

Mask Set Errata for Mask N29D

Introduction

This report applies to mask N29D for these products:

- PXD20

Errata ID	Errata Title
3526	Flash : Prefetch at flash boundary causes unexpected ECC errors
3664	Flash may not enter normal mode after reset or on wake from STANDBY
3570	MC_ME: Possibility of Machine Check on Low-Power Mode Exit
3574	MC_RGM: A non-monotonic ramp on the VDD_HV/BV supply can cause the RGM module to clear all flags in the DES register
3624	RLE: Decode operation freezes when configured for very small data transfers

e3526: Flash : Prefetch at flash boundary causes unexpected ECC errors

Errata type: Errata

Description: For flash modules, where the implemented array size is less than the maximum allowed size of the flash revision (2MB for FL2 or 3MB for FL3) then if prefetch from the flash is enabled via the configuration bits in the Platform Flash Configuration Register (PFCR), accesses to the last page in the flash may cause an unexpected ECC error flagged by the ECC Event Error bit (EER) in the Module Configuration Register (MCR) of the flash module.

As 1 page is 4 words + ECC bits (146bits in total), the addresses affected are the upper most 16 bytes in the flash address space.

Workaround: Either :

(1) Don't use prefetch

or

(2) Do not use last page in flash array when prefetch is enabled;

or



(3) Ignore the setting of the Flash module MCR.EER register bit on the last page if you are using prefetch.

e3664: Flash may not enter normal mode after reset or on wake from STANDBY

Errata type: Errata

Description: One of the actions performed during a microcontroller reset is to reset the internal flash block. This takes place on all microcontroller resets except those where the short sequence reset option is selected. The short reset is controlled by the RGM_FESS register and applies to a limited number of functional resets.

The internal flash is also reset when the microcontroller wakes from STANDBY mode.

The expected behavior is that the flash block will return to normal operation after receiving an internal reset, however, this intermittently fails to occur.

The behavior of the microcontroller during this condition depends on the circumstances of the internal reset.

If the condition occurs on a microcontroller reset then the Reset Generation Module will stall the reset process while waiting for the internal flash to boot. As part of its normal operation the RGM will enable the Software Watchdog Timer and so a second watchdog reset will occur approximately 16 ms after the failed reset. It is likely that this second reset will behave normally.

There are two possible scenarios when waking from STANDBY mode.

If the device is configured to restart from flash on waking from STANDBY then the device will intermittently stall while waiting for the flash and will be unable to execute code. This stall can only be recovered by a microcontroller reset.

If the device is configured to restart from RAM on waking from STANDBY then the behavior depends on how the flash is configured in the ME_DRUN_MC register. If the flash is configured to be in normal mode then the device will intermittently stall while waiting for the flash and will be unable to execute code. This stall can only be recovered by a reset. If the flash is configured to be in power-down or low-power mode the device will execute code from RAM as expected however any attempt to change mode to place the flash into normal mode will intermittently fail and that mode entry transition will never complete.

Workaround: The original source of a reset can be determined by examining the bits in the RGM_DES and RGM_FES register. If no other flags are set then the SWT is the original reset source. If other flags are set then it is possible that they are the true source of reset. To simplify this detection ensure that the source flags in the RGM_DES and RGM_FES registers are cleared once the reset source is confirmed. This will ensure that the flag registers only contain flags related to unhandled resets.

If suitable for the application then configure functional resets to use the short sequence.

There are four workarounds possible for use in STANDBY mode:

1/ Do not use STANDBY mode. Instead use STOP mode and configure it to the lowest current configuration.

2/ Configure the device to recover to flash and configure the SWT watchdog to be active from reset by setting the flash NVUSR0[WATCHDOG_EN] bit. In software track entry into and out of STANDBY mode using a value in the RAM. If the SWT watchdog flag is present in the RGM_FES register then the value in RAM will indicate that the reset was caused by recovery from STANDBY as distinct from a normal SWT watchdog failure while executing code.

3/ Configure the device to recover into RAM and then perform a mode transition to place the flash into normal mode in a RUNn mode (for example RUN0). If the mode transition does not complete after a suitable period (for example 100 us) then abort the transition by performing a mode transition back to DRUN mode and in software set a value in RAM before executing a mode transition into RESET mode. This will cause the device to reset and the flash to recover. The software can distinguish between this reset and another by examining the flags in the RGM_FES register and the value in RAM.

4/ Configure the device to recover into RAM and then perform a mode transition to place the flash into normal mode in a RUNn mode (for example RUN0). If the mode transition does not complete after a suitable period (for example 100 us) then abort the transition by performing a mode transition back to DRUN mode. The flash can be recovered by cycling through STANDBY mode again. Configure a wakeup using the API set to a suitable period (for example 500 us) and then perform another mode transition into STANDBY mode. When the device recovers perform the flash mode transition once more. If this fails on multiple attempts perform a reset per step 3/. It is unlikely that this will fail more than once in a row.

e3570: MC_ME: Possibility of Machine Check on Low-Power Mode Exit

Errata type: Errata

Description: When executing from the flash and entering a Low-Power Mode (LPM) where the flash is in low-power or power-down mode, 2-4 clock cycles exist at the beginning of the RUNx to LPM transition during which a wakeup or interrupt will generate a checkstop due to the flash not being available on RUNx mode re-entry. This will cause either a checkstop reset or machine check interrupt.

Workaround: If the application must avoid the reset, two workarounds are suggested:

- 1) Configure the application to handle the machine check interrupt in RAM dealing with the problem only if it occurs
- 2) Configure the MCU to avoid the machine check interrupt, executing the transition into low power modes in RAM

There is no absolute requirement to work around the possibility of a checkstop reset if the application can accept the reset, and associated delays, and continue. In this event, the WKPU.WISR will not indicate the channel that triggered the wakeup though the F_CHKSTOP flag will indicate that the reset has occurred. The F_CHKSTOP flag could still be caused by other error conditions so the startup strategy from this condition should be considered alongside any pre-existing strategy for recovering from an F_CHKSTOP condition.

e3574: MC_RGM: A non-monotonic ramp on the VDD_HV/BV supply can cause the RGM module to clear all flags in the DES register

Errata type: Errata

Description: During power up, if there is non-monotonicity in power supply ramp with a voltage drop > 100mV due to external factors, such as battery cranking or weak board regulators, the SoC may show a no flag condition (F_POR==LVD12==LVD27==0).

Under these situations, it is recommended that customers use a workaround to detect a POR.

In all cases, initialization of the device will complete normally.

Workaround: The software workaround need only be applied when neither the F_POR, LVD27 nor LVD12 flag is set and involves checking SRAM contents and monitoring for ECC errors during this process. In all cases, an ECC error is assumed to signify a power-on reset (POR).

Three suggestions are made for software workarounds. In each case, if POR is detected all RAM should be initialized otherwise no power-on condition is detected and it is possible to initialize only needed parts of RAM while preserving required information.

Software workaround #1 :

An area of RAM can be reserved by the compiler into which a KEY, such as 0x3EC1_9678, is written. This area can be checked at boot and if the KEY is incorrect or an ECC error occurs, POR can be assumed and the KEY should be set. Use of a KEY increases detection rate to 31 bits ($\approx 10^{-9}$) or 23bits ($\approx 5 \cdot 10^{-6}$) instead of 7-bit linked to ECC ($\approx 10^{-2}$)

Software workaround #2 :

When runtime data should be retained and RAM only fully re-initialized in the case of POR, a checksum should be calculated on the runtime data area after each data write. In the event of a reset where no flags are set, the checksum should be read and compared with one calculated across the data area. If reading the checksum and the runtime data area succeeds without an ECC error, and the checksums match, it is assumed that no POR occurred. The checksum could be a CRC, a CMAC or any other suitable hash.

Software workaround #3 :

Perform a read of memory space that is expected to be retained across an LVD reset. If there are no ECC errors, it can be assumed that an LVD reset occurred rather than a POR.

e3624: RLE: Decode operation freezes when configured for very small data transfers

Errata type: Errata

Description: When the RLE decode module is configured to decode small blocks of encoded data it may enter a state where an operation either never completes or completes the current operation but causes the next decode operation to fail. The fail condition depends on the configuration of the RLE module and the eDMA channel used to copy data into the module.

Workaround: Use at least 32 byte images (compressed and non-compressed) when using DMA modes on RLE.

How to Reach Us:

Home Page:

www.freescale.com

Web Support:

<http://www.freescale.com/support>

USA/Europe or Locations Not Listed:

Freescale Semiconductor
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:

Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:

Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:

Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductors 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.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale Semiconductor 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 Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor 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 failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify Freescale Semiconductor 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 claims alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

RoHS-compliant and/or Pb-free versions of Freescale products have the functionality and electrical characteristics as their non-RoHS-complaint and/or non-Pb-free counterparts. For further information, see <http://www.freescale.com> or contact your Freescale sales representative.

For information on Freescale's Environmental Products program, go to <http://www.freescale.com/epp>.

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.

© 2012 Freescale Semiconductor, Inc.

