

# Errata to MPC8536E PowerQUICC III™ Integrated Processor Reference Manual, Rev. 1

This errata describes corrections to the *MPC8536E PowerQUICC III™ Integrated Processor Reference Manual*, Revision 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
1.3.14, 1-9	Change first sentence to read “The integrated four-channel DMA controller can transfer data between any I/O or memory ports or between two devices or locations on the same port.”
4.4.3.1, 4-12	In Table 4-9, “CCB Clock PLL Ratio,” change default value from 1111 to No Default.
4.4.3.2, 4-12	In Table 4-10, “e500 Core Clock PLL Ratios,” change default value from 111 to No Default.
5.3, 5-5	In Table 5-1, “Differences Between the e500 Core and the PowerQUICC III Core Implementation,” update bit settings in row “TCR (timer control register)” as follows: <ul style="list-style-type: none"> <li>00 No watchdog timer reset occurs.</li> <li>01 Second timeout generates machine check</li> <li>10 Second timeout generates HRESET_REQ output externally</li> <li>11 Second timeout automatically resets the given processor core</li> </ul>

Section, Page No.	Changes																				
8.4.1.4, 8-17	<p>In Table 8-9, “TIMING_CFG_3 Field Descriptions,” modify descriptions for 001 and 010 in CNTL_ADJ (bits 29–31) field description, as follows:</p> <p>“001 MODT[0:3], <math>\overline{MCS}</math>[0:3], and MCKE[0:3] are launched 1/4 DRAM cycle later than the other DRAM address and control signals.</p> <p>010 MODT[0:3], <math>\overline{MCS}</math>[0:3], and MCKE[0:3] are launched 1/2 DRAM cycle later than the other DRAM address and control signals.”</p>																				
8.4.1.5, 8-19	<p>In Table 8-10, “TIMING_CFG_0 Field Descriptions,” add the following statement to the ODT_PD_EXIT (bits 20–23) field description:</p> <p>“ODT_PD_EXIT must be greater than TIMING_CFG_5[RODT_ON] when using RODT_ON overrides and must be greater than TIMING_CFG_5[WODT_ON] when using WODT_ON overrides.”</p> <p>In same table, add 1 cycle to all settings for field ACT_PD_EXIT (but not 000), as follows:</p> <table border="0"> <tr><td>000</td><td>Reserved</td></tr> <tr><td>001</td><td>2 clocks</td></tr> <tr><td>010</td><td>3 clocks</td></tr> <tr><td>...</td><td></td></tr> <tr><td>111</td><td>8 clocks</td></tr> </table> <p>In same table, add 1 cycle to all settings for field PRE_PD_EXIT (but not 0000), as follows:</p> <table border="0"> <tr><td>0000</td><td>Reserved</td></tr> <tr><td>0001</td><td>2 clocks</td></tr> <tr><td>0010</td><td>3 clocks</td></tr> <tr><td>...</td><td></td></tr> <tr><td>1111</td><td>16 clocks</td></tr> </table>	000	Reserved	001	2 clocks	010	3 clocks	...		111	8 clocks	0000	Reserved	0001	2 clocks	0010	3 clocks	...		1111	16 clocks
000	Reserved																				
001	2 clocks																				
010	3 clocks																				
...																					
111	8 clocks																				
0000	Reserved																				
0001	2 clocks																				
0010	3 clocks																				
...																					
1111	16 clocks																				
8.4.1.7, 8-22	<p>In Table 8-12, “TIMING_CFG_2 Field Descriptions,” change 11111 encoding for CPO (bits 4–8) from “Reserved” to “Automatic CPO.”</p>																				
8.4.1.10, 8-29	<p>Add the following sentence to the existing DDR_SDRAM_MODE[ESDMODE] field description:</p> <p>The write leveling enable bit should be cleared by software in this field.</p>																				
8.4.1.12, 8-31	<p>In Table 8-17, “DDR_SDRAM_MD_CNTL Field Descriptions,” delete second row for field CS_SEL (shows field as bits 1–3). Note that the first row, showing field as bits 2–3, is correct.</p>																				
8.4.1.19, 8-38	<p>In Table 8-25, “TIMING_CFG_5 Field Descriptions,” modify the last sentence in RODT_ON (bits 3–7) and WODT_ON (bits 15–19) field descriptions to say the following: “If 2T/3T timing is used, one/two extra cycle(s) is/are automatically added to the value selected in this field.”</p>																				
8.4.1.21, 8-43	<p>In Table 8-27, “DDR_WRLVL_CNTL Field Descriptions,” change bit settings for WRLVL_START to the following:</p> <table border="0"> <tr><td>0000</td><td>0 clock delay</td></tr> <tr><td>0001</td><td>1/8 clock delay</td></tr> </table>	0000	0 clock delay	0001	1/8 clock delay																
0000	0 clock delay																				
0001	1/8 clock delay																				

0010 1/4 clock delay  
 0011 3/8 clock delay  
 0100 1/2 clock delay  
 0101 5/8 clock delay  
 0110 3/4 clock delay  
 0111 7/8 clock delay  
 1000 1 clock delay  
 1001 9/8 clock delay  
 1010 5/4 clock delay  
 1011 11/8 clock delay  
 1100 3/2 clock delay  
 1101 13/8 clock delay  
 1110 7/4 clock delay  
 1111 15/8 clock delay

**8.4.1.23, 8-44**

Correct the second sentence of this section to agree with the register field description table. (Formerly, the bit ordering was reversed.) The sentence should read as follows:

Each 4-bit field represents the value that is placed on MBA[1], MBA[0], MA[4], and MA[3] during register control word writes.

In same section, correct each of the field descriptions in Table 8-29, “DDR\_Register Control Word 1 Field Descriptions,” such that each description properly corresponds to the fieldname being described. (Formerly, each field description incorrectly referenced, “Register Control Word 0.”)

**8.4.1.24, 8-45**

Correct the second sentence of this section to agree with the register field description table. (Formerly, the bit ordering was reversed.) The sentence should read as follows:

Each 4-bit field represents the value that is placed on MBA[1], MBA[0], MA[4], and MA[3] during register control word writes.

In same section, correct each of the field descriptions in Table 8-30, “DDR\_Register Control Word 2 Field Descriptions,” such that each description properly corresponds to the fieldname being described. (Formerly, each field description incorrectly referenced, “Register Control Word 0.”)

**8.4.1.27, 8-48**

Modify text concerning legal impedance that follow the numbered list, as follows:

“Note that the legal impedance values (from lowest impedance to highest impedance) for DDR2 (1.8 V) are as follows:

- 0000
- 0001
- 0011
- 0010
- 0110
- 0111
- 0101

- 0100
- 1100
- 1101
- 1111 (used for half-strength mode when driver calibration is not used)
- 1110
- 1010 (default full-strength impedance)
- 1011
- 1001

Note that the legal impedance values (from highest impedance to lowest impedance) for DDR3 (1.5 V) are as follows:

- 0000
- 0001
- 0011
- 0010
- 0110
- 0111 (used for half-strength mode when driver calibration is not used)
- 0101
- 0100
- 1100
- 1101
- 1110
- 1010
- 1011 (default full-strength impedance)
- 1001

Note that the drivers may either be calibrated to full-strength or half-strength.”

9.1.1.2, 9-4

In Table 9-1, “Processor Interrupts Generated Outside the Core—Types and Sources,” replace description in rows “Unconditional debug event” and “Reset,” as follows:

**Table 9-1. Processor Interrupts Generated Outside the Core—Types and Sources**

Core Interrupt Type	Signaled by (Input to Core)	Sources
Unconditional debug event	<i>coren_ude</i>	$\overline{UDE}$ . Asserting $\overline{UDE}$ generates an unconditional debug exception type debug interrupt and sets a bit in the debug status register, DBSR[UDE], as described in the <i>e500 Core Family Reference Manual</i> .
Reset	<i>coren_hreset</i>	<ul style="list-style-type: none"> <li>• <math>\overline{HRESET}</math> assertion (and negation)</li> <li>• <i>core_hreset_req</i>. Output from core—caused by writing to the core DBCR0[RST]. This condition is additionally qualified with MSR[DE] and DBCR0[IDM] bits. Note that assertion of this signal causes a hard reset of the core only.</li> <li>• <i>core_reset</i>. Output from PIC. See <a href="#">Section 9.3.1.6, “Processor Core Initialization Register (PIR).”</a></li> </ul>

9.1.3.2, 9-5

Add sentence:

It is required that in pass-through mode the internal and external interrupt targets should be *int*, which is by default.

9.3.3.1, 9-29

Change note at start of section to read as follows:

ERQSR fields report only the current logic level of IRQ0–IRQ11 pins....

In same section, Figure 9-18, “External Interrupt Summary Register (ERQSR),” should appear as follows:



**Figure 9-18. External Interrupt Summary Register (ERQSR)**

10.5.4.4, 10-54

Change reset value of ID register to 0x0030\_0300\_0001\_0000.

10.5.4.5, 10-54

Change reset value of IP block revision register to 0x0030\_0300\_0001\_0000.

10.7.1.2, 10-59

In Table 10-22, “AESU Mode Register Field Descriptions” update bit 59 description to use context registers 5–6 in CMAC Cipher Mode description instead of context registers 3–4.

10.7.1.11.1, 10-69

In subsection, “Context for CBC, CBC-RBP, OFB, and CFB128 Cipher Modes,” modify the bullets to say the following:

- Context register 1 holds the *most* significant bytes of the initialization vector (bytes 1–8).
- Context register 2 holds the *least* significant bytes of the initialization vector (bytes 9–16).

10.7.1.11.1, 10-69

Update the following subsections, as follows:

### Context for Counter (CTR) Cipher Mode

In counter cipher mode, a random 128-bit initial counter value is incremented modulo  $2^M$  with each block processed. The running counter is encrypted and XORed either with the plaintext to derive the ciphertext or with the ciphertext to recover the plaintext. The modulus exponent M can be set between 8 and 128, in multiples of 8.

There are two options for loading CTR mode context. When using descriptor type 0001\_0, AES-CTR context is loaded as shown in Table 10-29. The Context In length must be set to 56B, and the context itself must be 32B of zeros (context registers 1–4), followed by the initial counter value (context registers 5–6), and finally the modulus exponent M in context register 7. When using descriptor type 0000\_0, 24B of context can be loaded as shown for SRT, dispensing with the need for the initial zeros.

### Context for SRT Cipher Mode

As mentioned in the footnote to Table 10-29, SRT is not a new AES cipher mode but rather an AESU method of performing AES-CTR cipher mode with reduced context loading overhead. This mode was originally developed for SRTP, but is also applicable to the use of AES-CTR for LTE EEA2. SRT cipher mode can be used with descriptor type 0000\_0 (aes\_ctr\_nosnoop) and type 0010\_1 (SRTP). As with CTR

cipher mode, a random 128-bit initial counter value is incremented modulo  $2^M$  with each block processed. The running counter is encrypted and XORed with the plaintext to derive the ciphertext, or with the ciphertext to recover the plaintext. The modulus exponent M can be set between 8 and 128 in multiples of 8.

As shown in [Table 10-29](#), in SRT mode context registers 1–2 hold the initial counter value, and context register 3 holds the modulus exponent M.

**NOTE**

There are two methods of performing AES-CTR with reduced context loading, as follows:

- Use descriptor type 0000\_0 with the AES mode register set to CTR.
- Use descriptor type 0001\_0 with the AES mode register set to SRT.

These methods are completely equivalent for cipher-only operations (no snooping for ICV generation). When performing AES-CTR in conjunction with HMAC-SHA-1, reduced context loading can only be achieved by using SRT.

**10.7.1.11.2 Context for Data Integrity Cipher Modes**

The context registers for the different cipher modes that provide data integrity only are summarized in [Table 10-2](#). The registers are described in more detail in the following subsections.

**Table 10-2. AESU Context Registers for Integrity Modes**

Context Register # (byte address)	Cipher Mode providing only Data Integrity		
	XCBC-MAC	GCM-GHASH	CMAC (OMAC1)
1 (0x34100)	Computed MAC	Computed MAC	Computed MAC
2 (0x34108)			
3 (0x34110)	Received MAC*		Received MAC*
4 (0x34118)			
5 (0x34120)	Key 3		E(K, 0 <sup>128</sup> )
6 (0x34128)			
7 (0x34130)	Key 2	len(AAD) <sup>T</sup>	
8 (0x34138)			
9 (0x34140)	Key 1**	H	
10 (0x34148)			

**Table 10-2. AESU Context Registers for Integrity Modes (continued)**

Context Register # (byte address)	Cipher Mode providing only Data Integrity		
	XCBC-MAC	GCM-GHASH	CMAC (OMAC1)
11 (0x34150)		len(AAD) <sup>C</sup>	

**Notes:**

Context register 12 is unused for these modes

\* Used only in ICV mode—must be written at start of new message for ICV checking

<sup>C</sup> Length of data processed with current descriptor (in bits)

<sup>T</sup> Length of total data (in bits)

**Context and Operation for XCBC-MAC Cipher Mode**

XCBC-MAC cipher mode is an authentication-only mode of AES. Normal CBC-MAC runs AES in CBC cipher mode and assigns the final ciphertext result as the MAC. XCBC-MAC supports only 16-byte keys.

The AESU supports three mode options while operating in XCBC-MAC cipher mode. These options are controlled by the AUX bits in the AESU mode register, as defined in the list below. The encrypt/decrypt bit has no meaning in an XCBC-MAC operation and is ignored by the AESU.

- AUX0 Controls whether the XCBC-MAC is completed with this descriptor, or whether this descriptor is only doing a portion of the total MAC generation, and context needs to be output so that it can be reloaded for subsequent operations.
- AUX1 Controls whether K1, K2, and K3 are initialized at the start of the descriptor or whether previously initialized keys are reloaded.
- AUX2 Controls whether the AESU is to perform automatic checking of a received MAC against the newly calculated MAC.

For a descriptor that generates an XCBC-MAC over a full message, set AUX0 = 0, AUX1 = 0, AUX2 = 0. The descriptor's key length is loaded into the AESU Key Size Register, and the key itself is loaded into the AESU Key Registers 1U and 1L. The generated MAC is held in AESU context registers 1–2 and output according to the ICV Out length and pointer in the descriptor.

For a descriptor that generates an XCBC-MAC over a full message and compares the calculated MAC with the MAC received with the message, set AUX0 = 0, AUX1 = 0, AUX2 = 1. The descriptor's key length is loaded into the AESU key size register, and the key itself is loaded into the AESU key registers 1U and 1L. The generated MAC is held in AESU context registers 1–2 and compared to the received MAC, which the channel loads into AESU context registers 3–4 using the Extent field in the descriptor. Success or failure of the MAC comparison can be reported by an interrupt or through the ICCR1 bits in the modified descriptor header if header writeback is enabled.

Sometimes a message cannot be processed in a single descriptor. All data may not be present or the message may be larger than a single XCBC-MAC descriptor can process (> 64KB-1). In either case, this situation can be handled through combinations of AUX mode bits.

For the first descriptor, which processes the initial portion of the message, set AUX0 = 1, AUX1 = 0, AUX2 = 0. The descriptor's key length is loaded into the AESU Key Size Register, and the key itself is loaded into the AESU key registers 1U and 1L. The AESU generates K3, K2, K1 and stores them

respectively in AESU context registers 5–6, 7–8, and 9–10. When the first descriptor has consumed all of its message data, it outputs the contents of context registers 1–10 using the Context Out length (80B) and pointer.

There can be an unlimited number of middle descriptors, which are neither the first nor last descriptor. Middle descriptors set  $AUX0 = 1$ ,  $AUX1 = 1$ ,  $AUX2 = 0$ . The descriptor's key length is set to 16B, and the key pointer is set to the address of key 1 (context registers 9–10) from the previous descriptor. This loads key 1 into AESU key size registers 1U and 1L. The descriptor's Context In length is set to 64B, and the pointer is set to the address of context registers 1–8 from the previous descriptor.

When the middle descriptor has consumed all of its message data, it outputs the contents of context registers 1–10 using the Context Out length (80B) and pointer.

For the last descriptor, which processes the final portion of the message, set  $AUX0 = 0$ ,  $AUX1 = 1$ ,  $AUX2 = 0$ . The descriptor's key length is set to 16B, and the key pointer is set to the address of key 1 (context registers 9–10) from the previous descriptor. This loads key 1 into AESU key size registers 1U and 1L. The descriptor's Context In length is set to 64B, and the pointer is set to the address of context registers 1–8 from the previous descriptor.

When the last descriptor has consumed all its message data, the generated MAC is held in AESU context registers 1–2 and output according to the ICV Out length and pointer in the descriptor. To compare the calculated MAC with the MAC received with the message, set  $AUX2 = 1$  on the final descriptor. The generated MAC is held in AESU context registers 1–2 and compared to the received MAC, which is input using the Extent field in the descriptor. The channel overwrites the context reload values in context registers 3–4 with the real received MAC.

### Context and Operation for GCM-GHASH Cipher Mode

GCM-GHASH denotes the authentication part of GCM cipher mode and is described in [Section 10.7.1.11.3, “Context for Confidentiality and Data Integrity Cipher Modes.”](#)

### Context and Operation for CMAC (OMAC1) Cipher Mode

CMAC cipher mode is an authentication-only mode of AES. CMAC may be specified using the following notation:

- $E(K,L)$  denotes the AES-encrypt function;
- $xtime(L)$  is defined as follows, where  $L$  is a 128-bit vector with  $L[127]$  as most significant bit:
  - If  $L[127] = 0$ , then  $xtime(L) = L \ll 1$  (where ‘ $\ll$ ’ denotes bitwise left shift)
  - Else  $xtime(L) = (L \ll 1) \text{ XOR } 0x87$ .

The AESU supports three mode options while operating in CMAC mode. These options are controlled by the AUX bits in the AESU Mode Register, as defined in the list below. The encrypt/decrypt bit has no meaning in a CMAC operation and is ignored by the AESU.

**AUX0** Controls whether the CMAC is completed with this descriptor or whether this descriptor is only doing a portion of the total MAC generation and context needs to be output so that it can be reloaded for subsequent operations.



Section, Page No.	Changes
AUX1	Controls whether K1, K2 are initialized at the start of the descriptor or whether previously initialized keys are reloaded.
AUX2	Controls whether the AESU is to perform automatic checking of a received MAC against the newly calculated MAC.

For a descriptor that generates a CMAC over a full message, set  $AUX0 = 0$ ,  $AUX1 = 0$ ,  $AUX2 = 0$ . The descriptor's key length is loaded into the AESU key size register, and the key itself is loaded into the AESU key registers. The generated MAC is held in AESU context registers 1–2 and output according to the ICV Out length and pointer in the descriptor.

Note that for some uses of CMAC, the message data itself may be modified to include packet-specific context in the CMAC generation process. For instance, when using AES-CMAC for the LTE EIA2 algorithm, a 64-bit, IV-like value is prepended to the PDCP header prior to calculating the MAC. The 64-bit value consists of COUNT (32 bit) || BEARER (5 bit) || Direction (1 bit) || Zeroes (26 bit).

For a descriptor that generates a CMAC over a full message and compares the calculated MAC with the MAC received with the message, set  $AUX0 = 0$ ,  $AUX1 = 0$ ,  $AUX2 = 1$ . The descriptor's key length is loaded into the AESU key size register, and the key itself is loaded into the AESU key registers. The generated CMAC is held in AESU context registers 1–2 and compared to the received CMAC, which the channel loads into AESU context registers 3–4 using the Extent field in the descriptor. Success or failure of the MAC comparison can be reported by an interrupt or through the ICCR1 bits in the modified descriptor header if header writeback is enabled.

Sometimes a message cannot be processed in a single descriptor. All data may not be present or the message may be larger than a single CMAC descriptor can process ( $> 64 \text{ KByte}-1$ ). In either case, this situation can be handled through combinations of AUX mode bits.

For the first descriptor, which processes the initial portion of the message, set  $AUX0 = 1$ ,  $AUX1 = 0$ ,  $AUX2 = 0$ . The descriptor's key length is loaded into the AESU key size register, and the key itself is loaded into the AESU key registers. The AESU internally generates CMAC context and stores it respectively in AESU context registers 5–6. When the first descriptor has consumed all its message data, it outputs the contents of context registers 1–6 using the Context Out length (48B) and pointer.

There can be an unlimited number of middle descriptors, which are neither the first nor last descriptor. Middle descriptors set  $AUX0 = 1$ ,  $AUX1 = 1$ ,  $AUX2 = 0$ . The descriptor's key length is set to 0. The descriptor's Context In length is set to 48B, and the pointer is set to the address of context registers 1–6 from the previous descriptor.

When the middle descriptor has consumed all its message data, it outputs the contents of context registers 1–6, using the Context Out length (48B) and pointer.

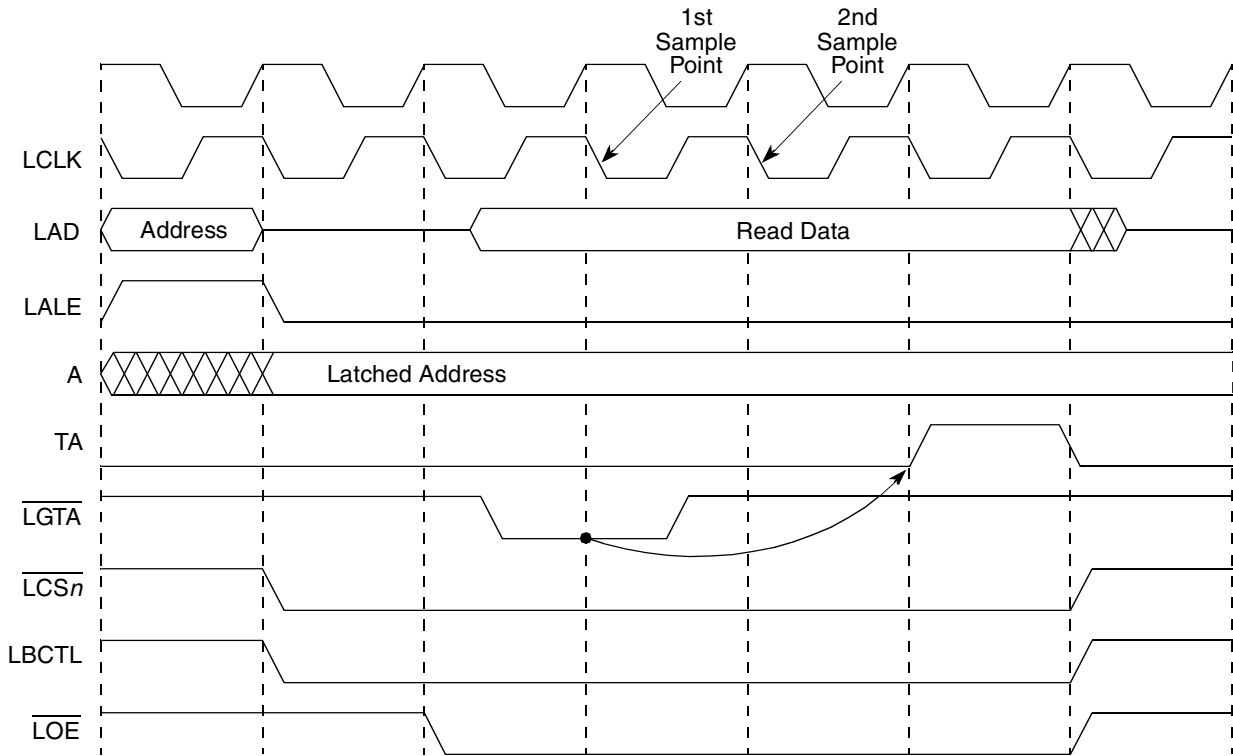
For the last descriptor, which processes the final portion of the message, set  $AUX0 = 0$ ,  $AUX1 = 1$ ,  $AUX2 = 0$ . The descriptor's key length is set to 0. The descriptor's Context In length is set to 48B, and the pointer is set to the address of context registers 1–6 from the previous descriptor.

When the last descriptor has consumed all its message data, the generated CMAC is held in AESU context registers 1–2, and output according to the ICV Out length and pointer in the descriptor. To compare the calculated MAC with the MAC received with the message, set  $AUX2 = 1$  on the final descriptor. The generated CMAC is held in AESU context registers 1–2 and compared to the received CMAC, which is

input using the Extent field in the descriptor. The channel overwrites the context reload values in context registers 3–4 with the real received MAC.

- 11.5.4, 11-23** Remove first sentence from the third paragraph, as follows: “The I<sup>2</sup>C controller automatically generates a STOP if I2CCR[TXAK] is set.”
- 13.1, 13-1 In Figure 13-1, “Enhanced Local Bus Controller Block Diagram,” change LA[7:31] to LA[27:31].
- 13.1.2, 13-2** Added top-level bullet to list of features, as follows:  
Different machines (FCM/GPCM/UPM) share the address, data, and control signals. While the eLBC is servicing a transaction, subsequent transactions are queued until the current transaction has completed.
- 13.2, 13-4 In Table 13-1, “Signal Properties—Summary,” change LA[7:31] to LA[27:31].
- 13.2, 13-7 In Table 13-2, “Enhanced Local Bus Controller Detailed Signal Descriptions,” change LA[7:31] to LA[27:31].
- 13.3.1.1, 13-11 In Figure 13-2, “Base Registers (BR<sub>n</sub>),” change reset value of reserved bit 30 from 1 to 0.
- 13.3.1.12, 13-30**
- 13.3.1.13, 13-31** Added note to both sections, as follows:  
LTEATR may not capture accurate information for errors that occur when an FCM special operation is in progress.
- 13.4, 13-43** Added note to end of section as follows:  
Different machines (FCM/GPCM/UPM) share the address, data, and control signals. While the eLBC is servicing a transaction, subsequent transactions are queued until the current transaction has completed.
- 13.4.1.7, 13-47** Added note to end of section as follows:  
When the FCM is in the middle of a long transaction (such as NAND erase, write, etc.), another transaction on the GPCM or UPM will trigger the bus monitor to start, even though the GPCM or UPM is waiting for the FCM to finish. If the bus monitor times out, it could corrupt the current NAND flash operation as well as terminate the GPCM or UPM operation. To avoid such cases, it is recommended that the bus monitor timeout be programmed to its maximum setting of LBCR[BMT] = 0 and LBCR[BMTPS] = 0xF.
- 13.4.2.4, 13-58** Relabel existing Figure 13-45 as “External Termination of GPCM Access (PLL Enabled Mode).” Add new Figure 13-46 as follows:

In PLL bypass mode, the timing of  $\overline{\text{LGTA}}$  is illustrated by the example in [Figure 10-19](#).



**Figure 10-19. External Termination of GPCM Access (PLL Bypass Mode)**

**13.4.4.4.7, 13-96**

Added note to section as follows:

Multiple-bank DRAM and SDRAM devices require that the bank address be driven during both RAS and CAS cycles. The UPM does not support a persistent bank address on both RAS and CAS cycles. Therefore, external logic must be used to supply a bank address to these devices.

**14.5.3.1.6, 14-36**

In Table 14-11, “ECNTRL Field Descriptions,” update GMIIM (bit 25) field description to read as follows:

GMII interface mode. If this bit is set, a PHY with a GMII interface is expected to be connected. If cleared, a PHY with an MII or RMII or RGMII interface is expected. The user should then set MACCFG2[I/F Mode] accordingly. The state of this status bit is defined during power-on reset. See [Section 4.4.3, “Power-On Reset Configuration.”](#)

- 0 MII or RMII or RGMII mode interface expected
- 1 GMII mode interface expected

In same section, in Table 14-12, “eTSEC Interface Configurations,” correct GMIIM column to be 1 for GMII only and 0 for all other modes.

**14.5.3.6.17, 14-96**

In Table 14-75, “RFLR Field Descriptions,” add the following text to the RFLR (bits 16–31) field description:

“Frames tagged with a single VLAN tag are checked for valid length based on bytes 17–18 (rather than 13–14). Frames tagged (stacked) with multiple VLAN tags are not checked for valid length.”

14.5.3.11.1, 14-122

Change TMR\_CTRL bits 20 and 21 to be reserved (were ESFDP and ESFDE).

14.6.2, 14-156

Replace first bullet under “The following restrictions apply in any of the FIFO modes” with the following:

- Transferred packets must be no more than 9600 bytes in length.
- If RCTRL[PRSFM]=0, received packets must be a minimum of 10 bytes. If RCTRL[PRSFM]=1, received packets must be a minimum of 14 bytes.
- Transmitted packets with L2 headers must be a minimum of 14 bytes. Transmitted packets without L2 headers must be a minimum of 10 bytes.

14.6.5.2.3, 14-186

Add RQFCR[GPI] column to Table 14-161, “Special Filer Rules,” as follows:

**Table 14-161. Special Filer Rules**

Rule Description	RQCTRL Fields							RQPROP Word	RQCTRL Word <sup>1</sup>
	GPI	CLE	REJ	AND	Q	CMP	PID		
Default file—Always file frame to ring Q	0	0	0	0	Q	01	0000	All zeros	0x0000_qq20
Default reject—Always discard frame	0	0	1	0	000_000	01	0000	All zeros	0x0000_0120
Empty rule in AND—Always matches	0	0/1 <sup>2</sup>	0	1	000_000	01	0000	0xFFFF_FFFF	0x0000_00A0
Empty rule in rule set—Always fails	0	0/1 <sup>3</sup>	0	0	000_000	11	0000	0xFFFF_FFFF	0x0000_0060

<sup>1</sup> Hexadecimal digits *qq* denotes field Q shifted left 2 bits.

<sup>2</sup> Set CLE = 1 if the empty rule guards a cluster.

<sup>3</sup> Set CLE = 1 if the empty rule occurs at the end of a cluster.

14.6.5.2.6, 14-187

Add RQCTRL[GPI] column to Table 14-163, “Filer Table Example—802.1p Priority Filing,” as follows:

**Table 14-163. Filer Table Example—802.1p Priority Filing**

Table Entry	RQCTRL Fields							RQPROP	Comment	RQCTRL Word
	GPI	CLE	REJ	AND	Q	CMP	PID			
0	0	0	0	0	000_000	00	1001	0x0000_0007	File priority 7 to ring 0	0x0000_0009
1	0	0	0	0	000_001	00	1001	0x0000_0006	File priority 6 to ring 1	0x0000_0409
2	0	0	0	0	000_010	00	1001	0x0000_0005	File priority 5 to ring 2	0x0000_0809
3	0	0	0	0	000_011	00	1001	0x0000_0004	File priority 4 to ring 3	0x0000_0C09
4	0	0	0	0	000_100	00	1001	0x0000_0003	File priority 3 to ring 4	0x0000_1009
5	0	0	0	0	000_101	00	1001	0x0000_0002	File priority 2 to ring 5	0x0000_1409
6	0	0	0	0	000_110	00	1001	0x0000_0001	File priority 1 to ring 6	0x0000_1809
7	0	0	0	0	000_111	00	1001	All zeros	File undefined 802.1p or priority 0 to ring 7—Default always matches	0x0000_1C09

**14.6.5.2.7, 14-188** Add RQCTRL[GPI] column to Table 14-164, “Filer Table Example—IP Diff-Serv Code Points Filing,” as follows:

**Table 14-164. Filer Table Example—IP Diff-Serv Code Points Filing**

Table Entry	RQCTRL Fields							RQPROP	Comment	RQCTRL Word
	GPI	CLE	REJ	AND	Q	CMP	PID			
0	0	0	0	0	001_000	01	1010	0x0000_00E0	File class 7 to queue 8 (TOS >= 0xE0)	0x0000_202A
1	0	0	0	0	001_001	01	1010	0x0000_00C0	File class 6 to queue 9 (TOS >= 0xC0)	0x0000_242A
2	0	0	0	0	001_010	01	1010	0x0000_00A0	File class 5 to queue 10 (TOS >= 0xA0)	0x0000_282A
3	0	0	0	0	001_011	01	1010	0x0000_0080	File class 4 to queue 11 (TOS >= 0x80)	0x0000_2C2A
4	0	0	0	0	000_100	01	1010	0x0000_0060	File class 3 to queue 4 (TOS >= 0x60)	0x0000_102A
5	0	0	0	0	001_100	01	1010	0x0000_0040	File class 2 to queue 12 (TOS >= 0x40)	0x0000_302A
6	0	0	0	0	010_100	01	1010	0x0000_0020	File class 1 to queue 20 (TOS >= 0x20)	0x0000_502A
7	0	0	0	0	011_100	01	1010	All zeros	File class 0 to queue 28 (TOS >= 0x00) or file to ring 4 by default	0x0000_702A

**14.6.5.2.8, 14-189** Add RQCTRL[GPI] column to Table 14-165, “Filer Table Example—TCP and UDP Port Filing,” as follows:

**Table 14-165. Filer Table Example—TCP and UDP Port Filing**

Table Entry	RQCTRL Fields							RQPROP	Comment	RQCTRL Word
	GPI	CLE	REJ	AND	Q	CMP	PID			
0	0	1	0	1	000_000	00	1011	0x0000_0006	Enter cluster if layer 4 is TCP	0x0000_028B
1	0	0	0	1	000_000	01	1111	0x0000_0014	AND rule—FTP from TCP ports 20 and 21: file to ring 2	0x0000_00AF
2	0	0	0	0	000_010	11	1111	0x0000_0016		0x0000_086F
3	0	0	0	0	000_011	00	1111	0x0000_0017	telnet from TCP port 23: file to ring 3	0x0000_0C0F
4	0	0	0	0	000_000	00	1111	All zeros	<i>empty entry reserved for future use</i>	0x0000_000F
5	0	0	0	0	000_000	00	1111	All zeros	<i>empty entry reserved for future use</i>	0x0000_000F
6	0	1	0	0	000_001	01	0000	All zeros	end cluster; default TCP: file to ring 1	0x0000_0620
7	0	1	0	1	000_000	00	1011	0x0000_0011	Enter cluster if layer 4 is UDP	0x0000_028B
8	0	0	0	0	000_101	00	1111	0x0000_0801	NFS from UDP port 2049	0x0000_140F
9	0	0	0	0	000_111	00	1111	0x0000_0208	Route from UDP port 520	0x0000_000F
10	0	0	0	0	000_110	00	1111	0x0000_0045	TFTP from UDP port 69	0x0000_180F
11	0	1	0	0	000_100	01	0000	All zeros	End cluster; default UDP: file to ring 4	0x0000_1220
12	0	0	0	0	000_000	01	0000	All zeros	By default, file to ring 0	0x0000_0020

**14.6.5.2.9, 14-189** Add new section, “Filer Example—Interrupt from Deep Sleep,” as follows:

### 14.6.5.2.9 Filer Example—Interrupt from Deep Sleep

The example in [Table 14-166](#) shows how the filer can facilitate exit from deep sleep if any of the following packets arrive:

- ARP packet with Target IP address matching either of two IP addresses (either static or link local address)
- IPv4/UDP multicast DNS query
- IPv4/UDP SNMP broadcast query

These packets are also be stored in memory. All other packets are dropped.

**Table 14-166. Filer Example—Interrupt from Deep Sleep**

Table Entry	RQCTRL Fields							RQPROP	Comment
	GPI	CLE	REJ	AND	Q	CMP	PID		
0	0	0	0	0	000_000	11	0000	0x0001_0000	Check for ARP request; set mask register to mask off everything but the ARP request flag.
1	0	1	0	1	000_000	00	0001	0x0001_0000	Check to see if ARP request flag is set by doing a =1 comparison. Enter the "ARP Request Cluster" if true.
2	0	0	0	0	000_000	11	0000	0xFFFF_FFFF	ARP Cluster: Set Mask to unmask everything (Reset mask to all F's)
3	1	0	0	0	000_001	00	1100	0XXXXX_XXXX	Compare the ARP Target IP address against "MY_IP_1", which is indicated by the user-defined value of 0xXXXX_XXXX; if they match, accept the frame and generate an interrupt.
4	1	0	0	0	000_001	00	1100	0YYYYY_YYYY	Compare the ARP Target IP address against "MY_IP_2", which is indicated by the user-defined value of 0xYYYY_YYYY; if they match, accept the frame.
5	0	1	1	0	000_000	01	0000	0x0000_0000	Default rule that will always discard the packet; inserted here because an ARP request was received, but the Target IP address did not match either local IP addresses; therefore drop the packet and exit the cluster.
6	0	0	0	0	000_000	11	0000	0x0000_02D0	Set Mask for IP4 Packet (2D0), with IPv4 checksum checked and verified, and UDP header located.
7	0	0	0	1	000_000	00	0001	0x0000_02D0	Check to see if IP4 Packet, with IPv4 checksum checked and verified, and UDP header.
8	0	0	0	1	000_000	00	0000	0xFFFF_FFFF	Set Mask to unmask everything (Reset mask to all F's).
9	0	1	0	1	000_000	00	1011	0x0000_0011	Check against UDP protocol; if this passes, enter the cluster - all packets in the cluster are IPv4 packets with UDP protocol identified as the L4 protocol type.
10	0	0	0	1	000_000	00	0011	0x00XX_XXXX	Compare upper L2 DA bits to XX_XXXX (for multicast DNS query)

**Table 14-166. Filer Example—Interrupt from Deep Sleep (continued)**

Table Entry	RQCTRL Fields							RQPROP	Comment
	GPI	CLE	REJ	AND	Q	CMP	PID		
11	0	0	0	1	000_000	00	0100	0x00YY_YYYY	Compare lower L2 DA bits to YY_YYYY
12	0	0	0	1	000_000	00	1100	0xZZZZ_ZZZZ	Compare L3 Destination IP address to ZZZZ_ZZZZ
13	0	0	0	1	000_000	00	1110	0x0000_XXXX	Compare L4 destination port to XXXX
14	1	0	0	0	000_001	00	1111	0x0000_YYYY	If all of the previously consecutive ANDed conditions pass, multicast DNS Query has matched.
15	0	0	0	1	000_000	00	0011	0x00XX_XXXX	Compare upper L2 DA bits to XX_XXXX (for SNMP broadcast query).
16	0	0	0	1	000_000	00	0100	0x00YY_YYYY	Compare lower L2 DA bits to YY_YYYY.
17	1	0	0	0	000_001	00	1110	0x0000_ZZZZ	If all of the previously consecutive ANDed conditions pass, SNMP broadcast Query has matched.
18	0	1	1	0	000_000	01	0000	0x0000_0000	Cluster End: IPv4, UDP Comparison Default rule that will always discard the packet; inserted here because an IPv4 packet with L4=UDP request was received, but the profiles didn't match anything "interesting"; therefore drop the packet and exit the cluster.
19	0	0	1	0	000_000	01	0000	0x0000_0000	Default rule that will always discard the packet; inserted here no matches for any "interesting" packets were received that are used to wake up the CPU. All packets that reach this rule are discarded.

**14.7.1.1, 14-210** In Table 14-173, “MII Mode Register Initialization Steps,” in rows for “Setup the MII Mgmt clock speed,” change last sentence to say that minimum frequency is 2.5 MHz.

**14.7.1.5, 14-225** In Table 14-185, “RMII Mode Register Initialization Steps,” replace third step with the following:

Initialize MACCFG2,  
MACCFG2[0000\_0000\_0000\_0000\_0111\_0001\_0000\_0101]  
(I/F Mode = 1, Full Duplex = 1)

15.4.1.1.4, 15-27 Revise first sentence of Step 1 to say, “Set MRn[CDSM/SWSM] and MRn[XFE] and clear MRn[CTM] to indicate extended chaining and single-write start mode.”

16.3.1, 16-14 In Table 16-3, “PCI Memory-Mapped Register Map,” change address range of next-to-last row of reserved registers to 0xE24–0xEFC.

16.3.2.2, 16-31 In Table 16-25. “PCI Device ID Register Field Description,” add PCI device IDs for MPC8535E, as follows:  
MPC8535E: 0x0052  
MPC8535: 0x0053



Section, Page No.	Changes
17.3.7.1.1, 17-44	In the last bullet, add the statement that, “In this case [no bus number matches], PEX_ERR_DR[ICCA] is set.”
17.3.8.1.2, 17-46	In Table 17-38, “PCI Express Device ID Register Field Description,” add PCI Express device IDs for MPC8535E, as follows: MPC8535E: 0x0052 MPC8535: 0x0053
Chapter 20	Remove references to “read wait,” “suspend,” “resume,” “read-wait control,” and “interrupt detect.”
20.4.8, 20-17	In Figure 20-10, “Protocol Control Register (PROCTL),” and Table 20-12, “PROCTL Field Descriptions,” make WECINT (bit 7) reserved.
20.4.10, 20-23	In Figure 20-12, “Interrupt Status Register (IRQSTAT),” and Table 20-14, “IRQSTAT Field Descriptions,” make CINT (bit 23) reserved.
20.4.11, 20-27	In Figure 20-13, “Interrupt Status Enable Register (IRQSTATEN),” and Table 20-18, “IRQSTATEN Field Descriptions,” make CINTSEN (bit 23) reserved.
20.4.12, 20-29	In Figure 20-14, “Interrupt Signal Enable Register (IRQSIGEN),” and Table 20-19, “IRQSIGEN Field Descriptions” make bit CINTIEN (bit 23) reserved.
20.4.14, 20-33	In Figure 20-16, “Host Capabilities Register (HOSTCAPBLT),” and Table 20-22, “HOSTCAPBLT Field Descriptions,” make SRS (bit 8) reserved.
20.4.16, 20-35	In Figure 20-18, “Force Event Register (FEVT),” and Table 20-24, “FEVT Field Descriptions,” make FEVTCINT (bit 0) reserved.
20.6.2.2, 20-46	Replace second sentence of second paragraph with the following: “See <a href="#">Figure 20-26</a> for the software flow chart.”  In the same section, modify the pseudocode and add an introductory sentence, as follows:  “The following pseudocode shows how to initialize the eSDHC host controller and the memory card.
	<pre>software_reset() {     Configure the I/O muxes to select SD signals;     set_bit(SYSCTL, RSTA);           // software reset the host     set SYSCTL[DTOCV and SDCLKFS]; // get the SDHC_CLK of frequency around 400 kHz     poll PRSSTAT[CIHB and CDIHB];  // wait until both bits are cleared     set_bit(SYSCTRL, INTIA);        // send 80 clock ticks for card to power-up     If the card is SD/MMC         send_command(CMD_GO_IDLE_STATE, &lt;other parameters&gt;); // reset the card with CMD0     If the card is SDIO         send_command(CMD_IO_RW_DIRECT, &lt;other parameters&gt;); //reset the card with CMD52 }</pre>
20.6.5, 20-59	Add the following rows to Table 20-27, “Commands for MMC/SD.” Note that only the new rows are shown:



**Table 20-27. Commands for MMC/SD**

CMD INDEX	Type	Argument	Resp	Abbreviation	Description <sup>1</sup>
ACMD18	adtc	[31:0] stuff bits	R1	SECURE_READ_MULTI_BLOCK	Protected Area Access Command: Reads continuously transfer data blocks from Protected Area of SD Memory Card. Refer to Security Specification Version 2.00 for more details.
ACMD25	adtc	[31:0] stuff bits	R1	SECURE_WRITE_MULTI_BLOCK	Protected Area Access Command: Writes continuously transfer data blocks to Protected Area of SD memory card. Refer to Security Specification Version 2.00 for more details.
ACMD26	adtc	[31:0] stuff bits	R1	SECURE_WRITE_MKB	System Area Access Command: Overwrite the existing Media Key Block (MKB) on System Area of SD Memory Card with new MKB. This command is used in dynamic update MKB scheme. Refer to Security Specification Version 2.00 for more details.
ACMD38	ac	[31:0] stuff bits	R1b	SECURE_ERASE	Protected Area Access Command: Erase a specified region of the Protected Area of SD Memory Card. Refer to Security Specification Version 2.00 for more details.
ACMD43	adtc	[31:24]Unit_Count: [23:16] MKB_ID: [15:0]Unit_Offset:	R1	GET_MKB	Reads Media Key Block from System Area of SD Memory Card. <ul style="list-style-type: none"> <li>Unit_Count specifies the Number of units to read. (Here, a unit=512 bytes (fixed).)</li> <li>MKB_ID specifies the application unique number.</li> <li>Unit_Offset specifies the start address(offset) to read.</li> </ul> Refer to Security Specification Version 2.00 for more details.
ACMD44	adtc	[31:0] stuff bits	R1	GET_MID	Reads Media ID from the System Area of SD Memory Card. Refer to Security Specification Version 2.00 for more details.
ACMD45	adtc	[31:0] stuff bits	R1	SET_CER_RN1	AKE Command: Writes random number RN1 as challenge1 in AKE process. Refer to Security Specification Version 2.00 for more details.
ACMD46	adtc	[31:0] stuff bits	R1	GET_CER_RN2	AKE Command: Reads random number RN2 as challenge2 in AKE process. Refer to Security Specification Version 2.00 for more details.

**Table 20-27. Commands for MMC/SD (continued)**

CMD INDEX	Type	Argument	Resp	Abbreviation	Description <sup>1</sup>
ACMD47	adtc	[31:0] stuff bits	R1	SET_CER_RES2	AKE Command: Writes RES2 as response2 to RN2 in AKE process. Refer to Security Specification Version 2.00 for more details.
ACMD48	adtc	[31:0] stuff bits	R1	GET_CER_RES1	AKE Command: Reads RES1 as response1 to RN1 in AKE process. Refer to Security Specification Version 2.00 for more details.
ACMD49	ac	[31:0] stuff bits	R1b	CHANGE_SECURE_AREA	Protected Area Access Command: Change size of Protected Area. Refer to Security Specification Version 2.00 for more details.

<sup>1</sup> Registers mentioned in this table are SD card registers.

22.3, 22-2                      In Table 22-2, “GPIO Register Address Map,” add header row as follows:

<b>General Purpose I/O (GPIO)—Block Base Address 0x0_F000</b>
---------------------------------------------------------------

23.4.1.4, 23-11                In Table 23-7, “PORDEVSR Field Descriptions” update values for PCI\_SPD (bit 16) field description to the following:

- 0    PCI set for low-speed operation—PCI at or below 33 MHz
- 1    PCI set for normal-speed operation—PCI above 33 MHz

23.4.1.21, 23-30              In Table 23-24, “SVR Field Descriptions,” add SVR values for MPC8535E, as follows:

- “0x803F\_0191 for MPC8535E Rev 1.1 (with security)
- 0x8037\_0191 for MPC8535 Rev 1.1 (without security)”

23.4.1.30, 23-39              In Figure 23-30, “SerDes2 Control Register 0 (SRDS2CR0),” and Table 23-33, “SRDS2CR0 Field Descriptions,” add bit fields LSTSA (bit 2) and LSTSE (bit 6), as follows. Note that only the new rows are shown:

**Table 23-33. SRDS2CR0 Field Descriptions**

Bits	Name	Description
2	LSTSA	Loss of Signal Threshold Level Selection for lane A of SerDes 2 in SGMII mode. This bit, when set, increases the SerDes 2 lane A receiver's Vrx_diffp-p level on electrical idle detection. Refer to the Vlos parameter (Loss of signal threshold) in the DC Receiver Electrical Characteristics of section "SGMII DC Electrical Characteristics" in <i>MPC8536E PowerQUICC III Integrated Processor Hardware Specifications</i> for the details of how the LSTS bit affects the loss of signal threshold level. Recommended setting per protocol: <ul style="list-style-type: none"> <li>• SGMII: 0</li> <li>• SATA: 0</li> </ul>
6	LSTSE	Loss of Signal Threshold Level Selection for lane E of SerDes 2 in SGMII mode. This bit, when set, increases the SerDes 2 lane E receiver's Vrx_diffp-p level on electrical idle detection. Refer to the Vlos parameter (Loss of signal threshold) in the DC Receiver Electrical Characteristics of section "SGMII DC Electrical Characteristics" in <i>MPC8536E PowerQUICC III Integrated Processor Hardware Specifications</i> for the details of how the LSTS bit affects the loss of signal threshold level. Recommended setting per protocol: <ul style="list-style-type: none"> <li>• SGMII: 0</li> <li>• SATA: 0</li> </ul>

23.5.1.16.1, 23-6      Replace first sentence of paragraph with the following:  
POWER\_OK is an external input indication that VDD is within the specified levels. Its assertion is used when waking from deep sleep to indicate that a previously switched off VDD has returned to the specified levels after a wakeup event has occurred.

23.5.1.16.2, 23-61      Replace second paragraph with the following:  
This signal is asserted during the power-on reset sequence, prior to  $\overline{\text{HRESET}}$  deassertion. The signal deasserts when entering deep sleep. When the system is in deep sleep mode and a wake-up event is accepted, the signal reasserts. It will only reassert after it has been low for at least a minimum of the PMCCR[PDCNT] counter value.

A.1.2, A-1      In Table A-2, "Local Access Window Registers," add the following rows at the beginning of the table:

Local Memory Offset (Hex)	Register	Access	Reset	Section/Page
0x0_0BF8	LAIPBRR1—Local access IP block revision register 1	R	All zeros	<a href="#">2.2.3.2/2-6</a>
0x0_0BFC	LAIPBRR2—Local access IP block revision register 2	R	All zeros	<a href="#">2.2.3.3/2-6</a>

A.1.5, A-5      In Table A-5, "I<sup>2</sup>C Controller 1 & 2 Registers," change offset for I2C controller 2 to the same as that for I2C controller 1 (0x0\_3000).

A.1.9, A-11      In Table A-9, "PCI Controller Registers," change address range of next-to-last row of reserved registers to 0xE24–0xEFC.

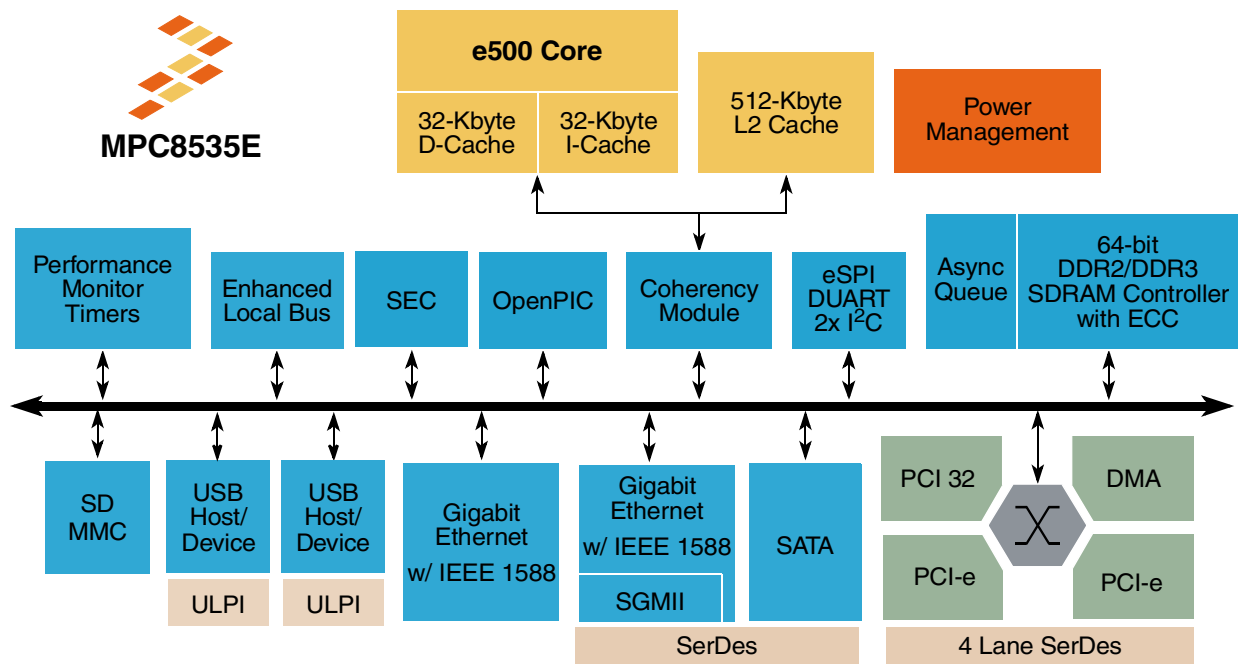
A.1.11, A-15      In Table A-11, "GPIO Register Address Map," add header row as follows:

<b>General Purpose I/O (GPIO)—Block Base Address 0x0_F000</b>
---------------------------------------------------------------

**Section, Page No.**

**Changes**

- A.1.14, A-18      Insert Table A-14, “DMA Register Summary.” Table A-14 is identical to Table 15-4, “DMA Register Summary,” on page 15-6.
- A.1.16, A-20      In Table A-16, “eTSEC Registers,” remove extra digits from column eTSEC1 Offset as follows:  
                          Replace 0x2\_4000 with 0x000,  
                          Replace 0x2\_4004 with 0x004,  
                          ... and so on.
- A.3.1, A-50        In Table A-17, “Global Utilities Registers,” remove extra digits from column Offset as follows:  
                          Replace 0xE\_3000 with 0x000  
                          Replace 0xE\_3008 with 0x008  
                          ... and so on (applies to SerDes registers only).
- C.1, C-1            Replace Figure C-1, “MPC8535E Block Diagram,” with the following:



**Figure C-1. MPC8535E Block Diagram**

**THIS PAGE INTENTIONALLY LEFT BLANK**

## How to Reach Us:

### Home Page:

[www.freescale.com](http://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](http://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](http://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](mailto: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](mailto: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](mailto: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.

Freescale are trademarks or registered trademarks of Freescale Semiconductor, Inc. in the U.S. and other countries. All other product or service names are the property of their respective owners. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org. IEEE 1588, 802.3, and 802.1 are registered trademarks of the Institute of Electrical and Electronics Engineers, Inc. (IEEE). This product is not endorsed or approved by the IEEE.

© Freescale Semiconductor, Inc., 2010. All rights reserved.

