Freescale Semiconductor
Addendum

Document Number: MPC8560RMAD
Rev. 1.4, 02/2010

Errata to MPC8560 PowerQUICC III Integrated Communications Processor Reference Manual, Rev. 1

This errata describes corrections to the *MPC8560 PowerQUICC III Integrated Communications Processor Reference Manual, Rev. 1*. For convenience, the section number and page number of the errata item in the reference manual are provided. Items in bold are new since the last revision of this document.

To locate any published updates for this document, visit our website listed on the back cover of this document.

Section, Page No.	Changes
1.3.1, 1-10	Replace the second paragraph of the second note with the following: “Freescale offers a libcfsl_e500 library that uses SPE APU instructions. Freescale will also provide libraries to support next-generation PowerQUICC devices.”
2.3.1, 2-11	Insert the following new Section 2.3.1 and, subsequently, renumber existing sections.

2.3.1Accessing CCSR Memory from the e500 Core

When the local e500 processor is used to configure CCSR space, the CCSR memory space should typically be marked as cache-inhibited and guarded.

In addition, many configuration registers affect accesses to other memory regions; therefore, writes to these registers must be guaranteed to have taken effect before accesses are made to the associated memory regions.

To guarantee that the results of any sequence of writes to configuration registers are in effect, the final configuration register write should be chased by a read of the same register, and that should be followed by a SYNC instruction. Then, accesses can safely be made to memory regions affected by the configuration register write.

Section, Page No.	Changes
2.4, 2-50	In Table 2-9, “Memory Map,” change the access for registers TODR1 at offset 0x9_1A0C, TODR2 at offset 0x9_1A2C, TODR3 at offset 0x9_1A4C, and TODR4 at offset 0x9_1A6C from R/W to W.
3.1, 3-1	In Figure 3-1, “MPC8560 Signal Groupings,” remove LSDAMUX.
3.2, 3-14	In Table 3-3, “MPC8560 Reset Configuration Signals,” remove LSDAMUX.
4.2.2, 4-3	In Table 4-3, “Clock Signals—Detailed Signal Descriptions,” modify RTC description, as follows: “Real time clock. May be used (optionally) to clock the time base of the e500 core. The RTC timing specifications are given in the *MPC8560 Integrated Processor Hardware Specifications*, but the maximum input frequency should not exceed 1/4th the CCB frequency. See Section 4.4.4.4, “Real Time Clock.” This signal can also be used (optionally) to clock the global timers in the programmable interrupt controller (PIC).”
4.4.2, 4-10	Add the following note after step 4:

NOTE

If the JTAG signals are not used, $\overline{\text{TRST}}$ may be tied active; however it is recommended that $\overline{\text{TRST}}$ not remain asserted after the negation of $\overline{\text{HRESET}}$. $\overline{\text{TRST}}$ may be connected directly to $\overline{\text{HRESET}}$.

There is no need to assert the $\overline{\text{SRESET}}$ signal when $\overline{\text{HRESET}}$ is asserted. If $\overline{\text{SRESET}}$ is asserted upon negation of $\overline{\text{HRESET}}$, the POR sequence will be paused after the e500 core PLL is locked and before the e500 reset is negated. The POR sequence will be resumed when $\overline{\text{SRESET}}$ is negated.

Section, Page No. **Changes**

4.4.3, 4-11 Replace the second paragraph with the following:
"All POR configuration signals have internal pull-up resistors so that if the desired setting is high, there is no need for a pull-up resistor on the board."

5.1.1, 5-3 Replace the second paragraph of the note with the following:

"Freescale Semiconductor offers a libcfsl_e500 library that uses SPE instructions. Freescale will also provide libraries to support next-generation PowerQUICC devices."

5.2, 5-5 Change title of Table 5-1 from "Revision Level-to-Device Marking Cross-Reference" to "Device Revision Level Cross-References." In addition, update SVR numbers, as follows:

Table 5-1. Device Revision Level Cross-Reference

MPC8560 Revision	Core Revision	Processor Version Register (PVR)	System Version Register (SVR)
1.0	1.0	0x8020_0010	0x8070_0010
2.0	2.0	0x8021_0010	0x8070_0020

5.6, 5-18 In Figure 5-6, "e500 Core Programming Model," remove DVC1 and DVC2 registers.

5-14, 5-32 In Table 5-8, "Differences Between the e500 Core and the PowerQUICC III Core Implementation," replace the last sentence in the description of SPE and SPFP APUs with the following:

"Freescale offers a libcfsl_e500 library that uses SPE APU instructions. Freescale will also provide libraries to support next-generation PowerQUICC devices."

Section, Page No. **Changes**

5-14, 5-33 In Table 5-8, “Differences Between the e500 Core and the PowerQUICC III Core Implementation,” the description of HID1 fields should appear as follows:

Table 5-8. Differences Between the e500 Core and the PowerQUICC III Core Implementation

Feature	PowerQUICC Implementation
HID1 Implementation	PLL_MODE. Set to 01 PLL_CFG. PowerQUICC III devices support the following: 0001_00Ratio of 2:1 0001_01Ratio of 5:2 (2.5:1) 0001_10Ratio of 3:1 0001_11Ratio of 7:2 (3.5:1) NEXEN, R1DPE, R2DPE, MPXTT, MSHARS, SSHAR, ATS, and MID are not implemented On PowerQUICC III devices, ABE must be set to ensure that cache and TLB management instructions operate properly on the L2 cache.
	Please refer to the description of HID1[RFXE] in Section 6.10.2, “Hardware Implementation-Dependent Register 1 (HID1).” If RFXE is 0, conditions that cause the assertion of $\overline{\mathit{core\_fault\_in}}$ cannot directly cause the e500 to generate a machine check; however, PowerQUICC III devices must be configured to detect and enable such conditions. The following describes how error bits should be configured: • ECM mapping errors: EEER[LAEE] must be set. See Section 8.2.1.4, “ECM Error Enable Register (EEER).” • L2 multiple-bit ECC errors: L2ERRDIS[MBECCDIS] must be cleared to ensure that error can be detected. L2ERRINTEN[MBECCINTEN] must be set. See Section 7.3.1.5, “L2 Error Registers.” • DDR multiple-bit ECC errors. ERR_DISABLE[MBED] and ERR_INT_EN[MBEE] must be zero and DDR_SDRAM_CFG[ECC_EN] must be one to ensure that an interrupt is generated. See Section 9.4.1, “Register Descriptions.” • PCI. The appropriate parity detect and master-abort bits in ERR_DR must be cleared and the corresponding enable bits in ERR_EN must be set to ensure that an interrupt is generated. • Local bus controller parity errors. LTEDR[PARD] must be cleared and LTEIR[PARI] must be set to ensure that an parity errors can generate an interrupt. See Section 12.3.1.11, “Transfer Error Check Disable Register (LTEDR),” and Section 12.3.1.12, “Transfer Error Interrupt Enable Register (LTEIR).”

6.1.1, 6-2 Remove DVC 1 and DVC 2 from Figure 6-1, “Core Register Model.”

6.7.2.4, 6-23 In Figure 6-26, “Machine Check Syndrome Register (MCSR),” and in Table 6-12, “MCSR Field Descriptions,” remove GL_CI field (bit 47) and make it Reserved.

6.10.2, 6-26 In Table 6-17, “HID1 Field Descriptions,” change “01” to “11” in PLL_Mode description.

Also replace the description of HID1[RFXE] with the following:

Section, Page No. **Changes**

Table 6-17. HID1 Field Descriptions

Bits	Name	Description
46	RFXE	Read fault exception enable. Enables the core to internally generate a machine check interrupt when *core_fault_in* is asserted. Depending on the value of MSR[ME], this results in either a machine check interrupt or a checkstop. 0 Assertion of *core_fault_in* cannot cause a machine check. The core does not execute any instructions from a faulty instruction fetch and does not execute any load instructions that get their data from a faulty data fetch. On the e500v2, if these instructions are eventually required by the sequential programming model (that is, they are not in a speculative execution path), the e500v2 stalls until an asynchronous interrupt is taken. The e500v1 does not stall when faulty instructions or data are received, as described in the following note. **Note:** The e500v1 does not stall when faulty instructions or data are received. Instead, it continues processing with faulty instructions or data. The only reliable way to prevent such behavior is to set RFXE, which causes a machine check before the faulty instructions or data are used. To avoid the use of faulty instructions or data and to have good error determination, software must set RFXE and program the PIC to interrupt the processor when errors occur. As a result, software must deal with multiple interrupts for the same fundamental problem. 1 Assertion of *core_fault_in* causes a machine check if MSR[ME] = 1 or a checkstop if MSR[ME] = 0. The *core_fault_in* signal is asserted to the core when logic outside of the core has a problem delivering good data to the core. For example, the front-side L2 cache asserts *core_fault_in* when an ECC error occurs and ECC is enabled. As a second example, it is asserted when there is a master abort on a PCI transaction. See “Proper Reporting of Bus Faults” in the core complex bus chapter of the *PowerPC e500 Core Family Reference Manual*. The RFXE bit provides flexibility in error recovery. Typically, devices outside of the core have some way other than the assertion of *core_fault_in* to signal the core that an error occurred. Usually, this is done by channeling interrupt requests through a programmable interrupt controller (PIC) to the core. In these cases, the assertion of *core_fault_in* is used only to prevent the core from using bad data before receiving an interrupt from the PIC (for example, an external or critical input interrupt). Possible combinations of RFXE and PIC configuration are as follows: • RFXE = 0 and the PIC is configured to interrupt the processor. In this configuration, the assertion of *core_fault_in* does not trigger a machine check interrupt. The core does not use the faulty instructions or data and may stall. The PIC interrupts the core so that error recovery can begin. This configuration allows the core to query the PIC and the rest of the system for more information about the cause of the interrupt, and generally provides the best error recovery capabilities. • RFXE = 1 and the PIC is not configured to interrupt the processor. This configuration provides quick error detection without the overhead of configuring the PIC. When the PIC is not configured, setting RFXE avoids stalling the core when *core_fault_in* is asserted. Determination of the root cause of the problem may be somewhat more difficult than it would be if the PIC were enabled. • RFXE = 1 and the PIC is configured to interrupt the processor. In this configuration, the core may receive two interrupts for the same fundamental error. The two interrupts may occur in any order, which may complicate error handling. Therefore, this is usually not an interesting configuration for a single-core device. This may, however, be an interesting configuration for multi-core devices in which the PIC may steer interrupts to a processor other than the one that attempted to fetch the faulty data. • RFXE = 0 and the PIC is not configured to interrupt the processor. This is not a recommended configuration. The processor may stall indefinitely due to an unreported error.

7.3.1.5.2, 7-17	In Figure 7-17, “L2 Error Detect Register (L2ERRDET),” change field MULL2ERR, bit 0, from read-only to read/write.
7.10.1.2, 7-35	Add the following paragraph to the end of this section: “When enabling the L2, both in L2 cache mode and in memory-mapped SRAM mode, L2BLKSIZ must be set to a non-reserved value. By default L2BLKSZ value is 1b00; this is a reserved value and will cause the L2ERRDET[L2CFGERR] and L2CTL[L2E] not to be set.”
11.3, 11-4	Replace the introductory sentence with the following paragraphs: “Table 11-3 lists the I^{2}C-specific registers and their offsets. It lists the offset, name, and a cross-reference to the complete description of each register. Note that

Section, Page No. **Changes**

the full register address is comprised of CCSRBAR together with the block base address and offset listed in Table 11-3.

In this table and in the register figures and field descriptions, the following access definitions apply:

- Reserved fields are always ignored for the purposes of determining access type.
- R/W, R, and W (read/write, read only, and write only) indicate that all the non-reserved fields in a register have the same access type.
- w1c indicates that all of the non-reserved fields in a register are cleared by writing ones to them.
- Mixed indicates a combination of access types.
- Special is used when no other category applies. In this case the register figure and field description table should be read carefully.

All I^{2}C registers are 1 byte wide. Reads and writes to these registers must be byte-wide operations.

11.3.1.2, 11-6 In Table 11-5, "I2CFDR Field Descriptions," update I2CFDR[FDR] values 0x00–0x07 and 0x020–0x027.

Also, replace the bit description with the following:

Table 11-5. I2CFDR Field Descriptions

Bits	Name	Description
2–7	FDR	Frequency divider ratio. Used to prescale the clock for bit rate selection. The serial bit clock frequency of SCL is equal to the platform (CCB) clock divided by the designated divider. Note that the frequency divider value can be changed at any point in a program. The serial bit clock frequency divider selections are described as follows:

FDR	Divider (Decimal)	FDR	Divider (Decimal)	FDR	Divider (Decimal)
0x00	384	0x16	12288	0x2B	1024
0x01	416	0x17	15360	0x2C	1280
0x02	480	0x18	18432	0x2D	1536
0x03	576	0x19	20480	0x2E	1792
0x04	640	0x1A	24576	0x2F	2048
0x05	704	0x1B	30720	0x30	2560
0x06	832	0x1C	36864	0x31	3072
0x07	1024	0x1D	40960	0x32	3584
0x08	1152	0x1E	49152	0x33	4096
0x09	1280	0x1F	61440	0x34	5120
0x0A	1536	0x20	256	0x35	6144
0x0B	1920	0x21	288	0x36	7168
0x0C	2304	0x22	320	0x37	8192
0x0D	2560	0x23	352	0x38	10240
0x0E	3072	0x24	384	0x39	12288
0x0F	3840	0x25	448	0x3A	14336
0x10	4608	0x26	512	0x3B	16384
0x11	5120	0x27	576	0x3C	20480
0x12	6144	0x28	640	0x3D	24576
0x13	7680	0x29	768	0x3E	28672
0x14	9216	0x2A	896	0x3F	32768
0x15	10240				

Section, Page No. **Changes**

11.3.1.4, 11-7

In Table 11-7, “I2CSR Field Descriptions,” change “I2CSR” to “I2CDR” in the MCF bit description bullet point discussing the received case condition.

11.3.1.5, 11-9

In Table 11-8, “I2CDR Field Description,” modify the final sentence in the DATA description to read, “Note that in both master receive and slave receive modes, the very first read is always a dummy read.”

11.4.1.2, 11-11

Replace the last sentence of the first paragraph with the following:

“An I^{2}C device cannot be master and slave at the same time; if this is attempted, the results are boundedly undefined.”

11.4.5, 11-17

Replace the initial paragraph of this section with the following paragraphs:

“If boot sequencer mode is selected on POR (by the settings on the cfg_boot_seq[0:1] reset configuration signals, as described in Section 3.4.3.6, “Boot Sequencer Configuration”), the I^{2}C module communicates with one or more EEPROMs through the I^{2}C interface. The boot sequencer accesses the I^{2}C serial ROM device at a serial bit clock frequency equal to the platform (CCB) clock frequency divided by 2560. The EEPROM(s) can be programmed to initialize one or more configuration registers of this integrated device.

If the boot sequencer is enabled for normal I^{2}C addressing mode, the I^{2}C interface initiates the following sequence during reset:

1. Generate RESET sequence (START then 9 SCL cycles) to the EEPROM twice. This clears any transactions that may have been in progress prior to the reset.
2. Generate START
3. Transmit 0xA0 which is the 7-bit calling address (0b101_0000) with a write command appended (0 as the least significant bit).
4. Transmit 0x00 which is the 8-bit starting address
5. Generate a repeated START
6. Transmit 0xA1 which is the 7-bit calling address (0b101_0000) with a read command appended (1 as the least significant bit).
7. Receive 256 bytes of data from the EEPROM (unless the CONT bit is cleared in the data structure).
8. Generate a repeated START
9. Transmit 0xA2 which is the 7-bit calling address of the second target (0b101_0001) with a write command appended (0 as the least significant bit).
10. Transmit 0x00 which is the 8-bit starting address for the second target.
11. Generate a repeated START
12. Transmit 0xA3 which is the 7-bit calling address (0b101_0001) with a read command appended (1 as the least significant bit).
13. Receive another 256 bytes of data from the second EEPROM (unless the CONT bit is cleared in the data structure).

The sequence repeats with successive targets until the CONT bit in the data structure is cleared and the CRC check is executed. If the last register is not

Section, Page No.	Changes
	detected (that is, the CONT bit is never cleared) before wrapping back to the first address, an error condition is detected, causing the device to hang and the $\overline{\text{HRESET\_REQ}}$ signal to assert externally. The I^{2}C module continues to read from the EEPROM(s) as long as the continue (CONT) bit is set in the EEPROM(s). The CONT bit resides in the address/attributes field that is transferred from the EEPROM, as described in Section 11.4.5.1, “EEPROM Calling Address.” There should be no other I^{2}C traffic when the boot sequencer is active.”
11.5.4, 11-22	Remove the following sentence from the second paragraph: “For 1-byte transfers, a dummy read should be performed by the interrupt service routine (see Section 11.5.8, “Interrupt Service Routine Flowchart”).”
12.1.2, 12-2	Correct the first sub-bullet under primary bullet “Memory controller with eight memory banks” to read as follows: “— 32-bit address decoding with mask”
12.1.5, 12-4	Remove entire section.

Section, Page No.	Changes
12.3, 12-9	Update the first paragraph of the section, as follows: “Table 12-3 shows the memory mapped registers of the LBC and their offsets. It lists the offset, name, and a cross-reference to the complete description of each register. Note that the full register address is comprised of CCSRBAR together with the block base address and offset listed in Table 12-3. Undefined 4-byte address spaces within offset 0x000–0xFFF are reserved. In this table and in the register figures and field descriptions, the following access definitions apply: •Reserved fields are always ignored for the purposes of determining access type. •R/W, R, and W (read/write, read only, and write only) indicate that all the non-reserved fields in a register have the same access type. •w1c indicates that all of the non-reserved fields in a register are cleared by writing ones to them. •Mixed indicates a combination of access types. Special is used when no other category applies. In this case the register figure and field description table should be read carefully.”
12.3.1.2.1, 12-12	Correct the first two sentences of the first paragraph to read as follows: “The address mask field of the option registers (OR*n*[AM]) mask up to 17 corresponding BR*n*[BA] fields. The 15 lsbs of the 32-bit internal address do not participate in bank address matching in selecting a bank for access.”
12.3.1.2.1, 12-12	Remove the left-most column of Table 12-5.
12.3.1.10, 12-25	Add the following sentence to the end of the second paragraph: “Note that LTEATR[V] bit has to be cleared to register subsequent errors in LTESR.”
12.3.1.13, 12-28	Revise the last sentence of the first paragraph to read: “...., software must clear this bit to allow LBC error registers to update following subsequent errors.”
12.3.1.13, 12-28	In Figure 12-16, “Transfer Error Attributes Register (LTEATR),” and Table 12-19, “LTEATR Field Descriptions,” remove XA field, bits 28–29, and make it Reserved.
12.3.1.14, 12-29	In Table 12-20, “LTEAR Field Descriptions,” correct A field description to read as follows: “Transaction address for the error. Holds the 32-bit address of the transaction resulting in an error.”
12.3.1.15, 12-30	In Table 12-21, “LBCR Field Descriptions,” change AHD bit description, as follows: “0 During address phases on the local bus, the LALE signal negates two platform clock periods prior to the address being invalidated. At 666 MHz, this provides 3 ns of additional address hold time at the external address latch.

Section, Page No. **Changes**

1 During address phases on the local bus, the LALE signal negates one platform clock period prior to the address being invalidated. This halves the address hold time, but extends the latch enable duration. This may be necessary for very high frequency designs."

12.3.1.16, 12-31

In Table 12-22, "LCRR Field Descriptions," add the following note to the description of CLKDIV:

"**Note:** It is critical that no transactions are being executed through the local bus while CLKDIV is being modified. As such, prior to modification, the user must ensure that code is not executing out of the local bus. Once LCRR[CLKDIV] is written, the register should be read, and then an isync should be executed."

12.4, 12-33

In Figure 12-20, correct the label for the arrow in the upper left corner to read "32-bit System Address."

12.4.1.1, 12-33

Modify the third sentence of the first paragraph to read as follows:

"Addresses are decoded by comparing the 17 msbs of the address, masked by OR*n*[AM], with the base address for each bank (BR*n*[BA])."

12.4.1.7, 12-37

Replace the last sentence of the first paragraph with the following:

"Setting LTEDR[BMD] disables bus monitor error checking (i.e. the LTESR[BM] bit is not set by a bus monitor time-out); however, the bus monitor is still active and can generate a UPM exception (as noted in Section 12.4.4.1.4, "Exception Requests") or terminate a GPCM access."

12.4.2.2, 12-43

Update bullet list as follows:

- Simultaneous with the latched memory address. (This refers to the externally latched address, not the address timing on LAD[0:31]. That is, chip select does not assert during LALE).
- One quarter of a clock cycle later (for LCRR[CLKDIV] = 4 or 8).
- One half of a clock cycle later (for LCRR[CLKDIV] = 2, 4 or 8).
- One clock cycle later (for LCRR[CLKDIV] = 4) when OR*n*[XACS] = 1.
- Two clock cycles later (for LCRR[CLKDIV] = 2, 4 or 8), when OR*n*[XACS] = 1.
- Three clock cycles later (for LCRR[CLKDIV] = 2, 4 or 8), when OR*n*[XACS] = 1 and OR*n*[TRLX] = 1.

Section, Page No. **Changes**

12.4.2.2.3, 12-44 Change the title of Figure 12-26 to “GPCM Relaxed Timing Read (XACS = 0, ACS = 1x, SCY = 1, EHTR = 0, TRLX = 1,” and replace the figure with the following:

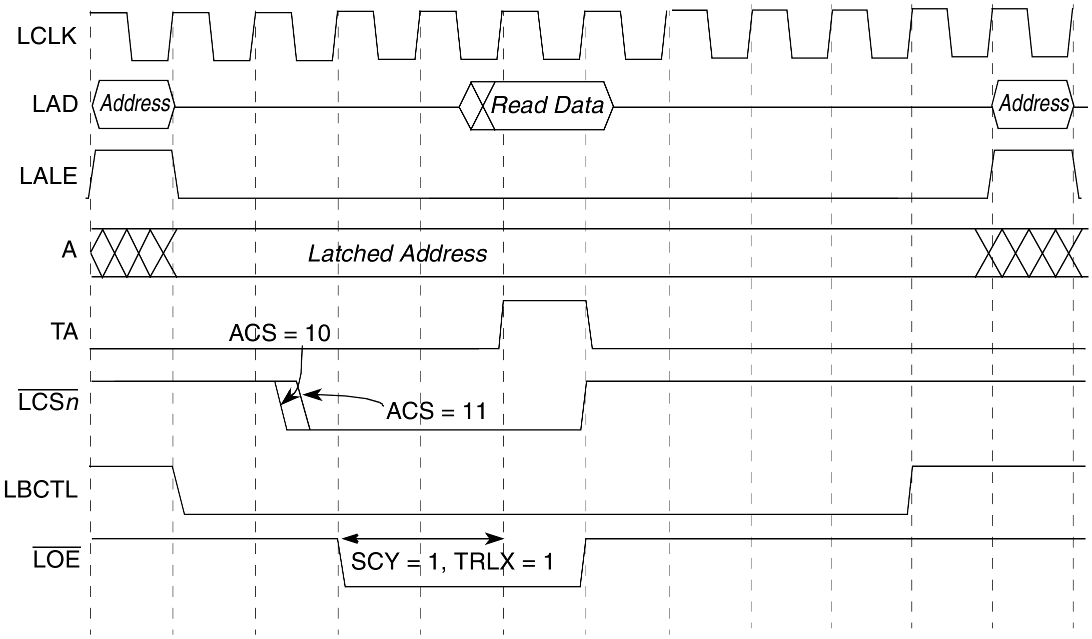

Figure 12-26. GPCM Relaxed Timing Read (XACS = 0, ACS = 1x, SCY = 1, EHTR = 0, TRLX = 1

12.4.2.4, 12-50 In Table 12-27, remove fields BR0[XBA] and OR0[XAM].

12.4.3.1, 12-50 Modify the last sentence of the first paragraph to read “Note that address signals A[2:0] of the SDRAM connect directly to LA[27:29], address signal A10 connects to the LBCs dedicated LSDA10 signal, while the remaining address bits (except A10) are latched from LAD[20:26].”

12.4.3.2, 12-51 Add the following paragraph after the bulleted list:

“The initial commands are executed by setting LSDMR[OP] and accessing the SDRAM with any write that hits the relevant bank. Since the result of any update to the LSDMR must be in effect before accessing the SDRAM with any write, a write to LSDMR should be followed immediately by a read from LSDMR, which must complete prior to an initial write to SDRAM. Further, the first write to SDRAM should be followed immediately by an SDRAM read, which must complete prior to additional LSDMR updates. This enforces a proper ordering between updates to the LSDMR and write accesses to the SDRAM. If the initialization is being done by the e500, this described protocol is guaranteed only if the SDRAM is mapped as cache-inhibited and guarded, as the CCSR memory region containing LSDMR should be. If the initialization is from an external host, said host must ensure completion of LSDMR and SDRAM reads prior to subsequent writes, as described above.”

Section, Page No.	Changes
12.4.4.2, 12-65	Remove the third paragraph, beginning with “Note that the UPM memory region....” In addition, add the following note to the end of the section:

NOTE

In order to enforce proper ordering between updates to the MxMR register and the dummy accesses to the UPM memory region, two rules must be followed:

1. Since the result of any update to the MxMR register must be in effect before the dummy read or write to the UPM region, a write to MxMR should be followed immediately by a read of MxMR.
2. The UPM memory region should have the same MMU settings as the memory region containing the MxMR configuration register; both should be mapped by the MMU as Cache-Inhibited and Guarded. This prevents the e500 core from re-ordering a read of the UPM memory around the read of MxMR. Once the programming of the UPM array is complete, the MMU setting for the associated address range can be set to the proper mode for normal operation, such as cacheable and copyback.

12.4.4.2, 12-65	Add the following two subsections to Section 12.4.4.2, “Programming the UPMs.”

12.4.4.2.1 UPM Programming Example (Two Sequential Writes to the RAM Array

The following example further illustrates the steps required to perform two writes to the RAM array at non-sequential addresses assuming that the relevant BR*n* and OR*n* registers have been previously setup.

1. Program M*x*MR for the first write (with desired RAM array address).
2. Write pattern/data to MDR to ensure that the M*x*MR has already been updated with the desired configuration.
3. Read MDR to ensure that the MDR has already been updated with the desired pattern. (Or, read M*x*MR if step 2 is not performed.)
4. Preform a dummy write transaction.(Write transaction can now be performed.)
5. Read/check M*x*MR[MAD]. If incremented, then the previous dummy write transaction is completed; proceed to step 6. Repeat step 5 until incremented.
6. Program M*x*MR for the second write with the desired RAM array address.
7. Write pattern/data to MDR to ensure that the M*x*MR has already been updated with the desired configuration.
8. Read MDR to ensure that the MDR has already been updated with the desired pattern.
9. Perform a dummy write transaction.(Write transaction can now be performed.)
10. Read/check M*x*MR[MAD]. If incremented, then the previous dummy write transaction is completed.
11. Note that if step 1 (or 6) and 2 (or 7) are reversed, then step 3 (or 8) is replaced by the following:

12. Read M*x*MR to ensure that the M*x*MR has already been updated with the desired configuration.

12.4.4.2.2 UPM Programming Example (Two Sequential Reads from the RAM Array)

RAM array contents may also be read for debug purposes, for example, by alternating dummy read transactions, each time followed by reads of MDR (when M*x*MR[OP] = 0b10). The following example further illustrates the steps required to perform two reads from the RAM array at non-sequential addresses assuming that the relevant BR*n* and OR*n* registers have been previously setup.

1. Program M*x*MR for the first read with the desired RAM array address.
2. Read M*x*MR to ensure that the M*x*MR has already been updated with the desired configuration, such as RAM array address.
3. Perform a dummy read transaction.(Read transaction can now be performed.)
4. Read/check M*x*MR[MAD]. If incremented, then the previous dummy read transaction is completed; proceed to step 5. Repeat step 4 until incremented.
5. Read MDR.
6. Program M*x*MR for the second read with the desired RAM array address.
7. Read M*x*MR to ensure that the M*x*MR has already been updated with the desired configuration, such as RAM array address.
8. Perform a dummy read transaction.(Read transaction can now be performed.)
9. Read/check M*x*MR[MAD]. If incremented, then the previous dummy read transaction is completed; proceed to step 10. Repeat step 9 until incremented.
10. Read MDR.

12.4.4.4.1, 12-69	In Table 12-30, “RAM Word Field Descriptions,” add note stating, “AMX must not be changed from its previous value in any RAM word which begins a loop” to LOOP and AMX field descriptions.
12.4.4.4.7, 12-74	Add the following note to end of section: “AMX must not change values in any RAM word which begins a loop.”
12.5.3, 12-87	Add the following note after the first paragraph: “**Note:** It may not be possible to write to 16-bit devices on the local bus using 16-bit transactions on one of the external peripheral interfaces. Refer to the chapter describing the specific external interface controller for more information.”
12.5.4.3.3, 12-94	In Table 12-39, remove fields BR*n*[XBA] and OR*n*[XAM].
12.5.4.3.5, 12-95	In Table 12-43, remove fields BR*n*[XBA] and OR*n*[XAM].
13.2, 13-7	Change the fifth element of the second bullet item in the TSEC features list to read: “— 10/100 Mbps RGMII”
13.5.3.2.1, 13-29	In Table 13-12, “FIFO_PAUSE_CTRL Field Descriptions,” update the TFC_PAUSE_EN field description as follows: “TFC_PAUSE enable. This bit enables the ability to transmit a pause control frame by setting the

Section, Page No.	Changes
	TCTRL[TFC_PAUSE] bit. This bit is cleared at reset but should always be set during initialization as undefined behavior results during normal operation when left cleared. 0 Pause control frame transmission disabled (default, but must be set during initialization). 1 Pause control frame transmission enabled.”
13.5.3.9.1, 13-89	In Table 13-96, change ATTR[ELCWT] and ATTR[BDLWT] field values of 11 to be “Reserved.”
13.5.4.2, 13-92	In Figure 13-100 and Table 13-99, change default reset values of control register fields AN Enable, Full Duplex, and Speed_1 to all be zero.
13.5.4.6, 13-98	In Figure 13-104, change the reset value of the TBI ANEX register to be 0000_0000_0000_0100.
13.6.1.1, 13-104	In Figure 13-110, “TSEC-MII Connection,” remove TSECn_COL signal.
14.1.4, 14-3	Replace Table 14-2, “DMA Mode Field Descriptions,” with the following table:

Table 14-2. DMA Mode Field Descriptions

Modes with Features	MR*n*[XFE]	MR*n*[CTM]	MR*n*[SRW]	MR*n*[CDSM/SWSM]	MR*n*[EMS_EN]
Basic Direct Modes					
Basic direct	0	1	0	0	0
Basic direct external control	0	1	0	0	1
Basic direct single-write start	0	1	1	1 or 0	0
Basic Chaining Modes					
Basic chaining	0	0	Reserved	0	0
Basic chaining external control	0	0	Reserved	0	1
Basic chaining single-write start	0	0	Reserved	1	0
Extended Direct Modes					
Extended direct	1	1	0	0	0
Extended direct external control	1	1	0	0	1
Extended direct single-write start	1	1	1	1 or 0	0
Extended Chaining Modes					
Extended chaining	1	0	Reserved	0	0
Extended chaining external control	1	0	Reserved	0	1
Extended chaining single-write start	1	0	Reserved	1	0

Section, Page No.	Changes
14.4.1.3, 14-34	In the third paragraph, add the following after the second sentence: “Note, however, that write data for a paused transfer may not have reached the target interface when so indicated.”
15.3.1.2.1, 15-21	In Table 15-7, “POTAR*n* Field Descriptions,” update TA description to read: “Translation address. Represents bits 31–12 of the PCI address. The specified address must be aligned to the window size, as defined by POWAR*n*[OWS].”
15.3.1.3, 15-24	Change final sentence of the paragraph beginning “Each window’s base address and translation address...” to read, “In addition, if inbound ATMU windows are overlapped, the ATMU windows must not map to the same address with different sets of attributes (other than window size).”
15.3.1.3.1, 15-25	In Table 15-11, “PITAR*n* Field Descriptions,” update TA description to read: “Translation address. Indicates the starting point of the inbound translated address. The specified address must be aligned to the window size, as defined by PIWAR*n*[IWS]. TA corresponds to the high-order 20 bits of a 32-bit local address.”
15.3.1.3.2, 15-26	In Table 15-12, “PIWBAR*n* Field Descriptions,” add the following to BA field description: “The specified address must be aligned to the window size, as defined by PIWAR*n*[IWS].”
15.4.2.2, 15-58	In Table 15-52, “PCI Bus Commands,” add the following sentence to the Memory-write command (0b0111) definition: “When a PCI master issues this command to local memory, the MPC8560 (the target) fetches data from the requested address to the end of the cache line (32 bytes) from local memory, even though all of the data may not be requested by (or sent to) the initiator.”
15.4.2.8.2, 15-66	Replace the bullet item reading, “The 16-clock latency timer has expired, and the first data phase has not begun” with “The 32-clock latency timer has expired, and the first data phase has not begun.”
15.4.2.11.3, 15-74	Replace the second to the last sentence of the first paragraph with the following: “When the MPC8560 is in agent lock mode, it retries all externally-generated PCI/X configuration cycles until the ACL bit in the PCI bus function register (0x44) is cleared.”
15.4.2.11.4, 15-75	In Table 15-55, change the title from “AD*n* Used in IIDSEL,” to “AD*n* Used in IDSEL.”
15.4.2.11.4, 15-75	In Table 15-55, separate entries for device number 0 and devices 1–9. Also, add the following table footnote related to IDSEL used for device number 0: “No external configuration transaction takes place; rather, internal registers are accessed.” In addition, insert the following table footnote related to IDSEL used for device numbers 1-9: “No IDSEL line asserted. Type0 configuration transaction is run, but ends with a master abort since no device responds.”

Section, Page No.	Changes
15.5.1.3, 15-100	Insert the following sentence after the first sentence: “The purpose of this mode is to allow initial configuration on the post by the local processor before opening the port to be further configured by the external host.”
16.3.2.2.3, 16-40	In Table 16-37, add the following note to the ROWAR*n*[WRTYP] field description: “**Note:** A sub-eight byte transaction that maps to an SWRITE is converted to an NWRITE. SWRITE requires the write size to be a multiple of 8 bytes.”
16.3.2.3.3, 16-50	In Table 16-43, “PNFEIER Field Descriptions,” for all field descriptions, change bit setting for 0 to “interrupt disabled” and bit setting for 1 to “interrupt enabled.”
16.5.1.1, 16-97	In Table 16-92, “General Device 8/16 LP-LVDS Physical Layer Nonrecoverable Errors List,” update item 2’s “Meets Reqs?” column to read “n/a” with the following footnote: “Since the RapidIO controller does not generate throttle requests it will never experience a late or non-existent response to a throttle request. The RapidIO controller will respond to throttle requests generated by its link partner.”
17.4.1.11, 17-13	In Figure 17-11 and Table 17-14, document bit 11 as ‘Reserved’.
17.4.1.11, 17-14	In Table 17-14, change the last sentence of the asserted state description for bit DEVDISR[E500] to read as follows: “Instruction fetching is stopped, snooping is disabled, and clocks are shut down to all functional units of the core, including the timer facilities.”
17.5.1.7, 17-26	In Figure 17-20, change the input of the NOR gate which drives internal signal core_tben, formerly noted as being driven by DEVDISR[TB], to be noted as being driven by DEVDISR[TB] or [E500].
Chapter 18, 18-1	Change chapter title to “Device Performance Monitor.”
18.4.7, 18-23	In Table 18-10, insert a centered header row labeled “TSEC1 DMA Events” directly between rows for events “Receive FIFO above 3/4” and “DMA reads,” as follows:

Receive FIFO above 3/4	Ref:49	—
	TSEC1 DMA Events	
DMA reads	C1:47	Descriptor and data reads

Section, Page No.	Changes
20.1.1, 20-5	In Table 20-1, “MPC8560 Internal Memory Map, update SCCR reset value to “0x0000_0001.”
20.1.1, 20-7	In Table 20-1, “MPC8560 Internal Memory Map,” remove reserved registers between each FCCE*n* and FCCM*n* at offsets 0x9_1312, 0x9_1332, and 0x9_1352.
20.1.1, 20-15	In Table 20-1, change the access for registers TODR1 at offset 0x9_1A0C, TODR2 at offset 0x9_1A2C, TODR3 at offset 0x9_1A4C, and TODR4 at offset 0x9_1A6C from read/write to write-only.
20.1.1, 20-15	In Table 20-1, “MPC8560 Internal Memory Map,” add reserved registers between each SCCE*n* and SCCM*n* at offsets 0x9_1A12, 0x9_1A32, 0x9_1A52, and 0x9_1A72.

Section, Page No.	Changes
20.4.1, 20-31	In Figure 20-9, “CP Command Register,” for bits 26–27 replace ‘—’ with ‘EN.’ In Table 20-10, “CP Command Register Field Descriptions,” replace ‘Reserved’ bits 26–27 with ‘EN’ and add the following description. For bits 18–25, add 0x0C Ethernet to the list of FCC protocols.

Bits	Name	Description
18–25	MCN	In FCC protocols, this field contains the protocol code as follows: 0x00 HDLC 0x0A ATM 0x0C Ethernet 0x0F Transparent
26–27	EN	Endpoint: Logical pipe number (only in USB) 00 ENDPOINT 0 01 ENDPOINT 1 10 ENDPOINT 2 11 ENDPOINT 3

Section, Page No.	Changes
20.4.1.1, 20-33	In Table 20-11, “CP Command Opcodes,” add row for 0x1110. In addition, place “RESET SU FILTER” in the MCC column.
20.4.1.1, 20-34	In Table 20-12, “Command Descriptions,” add RESET SU FILTER after MCC RESET. Use the description, “This command resets the filtering algorithm to ensure that the next SU will be received, even if it would normally have been filtered. This command could be issued periodically so that the CPU can check to make sure that the link is really up and not simply receiving flags.”
27.1, 27-2	Update the second bullet point beginning “Additional protocols supported...” to read, “Additional protocols may be available through the use of RAM-based microcodes. Please contact your Freescale sales office.”
27.2.3, 27-10	In Figure 27-5, “Transmit-on-Demand Register (TODR),” change the TODR register access from read/write to write-only. Also, in Table 27-3, revise the second sentence of bit setting 1 in the TODR[TOD] description to read “TOD is cleared automatically after one system clock cycle, but...”.
29.14, 29-19	In Figure 29-10, “Typical HDLC Bus Multimaster Configuration,” Figure 29-11, “Typical HDLC Bus Single-Master Configuration,” Figure 29-14, “HDLC Bus Transmission Line Configuration,” and Figure 29-16, “HDLC Bus TDM Transmission Line Configuration,” changed power supply voltage from 5 V to 3.3 V.
33.13.2, 33-34	In Table 33-20, “TxBD Field Descriptions,” add the following note to the bit 4, L field description**,** “In Transparent Mode operation, setting Last puts the channel to Idle state after sending the last byte of the buffer (or the CRC, if enabled). To resume transmit, set the POL bit in the CHAMR. For continuous transmission—for example, with no concept of a frame boundary—Last should NOT be set in Transparent Mode.”

Section, Page No. **Changes**

33.13.2, 33-36 In Table 33-20, “TxBD Field Descriptions,” update the end of I field (bit 3) description, as follows:

1 The TXB bit in the event register is set when this buffer is serviced. TXB can cause an interrupt if enabled.

35.10.1.3, 35-43 In Figure 35-24, “Global Mode Entry (GMODE),” and Table 35-13, “GMODE Field Descriptions,” add field GBL (bit 2), as follows:. Note that only the affected rows are shown:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Field	0	0	GBL	0	0	0	ALB	CTB	REM	0	0	UEAD	CUAB	EVPT	0	ALM

Figure 35-24. Global Mode Entry (GMODE)

Table 35-13. GMODE Field Descriptions

Bits	Name	Description
0–1	—	Reserved
2	GBL	Global. Asserting GBL enables snooping of connection tables. GBL should not be asserted if any of the related DMAs access the local bus.
3–5	—	Reserved

40.18.1, 40-21 In Table 40-9, “FPSMR Ethernet Field Descriptions,” modify LPB (bit 3) field description, as follows:

“Local protect bit.

0 Receiver is blocked when transmitter sends (default).

1 Receiver is not blocked when transmitter sends. Must set for full-duplex operation. For external loopback, GFMR[DIAG] must be programmed also; see Section 34.2, ‘General FCC Mode Registers (GFMRx).’”

45.5, 45-16 In Table 45-7, “Port C Dedicated Pin Assignment (PPARC = 1),” change entry for Default Input at PC15 to “by PC29,” and change entry for Default Input at PC13 to “by PC28.”

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor, Inc.
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
1-800-521-6274 or
+1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku
Tokyo 153-0064
Japan
0120 191014 or
+81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor
Literature Distribution Center
1-800 441-2447 or
+1-303-675-2140
Fax: +1-303-675-2150
LDCForFreescaleSemiconductor
@hibbertgroup.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor 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 which may be provided in Freescale Semiconductor 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 Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

Document Number: MPC8560RMAD
Rev. 1.4
02/2010