

MCF5208 Device Errata

Supports MCF5207 & MCF5208

by: Microcontroller Solutions Group

This document identifies implementation differences between the MCF5208 and MCF5207 processors and the description contained in the *MCF5208 ColdFire® Reference Manual*. Refer to <http://www.freescale.com/coldfire> for the latest updates. The errata items listed in this document (summarized in [Table 1](#)) describe differences from the following documents:

- *MCF5208 ColdFire® Reference Manual*
- *ColdFire Microprocessor Family Programmer's Reference Manual*

Table of Contents

1	PLL Behavior in Stop Mode	2
2	SDRAMC Incorrect Operation After Exiting Limp Mode	3
3	Possible Cache Corruption After Clearing Cache (Setting CACR[CINV])	4
4	Incorrect Operation of Cache Freeze (CACR[CFRZ])	4
5	Late Collision, Retry Limit, and Underrun FEC Interrupts Will Not Trigger on Consecutive Transmit Frames.	5
6	No Bus Monitor or Watchdog Recovery for Hung FlexBus Accesses.	6
7	Potential boot failure when using 32-bit wide SDRAM	6
8	Programmable Address Hold Does Not Function Correctly	7
9	Level 2 Trigger Operation Controlled by TDR[31].	8
10	Document Revision History.	8

PLL Behavior in Stop Mode

Early MCF5208 devices are marked as M28B mask set. The latest version of the MCF5208 is marked as 2M28B.

Table 1. Summary of MCF5208 Errata

Errata ID	Module Affected	Date Errata Added	Mask Set Affected?		Errata Title
			M28B	2M28B	
1	Clock	08/11/05	Yes	No	PLL behavior in STOP mode
2	SDRAMC	10/04/05	Yes	No	SDRAMC incorrect operation after exiting LIMP mode
3	Cache	01/23/06	Yes	Yes	Possible cache corruption after setting CACR[CINV]
4	Cache	01/23/06	Yes	Yes	Incorrect operation of CACR[CFRZ]
5	FEC ¹	01/23/06	Yes	Yes	Late collision, retry limit, and underrun interrupts will not trigger on consecutive transmit frames
6	FlexBus	01/23/06	Yes	Yes	No bus monitor or watchdog recovery for hung FlexBus accesses
7	SDRAMC	06/14/06	Yes	No	Potential boot failure when using 32-bit wide SDRAM
8	FlexBus	06/14/06	Yes	No	Programmable address hold does not function correctly
9	Debug	03/20/08	Yes	Yes	Level 2 Trigger Operation Controlled by TDR[31]

NOTES:

¹ Errata applies to MCF5208 only.

The chip identification register (CIR) can also be used to determine the silicon revision. [Table 2](#) list the CIR[PRN] field values that correspond to given datecode ranges.

Table 2. CIR[PRN] to Datecode

CIR[PRN] Value	Mask Set
0	M28B
2	2M28B

1 PLL Behavior in Stop Mode

1.1 Description

If the PLL is disabled in stop mode, then the device will exit stop mode with its booted frequency regardless of the operating frequency prior to entering stop mode. Entering stop mode with an

LPCR[STPMD] setting of 0b10 or 0b11 will cause a hard reset to the PLL module. When stop mode is exited, the PLL will re-latch the reset configuration setting for the PLL. For example:

1. The device is booted with a 166.67/83.33 MHz frequency selected (D1 and \overline{RCON} asserted upon reset).
2. The device is placed into limp mode to change the operating frequency to 144/72 MHz and limp mode is exited.
3. The LPCR is programmed to disable the PLL in stop mode (LPCR = 0xD0 or 0xD8).
4. The stop instruction is executed.
5. An interrupt is generated to exit the processor from stop mode.
6. The device will exit stop mode at an operating frequency of 166.67/83.33 MHz instead of 144/72 MHz.

1.2 Workaround

If the PLL frequency has not been modified since the last reset or if limp mode is being used, then no workaround is needed.

If the PLL was reprogrammed after reset, then the PLL needs to be reprogrammed after exiting stop mode. The following steps show how this can be done:

1. Program the LPCR register to disable the PLL in stop mode (LPCR = 0xD0 or 0xD8).
2. Save the values of the PLL registers.
3. Place the device into limp mode by clearing the MISCCR[LIMP] bit.
4. Execute the STOP instruction.
5. After stop mode is exited, reload the PLL with the register values saved in step 2.
6. Exit limp mode by clearing the MISCCR[LIMP] bit.

1.3 Status

Fixed starting with 2M28B mask set.

2 SDRAMC Incorrect Operation After Exiting Limp Mode

2.1 Description

The circuitry that controls the SDRAMC's SD_DQS signals will attempt to lock to the clock when the device is in limp mode. The large difference in clock speeds between limp mode and normal operation will cause the SD_DQS logic to become unsynchronized when limp mode is exited.

2.2 Workaround

After exiting LIMP mode, the value of 0x40000000 should be written to address 0xFC0A8080 before attempting to initialize the SDRAMC or exit the SDRAM from self-refresh mode.

2.3 Status

Fixed starting with 2M28B mask set.

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

3.1 Description

The cache on the V2 ColdFire core was enhanced to function as a unified data and instruction cache, an instruction cache, or a data cache. The cache function and organization is controlled by the cache control register (CACR). The CACR[CINV] 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 and CACR[INVD] invalidates 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 (CACR[CINV] = 1) and only a partial clear is done (CACR[INVI] or CACR[INVD] set), then cache corruption may occur.

3.2 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.

3.3 Status

Currently, there are no plans to fix this.

4 Incorrect Operation of Cache Freeze (CACR[CFRZ])

4.1 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 will be displaced. However, this feature does not always 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 CACR[CINV] bit in the CACR.

4.2 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.

4.3 Status

Currently, there are no plans to fix this.

5 Late Collision, Retry Limit, and Underrun FEC Interrupts Will Not Trigger on Consecutive Transmit Frames

5.1 Description

The late collision (LC), retry limit (RL), and underrun (UN) interrupts will not trigger on consecutive transmit frames. For example, if back-to-back frames cause a transmit underrun, only the first frame will generate an underrun interrupt. No other underrun interrupts will be generated until either a frame is transmitted that does not underrun, or the FEC is reset.

5.2 Workaround

Since 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, then 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 is detected.

5.3 Status

Currently, there are no plans to fix this.

6 No Bus Monitor or Watchdog Recovery for Hung FlexBus Accesses

6.1 Description

This device does not include a bus monitor or watchdog timer that is capable of forcing the termination of a hung FlexBus access. There are two possible cases that could lead to a hung FlexBus access:

1. If an access to one of the FlexBus memory areas (0x0000_0000–0x3FFF_FFFF or 0xC000_0000–0xDFFF_FFFF) does not hit in the valid range for one of the chip selects, a 32-bit access with no chip select asserted and no internal acknowledge is generated. The resulting bus cycle continues indefinitely waiting for the assertion of the \overline{TA} signal (external acknowledge).
2. If a chip select is configured for external termination, but no termination is received or a timing problem prevents \overline{TA} from being recognized, the bus cycle is hung and continues indefinitely.

There is no on-chip fail-safe to exit from the hung bus state. The system needs to either assert \overline{TA} to terminate the hung cycle or reset the entire device.

6.2 Workaround

1. Avoid issuing non-terminated FlexBus accesses. Ideally, a complete system will not access unused memory areas. This is harder to achieve while debugging, but as long as the debugger has access to the \overline{TA} signal on the BDM header, it should be able to force termination of any hung bus cycles.
2. Use an external bus monitor circuit that can detect hung bus cycles and generate a \overline{TA} to terminate the access. The bus monitor circuit can optionally assert an external interrupt signal along with \overline{TA} to indicate the error condition back to the CPU.
3. If there are unused chip selects in the system, then they can be mapped over unused memory areas. The only purpose of these chip selects would be to terminate accesses to addresses that are not used. BDM address breakpoints can then be used to generate an interrupt error.

6.3 Status

Currently, there are no plans to fix this.

7 Potential boot failure when using 32-bit wide SDRAM

7.1 Description

If the SD_CKE signal to the SDRAM deasserts while one or more banks are active, then the SDRAM enters a “clock suspend” state. If the SDRAM was driving the data lines before entering the clock suspend state, then the buffers will continue to drive.

During a reset, the processor will deassert SD_CKE without any graceful stop period to ensure that the SDRAM banks are all in an IDLE state. In some cases the SDRAM could be driving the data bus during and immediately after reset. This can lead to possible bus contention while latching reset configuration (RCON) values and/or while reading boot code from flash, and thereby cause the processor to enter an undefined state.

7.2 Workaround

- Use a split bus mode configuration with either DDR SDRAM or SDR SDRAM. This creates a dedicated 16-bit port for the SDRAM on D[31:16]. In this configuration the SDRAM doesn't share data lines with other devices so bus contention is not an issue.
- Additional workarounds TBD. We are investigating workarounds that can be used for 32-bit wide SDRAM configurations.

7.3 Status

Fixed starting with 2M28B mask set.

8 Programmable Address Hold Does Not Function Correctly

8.1 Description

The programmable address hold feature for the FlexBus chip selects does not function correctly. The address will be held for one clock after the chip select deasserts regardless of the address hold value programmed in the CSCR[WRAH] or CSCR[RDAH] fields. The address hold fields will still create a delay at the end of the bus cycle, so they can still be used to ensure a minimum delay between bus cycles.

8.2 Workaround

No workarounds.

8.3 Status

Fixed starting with 2M28B mask set.

9 Level 2 Trigger Operation Controlled by TDR[31]

9.1 Description

The TDR[L2T] bit (TDR bit 15) has no effect on the level 2 trigger. Bit 31 of the TDR register provides both trigger response control and logical operation of the level 2 trigger.

9.2 Workaround

Use the TDR[31] bit to control the logical operation for the level 2 trigger as follows:

- 0 — Level 2 trigger = PC_condition & Address_range & Data_condition
- 1 — Level 2 trigger = PC_condition | (Address_range & Data_condition)

Since TDR[31] is also part of the trigger response control, only certain combinations of trigger responses and logical operations are available as shown below:

Table 3. TDR[31:30] Definitions

TDR[31:30]	Level 2 Trigger	Trigger Response
00	PC_cond & (Add_range & Data_cond)	Display on DDATA
01		Processor Halt
10	PC_cond (Add_range & Data_cond)	Debug Interrupt
11		Reserved

9.3 Status

Currently, there are no plans to fix this.

10 Document Revision History

Table 4 provides a revision history for this document.

Table 4. Document Revision History

Rev. No.	Substantive Change(s)
1.0	Initial release.
1.1	Added errata: "SDRAMC incorrect operation after exiting LIMP mode."
1.2	Added erratas: <ul style="list-style-type: none"> • "Possible cache corruption after setting CACR[CINV]" • "Incorrect operation of CACR[CFRZ]" • "Late collision, retry limit, and underrun interrupts will not trigger on consecutive transmit frames" • "No bus monitor or watchdog recovery for hung FlexBus accesses"

Table 4. Document Revision History (continued)

Rev. No.	Substantive Change(s)
1.3	Added erratas: <ul style="list-style-type: none"><li data-bbox="427 304 1177 336">• Section 7, "Potential boot failure when using 32-bit wide SDRAM"<li data-bbox="427 336 1193 367">• Section 8, "Programmable Address Hold Does Not Function Correctly"
1.4	Updated errata to reflect fixed bugs for the 2M28B mask set.
2	Added errata: <ul style="list-style-type: none"><li data-bbox="427 455 1096 487">• Section 9, "Level 2 Trigger Operation Controlled by TDR[31]"

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.
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 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. 2008. All rights reserved.