Freescale Semiconductor
Addendum

Document Number: MPC7450UMAD
Rev. 5.3, 07/2013

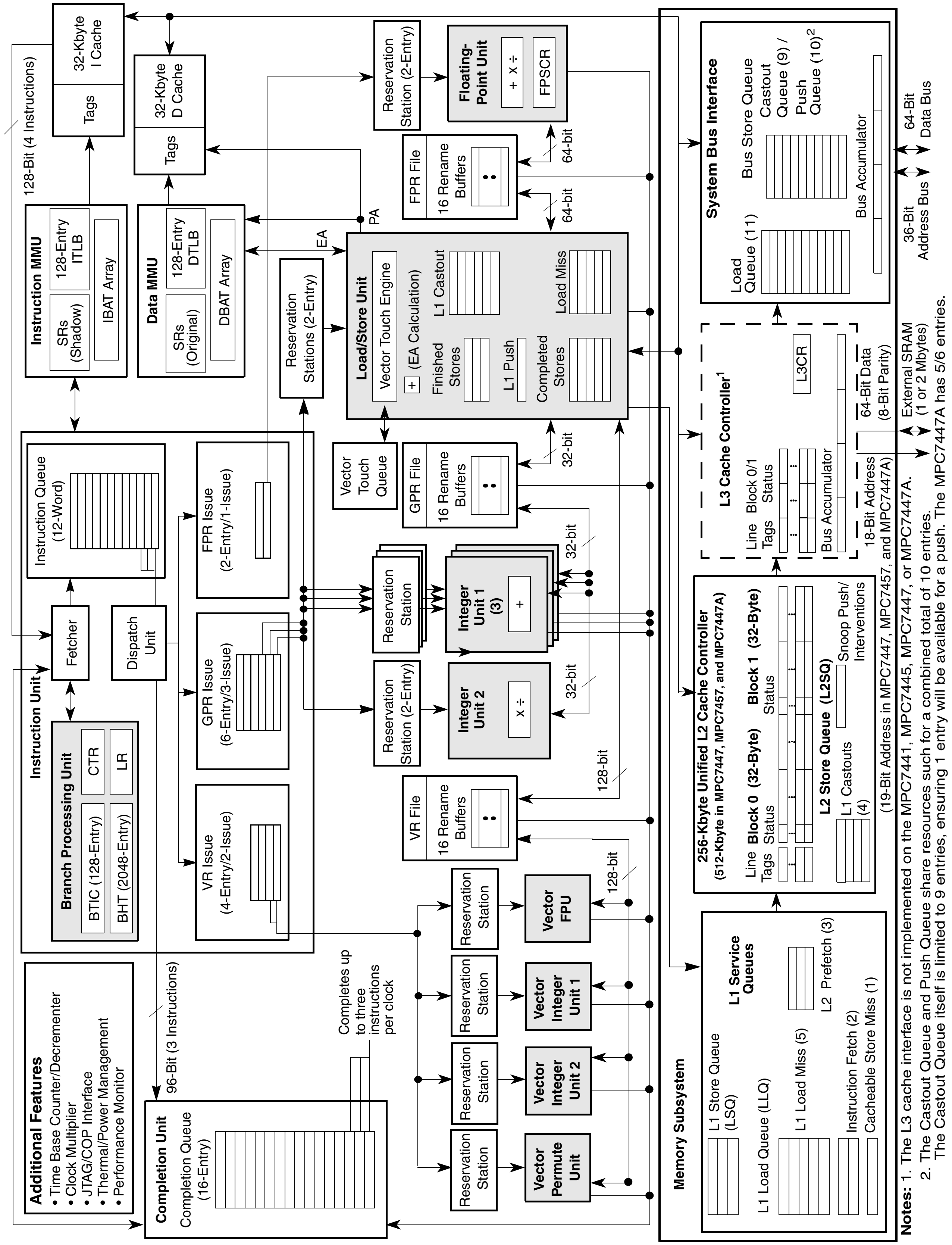
# Errata to the MPC7450 RISC Microprocessor Family Reference Manual, Rev. 5

This errata describes corrections to the *MPC7450 RISC Microprocessor Family Reference Manual*, Rev. 5. The MPC7450 is a PowerPC® microprocessor built on Power Architecture® technology. For convenience, the section number and page number of the errata item in the reference manual are provided in the left margin. Items in bold are new since the last revision of this document.

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

| Section, Page | Changes |
|---|---|
| 1.1, 1-4 | The LSU, IU2, and FPU each have 2-entry reservation stations; all other execution units have 1-entry reservation stations. Clarify Figure 1-1, “MPC7450 Microprocessor Block Diagram,” to show this consistently, as follows: |

Section, Page No. Changes



Notes: 1. The L3 cache interface is not implemented on the MPC7441, MPC7445, MPC7447, or MPC7447A.
2. The Castout Queue and Push Queue share resources such for a combined total of 10 entries. The Castout Queue itself is limited to 9 entries, ensuring 1 entry will be available for a push. The MPC7447A has 5/6 entries.

1.2.7, 1-23

Third paragraph, replace with the following:

The bus interface protocol is configured using the $\overline{\mathrm{BMODE0}}$ configuration signal at reset. If $\overline{\mathrm{BMODE0}}$ is sampled asserted just prior to the negation of $\overline{\mathrm{HRESET}}$, the MPC7450 uses the MPX bus protocol; if $\overline{\mathrm{BMODE0}}$ is sampled negated just prior to the negation of $\overline{\mathrm{HRESET}}$, the MPC7450 uses a limited subset of the 60x bus protocol. Note that the inverse state of $\overline{\mathrm{BMODE}}$[0:1] at the negation of $\overline{\mathrm{HRESET}}$ is saved in MSSCR0[BMODE].

**2.2.3.4, 2-13**

In Figure 2-4, add 0 to the 6-bit reserved field in bits 0-5 of the MSR.

2.2.5.1, 2-21

Table 2-7, replace bits 22 and 23 with the following:

| | | |
|---|---|---|
| 22 | SPD[1] | Speculative data cache and instruction cache access disable<br>0 Speculative bus accesses to nonguarded space (G = 0) from both the instruction and data caches is enabled.<br>1 Speculative bus accesses to nonguarded space in both caches is disabled.<br>Thus, setting this bit prevents L1 data cache misses from going to the memory subsystem until the instruction that caused the miss is next to complete. The HID0[SPD] bit also prevents instruction cache misses from going to the memory subsystem until there are no unresolved branches. For more information on this bit and its effect on re-ordering of loads and stores, see Section 3.3.1.2, “Out-of-Order Accesses to Guarded Memory” and Section 3.3.3.3, “Load Ordering with Respect to Other Loads.” |
| 23 | —[8] | Reserved on the MPC7450, MPC7441, and MPC7451. |
| | XBSEN[9] | Extended BAT block size enable for the MPC7445, MPC7455, MPC7447, MPC7457, MPC7447A, and MPC7448<br>0 Causes IBATnU[XBL] and DBATnU[XBL] to be cleared when the BAT register is written by an **mtspr** instruction.<br>1 Allows software to write IBATnU[XBL] and DBATnU[XBL], allowing extended BAT block sizes of 512 MB, 1 GB, 2 GB, and 4 GB.<br>If the extended block size feature is to be used, software must set this bit before configuring the IBAT and DBAT registers. Note that clearing this bit does not automatically clear the IBATnU[XBL] and DBATnU[XBL] bits, and the XBL fields will still be used for address translation. To disable the extended block size feature, software must clear the IBATnU[XBL] and DBATnU[XBL] bits by writing zero to these bits, or by clearing the XBSEN bit then writing the IBATnU and DBATnU registers to cause the XBL fields to be cleared. If backwards compatibility with previous processors is a concern, then HID0[XBSEN] should stay cleared so that the XBL bits are always zero, which allows the BAT translation to have a maximum block length of 256 MB. |

2.2.5.2, 2-26

First paragraph following note, remove the second sentence referencing Section 8.4.6.3.

**2.2.5.3, 2-27**

Make the following changes in Table 2-10, “MSSCR0 Field Descriptions”:

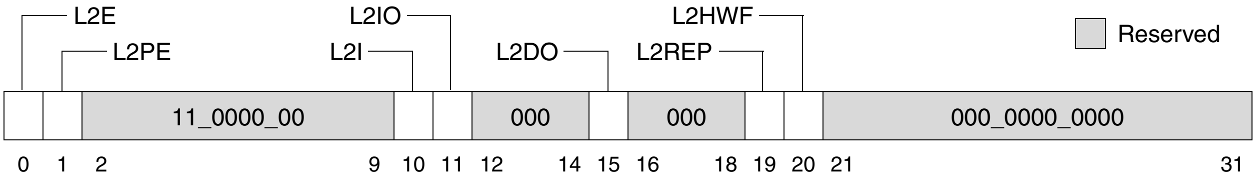
- In the bit 11 field description, update the first sentence to read, “The read-only bit reflects the inverted state of the $\overline{\mathrm{BMODE0}}$ signal after $\overline{\mathrm{HRSET}}$ negation and indicates whether the processor is address bus driven mode.”
- In the bits 16–17 field description, replace “while $\overline{\mathrm{HRESET}}$ is asserted” with “just prior to $\overline{\mathrm{HRESET}}$ negation” in the second sentence.
- Also in the bits 16–17 field description, modify the BMODE[0:1] → MSSCR0 list as shown.

| Section, Page No. | Changes |
|---|---|
| | •In the bit 26 field description, update the third sentence to read, “Determined by the inverse of the voltage levels on $\overline{\text{BMODE1}}$ after $\overline{\text{HRESET}}$ is negated.”<br>Affected rows are shown as follows. |

**Table 2-10. MSSCR0 Field Descriptions**

| Bits | Name | Function |
|---|---|---|
| 11 | ABD | Address bus driven mode<br>0 Address bus driven mode disabled<br>1 Address bus driven mode enabled<br>The read-only bit reflects the state of the inverted $\overline{\text{BMODE0}}$ signal after $\overline{\text{HRESET}}$ negation and indicates whether the processor is in address bus driven mode. See Section 9.3.2.1, “Address Bus Driven Mode,” for more information. |
| 16–17 | BMODE | Bus mode (read-only). Reflects the inverse of the voltage levels on $\overline{\text{BMODE}}$[0:1] just prior to $\overline{\text{HRESET}}$ negation. Indicates whether the system interface uses the 60x or MPX bus protocol as described in Chapter 9, “System Interface Operation.”<br>00 60x bus mode<br>01 Reserved<br>10 MPX bus mode<br>11 Reserved<br>Note that the value on $\overline{\text{BMODE}}$[0:1] after reset negation determines other values of MSSCR0 as follows:<br>$\overline{\text{BMODE0}}$ (post reset) → MSSCR0[ABD] (field value is inverted from signal)<br>$\overline{\text{BMODE1}}$ (post reset) → MSSCR0[ID] (field value is inverted from signal) |
| 26 | ID | Processor identification. Sets the processor ID to either processor 0 or 1. Determined by the inverse of the voltage levels on $\overline{\text{BMODE1}}$ while $\overline{\text{HRESET}}$ is negated.<br>0 $\overline{\text{BMODE1}}$ negated after $\overline{\text{HRESET}}$ negated<br>1 $\overline{\text{BMODE1}}$ asserted after $\overline{\text{HRESET}}$ negated<br>In a multiprocessor system, one processor can be assigned by the $\overline{\text{BMODE1}}$ as processor 0 and all other processor can be assigned as processor 1. Then software can find processor 0 and use it to re-identify the other processors by writing unique values to the PIR of the other CPUs. |

| | |
|---|---|
| 2.2.5.5.1, 2-31 | In Figure 2-16, correct the reset value of bits 4–5 to be 00 instead of 11 for MPC7450–MPC7447A. |
| **2.2.5.5.1, 2-31** | Replace Figure 2-17, “L2 Control Register (L2CR) for the MPC7448” with the following updated figure: |



| | |
|---|---|
| **2.2.5.5.1, 2-32** | In Table 2-12, remove rows for bits 24–31 and row for bits 21–23 that says “Reserved on the MPC7448.” |

**Section, Page No.** **Changes**

2.2.5.5.11, 2-39 In Table 2-22, replace the bit description for bits 5–7 with the following:

| | | |
|---|---|---|
| 5–7 | TRANSSIZ | Transaction size for detected error (read only). Reserved for read transactions (TRANSTYPE = 10). The transaction size for a read to the L2 will always be a 32-byte burst.<br>000 8 bytes (single-beat) or reserved (burst)<br>001 1 byte (single-beat) or 16 bytes (burst)<br>010 2 bytes (single-beat) or 32 bytes (burst)<br>011 3 bytes (single beat) or reserved (burst)<br>100 4 bytes (single-beat) or reserved (burst)<br>101 5 bytes (single-beat) or reserved (burst)<br>110 6 bytes (single-beat) or reserved (burst)<br>111 7 bytes (single-beat) or reserved (burst) |

2.2.5.5.14, 2-40 Table 2-25, bit description for bits 8–15, remove '(read only)' from bit field description. This field is writable.

2.2.5.7.2, 2-55 Figure 2-43, add the following footnote to bit 23:

See the bit description for bit 23 in Table 2-38. This bit is hardwired to 0 on earlier processors and to 1 on later processors.

2.2.5.7.2, 2-56 Table 2-38, register PTELO, bit 23, replace with the following:

| | | | |
|---|---|---|---|
| | 23 | — | Reserved. Corresponds to the referenced bit in a PTE. The referenced bit is not stored in the page tables, so this bit is ignored in the PTELO register. To simplify software, this bit is hardwired to 1 in the MPC7447, MPC7447A, and MPC7448 and to 0 in the MPC7441, MPC7450, MPC7451, MPC7455, and MPC7457. All the other bits in PTELO correspond to the bits in the low word of the PTE. |

2.2.5.7.2, 2-56 First paragraph following Table 2-38, replace with the following:

When extended addressing is not enabled, (HID0[XAEN] = 0), the software must clear the PTELO[XPN] and PTELO[X] bits; otherwise whatever values are in the fields become the four most-significant bits of the physical address. **Note:** The PTEHI register is accessed with **mtspr** and **mfspr** as SPR 981 and PTELO is accessed as SPR 982.

3.3.1, 3-14 Add the following sentence to the end of the second paragraph:

Note that attempting to fetch instructions from guarded memory when MSR[IR] = 1 will cause an exception, see Section 3.3.1.2, "Out-of-Order Accesses to Guarded Memory," for more information.

3.4.1.5, 3-32 Remove the third sentence of the first paragraph starting with "Note that HID0[ICFI] and HID0[DCFI] must not both be set..." It is an inaccurate restriction.

**3.4.2, 3-33** Added the following sentence after the second sentence:

"Bit 24 corresponds to way 0, and bit 31 corresponds to way 7."

**3.4.3.1, 3-33** Added the following sentence after the second sentence:

"Bit 24 corresponds to way 0, and bit 31 corresponds to way 7."

3.4.4.6, 3-37 Add the following sentence to the end of the last paragraph:

| Section, Page No. | Changes |
|---|---|
| | If the effective address of a **dcba** instruction targets a direct-store segment (T = 1), the instruction will not no-op, and the results are undefined. |
| **3.5.8, 3-44** | In Section 3.5.8, “L1 Cache Invalidation and Flushing,” change step 2 to the following: “Way n + 1: Repeat the process shown in step 1 for the next way in the cache. This is started by incrementing the base offset used for the last set in way n by 32 bytes. Now PA[21:23] is incremented by one. Then repeat the remainder of step 1. |
| **3.5.8, 3-44** | In Section 3.5.8, “L1 Cache Invalidation and Flushing,” change step 3 to the following: “Way n + 2 to way n + 7: Repeat the process described in step 2 six more times (effectively progressing through all 8 combinations of PA[21:23]). |
| **3.6.3.2, 3-54** | Change the second sentence of the second paragraph to read as follows:<br>Note that while it is not an error to enable the prefetch engines when the L2 cache is disabled, it will result in a performance loss and is strongly discouraged. |
| 3.6.4, 3-57 | Replace the three sentences following the bullet list and before Table 3-11 with the following:<br>For the MPC7448, accesses that meet one of these three conditions always propagate to the MPX bus as write-through stores.<br>For write-through stores ≥ 8 bytes that hit in the L2, the line is updated in the L2, but the status bits for the block are not changed, and the write-through store propagates to the MPX bus. Otherwise, for write-through stores < 8 bytes, if the L2 block is modified, the block is cast out to the MPX bus and the write-through store propagates to the MPX bus; if the L2 block is not modified, the block is invalidated, and the write-through store propagates to the MPX bus.<br>In case of multiple pending requests to the L2 cache, the priorities are as shown in Table 3-12. |
| 3.6.4.3, 3-59 | Add the following paragraphs below the one paragraph in this section.<br>In the MPC7448, write-through stores use byte enables in the L1 data cache to merge the write data with the current cache contents (if it hits). For write-through stores ≥ 8 bytes that hit in the L2, the line is updated in the L2 with the store data, but the status bits for the block are not changed, and only the write-through store propagates to the MPX bus.<br>If a store of less than 8 bytes hits in the L2 cache then one of the following applies:<br>—If the L2 block is modified, the block is cast out to the MPX bus, and only the write-through store propagates to the MPX bus<br>—If the L2 block is not modified, the block is invalidated, and only the write-through store propagates to the MPX bus. |
| **4.3, 4-10** | In Figure 4-3, add 0 to the 6-bit reserved field in bits 0-5 of the MSR as follows: |

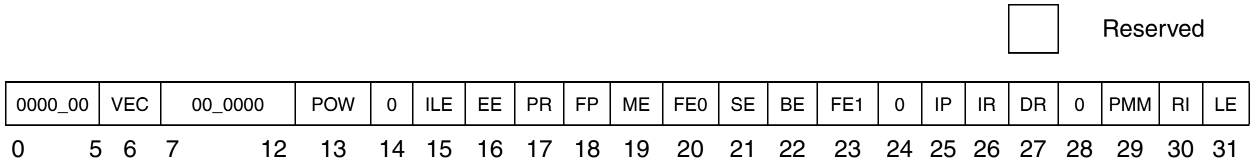


Figure 4-3. Machine State Register (MSR)

**4.6.2.1, 4-20** In Table 4-9, replace the description for the SRR1 register ‘11’ setting description with the following:

11MSS error. Set for an L2 cache tag parity or L2 data parity error, L3 SRAM data parity error, L3 tag parity error, address bus parity error, data bus parity error, or bus transfer error acknowledge error; otherwise zero. Refer to Section 2.2.5.4, “Memory Subsystem Status Register (MSSSR0),” for more information.

4.6.3.1, 4-33 In Table 4-10, replaced DSISR Setting Description 5 with the following:

Set if the **lwarx** or **stwcx.** instruction is attempted to write-through (W =1) or caching-inhibited (I = 1) memory or when the data cache is disabled or locked.

5.3.1, 5-26 Table 5-7, Upper BAT Register (BAT*n*U), Bits 15–18, XBL, replace with the following:

| | | | |
|---|---|---|---|
| Upper BAT Register (BAT*n*U) | 15–18 | — | Reserved on the MPC7441, MPC7450, and the MPC7451. |
| | | XBL | Extended block length. This XBL field is used by the MPC7445, MPC7447, MPC7447A, MPC7448, MPC7455, and MPC7557 as an extension of the BL field to lengthen the block size. When HID0[XBBSEN] is cleared, BATnU[15–18] will be automatically cleared (0b0000) when a **mtspr** instruction writes the BAT*n*U register. When HID0[XBBSEN] is set, software is allowed to write bits BATU[15–18], which become the 4 MSBs of the extended 15-bit BL field (BATU[15–29]). This allows for extended BAT block sizes of 512 MB, 1 GB, 2 GB, and 4 GB.<br>Values for the extended block length mask are listed in Table 5-9. Note that the XBL field is always used to determine the block length, regardless of the state of the HID0[XBSEN] bit. Additionally, if HID0[XBBSEN] is set at startup and then cleared after startup, the XBL bits will not clear but stay the same and the block lengths will continue to be used according to the value of the XBL and BL bits. See Table 2-7 and Table 5-9 for more information. |

5.3.2.1, 5-29 First paragraph, replace with the following:

On the MPC7445, MPC7447, MPC7447A, MPC7455, MPC7457, and MPC7448, the BAT block size is increased through the XBL field in the Upper BAT register, as shown in Figure 5-10. This allows for extended BAT block sizes of 512 MB, 1 GB, 2 GB, and 4 GB. The BL field is extended to 15 bits, with the XBL bits becoming the 4 most significant bits (MSBs) for the block size. The XBL bits can only be written when the extended BAT block size is enabled (HID0[XBBSEN] = 1). Note that XBL fields are always used for address translation and clearing HID0[XBSEN] does not automatically clear the XBL bits; see Table 2-7 for more information. The encoding for the extended BL field is shown in Table 5-9.

5.3.3, 5-32 First paragraph, add the following sentence after the first sentence:

**Section, Page No.** **Changes**

(Note that extended addressing does not have to be turned on to extend the BAT size.)

Chapter 8, 8-1

Second paragraph, replace with the following:

The MPC7450 provides a mode switch through the $\overline{\mathrm{BMODE0}}$ signal that, when sampled asserted just prior to $\overline{\mathrm{HRESET}}$ negation, enables the MPX bus protocol operation, and when sampled negated just prior to $\overline{\mathrm{HRESET}}$ negation, enables a limited subset of the 60x bus protocol. The MPX bus is derived from the 60x bus interface and includes a 72-bit data bus (including 8 parity bits) and a 44-bit address bus (including 5 parity bits) along with sufficient control signals to allow for unique system level optimizations.

8.2, 8-5

Second paragraph, replace with the following:

The value of the $\overline{\mathrm{BMODE0}}$ signal just prior to $\overline{\mathrm{HRESET}}$ negation determines whether the MPC7450 operates with the 60x bus or the MPX bus. The inverse of this value is stored in bits 16–17 of the BMODE field in MSSCR0. The state of MSSCR0[BMODE] is active high, meaning that if $\overline{\mathrm{BMODE0}}$ is detected as asserted immediately before the negation of $\overline{\mathrm{HRESET}}$, MSSCR0[16] = 1 and MPX bus mode is selected; if negated immediately before the negation of $\overline{\mathrm{HRESET}}$, MSSCR0[16] = 0 and 60x bus mode is selected.

8.2.6, 8-13

Replace the parenthetical phrase in the second sentence with:

(assertion of $\overline{\mathrm{BMODE0}}$ just prior to $\overline{\mathrm{HRESET}}$ negation sets the MPC7450 bus mode).

8.4.4.7, 8-50

Replaced Section 8.4.4.7 text and Table 8-10 with the following:

Refer to the *MPC7448 RISC Microprocessor Hardware Specifications* for specific information on the $\overline{\mathrm{LVRAM}}$ signal.

8.4.4.8, 8-51

Replace Section 8.4.4.8 and Table 8-11 with the following:

The $\overline{\mathrm{BMODE}}$[0:1] signals are sampled just prior to $\overline{\mathrm{HRESET}}$ negation to select the bus interface mode. (Please refer to the appropriate hardware specification document for the exact timing.) The MSSCR0[BMODE] field reflects the inverse of the voltage levels on $\overline{\mathrm{BMODE}}$[0:1] at this time.

The $\overline{\mathrm{BMODE}}$[0:1] signals are also sampled after $\overline{\mathrm{HRESET}}$ negation to configure the MPC7450's address bus driven mode and processor identification. Note that the value on $\overline{\mathrm{BMODE}}$[0:1] after reset negation determines other values of MSSCR0 as follows:

- $\overline{\mathrm{BMODE0}}$ (post reset) → MSSCR0[ABD] (field value is inverted from signal)
- $\overline{\mathrm{BMODE1}}$ (post reset) → MSSCR0[ID] (field value is inverted from signal)

Table 8-11 shows the meaning of $\overline{\mathrm{BMODE}}$[0:1] signal values, both during $\overline{\mathrm{HRESET}}$ assertion and after $\overline{\mathrm{HRESET}}$ negation.

**NOTE**

$\overline{\mathrm{BMODE1}}$ must not be asserted while $\overline{\mathrm{HRESET}}$ is asserted; this mode is reserved.

**Table 0-1. $\overline{\text{BMODE}}$[0:1] Signal Meanings**

| Signal | Value | Signal Meaning | |
|---|---|---|---|
| | | During $\overline{\text{HRESET}}$ | After $\overline{\text{HRESET}}$ |
| $\overline{\text{BMODE0}}$ | 0 | MPX bus mode | ABD mode enabled |
| | 1 | 60x bus mode | ABD mode disabled |
| $\overline{\text{BMODE1}}$ | 0 | Reserved—Do not use | Processor ID 1 |
| | 1 | Required | Processor ID 0 |

Table 8-12 shows how $\overline{\text{BMODE}}$[0:1] may be connected to achieve various configurations.

**Table 0-2. BMODE Configuration**

| Desired Configuration | | | Signal Connection[1] | |
|---|---|---|---|---|
| Bus Mode | ABD Mode Enabled? (MSSCR0[ABD]) | Processor ID (MSSCR0[ID]) | $\overline{\text{BMODE0}}$ | $\overline{\text{BMODE1}}$ |
| MPX | Yes (1) | 1 | 0 | ¬$\overline{\text{HRESET}}$ |
| MPX | No (0) | 1 | $\overline{\text{HRESET}}$ | ¬$\overline{\text{HRESET}}$ |
| 60x | Yes (1) | 1 | ¬$\overline{\text{HRESET}}$ | ¬$\overline{\text{HRESET}}$ |
| 60x | No (0) | 1 | 1 | ¬$\overline{\text{HRESET}}$ |
| MPX | Yes (1) | 0 | 0 | 1 |
| MPX | No (0) | 0 | $\overline{\text{HRESET}}$ | 1 |
| 60x | Yes (1) | 0 | ¬$\overline{\text{HRESET}}$ | 1 |
| 60x | No (0) | 0 | 1 | 1 |

[1] ¬$\overline{\text{HRESET}}$ is the inverse of $\overline{\text{HRESET}}$.

8.4.4.8.1, 8-51 First paragraph, replace with the following:

If the $\overline{\text{BMODE0}}$ input signal is sampled asserted just prior to $\overline{\text{HRESET}}$ negation, MPX bus mode is selected. However, if the $\overline{\text{BMODE0}}$ input signal is negated just prior to $\overline{\text{HRESET}}$ negation, 60x bus mode is selected.

8.4.4.8.3, 8-52 First paragraph, replace ‘during $\overline{\text{HRESET}}$ negation’ with ‘just prior to $\overline{\text{HRESET}}$ negation.’

8.4.5.5, 8-56 Under State Meaning, second sentence, replace ‘the bus clock frequency’ with ‘half the processor clock frequency.’

9.3.2.1, 9-17 Replaced the third sentence of the first paragraph with the following two sentences:
When the MPC7450 is driving the address bus but not running an address transaction, only transfer start ($\overline{\text{TS}}$), is driven to a 1. The other address bus signals—A[0:35], AP[0:4], TT[0:4], TSIZ[0:2], $\overline{\text{TBST}}$, $\overline{\text{GBL}}$, $\overline{\text{WT}}$, and $\overline{\text{CI}}$—are driven, but not to any specific value.

| Section, Page No. | Changes |
|---|---|
| 10.2.3, 10-4 | Add the following sentence to the end of the second paragraph:<br>Note that when asserting $\overline{\text{HRESET}}$, the processor will take the necessary number of clock cycles to initialize the processor. See the hardware specification for your device to determine the required number of clock cycles. |
| 10.3.2, 10-4 | Add the following sentence to the first paragraph after the first sentence:<br>Note that when asserting $\overline{\text{HRESET}}$, the processor will take the necessary number of clock cycles to initialize the processor. See the hardware specification for your device to determine the required number of clock cycles. |
| 11.5.1, 11-15 | Table 11-9, add the following note to PMC1 Event 21:<br>Note: This event increments even for accesses with stalls resulting from full buffers and other situations. This can lead to significant overcounting compared to the number of reloads to the cache. For that reason, to more accurately characterize L1 cache misses, it is recommended that the instruction cache reload event PMC2SEL[48] be used instead. |
| 11.5.2, 11-20 | Table 11-10, add the following note to PMC2 Event 21:<br>Note: This event increments even for accesses with stalls resulting from full buffers and other situations. This can lead to significant overcounting compared to the number of reloads to the cache. For that reason, to more accurately characterize L1 cache misses, it is recommended that the instruction cache reload event PMC2SEL[48] be used instead. |
| 11.5.2, 11-21 & 11-22 | Table 11-10, add the following note to PMC2 Events 23, 37, and 38:<br>Note: This event increments even for accesses with stalls associated with address collisions (such as when two loads to the same cache line miss in the L1), loads and stores to the same address, full buffers, and other situations. This can lead to significant overcounting compared to the number of reloads to the cache. For that reason, to more accurately characterize L1 cache misses, it is recommended that data cache reload event PMC2SEL[49] be used instead. |
| 11.5.2, 11-22 | Table 11-10, PMC2 Event 48, replace description with the following:<br>Counts times that the L1 instruction cache is reloaded with a new cache line. All accesses are counted, including cache-inhibited accesses, accesses with the cache disabled, and accesses that miss in the instruction cache when all ways are locked. |
| 11.5.3, 11-25 | Table 11-11, add the following note to PMC3 Event 23:<br>Note: This event increments even for accesses with stalls associated with address collisions (such as when two loads to the same cache line miss in the L1), loads and stores to the same address, full buffers, and other situations. This can lead to significant overcounting compared to the number of reloads to the cache. For that reason, to more accurately characterize L1 cache misses, it is recommended that data cache reload event PMC2SEL[49] be used instead. |
| **Appendix C, C-1** | Remove footnote 8 from Table C-1, “PowerPC SPR Encodings Ordered by Decimal Value.” This affects the following rows. Note only affected rows are shown. |

**Table C-1. PowerPC SPR Encodings Ordered by Decimal Value**

| Register Name | SPR[1] | | | Access | mfspr/mtspr |
|---|---|---|---|---|---|
| | Decimal | spr[5–9] | spr[0–4] | | |
| SVR | 286 | 01000 | 11110 | Supervisor (OEA) | **mfspr** |
| L2ERRINJHI | 985 | 11110 | 11001 | Supervisor (OEA) | Both |
| L2ERRINJLO | 986 | 11110 | 11010 | Supervisor (OEA) | Both |
| L2ERRINJCTL | 987 | 11110 | 11011 | Supervisor (OEA) | Both |
| L2CAPTDATAHI | 988 | 11110 | 11100 | Supervisor (OEA) | **mfspr** |
| L2CAPTDATALO | 989 | 11110 | 11101 | Supervisor (OEA) | **mfspr** |
| L2CAPTECC | 990 | 11110 | 11110 | Supervisor (OEA) | **mfspr** |
| L2ERRDET | 991 | 11110 | 11111 | Supervisor (OEA) | Special [2] |
| L2ERRDIS | 992 | 11111 | 00000 | Supervisor (OEA) | Both |
| L2ERRINTEN | 993 | 11111 | 00001 | Supervisor (OEA) | Both |
| L2ERRATTR | 994 | 11111 | 00010 | Supervisor (OEA) | Both |
| L2ERRADDR | 995 | 11111 | 00011 | Supervisor (OEA) | **mfspr** |
| L2ERREADDR | 996 | 11111 | 00100 | Supervisor (OEA) | **mfspr** |
| L2ERRCTL | 997 | 11111 | 00101 | Supervisor (OEA) | Both |

Appendix C, C-5

In Table C-2, “PowerPC SPR Encodings Ordered by Register Name,” make the following changes:

•Remove footnote 1 from all register name cells and footnote 5 from the table. Renumber accordingly.

•Update IBATSU register name cell to IBATSU[4]

•Update the decimal cell of L3ITCR0 to 984

•Update the decimal cell of L3OHCR to 1000

Affected rows are shown as follows.

**Table C-2. PowerPC SPR Encodings Ordered by Register Name**

| Register Name | SPR[1] | | | Access | mfspr/mtspr |
|---|---|---|---|---|---|
| | Decimal | spr[5–9] | spr[0–4] | | |
| IBAT5U [4] | 562 | 10001 | 10010 | Supervisor (OEA) | Both |
| L2CAPTECC | 990 | 11110 | 11110 | Supervisor (OEA) | **mfspr** |
| L2CAPTDATAHI | 988 | 11110 | 11100 | Supervisor (OEA) | **mfspr** |
| L2CAPTDATALO | 989 | 11110 | 11101 | Supervisor (OEA) | **mfspr** |
| L2ERRADDR | 995 | 11111 | 00011 | Supervisor (OEA) | **mfspr** |
| L2ERRATTR | 994 | 11111 | 00010 | Supervisor (OEA) | Both |

**Table C-2. PowerPC SPR Encodings Ordered by Register Name (continued)**

| Register Name | SPR[1] | | | Access | mfspr/mtspr |
|---|---|---|---|---|---|
| | Decimal | spr[5–9] | spr[0–4] | | |
| L2ERRCTL | 997 | 11111 | 00101 | Supervisor (OEA) | Both |
| L2ERRDET | 991 | 11110 | 11111 | Supervisor (OEA) | Special [2] |
| L2ERRDIS | 992 | 11111 | 00000 | Supervisor (OEA) | Both |
| L2ERREADDR | 996 | 11111 | 00100 | Supervisor (OEA) | **mfspr** |
| L2ERRINJCTL | 987 | 11110 | 11011 | Supervisor (OEA) | Both |
| L2ERRINJH I | 985 | 11110 | 11001 | Supervisor (OEA) | Both |
| L2ERRINJLO | 986 | 11110 | 11010 | Supervisor (OEA) | Both |
| L2ERRINTEN | 993 | 11111 | 00001 | Supervisor (OEA) | Both |
| L3ITCR0 [6] | 984 | 11111 | 01000 | Supervisor (OEA) | Both |
| L3OHCR [7] | 1000 | 11110 | 11000 | Supervisor (OEA) | Both |
| SVR | 286 | 01000 | 11110 | Supervisor (OEA) | **mfspr** |

[1] Note that the order of the two 5-bit halves of the SPR number is reversed compared with actual instruction coding. For **mtspr** and **mfspr** instructions, the SPR number coded in assembly language does not appear directly as a 10-bit binary number in the instruction. The number coded is split into two 5-bit halves that are reversed in the instruction, with the high-order 5 bits appearing in bits 16–20 of the instruction and the low-order 5 bits in bits 11–15.

[2] Most bits are bit reset/write 1 clear. A write of 0 to a bit does not change it. A write of 1 to a bit clears it. Reads act normally.

***How to Reach Us:***

**Home Page:**
freescale.com

**Web Support:**
freescale.com/support

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

Freescale reserves the right to make changes without further notice to any products herein. Freescale makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in Freescale data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer’s technical experts. Freescale does not convey any license under its patent rights nor the rights of others. Freescale sells products pursuant to standard terms and conditions of sale, which can be found at the following address: freescale.com/SalesTermsandConditions.

Document Number:  MPC7450UMAD
Rev. 5.3
07/2013

BUILT ON
Power™

freescale™