

MPC8323E Chip Errata

This document details all known silicon errata for the MPC8323E and MPC8321E. The following table provides a revision history for this document.

Table 1. Document revision history

Revision	Date	Significant changes
7	10/2014	<ul style="list-style-type: none"> Added QE erratum A-004261 Modified Table 2 and Table 3 for rev 1.3
6	08/2011	<ul style="list-style-type: none"> Added CPU-A022
5	02/2011	<ul style="list-style-type: none"> Changed workaround for General17, QE_HDLC2 and QE_UART6 Added SEC-A001, QE_ENET23, QE_UART-A001, QE_QMC-A001, QE_General-A003 Changed fix plans that were "Will be fixed in Rev 1.1" to "Fixed in Rev 1.1"
4	3/2010	<ul style="list-style-type: none"> Added QE_HDLC2, CPU-A002, QE_ATM-001
3	11/2009	<ul style="list-style-type: none"> Added General17, CPU6, QE_ENET24, QE_ENET25, QE_UART6, QE_UART7, QE_BISYNC3, QE_QMC3, QE_TRANSPARENT1, QE_TRANSPARENT2 Changed name of QE_ENET21 to QE_ENET22
2	8/2008	<ul style="list-style-type: none"> Added General 15, QE_HDLC3, QE_QMC2, and QE_ENET21 Removed DDR20 Changed DDR21 to be only applicable to silicon revision 1.1
1	6/2007	<ul style="list-style-type: none"> Added SEC13, DDR20 and DDR21
0	5/2007	Initial revision

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

Table 2. MPC8323E Family Devices and Silicon Revisions

Device	Silicon Revision	1.0	1.1	1.3
	Mask	1M39D	1M39D	3M39D
MPC8323E		√	√	√
MPC8321E		√	√	√

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

Table 3. Summary of Silicon Errata and Applicable Revision

Errata	Name	Projected Solution	Silicon Rev.		
			1.0	1.1	1.3
General					
General4	I ² C boot sequencer may fail if HRESET happens when SDA is driven low	No plans to fix	Yes	Yes	Yes
General7	DDR2 controller might fail after initialization	No plans to fix	Yes	Yes	Yes
General8	I ² C boot sequencer remains active while it is programmed to be disabled	No plans to fix	Yes	Yes	Yes
General15	Enabling I ² C could cause I ² C bus freeze when other I ² C devices communicate	No plans to fix	No	Yes	Yes
General17	DUART: Break detection triggered multiple times for a single break assertion	No plans to fix	Yes	Yes	Yes
RESET					
RESET3	External Soft reset functionality is not functional	No plans to fix	Yes	Yes	Yes
SEC					
SEC6	AES-CTR mode data size error	Fixed in Rev 1.1	Yes	No	No
SEC7	Single descriptor SRTP error	No plans to fix	Yes	Yes	Yes
SEC13	AESU/DEU initial reset requirement	No plans to fix	Yes	Yes	Yes
SEC-A001	Channel Hang with Zero Length Data	No plans to fix	Yes	Yes	Yes
DDR					
DDR21	MCK/MCK AC differential crosspoint voltage outside JEDEC specifications	No plans to fix	No	Yes	Yes
ARB					
ARB2	Data time out on the coherent system bus	No plans to fix	Yes	Yes	Yes
ARB3	Changing the value of ACR[PIPE_DEP] can cause the arbiter to generate false data time out	Fixed in Rev 1.1	Yes	No	No
CPU					
CPU6	DTLB LRU logic does not function correctly	No plans to fix	Yes	Yes	Yes
CPU-A002	CPU may hang after load from cache-inhibited, unguarded memory	No plans to fix	Yes	Yes	Yes
CPU-A022	The e300 core may hang while using critical interrupt	No plans to fix	Yes	Yes	Yes
QE					
QE_ENET1	RMON errors	No plans to fix	Yes	Yes	Yes
QE_ENET2	Rx MAC might accept dropped frame	No plans to fix	Yes	Yes	Yes
QE_ENET4	TXE event is not set after a Tx error if RMON is disabled	Fixed in Rev 1.1	Yes	No	No
QE_ENET6	The 2/3 (IPGR1) and 1/3 (IPGR2) IPG rule is violated (half duplex mode)	No plans to fix	Yes	Yes	Yes
QE_ENET7	Incorrect pause timer counter behavior	No plans to fix	Yes	Yes	Yes

Table continues on the next page...

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

Errata	Name	Projected Solution	Silicon Rev.		
			1.0	1.1	1.3
QE_ENET8	Transmitter might get stuck after late collision/maximum collisions events	Fixed in Rev 1.1	Yes	No	No
QE_ENET9	Discarded frames with no indication	No plans to fix	Yes	Yes	Yes
QE_ENET12	Frame discarded after busy condition for frames shorter or equal to MRBLR	Fixed in Rev 1.1	Yes	No	No
QE_ENET13	External eight-way hash lookup does not work	Fixed in Rev 1.1	Yes	No	No
QE_ENET16	Rx might get deadlocked in a certain mode	Fixed in Rev 1.1	Yes	No	No
QE_ENET17	Multiple identical keys in the hash lookup table	Fixed in Rev 1.1	Yes	No	No
QE_ENET20	UEC may stop transmitting after late collision	No plans to fix	Yes	Yes	Yes
QE_ENET22	Ethernet receiver may become stuck while in a multithreaded configuration	No plans to fix	Yes	Yes	Yes
QE_ENET23	Half-duplex collision on FCS of short frame may cause transmit lockup	No plans to fix	Yes	Yes	Yes
QE_ENET24	Magic Packet sequence embedded in partial sequence not recognized	No plans to fix	Yes	Yes	Yes
QE_ENET25	Malformed Magic Packet triggers Magic Packet exit	No plans to fix	Yes	Yes	Yes
QE_ATM1	APC PUBR+ mode does not work with CPS>1	Fixed in Rev 1.1	Yes	No	No
QE_ATM2	SAM automatic HEC disablement	Fixed in Rev 1.1	Yes	No	No
QE_ATM3	IMA DSL fails all other links	Fixed in Rev 1.1	Yes	No	No
QE_ATM4	AAL5 AVCF disablement in hierarchical frame based mode	Fixed in Rev 1.1	Yes	No	No
QE_ATM5	AAL2 round-robin mode with TCT[MaxStep] equals 1 for several external TxQDs	Fixed in Rev 1.1	Yes	No	No
QE_ATM6	Unpredictable channels could be transmitted when using ATM GCRA scheduler	Fixed in Rev 1.1	Yes	No	No
QE_ATM7	Overflow event in ATM GCRA scheduler	Fixed in Rev 1.1	Yes	No	No
QE_ATM8	ATM APC auto-activation mode may not work for MSP and AAL2 switch	Fixed in Rev 1.1	Yes	No	No
QE_ATM9	ATM policer is not functional	No plans to fix	Yes	Yes	Yes
QE_ATM-A001	ATM APC flux compensation mode is not functional	No plans to fix	Yes	Yes	Yes
QE_UPC1	Wrong HEC check error indication in Rx slave mode	No plans to fix	Yes	Yes	Yes
QE_UPC2	User-defined cell Rx extra header size limited to 8 bytes	No plans to fix	Yes	Yes	Yes
QE_USB3	Automatic SOF generation is not functional	No plans to fix	Yes	Yes	Yes
QE_HDLC1	UCC does not support HDLC bus mode through the TDM interface	No plans to fix	Yes	Yes	Yes
QE_HDLC2	UCC TxD is always set as open drain in HDLC bus NMSI mode	No plans to fix	Yes	Yes	Yes
QE_HDLC3	"Enter Hunt Mode" host command is not functional for UCC in HDLC mode	No plans to fix	Yes	Yes	Yes
QE_UART1	Simultaneous loopback and echo mode is not functional	No plans to fix	Yes	Yes	Yes
QE_UART2	Overrun indication is not reported on the current BD	No plans to fix	Yes	Yes	Yes
QE_UART3	DRT mode (UPSMR[DRT] = 1) is not functional	No plans to fix	Yes	Yes	Yes

Table continues on the next page...

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

Errata	Name	Projected Solution	Silicon Rev.		
			1.0	1.1	1.3
QE_UART5	Rx FIFO data can be lost when ENTER HUNT MODE command is executed	No plans to fix	Yes	Yes	Yes
QE_UART6	UART/AHDLC receive data corruption on QUICC Engine UCC	No plans to fix	Yes	Yes	Yes
QE_UART7	UART does not set UCCE[AB] correctly for autobaud	No plans to fix	Yes	Yes	Yes
QE_UART-A001	QE UART controller idle status bit may not be reliable	No plans to fix	Yes	Yes	Yes
QE_SI1	Change shadow RAM command might be executed twice	No plans to fix	Yes	Yes	Yes
QE_SI2	TDM to QE clock ratio limitations	No plans to fix	Yes	Yes	Yes
QE_SI3	UCC receiver data corruption might happen when using SWTR or per entry loopback	No plans to fix	Yes	Yes	Yes
QE_BISYNC1	An Underrun condition might occur after transmitting an eight byte frame	No plans to fix	Yes	Yes	Yes
QE_BISYNC2	RxBD in continuous mode is not closed after an Rx parity error	No plans to fix	Yes	Yes	Yes
QE_BISYNC3	BISYNC controller does not enter HUNT MODE	No plans to fix	Yes	Yes	Yes
QE_IW_ATM2ETH1	ATM AAL5 to Ethernet interworking could sometimes enter a deadlock situation in one of the ATM channels	No plans to fix	No	Yes	Yes
QE_QMC2	Data may be shifted if QMC is disabled and enabled during reinitialization or reconfiguration	No plans to fix	Yes	Yes	Yes
QE_QMC3	Missing Interrupt after Stop Rx Host command	No plans to fix	-	Yes	Yes
QE_QMC-A001	QMC GOV event is not reported in the UCC event register	No plans to fix.	Yes	Yes	Yes
QE_Transparent1	The CRC error is not checked in transparent mode if \overline{CD} is de-asserted	No plans to fix	Yes	Yes	Yes
QE_Transparent2	QE may erroneously report CRC error if GUMR[REVD] is set to 1 in transparent mode	No plans to fix	Yes	Yes	Yes
QE_General2	Forbidden access to reserved UCC1/UCC2 memory space	No plans to fix	Yes	Yes	Yes
QE_General3	QE read/write transactions may be corrupted at non-integer QE and the system bus clock ratios	Fixed in Rev 1.1	Yes	No	No
QE_General4	Potential spikes on BRG output clock when using odd divisions	No plans to fix	Yes	Yes	Yes
QE_General-A003	High Tx Virtual FIFO threshold size can cause UCC to halt	No plans to fix	Yes	Yes	Yes
A-004261	QE-ENET: After the reception of a PAUSE frame, an Ethernet frame may be transmitted during the PAUSE window	No plans to fix	Yes	Yes	Yes
PCI					
PCI13	Data corruption occurs when an external PCI initiator issues the "Read Multiple" command	Fixed in Rev 1.1	Yes	No	No
PCI15	Assertion of \overline{STOP} by a target device on the last beat of a PCI memory write transaction can cause a hang	No plans to fix	Yes	Yes	Yes
DMA					
DMA2	Data corruption by DMA when destination address hold (DAHE) bit is used	No plans to fix	Yes	Yes	Yes

General4: I²C boot sequencer may fail if $\overline{\text{HRESET}}$ happens when SDA is driven low

Description: Devices: MPC8323E, MPC8321E

I²C boot sequencer may not succeed in loading data if HRESET occurs when SDA is driven low by an external slave. The I²C slave does not have reset input and it keeps driving the SDA line. As a result, the boot sequencer cannot generate a START command (SCL high and SDA transition from high to low) and the device hangs.

Impact: Failure to load reset configuration words if load from I²C EEPROM is defined. Failure to load boot sequencer commands if boot sequencer is defined. In this case, the chip will hang.

Workaround: None

Fix plan: No plans to fix

General7: DDR2 controller might fail after initialization

Description: Devices: MPC8323E, MPC8321E

After initial power up, by default, the DDR IO pins are in DDR1 mode. To switch to DDR2 mode, the user should clear DDRCDDR[DDR_cfg]. Once the value is written to the register, it takes some time for DDR IO cells to switch their level to DDR2 requirements.

Impact: If enough time has not passed after writing to DDRCDDR[DDR_cfg], the DDR2 controller might fail to initialize correctly.

Workaround: After writing to DDRCDDR[DDR_cfg], allow 50 ms before continuing with the rest of the DDR controller initialization.

Fix plan: No plans to fix

General8: I²C boot sequencer remains active while it is programmed to be disabled

Description: Devices: MPC8323E, MPC8321E

I²C Boot sequencer can be used during following different time intervals:

- During hard reset active period to load the reset configuration words (RCW) from EEPROM. (CFG_RESET_SOURCE = 010)
- After negation of HRESET to configure some registers after reset from EEPROM. (BOOTSEQ = 10)

The I²C boot sequencer can be used in the following modes:

- I²C boot sequencer is used during reset to load the RCW and is disabled after reset. (CFG_RESET_SOURCE = 010, BOOTSEQ = 00)
- I²C boot sequencer is disabled during reset and used after reset to configure some registers. (CFG_RESET_SOURCE != 010, BOOTSEQ = 10)
- I²C boot sequencer is used both during reset and after reset. (CFG_RESET_SOURCE = 010, BOOTSEQ = 10)
- I²C boot sequencer is disabled both during reset and after reset. (CFG_RESET_SOURCE != 010, BOOTSEQ = 00)

The last three modes work as expected, but the first mode does not work as expected. In the first mode, during initial power-up the I²C boot sequencer should be disabled after reset. However, it still remains enabled and configures the registers stored in EEPROM. Note that this problem occurs only during PORESET sequence and not during HRESET. If HRESET is applied again, the boot sequencer works fine and is disabled after reset.

Impact: If the I²C boot sequencer is used after reset, the core should be disabled (in RCW, COREDIS = 1) and then enabled in the last register write from I²C EEPROM. Therefore, if there is valid data stored in EEPROM, both the core and boot sequencer try to access the system bus after negation of reset, which may lead to a system error.

Workaround: The CONT bit in EEPROM should be cleared following the RCWHR data in the EEPROM. This ensures that the boot sequencer is disabled and hence will not initiate any unnecessary activity. Use the local bus to load the boot code (initial register configuration).

Fix plan: No plans to fix

General15: Enabling I²C could cause I²C bus freeze when other I²C devices communicate

Description: Devices: MPC8323E, MPC8321E

When the I²C controller is enabled by software, if the signal SCL is high, the signal SDA is low, and the I²C address matches the data pattern on the SDA bus right after enabling, an ACK is issued on the bus. The ACK is issued because the I²C 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 I²C bus to freeze. However, it happens very rarely due to the need for two conditions to occur at the same time.

Impact: Enabling the I²C controller may cause the I²C bus to freeze while other I²C devices communicate on the bus.

Workaround: Use one of the following workarounds:

- Enable the I²C controller before starting any I²C communications on the bus. This is the preferred solution.
- If the I²C controller is configured as a slave, implement the following steps:
 - a. Software enables the device by setting I2CnCR[MEN] = 1 and starts a timer.
 - b. Delay for 4 I²C bus clocks.
 - c. Check Bus Busy bit (I2CnSR[MBB])

```

if MBB == 0
    jump to Step f; (Good condition. Go to Normal operation)
else
    Disable Device (I2CnCR[MEN] = 0)
  
```

- d. Reconfigure all I²C registers if necessary.
- e. Go back to Step a.
- f. Normal operation.

Fix plan: No plans to fix

General17: DUART: Break detection triggered multiple times for a single break assertion

Description: Devices: MPC8323E, MPC8321E

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

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

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

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

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

1. Read URBRn, which will return a value of zero, and will clear the ULSRn[DR] bit
2. Delay at least 1 character period
3. Read URBRn again

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

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

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

Fix plan: No plans to fix

RESET3: External Soft reset functionality is not functional

Description: Devices: MPC8323E, MPC8321E

External Soft reset is defined such that the DDR controller and the local bus controller will not be reset in the event of an external soft reset. However, the internal bus masters and the internal bus components will be held in reset immediately and as long as external soft reset event is valid and the soft reset sequence is in progress. As a result, if the soft reset event happens in the middle of an outstanding write transaction which is targeted to the DDR main memory or to one of the local bus slaves, the internal bus component will transition to reset state while it needs to supply the data to be written and as a result wrong data could be written to the main memory or to the local bus slave.

External Soft reset is defined such that it has no effect on system configuration registers, including IMMRBAR. As a result, software should behave different in regards to IMMRBAR initialization when recovering from soft reset as opposed to recovering from hard reset (in which the IMMRBAR is guaranteed to be located in its default mapping). Since software cannot tell if it is recovering from soft reset or hard reset without reading the platform's RSR register, and since software has to know the IMMRBAR value in order to read the RSR register, there is no way to recover from soft reset (unless the IMMRBAR is kept in its default value at all time).

Impact: Recovery from a soft reset event is not guaranteed.

Workaround: Do not use soft reset at all. Use hard reset or power on reset. SRESET signal must be used as an output-only signal. Software must not write 1 to the RCR[SWSR] bit.

Fix plan: No plans to fix

SEC6: AES-CTR mode data size error

Description: Devices: MPC8323E, MPC8321E

The SEC 2.1 supports acceleration of AES Counter mode, an underlying algorithm in Secure Realtime Transport Protocol (SRTP), and optionally, IPSec. The SEC is designed to accelerate AES-CTR alone (using descriptor type 0001_0) or in parallel with an HMAC-SHA-1 using a special SRTP descriptor type 0010_1. SRTP uses AES-CTR with HMAC-SHA-1. Although AES in counter mode (AES-CTR) is meant to act as a stream cipher, the AESU considers any input data size that is not an even multiple of 16 bytes to be an error.

Impact: None

Workaround: Use one of the following options:

- The AESU Data Size error can be disabled via the AESU Interrupt Control Register to prevent a nuisance interrupt.
- The input data length (in the descriptor) can be rounded up to the nearest 16B. Set the data-in length (in the descriptor) to include X bytes of data beyond the payload. Set the data-out length to only output the relevant payload (don't need to output the padding). SEC reads from memory are not destructive, so the extra bytes included in the AES-CTR operation can be whatever bytes are contiguously trailing the payload.

Fix plan: Fixed in Rev 1.1

SEC7: Single descriptor SRTP error

Description: Devices: MPC8323E, MPC8321E

The SRTP protocol specifies the use of AES-CTR with HMAC-SHA-1. The SEC 2.1 is designed to accelerate AES-CTR in parallel with HMAC-SHA-1 using a special SRTP descriptor type 0010_1. Single descriptor SRTP does not work for data sizes that are not even multiples of 16 bytes.

Impact: When operating on input data which is $N*16B$, the AESU and MDEU can each read in the portion of the datastream relevant to their respective operations. When the input data is not $N*16B$, the AESU data size workaround described in SEC5 (rounding up to the next 16B) does not work because these excess bytes are 'snooped' by the MDEU and corrupt the HMAC-SHA-1 operation.

Workaround: Because the SRTP protocol does not require the payload to be $N*16B$, most packets will likely be of a data length that hits this erratum. The workaround is to use two descriptors to perform SRTP.

Outbound: The first descriptor is type 0001_0 "common non-snoop" set for an AES-CTR operation using either of the workarounds described in SEC5. The second descriptor is type 0001_0 "common non-snoop" set for an HMAC-SHA-1 of the headers + unpadded encrypted payload + MKI (when present).

Inbound: Same descriptors as outbound, but in reverse order.

To minimize the performance impact of this workaround, these two descriptors should be created simultaneously and launched back-to-back. By configuring the SEC Crypto-channel to perform Done notification on selected descriptors, the first descriptor should be set to not generate a Done interrupt, while the second descriptor (which completes the SRTP operation) should be set to generate the Done interrupt. All the parameters required to build both descriptors are available at the start of the request to the SEC device driver, so there is no reason to wait for the first descriptor to complete before building and launching the second.

Fix plan: No plans to fix

SEC13: AESU/DEU initial reset requirement

Description: Devices: MPC8323E, MPC8321E

The SEC 2.2 is a size optimized version of the Freescale SEC core. One size optimization is the AESU and DEU share a FIFO and other bus interface logic used in descriptor-based operations. There is a bug in the bus interface logic that causes the bus interface to lock out either the AESU or DEU if those blocks are not reset before descriptor based operations begin.

Impact: When the PowerQUICC device comes out of reset, the user needs to initialize the SEC, including configuration of registers in the controller, channel, and EUs. EU initialization operations primarily consist of masking off low level interrupts. This host access to the EUs causes the SEC to lock out either the DEU or AESU, so when the user switches to descriptor based operations, one of the EUs does not work.

This errata does not affect the run-time performance of the SEC 2.2 or reduce functionality. Avoiding the errata requires an extra step at initialization.

Workaround: Following host configuration of each EU, the host needs to perform a soft reset of each EU by writing the value "0x2" to the DEU/AESU Reset Control Register. This soft reset will reset most registers within the DEU/AESU, but will not clear the settings of the EU Interrupt Control Registers. This soft reset has the effect of resetting the bus interface to allow both the DEU and AESU to function properly in descriptor based operations.

Any further host access to the EU registers once descriptor based operations have begun requires an EU soft reset before continuing with descriptor based operations. Note that direct EU access once the EUs are configured is rarely needed. If an EU error causes the host to read EU registers for diagnosis and interrupt clearing, the host should include an EU soft reset as the final step in the interrupt service routine.

Fix plan: No plans to fix

SEC-A001: Channel Hang with Zero Length Data

Description: Devices: MPC8323E, MPC8321E

Many algorithms have a minimum data size or block size on which they must operate. The SEC EUs detect when the input data size is not a legal value and signal this as an error. For most EUs, a zero byte length data input should be considered illegal, however the EUs do not properly notify the crypto-channel using the CHA of a data size error. Instead, the EUs wait forever for a valid data length, leading to an apparent channel hang condition.

Impact: When EUs detect illegal input data size, EUs wait forever for a valid data length, leading to an apparent channel hang condition.

Workaround: Option 1: Ensure that software does not create SEC descriptors to encrypt or decrypt zero length data.

Option 2: Use the SEC crypto-channel watchdog timer to detect hung channels. This is accomplished by enabling each channel's watchdog timer via the Channel Configuration Register CCRx[WGN]. This is a one time configuration. The timer stops when the channel completes a descriptor and restarts at zero each time the channel fetches a new descriptor. The default setting for the watchdog timer is 2^{27} SEC clock cycles. If the timer expires, the channel will generate an interrupt and the Channel Status Register will show the cause as a Watchdog Timeout [WDT]. The channel hang is cleared by resetting the channel via CCR[RST]. The offending EU is reset automatically by the channel.

Fix plan: No plans to fix

DDR21: MCK/ $\overline{\text{MCK}}$ AC differential crosspoint voltage outside JEDEC specifications

Description: Devices: MPC8323E, MPC8321E

The crossover points of the MCK and $\overline{\text{MCK}}$ signals of the DDR controller are not meeting JESD79-2C specifications, which indicates that the crossover points should lie within ± 125 mV range of reference voltage.

Impact: Disagreement with the JESD79-2C standard.

Workaround: To meet the JESD79-2C crosspoint voltage specifications, we recommend implementing the following connections between the DDR controller and DDR2 memory:

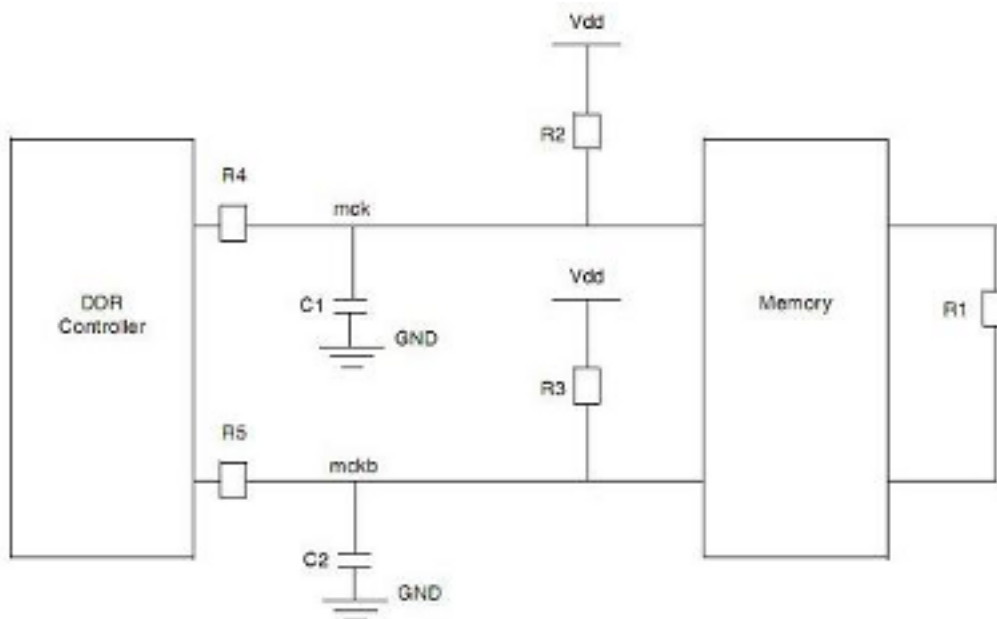


Figure 1. Recommended connections for DDR controller and memory

Effect of different circuit components on MCK and $\overline{\text{MCK}}$ signals:

1. Increasing R2 and R3 shifts signal levels up (used as pull up).
2. Increasing C1 increases rise/fall time of mck signal.
3. Increasing C2 increases rise/fall time of mckb signal.
4. Reducing R1 can help in shifting the signals up.
5. R4 and R5 can be used to make termination proper.

Component Placing:

R4, R5, C1, C2 near to SOC, R1, R2, and R3 have been placed at the end of clock transmission lines.

By considering these suggestions and choosing proper values of components for a particular board layout, one can keep crossover points within range.

We recommend varying termination resistor R1 first. Lowering R1 can help increase the voltage levels up. The only disadvantage is that it may add up more reflections. Therefore, the user must decide accordingly. If it does not give sufficient margin, the user can add the pull-up resistor (R3 and R4) option to further increase the voltage levels.

Exact values of components vary from design to design and some of them may not be required for all designs. We recommend that you make provisions for all these options in your design.

Fix plan: No plans to fix

ARB2: Data time out on the coherent system bus

Description: Devices: MPC8323E, MPC8321E

A data time out may occur on the internal coherent system bus when a master on this bus accesses a very slow device connected to the local bus controller and at the same time, the QUICC Engine block accesses this device (or any other slow device) using the QUICC Engine secondary bus. Because the local bus controller arbitrates between transactions that are originated by the coherent system bus master (CSB port) and transactions that are originated by the QUICC Engine block on the QUICC Engine secondary bus (QE port), the CSB master data phase can potentially last twice the response time of the slowest device connected to the local bus.

Impact: False data time out occurs on the coherent system bus.

Workaround: When data or buffer descriptors needing to be accessed by the QUICC Engine block are stored in a slow external memory device, the QUICC Engine block should be programmed to access this device using the coherent system bus, not the QUICC Engine secondary bus.

Fix plan: No plans to fix

ARB3: Changing the value of ACR[PIPE_DEP] can cause the arbiter to generate false data time out

Description: Devices: MPC8323E, MPC8321E

If the pipeline depth is changed to any value other than zero by writing to ACR[PIPE_DEP], and the next transaction is acknowledged or retried at the same cycle that the actual pipe depth value change is registered in the arbiter logic, the arbiter internal logic will be broken.

Impact: The arbiter generates a false data time out.

Workaround: If the system is configured by the e300 CPU:

- Make sure no other masters are active when pipe depth is configured.
- Add a loop of 100 NOP instructions following the store to ACR. In order to avoid an instruction fetch after the ACR write, make sure the instruction cache is on and that the ACR write and the loop of 100 NOPs are in the same cache line. This will ensure that no instruction is fetched after the ACR write and that, therefore, CSB is idle while the pipe depth is being updated.

For PCI agent mode, use an external PCI host to configure the system. Hold the e300 in core disable mode by setting the CORE_DIS bit in RCWH. Set ACR[PIPE_DEP] to the desired value and then wait for at least 1 μ s before issuing further transactions to the device. At this point the e300 core can be enabled to fetch its boot code by clearing ACR[CORE_DIS].

If possible, use the I²C boot sequencer to configure initial settings of the system, including ACR[PIPE_DEP].

Fix plan: Fixed in Rev 1.1

CPU6: DTLB LRU logic does not function correctly

Description: Devices: MPC8323E, MPC8321E

The DTLB is implemented as 2-way set associative with 32 entries per way. EA[15:19] is used to determine which one of the 32 entries of both ways. When a DTLB miss occurs, normally the CPU provides information (through SRR1 bit 14, DTLB replacement way) to indicate which of the two ways the software (DTLB exception handler or the software table walk routine) should use to bring in the new page. Presumably, the CPU provides the information, through SRR1 bit 14, based on the LRU (least recently used) algorithm. However, because of this bug, this is not the case.

In fact, for any given EA[15:19], when a DTLB miss occurs, SRR1 bit 14 always indicates that way1 should be used or replaced. (Except if it is the very first DTLB miss after hard reset. In this case, SRR1 bit 14 indicates way0 should be used.)

In other words, if the DTLB exception routine follows the SRR1 bit 14's suggestion to do the TLB replacement, it always replaces the one in way1. In addition, whatever has been loaded in way0 is effectively locked and is not replaced.

Impact: Performance degradation due to the reduction of usable DTLB entries.

Workaround: In the software, use one word (32 bits) to keep the record of the DTLB way that is Least Recently Written (LRW). Use this information to overwrite the SRR1 bit 14 (DTLB replacement way) when a Data Translation Miss exception occurs. Basically, the LRU (least recently used) hardware algorithm is changed to an LRW (least recently written) software algorithm.

For the system that has no secondary storage (such as a hard drive), it is highly recommended to set the C (Change) bit during the DTLB load exception to optimize the performance.

For a system that has a secondary storage and the OS does a page swap, the OS can choose whether to set the C bit in the DTLB load exception.

If the C bit is set in the DTLB load exception, it can preempt the subsequent DTLB store exception to the same page. However, since all the pages are marked as changed, during the page swap all pages must be written back to the secondary storage regardless of whether they have really been changed or not.

If the C bit is not set in the DTLB load exception with the LRW algorithm, a subsequent store to the same page as the previous load will use a separate entry. Therefore, one page occupies both ways. This causes inefficiency for the DTLB allocation but may save time during the page swap since the page change status is correctly marked.

Fix plan: No plans to fix

CPU-A002: CPU may hang after load from cache-inhibited, unguarded memory

Description: Devices: MPC8323E, MPC8321E

There is a one-core-clock-cycle window of opportunity for a snoop to collide with the data returned from cache-inhibited memory to a load that has been cancelled. This collision can hang the data cache in a busy state, prohibiting further data cache accesses. The snoop address is immaterial.

The load can be cancelled if it meets the following conditions: it was executing speculatively beyond a conditional branch that resolved unfavorably or was pre-empted by an external interrupt or decremter interrupt that took priority. The load must be from a cache-inhibited, non-guarded memory block. A load from cacheable memory, even if it misses in the cache, does not hang the data cache in a busy state, and if the memory block is marked guarded, the load will not be executed out of order.

An external master must put addresses on the bus with the attribute of memory coherency required (Global) so that the CPU allows snooping of the data cache. External masters have to be programmed to snoop.

Impact: CPU halts or stops executing on a subsequent load or store that finds the data cache busy. This load or store does not complete, and the data cache stays busy until a hardware or software reset ($\overline{\text{HRESET}}$ or $\overline{\text{SRESET}}$). Debug tools will reveal that the program counter is stopped and pointing to the load or store that cannot complete.

Workaround: Use one of the following options:

- Set bit 14 in the HID2 register. This bit disables a minor feature that attempted to take advantage of cache hits under a cancelled cache miss which was still waiting for data. There should be no performance loss from disabling this feature. HID2[14] is not implemented on any other e300 devices. Setting it will have no effect on other devices or any future revisions.
- Mark all data memory blocks from which loads could occur “cacheable” or “guarded.”

Fix plan: No plans to fix

CPU-A022: The e300 core may hang while using critical interrupt

Description: Devices: MPC8323E, MPC8321E

If BOTH critical interrupt AND normal interrupt types are used in a system, the e300 core may hang.

Impact: The processor may stop dispatching instructions until a hardware reset($\overline{\text{HRESET}}$). Debug tools will not be able to read any register correctly except program counter IAR which points to a location in the critical interrupt vector.

Workaround: If both critical interrupt and normal interrupt types are used, then instead of using an **rfi** instruction at the end of every exception handler, replace the **rfi** with the following:

1. Disable critical interrupts by setting MSR[CE] to 0 with a **mtspr** instruction.
2. Copy SRR0 and SRR1 to CSRR0 and CSRR1, respectively.
3. Execute an **rfci** instruction. This enables MSR[CE] and any other bits that the original **rfi** would have set including the MSR[EE].

Sample Code:

```
// Disable MSR[CE]
mfmsr r2
lis r3, 0xffff
ori r3, r3, 0xff7f
and r2, r2, r3
sync
mtmsr r2
isync
// Copy SRR0, SRR1 to CSRR0 and CSRR1
mfspr r2, srr0
mfspr r3, srr1
mtspr csrr0, r2
mtspr csrr1, r3
...restore GPRs
rfci
```

Fix plan: No plans to fix

QE_ENET1: RMON errors

Description: Devices: MPC8323E, MPC8321E

RMON statistics are inaccurate under the following conditions:

- Transmit collision counter status increments after underrun even though no collision has occurred. This bug only shows up when CRC is not appended.
- The MAC (TX module) does not report CRC error on a frame that is transmitted if the previous frame had underrun.
- When BPNB (back pressure no backoff) is enabled the MAC may report a wrong value of collision count (a value that is less than the real collision count).
- If all the following conditions occur, Tx CRC error might not be reported by the Tx MAC:
 - Underrun on last data byte of previous frame
 - No pad or CRC is set on MACCFG2
 - No pad or CRC bit in TxBD is set on current frame (TxBD[TC,PAD/CRC])
 - There is a CRC error on the current frame

Impact: Inaccurate RMON statistics.

Workaround: None

Fix plan: No plans to fix

QE_ENET2: Rx MAC might accept dropped frame

Description: Devices: MPC8323E, MPC8321E

When a dropped frame contains data with 5d pattern, the MAC assumes that it is a new frame and cancels the drop event. The MAC effectively accepts this partial/dropped frame. However, the MAC receive statistic vector (receive previous packet dropped) is asserted to indicate that the received packet is dropped.

Impact: Partial/incomplete frame being received and a false count in the MIBs. However, this partial frame will be rejected by the receiver (due to CRC error).

Workaround: None

Fix plan: No plans to fix

QE_ENET4: TXE event is not set after a Tx error if RMON is disabled

Description: Devices: MPC8323E, MPC8321E

TXE event is not set after a transmit error if RMONEN parameter is cleared.

Impact: Inaccurate value of retransmit count in TxBD. Note that this is a corner case only.

Workaround: Set RMONEN parameter or use a microcode patch.

Fix plan: Fixed in Rev 1.1

QE_ENET6: The 2/3 (IPGR1) and 1/3 (IPGR2) IPG rule is violated (half duplex mode)

Description: Devices: MPC8323E, MPC8321E

If carrier (CRS) is detected during the timing of IPGR1, the MAC does not defer to carrier and continues timing and then transmits.

Impact: This will cause more frequent collisions on the line.

Workaround: Set IPGR2 to some large (optimal) value (e.g. greater than 96).

Fix plan: No plans to fix

QE_ENET7: Incorrect pause timer counter behavior

Description: Devices: MPC8323E, MPC8321E

Receiving a flow control frame with the pause period parameter set to 0x0000 while the transmit side is still decrementing the pause counter due to a previous received flow control frame may cause the transmit counter to be set to 0xFFFF instead of 0x0000.

Impact: The transmitter is stuck/paused for a long time (but will recover eventually) due to the second pause frame (with pause parameter set to zero).

Workaround: None

Fix plan: No plans to fix

QE_ENET8: Transmitter might get stuck after late collision/maximum collisions events

Description: Devices: MPC8323E, MPC8321E

In case of maximum collision/late collision Tx errors on the last byte of the frame payload, the transmitter might get stuck. For maximum collision event, the bug only occurs if the data payload is less than 64 bytes.

Impact: The transmitter will be stuck.

Workaround: Initiate UCC Tx after late collision or max collision error.

Fix plan: Fixed in Rev 1.1

QE_ENET9: Discarded frames with no indication

Description: Devices: MPC8323E, MPC8321E

An overrun status for a received frame indicates that the frame has not been received in full due to temporary QUICC Engine block overloading and should be discarded by the application. In some rare conditions, an entire frame might be discarded without any QUICC Engine block indication.

Impact: The loss of Ethernet frames is detected by higher layer protocols (TCP/IP).

Workaround: None

Fix plan: No plans to fix

QE_ENET12: Frame discarded after busy condition for frames shorter or equal to MRBLR

Description: Devices: MPC8323E, MPC8321E

When the following conditions occur, the last indication for the incoming frame might be lost.

- An incoming frame length is lesser or equal to MRBLR
- VLAN operation for tagged incoming frames and/or IP address alignment are set
- There is no free buffer descriptor, resulting in a busy condition

In this case, several frames might be discarded.

Impact: VLAN operation for tagged incoming frames and/or IP address alignment modes can not be used.

Workaround: Use a microcode patch.

Fix plan: Fixed in Rev 1.1

QE_ENET13: External eight-way hash lookup does not work

Description: Devices: MPC8323E, MPC8321E

Eight-way hash table lookup commands using tables residing in external memory do not work.

Impact: Only four-way hash or internal memory eight-way hash lookup commands can be used.

Workaround: Use four-way hash or internal memory eight-way hash lookup commands or use a microcode patch.

Fix plan: Fixed in Rev 1.1

QE_ENET16: Rx might get deadlocked in a certain mode

Description: Devices: MPC8323E, MPC8321E

When the following conditions are met, Ethernet Rx may become deadlocked.

- Receive short frame mode is disabled.
- Remove VLAN mode is enabled and MRBLR is smaller than frame length.

Impact: Ethernet receiver can become deadlocked.

Workaround: Use one of the following options as a workaround:

- Enable receive short frame mode.
- Disable remove VLAN mode.
- Remove VLAN mode is enable and $mrblr \geq$ frame length.
- Use a microcode patch.

Fix plan: Fixed in Rev 1.1

QE_ENET17: Multiple identical keys in the hash lookup table

Description: Devices: MPC8323E, MPC8321E

Multiple identical keys can be stored in the hash lookup table, but only one TAD (termination action description) can be valid. Incorrect TAD might be chosen.

Impact: Incorrect TAD can be chosen.

Workaround: Use a microcode patch.

Fix plan: Fixed in Rev 1.1

QE_ENET20: UEC may stop transmitting after late collision

Description: Devices: MPC8323E, MPC8321E

When a late collision event coincides with the last byte of a transmitted packet, the UEC's transmitter may halt (TxBD[LC] and UCCE[TXE] will be set). In some cases, if the UEC also reaches the retransmit limit, TxBD[RL] will also be set.

Impact: The UEC might not self recover from a transmit error.

Workaround: The following steps are required to avoid halting the transmitter during the situation described above:

1. Disable automatic recovery from a transmit error by setting UPSMR[7] (reserved bit). The UEC will stop transmission when an error occurs.
2. On every transmit error event (TXE - indicating retransmission limit, late collision, excessive defer, or an underrun), issue a "resume transmit" host command to the UEC to resume transmission. The sequence is:
 - a. Write 0x0000_0001 to CECDR.
 - b. Write the following settings to CECR:
 - CECR[SBC] = UCCx (See the MPC8323E Reference Manual, Rev 1)
 - CECR[FLAG] = 1
 - CECR[MCN] = 0000_0010
 - CECR[Opcode] = 01_1000 (reserved opcode)
 - c. Poll CECR[FLAG] (should be zero) to ensure resume command was executed.

Fix plan: No plans to fix

QE_ENET22: Ethernet receiver may become stuck while in a multithreaded configuration

Description: Devices: MPC8323E, MPC8321E

When there is heavy traffic in a multithread configuration, the Ethernet receiver may become stuck. This condition is influenced by both the VFIFO size and the number of threads that are enabled.

Impact: Ethernet receiver may halt.

Workaround: At least one of these measures must be taken to avoid the situation described above:

- Only one Rx thread is enabled.
- The Rx VFIFO size should be equal to 0.5 Kbytes.
- If the Rx VFIFO size is between 0.5 to 1.1 Kbytes, at least 4 threads must be enabled.
- If the Rx VFIFO size is between 1.1 to 1.6 Kbytes, at least 6 threads must be enabled.
- If the Rx VFIFO size is between 1.6 to 2.2 Kbytes, at least 8 threads must be enabled.
- If the Rx VFIFO size is between 2.2 to 4.5 Kbytes, the user must allocate 512 bytes (must be initialized to zero) in the Multiuser RAM. The base address for this area must be 512 bytes aligned. Immediately after the Ethernet Rx INIT command is ended (and before the UCC receiver is enabled), this address (24 bits) must be written to the Rx GPRAM in offsets 0x09–0x0B and 0x11–0x13. The user must write the value "0x3F" to offsets 0x08 and 0x10 . For example, if the base address for this new structure is 0x12_3400, the Rx GPRAM[0x08–0x0b and 0x10–0x13] is equal to 0x3F12_3400. Note: In this case there is no limitation on the number of threads that must be enabled.
- If none of the above measures apply, use a microcode patch.

Fix plan: No plans to fix

QE_ENET23: Half-duplex collision on FCS of short frame may cause transmit lockup

Description: Devices: MPC8323E, MPC8321E

In half-duplex mode, if a collision occurs in the FCS bytes of a short (less than 64 bytes) frame, the Ethernet MAC may lock up and stop transmitting data or control frames. Only a reset of the controller can restore proper operation once it is locked up.

Impact: A collision on hardware-generated FCS bytes of a short frame in half-duplex mode may cause a transmit lockup.

Workaround: Use one of the following options:

- Set MACCFG2[**PAD/CRC**] = 1, which pads all short Tx frames to 64 bytes.
- Use software-generated CRC (MACCFG2[**PAD/CRC**] = 0, MACCFG2[**CRC_EN**] = 0 and TxBD[**TC**] = 0).

Fix plan: No plans to fix

QE_ENET24: Magic Packet sequence embedded in partial sequence not recognized

Description: Devices: MPC8323E, MPC8321E

The Ethernet MAC should recognize the Magic Packet sequences as follows: Any Ethernet frame containing a valid Ethernet header (Destination and Source Addresses) and valid FCS (CRC-32) and whose payload includes the specific Magic Packet byte sequence at any offset from the start of data payload. The specific byte sequence comprises an unbroken stream of 102 bytes, the first 6 bytes of which are 0xFFs, followed by 16 copies of the MAC's unique IEEE station address in the normal byte order for Ethernet addresses.

If a complete Magic Packet sequence (including 6 bytes of 0xFF) immediately follows a partial Magic Packet sequence, however, the complete sequence will not be recognized and the MAC does not exit Magic Packet mode.

The following are example partial sequences followed by the start of a complete sequence for station address 01_02_03_04_05_06:

- FF_FF_FF_FF_FF_FF_01_02_03_04_05_06_01...
Seventh byte of 0xFF does not match next expected byte of Magic Packet sequence (01). Pattern search restarts looking for 6 bytes of FF at byte 01.
- FF_FF_FF_FF_FF_FF_01_FF_FF_FF_FF_FF_01_02_03_04_05_06_01...
First FF byte following 01 does not match Magic Packet sequence. Pattern search restarts looking for 6 bytes of FF at second byte of FF following 01.

The following is an example partial sequence followed by the start of a complete sequence which is erroneously not recognized for station address 01_02_03_04_FF_06:

- FF_FF_FF_FF_FF_FF_01_02_03_04_FF_FF_FF_FF_FF_01_<complete sequence>
11th byte (0xFF) is seen as the 11 byte of the partial pattern and is not recognized as the start of a complete sequence. Pattern search restarts looking for 6 bytes of 0xFF at 12th byte, but sees only 5.

Impact: The Ethernet controller does not exit Magic Packet mode if the Magic Packet sequence is placed immediately after other frame data that partially matches the Magic Packet sequence.

Workaround: Place 1 byte of data that is not 0xFF and does not match any bytes of Destination Address before the start of the Magic Packet sequence in the frame.

Because the Magic Packet sequence pattern search starts at the third byte after Destination Address, the Magic Packet sequence can be placed at the start of the data payload as long as the second byte of the length/type field follows the above rule.

Fix plan: No plans to fix

QE_ENET25: Malformed Magic Packet triggers Magic Packet exit

Description: Devices: MPC8323E, MPC8321E

The Ethernet MAC should recognize Magic Packet sequences, as follows:

1. Any Ethernet frame containing a valid Ethernet header (Destination and Source Addresses) and valid FCS (CRC-32), and whose payload includes the specific Magic Packet byte sequence at any offset from the start of data payload. The specific byte sequence comprises an unbroken stream of 102 bytes, the first 6 bytes of which are 0xFFs, followed by 16 copies of the MAC's unique IEEE station address in the normal byte order for Ethernet addresses.
2. Once the Ethernet MAC has recognized a valid Destination Address for one frame, it continues searching for valid 102-byte Magic Packet sequences through multiple frames without checking for a valid Destination Address on each frame. The only events that cause the MAC to go back to check for valid Destination Address before checking for a Magic Packet sequence on new frames are:
 - A frame containing a recognized full Magic Packet sequence (with valid or invalid FCS)
 - Software disable of Magic packet mode (MACCFG2[MPE] = 0)
 - Perform soft reset via MACCFG1[Rx_EN] and MACCFG1[Tx_EN].

Impact: The Ethernet controller may exit Magic Packet mode if it receives a frame with Destination Address not matching station address, or invalid unicast or broadcast address in a valid Magic Packet sequence for the device.

Workaround: None

Fix plan: No plans to fix

QE_ATM1: APC PUBR+ mode does not work with CPS>1

Description: Devices: MPC8323E

When working in PUBR+ mode and the CPS parameter in the APC Parameter Table is bigger than one, there might be a situation that the priority levels marked as UBR+ priority will cease transmitting.

Impact: This happens only in systems that are not heavily loaded and APC scheduling yields no valid channel ready for transmission. In this IDLE state, the UBR+ priorities might not be serviced at all.

Workaround: Use CPS = 1 or use a microcode patch.

Fix plan: Fixed in Rev 1.1

QE_ATM2: SAM automatic HEC disablement

Description: Devices: MPC8323E, MPC8321E

In some cases the MTC_MODE[THD] can be unduly set due to a microcode coherency issue.

Impact: Transmitter will stop sending the ATM cell HEC. This corrupts the ATM cell stream causing the corresponding receiving entity to lose cell delineation.

Workaround: Use a microcode patch.

Fix plan: Fixed in Rev 1.1

QE_ATM3: IMA DSL fails all other links

Description: Devices: MPC8323E, MPC8321E

After a DSL interrupt on a link that was added to the group using the LDS process, all other links in the group may suffer a DCBO. This happens if the DCBX reaches and is equal to the DCBRP for the LDS added link before this link is removed using the link removal procedure.

Impact: Entire group will be torn down for a one link failure.

Workaround: Use a microcode patch.

Fix plan: Fixed in Rev 1.1

QE_ATM4: AAL5 AVCF disablement in hierarchical frame based mode

Description: Devices: MPC8323E, MPC8321E

In ATM AAL5 hierarchical frame based scheduling mode, in a situation which there are no buffers to transmit in WFQ channel under V-TCT. This channel will be taken out from the WFQ without checking AVCF bit in WFQ TCT.

Impact: In each buffer not ready event, the WFQ channel will stop transmit until the host will issue a start transmit host command.

Workaround: Use a microcode patch.

Fix plan: Fixed in Rev 1.1

QE_ATM5: AAL2 round-robin mode with TCT[MaxStep] equals 1 for several external TxQDs

Description: Devices: MPC8323E, MPC8321E

When working in round-robin AAL2 Tx mode, if the TCT[MaxStep] parameter equals one and there is nothing to transmit from the external TxQD, the QUICC Engine block becomes stuck on this TxQD forever.

Impact: Deadlock can occur in this condition.

Workaround: Set TCT[MaxStep] according to the actual number of TxQD under the TCT (or to a value bigger than one), even when working in round-robin mode, or use a microcode patch.

Fix plan: Fixed in Rev 1.1

QE_ATM6: Unpredictable channels could be transmitted when using ATM GCRA scheduler

Description: Devices: MPC8323E, MPC8321E

In some cases the GCRA scheduler will schedule unpredictable channels for transmission.

Impact: Since this condition might occur in GCRA ATM scheduler mode, then this mode is not functional.

Workaround: Use a microcode patch.

Fix plan: Fixed in Rev 1.1

QE_ATM7: Overflow event in ATM GCRA scheduler

Description: Devices: MPC8323E, MPC8321E

When an overflow occurs, the GCRA does not function if the following conditions are met:

- Working in ATM GCRA scheduler
- Using non-expanded mode (up to 32 channels)
- Using multi-priority levels

Impact: GCRA mode is not functional in this condition.

Workaround: Use expanded mode GCRA even if 32 channels or fewer are used, or use a microcode patch.

Fix plan: Fixed in Rev 1.1

QE_ATM8: ATM APC auto-activation mode may not work for MSP and AAL2 switch

Description: Devices: MPC8323E, MPC8321E

When working in switch mode using traditional APC, the Rx switch may want to activate the channel into the APC automatically, a mechanism which is called channel auto-activation. It may be that the channel that should be inserted into the APC slot will be overwritten by the APC rescheduling process, and therefore this channel will never be transmitted again.

Impact: ATM APC auto-activation for switched modes, such as AAL2 switch or MSP, may not work.

Workaround: Use a microcode patch.

Fix plan: Fixed in Rev 1.1

QE_ATM9: ATM policer is not functional

Description: Devices: MPC8323E, MPC8321E

ATM policer is not functional.

Impact: Memory corruption, false PM table updates, and wrong memory accesses by the QUICC Engine block can occur and cause AAL0 traffic to operate incorrectly.

Workaround: Use a microcode patch.

Fix plan: No plans to fix

QE_ATM-A001: ATM APC flux compensation mode is not functional

Description: Devices: MPC8323E, MPC8321E

APC flux compensation mode is not functional.

Impact: When APC flux compensation mode is used, half of the time frames are not transmitted.

Workaround: Use regular APC or GCRA schedulers, and disable AFC mode in ATM APC.

Fix plan: No plans to fix

QE_UPC1: Wrong HEC check error indication in Rx slave mode

Description: Devices: MPC8323E

HEC check can be wrong when using UTOPIA Rx slave mode with HEC check enabled (UPDC[HECC]=1) and if the master pauses the transfer (negates TX_EN_B) on the HEC cycle.

Impact: Applications that use UTOPIA Rx slave interface and enable the HEC check should not pause the cell transmission.

Workaround: None

Fix plan: No plans to fix

QE_UPC2: User-defined cell Rx extra header size limited to 8 bytes

Description: Devices: MPC8323E

User-defined cells in Rx slave mode with extra header size of 9–12 bytes may cause data corruption.

Impact: The maximum extra header size UPCDx[REHS] is limited to eight bytes.

Workaround: None

Fix plan: No plans to fix

QE_USB3: Automatic SOF generation is not functional

Description: Devices: MPC8323E, MPC8321E

USB automatic SOF generation is not functional.

Impact: USB controller will not generate SOF automatically.

Workaround: Use a microcode patch.

Fix plan: No plans to fix

QE_HDLC1: UCC does not support HDLC bus mode through the TDM interface

Description: Devices: MPC8323E, MPC8321E

The connectivity of $\overline{\text{CTS}}$ from the NMSI input to the UCC in TSA mode is not enabled.

Impact: HDLC bus mode is not supported through the TDM interface.

Workaround: Program the UCC to NMSI mode and use external logic to generate the gated clock for the UCC at the assigned time slot, see Figure 1 below.

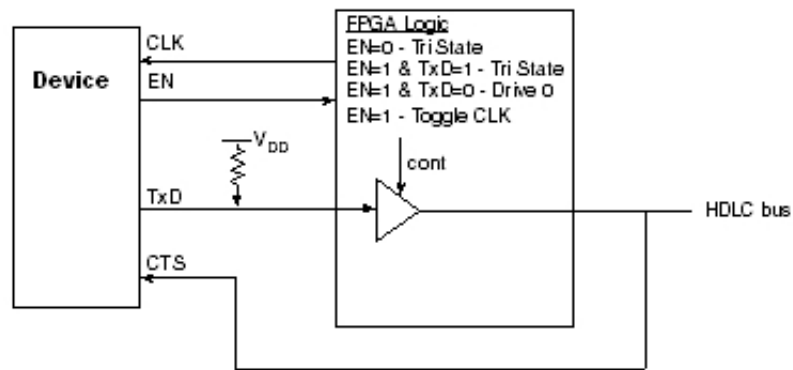


Figure 2. External Logic To Generate Gated Clock for the UCC

Fix plan: No plans to fix

QE_HDLC2: UCC TxD is always set as open drain in HDLC bus NMSI mode

Description: Devices: MPC8323E, MPC8321E

TxD output is always set as open drain in HDLC bus NMSI mode, independent of the parallel I/O configuration, which is the correct setting when there are two or more stations on the HDLC bus. However, it is not required for TxD to be set as open drain when there is only a single station connected to the bus through an external line driver (which implements the open drain capability).

Impact: HDLC bus with external line driver: TxD has to be pulled up even if TxD is not set as open-drain.

Workaround: Place pull-up resistor on TxD signal.

Fix plan: No plans to fix

QE_HDLC3: "Enter Hunt Mode" host command is not functional for UCC in HDLC mode

Description: Devices: MPC8323E, MPC8321E

"Enter Hunt Mode" host command may stall the UCC receiver in the following Fast Protocol modes:

- HDLC
- Transparent

Impact: UCC receiver may halt.

Workaround: Use a microcode patch.

Fix plan: No plans to fix

QE_UART1: Simultaneous loopback and echo mode is not functional

Description: Devices: MPC8323E, MPC8321E

Simultaneous loopback and echo diagnostic mode (GUMR_L[DIAG] = 11) is not functional.

Impact: Issue only in diagnostic mode.

Workaround: Test echo mode and loopback mode separately.

Fix plan: No plans to fix

QE_UART2: Overrun indication is not reported on the current BD

Description: Devices: MPC8323E, MPC8321E

The overrun indication is not reported on the current Rx buffer descriptor but on first buffer descriptor related to the received data after the overrun condition ended.

Impact: The user will not have any knowledge about the overrun event until the event ends and data is received. Not backward compatible with the PowerQUICC II family.

Workaround: None

Fix plan: No plans to fix

QE_UART3: DRT mode (UPSMR[DRT] = 1) is not functional

Description: Devices: MPC8323E, MPC8321E

Disable receive while in transmitting mode (UPSMR[DRT] = 1) is not functional.

Impact: DRT mode cannot be used.

Workaround: None

Fix plan: No plans to fix

QE_UART5: Rx FIFO data can be lost when ENTER HUNT MODE command is executed

Description: Devices: MPC8323E, MPC8321E

RX ENTER HUNT MODE host command is not functional.

Impact: Data may be lost in the UART receive FIFO.

Workaround: Use a microcode patch.

Fix plan: No plans to fix

QE_UART6: UART/AHDLC receive data corruption on QUICC Engine UCC

Description: Devices: MPC8323E, MPC8321E

The QE UCC hardware may experience synchronization problems when it is configured for UART or Asynchronous HDLC in both normal operation and multidrop operation. Framing errors and/or CRC errors may be reported in the receive buffer descriptors (BDs) when synchronization issues occur. It is also possible for received data to be corrupted.

Impact: Data might be received incorrectly leading to data integrity errors, e.g. framing errors (for UART) or CRC errors (for AHDLC).

Workaround: Use one of the following options:

- Use DUART controller (for UART), if possible
- Use the microcode patch found in QERAMPKG. The microcode patch disables UART/AHDLC mode, and instead collects all the data using Transparent mode. By over-sampling, the microcode processes the data based on either the UART or AHDLC protocol that has been configured. There are some limits due to this software solution. Please see QERAM Microcode Release Note for details.

If this workaround is used, all other QE UART errata are not applicable.

- Sample the receive data signal before it reaches the device and connect the sampled version of the receive data signal to the device instead of the original signal. The clock that should be used for this sampling of receive data should be the one that is used by the UCC's UART hardware (either the external clock or BRGO clock that is driven to an external pin and used for sampling receive data). The negative edge of the clock should be used to sample the receive data.

Fix plan: No plans to fix

QE_UART7: UART does not set UCCE[AB] correctly for autobaud

Description: Devices: MPC8323E, MPC8321E

The QE UCC hardware does not set UCCE[AB] correctly when it is configured to be UART for the autobaud operation. The autobaud control function detects the baud rate on the corresponding RXDx input. The control function then writes the detected divide ratio to BRGCx[CD] and BRGCx[DIV16]. UCCE[AB] should be set at the same time so that the software (interrupt server routine (ISR)) can adjust the ratio to be an exact value. However, UCCE[AB] is not set due to this erratum.

Impact: The UCCE[AB] bit cannot be used for checking the autobaud interrupt.

Workaround: The following steps are required to ensure that the UCC operates correctly during the situation described above:

1. Set UCCM[Rx] and UCCM[AB].
2. Set MRBLR to 1 (BD contains 1 char) on system initialization.
 - Receive data is driven (AT commands) and triggers autobaud mechanism that sets BRGCx[CD] and BRGCx[DIV16], receive data is received to data buffer (one char).
 - UCCE[Rx] is set and an interrupt is issued on the received data character. This bit can be used instead of UCCE[AB], because it does not work due to this erratum.
3. Configure as follows for the interrupt service routine (ISR):
 - a. Use the first receive(Rx) ISR to process the autobaud (AB) interrupt request. This can be done by setting BRGCx[CD] and BRGCx[DIV16], if necessary.
 - b. Re-configure MRBLR to the required value, so that later, the UCCE[Rx] bit can be used for a regular receive data interrupt.

Fix plan: No plans to fix

QE_UART-A001: QE UART controller idle status bit may not be reliable

Description: Devices: MPC8323E, MPC8321E

The idle status bit in the UCC UART status register, UCCS[ID], may not always reflect the actual status of the line.

Impact: The idle status bit cannot be used by software to monitor the line state.

Workaround: None. Software should not use the UART Idle status bit.

Fix plan: No plans to fix

QE_SI1: Change shadow RAM command might be executed twice

Description: Devices: MPC8323E, MPC8321E

Before issuing a change shadow RAM command the Host must poll the SI Command Register (CMDR_L or CMDR_H) and check that is all cleared otherwise the previous command might be executed twice.

Impact: If workaround is not implemented, routing table might be corrupted.

Workaround: Check that CMDR_H is cleared before issuing a new command to TDMA through TDMD.
Check that CMDR_L is cleared before issuing a new command to TDME through TDMH.

Fix plan: No plans to fix

QE_SI2: TDM to QE clock ratio limitations

Description: Devices: MPC8323E, MPC8321E

The clock ratio between the TDM clock and the QE clock is limited to 1:24 (the TDM clock can not be faster than QE clock divided by 24), except for the following clock to sync relationship:

- The limitation is 1:32 when $S1xMR[CE] = S1xMR[FE]$, $S1xMR[TFSD] = 00$, and the sync has the timing shown in Figure 1 and Figure 2. In these configurations, $S1xMR[8]$ should be set and in addition to this limitation, the maximum delay between L1CLK and L1SYNC t_d should be less than 8 QE clocks.

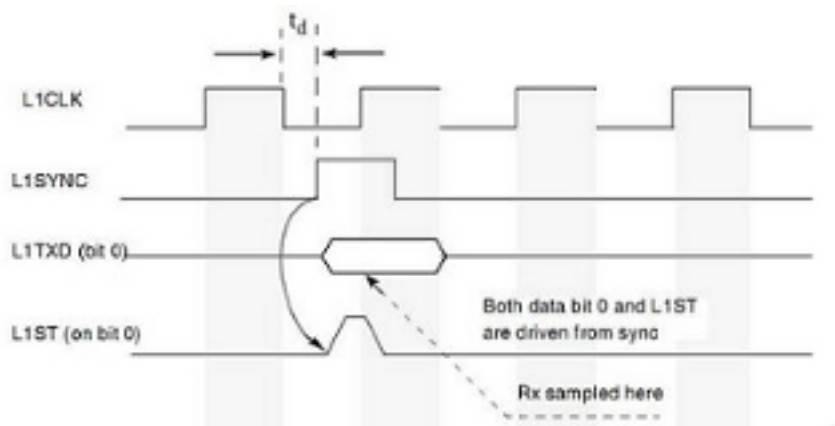


Figure 3. Falling Edge (FE) Effect When CE=1, FE=1 and xFSD=00.Sync starts before rising edge

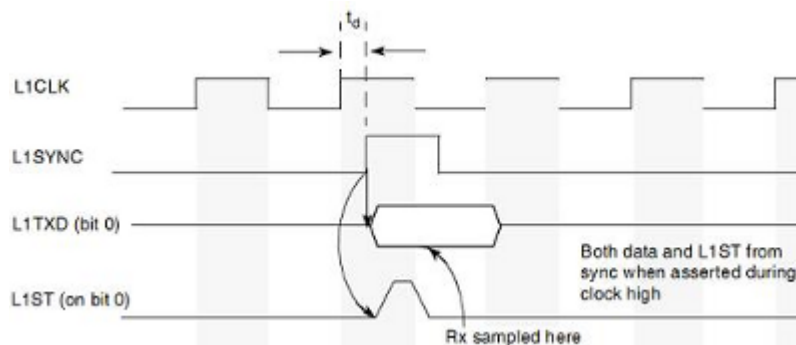


Figure 4. Falling Edge (FE) effect When CE=0,FE=0 and xFSD=00.SYNC starts before falling edge

Impact: Maximum TDM data rate cannot be achieved.

Workaround: Setting $S1xMR[8]$ as described above is a temporary work around. In future revisions, this mode bit might be used for a different mode setting and it will be required to clear it unless it is used for a different mode or purpose.

Fix plan: No plans to fix

QE_SI3: UCC receiver data corruption might happen when using SWTR or per entry loopback

Description: Devices: MPC8323E, MPC8321E

In the receive routing RAM, for a UCC assigned timeslot entry, when SWTR is enabled for one of the entries which is followed or preceded by an entry in which the SWTR is disabled, the data corruption might occur. This applies also for a UCC assigned Rx entry with per-entry loop back (refer to SIENS register).

Impact: Data corruption on the receive side when using SWTR or per entry loopback.

Workaround:

- Add an un-assigned entry (with CSEL = 0000) before and after a UCC entry with SWTR/ Loop mode enabled.
- Use MCC for the SWTR channel. (Refer to the SIENS register, extended diagnostic mode).
- Use global loop back mode instead of per-entry loop back for UCC entry.

Fix plan: No plans to fix

QE_BISYNC1: An Underrun condition might occur after transmitting an eight byte frame

Description: Devices: MPC8323E, MPC8321E

After transmitting a frame with eight data bytes (TxBD with last and length of 0x8) there is an increased probability for a transmit underrun for the following frame.

Impact: Underrun condition might occur

Workaround: Avoid using data length of eight bytes.

Fix plan: No plans to fix

QE_BISYNC2: RxBD in continuous mode is not closed after an Rx parity error

Description: Devices: MPC8323E, MPC8321E

RxBD in continuous mode (CM is set in the RxBD) is not closed after an Rx parity error. After receiving a parity error, the controller sets the PR status bit in the RxBD, but does not clear the Empty control bit as expected.

Impact: The receiver does not work as expected in continuous mode when an Rx parity error occurs.

Workaround: When using CM in the receiver, the software should poll PR status during the routine handling of RxBD.

Fix plan: No plans to fix

QE_BISYNC3: BISYNC controller does not enter HUNT MODE

Description: Devices: MPC8323E, MPC8321E

The BISYNC controller should revert to hunt mode (which means the current opened Receive BD is closed, no additional bytes will be written to the current data buffer, and the BISYNC controller will search for the SYN1/SYN2 sequence, after which reception continues using the next BD) when it receives an ENTER HUNT MODE command, upon an error condition, or after reception of a pre-defined control character (RCCM entry H bit is set). Instead, if the BISYNC controller was in a state of receiving a message, it continues the reception of characters.

Impact: The BISYNC controller keeps transferring data to the receive buffers and does not enter hunt mode.

Workaround: Use a microcode patch.

Fix plan: No plans to fix

QE_IW_ATM2ETH1: ATM AAL5 to Ethernet interworking could sometimes enter a deadlock situation in one of the ATM channels

Description: Devices: MPC8323E, MPC8321E

The ATM AAL5 to Ethernet interworking could sometimes enter a deadlock situation in one of the ATM channels.

Impact: When the problem occurs no traffic from the specific ATM channel is ever Interworked to the destination Ethernet port. The BD ring of the ATM channel presents a situation where a single frame has not finished the IW process causing a busy condition and continuous drop of frames due to this condition on this BD ring.

Workaround: Use a microcode patch.

Fix plan: No plans to fix

QE_QMC2: Data may be shifted if QMC is disabled and enabled during reinitialization or reconfiguration

Description: Devices: MPC8323E, MPC8321E

If the QMC transmitter is disabled and enabled through GUMR_L[ENT] during a reconfiguration or reinitialization procedure, data may be transmitted in the wrong timeslot.

Impact: Data may be shifted up to three timeslots if the QMC is disabled and enabled during a reconfiguration or reinitialization procedure.

Workaround: Before enabling the QMC transmit controller (before GUMR_L[ENT] is set), always initialize the UCC transmit startup ROM address to 0x80 with the PushSched host command and the UCC receive startup ROM address to 0x82 with the PushSched host command.

See the “Serial Number (SNUM)” and “QUICC Engine Commands” section of the device reference manual.

Fix plan: No plans to fix

QE_QMC3: Missing Interrupt after Stop Rx Host command

Description: Devices: MPC8323E, MPC8321E

When a Stop Rx Host Command is issued, channel specific RXB and RXF interrupts may not be generated for data written to buffer prior to the command. The receive BD status indications are not affected.

Impact: RXB and RXF interrupts may be missing after Stop Rx Host Command.

Workaround: After issuing Stop Rx Host Command to a specific channel, RxBD[E] should be polled for the indication of data that was written to Rx buffer.

Fix plan: No plans to fix

QE_QMC-A001: QMC GOV event is not reported in the UCC event register

Description: Devices: MPC8323E, MPC8321E

GOV events are not reported in the UCC event register.

Impact: UCCE[GOV] is not set and an interrupt is not issued on QMC GOV event.

Workaround: Use a microcode patch.

Fix plan: No plans to fix.

QE_Transparent1: The CRC error is not checked in transparent mode if \overline{CD} is de-asserted

Description: Devices: MPC8323E, MPC8321E

When operating the UCC transparent protocol in the following modes, the Carrier Detect (\overline{CD}) lost bit is always set when the \overline{CD} signal is de-asserted at the end of the packet:

- \overline{CD} envelope mode: GUMR[CDP] = 0
- External sync mode: GUMR[SYNL] = 00

However, when the QUICC Engine detects and reports the \overline{CD} loss, it no longer checks for CRC error.

Impact: The QUICC Engine might receive a bad packet with wrong CRC, but RxB[CR] will not be set by the QUICC Engine to indicate the CRC error.

Workaround: Use a microcode patch.

Fix plan: No plans to fix

QE_Transparent2: QE may erroneously report CRC error if GUMR[REVD] is set to 1 in transparent mode

Description: Devices: MPC8323E, MPC8321E

In UCC transparent mode, if GUMR[REVD] is set to 1, the receiver may calculate a wrong CRC for a packet.

Impact: Due to wrong CRC calculation, the QUICC Engine block may report a CRC error for a correct packet.

Workaround: Use a microcode patch.

Fix plan: No plans to fix

QE_General2: Forbidden access to reserved UCC1/UCC2 memory space

Description: Devices: MPC8323E, MPC8321E

Write access to the following address space in UCC1/UCC2 memory space can cause erroneous behavior of UPC1/UPC2, respectively, as it will override the register settings of these peripherals. Read from these addresses does not cause problem. The restricted space (offset from UCC1/UCC2 base address) is:

- 0x1C - 0x1F
- 0x80 - 0x8F

Impact: Users need to make sure software does not write to these addresses.

Workaround: Do not write to the described memory space.

Fix plan: No plans to fix

QE_General3: QE read/write transactions may be corrupted at non-integer QE and the system bus clock ratios

Description: Devices: MPC8323E, MPC8321E

QE read/write transactions may be corrupted when the QE and the coherent system bus clock ratio is a non-integer. Integer clock ratios can be used such as 266:266 Mhz or 400:200 Mhz (QE/system bus).

Impact: QE and system bus clock ratio is limited to integer values.

Workaround: Use integer clock ratio.

Fix plan: Fixed in Rev 1.1

QE_General4: Potential spikes on BRG output clock when using odd divisions

Description: Devices: MPC8323E, MPC8321E

When the BRG is dividing the clock using an odd division factor greater than 3, there are potentially spikes on the BRG output clock.

- Clock Division Factor = $BRGCx[CD] + 1$

Impact: BRG division factor cannot be odd (greater than 3).

Workaround: Use one of the following options:

- Use an even division factor, i.e. $(BRGCx[CD] + 1)$ is divisible by 2
- Use an odd division factor only in DIV16 mode

Fix plan: No plans to fix

QE_General-A003: High Tx Virtual FIFO threshold size can cause UCC to halt

Description: Devices: MPC8323E, MPC8321E

Due to the structure of Tx Virtual FIFO, it is possible for the Tx Virtual FIFO to contain part of a frame when there is no room for the remainder of the frame (i.e. when the frame size is on the scale of magnitude of the Tx Virtual FIFO size).

If the Tx Virtual FIFO contains a partial frame, the transmission of the frame may start even if fewer than UTFTT (UCC Tx Virtual FIFO Threshold) bytes of data reside in the Tx Virtual FIFO. The only reason the frame will not start transmission is if it meets a specific sequence in which the UCC transmit-start condition is not met for the frame. This is rare but possible.

The erroneous behavior is a result of incorrect recovery after the pausing data-retrieve phase from the BD buffer to the Tx Virtual FIFO in a certain internal stage.

Reaching the UTFTT watermark in the Tx Virtual FIFO for each transmit frame prevents this sequence. UTFTT has an upper limit value, which is less than the UTFS (UCC Tx Virtual FIFO Size). For each UTFS value there is a maximum value of UTFTT that corresponds to it. Setting UTFTT higher than the suggested value can result in a UCC halt, and Ethernet transmission stops.

NOTE

1. As defined in the reference manual, UTFTT refers to the payload of a single Ethernet frame.
2. Transmission may start before surpassing the threshold if there is no space for additional data in Tx Virtual FIFO. This is the correct behavior.

Impact: The UCC may stop transmitting when a large frame is partially stored in the Tx Virtual FIFO before the transmission of the frame starts.

Note that the Tx BD ring buffer configuration (data allocation to BD buffers) may adversely affect Tx Virtual FIFO utilization, thus preventing the Tx Virtual FIFO from reaching the threshold for transmission.

Workaround: Use one of the following options:

- To recover from a possible failure, the application code should configure UTFTT to 0x40 and then set it back to the original value. This sequence triggers transmission from the point it stopped. Detection of a UCC halt can be done by monitoring Tx BD ring vacancy. A full Tx BD ring may indicate the UCC has halted.
- To prevent the failure, configure UTFS and UTFTT so that it is possible to store UTFTT data (payload) bytes in Tx VFIFO.
 - a. For the usage of a single buffer per frame, here are the maximum UTFTT values that can be used:
 - Ethernet Controller: Set $UTFTT < [(0.9375 \times UTFS) - 128]$
 - HDLC Controller: Set $UTFTT < [(0.7500 \times UTFS) - 32]$
 - For example, for an Ethernet controller with $UTFS = 1024$, set $UTFTT < 832$.
 - b. For the usage of a multiple buffer per frame, here are the maximum UTFTT values that can be used: (M is the smallest buffer size used)
 - Ethernet Controller: Set $UTFTT < [(UTFS \times (M - 8) \div M) - 128]$
 - HDLC Controller: Set $UTFTT < [(UTFS \times (M - 8) \div M) - 32]$
 - For example, for an Ethernet Controller with $UTFS = 1024$ and $M = 64$, set $UTFTT < 768$.

Fix plan: No plans to fix

A-004261: QE-ENET: After the reception of a PAUSE frame, an Ethernet frame may be transmitted during the PAUSE window

Affects: QE-ENET

Description: The Ethernet MAC might transmit a frame while being in a PAUSE state and this frame could consume the intended PAUSE duration. This is a violation of the specification which states in the QEIWRM "Receiving a Flow control Frame" section that "transmission stops for the time specified in the control frame."

The indications of UCCE[CBPR] and UCCS[BPR] are incorrect as well; the PAUSE indication should have been set only when the UCC Ethernet controller (UEC) transmitter is in PAUSE state and is not transmitting on the line, but the actual behavior is that the PAUSE indication is set while the UEC is transmitting on the line.

Impact: The UCC Ethernet controller may (occasionally) violate the pause duration.

Workaround: None

Fix plan: No plans to fix

PCI13: Data corruption occurs when an external PCI initiator issues the “Read Multiple” command

Description: Devices: MPC8323E, MPC8321E

When the 83xx PCI controller is the target of a PCI Memory Multiple-Read (MRdMult) command, the IO sequencer may enter an illegal state, causing data corruption on the PCI bus and other erroneous behaviors.

Impact: When the IOS buffer control mechanism becomes out of sync, the following behaviors may occur:

- Data corruption on the PCI bus
- System-level data corruption
- Unexpected PCI transactions
- Missed PCI transactions

Workaround: Use one of the following options:

- Turn off the MRdMult command on any potential external PCI initiator.
- Limit the DMA transaction size to a cache line 32 Bytes or less when using the MRdMult command on the PCI initiator side.
- Do not mark any inbound windows to be prefetchable. This will likely impact performance due to significant reduction in the bandwidth of PCI target reads.

Fix plan: Fixed in Rev 1.1

PCI15: Assertion of \overline{STOP} by a target device on the last beat of a PCI memory write transaction can cause a hang

Description: Devices: MPC8323E, MPC8321E

As a master, the PCI IP block can combine a memory write to the last PCI double word (4 bytes) of a cacheline with a 4 byte memory write to the first PCI double word of the subsequent cacheline.

This only occurs if the second memory write arrives to the PCI IP block before the deassertion of \overline{FRAME} for the first write transaction. If the writes are combined, the PCI IP block masters a single memory-write transaction on the PCI bus. If for this transaction, the PCI target asserts \overline{STOP} during the last data beat of the transaction (\overline{FRAME} is deasserted, but \overline{TRDY} and \overline{IRDY} are asserted), the transaction completes correctly. A subsequent write transaction other than an 8-byte write transaction causes a hang on the bus. Two different hang conditions can occur:

- If the target disconnects with data on the first beat of this last write transaction, the PCI IP block deasserts \overline{IRDY} on the same cycle as it deasserts \overline{FRAME} (PCI protocol violation), and no more transactions will be mastered by the PCI IP block.
- If the target does not disconnect with data on the first beat of this last write transaction, \overline{IRDY} will be deasserted after the first beat is transferred and will not be asserted anymore after that, causing a hang.

Impact: This affects 32-bit PCI target devices that blindly assert \overline{STOP} on memory-write transactions, without detecting that the data beat being transferred is the last data beat of the transaction. It can cause a hang.

If the PCI transaction is a one data beat transaction and the target asserts \overline{STOP} during the transfer of that beat, there is no impact.

Workaround: Hardware workaround:

Ensure that the PCI target device does not assert \overline{STOP} during the last beat of a PCI memory write transaction that is greater than one data beat and crosses a cacheline boundary. It could assert \overline{STOP} during the last data beat of the 32-byte cacheline or not assert \overline{STOP} at all.

Software workarounds:

Set bit 10, the master disabling streaming (MDS) bit, of the PCI bus function register (address 0x44) to prevent the combining of discrete outbound PCI writes or the recombining of writes that cross the 32-byte cacheline boundary into a burst.

Set the PCI latency timer register (offset 0x0D) to zero. A value of zero is the reset value for this register, so keeping this register unmodified after reset prevents the PCI IP block from ever combining writes. It is not necessary to set the PCI latency timer register to zero if the MDS bit is set.

Fix plan: No plans to fix

DMA2: Data corruption by DMA when destination address hold (DAHE) bit is used

Description: Devices: MPC8323E, MPC8321E

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

Examples of error condition are as follows:

- 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 one of the following options:

- Use a source address aligned to the destination transaction size
- Do not access any DMA registers while this type of DMA transfer is active

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: freescale.com/SalesTermsandConditions.

Freescale, the Freescale logo, AltiVec, C-5, CodeTest, CodeWarrior, ColdFire, ColdFire+, C-Ware, Energy Efficient Solutions logo, Kinetis, mobileGT, PowerQUICC, Processor Expert, QorIQ, Qorivva, StarCore, Symphony, and VortiQa are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. Airfast, BeeKit, BeeStack, CoreNet, Flexis, Layerscape, MagniV, MXC, Platform in a Package, QorIQ Qonverge, QUICC Engine, Ready Play, SafeAssure, SafeAssure logo, SMARTMOS, Tower, TurboLink, Vybrid, and Xtrinsic are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.

© 2014 Freescale Semiconductor, Inc.

