Mask Set Errata

MSE908AS60_8H62A
12/2002

Mask Set Errata for 908AS60, Mask 8H62A

Introduction

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

- 8H62A

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 8H62A. 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.*

FLASH Module

SE43B-FLASH

Failure of Stress Test: FLASH Write/Erase Endurance at 125°C

These stress tests have been completed satisfactorily to the requirements for XC qualification:

- High temperature operating life test (168 hours)
- ESD human body model (1,000 volts)
- ESD machine model (200 volts)
- Latch up (200 mA)
- EEPROM write/erase endurance (10,000 cycles)
- EEPROM data retention (168 hours)
- FLASH data retention (168 hours)

However, failure has been detected in this stress test:

- FLASH write/erase endurance at 125°C (100 cycles)

This stress test consists of successive program/erase cycles of the entire FLASH array and is performed up to the specified target/erase lifetime of the FLASH array (100 program/erase cycles) at maximum specified temperature (125°C). Any unit that cannot be programmed properly within the 100 cycles is regarded as a failure.

It is believed that these failures could be screened in Motorola’s production testing by suitable enhancements to the burn-in process and/or FLASH specification. However, until these have been implemented and demonstrated to be effective, units supplied by Motorola could exhibit a level of fallout during repeated FLASH erase/programming operations.

Table 1 shows the predicted FLASH programming failure rate in ppms per programming cycle at different user temperatures using the Arrhenius equation (with an activation energy of 0.35 eV for FLASH failure mechanisms) to derate from the observed data.

Table 1. FLASH Programming Failure Rate Per Programming Cycle[1]

Confidence Level	Programming Temperatures			
	25°C	85°C	105°C	125°C
60%	59 ppm	579 ppm	1056 ppm	1806 ppm
90%	73 ppm	707 ppm	1290ppm	2207 ppm

[1] Based on this data, the following are recommended:

a. Keep the number of programming cycles to a minimum.

b. Programming should be performed at ambient temperature only to reduce the overall number of programming fails.

Preliminary experimental data gathered shows the acceleration factor due to temperature to be significantly greater than that calculated using the Arrhenius equation. Consequently, these numbers should be regarded as upper estimates.

All other aspects of FLASH operation (for instance, data retention) have passed testing successfully.

Recommendation

Provision for module replacement in line with the programming fallout estimates discussed here are recommended to be made at locations where FLASH re-programming is performed.

FLASH Memory Programming

Use of the smart programming algorithm, the iterative page program/margin read technique, is required.

When using the iterative page program/margin read technique, a maximum of 100 programming pulses is allowed. The minimum duration of the programming pulse (t_{STEP}) shall be 1 ms. The t_{STEP} duration is defined as the amount of time during one program cycle that HVEN is asserted (HVEN = 1). The maximum t_{STEP} is 1.2 ms. Therefore, the maximum cumulative program time (t_{PROG}) per page is 120 ms.

At 25°C and above, with the minimum page program step (t_{STEP}) of 1 ms, the maximum page program time is 100 ms.

The requisite smart programming algorithm will determine the necessary program time requirements of each page.

Cumulative program time exceeding 100 steps per page may cause unintentional programming of erased bits. This condition is known as program disturb.

The FLASH row program endurance is 100 cycles.

The FLASH row erase endurance is 100 cycles.

The minimum FLASH READ bus clock period (t_{CYC}) is 119 ns (8.4 MHz). The maximum bus clock period is 31,250 ns (32 kHz).

The minimum FLASH program/erase/margin read frequency is 1.8 MHz. The maximum is 2.5 MHz.

Note: This algorithm is mandatory for programming the FLASH 2TS.

Note: This page program algorithm assumes the page/s to be programmed are initially erased.

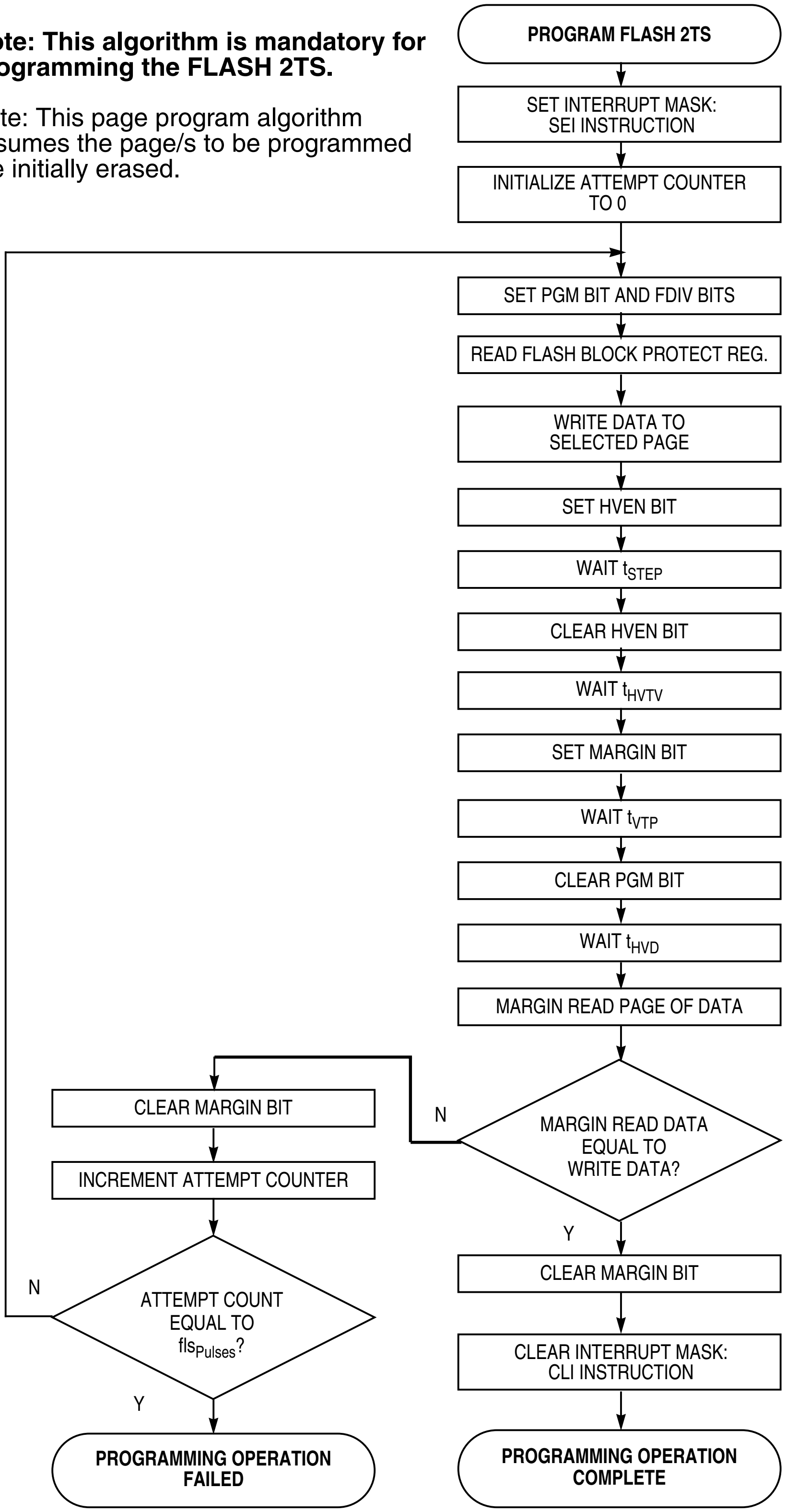
PROGRAM FLASH 2TS
SET INTERRUPT MASK: SEI INSTRUCTION
INITIALIZE ATTEMPT COUNTER TO 0
SET PGM BIT AND FDIV BITS
READ FLASH BLOCK PROTECT REG.
WRITE DATA TO SELECTED PAGE
SET HVEN BIT
WAIT t_{STEP}
CLEAR HVEN BIT
WAIT t_{HVTV}
SET MARGIN BIT
WAIT t_{VTP}
CLEAR PGM BIT
WAIT t_{HVD}
MARGIN READ PAGE OF DATA
MARGIN READ DATA EQUAL TO WRITE DATA?
N
Y
CLEAR MARGIN BIT
INCREMENT ATTEMPT COUNTER
ATTEMPT COUNT EQUAL TO fls_{Pulses}?
N
Y
CLEAR MARGIN BIT
CLEAR INTERRUPT MASK: CLI INSTRUCTION
PROGRAMMING OPERATION FAILED
PROGRAMMING OPERATION COMPLETE

Figure 1. Smart Programming Algorithm, Page Program/Margin Read Procedure

FLASH Security

SE44-FLASH

On the 8H62A mask set, the FLASH security feature is enabled. This security mechanism is designed to prevent unauthorized access to the FLASH contents in any non-user mode. The security feature relies on the assumption that only authorized users know the contents of the FLASH.

When using monitor mode to program the FLASH, eight bytes are downloaded serially after every reset and these eight bytes must match the contents of FLASH locations $FFF6 to $FFFD. If the bytes don’t match, then no read, program, or erase operations will be possible.

Motorola-supplied programming tools, such as the serial programmer, have the capability for downloading the eight security bytes. Contact a local Motorola tools support office for more information.

The presence of a voltage V_{HI} (defined as V_{DD} to V_{DD}+2) on the IRQ pin will bypass block protection. This means that no matter what block protect bits are set in the FLBPR (FLASH block protect register), all memory is available for programming or erasing. The block protect V_{HI} override is level sensitive only, not latched.

Electro-Static Discharge (ESD)

SE45-ESD

The 68HC908AS60 fails the human body model test above 1.0 kilovolts.

Floating Node in ADC Module Causes Variable Stop I_{DD} Current in Stop Mode

SE46-ADC

This mast set exhibits variable I_{DD} current in stop mode. A floating node within the ADC module causes the stop I_{DD} to be variable. When the floating node is corrected, the stop I_{DD} will be reduced.

Possible BDLC Missing Information

SE47-BDLC

The BDLC (byte data link controller) condition discussed here has been observed in a lab situation. The conditions are difficult to create. However, there is a remote possibility of creating a failed transmission of a message, depending on the method used to determine that a pending message has been sent. Some of the recommended practices described here will, most likely, be implemented in existing user software.

BDLC Message Transmission Problem

To transmit a message using the BDLC, the user writes the first byte that had been previously written to the BDR (byte data register). This will inhibit the transmission process at the beginning of the next idle bus state. An exception to this procedure has been identified and is described here.

An invalid symbol being received by the BDLC clears any byte that previously had been written to the BDR. This will inhibit the transmission process until the user writes another byte to the BDR.

The following scenario describes an event sequence that will prevent the user from knowing that an invalid symbol was received and that the BDR had been cleared.

1. BDLC starts to receive a message.
2. The user decides to ignore this message by setting the IMSG bit in BCR1. Setting this bit blocks all further BDLC interrupts until the next start of frame appears on the bus.
3. User writes a byte to the BDR to have the transmission process begin the next time an idle state is seen on the bus.
4. An invalid symbol is received by the BDLC and the BDR is cleared. The user is unaware of this event because of the setting of the IMSG bit.
5. End of frame occurs.
6. Idle state appears on the bus.
7. Transmission does not begin.

Work Around

A 2-level strategy has been developed that positively signals the need to restart the transmission of a message.

- The first level looks for the special case of reception of an illegal symbol with a byte pending transmission in the BDLC data register, as described earlier.
- The second level uses a transmit watchdog timer to spot any case of a transmission not occurring within a maximum amount of time.

1.
   a. If an illegal symbol interrupt occurs with a byte pending transmission in the BDR, reload the BDR with the first byte of that message to restart transmission.
   b. Do not load the first byte of a message to transmit into the BDR when the IMSG bit is set in BCR1. Wait until IMSG is cleared by reception of the next message on the bus. This will prevent the case of an illegal symbol interrupt clearing the BDR but no interrupt indication being given to the host due to IMSG blocking the interrupts.

c. If a byte is pending transmission in the BDR, do not set the IMSG bit in BCR1. This will require the continued handling of each byte received in the current message, but will allow the host to see an illegal symbol interrupt and perform the operation described in 1a.

2. When the first byte of a message to transmit is loaded into the BDR, start a 16.5 ms (minimum) transmit watchdog timer. This is the worst case time before the first TDRE interrupt of a transmission should occur. It is the sum of symbol times of a maximum length message (consisting of an SOF, 12 bytes of long symbols, and an IFS), and the SOF of the message to be transmitted. The worst case symbol receive times, as shown in Table A-2 of the BDLC Reference Manual, were used to calculate the watchdog timeout. When the first TDRE is received, this watchdog is deactivated. If the watchdog times out, the first byte of the message to transmit is reloaded into the BDR and the watchdog restarted. This timer will prevent all cases of the driver software transmit state machine getting out of synchronization with the BDLC and hanging up, either due to this issue or any other unexpected condition that causes the software to think that a transmission is pending when it is not.

Exceptions to Documentation

SE48-DOC

All information in the *68HC908AS60 Advance Information*, Motorola document order number MC68HC908AS60/D, applies to the 68HC708AS60 with these exceptions:

- Programming temperature is 25°C.
- FLASH clock endurance does not apply.
- FLASH erase time does not apply.

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

Information in this document is provided solely to enable system and software implementers to use Motorola products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Motorola reserves the right to make changes without further notice to any products herein. Motorola makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Motorola 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 which may be provided in Motorola 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. Motorola does not convey any license under its patent rights nor the rights of others. Motorola products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Motorola product could create a situation where personal injury or death may occur. Should Buyer purchase or use Motorola products for any such unintended or unauthorized application, Buyer shall indemnify and hold Motorola and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Motorola was negligent regarding the design or manufacture of the part.

MOTOROLA