Mask Set Errata for Mask 1N50A

Introduction

This report applies to mask 1N50A for these products:

- MC9S08AC16
- MC9S08AC8
- MC9S08AW16A
- MC9S08AW8A

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.

SE181-ICG-RST: ICG HGO = 1 long reset

Errata type: Silicon
Affects: ICG, Reset
Description: When the ICG (Internal Clock Generator) is configured in FBE (FLL bypassed external) mode and the HGO (high gain option) bit is set, an internally generated reset may cause a reset (low) pulse that is much longer than a typical 9 µs to 20 µs pulse width. The Reset low time can extend to many seconds at low temperatures (<0°C) if the oscillations on the EXTAL pin are attenuated due to series resistance or oscillator loading.

© 2013 Freescale Semiconductor, Inc.

The ICG contains logic to switch from the external clock to an internal clock when Reset is asserted (low) in FBE mode. This switch requires 2 full clocks to operate properly. However, in FBE mode with HGO = 1, the external clock input signal (EXTAL) may not have sufficient voltage swing, particularly the low level, to properly clock the logic just after Reset is asserted.

Workaround: The best workaround is to use HGO = 0 if FBE mode is desired. Note that when HGO = 0 is selected, the series resistor (RS) must be 0 Ω. This workaround is effective because the oscillation levels at the EXTAL pin are sufficient to clock the internal logic when Reset is asserted.

An alternate workaround is to use HGO = 1 with RS = 0 Ω and crystal/resonator load components that guarantee that the EXTAL low level is below 0.5 V.

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

Errata type: Silicon
Affects: ADC
Description: In normal 10-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 occurring.

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

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 occurring.

Minimizing the delay between conversion complete and processing the data can minimize the risk of the errata occurring. 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.

SE133-FLASH: Unexpected Flash Block Protection Errors

Errata type: Silicon
Affects: Flash
Description: If a portion of the nonvolatile memory (NVM) is block protected, unexpected flash block protect violation (FPVIOL) errors can result. These errors can occur during an attempt to program or erase locations in areas of the NVM that are not block protected. Software methods can be used to avoid this potential problem. The problem is more likely to be seen on devices that have multiple nonvolatile blocks, including devices with two or more separate flash blocks or with flash plus EEPROM. If block protection is not enabled, no errors occur.

This error is related to logic that compares current block protection settings to an internally latched address. This internal address is written (latched) at reset, at the end of most flash commands, and whenever there is a write to a location in NVM. If a read access to the partially protected NVM is performed immediately before the write to unprotected memory that starts a new flash command, the erroneous address that was previously in the internal latch can cause a false indication of a protection violation. A short sequence of instructions can be performed before starting normal flash commands to ensure that the address in the internal latch is not a protected address.

Workaround: The preferred workaround starts a command to a known unprotected address (which internally latches the known-unprotected address), forces an access error to abort that command, and then clears the resulting error flags before starting any new flash command. This workaround assumes the H:X index register points to the location or sector you want to program or erase, and accumulator A has the data value you plan to write to that location. Start your program or erase routine with the following instructions.

```
 STA    ,X     ;latch the unprotected address from H:X
               NOP           ;brief delay to allow the command state
machine to start
               STA    ,X     ;intentionally cause an access error to abort
this command
               PSHA          ;temporarily save data value
               LDA    #$30   ;1's in PVIOL and ACCERR bit positions
               STA    FSTAT  ;clear any error flags
               PULA          ;restore data value
               STA    ,X     ;STEP 1 write data to start new command
```

The only new instructions compared to the normal routine for flash commands are the first three instructions, which take three bytes of code space and five bus cycles. These instructions may be located anywhere in memory, including in the protected area of the flash memory.

SE194-STOP3DCO: Stop3 Mode DCO Erratum

Errata type: Silicon
Affects: ICG, stop3
Description: On some devices, during stop3 recovery in FEI, FEE, and SCM modes, the DCO clock can generate high frequency pulses that can cause the MCU to enter a code runaway condition due to excessive bus speed. This condition can occur when the Reduced Frequency Divider value (R) in the ICGC2 register is 1, 2, or 4. The Multiplication Factor value (N) in the ICGC2 register does not matter. Stop recovery by reset is not affected. Supply voltage is not a factor.

Workaround:
1. Select a value of R that is greater than or equal to 8 when choosing a system bus frequency. This divides the high frequency DCO clocks to within system specification.
2. Increase the R value to 8 by writing the ICGC2 register just before the stop instruction. Restore the R value by writing the ICGC2 register to its original value in the service routine of the interrupt that wakes the MCU from stop3 mode.

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: freescale.com/SalesTermsandConditions.

Freescale and the Freescale logo are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. All other product or service names are the property of their respective owners.

© 2013 Freescale Semiconductor, Inc.