Mask Set Errata

MSE908AS60_3J74Y
12/2002

Mask Set Errata for 908AS60, Mask 3J74Y

MOTOROLA intelligence everywhere™ | digital dna™

Introduction

This mask set errata applies to this 908AS60 MCU mask set:

- 3J74Y

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 3J74Y. 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 “0201” indicates the first week of the year 2002.

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.

BDLC 300 µs IFS Issue

SE18-BDLC

If two messages are received at 300 µs interframe separation (IFS) (+/– µs, as measured at the RX pin), the second message's start-of-frame (SOF) symbol generates an invalid symbol interrupt. This invalid symbol interrupt results in the second message being lost and, therefore, unavailable to the application software. This is the result of a race condition within the BDLC where it is changing states in its receive state machine at the same time a transition occurs on the RX pin (beginning of the SOF symbol of the second message).

Workarounds

- Ensure that no nodes on the J1850 network will transmit a message at 300 µs IFS separation from another message. Be certain that physical layer error is taken into account when calculating this case, as temperature changes and ground shifts can shift the timing seen at the RX pin of the microcontroller. Motorola silicon implementations of J1850 have not been shown to retransmit any faster than 320 µs, and are, therefore, not likely to cause this behavior.
- Design messaging and application software to properly handle loss of messages in the system. This is safe programming practice in any case and will protect the integrity of the system in the event of a lost message.

Index Mode Instructions

SE19-INDEX

When unmapped locations are accessed with data from an unmapped address using indexed mode instructions, an illegal address reset occurs. For example, the location $FE15 is not mapped for the 68HC08AZ32. When the location is read using the following instructions, an illegal address reset occurs when `LDA ,X` is executed.

```
LDHX  $FE15
LDA ,X
```

The indexed mode instructions that cause this problem are the same instructions that originally came from the M68HC05 such as STA and ORA. However, the newer M68HC08 instructions such as MOV do not cause the illegal address reset.

To avoid this illegal address reset, do not access data from an unmapped location using an instruction with an address determined by the contents of the H:X registers. However, an opcode fetch from an unmapped address generates an illegal reset.

EEPROM Protection

SE20-EEPROM

The MC68HC908AS60 contains two EEPROM blocks, called EEPROM1 and EEPROM2. Each block contains an EEPROM Nonvolatile Register (EENVR1 and EENVR2) and an EEPROM Array Configuration Register (EEACR1 and EEACR2). Reset loads the EEACRs with the contents of the EENVR registers.

According to the General Release Specification, “This protect function is enabled by programming the EEPRTCT bit in the EENVR to 0.”

In addition to the disabling of the program and erase operations on memory locations $08F0 to $08FF ($06F0 to $06FF on EEPROM2), the enabling of the protect option has the following effects:

- Bulk and block erase modes are disabled
- Programming and erasing of the EENVR is disabled
- Unsecure locations can be erased using the single byte erase function as normal
- Secured locations can be read as normal
- Writing to a secure location no longer qualifies as a “valid EEPROM write”

This indicates that clearing the EEPRTCT bit of either EENVR register is a “one time” function which should only have effect on the corresponding EEPROM array (EENVR1 should only affect EEPROM1 and EENVR2 should only affect EEPROM2).

Here is a summary of the incorrect operation of the EEPRTCT bits of the two EENVR registers:

When the EEPRTCT bit of the EENVR1 (address $FE1C) register is programmed to 0, EEPROM protection is enabled for the EEPROM1 located at $0800 to $09FF. This means that the software cannot program the EENVR1 register once protection is enabled for EEPROM1. An illegal address reset is issued upon a write to this location because the address of EENVR1 is inadvertently removed from the memory map when EEPRTCT of the EENVR1 is cleared.

Unfortunately, programming this bit to 0 also enables protection for the EENVR2 (address $FE18) register. This means that the software cannot program the EENVR2 register once protection is enabled for EEPROM1. An illegal address reset is issued upon a write to this location because the address of EENVR2 is inadvertently removed from the memory map when EEPRTCT of the EENVR1 is cleared.

When the EEPRTCT bit of the EENV2 (address $FE18) register is programmed to 0, EEPROM protection is enabled for the EEPROM2 located at $0600 to $07FF. This means that the software cannot program the EENVR2 register once protection is enabled for EEPROM2. An illegal address reset is issued upon a write to this location because the address of EENVR2 is inadvertently removed from the memory map when EEPRTCT of the EENVR2 is cleared.

Unfortunately, programming this bit to 0 also enables protection for the EENVR1 (address $FE1C) register. This means that the software cannot program the EENVR1 register once protection is enabled for EEPROM2. An illegal address reset is issued upon a write to this location because the address of EENVR1 is inadvertently removed from the memory map when EEPRTCT of the EENVR2 is cleared.

Workaround:

As a result of the problem described previously, EEPROM protection may be enabled for only one of the two EEPROM arrays. The two cases are described below:

Case 1 — To enable EEPROM protection for array 1, program the EEPRTCT bit of EENVR1 to 0. This will result in the following:

- Locations $08F0 to $08FF are protected from program and erase
- Bulk and block erase modes are disabled for EEPROM array 1
- Unprotected locations in EEPROM1 may be erased using the single byte erase function as normal
- Protected locations may be read as normal
- Writes to a protected EEPROM1 location no longer qualifies as a “valid EEPROM write”
- New Subsequent writes to EENVR1 ($FE1C) or EENVR2 ($FE18) or $08F0 through $08FF will result in an illegal address reset

NOTE: *Be sure that the correct data is programmed in EENVR2 prior to programming the EEPRTCT bit in EENVR1.*

Case 2 — To enable EEPROM protection for array 2, program the EEPRTCT bit of EENVR2 to 0. This will result in the following:

- Locations $06F0 to $06FF are protected from program and erase
- Bulk and block erase modes are disabled for EEPROM array 2
- Unprotected locations in EEPROM2 may be erased using the single byte erase function as normal
- Protected locations may be read as normal
- Writes to a protected EEPROM2 location no longer qualifies as a “valid EEPROM write”

- New Subsequent writes to EENVR2 ($FE18) or EENVR1 ($FE1C) or $06F0 through $06FF will result in an illegal address reset

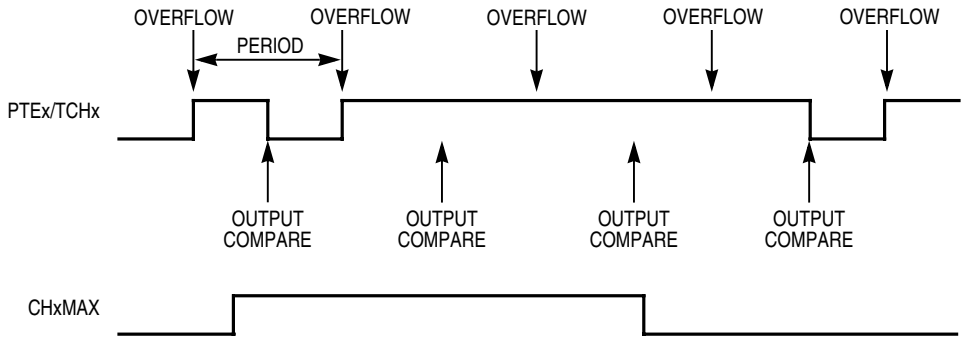
NOTE: *Be sure that the correct data is programmed in EENVR1 prior to programming the EEPRTCT bit in EENVR2.*

Timer Interface Module (TIM) SE38-TIM

The PWM 100% duty cycle is defined as output high all of the time. To generate the 100% duty cycle, use the CHxMAX bit in the TSCx register.

CHxMAX — Channel x Maximum Duty Cycle Bit

When the TOVx is at logic 1 and clear output on compare is selected, setting the CHxMAX bit forces the duty cycle of buffered and unbuffered PWM signals to 100%. As this figure shows, the CHxMAX bit takes effect in the cycle after it is set or cleared. The output stays at 100% duty cycle level until the cycle after CHxMAX is cleared.



NOTE: *The PWM 0% duty cycle is defined as output low all of the time. To generate the 0% duty cycle, select clear output on compare and then clear the TOVx bit (CHxMAX=0).*

HOW TO REACH US:

USA/EUROPE/LOCATIONS NOT LISTED:

Motorola Literature Distribution;
P.O. Box 5405, Denver, Colorado 80217
1-303-675-2140 or 1-800-441-2447

JAPAN:

Motorola Japan Ltd.; SPS, Technical Information Center,
3-20-1, Minami-Azabu Minato-ku, Tokyo 106-8573 Japan
81-3-3440-3569

ASIA/PACIFIC:

Motorola Semiconductors H.K. Ltd.;
Silicon Harbour Centre, 2 Dai King Street,
Tai Po Industrial Estate, Tai Po, N.T., Hong Kong
852-26668334

TECHNICAL INFORMATION CENTER:

1-800-521-6274

HOME PAGE:

http://www.motorola.com/semiconductors

MOTOROLA