Freescale Semiconductor
Chip Errata

MCF5235DE
Rev. 5, 6/2011

MCF5235 Chip Errata

Silicon Revision: All

This document identifies implementation differences between the MCF523*x* processors and the description contained in the *MCF5235 ColdFire*® *Reference Manual*. Refer to http://www.freescale.com/coldfire for the latest updates.

All current MCF523x devices are marked as L23W mask set. The date code on the marking can be used to determine which errata have been corrected on a particular device, as shown in Table 1. The datecode format is XXXXYYWW, where YY represents the year and WW represents the work week. The four leading digits can be ignored.

Table 1. Summary of MCF523x Errata

Errata	Module Affected	Date Errata Added	Date Code Affected?			
			< XXXX0501	≥ XXXX0501 & < XXXX0511	≥ XXXX0511 & < XXXX1031	≥ XXXX1031
SECF005	Cache	8/11/04	Yes	Yes	Yes	Yes
SECF001	Cache	8/11/04	Yes	Yes	Yes	Yes
SECF009	FEC	8/11/04	Yes	Yes	No	No
SECF007	FEC	8/11/04	Yes	Yes	No	No
SECF028	FEC	8/11/04	Yes	Yes	No	No
SECF010	FEC	8/11/04	Yes	Yes	Yes	Yes
SECF026	eTPU	8/11/04	Yes	Yes	No	No
SECF027	eTPU	8/11/04	Yes	Yes	Yes	Yes
SECF030	PLL	12/30/05	Yes	No	No	No
SECF131	PLL	9/15/08	Yes	Yes	Yes	Yes
SECF163	PLL	3/4/10	Yes	Yes	Yes	No

The table below provides a revision history for this document.

Table 2. Document Revision History

Rev. No.	Date	Substantive Changes
0		Initial revision
0.1		Reworded errata description in SECF028
1		Updated errata with new revision of silicon.
1.1		Added SECF030
2	9/2008	Added SECF131
3	4/2010	Added SECF163
4	12/2010	Added XXXX1011 datecode which fixes SECF163
5	6/2011	Changed datecode which fixes SECF163 to XXXX1031

SECF001: Incorrect Operation of Cache Freeze (CACR[CFRZ])

Errata type: Silicon
Affects: Version 2 ColdFire Cache
Description: The cache on the V2 ColdFire core is controlled by the cache control register (CACR). When the CACR[CFRZ] bit is set, the cache freeze function is enabled and no valid cache array entry is displaced. However, this feature does not always work as specified, sometimes allowing valid lines to be displaced when CACR[CFRZ] is enabled.

This does not cause any corrupted accesses. However, there could be cache misses for data that was originally loaded into the cache but was subsequently deallocated, even though the CACR[CFRZ] bit was set.

Also, incoherent cache states are possible when a frozen cache is cleared via the CACR[CINV] bit.

Workaround: Unfreeze the cache by clearing CACR[CFRZ] when invalidating the cache using the CACR[CINV] bit

Workaround: Use the internal SRAM to store critical code/data if the system cannot handle a potential cache miss

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

SECF005: Possible Cache Corruption After Clearing Cache (Setting CACR[CINV])

Errata type: Silicon
Affects: Version 2 ColdFire Cache
Description: The cache on the V2 ColdFire core may function as either:

- a unified data and instruction cache
- an instruction cache
- a data cache

The cache function and organization is controlled by the cache control register (CACR). The CACR[CINV] bit causes a cache clear. If the cache is configured as a unified cache and the CINV bit is set, the scope of the cache clear is controlled by two other bits in the CACR:

- CACR[INVI] invalidates instruction cache only
- CACR[INVD] invalidates data cache only

If a write to the CACR is performed to clear the cache (CACR[CINV] = 1) and only a partial clear is done (CACR[INVI] or CACR[INVD] set), then cache corruption may occur.

Workaround: All loads of the CACR that perform a cache clear operation (CACR[CINV] set) should be followed immediately by a NOP instruction. This avoids the cache corruption problem.

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

SECF009: FEC Receive Buffer Overrun in 10BaseT Mode

Errata type: Silicon
Affects: FEC
Description: When the FEC is connected to a 10BaseT network, if length of the data stored in a descriptor is not evenly divisible by 16 (not line-aligned), the FEC writes extra lines at the end of the buffer. The entire line that contains the last valid data is written and at least one extra line, but up to four lines after the end of the valid data can also be written. In most cases, this is not a problem because the extra lines of data continue falling within the limits of the buffer. However, if the valid data ends near the end of the buffer, the extra lines written by the FEC might be outside of the data buffer. This leads to corruption of the next buffer, descriptor, data, or code stored in the adjacent memory.

For example, as shown in the figure below, if the max buffer size is programmed to 0x600 and a frame that is 0x5F8 bytes long is received, a line is written starting at buffer start + 0x5F0. The first half of the line at buffer start + 0x5F0 is valid frame data that should be processed by the FEC driver; the second half of the line is additional data that is written because the FEC only writes complete lines. This data should be ignored by the FEC driver. So far, this is correct FEC behavior as originally specified. However, the FEC repeats the last line of valid data a number of times. The line at buffer start + 0x600 is written, and as many as three additional lines beyond the end of the data buffer could be written.

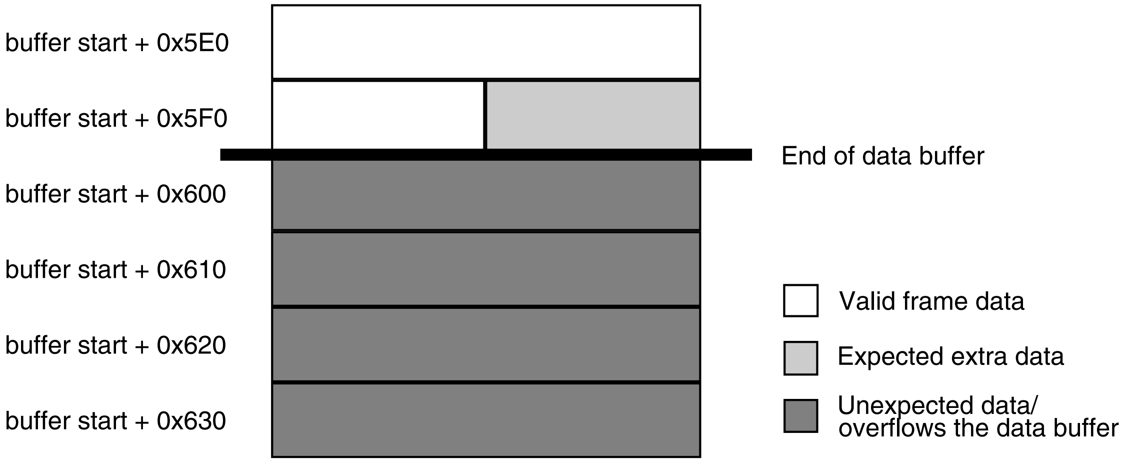

Figure 1. Buffer Overrun Example

Workaround: Only use 100BaseT.

Workaround: Allocate extra lines for the receive data buffers. The actual allocated memory for each buffer should be equal to the receive buffer size programmed in the FEC's EMRBR register plus four lines (16 byte-sized lines).

Workaround: Program the data buffer size one line larger than the max packet size (data buffer size = EMRBR + 0x40).

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

SECF007: Concatenation of Received Frames in 10BaseT Mode

Errata type: Silicon
Affects: FEC
Description: When the FEC is connected to a 10BaseT network, sometimes the FEC combines the data from multiple frames to generate a single frame. The data from the frames is received correctly, but the frame boundary is not reported correctly. This causes the descriptor to report the length as the data length for all of the concantenated frames added together. The incorrect data length might exceed the max frame length programmed in the RCR[MAX_FL] field.

When TCP is used as a transport mechanism, this errata manifests itself as lost packets and reduced throughput. Data continues to be received correctly because TCP requests retransmission of bad packets. However, UDP does not include any mechanism for packet retransmission, as it is a send and forget protocol. Consequently, while UDP should be able to identify an incorrectly received packet (because its checksum will fail), higher level software in the protocol stack must be capable of requesting retransmission to work around this errata.

Workaround: Higher level Ethernet layer code should compare the length reported by the descriptor to the length included in its header. If the lengths do not match, the packet should be truncated or discarded as needed. The protocol stack must be responsible for requesting retransmission of any frames that are discarded due to the data length mismatch.

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

SECF028: Internal Pull-ups on FEC Signals

Errata type: Silicon
Affects: FEC
Description: Some of the FEC pins have internal pull-up resistors of 18k ohms. This causes some of the common PHYs to be configured incorrectly. Internal pull-up resistors can be found on the following signals: ERXDV, ERXER, ECOL, ECRS.

Workaround: The FEC pins referenced above require a pull-down resistor for the PHYs to be configured properly.

Fix plan: Fixed in datecodes XX0511 and later.

SECF010: FEC Interrupts will not Trigger on Consecutive Transmit Frames

Errata type: Silicon
Affects: FEC
Description: The late collision (LC), retry limit (RL), and underrun (UN) interrupts do not trigger on consecutive transmit frames. For example, if back-to-back frames cause a transmit underrun, only the first frame generates an underrun interrupt. No other underrun interrupts are generated until a frame is transmitted that does not underrun or the FEC is reset.

Workaround: Because late collision, retry limit, and underrun errors are not directly correlated to a specific transmit frame, in most cases a workaround for this problem is not needed. If a workaround is required, there are two independent workarounds:

- Ensure that a correct frame is transmitted after a late collision, retry limit, or underrun errors are detected.
- Perform a soft reset of the FEC by setting ECR[RESET] when a late collision, retry limit, or underrun errors are detected.

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

SECF026: eTPU SCMMISF Asserted on Valid SCM Contents

Errata type: Silicon
Affects: eTPU
Description: After the first multiple input signature calculation (MISC) concludes, if the eTPU tries to access the shared code memory (SCM) at the same time as the next MISC calculation begins, the ETPUMCR[SCMMISF] flag could assert. This incorrectly indicates an error in the SCM.

Workaround: Before any channel is enabled for thread execution, disable the MISC feature by clearing ETPUMCR[SCMMISEN]. MISC can still be used on initialization for a SCM load check.

This errata does not require any action by the eTPU function library user, because it will be handled at the API level.

Fix plan: Fixed on datecodes XXXX0511 and later.

SECF027: eTPU Code That Jumps to an Invalid SCM Address Does Not Finish

Errata type: Silicon
Affects: eTPU
Description: If eTPU runaway code jumps to an out-of-range shared coded memory (SCM) address, the thread does not finish and neither of the channels flags are cleared. An illegal instruction is flagged, and a global exception is issued.

Workaround: Enable the eTPU global exception interrupt in interrupt controller (INTC). The exception handler should:

- Disable the channel by clearing ETPU_C*n*CR[CPR]
- Issue a forced end by setting ETPU_ECR[FEND].

This errata does not require any action by the eTPU function library user, because it will be handled at the API level.

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

SECF030: Frequency Modulation Mode on PLL Not Functional

Errata type: Silicon
Affects: FMPLL
Description: A short exists between internal test points within the frequency modulation (FM) portion of the PLL. The spacing between the shorted test points was not sufficient, preventing FM from operating as expected. This results in the PLL not locking after FM is enabled.

Workaround: No workaround.

Fix plan: Fixed in datecodes XX0511 and later.

SECF131: PLL Does Not Lock in Normal PLL Mode with External Clock Reference

Errata type: Silicon
Affects: PLL
Description: During a power on reset, if the CLKMOD[1:0] equals 10 setting is used (normal PLL mode with external clock reference), the PLL does not lock and the device never comes out of reset.

Workaround:

NOTE

If a workaround for errata SECF163 is implemented, a workaround for this errata is not necessary.

When configuring the PLL for normal PLL mode with external clock reference, tie CLKMOD1 to $\overline{\text{RESET}}$ and not straight to 3.3V. This allows the PLL to correctly detect the desired operating mode and lock.

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

SECF163: Some Devices May Not Exit Reset After Power On

Errata type: Silicon

Affects: PLL

Description: During power-on reset (POR), the frequency of the PLL's internal oscillator (ICO) may overrun the ability of the ICO's internal feedback to track to that frequency. This results in an open-loop condition where the frequency of the ICO remains at its upper limit. The internal feedback loop recovers, but might not be able to reach the target frequency and exit reset. This affects all PLL-enabled modes, but does not affect bypass mode.

Workaround: After $\overline{\text{RESET}}$ deasserts and before $\overline{\text{RSTOUT}}$ deasserts, change the target clock mode to a PLL-enabled mode in which the input clock is disabled for more than 160us. Then, change back to the target clock mode.

Table 3. Workaround #1

Start in	CLK MOD	After $\overline{\text{RESET}}$ deasserts and before $\overline{\text{RSTOUT}}$ deasserts, switch to	CLK MOD	After 160us, switch back to the target clock mode	CLK MOD
Normal mode with crystal reference	11	Normal mode with external reference	10	Normal mode with crystal reference	11
Normal mode with external reference	10	Normal mode with crystal reference	11	Normal mode with external reference	10
1:1 mode	01	Normal mode with crystal reference	11	1:1 mode	01

Workaround: At POR, start in a PLL-enabled mode in which the input clock is disabled. Keep $\overline{\text{RESET}}$ asserted for at least 160us. After $\overline{\text{RESET}}$ deasserts and before $\overline{\text{RSTOUT}}$ deasserts change to the desired clock mode.

Table 4. Workaround #2

Start in	CLK MOD	After $\overline{\text{RESET}}$ deasserts and before $\overline{\text{RSTOUT}}$ deasserts, switch to target clock mode	CLK MOD
Normal mode with external reference	10	Normal mode with crystal reference	11
Normal mode with crystal reference	11	Normal mode with external reference	10
Normal mode with crystal reference	11	1:1 mode	01

Fix plan: Fixed on datecodes XXXX1031 and later.

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