

T2080 and T2081 Chip Errata

This document details all known silicon errata for the T2080 and T2081.

This table provides a revision history for this document.

Table 1. Document revision history

Revision	Date	Significant changes
5	12/2016	<p>Added the following errata:</p> <ul style="list-style-type: none"> • Debug A-009582 • DDR A-010165 • QMAN A-010383 <p>Modified the DDR erratum A-009942</p> <p>Removed CPC erratum A-002719</p>
4	02/2016	<p>Added the following errata:</p> <ul style="list-style-type: none"> • DDR A-009942 • Ethernet A-009885 • PCIe A-009518 <p>Modified the following errata:</p> <ul style="list-style-type: none"> • eSDHC A-009620 • Ethernet A-005177 • PCIe A-007999 • SerDes A-004985 <p>Removed IFC A-008175</p>
3	10/2015	<p>Added the following errata:</p> <ul style="list-style-type: none"> • eSDHC A-009620 • FMAN A-009450 • QMan A-009300 and A-009473 <p>Modified the following errata:</p> <ul style="list-style-type: none"> • FMAN A-008975 • PCIe A-009220
2	04/2015	<p>Added the following errata:</p> <ul style="list-style-type: none"> • PCIe A-008851, A-008913, A-009151, and A-009220 • SerDes A-004985

Table continues on the next page...



Table 1. Document revision history (continued)

Revision	Date	Significant changes
		Modified the following errata: <ul style="list-style-type: none"> • CPU A-008083 • FMAN A-006675 • SRIO A-008116 Changed the affected component of A-008968 and A-009006 from eMSG to RMAN
1	03/2015	Updated SerDes A-007186 from "Fixed in Rev 1.1" to "No plans to fix" and added the Freescale software workaround. Modified USB A-007992. Removed PME A-008659.
0	02/2015	The device silicon has qualified.

This table provides a cross-reference to match the revision code in the processor version register to the revision level marked on the device.

Table 2. Revision Level to Part Marking Cross-Reference

Part	Revision	e6500 Core Revision	Processor Version Register Value	System Version Register Value	Note
T2080E	1.0	2.0	8040_0020h	8538_0010h	—
T2080	1.0	2.0	8040_0020h	8530_0010h	—
T2081E	1.0	2.0	8040_0020h	8539_0010h	—
T2081	1.0	2.0	8040_0020h	8531_0010h	—
T2080E	1.1	2.0	8040_0120h	8538_0011h	—
T2080	1.1	2.0	8040_0120h	8530_0011h	—
T2081E	1.1	2.0	8040_0120h	8539_0011h	—
T2081	1.1	2.0	8040_0120h	8531_0011h	—

This table summarizes all known errata and lists the corresponding silicon revision level to which they apply. A 'Yes' entry indicates the erratum applies to a particular revision level, and an 'No' entry means it does not apply.

Table 3. Summary of Silicon Errata and Applicable Revision

Errata	Name	Projected Solution	Silicon Rev.	
			1.0	1.1
AltiVec				
A-007775	AltiVec unit must be powered up before attempting to enter the PH20 power management state	No plans to fix	Yes	Yes
CPC				

Table continues on the next page...

Table 3. Summary of Silicon Errata and Applicable Revision (continued)

Errata	Name	Projected Solution	Silicon Rev.	
			1.0	1.1
A-006001	CPC may not allocate LoadECs targeting caches other than CPC	No plans to fix	Yes	Yes
A-006379	CoreNet Platform Cache (CPC) CPCHDBCR0 register bit field [10:14] has incorrect default value after POR	No plans to fix	Yes	Yes
A-006380	Certain transaction types that should not affect the PLRU algorithm can inadvertently corrupt the algorithm state	No plans to fix	Yes	Yes
CPU				
A-006400	Guest state ISI handler is invoked on an ISI Virtualization Fault when EPCR[ISIGS] = 1	No plans to fix	Yes	Yes
A-006593	Atomic store may report failure but still allow the store data to be visible	No plans to fix	Yes	Yes
A-006958	A 64-bit read of TBL may appear to count backwards if the timebase increment causes a carry out from TBL into TBU	No plans to fix	Yes	Yes
A-007907	e6500 core may enter a state where a non-flushable transaction (L1 stash, dcbtIs CT = 0, dcbtstIs CT = 0) is stranded in the internal queue, resulting in a core hang	No plans to fix	Yes	Yes
A-008083	The dynamic frequency switch (DFS) can hang the SoC when changing frequency of a cluster with active cores or snoop transactions	No plans to fix	Yes	Yes
A-008139	Duplicate TLB0 entry may be created if a Hardware Tablewalk is active on one thread and the corresponding TLB1 indirect entry is replaced by the other thread	No plans to fix	Yes	Yes
DCE				
A-008261	T2080 DCE/PME cannot use memory coherency when PAMU is in bypass mode	No plans to fix	Yes	Yes
DCFG				
A-006748	DCFG_CRSTSRn momentarily reports core ready status while the threads are still coming out of boot hold-off state	No plans to fix	Yes	Yes
DDR				
A-007212	Runaway condition (WOT) on the DDR PLL oscillator	Fixed in Rev 1.1	Yes	No
A-007495	CKE is tri-stated until the HRESET signal is de-asserted	Fixed in Rev 1.1	Yes	No
A-008109	Memory controller could fail initialization with soldered down register and discrete DRAM	No plans to fix	Yes	Yes
A-009942	DDR controller can train to non-optimal setting	No plans to fix	Yes	Yes
A-010165	DDRC register write required to improve data eye margins for DDR-2133	No plans to fix	Yes	Yes
DUART				
A-004737	BREAK detection triggered multiple times for a single break assertion	No plans to fix	Yes	Yes
eSDHC				
A-008171	eSDHC transactions may fail in tuning modes of operation when the input data window is close to 1 UI during tuning	No plans to fix	Yes	Yes

Table continues on the next page...

Table 3. Summary of Silicon Errata and Applicable Revision (continued)

Errata	Name	Projected Solution	Silicon Rev.	
			1.0	1.1
A-009620	Data timeout error not getting set in case of command with busy response (R1b) as well as for busy period after last write block transfer	No plans to fix	Yes	Yes
Ethernet				
A-005177	RGMIIC AC timing specification for t_{SKRGT_TX} is not met	No plans to fix	Yes	Yes
A-007900	MDIO_CFG[MDIO_RD_ERR] bit is always cleared for external MDIO reads using EMI1 or EMI2	No plans to fix	Yes	Yes
A-007910	XAUI receive energy efficient Ethernet power management error detection function is controlled by transmit LPI enable	No plans to fix	Yes	Yes
A-008105	MDIO_CFG[BSY] of external MDIO may be cleared before read data is valid	No plans to fix	Yes	Yes
A-009056	RGMIIC internal loopback mode is non-functional for MAC3, MAC4, and MAC10	No plans to fix	Yes	Yes
A-009057	MAC4 does not support 10/100Mbps operation when connected to the RGMIIC interface	No plans to fix	Yes	Yes
A-009885	MDIO read transaction from external PHY may return FFFFh instead of valid data	No plans to fix	Yes	Yes
FMAN				
A-006675	FMan processing in an offline port results in a buffer leak if the port is not configured to release external buffers and classification decides to discard	No plans to fix	Yes	Yes
A-007273	FMAN soft reset is not finished properly if one of the Ethernet MAC clocks is disabled	No plans to fix	Yes	Yes
A-008975	The FMAN controller cannot allow empty look-up tables	No plans to fix	Yes	Yes
A-009450	Potential mismatch between VSP and error queue	No plans to fix	Yes	Yes
GEN				
A-007432	ASLEEP pin is not aligned with posedge of SYSCLK	No plans to fix	Yes	Yes
I2C				
A-006037	I ² C could hang if disabled after enabling	No plans to fix	Yes	Yes
A-007188	PBL wait command with a delay value greater than 2f_ffffh causes the device to get stuck in the reset state	No plans to fix	Yes	Yes
IFC				
A-007101	Problems with ECC mode when using 8 KB page size NAND devices	No plans to fix	Yes	Yes
A-008172	Issue with write protect toggling in NAND NVDDR mode	No plans to fix	Yes	Yes
MPIC				
A-008312	Duplicate edge-triggered interrupt after priority re arbitration	No plans to fix	Yes	Yes
PBL				
A-008665	A PBL wait command with a delay value greater than 1fff00h causes the device to stick in the reset state	No plans to fix	Yes	Yes
PCIE				
A-007189	PCI Express inbound error message handled incorrectly in RC mode	No plans to fix	Yes	Yes

Table continues on the next page...

Table 3. Summary of Silicon Errata and Applicable Revision (continued)

Errata	Name	Projected Solution	Silicon Rev.	
			1.0	1.1
A-007758	False window crossing error behavior when accessing the last byte of a PCI Express outbound ATMU window	No plans to fix	Yes	Yes
A-007815	The read-only-write-enable bit must be cleared to prevent overwriting read-only registers	No plans to fix	Yes	Yes
A-007941	Device ID and Revision ID registers fail to retain their updated values in EP mode after link-down or hot reset event	No plans to fix	Yes	Yes
A-007997	PCI Express hot-plug-related bits in the Slot Capabilities Register need to be cleared	No plans to fix	Yes	Yes
A-007999	Incorrect reset value for the multifunction bit in the PCI Express Header Type Register in RC mode	No plans to fix	Yes	Yes
A-008002	FLR is not supported despite the value that the Device Capabilities Register indicates for the SRIOV-capable PCI Express controller in EP mode	Fixed in Rev 1.1	Yes	No
A-008027	Incorrect values in the VF Device ID registers for both PF0 and PF1 of PCI Express controller 1 in SR-IOV EP mode	Fixed in Rev 1.1	Yes	No
A-008053	The PCI Express replay timer value needs to be increased	No plans to fix	Yes	Yes
A-008073	The PCI Express controller's next capability pointers are incorrect in some cases	No plans to fix	Yes	Yes
A-008098	PCI Express may corrupt memory when there are excessive correctable errors	No plans to fix	Yes	Yes
A-008099	PCI Express erroneously flags correctable errors when the link's ASPM is enabled in Gen 3 speed	No plans to fix	Yes	Yes
A-008100	PCI Express erroneously sets the Signaled Target Abort bit in the Status Register in EP mode	No plans to fix	Yes	Yes
A-008236	PCI Express controller does not exit disabled state if the Link Control register [Link Disable] bit is cleared before lanes enter electrical idle in RC mode	No plans to fix	Yes	Yes
A-008339	PBA offset value in MSI-X PBA Offset register is incorrect and PF1 VFs generate incorrect MSI-X transaction in SR-IOV EP mode	No plans to fix	Yes	Yes
A-008405	PCI Express Gen3 equalization cannot complete when lanes are reversed	No plans to fix	Yes	Yes
A-008851	Invalid transmitter/receiver preset values are used in Gen3 equalization phases during link training for RC mode	No plans to fix	Yes	Yes
A-008913	New A11 message request handling rules not supported	No plans to fix	Yes	Yes
A-009151	PCI Express controller does not correctly set the Link Autonomous Bandwidth Status and Link Bandwidth Management Status bits of the Link Status Register in RC mode	No plans to fix	Yes	Yes
A-009220	PCI Express controller upstream port does not advertise 8 GT/s data rate after receiving Gen3 equalization (EQ) TS Ordered Sets in EP mode	No plans to fix	Yes	Yes
A-009518	PCI Express EP controller's non-D0 PowerState is overridden to D0 when any single or combination of the Bus Master Enable (BME) or Memory Space Enable (MSE) bits are enabled from disabled setting by host software	No plans to fix	Yes	Yes
PM				

Table continues on the next page...

Table 3. Summary of Silicon Errata and Applicable Revision (continued)

Errata	Name	Projected Solution	Silicon Rev.	
			1.0	1.1
A-008088	e6500 does not exit PW10/PW20 after core warm reset	No plans to fix	Yes	Yes
QMAN				
A-009300	When a management command is performed to the Logical FQ Mapping Table while enqueues to the CEETM class queues are in progress, it could result in frames enqueued to the wrong class queue	No plans to fix	Yes	Yes
A-009473	PFDR reserved by order restoration lists (ORL) and notification message queues not accounted for in PFDR free pool count and low watermark threshold	No plans to fix	Yes	Yes
A-010383	CEETM Effective Packet Size used to update shaper credits is incorrectly calculated when using both Overhead Accounting Length and Minimum Packet Size	No plans to fix	Yes	Yes
RMAN				
A-002518	RapidIO Type 9 data streaming segmentation interleaving rule violated by RMan	No plans to fix	Yes	Yes
A-008968	RMan Type11 duplicate segment	No plans to fix	Yes	Yes
A-009006	Loss of inbound RMan hardware reassembly contexts for Type11	No plans to fix	Yes	Yes
SATA				
A-005619	ATAPI non-data commands fail when TTL=0 in command header	No plans to fix	Yes	Yes
A-005637	When a received data packet is smaller than the programmed length in the ATAPI command, the SATA host controller raises a false fatal error	No plans to fix	Yes	Yes
A-008077	Intermittent failures on some hard drives	No plans to fix	Yes	Yes
SEC				
A-006385	SEC watchdog timer does not prevent all cases of illogical descriptors from hanging DECOs	No plans to fix	Yes	Yes
A-008857	Anti-replay early-rollover error	No plans to fix	Yes	Yes
SerDes				
A-004985	SerDes receiver high-speed data path is short in gain	No plans to fix	Yes	Yes
A-007186	SerDes Ring VCO does not maintain lock throughout specified temperature range	No plans to fix	Yes	Yes
A-007809	SerDes 2 LC VCO does not work for SRDS_PRTCL_S2 = 0x29	No plans to fix	Yes	Yes
SRIO				
A-007837	When SRIO is configured in 4x port width mode, enabling 2x training may cause downtraining from 4x to 2x port width mode	No plans to fix	Yes	Yes
A-008116	A Serial RapidIO physical or logical/transport error interrupt (PINTn) that originates on port 2 is incorrectly reported as originating on port 1	No plans to fix	Yes	Yes
USB				
A-005697	Suspend bit asserted before the port is in Suspend state	No plans to fix	Yes	Yes
A-006261	The USB Host controller may falsely go into disconnect condition	No plans to fix	Yes	Yes

Table continues on the next page...

Table 3. Summary of Silicon Errata and Applicable Revision (continued)

Errata	Name	Projected Solution	Silicon Rev.	
			1.0	1.1
A-007792	USB reset hangs if ULPI phy is not connected and reset is issued before programming PTS field in UTMI mode	No plans to fix	Yes	Yes
A-007903	The initial bits of USB high speed packet show amplitude of about 70 mV higher than the rest of HS packet amplitude	No plans to fix	Yes	Yes
A-007992	Reduced HBM rating on USB_PWR_FAULT, USB_VBUS_CLAMP, USB_UDP, and USB_UDM pins	Fixed in Rev 1.1	Yes	No

A-007775: AltiVec unit must be powered up before attempting to enter the PH20 power management state

Affects: AltiVec

Description: The e6500 AltiVec unit has the ability to power down when there is no AltiVec activity and then power up under software control by setting AltiVec CDCSR0[AV_CNTL] = ON or under hardware control when an AltiVec instruction is executed.

If software running on one core programs the SoC RCPM registers to transition another targeted e6500 core into the PH10 power management state (halt or debug halt) while the targeted e6500 core is in the process of powering up its AltiVec unit, then the targeted core may hang when attempting to enter the PH20 power management state.

This erratum does not affect the PW20 power management state.

Impact: A core may hang if it attempts to go into the PH20 power management state while in the process of powering up the AltiVec unit.

Workaround: Choose one of the following three workaround options.

OPTION 1:

Prior to software configuring RCPM to issue power management halt or debug halt to the targeted core, do the following on the targeted core:

1. Set PWRMGTCR0[AV_IDLE_PD_EN] = 0
2. Set CDCSR0[AV_CNTL] = ON
3. isync
4. Poll CDCSR0[AV_STATE] until AV_STATE == READY

Then follow the normal procedure for configuring RCPM to issue power management halt or debug halt to the targeted core.

OPTION 2:

Always keep the AltiVec unit powered up and do not turn it off:

1. Set PWRMGTCR0[AV_IDLE_PD_EN] = 0
2. Do not set CDCSR0[AV_CNTL] = STANDBY

OPTION 3:

Disable the AltiVec unit by writing MSR[SPV] = 0, and do not issue any AV instructions.

Fix plan: No plans to fix

A-006001: CPC may not allocate LoadECs targeting caches other than CPC

Affects: CPC

Description: The CPCPAR configuration register's DRDALLOC, DRDPEALLOC, and DRDFEALLOC allocation control bits are intended to allocate LOADEC, LOADEC-PE, and LOADEC-FE transactions that target caches other than the CPC and miss the CPC. These transactions will only allocate the CPC if they hit in a processor cache and dirty data is cleaned to memory; LOADEC, LOADEC-PE, and LOADEC-FE transactions targeting caches other than CPC that do not clean data to memory will not allocate. Data integrity is not affected.

Additionally, the CPC's read allocate performance monitor will falsely count LOADEC, LOADEC-PE, and LOADEC-FE transactions targeting caches other than CPC that were intended to allocate, but did not allocate because they did not result in dirty data being cleaned to memory.

Impact: LOADEC, LOADEC-PE, and LOADEC-FE transactions targeting caches other than CPC may not allocate the CPC under these allocation policies. Data integrity is not affected.

The CPC's read allocate performance monitor may over-count these transactions that do not actually allocate

Workaround: None

Fix plan: No plans to fix

A-006379: CoreNet Platform Cache (CPC) CPCHDBCR0 register bit field [10:14] has incorrect default value after POR

Affects: CPC

Description: The intended replacement algorithm is for transitive streaming data. This does not occur.

Impact: The issue is that best performance cannot be achieved because the intended replacement algorithm is not chosen, thus the most recently used ways end up being victimized incorrectly.

Workaround: Set CPCHDBCR0 at offset 0xF00, bitfield [10:14] to 0x0F before enabling CPC or else mask CPCHDBCR0 register with 0x001E_0000 before enabling CPC.

Fix plan: No plans to fix

A-006380: Certain transaction types that should not affect the PLRU algorithm can inadvertently corrupt the algorithm state

Affects: CPC

Description: The CoreNet Platform Cache (CPC) does not filter certain transaction types (e.g. Sync) from affecting the PLRU algorithm it employs to select a way for eviction. The result is an unintended selection of a coherency granule currently residing in the CPC to be identified as having been recently accessed, which detects a hit to an allocated cache line. This, in turn, causes the PLRU algorithm to incorrectly protect the granule from being evicted. The unintended side effect is useful lines may be prematurely victimized.

Impact: This may cause recently used cache lines to be prematurely victimized. No data corruption occurs.

Workaround: None

Fix plan: No plans to fix

A-006400: Guest state ISI handler is invoked on an ISI Virtualization Fault when EPCR[ISIGS] = 1

Affects: CPU

Description: When an ISI Virtualization Fault (ISI-VF) is encountered and EPCR[ISIGS] = 1, the guest state ISI handler (GIVOR3) is invoked instead of the hypervisor state ISI handler (IVOR3).

In general, an ISI-VF should not occur since the hypervisor will only use the VF bit in indirect entries when a guest has page tables that the hypervisor wants to virtualize. However, if a hypervisor does try to virtualize a guest page table, the ISI-VF will be directed to the guest state on an instruction fetch from the guest state. The guest will not expect the ISI since the hypervisor set the VF bit for its own purposes.

Impact: If a guest receives an unexpected ISI, the guest may panic or return and re-execute the faulting instruction, which could result in a hang. This can take the partition out of service.

Note: if a hang occurs, other partitions will not be affected

Workaround: There are two workaround options:

1. Do not set the VF bit on indirect TLB entries.
2. Keep EPCR[ISIGS] = 0 to direct all ISI interrupts to the hypervisor state ISI and modify the hypervisor state ISI handler to handle guest state ISI cases.

Fix plan: No plans to fix

A-006593: Atomic store may report failure but still allow the store data to be visible

Affects: CPU

Description: A rare scenario exists where an atomic store may report failure but still allow the store data to be written to the L2 and be visible to snoopers. This scenario requires the following:

- The CoreNet Platform Cache must have sole ownership of the cache line.
- The cache line must not be in the L2 because it was evicted from the cache or killed by another master.

Impact: An atomic store may report failure but still allow the store data to be visible, which violates coherency rules.

Workaround: Set CoreNet Platform Cache register CPCHDBCR0 bit 21 to 1'b1. This may have a small impact on synthetic write bandwidth benchmarks but should have a negligible impact on real code.

Fix plan: No plans to fix

A-006958: A 64-bit read of TBL may appear to count backwards if the timebase increment causes a carry out from TBL into TBU

Affects: CPU

Description: When operating in 64-bit mode, if the timebase increment causes a carry out from TBL into TBU, there is a window where a 64-bit read of TBL may appear to count backwards.

Impact: Timebase may appear to count backwards.

Workaround: When operating in 64-bit mode, choose one of the following workarounds:

OPTION 1: Follow the same procedure for reading the timebase as for 32-bit implementations:

```
loop:
mfspr Rx,TBU      #load from TBU
mfspr Ry,TBL      #load from TBL
mfspr Rz,TBU      #load from TBU
cmp cr0,0,Rz,Rx   #see if 'old' = 'new'
bc 4,2,loop       #loop if carry occurred
```

The comparison and loop ensure that a consistent pair of values is obtained.

OR

OPTION 2: Replace 64-bit sequences that read the 64-bit timebase, such as

```
mfspr Rz,TBL      #get full timebase
```

with

```
loop:
mfspr Rz,TBL      #get full timebase
cmpwi Rz,0        #check to see if lower bits are 0
beq loop          #if timebase[32:63] = 0, wait until next tick to be sure
                  #it is correct
```

NOTE

When OPTION 2 is employed, if the Timebase freezes (that is, not incrementing), it is possible for the processor to hang if the Timebase is read and the value of the lower 32 bits is 0. This is because the workaround continually loops until the Timebase's lower 32 bits are non-zero.

In most systems, the Timebase will only freeze in very specific code sequences. OPTION 2 should not be used in those sections of code.

Fix plan: No plans to fix

A-007907: e6500 core may enter a state where a non-flushable transaction (L1 stash, dcbtIs CT = 0, dcbtstIs CT = 0) is stranded in the internal queue, resulting in a core hang

Affects: CPU

Description: The e6500 core may enter a state where a non-flushable transaction (L1 stash, dcbtIs CT = 0, dcbtstIs CT = 0) is stranded in an internal queue. A subsequent access to this address results in a core hang.

Conditions:

1. The internal queue is full and the newest entry is a non-flushable transaction (L1 stash, dcbtIs CT = 0, dcbtstIs CT = 0).
2. A core flush occurs, creating a hole or holes in the internal queue.
3. A new L1 stash, dcbtIs CT = 0, or dcbtstIs CT = 0 is in stage 2 of the Load/Store pipeline.

If all three conditions occur simultaneously, there is a one cycle window where the new L1 stash, dcbtIs CT = 0, or dcbtstIs CT = 0 could become stranded in the internal queue. The e6500 core hangs the next time this address is accessed.

Impact: The e6500 core may hang.

Workaround: Use the following steps:

1. Disable L1 stashes by writing L1CSR2[DCSTASHID] = 0. The issue does not affect L2 stashes, so software can direct stashes to the L2 Cache instead.
2. Insert a sync instruction before each dcbtIs CT = 0 or dcbtstIs CT = 0 instruction. Or, when performing a stream of only dcbtIs CT = 0 or dcbtstIs CT = 0 instructions with no other instructions in between, precede the stream of dcbtIs CT = 0 or dcbtstIs CT = 0 instructions with a sync instruction.

An embedded operating system may choose to set MSR[UCLE] = 0 and MSRP[UCLEP] = 1 so that all user-mode requests to lock or unlock cache lines are directed to the hypervisor.

Fix plan: No plans to fix

A-008083: The dynamic frequency switch (DFS) can hang the SoC when changing frequency of a cluster with active cores or snoop transactions

Affects: CPU

Description: The SoC can hang when changing cluster frequency using the DFS with active cores or snoop transactions targeting that cluster.

Impact: The SoC can hang when the DFS is used to change the frequency of a cluster with active cores or snoop transactions.

Workaround: Use only one of the following workarounds:

OPTION#1

Avoid DFS. The preferred method for power savings is by placing cores in the PW20 or PH20 power management states.

Option #2

Place the cluster in the PCL10 power management state before executing the SoC DFS sequence. The cluster can be returned to the active state after the DFS sequence has completed. The PC10 entry and exit sequences are as follows:

PCL10 entry sequence:

1. All cores in a cluster flush the L1 cache and enter wait state.
2. The master core polls the cores' states to make sure they have entered wait state.
3. The master core flushes and invalidates the L2 cache.
4. The master core places the cores into PH20 state.
5. The master core polls the cores' states to make sure they have entered PH20 state.
6. The master core disables the L2 cache for that cluster.
7. The master core places the cluster into PCL10.

PCL10 exit sequence:

1. The master core clears the cluster PCL10 status.
2. The master core invalidates and enables the L2 cache for that cluster.
3. The master core lets all cores of a cluster to exit PH20 state.
4. The master core polls the cores' states to make sure all cores of the cluster have exited PH20 state.
5. Cores of this cluster will reset and start again as a cold boot (reinitialize cache, TLB, MMU, ...)

NOTE

Option #2 is not applicable to single cluster devices (T2080, B4860, B3421).

Fix plan: No plans to fix

A-008139: Duplicate TLB0 entry may be created if a Hardware Tablewalk is active on one thread and the corresponding TLB1 indirect entry is replaced by the other thread

Affects: CPU

Description: When a TLB0 entry is written by a Hardware Tablewalk, the TID value written should be "0" if TID=0 in the indirect entry that caused the Hardware Tablewalk, else the TID value written should be written with the current PID value. This expected behavior is not always followed in the following scenario:

- Thread x: A Hardware Tablewalk is in progress and the thread has a non-zero PID or a non-zero LPID.
- Thread y: A **tlbwe** replaces a TLB1 indirect entry that has TID=0 or TLPID=0 that is in active use by the Hardware Tablewalk on Thread x with a different entry.

In this scenario, the error is that the TLB0 entry generated by the Hardware Tablewalk could incorrectly inherit its own non-zero PID/LPIDR when it should write "0".

Impact: TLB0 may have multiple valid entries for the same EA with different TID or TLPID fields. This could result in a multiple-hit exception if HID0[EN_L2MMU_MHD] is enabled. This could also result in confusing information from a **tlbsx** instruction.

Workaround: Ensure that a **tlbwe** is never used to overwrite a valid indirect entry, if the existing entry:

- Has TID=0 and could potentially be in use by the other thread that has a non-zero PID, OR
- Has TLPID=0 and could potentially be in use by the other thread that has a non-zero LPID.

Use **tlbilx** to invalidate the valid indirect entry before writing, and ensure that the other thread does not write to the TLB entry during this sequence.

Fix plan: No plans to fix

A-008261: T2080 DCE/PME cannot use memory coherency when PAMU is in bypass mode

Affects: DCE

Description: T2080 incorrectly interconnects with the snoop enable signal from DCE to CoreNet. The decompressor returns error code 81h, "invalid code lengths set" in *msg. However, it correctly reproduces the original text after inflating the compressed output generated by the compressor.

Impact: DCE/PME cannot use memory coherency if PAMU is in bypass mode.

Workaround: Configure snoop directly to PAMU.

Fix plan: No plans to fix

A-006748: DCFG_CRSTSRn momentarily reports core ready status while the threads are still coming out of boot hold-off state

Affects: DCFG

Description: When the cores exit the boot hold-off state at the end of the reset sequence, the Core Reset Status Register n (DCFG_CRSTSRn) momentarily reports an incorrect core ready status while the threads are still halted and are about to exit the halted state.

Impact: The DCFG_CRSTSRn reports an incorrect core ready status from the time the core HRESET is negated to the time the core negates the halted signal.

Workaround: Software should only read DCFG_CRSTSRn[READY] field 256 core clock cycles after the bit for the corresponding core is set in DCFG_BRRL to release the core from boot hold off.

Fix plan: No plans to fix

A-007212: Runaway condition (WOT) on the DDR PLL oscillator

Affects: DDR

Description: Runaway condition (WOT) exists on the DDR PLL oscillator, in all DDR PLLs on the chip. Therefore, a workaround must be used.

Impact: Odd DDR ratios are not supported. Only EVEN DDR ratios are supported by the device.

The PBI workaround is not sufficient for Secure Boot customers. The software workaround is not sufficient for boot from DDR customers. Consequently, secure boot from shared DDR is not supported.

Workaround: Choose one of the following workaround options.

NOTE

All the addresses mentioned are in the debug configuration space.

For both options, first write $RCW[MEM_PLL_RAT] = 0$.

OPTION 1-PBI

1. Write a value of 81d0_0000h to ALTCAR register at offset 18h, this step set the target ID to DCSR.
2. Wait for PBI (unsecure parts) before continuing with the workaround
3. Write a value of 0200_0001h (set bits 6 and 31) to the internal register at offset 2_1C28h and 2_1C48h
4. Write a value of e000_0000h (set bits 0:2) to the internal register at offset 2_1e80h
5. Write a value of 8800_0001h + (DDR PLL Ratio) to the internal register at offset 2_1C20h and 2_1C40h

For example, if 16:1 DDR PLL ratio (MEM_PLL_RAT) is required, set 21c40h = 21c20h = 8800_0011h. Odd ratios are not supported for this workaround.

6. Write a value of 0800_0001h +(DDR PLL Ratio)(clear bit 0) to the internal register at offset 2_1C20h and 2_1C40h

For example, if 16:1 DDR PLL ratio (MEM_PLL_RAT) is required, set 21c40h = 21c20h = 0800_0011h. Odd ratios are not supported for this workaround.

7. Wait for 100us (via PBI WAIT command)
8. Write a value of 0000_0000h (clear bits 6 and 31) to the internal register at offset 2_1C28h and 2_1C48h
9. Write a value of 0000_0000h (clear bits 0:2) to the internal register at offset 2_1e80h

OPTION 2—Software Patch:

A software patch is similar to the PBI patch, except the above WAIT should be implemented in software (instead of using the built-in WAIT command of the PBI).

Fix plan: Fixed in Rev 1.1

A-007495: CKE is tri-stated until the HRESET signal is de-asserted

Affects: DDR

Description: Historically all QorIQ devices drive CKE signal low while HRESET signal is asserted. But in this device the CKE signal is tri-stated while the HRESET signal is asserted. For proper operation of the DDR interface the CKE signal is required to be driven low for 10 ns before the DRAM reset signal is de-asserted. DRAM reset signal is not controlled or driven by memory controller.

Impact: DDR interface may not operate properly if DRAM reset signal is de-asserted when HRESET signal is asserted and CKE signal is tri-stated.

Workaround: De-assert the DRAM reset signal at least 10 ns after the HRESET signal is de-asserted and the CKE signal is driven low.

Fix plan: Fixed in Rev 1.1

A-008109: Memory controller could fail initialization with soldered down register and discrete DRAM

Affects: DDR

Description: Memory controller could fail to complete initialization when operating with soldered down DRAM devices and DDR3 registering clock driver (compliant to JEDEC SSTE32882). No failure has been observed when JEDEC compliant RDIMMs are used.

Impact: Memory controller could fail to initialize and operate.

Workaround: Change the following registers when all other DDR registers are initialized by software; before the 500 microseconds delay that is required before DDR_SDRAM_CFG[MEM_EN] is set:

1. Set DDR_SDRAM_CFG_2[DDR_SLOW]
2. Read the DEBUG_19 register and then OR with 0x00000002. This will only set the bit 30 of DEBUG_19 register, keeping all other bits unchanged.
3. Write DEBUG_29 = 0x30000000

NOTE

A-008109 erratum workaround must be executed in the following order:

1. Execute A-008378 erratum workaround first
2. Execute A-008109 erratum workaround second
3. Execute A-009942 erratum workaround third

[For this workaround, DEBUG_19 offset = 0xf48; DEBUG_29 offset = 0xf70]

Fix plan: No plans to fix

A-009942: DDR controller can train to non-optimal setting

Affects: DDR

Description: During the receive data training, the DDRC may complete on a non-optimal setting.

Impact: Memory controller fails to complete initialization. In the very unlikely event that the memory controller completes initialization with non-optimized training results, most of the subsequent read transactions fail.

Workaround: Before setting MEM_EN, ensure the following:

If operating at...	Then logical OR the DEBUG_29 with a value of...
DDR-1333	0080006ah
DDR-1600	0070006fh
DDR-1867	00700076h
DDR-2133	0060007bh

First read the DEBUG_29 register and then perform the following steps:

1. Logical AND the read value with 0xFF0FFF00.
2. Logical OR the read value with the value provide in the table above.
3. Write the result back to DEBUG_29.

NOTE

After implementing the existing workaround listed above, read the DEBUG_10, 11, 12, 13 and 14 from a working unit. Each debug register holds the CPO value for each byte lane as shown below (big-endian convention is used for all the following debug registers):

- DEBUG_10 [0:7] and DEBUG_10 [16:23] hold the CPO value for byte lane 0 and 1, respectively.
- DEBUG_11 [0:7] and DEBUG_11 [16:23] hold the CPO value for byte lane 2 and 3, respectively.
- DEBUG_12 [0:7] and DEBUG_12 [16:23] hold the CPO value for byte lane 4 and 5, respectively.
- DEBUG_13 [0:7] and DEBUG_13 [16:23] hold the CPO value for byte lane 6 and 7, respectively.
- DEBUG_14 [0:7] hold the CPO value for ECC byte.

When ECC byte lane is not used, ignore the ECC CPO information. Similarly, ignore the CPO information for any unused byte lanes. Determine the maximum and minimum CPO values among the CPO values that are read. Then check the condition in steps below to determine if the CPO value in the existing workaround for A009942 needs to be updated. The maximum and minimum CPO values will be used in the following steps:

1. In the case of DDR4, if the minimum CPO value read + 0x3B < the DEBUG_29 [24:31] listed in the table above, then go to step 3; otherwise CPO update in step 3 is not required.
2. In the case of DD3/3L, if the minimum CPO value read + 0x3F < the DEBUG_29 [24:31] listed in the table above, then go to step 3; otherwise CPO update in step 3 is not required.

3. Update DEBUG_29 [24:31] = (Maximum CPO value read + Minimum CPO value read)/2 + 0x27.

For this erratum, DEBUG_29 is at CCSRBAR + DDR_OFFSET + f70h. DEBUG_10, 11, 12, 13, and 14 are at CCSRBAR + DDR_OFFSET + 0xF24, 0xF28, 0xF2C, 0xF30, and 0xF34 respectively, these are read-only registers. All the debug registers are 32 bit.

Fix plan: No plans to fix

A-010165: DDRC register write required to improve data eye margins for DDR-2133

Affects: DDR

Description: During DDR-2133 operation, the transmit data eye margins determined during the memory controller initialization may be sub-optimal.

Impact: DDR failures may be observed when the transmit data eye margins are not optimal.

Workaround: Before MEM_EN is set, read and modify the write to set DEBUG_29[12] and DEBUG_29[13:16] = 4'b0100. Only bit 12 and [13:16] in DEBUG_29 are set, all other bits must not be modified.

NOTE

DEBUG_29 is offset f70h in the DDR memory mapped register space.

Fix plan: No plans to fix

A-004737: BREAK detection triggered multiple times for a single break assertion

Description: A UART break signal is defined as a logic zero being present on the UART data pin for a time longer than (START bit + Data bits + Parity bit + Stop bits). The break signal persists until the data signal rises to a logic one.

A received break is detected by reading the ULSR and checking for BI = 1. This read to ULSR clears the BI bit. After the break is detected, the normal handling of the break condition is to read the URBR to clear the ULSR[DR] bit. The expected behavior is that the ULSR[Bi] and ULSR[DR] bits do not get set again for the duration of the break signal assertion. However, the ULSR[Bi] and ULSR[DR] bits continue to get set each character period after they are cleared. This continues for the entire duration of the break signal.

At the end of the break signal, a random character may be falsely detected and received in the URBR, with the ULSR[DR] being set.

Impact: The ULSR[Bi] and ULSR[DR] bits get set multiple times, approximately once every character period, for a single break signal. A random character may be mistakenly received at the end of the break.

Workaround: The break is first detected when ULSR is read and ULSR[Bi]=1. To prevent the problem from occurring, perform the following sequence when a break is detected:

1. Read URBR, which returns a value of zero, and clears the ULSR[DR] bit
2. Delay at least 1 character period
3. Read URBR again, which return a value of zero, and clears the ULSR[DR] bit

ULSR[Bi] remains asserted for the duration of the break. The UART block does not trigger any additional interrupts for the duration of the break.

This workaround requires that the break signal be at least 2 character-lengths in duration.

This workaround applies to both polling and interrupt-driven implementations.

Fix plan: No plans to fix

A-008171: eSDHC transactions may fail in tuning modes of operation when the input data window is close to 1 UI during tuning

Affects: eSDHC

Description: In tuning mode of operation, when TBCTL[TB_EN] is set, eSDHC may report one of the following errors :

- Tuning error while running tuning operation where SYSCTL2[SAMPCLKSEL] will not get set even when SYSCTL2[EXTN] is reset. OR
- Data transaction error (e.g. IRQSTAT[DCE], IRQSTAT[DEBE]) during data transaction errors.

This issue occurs when the data window sampled within eSDHC is in full cycle. So, in that case, eSDHC is not able to find out the start and end points of the data window and sets the sampling pointer at default location (which is middle of the internal SD clock). If this sampling point coincides with the data eye boundary, then it can result in the above mentioned errors.

Impact: Tuning mode of operation for SDR50, SDR104 or HS200 speed modes may not work properly

Workaround: In case eSDHC reports tuning error or data errors in tuning mode of operation, then software can use one of the below mention workarounds :

- Shift the sampling pointer by half SD_CLK cycle from its default location. OR
- Increase the sampling clock (per_clk) frequency so that data window is sampled less than full cycle. OR
- Change the SD_CLK frequency so that the sampling point doesn't coincide with the start or end of data eye. OR
- Use fixed sampling technique for SDR50 mode.

Procedure to shift the sampling pointer by half SD_CLK period from its default location:

1. Program TBPTR[TB_WNDW_END_PTR] = 3*DIV_RATIO (i.e. three times division ration of SD_CLK)
2. Program TBPTR[TB_WNDW_START_PTR] = 5*DIV_RATIO
3. Program the software tuning mode by setting TBCTL[TB_MODE] = 2'h3
4. Set SYSCTL2[EXTN] and SYSCTL2[SAMPCLKSEL]
5. Issue SEND_TUNING_BLK Command (CMD19 for SD, CMD21 for MMC).
6. Wait for IRQSTAT[BRR], buffer read ready, to be set.
7. Clear IRQSTAT[BRR].
8. Check SYSCTL2[EXTN] to be cleared
9. Check SYSCTL2[SAMPCLKSEL], Sampling Clock Select. It's set value indicate tuning procedure success, and clear indicate failure. In case of tuning failure, fixed sampling scheme could be used by clearing TBCTL[TB_EN]

Fix plan: No plans to fix

A-009620: Data timeout error not getting set in case of command with busy response (R1b) as well as for busy period after last write block transfer

Affects: eSDHC

Description: In the event that a busy timeout occurs for a command with a busy response (for example, R1b response) as well as busy period after the last write block, the eSDHC does not set the IRQSTAT[DTOE] bit or the IRQSTAT[TC]. Therefore, the current command transfer is never completed.

Impact: Software has to track the busy timeout error in this case.

Workaround: Choose one of the following workarounds for commands with a busy response:

- Implement a software timer to track the busy timeout error. If this software timer expires before the IRQSTAT[TC] event then consider it as data timeout error event (same as if IRQSTAT[DTOE] gets set) and execute the error recovery sequence for data timeout error specified in section 3.10.1 "Error Interrupt Recovery" in SD Host specification 3.0.
- OR
- Don't set the XFRTYP[RSP]=2'b11 for commands with a busy response. Rather, poll the busy status of the card from the PRSSTAT[DLSL].

The workaround sequence for a busy period after last write block is as follows:

1. After the command completion interrupt (IRQSTAT[CC]), wait for de-assertion of PRSTAT[WTA].
2. As soon as PRSTAT[WTA] is de-asserted, start the software timer and poll the busy signal (DAT0) using PRSTAT[DLSL[0]].
3. Wait for DAT0 signal to go high (which indicates that the transfer is complete) or for the software timer expiry (which indicates a data timeout error).
4. Issue a soft reset for data (SYSCTL[RSTD]), if PRSTAT[DLA] bit is set.
5. In the case of a data timeout error (detected in step 3), perform error recovery.

Fix plan: No plans to fix

A-005177: RGMII AC timing specification for t_{SKRGT_TX} is not met

Affects: Ethernet

Description: In RGMII AC timing, the t_{SKRGT_TX} parameter is defined as data-to-clock output skew (at transmitter).

On Rev 1.1 silicon of this device, consider the following:

- In MAC4, t_{SKRGT_TX} ranges from -0.75 ns to +0.80 ns, instead of ± 0.5 ns for the RCW[EC2] RGMII configuration.
- In MAC3, t_{SKRGT_TX} ranges from -0.75 ns to +1.0 ns, instead of ± 0.5 ns for the RCW[EC1] RGMII configuration.

Impact: Out-of-specification operation could lead to boundedly undefined behavior, which may include false errors, false passes, or CRC errors.

Workaround: Use one of the following workarounds:

Option#1

To operate RGMII at 10 Mbps or 100 Mbps (these speeds inherently have larger receive skews), use MAC3 or MAC10. MAC4 only operates at 1 Gbps for RGMII (see A-009057).

Option#2

Terminate 1000 Mbps RGMII links with PHYs that accommodate larger receive skews.

NOTE

The t_{SKRGT_TX} timing of ± 0.5 ns typically requires a 1.5 ns to 2.1 ns of introduced delay in PCB, and/or some optional internal PHY delays, to the transmit clock signal relative to transmitted data to ensure that the receiving RGMII data to clock skew is between 1 ns to 2.6 ns. This erratum implies that there is no delay that would satisfy the standard receiver skew at 1000 Mbps since the transmitted skew span (1.0 ns + 0.75 ns = 1.75 ns) is larger than the receiver standard skew span (2.6 ns - 1.0 ns = 1.6 ns). Also, MAC10 meets RGMII industry specifications.

Fix plan: No plans to fix

A-007900: MDIO_CFG[MDIO_RD_ERR] bit is always cleared for external MDIO reads using EMI1 or EMI2

Affects: Ethernet

Description: The error indication bit MDIO_CFG[MDIO_RD_ERR] should be set when a read to an undefined address occurs. The improper behavior of the controller has this bit always cleared for external accesses.

For Clause 22 external reads (EMI1, MDIO_CFG[ENC45]=0), undefined values for MDIO_CTL[PHY_ADDR] and MDIO_CTL[REG_ADDR] are affected.

For Clause 45 external reads (EMI2, MDIO_CFG[ENC45]=1), undefined values for MDIO_CTL[PORT_ADDR], MDIO_CTL[DEV_ADDR], and MDIO_ADDR[REGADDR] are affected.

Impact: Software does not know if a read error occurred if an external access is made to an undefined or unsupported register address (Clause 22 or Clause 45).

Workaround: MDIO_DATA[MDIO_DATA]=0xffff can be used as an indication of a read error for both Clause 22 and Clause 45 external reads instead of MDIO_CFG[MDIO_RD_ERR].

Fix plan: No plans to fix

A-007910: XAUI receive energy efficient Ethernet power management error detection function is controlled by transmit LPI enable

Affects: Ethernet

Description: For RX LPI, when the MAC's low power idle status bit indicates the XAUI interface is in a RX LPI state (IEVENT[RX_LOWP]=1) and TX LPI is disabled (COMMAND_CONFIG[TX_LOWP_ENA]=0), errors during RX LPI state sequencing will not always be reported.

In addition, the MAC's IEVENT[RX_LOWP] status bit may remain high when the link partner has exited its TX LPI state.

Impact: Occurrence of an error during RX LPI may not be indicated or may be delayed.

Workaround: To track RX LPI errors, use one of the following options:

Option 1: Enable TX EEE, by setting the MAC COMMAND_CONFIG[TX_LOWP_ENA]. An error has occurred if IEVENT[RX_LOWP] transitions from 1 to 0 and SRDSxXAUIInCR1[ALGD]=0.

Option 2: If TX LPI is not desired, set MAC COMMAND_CONFIG[REG_LOWP_RXEMPTY] and follow IEEE 802.3az Table 48-9b and Table 48-10 to manually track the RX LPI states. For tracking the RX LPI states, trigger the reset and start of a software timer when RX_SLEEP, RX_QUIET or RX_WAKE state is initially observed. A way to define and track the states is described below:

- Set timer value to count Tqr, 4.0ms, when entering RX_SLEEP
- Set timer value to count Tqr, 4.0ms, when entering RX_QUIET
- Set timer value to count Twr + Twtf, 9.0us + 1.0ms, when entering RX_WAKE

On a timeout:

- Record RX_LINK_FAIL event
- Wait for link up indication, SRDSxXAUIInCR1[ALGD]==1 AND SRDSxXAUIInCR1[SYNC[3:0]]==4'b1111 AND IEVENT[RX_LOWP] ==0

RX LPI state	State transition conditions
RX_ACTIVE	MAC's low power idle, IEVENT[RX_LOWP] is cleared
RX_SLEEP ¹	MAC's low power idle, IEVENT[RX_LOWP] is set AND SRDSxXAUIInCR1[SYNC[3:0]]==4'b1111
RX_SLEEP_QEXIT ²	MAC's low power idle, IEVENT[RX_LOWP] is set AND SRDSxXAUIInCR1[SYNC[3:0]] != 4'b1111
RX_QUIET	From time of entry into RX_SLEEP_QEXIT, before Tqr, 4.0ms timer expires, IEVENT[RX_LOWP] is set AND SRDSxXAUIInCR1[SYNC[3:0]]==4'b0
RX_WAKE	From time of entry into RX_QUIET, before Tqr, 4.0ms timer expires, IEVENT[RX_LOWP] is set AND SRDSxXAUIInCR1[SYNC[3:0]]==4'b1111
RX_QUIET	From time of entry into RX_WAKE, before Twr + Twtf, 9.0us + 1.0ms timer expires, IEVENT[RX_LOWP] is set AND SRDSxXAUIInCR1[SYNC[3:0]]==4'b0
RX_ACTIVE	From time of entry into RX_WAKE, before Twr + Twtf, 9.0us + 1.0ms timer expires, IEVENT[RX_LOWP] is cleared

Table continues on the next page...

RX LPI state	State transition conditions
RX_SLEEP ¹	From time of entry into RX_WAKE, before T _{wr} + T _{wtf} , 9.0us + 1.0ms timer expires, IEVENT[RX_LOWP] is set AND SRDSxXAUInCR1[SYNC[3:0]]==4'b1111
RX_LINK_FAIL/RX_WTF	Any timeout from an expired timer OR, IEVENT[RX_LOWP] is cleared AND (SRDSxXAUInCR1[ALGD]==0 OR SRDSxXAUInCR1[SYNC[3:0]] != 4'b1111)
Notes: 1. Once in RX_SLEEP, there's no timeout if SRDSxXAUInCR1[SYNC[3:0]] never changes from 4'b1111 2. Placeholder state, the actual state is not in 802.3az	

Fix plan: No plans to fix

A-008105: MDIO_CFG[BSY] of external MDIO may be cleared before read data is valid

Affects: Ethernet

Description: Normally, the MDIO read routine waits for the MDIO_CFG[BSY] to clear as an indication that the data is ready in the MDIO_DATA register. However, the MDIO_CFG[BSY] may get cleared before a valid data is loaded into the MDIO_DATA register, resulting in incorrect data being read for external MDIO Clause 45 and Clause 22 accesses.

Impact: MDIO read routine of external MDIO ports may read invalid data

Workaround: The MDIO read routine should add sync instructions and wait for the MDIO_CFG[*CMP*] for command completion to ensure valid data is ready before reading MDIO_DATA.

The MDIO read sequence for Clause 45:

1. Wait until MDIO_CFG[BSY] = 0
2. Write MDIO_CTL with proper PORT_ADDR and DEV_ADDR and with bits READ and POST_INC cleared
3. Write MDIO_ADDR with proper REGADDR
4. Wait until MDIO_CFG[BSY] = 0
5. Wait until MDIO_CFG[*CMP*] = 1
6. Issue sync instruction
7. Clear MDIO_CFG[*CMP*] bit by writing “1” and wait until MDIO_CFG[*CMP*] = 0
8. Issue sync instruction
9. Write MDIO_CTL with proper PORT_ADDR and DEV_ADDR and with READ bit set and POST_INC bit optionally set
 - READ – A normal MDIO read transaction is initiated
 - POST_INC – A MDIO read with address post-increment transaction is initiated
10. Wait until MDIO_CFG[BSY] = 0
11. Wait until MDIO_CFG[*CMP*] = 1
12. Issue sync instruction
13. Clear MDIO_CFG[*CMP*] bit by writing “1” and wait until MDIO_CFG[*CMP*] = 0
14. Issue sync instruction
15. Read the data from MDIO_DATA.
 - If MDIO_DATA[MDIO_DATA]=0xffff, the read transaction did not receive a response from the addressed PHY. See A-007900.

The MDIO read sequence for Clause 22:

1. Wait until MDIO_CFG[BSY] = 0
2. Write MDIO_CTL with proper PHY_ADDR and REGISTER_ADDR and with the READ bit set
3. Wait until MDIO_CFG[BSY] = 0
4. Wait until MDIO_CFG[*CMP*] = 1
5. Issue sync instruction
6. Clear MDIO_CFG[*CMP*] bit by writing “1” and wait until MDIO_CFG[*CMP*] = 0
7. Issue sync instruction
8. Read the data from MDIO_DATA.
 - If MDIO_DATA[MDIO_DATA]=0xffff, the read transaction did not receive a response from the addressed PHY. See A-007900.

Fix plan: No plans to fix

A-009056: RGMII internal loopback mode is non-functional for MAC3, MAC4, and MAC10

Affects: Ethernet

Description: There are two internal loopback modes available for the RGMII interface. One internal loopback is at the MAC level and the other is at the RGMII interface level. The internal loopback at the RGMII interface does not operate as expected.

Impact: Internal loopback at the RGMII interface level is not fully functional and should not be used.

Workaround: Use MAC internal loopback.

MAC level loopback (1Gbps only) is enabled with the following register bit settings:

- COMMAND_CONFIG[XGLP]=1
- IF_MODE[RG]=0
- IF_MODE[RLP]=0
- IF_MODE[INV]=0

Table 5. Register offsets

Register bit	MAC3	MAC4	MAC10
COMMAND_CONFIG	CCSRBAR+4E_4008h	CCSRBAR+4E_6008h	CCSRBAR+4F_2008h
IF_MODE	CCSRBAR+4E_4300h	CCSRBAR+4E_6300h	CCSRBAR+4F_2300h

Fix plan: No plans to fix

A-009057: MAC4 does not support 10/100Mbps operation when connected to the RGMII interface

Affects: Ethernet

Description: There are up to two physical RGMII ports available based on RCW selection. One port is connected to MAC3 when RCW[EC1]=0b00 and the other port can be connected to either MAC4 (RCW[EC2]=0b00) or MAC10 (RCW[EC2]=0b10). RGMII on MAC4 is not functional at 10/100Mbps operation.

Impact: MAC4 is limited to 1Gbps only when connected to the RGMII interface.

Workaround: Use MAC4 only at 1Gbps. Use either MAC3 or MAC10 without any speed restrictions (1Gbps, 100Mbps, and 10Mbps are fully supported).

NOTE

When RGMII is enabled for a particular MAC, it always takes precedence over SerDes-based interfaces for which that MAC is mapped. RGMII speed selection is determined by IF_MODE[SETSP].

Fix plan: No plans to fix

A-009885: MDIO read transaction from external PHY may return FFFFh instead of valid data

Affects: Ethernet

Description: MDIO read transaction (Clause 22 or Clause 45) from an external PHY returns the result to the MDIO_DATA register. However, the MDIO_DATA register may not hold the valid information. The read data may get corrupted when the following conditions happen:

- The pattern of first 14 bits of read transaction exactly matches anywhere within the returned data at MDIO_DATA register. The first 14 bits of read transaction is the sequence of Start of frame(ST), Operation code(OP) and MDIO_CTL[22 : 31] bits.
- If MDIO_CFG[PRE_DIS]=1b, the time between MDIO_CTL register configuration and reading MDIO_DATA register is more than 48 x MDC clock period.
- If MDIO_CFG[PRE_DIS]=0b, the time between MDIO_CTL register configuration and reading MDIO_DATA register is more than 80 x MDC clock period.

The above conditions are to ensure that the time between MDIO_CFG[BSY]=0 (after configuring MDIO_CTL) and reading MDIO_DATA register is less than 16 x MDC clock period.

This condition is rare and may occur when there are core related delays when reading the MDIO_DATA register after configuring the MDIO_CTL register.

This errata is applicable to EMI1 (Clause 22) and EMI2 (Clause 45).

Impact: MDIO read transaction from external PHY may return corrupted data.

Workaround: When MDIO read transaction (Clause 22 or Clause 45) returns MDIO_DATA=FFFFh, MDIO read sequence should be repeated to ensure the following:

- If MDIO_CFG[PRE_DIS]=1b, the time between MDIO_CTL configuration and reading MDIO_DATA is less than 48 x MDC clock period.
- If MDIO_CFG[PRE_DIS]=0b, the time between MDIO_CTL configuration and reading MDIO_DATA is less than 80 x MDC clock period.

Timer or timestamp may be used to determine the time difference between the access events of the registers.

For example, create timestamp at following instances:

- Timestamp T1 = Configure MDIO_CTL register with read bit set
- Timestamp T2 = Read MDIO_DATA

The following conditions should be met for the workaround:

- If MDIO_CFG[PRE_DIS]=1b, T2 – T1 should be less than 48 x MDC clock period
- If MDIO_CFG[PRE_DIS]=0b, T2 – T1 should be less than 80 x MDC clock period

During the T2-T1, the process reading the MDIO_DATA could be interrupted by interrupt or be prompted by higher priority process which breaks the time conditions. In this case, the interrupt needs to be disabled to make sure the process is not interrupted.

If the workaround is implemented and MDIO_DATA register still returns FFFFh, it may mean that this is a valid read data from PHY.

Fix plan: No plans to fix

A-006675: FMan processing in an offline port results in a buffer leak if the port is not configured to release external buffers and classification decides to discard

Affects: FMan

Description: FMan processing in an offline port results in a buffer leak if the port is not configured to release external buffers (Frame queue context.EBD = 0) and the classification results in a discard. This means that FD[status] and the state of bits FMBM_OFSDM and FMBM_OFSEM result in the decision to discard and FMBM_CFG[FDOVR] is not set.

Impact: May result in a buffer leak if workaround is not used.

Workaround: Use one of the following workarounds:

Option #1:

Use NXP drivers and microcode. The workaround is included in FMD21_1 or later and FMan microcode IPACC 106.x.9 or later. For NCSW users, these are included in NCSW4.7 or later. For SDK users, these are included in SDK v1.5 or later.

Option #2:

Do not configure the BMI offline port to discard frames when the BMI is called with the "prepare to enqueue" action code if this port processes data from a frame queue where EBD = 0. Use only one of the following possible solutions:

- Direct all errors to the normal queue (result of classification). Clear all bits in FMBM_OFSDM.
- Direct certain errors to the error queue (set bits in FMBM_OFSEM) and configure this queue to tail drop = 0 so that hardware will automatically consume the queue and return its associated buffers to the appropriate buffer pools.
- Direct certain errors to the error queue (set bits in FMBM_OFSEM) and consume this queue in the software routine so that it returns the associated buffers to the appropriate buffer pools.

Fix plan: No plans to fix

A-007273: FMAN soft reset is not finished properly if one of the Ethernet MAC clocks is disabled

Affects: FMAN

Description: FMAN soft reset is not finished properly if at least one of the Ethernet MAC clocks is disabled by setting the corresponding bit of the DCFG_CCSR_DEVDISR2 register.

Impact: FMAN hangs if soft reset is done when at least one of the Ethernet MAC clocks is disabled.

Workaround: Choose one of the following workaround options.

OPTION 1:

Do not disable any of the Ethernet MAC clocks through the DCFG_CCSR_DEVDISR2 register. The impact of this workaround is that no power saving is obtained by disabling unused MACs.

OPTION 2:

Re-enable all disabled MAC clocks through the DCFG_CCSR_DEVDISR2 register prior to issuing an FMAN soft reset. Re-disable the MAC clocks after the FMAN soft reset is done. Inactive MAC clocks can be re-enabled because they are inactive. However, functional MACs cannot be re-enabled, as described in the reference manual.

Fix plan: No plans to fix

A-008975: The FMAN controller cannot allow empty look-up tables

Affects: FMAN

Description: Custom classification nodes are created using the FMAN controller driver (FMD). The FMAN controller cannot allow empty look-up tables.

Impact: Users may experience boundedly undefined behavior.

Workaround: Create at least one dummy entry in the look-up table that results with the missed action descriptor.

NXP plans to provide the workaround in FMAN microcode release 106_x_16.

Fix plan: No plans to fix

A-009450: Potential mismatch between VSP and error queue

Affects: FMAN

Description: If using classification, multiple VSPs may be used when frames are sent to the error FQID.

Impact: The incorrect VSP may be used when the classification process results in sending the frame to an error FQID.

Workaround: Use the provided FMD and microcode routines that change the VSP ID to the default VSP ID set in the FMBM_RFQID[RSPID] field before the BMI prepare-to-enqueue. The logic uses the default VSP if it determines the frame needs to be sent to the error FQID. A frame is sent to the error FQID based on the current method of comparing the error mask and discard mask bits (see BMI FMBM_xFSEM and FMBM_xFSDM).

This workaround is available in FMD Version FMD21_1 and microcode version IPACC 106.x.9. In FMD, the define FM_ERROR_VSP_NO_MATCH_SW006 should be set.

NCSW includes this workaround in NCSW4.7 and later. If the SDK is used, this workaround is available in SDK version 1.4 and later.

Fix plan: No plans to fix

A-007432: ASLEEP pin is not aligned with posedge of SYSCLK

Affects: GEN

Description: The assertion of the ASLEEP pin is one platform clock late to the posedge of SYSCLK due to the internal pipelined stage.

Impact: ASLEEP pin asserts one platform clock late to posedge of SYSCLK.

Workaround: Take into account that the ASLEEP pin asserts one platform clock late to posedge of SYSCLK.

Fix plan: No plans to fix

A-006037: I²C could hang if disabled after enabling

Affects: I²C

Description: The I²C controller starts a 64 K cycle counter when the software enables it via I2CCR[MEN]. It restarts this counter anytime SCL toggles, which could happen in a multi-master system if another transaction is in progress. If software clears I2CCR[MEN] before the 64 K cycle counter completes in either a single-master or multi-master system, then the I²C controller could enter into a bad state and hang future accesses to I²C registers.

Impact: I²C controller could hang if it is disabled after enabling.

Workaround: Choose one of the following workaround options.

OPTION 1:

During the first I²C transaction after enabling the controller, poll I2CSR[MIF] until it is set before allowing software to disable the I²C controller again.

OPTION 2:

Whenever enabling the I²C controller, software needs to set bit 6 of the I2CCR to 1. If the I²C controller is configured as a slave, the following additional steps are required:

1. Software enables the device by setting I2CnCR[MEN] = 1 and starts a timer.
2. Delay for 4 I²C bus clocks.
3. Check Bus Busy bit (I2CnSR[MBB]):
 - If MBB == 0, jump to Step 6 (good condition; go to normal operation).
 - Else, disable the device (I2CnCR[MEN] = 0).
4. Reconfigure all I²C registers if necessary.
5. Go back to Step 1.
6. Normal operation.

Fix plan: No plans to fix

A-007188: PBL wait command with a delay value greater than 2f_ffffh causes the device to get stuck in the reset state

Affects: I2C

Description: When using I²C as RCW/PBI source, the PBL wait command with a delay value greater than 2f_ffffh causes the device to get stuck in the reset state.

Impact: Device does not come out of reset when the delay value exceeds 2f_ffffh.

Workaround: The PBL wait command must have a delay value less than 2f_ffffh when using I²C as a RCW/PBI source.

Fix plan: No plans to fix

A-007101: Problems with ECC mode when using 8 KB page size NAND devices

Affects: IFC

Description: 8 KB page size NAND support has following issues:

1. During boot in ECC mode, an incorrect default spare-size value is loaded for an 8 KB page size NAND device. This restriction applies only when the device boots from NAND. In normal operation, a valid spare-region size can be programmed by software in the CSORn_EXT[SPARE_BYTES_CS_n] register.
2. Operations can not be performed on 8KB page size NAND devices when operated in 24/40 bit ECC mode. This restriction applies to boot as well as normal operation.

The following table explains the required size and default values chosen by IFC.

Table 6. Required size and default values

ECC mode	Minimum spare size required	Default value chosen during boot
4 bit/sector	136	128
8 bit/sector	264	210

Impact: The device cannot boot from 8 KB page size NAND in ECC enable mode. The 4/8b ECC mode restriction does not apply if NAND is not a boot source because a valid spare region size can be programmed by software in the CSORn_EXT[SPARE_BYTES_CS_n] register. 8 KB page size NAND devices do not work in 24/40 bit ECC modes during normal operation or during boot.

Workaround: Choose one of the following workarounds:

1. Disable ECC check in 4-bit and 8-bit ECC mode when booting from 8-KB NAND, and use software algorithm to protect the NAND page data.
2. Disable ECC check in 24-bit and 40-bit ECC mode when using 8-KB NAND, and use software algorithm to protect the NAND page data.

Fix plan: No plans to fix

A-008172: Issue with write protect toggling in NAND NVDDR mode

Affects: IFC

Description: IFC does not work properly in the following scenarios:

- CSPRn[WP] is changed immediately after switching from asynchronous SDR mode to synchronous NV-DDR mode.
- NAND device is in NVDDR mode and write protect value in CSPRn[WP] is changed.

Impact: Write protect WP# can't be toggled in NVDDR mode.

Workaround: WP# (CSPRn[WP]) value can only be changed when NAND device is in asynchronous SDR mode. To toggle the value of WP# when device is in synchronous NVDDR mode, use the following steps:

1. Transition the NAND device to the asynchronous mode by issuing reset command (Refer to IFC RM section "Switching to the asynchronous interface").
2. Change the value of WP# in corresponding CSPRn[WP] register per the requirement.
3. Transition the NAND device back to NVDDR mode using set feature command (Refer to IFC RM section "Activating the NVDDR Interface").
4. Clear NAND_EVTER_STAT[WPER] bit if it is set.

Fix plan: No plans to fix

A-008312: Duplicate edge-triggered interrupt after priority re arbitration

Affects: MPIC

Description: There is an occurrence of duplicate interrupt when an edge-triggered interrupt higher in priority comes closely to any other enabled interrupts. The following is the sequence of events that leads to the duplicate edge-triggered interrupt:

1. An active interrupt is waiting for acknowledgement
2. An edge-triggered interrupt of higher priority triggers closely to the lower priority interrupt just when it is acknowledged
3. The higher priority edge-triggered interrupt supersedes and fires a new interrupt to the core
4. The core acknowledges the higher priority interrupt without clearing the pending state and finishes the interrupt service routine with EOI
5. A duplicate of the higher priority edge-triggered interrupt is triggered because of the uncleared pending state

Impact: Enabling any edge-triggered interrupts higher in priority than other enabled interrupts may lead to the duplicate edge-triggered interrupt. This includes edge-triggered IRQs, global timers and IPI.

Workaround: Choose one of the following workarounds based on the interrupt type:

- Configure the higher priority interrupts as level-sensitive only
 - a. In case of IRQs this can be configured in the Vector/Priority Register.
 - b. It is not an option for global timers or IPI.
- Any enabled edge-triggered interrupts must be no higher in priority than the other enabled interrupts.

Fix plan: No plans to fix

A-008665: A PBL wait command with a delay value greater than 1fff00h causes the device to stick in the reset state

Affects: PBL

Description: When using I2C as a RCW/PBI source, a PBL wait command with a delay value greater than 1fff00h causes the device to stick in the reset state.

Impact: The device does not come out of reset when the delay value exceeds 1fff00h.

Workaround: The PBL wait command must have a delay value less than 1fff00h when using I2C as a RCW/PBI source.

Fix plan: No plans to fix

A-007189: PCI Express inbound error message handled incorrectly in RC mode

Affects: PCIe

Description: According to Figure 6-3 of the PCI Express base specification REV 3.0, Inbound errors reported to PCI Express RC:

- In its Root Error Status Register can be gated by the "SERR# Enable" bit in the PCI Express Command Register and the "Error Reporting Enables" bit fields (the URR, FER, NFER, or CER bit) in the Device Control Register.
- In its PCI Express Status Register can be gated by the "SERR# Enable" bit in the PCI Express Command Register

However, due to this error, the PCI Express controller in RC mode does not allow the error status reporting in both the PCI Status Register and the Root Error Status Register to be controllable by the PCI Express Command Register [SERR# Enable] bit and the Device Control Register [URR, FER, NFER, CER] bits.

Impact: The inbound errors, including both detected by the PCI Express RC itself and from externally-received error messages, are directly reported to both the PCI Express Status Register [Signaled System Error] bit and the Root Error Status Register [FEMR, NFEMR, FUF, MEFNFR, EFNFR, MECR, ECR] bits, regardless of the setting of the PCI Express Command Register [SERR# Enable] bit or the Device Control Register [URR, FER, NFER, CER] bits.

However, for externally-received error messages, the error status reporting in both the PCI Express Status Register [Signaled System Error] bit and the Root Error Status Register [FEMR, NFEMR, FUF, MEFNFR, EFNFR, MECR, ECR] bits can still be controlled by the Bridge Control Register [SERR_EN] bit as normal, when propagating the error messages from the secondary to primary side of the root port.

Workaround: Although the PCI Express RC error status reporting in both the PCI Status Register and the Root Error Status Register cannot be controllable by the PCI Express Command Register [SERR# Enable] bit and the Device Control Register [URR, FER, NFER, CER] bits, the system software can still use the Root Control Register [SEFEE, SENFEE, SECEE] bits and Root Error Command Register [FERE, NFERE, CERE] bits to control the interrupt generation, for the inbound errors reported in the Root Error Status Register [FEMR, NFEMR, FUF, MEFNFR, EFNFR, MECR, ECR] bits.

Note that the inbound errors reported in the PCI Express Status Register [Signaled System Error] bit does not trigger an interrupt.

Fix plan: No plans to fix

A-007758: False window crossing error behavior when accessing the last byte of a PCI Express outbound ATMU window

Affects: PCIe

Description: When a PCI Express outbound transaction attempts to access the last byte of a PCI Express outbound ATMU window, the logic falsely detects an outbound window crossing error and discards the transaction.

Impact: PCI Express outbound transactions that attempt to access the last byte of a PCI Express outbound ATMU window falsely trigger an outbound window crossing error and are discarded.

Workaround: Software can choose to either:

- Double the size of the outbound ATMU window to prevent accessing the last byte of the intended, smaller window size
- Prevent using the last byte of the window

Fix plan: No plans to fix

A-007815: The read-only-write-enable bit must be cleared to prevent overwriting read-only registers

Affects: PCIe

Description: The PCI Express controller features overwrite capability for read-only registers within its configuration space. This capability is turned on by default, which might cause read-only registers in the PCI Express configuration space to be overwritten unintentionally.

Impact: The read-only registers in the PCI Express configuration space may be inadvertently overwritten, which might cause system to malfunction.

Workaround: To prevent unintentionally overwriting to read-only registers in the PCI Express configuration space, the DBI_RO_WR_EN bit (bit 0) of the DBI Read-Only Write Enable Register at PCI Express configuration space offset 8BCh must be cleared.

Since system software might have a need to perform overwrite to certain PCI Express configuration space read-only registers during initialization, it's recommended to perform this clear action to the DBI_RO_WR_EN bit at the end of the Pre-Boot Initialization (PBI) process.

If the local software (for example, boot loader) has a need to utilize this overwrite capability after PBI, the software must clear the DBI_RO_WR_EN bit after finishing the overwrite task.

For PCI Express EP implementation, the EP's local firmware like boot loader, must ensure that the DBI_RO_WR_EN bit is cleared before setting the CFG_READY bit in the PEX_CONFIG register to avoid remote RC software from overwriting EP's read-only registers unintentionally.

NOTE

The PCI Express specification requires that PCI Express EP must be able to accept RC's configuration cycles (not configure retry) within 100 ms after the de-assertion edge of the PCI Express slot reset, which implies that the link must finish training and the CFG_READY bit must be set within 100 ms.

Similar to A-007941, if there is any link-down, hot reset or FLR event occurred, the above registers fail to retain their updated value. In order to retain the updated value, a PCI Express link-down-and-link-up interrupt handler has to be developed for both bootloader and operating system (OS) environment running locally to re-initialize these registers. Refer to the workaround section of A-007941 for the details on how to implement the required interrupt handler.

NOTE

The above workaround does not address the impact caused by FLR event, since the FLR is currently not supported. Refer to A-008002 for further detail.

Fix plan: No plans to fix

A-007941: Device ID and Revision ID registers fail to retain their updated values in EP mode after link-down or hot reset event

Affects: PCIe

Description: When the PCI Express controller is configured in EP mode, during a link-down or hot reset event, the following registers fail to retain their updated values:

- Device ID registers (located at PCI Express EP controller's Type 0 configuration header offset 0x02)
- Revision ID registers (located at PCI Express EP controller's Type 0 configuration header offset 0x08)

NOTE

The above registers are defined by the PCI Express base specification as read-only registers.

In some EP application, there might be some desire to alter the reset default value and update them with values appropriate for the end product application. To support such requirement, the device features an overwrite capability for read-only registers within the configuration space of the PCI Express controller in either RC or EP mode. This can be enabled by setting the DBI_RO_WR_EN bit (Bit 0) of the DBI Read-Only Write Enable Register at the PCI Express configuration space offset 0x8BC, before writing to any read-only register within the controller's configuration space.

Although the read-only registers within the configuration space can be overwritten by software running either locally or at the remote PCI Express RC, this overwrite capability can only be enabled by software running locally to PCI Express controller. In other words, if the PCI Express controller is configured in EP mode, the remote PCI Express RC cannot enable this PCI Express EP controller's overwrite capability.

Refer to A-007815 for further detail regarding the overwrite capability feature.

Impact: When the PCI Express controller operates in EP mode after a link-down or hot reset event, the Device ID and Revision ID registers fail to retain their updated values. This may cause improper identification to the PCI Express EP controller.

Workaround: As mentioned in the description section, if the end product application has any desire to alter the default content of these read-only registers, the overwrite capability for read-only configuration registers must be used, which is a normal feature irrelevant to this erratum. It is assumed that the end product system utilizes the overwrite capability to initialize these read-only registers correctly during Pre-boot Initialization (PBI) stage. As such, the following workaround focuses on how to retain the updated values of these read-only registers after link-down or hot reset event.

In order to retain the updated values, a PCI Express link-down-and-link-up interrupt handler has to be developed for both bootloader and operating system (OS) environment running locally to cover the following possible scenarios:

- If the link goes down during the bootloader stage, the above interrupt handler has to be called and re-initialize these read-only registers with their original updated values.
- If the link goes down after the control is passed from the bootloader (like uBoot) to OS (like Linux kernel), a similar interrupt handler must be called to re-initialize these read-only registers with their original updated values.

System and software developers must take precaution to cover any uncertainty when the control is transitioning from bootloader to OS.

The following sequence represents guidelines for implementing the PCI Express link-down-and-link-up interrupt handler. It uses the bootloader stage as an example. The OS handler implementation should be the same.

1. Assume these read-only registers have been programmed with their updated value during the PBI stage.
2. After POR (Power-On Reset), the PCI Express link should be up after link training. Once the PCI Express initialization is done, the bootloader should clear all the false error bits in the PEX_PME_MES_DR register and other error status registers in PCI Express controller's memory-mapped space and configuration space. The bootloader should then program the PEX_PME_MES_DISR and PEX_PME_MES_IER registers to enable at least the detection and interrupt generation for both link-down and link-up events. The handler should be activated to function at this time.
3. If any link-down event occurs, the handler should be called to service. It should only set a flag at this time to remember that a link-down event has occurred. Once the flag is set, the handler should clear PEX_PME_MES_DR [LDD] bit and the related interrupt. The handler should then wait for the link-up event to occur.
4. After the link comes back up due to a link-up event, the handler should be called quickly to service. Within the handler, it should check whether the link-down flag has been set. Once confirmed, this is an indication that the system experienced a link-down followed by a link-up event sequence. The handler can now proceed to program the original updated value back to these read-only configuration registers. After the register re-initialization task is finished, the handler can clear the link-down flag, the PEX_PME_MES_DR [LUD] bit and the related interrupt. This allows the handler to be called and service in the next round of link-down followed by link-up event sequence. It's important to note that the register re-initialization task shall only be carried out after the link comes back up.

The handler has to ensure that the overwrite capability for read-only registers is enabled locally by setting the DBI_RO_WR_EN bit (Bit 0) of the DBI Read-Only Write Enable Register at PCI Express configuration space offset 0x8BC, before writing to read-only registers in the configuration space. To prevent unintentional overwrite by software, the local handler should minimize the timing window of making DBI_RO_WR_EN bit enabled. In other words, the local handler should only set the DBI_RO_WR_EN bit when it is ready for the overwrite task and must clear the DBI_RO_WR_EN bit right after the overwrite task is finished. Refer to A-007815 for further detail regarding the overwrite capability feature.

Fix plan: No plans to fix

A-007997: PCI Express hot-plug-related bits in the Slot Capabilities Register need to be cleared

Affects: PCIe

Description: The device does not support PCI Express hot-plug functionality. However, the value of the Slot Capabilities Register of the PCI Express controllers indicates that various items related to PCI Express hot-plug are supported.

Impact: Without clearing the bit value of all the PCI Express hot-plug-related bits in the Slot Capabilities Register, it appears that the device supports PCI Express hot-plug while in fact it doesn't.

Workaround: In order to avoid indicating that the device supports PCI Express hot-plug, the following bits in the Slot Capabilities Register need to be cleared during Pre-boot Initialization (PBI):

- Bit 18, No Command Completed Support
- Bit 17, Electromechanical Interlock Present
- Bit 6, Hot-Plug Capable
- Bit 5, Hot-Plug Surprise
- Bit 4, Power Indicator Present
- Bit 3, Attention Indicator Present
- Bit 2, MRL Sensor Present
- Bit 1, Power Controller Present
- Bit 0, Attention Button Present

The above register is defined by the PCI Express base specification as read-only register. The overwrite capability should be turned on by setting the DBI_RO_WR_EN bit (Bit 0) of the DBI Read-Only Write Enable Register at the PCI Express configuration space offset 0x8BC, before writing to any Read-Only register within the controller's configuration space. Refer to the A-007815 for further detail regarding the overwrite capability feature.

Similar to A-007941, if there is any link-down, hot reset or FLR event occurred, the above register fails to retain their updated value. In order to retain the updated value, a "PCI Express link-down and link-up interrupt handler" has to be developed for both bootloader and operating system (OS) environment running locally to re-initialize these registers. Refer to the workaround section of A-007941 for the details on how to implement the required interrupt handler.

NOTE

The above workaround does not address the impact caused by FLR event, since the FLR is currently not supported. Refer to A-008002 for further detail.

Fix plan: No plans to fix

A-007999: Incorrect reset value for the multifunction bit in the PCI Express Header Type Register in RC mode

Affects: PCIe

Description: All PCI Express controllers configured in RC mode do not feature multiple functions. However, for PCI Express controller 1, the multifunction bit (Bit 7) of the PCI Express Header Type Register at the RC's configure space offset Eh is incorrectly set at reset.

Impact: The device appears to support multiple functions in RC mode when in fact it is a single function device.

NOTE

Only the specific PCI Express controller mentioned is affected.

Workaround: When the PCI Express controller 1 is configured in RC mode, with the DBI_RO_WR_EN bit (Bit 0) set in the DBI Read-Only Write Enable Register at PCI Express configuration space offset 8BCh, clear the multifunction bit (Bit 7) of the PCI Express Header Type Register to indicate that the RC controller is a single function device. The value in RC mode for the Header Type Register should be 01h.

For best results, perform the workaround during the pre-boot initialization (PBI) process.

Similar to A-007941, if there is any link-down, hot reset or FLR event that occurs, the above register fails to retain its updated value. In order to retain the updated value, a PCI Express link-down and link-up interrupt handler has to be developed for both bootloader and operating system (OS) environment running locally to re-initialize these registers. Refer to the workaround section of A-007941 for the details on how to implement the required interrupt handler.

NOTE

The above workaround does not address the impact caused by FLR event, since the FLR is currently not supported. Refer to A-008002 for further details.

Fix plan: No plans to fix

A-008002: FLR is not supported despite the value that the Device Capabilities Register indicates for the SRIOV-capable PCI Express controller in EP mode

Affects: PCIe

Description: Although the Device Capabilities Register indicates (bit 28, FLR capability = 1) that the device is capable of FLR (function level reset), it is not supported and must not be used. In other words, system software must not write a value of 1b to the bit 15 (IFLR, initiate an FLR) of the Device Control Register for any PF or VF in the SRIOV-capable PCI Express EP controller.

Impact: The device does not function correctly if FLR is issued to SRIOV-capable EP controller's PF or VF.

NOTE

- Only PCI Express controller 1 in EP mode is affected, because it is the only PCI Express controller intended to support FLR in EP mode.
- PCI Express controllers configured in RC mode are not affected, because the FLR is not applicable in RC mode.

Workaround: Instead of issuing FLR, the upstream PCI Express RC controller can use hot reset instead, although hot reset will reset the whole PCI Express EP controller in use.

Fix plan: Fixed in Rev 1.1

A-008027: Incorrect values in the VF Device ID registers for both PF0 and PF1 of PCI Express controller 1 in SR-IOV EP mode

Affects: PCIe

Description: When the PCI Express controller 1 is configured to operate in SR-IOV EP mode, the VF Device ID register at each PF's configuration space offset 0x192 for both PF0 and PF1 contains the following incorrect values:

- PF0's VF Device ID register value = 0x0000
- PF1's VF Device ID register value = 0x1957

Impact: When the PCI Express controller 1 is configured to operate in SR-IOV EP mode, the VF Device ID registers for both PF0 and PF1 return incorrect values upon read. This may cause improper identification to the VFs.

NOTE

Only PCI Express controller 1 operating in SR-IOV EP mode is affected. All other PCI Express controllers are not SR-IOV-capable and are not affected.

Workaround: There are two aspects of the problem related to the workaround:

1. Determine the appropriate value to update, and update each PF's VF Device ID Register. According to, *Single Root I/O Virtualization and Sharing Specification, Revision 1.1*, all the VFs within a PF share the same VF Device ID, which may be different from its PF's Device ID located at PF's configuration space offset 0x02. As such, the software running on the SR-IOV EP may choose to update each PF's VF Device ID register with either one of the following:
 - The same value from its PF's Device ID register (normally, this is the preferred approach), or
 - A different value suitable for the end product application.

Because the PF's VF Device ID is a read-only register, before updating its value, the DBI_RO_WR_EN bit (Bit 0) of the DBI Read-Only Write Enable Register at PCI Express configuration space offset 0x8BC must be set. The register update should be performed during pre-boot initialization (PBI) stage to avoid the incorrect value in the VF Device ID register being read by remote PCI Express RC during initialization.

2. Retain the updated value.

Similar to A-007941, if there is any link-down, hot reset or FLR event occurred, each PF's VF Device ID register fails to retain their updated value. In order to retain the updated value, a PCI Express link-down-and-link-up interrupt handler has to be developed for both bootloader and operating system (OS) environment running locally to re-initialize each PF's VF Device ID register. Refer to the workaround section of A-007941 for the details on how to implement the required interrupt handler.

NOTE

The above workaround does not address the impact caused by FLR event, since the FLR is currently not supported. Refer to A-008002 for further detail.

Fix plan: Fixed in Rev 1.1

A-008053: The PCI Express replay timer value needs to be increased

Affects: PCIe

Description: The default replay timer time-out value for the PCI Express controller is too small which results in unnecessary correctable errors in x1 and x2 configurations.

Impact: Correctable errors are logged with the PCI Express controller's default replay timer time-out value in x1 or x2 configuration.

Workaround: To avoid unnecessary correctable errors, the replay timer time-out value (Bits [18:14]) in the register at configuration space offset 718h should be changed to 0_1111b. To do so, perform the following sequence in the order listed:

1. Perform a configuration read to the PCI Express controller's configuration space offset 718h. Save the read return value to a temporary location.
2. AND the temporary value obtained in Step 1 with FFF8_3FFFh and save the result.
3. OR the result from Step 2 with 0003_C000h and write back the result to the same configuration space offset 718h.

Fix plan: No plans to fix

A-008073: The PCI Express controller's next capability pointers are incorrect in some cases

Affects: PCIe

Description: The next capability pointers (offsets) for PCI Express controller 1 are incorrect in some cases.

- In RC mode, the next capability pointers (offsets) may point to the SR-IOV Extended Capability structure that is not applicable for RC, as indicated in the following capabilities registers:
 - Register at configure space offset 100h = 1582_0001h, Advanced Error Reporting Capability structure with Next Capability Offset = 158h,
 - Register at configure space offset 158h = 1781_0019h, Secondary PCI Express Capability structure with Next Capability Offset = 178h,
 - Register at configure space offset 178h = 0001_0010h, SR-IOV Capability structure with Next Capability Offset = 000h (NULL)
- In EP mode, PF1's next capability pointers (offsets) may point to the Secondary PCI Express Capability structure that is not applicable for PF1, as indicated in the following capabilities registers:
 - Register at configure space offset 100h = 1482_0001h, Advanced Error Reporting Capability structure with Next Capability Offset = 148h,
 - Register at configure space offset 148h = 1581_000Eh, Alternate Routing ID (ARI) Capability structure with Next Capability Offset = 158h
 - Register at configure space offset 0x158 = 1781_0019h, Secondary PCI Express Capability structure with Next Capability Offset = 178h,
 - Register at configure space offset 178h = 0001_0010h, SR-IOV Capability structure with Next Capability Offset = 000h (NULL)

For all other PCI Express controllers, the next capability pointers may point to the Secondary PCI Express Extended Capability structure even when the controller is configured as non-Gen3 speed via RCW.

Impact: The PCI Express capability structure linked list may erroneously point to a PCI Express extended capability structure that is not applicable in some cases. Such extended capability structure irrelevant to the current controller configuration may cause confusion to the PCI Express enumeration software, if the software can't ignore the extra or irrelevant PCI Express capabilities structure presented in the linked list.

Workaround: Workaround procedure to overwrite the affected capabilities pointers is not feasible since the affected PCI Express capabilities pointer registers cannot be updated. The alternative approach described below can be adopted for the host system software running at RC side to ignore the extra or irrelevant PCI Express capabilities structure presented in the linked list.

- When the PCI Express controller is configured in RC mode, the host system software has to build its PCI Express capabilities linked list based on the information provided below, instead of using the information read from RC configuration space registers.
- When the controller is configured in EP mode, the similar approach can be adopted if the remote host system software can be updated with the EP linked list information provided below prior to PCIe bus enumeration. In case this is not feasible for some system environment, the overall system behavior might be unpredictable, since it totally depends on how the remote host system software behaves when encountering the extra or irrelevant PCI Express capabilities structure presented in EP controller's linked list.

For the rest of section, the speed the controller configured to support refers to the maximum link speed that a PCI Express controller is configured to operate for a chosen SRDS_PRTCL_Sn setting within a RCW.

For PCI Express controller 1, depending on the RCW's setting for the mode and speed required by the end product application, the host system software running at RC side should build its PCI Express capabilities linked list as described below:

1. If the PCI Express controller 1 is configured in RC mode,
 - a. If the controller is configured to support Gen3 speed, host software should ignore the SR-IOV Capability structure presented at the end of the linked list.

The correct linked list should only have the following for the Gen3-capable PCI Express RC controller 1:

- Offset 100h, Advanced Error Reporting Capability structure with Next Capability Offset = 158h →
 - Offset 158h, Secondary PCI Express Capability structure.
- b. If the controller is configured to support Gen2 or Gen1 speed only, host software should ignore the both the Secondary PCI Express Capability and SR-IOV Capability structures presented in the linked list.

The correct linked list should only have the Advanced Error Reporting Capability structure for the non-Gen3-capable PCI Express RC controller 1.

2. If the PCI Express controller 1 is configured in EP mode,
 - a. The remote host software should always ignore the Secondary PCI Express Capability structure presented in PF1's linked list at offset 0x158 regardless what speed the controller is configured to support.

The correct linked list should only have the following for the PCI Express EP controller 1's PF1:

- PF1's offset 100h, Advanced Error Reporting Capability structure with Next Capability Offset = 148h →
 - PF1's offset 148h, Alternate Routing ID (ARI) Capability structure, use Next Capability Offset = 178h instead of 158h in software (or skip the capabilities in offset 158h)→
 - PF1's offset 178h, SR-IOV Capability structure with Next Capability Offset = 000h (NULL)
- b. In addition, if the EP controller is configured to support Gen2 or Gen1 speed only, host software should ignore the Secondary PCI Express Capability structure presented in PF0's linked list at offset 158h.

The correct linked list should only have the following for the non-Gen3-capable PCI Express EP controller 1's PF0:

- PF0's offset 100h, Advanced Error Reporting Capability structure with Next Capability Offset = 148h →
- PF0's offset 148h, Alternate Routing ID (ARI) Capability structure, use Next Capability Offset = 178h instead of 158h in software (or skip the capabilities in offset 158h)→
- PF0's offset 178h, SR-IOV Capability structure with Next Capability Offset = 000h (NULL)

For PCI Express controller 2, 3, and 4, if the speed from the chosen SRDS_PRTCL_Sn setting of a RCW required by the end product application is Gen1 or Gen2 only, regardless the mode is RC or EP, the host system software running at RC side should always ignore the Secondary

PCI Express Capability structure presented at the end of the linked list. The correct linked list should only have the Advanced Error Reporting Capability structure for the non-Gen3-capable PCI Express controller 2, 3 and 4.

Fix plan: No plans to fix

A-008098: PCI Express may corrupt memory when there are excessive correctable errors

Affects: PCIe

Description: The PCI Express controller may fail to discard a bad packet when there are excessive, correctable errors resulting in corrupted memory.

Impact: While running in an environment with many correctable errors the PCI Express controller can corrupt memory.

Workaround: To prevent possible data corruption, update the following registers in the PCI Express controller's configuration space with the little-endian values specified. It's recommended to perform the update during Pre-boot Initialization (PBI) stage.

Register at configure space offset	Value
748h	4020_8080h
74Ch	0020_C002h
750h	0020_0000h

Similar to A-007941, if there is any link-down, hot reset or FLR event occurred, the above registers fail to retain their updated value. In order to retain the updated value, a "PCI Express link-down and link-up interrupt handler" has to be developed for both boot-loader and operating system (OS) environment running locally to re-initialize these registers. Refer to the workaround section of A-007941 for the details on how to implement the required interrupt handler.

NOTE

The above workaround does not address the impact caused by FLR event, since the FLR is currently not supported. Refer to A-008002 for further detail.

Fix plan: No plans to fix

A-008099: PCI Express erroneously flags correctable errors when the link's ASPM is enabled in Gen 3 speed

Affects: PCIe

Description: When the link running at Gen3 speed with its ASPM enabled, the PCI Express controller erroneously flags correctable errors.

Impact: When the PCI Express controller operates in Gen 3 speed with its link's Active State Power Management (ASPM) enabled, there may be false correctable errors reported that can impact performance due to the extra delay when going from L0s through recovery. The correctable errors may also generate interrupts if enabled.

Workaround: If PCI Express ASPM is really desired for the link that this PCI Express controller belongs to when operating at Gen3 speed, interrupt generation from correctable errors should be disabled, given the system is able to tolerate the above mentioned impact.

If a system cannot tolerate the above mentioned impact from false PCI Express correctable errors, the PCI Express ASPM should be disabled for the link that this PCI Express controller belongs to when operating at Gen3 speed in either one of the following options:

- The PCI Express ASPM policy is normally controlled globally by the operating system. If it's feasible, turn off the ASPM support globally in the OS for the whole PCI Express fabric that the device belongs to.
- If ASPM cannot be turned off globally, the ASPM support of the affected device's PCI Express link can be disabled by setting the PCI Express link Control register [ASPM_CTL] = 00b for both the Freescale PCI Express controller and its link partner.

Fix plan: No plans to fix

A-008100: PCI Express erroneously sets the Signaled Target Abort bit in the Status Register in EP mode

Affects: PCIe

Description: In endpoint (EP) mode, the Signaled Target Abort bit in the PCI Express Status Register at configuration offset 0x6 is erroneously set even though no Completer Abort error is generated.

Impact: The PCI Express Status Register may falsely indicate that a Completer Abort error is detected in EP mode. Note that this error bit does not generate any interrupt.

Workaround: Since a PCI Express EP rarely generates Completer Abort error, the Signaled Target Abort bit in the PCI Express Status Register at configuration offset 0x6 can be disregarded.

Fix plan: No plans to fix

A-008236: PCI Express controller does not exit disabled state if the Link Control register [Link Disable] bit is cleared before lanes enter electrical idle in RC mode

Affects: PCIe

Description: When the PCI Express controller is configured in RC mode, software can disable the link by setting the configuration space Link Control register [Link Disable] bit to direct the LTSSM (Link Training and Status State Machine) to the disabled state, which is reflected with the PCI Express controller memory mapped space CSR0 register [LTSSM_SC] = 0x19.

Once the LTSSM is in the disabled state, the lanes enter electrical idle.

Because of this erratum, if the software immediately clears the Link Disable bit, the PCI Express RC controller may fail to detect this action from software.

Impact: The PCI Express RC controller may remain in the disabled LTSSM state and does not move to the detect state required for link retrain later.

Workaround: If software needs to disable and re-enable the link, the following procedure must be followed:

1. After software sets the PCI Express RC controller's Link Control register [Link Disable] bit, it should poll the CSR0 register to confirm that the LTSSM is indeed in the disabled state, with read return value of "LTSSM_SC = 0x19" for three times consecutively.
2. Once the LTSSM is confirmed in the disabled state, software may proceed to clear the Link Control register [Link Disable] bit to bring the link out of the disabled state if it needs to.

Fix plan: No plans to fix

A-008339: PBA offset value in MSI-X PBA Offset register is incorrect and PF1 VFs generate incorrect MSI-X transaction in SR-IOV EP mode

Affects: PCIe

Description: There are two issues related to MSI-X when the PCI Express controller 1 is configured in SR-IOV EP mode:

Issue# 1: PBA offset value in MSI-X PBA Offset register is incorrect.

Before a PCI Express EP can utilize the MSI-X feature, the remote PCI Express RC (host) software has to initialize the EP's MSI-X capability structure. During the initialization, the RC software performs configuration reads to EP's MSI-X Table Offset/BIR register (at each PF or VF's configure space offset B4h) and MSI-X PBA Offset/BIR register (at each PF or VF's configure space offset B8h) for each PF or VF. Both registers are defined as read-only by PCI Local Bus Specification Rev 3.0.

Based on the Table BIR (BAR Indicator Register) bit field return value of these registers, which is 1h for the T4240 and T2080, the RC software accesses the BAR1 register of the PF and VF configure space to find out the PF and VF aperture (size) implemented for PF and VF MSI-X vector table and PBA (Pending Bit Array), which is 8 KB for each PF or VF. The RC software can then proceed to program the BAR1 for each PF and its associated VFs, which becomes the upper address bits of the memory location for the MSI-X Table and PBA for each PF and its associated first VF.

To be able to access the actual MSI-X Table and PBA for each PF or VF, the RC software needs to read the dedicated MSI-X Table Offset/BIR and PBA Offset/BIR registers for each PF or VF and find out the values of MSI-X Table Offset and PBA Offset. Based on the offset values from these two registers and the previously programmed BAR1 value, the RC software is able to calculate the MSI-X Table and PBA address of each PF or VF for accessing their MSI-X vector table and PBA later.

Because of this erratum, the MSI-X PBA/BIR register contains the incorrect MSI-X PBA offset value.

Issue# 2: PF1 VFs generate incorrect MSI-X transaction.

For PCI Express SR-IOV capable EP, each PF or VF has its dedicated MSI-X Capability Structure, which contains the enable bit in the MSI-X Control register. For each PF's VF, setting the enable bit in its dedicated MSI-X Control register should be sufficient for MSI-X transaction generation. Due to this erratum, to generate correct MSI-X transaction for PF1's VFs, setting PF1's MSI-X Control [Enable] bit is required in addition to setting VF's enable bit.

Impact: Issue# 1:

Remote RC software might not be able to access SR-IOV EP's PF or VF's PBA based on the PBA offset bit field read return value obtained from PF or VF's MSI-X PBA Offset/BIR register.

Issue# 2:

PF1 VFs are unable to generate correct MSI-X transaction without setting the PF1's MSI-X Control [Enable] bit. There is no impact to other PF or VF's MSI-X outbound transaction. For example, PF0 VFs can generate correct MSI-X transaction without requiring the PF0's MSI-X Control [Enable] bit to be set.

Workaround: Issue# 1:

Remote RC software should ignore the incorrect PBA offset value returned when reading PF or VF's MSI-X PBA Offset/BIR register. Instead, software should use 1000h as PBA offset

instead. This may not be feasible when the SR-IOV EP card is plugged into a third-party remote RC, like x86 PC host.

The following is the MSI-X table organization of PF0 or PF1 for reference:

Range	Address	Meaning	Note
0000h — 007Fh	0000h	PF's MSI-X table offset	This is the beginning of the PF's MSI-X vector table, pointed to by the base address in PF BAR1.
	007Fh	—	This is the end of the PF's MSI-X vector table for its 8 vectors.
1000h — 1FFFh	1000h	PF's MSI-X PBA offset	This is the beginning of the PF's PBA.
	1FFFh	—	This is the end of the PF's BAR1 aperture with the size of 8 KB.

The following is the MSI-X table organization of all VFs for reference:

Range	Address	Meaning	Note
0000h — 007Fh	0000h	VF1's MSI-X table offset	This is the beginning of the VF1's MSI-X vector table, pointed to by the base address in VF BAR1.
	007Fh	—	This is the end of the VF1's MSI-X vector table for its 8 vectors.
1000h — 1FFFh	1000h	VF1's MSI-X PBA offset	This is the beginning of the VF1's PBA.
	1FFFh	—	This is the end of the VF1's BAR1 aperture with the size of 8 KB.
2000h — 207Fh	2000h	VF2's MSI-X table offset	This is the beginning of the VF2's MSI-X vector table.
	207Fh	—	This is the end of the VF2's MSI-X vector table for its 8 vectors
3000h — 3FFFh	3000h	VF2's MSI-X PBA Offset	This is the beginning of the VF2's PBA
	3FFFh	—	This is the end of the VF2's BAR1 aperture with the size of 8 KB.

Note: Repeat above organization to cover the BAR1 aperture for the remaining VFs from VF3 to VF64.

Issue# 2:

In the PBI, the SR-IOV EP can set the PF1's MSI-X Control [Enable] bit before the remote RC performs enumeration and initialization. Similar to A-007941, if there is any link-down, hot reset or FLR event occurred, the above register fails to retain its updated value. In order to retain the updated value, a "PCI Express link-down and link-up interrupt handler" has to be developed for both bootloader and operating system (OS) environment running locally to re-initialize this register. Refer to the workaround section of A-007941 for the details on how to implement the required interrupt handler.

Fix plan: No plans to fix

A-008405: PCI Express Gen3 equalization cannot complete when lanes are reversed

Affects: PCIe

Description: During PCI Express Gen3 link training equalization process, as equalization master, the controller sends TS1 Ordered Sets with new coefficients to the link partner, which replies back with TS1 Ordered Sets with the same coefficients. The controller checks whether the transmitted and received coefficients match before proceeding further on the equalization handshake process. This can happen at the Phase 3 for RCs downstream port or Phase 2 for EP's upstream port.

If lane reversal is active, the above check fails, because the equalization module does not take lane reversal into account when doing the comparison. As such, the Gen3 equalization cannot complete and times out.

Impact: When PCI Express lanes are reversed, Gen3 equalization cannot complete and times out. This causes the link to eventually settle at Gen1 speed. Therefore, if Gen3 operation speed is desired, lane reversal cannot be used, although T4240 and T2080 support limited lane reversal options as described in section 20.6.1.3, "Lane Reversal" of *T2080 Rev 0 Reference Manual*. This is especially essential in the end product system that has a non-fixed link partner (like a RC controller implemented with slot or EP controller implemented as card that might be plugged into a PCI Express slot of any random host).

This issue is not applicable for Gen1 or Gen2 links.

Workaround: There is no workaround for Gen3 link with lane reversal active. Therefore, the end product board layout must avoid lane reversal if Gen3 operation speed is desired. This is feasible if the link partner is fixed where the link between the RC and EP is directly connected with PCB trace.

In end product system with non-fixed Gen3 link partner, there might be some situation that the link reversal is caused by the board layout of a random link partner, such that the overall lane reversal is unavoidable for the link. If such condition happens, the only workaround is to change the RCW [SRDS_DIV_PEX_Sn] setting in PBI from 00b to either one of the following settings:

- 01b -> This limits the maximum speed of the link to Gen2 (5GT/s), or
- 10b -> This limits the maximum speed of the link to Gen1 (2.5 GT/s)

Fix plan: No plans to fix

A-008851: Invalid transmitter/receiver preset values are used in Gen3 equalization phases during link training for RC mode

Affects: PCIe

Description: During PCI Express Gen3 link training equalization Phase 0 and Phase 1, if the PCI Express controller is configured in RC mode, the values of the following bit fields in the Lane Equalization Control Register (16-bit little endian read-only register for each lane, starting at RC's configure space 164h for lane 0 to 172h for lane 7, depending on number of lanes in use for the controller) are used initially to communicate the Transmit Preset and Receiver Preset Hint (optional) for each lane between RC's downstream port and link partner's upstream port:

- Bits [14:12], Upstream Port Receiver (Rx) Preset Hint
- Bits [11:8], Upstream Port Transmitter (Tx) Preset
- Bits [6:4], Downstream Port Receiver (Rx) Preset Hint
- Bits [3:0], Downstream Port Transmitter (Tx) Preset

Currently, the value of the above bit fields in the read-only Lane Equalization Control Registers for each lane are implemented as below:

- Receiver (Rx) Preset Hint = 7h (same for Bits [14:12] and Bits [6:4]), and,
- Transmitter (Tx) Preset = Fh (same for Bits [11:8] and [3:0])

Therefore the values of the register for each lane are 7F7Fh. However, according to PCI Express Base Specification Rev 3.0, Table 4-4 and 4-3, the above encoding values are reserved, although these values are shown as the default in the bit field definition within the same specification's section 7.27.4, "Lane Equalization Control Register."

Impact: Due to the ambiguity in the PCI Express Base Specification Rev 3.0, when communicating invalid preset values during Gen3 link training equalization process, confusion might be caused to the link partner. It may result in Gen3 equalization to be concluded with non-optimal settings.

This issue is not applicable for Gen1 or Gen2 link. It's also not applicable for EP mode.

Workaround: The recommended values are 4h for the optional Receiver (Rx) Preset Hint and 7h for the Transmitter (Tx) Preset. In PBI, if the PCI Express controller is configured as Gen3 RC mode, each lane's Lane Equalization Control Register of the controller should be programmed with the value of 4747h.

Since the register mentioned above is defined by the PCI Express base specification as a read-only register, the overwrite capability should be turned on by setting the DBI_RO_WR_EN bit (Bit 0) of the DBI Read-Only Write Enable Register at the PCI Express configuration space offset 8BCh, before writing to any read-only register within the controller's configuration space.

Fix plan: No plans to fix

A-008913: New A11 message request handling rules not supported

Affects: PCIe

Description: As defined in the section 2.3.1, Request Handling Rules of the PCI Express Base Specification Rev 3.0, a received Message TLP can be treated as a valid request as long as the Message Code field of the TLP header is valid.

The Errata for the PCI Express Base Specification Rev 3.0 released on March 31, 2013 revises the above message request handling rules. Before a received message can be considered as a valid request, the new A11 request handling rules require checking the valid Message Routing subfield (Type [2:0] bits) and Msg/MsgD indication (Fmt [1:0] bits) within the Message TLP header in addition to checking the previous Message Code field.

The errata described above for the PCI Express Base Specification Rev 3.0 has been integrated into the PCI Express Base Specification Rev 3.1, released on October 8, 2014.

Nevertheless, since the PCI Express controller core follows the requirement of the PCI Express Base Specification Rev 3.0, it does not support the new A11 request handling rules defined in PCI Express Base Specification Rev 3.0 errata.

Impact: This might lead to inappropriate handling for some received Message TLP generated by some third-party link partner with invalid Message Routing subfield (type [2:0] bits) and Msg/MsgD indication (Fmt [1:0] bits) within the Message TLP header. For example:

- Inbound Message TLP with invalid Message Routing subfield (type [2:0] bits) within the TLP header is still treated as a valid message that is processed normally.
- Inbound Message TLP with invalid Msg/MsgD indication (Fmt [1:0] bits) within the Message TLP header is detected as malformed TLP and discarded. Since Message TLPs are posted TLPs, there is no UR completion TLP returned to link partner for such invalid message requests.

Workaround: None. The link partner should constrain itself to ensure that the Message TLPs sent out contain valid encoding for both the Message Routing subfield (type [2:0] bits) and Msg/MsgD indication (Fmt [1:0] bits), in addition to the Message Code field, within the Message TLP header.

Fix plan: No plans to fix

A-009151: PCI Express controller does not correctly set the Link Autonomous Bandwidth Status and Link Bandwidth Management Status bits of the Link Status Register in RC mode

Affects: PCIe

Description: The Link Status Register Bit 15, Link Autonomous Bandwidth Status (LABS) and Bit 14, Link Bandwidth Management Status (LBMS) are only meaningful for PCI Express RC's downstream port. The PCI Express base specification requires the RC downstream ports to:

- Set the LABS bit when the link width or speed is changed by downstream link partner for autonomous reasons only, not for reason of attempting to correct unreliable link operation
- Set the LBMS bit when the link width or speed is changed due to reliability issues

Due to this erratum, the PCI Express RC controller behaves incorrectly as below:

- Does not set the LABS bit when the link width or speed is changed by downstream link partner for autonomous reasons
- Does not set the LBMS bit when the link width or speed is changed due to reliability issues
- Set the LBMS bit when the link width or speed is changed by the downstream link partner for autonomous reasons, which should be the function of LABS bit

Impact: The system software cannot utilize the PCI Express RC controller's Link Status Register [LABS, LBMS] bits to determine the cause of the possible link width and/or speed change, especially for test and debug purpose.

Workaround: Software should not utilize the functionality of the PCI Express RC controller's LABS and LBMS bits of the Link Status Register.

Fix plan: No plans to fix

A-009220: PCI Express controller upstream port does not advertise 8 GT/s data rate after receiving Gen3 equalization (EQ) TS Ordered Sets in EP mode

Affects: PCIe

Description: According to the PCI Express Base Specification Rev 3.0, a Gen3 capable PCI Express EP controller (initially linked up at either Gen1 or Gen2 speed) must advertise 8.0 GT/s data rate support after receiving eight consecutive Gen3 equalization Ordered Sets (either EQ TS1 or EQ TS2) from the downstream port of its link partner.

The Gen3-capable PCI Express EP controller refers to a PCI Express EP controller currently reporting its Link Capabilities Register [Maximum Link Speed] as 0011b.

Due to this erratum, if the EP controller's Link Control 2 Register [Target Link Speed] bit field is set to either 0001b (2.5 GT/s) or 0010 (5.0 GT/s) the PCI Express EP controller (operating at Gen1 or Gen2 speed) does not advertise the required 8 GT/s data rate when the upstream link partner requests a speed change re-training toward Gen3.

Impact: The link is not linked up at Gen3 speed. Note that the failing condition is not a normal operation scenario. The remote RC/host software only updates the Link Control 2 Register [Target Link Speed] bit field of a PCI Express EP controller to a value different from the setting of the EP's Link Capabilities Register [Maximum Link Speed] bit field during the compliance testing.

Workaround: Software should ensure that the Link Control 2 Register [Target Link Speed] bit field of a PCI Express EP controller is untouched from the default value during normal operation, or configure it back to the same setting shown in the EP's Link Capabilities Register [Maximum Link Speed] bit field after finishing testing.

Fix plan: No plans to fix

A-009518: PCI Express EP controller's non-D0 PowerState is overridden to D0 when any single or combination of the Bus Master Enable (BME) or Memory Space Enable (MSE) bits are enabled from disabled setting by host software

Affects: PCIe

Description: When the PCI Express EP controller's PMCSR [PowerState] is set to non-D0 (including D1, D2 or D3), host software's update to any single or combination of the BME or MSE bits in EP's Command Register should not cause the EP's PowerState to settle at D0.

Due to this erratum, when the EP's PowerState is at non-D0, if the host software enables BME and/or MSE bits from their disabled setting (writing an 1b to a bit when its previous value is 0b), the EP controller's PowerState is overridden to D0.

Impact: Instead of remaining in Non-D0, PCI Express EP controller's PMCSR [PowerState] bit field is overridden to D0 due to the host software's action of enabling any single or combination of the BME or MSE bit fields in EP's Command Register from their disabled setting. As the result, the link will exit from L1.

However the impact should be small, since host software should not update the EP controller's Command Register when the EP is in non-D0 PowerState. Instead, it should update EP's PMCSR [PowerState] first.

Workaround: When the PCI Express EP controller's PMCSR [PowerState] is in non-D0, host software should not issue configure cycles to write to EP Command Register's BME or MSE bit fields.

Fix plan: No plans to fix

A-008088: e6500 does not exit PW10/PW20 after core warm reset

Affects: PM

Description: When the e6500 core is in the power management state PW10/PW20, it does not exit PW10/PW20 after a core warm reset is issued.

Impact: The core stays in PW10/PW20 state after a core warm reset instead of returning to normal running state PH00.

Workaround: Software should take the cores out of PW10 or PW20 power management state before requesting core warm reset.

Fix plan: No plans to fix

A-009300: When a management command is performed to the Logical FQ Mapping Table while enqueues to the CEETM class queues are in progress, it could result in frames enqueued to the wrong class queue

Affects: QMAN

Description: Back-to-back cycles of access to the Logical FQ Mapping Table (like look-up for an enqueued request to CEETM, followed by a CEETM Logical FQ Mapping Table Configure/Query command) could cause the enqueued frame to be queued in the wrong class queue.

Impact: CEETM Logical FQ Mapping Table (LFQMT) Configure or Query commands are not usable when enqueues are in progress.

Workaround: Do not use the CEETM Logical FQ Mapping Table (LFQMT) Configure or Query commands while enqueues to CEETM are in progress. All LFQMT configuration should be done before traffic is sent to CEETM.

Fix plan: No plans to fix

A-009473: PFDR reserved by order restoration lists (ORL) and notification message queues not accounted for in PFDR free pool count and low watermark threshold

Affects: QMAN

Description: When QMan adds deferred enqueues to an order restoration list (ORL), or adds enqueue rejection notification (ERN) messages or FQ state change notification messages (FQRN, FQRNI, FQRL, FQPN) to an ERN queue destined to the message ring of a software portal, each item added to the list or queue causes a PFDR to be reserved, but not yet used. These reserved PFDRs are in addition to any PFDRs that may be used by the lists or queues themselves. These reservations are tracked by an internal counter, and the number of reserved PFDRs is used to determine PFDR depletion, which triggers an interrupt (PEBI) and blocks enqueues when $PFDR_FPC[FPC]$ is less than or equal to $(PFDR_CFG[K] + \text{number of reserved PFDRs})$.

However the number of internally reserved PFDRs is not visible to the user, and is not reflected in the free pool count $PFDR_FPC[FPC]$, nor is it used in generating the PFDR low watermark interrupt (PLWI). Therefore a user who encounters PFDR depletion (PEBI interrupt and blocked enqueues) caused in whole or in part by a large number of internally reserved PFDRs may not realize this because reading the PFDR free pool count shows that sufficient PFDR are still available, since the free pool count does not reflect the internally reserved PFDRs.

Impact: A user may not realize that internally reserved PFDRs are the cause of PFDR depletion.

Workaround: To get an accurate value of PFDR free pool count and low watermark threshold. Drain the order restoration points (ORPs) and software portal message rings which contain deferred enqueues or messages, and the internal PFDR reservation count drops to 0.

Fix plan: No plans to fix

A-010383: CEETM Effective Packet Size used to update shaper credits is incorrectly calculated when using both Overhead Accounting Length and Minimum Packet Size

Affects: QMAN

Description: When a frame is dequeued by the CEETM, the effective packet size is a function of the actual packet size, the configured Overhead Accounting Length (OAL), and the configured Minimum Packet Size (MPS). This effective packet size is used to update the CEETM shaper credits. The MPS value should be applied to the actual packet size before the OAL value is applied. However, the OAL value is applied to the actual packet size first. Then, the MPS is applied. This results in an incorrect value being subtracted from the CEETM credit for that particular Logical Network Interface (LNI) shaper.

Impact: If OAL and MPS are used together on an LNI shaper, then the credit value tracked for a CEETM shaper will not be accurate. This results in incorrect traffic shaping.

Workaround: Do not use both OAL and MPS together on an LNI shaper.

Fix plan: No plans to fix

A-002518: RapidIO Type 9 data streaming segmentation interleaving rule violated by RMan

Affects: RMAN

Description: RMan supports data streaming via RapidIO Type 9 packets. One of the features of RMan for high performance operation is segmentation interleaving, which allows the message manager to interleave segments from two or more transactions with the same destination device ID (regardless of type). RapidIO Interconnect Spec Rev. 2.1, Part 10 Section 3.2.5 adds clarification to the segmentation rules. Rule 1 states "No more than one PDU from a given stream shall be segmented at a time" and "No more than one PDU from a given RapidIO flow shall be segmented at a time". RMan was implemented with an 8-bit CoS field in the Type 9 Outbound PDU descriptor. This CoS field is used to differentiate between two different virtual stream IDs for outbound segmentation interleaving purposes which violates Rule 1.

Impact: A non-NXP target device may not complete Type 9 reassembly correctly when segmentation interleaving is performed on two PDUs with the same flow but different CoS.

Workaround: No workaround is required if all Type 9 traffic is between two NXP devices. For interoperability of Type 9 traffic with non-NXP devices: Option 1 (preferred): Issue the same CoS for Type 9 Outbound PDUs that belong to the same flow to avoid interleaving by the source. Option 2 (if CoS per flow cannot be controlled): Disable segmentation interleaving by setting MMMR[OSID]=1

Fix plan: No plans to fix

A-008968: RMan Type11 duplicate segment

Affects: RMAN

Description: A Type11 outbound message consists of one to sixteen segments. Duplication of the last transmitted segment for a non-multicast (Message Descriptor[Multi-cast mode]=0) message may occur if the message starts outbound segmentation after a non-multicast Type11 has previously detected a segment response time-out. This only occurs if the messages are assigned to the same segmentation unit.

The most common scenario is for a source A to receive a segment response timeout when a single segment Type11 message is transmitted to an unresponsive destination B. Following this event, source A sends the last segment of the next Type11 message twice to destination C.

Impact: Duplication of a Type11 segment causes destination to receive additional payload. This could lead to one of the following:

- Partial message re-assembled at destination may time-out, causing an incomplete message with error status.
- If a single segment message is duplicated, destination does not detect an error, but receives a duplicate message.
- A segment of message A may be combined with subsequent segment(s) of message B and create a new message C which has a bad payload likely undetectable by the destination at the time of reception.

Workaround: Possible workarounds are listed below, each with their own limitations:

Workaround #1

Use Type9 data streaming protocol if data delivery does not require a response. Type9 is unaffected by the duplication issue and provides a much better performance than Type11 messages. Type9 may cause loss of packets without the sender knowing it if SRIO fabric fails to deliver data. Protocols such as Ethernet TCP/IP using sequence numbers can handle this potential temporary loss of data.

Workaround #2

Use Type11 multicast by setting message descriptor field MD[MM]=1. This has a limit of max 256B of payload. MD[EMG/MG/ML] (MD[Extended Multicast group/Multicast group/Multicast list]) must be set accordingly to indicate a single destination.

Workaround #3

Allocate a segmentation engine, using arbitration group (AG) assignment with RMan Message Manager Segmentation Execution Privilege Register MMSEPR0, per destination. This works under the assumption that sending a duplicate segment to an unresponsive destination (link down), does not have any adverse effects. This first restriction is a limit of 4 AGs (4 potential destinations that can cause time-out). The second restriction is the desired arbitration between messages to these destinations may not be as desired. The details are discussed below:

Workaround #3 - Details

To separate outbound RMan traffic for two destinations onto separate segmentation engines, do the following:

1. Assign RMan sub-portal 0/1 for messages to destination A/B respectively. This implies using a separate frame/work queue for each destination. RMan arbitrates using round-robin between sub-portals to balance traffic between SRIO ports, if used this way

2. Use a different AG on each sub-portal. For example, AG0 (WQ0) for destination A and AG1 (WQ1) for destination B. Unique arbitration groups are needed to designate traffic to a specific segmentation engine.
3. Program engines for arbitration groups using register MMSEPR0. For example, MMSEPR0= 01010202h. This allows destination A to use segmentation engine 0/1 (AG0 only), and destination B to use segmentation engine 2/3 (AG1 only).

Fix plan: No plans to fix

A-009006: Loss of inbound RMan hardware reassembly contexts for Type11

Affects: RMAN

Description: RMan receives inbound messages (up to 4KB in size) as 1 to 16 messages segments. When the first segment of a multi-segment message is received, a hardware context is opened to track reassembly of the entire message in memory. If all segment numbers of a Type11 inbound message are received, then the hardware context is closed and set back into invalid state ready to receive a new message. However, if the last segment of a multi-segment message has a different message length than the original starting message segment, loss of one reassembly hardware context occurs. The most likely source of this event would be a badly generated RapidIO message header or a duplication/loss of a Type11 segment by transmitter.

Loss of one inbound reassembly hardware context occurs for a flow based on the MMRCARn setup. This is detected as inbound reassembly idle (MMSR[IMUB]=0) and open hardware reassembly contexts (MMSR[OHWC]>0). The loss of contexts is compounded by additional errors until the dedicated/generic flow count has been exhausted. An exhausted dedicated/generic flow may result in dropped packets or retried messages.

Impact: One or more losses of inbound hardware reassembly context for Type11 may eventually result in zero contexts available as defined by MMRCARn. When this occurs, Type11 has retried for an exhausted flow or generic context.

Workaround: The following workarounds are available for this issue:

Workaround #1

User may use Type9 data packets if data delivery notification is not required since these are unaffected by the errata.

Workaround #2

User may redirect inbound reassembly errors to a unique Type11 error frame queue by setting MMMR[EFQ]=1. If FD[MFE]=1 and FD[SRT]=0, potential loss of inbound reassembly context may have occurred. User should quiesce RMan by setting MMMR[MMQ]=1 and stop inbound traffic from sender to check if MMSR[OHWC]>0. To reclaim lost reassembly H/W contexts, software must perform a soft reset of RMan as defined by register field MMMR[SR]. A soft reset of RMan does not cause the SRIO link to go down. The Type11 messages in this scenario is retried.

Fix plan: No plans to fix

A-005619: ATAPI non-data commands fail when TTL=0 in command header

Affects: SATA

Description: When the Total Transfer Length (TTL) field in Command Header is set to zero, the SATA controller does not issue non-data ATAPI commands like Test Unit Ready (TUR) to ATAPI drives. For a non-data command no Physical Region Descriptors (PRDs) which are essential for data transfer, are required. As no data is involved the TTL field in the command header is programmed to zero.

Impact: In the previous version of the controller, programming TTL to zero was allowed. This can cause compatibility issues in software.

Workaround: The TTL field needs to be programmed with a value equal to the ATAPI packet length, for example the Test Unit Ready command has a packet length of 12

Fix plan: No plans to fix

A-005637: When a received data packet is smaller than the programmed length in the ATAPI command, the SATA host controller raises a false fatal error

Affects: SATA

Description: When an ATAPI device (for example, a CD or DVD-ROM drive) is connected to the SATA interface, the following sequence may occur:

1. The SATA controller issues an ATAPI command (the 'A' bit in command Header DW3 is set) with a certain value in the *ttl* field (that is, the DW2 of the Command Header).
2. The ATAPI device responds with a data FIS.
3. The length of data in the data FIS is smaller than the value programmed in the *ttl* field.
4. The SATA controller raises the following errors:
 - HSTATUS[DLM]
 - HSTATUS[FE]
 - HSTATUS[DE]
 - SERROR[E]
 - CE
 - DE
5. On receiving a fatal error (HSTATUS[FE]), the SATA driver issues a hardware reset to the device.

The *ttl* field is based on maximum ALLOCATION LENGTH as per the *SCSI command Reference Manual*. The actual data transfer length may be less than the ALLOCATION LENGTH and should not result in an error.

Impact: Frequent resets are observed on the bus.

Workaround: When a fatal error is received along with the Data Length Mismatch error in response to an ATAPI command, perform the following:

1. Set HCONTROL[27].
2. Clear HCONTROL[27].
3. Clear the SERROR[E].
4. Do not issue a hardware reset on the bus.

Fix plan: No plans to fix

A-008077: Intermittent failures on some hard drives

Affects: SATA

Description: Intermittent failures are seen with some hard drives. The host is taking more than 20 DWORDs to send a HOLDAp in response to receiving a HOLDp from the disk during a write-data command. This might cause receive error(R_ERR) from some hard drives.

Impact: There may be intermittent failures during write operation

Workaround: The workaround is to increase the ALIGNp insertion rate. Default value (0xFF) of bits 15:8 of LinkCfg (Offset 0x148) should be replaced with value 0x10. The new ALIGNp insertion rate will guarantee that the run length of consecutive data beats will not exceed 20 DWord.

Fix plan: No plans to fix

A-006385: SEC watchdog timer does not prevent all cases of illogical descriptors from hanging DECOs

Affects: SEC

Description: The SEC block contains a watchdog timer designed to clear errors that have hung a DECO. Although mostly effective, there are cases in which poorly constructed descriptors can hang a DECO from which a watchdog cannot recover.

The SEC's programming model is based on the construction of descriptors, which include commands and, optionally, embedded keys and context. Users are expected to use the SEC's driver (SEC Descriptor Runtime Assembler) to build descriptors; however, because there is considerable innate flexibility in descriptor construction — even with the Descriptor Runtime Assembler — it is possible to create descriptors that appear to follow the rules of descriptor construction but still lead to errors, hangs, or corrupted data.

A few examples of descriptor constructions that can lead to hangs not cleared by the watchdog are:

- Ending a descriptor with a LOAD IMM command.
- Creating descriptors that command the PKHA to unload more data than the RAMs actually store.

Impact: Certain non-standard descriptor constructions can lead to errors, hangs, or corrupted data.

Workaround: To avoid this erratum, it is highly recommended that users model their descriptors after the examples provided in the NXP reference software. If there is a need to create a descriptor for which an example does not exist, do so based on the SEC Programmer's reference manual.

Fix plan: No plans to fix

A-008857: Anti-replay early-rollover error

Affects: SEC

Description: The anti-replay functionality is implemented as a protocol. It's designed to be used by other protocols, and is used by IPsec, DTLS, SRTP, MacSEC and WiMAX. However, this bug only affects the SRTP protocol. The functionality includes optional rollover checking of the sequence number. When rollover checking is enabled, it will incorrectly detect a rollover half-way... that is if it was supposed to detect a rollover at 2^{48} in SRTP, it detects, instead, at $2^{(48-1)}$. If the count is started at or near zero it is highly unlikely that a count of $2^{(48-1)}$ will ever be reached in practice.

Impact: The impact is minor. Our research indicates that even if SRTP were to run at 10x the fastest rates, it would take 47 years for the rollover error to issue prematurely when counts are started at or near zero. This defect is in all implementations of SEC back to ERA4, when the separate Anti-Replay functionality was first introduced as a stand-alone protocol.

Workaround: The only workaround for SRTP is to turn off anti-replay checking if a count of 2^{47} is ever expected. For the stand-alone protocol, the rollover check can be turned off and implemented more correctly by separate DECO commands.

Fix plan: No plans to fix

A-004985: SerDes receiver high-speed data path is short in gain

Affects: SerDes

Description: The SerDes receiver interface for certain protocols can only support up to 17 dB of channel loss.

Impact: The smaller gain limits the amount of channel loss that can be overcome at the receiver input.

The maximum channel loss allowed is 17 dB. If the protocol restricts the channel loss to less than the above limit, the channel loss limit specified in the protocol specification should be followed.

The channel loss for the following protocols/rates impacted by this erratum must be restricted to 17 dB:

- PCI Express 8 Gbaud
- Serial RapidIO 5 Gbaud
- 10GBASE-KR

The following protocols/rates are not impacted:

- PCI Express 2.5 Gbaud and 5 Gbaud
- Serial RapidIO 2.5 Gbaud and 3.125 Gbaud
- XAUI 3.125 Gbaud
- Aurora 2.5 Gbaud, 3.125 Gbaud, and 5 Gbaud
- SATA 1.5 Gbaud and 3 Gbaud
- SGMII (1.25 Gbaud) and SGMII 2.5x (3.125 Gbaud)
- 1000Base-KX
- XFI 10.3125 Gbaud

Workaround: Restrict channel loss to 17 dB for the impacted protocols.

Maximum allowed total channel loss budget (channel + interconnect + other loss) at Nyquist rate is 17 dB. Some provision has to be made for channel cross talk and reflection. For example, XFI protocol recommends the channel crosstalk and reflection margin of 3.6 dB.

Fix plan: No plans to fix

A-007186: SerDes Ring VCO does not maintain lock throughout specified temperature range

Affects: SerDes

Description: SerDes PLL is calibrated at reset. When the junction temperature delta from the time the PLL is calibrated exceeds +56°C/-66°C when using XnV_{DD} of 1.35 V, there is a possibility for jitter to increase and cause the PLL to unlock. No issues are seen with LC VCO. Only the protocols using Ring VCOs are impacted.

The following table shows the impacted protocols and rates.

Table 7. Summary of protocols and rates

Protocol	Bit Rate (Gbps)	VCO Frequency (GHz)	VCO	Workaround
SATA	3.0	3.00	Ring	2
	1.5	3.00	Ring	2
SRIO	3.125	3.125	Ring	2
	5.0	5.00	Ring or LC	1
	2.5	5.00	Ring or LC	1
XAUI	3.125	3.125	Ring	2
Aurora	5.0	5.00	Ring or LC	1
	2.5	5.00	Ring or LC	1
PEX	8.0	4.00	LC	Not impacted
	5.0	5.00	Ring or LC	1
	2.5	5.00	Ring or LC	1
SGMII	3.125	3.125	Ring	2
	1.25	5.00	Ring or LC	1
HiGig	3.125	3.125	Ring	2
HiGig2	3.75	3.75	Ring	2
XFI	10.3125	5.15625	LC	Not impacted
10GBase-KR	10.3125	5.15625	LC	Not impacted

Impact: Once the SerDes PLL unlocks, no further communication on that SerDes link is possible until the PLL is reset and allowed to re-lock.

Workaround: Choose one of the following workaround options:

OPTION 1:

Table 8. SerDes options

SerDes 1 Options	
SRDS_PRTCL_S1	Alternate SRDS_PRTCL_S1
1C	1B
51	50

Table continues on the next page...

Table 8. SerDes options (continued)

SerDes 1 Options	
SRDS_PRTCL_S1	Alternate SRDS_PRTCL_S1
5F	5e
65	64
67	66
6B	6A
6D	6C
71	70
83	82
8F	8E
95	94
A4	A4 ¹
A6	A6 ²
AB	AA
BC	BC ¹
CB	CA
D3	D2
D6	D6 ²
D8	D8 ²
D9	D8 ²
DB	DA
DE	DE ¹
E0	E0 ¹
F2	F2 ¹
SerDes 2 Options	
SRDS_PRTCL_S2	Alternate SRDS_PRTCL_S2
02	1
16	15
18	17
1F	1F ³
2E	2D
36	35
<p>1. For options with Gen 3 PCIe, PLL2 is automatically enabled during speed switch. Therefore, the system must supply a reference clock to PLL2. The RCW field SRDS_PLL_REF_CLK_SEL_S1 must be configured to 100 MHz or 125 MHz to apply the reference clock to PLL2.</p> <p>2. Using PBI command, set SRDS1PLL1CR1= 800_4100h</p> <p>3. Using PBI command, set SRDS2PLL2CR1= 800_4100h</p>	

OPTION 2:

The OPTION 2 workaround requires factory pre-set SerDes calibration values to be read from a fuse block. These values have been shown to work across the entire temperature range for all SerDes. These values are then written into the SerDes registers to calibrate the SerDes PLL. Use this workaround for the protocols and rates that only have the Ring VCO, which includes all 3.0/3.125/3.75 G protocols.

1. After device reset, read the SerDes PLL calibration values from the NXP Scratch Pad Fuse 0 (SFP_FSPFR0) register.
 - a. For 3.0 G protocols:
 1. Read BC = SFP_FSPFR0[2]
 2. Read DC = SFP_FSPFR0[3:5]
 3. Read FC = SFP_FSPFR0[6:11]
 - b. For 3.125 G protocols:
 1. Read BC = SFP_FSPFR0[12]
 2. Read DC = SFP_FSPFR0[13:15]
 3. Read FC = SFP_FSPFR0[16:21]
 - c. For 3.75 G protocols:
 1. Read BC = SFP_FSPFR0[22]
 2. Read DC = SFP_FSPFR0[23:25]
 3. Read FC = SFP_FSPFR0[26:31]
2. Write the BC, DC, and FC values into the SerDes PLL control registers. Do not modify other bits in SRDSxPLL n CR0 and SRDSxPLL n CR1.
 - a. Write SRDSxPLL n CR1[11:16] = FC
 - b. Write SRDSxPLL n CR1[2] = BC
 - c. Write SRDSxPLL n CR0[24:26] = DC
 - d. Write SRDSxPLL n CR1[3] = 1
 - e. Write SRDSxPLL n CR1[6] = 1
3. Read the SerDes PLL status registers to verify the calibration values have been written correctly.
 - a. Read SRDSxPLL n SR2[8] = BC
 - b. Read SRDSxPLL n SR2[10:15] = FC
 - c. Write SRDSxPLL n CR0[6] = 1
 - d. Read SRDSxPLL n SR2[12:14] = DC
 - e. Write SRDSxPLL n CR0[6] = 0
4. Wait 750 μ s to verify the PLL is locked by checking SRDSxPLL n CR0[8] = 1.

NXP software workaround:

The software patch is available at:

<http://git.denx.de/?p=u-boot.git;a=commit;h=b6808cd82d616bec2c357fb1b95116>

Fix plan: No plans to fix

A-007809: SerDes 2 LC VCO does not work for SRDS_PRTCL_S2 = 0x29

Affects: SerDes

Description: When choosing SRDS_PRTCL_S2 = 0x29, the SerDes 2 PLL tries to use LC VCO to lock at 3.125 G for SRIO protocol. However, because of the wrong definition in the PLL oscillator, the PLL fails to lock. Using Ring VCO to override LC VCO fixes the problem.

Impact: The LC VCO does not work for SerDes SRIO1 and SRIO2.

Workaround: Write SRDS2 PLL Control Register 1 [VCO_EN] to 1 to override LC VCO through PBI instruction.

Example:

- Write SRDS2PLL1CR1 = 0x0820_4100
- Write SRDS2PLL2CR1 = 0x0820_4100

Fix plan: No plans to fix

A-007837: When SRIO is configured in 4x port width mode, enabling 2x training may cause downtraining from 4x to 2x port width mode

Affects: SRIO

Description: SRIO supports training to 4x, 2x, and 1x port width modes when the port is configured as 4x. However, allowing 2x training in a 4x configuration may cause 4x training to fail, resulting in downtraining to 2x mode of operation.

Impact: SRIO ports configured as 4x may downtrain to 2x even in the absence of link errors.

Workaround: Before initiating training on an SRIO port configured as 4x via Control Command and Status Register (SRIO_PnCCSR[PW] = 1x as set by RCW), write SRIO_PnCCSR[PWO] = 110. This allows 4x and 1x training, but disables 2x training.

A write to SRIO_PnCCSR[PWO] = 110 can be implemented by:

- PBI: No special sequence is needed.
- Early initialization: Changing PWO causes SRIO to retrain.
- Re-initialization: PWO can also be changed while SRIO_PnCCSR[PD] = 1 or when clearing PD.

NOTE

With the workaround, SRIO ports configured as 4x that are unable to train to 4x because of link errors will downtrain to 1x.

Fix plan: No plans to fix

A-008116: A Serial RapidIO physical or logical/transport error interrupt (PINTn) that originates on port 2 is incorrectly reported as originating on port 1

Affects: SRIO

Description: When an interrupt occurs, examine the Serial RapidIO register EPWISR to determine which one of the ports caused the interrupt. There are individual PINTn bits that represent a corresponding port. Additionally, observe the LTLCCCSR[EPN] field to read the proper encoded port number.

With the erratum, if an interrupt occurs on port 2, the EPWISR and LTLCCCSR[EPN] register/field incorrectly reports that the interrupt originated on port 1 EPWISR[PINT1], instead of reporting port 2 EPWISR[PINT2].

Impact: If a user is using port 2 and does not apply workaround, a user does not know which port has caused the specified interrupt.

Workaround: Valid workaround needed if using Serial RapidIO port 2. Upon detection of a Serial RapidIO interrupt, a user has two options to determine the port origin.

If LTLEDCSR == 0x0: Physical layer type error. A user must examine each port's PnEDCSR register to determine the port origin.

If LTLEDCSR != 0x0: Logic/Transport layer type error. Since the LTLEDCSR register is shared among all ports a user must examine the following capture registers to aid in determining the port origin:

- LTLACCSR to obtain the address
- LTLIDCCSR to obtain source and destination IDs
- LTLCCCSR to obtain format type and transaction type

Fix plan: No plans to fix

A-005697: Suspend bit asserted before the port is in Suspend state

Affects: USB

Description: The EHCI specification states the following in the SUSP bit description:

In the Suspend state, the port is sensitive to resume detection. Note that the bit status does not change until the port is suspended and that there may be a delay in suspending a port if there is a transaction currently in progress on the USB.

In the USBDR controller, the PORTSCx[SUSP] bit changes immediately when the application sets it and not when the port is actually suspended.

Impact: Even though the behavior of the USBDR violates the EHCI specification statement, it does not cause any functional issue as, according to the EHCI specification, the application must wait for at least 10 milliseconds after a port indicates that it is suspended before initiating a port resume using the Force Port Resume bit.

Workaround: The software driver must follow the EHCI specification by waiting for at least 10 ms after setting the PORTSCx[SUSP] bit.

Fix plan: No plans to fix

A-006261: The USB Host controller may falsely go into disconnect condition

Affects: USB

Description: USB specs request that the minimum disconnect threshold should be greater than 525 mV. However, it has been found that some USB devices may be falsely disconnected due to host disconnect threshold being at a signal as low as 500 mV.

Impact: The USB device may be erroneously disconnected.

Workaround: Use a software write to shift the disconnect threshold to the higher side.

Write these fields into internal registers:

- R1: Register offset: 0x40
- R2: Register offset: 0xBC
- R1[7] = 1, R1[5:4] = 01
- R2[7] = 1, R2[5:4] = 01

Fix plan: No plans to fix

A-007792: USB reset hangs if ULPI phy is not connected and reset is issued before programming PTS field in UTMI mode

Affects: USB

Description: Whenever USB soft reset bit USBCMD[RST] asserts to reset the controller, it also resets the connected corresponding USB PHY. In order to provide time to ULPI PHY to come out of reset, the controller waits for the ULPI PHY to come out of reset before de-asserting USBCMD[RST] bit if the selected configuration is ULPI PHY. For the cases where the PHY is a UTMI PHY, USBCMD[RST] does not de-assert unless the configuration of the controller is changed to UTMI by configuring PORTSC[PTS] field. This is true because the default configuration of controller is ULPI. Therefore a new requirement is available for the controller while communicating with UTMI PHY. In this case, the soft reset normally issued by writing in USBCMD[RST] is not issued before changing the mode of the controller to UTMI by writing PORTSC[PTS].

Impact: USB will not work with open source Linux code.

Workaround: Do not check for USBCMD[RST] bit de-assertion before programming the controller.

Fix plan: No plans to fix

A-007903: The initial bits of USB high speed packet show amplitude of about 70 mV higher than the rest of HS packet amplitude

Affects: USB

Description: Initial bits of USB high speed packet show amplitude of about 70 mV higher than the rest of HS packet amplitude. This tends to occupy the EYE diagram mask thereby reducing the number of available software settings to adjust the EYE diagram to pass the EYE diagram. There is no functional data transfer impact because of this issue.

Impact: The system may fail the USB compliance test. However, functional data/packet transfer is not impacted.

Workaround: The following settings have been used to pass EYE diagrams on FSL boards. Use these settings and vary the bits to pass EYE diagram in case of issue. Program the following registers below with the corresponding values:

Register name	Offset	Value
USB_PHY_P1XCVRPRG	040h	00880000h ¹
USB_PHY_P2XCVRPRG	0BCh	00880000h ¹
USB_PHY_TVR	05Ch	00004406h ²
Notes: 1. -50% slew, +10% impedance 2. +10% HSTX amplitude "HSTX+1"		

Fix plan: No plans to fix

A-007992: Reduced HBM rating on USB_PWR_FAULT, USB_VBUS_CLAMP, USB_UDP, and USB_UDM pins

Affects: USB

Description: USB_HVDD is rated to 1KV HBM, while USB_UDP and USB_UDM have a risk of rating below 500V HBM. Only HBM levels below 500V are considered by the ESD Industry Council to merit additional factory safeguards. However, the USB_UDP and USB_UDM pins are immediately adjacent to at least one GND pin. As a result, the exposure to an actual field issue is considered virtually null. To safeguard even further, customers not enabling USB on this product are advised to tie all USB pins to GND. No additional board-level safeguards are required due to this errata notice.

Impact: These pins are vulnerable to human body electrostatic discharge stress. If it is inadvertently handled without proper precautions, device rupture could occur.

Workaround: Customers not enabling USB are advised to tie all USB pins to GND. No additional board-level safeguards are required due to this errata notice if USB is used.

Fix plan: Fixed in Rev 1.1

How to Reach Us:**Home Page:**nxp.com**Web Support:**nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including "typicals," must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, I2C BUS, ICODE, JCOP, LIFE VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, AltiVec, C-5, CodeTest, CodeWarrior, ColdFire, ColdFire+, C-Ware, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, Ready Play, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, SMARTMOS, Tower, TurboLink, and UMEMS are trademarks of NXP B.V. All other product or service names are the property of their respective owners. ARM, AMBA, ARM Powered, Artisan, Cortex, Jazelle, Keil, SecurCore, Thumb, TrustZone, and μ Vision are registered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. ARM7, ARM9, ARM11, big.LITTLE, CoreLink, CoreSight, DesignStart, Mali, mbed, NEON, POP, Sensinode, Socrates, ULINK and Versatile are trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

© 2016 NXP B.V.

