Mask Set Errata for Mask 1N39G

Introduction

This report applies to mask 1N39G for these products:

- MSE9S12ZVHY64

ID before 15 MAY 2008	ID from 15 May 2008 to 30 JUNE 2010	ID after 1 JULY 2010	Errata Title
		6964	ADC: Flag RSTAR_EIF is set unexpectedly in corner cases
		6965	ADC: Flag TRIG_EIF not set if erroneous TRIG request occurs short before STOP Mode entry or WAIT Mode entry with bit SWAI set
		7019	FTMRZ: Problem with Flash commands 'Set User Margin Level' and 'Set Field Margin Level'
		6722	S12Z_BDC: WRITE_MEM, FILL_MEM commands issued during STOP with the core clock disabled can write incorrect data to the addressed location.

e6964: ADC: Flag RSTAR_EIF is set unexpectedly in corner cases

Errata type: Errata

Description: In case of repeated Restart Request overruns (Restart overrun occurs again before previous Restart overrun could be handled) the RSTAR_EIF is erroneously set if:

- The first Restart Request overrun occurs correctly with a simultaneous Sequence Abort Request AND

- The second Restart Request overrun occurs without Sequence Abort Request, which is a flow control failure but for overrun situation the issue should not be flagged.

Misbehavior:

The ADC flags the flow control failure (RSTAR_EIF bit is set) in the second overrun which should not happen. Overruns are tracked quietly and executed as soon as Sequence Abort turnaround time allows.

Workaround: If possible, avoid multiple Restart Request overrun scenarios.

e6965: ADC: Flag TRIG_EIF not set if erroneous TRIG request occurs short before STOP Mode entry or WAIT Mode entry with bit SWAI set

Errata type: Errata

Description: At Low power Mode entry (STOP or WAIT with bit SWAI set) all current flow control information as well as all pending flow control information gets deleted immediately.

In case of a flow control error, the detection can take several cycles until it is flagged by TRIG_EIF.

This leads to the corner case in which TRIG_EIF does not get set because the device enters STOP or WAIT with SWAI set .

Workaround: The ADC should be idle (no conversion or conversion sequence or Command Sequence List ongoing) when entering STOP mode or WAIT with bit SWAI set.

e7019: FTMRZ: Problem with Flash commands 'Set User Margin Level' and 'Set Field Margin Level'

Errata type: Errata

Description: When commands 'Set User Margin Level' and/or 'Set Field Margin Level' are executed targeting P-Flash, configured to return to the normal read level (FCCOB2=0, thus effectively disabling any margin level settings previously applied to Flash reads), the margin level is deactivated correctly but execution of further commands on EEPROM is compromised: concurrent reads to P-Flash while an EEPROM command is executed might result in corrupted data being read, and the EEPROM commands might not execute properly with status flags indicating that errors might have occured.

If any of the commands 'Set User Margin Level' or 'Set Field Margin Level' run on EEPROM (enabling and/or disabling the margin levels) there is no issue.

Workaround: The way to properly restore the functionality is to run 'Set User Margin Level' or 'Set Field Margin Level' to return to the normal read level (FCCOB2=0) loading an EEPROM address in FCCOB0 / FCCOB1, even if at the beginning of the procedure a P-Flash address was used to set the margin level.

e6722: S12Z_BDC: WRITE_MEM, FILL_MEM commands issued during STOP with the core clock disabled can write incorrect data to the addressed location.

Errata type: Errata

Description: With the device core clock disabled during STOP mode debugging (BDCCIS=0) then WRITE_MEM and FILL_MEM commands can cause incorrect data to be written to the addressed location. This can happen if the device leaves STOP mode during execution of the WRITE_MEM, FILL_MEM command.

Workaround: Three workarounds are proposed. The appropriate workaround depends on the individual debugging requirements.

1. Do not disable the device core clock when debugging through STOP (set BDCCIS).

2. If BDCCIS=0 then only use WRITE_MEM, FILL_MEM whilst in active BDM (rather than during code execution).

3. If BDCCIS=0 then use WRITE_MEM.WS/FILL_MEM.WS which indicate that STOP mode occurred. If STOP=1 then read data back to verify correct write execution.

OR Use ACK and then check BDCCSR if a long ACK occurs. If STOP=1 then read data back to verify.

Note that the data could have been overwritten by execution code because the device has since left STOP mode. This is only of use if the user can be sure that the application code does not write to the same location.

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Information in this document is provided solely to enable system and software implementers to use Freescale products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. Freescale reserves the right to make changes without further notice to any products herein.

Freescale makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale 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 that may be provided in Freescale 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. Freescale does not convey any license under its patent rights nor the rights of others. Freescale sells products pursuant to standard terms and conditions of sale, which can be found at the following address: freescale.com/SalesTermsandConditions.