MCF54418 Chip Errata

Silicon Revision: All

This document identifies implementation differences between the MCF5441*x* processors and the description contained in the *MCF54418 ColdFire® Reference Manual and Datasheet*. Refer to http://www.freescale.com/coldfire for the latest updates.

The latest mask of the MCF5441x family is 1M38W.

Table 1. Summary of MCF5441x Errata

Errata	Module Affected	Date Errata Added	Revision Affected?
			1M38W
SECF171	ADC/DAC pin	3/4/2010	Yes
SECF180	Interrupt Controller	8/12/2010	Yes
SECF181	Synchronous Serial Interface	10/20/2010	Yes
SECF182	Synchronous Serial Interface	10/20/2010	Yes
SECF183	Synchronous Serial Interface	10/20/2010	Yes
SECF184	Universal Serial Bus Interface	10/20/2010	Yes
SECF193	FlexCAN	10/20/2010	Yes
SECF197	Motor Control PWM	10/20/2010	Yes
SECF198	10/100Mbps Ethernet MAC-NET Core	04/15/2011	Yes
SECF199	Universal Serial Bus Interface	04/15/2011	Yes
SECF200	Universal Serial Bus Interface	04/15/2011	Yes
SECF201	Universal Serial Bus Interface	04/15/2011	Yes
SECF202	Universal Serial Bus Interface	04/15/2011	Yes
SECF203	Universal Serial Bus Interface	04/15/2011	Yes

Table continues on the next page...

Table 1. Summary of MCF5441x Errata (continued)

Errata	Module Affected	Date Errata Added	Revision Affected?
			1M38W
SECF204	Universal Serial Bus Interface	04/15/2011	Yes
SECF205	DDR2 SDRAM Memory Controller	04/15/2011	Yes

The table below provides a revision history for this document.

Table 2. Document Revision History

Rev. No.	Date	Substantive Changes
0	4/2010	Initial revision
1	8/2010	Added SECF180.
2	10/2010	Added SECF181. Added SECF182. Added SECF183. Added SECF184. Added SECF185. Added SECF186. Added SECF187. Added SECF188.
3	04/2011	Added SECF193. Added SECF197. Added SECF198. Added SECF199. Added SECF200. Added SECF201. Added SECF202. Added SECF203. Added SECF204.
4	07/2011	Removed SECF188.

SECF171: Latch-up susceptibility on ADC_IN7/DAC1_OUT pin

Errata type: Silicon
Affects: ADC/DAC

Description: ADC_IN7/DAC1_OUT pin has latch-up susceptibility at a low injection current level (40mA vs. 100mA spec). This is true for positive polarity injection at 125°C.

Workaround: Must prevent large injection currents with board level protection such as a series resistance.

Fix plan: Will be fixed in revision 2.0

SECF180: Spurious Interrupts Can Cause Incorrect Vector Fetch

Errata type: Silicon
Affects: INTC
Description: In rare cases the interrupt controller's spurious detection logic can cause a fetch to an incorrect vector number. This can occur when the core is starting the IACK for a spurious interrupt. During this small window of time, if a second interrupt at a different level arrives, the second interrupt causes the interrupt controller logic to clear the spurious request. Therefore, the interrupt controller sees no valid interrupt pending at the requested level and returns vector number 0 for INTC0, or vector number 64 for INTC1, or vector number 128 for INTC2.

The second interrupt can be at any level other than the level that caused the spurious interrupt (it can even be a lower priority than the spurious interrupt). If the second interrupt is at the same level as the spurious interrupt, then the correct vector number for the second interrupt is returned.

Workaround: In many systems spurious interrupts represent error conditions in and of themselves. So, it is always a good design practice to eliminate potential causes of spurious interrupts during product development. Proper interrupt management can help to prevent or reduce the possibility of spurious interrupts (and the potential occurrence of this errata). The correct procedure for masking an interrupt in the INTC or inside the module is:

1. Write the interrupt level mask in the core’s status register (SR[I]) to a value higher than the priority level of the interrupt you want to mask.
2. Mask the interrupt using the INTC’s IMR and/or an interrupt mask register inside the module.
3. Write the original value back to the core’s status register.

Even when steps are taken to remove spurious interrupts, it is still desirable to have a spurious interrupt handler to help manage unexpected events and glitches in a system. A workaround to allow for correct spurious interrupt handling is to:

1. After boot, copy the vector table to RAM
2. Modify the vector 0, and vector 64, and vector 128 entries so that they point to the spurious interrupt handler.

This way the system performs the same for any potential spurious interrupt vectors. Vectors 0, 64, 128, and 24 (the correct spurious interrupt vector) should point to the same handler.

Fix plan: Currently, there are no plans to fix this.

SECF181: Rx re-eanbled with Tx disabled and TFR_CLK DIS set

Errata type: Silicon
Affects: SSI

Description: If RX is re-enabled in I2S_master mode with TX disabled along with TFR_CLK DIS set, the first data after re-enabling is received in the second timeslot and hence channel swapping takes place.

In Network Sync mode if RX is re-enabled with TX disabled when TFR_CLK_DIS is set, data is not accepted into the FIFO according to the masking bits.

Workaround: Reset SSI before enabling RX or do not use TFR_CLK_DIS feature

Fix plan: Will not be fixed

SECF182: If TX disabled before frame sync, data not transmitted in 1st timeslot of next frame

Errata type: Silicon
Affects: SSI
Description: I2S_slave mode: If TX is disabled just before frame sync, data is not transmitted for the first timeslot in the next frame but DDR signal is high for the first timeslot.

This issue occurs if TX is disabled within four-bit clock cycles of active edge of frame sync

Workaround: WAIT for TFS and then disable TX

Fix plan: No plans to fix

SECF183: TX enabling with respect to full speed in normal sync mode

Errata type: Silicon
Affects: SSI
Description: In Normal Sync Mode if TX is enabled 4 clock cycle before external early word length frame sync with F0 disabled, data is re-transmitted from STX0 since TDE does not occur

In Normal Sync Mode if TX is re-enabled 2 clock cycle before external non-early word length frame sync , data is not transmitted for 1 clock cycle since ddr_stxd is low for 1 clock cycle.

This issue occurs if TX is enabled within 4 bit clock cycles of active edge of frame sync

Workaround: To enable TX, use the following sequence:

1. Enable RX in SCR register.
2. Enable RIE and RFS_EN bit in SIER register.
3. Wait for occurrence of RFS interrupt.
4. Enable TX and disable RX.

Fix plan: Will not be fixed

SECF184: Marginal crossover failures across Voltage / Temp corners

Errata type: Silicon
Affects: USB HOST / OTG
Description: Description: Timing for the first bit transmitted in the controllers is too early and makes it difficult to achieve clean USB eye diagrams. While this has not been seen as an operational issue, it may result in a USB a specification violation. The behavior in question is related to the controller not pre-driving the transmit enable pin before the first J/K transition.

Workaround: May require compliance waiver if full USB certification is requirement.

Fix plan: No plans to fix

SECF193: FlexCAN: Global Masks misalignment

Errata type: Silicon
Affects: FlexCAN
Description: During CAN messages reception by FlexCAN, the RXGMASK (Rx Global Mask) is used as acceptance mask for most of the Rx Message Buffers (MB). When the FIFO Enable bit in the FlexCAN Module Configuration Register (CANx_MCR[FEN], bit 2) is set, the RXGMASK also applies to most of the elements of the ID filter table. However, there is a misalignment between the position of the ID field in the Rx MB and in RXIDA, RXIDB and RXIDC fields of the ID Tables. In fact RXIDA filter in the ID Tables is shifted one bit to the left from Rx MBs ID position as shown below:

Rx MB ID = bits 3-31 of ID word corresponding to message ID bits 0-28

RXIDA = bits 2-30 of ID Table corresponding to message ID bits 0-28

Note that the mask bits one-to-one correspondence occurs with the filters bits, not with the incoming message ID bits. This leads the RXGMASK to affect Rx MB and Rx FIFO filtering in different ways.

For example, if the user intends to mask out the bit 24 of the ID filter of Message Buffers then the RXGMASK will be configured as 0xffff_ffef. As result, bit 24 of the ID field of the incoming message will be ignored during filtering process for Message Buffers. This very same configuration of RXGMASK would lead bit 24 of RXIDA to be "don't care" and thus bit 25 of the ID field of the incoming message would be ignored during filtering process for Rx FIFO.

Similarly, both RXIDB and RXIDC filters have multiple misalignments with regards to position of ID field in Rx MBs, which can lead to erroneous masking during filtering process for either Rx FIFO or MBs. RX14MASK (Rx 14 Mask) and RX15MASK (Rx 15 Mask) have the same structure as the RXGMASK. This includes the misalignment problem between the position of the ID field in the Rx MBs and in RXIDA, RXIDB and RXIDC fields of the ID Tables.

Workaround: It is recommended that one of the following actions be taken to avoid problems:

1. Do not enable the RxFIFO. If CANx_MCR[FEN]=0 then the Rx FIFO is disabled and thus the masks RXGMASK, RX14MASK and RX15MASK do not affect it.
2. Enable Rx Individual Mask Registers. If the Backwards Compatibility Configuration bit in the FlexCAN Module Configuration Register (CANx_MCR[BCC], bit 15) is set then the Rx Individual Mask Registers (RXIMR0-63) are enabled and thus the masks RXGMASK, RX14MASK and RX15MASK are not used.
3. Do not use masks RXGMASK, RX14MASK and RX15MASK (i.e. let them in reset value which is 0xffff_ffff) when CANx_MCR[FEN]=1 and CANx_MCR[BCC]=0. In this case, filtering processes for bothRx MBs and Rx FIFO are not affected by those masks.
4. Do not configure any MB as Rx (i.e. let all MBs as either Tx or inactive) when CANx_MCR[FEN]=1 and CANx_MCR[BCC]=0. In this case, the masks RXGMASK, RX14MASK and RX15MASK can be used to affect ID tables without affecting filtering process for Rx MBs.

Fix plan: No plans to fix.

SECF197: Trouble creating phase-delayed pulses

Errata type: Silicon
Affects: Motor control PWM
Description: The current PWM design cannot easily create phase-delayed pulses that go into the PWM period. These pulses require that the turn off time (VAL3) be less than the turn on time (VAL2). This is an issue because the comparisons between the counter and VAL2 and VAL3 are using is "greater than or equal to" instead of just "equal to".

The VAL2 and VAL4 registers define the turn-on edge and the VAL3 and VAL5 registers define the turn off edge of the PWMA/PWMB signals respectively. VAL3 cannot be less than VAL2 and VAL5 cannot be less than VAL4. Doing so will cause the PWM signal to turn off at the correct time (VAL3 or VAL5), but it will not turn on at the time defined by VAL2 or VAL4.

This can be an issue during the generation of phase delayed pulses where the PWM signal goes high late in PWM cycle N and remains high across the cycle boundary before going low early in cycle N+1 and goes high again in PWM cycle N+1. These errata will allow that to happen. VAL3 register must be "greater than or equal to" VAL2 register and VAL5 must be "greater than or equal to" VAL4.

Workaround: Software will have to guarantee the condition never occurs.

Fix plan: No plans to fix.

SECF198: Magic packet wake-up not to specification

Errata type: Silicon
Affects: Ethernet MAC-NET
Description: Ethernet MAC-NET should require 16 Magic Packets to wake-up. Instead, it wakes after 6. Currently Ethernet MAC-NET will wake up if there are only 6 duplications of the IEEE address in the expected pattern in the frame (i.e., it will definitely wake up if there are 16 duplications).

To be strictly compliant with the Magic Packet spec as stated in the AMD Magic Packet White Paper (or as commonly/widely implemented in the industry), the Ethernet MAC-NET should only wake up if there are 16 duplications (instead of just only 6) of the IEEE address in the expected pattern in the frame.

Example :

If the IEEE address of a target computer is 01:02:03:04:05:06 (6 bytes), then the LAN controller of that machine should be scanning for the following sequence (less the Destination, Source, CRC, and Misc fields needed to complete an IP packet).

```
FFFFFFFFFFFF010203040506010203040506010203040506010203040506
010203040506010203040506010203040506010203040506010203040506
010203040506010203040506010203040506010203040506010203040506
010203040506010203040506
```

Notice that there are 16 duplication of ‘01h 02h 03h 04h 05h 06h’ in the frame.

Workaround: None needed.

Fix plan: No plans to fix.

SECF199: Wrong value read in TXPBURST/RXPBURST when AHBBRST=000/100

Errata type: Silicon
Affects: USB HOST / OTG
Description: When reading registers TXPBURST (BURSTSIZE[15:8]) and RXPBURST (BURSTSIZE[7:0]), if SBUSCFG [2:0] is set to 000 or 100, the read value is always 5'b 10000, regardless of the previously programmed value in either TXPBURST or RXPBURST.

Workaround: Software must maintain a copy of this value if needed.

Fix plan: No plans to fix.

SECF200: Short inter-packet delay between OUT transactions may terminate transfer

Errata type: Silicon
Affects: USB HOST / OTG
Description: This issue occurs when controller is configured as device and the host sends two consecutive OUT transactions (for example, two ISO OUT) with a short inter-packet delay between them (less than 200xns). In this case, the clear command from the protocol engine state machine to the protocol engine data path is not sent (and internal byte count in the data path module is not cleared). This causes a short packet to be reported to the DMA engine, which finishes the transfer and the current dTD is retired.

Workaround: It is up to the software to check the transferred number of bytes.

Fix plan: No plans to fix.

SECF201: Extra resume interrupt in HOST mode

Errata type: Silicon
Affects: USB HOST / OTG
Description: When working as host, and doing a resume (software-driven by writing to bit 6 of PORTSCx register), a port change interrupt will fire at the end of resume. This is an EHCI specification violation: section is 4.3.1, end of 3rd paragraph.

Workaround: The extra interrupt must be taken into account when doing resume.

Fix plan: No plans to fix.

SECF202: FSL Serial mode resets to parallel during state PORT_RESET

Errata type: Silicon
Affects: USB HOST / OTG
Description: During a port reset, in either host mode or device mode, the software selection of the serial PHY interface can be lost.

Workaround: At each port reset, software needs to re-write PORTCTR.

Fix plan: No plans to fix.

SECF203: Host RX FIFO overflow too close to End-Of-Frame generates false frame babble or USB reset

Errata type: Silicon
Affects: USB HOST / OTG
Description: During normal operation, if the RX FIFO becomes full and the protocol engine needs to send a command to the DMA state machine, it will wait in that state until the RX FIFO becomes not full. As the protocol state machine also handles the SOF generation, the SOFs will not be sent during this interval. If one SOF is missed, the host controller will issue a false babble detection. If more than 3.125ms are elapsed without SOFs, the peripheral will recognize the idle bus as a USB reset.

This issue only happens if the RX FIFO is full long enough for a SOF to be missed. For this to happen, the host controller must have lost access to the main bus and/or the RX FIFO is much too small, being this is a throughput issue. If using non-streaming mode, this issue does not apply.

Workaround: RX FIFO overflow events must be avoided.

Fix plan: No plans to fix.

SECF204: When using serial PHY interface the host may start transmitting next IN token before packet being received from device is finished

Errata type: Silicon
Affects: USB HOST / OTG
Description: When using ISO IN endpoints with MULT=3 and low bandwidth system bus access, the controller may enter into a wait loop situation without warning the software. Due to the low bandwidth the last packet from a mult3 sequence may not be fetched in time before the last token IN is received (for that uframe/endpoint). This will cause the controller to reply with a zero length packet (ZLP), breaking the prime sequence. The DMA state machine will not be warned of this situation and the controller will send a ZLP to all the following IN tokens for that endpoint. The transaction will not be completed because the DMA state machine is waiting for the TX complete command to come from the protocol engine, which will not happen (because it is unprimed).

Workaround: Software must set correct MULT, matching the number of packets to be transmitted in a given dTD.

Fix plan: No plans to fix.

SECF205: Bus masters may fetch corrupt data/instructions from DDR2 memory.

Errata type: Silicon
Affects: MAC_NET
Description: The DDR Memory Controller (DDRMC) may fetch corrupt data when both DDR ports are accessed by different bus masters concurrently. These two masters can generate read-to-write transactions with a single DDR cycle between transactions with both ports active. This short cycle may cause an errant data fetch.

Workaround: To Eliminate potential for this issue all of the following must be implemented.

1. Supply 1.3V to Ivdd.
2. Configure the DDRMC for 125 MHz operation, and bypass duty cycle correction:

   MISCCR2[DDR2CLK] = 0, MISCCR2[DCCBYP] = 1

3. Adjust DDRMC controller timing to match your DDR2 timing at 125 MHz.

Fix plan: Will be fixed.

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

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)
www.freescale.com/support

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
1-800-441-2447 or +1-303-675-2140
Fax: +1-303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductors 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.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale Semiconductor 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. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor 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 failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify Freescale Semiconductor 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 claims alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

RoHS-compliant and/or Pb-free versions of Freescale products have the functionality and electrical characteristics as their non-RoHS-complaint and/or non-Pb-free counterparts. For further information, see http://www.freescale.com or contact your Freescale sales representative.

For information on Freescale's Environmental Products program, go to http://www.freescale.com/epp.

freescale™
semiconductor