Chip Errata for the MPC8272 PowerQUICC II Family

This document describes all known silicon errata for the MPC8272 PowerQUICC II family of 0.13-μm (HiP7) communications processors. See Table 2 for a list of devices.

Table 1 summarizes this document's revision history.

Table 1. Document Revision History

Revision	Date	Substantive Changes
4	07/2012	• Added G14 • Updated the affected devices in CPM130 • Added CPM131 and CPM132
3	07/2010	• Added I2C1.
2	3/2008	• Added G13.
1.3	10/2007	• Modified PCI20 "Impact" and "Workaround" sections.
1.2	10/2007	• New errata: PCI19, PCI20, CPM130
1.1	6/2006	• Modified CPM125 in Table 4, removed from Rev. 0.0 column. • Modified CPM75 on page 32, should be CPM119. • Deleted errata: CPM124, CMP126
1	5/2006	• New errata: CPM127, CPM128
0.9	4/2006	• Updated template and made editorial corrections.
0.8	4/2006	• New errata: G12, JTAG1, PCI17, PCI18, CPM124, CPM125, CPM126
0.7	6/2005	• New errata: SIU20, PCI16, CPM106, CPM122, CPM123 • Modified errata: G10, PCI12, CPM75
0.6	2/2005	• Modified errata: CPM119

freescale™

Table 1. Document Revision History (continued)

Revision	Date	Substantive Changes
0.5	1/2005	• Modified errata: G10
0.4	11/2004	• New errata: G11
0.3	11/2004	• New errata: PCI14, PCI15, CPM120, and CPM 121
0.2	6/2004	• New errata: G10,PCI11, PCI12, and CPM119 • Modified errata: CPM101, SEC2, and SEC3
0.1	4/2004	• Addition of CPM117 • Modification of CPM101 and CPM114 descriptions
0	1/2004	• Initial release of document

Table 2 shows all MPC8272 PowerQUICC II family devices and silicon revisions.

Table 2. MPC8272 Family Devices and Silicon Revisions

Device	0.13μm (HiP7) Silicon		
	Revision	0.0	A.0
	Mask	0K50M	1K50M
MPC8272		√	√
MPC8271		√	√
MPC8248		√	√
MPC8247		√	√

Table 3 lists the silicon revisions to which each erratum applies and a reference to the page where each erratum is described.

Table 3. MPC8272 PowerQUICC II Family Silicon Errata Summary

Errata	Revision		Work-around exists	Description
	0.0	A.0		
System Interface Unit				
SIU18	√	√	—	$\overline{\text{ARTRY}}$ assertion when using pipeline depth of zero
SIU20	√	√	Yes	Hard reset configuration from boot EPROM in multi-master mode
I2C				
I2C1	√	√	Yes	Enabling I^{2}C could cause I^{2}C bus freeze when other I^{2}C devices communicate
General				
G6	√		—	Insufficient ESD protection on VCCSYN and VCCSYN1
G10	√	√	Yes	Possible I/O glitches during reset
G11	√	√	Yes	IMMR value of MPC8271/MPC8247 datecoded 2004

Chip Errata for the MPC8272 PowerQUICC II Family, Rev. 4

Errata	Revision 0.0	A.0	Work-around exists	Description
G12	√	√	Yes	CPU JTAG machine is not reset during $\overline{\text{PORESET}}$
G13	√	√	Yes	/SRESET might hang the device in PCI host mode
G14	√	√	Yes	DWLCK[0–2] bits behave as binary mask for data cache ways [0–2]
JTAG				
JTAG1	√	√	—	JTAG does not support the sample command
PCI				
PCI11	√	√	—	Outbound translation window can overlap PCI memory-mapped configuration space
PCI12	√	√	Yes	Deassertion of $\overline{\text{GNT}}$ during the address stepping cycle of an outbound configuration write transaction can cause PCI bus to hang
PCI14	√	√	Yes	PCI returns bad data on a master read following perr_response assertion
PCI15	√	√	Yes	Possible data corruption on PCI DMA writes with unaligned address
PCI16	√	√	—	PCI subsystem ID registers are not read-only
PCI17	√	√	Yes	PCI signals might be asserted on the falling edge of PCI clock after $\overline{\text{HRESET}}$
PCI18	√	√	Yes	PCI streaming problem when the latency timer is zero
PCI19	√	√	Yes	Inbound PCI Transaction with No Bytes Enabled May Cause the PCI Bus to Hang.
PCI20	√	√	Yes	Data corruption by DMA when destination address hold (DAHE) is used
Communications Processor Module				
CPM75	√	√	Yes	AAL2 microcode in ROM is not fully functional
CPM94	√	√	Yes	FCC $\overline{\text{RTS}}$ signal not asserted correctly
CPM98	√	√	Yes	I^2C erratic behavior can occur if extra clock pulse is detected on SCL
CPM101	√		Yes	FCC RxClav timing violation (slave)
CPM106	√		Yes	USCOM register flush command
CPM111	√		Yes	FCC missing reset at overrun
CPM112	√		Yes	FCC missing status
CPM114	√	√	Yes	IDMA transfer has an extra $\overline{\text{DACKx}}$
CPM115	√		Yes	APC transmits unwanted idle cells
CPM116	√		Yes	The pointer value of 93 is not supported in PFM mode of AAL1 CES
CPM117	√	√	Yes	False address compression
CPM119	√	√	Yes	FCC Tx, incorrect handling of Ethernet collision
CPM120	√	√	Yes	SS7_OPT[FISU_PAD] parameter has no effect on the number of flags between FISUs
CPM121	√	√	Yes	Data frame may be corrupted if writing to xMR registers while other TDM channels are active
CPM122	√	√	Yes	QMC—Missing flag between frames
CPM123	√	√	Yes	FCC ATM AAL5, underrun and idle cells

Chip Errata for the MPC8272 PowerQUICC II Family, Rev. 4

Table 3. MPC8272 PowerQUICC II Family Silicon Errata Summary (continued)

Errata	Revision		Work-around exists	Description
	0.0	A.0		
CPM125		√	Yes	An IN transaction might cause internal memory corruption when using USB transaction-level interface
CPM127	√	√	Yes	SCC-UART-Length field in the BD might not be updated correctly in polling mode
CPM128	√	√	Yes	Possible DPR data corruption in ATM APC mode
CPM130	√	√	Yes	Possible corrupted SP in AAL1 STF mode
CPM131	√	√	Yes	SS7 OCM is incorrectly left based on IDL-NIDL state transition
CPM132	√	√	Yes	SS7 OCM not entered when HDLC ABORT encountered in the middle of the frame
Integrated Security Engine				
SEC1	√		—	Public key execution unit (PKEU) failure
SEC2			—	**Note:** SEC2 has been re-classified as reference manual errata. Therefore, it has been removed from this document. ARC-4 Execution Unit (AFEU) slave mode RAM irregularity
SEC3			—	**Note:** SEC3 has been re-classified as reference manual errata. Therefore, it has been removed from this document. Improper key size, data size error interrupt in debug mode
SEC4	√		Yes	Potential MDEU snooping error
SEC5	√		Yes	PKEU, RSA_SSTEP data size

SIU1: $\overline{\text{ARTRY}}$ assertion when using pipeline depth of zero

Devices:

MPC8272, MPC8271, MPC8248, MPC8247

Description:

Internal (60x) slave maintains a pipeline depth of zero by asserting $\overline{\text{AACK}}$ only after $\overline{\text{TA}}$. When $\overline{\text{ARTRY}}$ is asserted, the 60x bus access is terminated and $\overline{\text{TA}}$ is not asserted. The internal (60x) slave does not assert $\overline{\text{AACK}}$, since $\overline{\text{TA}}$ was not asserted.

Workaround:

Use a pipeline depth of one (BCR[PLDP] = 0) for applications that require memory coherency.

Fix Plan:

No fix plan at this time.

SIU20:   Hard reset configuration from boot EPROM in multi-master mode

Devices:

MPC8272, MPC8271, MPC8248, MPC8247

Description:

BADDR27/$\overline{\text{IRQ1}}$, BADDR28/$\overline{\text{IRQ2}}$ and ALE/$\overline{\text{IRQ4}}$ pins are configured as inputs during the reset configuration process. If the MPC8272 is used in a multi-master environment (60x-compatible bus mode) the BADDR27, BADDR28, and ALE will not drive the boot EPROM during reset and will not access the desired reset configuration word. If the MPC8272 is used in a single master environment (single PowerQUICC II bus mode) no issue exists since A[27:28] is connected to the boot EPROM instead of BADDR[27:28] and the ALE control is not relevant.

Workaround:

Use external glue logic to implement a mux between BADDR[27:28] and A[27:28]. The output of the mux should be connected to the boot EPROM and the mux select should be the hard reset signal. When hard reset is asserted, the mux should select A[27:28] and when hard reset is negated, it should select BADDR[27:28].

Fix Plan:

No fix plan at this time.

I2C1: Enabling I^{2}C could cause I^{2}C bus freeze when other I^{2}C devices communicate

Devices:

MPC8260, MPC8255, MPC8260A, MPC8255A, MPC8250, MPC8264, MPC8265, MPC8266

Description:

When the I^{2}C controller is enabled by software, if the signal SCL is high and the signal SDA is low, and the I^{2}C address matches the data pattern on the SDA bus right after the enabling, an ACK is issued on the bus. The ACK is issued because the I^{2}C controller detects a START condition due to the nature of the SCL and SDA signals at the point of enablement. When this occurs, it may cause the I^{2}C bus to freeze. However, it happens very rarely due to the need of two conditions occuring at the same time.

Impact:

Enabling the I^{2}C controller may cause the I^{2}C bus to freeze while other I2C devices communicate on the bus.

Workaround:

Option 1: Enable the I2C controller before starting any I2C communications on the bus.

Option 2: If the I2C is configured as a slave, the following steps are needed:

1. Software enables device by setting I2CnCR[MEN] = 1, and starts a timer;
2. Delay 4 I^{2}C bus clocks.
3. Check Bus Busy bit (I2CnSR[MBB]).
   If MBB = 0, jump to step 6; (Good condition. Go to normal operation)
   else,
   Disable device (I2CnCR[MEN] = 0).
4. Re-configure all the I^{2}C registers, if necessary.
5. Go back to step 1.
6. Normal operation

Fix Plan:

No fix plan at this time.

G6: Insufficient ESD protection on VCCSYN and VCCSYN1

Devices:

MPC8272, MPC8271, MPC8248, MPC8247

Description:

The electrostatic discharge protection on VCCSYN and VCCSYN1 does not meet Freescale standards.

Workaround:

None

Fix Plan:

Fixed on Rev. A.0.

G10:   Possible I/O glitches during reset

Devices:

MPC8272, MPC8271, MPC8248, MPC8247

Description:

If core supply voltage (VDD) is below recommended operating conditions while I/O supply voltage (VDDH) is high during a power-on reset sequence, I/O pins (including those designated input only) might drive a value instead of being Hi-Z. This could confuse connected devices and, in turn, cause the PowerQUICC II device to behave improperly. Affected pins include such interfaces as the communication parallel I/O, reset, address and data bus pins, and JTAG pins. The value a pin may drive is random. As soon as core voltage has stabilized at its nominal level within recommended operating conditions, all pins will behave normally and the PowerQUICC II will continue to function properly.

Workaround:

It is recommended that VDD/VCCSYN be raised before or simultaneously with VDDH during the power-on reset sequence; that is, while VDDH <= recommended operating condition for VDD/VCCSYN during ramp of all voltages, it should be ensured that VDD/VCCSYN >= VDDH at all times. Refer to Figure 1.
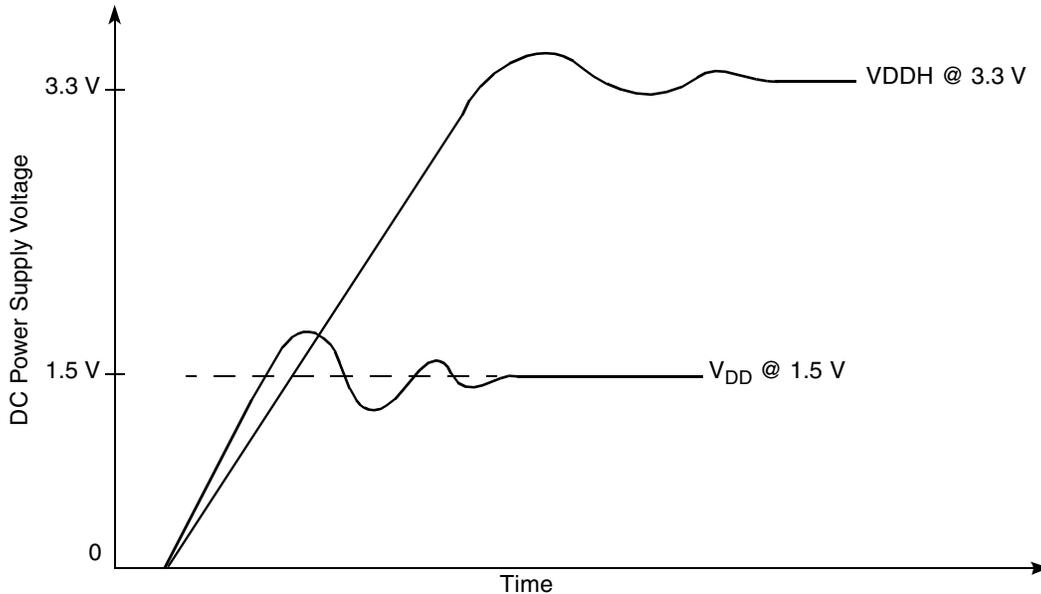

Figure 1. Power Supply Ramping

If the power supply design cannot be altered as recommended, then any logic connected to the PowerQUICC II which may react or be affected in an undesirable manner by unexpected low or high signal values during power ramp up should be appropriately secured in the design. Using

SDRAM as an example, it is usually sufficient to connect the CKE input to $\overline{\text{PORESET}}$. The practice of tying CKE high may not be safe and may lead to an unrecoverable system state.

Fix Plan:

No fix plan at this time.

G11: IMMR value of MPC8271/MPC8247 datecoded 2004

Devices:

MPC8271, MPC8247

Description:

For the MPC8247 and MPC8271 (non-encryption versions of the MPC8248 and MPC8272, respectively), the mask value located in IMMR[16–31] can be either 0x0D10 or a 0x0D90 for product with 2004 date code.

Workaround:

Customers should ensure that any software operation dependent on recognizing the mask value located at IMMR[16–31] should update that software to also recognize the 0x0D90 value, in addition to the usual 0x0D10 value.

Fix Plan:

For product with date code of 2005 and higher, the IMMR[16–31] mask value will be only a 0x0D10.

G12: CPU JTAG machine is not reset during $\overline{\text{PORESET}}$

Devices:

MPC8272, MPC8271, MPC8248, MPC8247

Description:

$\overline{\text{PORESET}}$ does not reset the CPU JTAG machine and, therefore, $\overline{\text{TRST}}$ must be asserted during $\overline{\text{PORESET}}$ in order for the CPU to work correctly.

Workaround:

For boards that have no need of third party debug tools:

- $\overline{\text{PORESET}}$ may be connected to $\overline{\text{TRST}}$, or
- $\overline{\text{TRST}}$ may be tied low

For boards that require debugging capability, implement an external circuit as shown in Figure 2.
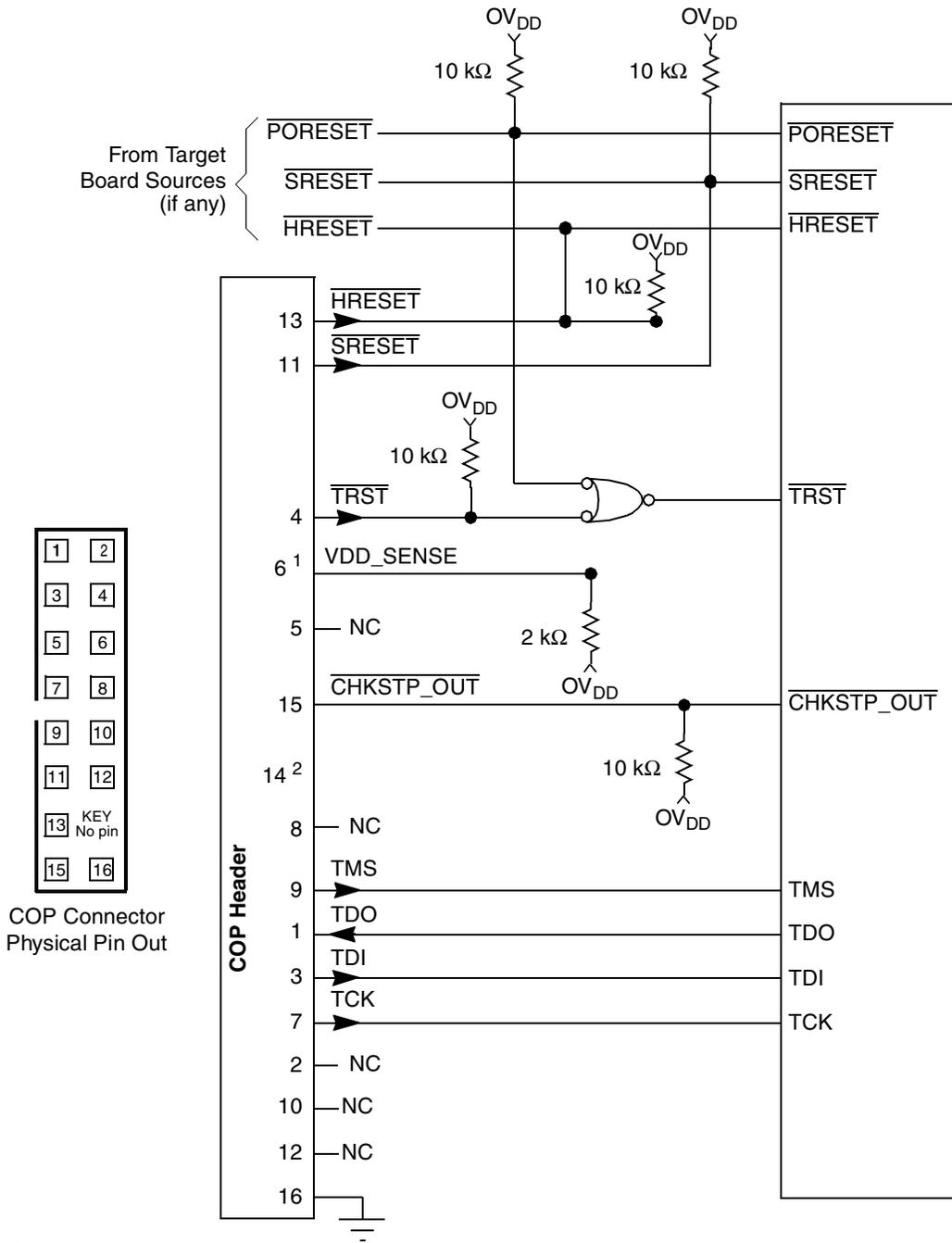
Notes:

1. Some systems require power to be fed from the application board into the debugger repeater card via the COP header. The resistor value for VDD_SENSE is around 20 Ω.

2. Key location; pin 14 is not physically present on the COP header.

Figure 2. COP Connections to PowerQUICC II Processors

Fix Plan:

No fix plan at this time.

G13: /SRESET might hang the device in PCI host mode

Devices:

8250, 8265, 8266, 8270, 8275, 8280

Description:

In PCI host mode, if /SRESET is used to reset the device, it is possible that the PCI module might miss the /SRESET negation and stay in the reset state when CPM/60x bus clock ratio is half integer.

Impact:

/SRESET assertion might cause the PCI controller to hang.

Work Around:

- Use /HRESET instead of /SRESET

OR

- Use integer CPM/60x bus clock ratio

OR

- Use software to emulate /SRESET
    - Emulate CPM Soft Reset:
        - Disable all active peripherals through mode register.
        - Reset the CPM by writing 0x80000000 to CPCR, check that RST bit is cleared. This cause the CPM to execute Reset routine.
    - Emulate SIU Soft Reset
        - Clear SIPNR by writing 0xFFFFFFFF to SIPNR_H, SIPNR_L
    - Emulate CPU Soft-Reset:
        - Configure MSR, EE=0, PR=0, ME=0, IR=0, DR=0, RI=0; other bits are unchanged.
        - Clear HID0, HID2
        - Set reset exception handler routine start address in CTR (0x100 or 0xfff00100 depending on MSR[IP]) and branch to reset exception using **bctr** instruction.

Fix Plan:

No plans to fix.

G14: DWLCK[0–2] bits behave as binary mask for data cache ways [0–2]

Devices:

MPC8272, MPC8271, MPC8248, MPC8247

Description:

The 3 DWLCK[0–2] bits behave as a mask for data cache ways [0–2] as shown in this table.

Value	Description
000	No ways locked
001	Way 2 locked
010	Way 1 locked
011	Way 1 + way 2 locked
100	Way 0 locked
101	Way 0 + way 2 locked
110	Way 0 + way 1 locked
111	Way 0 + way 1 + way 2 locked

Impact:

Ways 1–7 in the G2_LE cannot be locked on a per-way basis.

Work Around:

Only way 0 can be locked. Do not lock ways 1–7.

Fix Plan:

No plans to fix

JTAG1: JTAG does not support the sample command

Devices:

MPC8272, MPC8271, MPC8248, MPC8247

Description:

JTAG does not support the sample command.

Workaround:

None

Fix Plan:

No fix plan at this time.

PCI11: Outbound translation window can overlap PCI memory-mapped configuration space

Devices:

MPC8272, MPC8271, MPC8248, MPC8247

Description:

If an outbound translation window is programmed to have a translation that maps an address to any of the following addresses: IMMR+0x10900, IMMR+0x10904, IMMR+0x10908, a memory transaction will not be generated on PCI. Instead, the PCI CFG_ADDR, PCI CFG_DATA, or PCI INT_ACK registers of the memory-mapped configuration space will be accessed.

Workaround:

Do not allow software to program the outbound translation window such that it maps an address to IMMR+10900, IMMR+10904, IMMR+10908. To make this more general, software can be restricted so that an outbound translation window cannot overlap the internal memory map configuration window.

Fix Plan:

No fix plan at this time.

PCI12: Deassertion of $\overline{\text{GNT}}$ during the address stepping cycle of an outbound configuration write transaction can cause PCI bus to hang

Devices:

MPC8272, MPC8271, MPC8248, MPC8247

Description:

A configuration write transaction is mastered by the IOU and this transaction is retried. The configuration write transaction is then mastered again on the PCI bus. If during the address stepping cycle of the configuration transaction (the cycle before $\overline{\text{FRAME}}$ is asserted) the IOU $\overline{\text{GNT}}$ signal is negated, the PCI bus can hang. The IOU $\overline{\text{GNT}}$ signal is provided either by the internal or by the external arbiter, depending on arbiter configuration.

The PCI bus can potentially hang if configuration write transactions are retried in host mode and other masters are requesting the PCI bus.

The hang on the PCI bus manifests itself either as no assertion of $\overline{\text{FRAME}}$ or as the assertion of $\overline{\text{FRAME}}$ without the assertion of $\overline{\text{IRDY}}$.

Workaround:

Program the arbiter (internal or external) so that there are no other masters with a higher priority than the PowerQUICC II.

Fix Plan:

No fix plan at this time.

PCI14:   PCI returns bad data on a master read following perr_response assertion

Devices:

MPC8272, MPC8271, MPC8248, MPC8247

Description:

If the value of the PERR bit of the PCI bus command register (0x04 in the configuration space) is changed from 0 to 1 by using the CFG_DATA register and if there is a master read immediately following, the wrong read data is returned to the IOS.

Workaround:

- Unless the core is fetching its instructions from the PCI space, writing to the register twice or writing and then reading it prevents the problematic case from occurring.
- Do not use clock ratios above 6:1.

Fix Plan:

No fix plan at this time.

PCI15: Possible data corruption on PCI DMA writes with unaligned address

Devices:

MPC8272, MPC8271, MPC8248, MPC8247

Description:

If the PCI DMA destination address is in the 60x space and the data transfers are not multiples of 8 bytes and/or are not aligned to 8 bytes, the DMA might generate multi-beat write transactions with invalid bytes. As a result, the PCM generates a 60x transaction that writes beyond the allocated buffer. The PCM may also get stuck.

Workaround:

When transferring data to the 60x space using PCI DMA, use only destination address and byte counts that are multiples of 8.

Fix Plan:

No fix plan at this time.

PCI16: PCI subsystem ID registers are not read-only

Devices:

MPC8272, MPC8271, MPC8248, MPC8247

Description:

The subsystem vendor ID and subsystem device ID registers in the PCI configuration space are not read-only. Although the PCI specification does not state explicitly that these registers should be read-only from the PCI configuration space, Microsoft WHQL certification requires that these registers be read-only.

Workaround:

None

Fix Plan:

No fix plan at this time.

PCI17:   PCI signals might be asserted on the falling edge of PCI clock after $\overline{\text{HRESET}}$

Devices:

MPC8272, MPC8271, MPC8248, MPC8247

Description:

After $\overline{\text{PORESET}}$ sequence is over, if $\overline{\text{HRESET}}$ is asserted, PQ27E may lose synchronization between PCI clock and the internal clock the PCI block receives. As a consequence, PCI signals might be asserted on the falling edge of PCI clock. This issue is caused because the sampling of the PCI MODCK reset configuration bit (60x bus D[27]) starts when the bus is still floating.

Workaround:

- D[27] should be driven to 0 (for PCI_MODCK = 0, 66 MHz) or to 1 (for PCI_MODCK = 1, 33 MHz) for the duration of the reset configuration sequence.
- Put a pull-down (for PCI_MODCK = 0, 66 MHz) or pull-up (for PCI_MODCK = 1, 33 MHz) on D[27].
- Use $\overline{\text{PORESET}}$ instead of $\overline{\text{HRESET}}$, and disable software watchdog and checkstop reset.

Fix Plan:

No fix plan at this time.

PCI18: PCI streaming problem when the latency timer is zero

Devices:

MPC8272, MPC8271, MPC8248, MPC8247

Description:

When the latency timer is set to its default value (zero) there might be a protocol violation. As a consequence, the PCI bus may be locked.

Workaround:

Set the latency timer to a value other than zero.

Fix Plan:

No fix plan at this time.

PCI19: Inbound PCI Transaction with No Bytes Enabled May Cause the PCI Bus to Hang.

Devices:

MPC8272, MPC8271, MPC8248, MPC8247

Description:

An inbound PCI transaction of 1 or 2 32-bit data beats with all byte enables negated may cause the previous inbound transaction to be handled incorrectly. When this happens on a read transaction, a buffer becomes stuck in the IOS and the PCI repeatedly retries any attempt to read from that address.

Impact

There is no expected impact for systems using PCI devices that do not generate transactions with no byte enables. The issue was detected on a single apparatus. There is no expected impact for systems with D-cache disabled or PCI prefetch disabled.

Workaround:

Use only PCI devices and/or bus topologies that do not generate 1-2 beat transactions with no byte enables.

Fix Plan:

No fix plan at this time.

PCI20: Data corruption by DMA when destination address hold (DAHE) is used

Devices:

MPC8272, MPC8271, MPC8248, MPC8247

Description:

There can be corruption of the DMA data under the following conditions:

• DMAMR[DAHE] = 1 (destination address hold)

• DMAMR[DAHTS] = 10 (4 bytes) or 11 (8 bytes)

• DMA source address is not aligned to the transaction size specified by DAHTS

• The source port width is smaller than the destination transaction size OR the source port returns valid read data only in the valid byte lanes

Example of Error Condition:

• DAHTS is 8 bytes and the source port is a 32-bit PCI bus

• The source memory space is on the PCI bus and is not prefetchable

Impact

Corrupted data written to the destination peripheral or memory.

Workaround:

• Use a source address aligned to the destination transaction size

    or

• Do not access any DMA registers while this type of DMA transfer is active

Fix Plan:

No fix plan at this time.

CPM75: AAL2 microcode in ROM is not fully functional

Devices:

MPC8272, MPC8271, MPC8248, MPC8247

Description:

The enhanced AAL2 microcode integrated into ROM is not fully functional.

Workaround:

Use the RAM-based enhanced AAL2 microcode package available from Freescale.

Fix Plan:

No fix plan at this time.

CPM94: FCC $\overline{\text{RTS}}$ signal not asserted correctly

Devices:

MPC8272, MPC8271, MPC8248, MPC8247

Description:

At the beginning of an HDLC frame transmission that is preceded by more than one opening flag, $\overline{\text{RTS}}$ is not asserted if $\overline{\text{CTS}}$ is negated. This situation may cause a deadlock if the modem waits for the assertion of $\overline{\text{RTS}}$ before asserting $\overline{\text{CTS}}$.

Workaround:

- Transmit no flags between or before frames. Clear the FPSMR[NOF] bit.
- Set GFMR[RTSM] = 1 to ensure $\overline{\text{RTS}}$ is asserted when FCC is enabled. However, no hand shaking activities with the modem will occur for all the proceeding frames.

Fix Plan:

No fix plan at this time.

CPM98: I^{2}C erratic behavior can occur if extra clock pulse is detected on SCL

Devices:

MPC8272, MPC8271, MPC8248, MPC8247

Description:

The I^{2}C controller has an internal counter that counts the number of bits sent. This counter is reset when the I^{2}C controller detects a START condition. When an extra SCL clock pulse is inserted in between transactions (before START and after STOP conditions), the internal counter may not be reset correctly, which could generate partial frames (less than 8 bits) in the next transaction.

Workaround:

Do not generate extra SCL pulses on the I^{2}C bus. In a noisy environment, the digital filter I2MOD[FLT] and additional filtering capacitors should be used on SCL to eliminate clock spikes that may be misinterpreted as clock pulses.

Fix Plan:

No fix plan at this time.

CPM101: FCC RxClav timing violation (slave)

Devices:

MPC8272, MPC8271, MPC8248, MPC8247

Description:

FCC ATM receive UTOPIA slave mode. When the RxFIFO is full, RxClav is negated 2 cycles before the end of the cell transfer instead of 4. A master that polls RxClav or pauses 3 or 4 cycles before the end of the cell transfer may sample a false RxClav and an overrun condition may occur. The dashed line in the timing diagram below depicts the actual RxClav negation (2 cycles before the end of the cell transfer instead of 4 cycles). The signals in the timing diagram shown in Figure 3 are with respect to the master, hence, Tx interface is shown.

Workaround:

- Master should not poll RxClav or pause cell transfer at 4 cycles before the end of cell transfer. Master should poll 2 cycles before the end of the current cell or later. This can be achieved by introducing a cell-to-cell polling (and transfer) delay, which is equal or larger then one cell transfer time. If this can be achieved, the impact on performance is minimal.

- Configuring ATM only on FCC1 and setting FPSMR[TPRI] ensures highest priority to FCC1 Rx. In addition, for CPM utilization lower then 80% (as reported by the CPM performance tool based on UTOPIA maximal bus rate) the CPM performance is enough to guarantee that the RxFIFO does not fill up.
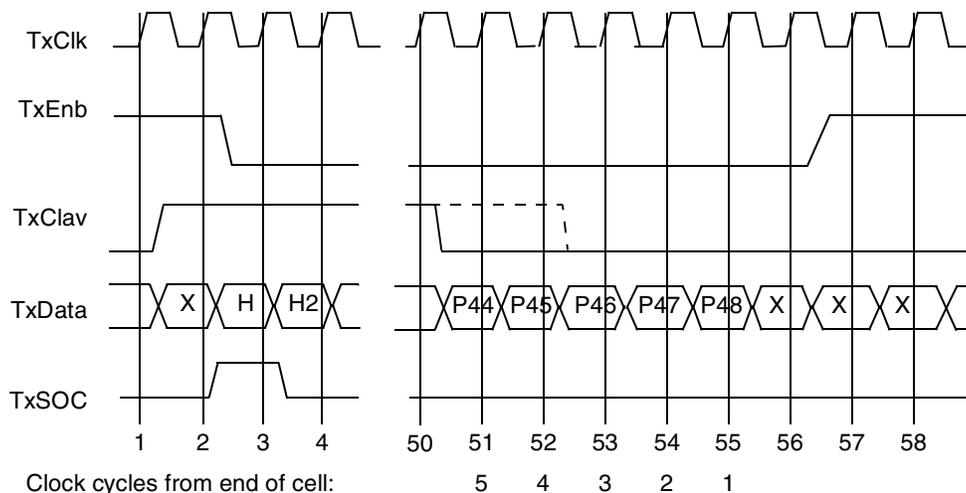

Figure 3. Transmit Timing for Cell-Level Handshake

Fix Plan:

Fixed on Rev. A.0.

CPM106: USCOM register flush command

Devices:

MPC8272, MPC8271, MPC8248, MPC8247

Description:

When the USCOM register is written with a flush command while the USB controller is starting to transmit from the FIFO, unpredictable behavior may result.

Workaround:

Whenever a flush command is used, the command should be written to USCOM twice with at least 3 bit-times between them.

Fix Plan:

Fixed on Rev. A.0.

CPM111: FCC missing reset at overrun

Devices:

MPC8272, MPC8271, MPC8248, MPC8247

Description:

Overrun error condition does not reset the FCC receiver in Ethernet mode, and may not set OV status in the RxBD. If RMON is not set, frames may be received with CR status continuously. CR and LG status or JBRC counts might be due to overrun condition. Fragment of a later frame may be appended to a fragment of an earlier one. If this frame length exceeds MFLR, it is discarded without indication on the RxBD. RMON JBRC will be incremented (false jabber).

Workaround:

Use the microcode patch provided by Freescale.

Fix Plan:

Fixed on Rev. A.0.

CPM112: FCC missing status

Devices:

MPC8272, MPC8271, MPC8248, MPC8247

Description:

TxBD may not be closed for FCC in half-duplex 10BaseT Ethernet. A mismatch could occur between the actual transmitted BD and the BD for which status is updated. As a result, the status of one to three BDs may not be updated, and they would appear 'Ready,' although the associated frames were transmitted (assuming a frame per BD).

Workaround:

Use the microcode patch provided by Freescale.

Fix Plan:

Fixed on Rev. A.0.

CPM114: IDMA transfer has an extra $\overline{\text{DACK}x}$

Devices:

MPC8272, MPC8271, MPC8248, MPC8247

Description:

In rare cases certain systems that use $\overline{\text{DREQ}}$ level-sensitive mode may show an additional $\overline{\text{DACK}x}$ cycle after $\overline{\text{DREQ}}$ has been deactivated. This causes extra data to be sent.

When the following conditions are all true:

- CPM IDMA operates in external request mode
- $\overline{\text{DREQ}}$ signal is set to be level-sensitive
- IDMA is writing to an external peripheral

The CPM may sample $\overline{\text{DREQ}}$ too early and, thus, erroneously start another DTS byte transfer sequence.

This erratum is resolved by a microcode patch. The effect of the patch is to have the IDMA perform a read bus transaction at the end of every DTS byte transfer sequence. $\overline{\text{DREQ}}$ is not sampled until this read completes. The address of the read must be on the same bus as the external peripheral. Refer to the README file of the CPM114 microcode patch for more details.

Workaround:

Use $\overline{\text{DREQ}}$ edge-sensitive mode.

Fix Plan:

No fix plan at this time.

CPM115: APC transmits unwanted idle cells

Devices:

MPC8272, MPC8271, MPC8248, MPC8247

Description:

In heavily loaded ATM applications, if the ATM pace controller (APC) is configured for multiple priority levels and a burst of traffic for transmission is sustained long enough on the highest priority APC table, an unwanted idle cell could be transmitted on the lower priority APC tables when cells are available in a lower priority APC scheduling table for transmission. The transmission of the unwanted idles could cause the valid ATM cells on lower priority APC scheduling tables not to be transmitted. This situation could affect all ATM channels that are not located in highest priority APC scheduling table.

Workaround:

Increase the size of lower priority APC scheduling tables so that they are large enough to absorb any burst or back-to-back bursts on the highest priority APC scheduling table, or use the microcode patch.

Fix Plan:

Fixed on Rev. A.0.

CPM116: The pointer value of 93 is not supported in PFM mode of AAL1 CES

Devices:

MPC8272, MPC8271, MPC8248, MPC8247

Description:

When working in partially filled mode (PFM), the pointer value of 93 is not generated, causing loss of synchronization at the far end. Furthermore, when receiving the pointer value of 93, synchronization is lost, causing loss of data and resynchronization routine.

Workaround:

Use the microcode patch provided by Freescale.

Fix Plan:

Fixed on Rev. A.0.

CPM117:   False address compression

Devices:

MPC8272, MPC8271, MPC8248, MPC8247

Description:

If there are active AAL0 channels and a CRC-10 error has been received, VP-level address compression might have false results, which could lead to one of the following:

- Wrong calculation of a VP pointer address
- Cells might be falsely discarded as misinserted cells
- Misidentification of misinserted cells (in CUAB mode)

This is a statistical error, which is conditional on the reception of AAL0 cells with CRC-10 error. The probability of false address compression is directly correlated with higher CPM bit rate and longer system bus latency.

Note: While the false address compression is possible only if there are active AAL0 channels, it might impact all AAL types. However, it cannot occur unless AAL0 cells with CRC-10 error have been received beforehand.

Workaround:

Use the microcode RAM patch provided by Freescale.

Fix Plan:

No fix plan at this time.

CPM119: FCC Tx, incorrect handling of Ethernet collision

Devices:

MPC8272, MPC8271, MPC8248, MPC8247

Description:

When an Ethernet collision occurs on the line 125 clocks after Tx-En assertion, late collision will be reported even though this is only 63 bytes into the frame instead of 64. When a collision occurs 124 cycles after Tx_En assertion, no event is reported, the TxBD is not closed, and transmission halts. Retransmission behavior is correct for collisions occurring between assertion of Tx_En and 123 clocks.

Workaround:

Use the microcode RAM patch provided by Freescale.

Fix Plan:

No fix plan at this time.

CPM120: SS7_OPT[FISU_PAD] parameter has no effect on the number of flags between FISUs

Devices:

MPC8272, MPC8271, MPC8248, MPC8247

Description:

The SS7_OPT[FISU_PAD] parameter has no effect on the number of flags between FISUs. Regardless of the value of this field, one flag will be present between back-to-back FISUs.

Workaround:

Use the latest SS7 microcode package provided by Freescale.

Fix Plan:

No fix plan at this time.

CPM121: Data frame may be corrupted if writing to xMR registers while other TDM channels are active

Devices:

MPC8272, MPC8271, MPC8248, MPC8247

Description:

The issue is data corruption in a working TDM during the enabling/disabling of the second TDM in the system. When writing to one of the following SI registers—GMR, AMR, BMR, CMR, DMR—while one or more TDMs are working, one data frame of a working TDM might get corrupted.

Workaround:

It is recommended to work with the shadow RAM when wanting to change data and not to disable and then enable the TDM.

Fix Plan:

No fix plan at this time.

CPM122: QMC—Missing flag between frames

Devices:

MPC8272, MPC8271, MPC8248, MPC8247

Description:

When QMC is configured to work in HDLC-flag mode and NOF = 0 (flag sharing between two consecutive packets), it is possible that the flag between frames is not transmitted. Two packets could then be transmitted as a single packet with no closing/opening flag in between.

Workaround:

Setting NOF = 1 meaning that there will be a minimum of two flags between consecutive frames. This will eliminate the problem. It is also possible to use a patch provided by Freescale.

Fix Plan:

No fix plan at this time.

CPM123: FCC ATM AAL5, underrun and idle cells

Devices:

MPC8272, MPC8271, MPC8248, MPC8247

Description:

The FCC in ATM AAL5 mode may experience unexpected transmit underruns, followed by transmission of idle cells even when in internal rate mode (when idle cells should not be generated). Technically, this problem can potentially occur on any PowerQUICC II processor using AAL5. However, the problem is more likely to occur when using a fast (50 MHz) 16-bit UTOPIA bus alongside other heavy DMA activity being performed by the CPM (such other high-speed communication peripherals with small buffers).

Workaround:

Acquire microcode patch or, if applicable, acquire the latest revision of the eFDS microcode package.

Fix Plan:

No fix plan at this time.

CPM125: An IN transaction might cause internal memory corruption when using USB transaction-level interface

Devices:

MPC8272, MPC8271, MPC8248, MPC8247

Description:

When using the USB transaction-level interface, an IN transaction can cause corruption of the DPRAM. This is likely to occur when the transaction is an isochronous packet, but it may occur for other types of packets as well.

Workaround:

- Use the microcode patch provided, or
- Use the packet-level interface

Fix Plan:

No fix plan at this time.

CPM127:   SCC-UART-Length field in the BD might not be updated correctly in polling mode

Devices:

MPC8272, MPC8271, MPC8248, MPC8247

Description:

When working in polling mode on the BD ring of a UART receiver there might be a rare situation where a BD status is indicating a full BD e.g. the Empty bit is cleared but the length field in this BD is not updated correctly.

Workarounds:

- Work in Interrupt mode.
- If working in polling mode, perform the polling of the BD status and if the Empty bit is cleared perform another read for the length field in this BD. The second read is guaranteed to have the correct length value.
- Use a ucode patch provided by Freescale.

Fix Plan:

No fix plan at this time.

CPM128: Possible DPR data corruption in ATM APC mode

Devices:

MPC8272, MPC8271, MPC8248, MPC8247

Description:

When the following 3 statements are true:

- When the ATM receiver is in emergency state
- The user has configured GMODE[REM] = 0 (enabled)
- PHY0 is not used in the system and has no APC tables configured

The code will update PHY 0 APC regardless if it exists and could cause DPR data corruption if this space is used by other protocols/different purposes.

Workaround:

Use a ucode patch provided by Freescale.

Fix Plan:

No fix plan at this time.

CPM130: Possible corrupted SP in AAL1 STF mode

Devices:

MPC8272, MPC8271, MPC8248, MPC8247

Description:

This erratum can manifest itself in either of the following two ways:

— The ATM controller does not generate SP parity (transmitter) or check it (receiver) for AAL1 structured format cells (AAL Type = 001, STF=1). This may lead to false SP error detection on the remote node.

— An ATM transmit busy event (TxBD not ready) may lead to corrupted SP.

Workaround:

Migrate to enhanced AAL1 code using AAL Type = 101 (AAL1-CES), while disabling CES mode using the following three steps:

1. ATM AAL1 parameter RAM extension[1] and dummy cell structure:
   Allocate memory for AAL1 internal, external statistics, dummy cell template and for partially filled cells and initialize the following parameters as shown in Table 1.

Table 1. AAL1 CES Field Descriptions[1]

Offset[2]	Size	AAL1 CES Field	Notes
0xE0	Word	TCELL_TMP_BASE_EXT (points to external memory, 64 bytes aligned, 64 bytes per channel).	
0xE8	Half-word	AAL1_Int_STATT_BASE (points to DPRAM, 16 bytes aligned, 16-byte size)	
0xEA	Half-word	AAL1_DUMMY_CELL_BASE (points to DPRAM, 64 bytes aligned, 64-byte size. If a cell drop is detected the receiver will insert a dummy cell).	
0xF0	Word	AAL1_Ext_STATT_BASE (points to external memory, 16 bytes aligned, 16 bytes per channel).	6

[1] For reference, see the "AAL1 CES Parameters" table in the "ATM AAL1 Circuit Emulation Service" chapter of the MPC8260 PowerQUICC™ II Family Reference Manual or the MPC8280 PowerQUICC™ II Family Reference Manual.

[2] Offset from FCC page base.

3 Allocate size larger then max AAL1 channel code $\times$ 64.

4 Allocate size 16 bytes, 16 bytes aligned.

5 Initialize dummy cell payload template with user defined data pattern.

[6] Allocate size larger then max AAL1 channel code $\times$ 16

2. Configure ATM connection tables:

— For AAL1 channels, change AAL-Type in RCT/TCT from AAL1 (001) to AAL1-CES (101).

— Protocol specific TCT remains unchanged.

— Protocol specifc RCT structure is changed.

3. Configure AAL1-specific RCT as listed below. Refer to Figure 1 and Figure 2.

— Initialize RCT[Block Size] (offset 0x16). This field doesn't exist in AAL1 RCT.

— The user should clear "Super Channel Number" and CASBS fields.

— SNE are reported in channel external statistics.

— Clear SLIPIM to disable SLIP interrupts.

Figure 1 shows the AAL1 protocol-specific area of an RCT entry.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Offset + 0x0E	—								PFM	SRT	INVE	STF	—			
Offset + 0x10	SRTS_TMP				—				—				SRTS_DEV			
Offset + 0x12	—	Valid Octet Size (VOS)							SPV	Structured Pointer (SP)						
Offset + 0x14	RBDCNT															
Offset + 0x16	—													SN		
Offset + 0x18	—				SNEM	—			RXBM	—						

Figure 1. AAL1 Protocol-Specific RCT

Figure 2 shows the AAL1 CES protocol-specific area of an RCT entry.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Offset + 0x0E	SRTS_DEV				—				PFM	SRT	INVE	STF	—			
Offset + 0x10	OCASB/SRTS_TMP				—				—				Valid Octet Size (VOS)			
Offset + 0x12	SPV	—			Structured Pointer (SP)											
Offset + 0x14	RBDCNT															
Offset + 0x16	Block Size												—	SN		
Offset + 0x18	Super Channel Number								RXBM	SLIPIM	—		CASBS			

Figure 2. AAL1 CES Protocol-Specific RCT

Note the following improved AAL1 features in the AAL1-CES code:

— SP parity generation and check.

— Support for Busy event (TxBD not ready).

— SNE statistics.

Fix Plan:

No fix plan at this time.

CPM131: SS7 OCM is incorrectly left based on IDL-NIDL state transition

Devices:

MPC8272, MPC8271, MPC8248, MPC8247

Description:

SS7 OCM does not count HDLC flags; it counts only data bytes and octets with the value 0xFF.

Impact: SUERM interrupts are generated too late or not at all. The SUERM mechanism does not behave according to the definition in Q.703 standard.

Behaviours experienced:

a) HDLC flags (0x7E) are not counted in OCM. Example:

Data pattern: 7E_LARGE_7E_FISU_7E_FISU_7E_FOREVER

OCM is entered due to the "large frame" (LG) event. According to Q.703, in OCM every byte must be counter, however in this case the HDLC flags (0x7E) are not counted.As a consequence, OCT and SUERM counters are not incremented correctly. Due to this, SUERM interrupt(s) will be generated too late or not at all.

b) Incorrect decision to leave OCM. Example:

Data pattern: FFFF_7E_DATA_7E_FFFF_5454_FOREVER

Events: IDL, NIDL, RxF, IDL, NIDL

Decision whether to enter / exit OCM is taken based on HDLC framer IDL / NIDL transitions. However, this is not correct - current example being a proof: the last transition (IDL / NIDL) will trigger the exit of OCM, which is not in compliance with Q.703, as we have a "loss of flag" case). As a consequence, SUERM interrupt(s) will be generated too late or not at all.

NOTE

This erratum applies not only to SS7, but also to ESS7 (enhanced SS7). Of course, for ESS7 things are a bit different than for SS7 – meaning that (according to Q.703 Annex A) OCM is not used at all and EIM is used instead of SUERM. Thus, description changes a bit: "As a consequence, interval is not marked as errored, thus EIM interrupt(s) will be generated too late or not at all."

Workaround:

Use the latest (depending on case) SS7/ESS7 microcode package provided by Freescale.

Fix plan:

No plans to fix

CPM132: SS7 OCM not entered when HDLC ABORT encountered in the middle of the frame

Devices:

MPC8272, MPC8271, MPC8248, MPC8247

Description:

If there is an ABORT character during frame reception and the subsequent data patter is random data, SS7 OCM is not entered.

Impact: SUERM interrupt is not generated for loss of flags in this instance.

NOTE

According to HDLC protocol (ISO13239/2000 section 5.1.1.2 Abort), Abort is defined as 7-14 consecutive "1"s on the line. 15 or more consecutive "1"s is interpreted as "line idle". The SS7 microcode entered octet count mode (OCM) only after an idle pattern was received; the abort sequence did not cause the receiver to enter the OCM . The Q.703 loss of flags case was not handled correctly in all scenarios.

Example:

Data pattern: 7E_DATA_7F_DATA_7E

OCM is not entered, although an abort character (0x7F – 7 consecutive "1"s) is present in the middle of the HDLC frame.

NOTE

This erratum applies not only to SS7, but also to ESS7 (enhanced SS7). Of course, for ESS7 things are a bit different than for SS7 – meaning that (according to Q.703 Annex A) OCM is not used at all and EIM is used instead of SUERM. Thus, description changes a bit: "Interval is not marked as errored, thus EIM interrupt(s) will be generated too late or not at all."

Workaround:

Use the latest (depending on case) SS7/ESS7 microcode package provided by Freescale.

Fix plan:

No plans to fix

SEC1:   Public key execution unit (PKEU) failure

Devices:

MPC8272, MPC8248

Description:

The security engine (SEC) PKEU is not functional.

Workaround:

None

Fix Plan:

Fixed on Rev. A.0.

SEC4: Potential MDEU snooping error

Devices:

MPC8272, MPC8248

Description:

Though highly unlikely with typical packet sizes (< 2 KB), when performing 'out-snooping' as required for IPSec out-bound processing, the SEC MDEU can fail to snoop a word from the symmetric encryption unit and consequently generate an incorrect HMAC.

Workaround:

If it is known that higher layer software in the IPSec tunnel termination point will recover from a failed HMAC and the remote possibility of an HMAC failure can be tolerated, the user can continue to use the higher performance, single-descriptor method.

If this use is unacceptable, rather than using a single descriptor, 0x2053_1C20 (SDES-CBC-Encrypt-HMAC-SHA-1) for outbound IPSec, processing could be split into two descriptors. The first descriptor 0x2050_0000 would perform SDES-CBC-Encrypt, and the second 0x31C0_0000 would perform HMAC-SHA-1.

Fix Plan:

Fixed on Rev. A.0.

SEC5: PKEU, RSA_SSTEP data size

Devices:

MPC8272, MPC8248

Description:

The SEC PKEU ROM routine RSA_SSTEP returns an incorrect result if any data size other than 1024 bits is used.

Workaround:

Software must switch back to the using a concatenation of MOD_R2MODN, POLY_F2M_MULT1_MONT, and MOD_EXP.

Fix Plan:

Fixed on Rev. A.0.