Mask Set Errata For 0M84G

Introduction

This mask set errata applies to the mask for these products:

- MC9S08SH8
- MC9S08SH4

MCU Device Mask Set Identification

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.

MCU Device Date Codes

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.

MCU Device Part Number Prefixes

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.

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

Errata type: *Silicon*
Affected componenet:*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-cylce (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.

Workarounds

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.

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

Errata type: *Silicon*
Affected componenet:*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.

Workarounds

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.

Errata: SE147-GNGC : Ganged Output Drive Control Register is not one-time writable

Errata type: *Silicon*
Affected componenet:*ICS*

Description

The ganged output drive control register (GNGC) is intended to be a write once register that enables the ganged output to take over up to 8 pins, enabling them as outputs and tying their data, drive strength, and slew rate control to PTC0 control bits. In normal usage, these pins would be shorted externally to be able to drive higher mA currents to an external component (i.e., H-bridge).

For this maskset, the ganged port control register can be written more than once.

In the scenario of code runaway, it might be possible to overwrite this register, relinquishing control of certain pins back to their respective control bits (i.e., PTCD_PTCD1, PTCDD_PTCDD1, PTCDS_PTCDS1, etc). In this scenario, depending on what is in those control bits for the shorted pins, it is possible to end up with shorted output trying to drive different logic levels. This may result in a large current short, that could lead to damage of the MCU.

Workarounds

Assuming user code avoids improperly configuring pins, this condition would be caused by code runaway. As such, proper precautions need to be taken to protect against code runaway and minimize any stresses on the part or application failures.

The COP/watchdog timer should be used to reset the part in the instance of code runaway. And, the LVD should be used to ensure device integrity with proper operating voltage.

Additionally, stress to the MCU can be minimized by configuring any shorted ganged ports as inputs when not controlled by the ganged output. This way, if runaway code affects the GNGC register, any outputs will be driving inputs, as opposed to outputs.

Errata: SE143-ICS : ICS Internal Reference Can Remain Enabled in Stop3 Mode

Errata type: *Silicon*
Affected componenet:*ICS*

Description

When transitioning from FEI or FBI modes to FEE or FBE modes, the internal reference clock may remain active in stop3 mode if the STOP instruction is executed soon after the IREFST bit in the ICSSC register clears. This can lead to elevated stop3 I_{DD}.

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

Workarounds

To ensure the internal reference clock is disabled before entering stop3, wait three internal reference clock periods after the IREFST bit has cleared before entering stop3. On a device with a trimmed internal reference, one period will be between 25.6 μs and 32 μs, therefore waiting 100 μs is adequate for all trimmed devices.

Or

To ensure the internal reference clock is disabled before entering stop3, transition into FEE mode and wait until the LOCK status bit indicates the FLL has attained lock before entering stop3 or transitioning into FBE mode and entering stop3.

Errata: SE133-FLASH : Unexpected Flash Block Protection Errors

Errata type: *Silicon*
Affected componenet:*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.

Workarounds

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.

Errata: SE128B-ICSV2 : TPM — TPM2CH1 Pin Reassignment Error

Errata type: *Silicon*
Affected componenet:*ICSV2*

Description

The ICS module V2 — when configured with the FLL enabled and with BDIV set to divide-by one — can sometimes produce a very short clock pulse. This short clock pulse can cause the device to malfunction, causing illegal address or illegal opcode resets. The short clock pulse is caused when the digitally controlled oscillator (DCO) is switching between certain input values as it continually reacts to the output frequency error.

- When operating from the internal reference clock, certain trim values can cause the error more often. The trim value for any particular clock frequency is unique to each device.
- The temperature coefficient of the DCO is such that the unique reference frequency causing the error, either internally or externally generated, will not be constant over temperature.

Workarounds

- If using FLL enabled with internal reference (FEI) or FLL enabled with external reference (FEE) modes, operate the device with a bus frequency equal to or below 10 MHz. This is accomplished by setting BDIV divide-by value to two or higher (BDIV[1,0] bit field value of 01, 10 or 11).
- Use the ICS in any of the modes with the FLL disabled. This includes: FLL bypassed internal (FBI), FLL bypassed internal low power (FBILP), FLL bypassed external (FBI), FLL bypassed external low power (FBELP) modes. (Not all devices have EXTAL and XTAL pins available to run the device with an external reference.)

SE120-IIC: IIC: Incorrect Clocking in 10-Bit Addressing Mode

Description

This erratum is relevant only for applications using 10-bit addressing mode and does not affect 7-bit addressing mode operations.

When hexidecimal values E0,E1,E4,E5,E8,E9,EC,ED,F0,F2,F4, or F6 are used as data in master receive slave transmissions from the IIC module using 10-bit addresses, the IIC clock may produce an incorrect number of pulses. This will result in IIC communication errors.

Workaround

Avoid using the values listed in the erratum description as data in any master receive slave transmission from the IIC module while in 10-bit addressing mode.

Errata: SE118-RTC : LPO Enabled in Stop Modes When RTC is in Reset State

Errata type: *Silicon*
Affected componenet:*RTC*

Description

The RTC module comes out of reset with the low power oscillator (LPO) as the selected clock source (RTCLKS = 0:0), but with the RTC module itself disabled (RTCPS = 0:0:0:0). This configuration signals the LPO to be enabled in all modes including stop2 and stop3.

This results in higher than expected stop currents due to the LPO stop mode current adder.

Workarounds

To disable the LPO when not needed in the stop modes, set the RTCLKS bits in the RTCSC register to any value other than 0:0. This will select an alternative clock source for the RTC, either the external clock (ERCLK) or the internal clock (IRCLK). Unlike the LPO, the RTC module does not automatically enable either of these

clocks to run in stop modes. ERCLK or IRCLK enable is determined by other MCU control bits, not the RTC's. Before entering stop mode, one period of the LPO must expire after setting the RTCLKS bits to ensure that the change has been processed.