MCF53017 Chip Errata

Silicon Revision: All

Supports: MCF53010, MCF53011, MCF53012, MCF53013, MCF53014, MCF53015, MCF53016, and MCF53017

Introduction

This document identifies implementation differences between the MCF5301*x* processors and the description contained in the *MCF5301x ColdFire® Reference Manual*. Refer to http://www.freescale.com/coldfire for the latest updates.

Summary of MCF5301*x* Errata

All current MCF5301x devices are marked as 1M23H mask set.

Table 1. Summary of MCF5301*x* Errata

Errata	Module Affected	Date Errata Added	Revision Affected?
			1M23H
SECF008	FlexBus	11/2/07	Yes
SECF145	RNG	6/6/08	Yes
SECF146	I/O	3/25/08	Yes
SECF180	Interrupt Controller	8/12/10	Yes
SECF189	208 LQFP	10/13/2010	Yes

You can also use the chip identification register (CIR) to determine the silicon revision. The table below lists the CIR[PRN] field values that correspond to given revisions.

Table 2. CIR[PRN] to Revision

CIR[PRN] value	Mask
1	1M23H

Revision History

The table below provides a revision history for this document.

Table 3. Document Revision History

Rev. No.	Date	Substantive Changes
1	8/2009	Initial public revision
2	8/2010	Added SECF180.
3	10/2010	Added SECF189.

SECF008: Programmable Address Hold Does Not Function Correctly

Errata type: Silicon

Affected component: FlexBus

Description: The programmable address hold feature for the FlexBus chip selects does not function correctly. The address is 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 creates a delay at the end of the bus cycle. They can continue to be used to ensure a minimum delay between bus cycles.

Workaround: No workarounds.

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

SECF145: RNG Does Not Implement NIST-Approved Random Number Generator

Errata type: Silicon

Affected component: RNG

Description: The RNG was designed to implement one of the NIST-approved deterministic random number generators (specifically, FIPS186-2, Appendix 3.1). It was discovered than RNGB incorrectly implements this algorithm.

To describe the errata, two variables need to be introduced: XKEY is the 256-bit internal seed value, and HASH is the 160-bit output of SHA-1.

- Let XKEY = X0 || X1 || X2 || X3 || X4 || X5 || X6 || X7, where each Xi is a 32-bit word.
- Let HASH = H0 || H1 || H2 || H3 || H4, where each Hi is a 32-bit word.

FIPS186-2, Appendix 3.1 includes a step for updating the seed after each iteration.

```
XKEY = XKEY + HASH + 1
```

The RNG actually defines XKEY' and HASH' as:

```
XKEY' = X7 || X6 || X5 || X4 || X3 || X2 || X1 || X0
HASH' = H4 || H3 || H2 || H1 || H0
```

RNG then calculates the updated seed as:

```
XKEY' = XKEY' + HASH' + 1
```

As can be seen from the above, the RNG currently performs the addition in the wrong direction.

The seed update function used in the RNG is equivalent to the actual update seed function. So, the RNG does generate a stream of random data with the equivalent cryptographic strength of the NIST approved algorithm.

Due to this slightly different implementation of the algorithm, this version of the RNG cannot be given a certificate that is passes the NIST Algorithm Validation Suite.

Workaround: To work around this errata requires the user to implement the correct equation in a software routine.

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

SECF146: Latch-up Susceptibility When Maximum I/O Sink Current Is Exceeded

Errata type: Silicon

Affected component: I/O

Description: Various pins are susceptible to latch-up when the current injection exceeds 60 mA at 85°C per pin on the pins listed in Table 3. Per the JESD78A, the maximum allowable latch-up current per pin is 100 mA at 85°C.

Table 4. Pins Susceptible to Latch-Up

Signal Name	256-MAPBGA Pin Location	Signal Name	256-MAPBGA Pin Location
$\overline{\text{IRQ04}}$	D1	$\overline{\text{IRQ1DEBUG5}}$	H3
SIM1_VEN	D16	$\overline{\text{IRQ1DEBUG6}}$	H4
$\overline{\text{IRQ07}}$	E4	$\overline{\text{IRQ1DEBUG0}}$	J13
SIM1_RST	E13	$\overline{\text{IRQ1DEBUG3}}$	K14
SIM1_DATA	E14	$\overline{\text{IRQ1DEBUG2}}$	H14
SIM1_PD	E15	$\overline{\text{IRQ1DEBUG1}}$	K15
U2TXD	F2	I2C_SDA	K3
$\overline{\text{IRQ01}}$	F4	SIM0_DATA	L3
SIM1_CLK	F13	$\overline{\text{IRQ06}}$	L13
DSPI_PCS1	F14	SIM0_PD	L14
DSPI_SOUT	F15	I2C_SCL	M1
SIM0_RST	F16	SIM0_VEN	M2
$\overline{\text{U0CTS}}$	G4	SIM0_CLK	M16
DSPI_PCS0	G13	$\overline{\text{RSTOUT}}$	N1
$\overline{\text{IRQ1DEBUG7}}$	H1	TDI	T4
$\overline{\text{IRQ1DEBUG4}}$	H2	PSTCLK	T5

Workaround: No workaround.

Fix plan: Will be fixed on the next revision.

SECF180: Spurious Interrupts Can Cause Incorrect Vector Fetch

Errata type: Silicon

Affected component: 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.

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 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, and 24 (the correct spurious interrupt vector) should point to the same handler.

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

SECF189: 208-LQFP: Modules not available in 208 LQFP package

Errata type: Silicon

Affected component: 208-LQFP

Description:	The following devices available in the 208 LQFP package: MCF53010, MCF53011, MCF53012, and MCF53013 do not have the SSI, RTC, and eSDHC modules enabled. This is because of errors in the design and definition of the device for the 208-LQFP package. The SIM module is also impacted because the RTC is required for auto powerdown mode. This feature will not be available. The eSDHC pins on the package will not be available as GPIO, so leave them as "no connects."
Workaround:	The only currently available workaround is to use the MAPBGA packaged device as it is not impacted by this issue.

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

Document Number: MCF53017DE
Rev. 3, 10/2010