Errata

AM62Px Sitara™ Processors Silicon Errata, Silicon Revision 1.0, 1.1, 1.2

TEXAS INSTRUMENTS

ABSTRACT

This document describes the known exceptions to the functional specifications (advisories). This document may also contain 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.

Table of Contents

Copyright © 2025 Texas Instruments Incorporated

1 Usage Notes and Advisories Matrices

Table 1-1 lists all usage notes and the applicable silicon revision(s). Table 1-2 lists all advisories, modules affected, and the applicable silicon revision(s).

Table 1-1. Usage Notes Matrix

MODULE	DESCRIPTION	SILICON REVISIONS AFFECTED		
		AM62Px 1.0, AM62Px-Q1 1.0	AM62Px 1.1, AM62Px-Q1 1.1, AM62Px 1.2	AM62Px-Q1 1.2
OSPI	i2351 — OSPI: Controller does not support Continuous Read mode with NAND Flash	YES	YES	YES
PLL	i2424 — PLL: PLL Programming Sequence May Introduce PLL Instability	YES	YES	YES
DDR	i2330 — DDRSS Register Configuration Tool Updates	YES	YES	YES
Package	i2496 — Package Pin Assignment Difference between SR1.0/SR1.1 and SR1.2	YES	YES	YES

Table 1-2. Advisories Matrix

MODULE	DESCRIPTION	SILICON REVISIONS AFFECTED		
		AM62Px 1.0, AM62Px-Q1 1.0	AM62Px 1.1, AM62Px-Q1 1.1, AM62Px 1.2	AM62Px-Q1 1.2
BCDMA	i2431 — BCDMA: RX Channel can lockup in certain scenarios	YES	YES	YES
BCDMA	i2436 — BCDMA: BCDMA RX_IGNORE_LONG setting in RX CHAN CFG register doesn't work	YES	YES	YES
Boot	i2410 — Boot: ROM may fail to boot due to i2409	YES	NO	NO
Boot	i2419 — Boot: When disabling deskew calibration, ROM does not check if deskew calibration was enabled	YES	NO	NO
Boot	i2423 — Boot: HS-FS ROM applies debug access restrictions to all address space covered by the efuse controller firewall	YES	NO	NO
Boot	i2435 — Boot: ROM timeout for eMMC boot too long	YES	YES	YES
Boot	i2457 — Boot: Missing data in UART transfer causes boot failures	YES	NO[1]	NO[1]
Boot	i2464 — Boot: ROM is unable to boot from SD cards with incorrect formatting	YES	YES	YES
Boot	i2482 — Boot: ROM does not provide enough clocks during SD card initialization	YES	YES	YES
CPSW	i2208 — CPSW: ALE IET Express Packet Drops	YES	YES	YES
CPSW	i2401 — CPSW: Host Timestamps Cause CPSW Port to Lock up	YES	YES	YES
CSI_RX	i2190 — CSI_RX_IF may enter unknown state following an incomplete frame	YES	YES	YES
DDR	i2160 — DDR: Valid VRef range must be defined during LPDDR4 Command Bus Training	YES	YES	YES
DSS	i2097 — DSS: Disabling a layer connected to Overlay may result in synclost during the next frame	YES	YES	YES
Internal Diagnostic Modules	i2103 — Incorrect Reporting of ECC_GRP, ECC_BIT and ECC_TYPE Information for Functional Safety Errors	YES	YES	YES
Interrupt Aggregator	i2196 — IA: Potential deadlock scenarios in IA	YES	YES	YES
LPM	i2487 — LPM: Low power modes may inadvertently corrupt DDR contents	YES	YES	YES
MCAN	i2278 — MCAN: Message Transmit order is not ensured from dedicated Tx Buffers configured with same Message ID	YES	YES	YES
MCAN	i2279 — MCAN: Specification Update for dedicated Tx Buffers and Tx Queues configured with same Message ID	YES	YES	YES
MMCHS	i2312 — MMCHS HS200 and SDR104 Command Timeout Window Too Small	YES	YES	YES
MMCHS	i2458 — MMCHS: eMMC HS400 tDCD timing marginal to JEDEC spec	YES	YES	NO
OSPI	i2189 — OSPI: Controller PHY Tuning Algorithm	YES	YES	YES
OSPI	i2249 — OSPI: Failing OSPI DDR PHY Internal Pad Loopback and No Loopback timing modes	YES	YES	YES
OSPI	i2383 — OSPI: 2-byte address is not supported in PHY DDR mode	YES	YES	YES

Table 1-2. Advisories Matrix (continued)

MODULE	DESCRIPTION	SILICON REVISIONS AFFECTED		
		AM62Px 1.0, AM62Px-Q1 1.0	AM62Px 1.1, AM62Px-Q1 1.1, AM62Px 1.2	AM62Px-Q1 1.2
PRG	i2253 —PRG: CTRL_MMR STAT registers are unreliable indicators of POK threshold failure	YES	YES	YES
PSIL	i2137 — PSIL: Clock stop operation can result in undefined behavior	YES	YES	YES
RAT	i2062 — RAT: Error Interrupt Triggered Even When Error Logging Disable Is Set	YES	YES	YES
RESET	i2407— RESET: MCU_RESETSTATz unreliable when MCU_RESETz is asserted low	YES	YES	YES
USART	i2310 — USART: Erroneous triggering of timeout interrupt	YES	YES	YES
USART	i2311 — USART Spurious DMA Interrupts	YES	YES	YES
USB	i2409 — USB: USB2 PHY locks up due to short suspend	YES	YES	YES

(1)   See Advisory for important information

1.1 Devices Supported

This document supports the following devices:

• AM62Px
• AM62Px-Q1

Reference documents for the supported devices are:

• AM62Px Processors Technical Reference Manual (SPRUJ83)
• AM62Px Processors Data Sheet (SPRSP89)

2 Silicon Usage Notes and Advisories

This section lists the usage notes and advisories for this silicon revision.

2.1 Silicon Usage Notes

i2351	***OSPI: Direct Access Controller (DAC) does not support Continuous Read mode with NAND Flash***

Details:

The OSPI Direct Access Controller (DAC) doesn't support Continuous Read mode with NAND Flash since the OSPI controller can deassert the CSn signal (by design intent) to the Flash memory between internal DMA bus requests to the OSPI controller.

The issue occurs because "Continuous Read" mode offered by some OSPI/QSPI NAND Flash memories requires the Chip Select input to remain asserted for an entire burst transaction.

The SoC internal DMA controllers and other initiators are limited to 1023 B or smaller transactions, and arbitration/queuing can happen both inside of the various DMA controllers or in the interconnect between any DMA controller and the OSPI peripheral. This results in delays in bus requests to the OSPI controller that result in the external CSn signal being deasserted.

NOR Flash memories are not affected by CSn de-assertion and Continuous Read mode works as expected.

Workaround(s):

Software can use page/buffered read modes to access NAND flash.

i2424	***PLL: PLL Programming Sequence May Introduce PLL Instability***

Details:

PLL programming sequence has been changed to ensure that, if used, all calibration fields are configured prior to enabling the PLL calibration. In addition to the change to the control of the calibration logic, other changes are implemented so that PLL parameters are unchanged while the PLL is enabled.

When in integer mode, the software enables the PLL calibration feature on calibration-capable PLLs. The previous software adjusted calibration modes after CAL_LOCK was asserted. These writes have been observed to cause a loss of PLL lock on some devices. Additionally, even on susceptible devices, the loss of lock is intermittent, but when the loss occurs, dependent circuitry runs at an incorrect frequency; this wrong frequency can show up as slow algorithm execution or communication failures.

Limit on the impact: The calibration logic cannot be used when the PLL is in fractional mode. Therefore, PLLs that are programmed to use fractional mode should not see a failure related to the calibration programmation. Nevertheless, because of the change to the full PLL sequence, the new software is recommended for all users.

Workaround(s):

Do not use clk_pll_16fft_cal_option4() in SYSFW. Ensure to use updated PLL programming sequences in SDK v10.0 or later when performing any PLL configuration change.

i2330	***DDRSS Register Configuration Tool Updates***

Details:

The DDR Register Configuration Tool provides custom register settings based on system level details such as the architecture (density, data width, ranks) of the DDR device,

i2330 (continued)   ***DDRSS Register Configuration Tool Updates***

frequency of operation, and IO settings determined through board simulations. This tool may be updated over time to support new devices and/or features, fix issues identified with the tool, and most importantly, capture work-arounds of errata or recent updates identified to register calculations which improve performance, signal integrity, or timing relationships between signals.

Workaround(s):

In order to ensure that parameters are set appropriately based on lessons learned and reduce the risk of functional failure, the latest DDR register configuration tool should always be used to generate register values. As the DDR register configuration tool can periodically be updated, the revision history of the tool should be reviewed and evaluated whether tool changes apply to existing systems. When applicable, the configuration of an existing system should be updated appropriately. The latest version of the tool can be found at http://dev.ti.com/sysconfig, and choosing DDR Configuration under Software Product drop down for the applicable device that is being used.

i2496   ***Package Pin Assignment Difference between SR1.0/SR1.1 and SR1.2***

Details:

Two AMH package terminals previously assigned to VDD_MMC0 and VDDA_0P85_DLL_MMC0 power rails have been assigned to a new signal function, VDDR_CORE, and the connectivity was merged with the VDDR_CORE power rail inside of the package as the device transitions from Device Revision A/B (SR1.0/SR1.1) to Device Revision C (SR1.2) and newer devices.

- Terminal 1J1
  - SR1.0/SR1.1: Assigned to VDDA_0P85_DLL_MMC0
  - SR1.2 and newer: Assigned to VDDR_CORE
- Terminal 1K3
  - SR1.0/SR1.1: Assigned to VDD_MMC0
  - SR1.2 and newer: Assigned to VDDR_CORE

Note

For new designs using SR1.2 or when installing SR1.2 devices on boards designed for SR1.0/SR1.1, terminals 1J1 and 1K3 must be supplied by the same power source as VDDR_CORE.

Workaround(s):

None

2.2 Silicon Advisories

i2062

RAT: Error Interrupt Triggered Even When Error Logging Disable Is Set

Details:

If the RAT error logging is programmed to disable logging and enable interrupts, then an error will incorrectly trigger an interrupt but the error log registers will correctly not be updated. The error interrupt should not have been generated.
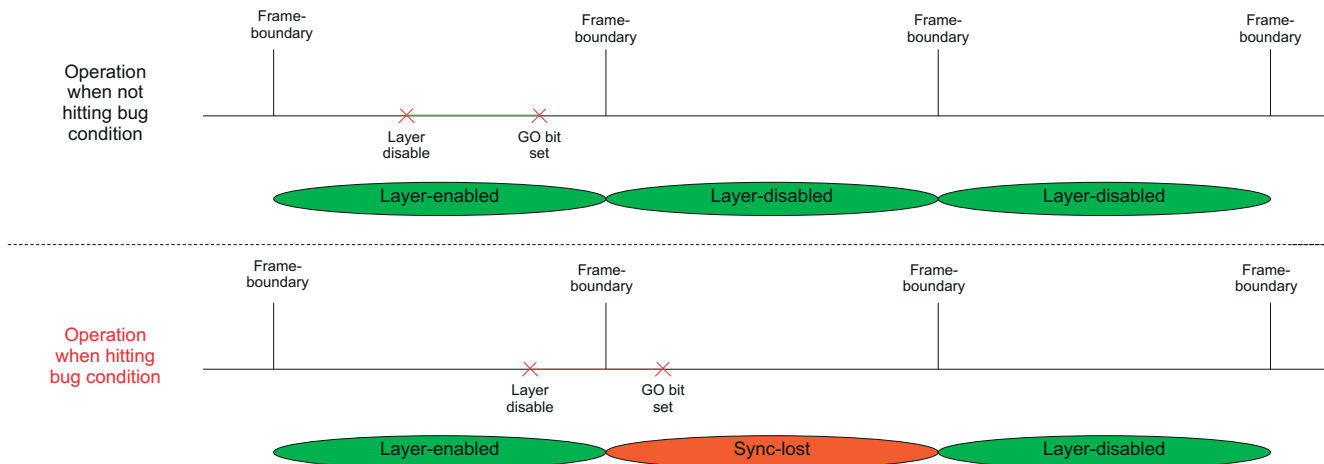
Workaround(s):

If the RAT error logging is disabled, then the error interrupt should also be disabled by software.

i2097

DSS: Disabling a Layer Connected to Overlay May Result in Synclost During the Next Frame

Details:

Disabling a layer (for example VID1) connected to an OVR (that is toggling DSS_VID_ATTRIBUTESx[0] ENABLE from 1 to 0) may result in synclost during the next frame. The synclost may result in a corrupted or blank frame (all pixel data sent out of DSS during the frame is 0x0). The occurrence of synclost is dependent on the timing of setting the GO bit (that is DSS_VP_CONTROL[5] GOBIT to 1) vis-à-vis the disabling of the layer. If the "disable layer" MMR write operation and "set GO bit" MMR write operation happens within the same frame boundary, no synclost occurs. If the operations happen across the frame boundary, then synclost occurs (for one frame). The design automatically recovers and returns to normal operation from the next frame after GO bit is set, see Figure 2-1.

Figure 2-1. Bug Condition

Workaround(s):

A simple software workaround exists. In the workaround, prior to disabling a layer on the OVR, it is moved to the "non-visible" area of the OVR (for example: DSS_OVR_ATTRIBUTES_x[17-6] POSX = max_value_of_posx or DSS_OVR_ATTRIBUTES_x[30-19] POSY = max_value_of_posy). This avoids the synclost when the layer is disabled.

A sample software workaround pseudo-code is shown on Figure 2-2. In this case, the regular "disable layer" MMR write operation and "set GO bit set" MMR write operation are replaced with macros which implement the software workaround.

6   *AM62Px Sitara™ Processors Silicon Errata, Silicon Revision 1.0, 1.1, 1.2*   SPRZ574B – DECEMBER 2023 – REVISED OCTOBER 2025

Submit Document Feedback

i2097 (continued) ***DSS: Disabling a Layer Connected to Overlay May Result in Synclost During the Next Frame***

```
macro disable_layer (overlay n , layer m)
set OVR[n].ATTRIBUTES2[m].POSX = posx_max;
set OVR[n].ATTRIBUTES2[m].POSY = posy_max;
global_ovr_layer_disable_tracker[n][m] = 1;
endmacro

macro set_go_bit (vp n)
if(|(global_ovr_layer_disable_tracker[n])//any bit set
{
   set VP[n].CONTROL.GOBIT  = 1;
   Wait for 10 DSS FUNC CLK cycles;
   for (i=0;i<NUM_LAYERS;i++)
   {
     if(global_ovr_layer_disable_tracker[n][i])
     {
       Clear OVR[n].ATTRIBUTES[i].ENABLE = 0;
       global_ovr_layer_disable_tracker[n][i] = 0;
     }
   }
}
set VP[n].CONTROL.GOBIT  = 1;
endmacro
```

- Replace layer disable MMR write operation with a macro which positions the layer to the non-visible area of the OVR

- Track which layers are disabled. This will be used while GO bit is set

- Replace GO bit set MMR write operation with this macro
- First, set GO Bit for the changes in "disable_layer" macro (and any other earlier changes) to take effect
- After the first GO bit set, few idle_cycles (10 DSS functional clock cycles) are necessary before we move to the second step

- In the second step, actually disable the layers based on the previously tracked information

- Set the GO bit for the second time for the disable of the layers to take effect

Figure 2-2. Workaround Pseudo-code

---

i2103 ***Safety Modules: Incorrect Reporting of ECC_GRP, ECC_BIT and ECC_TYPE Information for Functional Safety Errors***

Details:

For functional safety errors, the logged information - ECC_GRP, ECC_BIT, and ECC_TYPE in the Error Status Registers may be incorrect for certain safety checkers. This only applies to safety checkers that map to ECC_GRP = 0,15,31,47,63...(N*16-1). In the case for the DDR Bridge/Controller, the issue only applies to the safety checkers where ECC_GRP = 0,31,63...(N*32-1).

This issue affects all Safety Module instances and their sub-banks. Refer to section Safety Modules of the device TRM.

Note: The detection and interrupt signaling of these safety errors is unaffected. Only the logging of the aforementioned fields of the Error Status Registers are affected.

Workaround(s):

None. For these specific safety checkers, software is limited to knowing whether a correctable or uncorrectable error occurred and which Safety Module instance had the error (thus knowing the IP module), but not which exact safety checker encountered the error.

---

i2137 ***PSIL: Clock stop operation can result in undefined behavior***

Details:

The clock stop interface is a request/acknowledge interface used to coordinate the handshaking of properly stopping the main clock to the module. Attempting a clock stop on the module without first performing the channel teardowns or clearing of global enable bits will result in module-specific behavior that may be undefined.

The impacted modules are PDMA, SA2UL, Ethernet SW, CSI, UDMAP, ICSS, and CAL.

Workaround(s):

Before attempting to perform a clock stop operation, software is required to teardown all active channels (via UDMAP "real time" registers in the UDMAP, or PSIL register 0x408 in PSIL based modules), and after this is complete, also clear the global enable bit for all channels (via PSIL register 0x2 in both the UDMAP and PSIL based modules).

SPRZ574B – DECEMBER 2023 – REVISED OCTOBER 2025   *AM62Px Sitara™ Processors Silicon Errata, Silicon Revision 1.0, 1.1, 1.2*   7
Submit Document Feedback

Copyright © 2025 Texas Instruments Incorporated

i2160	**DDR: Valid VRef Range Must be Defined During LPDDR4 Command Bus Training**

Details:

The DDR PHY updates VREF(ca) for the command/address bus during LPDDR4 Command Bus Training (CBT). If VREF(ca) search range is set to invalid values such as no working settings can be found during CBT, the training process could fail or hang.

Workaround(s):

Set the following fields to known valid working values before enabling CBT.

For frequency set 0: PI_CALVL_VREF_INITIAL_START_POINT_F0 and PI_CALVL_VREF_INITIAL_STOP_POINT_F0

For frequency set 1: PI_CALVL_VREF_INITIAL_START_POINT_F1 and PI_CALVL_VREF_INITIAL_STOP_POINT_F1

For frequency set 2: PI_CALVL_VREF_INITIAL_START_POINT_F2 and PI_CALVL_VREF_INITIAL_STOP_POINT_F2

Recommendation is to use the nominal VRef value (based on the device programming of drive strength on the processor and termination in the memory) +/- 4%. Please use the online DDR Register Configuration Tool at http://dev.ti.com/sysconfig to program these registers and check the Revision History to ensure this workaround has been addressed in the version of the tool being used.

i2189	**OSPI: Controller PHY Tuning Algorithm**

Details:

The OSPI controller uses a DQS signal to sample data when the PHY Module is enabled. However, there is an issue in the module which requires that this sample must occur within a window defined by the internal clock. Read operations are subject to external delays, which change with temperature. To ensure valid reads at any temperature, a special tuning algorithm must be implemented which selects the most robust TX, RX, and Read Delay values.

Workaround(s):

The workaround for this bug is described in detail in SPRACT2. To sample data under some PVT conditions, users are required to increment the Read Delay field to shift the internal clock sampling window. This allows sampling of the data anywhere within the data eye. However, this has these side effects:

1. PHY Pipeline mode must be enabled for all read operations. Because PHY Pipeline mode must be disabled for writes, reads and writes must be handled separately.
2. Hardware polling of the busy bit is broken when the workaround is in place, so SW polling must be used instead. Writes must occur through DMA accesses, within page boundaries, to prevent interruption from either the host or the flash device. Software must poll the busy bit between page writes. Alternatively, writes can be performed in non-PHY mode with hardware polling enabled.
3. STIG reads must be padded with extra bytes, and the received data must be right-shifted.

i2190	**CSI: CSI_RX_IF may enter unknown state following an incomplete frame**

Details:

i2190 (continued)   ***CSI: CSI_RX_IF may enter unknown state following an incomplete frame***

When an incomplete frame with potential CRC error is received by the CSI2 interface, the module may enter an unknown state. In which case all the subsequent image frames will not be captured.

Workaround(s):

Reset the CSI_RX_IF module.

i2196   ***IA: Potential deadlock scenarios in IA***
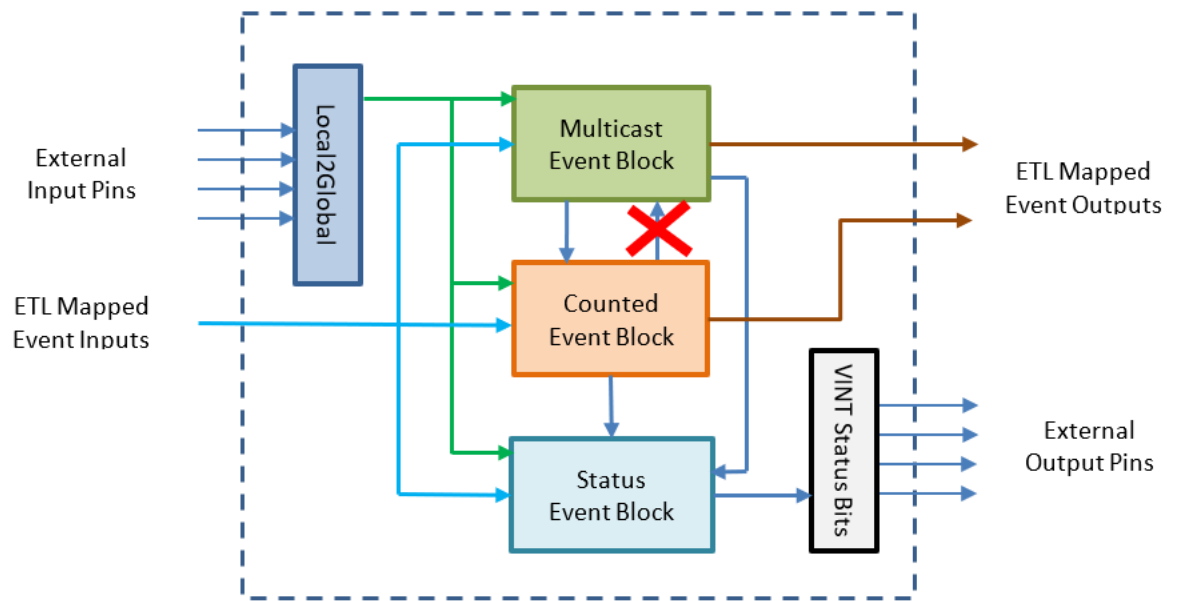
Details:

The interrupt Aggregator (IA) has one main function, which is to convert events arriving on the Event Transport Lane (ETL) bus, can convert them to interrupt status bits which are used to generate level interrupts. The block that performed this function in IA version 1.0 was called the status event block.

In addition to the status event block, there are two other main processing blocks; the multicast event block, and the counted event block. The multicast block really functions as an event splitter. For every event it takes in, it can generate two output events. The counted event block is used to convert high frequency events into a readable count. It counts input events and generates output events on count transitions to/from 0 to/from non-zero count values. Unlike the status event block, the multicast and counted event blocks generate output ETL events that are then mapped to other processing blocks.

An issue was found after design that could cause the IA to deadlock. The issue occurs when event "loops" occur between these three processing blocks. It is possible to create a situation where a processing block can not output an event because the path is blocked, and since it can not output an event, it can not take any new input events. This inability to take input events prevents the output path from being able to unwind, and thus both paths remain blocked.

Workaround(s):

Figure 2-3 shows the conceptual block diagram of IA 1.0. Potential loops are avoided by adopting the policy of not allowing the counted event block to send events to the multicast block. This method was chosen because it is more common to split an event first, and then count one while sending the other elsewhere. With this path blocked by convention, it is not possible for a single event to visit any block more than once and thus not possible for paths to become blocked so long as the outputs remain unblocked.

i2196 (continued)   ***IA: Potential deadlock scenarios in IA***

Figure 2-3. Interrupt Aggregator Version 1.0

By following the conventions outlined here, the system is safe from looping hazards that can create a deadlock scenario.

i2208   ***CPSW: ALE IET Express Packet Drops***

Details:

This issue impacts the following Module:

The issue with ALE is due to CPSW frequency and IET operation with short express traffic and pre-empted packets that get pre-empted between 60-69 bytes on non-10G capable ports.

If an IET pre-emptible packet get interrupted at 60-69 bytes, the lookup will occur when the next chunk arrives. The CPSW only gives the ALE 64 bytes from the pre-emptible MAC.

As a result, a short express traffic lookup will start at the end of a 64 byte express traffic, but when the pre-empted queue continues, the pre-empted traffic will complete the 64 bytes and attempt a lookup for the pre-empt packet. But this lookup is less that 64 clocks from the express lookup start, so the express lookup will be aborted(express traffic dropped) and start the new lookup for the pre-empted traffic.

Rules to induce the issue:
1. You are in IET (Interspersed Express Traffic) mode on ports not capable of 5/10G operation
2. Remote express packets can be preempt packets as low as 60 bytes
3. Pre-empt packet traffic that is 128 bytes or more.
4. Express traffic that interrupts the pre-empt traffic between 60-69 bytes.
5. A short express traffic immediately followed by the continuation of the pre-empt traffic.
    a. Gap between express frame and pre-empt frame be its minimum.

6. The CPSW frequency is at its lowest capability for the speeds required.

i2208 (continued)   ***CPSW: ALE IET Express Packet Drops***

Workaround(s):   During IET negotiation, tell the remote to fragment at 128 bytes.

i2249   ***OSPI: Internal PHY Loopback and Internal Pad Loopback clocking modes with DDR timing inoperable***

Details   The OSPI Internal PHY Loopback mode and Internal Pad Loopback mode uses "launch edge as capture edge" (same edge capture, or 0-cycle timing).

The programmable receive delay line (Rx PDL) is used to compensate for the round trip delay (Tx clock to Flash device, Flash clock to output and Flash data to Controller).

In the case of internal and IO loopback modes, the total delay of the Rx PDL is not sufficient to compensate for the round trip delay, and thus these modes cannot be used.

The table below describes the recommended clocking topologies in the OSPI controller. All other modes not described here are affected by the advisory in DDR mode and are not recommended clocking topologies.

Table 2-1. OSPI Clocking Topologies

Clocking Mode Terminology	CONFIG_REG.PHY_MODE_ENABLE	READ_DATA_CAPTURE.BYPASS	READ_DATA_CAPTURE.DQS_EN	Board implementation
No Loopback, no PHY	0 (PHY disabled)	1 (disable adapted loopback clock)	X	None. Relying on internal clock. Max freq 50MHz.
External Board Loopback with PHY	1 (PHY enabled)	0 (enable adapted loopback clock)	0 (DQS disabled)	External Board Loopback (OSPI_LOOPBACK_CLK_SEL = 0)
DQS with PHY	1 (PHY enabled)	X (DQS enable has priority)	1 (DQS enabled)	Memory strobe connected to SOC DQS pin

Workaround   None. Please use one of the unaffected clocking modes based on the table in the description

i2253   ***PRG: CTRL_MMR STAT registers are unreliable indicators of POK threshold failure***

Details   The POK overvoltage and undervoltage flags in the CTRL_MMR PRG STAT registers are unreliable indicators of whether the POK has seen a failure. As a result, they are being marked as Reserved in the device Technical Reference Manual (TRM).

Workaround   The filtered POK output updates ESM flags.

Upon POK initialization (i.e. enable), the ESM flags should be cleared (due to comparisons carried out during the bandgap and / or the POK settling time). After this initial clear, the ESM flags can be used as a reliable indicator of failure (or no failure) from the POKs.

i2278	***MCAN: Message Transmit order is not ensured from dedicated Tx Buffers configured with same Message ID***

Details

The erratum is limited to the case when multiple Tx Buffers are configured with the same Message ID (TXBC.NDTB > 1).

Under the following conditions, a message may be transmitted out of order:
- Multiple Tx Buffers configured with the same Message ID
- Tx requests for these Tx Buffers are submitted sequentially with delays between each

Workaround

Workaround #1:

After writing the Tx messages with same Message ID to the Message RAM, request transmission of all these message concurrently by single write access to TXBAR. Make sure none of these messages have a pending Tx request before making the concurrent request.

Workaround #2:

Use the Tx FIFO instead of dedicated Tx Buffers (set bit MCAN_TXBC[30] TFQM = 0 to use Tx FIFO) for the transmission of several messages with the same Message ID in a specific order.

i2279	***MCAN: Specification Update for dedicated Tx Buffers and Tx Queues configured with same Message ID***

Details

The erratum updates the descriptions in Section 3.5.2 Dedicated Tx Buffers and 3.5.4 Tx Queue of the M_CAN User's Manual related to message transmission from multiple dedicated Tx Buffers configured with the same Message ID.

Workaround

Workaround #1:

After writing the Tx messages with same Message ID to the Message RAM, request transmission of all these message concurrently by single write access to TXBAR. Make sure none of these messages have a pending Tx request before making the concurrent request.

Workaround #2:

Use the Tx FIFO instead of dedicated Tx Buffers (set bit MCAN_TXBC[30] TFQM = 0 to use Tx FIFO) for the transmission of several messages with the same Message ID in a specific order.

i2310	***USART: Erroneous clear/trigger of timeout interrupt***

Details:

The USART may erroneously clear or trigger the timeout interrupt when RHR/MSR/LSR registers are read.

Workaround(s):

For CPU use-case.
- If the timeout interrupt is erroneously cleared:
  – This is Valid since the pending data inside the FIFO retriggers the timeout interrupt

12   *AM62Px Sitara™ Processors Silicon Errata, Silicon Revision 1.0, 1.1, 1.2*   SPRZ574B – DECEMBER 2023 – REVISED OCTOBER 2025

Submit Document Feedback

i2310 (continued) **USART: Erroneous clear/trigger of timeout interrupt**

- If timeout interrupt is erroneously set, and the FIFO is empty, use the following SW workaround to clear the interrupt:
  – Set a high value of timeout counter in TIMEOUTH and TIMEOUTL registers
  – Set EFR2 bit 6 to 1 to change timeout mode to periodic
  – Read the IIR register to clear the interrupt
  – Set EFR2 bit 6 back to 0 to change timeout mode back to the original mode

For DMA use-case.

- If timeout interrupt is erroneously cleared:
  – This is valid since the next periodic event retriggers the timeout interrupt
  – User must ensure that RX timeout behavior is in periodic mode by setting EFR2 bit6 to 1
- If timeout interrupt is erroneously set:
  – This causes DMA to be torn down by the SW driver
  – Valid since next incoming data causes SW to setup DMA again

i2311 **USART Spurious DMA Interrupts**

Details:

Spurious DMA interrupts may occur when DMA is used to access TX/RX FIFO with a non-power-of-2 trigger level in the TLR register.

Workaround(s):

Use power of 2 values for TX/RX FIFO trigger levels (1, 2, 4, 8, 16, and 32).

i2312 **MMCSD: HS200 and SDR104 Command Timeout Window Too Small**

Details:

Under high speed HS200 and SDR104 modes, the functional clock for MMC modules will reach up to 192 MHz. At this frequency, the maximum obtainable timeout through of MMC host controller using MMCSD_SYSCTL[19:16] DTO = 0xE is (1/192MHz)*2^27 = 700ms. Commands taking longer than 700ms may be affected by this small window frame.

Workaround(s):

If the command requires a timeout longer than 700ms, then the MMC host controller command timeout can be disabled (MMCSD_CON[6] MIT=0x1) and a software implementation may be used in its place. Detailed steps as follows (in Linux):

1. During MMC host controller probe function (omap_hsmmc.c:omap_hsmmc_probe()), inform processor that the host controller is incapable of supporting all the necessary timeouts.

2. Modify the MMC core software layer functionality so the core times out on its own when the underlying MMC host controller is unable to support the required timeout.

| i2383 | ***OSPI: 2-byte address is not supported in PHY DDR mode*** |

Details:

When the OSPI controller is configured for 2-byte addressing in PHY DDR Mode, an internal state machine mis-compares the number of address bytes transmitted to a value of 1 (instead of 2). This results in a state machine lockup in the address phase, rendering PHY DDR mode non-operable.

This issue does not occur when using any Tap mode or PHY SDR mode. This issue also doesn't occur when using 4 byte addressing in PHY DDR mode.

Workaround(s):

For compatible OSPI memories that have programmable address byte settings, set the amount of address bytes required from 2 to 4 on the flash. This may involve sending a specific command to change address bytes and/or writing a configuration register on the flash. Once done, update the amount of address bytes sent in the controller settings from 2 to 4.

For compatible OSPI memories that only support 2-byte addressing and cannot be re-programmed, PHY DDR mode will not be compatible with that memory. Alternative modes include:

- PHY SDR mode
- TAP (no-PHY) DDR mode
- TAP (no-PHY) SDR mode

| i2401 | ***CPSW: Host Timestamps Cause CPSW Port to Lock up*** |

Details:

The CPSW offers two mechanisms for communicating packet ingress timestamp information to the host.

The first mechanism is via the CPTS Event FIFO which records timestamps when triggered by certain events. One such event is the reception of an Ethernet packet with a specified EtherType field. Most commonly this is used to capture ingress timestamps for PTP packets. With this mechanism the host must read the timestamp (from the CPTS FIFO) separately from the packet payload which is delivered via DMA. This mode is supported and is not affected by this errata.

The second mechanism is to enable receive timestamps for all packets, not just PTP packets. With this mechanism the timestamp is delivered alongside the packet payload via DMA. This second mechanism is the subject of this errata.

When the CPTS host timestamp is enabled, every packet to the internal CPSW port FIFO requires a timestamp from the CPTS. When the packet preamble is corrupted due to EMI or any other corruption mechanism a timestamp request may not be sent to the CPTS. In this case the CPTS will not produce the timestamp which causes a lockup condition in the CPSW port FIFO. When the CPTS host timestamp is disabled by clearing the tstamp_en bit in the CPTS_CONTROL register the lockup condition is prevented from occurring.

Workaround(s):

Ethernet to host timestamps must be disabled.

CPTS Event FIFO timestamping can be used instead of CPTS host timestamps.

| i2407 | ***RESET: MCU_RESETSTATz unreliable when MCU_RESETz is asserted low*** |

Details:

MCU_RESETSTATz goes high periodically for a short duration and then low again while MCU_RESETz is still asserted low. This issue is seen only when MCU_RESETz is

i2407 (continued)   ***RESET: MCU_RESETSTATz unreliable when MCU_RESETz is asserted low***

asserted low for greater than 100us. The device remains in reset while MCU_RESETz is low; the advisory only applies to the signal MCU_RESETSTATz.

Workaround(s):

Any one of the following could be used as a workaround for this advisory
- Do not use MCU_RESETz in a functional system. MCU_RESETz can still be used for debug, realizing the errata limitation.
- Limit the maximum low duration of MCU_RESETz to less than 100us.
- Use Main Domain RESETSTATz instead of MCU_RESETSTATz. MCU_RESETz also causes a Main reset, so Main Domain RESETSTATz could be used for device reset observation. Consult the datasheet for RESETSTATz timing specifications.
- For new designs, the circuits which produce Main Domain reset and MCU Domain reset should be combined with an AND gate as an input to RESETz. Also connect MCU Domain reset circuit to the MCU_RESETz input. This will provide full functionality of MCU warm reset using MCU_RESETz and MCU_RESETSTATz to indicate status of MCU domain reset. RESETz will be triggered on either a MAIN domain reset or MCU domain reset by using the AND gate.

i2409   ***USB: USB2 PHY locks up due to short suspend***

Details:

The USB 2.0 PHY may hang in response to a USB wake-up event that occurs within 3 microseconds of the USB controller entering suspend. This PHY hang can only be recovered via a power cycle as warm reset is ineffectual.

Workaround(s):

Note: this workaround is only applicable if USB is not the primary boot mode. If USB is the primary boot mode, no workaround is available.

In order to prevent this issue from occurring, a specific order of operations must be observed during the USB controller initialization process:

1. Remove USB controller reset via the LPSC.
2. Set PLL_REG12.pll_ldo_ref_en field (bit 5) in PHY2 region to '1'.
3. Set PLL_REG12.pll_ldo_ref_en_en field (bit 4) in PHY2 region to '1'.
4. Proceed with normal USB controller initialization.

i2410   ***Boot: ROM may fail to boot due to i2409***

Details:

Due to i2409, the ROM may fail to boot in USB boot mode after a warm reset. If the USB 2.0 PHY locks up, the ROM does not implement any of the workarounds listed in i2409, and thus the ROM will hang and fail to boot.

Workaround(s):

The advisory described in i2409 should be avoided by implementing one of the workarounds described in the advisory in software.

i2419	**Boot: When disabling deskew calibration, ROM does not check if deskew calibration was enabled**

Details:

If PLL Deskew calibration is being disabled, the ROM driver code intends to check if deskew calibration is enabled and if lock has failed. However the current code has an assignment in an if condition. As a result, it does not check if deskew calibration is enabled before clearing the config bit. There is no functional issue.

Workaround(s):

None

i2423	**Boot: HS-FS ROM applies debug access restrictions to all address space covered by the efuse controller firewall**

Details:

On HS-FS device ROM applies debug restrictions to FWL 33 and 66 which contains secure assets. The debug access restriction was applied to the entire firewall region and not just to the region which applies to the secure asset. This prevent the use of an external emulator to be able to perform initial flash programming for example without requiring TIFS SW to in the picture.

Workaround(s):

TIFS SW is needed to be able to request the needed firewall to be open.

i2431	**BCDMA: RX Channel can lockup in certain scenarios**

Details:

BCDMA RX chan Teardown can lockup channel and cannot be used for subsequent transfers if none of the TRs have EOP flag set in configuration specific flags field. Subsequently when channel is re-enabled, transfer does not not complete and terminates with various errors in TR response.

Workaround(s):

a) When receiving data from a PSIL/PDMA peripheral, EOP flag needs to be set in the each TR's configuration specific flag field and PDMA's 1 X-Y FIFO Mode Static TR "Z" parameter should be set to non zero value for channel teardown to function properly and cleanup the internal state memory. Otherwise it leads to channel lockups on subsequent runs. The PDMA Z count should also match the TR size, so that PDMA delineates each transfer as an individual packet. This is especially problematic in cases like where TRPD has infinite reload count set to perform cyclic transfer using a single set of TRs in streaming mode, in which case each TR could potentially be the last one.

b) If the usecase doesn't allow for PDMA Z count to be set in advance or packet EOP cannot be set then alternate is to use PKTDMA in single buffer mode instead of BCDMA.

i2435	**Boot: ROM timeout for eMMC boot too long**

Details:

Due to a bug in ROM, if attempting to boot in eMMC boot mode (ie, from eMMC boot partitions, sometimes referred to as eMMC alternative mode) from an eMMC device that is empty or erased (or factory fresh), the normal boot timeout to switch to backup boot mode takes 10 seconds.

i2435 (continued)	***Boot: ROM timeout for eMMC boot too long***
Workaround(s):	
	Need to boot from another boot mode if this timeout considered too long in the system.

i2436	***BCDMA: BCDMA RX_IGNORE_LONG setting in RX CHAN CFG register doesn't work***
Details:	RX_IGNORE_LONG flag in RXCHAN CFG register of BCDMA gets ignored and BCDMA reports errors in TR response when remote endpoints don't send EOP to match TR boundary.
Workaround(s):	RX_IGNORE_LONG is unusable, so remote endpoint such as PDMA should close packet by sending EOP to match TR boundary (PDMA X*Y*Z should match TR ICNT0*ICNT1*ICNT2*ICNT3)
	If infinite stream is desired (PDMA Z=0) then switch to PKTDMA and use Single Buffer Mode

i2457	***Boot: Missing data in UART transfer causes boot failures***
Details:	The ROM fix implemented to workaround a previous errata causes a race condition which leads to missing in-flight data.This increases the chance of causing xmodem NAK. Current xmodem implementation would cancel transfer after receiving more than 10 NAK.This causes image transfer to fail and therefore a boot failure.

> **Note**
> Silicon device revisions which are not affected by this advisory must set BOOTMODE7=0 (when UART is primary boot mode) or BOOTMODE13=0 (when UART is backup boot mode) in order to enable the ROM code that fixes this advisory. See latest version of the device specific TRM for more information on boot mode signals.

Workaround(s):	
	Build a special version of the xmodem transfer tool 'sx' to insert an inter-character delay of 50ms.This will reduce the throughput accordingly.

i2458	***MMCHS: eMMC HS400 tDCD timing marginal to JEDEC spec***
Details:	In some system corner cases, MMC0 interface may marginally exceed the tDCD JEDEC spec requirement of 300ps.
Workaround(s):	Use HS200 instead of HS400 mode

i2464	**Boot: ROM is unable to boot from SD cards with incorrect formatting**

Details:

MMCSD boot mode may be unable to boot from SD cards if the filesystem is formatted incorrectly during the creation of the filesystem.

Workaround(s):

When creating the SD from Ubuntu 22.04 system, add "-a" argument to the mkfs.vfat command to generate correct boot partitioning.

i2482	**Boot: ROM does not provide enough clocks during SD card initialization**

Details:

ROM code is not providing 74 clocks before sending first command as specified in the SD Card Physical Layer Specification ver. 2.00. This may cause SD card boot to fail, however, a failure has never been observed due to this errata on affected devices.

Workaround(s):

None

i2487	**LPM: Low power modes may inadvertently corrupt DDR contents**

Details:

During entry to the Deep Sleep or RTC+IO+DDR low-power modes, the device may not properly transition the attached DDR into retention mode, which will lead to corruption of the DDR data.

Affected devices should not utilize these low power modes without applying the software workaround.

Workaround(s):

A workaround to this issue is provided in SDK versions 11.1 and later. The workaround ensures that an internal retention latch which controls DDR self-refresh is in the correct state through various reset scenarios.

Trademarks

All trademarks are the property of their respective owners.

Revision History

Revision B: Added Silicon Revision 1.2

20   *AM62Px Sitara™ Processors Silicon Errata, Silicon Revision 1.0, 1.1, 1.2*   SPRZ574B – DECEMBER 2023 – REVISED OCTOBER 2025

Submit Document Feedback

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), 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 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, TI's General Quality Guidelines, 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. Unless TI explicitly designates a product as custom or customer-specified, TI products are standard, catalog, general purpose devices.

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