Errata

MCF5282DE
Rev. 1.3, 9/2003

MCF5282 Device Errata

MOTOROLA intelligence everywhere™ | digital dna™

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*

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

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.

This is the old errata description from the initial release of the device errata.

The Flash controller includes a number of performance-enhancing features, including logic that generates "speculative" Flash reads and logic that detects consecutive accesses to the same longword address. In some cases, the speculative reads can cause corrupted instruction fetches and/or operand read 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

THIS PAGE INTENTIONALLY LEFT BLANK

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

MOTOROLA

MCF5282DE, **Rev. 1.3, 9/2003**