Mask Set Errata for Mask 3M12J

This report applies to mask 3M12J for these products:

- MCF51QE128
- MCF51QE96
- MCF51QE64
- MCF51QE32

The mask set is identified by a 5-character code consisting of a version number, a letter, two numerical digits, and a letter, for example 0J27F. All standard devices are marked with a mask set number and a date code.

Device markings indicate the week of manufacture and the mask set used. The date is coded as four numerical digits where the first two digits indicate the year and the last two digits indicate the work week. For instance, the date code "0301" indicates the first week of the year 2003.

Some MCU samples and devices are marked with an SC, PC, or XC prefix. An SC prefix denotes special/custom device. A PC prefix indicates a prototype device which has undergone basic testing only. An XC prefix denotes that the device is tested but is not fully characterized or qualified over the full range of normal manufacturing process variations. After full characterization and qualification, devices will be marked with the MC or SC prefix.

SE157-ADC-INCORRECT-DATA: Boundary case may result in incorrect data being read in 10- and 12-bit modes

Errata type: Silicon
Affects: ADC
Description: In normal 10-bit or 12-bit operation of the ADC, the coherency mechanism will freeze the conversion data such that when the high byte of data is read, the low byte of data is frozen, ensuring that the high and low bytes represent result data from the same conversion.

In the errata case, there is a single-cycle (bus clock) window per conversion cycle when a high byte may be read on the same cycle that subsequent a conversion is completing. Although extremely rare due to the precise timing required, in this case, it is possible that the data transfer occurs, and the low byte read may be from the most recently completed conversion.

In systems where the ADC is running off the bus clock, and the data is read immediately upon completion of the conversion, the errata will not occur. Also, in single conversion mode, if the data is read prior to starting a new conversion, then the errata will not occur.

The errata does not impact 8-bit operation.

Introducing significant delay between the conversion completion and reading the data, while a following conversion is executing/pending, could increase the probability for the errata to occur. Nested interrupts, significant differences between the bus clock and the ADC clock , and not handling the result register reads consecutively, can increase the delay and therefore the probability of the errata occuring.

Workaround: Using the device in 8-bit mode will eliminate the possibility of the errata occuring.

Using the ADC in single conversion mode, and reading the data register prior to initiating a subsequent conversion will eliminate the possibility of the errata occuring.

Minimizing the delay between conversion complete and processing the data can minimize the risk of the errata occuring. Disabling interrupts on higher priority modules and avoiding nested interrupts can reduce possible contentions that may delay the time from completing a conversion and handling the data. Additionally, increasing the bus frequency when running the ADC off the asynchronous clock, may reduce the delay from conversion complete to handling of the data.

SE156-ADC-COCO: COCO bit may not get cleared when ADCSC1 is written to

Errata type: Silicon
Affects: ADC
Description: If an ADC conversion is near completion when the ADC Status and Control 1 Register (ADCSC1) is written to (i.e., to change channels), it is possible for the conversion to complete, setting the COCO bit, before the write instruction is fully executed. In this scenario, the write may not clear the COCO bit, and the data in the ADC Result register (ADCR) will be that of the recently completed conversion.

If interrupts are enabled, then the interrupt vector will be taken immediately following the write to the ADCSC1 register.

Workaround: It is recommended when writing to the ADCSC1 to change channels or stop continuous conversion, that you write to the register twice. The first time should be to turn the ADC off and disable interrupts, and the second should be to select the mode/channel and re-enable the interrupts.

SE141-DEBUG: Debug Trace Captures Incorrect Data on Access Errors

Description: Memory referencing operations that are targeted for debug trace data capture but receive an access error on their memory reference incorrectly perform the debug trace data capture. This puts corrupt information in the trace buffer. The trace buffer should not capture information from memory accesses that receive access error

Workaround: Do not have debug trace information capture enabled for accesses to memory locations that might fault.

or

Do not depend on the content of the trace buffer for information captured on an access error.

Note first that the default ColdFire V1 configuration responds to any access error by causing a reset.

Also note that, in general, access errors are not recoverable. It is believed that this behavior will not adversely affect debug.

SE140-MEMORY: Certain Read Memory Instructions Incorrectly Update Target Register and Condition Codes on Access Errors

Description: Certain cases of memory read instructions that receive an access error on their read operation incorrectly update their target register. This corrupts the content of this register, which should be unchanged by a memory read operation that receives an access error. The condition codes affected by these instructions are also incorrectly updated.

The problem instructions are mov.b, mov.w, mvz.b, mvz.w, mvs.b, mvs.w, tst.b, and tst.w.

Workaround: Do not use these instructions to access memory locations that might fault.

or

Do not depend on the content of the target register or condition codes immediately after these instructions receive an access error.

Note first that the default ColdFire V1 configuration responds to any access error by causing a reset.

Also note that, in general, read access errors are not recoverable. It is believed that most software currently follows the second workaround and just aborts the faulting software module or resets on an access error. Very specific software modules that disable resetting on access errors to size memory, check for existing peripherals, or some

SE139-TPM: Incorrect 16-Bit Write to 16-Bit TPM Registers via BDM Single Step

Description: Due to an integration error between the ColdFire V1 BDM access and TPM coherency mechanism, if a user conducts BDM single stepping through software that executes 16-bit writes to 16-bit TPM registers (TPMxMOD and TPMxCxV), reading back the registers will show an invalid value. The read will show that only the data associated with the low byte of the TPM register was written correctly and the high byte remains unchanged. This issue occurs only in BDM mode, therefore 16-bit writes to any 16-bit TPM register (TPMxMOD or TPMxCxV) in user mode will be valid.

Workaround: Executing 8-bit writes to appropriate high and low registers of 16-bit TPM registers (TPMxMODH, TPMxMODL, TPMxCxVH, and TPMxCxVl) will avoid the errata in BDM mode.

Another workaround is for the user to set a breakpoint after the 16-bit TPM registers and run the software to the breakpoint. This method does not require a software change for debugging.

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Information in this document is provided solely to enable system and software implementers to use Freescale products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. Freescale reserves the right to make changes without further notice to any products herein.

Freescale makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale 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 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 does not convey any license under its patent rights nor the rights of others. Freescale sells products pursuant to standard terms and conditions of sale, which can be found at the following address: www.freescale.com/salestermsandconditions.