sitara Processors* data manual (SPRS717).

Table 7 provides latch-up performance limits published in previous revisions of the AM335x data manual and these limits apply to silicon revisions 1.0 and 2.0.

**Table 7. Latch-up Performance Limits - Silicon Revisions 1.0 and 2.0**

| PARAMETER | | MIN | MAX | UNITS |
|---|---|---|---|---|
| Latch-up Performance | Class II (105°C) | 25 | | mA |

**Workaround** Not applicable.

## Advisory 1.0.30 — *OSC0 and OSC1: Noise Immunity Improved When Crystal Circuit is Connected Directly to PCB Digital Ground*

**Revisions Affected** 2.1, 2.0, 1.0

**Details** The *AM335x Sitara Processors* data manual (SPRS717), revision F and earlier, has specified OSC0 crystal circuit grounds only be connected to VSS_OSC in both ZCE and ZCZ packages, and OSC1 crystal circuit grounds only be connected to VSS_RTC in the ZCZ package. The VSS_OSC and VSS_RTC terminals are connected to the VSS terminals inside the AM335x device, which connects the crystal circuit to the PCB digital ground, but this ground connection is a higher impedance connection than a direct connection to the PCB digital ground.

The higher impedance connection through the AM335x device to the PCB digital ground makes it easier for electrical noise to couple into the crystal circuit. If the noise is large enough, it is possible that the oscillator output may produce clock glitches to various internal logic circuits. These clock glitches may cause unexpected behavior.

**Workaround** Connect the VSS_OSC and VSS_RTC terminals and respective crystal circuit component grounds directly to the nearest PCB digital ground, making it more difficult for noise to couple into the crystal circuit.

## Advisory 1.0.31 — *TSC_ADC: False Pen-up Interrupts*

**Revisions Affected** 2.1, 2.0, 1.0

**Details** The touchscreen controller (TSC) determines the pen state (up or down) by checking the respective analog input (AIN0 for 4-wire or AIN4 for 5-wire) voltage level immediately after the Charge step. This does not provide enough time for the analog input voltage to return to the normal Pen-down voltage before testing the pen state. This will cause the TSC to generate a false Pen-up interrupt if the Charge step is enabled with the strong pull-up turned on when using hardware synchronized steps. Figure 5 illustrates the effect on the analog voltage when constant pen pressure is applied to the touchscreen and held through the Charge step.

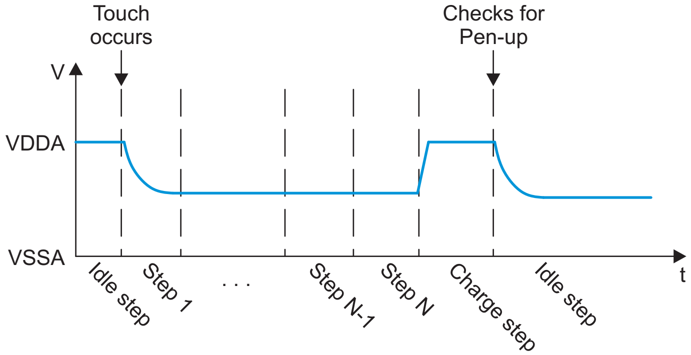

**Figure 5. TSC Generates False Pen-up Interrupt**

**Workarounds** There are two possible workarounds for this problem:

- The first workaround is implemented by configuring the Charge step exactly like the Idle step, where the internal strong pull-up is not turned on. This workaround will remove the strong pull-up on the analog input during the Charge step and leave only the internal weak pull-up. The weak pull-up is not strong enough to effect the normal Pen-down voltage. Therefore, the voltage applied to the analog input will not rise and cause the TSC to generate a false Pen-up interrupt. There is a drawback to this approach; this workaround will cause the Pen-down event to remain active until the weak pull-up turned on in the Idle step charges the touchscreen capacitance, as shown in Figure 6.

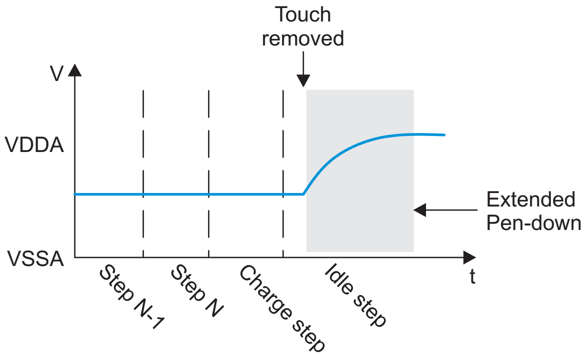

**Figure 6. Pen-down Remains Active in Idle Step**

- Another method is available for 4-wire touchscreen implementations. It this case, it is possible to enable and configure one of the 16 steps to emulate the Charge step with the strong pull-up turned on as the step before the actual Charge step that has been configured like the Idle step with the strong pull-up turned off. The advantage of this implementation is that you are able to charge the touchscreen quicker and avoid the possible false Pen-down interrupts. The disadvantage is that you lose one of the 16 programmable steps and during this step there will be a garbage result stored in the FIFO. You will need to set the Step_ID_tag bit in the CTRL register so that you can identify the garbage result in the FIFO and discard it.

## Advisory 1.0.32 *TSC_ADC: Terminals May be Temporarily Connected Together Through Internal Paths During Power-up Sequence*

**Revisions Affected** 2.1, 2.0, 1.0

**Details**

Any one or more of the TSC_ADC terminals (AIN[7:0], VDDA_ADC, VREFP, VSSA_ADC, and VREFN) may be temporarily connected together through internal paths during the power-up sequence. These internal connections can occur when the VDDA_ADC power source is applied while the VDD_CORE supply is not applied.

The temporary internal connections may occur in any or all of the four analog multiplexers in the TSC_ADC analog front end (AFE). This occurs because the VDD_CORE supply is used to power the decoder logic that selects which analog switch is enabled in each analog multiplexer. It is possible for multiple analog switches in each multiplexer to be enabled while the decoder logic does not have a valid power source. When this occurs, all multiplexer inputs that have their analog switch enabled will simultaneously be connected to the multiplexer output which causes the multiplexer output and all enabled inputs to be connected together.

To better understand all of the possible internal connection paths, read the following paragraphs while referencing the Analog Front End (AFE) Functional Block Diagram, which can be found in the Touchscreen Controller chapter of the *AM335x Sitara Processors Technical Reference Manual* (SPRUH73).

The VDDA_ADC, AN0, AN2, and VREFP terminals are connected to four of the five INT_VREFP (5:1 multiplexer) inputs which allow either one or both power sources (VDDA_ADC and VREFP) to be connected to the AIN0 terminal, AIN2 terminal, and two inputs of each 9:1 multiplexer.

The VSSA_ADC, AIN1, AIN3, and VREFN terminals are connected to the four VREFM (4:1 multiplexer) inputs which allows one or both ground references (VSSA_ADC and VREFN) to be connected to the AIN1 terminal, AIN3 terminal, and two inputs of each 9:1 multiplexer.

The AIN[7:0] and VREFN terminals are connected to the nine INP (9:1 multiplexer) and INM (9:1 multiplexer) inputs which allow any or all AIN[7:0] and VREFN terminals to be connected together.

**Workarounds**

- **TSC_ADC AIN terminals used with a resistive touchscreen:**

  No action required for these AIN terminals.

- **TSC_ADC AIN terminals used as ADC inputs:**

  Insert a series resistor on each of these AIN terminals to limit the output current to an acceptable level for the attached device during this temporary period when VDD_CORE is not applied during power-up. For example: If the attached device has a minimum output impedance (Rsource_min) of 20 Ω and can drive its output as low as VSSA_ADC (0 V) with a maximum output current of 1 mA and the respective AIN terminal is temporarily connected to VDDA_ADC (1.89-V maximum supply voltage), the minimum series resistance (Rseries_min) would need to be 1890 Ω minus Rsource_min to limit the current to 1 mA. Therefore, a series resistor greater than 1870 Ω should be inserted on the analog input signal to protect the attached device.

  The effect on ADC performance also needs to be considered when inserting a series resistor to limit current. Additional series resistance will increase the time required for the source to charge the ADC input capacitor. The ADC accuracy will be reduced if the input capacitor is not completely charged during the sampling time. The following formula should be used to calculate the maximum resistance of the series resistor.

  Rseries_max = [(ADC_ClkDiv + 1) *( SampleDelay + 2) / (116 x $10^{-12}$) * (CLK_M_OSC)] - 200 Ω - Rsource_max

Where:

– Rseries_max is the maximum value of the series resistor

– ADC_ClkDiv is the decimal value of the ADC_CLKDIV register bits [15:0]

– SampleDelay is the decimal value of the respective STEPDELAY register bits [31:24]

– CLK_M_OSC is the frequency of the reference clock connected to OSC0

– Rsource_max is the maximum output impedance of the source.

In the example above, if we assume ADC_ClkDiv is configured to 7 with SampleDelay configured to 0 and a 24-MHz reference clock, Rseries_max would be 5547 Ω minus Rsource_max. Assuming Rsource_max is 47 Ω, the series resistor inserted on the analog input signal should be less than 5500 Ω to provide an accurate voltage measurement.

Note: If a higher value of Rseries_max is required, it may be necessary to increase the sample time by increasing SampleDelay to provide more time for the input capacitance to charge.

- **TSC_ADC AIN terminals not used:**

  The preferred option is to have each unused AIN terminal connected to VSSA_ADC through a 500-Ω resistor. Another option would be to leave the unused AIN terminals open-circuit without any PCB traces connected to these terminals.

- **TSC_ADC VREFP terminal connected to VDDA_ADC and VREFN terminal connected VSSA_ADC:**

  No action required for the VREFP and VREFN terminals.

- **TSC_ADC VREFP terminal not connected to VDDA_ADC and VREFN terminal not connected to VSSA_ADC:**

  Insert a series resistor on each VREFP and VREFN terminal with a 0.1-uF decoupling capacitor connected directly to each VREFP and VREFN terminal. The series resistor value should be selected to limit output current to an acceptable level for the VREFP and VREFN sources during this temporary period when VDD_CORE is not applied during power-up.

  The effect on ADC performance also needs to be considered when selecting the series resistor value. The combined resistance of the series resistor and maximum output impedance of each VREFP and VREFN source should be less than 100 Ω to minimize reduction in ADC performance. This low source impedance requirement may cause the VREFP and VREFN sources to source or sink several mA of current during this temporary period. For example: If the combined minimum resistance of the series resistor and output impedance of the source is 50 Ω, the worst case current requirement during this temporary period would be [1.89 V (maximum supply voltage) / 50 Ω] = 37.8 mA while the typical current requirement for these sources during normal operation would be about 90 uA.

## Advisory 1.0.33 *USB Host: USB Low-Speed Receive-to-Transmit Inter-Packet Delay*

**Revisions Affected** 2.1, 2.0, 1.0

**Details** The universal serial bus (USB) Specification Revision 2.0 requires that a host provide a minimum of 2 bit-times of 'J' state between the single-ended zero (SE0) of an end-of-packet (EOP) and the start of a new packet. While operating as a host in USB low-speed mode, the AM335x device only provides a single bit-time of 'J' state after the SE0 of an EOP that marks the end of a received packet.

The absence of the required minimum 2-bit time 'J' state as part of a USB low-speed RX-to-TX bus transition can result in the failure of some low-speed peripheral devices to function.

Peripherals connected at USB full-speed or high-speed to an AM335x operating in Host mode are not affected by the issue described in this advisory.

**Workaround** None.

## Advisory 1.0.34 *USB2PHY: Register Accesses After a USB Subsystem Soft Reset May Lock Up the Entire System*

**Revisions Affected** 2.1, 2.0, 1.0

**Details** The synchronization bridge connecting the USB2PHY register interface to the L3S interconnect may hang and lock up the entire system. When there is a sequence of any USB2PHY register access, followed by a USB subsystem soft reset initiated with the SOFT_RESET bit in the SYSCONFIG register, followed by any USB2PHY register access, the L3S interconnect may hang on the second USB2PHY register access.

**Workaround** Normal USB operation does not require any USB2PHY register changes. Therefore, one solution for preventing this issue is never access any USB2PHY register.

If there is a requirement to access USB2PHY registers, the system software must ensure there are no USB2PHY register access followed by a USB subsystem soft reset followed by another USB2PHY register access.

## Advisory 1.0.35 *UART: Transactions to MDR1 Register May Cause Undesired Effect on UART Operation*

**Revisions Affected** 2.1, 2.0, 1.0

**Details** The UART logic may generate an internal glitch when accessing the MDR1 registers that causes a dummy under-run condition that will freeze the UART in IrDA transmission. In UART mode, this may corrupt the transferred data (received or transmitted).

**Workaround** To ensure this problem does not occur, the following software initialization sequence must be used each time MDR1 must be changed.

1. If needed, set up the UART by writing the required registers, except MDR1.
2. Set the MDR1.MODE_SELECT bit field appropriately.
3. Wait for five L4 clock cycles + five UART functional clock cycles.
4. Clear TX and RX FIFO in the FCR register to reset its counter logic.
5. Read RESUME register to resume the halted operation.

Note: Step 5 is for IrDA mode only and can be omitted in UART mode.

## Advisory 1.0.36 *EMU0 and EMU1: Terminals Must be Pulled High Before ICEPick Samples*

**Revisions Affected** 2.1, 2.0, 1.0

**Details** The state of the EMU[1:0] terminals are latched during reset to determine ICEPick boot mode. For normal device operation, these terminals must be pulled up to a valid high logic level ( > $V_{IH}$ min) before ICEPick samples the state of these terminals, which occurs [five CLK_M_OSC clock cycles - 10 ns] after the falling edge of WARMRSTn.

Many applications may not require the secondary GPIO function of the EMU[1:0] terminals. In this case, they would only be connected to pull-up resistors, which ensures they are always high when ICEPick samples. However, some applications may need to use these terminals as GPIO where they could be driven low before reset is asserted. This usage of the EMU[1:0] terminals may require special attention to ensure the terminals are allowed to return to a valid high-logic level before ICEPick samples the state of these terminals.

When any device reset is asserted, the pin mux mode of EMU[1:0] terminals configured to operate as GPIO (mode 7) will change back to EMU input (mode 0) on the falling edge of WARMRSTn. This only provides a short period of time for the terminals to return high if driven low before reset is asserted.

**Workaround** If the secondary GPIO function of the EMU[1:0] terminals is not required, only connect these terminals to pull-up resistors.

If the EMU[1:0] terminals are configured to operate as GPIO, the product should be designed such these terminals can be pulled to a valid high-logic level within 190 ns after the falling edge of WARMRSTn.

### 1.0.37 *Watchdog Timers: Delay Needed to Read Some WDTimer Registers After Wakeup*

**Revisions Affected** 2.1, 2.0, 1.0

**Details** Due to internal resynchronization, values read in Watchdog timers WCRR registers right after the timer interface clock (L4) goes from stopped to active may not return the expected values. The most common event leading to this situation occurs upon wakeup from idle.

All the Watchdog timers support only POSTED internal synchronization mode. There is no capability to change the internal synchronization scheme to NON-POSTED by software.

**Workaround** Software has to wait at least (2 timer interface clock cycles + 1 timer functional clock cycle) after L4 clock wakeup before reading the WCRR register of the Watchdog timers.

### 1.0.38 *McASP: McASP to EDMA Synchronization Level Event Can Be Lost*

**Revisions Affected** 2.1, 2.0, 1.0

**Details** The McASP FIFO events to the EDMA can be lost depending on the timing between the McASP side activity and the EDMA side activity. The problem is most likely to occur in a heavily loaded system which can cause the EDMA latency to increase and potentially hit the problematic timing window. When an event is lost, the McASP FIFO Rx path will overflow or the Tx path will underflow. Software intervention is required to recover from this condition.

The issue results due to a state machine boundary condition in the McASP FIFO logic. In normal operation, when "Threshold" (set by the RNUMEVT and WNUMEVT registers) words of data are read/written by the EDMA then the previous event would be cleared. Similarly, when "Threshold" words of data are written/read from the pins, a new event should be set. If these two conditions occur at the same exact time (within a 2-cycle window), then there is a conflict in the set/clear logic and the event is cleared but is not re-asserted to the EDMA.

**Workaround** Since the McASP is a real-time peripheral, any loss of data due to underflow/overflow should be avoided by eliminating the possibility of EDMA read/write completing at the same time as a new McASP Event. Software should configure the system to 1) Maximize time until the deadline for the McASP FIFO, and 2) Minimize EDMA service time for McASP related transfers.

In order to maximize time until deadline, the RNUMEVT and WNUMEVT registers should be set to the largest multiple of "number of serializers active" that is less-than-equal-to 32 words. Since the FIFO is 64-words deep (each word is 32-bits), this gives the maximum time to avoid the boundary condition.

In order to minimize EDMA service time for McASP related transfers multiple options are possible. For example, McASP buffers can be placed in OCMCRAM (since on-chip memories provide a more deterministic and lower-latency path compared to DDR memory) In addition, a dedicated Queue/TC can be allocated to McASP transfers. At minimum, care should be taken to avoid any long transfers on the same Queue/TC to avoid head-of-line blocking latency.

### 1.0.39 *GPTimer: Delay Needed to Read Some GPTimer Registers After Wakeup*

**Revisions Affected** 2.1, 2.0, 1.0

**Details** If a general-purpose timer (GPTimer) is in posted mode (TSICR [2].POSTED=1), due to internal resynchronizations, values read in the TCRR, TCAR1 and TCAR2 registers right after the timer interface clock (L4) goes from stopped to active may not return the expected values. The most common event leading to this situation occurs upon wakeup from idle.

GPTimer non-posted synchronization mode is not impacted by this limitation.

**Workaround** For reliable counter read upon wakeup from IDLE state, software needs to issue a non-posted read to get an accurate value. To get this non-posted read, TSICR [2].POSTED needs to be set to '0'.

### 1.0.41 *UART: UART0-5 Do Not Acknowledge Idle Request After DMA Has Been Enabled*

**Revisions Affected** 2.1, 2.0, 1.0

**Details** All UART modules (UART0-5) in the device do not acknowledge an idle request after enabling the module's DMA feature, even if the DMA is subsequently disabled. Thus, the UART module cannot be clock idled after enabling DMA with

- UART_SCR.DMAMODECTL = 1 and UART_SCR.DMAMODE2 != 0

  OR
- UART_SCR.DMAMODECTL = 0 and UART_FCR.DMA_MODE = 1

A consequence of this is that CM_WKUP_UARTx_CLKCTRL will remain in transition when trying to disable the module (UARTx_CLKCTRL = 0x10000) and the associated CLKACTIVITY bit will remain active.

**Workaround** Initiating a soft reset (UART_SYSC.SOFTRESET = 1) will allow the module to acknowledge the idle request.

## Advisory 1.0.42 *DebugSS: DebugSS Does Not Acknowledge Idle Request*

**Revisions Affected** 2.1, 2.0, 1.0

**Details** The Debug Subsystem (DEBUGSS) does not acknowledge an idle request. Thus the DEBUGSS module cannot be clock idled during normal operation. The idle status in register PRCM_CM_WKUP_DBGSS_CLKCTRL continually reports an 'in-transition' state (IDLEST=1) after attempting to turn off its clock (MODULEMODE=0).

**Workaround** None

# Revision History

**Changes from F Revision (November2013) to G Revision** **Page**

- Added new advisories to Table 4 .......... 6
- Added Advisory 1.0.37, Watchdog Timers: Delay Needed to Read Some WD Timer Registers After Wakeup .......... 38
- Added Advisory 1.0.38, McASP: McASP to EDMA Synchronization Level Event Can Be Lost .......... 38
- Added Advisory 1.0.39, GPTimer: Delay Needed to Read Some GPTimer Registers After Wakeup .......... 39
- Added Advisory 1.0.41, UART: UART0-5 Do Not Acknowledge Idle Request After DMA Has Been Enabled .......... 39
- Added Advisory 1.0.42, DebugSS: DebugSS Does Not Acknowledge Idle Request .......... 40

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

# IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as “components”) are sold subject to TI’s terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI’s terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

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

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

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

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI’s goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or “enhanced plastic” are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have ***not*** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

| **Products** | | **Applications** | |
|---|---|---|---|
| Audio | www.ti.com/audio | Automotive and Transportation | www.ti.com/automotive |
| Amplifiers | amplifier.ti.com | Communications and Telecom | www.ti.com/communications |
| Data Converters | dataconverter.ti.com | Computers and Peripherals | www.ti.com/computers |
| DLP® Products | www.dlp.com | Consumer Electronics | www.ti.com/consumer-apps |
| DSP | dsp.ti.com | Energy and Lighting | www.ti.com/energy |
| Clocks and Timers | www.ti.com/clocks | Industrial | www.ti.com/industrial |
| Interface | interface.ti.com | Medical | www.ti.com/medical |
| Logic | logic.ti.com | Security | www.ti.com/security |
| Power Mgmt | power.ti.com | Space, Avionics and Defense | www.ti.com/space-avionics-defense |
| Microcontrollers | microcontroller.ti.com | Video and Imaging | www.ti.com/video |
| RFID | www.ti-rfid.com | | |
| OMAP Applications Processors | www.ti.com/omap | **TI E2E Community** | e2e.ti.com |
| Wireless Connectivity | www.ti.com/wirelessconnectivity | | |

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265