Errata

AM437x Sitara™ Processors Silicon Revisions 1.1, 1.2

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 AM437x Sitara Cortex®-A9 Processors. See *AM437x 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 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, XAM4376ZDN). 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.

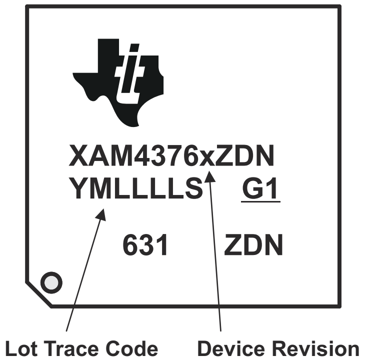

Figure 1-1. Example of Device Revision Codes for the Device Processor

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 AM4376x, 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
A	1.1	Silicon revision PG1.1
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
1.1	r2p10	0001b	27.01	r3p2
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	
		1.1	1.2
Section 3.1.1	LPDDR2/DDR3: JEDEC Compliance for Minimum Self-Refresh Command Interval	X	X
Section 3.1.2	DDR3/DDR3L: JEDEC Specification Violation for DDR3 RESET Signal When Implementing RTC+DDR Mode	X	X

Table 2-2. All Design Exceptions to Functional Specifications

NUMBER	TITLE	SILICON REVISION AFFECTED	
		1.1	1.2
Advisory 1	UART: Extra Assertion of FIFO Transmit DMA Request, UARTi_DMA_TX	X	X
Advisory 2	ROM: USB Host Boot is Unsupported	X	
Advisory 3	ROM: USB Client Boot is Unsupported	X	
Advisory 4	ROM: RGMII Clocking Register is Not Configured Properly at OPP50	X	
Advisory 5	ROM: Trace Vector Does Not Reflect that TFTP Transfer Has Been Initiated	X	
Advisory 6	ROM: Booting from Redundant Image in NAND Does Not Work as Expected	X	
Advisory 7	ROM: NAND Booting is Slower than Expected	X	
Advisory 8	ROM: In NOR Low Latency Boot Mode, Wait Monitoring Will Not Work	X	
Advisory 9	ROM: NAND ECC May Not Be Chosen Correctly by the ROM	X	
Advisory 10	ROM: Peripheral Boot is Not Supported	X	
Advisory 11	Asynchronous Bridge Corruption	X	X
Advisory 12	DebugSS: Register Identifier Field (MasterID) of Statistics Collector Has a Default Value of 0x0 Instead of the Expected ID	X	X
Advisory 13	DSS: DSS Smart Standby May Cause Synchronization Issues	X	X
Advisory 14	DSS: DSS Limitations	X	X
Advisory 15	ROM: NAND Boot Mode is Unsupported	X	
Advisory 16	McASP: McASP to EDMA Synchronization Level Event Can Be Lost	X	X
Advisory 17	DebugSS: DebugSS Does Not Acknowledge Idle Request	X	X
Advisory 19	TSC_ADC: False Pen-up Interrupts	X	X
Advisory 20	GPTimer: Delay Needed to Read Some GPTimer Registers After Wakeup	X	X
Advisory 21	UART: UART0-5 Do Not Acknowledge Idle Request After DMA Has Been Enabled	X	X
Advisory 22	Watchdog Timers: Delay Needed to Read Some WDTimer Registers After Wakeup	X	X
Advisory 24	VDD_MPU_MON Not Connected to Die	X	X
Advisory 25	Ethernet Boot: ROM May Select 1-Gbit, Half-Duplex Mode During Auto-negotiation and Fail to Boot	X	X

Table 2-2. All Design Exceptions to Functional Specifications (continued)

NUMBER	TITLE	SILICON REVISION AFFECTED	
		1.1	1.2
Advisory 26	AutoCMD12 Mode: CMD12 Command is Not Issued on Write Transfer Completion	X	X
Advisory 27	UART: Spurious UART Interrupts When Using EDMA	X	X
Advisory 28	UART: Transactions to MDR1 Register May Cause Undesired Effect on UART Operation	X	X
i2223	ROM: Non-muxed Fast NOR boot does not onfigure A26 and A27	X	X
i2224	DCAN: RAMINIT_DONE intermittently fails to latch completion	X	X
i912	QSPI: QSPI register bitfield incorrectly masked when read	X	X
i2225	Possible underflow condition when using EDMA with UART1, UART2, and UART3	X	X
i2226	PRU-ICSS: Burst data transfer between ICSS instances not supported	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 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 AM437x 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 AM437x 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 1.1, 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 2 *ROM: USB Host Boot is Unsupported*

Revisions Affected 1.1

Details Boot modes affected: USB_MS

USB Host boot (USB_MS) is unsupported in the affected revisions.

Workaround None

Advisory 3 *ROM: USB Client Boot is Unsupported*

Revisions Affected 1.1

Details Boot modes affected: USB_CL

USB Client boot (USB_CL) is unsupported in affected revisions.

Workaround None

Advisory 4 *ROM: RGMII Clocking Register is Not Configured Properly at OPP50*

Revisions Affected 1.1

Details Boot modes affected: EMAC1 (RGMII mode only)

ROM incorrectly configures a divisor in the PRCM and results in the RGMII modules receiving a 20-MHz clock instead of the required 50-MHz clock.

Workaround None

Advisory 5 *ROM: Trace Vector Does Not Reflect that TFTP Transfer Has Been Initiated*

Revisions Affected 1.1

Details Boot mode affected: EMAC1

Bit number 16 of Trace Vector 5 indicates if the TFTP transfer for the EMAC Boot has been initiated. This bit is not implemented by the ROM Code.

Workaround None

Advisory 6 *ROM: Booting from Redundant Image in NAND Does Not Work as Expected*

Revisions Affected 1.1

Details This original Advisory has been superseded by Advisory 15.

Workaround None

Advisory 7 ***ROM: NAND Booting is Slower than Expected***

Revisions Affected 1.1

Details This original Advisory has been superseded by Advisory 15.

Workaround None

Advisory 8 — *ROM: In NOR Low Latency Boot Mode, Wait Monitoring Will Not Work*

Revisions Affected 1.1

Details Boot mode affected: NOR Low Latency (FAST_NOR)

In the NOR Low Latency Boot Mode, the pin mux configuration for the GPMC_WAIT0 is not performed by the ROM Code. As a result, a NOR Flash that needs wait pin configuration is not supported. Wait monitoring during boot should not be enabled (SYSBOOT[9] should remain 0).

This affects only NOR Low Latency Boot Mode (FAST_NOR). NOR boot is not affected by this issue.

Workaround The GPMC_OE (Output Enable) signal is active for 833.32 ns.

If the Flash is able to receive data within 833.32 ns, FAST_NOR will work as normal.

Advisory 9 — *ROM: NAND ECC May Not Be Chosen Correctly by the ROM*

Revisions Affected 1.1

Details This original Advisory has been superseded by Advisory 15.

Workaround None

Advisory 10 — *ROM: Peripheral Boot is Not Supported*

Revisions Affected 1.1

Details Boot modes affected: USB_CL, UART, EMAC

Peripheral boot modes USB_CL, UART, and EMAC are not supported on affected devices.

Workaround None

Advisory 11 — *Asynchronous Bridge Corruption*

Revisions Affected 1.1, 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 1.1, 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 13 — *DSS: DSS Smart Standby May Cause Synchronization Issues*

Revisions Affected 1.1, 1.2

Details Enabling DSS Smart Standby may cause synchronization issues with the PRCM, which may render the DSS inoperable.

DSS MIDLE_MODE = 2 (SmartStandby) should not be used because of possible synchronization issues.

Workarounds The following sequence must be followed to avoid synchronization issues.

During DSS initialization

- Bring DSS module out of Standby by configuring *DISPC_SYSCFG.MIDLEMODE = NO_STANDBY* (Value: 1).
- Configure the remaining DSS memory mapped registers.
- Enable the LCD channel by configuring DISPC_CTRL.LCD_EN.

During DSS Disable

- Disable LCD channel by configuring DISPC_CTRL.LCD_EN.
- Put DSS in Standby by configuring *DISPC_SYSCFG.MIDLEMODE = FORCE_STANDBY* (Value: 0).

Advisory 14 *DSS: DSS Limitations*

Revisions Affected 1.1, 1.2

Details Under certain system-level conditions where bandwidth usage is high or spikes, the DSS can experience a FIFO underflow condition causing screen tearing, flickering, or otherwise corrupted LCD data output. Each pipeline in the DSS has its own dedicated 1KB FIFO which, in certain conditions, is not large enough to sustain constant LCD data output when concurrent continuous writes to memory are being performed by other device peripherals. Therefore, system-level peak bandwidth usage must be taken into consideration in order to avoid frequent FIFO underflows from occurring. In the case of high resolution displays, typically for resolutions over 1024x768, FIFO underflows can occur regularly if no steps are taken to properly manage the DSS memory bandwidth requirements. It is recommended to perform system-level stress tests based on the end-application requirements to determine if the worst-case bandwidth consumption causes the DSS FIFO to underflow, especially when using the DSS to drive LCD panels over 1024x768 resolution. In the case where frequent FIFO underflows occur, the following workaround is required for stable operation.

Workarounds There are three features than can be used to reduce the likelihood of experiencing a DSS underflow: EMIF Class of Service mapping, DSS FIFO merge, and L3 interconnect A9 bandwidth limit. Depending on the end application, a combination of these features can be used to allow for proper DSS operation while minimizing overall system performance impact.

EMIF Class of Service

The Class of Service feature within the EMIF allows the user to assign the DSS to a priority that is higher than the other masters within the system. This ensures that during EMIF arbitration the DSS maintains the highest possible priority. When configuring the EMIF, the DSS should be assigned to Class of Service 1 (COS1). It is recommended to use this feature anytime the DSS is being used and the DSS must always remain the only peripheral assigned to Class of Service 1 to prevent other masters from being selected before the DSS during arbitration. To assign the DSS to COS1 and enable class of service mapping on COS1, the user must write the DSS connection ID, 0x25, to the CONNID_3_COS_1 bit field and set the CONNID_COS_1_MAP_EN bit of the EMIF4D_CONNECTION_ID_TO_CLASS_OF_SERVICE_1_MAPPING register.

To limit the amount of time a DSS command will remain in the EMIF command queue, the user must modify the COS_COUNT_1 bit field in the EMIF4D_COS_CONFIG register. It is recommended to start with the COS_COUNT_1 field be set to 0x0F, which means that if a command sits in the queue for 16 clock cycles then it will be executed next. If a FIFO underflow still occurs with the COS_COUNT_1 field set to 0x0F, the user should decrease this value to further reduce the maximum latency in DSS command execution. On the other hand, the COS_COUNT_1 field can be increased to improve overall system performance as long as system stress tests show there are no DSS FIFO underflows occurring.

MODULE	REGISTER	FIELD	VALUE
EMIF	EMIF4D_CONNECTION_ID_TO_CLASS_OF_SERVICE_1_MAPPING	CONNID_3_COS_1	0x25
EMIF	EMIF4D_CONNECTION_ID_TO_CLASS_OF_SERVICE_1_MAPPING	CONNID_COS_1_MAP_EN	0x1
EMIF	EMIF4D_COS_CONFIG	COS_COUNT_1	0x0-0xF[1]

[1] See description of EMIF Class of Service workaround above.

FIFO Merge

The FIFO merge feature will consolidate the three DSS pipeline FIFOs into a single 3KB deep FIFO. However, using the FIFO merge feature introduces some feature limitations within the DSS. By performing a FIFO merge, the DSS is limited to the use of only a single plane which makes scaling and overlay features less useful. The FIFO merge feature can be enabled by setting bit 14, FIFO_MERGE, of the DISPC_CFG register in the DSS.

MODULE	REGISTER	FIELD	VALUE
DSS	DISPC_CFG	FIFO_MERGE	0x1

L3 Interconnect A9 Bandwidth Limit

The L3 interconnect can be configured to place a bandwidth limit on the A9 core in order to prevent high-load tasks or sudden spikes in processors activity from consuming too much memory bandwidth which could result in a DSS FIFO underflow. There are two registers within the L3Fast interconnect register space that must be configured to set the A9 bandwidth limit: the NOC_200F_BWLIMITER_MODENA_INIT0_BANDWIDTH_FRACTIONAL and NOC_200F_BWLIMITER_MODENA_INIT0_BANDWIDTH _INTEGER registers. The configuration values required for a given bandwidth limit can be calculated using the following equation where x is the desired maximum A9 bandwidth on the L3 interconnect and y is resulting bandwidth factor.

$$x/6.25 = y$$

The least significant 5 bits of the bandwidth factor (y), [4:0], should be programmed into the BANDWIDTH_FRACTIONAL field of the NOC_200F_BWLIMITER_MODENA_INIT0_BANDWIDTH_FRACTIONAL register and the remaining most significant bits should be programmed into BANDWIDTH_INTEGER field of the NOC_200F_BWLIMITER _MODENA_INIT0_BANDWIDTH_INTEGER register.

For example, if y = 0x30

BANDWIDTH_FRACTIONAL = 0x10

BANDWIDTH_INTEGER = 0x1

Figure 3-1. NOC_200F_BWLIMITER_MODENA_INIT0_BANDWIDTH_FRACTIONAL Register (Address Offset: 0x00 5208)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED											BANDWIDTH_FRACTIONAL				
											R/W-0h				

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

Table 3-1. NOC_200F_BWLIMITER_MODENA_INIT0_BANDWIDTH_FRACTIONAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED			
4-0	BANDWIDTH_FRACTIONAL	R/W	0h	Fractional part of bandwidth in terms of bytes per second

Figure 3-2. NOC_200F_BWLIMITER_MODENA_INIT0_BANDWIDTH_INTEGER Register (Address Offset: 0x00 520C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
0h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												BANDWIDTH_INTEGER			
0h												R/W-1h			

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

Table 3-2. NOC_200F_BWLIMITER_MODENA_INIT0_BANDWIDTH_INTEGER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED		0h	
3-0	BANDWIDTH_INTEGER	R/W	1h	Integer part of bandwidth in terms of bytes per second

By default, the L3 interconnect flow control mechanism allows for the peak bandwidth to temporarily exceed the average bandwidth limit configured in the previous step through to use of the Watermark feature. The Watermark feature defines the amount of peak bandwidth, in bytes, to allocate for temporary use in high load scenarios. These temporary peaks of bandwidth usage could cause the DSS FIFO to underflow. As a result, the user must disable this feature by writing 0x0 to the NOC_200F_BWLIMITER _MODENA_INIT0_WATERMARK_0 register.

Figure 3-3. NOC_200F_BWLIMITER_MODENA_INIT0_WATERMARK_0 Register (Address Offset: 0x00 5210)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					WATERMARK_0										
					R/W-7FFh										

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

Table 3-3. NOC_200F_BWLIMITER_MODENA_INIT0_WATERMARK_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED		0h	
10-0	WATERMARK_0	R/W	7FFh	Peak bandwidth allowed. Program to value + 1. 0x0 disables the feature

Advisory 15 *ROM: NAND Boot Mode is Unsupported*

Revisions Affected	1.1
Details	Boot modes affected: NAND, NAND_I2C The ROM does not send the proper commands to determine if the boot blocks are valid, thus NAND boot (NAND, NAND_I2C) is unreliable.
Workaround	None

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

Revisions Affected 1.1, 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 1.1, 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 19 — *TSC_ADC: False Pen-up Interrupts*

Revisions Affected 1.1, 1.2

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 3-4 illustrates the effect on the analog voltage when constant pen pressure is applied to the touchscreen and held through the Charge step.

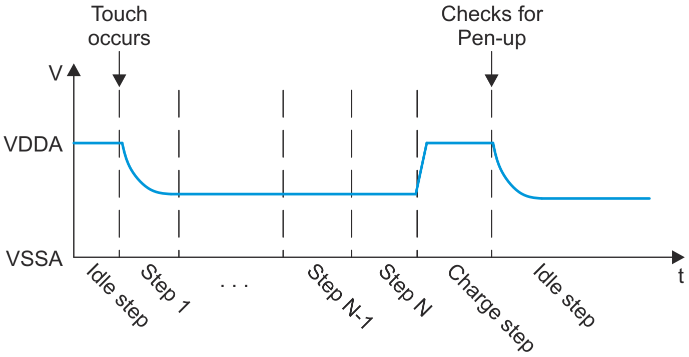

Figure 3-4. TSC Generates False Pen-up Interrupt

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 3-5.

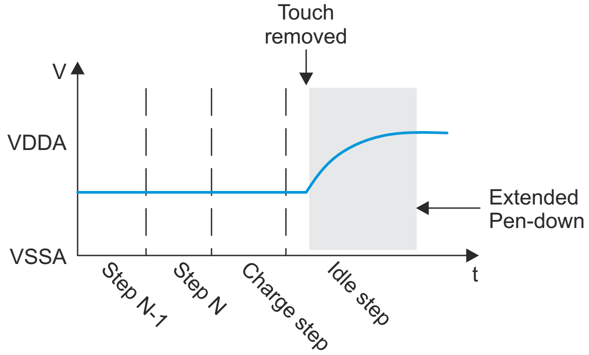

Figure 3-5. Pen-down Remains Active in Idle Step

- 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 20 — *GPTimer: Delay Needed to Read Some GPTimer Registers After Wakeup*

Revisions Affected 1.1, 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 1.1, 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 1.1, 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 1.1, 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 25 — *Ethernet Boot: ROM May Select 1-Gbit, Half-Duplex Mode During Auto-negotiation and Fail to Boot*

Revisions Affected 1.1, 1.2

Details The Ethernet controller on the device does not support 1-Gbit, half-duplex mode. If a connected Ethernet PHY is boot strapped to enable auto-negotiation at 1-Gbit, half-duplex, the ROM may choose this mode inadvertently and never sends out a BOOTP packet to initiate Ethernet boot. Ethernet boot fails in this case.

Workaround Ensure that a connected Ethernet PHY precludes the advertisement of 1-Gbit, half-duplex mode. This can typically be done with proper boot strapping options on the Ethernet PHY.

Advisory 26 — *AutoCMD12 Mode: CMD12 Command is Not Issued on Write Transfer Completion*

Revisions Affected 1.1, 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 1.1, 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 1.1, 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 1.1, 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 1.1, 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 1.1, 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 1.1, 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 1.1, 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.

4 Revision History

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

Changes from May 31, 2017 to June 30, 2021 (from Revision C (June 2017) to Revision D (June 2021)) **Page**

- Added i2223 - ROM: Non-muxed Fast NOR boot does not onfigure A26 and A27... 5
- Added i2224 - DCAN: RAMINIT_DONE intermittently fails to latch completion.. 5
- Added i912 - QSPI: QSPI register bitfield incorrectly masked when read.. 5
- Added i2225 - Possible underflow condition when using EDMA with UART1, UART2, and UART3.................. 5
- Added i2226 - PRU-ICSS: Burst data transfer between ICSS instances not supported.................................... 5

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to TI's Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

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