Device Errata
MPC821

Silicon Revision B.2 -- Mask Set 4J24C

December 31, 1997 (Version - 0)

These erratum are valid on Revision B.2 silicon. Please note that any errata listed in this document applies to B revision of the silicon, unless otherwise stated. There are errata that carry over from the Revision A.x silicon and are renumbered here along with the Rev A.x errata number in parenthesis [i.e., (CPM1)]. Those errata that are scheduled to be fixed in the next revision of the silicon are so marked.

CPU Errata

CPU1 Bus Error unsupported by the Data Cache burst.

The Data Cache does not support a bus error which might occur on the 2nd or 3rd data beat of a burst. (burt_232) **Fixed in revision D.**
Workaround: Avoid using bus error in this case.

CPU2 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) **Fixed in revision C.**
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 prodcedure 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 this
mtspr  M_TW,  r1                  # 3 instructions
```

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

This document contains information on a product under development. Motorola reserves the right to change or discontinue this product without notice

MOTOROLA

CPU Errata (cont.)

CPU3 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) **Fixed in revision C.**

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

CPU4 (CPU2) Incorrect Data Breakpoint Detection on Store Instructions.

When a breakpoint on data occurs with size elements of byte or half-word is programmed for store instructions, the following erroneous scenarios might happen:

1.) Breakpoint might be detected when it should not.
2.) Breakpoint might not be detected when it should.

Either of these two cases may 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.

Examples:

A.) Assume that the user has programmed a byte data breakpoint on store instruction and is looking for the byte element: 0x55. Assume that register R1 has the value: 0x00080000, R10 has the value: 0x55443322 and the following store instruction is performed:

```
stb R10,0x3(R1)
```

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 shouldn’t.

B) Assume that in the above case the user is programming for byte element of 0x22, then it might happen that a breakpoint condition will not be detected, although it should.

NOTE: These fault cases depend on the previous Load-Store instruction address. If the previous Load-Store instruction address’s LSB is different from the current instruction address’s LSB, then incorrect breakpoint detection might occur. (burt_246) **Fixed in revision C (orginally reported to be fixed in revision B).**

Workaround:. none.

CPU5 Program Trace Mechanism Error

In the following case there is an error in the program trace mechanism.

```
The program is:
0x0004FF0:  divw. r25,r27,r26
0x00004FF4:  divw. r28,r27,r26
0x00004FF8:  unimplemented
0x00004FFC: b 0x00005010

where 0x00005010 belongs to a page with a page fault.
```

The divide takes a long time so the instruction queue gets filled with the unmimplemented instruction, the branch and the branch target (page fault).

When the sequencer takes the unimplemeneted instruction it releases the fetch (that was blocked by the MMU error) this causes the queue to get another instruction besides the first page fault. Because the second fault is sequential to the branch target it is not reported by the vf. This causes a wrong vf flush information to be reported when the umimplemnted exception occurs. (burt_251) **Fixed in revision C.**

Workaround: none

CPM Errata

CPM1 (CPM3). Concurrent Operation of Ethernet & I^{2}C or SPI Has Overlapping Parameter RAM Tables

When concurrent operation of the the Ethernet protocol and either I^{2}C or SPI is set up and used at the same time, there is an overlap in the parameter RAM. (burt_2xx)
Workaround: There is microcode available that moves the I^{2}C/SPI parameter RAM entries to another location in the dual port RAM. 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 MPC8XX I^{2}C/SPI Microcode Package.

CPM2. I2C Receive Problem in Arbitration-Lost State

If the MPC821 I2C master transmitter loses arbitration to another I2C master which is transmitting to the MPC821, the 821 receiver will not accept the message (address byte not acknowledged) (burt_xxx).
Fixed in revision C.
Workaround:
1. Avoid multimaster configuration
2. The operation should be retried by the other master through software.

CPM3. I2C Error in FLT Bit.

An error will occur if the FLT bit is set to turn on the digital filter for the I2C. The digital filter is activated by setting the FLT bit in the I2C mode register and is default off at reset. Note, that this digital filter is not required for normal operation. The MPC821 I2C is fully compliant to the I2C specification without this digital filter option. (burt_282). **Fixed in revision C.**
Workaround Do not turn o the digital filter for the I2C clock filter.

SIU Errata

SIU1. Timing for Address Latch Enable of PCMCIA Is Åsserted Early.

The Address Latch Enable (ALE) pin should change with external address phase (t3) of the SIU; but it actually changes at time t1. (burt 284) **Fixed in revision C.**
Workaround: None.

General Errata

G1 (G2). Core Operation Is Limited to 3.0 Volt Minimum.

The current versions of the 821 silicon are only tested and verified at 3.0V to 3.6V power. Because of this, low voltage operation (@2.2V) cannot be used for powering the core.
Workaround: None. To be tested and verified in this or future silicon.

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

The XFC pin (T2) of the B.1 (3J24C) version of the 821 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 as a reference to prior errata.

General Errata (cont.)

G4. EXTCLK and CLKOUT Clocks May Not Be In Phase In Half Bus Speed Mode.

When the MPC821 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 degrees out of phase from the input clock. This will effect synchronous designs where the same clock source (i.e., EXTCLK) is shared between the MPC821 and another synchronous device. (burt 293) **Fixed in Revision C.**

Workaround: Use the CLKOUT as the only source clock to all synchronous devices.

G5. PLL May Lock On The Falling Edge Of EXTCLK.

The PLL of the MPC821 can lock on either the rising or the falling edge of the input clock (clock at EXTCLK pin). If it locks on the falling edge, this will effect the skew between EXTCLK and CLKOUT at the rising edge. Effectively, the skew at the rising edge will depend on the duty cycle of the input clock. This will effect synchronous designs where the same clock source (i.e., EXTCLK) is shared between the MPC821 and any other syschronous devices. **Fixed in future revision.**

Workaround: Use the CLKOUT as the only source clock to all synchronous devices.

G6. LCD Controller Off Sequence When LAM Bit Is Set May Cause the CPU to Lockup.

Starting with Revison B, an additional bit was added to provide more aggressive arbitration for the LCD Controller when doing DMAs to system memory. This bit is bit 25, called the LCD Aggressive Mode bit or LAM bit, of the SDCR Register. If this bit is set in the SDCR and the LCD Controller is turned off, the LCD Controller generates a spurious request to the SDMA that may cause the CPU to lockup. **Fixed in future revision.**

Workaround: Clear the LAM bit of the SDCR before turning the LCD Controller off .

G7. LCD Off then 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. **Fixed in future revision.**

Workaround: Ensure 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 the LCD Controller on .

G8. 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) **Fixed in Revision C.**

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.

G9. 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 **Fixed in Revision C.**

Workaround: Use a smaller value pull-up resistor on these pins.

General Errata (cont.)

G10. Potential External Bus Transaction Hang After PLPRCR Write Access.

An endless external bus transaction can occur on the next external bus access after executing aPLPRCR 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 due to the clock unit indicating operation complete prior to actual completion of all necessary tasks. The next external bus request is driven by the SIU and suddenly all clocks are stopped.
Workaround: Being investigated.

General Customer Information

Although not generally considered to be errata the following items are provided as guidelines in the appropriate use of the device.

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 makes sure that the interrupt will be recognized even if interrupts are temporarily blocked or disabled via software. The interrupt service routine (ISR) should clear the edge status flag after the ISR is entered and prior to setting the MSR (EE) bit [if it waits until after the EE bit is set, a 2nd 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 as above.
In order to avoid spurious interrupts, any 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 what the source of the interrupt is. 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.

Setting the ICTRL(IFM) bit (IFM=1), which is the Ignore First Match, by mtspr together with setting an instruction breakpoint on this instruction causes unpredictable behaviour of the chip.
Workaround: Disable instruction breakpoints when setting the IFM bit of the ICTRL Register.

General Documentation Errata Associated with Silicon Operation

The following items reflect operation of the MPC821 and how this operation may be currently misrepresented in the MPC821 User's Manual (MPC821UM/AD). Please refer to the sections mentioned, in the UM, for clarification and/or replacement by the following information.

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 the fetch request was issued to the Instruction Cache and the time the Instruction Cache delivers the instruction to the sequencer. Since Instruction Cache inhibit is used only 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. (burt_237)

DOC2. DAR and DSISR Updating with Debug Counter Operation.

If a load/store breakpoint occurs as a result of debug counter expiration when one of the following three interrupts occur, Machine Check interrupt due to an error in a load/store cycle, Data Storage interrupt, and Alignment interrupt; the DAR & DSISR is set to the effective address associated with the interrupting instruction. In some cases, when a load/store breakpoint occurs due to one of the debug counters expiration; just before one of the above interrupts occur, the value of the DAR & 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 & DSISR is normally used by the software inside these interrupt routines and therefore this anomaly may influence program flow only if these interrupts are nested one inside the other and load/store breakpoint is used inside one of these interrupt routines. See Section 6.6.13, Core, and Section 19, Development Support. (burt 253)

DOC3. Ethernet Is Only Supported on SCC1.

All references to Ethernet support on SCC2 is in error. The MPC821 only supports the Ethernet protocol on SCC1. Disregard all references to Ethernet on SCC2 in Section 16 of the MPC821 User's Manual.

DOC4. LCD Aggressive Mode Bit Added to the SDCR.

Referring to Section 16.10.2.1 (page 16-78) of the MPC821 User's Manual, a new feature has been added to the Revison B and subsequent silicon. Bit position 25 is no longer reserved and is now referred to as the LCD Aggressive Mode bit, or LAM. When this bit is cleared, LAM=0, the SDMA arbitration operates as described in the User's Manual. When the LAM=1, then the LAID and RAID fields must be set equal to '00'. When LAM=1, LAID=00, and RAID=00, the SDMA arbitration function is set to provide the LCD Controller with the highest possilbe priority, to allow for improved display memory to LCD panel data transfer, thus minimizing the chance for an underrun condition.

DOC5. Reserved Bits in the PDPAR.

Referring to Section 16.19.12.3 (page 16-466) of the MPC821 User's Manual, Port D Pin Assignment Register; note that bits 0-2 are marked with a '-' and these bits should be considered "reserved" on the MPC821. Anytime this register is written to, care should be taken to ensure that those three bits are alway written as '0', the reset value. Prior revisions of the silicon allowed these bits to be 0 or 1, but with Revison B of the MPC821 these bits are used internally and when set to 1 , improper operation may occur.