AMIC110 Sitara™ SoC
Silicon Revision 2.1

Silicon Errata

TEXAS INSTRUMENTS

Literature Number: SPRZ446
January 2017

Contents

Silicon Errata

SPRZ446–January 2017

AMIC110 Sitara™ SoC Silicon Revision 2.1

1 Introduction

This document describes the known exceptions to the functional specifications for the AMIC110 Sitara™ ARM® Cortex®-A8 processors. (See the device-specific data manual, *AMIC110 Sitara SoC*.)

For additional information, see the latest version of the *AM335x and AMIC110 Sitara Processors Technical Reference Manual*.

1.1 AMIC110 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, XAMIC110BZCZA). 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 AMIC110 device markings.

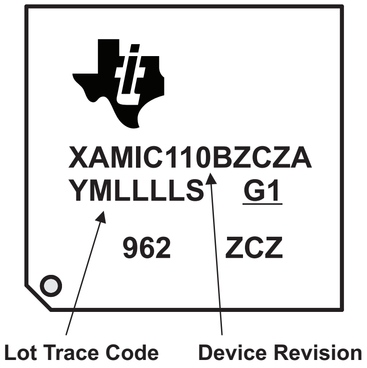

Figure 1. Example of Device Revision Code for the AMIC110 Microprocessor

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 AMIC110 device shown in this device marking example is one of several valid part numbers for the AMIC110 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) 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 AMIC110x, 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
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
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
		2.1
Section 3.1.1	DDR3: JEDEC Compliance for Maximum Self-Refresh Command Limit	X
Section 3.1.2	Boot: USB Boot ROM Code Uses Default DATAPOLARITY	X
Section 3.1.3	Pin Multiplexing: Valid IO Sets and Restrictions	X
Section 3.1.4	OSC1: RTC_XTALIN Terminal Has an Internal Pull-up Resistor When OSC1 is Disabled	X

Table 4. All Design Exceptions to Functional Specifications

NUMBER	TITLE	SILICON REVISION AFFECTED
		2.1
Advisory 1.0.1	DDR2, DDR3, mDDR PHY: Control and Status Registers Configured for Write Only	X
Advisory 1.0.2	Debug Subsystem: EMU[4:2] Signals Are Not Available by Default After Reset	X
Advisory 1.0.3	Debug Subsystem: Internal Inputs Tied-off to the Wrong Value	X
Advisory 1.0.12	UART: Extra Assertion of FIFO Transmit DMA Request, UARTi_DMA_TX	X
Advisory 1.0.19	DDR3: Fully-Automated Hardware READ and WRITE Leveling Not Supported	X
Advisory 1.0.22	EMIF: Dynamic Voltage Frequency Scaling (DVFS) is Not Supported	X
Advisory 1.0.24	Boot: System Boot is Not Reliable if Reset is Asserted While Operating in OPP50	X
Advisory 1.0.30	OSC0 and OSC1: Noise Immunity Improved When Crystal Circuit is Connected Directly to PCB Digital Ground	X
Advisory 1.0.33	USB Host: USB Low-Speed Receive-to-Transmit Inter-Packet Delay	X
Advisory 1.0.34	USB2PHY: Register Accesses After a USB Subsystem Soft Reset May Lock Up the Entire System	X
Advisory 1.0.35	UART: Transactions to MDR1 Register May Cause Undesired Effect on UART Operation	X
Advisory 1.0.36	EMU0 and EMU1: Terminals Must Be Pulled High Before ICEPick Samples	X
Advisory 1.0.37	Watchdog Timers: Delay Needed to Read Some WDTimer Registers After Wakeup	X
Advisory 1.0.38	McASP: McASP to EDMA Synchronization Level Event Can Be Lost	X
Advisory 1.0.39	GPTimer: Delay Needed to Read Some GPTimer Registers After Wakeup	X
Advisory 1.0.41	UART: UART0-5 Do Not Acknowledge Idle Request After DMA Has Been Enabled	X
Advisory 1.0.42	DebugSS: DebugSS Does Not Acknowledge Idle Request	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 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.

3.1.2 Boot: USB Boot ROM Code Uses Default DATAPOLARITY

The AMIC110 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.

3.1.3 Pin Multiplexing: Valid IO Sets and Restrictions

The AMIC110 device contains many peripheral interfaces. In order to reduce package size and lower overall system cost while maintaining maximum functionality, many of the AMIC110 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 AMIC110 device-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 AMIC110 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.

3.1.4 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.

```
This is how code is displayed
```

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

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

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 AMIC110 EMU[4:2] signals can not be used to export trace messages from the Debug subsystem since AMIC110 device supports 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

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 AMIC110 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 AMIC110 EMU[1:0] pins to operate as GPIO if you need to export trace messages.

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

Revisions Affected 2.1

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

Revisions Affected 2.1

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

Revisions Affected 2.1

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

Revisions Affected 2.1

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. However, this issue can be avoided by selecting a boot sequence that does not include Ethernet boot. Alternatively, the effect of this issue can be mitigated by selecting a boot sequence that places Ethernet boot after all desired boot options.

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

Revisions Affected 2.1

Details

AMIC110 Sitara Processors, revision A and earlier, has specified OSC0 crystal circuit grounds only be connected to VSS_OSC in the ZCZ package, 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 AMIC110 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 AMIC110 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 1.0.33 *USB Host: USB Low-Speed Receive-to-Transmit Inter-Packet Delay*

Revisions Affected 2.1

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 AMIC110 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 AMIC110 device 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

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

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

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.

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

Revisions Affected 2.1

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.

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

Revisions Affected 2.1

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.

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

Revisions Affected 2.1

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'.

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

Revisions Affected 2.1

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

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

Trademarks

Sitara is a trademark of Texas Instruments.
ARM, Cortex are registered trademarks of ARM Ltd or its subsidiaries.
All other trademarks are the property of their respective owners.

IMPORTANT NOTICE FOR TI DESIGN INFORMATION AND RESOURCES

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