Freescale Semiconductor
Errata

Document Number: MPC8260CE
Rev. 10, 07/2012

Chip Errata for the MPC8260 PowerQUICC II Family

This document describes all known silicon errata for the MPC8260 PowerQUICC II family of integrated communications processors. See Table 2 for a list of devices.

Table 1 summarizes this document’s revision history.

Table 1. Document Revision History

Rev. Number	Date	Substantive Changes
10	07/2012	• Added CPM131, CPM132, and A-005319 • Updated the affected devices in CPM130
9	07/2010	• Updated CPU2 to include a list of affected devices. • Added I2C1.
8	02/2010	• Added CPU2 • Updated CPM130
7	03/2008	• New errata: G13
6	11/2006	• New errata: PCI19, PCI20, CPM130
5	5/2006	• Updated template and made minor editorial corrections. • New errata: JTAG1, PCI18, CPM124, CPM126, CPM127, CPM128
4.7	7/2005	• Modified: PCI12: Added workaround • New errata: PCI16, CPM123
4.6	11/2004	• New errata: PCI14, PCI15, CPM120, CPM12 • Modified: CPM110: Replaced missing description and workaround.
4.5	8/2004	• Modified: CPM101 • New Errata: CPM118, CPM119

© 2003-2012 Freescale Semiconductor, Inc. All rights reserved.

Table 1. Document Revision History (continued)

Rev. Number	Date	Substantive Changes
4.4	4/2004	• Modified: CPM101, CPM114 • New Errata: CPM117
4.3	2/2004	• New errata: G8
4.2	12/2003	• New errata: CPM116
4.1	10/2003	• New errata:CPM114, CPM115
4	9/2003	• New errata: PCI11–PCI12, CPM10–CPM113 • Modified description of CPM57 • Addition of HiP7 devices • Addition of HiP4 Rev C.0
Prior to Revision 4		This document replaces three previous errata documents: • “MPC826x Family Device Errata” (MPC8260CE), Rev 3.1 • “XPC826xA Family Device Errata” (XPC8260ACE), Rev 1.3 • “MPC8260/XPC8260A Family Device Errata Summary” (MPC8260CESUMM), Rev 4.9

MPC8260 PowerQUICC II family devices are available in multiple silicon revisions, as shown in Table 2. To find which errata apply to a particular device, please refer to Table 3.

Table 2. MPC8260 Family Devices and Silicon Revisions

Device		Silicon							
	Process	0.29 µm (HiP3)					0.25 µm (HiP4)		
	Revision	A.1	B.1	B.2	B.3	C.2	A.0	B.1	C.0
	Mask	1K22A	1K23A	2K23A	3K23A	6K23A, 7K23A	2K25A	4K25A	5K25A
MPC8260(A)[1]		√	√	√	√	√	√	√	
MPC8250[2]							√	√[2]	√[2]
MPC8255(A)[1]		√	√	√	√	√	√	√	
MPC8264							√	√	
MPC8265							√	√	√
MPC8266							√	√	√

[1] ‘A’ designates HiP4 revisions of a device that was originally available in a HiP3 version.

[2] Also available in 516 PBGA (VR or ZQ) package in HiP4 Rev. B.1 and Rev. C.0 only.

Table 3 lists the silicon revisions to which each erratum applies.

Table 3. Errata Summary

Errata	.29 µm (HiP3)					.25 µm (HiP4)			Workaround	Description
	A.1	B.1	B.2	B.3	C.2	A.0	B.1	C.0		
System Interface Unit										
SIU1	√								—	Wrong timer advancement on RCCR
SIU2	√								Yes	DMA error on assertion of ARTRY
SIU4	√								Yes	Incorrect masking of MCP
SIU5	√								Yes	Incorrect report of TEA
SIU6	√								—	Software watchdog reports soft reset
SIU8	√	√	√	√	√				—	Parity when bus port size is less than 64 bits
SIU9		√	√	√					Yes	Bus monitor erroneously asserts TEA after ARTRY
SIU10		√	√	√					Yes	Strict enforcement of requirement to assert DBG and TS In the same cycle when core enabled
SIU12	√	√	√	√					Yes	In the core disabled mode, CPU_BR_B powers up in a random state
SIU13	√	√	√	√					—	SDAMUX not valid in single MPC8260 mode
SIU14						√			Yes	Errata in parity operation when BRx[DR] = 1
SIU16	√	√	√	√	√	√	√	√	Yes	Bus busy disable mode
SIU17	√	√	√	√	√	√	√	√	Yes	Bus error causes TEA to asserted twice
SIU18		√	√	√	√	√	√	√	—	ARTRY assertion when using pipeline depth of zero
SIU19	√	√	√	√	√	√	√	√	Yes	Bus monitor time-out when using external slave
I2C										
I2C1	√	√	√	√	√	√	√	√	Yes	Enabling I^{2}C could cause I^{2}C bus freeze when other I^{2}C devices communicate
General										
G1	√								—	Incorrect AC timings: outputs switch later than specified
G3						√			Yes	PLL does not lock on the rising edge of the input clock CLKIN
G4	√								Yes	Incorrect AC timings: outputs switch later than specified
G8	√	√	√	√	√	√	√	√	—	Assert PORESET to ensure correct JTAG operation
G13	√	√	√	√	√	√	√	√	Yes	SRESET might hang the device in PCI host mode
G14	√	√	√	√	√	√	√	√	Yes	
JTAG										
JTAG1	√	√	√	√	√	√	√	√	—	JTAG does not support the sample command

Table 3. Errata Summary (continued)

Errata	.29 μm (HiP3)					.25 μm (HiP4)			Workaround	Description
	A.1	B.1	B.2	B.3	C.2	A.0	B.1	C.0		
PCI										
PCI1						√	√	√	Yes	PCI DMA operation after bus error
PCI2						√	√	√	Yes	PCI I_2O operation
PCI3						√	√	√	—	PCI configuration registers, class code
PCI4						√			—	PCI TVAL hold time
PCI5						√	√	√	Yes	PCI does not negate ARTRY properly
PCI6						√	√	√	—	CPM frequency limitation in non-integer bus-to-CPM clock ratios in PCI mode
PCI7						√	√	√	Yes	Access to PCI memory-mapped configuration registers in non-PCI mode
PCI8						√	√	√	Yes	Output bus clock in PCI agent mode
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						√	√	√	Yes	Deassertion of 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
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	Inbound PCI Transaction with No Bytes Enabled May Cause the PCI Bus to Hang.
CPU										
CPU1	√								—	Error in MCP reporting
CPU2	√	√	√	√	√	√	√	√	—	Floating-point registers must be initialized before they are used for arithmetic operations
CPM										
CPM1	√								—	Erroneous LG error indication in MCC
CPM2	√								—	CAM access not atomic

Table 3. Errata Summary (continued)

Errata	.29 µm (HiP3)					.25 µm (HiP4)			Workaround	Description
	A.1	B.1	B.2	B.3	C.2	A.0	B.1	C.0		
CPM4	√	√	√	√	√				—	No CTS lost indication with HDLC
CPM5	√								—	Data corruption on DMA fly-by
CPM6	√	√	√	√	√				—	Erroneous report of overrun on FCC
CPM7	√	√	√	√	√				—	Erroneous report of overrun with fast Ethernet
CPM8	√	√	√	√	√				Yes	Error using FCC transmit on demand register
CPM9	√	√	√	√	√				Yes	Erroneous reception of ATM cell
CPM10	√	√	√	√	√				—	Error in ATM underrun report
CPM11	√	√	√	√	√				—	False indication of shared flag
CPM13	√	√	√	√	√				—	Error in random number generation
CPM14	√								Yes	Corruption of ATM cells using AAL1 and UDC
CPM15	√								Yes	Corruption of port D registers
CPM17	√	√	√	√	√				—	Error in reporting UTOPIA error condition
CPM18	√	√	√	√	√				Yes	Error in UTOPIA slave transmit mode
CPM21	√	√	√	√	√				Yes	False indication of collision in fast Ethernet
CPM22	√	√	√	√	√				—	False defer indication in fast Ethernet
CPM23	√								Yes	Corruption of AAL5 header
CPM24	√	√	√	√	√				—	Error in indicating IDLE between frame
CPM27	√	√	√	√	√				—	Error in heartbeat checking in FCC
CPM28	√								Yes	Error in receive frame threshold
CPM29	√								—	MAXD1 and MAXD2 may not be less than MFLR
CPM30	√								—	Graceful stop command does not work
CPM35	√								Yes	Data corruption in SCC transparent mode
CPM36	√								Yes	SI sync timing restriction
CPM38	√	√	√	√	√				—	Heartbeat error and carrier sense lost error on two frames
CPM39	√								Yes	Corruption in AAL0 cell payload
CPM40	√								Yes	Corruption in AAL0 idle cell
CPM41	√								Yes	Limitation in ATM controller
CPM42	√								Yes	Data corruption in MCC
CPM43	√								Yes	TxCLAV ignored by UTOPIA in single PHY mode
CPM44	√								Yes	Zero insertion error on MCC
CPM45	√								Yes	Error in CLAV sample point

Table 3. Errata Summary (continued)

Errata	.29 µm (HiP3)					.25 µm (HiP4)			Workaround	Description
	A.1	B.1	B.2	B.3	C.2	A.0	B.1	C.0		
CPM46	√								—	Error in internal prioritization of CPM resource
CPM47	√								—	Error in three-state ability of two TxDATA signals using 16-bit UTOPIA interface on FCC1
CPM48	√	√	√	√	√				Yes	Error in TDM
CPM49	√								Yes	Error in FEC CAM address recognition
CPM50	√	√	√	√	√				Yes	Error in loss of alignment
CPM51	√	√	√						Yes	Pointer insertion/extraction error in AAL1 CES
CPM52			√						Yes	Error in ATM internal rate mode
CPM53		√	√						—	Inability to run RAM microcode
CPM54	√	√	√	√	√				Yes	Error in switching to and from shadow SI RAM
CPM55	√	√	√						—	Error in ATM_Transmit command
CPM56				√	√				Yes	AAL2 microcode in ROM does not function
CPM57	√	√	√	√					Yes	AAL5 cell corruption
CPM62	√	√	√	√					Yes	CPM PLL does not lock reliably for certain multiplication factors
CPM64	√	√	√	√		√			Yes	AAL5 RxBD[LNE] error generated if PDU length exceeds 65,512 bytes
CPM65		√	√	√	√				Yes	SS7 microcode in ROM is not fully functional
CPM71		√	√	√					Yes	CPM does not snoop MCC buffer descriptors
CPM72	√	√	√	√					—	MCC global underruns
CPM73	√	√	√	√					Yes	SIRAM corruption
CPM75					√	√	√	√	Yes	AAL2 microcode in ROM is not fully functional
CPM77						√			Yes	TC layer transmits and receives data LSB first instead of MSB first
CPM78						√	√	√	—	IMA microcode in ROM is not fully functional
CPM79	√								—	FCC fast Ethernet flow control
CPM80		√	√	√	√	√			Yes	MCC CES user template
CPM81						√			Yes	Japanese SS7 error interval timer problem
CPM85	√	√	√	√	√	√			—	AAL0 only one BSY interrupt generated
CPM86	√	√	√	√	√	√			Yes	Random PHY number for FCC Rx in single-PHY master mode
CPM88	√	√	√	√	√	√			Yes	MCC transmit GUN when ‘MCC STOP RX’ CPCR command is used
CPM92							√	√	Yes	TC Layer when disabled can be selected by FCC2

Table 3. Errata Summary (continued)

Errata	.29 µm (HiP3)					.25 µm (HiP4)			Workaround	Description
	A.1	B.1	B.2	B.3	C.2	A.0	B.1	C.0		
CPM93	√								Yes	IDMA microcode in ROM is not fully functional
CPM94						√	√	√	Yes	FCC $\overline{RTS}$ signal not asserted correctly
CPM95	√	√	√	√	√				Yes	ATM false indication of miss inserted cells
CPM96		√	√	√	√	√	√	√	Yes	ATM performance monitoring with AAL1 CES
CPM97						√	√	√	Yes	MCC SS7—No SUERM interrupt generated after an ABORT
CPM98	√	√	√	√	√	√	√	√	Yes	I^2C erratic behavior can occur if extra clock pulse is detected on SCL
CPM99		√	√	√	√	√	√	√	Yes	ABR TCTE[ER-TA] corruption
CPM100		√	√	√	√	√			Yes	ABR TCTE address miscalculation
CPM101						√	√	√	Yes	FCC RxClav timing violation (slave)
CPM110	√	√	√	√	√	√	√	√	Yes	FCC1 prioritization
CPM111	√	√	√	√	√	√	√	√	Yes	FCC missing reset at overrun
CPM112	√	√	√	√	√	√	√	√	Yes	FCC missing status
CPM113	√	√	√	√	√	√	√	√	—	Incorrect return value from event register read (SCC, SPI, I2C, and SMC)
CPM114	√	√	√	√	√	√	√	√	Yes	IDMA transfer has an extra 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
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
CPM130	√	√	√	√	√	√	√	√	Yes	Possible corrupted SP in AAL1 STF mode

Table 3. Errata Summary (continued)

Errata	.29 µm (HiP3)					.25 µm (HiP4)			Workaround	Description
	A.1	B.1	B.2	B.3	C.2	A.0	B.1	C.0		
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

SIU1: Wrong timer advancement on RCCR

Devices:

MPC8260, MPC8255

Description:

The MPC8260 treats the RCCR[TIMEP] value (UC timer) differently than QUICC. In QUICC, the timer advanced (N+1)*1024 cycles and in XPC8260 the timer advances N*1024 cycles.

Workaround:

None

Fix Plan:

Fixed on Rev. B.1.

SIU2: DMA error on assertion of $\overline{\text{ARTRY}}$

Devices:

MPC8260, MPC8255

Description:

Address retry assertion by a bus master on the 60x bus may confuse the XPC8260 DMA. Examples of external masters that use address retry are PCI bridge processor, which uses cache in copy-back mode, and some ASICS.

Workaround:

- Place XPC8260 in single 8260 bus mode.
- Work with external L2 cache in write-through mode.
- Inhibit external PCI bridge from using address retry.
- Design ASIC interface not to use address retry.

If an external PCI bridge exists and it has the capability to drive $\overline{\text{ARTRY}}$, only the CPM DMA must be allowed to perform accesses through the PCI bridge after the CPM has been activated (that is, no core accesses through the PCI bridge after this point).

Fix Plan:

Fixed on Rev. B.1.

SIU4: Incorrect masking of $\overline{\text{MCP}}$

Devices:

MPC8260, MPC8255

Description:

$\overline{\text{MCP}}$ (machine check interrupt) due to data errors (parity/ECC) is masked by the SWRI bit in SYPCR.

Workaround:

Clear the SWRI bit in SYPCR to get data error indication.

Fix Plan:

Fixed on Rev. B.1.

SIU5: Incorrect report of $\overline{\text{TEA}}$

Devices:

MPC8260, MPC8255

Description:

Data parity error in the local bus does not cause $\overline{\text{MCP}}$ interrupt.

Workaround:

Use $\overline{\text{TEA}}$ for parity error in local bus by setting bit 16 of L_TESCR1.

Fix Plan:

Fixed on Rev. B.1.

SIU6: Software watchdog reports soft reset

Devices:

MPC8260, MPC8255

Description:

The events from software watchdog and bus monitor should cause assertion of hard reset. Currently, these events cause assertion of soft reset.

Workaround:

None

Fix Plan:

Fixed on Rev. B.1.

SIU8: Parity when bus port size is less than 64 bits

Devices:

MPC8260, MPC8255

Description:

When reading from a device with a port size of less than 64 bits, from an address not aligned to 64 bits, the parity bits for parity check are not taken from the write locations (for example, for a read of 4 bytes from a 32-bit port size from address 4, the parity is checked against dp[4:7] when it should be checked against dp[0:3]). The bug exists for both normal and RMW parity, and for both the 60x and local buses.

Workaround:

None

Fix Plan:

Fixed on Rev. A.0.

SIU9: Bus monitor erroneously asserts $\overline{\text{TEA}}$ after $\overline{\text{ARTRY}}$

Devices:

MPC8260, MPC8255

Description:

The bus monitor will assert $\overline{\text{TEA}}$ if a bus cycle does not complete in a certain amount of time. In case there is an $\overline{\text{ARTRY}}$ cycle, the bus monitor will not recognize the complication of the transaction and will assert $\overline{\text{TEA}}$ if there is no bus activity following the $\overline{\text{TEA}}$ for a time equal to the bus monitor time-out.

Workaround:

Disable the bus monitor in system where $\overline{\text{ARTRY}}$ cycles are used, for example, systems where external PCI bridge chip is used.

Fix Plan:

Fixed on Rev. C.2.

SIU10: Strict enforcement of requirement to assert $\overline{\text{DBG}}$ and $\overline{\text{TS}}$ in the same cycle when core enabled

Devices:

MPC8260, MPC8255

Description:

In systems where the XPC8260 core is enabled, an external arbiter must assert $\overline{\text{DBG}}$ in the same clock in which $\overline{\text{TS}}$ is asserted (there may be a one-clock delay in the PPC_ACR[DBGD] bit is set; however, out of reset this bit is not set by default). Some external arbiters, including the one implemented in Tundra PowerSpan device, do not meet this requirement. As a result, the system gets stuck following the first bus access after reset. In XPC8260 Rev. A.1 silicon this condition was relaxed and $\overline{\text{DBG}}$ could be asserted one clock later. However, in XPC8260 Rev. B.X silicon, this requirement is strictly enforced, and some systems that include external arbiters that worked with Rev. A.1 silicon will not work with Rev. B.X.

Workaround:

Do not use the external arbiter with Rev. B.X silicon, which does not meet the above requirement to assert $\overline{\text{DBG}}$ and $\overline{\text{TS}}$ in the same cycle, or, alternatively, use the internal MPC8260 bus arbiter.

Fix Plan:

Fixed on Rev. C.2.

SIU12: In the core disabled mode, CPU_BR_B powers up in a random state

Devices:

MPC8260, MPC8255

Description:

When the core is disabled, the CPU_BR_B (bus request by the 603 core) powers up in random state. If it happens to power-up asserted it will request the bus all the time, and the internal arbiter will be deadlocked, disregarding the bus requests from the external devices. As a result, the system does not start fetching instructions after reset.

Workaround:

Set the MMR field in the hard reset configuration word (HRCW) to 0b10. This will mask the core bus request and allow the bus to be granted to external master 1. As a side effect it will also mask the bus requests from the external masters 2 and 3. The SIUMCR[MMR] field must not be changed after power-on reset. The bus requests issued by the CPM (that is, SDMA) will not be masked.

Fix Plan:

Fixed on Rev. C.2.

SIU13: SDAMUX not valid in single MPC8260 mode

Devices:

MPC8260, MPC8255

Description:

SDAMUX signal is disabled (stuck at 0) when SDRAM machine handles the memory access and the chip is programmed to single MPC8260 mode (BCR[EBM] = 0).

Workaround:

None

Fix Plan:

Fixed on Rev. C.2.

SIU14: Errata in parity operation when BRx[DR] = 1

Devices:

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

Description:

When reading from a device with a port size less then 64 bits, parity checking is not performed at the right location when BRx[DR] = 1, and parity errors will result. This problem exists for both normal and RMW parity, and for both the 60x and local buses.

Workaround:

Use BRx[DR] = 0 for no data pipelining.

Fix Plan:

Fixed on Rev. B.1.

SIU16: Bus busy disable mode

Devices:

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

Description:

Bus busy disable mode (SIUMCR[BBD = 1]) cannot be used if the 826x or the 826xA is not the only master on the 60x bus. Using this mode in such a system can cause the 60x bus to hang.

Workaround:

- If the external master supports the $\overline{\text{ABB}}$ signal, do not use bus busy disable mode and connect this signal to the 826x or the 826xA. The DBB signal can either be connected or can be pulled up.
- If the external master doesn't support the $\overline{\text{ABB}}$ signal, do one of the following:
  - — Do not use bus busy disable mode and generate the $\overline{\text{ABB}}$ signal externally. The $\overline{\text{DBB}}$ signal can either be connected or can be pulled up. The following external $\overline{\text{ABB}}$ implementation should be enough to work around the problem: assert the $\overline{\text{ABB}}$ signal whenever a qualified bus grant for the external master is sampled (bus grant asserted while $\overline{\text{ARTRY}}$ and $\overline{\text{ABB}}$ are negated). Negate the $\overline{\text{ABB}}$ signal when there is no qualified bus grant. The negation of $\overline{\text{ABB}}$ should be as follows: drive $\overline{\text{ABB}}$ to VDD for half a clock cycle and then stop driving it (HIGH-Z).
  - — If using the internal arbiter and up to two external masters, connect the external bus grants (through an AND gate if more than one) to an available external bus request and define the priority for that request to be the highest in the PPC_ALRH register. The DBB signal can either be connected or can be pulled up.

Note that the second workaround should be implemented only for external masters that do not support $\overline{\text{ABB}}$.

If some of the masters support $\overline{\text{ABB}}$ and another masters do not, then the masters that do support it should simply connect the $\overline{\text{ABB}}$ signal to the PQ2.

Fix Plan:

No fix plan at this time.

SIU17: Bus error causes $\overline{\text{TEA}}$ to asserted twice

Devices:

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

Description:

When the local bus is accessed through the 60x bus, this single transaction turns on both local bus and 60x bus monitors. If there is bus error, then $\overline{\text{TEA}}$ will be asserted twice a few cycles apart. If the 826x initiates this transaction, then $\overline{\text{TEA}}$ being asserted twice will result in the CPU entering a checkstop state.

Workaround:

Disable 60x bus monitor, SYPCR[PBME] = 0.

Fix Plan:

No fix plan at this time.

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

Devices:

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

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 $\overline{\text{AACK}}$, 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.

SIU19: Bus monitor time-out when using external slave

Devices:

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

Description:

When using an external 60x bus slave with the bus monitor activated. $\overline{\text{PSDVAL}}$ is not asserted when the external slave is accessed, which could cause the bus monitor to time-out and $\overline{\text{TEA}}$ to be asserted.

Workaround:

- Use pipeline depth of zero (BCR[PLDP] = 1) when using an external 60x bus slave
- Disable 60x bus monitor, SYPCR[PBME] = 0
- If the external 60x bus slave is another 826x device, connect the $\overline{\text{PSDVAL}}$ signals together

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 I2C*n*CR[MEN] = 1, and starts a timer;
2. Delay 4 I^{2}C bus clocks.
3. Check Bus Busy bit (I2C*n*SR[MBB]).
   If MBB = 0, jump to step 6; (Good condition. Go to normal operation)
   else,
   Disable device (I2C*n*CR[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.

G1: Incorrect AC timings: outputs switch later than specified

Devices:

MPC8260, MPC8255

Description:

The maximum output delay defined by sp32 is 10 ns. The specification is 8 ns.

The maximum output delay defined by sp35 is 7.5 ns. The specification is 7 ns.

The maximum output delay defined by sp36a is 6.5 ns. The specification is 6 ns.

Workaround:

None

Fix Plan:

Fixed on Rev. C.2.

G2: PLL does not lock on the rising edge of the input clock CLKIN

Devices:

MPC8260A, MPC8255A, MPC8250, MPC8264, MPC8265, MPC8266

Description:

In correct operation, the PLL of the MPC826x devices will lock on the rising edge of the input clock. However, on the MPC826xA (Hip4) silicon, the PLL locks on the falling edge of the input clock if an integer CPM multiplication factor is used. This will affect the skew between CLKIN and internal clock at the rising edge since the skew is dependent on the duty cycle of the input clock. This will affect synchronous designs where the same clock source is used as an input to CLKIN as well as to an external synchronous device (for example, a peripheral or ASIC). The MPC826xA internal logic assumes that the internal clock's rising edge will be in sync with CLKIN.

Workaround:

Use a non-integer CPM multiplication factor. This workaround does not apply to PCI agent mode, since the multiplication factor in PCI agent mode must be an integer.

NOTE

Please note that in PCI agent mode, since the clock provided by the user is the PCI clock, the skew due to non-50% duty cycle will be seen between the PCI clock and the internal clock. The bus clock in this case is supplied by the 8260, and if the clk2 skew elimination function is used, then the internal clock will be in phase with the bus clock.

Fix Plan:

Fixed on Rev. B.1.

G3: Incorrect AC timings: outputs switch later than specified

Devices:

MPC8260, MPC8255

Description:

The maximum output delay defined by sp34 is 11 ns. The specification is 6 ns.

Workaround:

Set P/LSDMR [BUFCMD] = 1 or set P/LSDMR [EAMUX] = 1.

Fix Plan:

Fixed on Rev. B.1.

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

Devices:

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

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.

Workaround:

None

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.

JTAG1: JTAG does not support the sample command

Devices:

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

Description:

JTAG does not support the sample command.

Workaround:

None

Fix Plan:

No fix plan at this time.

PCI1: PCI DMA operation after bus error

Devices:

MPC8250, MPC8265, MPC8266

Description:

The PCI DMA may lock up if it attempts to perform a DMA cycle following a cycle that was terminated by a bus error.

Workaround:

On a PCI-enabled system, one must reset the system if a bus error is detected.

Fix Plan:

No fix plan at this time.

PCI2: PCI I_2O operation

Devices:

MPC8250, MPC8265, MPC8266

Description:

When the conditions for the outbound post queue interrupt assertion are valid, and OMIMR[OPQIM] is set, OMISR[OPQI] is cleared.

Workaround:

Do not read the state of OMISR[OPQI] when OMIMR[OPQIM] is set.

Fix Plan:

No fix plan at this time.

PCI3: PCI configuration registers, class code

Devices:

MPC8250, MPC8265, MPC8266

Description:

When configured as a PCI agent device, the value of the base class code, subclass, and interface registers are 0x0e, 0x00, and 0x01 respectively, indicating that it supports the I_2O protocol. Since the I_2O support is not fully compliant, future revisions of the MPC826x will not use this value in the class code fields.

Workaround:

None

Fix Plan:

No fix plan at this time.

PCI4: PCI TVAL hold time

Devices:

MPC8250, MPC8265, MPC8266

Description:

PCI bridge only supports a 1 ns hold time for Tval. Tval is a timing spec listed in the PCI-SIG that specifies CLK to signal valid. To support 33 MHz, the minimum must be 2 ns.

Workaround:

None

Fix Plan:

Fixed on Rev. B.1.

PCI5: PCI does not negate $\overline{\text{ARTRY}}$ properly

Devices:

MPC8250, MPC8265, MPC8266

Description:

For normal recommended pull-up resistor (10 KΩ), $\overline{\text{ARTRY}}$ is negated 2 cycles after $\overline{\text{AACK}}$ assertion instead of 1. Under certain PCI traffic patterns, this extra cycle $\overline{\text{ARTRY}}$ assertion could lead to 60x bus deadlock.

Workaround:

Use a 300-Ω pull-up resistor on $\overline{\text{ARTRY}}$

Fix Plan:

No fix plan at this time.

PCI6: CPM frequency limitation in non-integer bus-to-CPM clock ratios in PCI mode

Devices:

MPC8250, MPC8265, MPC8266

Description:

In PCI mode (agent or host), when the clock ratio between bus clock and CPM clock is not an integer ratio (1:2.5, 1:3.5) the frequency of the CPM is limited to 166 MHz.

Workaround:

None

Fix Plan:

No fix plan at this time.

PCI7: Access to PCI memory-mapped configuration registers in non-PCI mode

Devices:

MPC8250, MPC8265, MPC8266

Description:

In non-PCI mode, the internal memory space (DPRAM, registers, and the local bus bridge) will not be accessible any more after an access to the PCI memory-mapped configuration registers area (offsets 10400–10BFF). The access to the above area and any following access to the internal memory space will not be terminated normally and can only be terminated by $\overline{\text{TEA}}$ if the 60x bus monitor is activated. The system can recover only after a soft reset.

Workaround:

In non-PCI mode, do not access the described area. Note that this area is reserved in non-PCI mode.

Fix Plan:

No fix plan at this time.

PCI8: Output bus clock in PCI agent mode

Devices:

MPC8250, MPC8265, MPC8266

Description:

In PCI agent mode, the DLL output reference clock is used externally as the bus clock. If the CPM/BUS ratio is non-integer (2.5 and 3.5), the output bus clock from the DLL will not have a 50% duty cycle. Instead the output bus clock will have a duty cycle of:

- 2.5—40% or 60%
- 3.5—43% or 57%

Workaround:

In hardware add a zero delay buffer to the DLLOUT signal.

Fix Plan:

No fix plan at this time.

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

Devices:

MPC8250, MPC8265, MPC8266

Description:

In PCI mode, when both outbound and inbound traffic happens simultaneously, it is possible that after some random number of transactions the system will hang. The deadlock situation might 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, or
- 60x master reads from PCI memory/IO/config space while a PCI bridge's DMA channel writes into 60x memory, or
- 60x master reads from PCI bridge's internal register while a PCI master writes into 60x memory, or
- 60x master reads from PCI bridge's internal register while a PCI bridge's DMA channel writes into 60x memory.

Workaround:

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) should be selected to locate on local bus in the IDMA BD (SDTB = 1).

Fix Plan:

Fixed on Rev. C.0.

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

Devices:

MPC8250, MPC8265, MPC8266

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, or 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 an outbound translation window can not 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:

MPC8250, MPC8265, MPC8266

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 deasserted, the PCI bus can hang.

The PCI bus can potentially hang if configuration write transactions are retried in host mode and other masters are requesting the PCI bus. In agent mode the IOU will not be mastering configuration transactions so there shouldn't be any problems.

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 deasserted, 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 hang on the PCI bus manifests itself by $\overline{\text{FRAME}}$ being asserted without the assertion of $\overline{\text{IRDY}}$ indefinitely or no assertion of $\overline{\text{FRAME}}$ at all.

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:

MPC8250, MPC8265, MPC8266

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:

MPC8250, MPC8265, MPC8266

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:

MPC8250, MPC8265, MPC8266

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.

PCI18: PCI streaming problem when the latency timer is zero

Devices:

MPC8250, MPC8265, MPC8266

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:

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.

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:

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.

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.

CPU1: Error in $\overline{\text{MCP}}$ reporting

Devices:

MPC8260, MPC8255

Description:

Errors that cause $\overline{\text{MCP}}$ assertion are not reported in any status register. When $\overline{\text{MCP}}$ is asserted the CPU will branch to bus error vector (0x200) but it is not possible to know the cause of the error. Data errors including data parity and ECC are also reported by $\overline{\text{MCP}}$ assertion.

Workaround:

None

Fix Plan:

Fixed on Rev. B.1.

CPU2: Floating-point registers must be initialized before they are used for arithmetic operations

Devices:

MPC8260, MPC8255, MPC8250, MPC8264, MPC8265, MPC8266

Detailed Description:

The 64-bit FPRs each have additional internal bits with which they are associated that specify the type of floating-point number in the register. These bits are set properly whenever the FPR is loaded. However, the part could power up with the internal bits randomly set. The FPR could be interpreted as containing a denormalized number with a mantissa that contains all zeros. If this random state is used for arithmetic operations or stored with an **stfd** instruction before a floating-point load operation corrects the internal bits, the part hangs or has floating point exceptions while searching for a leading 1 in the mantissa. This behavior may be seen when doing arithmetic operations (for example, addition or subtraction) or when the **stfd** instruction is used.

Projected Impact:

This error affects all systems that use floating-point operations.

Workaround:

When emerging from reset, initialize all the FPRs that will be used. The initialization value is not important.

Projected Solution:

Fixed in documentation.

CPM2: Erroneous LG error indication in MCC

Devices:

MPC8260, MPC8255

Description:

In MCC HDLC when MFLR is a multiple of 8 and the frame length is exactly MFLR there might be LG error indication and interrupt on this frame.

Workaround:

None

Fix Plan:

Fixed on Rev. B.1.

CPM3: CAM access not atomic

Devices:

MPC8260, MPC8255

Description:

The bus tonicity mechanism for CAM access may not function correctly when the CPM’s DMA accesses the CAM. This only affects systems in which multiple CPM’s will access the CAM.

Workaround:

None

Fix Plan:

Fixed on Rev. B.1.

CPM4: No $\overline{\text{CTS}}$ lost indication with HDLC

Devices:

MPC8260, MPC8255

Description:

When $\overline{\text{CTS}}$ is negated at the end of HDLC frame, (last flag or 1 byte before) transmission will be aborted, however, there will be no $\overline{\text{CTS}}$-lost indication. There will be only an abort indication.

Workaround:

None

Fix Plan:

Fixed on Rev. A.0.

CPM5: Data corruption on DMA fly-by

Devices:

MPC8260, MPC8255

Description:

The data of a DMA write, which follows a DMA fly-by read in the local bus, may be corrupted.

Workaround:

None

Fix Plan:

Fixed on Rev. B.1.

CPM6: Erroneous report of overrun on FCC

Devices:

MPC8260, MPC8255

Description:

Spurious overrun indications on the FCC may occur in the following cases:

- After issuing stop transmit command.
- Following $\overline{\text{CTS}}$ lost condition.
- Late collision under Ethernet.

Workaround:

None

Fix Plan:

Fixed on Rev. A.0.

CPM7: Erroneous report of overrun with fast Ethernet

Devices:

MPC8260, MPC8255

Description:

In case the CRS (carrier sense) signal is negated while fast Ethernet frame is transmitted, an overrun error might occur and the FCC may have to be reset.

Workaround:

None

Fix Plan:

Fixed on Rev. A.0.

CPM8: Error using FCC transmit on demand register

Devices:

MPC8260, MPC8255

Description:

The TODR mechanism may freeze an FCC serial channels.

Workaround:

For an FCC do not use the TODR.

Fix Plan:

Fixed on Rev. A.0.

CPM9: Erroneous reception of ATM cell

Devices:

MPC8260, MPC8255

Description:

Under certain condition ATM receiver may receive cells of PHYs that were not addressed for it.

Details of the condition:

- ATM receiver in UTOPIA slave mode.
- FIFO full condition occurred (this will happen only when the transmitter violates).
- The UTOPIA standard requirements: transmits data without CLAV).
- Transmitter changed selected PHY number.
- FIFO full condition ended (CPM read some data from FIFO).

Workaround:

Use different VPI/VCI for different PHYs or expect the cells to be discarded by higher level protocol software.

Fix Plan:

Fixed on Rev. A.0.

CPM10: Error in ATM underrun report

Devices:

MPC8260, MPC8255

Description:

In ATM, transmit internal rate underrun error is not reported correctly in TIRU in FCCE register. In most cases, TIRU will not be set in FCCE when internal rate underrun error occurs. In some rare cases that depend on internal sequences within the communication controller, the TIRU bit might be set as expected when the error should be reported.

Workaround:

None

Fix Plan:

Fixed on Rev. A.0.

CPM11: False indication of shared flag

Devices:

MPC8260, MPC8255

Description:

FCC-TX HDLC–FCT_TXD (data out) changes from 1-->0 for 1 ser_clock period, few clocks after reset command from MAIN is given. A false shared flag can be detected at the receiver, if last bit before reset was 0, and the receiver will consider it as a closing flag of the frame. In most cases, a CRC error will be generated and the frame will be discarded.

Workaround:

None

Fix Plan:

Fixed on Rev. A.0.

CPM13: Error in random number generation

Devices:

MPC8260, MPC8255

Description:

In fast Ethernet transmit, when more than one (up to four) frames reside in the FCC FIFO, random number generation (for collision wait) may produce the same number for all four frames.

Workaround:

None

Fix Plan:

Fixed on Rev. A.0.

CPM14: Corruption of ATM cells using AAL1 and UDC

Devices:

MPC8260, MPC8255

Description:

Corruption of ATM cells may occur when using the following combination: AAL1 with UDC in which the user defined header size is 9 to 12 octets and PM is not used.

Workaround:

Since this problem appears in a very specific condition as described above, avoiding any of the elements (for example, use cell header of 8 octets) will eliminate it.

Fix Plan:

Fixed on Rev. B.1.

CPM15: Corruption of port D registers

Devices:

MPC8260, MPC8255

Description:

The PDATA, PDATB, PDATC, and PDATD registers can only be written with a 32-bit write instruction. (that is, store word stw) When using 8- or 16-bit write instruction (that is, store half-word sth, or store byte stb instructions), the bits not being written to may be corrupted.

Workaround:

Use a 32-bit write instruction only to write to PDATA, PDATB, PDATC, and PDATD registers.

Fix Plan:

Fixed on Rev. B.1.

CPM17: Error in reporting UTOPIA error condition

Devices:

MPC8260, MPC8255

Description:

The FCC receiver, which is configured as single PHY master, does not detect UTOPIA error condition when SOC and CLAV are not asserted simultaneously.

Workaround:

None

Fix Plan:

Fixed on Rev. A.0.

CPM18: Error in UTOPIA slave transmit mode

Devices:

MPC8260, MPC8255

Description:

When the XPC8260 is configured to work as UTOPIA slave device in multi-slave PHY mode, and the TXEN input signal is asserted constantly, the XPC8260 will transmit one cell and transmission will stop.

Workaround:

Ensure that TXEN is negated whenever the TXCLAV is negated.

Fix Plan:

Fixed on Rev. A.0.

CPM21: False indication of collision in fast Ethernet

Devices:

MPC8260, MPC8255

Description:

In the fast Ethernet a false COL will be reported whenever a collision occurs on the preamble of the previous frame.

Workaround:

Software should ignore COL indications when the CRC of the frame is correct.

Fix Plan:

Fixed on Rev. A.0.

CPM22: False defer indication in fast Ethernet

Devices:

MPC8260, MPC8255

Description:

In the fast Ethernet if a frame was transmitted due to defer and this frame also got late collision, a false defer indication will be indicated for the next frame.

Workaround:

None

Fix Plan:

Fixed on Rev. A.0.

CPM23: Corruption of AAL5 header

Devices:

MPC8260, MPC8255

Description:

When working in AAL5 user defined cell mode and CPCS_UU is disabled, the UDC header may be corrupted.

Workaround:

When working with AAL5 UDC, use CPCS enabled mode.

Fix Plan:

Fixed on Rev. B.3.

CPM24: Error in indicating IDLE between frame

Devices:

MPC8260, MPC8255

Description:

In FCC HDLC transmitter, if slow serial clock (CPM _freq/serial_clock > 16) is used, $\overline{\text{RTS}}$ will not transition to IDLE between frames. This means that all the frames will be transmitted back-to-back in case there is valid data in the transmitter’s FIFO.

Workaround:

None

Fix Plan:

Fixed on Rev. A.0.

CPM27: Error in heartbeat checking in FCC

Devices:

MPC8260, MPC8255

Description:

The heartbeat checking in FCC transmit Ethernet 10 Mbps does not work properly. The standard requires that the collision pulse from the PHY should be checked within a window of 4 µsec from the falling edge of the carrier sense. The XPC8260 samples the collision signal only once at exactly 4 µsec (10 serial clocks) after the falling edge of the carrier sense signal.

Workaround:

None

Fix Plan:

Fixed on Rev. A.0.

CPM28: Error in receive frame threshold

Devices:

MPC8260, MPC8255

Description:

In SCC Rx in HDLC mode RFTHR does not work. There is no way to get interrupt on the receive side after a programmable number of frames.

Workaround:

RFTHR should be programmed to 1.

Fix Plan:

Fixed on Rev. B.1.

CPM29: MAXD1 and MAXD2 may not be less than MFLR

Devices:

MPC8260, MPC8255

Description:

In SCC Rx Ethernet, the option of transferring only part of a frame into memory (MAXD1 and MAXD2 < MFLR) does not work.

Workaround:

None

Fix Plan:

Fixed on Rev. B.1.

CPM30: Graceful stop command does not work

Devices:

MPC8260, MPC8255

Description:

Graceful stop command does not work in SCC Tx in the following protocols: Ethernet, HDLC, transparent.

Workaround:

None

Fix Plan:

Fixed on Rev. B.1.

CPM35: Data corruption in SCC transparent mode

Devices:

MPC8260, MPC8255

Description:

(4350) When using SCC transparent, envelope mode and the received frame size is (4*n) + 1, the last byte is corrupted. When using GSMR_H(RFW) – rx FIFO width, the received data is completely corrupted, not just the last byte.

Workaround:

Use the microcode patch available from Freescale.

Fix Plan:

Fixed on Rev. B.1.

CPM36: SI sync timing restriction

Devices:

MPC8260, MPC8255

Description:

SI’s sync signal may not change exactly on clock edge in the following cases.

The bug affects operation only when the SI is in one of two modes:

- sd = 00, ce = 0, fe = 0, dsc=1
  - (Sync sampled with falling edge -> sync should not change on rising edge)
- sd = 00, ce = 1, fe = 1, dsc=1
  - (Sync sampled with rising edge -> sync should not change on falling edge)

Workaround:

When working in these modes the sync signal to the SI should be manipulated such that it will not change on the exact edge of the serial clock. The errata can be worked around by toggling the sync at least 5 ns after the edge. One way to implement such a workaround is adding a non-inverting buffer between the device that generates the sync signal and the XPC8260 that uses it.

Fix Plan:

Fixed on Rev. B.1.

CPM38: Heartbeat error and carrier sense lost error on two frames

Devices:

MPC8260, MPC8255

Description:

There are rare cases when heartbeat error and carrier sense lost error will be reported on two frames. The error is reported in the frame in which it occurred, but in those rare cases it is also reported on an adjacent frame.

Workaround:

None

Fix Plan:

Fixed on Rev. A.0.

CPM39: Corruption in AAL0 cell payload

Devices:

MPC8260, MPC8255

Description:

There is a rare case when using ATM AAL0 transmitter that the AAL0 cell payload may be corrupted. This can occur in certain internal sequence of events that can not be controlled or detected by the user.

Workaround:

Use the microcode patch available from Freescale. Alternatively, when working with AAL0 SAR, place the TCELL_TMP_BASE 64-byte align plus 4. For example, use TCELL_TMB_BASE = 0x2d04 not 0x2d00.

Fix Plan:

Fixed on Rev. B.1.

CPM40: Corruption in AAL0 idle cell

Devices:

MPC8260, MPC8255

Description:

There is a rare case when transmitting ATM idle cell that the idle cell may be corrupted. This can occur in certain internal sequence of events that can not be controlled or detected by the user.

Workaround:

Place the idle base template at address 64-byte align minus 4. For example, use Idle_BASE = 0x2cfc not 0x2d00.

Fix Plan:

Fixed on Rev. B.1.

CPM41: Limitation in ATM controller

Devices:

MPC8260, MPC8255

Description:

There are some limitations in the ATM controller. The first limitation is that only the first 8 PM tables can be used instead of 64. When using these 8 tables, the user must clear the 5 most significant bits of TBD_BASE (in case of Tx PM) or RBD_BASE (in case of Rx PM).

The second limitation is that only the first 2048 ATM channel numbers can be used.

Workaround:

Use the microcode patch available from Freescale which fixes the performance monitoring limitation problem described above. However, the ATM channel number limitation has no workaround.

Fix Plan:

Fixed on Rev. B.1.

CPM42: Data corruption in MCC

Devices:

MPC8260, MPC8255

Description:

Data corruption may occur in the receive buffers of MCC channels when more than 1 TDM slot uses 7 bits of contiguous data.

Workaround:

It is possible to avoid this problem by splitting the 7-bit slots between 2 SI RAM entries such that 1 entry will represent 4 bits of the slot and the other SI entry will represent 3 bits of the slot. When initializing the 2 spliced channels, shadow RAM must be used to keep the 2 channels synchronized. This problem occurs only when all the 7 bits are represented by 1 entry in the SI RAM.

Fix Plan:

Fixed on Rev. B.1.

CPM43: TxCLAV ignored by UTOPIA in single PHY mode

Devices:

MPC8260, MPC8255

Description:

When the FCC transmitter is configured to work in UTOPIA single PHY master mode, it will ignore negation of the TxCLAV signal. Thus, due to this errata the UTOPIA slave will be unable to control the flow of cells by negating TxCLAV.

Workaround:

Configure the FCC transmitter and receiver to work in multi-PHY master mode by programming FPSMR[TUMP/RUMP] = 1 and limit the number of ATM PHYs to 1 by programming FPSMR[LAST PHY] = 00000.

Fix Plan:

Fixed on Rev. B.1.

CPM44: Zero insertion error on MCC

Devices:

MPC8260, MPC8255

Description:

When using MCC transmitter in HDLC super channel mode, a zero insertion at the last bit before the flag may fail to occur.

Workaround:

Use the microcode patch available from Freescale.

Fix Plan:

Fixed on Rev. B.1.

CPM45: Error in CLAV sample point

Devices:

MPC8260, MPC8255

Description:

In FCC ATM transmit master mode (multiple PHY only), the CLAV signal is sampled 5 clocks before the end of the cell instead of 4 clocks. This is relevant only for back-to-back transmission sequence.

Workaround:

In multiple PHY fix priority polling mode, by adding a dummy PHY, it is possible to ensure that the dummy PHY will be polled at payload 44 (5 clocks before the end of the cell). This is possible since the cell length is constant and the number of PHY to poll is also constant.

Fix Plan:

Fixed on Rev. B.1.

CPM46: Error in internal prioritization of CPM resource

Devices:

MPC8260, MPC8255

Description:

Each of the communication controllers (FCC, MCC, SCC, ...) issue request for service to the CPM with different priorities in order to receive the necessary assistance in time. Because of an internal connection error, the FCC3 request for service is issued with a much lower priority than intended. Because of this, FCC3 might sporadically overrun when the CPM is heavily loaded.

Workaround:

None

Fix Plan:

Fixed on Rev. B.1.

CPM47: Error in three-state ability of two TxDATA signals using 16-bit UTOPIA interface on FCC1

Devices:

MPC8260, MPC8255

Description:

Two of the TXDATA signals using the 16-bit UTOPIA bus on FCC1 cannot be three-stated. The two signals are PD[5] (txdata[3]) and PD[6] (txdata[4]). In a MPHY system where the XPC8260 is configured as a UTOPIA slave, three-stating of these signals is required when the XPC8260 is not the selected UTOPIA slave or polled device. Additionally, the signals PD[11] (TDMB2: L1RQ) and PD[10] (TDMB2: L1CLKO) may be three-stated sometimes (erroneously) when FCC1 is configured to work in ATM.

Workaround:

None

Fix Plan:

Fixed on Rev. B.1.

CPM48: Error in TDM

Devices:

MPC8260, MPC8255

Description:

Disabling TDMx may interfere with the operation of TDMy in case TDMy uses the SI-RAM blocks directly above those used by TDMx. For example:

- start address of TDMc = 0
- start address of TDMb = 2
- start address of TDMa = 4
- start address of TDMd = 6
- When disabling TDMa, TDMb will be affected.
- When disabling TDMb, TDMc will be affected.
- When disabling TDMd, TDMa will be affected.
- When disabling TDMc, no TDM will be affected.

Workaround:

Instead of disabling a TDM, the user can switch to a shadow RAM, which contains only non-supported slots in its entries.

Fix Plan:

Fixed on Rev. A.0.

CPM49: Error in FEC CAM address recognition

Devices:

MPC8260, MPC8255

Description:

External CAM address recognition in fast Ethernet controller does not function.

Workaround:

Use the microcode patch available from Freescale.

Fix Plan:

Fixed on Rev. B.1.

CPM50: Error in loss of alignment

Devices:

MPC8260, MPC8255

Description:

When configuring the MCC to work in transparent, super channel first sync slot synchronization, loss of alignment may occur. This will occur when the first data (idles) on the Rx data line matches the value of the RCVSYNC parameter.

Workaround:

Write to RCVSYNC a pattern that cannot appear in the Rx data line.

Fix Plan:

Fixed on Rev. A.0.

CPM51: Pointer insertion/extraction error in AAL1 CES

Devices:

MPC8260, MPC8255

Description:

When a pointer value of 93 needs to be inserted/extracted to/from a cell with SN different then 6, the pointer will not be treated correctly.

Workaround:

Use the microcode patch available from Freescale.

Fix Plan:

Fixed on Rev. B.3.

CPM52: Error in ATM internal rate mode

Devices:

MPC8260, MPC8255

Description:

ATM internal rate mode underruns can cause the CPM to behave erratically, and possibly to crash. Symptoms are either loss of a transmitted cell, or possibly a CPM crash requiring a CP reset. This only affects systems working in internal rate mode (that is, FTIRRx[TRM] = 1). Systems working in external rate mode will never experience this bug.

Workaround:

Use the microcode patch available from Freescale.

Fix Plan:

Fixed on Rev. B.3.

CPM53: Inability to run RAM microcode

Devices:

MPC8260, MPC8255

Description:

Large RAM microcode packages will not run on Rev. B.1 or Rev. B.2 device. This includes AAL2 and the enhanced SS7 microcode packages. Small microcode patches (less than 2 Kbytes) for CPM errata fixes are not effected.

Workaround:

None

Fix Plan:

Fixed on Rev. B.3.

CPM54: Error in switching to and from shadow SI RAM

Devices:

MPC8260, MPC8255

Description:

Dynamic switching in SIRAM may not occur properly if the following restrictions are not applied.

Workaround:

In SI RAM, when working with shadow RAM, the last entry (n) and the entry immediately before the last entry (n-1) MUST have at least one common bit in the CNT or BYT fields. For example:

SIRAM Entry	CNT Field	Byte Field
n-1	000	1
n	010	1

The above is okay

SIRAM Entry	CNT Field	Byte Field
n-1	101	0
n	001	0

The above is okay

SIRAM Entry	CNT Field	Byte Field
n-1	100	0
n	001	0

The above is not okay.

Fix Plan:

Fixed on Rev. A.0.

CPM55: Error in ATM_Transmit command

Devices:

MPC8260, MPC8255

Description:

The ATM_Transmit command does not execute correctly when used on APC priority above 4.

Workaround:

None

Fix Plan:

Fixed on Rev. B.3.

CPM56: AAL2 microcode in ROM does not function

Devices:

MPC8260, MPC8255

Description:

Due to a contention on the CPM internal bus, the enhanced AAL2 microcode integrated into ROM on Rev. B.3 is not functional.

Workaround:

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

Fix Plan:

Fixed on Rev. A.0.

CPM57: AAL5 cell corruption

Devices:

MPC8260, MPC8255

Description:

When configured for ATM AAL5 operation, sometimes a small percentage of data packets received are corrupted. The second part of a second ATM cell received is incorrectly DMAed to memory and overwrites the second part of the first ATM cell received in a frame, and the CPM indicates no errors but data in memory is wrong. Note that this data corruption happens only if the AAL5 free buffer pool feature is enabled and the BDs' buffers are on the 60x bus and CTs are on the local bus.

Workaround:

Use the microcode patch available from Freescale.

Fix Plan:

Fixed on Rev. C.2.

CPM62: CPM PLL does not lock reliably for certain multiplication factors

Devices:

MPC8260, MPC8255

Description:

If the CPM multiplication factor is 2.5, 3.5, 5, or 6, the internal PLL will not lock reliably, which will result in erratic behavior. Sometimes (mainly at bus frequencies lower than 50 MHz) it might lock, however, it might loose the lock after worth. If one of the other multiplication factors (2, 3, or 4) is selected, the CPM PLL will lock reliably, even at 66 MHz bus speed.

Workaround:

Do not use the non-reliable multiplication factors. For 166 MHz CPM, use 55 MHz bus frequency and a CPM multiplication factor of 3.

Fix Plan:

Fixed on Rev. C.2.

CPM64: AAL5 RxBD[LNE] error generated if PDU length exceeds 65,512 bytes

Devices:

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

Description:

When the CPM receives an AAL5 PDU between 65,512–65,535 bytes (maximum length). The CPM sets the RxBD[LNE] indicating a receive length error, however, the memory buffer contents for the PDU are correct.

Workaround:

Use the microcode patch available from Freescale, or, alternatively, receive AAL5 PDU less than 65,512 bytes.

Fix Plan:

Fixed on Rev. B.1.

CPM65: SS7 microcode in ROM is not fully functional

Devices:

MPC8260, MPC8255

Description:

The SS7 microcode in ROM on Rev. B.x silicon is not fully functional and should not be used. Refer to the release note available with the latest SS7 RAM microcode release for Rev. B.3 for more details.

Workaround:

Use the enhanced SS7 microcode package (available for Rev. B.3 only). This microcode is not available for Rev. B.1 and Rev. B.2 silicon due to errata CPM 53.

Fix Plan:

Fixed on Rev. B.0.

CPM71: CPM does not snoop MCC buffer descriptors

Devices:

MPC8260, MPC8255

Description:

When the MCC performs a DMA read or write of the buffer descriptor, GBL is not asserted and TC2 is always driven low. Therefore, cache snooping will not be enabled for MCC BDs, and BDs in memory will not match the data cache. Also the bus used for DMA is always 60x, therefore, if the BDs are on the local bus then the DMA consumes bandwidth on both the 60x and local bus.

Workaround:

If GBL and/or TC2 are set in the MCC TSTATE/RSTATE parameters, use the microcode patch available from Freescale. If GBL and TC2 are not set to improve performance move the MCC BDs to the 60x bus.

The microcode patch will fix both the GBL/TC2 and bus performance issue.

Fix Plan:

Fixed on Rev. C.2.

CPM72: MCC global underruns

Devices:

MPC8260, MPC8255

Description:

There is a rare case when MCC transmitter global underrun (GUN) errors may occur during intensive CPM activity even when estimated CPM bandwidth is less than 100%. This is due to the prioritization of the MCC transmitter relative to other CPM resources.

In Rev. C.2 silicon and forward a new feature has been added to the MPC8260, which will allow users to condition the MCC transmitter priority mechanism and remove expected MCC GUNs errors without effecting the performance of other peripherals in the CPM. This feature will be controlled by a new MCC mode bit in the RCCR, which will allow users to continue to use the current CPM priority scheme in their applications if required.

Please refer to the *MPC8260 PowerQUICC II Family Reference Manual* for more details.

Workaround:

None

Fix Plan:

Fixed on Rev. C.2.

CPM73: SIRAM corruption

Devices:

MPC8260, MPC8255

Description:

An access to the SI RAM bank from the 60x bus while the corresponding TDM is active may result in data corruption within the SI RAM.

Workaround:

Associate the SI RAM bank with an inactive TDM before attempting to access it. Once the accesses has been made, the SI RAM bank should be re-assigned to the active TDM.

Fix Plan:

Fixed on Rev. C.2.

CPM75: AAL2 microcode in ROM is not fully functional

Devices:

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

Description:

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

Workaround:

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

Fix Plan:

No fix plan at this time.

CPM76: TC layer transmits and receives data LSB first instead of MSB first

Devices:

MPC8264, MPC8266

Description:

The internal TC layer hardware is connected to the UTOPIA data bus in the reverse order. This means the transmitter transmits the data LSB first and not MSB first as required. The HEC will be generated according to the reversed data. On the receive side data will be received in the right order, the HEC will be checked correctly but the data will be delivered to the UTOPIA Rx data bus in the reverse order.

Workaround:

Use the MPC8260 TC layer in internal loopback mode for development purposes.

Fix Plan:

Fixed on Rev. B.1.

CPM77: IMA microcode in ROM is not fully functional

Devices:

MPC8264, MPC8266

Description:

The IMA microcode integrated into ROM is not fully functional. Refer to latest IMA RAM microcode release for more details.

Workaround:

Use the IMA RAM microcode package available from Freescale.

Fix Plan:

No fix plan at this time.

CPM78: FCC fast Ethernet flow control

Devices:

MPC8260, MPC8255

Description:

When the FCC receives a flow control pause message with MAC parameter = 0xffff, it sets a zero delay instead of maximum delay.

Workaround:

None

Fix Plan:

Fixed on Rev. B.1.

CPM79: MCC CES user template

Devices:

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

Description:

If the transparent MCC Tx CES channel requires the user template (CHAMR[UTM] = 1) then only the first 8 bytes of the user defined pattern are transmitted. Then the transmitter will continue to send bytes 4–7 of the pattern continuously until the counter reaches 0. Any bytes defined in the pattern after byte 7 are never transmitted.

Workaround:

Use a template size of 8 bytes.

Fix Plan:

Fixed on Rev. B.1.

CPM80: Japanese SS7 error interval timer problem

Devices:

MPC8260A, MPC8255A, MPC8250, MPC8264, MPC8265, MPC8266

Description:

The Japanese error interval for the SS7 code is not timed correctly. The error interval timer is started on the reception of every SU, it should be running in the background and expiring on a user programmable time of every N*512 µs (where 0<=N<=65535). This problem only affects the SS7 microcode in Japanese mode.

Workaround:

Use the RAM based SS7 microcode package available from Freescale.

Fix Plan:

Fixed on Rev. B.1.

CPM85: AAL0 only one BSY interrupt generated

Devices:

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

Description:

When using AAL0, only one BSY interrupt will be received regardless of the number of BSY events that are generated.

Workaround:

None

Fix Plan:

Fixed on Rev. B.1.

CPM86: Random PHY number for FCC Rx in single-PHY master mode

Devices:

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

Description:

When FCC receive ATM controller is configured for single PHY master mode (FPSMR[RUMP] = 0, FPSMR[RUMS] = 0) and FPSMR [LAST PHY/PHY ID] is not equal to zero, a random PHY ID might be allocated to the incoming cells instead of the expected zero (for single-PHY).This will result in a loss of cells. This configuration is typical when using the FCC transmit ATM controller in multi-PHY master mode together with the FCC receive ATM controller in single-PHY master mode.

Workaround:

The address lookup mechanism should be created in such a way that for any PHY address input, the output will be as for PHY 0.

Fix Plan:

Fixed on Rev. B.1.

CPM88: MCC transmit GUN when ‘MCC STOP RX’ CPCR command is used

Devices:

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

Description:

An MCC may experience a highly intermittent transmit GUN event indication, related to MCC receive channels that have been stopped through the ‘MCC STOP RX’ host CPCR command. This GUN can happen unrelated to internal CPM loading or other external factors.

Workaround:

Avoid using ‘MCC STOP RX’ command using one of the following mechanisms:

- Simply stop the TDM
- Use shadow RAM and dynamically remove the desired MCC RX channel entry from SIRAM programming (see Chapter 14 of the MPC8260UM).

To use the second mechanism, the following procedure should be utilized, using an extra redundant shadow RAM switch. This is done to provide a full TDM frame's amount of time to ensure receive activity is complete and will avoid the issue:

1. Re-program shadow SIRAM to remove channel to be stopped
2. Switch to shadow SIRAM and wait for that TDM's bit in the SIxSTR register to change to indicate switch complete
3. Copy this new shadow RAM programming back to the main SIRAM bank
4. Switch to active RAM, again wait for switch to complete

Then software can re-initialize or modify the removed channel's RX parameters.

Fix Plan:

Fixed on Rev. B.1.

CPM92: TC Layer when disabled can be selected by FCC2

Devices:

MPC8264, MPC8266

Description:

When the TC layer is enabled for transmit or receive (TCMODEx[TXEN/RXEN] = 1) and transmits or receives an ATM cell (by asserting TxClav or RxClav). The FCC continues to poll the CLAV signal and will respond to a disabled TC layer before it moves to the next TC layer (enabled or disabled). This will have an impact on the ATM CPM performance.

Workaround:

- Use sequential TC Layers (TCMODEx[TXEN], [RXEN] = 1) from TC layer number 1 to TC layer number n and program FPSMR[LastPHY] = n–1.
- For transmit program the disabled TC layer to transmit idle by preparing the APC parameter table, priority-level tables for the corresponding PHY with CPS = 0 and no channel scheduled. For receive, if the user is working with CAM, set the MS bit (MS = 1:not match) of the entries correspondent to the disabled TCs. If the user is working with address compression, program the VP_MASK and the VCOFFSET so the cells coming from the disabled PHYs will get an MS = 1.

Fix Plan:

No fix plan at this time.

CPM93: IDMA microcode in ROM is not fully functional

Devices:

MPC8260, MPC8255

Description:

The IDMA microcode integrated into ROM on Rev. A.1 is not fully functional.

Workaround:

Use the microcode patch available from Freescale.

Fix Plan:

Fixed on Rev. B.1.

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

Devices:

MPC8260A, MPC8255A, MPC8250, MPC8264, MPC8265, MPC8266

Description:

At the beginning of an HDLC frame transmission that is preceded by more than one opening flag, $\overline{\text{RTS}}$ 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 CTS.

Workaround:

- 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 hand shaking activities with the modem will occur for all the proceeding frames.

Fix Plan:

No fix plan at this time.

CPM95: ATM false indication of miss inserted cells

Devices:

MPC8260, MPC8255

Description:

There is a false indication of unassigned bits in the PHY:VPI:VCI, which could cause ATM cells to be treated as miss-inserted cells and, therefore, discarded.

Workaround:

Use the microcode patch available from Freescale.

Fix Plan:

Fixed on Rev. A.0.

CPM96: ATM performance monitoring with AAL1 CES

Devices:

MPC8260A, MPC8255A, MPC8250, MPC8264, MPC8265, MPC8266

Description:

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

Workaround:

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

Fix Plan:

No fix plan at this time.

CPM97: MCC SS7—No SUERM interrupt generated after an ABORT

Devices:

MPC8260A, MPC8255A, MPC8250, MPC8264, MPC8265, MPC8266

Description:

After an ABORT octet count mode is not entered properly when idles are received. Therefore, N_Cnt is not decremented and no SUERM interrupt will be generated. This problem only affects the SS7 microcode in ITU-T/ANSI mode (SS7_OPT[STD] = 0).

Workaround:

Use the latest RAM based SS7 microcode package available from Freescale.

Fix Plan:

No fix plan at this time.

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

Devices:

MPC8260A, MPC8255A, MPC8250, MPC8264, MPC8265, MPC8266

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 could generate partial frames (less than 8 bits) in the next transaction.

Workaround:

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:

MPC8260A, MPC8255A, MPC8250, MPC8264, MPC8265, MPC8266

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 could have a detrimental effect on the network performance.

Workaround:

Use the microcode patch available from Freescale.

Fix Plan:

No fix plan at this time.

CPM100: ABR TCTE address miscalculation

Devices:

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

Description:

When using the AAL5 ABR ROM microcode with external ATM channels, it is possible for the EXT_TCTE_BASE word value (written by the user to DPRAM) to be misread. In this case, calculations performed by the microcode to access the users programmed external TCTE will be incorrect with a high chance of the access resulting in a CPM crash.

Workaround:

Use the microcode patch available from Freescale.

Fix Plan:

Fixed on Rev. B.1.

CPM101: FCC RxClav timing violation (slave)

Devices:

MPC8260A, MPC8255A, MPC8250, MPC8264, MPC8265, MPC8266

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.

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 than 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.

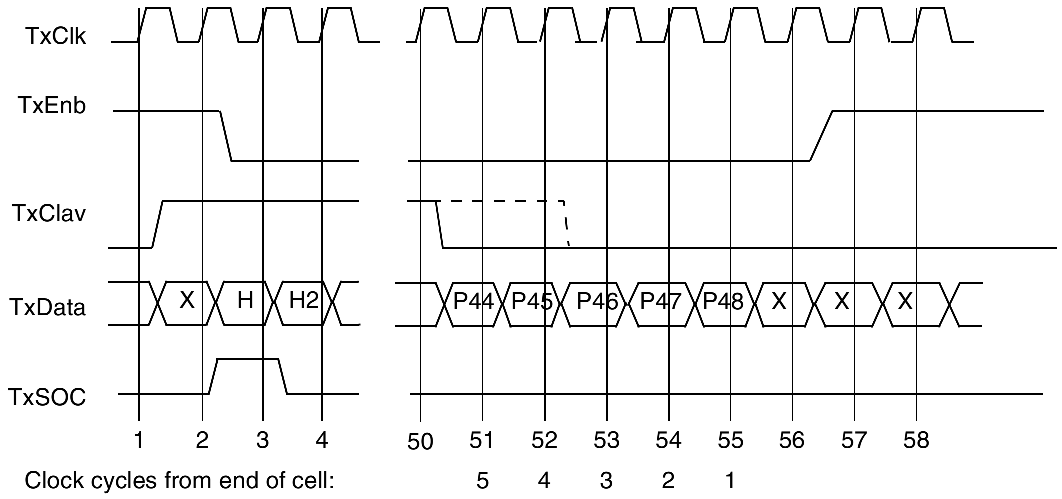

Figure 1. Transmit Timing for Cell-Level Handshake

- Multi-PHY with single Clav polling—Master should ensure that the address corresponding to an MPC826xA slave is not placed on the address bus before 5 cycles before the end of a cell transfer to that slave.

Fix Plan:

No fix plan at this time.

CPM110: FCC1 prioritization

Devices:

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

Description:

FCC1 receiver in Ethernet, HDLC or transparent controller mode is not elevated to emergency status (priority 4 in Table 14-2, “Peripheral Prioritization,” in the *MPC8260 PowerQUICC II Family Reference Manual*), which may lead to FIFO overrun if the system is heavily loaded (FCC1 receiver has the highest priority excluding emergency status of other peripherals).

Workaround:

- When allocating FCCs, assign FCC2 and/or FCC3 for Ethernet, HDLC or transparent before FCC1, or assign FCC1 to the lowest bit rate interface. If FCC1 is allocated for ATM and requires higher CPM utilization then the other FCCs, disable its emergency status.
- If FCC1 is allocated for Ethernet or HDLC and FCC2 for ATM, disable emergency status of FCC2 by setting FPSMR[TPRI].

Fix Plan:

No fix plan at this time.

CPM111: FCC missing reset at overrun

Devices:

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

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 exceed MFLR, it will be discarded without indication on the RxBD. RMON JBRC will be incremented (false jabber).

Workaround:

Use the microcode patch provided by Freescale.

Fix Plan:

No fix plan at this time.

CPM112: FCC missing status

Devices:

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

Description:

TxBD may not be closed for FCC in half duplex 10BaseT Ethernet. There might become a mismatch 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 have been transmitted (assuming a frame per BD).

Workaround:

Use the microcode patch provided by Freescale.

Fix Plan:

No fix plan at this time.

CPM113: Incorrect return value from event register read (SCC, SPI, I2C, and SMC)

Devices:

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

Description:

Under specific conditions, the value returned from an event register (SCC, SPI, I2C, SMC) read might be zero, although some event bits are actually set. If the read operation is done as part of the interrupt handling routine, and no action is taken due to the zero value, the interrupt handler will be invoked again since the pending register bit will still be set. The subsequent read will most probably return the correct value. This should not cause any issue other than some minor performance impact. This issue exists only in the SCC, SMC, I2C. and SPI.

Workaround:

None

Fix Plan:

No fix plan at this time.

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

Devices:

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

Description:

In rare cases certain systems that use $\overline{\text{DREQ}}$ level-sensitive mode may show an additional $\overline{\text{DACK}\textit{x}}$ 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.

Workaround:

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

Fix Plan:

Use the microcode patch provided by Freescale.

CPM115: APC transmits unwanted idle cells

Devices:

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

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 for long enough on the highest priority APC table, then unwanted idle cell could be transmitted on the lower priority APC tables when there are cells available in 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.

Workaround:

Increase the size of lower priority APC scheduling tables so 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:

No fix plan at this time.

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

Devices:

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

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.

Workaround:

Use the microcode patch available from Freescale.

Fix Plan:

No fix plan at this time.

CPM117: False address compression

Devices:

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

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.

CPM118: MCC Rx, aborted HDLC frames

Devices:

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

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.

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:

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

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:

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

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:

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

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.

CPM123: FCC ATM AAL5, underrun and idle cells

Devices:

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

Fix Plan:

No fix plan at this time.

CPM124: AAL1 CES slip-end data corruption

Devices:

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

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.

Workaround

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:

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

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.

Workaround:

- 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:

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

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.

Workaround:

- 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:

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

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.

Workaround:

Use the ucode patch provided by Freescale.

Fix Plan:

No fix plan at this time.

CPM130: Possible corrupted SP in AAL1 STF mode

Devices:

MPC8260, MPC8255, MPC8264, MPC8625, MPC8266

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
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 Manual* 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 specific 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:

MPC8260A, MPC8255A, MPC8250, MPC8264, MPC8265, MPC8266

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:

MPC8260A, MPC8255A, MPC8250, MPC8264, MPC8265, MPC8266

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

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

Devices:

MPC8260, MPC8250, MPC8255, MPC8264, MPC8265, MPC8266

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.

Freescale, the Freescale logo, CodeWarrior, ColdFire, PowerQUICC, QorIQ, StarCore, and Symphony are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. CoreNet, QorIQ Qonverge, QUICC Engine, and VortiQa are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.
© 2003-2012 Freescale Semiconductor, Inc.

Document Number: MPC8260CE
Rev. 10
07/2012