Errata

MCF5206ECE
Rev. 2.1, 1/2004

MCF5206e Device Errata

This document identifies implementation differences between particular versions of the MCF5206e processor and the description of the MCF5206e processor contained in the MCF5206e User's Manual. Please check the WWW at http://www.motorola.com/ColdFire for the latest updates. This errata lists any processor differences from the following documents:

- MCF5206e User's Manual
- ColdFire Microprocessor Family Programmer's Reference Manual

This document applies to the following mask set: J22G.

Designers can differentiate what mask set by reading the idcode register in the JTAG module of the MCF5206e.

The idcode register shown in Table 1. will be identical between mask sets except for the version number. For an J22G mask, idcode[31:28]=version no.= 0000.

Table 1. . MCF5206e IDCODE Register

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
x	x	x	x	0	1	0	0	1	1	0	0	0	0	0	0
Version No.				Design Center= ColdFire						Device No.					
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0	0	1	1	0	0	0	0	0	0	0	1	1	1	0	1
Device No.				JEDEC No.											-

Table 11. Summary of MCF5206e Errata

Current Errata ID	Date MCF5206e Errata Created	Applicable to Mask	Errata Title
1	10/31/1998	J22G	Incorrect Read from Bits 7:0 of DMA Source and Destination Registers
2	10/31/1998	J22G	Incorrect Read from Interrupt Mask Register (IMR) bits[15:14]
3	10/31/1998	J22G	Internal DMA Access to S-Bus Module Address Space]
4	10/31/1998	J22G	Section 4 - DC Parametric
5	10/31/1998	J22G	+5V Input Leakage Current
6	10/31/1998	J22G	Changes to Timing Specifications
7	09/22/1999	J22G	Test Access Port (TAP) Reset
8	09/22/1999	J22G	CF2 Debug Captured Write after Exception, may cause an Extraneous PST = 1
9	09/22/1999	J22G	Problem Involving Undetected Level 7 Interrupt
10	09/22/1999	J22G	CF{2,3} Debug—Non-contiguous Stream of PST=D Values during a Debug Interrupt Exception
11	09/22/1999	J22G	CFxCpu—Forcing Emulator Mode from Reset by Assertion of CSR[13] Does Not Work
12	09/22/1999	J22G	When using multiple DMA channels, a write by a DMA channel could get lost if a higher priority DMA channel requests the internal bus simultaneously.
13	09/22/1999	J22G	Direct Memory Access Controller (DMA) Cycle Steal Mode Mis-operation
14	09/22/1999	J22G	Source address increment bit for DMA channels does not increment properly
15	01/22/2004	J22G	Corrupted Return PC in Exception Stack Frame

1. Incorrect Read from Bits 7:0 of DMA Source and Destination Registers

1.1 Description

When reading the four DMA source and destination registers (DMASAR1, DMASAR2, DMADAR1, DMADAR2), the data for bits [7:0] will echo the data on bits [15:8]. Writes to all of these registers function correctly.

1.2 Workaround

This errata has the effect of complicating the observing DMA transfers. If the source increment (SINC) or the destination increment (DINC) bit of the DMA control register are set, the SAR and DAR registers will increment, respectively, with each transfer. User code will not be able to read a correct value of these registers. A workaround is to track DMA progress by reading the DMA byte count register.

MASKS: J22G 10/13/1998

2 Incorrect Read from Interrupt Mask Register (IMR) Bits[15:14]

2.1 Description

Bits [15:14] of the interrupt mask register will always return 0. Writes to these registers function correctly.

2.2 Workaround

If reads are required, store the written value in a separate memory location whenever the IMR is updated.

MASKS: J22G 10/13/1998

3 Internal DMA Access to S-Bus Module Address Space

3.1 Description

The internal DMA cannot perform transfers in or out of the S-Bus address space. This address space includes the UART, Parallel Port, Timers and M-Bus.

3.2 Workaround

Use the core CPU to access S-Bus space.

MASKS: J22G 10/13/1998

4 Section 4—DC Parametric

4.1 Description

All outputs which are rated to drive 8mA Ioh and Iol (XBD, XBR, XCS[3:0], A[23:0], ATM, D[31:0], PP[7:0], RTS[2:1], SIZ[1:0]. TDO, TOUT[1:0], TT[1:0], TXD[2:1], SCL*, SDA*) will, under worst case conditions, maintain the specified 2.4 V Voh and 0.5 V Vol while loaded up to 5 mA. All outputs which are rated to drive 16mA Ioh and Iol (XCAS[3:0], XRAS[1:0]. XDRAMW, XTS, RW, A[27:24]) will, under worst case conditions, maintain the specified 2.4 V Voh and 0.5 V Vol while loaded up to 11 mA.

* Note that open-drain pins require pull-ups and cannot driver Ioh.

4.2 Workaround

Not Applicable.

MASKS: J22G 10/13/1998

5 +5V Input Leakage Current

5.1 Description

The maximum input leakage current when VIH=5.5 shall be 200uA.

5.2 Workaround

Not Applicable.

MASKS: J22G 10/13/1998

6 Changes to Timing Specifications

6.1 Description

The following table describes changes to several published input timing specifications:

Table 1. . Input Timing Relaxations

NAME	CHARACTERISTICS	40 MHZ OLD	40 MHZ NEW	54 MHZ OLD	54 MHZ NEW	UNITS
B1F	IPL[2:0]/IRQ[7,4] Valid to CLK (SETUP)	1.5	3.0	1.0	2.0	NS
B5	Data Input (D[31:0]) valid TO CLK (SETUP)	6.0	7.0	3.0	3.5	NS
T2	TIN[1:0] Valid to CLK (SETUP)	4.0	5.0	2.5	3.0	NS
U1	RXD[2:1] Valid to CLK (SETUP)	2.0	3.0	1.5	2.0	NS
P1	PP[7:0] Input setup to CLK	3.0	4.5	2.0	3.0	NS

6.2 Workaround

- Not Applicable.

MASKS: J22G 10/13/1998

7 Test Access Port (TAP) Reset

7.1 Description

The $\overline{\text{TRSTX}}$ signal does not work.

7.2 Workaround

To reset the JTAG TAP controller, TMS must be held high for 5 consecutive rising edges of TCK, as specified in IEEE 1149.1.

MASKS: J22G 09/22/1999

8 CF2 Debug Captured Write after Exception may cause an Extraneous PST = 1

8.1 Description

There is a complex sequence of events that may cause the Version 2 debug module to output an extraneous PST=1 value when capturing and displaying operand write data. The specific sequence is:

1) The processor takes any type of exception.

2) Once the exception handler is entered, the very first operand to be captured and displayed by the PST/DDATA logic is an operand write.

3) If the next instruction after the operand write is a non-memory- referencing opcode (e.g., a register-to-register or immediate-to- register instruction), then the debug module may incorrectly output an extraneous PST = 1 value before the write operand is captured and displayed.

The resulting stream of PST values is as follows:

Table 2. . PST Values

PST	PST[3:0] Stream (HEX)															
Current	C	C	...	5	0	...	1	0	...	1	0	...	[89B]	1	0	...
Correct	C	C	...	5	0	...	1	0	...	0	0		[89B]	1	0	...
Position Comment	exception						wrt inst			Extraneous PST			wrt marker			

Note that the processor's operation is perfectly correct throughout this sequence; it is the potential occurrence of the extraneous PST = 1 value that is the error.

8.2 Workaround

If a WDDATA.{B,W,L} instruction is included in the exception handler before any instructions with operand writes, the problem does not occur.

If the first memory-referencing instruction in an exception handler is a write, insert a NOP instruction immediately after it. If any operand reads are performed before the first write, the NOP is not required.

MASKS: J22G 09/22/1999

9 Problem Involving Undetected Level 7 Interrupt

9.1 Description

There is a problem involving the execution of a MOVE to SR or RTE instruction which may cause the assertion of a level 7 interrupt request to be undetected by the processor (that is, the core never responds with a level 7 interrupt exception).

Level 7 interrupts are treated differently than all other interrupts since they are viewed as being edge-sensitive (versus level-sensitive). Consequently, the processor has special logic to recognize the high-to-low assertion edge of the active-low interrupt 7 request. The error involves a very small window of time where the assertion of a level 7 interrupt request, in conjunction with the execution of certain variations of the MOVE to SR or RTE instructions, may incorrectly set a control state inhibiting the processing of this interrupt request.

In particular, the problem exists only when the execution of the MOVE to SR or RTE instruction loads a value of “7” into the 3-bit interrupt mask level of the status register. For all other interrupt mask values the failure cannot occur.

It should also be noted that the typical level 7 interrupt service routine, where the interrupt request is negated before the RTE is executed, would never encounter this problem.

9.2 Workaround

For the MOVE to SR or RTE instructions, load an operand of “6” into the 3-bit interrupt mask of the status register to mask interrupt levels 1-6. The use of this new interrupt mask level does not effect the ability to inhibit interrupts, since values of 6 or 7 both mask levels 1-6. A level 7 interrupt service routine would need to load an operand of “7” into the interrupt mask until the level 7 interrupt source is negated, otherwise another level 7 interrupt would be generated if the interrupt mask is lowered and the level 7 request is still present.

MASKS: J22G 09/22/1999

10 CF{2,3} Debug—Non-Contiguous Stream of PST = D Values during a Debug Interrupt Exception

10.1 Description

If the ColdFire debug module is configured to generate a debug interrupt exception in response to a breakpoint trigger, the processor responds by taking a special exception. While processing this exception, the debug module is required to output a contiguous stream of PST = D values until the exception completes and control is passed to the instruction defined by the interrupt vector. This change-of-flow is signaled by PST = 5, which marks the end of the exception processing. The only allowable deviations to the PST = D stream are operand markers (PST = B) associated with operand captures during the writing of the exception stack frame.

If operand writes are being captured, there is the possibility that the stream of PST = D values is non-contiguous, with PST = 0 values incorrectly inserted into the output stream.

Note that the processor's operation is perfectly correct throughout this sequence; it is the potential occurrence of extraneous PST = 0 values that is the error.

10.2 Workaround

Disable the capturing of operand writes if debug interrupts are enabled. Another possibility is to simply ignore PST = 0 values occurring during a debug interrupt. That is, the debug interrupt exception processing is defined from the initial PST = D until the PST = 5 value.

MASKS: J22G 09/22/1999

11 CFxCpu—Forcing Emulator Mode from Reset by Assertion of CSR[13] Does Not Work

11.1 Description

The ColdFire debug architecture defines a control bit in the configuration and status register, CSR[13], which specifies the processor should begin execution in emulator mode. After reset is negated, the processor samples for certain conditions, that is, the assertion of the breakpoint input signal before beginning reset exception processing. A typical sequence may involve the assertion of the breakpoint signal immediately after the negation of reset, followed by a BDM-controlled initiation of the microprocessor and/or system. Once the BDM "go" command is received, the processor continues with reset exception processing.

If CSR[13] is set during this initialization sequence, the processor is supposed to begin the reset exception processing in emulator mode. Unfortunately in all ColdFire core designs, the assertion of CSR[13] in this type of sequence does not force entry in emulator mode.

11.2 Workaround

Do not set CSR[13]. The quickest entry into emulator mode after reset is created with the following sequence:

1. While in the BDM initiation sequence, program a debug breakpoint trigger event by an operand reference to address 0 or 4. As part of this sequence, the debug interrupt vector must also be initialized to the same address as the initial PC defined at address 4.
2. 2) When the BDM go command is received by the processor, the reset exception processing, fetches the long words at addresses 0 and 4 in normal mode. Then a debug interrupt is immediately generated before the first instruction is executed.
3. 3) Execution continues in emulator mode.

MASKS: J22G 09/22/1999

12 When using multiple DMA channels, a write by a DMA channel could get lost if a higher priority DMA channel requests the internal bus simultaneously.

12.1 Description

If a channel requests the internal bus at the same time that a higher priority channel does, the write access of the lower priority channel is lost. This results in a read access by the lower priority channel with the address equal to the value of its source address register (SAR). The second and third accesses are a read and write by the higher priority channel and the write by the lower priority channel is lost. Another result could be that if the lower priority channel is in cycle steal mode and that channel is using the external request signal then the external request signal does not get recognized after the failure.

12.2 Workaround

- The DMA start bits can be set with writes by the processor. This means not to set the external DMA request bit DCR[EEXT].
- If multiple DMA channels are going to be used simultaneously, then have only a single channel running at one time by using the external DMA request.

MASKS: J22G 09/22/1999

13 Direct Memory Access Controller (DMA) Cycle Steal Mode Misoperation

13.1 Description:

When the Cycle Steal (CS) mode is set in DCR, the DMA controller module does not operate properly if multiple channels are active. In other words, when CS is set, it does not do a single read/write transfer. For example, if DMA channel 1 is set to Cycle Steal mode, and the first request for DMA channel 1 has already occurred, and then a second DMA channel is started, DMA channel 1 will completely transfer the number of bytes programmed in the BCR instead of a single transfer, thus ignoring the setting of the Cycle Steal mode bit.

13.2 Workarounds:

The only guaranteed way for Cycle Steal mode to work correctly is to use only the lowest priority channel in Cycle Steal mode.

MASKS: J22G 09/22/1999

14 Source address increment bit for DMA channels does not increment properly

14.1 Description

If multiple internal DMA channels are being used and one channel is programmed to increment its source address register (SAR) while the other is not, the source address may not increment correctly on arbitration boundaries.

For example, DMA channel 1's SAR may be programmed to increment source addresses and channel 2's SAR may be programmed not to increment source addresses. If channel 2 initiates a DMA request while channel 1 is doing a transfer, when channel 2 is arbitrated the bus, it's SAR WILL increment after the first transfer when it is NOT supposed to be incrementing. This could cause complications if channel 2 was being used as a FIFO. Furthermore, if channel 1 is programmed not to increment its SAR and channel 2 is programmed to increment its SAR, the opposite effect will take place. If channel 2 initiates a DMA request while channel 1 is doing a transfer, when channel 2 arbitrates the bus, it's SAR will NOT increment when it is suppose to. This would cause a duplicate write in the same address location to take place.

14.2 Workaround

- Guarantee no multiple concurrent requests
- Make all channels have identical incrementing schemes (that is, all enabled to increment, or all enabled not to increment).

MASKS: J22G 09/22/1999

15 Corrupted Return PC in Exception Stack Frame

15.1 Description

When processing an autovectored interrupt an error can occur that causes 0xFFFFFFFF to be written as the return PC value in the exception stack frame. The problem is caused by a conflict between an internal autovector access and a chip select mapped to the IACK address space (0xFFFFXXXX).

15.2 Workaround

- Set the C/I bit in the chip select mask register (CSMR) for the chip select that is mapped to 0xFFFFXXXX. This will prevent the chip select from asserting for IACK accesses.
- Remap the chip select to a different address range.
- Use external logic to provide external vectors for all interrupts instead of autovectoring.

MASKS: J22G 01/22/2004

16 Revision History

Table 3. . MCF5206e Processor Errata Change HIstory

Revision	Date	Change History:
Rev 1.0	10/13/1998	Initial Errata
Rev 2.0	09/22/1999	Add - Errata #7 – 14.
Rev 2.1	01/22/2004	Add - Errata #15.

THIS PAGE INTENTIONALLY LEFT BLANK

HOW TO REACH US:

USA / EUROPE / Locations Not Listed:

Motorola Literature Distribution
P.O. Box 5405
Denver, Colorado 80217
1-800-521-6274 or 480-768-2130

JAPAN:

Motorola Japan Ltd.
SPS, Technical Information Center
3-20-1, Minami-Azabu Minato-ku
Tokyo, 106-8573 Japan
81-3-3440-3569

ASIA/PACIFIC:

Motorola Semiconductors H.K. Ltd.
Silicon Harbour Centre
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
852-26668334

HOME PAGE:

http://motorola.com/semiconductors

Information in this document is provided solely to enable system and software implementers to use Motorola products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Motorola reserves the right to make changes without further notice to any products herein. Motorola makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Motorola assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters which may be provided in Motorola data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including “Typicals” must be validated for each customer application by customer’s technical experts. Motorola does not convey any license under its patent rights nor the rights of others. Motorola products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Motorola product could create a situation where personal injury or death may occur. Should Buyer purchase or use Motorola products for any such unintended or unauthorized application, Buyer shall indemnify and hold Motorola and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Motorola was negligent regarding the design or manufacture of the part.

MCF5206ECE Rev. 2.1, 1/2004