MSE908LJ12_2L79A
Rev. 1, 4/2007

Mask Set Errata for Mask 2L79A

Covers MC68HC908LJ12 MCUs

Introduction

This mask set errata applies to the mask 2L79A for this products:

- MC68HC908LJ12

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 2L79A. All standard devices are marked with a mask set number and a date code.

Possible LVI Reset

SE126-LVI

Description

An LVI reset may be occur immediately after the LVI is enabled by writing to the CONFIG1 register. On most HC08 devices, the low voltage inhibit feature is enabled out of reset. On this part, the LVI feature is disabled after reset, and is enabled by user software.

The internal voltage reference may not be stable after the LVI is initially enabled, causing a brief LVITRIP pulse. This pulse may not be long enough to set the LVIOUT bit, but it is latched so that when the LVI is enabled (CONFIG1 bit LVIPWRD = 0) an immediate LVI reset occurs. In the subsequent reset cycle, the

voltage reference will be stable so the LVITRIP pulse will not be generated. This fault does not occur on all LJ12 devices.

Workaround

For systems that use the LVI reset feature, write to the CONFIG2 (to set trip voltage) and CONFIG1 (to enable LVI resets) registers immediately after power up. This will not prevent the inadvertent LVI reset, but it will minimize the impact to the system by allowing the reset to occur before other system initialization is done. Note that this is the recommended procedure for safe system operation.

For systems that use the LVI interrupt or polling feature (LVI reset disabled), write to the CONFIG2 (to set trip voltage) and CONFIG1 (to enable LVI detector) registers immediately after power up. Set the LVIIAK bit in the LVISR register to clear the LVIIF flag before enabling the interrupt or before polling the LVIIF flag. Flag clearing is a recommended procedure before enabling interrupts or using polling.

This issue will not affect applications that do not use the LVI feature.

Stop Recovery May Cause Illegal Address Reset or Illegal Opcode Reset

SE117-STOP_RECOVERY

Description

During stop mode wake up the MCU may experience an illegal opcode reset or illegal address reset instead of the normal stop mode recovery sequence. This problem is observed during periodic recovery from stop mode. It is due to the asynchronous clock switching from the internal ICLK to the external clock in the stop recovery process. Recovery from wait mode does not have this problem as there is no internal clock switching.

Workaround

Two recommendations are described below to reduce the impact caused by this problem on the end applications.

- If the application can accept the larger wait I_{DD}, use wait mode instead of stop mode, because clock switching is not required when the application system recovers from wait mode.

- If the application cannot accept the wait I_{DD}, use the system reset status register (SRSR) to identify the reset source. Eliminating the improper reset source using the SRSR can reduce the impact on the system when the MCU restarts after the inadvertent reset. This method is described below:

  Before entering the stop mode, application code must save into RAM those registers and variables that are critical to the application system. Most of the I/O registers will be reset to their default value when the inadvertent reset happens. However, RAM data will not be affected when reset occurs. Saving critical registers to RAM enables the system to recover to the mode it was in before entering stop.

Besides saving the critical registers into RAM before entering stop mode, other processing is required after the inadvertent reset occurs. User firmware must immediately check the source of the reset by reading SRSR. If the reset source is illegal opcode reset or illegal address reset, user firmware must bypass any clear RAM routine and variables initialization routines. Instead, restore those values saved before entering stop mode into the corresponding registers. If the reset source is not illegal opcode or illegal address reset, user code can execute the normal initialization routine.