MSE908LJ24_0L83S
Rev. 0, 9/2006

Mask Set Errata for Mask 0L83S

Covers MC68HC908LJ24 MCUs

Introduction

This mask set errata applies to the mask 0L83S for this products:

- MC68HC908LJ24

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 0L83S. All standard devices are marked with a mask set number and a date code.

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.

freescale™ semiconductor

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.