Errata

J7200 DRA821 Processor Silicon Revision 1.0, 2.0

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

1 Modules Affected

Table 1-1 shows the module(s) that are affected by each usage note.

Table 1-1. Usage Note by Modules

MODULE	USAGE NOTE
N/A	

Table 1-2 shows the module(s) that are affected by each advisory.

Table 1-2. Advisories by Modules

MODULE	ADVISORY	SILICON REVISIONS AFFECTED	
		SR 1.0	SR 2.0
ADC	i2151 — ADC: Debounce time control register	YES	YES
Boot	i2307 — Boot: ROM does not properly select OSPI clocking modes based on BOOTMODE	YES	YES
	i2360 — Boot: Ethernet RMII Boot Mode is not supported	YES	YES
	i2361 — Boot: SPI and xSPI BOOTMODE Pin Mapping changes for SR2.0	NO	YES
	i2366 — Boot: ROM does not comprehend specific JEDEC SFDP features for 8D-8D-8D operation	YES	YES
	i2371 — Boot: ROM code may hang in UART boot mode during data transfer	YES	YES
	i2372 — Boot: ROM doesn't support select multi-plane addressing schemes in Serial NAND boot	NO	YES
	i2459 — Boot: PCIe Boot Mode is not supported	YES	YES
CBASS	i2207 — CBASS: Command Arbitration Blocking	YES	YES
	i2235 — CBASS Null Error Interrupt Not Masked By Enable Register	YES	YES
CC	i2221 — CC: Invasive and Non-Invasive debug enable settings are reset by MCU_RESETz	YES	YES
	i2222 — Compute Cluster: A72 Corepac unable to be powered down	YES	YES
CP	i2283 — Restrictions on how CP Tracer Debug Probes can be used	YES	YES
CPSW	i2184 — CPSW: IET express traffic policing issue	YES	YES
	i2185 — CPSW: Policer color marking issue	YES	YES
	i2208 — CPSW: ALE IET Express Packet Drops	YES	YES
DCC	i2209 — DCC: Incorrect clock selection	YES	NO
DDR	i2157 — DDR: Controller anomaly in setting wakeup time for low power states	YES	YES
	i2159 — DDR: VRCG high current mode must be used during LPDDR4 CBT	YES	YES
	i2160 — DDR: Valid VRef Range Must be Defined During LPDDR4 Command Bus Training	YES	YES
	i2166 — DDR: Entry and exit to/from Deep Sleep low-power state can cause PHY internal clock misalignment	YES	YES
	i2182 — DDR: Dual-rank non-power-of-2 density not supported with row-cs-bank-col address mapping	YES	YES
	i2186 — DDR rate limited to 2666 MT/s 1333 MHz clock	YES	NO
	i2232 — DDR: Controller postpones more than allowed refreshes after frequency change	YES	YES
	i2244 — DDR: Valid stop value must be defined for write DQ VREF training	YES	YES
	i2274 — DDR: Including DDR in BSCAN causes current alarm on the DDR supply	YES	NO
DMSC	i2245 — DMSC: Firewall Region requires specific configuration	YES	YES
	i2275 — DMSC Secure Boot ROM: Potential Secure Boot vulnerability with explicit EC curve parameters in X.509 certificate	YES	NO
ECC AGGR	i2049 — ECC AGGR: Potential IP Clockstop/reset sequence hang due to pending ECC Aggregator interrupts	YES	YES
I3C	i2197 — I3C: Slave mode is not supported	YES	YES
	i2205 — I3C Command fetched during pending IBI is not properly processed in some cases	YES	YES
	i2216 — I3C: Command execution may fail during slave-initiated IBI address byte reception	YES	YES
IA	i2196 — IA: Potential deadlock scenarios in IA	YES	YES

Table 1-2. Advisories by Modules (continued)

MODULE	ADVISORY	SILICON REVISIONS AFFECTED	
		SR 1.0	SR 2.0
JTAG	i2228 — JTAG: TAP used by Debuggers may be inaccessible if TRSTn device pin is never asserted	YES	NO
MCAN	i2278 — MCAN: Message Transmit order not guaranteed from dedicated Tx Buffers configured with same Message ID	YES	YES
	i2279 — MCAN: Specification Update for dedicated Tx Buffers and Tx Queues configured with same Message ID	YES	YES
MCU	i2217 — Recommended POST selection via MCU_BOOTMODE[09:08]	YES	NO
MDIO	i2329 — MDIO: MDIO interface corruption (CPSW and PRU-ICSS)	YES	YES
MMCSD	i2312 — MMCSD: HS200 and SDR104 Command Timeout Window Too Small	YES	YES
MSMC	i2116 — MSMC: Set-hazarding logic withholding RT access waiting on NRT access completion	YES	YES
	i2187 — MSMC: Cache Resize to 0 Refreshes Tags instead of Updating them	YES	YES
	i2201 — MSMC: Incorrect Parity Detect on bytecount	YES	NO
OSPI	i2189 — OSPI: Controller PHY Tuning Algorithm	YES	YES
	i2249 — OSPI: Internal PHY Loopback and Internal Pad Loopback clocking modes with DDR timing inoperable	YES	YES
	i2351 — OSPI: Controller does not support Continuous Read mode with NAND Flash	YES	YES
	i2383 — OSPI: 2-byte address is not supported in PHY DDR mode	YES	YES
PCIe	i2183 — PCIe: Link up failure when unused lanes are not assigned to PCIe Controller	YES	YES
	i2237 — PCIe: SerDes Reference Clock Output does not comply to Vcross, Rise-Fall Matching, and Edge Rate limits	YES	NO
	i2241 — PCIe: The SerDes PCIe Reference Clock Output can exceed the 5.0 GT/s Data Rate RMS jitter limit	YES	NO
	i2242 — PCIe: The 4-L SerDes PCIe Reference Clock Output is temporarily disabled while changing Data Rates	YES	YES
	i2243 — PCIe: Timing requirement for disabling output refclk during L1.2 substate is not met	YES	YES
	i2246 — PCIe: Automatic compliance entry fails when unused SERDES lanes are not assigned to PCIe Controller	YES	YES
	i2326 — PCIe: MAIN_PLLx operating in fractional mode, which is required for enabling SSC, is not compliant with PCIe Refclk jitter limits	YES	YES
POK	i2277 — POK: De-Glitch (filter) is based upon only two samples	YES	NO
PRG	i2253 — PRG: CTRL_MMR STAT registers are unreliable indicators of POK threshold failure	YES	YES
PSIL	i2137 — Clock stop operation can result in undefined behavior	YES	YES
R5FSS	i2161 — R5FSS: Debugger cannot access VIM module while it is active	YES	NO
	i2227 — R5FSS: Error interrupt CCM_COMPARE_STAT_PULSE_INTR incorrectly driven	YES	YES
RAT	i2062 — RAT: Error Interrupt Triggered Even When Error Logging Disable Is Set	YES	YES
RINGACC	i2177 — RINGACC: The ring accelerator's debug transaction trace stream can be corrupted by certain ring access sequences	YES	YES
ROM Code	i2306 — ROM Code: Need to turn off internal termination resistors in SERDES	YES	NO
SGMII	i2362 — 10-100M SGMII: Marvell PHY does not ignore the preamble byte resulting in link failure	YES	YES
STOG	i2123 — STOG: Timed Out Emulation Debug write responses from the Slave Gasket always return Success	YES	YES
	i2126 — STOG: Error miscounting when there are two concurrent timeouts or two concurrent unexpected responses	YES	YES
	i2127 — STOG: SRC side write data bus hang when a write command timeout occurs the same cycle as last acceptance on DST side	YES	YES
UDMA	i2146 — UDMA: Force teardown bitfield readback is masked in realtime TX/RX registers	YES	YES
	i2320 — UDMA, UDMAP: Descriptors and TRs required to be returned unfragmented	YES	YES
UDMAP	i2163 — UDMAP: UDMA transfers with ICNTs and/or src/dst addr NOT aligned to 64B fail when used in "event trigger" mode	YES	YES

Table 1-2. Advisories by Modules (continued)

MODULE	ADVISORY	SILICON REVISIONS AFFECTED	
		SR 1.0	SR 2.0
	i2234 — UDMA: TR15 hangs if ICNT0 is less than 64 bytes	YES	YES
USART	i2310 — USART: Erroneous clear/trigger of timeout interrupt	YES	YES
	i2311 — USART: Spurious DMA Interrupts	YES	YES
USB	i2091 — 2.0 PHY hangs if received signal amplitude crosses squelch threshold multiple times within the same packet	YES	YES
	i2134 — USB: 2.0 compliance receive sensitivity test limitation	YES	YES
xSPI	i2257 — xSPI boot mode redundant image boot failure	YES	NO

2 Nomenclature, Package Symbolization, and Revision Identification

2.1 Device and Development-Support Tool Nomenclature

To designate the stages in the product development cycle, TI assigns prefixes to the part numbers of all microprocessors (MPUs) and support tools. Each device has one of three prefixes: X, P, or null (no prefix) (for example, DRA821U4TCBALMRQ1

). 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.

For additional information how to read the complete device name for any DRA821 device, see the specific-device Datasheet (SRPSP57).

2.2 Devices Supported

This document supports the following devices:

- DRA821

Reference documents for the supported devices are:

- Jacinto™ DRA821 Automotive Processors Datasheet (SPRSP57)

2.3 Package Symbolization and Revision Identification

Figure 2-1 shows an example of package symbolization.

Table 2-1 lists the device revision codes.

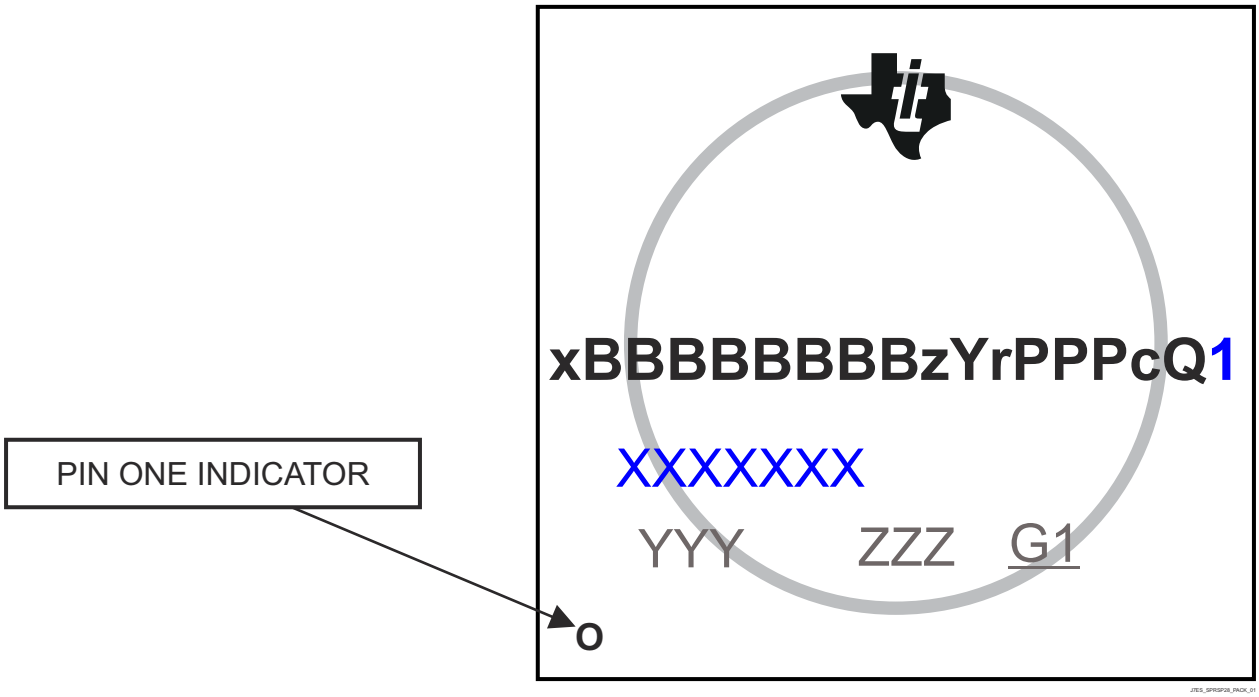

Figure 2-1. Package Symbolization

Table 2-1. Revision Identification

DEVICE REVISION CODE	SILICON REVISION	COMMENTS
A or BLANK	1.0	
B	2.0	

3 Silicon Revision 1.0, 2.0 Usage Notes and Advisories

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

3.1 Silicon Revision 1.0, 2.0 Usage Notes

No known usage notes for this silicon revision.

3.2 Silicon Revision 1.0, 2.0 Advisories

i2049 ***ECC_AGGR: Potential IP Clockstop/Reset Sequence Hang due to Pending ECC Aggregator Interrupts***

Details:

The ECC Aggregator module is used to aggregate safety error occurrences (which are rare) and generate interrupts to notify software. The ECC Aggregator provides software control over the enabling/disabling and clearing of safety errors interrupts.

When software is performing a clockstop/reset sequence on an IP, the sequence can potentially not complete because the IP's associated ECC Aggregator instance is not idle. The ECC Aggregator idle status is dependent upon any pending safety error interrupts either enabled or disabled, which have not been cleared by software. As a result, the IP's clockstop/reset sequence may never complete (hang) if there are any pending safety errors interrupts that remain uncleared.

The affected ECC_AGGRs can be determined by the value listed in the Technical Reference Manual (TRM) for their REV register at Register Offset 0h. The REV register encodes the ECC_AGGR version in its fields as follows:

v[REVMAJ].[REVMIN].[REVRTL]

ECC_AGGR versions before v2.1.1 are affected. ECC_AGGR versions v2.1.1 and later are not affected.

Affected Example:

REVMAJ = 2

REVMIN = 1

REVRTL = 0

The above values decode to ECC_AGGR Version v2.1.0, which is Affected.

Not Affected Example:

REVMAJ = 2

REVMIN = 1

REVRTL = 1

The above values decode ECC_AGGR Version v2.1.1, which is Not Affected.

Workaround(s):

General Note:

Clockstopping the ECC Aggregator is not supported in functional safety use-cases.

Software should use the following workaround for non-functional safety use-cases:

1. Enable all ECC Aggregator interrupts for the IP
2. Service and clear all Pending interrupts
3. Step 3:

i2049 (continued) ***ECC_AGGR: Potential IP Clockstop/Reset Sequence Hang due to Pending ECC Aggregator Interrupts***

a. Disable all interrupt sources to the ECC Aggregator, followed by performing Clockstop/reset sequence.
b. Perform Clockstop/reset sequence, while continuing to service/clear pending interrupts.

Due to interrupts being external stimuli, software has two options for step 3:

1. Disable all interrupt sources (EDC CTRL checkers) that can generate pending ECC_AGGR interrupts prior to performing the clockstop/reset sequence
2. Continue to service/clear pending interrupts that occur while performing the clkstop/reset sequence. The sequence would proceed when all interrupts are cleared.

Software in general may need to detect pending interrupts that continuously fire during this entire sequence (ex. in the case of a stuck-at fault scenario), and disable their associated EDC CTRL safety checkers to allow the clockstop/reset sequence to progress towards completion.

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.

i2091 ***USB: 2.0 PHY Hangs if Received Signal Amplitude Crosses Squelch Threshold Mmultiple Times Within the Same Packet***

Details:

USB 2.0 PHY implements a squelch detection circuit on the receiver to ensure noise is not interpreted as valid data when the bus is idle. The squelch circuit blocks invalid data by disabling the receiver output while the DP/DM differential signal amplitude is less than the squelch threshold.

The PHY may hang if the DP/DM differential signal amplitude drops below the squelch threshold for a brief period of time and increases back above the squelch threshold within the same packet. The issue does not occur if the DP/DM differential signal amplitude crosses the squelch threshold during the idle time between two packets.

Workaround(s):

The issue can be avoided by ensuring the DP/DM differential signal amplitude applied to the receiver input remains above the squelch threshold during valid data transfers.

i2116 ***MSMC: Set-hazarding logic withholding RT access waiting on NRT access completion***

Details:

The DDR controller prioritizes writes over reads to the same page. Additionally, MSMC hazards transactions on the same set regardless of the real-time attribute. Due to these two facts, a stream of writes to the same page followed by a non real-time read to the same page can effectively block out a real-time access command indefinitely.

i2116 (continued) ***MSMC: Set-hazarding logic withholding RT access waiting on NRT access completion***

Example sequence:

1. Stream of Writes to page A sent from MSMC to DDR Controller
2. Non Real-Time Read to page A sent from MSMC to DDR Controller
   - This command will be stalled in the DDR Controller behind the completion of the 1) Stream of Writes
3. Real-Time Access to same set as the 2) Non Real-Time Read will be stalled inside MSMC due to Set Hazarding

Workaround(s):

Software should attempt the following workarounds in order of least to most impact to SW.

1. Cadence DDR controller prioritizes writes to the same page over a read from another page causing a delay in returning the read. Try reducing the DDR controller command_age_count from 0xto 0xF - corresponding to reducing the command age count from 16 DDR refresh cycles (62 us) to 1 refresh cycle (3.9 us). In most of the cases issue is resolved with this setting, but in some cases there are still some underflows. In that case SW may require either 2 or 3 workaround.
2. If possible set the ARM MMU attribute to configure DDR as “Normal memory” instead of “Device memory” type. This makes ARM to DDR access to be more efficient and helps to alleviate the problem. This is the observation based on test results so far, but it may need more analysis and further system testing. If this workaround is not possible in the system, SW may require workaround 3).
3. If possible make the Real-Time access as non IO-coherent. Set the RT access ATYPE = 3 for non-virtualized cases, and set ATYPE=1 & MEMTYPE=0 for PVU specific cases. This forces the RT traffic to bypass the MSMC set-hazarding logic. SW will have to do the cache operations.

i2123 ***STOG: Timed Out Emulation Debug write responses from the Slave Gasket always return Success***

Details:

When the gasket flushes transactions, all responses should go back with a time-out error, but in the case of emulation debug writes, the response is incorrectly returned as Success.

Workaround(s):

SW should not assume emulation debug writes are successful when there is a system timeout occurrence/interrupt.

i2126 ***STOG: Error miscounting when there are two concurrent timeouts or two concurrent unexpected responses***

Details:

When there is a read command and write command that timeout in the same cycle, the timeout counter will only increment by 1 instead of 2 in this situation. Likewise, if an unexpected read response and an unexpected write response both arrive in the same cycle, the unexpected response counter will only increment by 1 instead of 2.

Workaround(s):

The error counters are primarily supplemental information for software debug. Only one timeout error command/transaction info is recorded. The counters saturate at a count of 3, so the software should primarily focus on the error counter value being non-zero vs the exact counter value. The same approach should be applied to the unexpected response counter. Note: unexpected responses are dropped by the flush gasket.

i2127 ***STOG: SRC side write data bus hang when a write command timeout occurs the same cycle as last acceptance on DST side***

Details:

If a write command times out the same cycle the last write dataphase is accepted on the destination side of the gasket, the gasket's source side will permanently stop accepting write data and won't be able to flush/auto respond properly.

Programming the gasket with low timeout period can result in a system hang due to the time out gasket stop accepting write data.

Workaround(s):

Software should set a sufficiently large timeout period that well exceeds the longest possible write command burst transmission period. The default timeout period for the gasket is sufficient - 3 x 2^30 cycles.

i2134 ***USB: 2.0 Compliance Receive Sensitivity Test Limitation***

Details:

Performing receive sensitivity tests (EL_16 and EL_17) as defined in the USB-IF USB 2.0 Electrical Compliance Test Specification may invoke the problem described in Advisory i2091.

The issue was originally found while performing these tests using automation software, which increased USB signal amplitude while sending packets. The software was sweeping the amplitude from a value less than 100 mV to a value greater than 150 mV while verifying the device under test (DUT) NAK'd all packets below 100 mV and NAK'd no packets above 150 mV. However, increasing the amplitude through the squelch threshold while sending valid packets may lock the PHY as described in Advisory i2091.

Workaround(s):

Enable both of the following hardware workarounds.

Set cdr_eb_wr_reset bit (bit 7) to 1'b1 in UTMI_REG28 register present in USB*_PHY2 region.

Set phyrst_a_enable bit (bit 0) to 1'b1 in PHYRST_CFG register present in USB*_MMR_MMRVBP_USBSS_CMN region. Please note that phyrst_a_value (bits 12:8) in PHYRST_CFG register should be retained at default value of 0xE.

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).

i2146 ***UDMA: Force teardown bitfield readback is masked in realtime TX/RX registers***

Details:

The force teardown bit field will not remain set in the read back of the realtime TX/RX registers after a force teardown is initiated.

Workaround(s):

The Force Teardown operation is only used by software to intervene to address a catastrophic system condition, so software should separately track when it initiates a forced teardown verses a normal teardown, and thus not depend on the readback value of the force teardown bitfield to obtain this information.

i2151 *ADC: Debounce time control register*

Details:

CTRLMMR_WKUP_PADCONFIG76.DEBOUNCE_SEL controls the debounce time for MCU_ADC0_AIN0:7 and CTRLMMR_WKUP_PADCONFIG84.DEBOUNCE_SEL controls the debounce time for MCU_ADC1_AIN0:7. These registers set the debounce period for all of the input channels on the respective ADC whether or not the specific input (e.g. MCU0_ADC0_AIN0 or MCU_ADC1_AIN0) is used.

Workaround(s):

None

i2157 *DDR: Controller Anomaly in Setting Wakeup Time for Low Power States*

Details:

The DDR controller may erroneously decrease the wakeup time for the present low power state if the wakeup time for the next deeper power state is either disabled, or set to a lower value.

Workaround(s):

If a particular low power state is enabled by setting a bit in the DDRSS_CTL_139[29-24] LPI_WAKEUP_EN bit field, all deeper power state bits must also be enabled. From bit 0 through 4, low power states go deeper and deeper as the bit number increases. For example, if bit 0 is set, all bits from 1 through 4 must also be set. Similarly, if bit 2 is set, bit 3 and 4 must also be set.

In addition, the following wakeup values must be programmed in increasing order:

1. LPI_CTRL_IDLE_WAKEUP_FN related to LPI_WAKEUP_EN[0] -> value should be less than all fields below
2. LPI_PD_WAKEUP_FN related to LPI_WAKEUP_EN[1] -> value should be less than all fields below
3. LPI_SR_SHORT_WAKEUP_FN, LPI_SR_LONG_WAKEUP_FN, LPI_SRPD_SHORT_WAKEUP_FN, LPI_SRPD_LONG_WAKEUP_FN related to LPI_WAKEUP_EN[2] -> value should be less than all fields below
4. LPI_SR_LONG_MCCLK_GATE_WAKEUP_FN, LPI_SRPD_LONG_MCCLK_GATE_WAKEUP_FN related to LPI_WAKEUP_EN[3] -> value should be less than all fields below
5. LPI_TIMER_WAKEUP_FN related to LPI_WAKEUP_EN[4] -> highest value,

where FN = F0, F1, and F2 for different frequency set points.

i2159 *DDR: VRCG High Current Mode Must be Used During LPDDR4 CBT*

Details:

The DDR PHY updates VREFca for the command/address bus during LPDDR4 Command Bus Training (CBT). Bit 3 in LPDDR4 Mode Register 13 (MR13) defines the VRef Current Generator (VRCG) mode inside the LPDDR4 device. If this bit is set to 0, the VREFca settling time is too long for subsequent operations to work properly. To ensure proper operation of CBT, bit 3 in MR13 must be set to 1 (VRef Fast Response high current mode) during CBT.

Workaround(s):

To ensure proper operation, VRef Fast Response high current mode should be enabled during both Command Bus Training (CBT) and Write DQ Vref Training. This can be done by setting the following fields to 1:

For chip select 0: PI_MR13_DATA_0[3] in the DDRSS_PI_259 register

For chip select 1: PI_MR13_DATA_1[3] in the DDRSS_PI_261 register

i2159 (continued) ***DDR: VRCG High Current Mode Must be Used During LPDDR4 CBT***

For chip select 2: PI_MR13_DATA_2[3] in the DDRSS_PI_263 register

For chip select 3: PI_MR13_DATA_3[3] in the DDRSS_PI_265 register

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: DDRSS_PI_199[6-0] PI_CALVL_VREF_INITIAL_START_POINT_F0 and DDRSS_PI_199[14-8] PI_CALVL_VREF_INITIAL_STOP_POINT_F0 bit fields.

For frequency set 1: DDRSS_PI_199[22-16] PI_CALVL_VREF_INITIAL_START_POINT_F1 and DDRSS_PI_199[30-24] PI_CALVL_VREF_INITIAL_STOP_POINT_F1 bit fields.

For frequency set 2: DDRSS_PI_200[6-0] PI_CALVL_VREF_INITIAL_START_POINT_F2 and DDRSS_PI_200[14-8] PI_CALVL_VREF_INITIAL_STOP_POINT_F2 bit fields.

Recommendation is to use the nominal VRef value (based on the device programming of VDDQ/3 or VDDQ/2.5 along with the drive/termination settings used) +/- 4%.

i2161 ***R5FSS: Debugger Cannot Access VIM Module While It Is Active***

Details:

This issue impacts the Vectored Interrupt Module (VIM) inside R5FSS. There are registers inside VIM which change the state of the IP when they are read (such as VIM_IRQVEC). The expected behavior is that only functional reads should cause the state change. Debug reads (generated by TI debug tools such as CCS) to these registers should leave the state as it is. An issue exists currently where VIM treats debug register reads in the same way as functional register reads. This can cause a debug operation (such as opening a VIM register memory window in CCS) to inadvertently change the state of the VIM IP, making debug ineffective.

Workaround(s):

There is no work-around for this issue. The user should avoid accessing VIM registers while debugging.

i2163 ***UDMAP: UDMA transfers with ICNTs and/or src/dst addr NOT aligned to 64B fail when used in "event trigger" mode***

Details:

Note

The following description uses an example a C7x DSP core, but it applies to any other processing cores which can program the UDMA.

For DSP algorithm processing on C6x/C7x, the software often uses UDMA in NavSS or DRU in MSMC. In many cases, UDMA is used instead of DRU, because DRU channels are reserved in many use-cases for C7x/MMA deep learning operations. In a typical DSP algorithm processing, data is DMA'ed block by block to L2 memory for DSP, and DSP operates on the data in L2 memory instead of operating from DDR (through the cache). The typical DMA setup and event trigger for this operation is as below; this is referred to as "2D trigger and wait" in the following example.

For each "frame":

1. Setup a TR typically 3 or 4 dimension TR.
   a. Set TYPE = 4D_BLOCK_MOVE_REPACKING_INDIRECTION
   b. Set EVENT_SIZE = ICNT2_DEC
   c. Set TRIGGER0 = GLOBAL0
   d. Set TRIGGER0_TYPE = ICNT2_DEC
   e. Set TRIGGER1 = NONE
   f. ICNT0 x ICNT1 is block width x block height
   g. ICNT2 = number of blocks
   h. ICNT3 = 1
   i. src addr = DDR
   j. dst addr = C6x L2 memory
2. Submit this TR
   a. This TR starts a transfer on GLOBAL TRIGGER0 and transfers ICNT0xICNT1 bytes, then raises an event
3. For each block do the following:
   a. Trigger DMA by setting GLOBAL TRIGGER0
   b. Wait for the event that indicates that the block is transferred
   c. Do DSP processing

This sequence is a simplified sequence; in the actual algorithm, there can be multiple channels doing DDR to L2 or L2 DDR transfer in a "ping-pong" manner, such that DSP processing and DMA runs in parallel. The event itself is programmed appropriately at the channel OES registers, and the event status check is done using a free bit in IA for UDMA.

When the following conditions occur, the event in step 3.2 is not received for the first trigger:

- Condition 1: ICNT0xICT1 is NOT a multiple of 64.
- Condition 2: src or dst is NOT a multiple of 64.
- Condition 3: ICNT0xICT1 is NOT a multiple of 64 and src/dst address not a multiple of 64

Multiple of 16B or 32B for ICNT0xICNT1 and src/dst addr also has the same issue, where the event is not received. Only alignment of 64B makes it work.

Conditions in which it works:

- If ICNT0xICNT1 is made a multiple of 64 and src/dst address a multiple of 64, the test case passes.
- If DRU is used instead of UDMA, then the test passes. You must submit the TR to DRU through the UDMA DRU external channel. With DRU and with ICNTs and

i2163 (continued) ***UDMAP: UDMA transfers with ICNTs and/or src/dst addr NOT aligned to 64B fail when used in "event trigger" mode***

src/dst addr unaligned, the user can trigger and get events as expected when TR is programmed such that the number of events and number of triggers in a frame is 1, i.e ICNT2 = 1 in above case or EVENT_SIZE = COMPLETION and trigger is NONE. Then the completion event occurs as expected. This is not feasible to be used by the use-cases in question.

Above is a example for "2D trigger and wait", the same constraint applies for "1D trigger and wait" and "3D trigger and wait":

- For "1D trigger and wait", ICNT0 MUST be multiple of 64
- For "3D trigger and wait", ICNT0xICNT1xICNT2 MUST be multiple of 64

Workaround(s):

Set the EOL flag in TR for UDMAP as shown in following example:

- 1D trigger and wait
  - TR.FLAGS |= CSL_FMK(UDMAP_TR_FLAGS_EOL, CSL_UDMAP_TR_FLAGS_EOL_ICNT0);
- 2D trigger and wait
  - TR.FLAGS |= CSL_FMK(UDMAP_TR_FLAGS_EOL, CSL_UDMAP_TR_FLAGS_EOL_ICNT0_ICNT1);
- 3D trigger and wait
  - TR.FLAGS |= CSL_FMK(UDMAP_TR_FLAGS_EOL,CSL_UDMAP_TR_FLAGS_EOL_ICNT0_ICNT1_ICNT2);

There is no performance impact due to this workaround.

i2166 ***DDR: Entry and exit to/from Deep Sleep low-power state can cause PHY internal clock misalignment***

Details:

When DDR PHY enters the Deep Sleep low-power state, there is a delay before the PHY PLL is disabled and gated off. If exit from Deep Sleep occurs before the PHY PLL is disabled, the PHY internal clocks can get misaligned with respect to each other, resulting in timing failures inside the PHY.

Workaround(s):

If using software-initiated low-power mode by writing to LP_CMD in the DENALI_CTL_132 register, ensure that when entry into low-power mode has been acknowledged, wait for a minimum of 160 DDR clock cycles before requesting an exit from low-power mode. Another option is to use the following workaround.

If using PSC to disable the DDR interface, ensure that after disabling of DDR interface has been acknowledged, wait for a minimum of 160 DDR clock cycles before sending a request to enable it. Another option is to use the following workaround.

If using the controller's automatic mechanism for low power entry/exit using LP_AUTO_ENTRY_EN in the DENALI_CTL_141 register, use the following workaround.

Workaround: Ensure that DDR PHY does not enter Deep Sleep low-power state.

This can be ensured by programming the value of PHY_LP_WAKEUP[3:0] in the DENALI_PHY_1318 register is greater than the values of all the following thresholds in DDR controller registers.

i2166 (continued) ***DDR: Entry and exit to/from Deep Sleep low-power state can cause PHY internal clock misalignment***

LPI_CTRL_IDLE_WAKEUP_FN, LPI_PD_WAKEUP_FN, LPI_SR_SHORT_WAKEUP_FN, LPI_SR_LONG_WAKEUP_FN, LPI_SRPD_SHORT_WAKEUP_FN, LPI_SRPD_LONG_WAKEUP_FN, LPI_SR_LONG_MCCLK_GATE_WAKEUP_FN, LPI_SRPD_LONG_MCCLK_GATE_WAKEUP_FN, and LPI_TIMER_WAKEUP_FN

where FN = F0, F1, and F2 for different frequency set points.

i2177 ***RINGACC: The ring accelerator's debug transaction trace stream can be corrupted by certain ring access sequences***

Details:

The Ring Accelerator allows for hardware assisted debug through direct debugger access of its memory space and by the ability to export a trace stream of its transactions out to the cptracer network. Typically this debug information is enabled, collected and analyzed using a JTAG based debugger which interfaces with the ring accelerator through the SOC debug fabric. An errata exists which can result in a corruption or a hang of the ring debug trace information. This failure can be triggered by normal ring peek operation or if the debugger is used to initiate a ring pop operation. The corruption signature for this errata is a peek wrongly being reported as a pop in the trace. Additionally during non-ring modes (message or credential) a normal ring pop operation can result in incorrect information in the trace's empty filed or a debug pop operation can result in incorrect destination address.

Workaround(s):

To use the Ring Accelerator's hardware trace features for development, code should avoid using ring peek operations and debugger initiated pop operations.

i2182 ***DDR: Dual-rank non-power-of-2 density not supported with row-cs-bank-col address mapping***

Details:

DDR controller does not support dual-rank non-power-of-2 density LPDDR4 devices with row-cs-bank-col address mapping.

Please note that the above does not apply to single-rank non-power-of-2 density devices as well as all power-of-2 density devices.

Workaround(s):

Use cs-row-bank-col address mapping with dual-rank non-power-of-2 density LPDDR4 devices. To ensure cs-row-bank-col address mapping is selected, the cs_lower_addr_en field in the Cadence controller register must be set to 0.

i2183 ***PCIe: Link up failure when unused lanes are not assigned to PCIe Controller***

Details:

PCIe fails to link up if SERDES lanes not used by PCIe are assigned to another protocol. For example, link training fails if lanes 2 and 3 are assigned to another protocol while lanes 0 and 1 are used for PCIe to form a two lane link. This failure is due to an incorrect tie-off on an internal status signal indicating electrical idle.

Status signals going from SERDES to PCIe Controller are tied-off when a lane is not assigned to PCIe. Signal indicating electrical idle is incorrectly tied-off to a state that indicates non-idle. As a result, PCIe sees unused lanes to be out of electrical idle and

i2183 (continued) ***PCIe: Link up failure when unused lanes are not assigned to PCIe Controller***

this causes LTSSM to exit Detect.Quiet state without waiting for 12ms timeout to occur. If a receiver is not detected on the first receiver detection attempt in Detect.Active state, LTSSM goes back to Detect.Quiet and again moves forward to Detect.Active state without waiting for 12ms as required by PCIe base specification. Since wait time in Detect.Quiet is skipped, multiple receiver detect operations are performed back-to-back without allowing time for capacitances on the transmit lines to discharge. This causes subsequent receiver detections to always fail even if a receiver gets connected eventually.

Workaround(s):

Enable 2ms minimum wait time for Detect.Quiet state using DQMDC field in PCIE_CORE_LM_I_PL_CONFIG_2_REG register. This will cause LTSSM to wait for a minimum of 2ms in Detect.Quiet state. This allows sufficient time for capacitances on transmit lines to discharge between successive receiver detect operation.

i2184 ***CPSW: IET express traffic policing issue***

Details:

This applies to 9-port CPSW, 5-port CPSW, 3-port CPSW, and 2-port CPSW IET traffic.

In IET (Interspersed Express traffic), if a preempted packet was interrupted by an express packet, two things can occur:

1. If the express traffic is policed, the frame size for the preempted packets is applied to the express traffic policer. Assuming a policer was set up to rate schedule an express traffic stream, it would take a hit by the preempted packet size it interrupted. The preempted packet also takes on the express traffic policer status. As a result, preempted packets could get dropped along with other express traffic due to the express traffic policer.
2. If the express traffic was not policed, the interrupted preempt packet would not get its packet size applied to the preempted policer.

Workaround(s):

Do not police IET express traffic.

i2185 ***CPSW: Policer color marking issue***

Details:

Only applies to CPSW9G and CPSW5G.

When packets from two different ports hit the same policer such that one port has a large packet and the other has a short packet and the short packet arrives just after the large packet starts, the short packet will stop the backlog counting, resulting in potentially flagging the next frame for this policer as yellow when it should have been green. Because the policer is normally set up to not drop yellow, it should not cause an issue. This is only true of packets that arrive on different ports that share the same policer index.

Workaround(s):

Ensure policers are unique to ports.

i2186 ***DDR: LPDDR4 should be configured to 2666 MT/S***

Details:

LP4-3200 is not supported on pre-production units. Characterization is ongoing to determine the LPDDR4 maximum data rate supported.

i2186 (continued) ***DDR: LPDDR4 should be configured to 2666 MT/S***

Workaround(s):

LP4-2666 is recommended for SW development.

i2187 ***MSMC: Cache Resize to 0 Refreshes Tags instead of Updating them***

Details:

Data corruption (MSMC returning all 0's) occurs upon changing MSMC L3$ Size from non-zero to zero and back to non-zero for lines that previously had cached dirty data in MSMC's L3$ (DDR). A 0->N configuration directly after release of MSMC reset is not impacted by this issue.

MSMC internal cache resize transactions are always marked as *non-allocating* misses. Tags are only updated with new values on *allocating* misses and hits. This results in cache resize operations leaving the tags unchanged, while zeroing out the underlying data.

Because all existing TAGs remain in MSMC when changing L3 Cache Size but data is zeroed, subsequent reads to these previously cached lines will see all 0's returned for data.

Workaround(s):

Reset MSMC after L3 Cache is resized from N to 0 and prior to resizing L3 from 0 to X. This workaround preserves data because the L3 Cache Size N -> 0 transition forces data into DDR allowing DDR (in self refresh) to contain valid data.

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. In order to guarantee 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, it is necessary 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.

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 3-1 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.

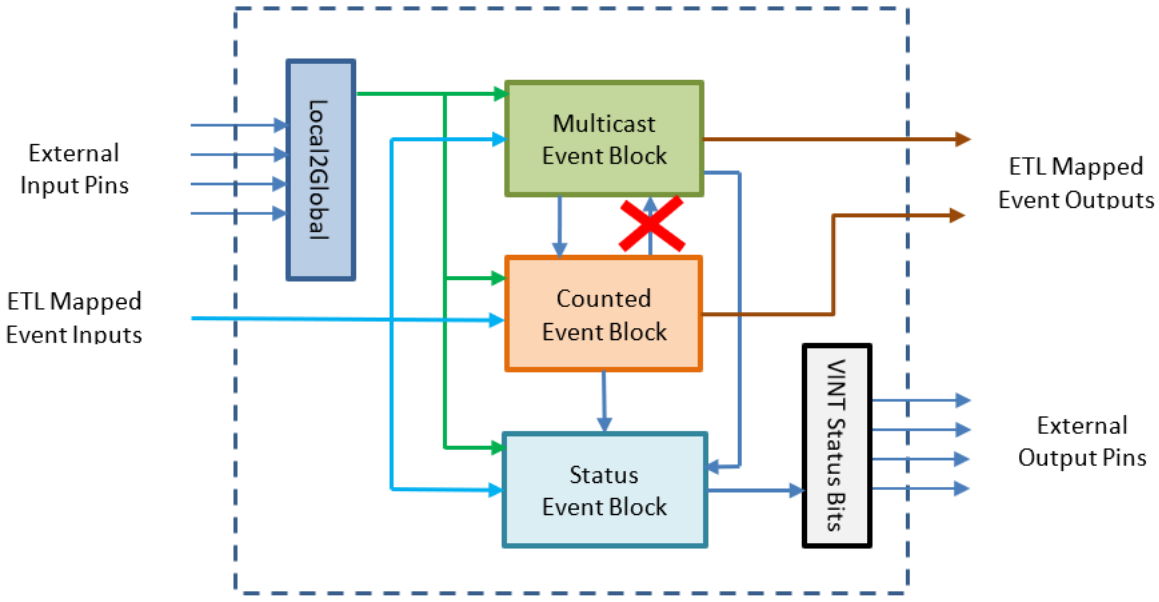

Figure 3-1. Interrupt Aggregator Version 1.0

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

i2197 ***I3C: Slave mode is not supported***

Details:

I3C Slave mode is not available. Only Master role on a single-master bus should be used.

Workaround(s):

None. Only Master role on a single-master bus should be used.

i2201 ***MSMC: Incorrect Parity Detect on bytecount***

Details:

A signal connection to check the parity of of byte enables for transactions coming from the navigator sub-system to the MSMC is incorrect, this causes false indications of a party error at the boundary of MSMC and throughout the path to DDR. To avoid these false error indications, the entire ECC aggregator must be disabled, which also disables detection of many other (potentially valid) errors and a loss of safety coverage.

Workaround(s):

The COMPUTE_CLUSTER0_MSMC_ECC_AGGR0 must be disabled. There is no direct way to reproduce the diagnostic coverage provided by these mechanisms. High-level system diagnostics may be implemented to mitigate the loss of coverage, but this will be application specific.

i2205 ***I3C: Command fetched during pending IBI is not properly processed in some cases***

Details:

Writing command by host during target-initiated IBI address byte reception may lead to improper command execution by controller, including incorrect frame generation.

Workaround(s):

Host must disable IBI by sending Broadcast DISEC CCC before sending commands to the controller.

i2207 ***CBASS: Command Arbitration Blocking***

Details:

When the interconnect arbitrates commands from multiple sources, the higher priority request always takes precedence. Requests that are at the same priority level are arbitrated in round-robin fashion. The issue is after the higher priority request goes idle and there are two or more pending requests that are at the same priority level, the hardware selects one of them arbitrarily. A potential issue may arise when software polls from multiple sources to the same endpoint: after servicing the higher priority source, the hardware may repeatedly select the same lower priority source for access. That means other same-lower priority requests may be blocked for a long time, and in the worst case if there are dependencies between the polling sequences, the software may run into a livelock state.

This issue only affects certain interconnect where in one switch module there are at least three sources that can access the same target simultaneously. Also note that when all requests are at the same priority level, the issue does not apply.

Workaround(s):

When multiple sources are simultaneously polling from the same endpoint, and there is expected dependency based on the read data, ensure that all sources are sending the read commands at the same priority level. The source that breaks the dependency should be at equal or higher priority than other dependent sources.

i2208 *CPSW: ALE IET Express Packet Drops*

Details:

This issue impacts the following Module:

[J7VCL] 5-port CPSW at 2.5G on ports 2-4

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.

Workaround(s):

During IET negotiation, tell the remote to fragment at 128 bytes.

i2209 *DCC: Incorrect clock selection*

Details:

An incorrect clock hookup on the MCU DCC means that the HFOSC1 clock cannot be selected for comparison. The primary purpose of the DCC on MCU island (run off of HFOSC0) is to compare with the independent clock source provided by HFOSC1.

Workaround(s):

Other clock sources for comparison are available (including the internal oscillator) with less accuracy. More attention should be paid to external watchdog mechanisms as well in order to indirectly check that clocks are working as expected by checking for loss of availability.

i2216 *I3C: Command execution may fail during slave-initiated IBI address byte reception*

Details:

An SoC host command to the I3C controller may lead to improper command execution by the controller, including incorrect frame generation, if the command was written while a slave-initiated IBI address byte reception is in progress.

In such case, the command response queue is incorrectly filled with responses. Additionally, if received IBI has no payload and is ACKed by Master, then slave fetched command causes incorrect frame issued over bus.

i2216 (continued) ***I3C: Command execution may fail during slave-initiated IBI address byte reception***

Workaround(s):

Host needs to disable IBI by sending Broadcast DISEC CCC before sending commands to the controller.

i2217 ***Recommended POST selection via MCU_BOOTMODE[09:08]***

Details:

The MCU_BOOTMODE[09:08] pins can be used to configure the Power-on-Selftest (POST) mode of operation. The effect of the MCU_BOOTMODE[09:08] depends on the TI factory setting for internal efuse override control. The options defined in the TRM are:

Table 4-6. POST Selection

POST Config Pins		POST Sequence
MCU 9	MCU 8	
0	0	DMSC LBIST followed by MCU LBIST followed by PBIST(2)
0	1	DMSC LBIST and MCU LBIST in parallel followed by PBIST(2)
1	0	Reserved(2)
1	1	POST bypass(1)

The recommended MCU_BOOTMODE[09:08] settings depends on the Device Type, as summarized in the Workaround section.

The Device Type is identified by the part number Y/Device Type designator, which is described in the SoC Data Manual chapter 10. This is illustrated in the following diagram:

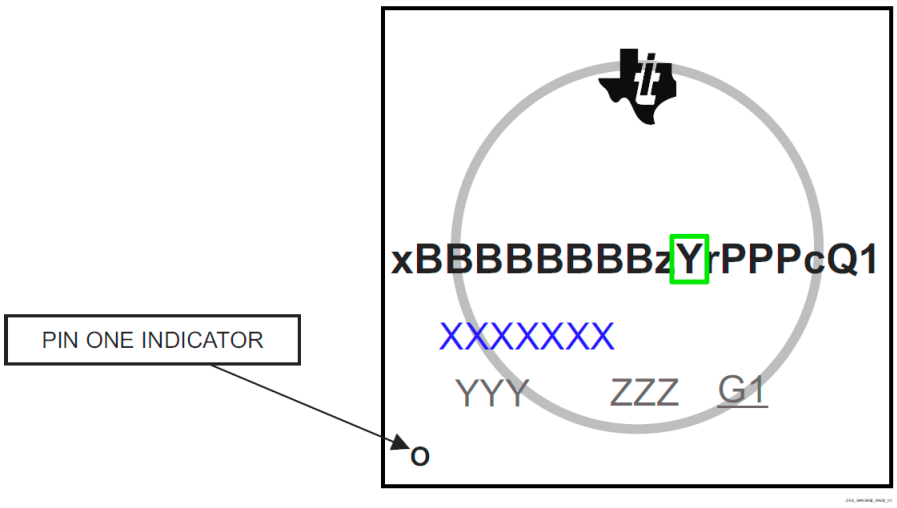

Figure 10-1. Printed Device Reference

Workaround(s):

For Device Type = C, 5, D

- MCU_BOOTMODE[09:08] pins are “don’t care” – overridden by efuse
- Factory efuse post_enable = 1
  - The SoC will run the POST sequence for “DMSC LBIST and MCU LBIST in parallel followed by PBIST” with a run-time of approximately 20 ms.
- TI recommends MCU_BOOTMODE[09:08] be set to ‘01’ to ensure compatibility with future devices.

For Device Type = G, 0

- MCU_BOOTMODE[09:08] must be set to ‘11’ for “POST Bypass”.

i2221 ***CC: Invasive and Non-Invasive debug enable settings are reset by MCU_RESETz***

Details:

The CTRLMMR_MCUSEC_CLSTR0_CORE[1:0]_DBG_CFG registers are erroneously reset to default values after asserting MCU_RESETz. They should only be reset on assertion of PORz. Therefore, any changes made to this register by software to affect Invasive and Non-invasive debug operation will be overwritten upon assertion of MCU_RESETz.

Workaround(s):

Software should reprogram the CTRLMMR_MCUSEC_CLSTR0_CORE[1:0]_DBG_CFG registers after MCU_RESETz is asserted.

i2222 ***Compute Cluster: A72 Corepac unable to be powered down***

Details:

If A72 Corepac powerdown is requested by software when DebugSS LPSC is OFF, Powerdown handshake could hang. When DebugSS LPSC is OFF, Clock to MSMC wrap's Debug block is gated and depending on initial state of A72 related debug components in this debug block, A72's LPSC may never receive disable_ack.

Workaround(s):

Before Initiating A72 Corepac Powerdown sequence software needs to make sure DebugSS LPSC is ON. This enables clock to MSMC wrap Debug block and ensures A72 Powerdown Handshake completes.

i2227 ***R5FSS: Error interrupt CCM_COMPARE_STAT_PULSE_INTR incorrectly driven***

Details

When modules on the device are functionally disabled/isolated to conserve power, any outputs from the device need to be held to a fixed value to avoid any downstream system issues.

Error interrupt CCM_COMPARE_STAT_PULSE_INTR from R5FSS is incorrectly driven to an active high value when the R5FSS is isolated/disabled. This will be recorded as an error occurring in the device if the detection logic is enabled in the Error Signaling Module (ESM). By default the detection logic is disabled.

Workaround

Do not enable ESM detection for this error until the R5FSS module is functionally active. Disable ESM detection for this error before disabling the R5FSS module.

i2228 ***JTAG: TAP used by Debuggers may be inaccessible if TRSTn device pin is never asserted***

Details

If TRSTn is never observed LOW, access to the embedded Debugger scan chains might be blocked by uninitialized logic. JTAG bypass and Boundary Scan functionality is not affected.

Workaround

Prior to connecting a Debugger, ensure that the TRSTn pin is asserted LOW for 100ns and subsequently de-asserted HIGH at-least one time after device power on.

i2232 *DDR: Controller postpones more than allowed refreshes after frequency change*

Details

When dynamically switching from a higher to lower clock frequency, the rolling window counters that control the postponing of refresh commands are not loaded correctly to scale to the lower clock frequency. This will result in controller postponing more refresh commands than allowed by the DRAM specification, thus violating refresh requirement for the DRAM.

Workaround

Workaround 1:Disable dynamic frequency change by programing DFS_ENABLE = 0

Workaround 2:If switching frequency, program the register field values based on the pseudo code listed below.Note that the controller requires AREF_*_THRESHOLD values to be programmed before triggering initialization. Their values cannot be changed during mission mode after initialization . Therefore, the value of these parameters must be the lowest of all values needed for every frequency change transition planned to be used.

```
if (old_freq/new_freq >= 7){
    if (PBR_EN==1) { // Per-bank refresh is enabled
        AREF_HIGH_THRESHOLD = 19
        AREF_NORM_THRESHOLD = 18
        AREF_PBR_CONT_EN_THRESHOLD = 17
        AREF_CMD_MAX_PER_TREF = 8
    }
    else { // Per-bank refresh is disabled
        AREF_HIGH_THRESHOLD = 18
        AREF_NORM_THRESHOLD = 17
        // AREF_PBR_CONT_EN_THRESHOLD <=== don't care, PBR not enabled
        AREF_CMD_MAX_PER_TREF = 8
    }
}
else {
    AREF_HIGH_THRESHOLD = 21
    AREF_NORM_THRESHOLD //<=== keep AREF_NORM_THRESHOLD < AREF_HIGH_THRESHOLD
    AREF_CMD_MAX_PER_TREF = 8
    if (PBR_EN==1) { // Per-bank refresh is enabled
    //keep AREF_PBR_CONT_EN_THRESHOLD<AREF_NORM_THRESHOLD<AREF_HIGH_THRESHOLD
        AREF_PBR_CONT_EN_THRESHOLD
    }
}
```

i2234 *UDMA: TR15 hangs if ICNT0 is less than 64 bytes*

Details

The UDMA always attempts to send the burst size for a transaction. If the actual ICNT0 is less than the minimum burst size of 64 the UDMA will wait for data that is never coming and will hang. If the EOL is set in the TR then the UDMA always sends the data for the last data regardless of the size allowing for the transfer to be sent.

Workaround

This can be worked around by setting the EOL to 1 in the TR

i2235 *CBASS Null Error Interrupt Not Masked By Enable Register*

Details

There is optional feature in CBASS that adds the null error reporting MMR and interrupt source. When the feature is present and the interrupt is enabled, these two output ports: "err_intr_intr" (level interrupt source) and "err_intr_pls_intr" (pulse interrupt source) will be asserted when an access to a null region occurs. The enable for the interrupt is in the ERR_INTR_ENABLE_SET register (address offset 0x58).

The issue is CBASS ignores this enable bit, and as a result any null access always produces the interrupt sources/events.

Workaround

There is no spurious event due to this bug because of the default disable status of processor events. At system level, processors don't receive any event unless it's enabled in the associated GIC/VIM interrupt controller.

When the interrupt is enabled, and an interrupt does occur, write to the following registers at cbass level to clear it:

write 0x1 to the err_intr_enabled_stat register, then write 0x1 to the err_eoi register.

i2237

PCIe: SerDes Reference Clock Output does not comply to Vcross, Rise-Fall Matching, and Edge Rate limits

Details

The PCIe Reference Clock Output of the SerDes does not comply with the PCI-SIG specifications for VCROSS and Edge Rate limits. Therefore, some external PCIe components may have an issue receiving and using the Reference Clock. However, the SerDes in this Device family does not have an issue accepting this non-compliant Reference Clock. This means that a link that connects the SerDes in one Device to the SerDes in a second Device will not have an issue when one Device generates the Reference Clock and the other Device receives the Reference Clock.

Workaround

Option 1:

Add an external circuit to the PCIe Reference Clock SERDES0_REFCLK_P/N Output to bring the signal into electrical compliance.

A passive re-biasing circuit can be used to achieve a compliant Vcross level:

- Use the SerDes internal 50-ohm termination
- On each leg of the SERDES0_REFCLK_P/N output, use 100nF AC coupling capacitors, followed by a bias network formed by 1k-ohm pull-down resistor and a 3.5k-ohm pull-up resistor toVDDA_1P8_SERDES0
- Component tolerances should be +/-5% for the resistors and +/-30% for the capacitors

There are two options to achieve a compliant Edge Rate:

- An external buffer can be added to the Output SERDES0_REFCLK_P/N signal. Depending on the buffer chosen, a re-biasing circuit may also be needed to comply with the external buffer input requirements.
- A reduced channel loss spec of -4dB@4MHz can be followed to achieve compliant Edge Rates.

Option 2:

Use an external clock source to supply the PCIe Reference Clock to both the Root Complex and End Point Devices of the Link.

i2241 ***PCIe: The SerDes PCIe Reference Clock Output can exceed the 5.0 GT/s Data Rate RMS jitter limit***

Details

When operating the SerDes PCIe Reference Clock in Output mode, the RMS jitter of the clock may exceed the PCIe specification limit for the 5.0 GT/s Data Rate.

Workaround

Option 1:

Configure the Reference Clock output in Derived Refclk mode (as opposed to Received Refclk mode) and program the PLL configuration registers as follows:

- Set CMN_PDIAG_PLL0_CP_PADJ_M0 = 0x0128 to enable lower jitter operation

(Note for Devices that support 8.0 GT/s operation: Derived Refclk mode has an associated errata i2242 related to temporary disabling of the Refclk while changing Data Rates to/from 8.0 GT/s in a Single PLL SerDes Configuration).

Option 2:

Do not operate the PCIe interface at the 5.0 GT/s Data Rate.

Option 3:

Use an external clock source to supply the PCIe Reference Clock to both the Root Complex and End Point Devices of the Link.

i2242 *PCIe: The SerDes PCIe Reference Clock Output is temporarily disabled while changing Data Rates*

Details

The SerDes PCIe Reference Clock Output will be temporarily disabled when changing Data Rates to or from 8.0 GT/s in Derived Refclk mode (as opposed to Received Refclk mode) and using a single SerDes PLL to generate the PCIe TX and RX clocks. This is due to the PLL reprogramming which must be performed when changing the data rate from 2.5 GT/s or 5.0 GT/s to 8.0 GT/s in this mode.

Some external PCIe components that are using the PCIe Reference Clock may not tolerate the disabling of the clock when changing data rates. However, the SerDes in this Device family does not have an issue accepting this Reference Clock behavior. This means that a link that connects the SerDes in one Device to the SerDes in a second Device will not have an issue when one Device generates the Reference Clock and the other Device receives the Reference Clock.

Workaround

Option 1:

Configure the SerDes to use one PLL to generate the clocks for 2.5 GT/s and 5.0 GT/s data rates, and a second PLL to generate the clocks for 8.0 GT/s data rate. This option imposes some limitations:

A) If Internal SSC mode is used, the two PLLs will not spread in sync with each other. This could result in up to 5000ppm difference between frequency of the two PLLs, and therefore between the TX and RX of the link partners. Because of this, Internal SSC mode is not recommended.

B) Protocols used simultaneously with PCIe on different Lanes of the SerDes must be compatible with sharing the PLL configuration of at least one of the two PLLs used for PCIe.

Option 2:

Use Received Refclk mode. Note that this mode is impacted by the separate Output Refclk jitter errata advisory (i2241)

Option 3:

Do not operate the PCIe interface at the 8.0 GT/s Data Rate

Option 4:

Use an external clock source to supply the PCIe Reference Clock to both the Root Complex and End Point Devices of the Link.

i2243 *PCIe: Timing requirement for disabling output refclk during L1.2 substate is not met*

Details

PCIe base specification requires Refclk to reach idle electrical state within 100ns of CLKREQ# deassertion when entering L1.2 substate (please refer TL1O_REFCLK_OFF parameter).

This timing requirement cannot be met when sourcing Refclk from the device since hardware does not automatically gate Refclk. Refclk gating has to be performed by software by writing to PHY_EN_REFCLK field in SERDES_RST register.

As a result, Refclk cannot be gated in L1.2 substate. Generally, allowing Refclk to run in L1.2 substate is not expected to cause any functional issues. However, if gating Refclk within 100ns is required by the system, L1.2 substate cannot be supported.

Workaround

Use an external Refclk generator to supply the PCIe reference clock

i2244 *DDR: Valid stop value must be defined for write DQ VREF training*

Details

The DDR PHY uses start, stop, and step-size values for write DQ VREF training. If the stop value is not equal to the start value + a multiple of the step-size, then the final VREF setting can go beyond the maximum VREF range, causing the training to hang.

Workaround

Program the stop value as follows:

PI_WDQLVL_VREF_INITIAL_STOP = (multiple of PI_WDQLVL_VREF_INITIAL_STEPSIZE) + PI_WDQLVL_VREF_INITIAL_START

i2245 *DMSC: Firewall Region requires specific configuration*

Details

The ECC Aggregator inside DMSC (DMSC0_ECC_AGGR) has an endpoint firewall which is used to protect this region. By default, this firewall blocks all the transactions except from the M3 core inside DMSC.

Workaround

If another processor or endpoint needs to access DMSC0_ECC_AGGR region, software shall configure the firewall region with starting address 0x0 and end address 0xFFFF_FFFF, using the CBASS_FW_REGION_i_START_ADDRESS and END_ADDRESS registers associated with the DMSC0_ECC_AGGR region. This is the only allowable address configuration for this region.

i2246 *PCIe: Automatic compliance entry fails when unused SERDES lanes are not assigned to PCIe Controller*

Details

PCIe fails to enter compliance state when connected to a passive load. This happens when unused SERDES lanes are not assigned to PCIe Controller. For example, if PCIe is configured in 1 lane mode, then compliance entry fails if only lane 0 of SERDES is assigned to PCIe Controller and lanes 1, 2, and 3 are not assigned to PCIe Controller.

Status signals going from SERDES to PCIe Controller are tied-off when a lane is not assigned to PCIe. Signal indicating electrical idle is incorrectly tied-off to a state that indicates non-idle. As a result, Controller sees unused lanes to be out of electrical idle (indicating that the lane is not connected to passive load) and this prevents compliance entry.

Please note that this issue only affects automatic compliance entry mechanism when connected to passive load (for example a scope that presents termination in its receive lines but does not bring its transmit lines out of electrical idle). This issue does not affect Enter Compliance or Compliance Receive mechanisms defined by PCIe specification.

Workaround

Only available workaround is to assign all SERDES lanes to PCIe during compliance validation.

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).

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

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 3-1. 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.

i2257 ***Boot: xSPI boot mode redundant image boot failure***

Details

xSPI boot is not able to boot from redundant image offset at 0x400000 when image at offset 0x0 is corrupted. xSPI boot failure API in the ROM does not handle the header check for xSPI properly.

Workaround

For xSPI 1S mode operation, enable SPI as backup boot mode. Note that this workaround does not apply to xSPI SFDP and 8D modes. No workaround exists for SFDP and 8D modes.

i2274 *DDR: Including DDR in BSCAN causes current alarm on the DDR supply*

Details

BSCAN causes current alarm trips when DDR is included. Customers using BSCAN should be warned of this issue to preclude the DDR in the scan chain during boundary scan. This only affects device packages which have DDR interface pinned out.

Workaround

Remove DDR from the scan chain when performing boundary scan. If DDR interface is not pinned out, this errata does not apply.

i2275 *DMSC Secure Boot ROM: Potential Secure Boot vulnerability with explicit EC curve parameters in X.509 certificate*

Details

Boot ROM supports use of EC Root-of-Trust keys. However, the ROM implementation used explicit curve parameters specified in the X.509 certificate to save ROM memory.

- The issue is that explicitly defined EC parameters could replicate known public keys — using legitimate, but different, private keys — if the curve parameters were explicitly defined.
- NIAP (the US Common Criteria Scheme) recently published a series of technical decisions (TDs) about the use of ECDSA X.509 certificates crossing numerous Protection Profiles.
- According to RFC 5480, section 2.1.1, using explicitly defined EC parameters is NOT permitted for X.509 certificates.

Use of named curve extensions binds the public key and curve type to prevent this.

Ref:

1. Overview https://lightshipsec.com/explicitly-parameterized-ecdsa-x-509-certificates/
2. Microsoft vulnerability https://msrc.microsoft.com/update-guide/vulnerability/CVE-2020-0601
3. Sect. 2.1.1 advises against explicit curve parameters in PKI https://tools.ietf.org/html/rfc5480

Workaround

Use RSA Root Keys for affected device variants and revisions. Do not use EC Private Root Keys which require explicit form.

i2277 *POK: De-Glitch (filter) is based upon only two samples*

Details

The POK is sampled on an approximate period of 1.25us. The "near-by" sample history is saved in a circular buffer. The De-Glitch (filter) is designed to AND the last *n* entries from the sample history in order to generate the output (to the ESM).

The De-Glitch filter is programmable to {4, 8, 12, 16} samples. The De-Glitch output is based upon a check of only the last entry (0th) and programmed number of samples ago (i.e. 3rd, 7th, 11th, or 15th). The filter ANDs these two results (instead of 4, 8, 12, or 16) in order to to generate the FAIL output to the ESM.

Notice that when the POK is set to monitor a fixed threshold (UV or OV but not set to ping-pong), the un-checked samples will be used.

When the POKs are controlled in a ping-pong manner, the skipped samples will be discarded.

Workaround

There is no workaround.

i2277 (continued) ***POK: De-Glitch (filter) is based upon only two samples***

However, the intent of the De-Glitch (filtering) is to insure that a discrete voltage dip or rise does not trigger FAIL. The sampling of two points significantly separated in time means that the voltage dip / rise was not a single isolated event.

Since the filter requires all N samples to fail before generating a FAIL signal to the ESM, the inclusion of 2 points instead of N makes this circuit more sensitive.

i2278 ***MCAN: Message Transmit order not guaranteed 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.

i2283 ***Restrictions on how CP Tracer Debug Probes can be used***

Details

Some CP Tracer bus probes do not receive the full SoC physical address but only a minimal set that was relevant to endpoint being monitored. This limits the usefulness of the probe in the SoC Analysis > Traffic Profiling feature in CCS.

1) Address Filtering / Matching : User would typically input the full 36b/40b (depending on device) address for any address-qualified bus probe jobs.

2) Decoding of transaction trace : User would expect that the address provided in the decoded stream was the full 36b/40b physical address of the transaction.

Affected probes:

J7VCL

Workaround

none

i2306 ***ROM Code: Need to turn off internal termination resistors in SERDES***

Details

The SERDES implementation in this device has internal termination resistors enabled by default. During PCIe boot, the ROM code does not disable these termination resistors, which results in reduced voltage swing of the PCIe reference clock, potentially less than the minimum limit defined for a PCIe reference clock. This can result in PCIe boot failure.

Workaround

None

i2307 ***Boot: ROM does not properly select OSPI clocking modes based on BOOTMODE***

Details

The ROM bootloader only selects an internal loopback mode for SPI/QSPI/OSPI/xSPI boot, regardless of the Iclk field value selected by the BOOTMODE pins (see the device specific TRM for BOOTMODE pin mappings), which is intended to allow the user to choose an internal or external clocking method. This results in less flexibility in board topology in customers designs. Customers intending to use the external board loopback mode could see timing issues in ROM boot because the external loopback clock is not being used.

Workaround

The topology of the OSPI design must not use "External Board Loopback" if planning to use OSPI as a boot source. All other clocking topologies (including internal loopback or DQS) can be used. Refer to the device specific datasheet, section "Applications, Implementation, and Layout" for supported clocking topologies using OSPI.

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:

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

  - This is Valid since the pending data inside the FIFO will retrigger the 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 will retrigger 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 will cause DMA to be torn down by the SW driver
  - Valid since next incoming data will cause 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.

i2320 ***UDMA and UDMAP : Descriptors and TRs required to be returned unfragmented***

Details

The UDMA and UDMAP require that the descriptors and TRs are placed in a memory subsystem that returns the descriptor or TR without any fragmenting of the descriptors. However, there are some memories that contain a fragmentation bridge, which makes them not available for holding the descriptors and TRs.

For this device, the R5 TCM memory cannot hold descriptors or TRs for UDMA or UDMAP

Workaround

None

i2326 ***PCIe: MAIN_PLLx operating in fractional mode, which is required for enabling SSC, is not compliant with PCIe Refclk jitter limits***

Details:

The MAIN_PLLx, which optionally supplies the 100MHz PCIe Refclk for SERDES and external components, does not comply to the PCIe Refclk jitter limits when configured in fractional mode. Fractional mode is required for enabling SSC, therefore SSC mode is not compliant to the PCIe Refclk jitter limits.

Workaround(s):

When sourcing the 100MHz PCIe Refclk from the MAIN_PLLx, the MAIN_PLLx should be configured in integer mode only (DACEN = 0, DSMEN = 0). This prevents the use of SSC for PCIe Refclk, which requires the PLL to operate in fractional mode. If SSC is required on the PCIe interface, an external Refclk generator with SSC should be used to provide the SERDES 100MHz Refclk.

i2329 ***MDIO: MDIO interface corruption (CPSW and PRU-ICSS)***

Details:

It is possible that the MDIO interface of all instances of CPSW and PRU-ICSS peripherals (if present) returns corrupt read data on MDIO reads (e.g. returning stale or previous data), or sends incorrect data on MDIO writes. It is also possible that the MDIO interface becomes unavailable until the next peripheral reset (either by LPSC reset or global device reset with reset isolation disabled in case of CPSW).

Possible system level manifestations of this issue could be (1) erroneous ethernet PHY link down status (2) inability to properly configure an ethernet PHY over MDIO (3) incorrect PHY detection (e.g. wrong address) (4) read or write timeouts when attempting to configure PHY over MDIO.

For boot mode (only CPSW if supported), there is no workaround to guarantee the primary ethernet boot is successful. If this exception occurs during primary boot, the boot may possibly initiate retries which may or may not be successful. If the retries are unsuccessful, this would result in an eventual timeout and transition to the backup boot mode (if one is selected). If no backup boot mode is selected, then such failure will result in a timeout and force device reset via chip watchdog after which the complete boot process will restart again.

To select a backup boot option (if supported), populate the appropriate pull resistors on the boot mode pins. See boot documentation for each specific device options, but the typical timeout for primary boot attempts over ethernet is 60 seconds.

Workaround(s):

On affected devices, following workaround should be used:

i2329 (continued) ***MDIO: MDIO interface corruption (CPSW and PRU-ICSS)***

MDIO manual mode: applicable for PRU-ICSS and for CPSW.

MDIO protocol can be emulated by reading and writing to the appropriate bits within the MDIO_MANUAL_IF_REG register of the MDIO peripheral to directly manipulate the MDIO clock and data pins. Refer to TRM for full details of manual mode register bits and their function.

In this case the device pin multiplexing should be configured to allow the IO to be controlled by the CPSW or PRU-ICSS peripherals (same as in normal intended operation), but the MDIO state machine must be disabled by ensuring MDIO_CONTROL_REG.ENABLE bit is 0 in the MDIO_CONTROL_REG and enable manual mode by setting MDIO_POLL_REG.MANUALMODE bit to 1.

Contact TI regarding implementation of software workaround.

Note

If using Ethernet DLR (Device Level Ring) (on CPSW or PRU-ICSS) or EtherCat protocol (on PRU-ICSS) there may be significant CPU or PRU loading impact to implement the run-time workaround 1 due to required polling interval for link status checks. Resulting system impact should be considered.

In case of PRU-ICSS, the loading of the software workaround may be reduced by using the MLINK feature of MDIO to do automatic polling of link status via the MIIx_RXLINK input pin to PRU-ICSS which must be connected to a status output from the external PHY which does not toggle while the link is active. Depending on the specified behavior of the external PHY device, this PHY status output may be LED_LINK or LED_SPEED or the logic OR of LED_LINK and LED SPEED. Refer to the MDIO section of TRM for details on using the MLINK feature of MDIO. This feature is not available on the CPSW peripheral.

For EtherCAT implementation on PRU-ICSS, the software workaround will be done in RTUx/ TX_PRUx Core. The core will have to be dedicated for workaround, which means this can’t be used for other purpose. The implementation will support two user access channels for MDIO access. This provides option for R5f core and PRU core to have independent access channel. The APIs will be similar to the ones we will have in RTOS Workaround implementation.

EtherCAT will continue to use PHY fast link detection via MDIO MLINK bypassing state m/c for link status (as this path is not affected by errata). This makes sure that cable redundancy related latency requirements are still met.

i2329 (continued) ***MDIO: MDIO interface corruption (CPSW and PRU-ICSS)***

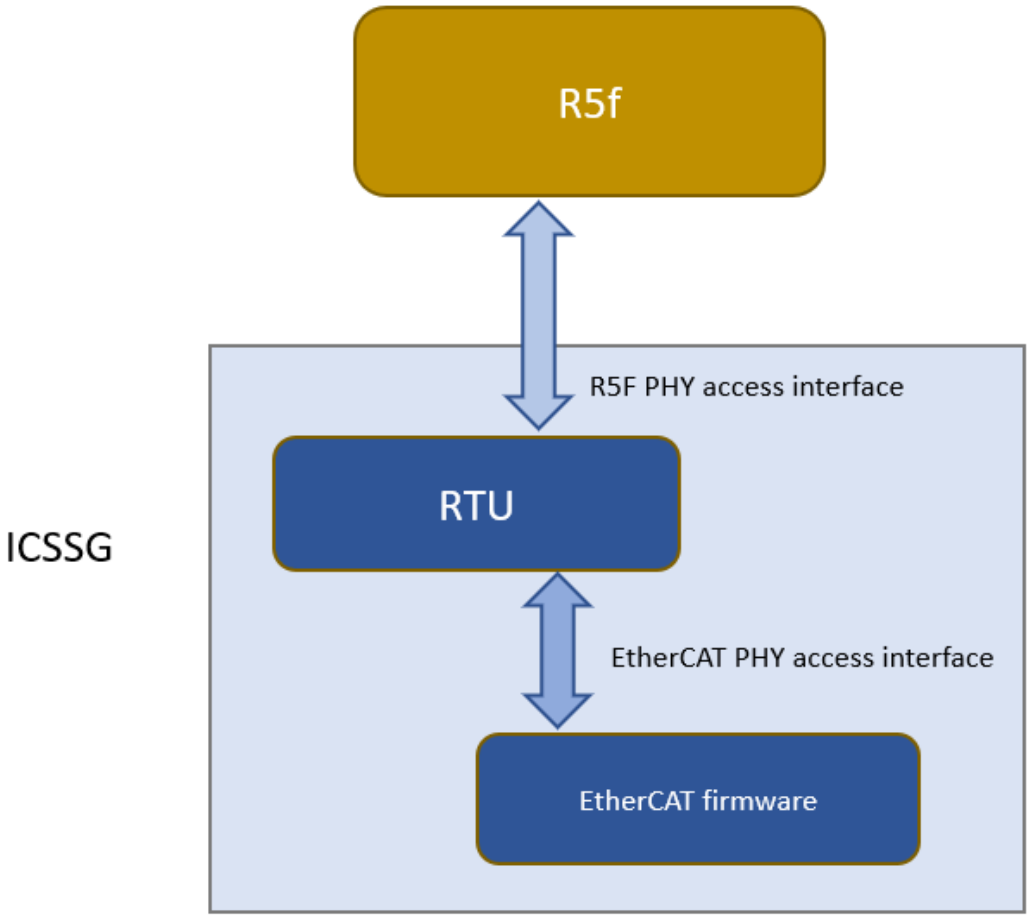

Figure 3-2. MDIO Emulation via Manual Mode using PRU Core

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.

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

Workaround(s):

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

i2360 ***Boot: Ethernet RMII Boot Mode is not supported***

Details:

Ethernet RMII Boot Mode is not supported and should not be used. It will be marked as Reserved in future revisions of the TRM.

Workaround(s):

None. An alternative Boot Mode should be selected.

i2361 ***Boot: SPI and xSPI BOOTMODE Pin Mapping changes for SR2.0***

Details:

The SPI and xSPI BOOTMODE pin mappings are changed between silicon revisions SR1.0 and SR2.0 (to align with other J7 Family device bootmode definitions), per the following table:

Primary Boot Mode B Pin	Primary Boot Mode A Pins	(merge left)	(merge left)	Boot Mode Selected for SR1.0	Boot Mode Selected for SR2.0
	MCU 5	MCU 4	MCU 3		
0	0	1	1	SPI	xSPI
1	1	1	0	xSPI	SPI

Workaround(s):

Configure the BOOTMODE pins per the above table to select the desired Boot Mode for each Silicon Revision.

i2362 ***10-100M SGMII: Marvell PHY does not ignore the preamble byte resulting in link failure***

Details:

The CPSW SGMII module outputs up to 5 bytes of 0x50 preamble data when in 10/100 mode and there is an odd number of clocks between packets. All bytes should be 0x55. In 1000Mbps mode, which does not have the issue, there are seven 0x55's in the preamble previous to the SFD. In 100Mbps mode there are 70 bytes in the preamble before the SFD (because the data is replicated 10 times from 1000Mbps mode). The first five bytes of the seventy can be 0x50 when the issue occurs. This issue has been undetected until now due to testing only with the PHYs that allow the preamble to be eroded and don't care about the actual data in the first number of bytes However, this issue was recently detected with a Marvel PHY (88Q1111 or similar) that looks at the preamble data and makes packet keep/discard decisions based on preamble data of 0x50.

Workaround(s):

The workaround options are:

1. Use 1000M mode which does not have the issue.

OR

2. Use a TI PHY (DP83869 or similar) or any other PHY which can erode/ignore the preamble data in 10/100/1000M mode.

i2366 *Boot: ROM does not comprehend specific JEDEC SFDP features for 8D-8D-8D operation*

Details:

JEDEC spec JESD216 - SERIAL FLASH DISCOVERABLE PARAMETERS (SFDP) details the parameter table used in certain serial flash devices to describe features and how to communicate/configure the device. The ROM interprets relevant portions of the SFDP for a device's features (such as a how to change from 1S-1S-1S to 8D-8D-8D mode), but does not properly comprehend a flash device that requires:

- A swapped byte order in 8D-8D-8D mode compared to 1S-1S-1S mode
- A command extension that in 8D-8D-8D mode that requires a different command than the first byte sent (such as an inversion of the opcode or another unique byte)

Workaround(s):

Review the SFDP table of any candidate flash memory that is compliant with JEDEC JESD216; in most cases vendors do not publish this table and can instead be requested from the flash vendor. If the 18th DWORD of the JEDEC Basic Flash Parameter table has bit 31 with a value of "1b", then the memory must be programmed with a swapped byte order from the factory or programmed with the SoC. If bits [30:29] have a value other than "00b" then it will not work with any bootmodes in 8D-8D-8D mode. Avoid using any 8D-8D-8D bootmodes with that flash device as a result.

i2371 *Boot: ROM code may hang in UART boot mode during data transfer*

Details:

Due to advisory i2310, it is possible for ROM code execution to hang during UART boot. The software workaround presented in i2310 is not implemented in ROM, and thus an erroneous timeout interrupt can be triggered in an unexpected state. This can prevent the ROM from being able to clear this interrupt and therefore hang.

This can manifest any time UART boot mode is used or when UART is used as the boot interface to enable production flows such as UniFlash or programing eFuses with OTP Keywriter.

Workaround(s):

None. Another boot interface should be used.

i2372 *Boot: ROM doesn't support select multi-plane addressing schemes in Serial NAND boot*

Details:

The ROM bootloader does not support certain multi-plane Serial SPI NAND flash memories that require the read from cache/buffer command to comprehend changing the cache/buffer/plane number to access the correct data.

Workaround(s):

Carefully review the addressing requirements of a candidate flash memory for references to a special bit for selecting a plane/buffer/cache in the read from cache/buffer command. Do not use memories that have such a requirement.

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

i2459 *Boot: PCIe Boot Mode is not supported*

Details:

PCIe Boot Mode is not supported and should not be used. It will be marked as Reserved in future revisions of the TRM.

Workaround(s):

None. An alternative Boot Mode should be selected.

Trademarks

All trademarks are the property of their respective owners.

Revision History

Changes from June 10, 2023 to December 31, 2024 (from Revision D (June 2023) to Revision E (December 2024)) **Page**

- Removed Advisory i2151; ADC: Debounce time control register.....14
- Updated Workaround for Advisory i2159, DDR: VRCG High Current Mode Must be Used During LPDDR4 CBT.....14
- Updated Details and Workaround for i2242, PCIe: The SerDes PCIe Reference Clock Output is temporarily disabled while changing Data Rates.....31
- Updated Workaround for Advisory i2326; PCIe: MAIN_PLLx operating in fractional mode, which is required for enabling SSC, is not compliant with PCIe Refclk jitter limits.....39
- Added Advisory i2459; Boot: PCIe Boot Mode is not supported.....44

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