Device Errata
MPC823

Silicon Revision 0.3—Mask Set 3F98S
March 17, 1998 (Version 1)

These errata apply to the MPC823 Revision 0.3 silicon. Those errata that are currently scheduled to be addressed in the future revision of the silicon are so marked. Notice that this device has soft modem specific microcode installed. *Changes to this version of the errata are in italics*.

CPU ERRATA

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

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

Workaround: Avoid using a bus error in this case.

CPU2. Incorrect Data Breakpoint Detection on Store Instructions

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.

Workaround: None. *FIxed in future revision*.

CPU3. Program Trace Mechanism Error

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)

Workaround: None. *FIxed in future revision*.

CPU4. D-Cache Presents Valid Data When Parity Error Present On A Burst.

If the LDST unit requests data that is not in the Data Cache, then the D-Cache will initiate a burst cycle to the memory. If during this burst cycle, a parity error is generated on the second or third words and not on the critical word; then the D-Cache will present the data to the LDST as valid data. (burt_277).

Workaround:. Disable parity.

CPU5. Instruction Cache LRU Mechanism Flawed.

Fixed in prior revision; shown here for reference to prior errata.

CPU6 Possible Data-Cache Corruption With Special Purpose Register Access Located In Data Cache, Data MMU or SIU.

A write access to a special-purpose register located in the Caches, MMUs, or the SIU might corrupt the contents of the data-cache. The special-purpose registers are: IMMR, IC_CST, IC_ADR, IC_DAT, DC_CST, DC_ADR, DC_DAT, MI_CTR, MI_AP, MI_EPN, MI_TWC, MI_RPN, MI_DBCAM, MI_DBRAM0, MI_DBRAM1, MD_CTR, M_CASID, MD_AP, MD_EPN, M_TWB, MD_TWC, MD_RPN, M_TW, MD_DBCAM, MD_DBRAM0, MD_DBRAM1, DEC, TB, TBU, AND DPDR. (burt_292)

Workaround: 1.) If the contents of the TLBs are not changed dynamically (fixed-page structure), any access to the above-mentioned registers should be avoided (except for initialization).

2a.) If the contents of the TLBs are changed dynamically (pages are loaded on demand), then each “mtspr” instruction which access one of these registers must be preceded by a store word and load word instruction with the data equal to the spr_address of the respective register. As an example, to write the data from the general purpose register r1 to the special purpose register M_TW, the following procedure should be followed:

```
lis      r2,   some_address_msb     # an address in RAM
li       r3,   0x3f80               # the spr_address of the M_TW from the table
stw      r3,   some_address_lsb(r2) # no interrupts
lwz      r3,   some_address_lsb(r2) # between these
mtspr M_TW,    r1                   # 3 instructions
```

2b.) Got to Low Normal Mode before any of the spr registers mentioned above are accessed.

Fixed in Revision A.

CPU7 Case of I-Cache Using Address of Old Page When Fetching New Page.

The Instruction Cache uses the address associated with the old page when fetching the first data froma 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)

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

VIDEO CONTROLLER ERRATA

VC1. VIDEO OUTPUT GETS STUCK IN BACKGROUND MODE.

There is a case in which the Video flush process that was caused by an underrun detection, is not treated appropriately by the underrun recovery mechanism. In this case, the video is stuck in background mode. (burt_280) Originally thought to beFxed in revision 0.1; but, still shows some deficiencies.

Workaround:. Setting the video controller off and then back on, release it from background mode. Fixed in revision A.

CPM ERRATA

CPM1. I2C DIGITAL FILTER PROBLEM WITH FAST I2C SLAVE DEVICES.

I2C slave devices that have hold times of less than approximately one microsecond can create a false detect of a stop/start on the I2C bus when the digital filter mode is enabled. The resulting effect is the I2C controller in the MPC823 sends clocks endlessly on SCL. (burt_282).

Workaround:. Do not enable Clock Filter (FLT) bit in I2MOD Register. Fixed in revision A

CPM2. USB Microcode May Duplicate First Byte For IN Token Transfer.

Fixed in this revision; shown here as a reference to prior errata.

CPM3. Infra-Red Mode Field Bits Reversed

In the Infra-Red Mode Field(see page 16-288 of the MPC823 User's Manual)the encoding/decoding for the M(OD) field of the IRMODE Register are reversed. This means that High Speed (4.0Mb/s) mode is represented by '01' rather than '10' and Middle-speed (0.576 Mb/s or 1.152 Mb/s) is represented by '10'.

Workaround: Use microcode patch. Retrieve downloadable file from website at:http://www.mot.com/mpc823. Fixed in Revision A.

CPM4. Port A Pin (PA13) May Consume Excess Current in Deep-Sleep Mode

When the Port A pin PA13 is configured as the SCC2 function RXD2 and the IrDA logic is not enabled (i.e.,the EN=0 in the IRMODE register), then the MPC823 may consume excess current due to internal contention after entering deep-sleep mode. Other than the approximate 1mA of excess current, there are no operational issues.

Workaround: Before entering deep-sleep mode, configure PA13 as a general-purpose input. When you exit deep-sleep mode, reconfigure PA13 as the SCC2-controlled RXD2, as required.

CPM5. Improper USB initialization May Cause Excess Current in Deep-Sleep Mode

An initialization problem in the USB block might cause excess current in the deep-sleep mode, typically around 500µA.

Workaround: As part of the power-on initialization sequence, the software should enable the baud rate generator clock1 (BRGC1) by setting the EN bit to 1 and leaving it set for at least 16 system clocks before changing the serial interface clock route register from its default value (0x00000000).

GENERAL ERRATA

G1. Core Operation Is Limited to a 3.0V Minimum

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 used to power the core.

Workaround: None. To be tested and verified in this silicon.

G2. ESD Breakdown Voltage for XFC Pin Less Than Motorola Imposed Requirements.

The XFC pin (B2) of the Rev 0.x (xF98S) version of the 823 silicon fails Motorola's XC qualification of 1K Volt for the ElectroStatic Discharge (ESD) breakdown voltage test. The maximum ESD voltage that can be applied to this pin on this silicon without damage is 750 Volts.

Workaround: Ensure devices on not exposed to greater than 750 volts of electrolstatic discharge.

G3. Higher Than Expected Keep Alive Power (KAPWR) Current When Main Power (VDDH & VDDL) Is Removed.

Fixed in this revision; shown here for reference to prior errata.

G4. The External Bus Transaction May Hang After a PLPRCR Write Access

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.

Workaround: 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.

G5. LCD Off and On Sequence With a Pending SDMA Cycle Causes Wrong Data Fetch

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.

Workaround: 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. Fixed in future revision.

G6. Lock/Unlock Command of RSR Also Locks/Unlocks SCCR.

When the Lock or Unlock mechanism of the KAPWR is used on the RSR the same function is performed on the SCCR. (burt_283)

Workaround: The user should modify the RSR and SCCR registers in the following sequence.
- The initial state is: both registers are locked.
- Unlock one register, then modify it and then lock it.
- Unlock the next register, then modify it and lock it.

Fixed in Revision A.

G7. Open Drain Pins De-assertion Drive Ends Too Soon.

All open-drain signal pin's output buffers stop driving the signal to its negated state earlier than they should. This causes the output signal to be negated slower that specified . (burt_288)

Workaround: Use a smaller value pull-up resistor on these pins. Fixed in Revision A.

G8. EXTCLK and CLKOUT Clocks May Not Be In Phase In Half Bus Speed Mode◦

When the MPC823 uses EXTCLK as an input clock source and MF=2 in the PLPRCR (i.e., the frequency of EXTCLK is 1/2 of the internal clock) and the half bus speed mode is used (EBDF=1 in the SCCR), the output

clock from CLKOUT could be 180° out of phase from the input clock. This will affect synchronous designs in which the same clock source (EXTCLK) is shared between the MPC823 and another synchronous device. (burt 293)

Workaround: Use the CLKOUT as the only source clock to all synchronous devices. *Fixed in future revision.*

G9. PLL May Lock On the Falling Edge of EXTCLK

The PLL of the MPC823 locks on the falling edge of the input clock EXTCLK. This will affect the skew between EXTCLK and CLKOUT at the rising edge. The skew depends on the duty-cycle of the input clock, but a 50% duty-cycle will not exceed 2nS. This will affect synchronous designs in which the same clock source is used as an input to EXTCLK, as well as to an external synchronous device (a peripheral or ASIC). Fixed in future revision.

Workarounds: 1) In cases where multiple external devices need to operate synchronously with the MPC823, use the CLKOUT pin of the MPC823 as the clock source for all external synchronous devices. CLKOUT is the effective system master clock to be used for distribution.

2) In cases where it is necessary to synchronize an external master clock (e.g. from a backplane), an MPC823 and external peripherals to allow data transfers in all three directions, insert a PLL between the external master clock and the EXTCLK pin of the MPC823. Connect the phase comparison pin of the PLL to the CLKOUT pin of the MPC823. Also use the CLKOUT signal as the reference clock for distribution to the local external peripherals. The PLL device used has to be capable of operating with a permanent offset of -2nS, so the range of lock should extend to about -4nS. NOTE: This workaround is a concept only. It has not been verified in hardware.

G10. LCD Controller Off Sequence When LAM Bit Is Set May Cause the CPU to Lock Up

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.

Workaround: Clear the LAM bit before turning off the LCD controller. Fixed in future revision.

G11. Possible External Bus Hang Occurs Under Certain Error Conditions

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 $\overline{TEA}$ assertion OR a bus monitor timeout occurs.*

AND

2. *The next transaction also ends with a $\overline{TEA}$ assertion OR a bus monitor timeout. (burt 300)*

Workaround: None. Fixed in future revision.

GENERAL CUSTOMER INFORMATION

Although not generally considered to be errata, the following items are guidelines for using the device appropriately.

CI-100. External Interrupt Handling

For external interrupt pins, if a request signal is a pulse, the interrupt request pin should be configured to “edge detect mode”. This ensures that the interrupt will be recognized even if interrupts are temporarily blocked or disabled by the software. The interrupt service routine (ISR) should clear the edge status flag after the ISR is entered and prior to setting the MSR’s EE bit (if it waits until after the EE bit is set, a second interrupt may be taken).

If a request signal is a “standard handshake”, the assertion is asynchronous, but the negation occurs upon request from the ISR. This ensures that the interrupt is taken and the source of the interrupt is known. The timing with respect to the EE bit is the same.

To avoid spurious interrupts, interrupt masks should not be set while interrupts might be sent to the core. Likewise, no interrupts should be disabled while the interrupt might be pending at the core. That way, when the core responds to the interrupt request, the request will still be pending and the core can determine the source of the interrupt. To accomplish all of the above, the EE bit should be disabled when masks are set or when interrupt enables are cleared.

CI-101. Move To Special Register (mtspr) Access to ICTRL Register

If you use mtspr to set the Ignore First Match (IFM) bit of the ICTRL register to 1 at the same time that you set an instruction breakpoint on this instruction, the chip will behave unpredictably.

Workaround: Disable instruction breakpoints when setting the IFM bit.

CI-102. Concurrent Operation Of Ethernet & I2C or SPI Has Overlapping Parameter RAM Tables.

When concurrent operationof the Ethernet protocol and either I2C or SPI is set up and used at the same timer, there is an overlap in the parameter RAM.

Workaround: There is microcode available that moves the I2C/SPI parameter RAM entries to another location in the dual port RAM. To use this, download the description of the change and the object code file from the website at:

http://www.mot.com/SPS/ADC/pps/_subpgs/_etoolbox/8XX/i2c_spi.html. This package is called the MPC8XX I2C/SPI Microcode Package.

GENERAL DOCUMENTATION ERRATA ASSOCIATED WITH SILICON OPERATION

The following items reflect additional information about the operation of the MPC823 and references made in the *MPC823 User's Manual.* Please refer to the manual for clarification.

DOC1. Cache-Inhibit Operation

In some cases, the last instruction executed from a certain page gets the caching inhibited attribute of the next page when the page change occurs between the time a fetch request was issued to the instruction cache and the time the instruction cache delivers the instruction to the sequencer. Since the instruction cache-inhibit is only used for performance reasons (mostly for not caching very fast memories or pages that include non real-time programs), the performance effect of this feature is negligible. See **Section 9 Instruction Cache** for more information. (burt_237)

DOC2. Updating the DAR and DSISR with Debug Counter Operation

If a load/store breakpoint occurs as a result of debug counter expiration when oa machine check interrupt due to an error in a load/store cycle, a data storage interrupt, or an alignment interrupt occurs, set the DAR and DSISR registers to the effective address associated with the interrupting instruction. In some cases, when a load/store breakpoint occurs when one of the debug counters expires just before one of the above interrupts occur, the value of the DAR and DSISR is changed. Although the interrupt is after the breakpoint and, therefore, should be ignored by the processor, the DAR and DSISR are updated. The value of the DAR and DSISR is normally used by the software inside these interrupt routines and may influence program flow only if these interrupts are nested one inside the other and a load/store breakpoint is used inside one of these interrupt routines. See **Section 6 Core** and **Section 20 Development Support** for details. (burt_253)

DOC3. Revision 0 to Revision A Silicon Changes.

The differences between the two revisions are shown below and are also documented in the MPC823 User's Manual.

a) the dual port ram is now 8k bytes instead of 5k with possible 4k ucode from ram (instead of 2k)

b) 4 timers vs 2 timers

c) 4 baud rate generators vs. 2

d) 3.3 V I/O Operation with 5V TTL Compatibility for the JTAG and CPM I/O ports pins only; other pins have only 3.3 V I/O Compatibility.