IBM PowerPC® 750FX RISC Microprocessor

Errata List

DD2.X

Version: 0.8

Preliminary - IBM Confidential

May 15, 2003

Printed in the United States of America May 2003

IBM Microelectronics Division
1580 Route 52, Bldg. 504
Hopewell Junction, NY 12533-6351

The IBM home page can be found at
http://www.ibm.com

The IBM Microelectronics Division home page
can be found at http://www-3.ibm.com/chips

Preface

This document identifies implementation differences between a referenced revision of the IBM PowerPC® 750FX RISC Microprocessor and its corresponding description in the *IBM PowerPC 750FX RISC Microprocessors User's Manual.*

The IBM PowerPC 750FX RISC Microprocessor , also known as the 750FX, has the following Processor Version Register (PVR) values for the respective design revision levels.

Note: The PowerPC 750FX PVR is 7000, which is not used in any previous PowerPC™ processor design. *Table 1, 750FX Processor Version Register (PVR)* lists the PVRs for the PowerPC 750FX.

Table 1. 750FX Processor Version Register (PVR)

750FX Design Revision Level	750FX PVR
DD2.0	0x700002X0
DD2.1	0x700002X1
DD2.2	0x700002X2
DD2.3	0x700002X3
Note: X is to be ignored and is for factory use only.	

This errata lists any processor differences from the following documents:

- *The PowerPC™ Architecture: A Specification for a New Family of RISC Processors*
- *IBM PowerPC 750FX RISC Microprocessors User's Manual*

Note: The section *Application Notes* on page 11 provides application-specific information that supplements this errata list.

Errata Fixed in DD2.0

The following DD1.X errata, which are listed and described in the *IBM PowerPC 750FX RISC Microprocessor Errata for DD1.1 and DD1.2*, have been fixed with DD2.0 and are not listed in this document:

- *Processor hangs with L2 data-only (L2DO)*
- *Asserting DBG speculatively may cause data corruption or a system hang*
- *Transferring data from BAT to GPR registers results in incorrect data*

Note: The numbering of the errata may change from one errata document version to another.

Summary of Errata

Table 2. Summary of IBM PowerPC 750FX RISC Microprocessor Errata for Revision DD2.X
(The section *Application Notes* on page 11 provides application-specific information.)

#	Problem	Overview	Impact	Workaround	Applicable to Revision?			
					2.0	2.1	2.2	2.3
1	*L2 cache invalidate may fail with DPM* [1]	If DPM is enabled during a global invalidate of the L2 cache, the global invalidate may not invalidate all of the L2's tags.	Possible system fail after L2 initialization and start-up.	Turn DPM off during an L2 tag invalidate.	Yes	Yes	Yes	Yes
2	*dcbz that hits in L1 cache may not snoop retry* [1]	If the target address of a **dcbz** instruction hits in the IBM PowerPC 750FX RISC Microprocessor L1 cache at the same time that a snoop is received to that address, the IBM PowerPC 750FX RISC Microprocessor may not react to the snoop, and may not generate a snoop retry to the other bus master.	For data that is shared in real time, stale data and data valid in multiple caches may result causing possible system failures.	Use of **dcbz** must be avoided for data that is shared in real time and which is not protected during writing through higher-level software synchronization protocols (such as semaphores). Use of **dcbz** must be avoided for managing semaphores themselves. (An alternative workaround is to prevent **dcbz** from hitting in the L1 cache by performing a **dcbf** to that address beforehand.)	Yes	Yes	Yes	Yes
3	*Segment register updates may corrupt data translation* [1]	A **mtsr,mtsrin** operation followed closely by an instruction causing a data address translation using page address translation can cause contention for the segment registers. This contention causes the data address translation to proceed using data from the wrong segment register.	Possible access to incorrect real address locations or false translation and data access exceptions.	Insert **isync**, **sc**, or **rfi** between any **mtsr**,**mtsrin** and instructions that cause a page data address translation.	Yes	Yes	Yes	Yes
4	*Stfd of uninitialized FPR can hang part* [1]	A stfd will hang the part, if its source FPR has powered up in a certain state.	All systems using stfd's.	Initialize all FPRs at POR.	Yes	Yes	Yes	Yes

Definitions: Yes indicates the errata is within the specified revision.
No indicates the errata has been fixed in the specified revision.

Notes: 1. This erratum is also present in the 750, 750CX, and 750CXe.
2: The DD2.3 revision has additional workaround capability. Refer to the workaround section in the applicable erratum.

Table 2. Summary of IBM PowerPC 750FX RISC Microprocessor Errata for Revision DD2.X (Continued)
(The section *Application Notes* on page 11 provides application-specific information.)

#	Problem	Overview	Impact	Workaround	Applicable to Revision?			
					2.0	2.1	2.2	2.3
5	*Tablewalk store collides with snoop copyback in clock bypass mode* [1]	A tablewalk completes by doing a store to update the pte (present bit) at the same time a snoop hit in the dcache results in a copyback. Both requests for the dcache collide in the dtag with data corruption.	A data integrity problem exists when snooping, address translation with paging, and clock bypass mode are all enabled.	Avoid using clock bypass mode with address translation (paging) enabled. If address translation is required, then use Block address translation (BAT).	Yes	Yes	Yes	Yes
6	*TEA causes processor hang or data corruption with MuM enabled*	A Processor hang condition or data corruption can occur if $\overline{\text{TEA}}$ is asserted in Miss under Miss mode.	If $\overline{\text{TEA}}$ is asserted during the first reload of a MuM, this may result in a processor hang or data corruption.	Do not allow assertion of $\overline{\text{TEA}}$ when MuM is enabled.	Yes	No	No	No
7	ARTRY signal may fail to precharge	In 2.5, 3.5, 4.5, and 5.5 PLL modes the processor may fail to precharge $\overline{\text{ARTRY}}$ while transitioning from nap mode to full power mode.	The processor and memory controller (and any other component observing the 60x bus) will observe $\overline{\text{ARTRY}}$ asserted on bus transactions and as $\overline{\text{ARTRY}}$ slowly charges the processor and memory controller may lose synchronization and hang and/or lose data integrity.	1. Avoid using 2.5, 3.5, 4.5, 5.5 PLL configuration modes. 2. Avoid using nap mode. Use doze mode instead. 3. Implement an external circuit that precharges $\overline{\text{ARTRY}}$ for one cycle 2 cycles after AACK.	Yes	No	No	No
8	*Nap mode is non-functional at frequencies above 400MHz*	Nap mode is non-functional at frequencies above 400MHz.	Unpredictable results if nap mode is used at frequencies greater than 400MHz	Either use doze mode for power savings or use nap mode at 400MHz.	Yes	Yes	Yes	No
9	*Double snoop push* [1]	Snooping may result with two castouts performed to the same address. The first castout may have stale data, but the second castout will have the most recent data.	May result in data corruption in the system or hang condition.	Use one of the following workarounds: 1. Allow both castouts to occur before using castout data. 2. Operate L2 in write-thru mode.	Yes	No	No	No

Definitions: Yes indicates the errata is within the specified revision.
No indicates the errata has been fixed in the specified revision.

Notes: 1. This erratum is also present in the 750, 750CX, and 750CXe.
2: The DD2.3 revision has additional workaround capability. Refer to the workaround section in the applicable erratum.

Table 2. Summary of IBM PowerPC 750FX RISC Microprocessor Errata for Revision DD2.X (Continued)
(The section *Application Notes* on page 11 provides application-specific information.)

#	Problem	Overview	Impact	Workaround	Applicable to Revision?			
					2.0	2.1	2.2	2.3
10	*SRR0 can become corrupted when Performance Monitor/Decrementer are enabled* [1]	A Performance Monitor Interrupt, taken immediately after a delayed Decrementer exception, can cause SRR0 to become corrupted.	The processor hangs.	When performance monitor interrupt signalling is enabled (MMCR0[ENINT] = 1), avoid decrementer interrupts by periodically setting the Decrementer register value to 0x7FFFFFFF. Performance monitor interrupts caused by Time Base bit transitions can be used to mediate this periodic setting of the Decrementer and in some cases to roughly simulate decrementer functionality during this time.	Yes	Yes	Yes	No
11	*Checkstop on parity error*	The processor will always checkstop on an I tag or L2 tag parity error even if the MSR ME bit is set.	The processor will always checkstop on an I tag or L2 tag parity error.	None.	Yes	No	No	No
12	*DPM is non-functional*	Dynamic Power Management (DPM) is incompatible with the data cache operation.	The added benefit of DPM power savings can not be realized.	DPM should be disabled (HID0 bit 11 set to b”0”).	Yes	No	No	No
13	*BAT usage restrictions*	Usage of the extended IBAT[4:7] and DBAT[4:7] with either 128k block sizes or uninitialized contents may result in incorrect block address translation.	Block sizes are limited to 256K-256M bytes on the 4 additional IBAT and DBAT registers. PowerPC 750 legacy applications that use 4 or less BAT registers are unaffected with respect to functionality. However, IBAT[4:7] and DBAT[4:7] must be initialized and held constant as described in the detailed description.	Block sizes of 128K cannot be used in the 4 additional BAT registers: IBAT[4:7] and DBAT[4:7]. The bits BEPI(14) and BL(10) must be set to a b'1' in IBATU[4:7] and DBATU[4:7] prior to enabling block address translation and must remain in this state thereafter. This workaround applies regardless of using IBAT[4:7] or DBAT[4:7].	Yes	No	No	No

Definitions: Yes indicates the errata is within the specified revision.
No indicates the errata has been fixed in the specified revision.

Notes: 1. This erratum is also present in the 750, 750CX, and 750CXe.
2: The DD2.3 revision has additional workaround capability. Refer to the workaround section in the applicable erratum.

Table 2. Summary of IBM PowerPC 750FX RISC Microprocessor Errata for Revision DD2.X (Continued)
(The section *Application Notes* on page 11 provides application-specific information.)

#	Problem	Overview	Impact	Workaround	Applicable to Revision?			
					2.0	2.1	2.2	2.3
14	*Snooping may cause system hang with MuM and DBWO*[2]	A Processor bus hang condition can occur if DBWO is required to force a snoop castout in Miss-under-Miss (MuM) mode.	MuM may result in a bus hang if DBWO is required to resolve system deadlock conditions.	Systems requiring DBWO to push snoop castouts should first disable MuM by setting HID0[14] to a zero (0).	Yes	Yes	Yes	Yes[2]
15	*3.3V I/O mode is non-functional*	The I/O devices on 750FX do not function with the processor configured in 3.3V I/O mode (BVSEL=b'1', L1_TSTCLK=b'0').	The 3.3V mode is not usable in production-level systems. However, for system prototyping and bring-up, the specified workaround may be used.	The processor can be interfaced with 3.3V I/O systems for limited usage. Connect OV_{DD} to 3.3V (3.3V max.) and configure the processor for 2.5V I/O mode (BVSEL=b'1', L1_TSTCLK=b'1'). To avoid reliability issues, restrict the usage of 3.3V I/O systems to less than 6000 power-on hours (POH) in this mode.	Yes	No	No	No
16	*LWARX or DCBT/ST causes data corruption with MuM enabled*	Store data may be lost when MuM is enabled and a **LWARX** or **DCBT/ST** instruction aliases a store in the store queue.	A data integrity problem or operating system hang may result.	Turn off MuM if **LWARX** or touch instructions are used.	Yes	No	No	No
17	*BTIC low voltage sensitivity*	The BTIC fails below 1.4V	The BTIC will fail below 1.4V causing an incorrect or illegal instruction to be executed.	Turn off the BTIC for voltages less than 1.4V (HID0 bit 26 = zero)	Yes	Yes	Yes	Yes

Definitions: **Yes** indicates the errata is within the specified revision.
No indicates the errata has been fixed in the specified revision.

Notes: 1. This erratum is also present in the 750, 750CX, and 750CXe.
2: The DD2.3 revision has additional workaround capability. Refer to the workaround section in the applicable erratum.

Table 2. Summary of IBM PowerPC 750FX RISC Microprocessor Errata for Revision DD2.X (Continued)
(The section *Application Notes* on page 11 provides application-specific information.)

#	Problem	Overview	Impact	Workaround	Applicable to Revision?			
					2.0	2.1	2.2	2.3
18	*PLL voltage sensitivity*	When V_{DD} is less than 1.35V, the phase-locked loop (PLL) circuit may not align the processor internal clock edges with the external clock (for example, SYSCLK), thus the processor may fail to meet I/O timing specifications or become non-functional.	Operation of this revision of the processor at V_{DD} less than 1.35V may cause a shift in the processor I/O timing specifications and result in a system malfunction. The amount of I/O timing shift increases as voltage is reduced from 1.35V. The I/O timing remains constant when V_{DD} is held constant.	Operate the processor at V_{DD} of 1.35V or greater.	Yes	No	No	No
19	*Dual-PLL switch may cause truncated internal clock pulses and processor malfunction*	Using the Dual-PLL feature to switch between PLL0 and PLL1 may result in truncated internal clock pulses and processor malfunction. *Table 3* provides a list of the revisions affected by this erratum.	Use of the Dual-PLL feature may result in unpredictable results.	Avoid using the Dual-PLL feature.	Yes	Yes	No	No
20	*Dual PLL switch at 4.5X mode may result in loss of ARTRY precharge*	ARTRY may not precharge after Dual PLL switching at 4.5X mode during snoop activity.	In the absence of an ARTRY pre-charge cycle, the processor and memory controller, and any other component observing the 60x bus, will observe ARTRY asserted on bus transactions. As ARTRY slowly charges, the processor and memory controller may lose synchronization causing a processor hang and/or data integrity issues.	1. Avoid using 4.5X mode with Dual PLL operation. 4.5X mode with a single PLL is not affected. 2. Implement an external circuit that precharges ARTRY. Two cycles after AACK, ARTRY is precharged for one cycle by the external circuit.	Yes	Yes	Yes	Yes
21	*MuM can cause invalid data with mispredicted branches*	Data from a cancelled load instruction may incorrectly be forwarded to a subsequent load miss, if MuM is enabled.	When Load MuM is enabled, incorrect data transfer may result.	Disable Load MuM with HID2 bit2 = b'1', or disable entire MuM with HID0 bit14 = b'0'.	Yes	Yes	Yes	No

Definitions: Yes indicates the errata is within the specified revision.
No indicates the errata has been fixed in the specified revision.

Notes: 1. This erratum is also present in the 750, 750CX, and 750CXe.
2: The DD2.3 revision has additional workaround capability. Refer to the workaround section in the applicable erratum.

Table 2. Summary of IBM PowerPC 750FX RISC Microprocessor Errata for Revision DD2.X (Continued)
(The section *Application Notes* on page 11 provides application-specific information.)

#	Problem	Overview	Impact	Workaround	Applicable to Revision?			
					2.0	2.1	2.2	2.3
22	*Cache parity checking is non-functional*	Parity checking of the L1 data cache and tag, L1 instruction cache and tag, and L2 tag array may malfunction at the rated maximum frequency of the processor. False parity errors may be reported and/or true parity errors may not be detected.	Parity checking is unreliable and therefore unusable on these processor versions. There is no impact to applications do not enable the internal cache parity checking function.	Parity checking must be disabled (HID2 bits 29,30,31 set to b'0'). This is the initial state of the processor after reset.	Yes	Yes	Yes	No
23	*L2 tag parity checking inaccurate at maximum frequency*	Parity checking of the L2 tag array may malfunction at the rated maximum frequency of the processor resulting in undetected parity errors.	No impact to applications that do not enable L2 tag parity checking (HID2 bit 31). Applications that do require L2 tag parity checking must operate the processor at a lower frequency than maximum rated frequency of the part.	Applications requiring L2 tag parity checking must operate the processor at a lower frequency than maximum rated frequency of the part. *** IMPORTANT NOTE *** Operation of the processor at or above the maximum rated frequency may result in undetected parity errors.	N/A	N/A	N/A	Yes[2]
24	*Enabling 60x data and/or address parity checking may result in a parity error*	Enabling 60x data and/or address bus parity checking may result in an erroneous parity error detected and cause either a machine check or processor checkstop condition.	An erroneous machine of checkstop can occur at the time that 60x address and/or data bus parity checking is enabled.	1) If 60x bus parity checking is to be enabled it should be done after hreset and before the L1 caches are enabled. 2) If 60x bus parity checking must be enabled after the L1 caches are already enabled: (a) the L1 caches must be disabled (b) 60x bus parity checking can be enabled (c) the L1 caches can then be re-enabled.	Yes	Yes	Yes	Yes

Definitions: Yes indicates the errata is within the specified revision.
No indicates the errata has been fixed in the specified revision.

Notes: 1. This erratum is also present in the 750, 750CX, and 750CXe.
2: The DD2.3 revision has additional workaround capability. Refer to the workaround section in the applicable erratum.

Table 2. Summary of IBM PowerPC 750FX RISC Microprocessor Errata for Revision DD2.X (Continued)
(The section *Application Notes* on page 11 provides application-specific information.)

#	Problem	Overview	Impact	Workaround	Applicable to Revision?			
					2.0	2.1	2.2	2.3
25	*L2 cache global invalidate may result in erroneous bus write*	A "global invalidate" of a L2 cache with "dirty" data can result in a burst write-with-kill to an erroneous address.	Data integrity at address 0xFFFFFXXX and/or system fails may occur due to the erroneous write. In some applications the memory controller will respond to the transaction with a $\overline{TEA}$ causing a processor checkstop or machine check.	Please refer to the full description found in *"L2 cache global invalidate may result in erroneous bus write" section on page 41.*	Yes	Yes	Yes	Yes
26	*$\overline{TEA}$ usage with miss-under-miss enabled may cause processor hang*	A Processor hang condition will occur if $\overline{TEA}$ is asserted in Miss under Miss mode.	$\overline{TEA}$ may result in a processor hang if it is asserted during a MuM request.	Don't allow assertion of $\overline{TEA}$ when MuM is enabled.	Yes	Yes	Yes	Yes
27	*HID2 Parity status bits may be incorrect when parity checking is disabled*	The cache array parity error status bits may be incorrect when parity checking is disabled. The status bits operate correctly when parity checking is enabled.	There is no impact to applications running with array parity checking enabled. Applications with parity checking disabled may experience false parity errors reported in the array parity status bits (HID2 bits 25-27). In this case, the processor will not execute a machine check or checkstop due to the false parity error.	Applications with array parity checking disabled should ignore the parity status bits (HID2 bits 25-27).	No	Yes	Yes	Yes

Definitions: **Yes** indicates the errata is within the specified revision.
No indicates the errata has been fixed in the specified revision.

Notes: 1. This erratum is also present in the 750, 750CX, and 750CXe.
2: The DD2.3 revision has additional workaround capability. Refer to the workaround section in the applicable erratum.

Application Notes

Note #1: Thermal Assist Unit (TAU)

The thermal sensor in the Thermal Assist Unit (TAU), in the DD levels covered by this errata, has not been characterized to determine the basic uncalibrated accuracy. The relationship between the actual junction temperature and the temperature indicated by THRM1 and THRM2 is not well known.

IBM recommends that the TAU in these devices be calibrated before use. Calibration methods are discussed in the IBM Application Note ***Calibrating the Thermal Assist Unit in the IBM25PPC750L Processors***. Although this note was written for the 750L, the calibration methods discussed in this document also apply to the 750FX.

IBM is planning to perform the characterization work necessary to establish the performance of the TAU. The results of this work are not expected to eliminate the calibration requirement, but will allow designers to make a more informed choice of calibration methods.

Note #2: MTMSR instruction for page address translation requires context-synchronizing instruction

Data corruption in the L1 cache can result if an instruction page address translation is enabled with a **MTMSR** that is not immediately followed by an context-synchronizing instruction. The following are context-synchronizing instructions:

- **ISYNC**
- **SC**
- **RFI**
- **RFID**

Note: The **MTMSR** and the context-synchronizing instructions work as designed. However, this application note was created to clarify that when the **MTMSR** instruction is used to enable page address translation, the next instruction must be a context-synchronizing instruction.

When enabling instruction page address translation with a **MTMSR** instruction (MSR, bit 26), there is a one-cycle window following the **MTMSR** where a **STORE** or **SYNC** can interfere with an instruction tablewalk reload into the dcache. The tablewalk reload is loaded into the data cache correctly, but the tag is not updated. Future accesses to the stale tag are seen as valid though it contains incorrect data.

This condition occurs if the **STORE** or **SYNC** following the **MTMSR** is in a new cache line that misses in the L1 icache and thus hits in the L2 cache.

Do not assume that the instructions immediately following the **MTMSR** will be address translated.

For example, this problem is experience when an **MTMSR** enables instruction address translation, followed by a **SYNC** or **STORE** instruction, a new cache line, which causes an icache miss, and then an L2 hit. In the previous example, prefetching the **SYNC**/**STORE** started before address translation was enabled and the prefetching was completed during an i-side tablewalk initiated after address translation was enabled. This caused corrupt data to appear in the L1 cache.

Data corruption in the L1 cache can occur if an instruction page address translation is enabled with a **MTMSR** that is not immediately followed by an context-synchronizing instruction.

The *PPC Programming Environments* manual strongly recommends using an **ISYNC** instruction immediately after a **MTMSR** that enables address translation. The **ISYNC** definition states that anytime an

instruction changes the address translation rules in the MMU then an **ISYNC** instruction should be executed to refetch instructions with the new translation rules. The section *Synchronization requirements for Special Registers and for lookaside Buffers*, in the *PowerPC Programming Environments* manual, specifies that a context-synchronizing instruction must be used immediately after a **MTMSR**(IR). The following are context-synchronizing instructions:

- **ISYNC**
- **SC**
- **RFI**
- **RFID**

Sample Code Sequence:

```
MTMSR RS
ISYNC
```

Erratum #1: L2 cache invalidate may fail with DPM

Overview

A “global invalidate” of the L2 cache (initiated by the L2I bit of L2CR) may not invalidate all of the L2 cache if dynamic power management (DPM bit of HID0) is also enabled.

Detailed Description

If dynamic power management is enabled (DPM=1 in HID0), a global invalidate of the L2 cache may not properly invalidate the L2 tag memory during the time that the L1’s data cache is waiting for reload data to be received from system memory. During that time, circuity in the L1 data cache is stopped to conserve power, which inadvertently affects the state machine performing the L2 global invalidate operation.

Projected Impact

System fails may occur due to stale or incorrect data in the L2 cache which should have been invalidated.

Workaround

Turn off DPM during the time that a global invalidate of the L2 cache is being performed.

Another workaround is to ensure that the processor is in a tight uninterrupted software loop monitoring the end of the global invalidate, so that the L1 data cache will never miss and initiate a reload from system memory during the global invalidate operation.

Status

A fix is not planned at the present time.

Erratum #2: dcbz that hits in L1 cache may not snoop retry

Overview

If the target address of a **dcbz** instruction hits in the IBM PowerPC 750FX RISC Microprocessor L1 cache at the same time that a snoop is received to that address, the IBM PowerPC 750FX RISC Microprocessor may not react to the snoop, and may not generate a snoop retry to the other bus master. As a result, the other bus master may continue to read the line from system memory (instead of reading IBM PowerPC 750FX RISC Microprocessor 's more recent copy of the data), resulting in stale data or a cache coherency violation.

Detailed Description

If the target address of a **dcbz** instruction hits in the IBM PowerPC 750FX RISC Microprocessor L1 cache, IBM PowerPC 750FX RISC Microprocessor will require 4 internal clock cycles to rewrite the cache line to zeros. On the 1st clock, the line is remarked as valid-unmodified, and on the last clock the line is marked as valid-modified. If a snoop request to that address is received during the middle 2 clocks of the **dcbz** operation, IBM PowerPC 750FX RISC Microprocessor will not properly react to the snoop operation or generate an address retry (via the ARTRY_ pin) to the other master. The other bus master will continue to read the data from system memory, and both IBM PowerPC 750FX RISC Microprocessor and the other bus master will end up with different copies of the data. In addition, if the other bus master has a cache, the line will end up valid in both caches which is not allowed in IBM PowerPC 750FX RISC Microprocessor 's 3-state cache environment.

Projected Impact

For data that is shared in real time, stale data and data valid in multiple caches may result causing possible system failures.

Workaround

Use of **dcbz** must be avoided for data that is shared in real time and which is not protected during writing through higher-level software synchronization protocols (such as semaphores). Use of **dcbz** must be avoided for managing semaphores themselves. (An alternative workaround could be to prevent **dcbz** from hitting in the L1 cache by performing a **dcbf** to that address beforehand.)

Status

A fix is not planned at the present time.

Erratum #3: Segment register updates may corrupt data translation

Overview

A **mtsr,mtsrin** operation followed closely by an instruction causing a data address translation using page address translation can cause contention for the segment registers. This contention causes the data address translation to proceed using data from the wrong segment register.

Detailed Description

Data page address translations, that attempt to read a segment register during the same cycle a **mtsr,mtsrin** instruction is writing to any of the segment registers, will cause the translation mechanism to receive the written data instead of the contents of the intended segment register. This can occur if there is no context synchronizing instruction between a **mtsr,mtsrin** and a succeeding data address translation that utilize any of the segment registers.

According to PowerPC Architecture, no context synchronizing instructions are required if the context of the surrounding instruction stream is unaffected by the segment register being altered by the **mtsr**.

> "If a sequence of instructions contains context-altering instructions and contains no instructions that are affected by any of the context alterations, no software synchronization is required within the sequence."
>
> "PowerPC Operating Environment Architecture Book III-AIM Version 1.07" Chapter 7, page 75.

This problem can, within specific timing windows, cause the incorrect segment data to be used for translation under these circumstances.

Instruction address and block address translations are not susceptible to this problem.

Projected Impact

Affected operations can receive incorrect data address translations resulting in access to incorrect real address locations or false translation and data access exceptions.

Workaround

A context-synchronizing instruction (for example, **isync**) should be placed between any **mtsr,mtsrin** instructions and succeeding instructions that cause a data address translation to take place utilizing any of the segment registers.

Status

A fix is not planned at the present time.

Erratum #4: Stfd of uninitialized FPR can hang part

Overview

A stfd can cause the part to hang if its source FPR has powered up in a certain state.

Detailed Description

The 64-bit FPRs each have additional internal bits associated with them that specify the type of floating point number that the register contains. These bits get properly set whenever the FPR is loaded. It is possible, however, for the part to power up with the internal bits randomly set, such that the FPR is interpreted as containing a denormalized number, but with the mantissa containing all zeroes. If this random state is stored out with an stfd before the internal bits are corrected via an FP load operation, the part will hang searching for a leading ‘1’ in the mantissa.

The stfd is the only instruction that causes this behavior.

Note that this problem was discovered when *compiled* code stored out FPRs in preparation for using them as scratch registers early in the boot sequence.

Projected Impact

This affects all systems that use floating point operations.

Workaround

Upon coming out of a Power-On-Reset (POR), initialize all of the FPRs that will be used. The value used for initialization is not important.

Status

A fix is not planned at the present time.

Erratum #5: Tablewalk store collides with snoop copyback in clock bypass mode

Overview

A tablewalk completes by doing a store to update the pte (present bit) at the same time a snoop hit in the dcache results in a copyback. If operating in clock bypass mode, both requests for the dcache collide in the dtag with data corruption.

Detailed description

The problem here is that a tablewalk store collides in the dtag with asnoop copyback. The snoop is incorrectly allowed to proceed with the copyback even though it will be retried. There is a one cycle window in bypass mode (1:1 clk ratio) where a snoop with copyback during a tablewalk reload will result in incorrect data written to the dcache.

Fail sequence:

1. Clock mode = Bypass
2. A tablewalk is in progress, reloading a PTE from the bus.
3. The PTE is a match in one of the dwords returned.
4. The TS for a snoop that hits a dirty line occurs in the cycle after the last TA for the tablewalk load.
5. The tablewalker is allowed to do its store a cycle after the last beat of data.
6. The snoop copyback incorrectly proceeds in the same cycle as the tablewalk store resulting in the snoop address being used for the tablewalk store. Data in the cache is corrupted.

Projected Impact

A data integrity problem exists when snooping, address translation with paging, and clock bypass mode are all enabled.

Workaround

Avoid using clock bypass mode with address translation (paging) enabled. If address translation is required, then use Block address translation (BAT).

Status

A fix is not planned at the present time.

Erratum #6: $\overline{\text{TEA}}$ causes processor hang or data corruption with MuM enabled

Overview

A Processor hang condition or data corruption can occur if $\overline{\text{TEA}}$ is asserted in Miss under Miss mode.

Detailed Description

Two types of failures exist when $\overline{\text{TEA}}$ is asserted with MuM.

Type 1: $\overline{\text{TEA}}$ asserted with MuM and No-Drtry mode enabled.

The L2 cache has two reload buffers which fill in a ping-pong fashion, writing current data into one while reading from the other into the L2 array. Assertion of $\overline{\text{TEA}}$ during a reload will terminate the data tenure of the current bus cycle, which also stops the reload process into the L2 reload buffer. If $\overline{\text{TEA}}$ is asserted during the first reload of a MuM (two reloads pipelined on the bus), the first reload is canceled and the buffer read pointer is advanced to the other reload buffer. However, the buffer write pointer is not advanced and a processor hang condition results because the wrong buffer is filled with the next request. The buffer to be read never fills and the other buffer fills and never clears.

Type 2: $\overline{\text{TEA}}$ asserted with MuM and No-Drtry mode disabled.

When No-Drtry mode is disabled, $\overline{\text{TEA}}$ may be asserted anytime during the data tenure. This is unlike the No-Drtry mode enabled case where $\overline{\text{TEA}}$ must be asserted before the first TA. If $\overline{\text{TEA}}$ is asserted before the first TA and during the first reload of a MuM, then the fail is a hang condition as in type 1 above. If $\overline{\text{TEA}}$ is asserted after at least one TA (of a burst of 4 TAs), and it is the first reload of a MuM, then a data corruption problem exits in the L2 cache. The internal data beat counter does not get reset with $\overline{\text{TEA}}$ and therefore signals that the reload buffer is full after receiving new beats from the next burst. This triggers a load of the reload buffer into the L2 array but part of the data will be from the previous burst. The result of the L2 corruption will not be observed until the L2 services an L1 miss. All L1 misses that were serviced from the bus will get the correct CW, even after a $\overline{\text{TEA}}$. Data corruption may result in a hang condition but may not for many instructions.

Projected Impact

If $\overline{\text{TEA}}$ is asserted during the first reload of a MuM, this may result in a processor hang or data corruption.

Workaround

Do not allow assertion of $\overline{\text{TEA}}$ when MuM is enabled.

Status

Fixed in the DD2.1 revision.

This erratum is superceded by *Erratum #26: TEA usage with miss-under-miss enabled may cause processor hang* on page 43.

Erratum #7: $\overline{\text{ARTRY}}$ signal may fail to precharge

Overview

In 2.5, 3.5, 4.5, and 5.5 PLL modes the processor may fail to precharge $\overline{\text{ARTRY}}$ while transitioning from nap mode to full power mode.

Detailed Description

The 750FX processor implementation for $\overline{\text{ARTRY}}$ deassertion is as follows:
- if asserted it remains asserted for one cycle after $\overline{\text{AACK}}$
- it is then tri-stated for 1/2 bus cycle
- it is then precharged (driven to one) for 1 bus cycle
- it is then tri-stated

The processor will fail to precharge $\overline{\text{ARTRY}}$ if:
- the PLL configuration is set to any of the following modes: 2.5, 3.5, 4.5 or 5.5
- the processor has entered nap state (both $\overline{\text{QREQ}}$ and $\overline{\text{QACK}}$ asserted)
- $\overline{\text{QACK}}$ is deasserted
- a snoop to the processor hits in a cache requiring an assertion of $\overline{\text{ARTRY}}$
- an external or decrementer interrupt occurs 2 bus cycles before the AACK for the snoop

Projected Impact

The processor and memory controller (and any other component observing the 60x bus) will observe $\overline{\text{ARTRY}}$ asserted on bus transactions and as $\overline{\text{ARTRY}}$ slowly charges the processor and memory controller may lose synchronization and hang and/or lose data integrity.

Workaround

1. Avoid using 2.5, 3.5, 4.5, 5.5 PLL configuration modes.
2. Avoid using nap mode. Use doze mode instead.
3. Implement an external circuit that prechargres $\overline{\text{ARTRY}}$ for one cycle 2 cycles after AACK.

Status

Fixed in the DD2.1 revision.

Erratum #8: Nap mode is non-functional at frequencies above 400MHz

Overview

Nap mode is non-functional at frequencies above 400MHz.

Detailed Description

Internal clocks within the processor are turned off when in nap mode to conserve power. An internal control signal is generated to wake-up the processor (for example, interrupt, snoop, and so forth) that activates the clocks. The control signal unexpectedly truncates the first clock cycle upon wake-up.

Projected Impact

Unpredictable results if nap mode is used at frequencies greater than 400MHz

Workaround

Use doze mode for power savings or use nap mode at 400MHz.

Status

Fixed in the DD2.3 revision.

Erratum #9: Double snoop push

Overview

Snooping may result with two castouts performed to the same address. The first castout may have stale data, but the second castout will have the most recent data.

Detailed Description

For the double snoop push to occur, the following sequence of events must occur.

1. Cache line “A” is dirty (modified with respect to system memory) in both the L1 data cache and in the L2 cache. This can occur as a result of normal tag re-allocations in the L1 data cache.
2. In the L2 cache, the cache tag for cache line “A” is then reallocated for another address as a result of normal reloading of instruction fetches or load/store operations. As a result, cache line “A” and at least one other dirty sector for that cache tag is sent from the L2 cache to the bus unit to be written out to memory. During this time, cache line “A” is still valid and modified in the L1 cache.
3. Another bus master then performs a bus transaction which is snooped and hits for cache line “A”. Internally, the snoop hits in both the L1 cache and in the bus unit’s L2 castout buffer. The correct response in this case would be to cancel the L2 castout in the bus unit and push the line from the L1 cache (which contains the most modified data). However, the presence of more than one sector in the L2 castout buffer results in incorrect behavior. The L2 castout buffer is pushed to the bus, and then the same cache line is again pushed (but with newer data) from the L1 cache.

   The result is that after the snoop appears to have completed on the bus, the snoop push buffer in the bus unit still contains the most modified cache line from the L1 cache, giving the appearance that the processor missed the snoop “window of opportunity”. This also blocks the snoop buffer from being available for an immediately subsequent snoop operation.

Projected Impact

If the second snoop push is not allowed to be performed before the original bus master that triggered the snoop is allowed to access system memory, the original bus master will then read stale data from memory instead of the latest data which is still in the processors snoop push buffer. This may result in data corruption in the system.

Also, if a new snoop triggers a snoop push from the processor while the snoop buffer still contains the previous uncompleted 2nd snoop push, the processor will respond to the new snoop by attempting to push the previous snoop address, in order to clear its buffer. If the bus master generating the new snoop does not allow this “different” address to be pushed, but rather keeps retrying it until the expected new snoop address is pushed instead, a system “livelock” may occur where the new snoop is never completed (usually requiring the system to be reset).

Workaround

Use one of the following workarounds:

1. Allow both snoop pushes to be performed before continuing the alternate master that triggered the snoop. In an MPC106-based system (Grackle) bridge chip, this can be accomplished by setting the “CF_LOOP_SNOOP” bit in the MPC106’s configuration registers.
2. Prevent dirty data from residing in the processor’s L2 cache. This can be achieved by operating the L2 cache in write-through mode by setting the L2WT bit in its L2 configuration register. This workaround works in any system configuration. In systems employing a 750FX, the MPC106, and a DMA device on the 60x bus which can generate snoops to the 750, a more narrow workaround could be to operate only those pages addressable by the DMA device in write-through (or cache-inhibit), while still setting the CF_LOOP_SNOOP bit in the MPC106.

Note: In Systems where all snoopable memory (for example, all system memory) is accessed directly and only over 60x bus, the occurrence of a double snoop push would not cause a failure, and a work-around would normally not be required. In this case, the 2 snoop pushes would get an opportunity to be performed before the alternate master continues on the 60x bus due to normal 60x bus retry protocols.

Status

Fixed in the DD2.1 revision.

Erratum #10: SRR0 can become corrupted when Performance Monitor/Decrementer are enabled

Overview

A Performance Monitor Interrupt, taken immediately after a delayed Decrementer exception, can cause SRR0 to become corrupted.

Detailed Description

A decrementer exception is signaled while MSR[EE]=0. The performance monitor is also enabled and counting events during this time. Sometime later, MSR[EE] gets set to “1” and the delayed decrementer exception starts to be taken. At the same time, a performance monitor interrupt is taken, which is a higher priority interrupt. One core cycle after the DEC exception is started, the PM interrupt is taken instead. SRR0 gets set to 0x00000900 - but the DEC exception handler never started, so no state was saved. At this point, the PM exception has become unrecoverable because it would return to the 0x900 handler and not be able to return to the normal program flow after that point.

Projected Impact

The processor hangs.

Workaround

When performance monitor interrupt signalling is enabled (MMCR0[ENINT] = 1), avoid decrementer interrupts by periodically setting the Decrementer register value to 0x7FFFFFFF. Performance monitor interrupts caused by Time Base bit transitions can be used to mediate this periodic setting of the Decrementer and in some cases to roughly simulate decrementer functionality during this time.

Status

Fixed in the DD2.3 revision.

Erratum #11: Checkstop on parity error

Overview

The processor will always checkstop on an I tag or L2 tag parity error even if the MSR ME bit is set.

Detailed Description

When a parity error occurs in the I Tag or the L2 tag, the parity error signal is active for more than one cycle. If the MSR ME bit is set, the processor will attempt to go to the machine check interrupt vector. On the way to the interrupt vector, the MSR ME bit is cleared by hardware. If the parity error signal is still active at that time, the processor will then checkstop and never reach the machine check interrupt vector.

Projected Impact

The processor will always checkstop on an I tag or L2 tag parity error.

Workaround

None.

Status

Fixed in the DD2.1 revision.

Erratum #12: DPM is non-functional

Overview

Dynamic Power Management (DPM) is incompatible with the data cache operation.

Detailed Description

Enabling the dynamic power management mode using the DPM bit in the HID0 register causes the data cache to be clocked improperly. Consequently, the data cache does not function.

Projected Impact

The added benefit of DPM power savings can not be realized.

Workaround

DPM should be disabled (HID0 bit 11 set to b”0”).

Status

Fixed in the DD2.1 revision.

Erratum #13: BAT usage restrictions

Overview

Usage of the extended IBAT[4:7] and DBAT[4:7] with either 128k block sizes or uninitialized contents may result in incorrect block address translation.

Detailed Description

The 750FX processor added 4 IBAT and DBAT registers for additional block translation capability. The BAT registers provide support for block sizes of 128KB, 256KB, 512KB, ..., 256MB. Due to a circuit timing sensitivity in DD1.x, usage of these additional BAT registers must be limited. The 128K byte block size can not be used for IBAT[4:7] and DBAT[4:7]. Also, IBATU[4:7] and DBATU[4:7] bits 14 and 29 (BEPI(14) and BL(10)) must be set to a b'1' prior to enabling block address translation and held constant thereafter.

Projected Impact

Block sizes are limited to 256KB-256MB on the 4 additional IBAT and DBAT registers. PowerPC 750 legacy applications that use 4 or less BAT registers are unaffected with respect to functionality. However, IBAT[4:7] and DBAT[4:7] must be initialized and held constant as described in the detailed description.

Workaround

Block sizes of 128KB cannot be used in the 4 additional BAT registers: IBAT[4:7] and DBAT[4:7]. The bits BEPI(14) and BL(10) must be set to a b'1' in IBATU[4:7] and DBATU[4:7] prior to enabling block address translation and must remain in this state thereafter. This workaround applies regardless of using IBAT[4:7] or DBAT[4:7].

Status

Fixed in the DD2.1 revision.

Erratum #14: Snooping may cause system hang with MuM and DBWO

Overview

A Processor bus hang condition can occur if DBWO is required to force a snoop castout in Miss-under-Miss (MuM) mode.

Detailed Description

The 750FX supports a maximum of 2 data tenures on the 60x bus. If MuM is enabled, then both data tenures may be occupied doing data reloads for load or store misses. If a snoop castout is required, at least one of the reloads must be serviced before the castout may be pushed. In systems using DBWO, this may cause a hang condition if the memory controller can not service one of the outstanding reloads until the snoop castout is complete. A bus hang results because the snoop castout can not be pushed ahead of the two outstanding reloads.

Projected Impact

MuM may result in a bus hang if DBWO is required to resolve system deadlock conditions.

Workaround

For the DD2.0 - DD2.2 revisions, systems requiring DBWO to push snoop castouts should disable MuM prior to the DBWO operation by setting HID0[14] to a zero (0).

DD2.3 has new hardware to support another solution that will allow DBWO when MuM is enabled. Systems that will only deadlock when accessing specific memory locations (such as io device space), may mark the space as guarded. Guarded loads will be limited to one data queue, reserving the other data queue for snoop castouts if needed. The MuM feature is dynamically disabled for all guarded loads, including instruction loads that might pipeline into a guarded load. This feature to turn off all pipelining of guarded loads is enabled by setting HID2[8] to a one.

Status

A workaround is available for DD2.3 and is defined in the above section.

A fix is not planned at this time.

Erratum #15: 3.3V I/O mode is non-functional

Overview

The I/O devices on 750FX do not function with the processor configured in 3.3V I/O mode (BVSEL=b'1', L1_TSTCLK=b'0').

Detailed Description

When 3.3V mode is selected, an internal voltage bias circuit is non-functional. As a result, the I/O devices are unable to drive a low voltage level or receive a high voltage.

Projected Impact

The 3.3V mode is not usable in production-level systems. However, for system prototyping and bring-up, the following workaround may be used.

Workaround

The processor can be interfaced with 3.3V I/O systems for limited usage. Connect OV_{DD} to 3.3V (3.3V max.) and configure the processor for 2.5V I/O mode (BVSEL=b’1’, L1_TSTCLK=b’1’).

To avoid reliability issues, restrict the usage of 3.3V I/O systems to less than 6000 power-on hours (POH) in this mode.

Status

Fixed in the DD2.1 revision.

Erratum #16: LWARX or DCBT/ST causes data corruption with MuM enabled

Overview

Store data may be lost when MuM is enabled and a **LWARX** or **DCBT/ST** instruction aliases a store in the store queue.

Detailed Description

The Load/Store unit store queue may stall due to changes in instruction flow or snooping activity to the L1 data cache. During this stall period with MuM enabled, if a **LWARX** or **DCBT/ST** instruction is dispatched to the L/S unit, and it is a miss in the L1 cache, and it has the same cache index as a store in the store queue, then the condition for fail is set up. The leading store will incorrectly advance out of the store queue, and it's data will never update the L1 cache.

Sample Code Sequence

```
lfd    miss
..
stswi  leading store in store queue
stfd   index A in store queue
..
lwarx  miss with index A
```

Projected Impact

A data integrity problem or operating system hang may result.

Workaround

Turn off MuM if **LWARX** or touch instructions are used.

Status

Fixed in the DD2.1 revision.

Erratum #17: BTIC low voltage sensitivity

Overview

For DD2.0 and DD2.1, the BTIC fails below 1.45V.

For DD2.2 and DD2.3, the BTIC fails below 1.4V.

Detailed Description

The BTIC displays a much higher sensitivity to reduced voltage. Its Voltage/Frequency curve is flatter than expected at lower voltages, with the frequency (Fmax) dropping off very quickly.

Projected Impact

For DD2.0 and DD2.1, the BTIC will fail below 1.45V causing an incorrect or illegal instruction to be executed.

For DD2.2 and DD2.3, the BTIC will fail below 1.4V causing an incorrect or illegal instruction to be executed.

Workaround

For DD2.0 and DD2.1,turn off the BTIC for voltages less than 1.45V (HID0 bit 26 = zero).

For DD2.2 and DD2.3, turn off the BTIC for voltages less than 1.4V (HID0 bit 26 = zero).

Status

A fix is not planned at this time.

Erratum #18: PLL voltage sensitivity

Overview

When V_{DD} is less than 1.35V, the phase-locked loop (PLL) circuit may not align the processor internal clock edges with the external clock (for example, SYSCLK), thus the processor may fail to meet I/O timing specifications or become non-functional.

Detailed Description

The PLL circuit contains a pass-gate circuit controlled by V_{DD} which is slow at low voltage (V_{DD} <1.35V). This creates an undesired offset in the alignment of internal, PLL-generated clock edges to the external clock (SYSCLK). I/O timing specifications (for example, input setup, output valid, and so forth) require tight tolerance between the internal clock edges and the external clock. Consequently, the processor is unable to meet I/O timing specifications.

Projected Impact

Operation of this revision of the processor at V_{DD} less than 1.35V may cause a shift in the processor I/O timing specifications and result in a system malfunction. The amount of I/O timing shift increases as voltage is reduced from 1.35V. The I/O timing remains constant when V_{DD} is held constant.

Workaround

Operate the processor at V_{DD} of 1.35V or greater.

Status

Fixed in the DD2.1 revision.

Erratum #19: Dual-PLL switch may cause truncated internal clock pulses and processor malfunction

Overview

Using the Dual-PLL feature to switch between PLL0 and PLL1 may result in truncated internal clock pulses and processor malfunction. *Table 3* provides a list of the part numbers and revisions affected by this erratum.

Table 3. Part Numbers and Revisions Affected by Erratum #19

750FX Revisions	
DD2.0 Part Numbers	DD2.1 Part Numbers
All Part Numbers	IBM25PPC750FX-EB0113T
	IBM25PPC750FX-EB0513T
	IBM25PPC750FX-EB1013T

Detailed Description

The switch from PLL0 to PLL1 or from PLL1 to PLL0 typically occurs when both PLLs are either driving a b'1' or b'0'. Though the voltage, temperature, and frequency operating conditions are within the specification, the internal signals controlling the switch may be delayed. This delay may allow the source PLL to switch to the opposite value of the target PLL which can cause an anomaly on the clock network. This anomaly causes the processor circuits to malfunction with unpredictable results.

Projected Impact

Use of the Dual-PLL feature may result in unpredictable results.

Workaround

Avoid using the Dual-PLL feature for the part numbers and revisions listed in *Table 3*.

Status

Fixed in the DD2.2 revision.

Erratum #20: Dual PLL switch at 4.5X mode may result in loss of $\overline{\text{ARTRY}}$ precharge

Overview

$\overline{\text{ARTRY}}$ may not precharge after Dual PLL switching at 4.5X mode during snoop activity.

Detailed Description

The following is the 750FX processor implementation for $\overline{\text{ARTRY}}$ assertion:

1. When asserted it remains asserted for one cycle after $\overline{\text{AACK}}$.
2. Tristated for 1/2 bus cycle.
3. Precharged (driven to one) for one bus cycle.
4. Tristated.

The processor does not precharge $\overline{\text{ARTRY}}$ if it coincides with a PLL switch from 4.5X mode to any other PLL configuration mode.

Projected Impact

In the absence of an $\overline{\text{ARTRY}}$ precharge cycle, the processor and memory controller, and any other component observing the 60x bus, will observe $\overline{\text{ARTRY}}$ asserted on bus transactions. As $\overline{\text{ARTRY}}$ slowly charges, the processor and memory controller may lose synchronization which may cause the processor to hang and/or data integrity issues.

Workaround

1. Avoid using 4.5X mode with Dual PLL operation.

   Note: 4.5X mode with a single PLL is not affected.

2. Implement an external circuit that precharges $\overline{\text{ARTRY}}$.
   Two cycles after $\overline{\text{AACK}}$, $\overline{\text{ARTRY}}$ is precharged for one cycle by the external circuit.

Status

A fix is not planned at this time.

Erratum #21: MuM can cause invalid data with mispredicted branches.

Overview

Data from a cancelled load instruction may incorrectly be forwarded to a subsequent load miss, if MuM is enabled.

Detailed Description

A MuM load in a mispredicted branch instruction stream may incorrectly forward it's CW (critical word from the bus to a load miss in the branch target stream. When the branch is finally resolved, the MuM load should be cancelled but isn't. The data returned from the bus is not ignored, but used for a new load miss. Several events must line up for the failing condition to occur.

1. A Load miss starts on the bus, but gets cancelled because of a mispredicted branch. Bus data will be ignored for this cancelled load.

2. In the new instruction stream, another load miss occurs after a conditional branch.

3. The branch resolution must be stalled at least 10 cycles, allowing MuM to start on the bus. Two outstanding load misses are now in progress on the bus. The first one is in a cancelled state.

4. The branch resolves and the MuM load is cancelled because it is in a mispredicted stream.

5. Incorrect data handling will occur if another load miss in the new instruction stream is pending immediately after the last beat of the first cancelled miss. If no load is pending, then the MuM on the bus is correctly cancelled and the CW ignored.

Example code stream:

```
bc(target A)  mispredicted branch
lwz           First load on bus cancelled

A:mulresult   stalls branch resolution
cmpi
bc (target B) 2'nd mispredicted branch.
lfd           MuM starts on bus under first cancelled load.

B:lwz         load miss gets MuM cancelled data.
```

Projected Impact

When Load MuM is enabled, incorrect data transfer may result.

Workaround

Disable Load MuM with Hid2 bit2 = b'1', or disable entire MuM with Hid0 bit14 = b'0'.

Status

Fixed in the DD2.3 revision.

Erratum #22: Cache parity checking is non-functional

Overview

Parity checking of the L1 data cache and tag, L1 instruction cache and tag, and L2 tag array may malfunction at the rated maximum frequency of the processor. False parity errors may be reported and/or true parity errors may not be detected.

Detailed Description

Parity checking for the internal arrays requires additional time compared to the normal read function for these arrays. The parity checking function cannot be completed reliably at the rated frequency of the processor. The result is false parity errors or undetected true parity errors.

Projected Impact

Parity checking is unreliable and therefore unusable on these processor versions. There is no impact to applications that do not enable the internal cache parity checking function.

Workaround

Parity checking must be disabled (HID2 bits 29,30,31 set to b'0'). This is the initial state of the processor after reset.

Note: There is no frequency derating curve at which parity checking is supported.

Status

Fixed in the DD2.3 revision.

Erratum #23: L2 tag parity checking inaccurate at maximum frequency

Overview

For DD2.3, parity checking of the L2 tag array may malfunction at the rated maximum frequency of the processor resulting in undetected parity errors.

Detailed Description

The parity checking function of the L2 tag array requires additional time compared to the data read function for this array. Also, the output of the parity checking circuit initializes to good parity every processor cycle and then evaluates to indicate a parity error if one is encountered. Consequently, when operating the processor at or above its maximum operating frequency, the parity checking circuit initialization value may be captured to always indicate good parity regardless of a true parity error.

Projected Impact

No impact to applications that do not enable L2 tag parity checking (HID2 bit 31).

For DD2.3, applications that do require L2 tag parity checking must operate the processor at a lower frequency than maximum rated frequency of the part.

Workaround

For DD2.3, applications requiring L2 tag parity checking must operate the processor at a lower frequency than the maximum rated frequency of the part.

- **Important note:** Operation of the processor at or above the maximum rated frequency may result in undetected parity errors (see detailed description above).
- Refer to the 750FX derating chart below for maximum frequency with the L2 tag parity disable/enabled.

Status

A workaround is available for DD2.3 and is defined in the above section.

A fix is not planned at this time.

750FX Derating Chart of Maximum Frequency with L2 tag Parity Disabled/Enabled

IBM PowerPC 750FX DD2.3 Internal Part Number Matrix								
	70P4668		70P4669		70P4670		70P4671	
	70P4672		70P4673		70P4674		70P4675	
VDD min	Max Freq w/ L2 tag parity disabled (MHz)	Max Freq w/ L2 tag parity enabled (MHz)	Max Freq w/ L2 tag parity disabled (MHz)	Max Freq w/ L2 tag parity enabled (MHz)	Max Freq w/ L2 tag parity disabled (MHz)	Max Freq w/ L2 tag parity enabled (MHz)	Max Freq w/ L2 tag parity disabled (MHz)	Max Freq w/ L2 tag parity enabled (MHz)
1.4V	800	700	733	650	700	600	600	533
1.35V	750	667	700	600	650	550	600	533
1.3V	700	600	650	550	600	533	566	500
1.25V	667	600	600	533	566	500	533	450
1.2V	600	533	566	500	533	450	500	400

Erratum #24: Enabling 60x data and/or address bus parity checking may result in a parity error

Overview

Enabling 60x data and/or address bus parity checking with either the L1 Icache and/or Dcache enabled may result in an erroneous parity error detected and cause either a machine check or processor checkstop condition.

Detailed Description

Enabling 60x address and/or data bus parity checking via hid0[EBA] and/or hid0[EBD] with either the L1 Icache and/or Dcache enabled may result in a parity error detected. The error may result in either a checkstop or machine check depending on the value of MSR[ME]. The exposure is only at the time that HID0[EBA] and/or HID0[EBD] transition from zero to one. Once enabled there is no exposure of getting this erroneous error at a later time.

Projected Impact

An erroneous machine check or checkstop condition can occur at the time that 60x address and/or data bus parity checking is enabled.

Workaround

1) If 60x bus parity checking is to be enabled it should be done after hreset and before the L1 caches are enabled.
2) If 60x bus parity checking must be enabled after the L1 caches are already enabled:

    a) the L1 caches must be disabled
    b) 60x bus parity checking can be enabled
    c) the L1 caches can then be re-enabled

Status

A fix is not planned at this time.

Erratum #25: L2 cache global invalidate may result in erroneous bus write

Overview

A "global invalidate" of the L2 cache with "dirty" data can result in a burst write-with-kill to an erroneous address.

Detailed Description

It is possible for the processor to issue an erroneous burst write-with-kill transaction to address 0xFFFFFXXX. This erroneous transaction can occur only if:

1) The L1 Icache is enabled.
2) Code is executed that disables and invalidates the L2 Cache
3) The L2 cache had dirty or modified data not yet written to the 60x bus.

The erroneous transaction can occur when a code stream that disables and invalidates the L2 cache executes. An instruction pre-fetch just before the L2 Cache is disabled allocates in the L2 Cache replacing a modified line and causes a castout to be queued. If the L2 Cache invalidate starts before the castout address phase is started on the 60x bus the castout address will get corrupted.

Projected Impact

Data integrity at address 0xFFFFFXXX and/or system fails may occur due to the erroneous write. In some applications the memory controller will respond to the transaction with a $\overline{\text{TEA}}$ causing a processor checkstop or machine check.

Workaround

It is always recommended to flush the L2 cache prior to disabling and invalidating it to prevent loss of modified data. The flush operation is described the processor user manual section 9.1.1.

To avoid the overhead of the flush operation when the modified data in the L2 Cache can be ignored another option is to disable the L1 Icache and then disable and invalidate the L2 as follows:

```
isync
hid0(ice)<-0    /* disable icache */
isync
 ....
l2cr(l2e)<-0    /* disable L2 Cache */
 ....
```

```
l2cr(l2i)<-1    /* invalidate L2 Cache */
...
isync
hid0(ice)<-1    /* re-enable icache if needed */
isync
 ....
wait for l2cr(ip)==0    /* wait for L2 Cache invalidate */
 ...
```

This will ensure that a pending L2 castout has been serviced by the 60x bus. If the L1 dcache is enabled, the user must also disable the L1 dcache prior to the L2 disable and invalidation.

Status

A fix is not planned at this time.

Erratum #26: $\overline{\text{TEA}}$ usage with miss-under-miss enabled may cause processor hang

Overview

A Processor hang condition will occur if $\overline{\text{TEA}}$ is asserted in Miss under Miss mode.

Detailed Description

When MuM is enabled, another request that also misses in the data cache is allowed to proceed to the L2 and bus interface while a current miss is in progress. If $\overline{\text{TEA}}$ is asserted after a MuM has started internally, then the processor will hang. The $\overline{\text{TEA}}$ correctly terminates the current request, but also terminates the MuM request. This is acceptable if a checkstop is enabled, but not for a machine check interrupt which expects to vector to an interrupt handler. No further bus transactions will proceed out of the processor and the machine check interrupt will not be taken because of the hang condition.

Projected Impact

$\overline{\text{TEA}}$ may result in a processor hang if it is asserted during a MuM request.

Workaround

Do not allow assertion of $\overline{\text{TEA}}$ when MuM is enabled.

Status

A fix is not planned at this time.

This erratum supercedes *Erratum #6: TEA causes processor hang or data corruption with MuM enabled* on page 18.

Erratum #27: HID2 parity status bits may be incorrect when parity checking is disabled

Overview

The cache array parity error status bits may be incorrect when parity checking is disabled. The status bits operate correctly when parity checking is enabled.

Detailed Description

The parity error status bits in the HID2 register (bits 25-27) are updated when either a) a parity error occurs or b) the register is written via instructions. Data placed in the register is either a) parity error status if parity checking is enabled or b) data from an internal bus such as that sourced from the general purpose registers. In the error case, a parity error is detected (independent of parity checking enabled or disabled) and the register is updated with the contents of the internal bus. Array parity checking is enabled or disabled using HID2 register bits 29-31. Only the status bits in HID2 (bits 25-27) are invalid when parity checking is disabled. The indication of a parity error in these bits will not cause a machine check or checkstop when array parity checking is disabled.

Projected Impact

There is no impact to applications running with array parity checking enabled. Applications with parity checking disabled may experience false parity errors reported in the array parity status bits (HID2 bits 25-27). In this case, the processor will not execute a machine check or checkstop due to the false parity error.

Workaround

Applications with array parity checking disabled should ignore the parity status bits (HID2 bits 25-27).

Status

A fix is not planned at this time.

Revision Log

Rev	Contents of Modification
March 20, 2002	Version (0.1) • Initial preliminary version. • The DD2.0 revision fixed the Erratum #2, #6, and #8 listed in the PowerPC 750FX RISC Microprocessor Errata List for DD1.1-1.2. For additional information, see *Errata Fixed in DD2.0* on page 3. • Added the following Errata: - Added *Checkstop on parity error* - Added *DPM is non-functional*
June 28, 2002	Version (0.2) • Preliminary version. • Expanded errata listing to include DD 2.X. • Updated *DCBZ that hits in L1 cache may not snoop retry.* • Removed *Transferring data from BAT to GPR registers results in incorrect data* because it was fixed. • Added the following Errata: - *BAT usage restrictions* - *Snooping may cause system hang with MuM and DBWO* - *3.3V I/O mode is non-functional* - *LWARX or DCBT/ST causes data corruption with Mum Enabled* - *MTMSR instruction for page address translation requires context-synchronizing instruction*
July 15, 2002	Version (0.3) • Preliminary version. • Added *PLL voltage sensitivity.*
September 3, 2002	Version (0.4) • Preliminary version. • Errata numbering has changed from previous document versions. • Updated the *Table 2. Summary of IBM PowerPC 750FX RISC Microprocessor Errata for Revision DD2.X*. Added the DD2.2 status and updated the status of all the errata to incorporate DD2.2. • Added *Application Awareness Note*s section on page 3. Added note to describe issue and workaround for using the thermal sensor in the Thermal Assist Unit (TAU). Moved *Erratum, MTMSR instruction for page address translation requires context-synchronizing instruction* to this section and deleted this erratum from *Table 2. Summary of IBM PowerPC 750FX RISC Microprocessor Errata for Revision DD2.X*. • Added *BTIC low voltage sensitivity.* (from IBM PowerPC 750FX RISC Microprocessor Errata for DD1.0, and DD1.1-DD1.2). • Added *Dual-PLL switch may cause truncated internal clock pulses and processor malfunction.* • DD2.2 fixed *Dual-PLL switch may cause truncated internal clock pulses and processor malfunction*. • Changed the status for *SRR0 can become corrupted when Performance Monitor/Decrementer are enabled* and *Snooping may cause system hang with MuM and DBWO*, • Incorporated general edits that included updates to the legal text • Updated the Preface. Added PVR for DD2.2 to*Table 1. 750FX Processor Version Register (PVR)* on page 3. • Removed *BTIC low voltage sensitivity* from the list of errata that were fixed for DD2.0. • Replaced the word *solution* with *workaround*.
October 2, 2002	Version (0.5) • Preliminary version. • Added *Dual PLL switch at 4.5X mode may result in loss of ARTRY precharge.* • Added a note to the *Preface* and a table description to the *Summary Table* that describes and links to the *Application Notes* section.
October 25, 2002	Version (0.6) • Preliminary version. • Added *Cache Parity checking sensitivity to frequency* • Added *MuM can cause invalid data with mispredicted branches* • Updated table to include DD2.3 Revision • Updated status of fixes planned in DD2.3 Revision

Rev	Contents of Modification
February 5, 2003	Version (0.7) • Preliminary version. • Updated the PVR note for DD2.3, in *Table 1. 750FX Processor Version Register (PVR)* on page 3, to read “0x700002X3, where X is reserved”. • Updated “Applicable to revision” status of DD2.3 to “Yes” for *Nap mode is non-functional at frequencies above 400MHz* . • Added note 2 regarding DD2.3 supporting a specific application of MuM and DBWO for *Snooping may cause system hang with MuM and DBWO* . • Updated “Applicable to revision” status of DD2.3 to “Yes” for *Dual PLL switch at 4.5X mode may result in loss of ARTRY precharge* . • Updated “Applicable to revision” status of DD2.3 to “Yes” for *Cache Parity checking sensitivity to frequency.* • Updated status of *Nap mode is non-functional at frequencies above 400MHz* . • Updated status of *Snooping may cause system hang with MuM and DBWO* . • Updated status of *Dual PLL switch at 4.5X mode may result in loss of ARTRY precharge* . • Updated status of *Cache Parity checking sensitivity to frequency.* • Updated status of *MuM can cause invalid data with mispredicted branches.* . • Updated status of fixes in DD2.3 Revision.
May 15, 2003	Version (0.8) • Preliminary version. • Updated *Table 1. 750FX Processor Version Register (PVR)* as follows: - Changed the next to last nibble to 'X' for all the PVRs. - Added the note “X is to be ignored and is for factory use only”. • Updated status of *Nap mode non-functional above 400MHz* to “fixed in DD2.3 revision”. • Added description of DD2.3 workaround for *Snooping may cause system hang with MuM and DBWO.* • For DD2.2 and DD2.3, the minimum operating voltage was changed from 1.45V to 1.4V in *BTIC low voltage sensitivity.* • Replaced *Cache Parity checking sensitivity to frequency* erratum with 2 separate errata: - *Cache parity checking is non-functional* - *L2 tag parity checking inaccurate at maximum frequency* • Added the following errata: - *Enabling 60x data and/or address parity checking may result in a parity error* - *L2 cache global invalidate may result in erroneous bus* - *$\overline{TEA}$ usage with miss-under-miss enabled may cause processor hang* - *HID2 parity status bits may be incorrect when parity checking is disabled*