Errata
AMIC120 Sitara™ Processors
Silicon Revision 1.2

![Texas Instruments]

Table of Contents

Trademarks

Sitara™ are trademarks of Texas Instruments.

Cortex® and ARM® are registered trademarks of ARM Ltd or its subsidiaries.

All trademarks are the property of their respective owners.

1 Introduction

This document describes the known exceptions to the functional specifications (advisories) for the AMIC120 Sitara Cortex®-A9 Processors. See *AMIC120 Sitara Processors*.

This document also contains usage notes. Usage notes describe situations where the device's behavior may not match presumed or documented behavior. This may include behaviors that affect device performance or functional correctness.

For additional information, see the latest version of the *AM437x and AMIC120 Sitara Processors Technical Reference Manual*.

1.1 Device and Development Support Tool Nomenclature

To designate the stages in the product development cycle, TI assigns prefixes to the part numbers of all processors and support tools. Each device has one of three prefixes: X, P, or null (no prefix) (for example, XAMIC120ZDN). 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 1-1 provides an example of the device markings.
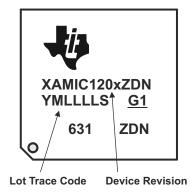

Figure 1-1. Example of Device Revision Codes for the Device Processor

NOTES:

1. 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.
2. The device shown in this device marking example are two of several valid part numbers for the family of devices.
3. The device revision code is the device revision (A, B, and so on).
4. YM denotes year and month.
5. LLLL denotes Lot Trace Code.
6. 631 is a generic family marking ID.
7. G1 denotes green, lead-free.
8. ZDN is the package designator.
9. S denotes Assembly Site Code.
10. 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 AMIC120, where "x" denotes the silicon revision. Table 1-1 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 1-1. Production Device Revision Codes

DEVICE REVISION CODE	SILICON REVISION	COMMENTS
B	1.2	Silicon revision PG1.2

Each silicon revision uses a specific revision of TI's ARM® Cortex®-A9 processor. The ARM Cortex-A9 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 0x3BFFC on silicon revision 1.1 and 0x3FFFC on silicon revision 1.2. The ROM code version consists of two decimal numbers: major and minor. The major number is 0x27, and the minor number counts ROM code version. The ROM code version is coded as hexadecimal readable values; for example, ROM version 27.02 is coded as 0x2702. Table 1-2 shows the ARM Cortex-A9 Variant and Revision, Device Revision, and ROM Code Revision values for each silicon revision of the device.

Table 1-2. Silicon Revision Variables

SILICON REVISION	ARM CORTEX-A9 VARIANT AND REVISION	DEVICE REVISION	ROM REVISION	PL310 CACHE CONTROLLER VERSION
1.2	r2p10	0002b	27.02	r3p2

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 2-1. All Usage Notes

NUMBER	TITLE	SILICON REVISION AFFECTED
		1.2
Section 3.1.1	LPDDR2/DDR3: JEDEC Compliance for Minimum Self-Refresh Command Interval	X
Section 3.1.2	DDR3/DDR3L: JEDEC Specification Violation for DDR3 RESET Signal When Implementing RTC+DDR Mode	X

Table 2-2. All Design Exceptions to Functional Specifications

NUMBER	TITLE	SILICON REVISION AFFECTED
		1.2
Advisory 1	UART: Extra Assertion of FIFO Transmit DMA Request, UARTi_DMA_TX	X
Advisory 11	Asynchronous Bridge Corruption	X
Advisory 12	DebugSS: Register Identifier Field (MasterID) of Statistics Collector Has a Default Value of 0x0 Instead of the Expected ID	X
Advisory 16	McASP: McASP to EDMA Synchronization Level Event Can Be Lost	X
Advisory 17	DebugSS: DebugSS Does Not Acknowledge Idle Request	X
Advisory 20	GPTimer: Delay Needed to Read Some GPTimer Registers After Wakeup	X
Advisory 21	UART: UART0-5 Do Not Acknowledge Idle Request After DMA Has Been Enabled	X
Advisory 22	Watchdog Timers: Delay Needed to Read Some WDTimer Registers After Wakeup	X
Advisory 24	VDD_MPU_MON Not Connected to Die	X
Advisory 26	AutoCMD12 Mode: CMD12 Command is Not Issued on Write Transfer Completion	X
Advisory 27	UART: Spurious UART Interrupts When Using EDMA	X
Advisory 28	UART: Transactions to MDR1 Register May Cause Undesired Effect on UART Operation	X
i2223	ROM: Non-muxed Fast NOR boot does not onfigure A26 and A27	X
i2224	DCAN: RAMINIT_DONE intermittently fails to latch completion	X
i912	QSPI: QSPI register bitfield incorrectly masked when read	X
i2225	UART: Possible underflow condition when using EDMA with UART1, UART2, and UART3	X
i2226	PRU-ICSS: Burst data transfer between ICSS instances not supported	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 LPDDR2/DDR3: JEDEC Compliance for Minimum Self-Refresh Command Interval

When using LPDDR2 or DDR3 EMIF Self-Refresh, it is possible to violate the minimum self-refresh command interval requirement specified in the JEDEC standard LPDDR2 Specifications (JESD209-2F, June 2013) and DDR3 SDRAM Specifications (JESD79-3F, July 2010). This requirement states: "The use of Self Refresh mode introduces the possibility that an internally timed refresh event can be missed when CKE is raised for exit from Self Refresh mode. Upon exit from Self Refresh, it is required that at least one Refresh command (8 per-bank or 1 all-bank) is issued before entry into a subsequent Self Refresh."

To meet this minimum when using the LPDDR2 or DDR3 EMIF and Self-Refresh mode (setting PWR_MGMT_CTRL.REG_LP_MODE=2), set the PWR_MGMT_CTRL.REG_SR_TIM register to a time greater than the refresh rate of the LPDDR2 or DDR3.

For example, if the refresh rate for the DDR3 is 7.8 µs and the DDR3 is running at 303 MHz, the minimum time to ensure the above requirement is:

7.8 µs / 3.3 ns = 2363 DDR clock cycles

Thus, SR_TIM must be no less than 0x9 (4096 clocks).

3.1.2 DDR3/DDR3L: JEDEC Specification Violation for DDR3 RESET Signal When Implementing RTC+DDR Mode

DDR3/DDR3L SDRAM specification (JESD79-J3, July 2010) states that "RESET# is recommended to be maintained below 0.2x VDDS_DDR" during initial power ramp. The main reason for this is to ensure the DDR3/DDR3L outputs are in High-Z to avoid an excessive current depending on bus activity. When implementing RTC+DDR mode, an external pull-up resistor of 1K or lower is required on DDR_RESETn to maintain DDR3/DDR3L memory in self-refresh. The external pull-up creates a spec violation during power up because DDR_RESETn will ramp during initial power cycle (the ramp will follow the voltage rail of the pull-up resistor). However, all DDR3/DDR3L I/Os of the AMIC120 DDR3/DDR3L interface are disabled during power ramp and until DDR3/DDR3L is initialized. Thus, there will be no signal contention and no excessive current on the DDR3/DDR3L interface. This specification violation will not negatively affect the AMIC120 device or the DDR3/DDR3L memory devices. Note, this violation is only applicable for low-power designs implementing RTC+DDR mode.

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 removed in future revisions of this document because the design exception was fixed or documented in the device-specific data manual or technical reference manual. When items are deleted, the remaining advisory numbers are not re-sequenced.

3.2.1 Advisory List

Advisory 1	***UART: Extra Assertion of FIFO Transmit DMA Request, UARTi_DMA_TX***
Revisions Affected	1.2
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 11	***Asynchronous Bridge Corruption***
Revisions Affected	1.2
Details	If data is stalled inside an asynchronous bridge because of back pressure, it may be accepted multiple times and create pointer misalignment that corrupts the next transfers on that data path until the system is reset. There is no recovery procedure once the issue is hit because the path remains consistently broken. The async bridge can be found on the path between MPU to L3 interconnect (to EMIF) and Cortex M3 to L3 interconnect (to EMIF). This situation can happen only when the idle is initiated by a master request disconnection, which is trigged by software when executing WFI.
Workaround	All the initiators connected through the asynchronous bridge must ensure that data path is properly drained before issuing WFI. This condition is met if one strongly ordered access is performed to the target right before executing the WFI.

Advisory 12	***DebugSS: Register Identifier Field (MasterID) of Statistics Collector Has a Default Value of 0x0 Instead of the Expected ID***

Revisions Affected   1.2

Details   The master ID for the statistics collectors in the Debug SubSystem is implemented as a programmable register. This register is usually read-only and must have a unique reset value for each statistics collector. The reset value of this register is 0x0 instead of the expected master ID.   As a result, there is incompatibility with the current programming model.

Workaround   Debug software must program and configure this field to a correct value when statistics collection in the Debug Subsystem is required.

Advisory 16	*McASP: McASP to EDMA Synchronization Level Event Can Be Lost*

Revisions Affected 1.2

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 17	***DebugSS: DebugSS Does Not Acknowledge Idle Request***
Revisions Affected	1.2
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

Advisory 20	***GPTimer: Delay Needed to Read Some GPTimer Registers After Wakeup***
Revisions Affected	1.2
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 21	***UART: UART0-5 Do Not Acknowledge Idle Request After DMA Has Been Enabled***
Revisions Affected	1.2
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 22	***Watchdog Timers: Delay Needed to Read Some WDTimer Registers After Wakeup***
Revisions Affected	1.2
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 24	***VDD_MPU_MON Not Connected to Die***
Revisions Affected	1.2
Details	VDD_MPU_MON (pin D20 on the ZDN package) is unconnected in the device and does not provide a kelvin connection of VDD_MPU. Therefore, this terminal cannot be used as VDD_MPU power supply feedback input for supporting remote sensing. Additionally, this pin can be left floating; that is, no-connect.
Workaround	Use VDD_MPU terminals for power supply feedback connection for enabling the remote sensing feature instead of VDD_MPU_MON.

Advisory 26	***AutoCMD12 Mode: CMD12 Command is Not Issued on Write Transfer Completion***
Revisions Affected	1.2
Details	When using AutoCMD12 mode in write transfer with DMA and SD_CMD.BCE is disabled, then the CMD12 command is not issued automatically after write transfer completion.
Workaround	Instead of setting the SD_CMD.ACEN bit to 0x1 to enable AutoCMD12 mode, software sends the CMD12 command at the end of write transfers (after the SD_STAT.TC bit goes High).

Advisory 27	***UART: Spurious UART Interrupts When Using EDMA***
Revisions Affected	1.2
Details	Spurious UART interrupts may occur when enabling DMA mode (FCR.DMA_MODE) using the EDMA. The Interrupt Controller flags that a UART interrupt has occurred; however, the associated IT_PENDING bit remains set to 1, indicating that no interrupt is pending.
Workaround	Acknowledge the spurious interrupts for every occurrence. The issue can be avoided by disabling receive data interrupts using the RHRIT bit; however, be aware that this also disables RX timeout interrupts, which may not be practical for all use cases.

Advisory 28	***UART: Transactions to MDR1 Register May Cause Undesired Effect on UART Operation***
Revisions Affected	1.2
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.

i2223	***ROM: Non-muxed Fast NOR boot does not configure A26 and A27***

Revisions Affected   1.2

Details

Boot mode affected: NOR Low Latency (FAST_NOR)

In the non-muxed NOR Low Latency Boot Mode, the ROM uses pinmux option 0 to configure the pin mux configuration, however A26 and A27 are not configured to pinmux mode 4, and thus are not driven during boot.

Workaround

Use non-muxed NOR boot, which correctly configures A26 and A27. Or, if FAST_NOR boot must be used, external pull-down resistors must be included on GPMC_A26 and GPMC_A27 signals if these signals are connected to the NOR flash. Appropriate resistor values must be determined to ensure these signals are held in a valid low state during boot.

i2224	***DCAN: RAMINIT_DONE intermittently fails to latch completion***

Revisions Affected   1.2

Details

The DCAN_RAMINIT bits inside the CTRL_DCAN_RAMINIT register do not always capture the completion signal after the RAM has been initialized.

Workaround

The DCAN RAM initialization takes 69 DCAN ICLK cycles (The DCAN ICLK is driven by the L4 interconnect clock L4_PER_CLK). For a typical system with L4_PER_CLK=100 MHz, this corresponds to 690 ns. After the 0->1 transition of RAMINIT_START, the CAN registers should not be accessed for at least 69 DCAN ICLK cycles.

i912	***QSPI: QSPI register bitfield incorrectly masked when read***

Revisions Affected 1.2

Details Due to an integration error in the device, bits [25:24] (part of bitfield WLEN) of register QSPI_CMD_REG are masked and always read zero.

Workaround The bitfield functions correctly, as all bits in WLEN are writable. The errata only applies to reading the bitfield WLEN.

i2225	***Possible underflow condition when using EDMA with UART1, UART2, and UART3***

Revisions Affected 1.2

Details UART1, UART2, and UART3 can trigger EDMA pre-maturely when set up to read data out of UART FIFO, resulting in an underflow condition which leads to data corruption. EDMA should not be used with these UART instances to read data from the UART Receive FIFO. Note that UART0, UART4, and UART5 are not affected by this errata, as these are different hardware IP modules.

Workaround If DMA is needed for reads from UART, use UART0, UART4, or UART5

i2226	***PRU-ICSS: Burst data transfer between ICSS instances not supported***

Revisions Affected   1.2

Details

The expansion port between PRU-ICSS0 and PRU-ICSS1 does not support burst reads or burst writes. In this case, "burst" means SBBO and LBBO assembly instructions that are moving data longer than 4 bytes from one ICSS to the other ICSS. These two assembly instructions do not work properly if the local data memory address for the expansion port is used. The expansion port to allow a PRU to access the other PRU-ICSS starts at local address 0x0004_0000. See the PRU-ICSS Memory Map Overview in the Technical Reference Manual for more information on memory maps.

Workaround

1) Use the global memory map address with SBBO/LBBO instructions that interact with data in the other ICSS.

OR

2) Use the local memory map address for the expansion port to load or store data in the other ICSS, but only move a maximum of 4 bytes per SBBO/LBBO instruction. The 4 byte copies may be performed in a while loop, in an unrolled jump table, etc.

IMPORTANT NOTICE AND DISCLAIMER