Document Number: MPC823CE
Rev. 3, 05/2005

Device Errata for the MPC823 Silicon Revision B.2—Mask Set 3H97G, 1K24A and 1K29A

These errata apply to the MPC823 Revision B.2 (CDR2) silicon. Changes to this errata are in *italics*.

Table 1 provides a revision history for this document.

Table 1. Document Revision History

Revision	Date	Significant Changes
Rev.	5/2005	• Updated template. • Added CPU16 errata.

Table 2 summarizes all known errata for the MPC823 and whether there is a work around.

Table 2. Summary of MPC823 Silicon Errata

Number	Errata Name	Work Arounds?
CPU		
CPU1	Bus Error Not Fully Supported by the Data Cache on a Burst	Yes
CPU2	Incorrect Data Breakpoint Detection on Store Instructions	None
CPU3	Program Trace Mechanism Error	None
CPU4	Case of I-Cache Using Address of Old Page When Fetching New Page	Yes
CPU14	RCCR and RMDS Should be Written to as a Single 32-bit Register	Yes
CPU16	*In Device Mode, an Endpoint Might Lock Up After the Reception of an IN Token*	*Yes*
General		
G1	Core Operation Is Limited to a 3.0V Minimum	None
G2	The External Bus Transaction May Hang After a PLPRCR Write Access	Yes

Table 2. Summary of MPC823 Silicon Errata (continued)

Number	Errata Name	Work Arounds?
G3	LCD Off and On Sequence With a Pending SDMA Cycle Causes Wrong Data Fetch	Yes
G6	LCD Controller Off Sequence When LAM Bit Is Set May Cause the CPU to Lock Up	Yes
G7	Possible External Bus Hang Occurs Under Certain Error Conditions	None
G8	Higher Than Normal Current Consumption Without Executing a MULLW Instruction	Yes
G12	Incorrect Reporting of Loss-of-Lock Reset Status	Yes
G13	Conflict Between Data Show Cycles and SDMA Burst Writes	Yes
G14	CPU Receives A Machine Check After Writing to the PLPRCR	Yes
G15	Address Bus May Consume Excess Current in Low Power Modes	Fixed in Rev. B
G16	Open Drain Pins Do Not Allow 5 Volt Pullups	None

CPU ERRATA

CPU1 Bus Error Not Fully Supported by the Data Cache on a Burst

Description:

The data cache does not support a bus error that might occur on the second or third data beat of a burst. (burt_232).

Work Arounds:

Avoid using a bus error in this case.

CPU2 Incorrect Data Breakpoint Detection on Store Instructions

Description:

When a breakpoint on data occurs and you have programmed the size elements as byte or half-word, the following may occur:

- A breakpoint might be detected when it should not
- A breakpoint might not be detected when it should

Either of these two cases can occur if the data that is programmed to be detected, matches some other portion of the register that is currently stored to memory by the store byte or store half-word instruction.

For example:

- Assume that you have programmed a byte data breakpoint on a store instruction and you are looking for the byte element 0x55. Assume that register R1 has the value 0x00080000, R10 has the value 0x55443322, and the stb R10,0x3(R1) store instruction is performed.
- What occurs is that byte 0x22 from R10 is stored to address 0x00080003, and this should not generate a breakpoint since 0x22 does not equal 0x55, but, in some cases, it can and does (in this scenario, R10 does include the data 0x55). The result is a breakpoint is executed when it should not be.
- Assume that in the above case you are programming for byte element 0x22, maybe a breakpoint condition will not be detected, even though it should. (burt_246).

NOTE

These fault cases depend on the previous Load-Store instruction address. If the previous Load-Store instruction address' LSB is different from the current instruction address' LSB, then an incorrect breakpoint detection might occur.

Work Arounds:

None.

CPU3 Program Trace Mechanism Error

Description:

In the following cases, there is an error in the program trace mechanism:

- 0x00004ff0: divw. r25,r27,r26
- 0x00004ff4: divw. r28,r27,r26
- 0x00004ff8: unimplemented
- 0x00004ffc: b 0x00005010
- and where: 0x00005010 belongs to a page where a page fault occurs

The divide takes a relatively long time, so the instruction queue gets filled with the unimplemented instruction, which is the branch and the branch target (page fault). When the sequencer takes the unimplemented instruction, it releases the fetch that was blocked by the mmu error, which causes the queue to get another instruction in addition to the first page fault. Because the second fault is sequential to the branch target, it is not reported by the queue flush (VF) and this causes an incorrect value to be present in the VF flush information when the unimplemented exception occurs. (burt_251)

Work Arounds:

None.

CPU4 Case of I-Cache Using Address of Old Page When Fetching New Page

Description:

The Instruction Cache uses the address associated with the old page when fetching the first data from a new page, under the following circumstances:

1. There is a show cycle on a sequential instruction which crosses the page boundary.
2. The internal bus is busy during the IMMU request with the old page number.

Thus, on the next cycle the I-Cache uses this incorrect address to access the external memory and internal cache. (burt_285)

Work Arounds:

Do not run in “show all” mode or do not put a sequential instruction in the last address of an MMU page.

CPU14 RCCR and RMDS Should be Written to as a Single 32-bit Register

Description:

The ERAM4K bit is cleared in the RISC Microcode Development Support Control Register, RMDS, if the register's location is accessed as either part of a half-word or byte access.

Work Arounds:

If the ERAM4K is to be set, the RMDS must be accessed as part of a word starting at IMMR+0x9C4 to IMMR+0x9C7. It is best to write to these two registers as if they were a single 32-bit register. See register table below.

RCCR-RMDS

BIT	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
FIELD	TIME	RES	TIMEP						DR2M	DR1M	DRQP		EIE	SCD	ERAM	
RESET	0	0	0						0	0	0		0	0	0	
R/W	R/W	R/W	R/W						R/W	R/W	R/W		R/W	R/W	R/W	
ADDR	(IMMR & 0xFFFF0000) + 0x9c4															

	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
FIELD	Reserved								ERAM4K	Reserved						
RESET	0								0	0						
R/W	R/W								R/W	R/W						
ADDR	(IMMR & 0xFFFF0000) + 0x9c6															

CPU16 In Device Mode, an Endpoint Might Lock Up After the Reception of an IN Token

Description:

The lockup condition is dependent on the sequence of tokens from the host and the condition of the endpoint's transmit FIFOs, including OUT-only endpoints (for which the transmit FIFO is not used). A lockup condition is observed by the endpoint's inability to respond to a valid IN request (with either data or a NAK/STALL).

Work Arounds:

A fix package exists on the product website. The package includes a microcode patch and the application software workaround procedure, as well as a text file with instructions on how to implement the workaround.

GENERAL ERRATA

G1 Core Operation Is Limited to a 3.0V Minimum

Description:

The current versions of the MPC823 silicon are only tested and verified at 3.0V–3.6V power. Because of this, low voltage operation at 2.2V cannot be guaranteed to power the core.

Work Arounds:

None.

G2 The External Bus Transaction May Hang After a PLPRCR Write Access

Description:

An endless external bus transaction can occur on the next external bus access after executing a PLPRCR write command that changes the MF field. The PLPRCR write command causes the PLL to freeze the clocks until it is locked again, according to the new MF value. The failure mechanism occurs because the clock unit indicates operation complete before all necessary tasks are actually completed. The next external bus request is driven by the system interface unit and suddenly all clocks are stopped.

Work Arounds:

The store instruction to the PLPRCR register should be in a burst-aligned address (cache line) followed by an isync instruction. The instruction cache should be enabled while executing this code sequence.

G3 LCD Off and On Sequence With a Pending SDMA Cycle Causes Wrong Data Fetch

Description:

If the LCD controller is turned off and there is a pending SDMA cycle for the LCD controller, and then the LCD is turned on again before the SDMA cycle completes, the LCD controller will start fetching from the start address + 16 instead of the start address.

Work Arounds:

Make sure that the pending SDMA cycle is completed before turning on the LCD controller. This can be done by performing an access to external memory before turning on the LCD controller.

G6 LCD Controller Off Sequence When LAM Bit Is Set May Cause the CPU to Lock Up

Description:

The LCD aggressive mode (LAM) bit of the SDCR register allows aggressive arbitration for the LCD controller when doing DMAs to system memory. If this bit is set and the LCD controller is turned off, the LCD controller generates a spurious request to the SDMA that may cause the CPU to lockup.

Work Arounds:

Clear the LAM bit before turning off the LCD controller.

G7 Possible External Bus Hang Occurs Under Certain Error Conditions

Description:

The external bus cycle may hang when the following sequence of events occur:

1. The transaction on the external bus ends as a result of TEA assertion OR a bus monitor timeout occurs.

AND

2. The next transaction also ends with a TEA assertion or a bus monitor timeout. (burt 300)

Work Arounds:

None.

G8 Higher Than Normal Current Consumption Without Executing a MULLW Instruction

Description:

In the integer multiply module in the core, there are internal latches that do not get initialized properly during reset. Due to this improper initialization there is a 1:8 chance of a driver contention which may create higher than normal current consumption in various modes. This current consumption is most noticeable in the lower power modes. (burt_361)

Work Arounds:

Adding a mullw instruction puts the two control lines into a valid state, eliminating the contention.

G12 Incorrect Reporting of Loss-of-Lock Reset Status

Description:

The RSR[LLRS] bit is set by both unintentional and software-initiated loss-of-lock. The RSR[LLRS] bit should be set only by an unintentional loss-of-lock. Software-initiated loss-of-lock (e.g. changing the SPLL multiplication factor or entering low-power modes) should not set this bit.

Work Arounds:

The PLPRCR[SPLSS] functions as intended. Reference this bit instead.

G13 Conflict Between Data Show Cycles and SDMA Burst Writes

Description:

IF data show cycles are enabled via SIUMCR[DSHW] ANDan internal register or dual-port RAM access is made immediately following an SDMA burst write, THEN the SDMA burst write may be corrupted. The observed phenomenon is that a burst write with four operands will hold the second operand into the third and fourth burst beats. For example, a burst write of A-B-C-D will be observed on the bus as A-B-B-B.

NOTE

This behavior can also occur when the SDMA burst is to burst-inhibited memory. Setting the memory to burst-inhibited will not solve the problem.

Work Arounds:

Do not use data show cycles in a system that performs SDMA bursts. These include systems that use ATM,Fast Ethernet, and memory-to-memory IDMA.

G14 CPU Receives A Machine Check After Writing to the PLPRCR

Description:

The CPU may receive a machine check after writing to the PLPRCR. This error is caused by an extra clock generated by the clock block after the SIU releases the bus. When the internal bus is released the CPU begins a transaction. The CPU's clocks are then stopped mid-cycle and it never receives the acknowledge from the bus. The failure mechanism occurs due to an internal logic synchronization issue aggravated by memory refreshes performed by the UPM. The problem is only evident when entering and exiting doze mode frequently, such as when using doze to conserve power. The possibility of encountering this problem is small but finite (1 in a million entries).

Work Arounds:

Prevent the CPU from getting the bus during the extra clock. To do so you must enable the instruction cache and insert a delay. To calculate how long of a delay is necessary, take the longest bus transaction including memory refresh and PCMCIA (in CPU clocks). The resultant number of clocks must be executed using instructions such as NOP (1 clock), ISYNC (2 clocks) or DIVW (13 clocks). If you use the DIVW instruction, then divide the resultant number by 13 and round up. Then insert this many DIVW instructions (dividing by one) after the isync (see errata G9). If more than 2 DIVWs are required then the additional CACHE lines must be loaded and locked into the ICACHE.

For example, If your longest transaction is 16 CPU clocks, then you must add 2 DIVW instructions:

```
        .global SetPLPRCR
        .align 16
SetPLPRCR:
                                                nop
        nop
        nop
        addi r5, r0, 1
        stw r4, PLPRCR(r3)      #This cache line MUST be in the cache.
        isync
        divw r4, r4, r5
        divw r4, r4, r5
        nop                     #This cache line MUST be in the cache.
        nop                     #These NOP's are to prevent the CPU from
        nop                     #trying to fetch from the BLR target by
        nop                     #filling the instruction prefetch queue.
        nop                     #This cache line MUST be in the cache.
        nop
        nop blr
```

G15 Address Bus May Consume Excess Current in Low Power Modes

Description:

When entering Sleep or Deep Sleep Mode, normally PLPRCR[FIOPD] is set to 1 to enable internal pulldowns on the Address and Data Bus. In some units, the Address Bus pins are not sufficently pulled down. This may cause excess current draw. The Data Bus is not affected.

Work Arounds:

None. Fixed in Rev. B.

G16 Open Drain Pins Do Not Allow 5 Volt Pullups

Description:

The Open Drain pins on the MPC823 will not allow external pullups to raise the voltage to the normal levels.

Work Arounds:

None.

How to Reach Us:

Home Page:
www.freescale.com

email:
support@freescale.com

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
(800) 521-6274
480-768-2130
support@freescale.com

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

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku
Tokyo 153-0064, Japan
0120 191014
+81 2666 8080
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate,
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor
Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
(800) 441-2447
303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor
@hibbertgroup.com

Document Number: MPC823CE
Rev. 3
05/2005

freescale™
semiconductor