Mask Set Errata 2
MC68HC912BC32 Microcontroller Unit

INTRODUCTION

This errata provides mask-set specific information applicable to the following MC68HC912BC32 MCU mask set devices:

- 0K25E

MCU DEVICE MASK SET IDENTIFICATION

The mask set is identified by a four-character code consisting of a letter, two numerical digits, and a letter, for example F74B. Slight variations to the mask set identification code may result in an optional numerical digit preceding the standard four-character code, for example 0F74B.

MCU DEVICE DATE CODES

Device markings indicate the week of manufacture and the mask set used. The data is coded as four numerical digits where the first two digits indicate the year and the last two digits indicate the work week. The date code “9115” would indicate the 15th week of the year 1991.

MCU DEVICE PART NUMBER PREFIXES

Some MCU samples and devices are marked with an SC, PC, ZC or XC prefix. An SC, PC or ZC prefix denotes special/custom device. An XC prefix denotes 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 prefix.

When contacting a Motorola representative for assistance, please have the MCU device mask set and date code information available.

MSE Published Date: March 1, 2001

ERRATA SUMMARY

Errata Number	Module affected	Brief description	Errata / for info only?(1)	Workaround available?	First Issued
AR311	ATD	Conversion of the (V_{rh}–V_{rl})/2 internal ref voltage returns $7F not $80	Information	Yes	Rev 0
AR564	CGM	Cannot interrupt out of STOP with DLY=1	Errata	Yes	Rev 0
AR600	IRQ	WAIT cannot be exited if XIRQ/IRQ Level deassertion occurs within particular window of time	Information	Yes	Rev 0
AR527	SWI	SWI fetched if I-bit not set when disabing interrupt	Information	Yes	Rev 0
		MCU device information and errata reference numbers (ARxxx) added			Rev 1

1. Please note that some items do not report bugs but only contain customer information.

ATD: CONVERSION OF THE (V_{RH}–V_{RL})/2 INTERNAL REF VOLTAGE RETURNS $7F, $80 OR $81 AR311

The (V_{RH}–V_{RL})/2 internal reference conversion result may be $7F, $80 or $81.

Work-around

If the (V_{RH}–V_{RL})/2 internal reference is used (perhaps for system diagnostics), expected pass result may be $7F, $80 or $81.

CGM: CANNOT INTERRUPT OUT OF STOP WITH DLY=1 AR564

STOP mode cannot be exited using interrupts when DLY=1 depending on where the Real-Time-Interrupt (RTI) counter is when the STOP instruction is executed. The RTI counter is free-running during normal operation and is only reset at the beginning of Reset, during Power-on-Reset, and after entry into STOP. The free-running counter will generate a one cycle pulse every 4096 cycles. If that pulse occurs at the exact same time that the stop signal from the CPU is asserted then the OSC is stopped but the internal stop signal will remain low. In this state the OSC is shut off until RESET.

Work-around

1. If you are not using the Real Time Interrupt function you can wait for a RTI flag before entering into STOP to guarantee the counter is in a safe state. When executing the following code all interrupt sources except for those used to exit STOP mode must be masked to prevent a loss of synchronization.
   A loss of synchronization can occur if an interrupt is processed between the setting of the RTIF and the execution of the STOP instruction. Also, you must enable the RTI counter in the initialization code, set to the fastest RTI time-out period, and the RTIE bit should NOT be set.

```
        BRCLRRTIFLG,#RTIF,RTIFClr; RTIF flag is already clear
        LDAB#RTIF; if it's set, clear the flag.
        STABRTIFLG
RTIFClr:BRCLRRTIFLG,#RTIF,*; wait until the RTIFLG is set.
        NOP
        NOP
        NOP
        STOP   ; enter stop mode
```

2. If you are driving a clock in (not using the OSC) then you could set DLY=0.
3. Pseudo-STOP and DLY=0 could be used. The oscillator will be kept alive during STOP at the expense of power consumption but no recovery delay is needed. Set the PSTP bit and clear DLY bit prior to going to STOP.
4. Limp Home and DLY=0 could be used. The part comes out of STOP in limp home mode while the crystal recovers. If a known frequency is not a requirement, this workaround avoids having the crystal alive in STOP mode. Clear the NOLHM and DLY bits prior to going to STOP.

INT: WAIT CANNOT BE EXITED IF XIRQ/IRQ LEVEL DEASSERTION OCCURS WITHIN PARTICULAR WINDOW OF TIME AR600

The device can get trapped in WAIT mode if, on exiting the WAIT instruction, the deassertion timing of the XIRQ or level-sensitive IRQ occurs within a particular timeframe. Only reset will allow recovery. Noise/bounce on the pins could also cause this problem.

Work-around

1. Use edge-triggered IRQ (IRQE=1) instead of XIRQ or level-triggered IRQ.
2. Use RTI , timer interrupts, KWU or other interrupts (except level-sensitive IRQ or XIRQ) to exit WAIT. If using RTI, it must be enabled in WAIT (RSWAI=0) and the COP must be disabled (CME=0).
3. Assert XIRQ or level-sensitive IRQ until the interrupt subroutine is entered.
4. Add de-bouncing logic to prevent inadvertent highs when exiting WAIT.

INT: DISABLING INTERRUPT WITH I MASK BIT CLEAR CAN CAUSE SWI

AR527

If the source of an interrupt is taken away by disabling the interrupt without setting the I mask bit in the CCR, an SWI interrupt may be fetched instead of the vector for the interrupt source that was disabled.

Work-around

Before disabling an interrupt using a local interrupt control bit, set the I mask bit in the CCR.

Additional mask set errata can be found on the World Wide Web at http://www.mcu.motsps.com/documentation/index.html