

MCF5282 Device Errata

This document identifies implementation differences between the MCF5282 processor and the description contained in the *MCF5282 ColdFire® Microcontroller User's Manual*. Refer to <http://motorola.com/ColdFire> for the latest updates. The errata items listed in this document (summarized in [Table 1](#)) describe differences from the following documents:

- MCF5282 ColdFire® Microcontroller User's Manual
- ColdFire Microprocessor Family Programmer's Reference Manual

Leakage Current on V_{DDPLL} pin

All current MCF5282 devices are marked as L95M 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 XXXYYWW, where YY represents the year and WW represents the work week. The three leading digits can be ignored.

Table 1. Summary of MCF5282 Errata

Errata ID	Module Affected	Date Errata Added	Date Code Affected			Errata Title
			<XXX0324	XXX0324 to XXX0326	≥XXX0327	
1	PLL	03/18/03	Yes	No	No	Leakage current on V _{DDPLL} pin
2	BDM	03/28/03	Yes	Yes	Yes	BDM load of SR does not enable stack pointer exchange
3	EMAC	03/28/03	Yes	Yes	No	Unexpected pipeline stall on EMAC load/store accumulator instruction
4	Cache	03/31/03	Yes	Yes	No	Incorrect cache size
5	Flash	04/09/03	Yes	Yes	No	Corrupted fetches from Flash
6	Cache	07/21/03	Yes	Yes	Yes	Possible cache corruption after setting CACR[CINV]
7	Cache	07/21/03	Yes	Yes	Yes	Incorrect operation of CACR[CFRZ]
8	FlexCAN	07/23/03	Yes	Yes	Yes	32-bit accesses to FlexCAN registers do not work properly
9	FEC	04/22/04	Yes	Yes	Yes	FEC receive buffer overrun in 10baseT
10	FEC	04/22/04	Yes	Yes	Yes	Concatenation of received frames in 10baseT
11	PLL	08/23/04	Yes	Yes	Yes	PLL does not lock when in "Normal PLL mode with external clock reference"

1 Leakage Current on V_{DDPLL} pin

1.1 Description

The MCF5282 exhibits a 65mA leakage current on the V_{DDPLL} supply regardless of chip configuration.

1.2 Workaround

No workaround.

DATECODES AFFECTED: XXX0323 and earlier

2 BDM Load of SR Does Not Enable Stack Pointer Exchange

2.1 Description

The V2 core used in the MCF5282 adds support for separate user and supervisor stack pointers. The hardware implements an active stack pointer and an “other_stack_pointer.” Whenever the operating mode of the processor changes (supervisor → user, user → supervisor), the processor hardware “exchanges” the active SP and the other SP.

This exchange operation does not work when the processor mode is changed by a write to the SR from the BDM port. The hardware in the processor core required to process the BDM load_SR operation and enable the stack pointer exchange is missing.

The exchange works properly when the SR is changed through software.

2.2 Workaround

Use software for any operations that require exchanging the stack pointers.

DATECODES AFFECTED: All

3 Unexpected Pipeline Stall on EMAC Load/store Accumulator Instruction

3.1 Description

An unexpected pipeline stall occurs for accumulator load and accumulator store instructions that immediately follow a load accumulator or MAC instruction.

Specifically, the operand execution pipeline (OEP) experiences a 2T pipeline stall when a load/store accumulator instruction enters the pipeline immediately after any load accumulator or MAC instruction. The pipeline is supposed to stall only if there is a store accumulator instruction immediately following a load or MAC instruction which updated the specified accumulator.

A simple example can be created to expose this problem:

```
mac.l ra,rb,acc0
mac.l rc,rd,acc0
mov.l acc1,rx
```

In the above example, the store of acc1 (mov.l acc1,rx) should not experience any stall since that accumulator is not being updated. In the current V2 + EMAC implementation, it incorrectly stalls for two cycles.

NOTE

The operation of the instructions is correct. The problem is that the expected timing is not met.

3.2 Workaround

No workaround.

DATECODES AFFECTED: XXX0326 and earlier

4 Incorrect Cache Size

4.1 Description

The MCF5282 operates as if it were connected to an 8KB cache; however, the cache size is in fact 2KB. Once the 2KB cache is full, the cache controller can have erroneous hits in the cache space resulting in data and/or instruction corruption.

4.2 Workaround

Do not enable the cache.

DATECODES AFFECTED: XXX0326 and earlier

5 Corrupted Fetches from Flash

5.1 Description

Leaving bit 6 in the FLASHBAR register cleared can cause corrupted fetches from the MCF5282's internal Flash. For datecodes after XXX0327, the bit is hardwired high to prevent the corrupted accesses.

5.2 Workaround

Set bit 6 in the FLASHBAR. This will prevent the corrupted fetches.

DATECODES AFFECTED: XXX0326 and earlier

6 Possible Cache Corruption after Setting CACR[CINV]

6.1 Description

The cache on the MCF5282 was enhanced to function as a unified data and instruction cache, an instruction cache, or an operand cache. The cache function and organization is controlled by the cache control register (CACR). The CINV (Bit 24 = cache invalidate) bit in the CACR 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, INVI (BIT 21 = CINV instruction cache only) and INVD (BIT 20 = CINV data cache only). These bits allow the entire cache, just the instruction portion of the cache, or just the data portion of the cache to be cleared. If a write to the CACR is performed to clear the cache (CINV = BIT 24 set) and only a partial clear will be done (INVI = BIT 21 or INVD = BIT20 set), then cache corruption may occur.

6.2 Workaround

All loads of the CACR that perform a cache clear operation (CINV = BIT 24) should be followed immediately by a NOP instruction. This avoids the cache corruption problem.

DATECODES AFFECTED: All

7 Incorrect Operation of CACR[CFRZ]

7.1 Description

The cache on the ColdFire V2 is controlled by the cache control register (CACR). When CACR[CFRZ] is set, the cache freeze function is enabled and no valid cache array entry will be displaced. However, this feature does not work as specified, sometimes allowing valid lines to be displaced when CACR[CFRZ] is enabled.

This will 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 CINV (bit 24 = cache invalidate) bit in the CACR.

7.2 Workaround

- Unfreeze the cache by clearing CACR[CFRZ] when invalidating the cache using the CACR[CINV] bit.
- Use the internal SRAM to store critical code/data if the system cannot handle a potential cache miss.

DATECODES AFFECTED: All

8 32-bit Accesses to FlexCAN Registers do not Work Properly

8.1 Description

Since the FlexCAN was originally designed for 16-bit architectures, all 32-bit register accesses are broken down into two back-to-back 16-bit accesses. However, the timing for the back-to-back accesses is incorrect and leads to corruption of the second 16-bit read or write.

8.2 Workaround

When reading or writing to the 32-bit RxMASK registers, use two 16-bit accesses instead of a single 32-bit access.

DATECODES AFFECTED: All

9 FEC Receive Buffer Overrun in 10BaseT Mode

9.1 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), then the FEC will write 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, since the extra lines of data still fall within the limits of the buffer. However, if the valid data ends near the end of the buffer, then 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 [Figure 1](#), if the max buffer size is programmed to 0x600 and a frame that is 0x5F8 bytes long is received, then 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 will only write complete lines. This data should be ignored by the FEC driver. So far, this is correct FEC behavior as originally specified. However, the FEC will repeat the last line of valid data a number of times. The line at buffer start + 0x600 will be written, and as many as three additional lines beyond the end of the data buffer could be written.

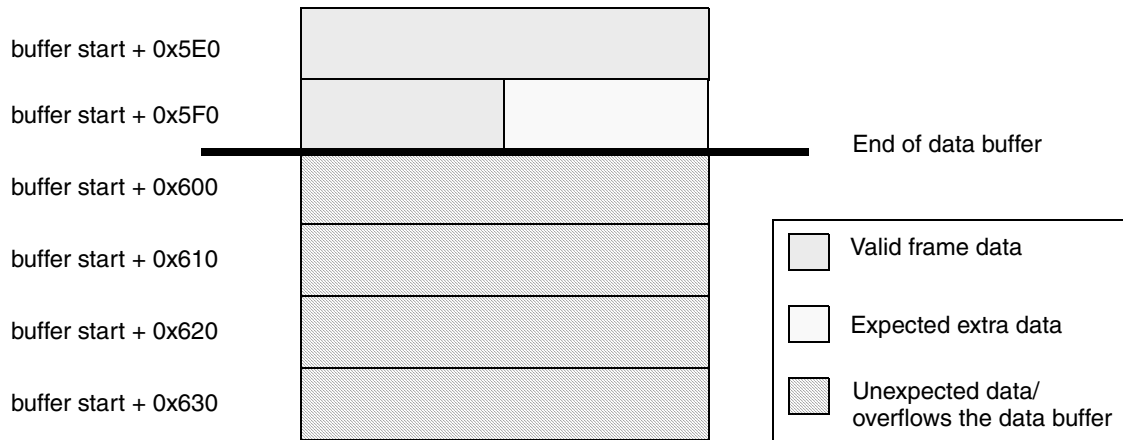


Figure 1. Buffer Overrun Example

9.2 Workaround

There are three independent workarounds:

1. Only use 100BaseT.
2. 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).
3. Program the data buffer size one line larger than the max packet size (data buffer size = EMRBR + 0x40).

STATUS: Will be fixed

DATECODES AFFECTED: All

10 Concatenation of Received Frames in 10BaseT Mode

10.1 Description

When the FEC is connected to a 10BaseT network, sometimes the FEC will combine 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 concatenated 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 will manifest itself as lost packets and reduced throughput. Data will still ultimately be received correctly, because TCP will request 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 a packet that is received incorrectly

PLL Does not Lock When in "Normal PLL mode with External Clock Reference"

(because its checksum will fail), higher level software in the protocol stack must be capable of requesting retransmission to work around this errata.

10.2 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, then 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.

STATUS: Will be fixed

DATECODES AFFECTED: All

11 PLL Does not Lock When in "Normal PLL mode with External Clock Reference"

11.1 Description

During a power on reset, if the CLKMOD[1:0] = 10 setting is used (normal PLL mode with external clock reference), then the MCF5282 PLL does not lock and the device never comes out of reset.

11.2 Workaround

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

STATUS: Will be fixed

DATECODES AFFECTED: All

THIS PAGE INTENTIONALLY LEFT BLANK



THIS PAGE INTENTIONALLY LEFT BLANK

THIS PAGE INTENTIONALLY LEFT BLANK

How to Reach Us:

Home Page:

www.freescale.com

E-mail:

support@freescale.com

USA/Europe or Locations Not Listed:

Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

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)
support@freescale.com

Japan:

Freescale Semiconductor Japan Ltd.
Technical Information Center
3-20-1, Minami-Azabu, Minato-ku
Tokyo 106-0047, Japan
0120 191014 or +81 3 3440 3569
support.japan@freescale.com

Asia/Pacific:

Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

For Literature Requests Only:

Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor 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 and all 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 the 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 and hold 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 claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.



Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © Freescale Semiconductor, Inc. 2004. All rights reserved.

MCF5282DE
Rev. 1.5
08/2004