Freescale Semiconductor
Errata

Document Number: MPC8280CE
Rev. 7, 07/2012

Chip Errata for the MPC8280 PowerQUICC II Family

This document describes all known silicon errata for the MPC8280 PowerQUICC II family of integrated communications processors. See Table 2 for a list of devices.

Table 1 provides a revision history for this document.

Table 1. Document Revision History

Rev. Number	Date	Substantive Changes
7	07/2012	• Added G14 • Updated affected devices for CPM130 • Added CPM131, CPM132, and A-005319
6	3/2008	• Added CPM130 • Added G13
5	10/2007	• Modified PCI20 “Impact” and “work around” sections
4	8/2007	• Modified SIU18, adding an “Impact” section.
3	11/2006	• New errata: PCI19, PCI20
2	5/2006	• New errata: PCI18, CPM127, CPM128.
1	4/2006	• Updated template and made minor editorial corrections. • New errata: G12, JTAG1, CPM124, CPM126.
0.8	10/2005	• G9: Clarified description.
0.7	10/2005	• Modified errata: G10, PCI12, CPM75 • New errata: PCI16, CPM123
0.6	2/2005	• Modified erratum: CPM119
0.5	1/2005	• Modified erratum: G10

Table 1. Document Revision History (continued)

Rev. Number	Date	Substantive Changes
0.4	11/2004	• New errata: PCI14, PCI15, CPM120, CPM121 • Modified erratum: G9
0.3	6/2004	• New errata: G10, CPM118,CPM119 • Modified erratum: CPM101
0.2	6/2004	• New erratum: CPM117 • Modified errata: CPM101, CPM114
0.1	4/2004	• New errata: G8,G9, PCI13
0	12/2003	• New erratum: CPM116 • Prior to this document, the MPC8280 family was supported by the *MPC8260 PowerQUICC II Family Device Errata* (MPC8260CE).

Table 2 shows all MPC8280 PowerQUICC II family devices and silicon revisions.

Table 2. MPC8280 Family Devices and Silicon Revisions

Device		0.13 µm (HiP7) Silicon		
	Revision	0.0	0.1	A.0
	Mask	0K49M	1K49M	2K49M, 3K49M
MPC8270		√	√	√
MPC8275		√	√	√
MPC8280		√	√	√

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

Table 3. Errata Summary

Errata	0.13 µm (HiP7)			Work Around	Description
	0.0	0.1	A.0		
System Interface Unit					
SIU18	√	√	√	—	$\overline{\text{ARTRY}}$ assertion when using pipeline depth of zero
General					
G6	√	√		—	Insufficient ESD protection on VCCSYN and VCCSYN1
G7	√			—	$\overline{\text{HRESET}}$ may cause loss synchronization between CLKIN and the internal bus clock
G8	√	√		—	Assert $\overline{\text{PORESET}}$ to ensure correct JTAG operation
G9	√	√		Yes	Minimum CPU operating frequency (Fmin) for CPU multiplication factors ≥ 3.5
G10	√	√	√	Yes	Possible I/O glitches during reset
G12	√	√	√	Yes	CPU JTAG machine is not reset during $\overline{\text{PORESET}}$

Table 3. Errata Summary (continued)

Errata	0.13 µm (HiP7)			Work Around	Description
	0.0	0.1	A.0		
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					
PCI9	√			Yes	Simultaneous PCI inbound write transactions and PCI outbound read transactions can cause bus deadlock
PCI11	√	√	√	Yes	Outbound translation window can overlap PCI memory-mapped configuration space
PCI12	√	√	√	—	Deassertion of $\overline{\text{GNT}}$ during the address stepping cycle of an outbound configuration write transaction can cause PCI bus to hang
PCI13	√	√		Yes	PCI timing non-compliance with 66 MHz PCI clock
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
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
CPM96	√	√		Yes	ATM performance monitoring with AAL1 CES
CPM98	√	√	√	Yes	I^{2}C erratic behavior can occur if extra clock pulse is detected on SCL
CPM99	√	√		Yes	ABR TCTE[ER-TA] corruption
CPM101	√	√		Yes	FCC RxClav timing violation (slave)
CPM102	√	√		Yes	Error in APC when FCC enabled for IMA
CPM103	√	√		Yes	IMA group parameter RSCCI overwritten when a link enters HUNT state
CPM104	√	√		Yes	In USB host mode, TxBD processing may get stuck
CPM105	√	√		—	In USB host mode, timeout error condition due to bit-stuffing error may not be reported
CPM106	√	√		Yes	USCOM register flush command
CPM107	√	√		Yes	USB small packet loading
CPM108	√			Yes	USB Rx flow control

Table 3. Errata Summary (continued)

Errata	0.13 µm (HiP7)			Work Around	Description
	0.0	0.1	A.0		
CPM109	√	√		Yes	IMA IDCR initialization command (OpCode = 0x0E) does not work correctly
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	Pointer value of 93 is not supported in PFM mode of AAL1 CES
CPM117	√	√	√	Yes	False address compression
CPM118	√	√	√	Yes	MCC Rx, aborted HDLC frames
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
CPM123	√	√	√	Yes	FCC ATM AAL5, underrun and idle cells
CPM124	√	√	√	Yes	AAL1 CES slip-end data corruption
CPM126	√	√		Yes	Possible frame corruption in FCC Ethernet when filtering short frames
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
CPM129	√	√		Yes	Random number generator may be stuck
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
A-005319	√	√	√	Yes	Zero-bit insertion error in MCC non-supper channel HDLC mode

SIU18: $\overline{\text{ARTRY}}$ assertion when using pipeline depth of zero

Devices:

MPC8280, MPC8275, MPC8270

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 will be terminated and $\overline{\text{TA}}$ will not be asserted. Therefore, the internal (60x) slave will not assert $\overline{\text{AACK}}$, because $\overline{\text{TA}}$ was not asserted.

Work Around:

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

Impact

Bus hang

Fix Plan:

No fix plan at this time.

G6: Insufficient ESD protection on VCCSYN and VCCSYN1

Devices:

MPC8280, MPC8275, MPC8270

Description:

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

Work Around:

None

Fix Plan:

Fixed on Rev. A.0.

G7: $\overline{\text{HRESET}}$ may cause loss synchronization between CLKIN and the internal bus clock

Devices:

MPC8280, MPC8275, MPC8270

Description:

When $\overline{\text{HRESET}}$ is asserted, PQ27 may lose synchronization between CLKIN and the internal bus clock. As a consequence, read and write accesses to external synchronous memories may be corrupted. This issue is caused because the sampling of the PLL bypass reset configuration bit (60x bus D[12]) starts when the bus is still floating.

Work Around:

D[12] should be driven to ‘0’ for the duration of the reset configuration sequence. Alternatively, use $\overline{\text{PORESET}}$ instead of $\overline{\text{HRESET}}$, and disable software watchdog and checkstop reset

Fix Plan:

Fixed on Rev. 0.1

G8: Assert $\overline{\text{PORESET}}$ to ensure correct JTAG operation

Devices:

MPC8280, MPC8275, MPC8270

Description:

To ensure correct operation of the JTAG module, it is required to assert the $\overline{\text{PORESET}}$ external pin at least once after the processor is powered up, for the duration of at least 240 ns, even though the clock input for the CLKIN is not required. Without asserting $\overline{\text{PORESET}}$ at least once, an internal test feature might randomly awaken and disable JTAG BSR testability. $\overline{\text{PORESET}}$ should be negated before starting JTAG operations.

Work Around:

None

Fix Plan:

Fixed on Rev. A.0.

G9: Minimum CPU operating frequency (Fmin) for CPU multiplication factors ≥ 3.5

Devices:

MPC8280, MPC8275, MPC8270

Description:

The correct operation of the PowerQUICC II cannot be guaranteed if both of the following conditions are met:

- Bus-to-core multiplication factor (MF) of 3.5 or greater is used
- PowerPC core is operating below the values indicated in the package-specific rules listed in the following table:

Rating	Package	Devices	Fmin (MHz)
Commercial	ZU	• MF ≥ 3.5	• 285
	VR/ZQ	• Devices with 266 MHz-rated core • Devices with a 200 MHz-rated core	• 250 • 150
Industrial	ZU	• Devices with 450 MHz-rated core • Devices with 333 MHz-rated core	• 350 • 300
	VR/ZQ	Devices with 266 MHz rated core are not available. It is recommended that customers consider using the 200/200/66 speed grade parts.	

Symptoms of this erratum include a halting of the core at potentially random points in software execution. Those locations in software may be related to triggering high amounts of bus activity, such as bursting. No errors may be reported. The halting may occur at different points of the same software between runs. Core-related activity on the bus will cease, but such bus activity as SDRAM refresh and CPM accesses may continue. The problem may not manifest itself until the device is operated at the cold end of the allowable temperature range for that device.

Work Around:

It is strongly recommended that customers migrate applications to use revision A silicon or later. For applications that use earlier silicon, do not operate the CPU below the aforementioned Fmin values when using an MF ≥3.5 or do not use MFs ≥3.5, as specified in the above table.

Fix Plan:

Fixed on Rev. A.0.

G10: Possible I/O glitches during reset

Devices:

MPC8280, MPC8275, MPC8270

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.

Work Around:

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.

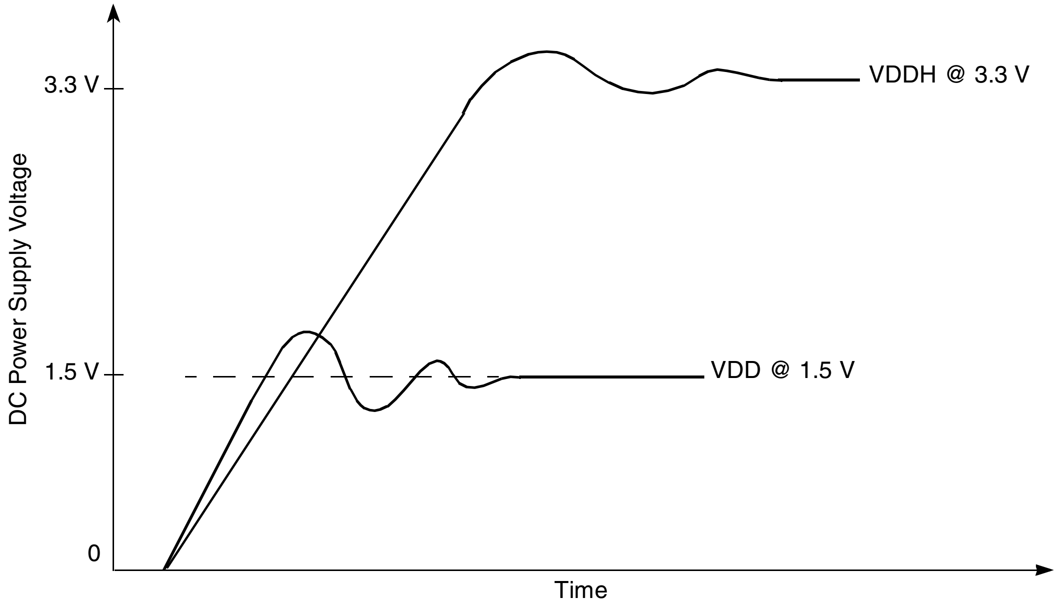


Figure 1. Power Supply Ramping

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.

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

Devices:

MPC8280, MPC8275, MPC8270

Description:

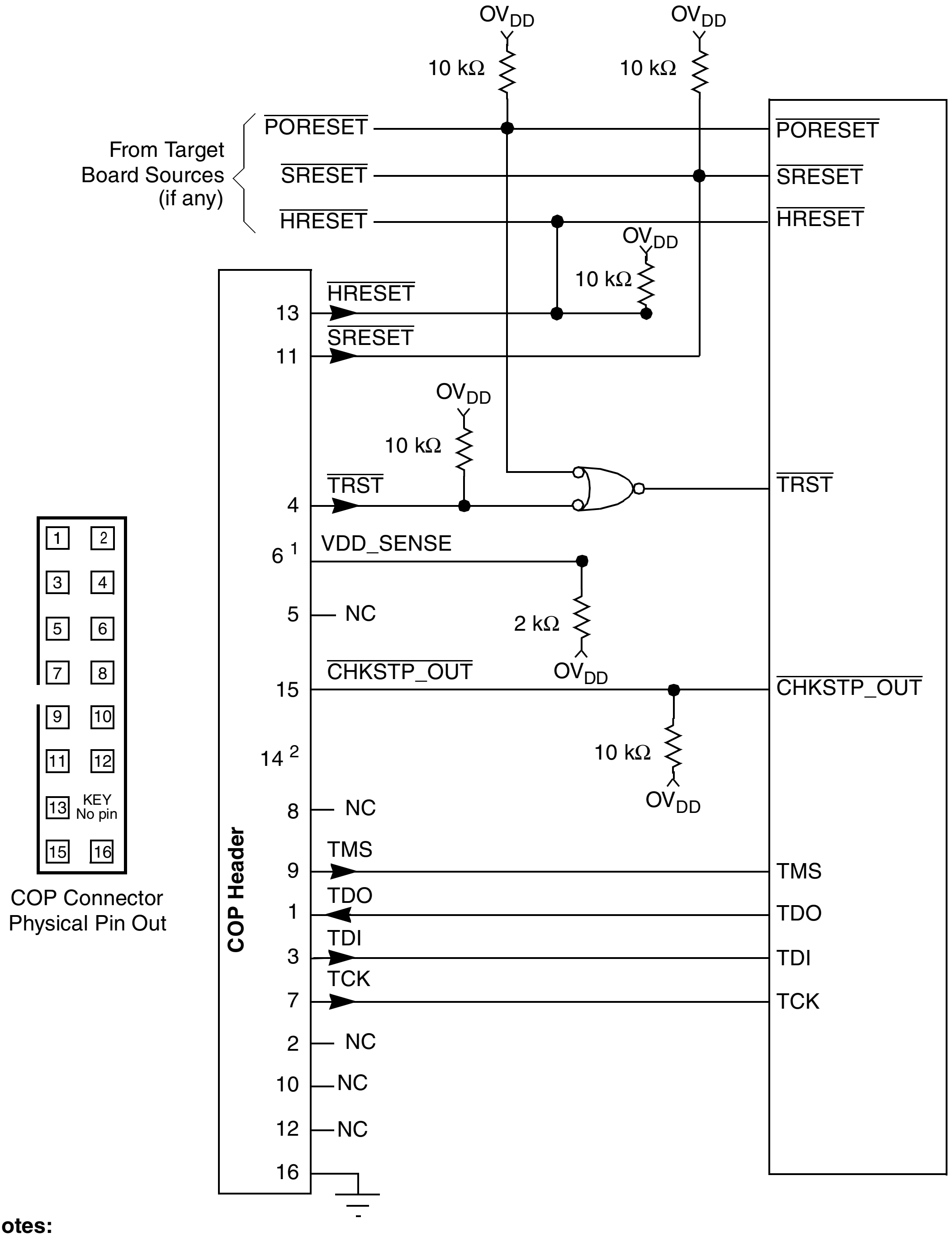
$\overline{\text{PORESET}}$ does not reset the CPU JTAG machine and, therefore, $\overline{\text{TRST}}$ must be asserted during $\overline{\text{PORESET}}$ in order for the CPU to work correctly.

Work Around:

For boards that have no need of third party debug tools, use the following work arounds:

- $\overline{\text{PORESET}}$ 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:

MPC8280, MPC8270, MPC8275

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 way0 can be locked. Do not lock ways 1–7.

Fix Plan:

No plans to fix

JTAG1: JTAG does not support the sample command

Devices:

MPC8280, MPC8275, MPC8270

Description:

JTAG does not support the sample command.

Work Around:

None

Fix Plan:

No fix plan at this time.

PCI9: Simultaneous PCI inbound write transactions and PCI outbound read transactions can cause bus deadlock

Devices:

MPC8280, MPC8275, MPC8270

Description:

In PCI mode, when outbound and inbound traffic happen simultaneously, it is possible that after some random number of transactions the system will hang. The deadlock situation may be caused by one of the following events:

- 60x master reads from PCI memory/IO/config space while a PCI master writes into 60x memory
- 60x master reads from PCI memory/IO/config space while a PCI bridge's DMA channel writes into 60x memory
- 60x master reads from PCI bridge's internal register while a PCI master writes into 60x memory
- 60x master reads from PCI bridge's internal register while a PCI bridge's DMA channel write into 60x memory

Work Around:

Do not allow simultaneous outbound read and inbound write transactions, or use IDMA mechanism to perform data read from PCI memory/IO/config space and from PCI bridge's internal registers. Also note that the IDMA should be initialized in such a way that the source (PCI memory/IO/config space/ PCI bridge's internal registers) is selected to locate on local bus in the IDMA BD (SDTB = 1).

Fix Plan:

Fixed on Rev. 0.1.

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

Devices:

MPC8280, MPC8275, MPC8270

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.

Work Around:

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:

MPC8280, MPC8275, MPC8270

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 $\overline{\text{IRDY}}$.

Work Around:

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.

PCI13: PCI timing non-compliance with 66 MHz PCI clock

Devices:

MPC8280, MPC8275, MPC8270

Description:

The PCI interface is not compliant with the industry PCI specification. The PCI specification calls for an input setup time of 3 ns. With a 66 MHz PCI clock, the PCI interface may require up to 3.8 ns input setup time.

Work Around:

Use a PCI clock of 33 MHz to eliminate the problem.

Fix Plan:

Fixed on Rev. A.0.

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

Devices:

MPC8280, MPC8275, MPC8270

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.

Work Around:

- 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:

MPC8280, MPC8275, MPC8270

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.

Work Around:

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:

MPC8280, MPC8275, MPC8270

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.

Work Around:

None

Fix Plan:

No fix plan at this time.

PCI18: PCI streaming problem when the latency timer is zero

Devices:

MPC8280, MPC8275, MPC8270

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.

Work Around:

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:

MPC8250, MPC8265, MPC8266

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.

Work Around:

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:

MPC8250, MPC8265, MPC8266

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.

Work Around:

- 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:

MPC8280, MPC8275, MPC8270

Description:

The enhanced AAL2 microcode integrated into ROM on Rev. C.2 is not fully functional.

Work Around:

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:

MPC8280, MPC8275, MPC8270

Description:

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

Work Around:

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

Fix Plan:

No fix plan at this time.

CPM96: ATM performance monitoring with AAL1 CES

Devices:

MPC8280, MPC8275, MPC8270

Description:

Data in DPRAM is corrupted when performance monitoring is enabled in the receiver.

Work Around:

- Disable receive performance monitoring RCT[PMT] = 0
- Use the microcode patch available from Freescale

Fix Plan:

Fixed on Rev. A.0.

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

Devices:

MPC8280, MPC8275, MPC8270

Description:

The I^2C controller has an internal counter that counts the number of bits sent. This counter is reset when the I^2C 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 get reset correctly. This can generate partial frames (less than 8 bits) in the next transaction.

Work Around:

Do not generate extra SCL pulses on the I^2C 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.

CPM99: ABR TCTE[ER-TA] corruption

Devices:

MPC8280, MPC8275, MPC8270

Description:

When using the AAL5 ABR ROM microcode it is possible for the TCTE[ER-TA] field to be overwritten with an erroneous value. This, in turn, will cause the TCTE[ER-BRM] to be updated with this value. As TCTE[ER-BRM] holds the maximum explicit rate value allowed for B-RM cells an erroneous value in this field can have a detrimental effect on the network performance.

Work Around:

Use the microcode patch available from Freescale.

Fix Plan:

Fixed on Rev. A.0.

CPM101: FCC RxClav timing violation (slave)

Devices:

MPC8280, MPC8275, MPC8270

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 are with respect to the master, hence, Tx interface is shown.

Work Around:

- 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.

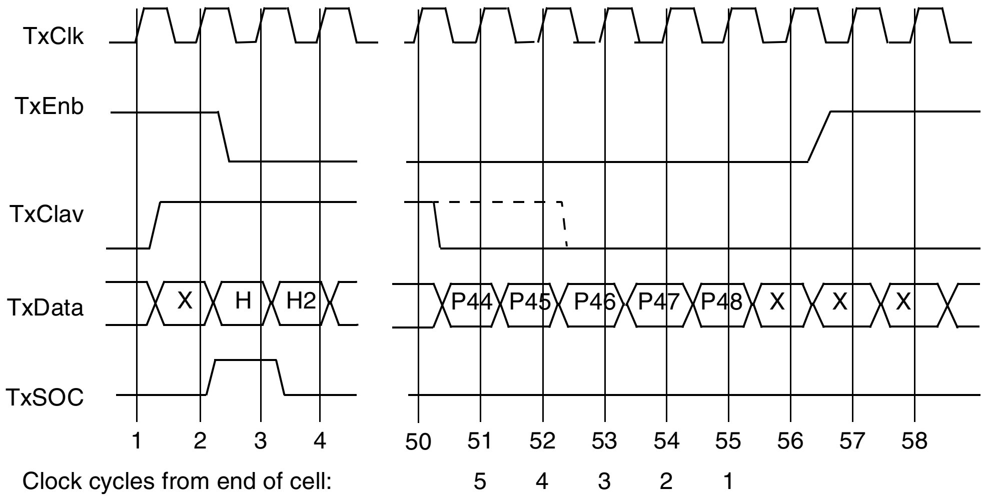


Figure 3. Transmit Timing for Cell-Level Handshake

Fix Plan:

Fixed on Rev. A.0.

CPM102: Error in APC when FCC enabled for IMA

Devices:

MPC8280

Description:

There is an error in the APC when the FCC is enabled for IMA that can cause the CPM to hang after max iterations is reached. This error is typically observed at high data rate, as ATM channels will be rescheduled further down the APC table as their pace is multiplied by TNUMLINKS.

Work Around:

Use the latest IMA microcode patch provided by Freescale.

Fix Plan:

Fixed on Rev. A.0.

CPM103: IMA group parameter RSCCI overwritten when a link enters HUNT state

Devices:

MPC8280

Description:

The IMA received group parameter RCCI, which stores the SCCI value of the last ICP received by a group, can be overwritten if a link goes into HUNT mode. This means ICP cells can stop being passed to the ATM layer for an unknown period.

Work Around:

Use the latest IMA microcode patch provided by Freescale.

Fix Plan:

Fixed on Rev. A.0.

CPM104: In USB host mode, TxBD processing may get stuck

Devices:

MPC8280, MPC8275, MPC8270

Description:

Under certain error conditions, the TxBD processing may get stuck: the BD R bit will never be cleared, and the TXEn event bit will not be set. If the SW is waiting for the processing to complete, a deadlock may occur.

Work Around:

The software should have a timeout mechanism. If the R bit is not cleared in a reasonable amount of time, reset the transmission process by doing the following:

1. STOP TX ENDPOINT command to CPCR
2. Flush FIFO through the USCOM register
3. RESTART TX ENDPOINT command to CPCR

Fix Plan:

Fixed on Rev. A.0.

CPM105: In USB host mode, timeout error condition due to bit-stuffing error may not be reported

Devices:

MPC8280, MPC8275, MPC8270

Description:

Under the following conditions, the TxBD will incorrectly be returned with no error reported, when actually a timeout error should be reported:

- A bit-stuffing error is detected while waiting for the handshake PID during an OUT/SETUP transaction.
- A bit-stuffing error is detected while waiting for a PID in response to an IN token.

Work Around:

No direct work around. The second case can be detected by software because there will be no received packet. The first case cannot be easily detected. If there is a problem, it will likely be detected by higher layer protocols.

Fix Plan:

Fixed on Rev. A.0.

CPM106: USCOM register flush command

Devices:

MPC8280, MPC8275, MPC8270

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.

Work Around:

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.

CPM107: USB small packet loading

Devices:

MPC8280, MPC8275, MPC8270

Description:

When using small packets (less than 32 bytes), the USB may sometimes load multiple packets for transmission even if the MF bit of the EP register is 0 and/or the CNF bit of the TxBD is 1. This may cause any errors that occur during the transaction to be reported in the wrong BD.

Work Around:

When using small packets, do not provide more than one packet at a time to the TxBD ring unless a multi-packet sequence is intended.

Fix Plan:

Fixed on Rev. A.0.

CPM108: USB Rx flow control

Devices:

MPC8280, MPC8275, MPC8270

Description:

The USB standard defines that a function that receives an OUT transaction and cannot accept the data due to flow control reasons returns a NAK handshake packet. While the USB module supports the NAK response, it still receives the data. Because the host will re-transmit the data, the load on the software is actually increased, and the flow control is not effective.

If the software performance is not sufficient to support the full rate of the USB reception, there is no way to use the USB flow control feature to significantly reduce the load. Therefore, a BSY condition occurs, and data may be lost.

Work Around:

Use a relatively large RxBD ring. Set the NAK condition as soon as the software starts to fall behind to ensure that it is set before the BSY condition is reached. Any duplicated packets can be identified by the repeated data PID and discarded. Then, clear the RxBDs and NAK condition to continue reception.

Fix Plan:

Fixed on Rev. 0.1.

CPM109: IMA IDCR initialization command (OpCode = 0x0E) does not work correctly

Devices:

MPC8280

Description:

IMA IDCR Initialization command (OpCode = 0x0E) does not work correctly.

Work Around:

User software must perform special initialization sequence before enabling IMA. Issue the following CPCR commands:

```
cp.cpcr = 0x32810fbf
cp.cpcr = 0x32a10fbf
cp.cpcr = 0x32c10fbf
cp.cpcr = 0x32e10fbf
```

User should wait until cpcr.flg clears after each issuance before software proceeds. Note that the values apply to both Rev. 0.0 and Rev. 0.1.

Fix Plan:

No fix plan at this time.

CPM111: FCC missing reset at overrun

Devices:

MPC8280, MPC8275, MPC8270

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 may be due to overrun condition. Fragment of a later frame may be appended to a fragment of an earlier one. If this frame length exceed MFLR, it will be discarded without indication on the RxBD. RMON JBRC will be incremented (false jabber).

Work Around:

Use the microcode patch provided by Freescale.

Fix Plan:

Fixed on Rev. A.0.

CPM112: FCC missing status

Devices:

MPC8280, MPC8275, MPC8270

Description:

TxBD may not be closed for FCC in half-duplex 10BaseT Ethernet. A mismatch may result 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 will appear as ‘Ready,’ although the associated frames have been transmitted (assuming a frame per BD).

Work Around:

Use the microcode patch provided by Freescale.

Fix Plan:

Fixed on Rev. A.0.

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

Devices:

MPC8280, MPC8275, MPC8270

Description:

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

When the following conditions are all true:

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

Work Around:

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

Fix Plan:

No fix plan at this time.

CPM115: APC transmits unwanted idle cells

Devices:

MPC8280, MPC8275, MPC8270

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 could affect all ATM channels that are not located in highest priority APC scheduling table.

Work Around:

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 available from Freescale.

Fix Plan:

Fixed on Rev. A.0.

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

Devices:

MPC8280, MPC8275, MPC8270

Description:

When working in PFM mode (partially filled mode), the pointer value of 93 is not generated. It causes the loss of synchronization at the far end. Also, when receiving the pointer value of 93, the synchronization will be lost, which causes loss of data and resynchronization routine.

Work Around:

Use the microcode patch available from Freescale.

Fix Plan:

Fixed on Rev. A.0.

CPM117: False address compression

Devices:

MPC8280, MPC8275, MPC8270

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.

Work Around:

Use the microcode RAM patch provided by Freescale.

Fix Plan:

No fix plan at this time.

CPM118: MCC Rx, aborted HDLC frames

Devices:

MPC8280, MPC8275, MPC8270

Description:

When an aborted HDLC frame is followed by a good frame, there may appear in the receive data buffer both the data of the aborted frame followed by the data of the good frame.

Work Around:

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:

MPC8280, MPC8275, MPC8270

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.

Work Around:

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:

MPC8280, MPC8275, MPC8270

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.

Work Around:

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:

MPC8280, MPC8275, MPC8270

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.

Work Around:

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.

CPM123: FCC ATM AAL5, underrun and idle cells

Devices:

MPC8280, MPC8275, MPC8270

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).

Work Around:

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

Fix Plan:

No fix plan at this time.

CPM124: AAL1 CES slip-end data corruption

Devices:

MPC8280, MPC8275, MPC8270

Description:

When using the AAL1 CES microcode with the user template mode (CHAMR.UTM = 1), on exit from slip mode, the framer of the MCC is incorrectly configured. This can cause data corruption to occur on the MCC transmit side, which results in erroneous behavior at the corresponding receive side.

Work Around:

Use the microcode patch provided by Freescale.

Fix Plan:

No fix plan at this time.

CPM126: Possible frame corruption in FCC Ethernet when filtering short frames

Devices:

MPC8280, MPC8275, MPC8270

Description:

The problem occurs when the Ethernet controller is configured to filter out short frames, and there is a short frame with a length of 32 bytes. In this case, the next frame will be concatenated to the short frame instead of overwriting it, resulting in data corruption of the frame.

Work Around:

- Use the microcode patch provided, or
- If working in a mode where short frames are being received, the software should ignore the short frames.

Fix Plan:

Fixed on Rev. A.0.

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

Devices:

MPC8280, MPC8275, MPC8270

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, that is, the empty bit is cleared but the length field in this BD is not updated correctly.

Work Around:

- 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 the ucode patch provided by Freescale.

Fix Plan:

No fix plan at this time.

CPM128: Possible DPR data corruption in ATM APC mode

Devices:

MPC8280, MPC8275, MPC8270

Description:

When the following three 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.

Work Around:

Use the ucode patch provided by Freescale.

Fix Plan:

No fix plan at this time.

CPM129: Random number generator may be stuck

Devices:

MPC8280, MPC8275, MPC8270

Description:

The CPM hardware random number generator might have a fault in it that cause it to generate a fixed number (typically zero).

Impact of the fault:

The back off time after collision on the Ethernet LAN would not be random. If two or more units with this fault are connected to the same collision domain, some of them may experience a retransmission-limit error.

The “RAND” host-command would return a constant number.

Work Around:

Use the ucode patch provided by Freescale.

Fix Plan:

Fixed on Rev. A.0

CPM130: Possible corrupted SP in AAL1 STF mode

Devices:

MPC8280, MPC8275

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).	—

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

[2] Offset from FCC page base.

3 Allocate size larger then max AAL1 channel code × 64.

4 Allocate size 16 bytes, 16 bytes aligned.

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

Allocate size larger then max AAL1 channel code × 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.

<table>
<tr><th></th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th></tr>
<tr><td>Offset + 0x0E</td><td colspan="8">—</td><td>PFM</td><td>SRT</td><td>INVE</td><td>STF</td><td colspan="4">—</td></tr>
<tr><td>Offset + 0x10</td><td colspan="4">SRTS_TMP</td><td colspan="4">—</td><td colspan="4">—</td><td colspan="4">SRTS_DEV</td></tr>
<tr><td>Offset + 0x12</td><td colspan="2">—</td><td colspan="6">Valid Octet Size (VOS)</td><td>SPV</td><td colspan="7">Structured Pointer (SP)</td></tr>
<tr><td>Offset + 0x14</td><td colspan="16">RBDCNT</td></tr>
<tr><td>Offset + 0x16</td><td colspan="13">—</td><td colspan="3">SN</td></tr>
<tr><td>Offset + 0x18</td><td colspan="4">—</td><td>SNEM</td><td colspan="3">—</td><td>RXBM</td><td colspan="7">—</td></tr>
</table>

Figure 1. AAL1 Protocol-Specific RCT

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

<table>
<tr><th></th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th></tr>
<tr><td>Offset + 0x0E</td><td colspan="4">SRTS_DEV</td><td colspan="4">—</td><td>PFM</td><td>SRT</td><td>INVE</td><td>STF</td><td colspan="4">—</td></tr>
<tr><td>Offset + 0x10</td><td colspan="4">OCASB/SRTS_TMP</td><td colspan="4">—</td><td colspan="2">—</td><td colspan="6">Valid Octet Size (VOS)</td></tr>
<tr><td>Offset + 0x12</td><td>SPV</td><td colspan="3">—</td><td colspan="12">Structured Pointer (SP)</td></tr>
<tr><td>Offset + 0x14</td><td colspan="16">RBDCNT</td></tr>
<tr><td>Offset + 0x16</td><td colspan="12">Block Size</td><td>—</td><td colspan="3">SN</td></tr>
<tr><td>Offset + 0x18</td><td colspan="8">Super Channel Number</td><td>RXBM</td><td>SLIPIM</td><td colspan="2">—</td><td colspan="4">CASBS</td></tr>
</table>

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:

MPC8270, MPC8275, MPC8280

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:

MPC8270, MPC8275, MPC8280

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 fix plan at this time.

A-005319 Zero-bit insertion error in MCC non-supper channel HDLC mode

Devices:

MPC8280, MPC8275, MPC8270

Description:

When using the MCC controller in non-supper channel HDLC mode, the zero-bit insertion may not occur, depending on the data patterns that require zero-bit insertion right after the last data bit before the closing flag.

Impact:

The transmitted data violates the HDLC zero-insertion definition, resulting in an error in the receiving end.

Workaround:

- Use the microcode patch available from Freescale.

Or

- Utilize the upper layer protocol to re-transmit the data.

Fix plan:

No plans to fix

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Information in this document is provided solely to enable system and software implementers to use Freescale 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.

Freescale reserves the right to make changes without further notice to any products herein. Freescale makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale 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 Freescale 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. Freescale does not convey any license under its patent rights nor the rights of others. Freescale sells products pursuant to standard terms and conditions of sale, which can be found at the following address: http://www.reg.net/v2/webservices/Freescale/Docs/TermsandConditions.htm.

Document Number: MPC8280CE
Rev. 7
07/2012