MOTOROLA
Semiconductor Products Sector

Order Number: MPC603EUMAD/D
Rev. 1, 10/2000

PowerPC™

Errata to
MPC603e & EC603e Microprocessor User's Manual, rev. 1

This errata describes corrections to the *MPC603e & EC603e RISC Microprocessor User's Manual*. For convenience, the section number and page number of the errata item in the user's manual are provided.

To locate any published updates for this document, refer to the world-wide web at http://www.motorola.com/semiconductors.

DigitalDNA
From Motorola

Section #/Page #	Changes
1.1.1, 1-6	The bus connecting the LSU to the GPRs should be 32 bits wide.
2.1.1, 2-3	Figure 2-1 in revision 1 of the published manual should be replaced by the following figure:

SUPERVISOR MODEL—OEA

Configuration Registers

Hardware Implementation Registers[1]

HID0	SPR 1008
HID1	SPR 1009

Machine State Register

MSR

Processor Version Register

PVR	SPR 287

Memory Management Registers

Instruction BAT Registers

IBAT0U	SPR 528
IBAT0L	SPR 529
IBAT1U	SPR 530
IBAT1L	SPR 531
IBAT2U	SPR 532
IBAT2L	SPR 533
IBAT3U	SPR 534
IBAT3L	SPR 535

Data BAT Registers

DBAT0U	SPR 536
DBAT0L	SPR 537
DBAT1U	SPR 538
DBAT1L	SPR 539
DBAT2U	SPR 540
DBAT2L	SPR 541
DBAT3U	SPR 542
DBAT3L	SPR 543

Software Table Search Registers[1]

DMISS	SPR 976
DCMP	SPR 977
HASH1	SPR 978
HASH2	SPR 979
IMISS	SPR 980
ICMP	SPR 981
RPA	SPR 982

SDR1

SDR1	SPR 25

Segment Registers

SR0
SR1
⋮
SR15

Exception Handling Registers

Data Address Register

DAR	SPR 19

DSISR

DSISR	SPR 18

SPRGs

SPRG0	SPR 272
SPRG1	SPR 273
SPRG2	SPR 274
SPRG3	SPR 275

Save and Restore

SRR0	SPR 26
SRR1	SPR 27

Miscellaneous Registers

Time Base Facility (For Writing)

TBL	SPR 284
TBU	SPR 285

Decrementer

DEC	SPR 22

Instruction Address Breakpoint Register[1]

IABR	SPR 1010

External Address Register (Optional)

EAR	SPR 282

USER MODEL—UISA

General-Purpose Registers

GPR0
GPR1
⋮
GPR31

Floating-Point Registers[2]

FPR0
FPR1
⋮
FPR31

Condition Register

CR

Floating-Point Status and Control Register[2]

FPSCR

XER

XER	SPR 1

Link Register

LR	SPR 8

Count Register

CTR	SPR 9

USER MODEL—VEA

TBL	TBR 268
TBU	TBR 269

Notes: [1]These registers are 603e–specific (PID6-603e and PID7v-603e) registers. They may not be supported by other PowerPC processors.
[2]Not supported on the EC603e microprocessor.

Section #/Page #	Changes
2.1.1, 2-4	Add the following text after the first paragraph of the sub-bullet for Floating-point registers (FPRs):

Before the **stfd** instruction is used to store the contents of an FPR to memory, the FPR must have been initialized after reset (explicitly loaded with any value) by using a floating point load instruction.

2.1.1, 2-5 — Replace Table 2-1 with the following:

Table 2-1. MSR[POW] and MSR[TGPR] Bits

Bit	Name	Description
13	POW	Power management enable (603e-specific) 0 Disables programmable power modes (normal operation mode). 1 Enables programmable power modes (nap, doze, or sleep mode). This bit controls the programmable power modes only; it has no effect on dynamic power management (DPM). MSR[POW] may be altered with an **mtmsr** instruction only. Also, when altering the POW bit, software may alter only this bit in the MSR and no others. The **mtmsr** instruction must be followed by a context-synchronizing instruction. See Chapter 9, “Power Management,” for more information on power management.
14	TGPR	Temporary GPR remapping (603e-specific) 0 Normal operation 1 TGPR mode. GPR0–GPR3 are remapped to TGPR0–TGPR3 for use by TLB miss routines. The contents of GPR0–GPR3 remain unchanged while MSR[TGPR] = 1. Attempts to use GPR4–GPR31 with MSR[TGPR] = 1 yield undefined results. When this bit is set, all instruction accesses to GPR0–GPR3 are mapped to TGPR0–TGPR3, respectively. The TGPR bit is set when either an instruction TLB miss, data TLB miss on load, or data TLB miss on store exception is taken. The TGPR bit is cleared by an **rfi** instruction.

2.1.2.1, 2-8 — Replace Table Table 2-2 with the HID0 bits descriptions as follows:

Table 2-2. HID0 Bit Functions

Bit	Name	Function
0	EMCP	Enable MCP. The primary purpose of this bit is to mask out further machine check exceptions caused by assertion of MCP, similar to how MSR[EE] can mask external interrupts. 0 Masks MCP. Asserting MCP does not generate a machine check exception or a checkstop. 1 Asserting MCP causes checkstop if MSR[ME] = 0 or a machine check exception if ME = 1.
1	—	Reserved
2	EBA	Enable/disable 60x bus address parity checking 0 Disables address parity checking. 1 Allows a address parity error to cause a checkstop if MSR[ME] = 0 or a machine check exception if MSR[ME] = 1. EBA and EBD allow the processor to operate with memory subsystems that do not generate parity.
3	EBD	Enable 60x bus data parity checking 0 Disables data parity checking. 1 Allows a data parity error to cause a checkstop if MSR[ME] = 0 or a machine check exception if MSR[ME] = 1. EBA and EBD allow the processor to operate with memory subsystems that do not generate parity.
4	BCLK	CLK_OUT output enable and clock type selection. Used in conjunction with HID0[ECLK] and the HRESET signal to configure CLK_OUT. See Table 2-3.

Table 2-2. HID0 Bit Functions (Continued)

Bit	Name	Function
5	EICE	Enables in-circuit emulator outputs for pipeline tracking. See Section 7.2.11, “Pipeline Tracking Support,” for more information.
6	ECLK	CLK_OUT output enable and clock type selection. Used in conjunction with HID0[BCLK] and the $\overline{\text{HRESET}}$ signal to configure CLK_OUT. See Table 2-3.
7	PAR	Disable precharge of $\overline{\text{ARTRY}}$. 0 Precharge of $\overline{\text{ARTRY}}$ enabled 1 Alters bus protocol slightly by preventing the processor from driving $\overline{\text{ARTRY}}$ to high (negated) state. If this is done, the system must restore the signals to the high state.
8	DOZE[1]	Doze mode enable. Operates in conjunction with MSR[POW]. 0 Doze mode disabled. 1 Doze mode enabled. Doze mode is invoked by setting MSR[POW] while this bit is set. In doze mode, the PLL, time base, and snooping remain active.
9	NAP[1]	Nap mode enable. Operates in conjunction with MSR[POW]. 0 Nap mode disabled. 1 Nap mode enabled. Doze mode is invoked by setting MSR[POW] while this bit is set. In nap mode, the PLL and the time base remain active.
10	SLEEP[1]	Sleep mode enable. Operates in conjunction with MSR[POW]. 0 Sleep mode disabled. 1 Sleep mode enabled. Sleep mode is invoked by setting MSR[POW] while this bit is set. $\overline{\text{QREQ}}$ is asserted to indicate that the processor is ready to enter sleep mode. If the system logic determines that the processor may enter sleep mode, the quiesce acknowledge signal, $\overline{\text{QACK}}$, is asserted back to the processor. Once $\overline{\text{QACK}}$ assertion is detected, the processor enters sleep mode after several processor clocks. At this point, the system logic may turn off the PLL by first configuring PLL_CFG[0–3] to PLL bypass mode, then disabling SYSCLK.
11	DPM[1]	Dynamic power management enable. 0 Dynamic power management is disabled. 1 Functional units enter a low-power mode automatically if the unit is idle. This does not affect operational performance and is transparent to software or any external hardware.
12–15	—	Reserved
16	ICE[2]	Instruction cache enable 0 The instruction cache is neither accessed nor updated. All pages are accessed as if they were marked cache-inhibited (WIM = X1X). Potential cache accesses from the bus (snoop and cache operations) are ignored. In the disabled state for the L1 caches, the cache tag state bits are ignored and all accesses are propagated to the L2 cache or bus as single-beat transactions. For those transactions, however, $\overline{\text{CI}}$ reflects the original state determined by address translation regardless of cache disabled status. ICE is zero at power-up. 1 The instruction cache is enabled
17	DCE[2]	Data cache enable 0 The data cache is neither accessed nor updated. All pages are accessed as if they were marked cache-inhibited (WIM = X1X). Potential cache accesses from the bus (snoop and cache operations) are ignored. In the disabled state for the L1 caches, the cache tag state bits are ignored and all accesses are propagated to the L2 cache or bus as single-beat transactions. For those transactions, however, $\overline{\text{CI}}$ reflects the original state determined by address translation regardless of cache disabled status. DCE is zero at power-up. 1 The data cache is enabled.

Table 2-2. HID0 Bit Functions (Continued)

Bit	Name	Function
18	ILOCK[2]	Instruction cache lock 0 Normal operation 1 Instruction cache is locked. A locked cache supplies data normally on a hit, but are treated as a cache-inhibited transaction on a miss. On a miss, the transaction to the bus or the L2 cache is single-beat, however, $\overline{CI}$ still reflects the original state as determined by address translation independent of cache locked or disabled status. To prevent locking during a cache access, an **isync** instruction must precede the setting of ILOCK.
19	DLOCK[2]	Data cache lock. 0 Normal operation 1 Data cache is locked. A locked cache supplies data normally on a hit but is treated as a cache-inhibited transaction on a miss. On a miss, the transaction to the bus or the L2 cache is single-beat, however, $\overline{CI}$ still reflects the original state as determined by address translation independent of cache locked or disabled status. A snoop hit to a locked L1 data cache performs as if the cache were not locked. A cache block invalidated by a snoop remains invalid until the cache is unlocked. To prevent locking during a cache access, a **sync** instruction must precede the setting of DLOCK.
20	ICFI[2]	Instruction cache flash invalidate 0 The instruction cache is not invalidated. The bit is cleared when the invalidation operation begins (usually the next cycle after the write operation to the register). The instruction cache must be enabled for the invalidation to occur. 1 An invalidate operation is issued that marks the state of each instruction cache block as invalid without writing back modified cache blocks to memory. Cache access is blocked during this time. Bus accesses to the cache are signaled as a miss during invalidate-all operations. Setting ICFI clears all the valid bits of the blocks and the PLRU bits to point to way L0 of each set. For 603e processors, the proper use of the ICFI and DCFI bits is to set them and clear them with two consecutive **mtspr** operations.
21	DCFI[2]	Data cache flash invalidate 0 The data cache is not invalidated. The bit is cleared when the invalidation operation begins (usually the next cycle after the write operation to the register). The data cache must be enabled for the invalidation to occur. 1 An invalidate operation is issued that marks the state of each data cache block as invalid without writing back modified cache blocks to memory. Cache access is blocked during this time. Bus accesses to the cache are signaled as a miss during invalidate-all operations. Setting DCFI clears all the valid bits of the blocks and the PLRU bits to point to way L0 of each set. For 603e processors, the proper use of the ICFI and DCFI bits is to set them and clear them with two consecutive **mtspr** operations.
22–23	—	Reserved
24	IFEM	Instruction fetch enable M (PID7v-603e only). Enables the M bit on the bus. Used for instruction fetches.
25–26	—	Reserved
27	FBIOB	Force branch indirect on bus. 0 Register indirect branch targets are fetched normally 1 Forces register indirect branch targets to be fetched externally
28	ABE[2]	Address broadcast enable—controls whether certain address-only operations (such as cache operations) are broadcast on the 60x bus. 0 Address-only operations affect only local caches and are not broadcast. 1 Address-only operations are broadcast on the 60x bus.Affected instructions are **dcbi**, **dcbf**, and **dcbst**. Note that these cache control instruction broadcasts are not snooped by the PID7v-603e. Refer to Section 3.2.3, “Data Cache Control,” for more information.

Table 2-2. HID0 Bit Functions (Continued)

Bit	Name	Function
29–30	—	Reserved
31	NOOPTI	No-op the data cache touch instructions. 0 The **dcbt** and **dcbtst** instructions are enabled. 1 The **dcbt** and **dcbtst** instructions are no-oped globally.

[1] See Chapter 9, "Power Management," for more information.

[2] See Chapter 3, "Instruction and Data Cache Operation," for more information

2.1.2.1, 2-8 Also, add to this section the following: Table 2-3, the paragraph preceding it, and the sentence immediately following it.

Table 2-3 shows how HID0[BCLK], HID0[ECLK], and $\overline{\text{HRESET}}$ are used to configure CLK_OUT. See Section 7.2.12.2, "Test Clock (CLK_OUT)—Output," for more information.

Table 2-3. HID0[BCLK] and HID0[ECLK] CLK_OUT Configuration

$\overline{\text{HRESET}}$	HID0[ECLK]	HID0[BCLK]	CLK_OUT
Asserted	x	x	Bus
Negated	0	0	High impedance
Negated	0	1	Core clock frequency
Negated	1	0	Bus
Negated	1	1	Core clock frequency

HID0 can be accessed with **mtspr** and **mfspr** using SPR1008.

Then, after Figure 2-3, add the following:

HID1 can be accessed with **mfspr** using SPR1009.

2.1.2.7, 2-12 Remove this section. The Run_N counter is not an SPR.

2.3.5.3, 2-41 Add the following text to the end of this section:

Note that incoherency may occur if the following sequence of accesses hits the same cache block: a write-through, a **dcbz** instruction, a snoop. This occurs when the logical address for the **dcbz** and the write-through store are different but aliased to the same physical page.

Section #/Page #	Changes
	To avoid potential adverse effects, **dcbz** should not address write-through memory that can be accessed through multiple logical addresses. Explicit store instructions that write all zeroes should be used instead.
	Note that broadcasting a sequence of **dcbz** instructions may cause snoop accesses to be retried indefinitely, which may cause the snoop originator to time out or may cause the snooped transaction to not complete. This can be avoided by disabling the broadcasting of **dcbz** by marking the memory space being addressed by the **dcbz** instruction as not global in the BAT or PTE.
2.3.5.4, 2-42	Replace the first four sentences of this section with the following:
	The **eciwx** instruction provides an alternative way to map special devices. The MMU translation of the EA is not used to select the special device, as it is used in loads and stores. Rather, it is used as an address operand that is passed to the device over the address bus. Four other signals (the burst and size signals on the 60x bus) are used to select the device; these four signals output the 4-bit resource ID (RID) field in the EAR register. The **eciwx** instruction also loads a word from the data bus that is output by the special device.
2.3.6.3.1, 2-44	The section should be completely replaced with the following:
	The supervisor-level cache management instruction in the PowerPC architecture, **dcbi**, should not be used on the 603e. The user-level **dcbf** instruction, described in Section 2.3.5.3, "Memory Control Instructions—VEA," and in Section 3.7, "Cache Control Instructions," should be used when the program needs to invalidate cache blocks. Note that the **dcbf** instruction causes modified blocks to be flushed to system memory if they are the target of a **dcbf** instruction whereas, by definition in the PowerPC architecture, the **dcbi** instruction only invalidates modified blocks.
3.1.3.2, 3-4	Change the last sentence of the paragraph to the following:
	To prevent the cache from being enabled or disabled in the middle of a data access, an **isync** instruction should be issued before changing the value of ICE.
3.2.3.2, 3-6	Change the last sentence of the first paragraph to the following:
	To prevent the cache from being enabled or disabled in the middle of a data access, a **sync** instruction should be issued before changing the value of DCE.
4.1, 4-3	The parenthetical clause in the second sentence of the first bullet should be “... (that is, all enabled floating-point exceptions are always precise on the 603e).”
4.1, 4-4	Figure 4-1 should be re-labeled as Table 4-1.

Section #/Page #	Changes
4.1, 4-5	At the end of the first sentence in the description of the floating-point unavailable exception, change "...when the floating-point available bit is disabled..." to "...when the floating-point available bit is cleared..."
	Also, change "... Note that the EC603e microprocessor... execution of a floating-point..." to "Note that the EC603e microprocessor... execution of any floating-point..."
	Also, the description for instruction translation miss should change from "... miss exception is caused when an effective address for an..." to "... miss exception is caused when the effective address for an..."
4.1, 4-6	The descriptions for Data load translation miss and Data store translation miss should change from "... miss exception is caused when an effective address for a..." to "... miss exception is caused when the effective address for a..."
4.2, 4-13	In Table 4-5 change the description of the FP bit from"...The processor prevents dispatch of floating-point instructions, including floating-point loads, stores, and moves, default..." to "...The processor prevents dispatch of floating-point instructions, including floating-point loads, stores, and moves; default..."
	Bit 24 is "Reserved" rather than "Reserved. Full Function."
4.2.2, 4-15	The first sentence of step 4 should be "The MSR is set as described in Table 4-7."
4.5, 4-17	In Table 4-7, the setting for the IP bit in the System reset row should be a 1 instead of the "—" shown in the table.
4.5, 4-18	The first note for Table 4-7 should read as "The floating-point available bit is always cleared to 0 on the EC603e microprocessor."
4.5.1.2, 4-20	The first sentence in the second paragraph should read as follows:
	Unlike a hard reset, no registers or latches are initialized; however, the instruction cache is disabled (HID0[ICE] = 0].
	The second sentence in the second paragraph should read as follows:
	After $\overline{\text{SRESET}}$ is recognized as asserted, the processor beings fetching instructions from the system reset routine at offset 0x0100.
	The third sentence should read as follows:
	When a soft reset occurs, registers are set as shown in Table 4-9 and HID0[ICE] is cleared.

Section #/Page #	Changes
4.5.5, 4-25	Replace the second sentence in the second paragraph with the following:

The 603e allows the next instruction in program order to complete, including handling any exceptions that instruction may generate. However, the 603e blocks subsequent instructions from completing and allows any outstanding stores to occur to system memory.

4.5.5, 4-26 — The last sentence of this section should be appended as follows:

The interrupt handler must send a command to the device that asserted $\overline{\text{INT}}$, acknowledging the interrupt and instructing the device to negate $\overline{\text{INT}}$ before the handler re-enables recognition of external interrupts.

4.5.6.1.1, 4-28 — Remove the heading "4.5.6.1.1 Page Address Translation Access." The text that follows this heading should just be part of section 4.5.6.1.

4.5.12, 4-34 — In Table 4-16, the KEY bit was omitted and the polarity of bit 15 was reversed. Therefore, replace the SRR1 row with the following:

Table 4-16. Instruction and Data TLB Miss Exceptions— Register Settings (partial)

SRR1	0–3	Loaded from condition register CR0 field
	4–11	Cleared
	12	KEY. Key for TLB miss (SR[Ks] or SR[Kp], depending on whether the access is a user or supervisor access).
	13	D/I. Data or instruction access 0 = Data TLB miss 1 = Instruction TLB miss
	14	WAY. Next TLB set to be replaced (set per LRU) 0 = Replace TLB associativity set 0 1 = Replace TLB associativity set 1
	15	S/L. Store or load data access 0 = Data TLB miss on load 1 = Data TLB miss on store (or C = 0)
	16–31	Loaded from MSR[16–31]

5.1.7, 5-16 — In Table 5-4, delete the sixth row (**lwarx** or **stwcx.** with W = 1).

5.1.8, 5-18 — In Table 5-5, add the following paragraph at the end of the **tlbie** description.

Software must ensure that instruction fetches or memory references to the virtual pages specified by the **tlbie** instruction have been completed prior to executing the **tlbie** instruction.

5.1.8, 5-18 — In Table 5-5, add the following sentence at the end of the **tlbsync** description.

For a complete description of the $\overline{\text{TLBISYNC}}$ signal, refer to Section 8.8.2, "$\overline{\text{TLBSYNC}}$ Input."

Section #/Page #	Changes
5.3, 5-20	Replace the Implementation Note for Block Address Translation with: The 603e BAT registers are not initialized by the hardware after the power-up or reset sequence. Consequently, all valid bits in both instruction and data BAT areas must be explicitly cleared before setting any BAT area for the first time and before enabling translation. Also, note that software must avoid overlapping blocks while updating a BAT area or areas. Even if translation is disabled, multiple BAT area hits (with the valid bits set) can corrupt the remaining portion (any bits except the valid bits) of the BAT registers. Thus, multiple BAT hits (with valid bits set) are considered a programming error whether translation is enabled or disabled and can lead to unpredictable results if translation is enabled (or if translation is disabled, when translation is eventually enabled). For the case of unused BATs (if translation is to be enabled) it is sufficient precaution to simply clear the valid bits of the unused BAT entries.”
5.4.3.1, 5-26	The second part of the second sentence in the first paragraph of that page should be deleted so that the sentence should read as follows: ITLB miss exception conditions are reported when there are no more instructions to be dispatched or retired (the pipeline is empty).
5.4.3.1, 5-26	The very last sentence on this page should be edited as follows: In order to uniquely identify a TLB entry as the required PTE, the TLB entry also contains...
5.4.3.1, 5-27	The phrase, “the valid entries are loaded and” should be deleted from the second sentence in the last paragraph of section 5.4.3.1.
5.4.4, 5-29	Figure 5-8 in the published manual incorrectly shows the loopback arrow on the left side pointing to the node above the word ‘otherwise’. Replace Figure 5-8 with the following:

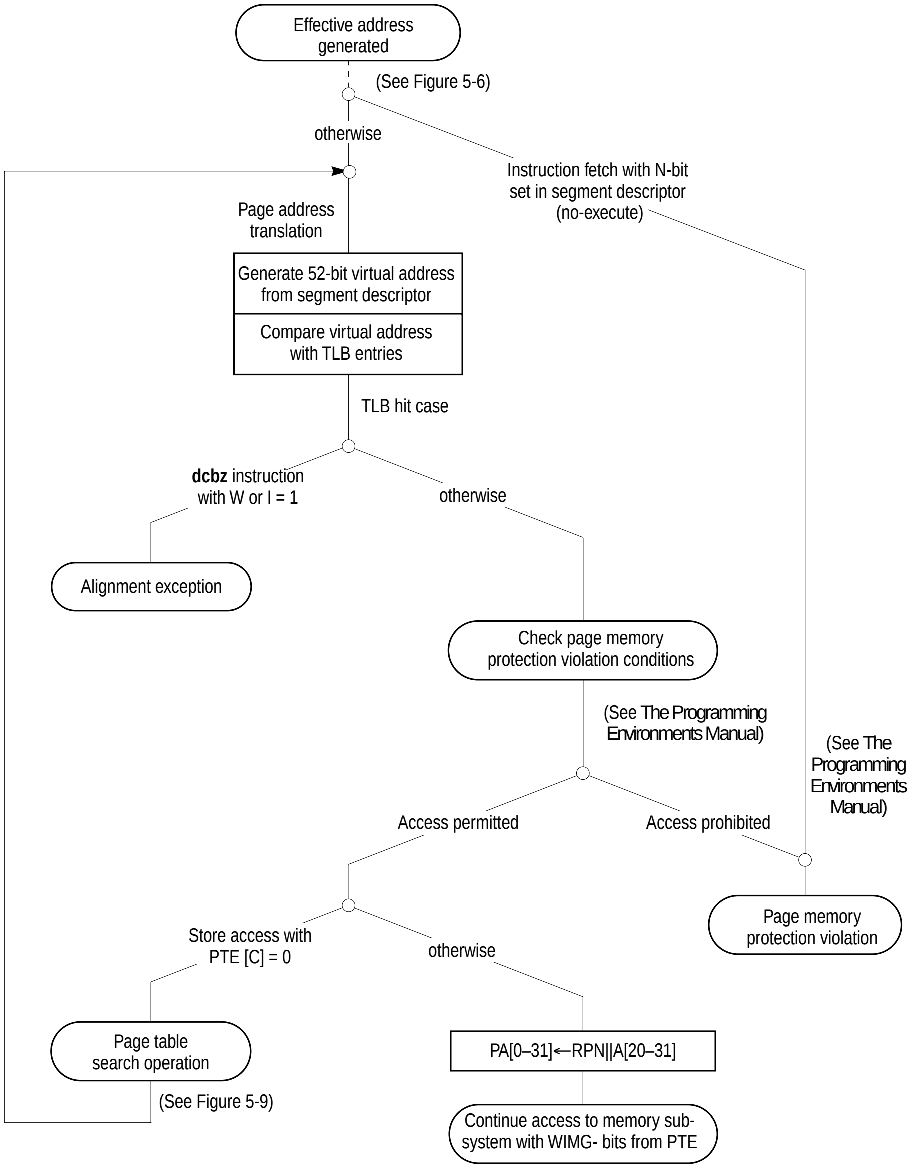

Figure 5-8. Page Address Translation Flow—TLB Hit

Section #/Page #	Changes
5.5.2.2.2, 5-46	Replace the doISI segment with the following:

```
doISI:
        mfsprr3, srr1 # get srr1
        andi.r2, r3, 0xffff# clean srr1
        addisr2, r2, 0x4000# or in srr1<1> = 1 to flag pte not found
isi1    mtctrr0      # restore counter
        mtsprsrr1, r2# set srr1
        mfmsrr0      # get msr
        xorisr0, r0, 0x8000# flip the msr<tgpr> bit
        mtcrf0x80, r3# restore CR0
        mtmsrr0      # flip back to the native gprs
        bvec400      # go to instr. access exception
#
```

5.5.2.2.2, 5-47 — Replace the third line of the dm1 segment with the following:

```
        bdnzf0, dm1# dec count br if cmp ne and if count not zero
```

5.5.2.2.2, 5-49 — Replace the comment of the second line of the chk0 segment with the following:

```
        andis.r3,r3,0x0008 # test the KEY bit (SRR1-bit 12)
```

5.5.2.2.2, 5-49 — Replace the second line in the chk2 segment with the following

```
        sthr1, 6(r2) # update page table
```

5.5.2.2.2, 5-50 — Replace the third line from the top of the page with the following

```
        beqdsi2:     # if little endian then:
```

8.2.1, 8-8 — The second part of the $\overline{\text{DBG}}$ bullet should be edited to read as follows: “The $\overline{\text{DBB}}$ signal is driven by the current bus master. $\overline{\text{DRTRY}}$ is only driven from the bus; $\overline{\text{ARTRY}}$ is driven from the bus, but only for the address bus tenure...”

8.3.2.2.1, 8-13 — The cross references at the end of this section should be to Table 7-1 and Table 7-2, instead of Table 8-1 and Table 8-2.

8.3.2.4, 8-17 — The fifth entry in the TSIZ[0–2] column (second access for the second misaligned entry) should be 010 instead of 011.

8.3.3, 8-20 — Replace sentences 3–5 with the following:

Although $\overline{\text{AACK}}$ can be asserted as early as the bus clock cycle following $\overline{\text{TS}}$ (see Figure 8-7), which allows a minimum address tenure of two bus cycles when the 603e clock is configured for 1:1 or 1.5: 1 processor-to-bus clock mode, the $\overline{\text{ARTRY}}$ snoop response cannot be determined in the minimum allowed address tenure period. Thus in a system with two or more 603e processors using 1:1 or 1.5:1 clock mode, $\overline{\text{AACK}}$ must not be asserted until the third clock of the address tenure (one address wait state) to allow the snooping 603e processors an opportunity to assert $\overline{\text{ARTRY}}$ on the bus. For other clock configurations (2:1, 2.5:1, 3:1, 3.5:1, and 4:1), the $\overline{\text{ARTRY}}$

Section #/Page #	Changes
	snoop response can be determined in the minimum address tenure period, and $\overline{\text{AACK}}$ may be asserted as early as the second bus clock of the address tenure.
8.4.4.1, 8-27	Insert a new section heading, 8.4.4.2, “Normal Burst Termination,” immediately after Figure 8-10.
8.7.4, 8-42	Replace the first sentence of the second paragraph with the following: The system quiesce state is entered by configuring the processor to assert the $\overline{\text{QREQ}}$ output.
8.8.1, 8-42	The first sentence of the second paragraph should read as follows: The reservation ($\overline{\text{RSRV}}$) output signal is driven synchronously with the bus clock and reflects the status of the reservation coherency bit in the reservation address buffer (see Section 3.9, “Instruction and Data Cache Operation,” for more information).
9.2, 9-2	In the third sentence of the last paragraph on this page, the SMI should have an overbar ($\overline{\text{SMI}}$).
9.2.1.4, 9-4	Change the fourth second-level bullet under the “Nap mode sequence” bullet to the following: The processor enters nap mode after several processor clocks.
9.2.1.5, 9-5	The last sub-bullet should have overbars on $\overline{\text{INT}}$ and $\overline{\text{SMI}}$.

Section #/Page #	Changes

Motorola Literature Distribution Centers:

USA/EUROPE: Motorola Literature Distribution; P.O. Box 5405; Denver, Colorado 80217; Tel.: 1-800-441-2447 or 1-303-675-2140;
World Wide Web Address: http://ldc.nmd.com/

JAPAN: Nippon Motorola Ltd SPD, Strategic Planning Office 4-32-1, Nishi-Gotanda Shinagawa-ku, Tokyo 141, Japan Tel.: 81-3-5487-8488

ASIA/PACIFIC: Motorola Semiconductors H.K. Ltd Silicon Harbour Centre 2, Dai King Street Tai Po Industrial Estate Tai Po, New Territories, Hong Kong

Mfax™: RMFAX0@email.sps.mot.com; TOUCHTONE 1-602-244-6609; US & Canada ONLY (800) 774-1848;
World Wide Web Address: http://sps.motorola.com/mfax

INTERNET: http://motorola.com/sps

Technical Information: Motorola Inc. SPS Customer Support Center 1-800-521-6274; electronic mail address: crc@wmkmail.sps.mot.com.

Document Comments: FAX (512) 895-2638, Attn: RISC Applications Engineering.

World Wide Web Addresses: http://www.motorola.com/PowerPC
http://www.motorola.com/netcomm
http://www.motorola.com/Coldfire

MOTOROLA

MPC603EUMAD/D