SPC560D30x, SPC560D40x
Errata sheet

32-bit MCU family built on the Power Architecture®category for automotive electronics

Introduction

This errata sheet describes all the functional and electrical problems known in the revision of SPC560D30x, SPC560D40x.

Device identification:

- MCU ID Register 1
  - MAJOR_MASK: 0
  - MINOR_MASK: 2
- Die Mask ID: FD40X1

This errata sheet applies to SPC560D30x, SPC560D40x,devices in accordance with *Table 1*.

Table 1. Device summary

Package	Part number	
	128 Kbyte code Flash	256 Kbyte code Flash
LQFP100	SPC560D30L3	SPC560D40L3
LQFP64	SPC560D30L1	SPC560D40L1

Contents

List of tables

1 Functional problems

1.1 ERR002656: FlexCAN: Abort request blocks the CODE field

Description:

An Abort request to a transmit Message Buffer (TxMB) can block any write operation into its CODE field. Therefore, the TxMB cannot be aborted or deactivated until it completes avalid transmission (by winning the CAN bus arbitration and transmitting the contents of theTxMB).

Workaround:

Instead of aborting the transmission, use deactivation instead.

Note: Note that there is a chance the deactivated TxMB can be transmitted without settingIFLAG and updating the CODE field if it is deactivated.

1.2 ERR002958: MC_RGM: Clearing a flag at RGM_DES or RGM_FES register may be prevented by a reset

Description:

Clearing a flag at RGM_DES and RGM_FES registers requires two clock cycles because of a synchronization mechanism. As a consequence if a reset occurs while clearing is on-going the reset may interrupt the clearing mechanism leaving the flag set.

Note: Note that this failed clearing has no impact on further flag clearing requests.

Workaround:

No workaround for all reset sources except SW reset.

Note: Note that in case the application requests a SW reset immediately after clearing a flag in RGM_xES the same issue may occur. To avoid this effect the application must ensure that flag clearing has completed by reading the RGM_xES register before the SW reset is requested.

1.3 ERR003060: MC_RGM: SAFE mode exit may be possible even though condition causing the SAFE mode request has not been cleared

Description:

A SAFE mode exit should not be possible as long as any condition that caused a SAFE mode entry is still active. However, if the corresponding status flag in the RGM_FES register has been cleared, the SAFE mode exit may incorrectly occur even though the actual condition is still active.

Workaround:

Software must clear the SAFE mode request condition at the source before clearing the corresponding RGM_FES flag. This will ensure that the condition is no longer active when the RGM_FES flag is cleared and thus the SAFE mode exit can occur under the correct conditions.

1.4 ERR003069: ADC: ADC_DMAE[DCLR] set to 1 clears the DMA request incorrectly

Description:

When ADC_DMAE[DCLR] is set the DMA request should be cleared only after the data registers are read. However for this case the DMA request is automatically cleared and will not be recognized by the eDMA.

Workaround:

None

1.5 ERR003103: Accesses to reserved PSMI registers do not return a transfer error

Description:

When ADC_DMAE[DCLR] is set the DMA request should be cleared only after the data registers are read. However for this case the DMA request is automatically cleared and will not be recognized by the eDMA.

Workaround:

None

1.6 ERR003190: MC_ME: Main VREG not disabled during STOP0 or HALT0 mode if RUN[0..3] mode selects FXOSC to be running and target mode selects FXOSC as system clock

Description:

If STOP0 or HALT0 is configured with ME_[mode]_MC.MVRON = '0', ME_[mode]_MC.FIRCON = '0' and ME_[mode]_MC.SYSCLK = '0010/0011' the Main VREG will nevertheless remain enabled during the STOP0 mode if the previous RUN[0..3] mode is configured with ME_RUN[0..3]_MC.FXOSCON = '1'. This will result in increased current consumption of 500uA than expected.

Workaround:

Before entering STOP0 or HALT0 mode with the following configuration - ME_[mode]_MC.MVRON = '0', ME_[mode]_MC.FIRCON = '0' and ME_[mode]_MC.SYSCLK = '0010/0011' - ensure the RUN[0..3] mode switches off FXOSC - ME_RUN[0..3]_MC.FXOSCON = '0' before attempting to low power mode transition

1.7 ERR003195: LINFlex: Limitations for DMA access to LINFlex

Description:

The DMA handshaking to the LINFlex can fail when the LINFlex operates on a divided peripheral clock.

Workaround:

Don't divide the LINFlex peripheral clock if DMA access is required.

1.8 ERR003209: NMI pin configuration limitation in standby mode

Description:

NMI pin cannot be configured to generate Non Maskable Interrupt event to the core (WKPU_NCR[NDSS] = "00") if the following standby mode is to be used:

- NMI pin enabled for wake-up event,
- standby exit sequence boot from RAM,
- code flash module power-down on standby exit sequence.

With following configuration following scenario may happen:

1. System is in standby
2. NMI event is triggered on PA[1]
3. System wakeup z0 core power domain.
4. z0 core reset is released and NMI event is sampled by core on first clock-edge.
5. z0 core attempt to fetch code at 0x10 address (IVPR is not yet initialized by application) and receive an exception since flash is not available
6. z0 core enter machine check and execution is stalled.

Workaround:

If NMI is configured as wake-up source, WKPU_NCR[NDSS] must be configures as "11". This will ensure no NMI event is trigger on the core but ensure system wakeup is triggered. After standby exit, core will boot and configure its IVOR/IVPR, it may then re-configure WKPU_NCR:DSS to the appropriate configuration for enabling NMI/CI/MCP.

1.9 ERR003219: MC_CGM: System clock may stop for case when target clock source stops during clock switching transition

Description:

The clock switching is a two step process. The availability of the target clock is first verified. Then the system clock is switched to the new target clock source within two target clock cycles. For the case when the FXOSC stops during the required two cycles, the switching process may not complete, causing the system clock to stop and prevent further clock switching. This may happen if one of the following cases occurs while the system clock source is switching to FXOSC:

- FXOSC oscillator failure
- SAFE mode request occurs, as this mode will immediately switch OFF the FXOSC (refer to ME_SAFE_MC register configuration)

Workaround:

The device is able to recover through any reset event ('functional', 'destructive', internal or external), so typically either the SWT (internal watchdog) will generate a reset or, in case it

is used in the application, the external watchdog will generate an external reset. In all cases the devices will restart properly after reset. To reduce the probability that this issue occurs in the application, disable SAFE mode transitions when the device is executing a mode transition with the FXOSC as the system clock source in the target mode.

1.10 ERR003242: PB[10],PD[0:1] pins configuration during standby

Description:

PB[10], PD[0:1] are the pins having both wake-up functionality and analog functionality. As for all wake-up pins, it must be driven either high level or low level (possibly using the internal pull-up) during standby. In case the pin is connected to external component providing analog signal, it is important to check that this external analog signal is either lower than 0.2*VDD_HV or higher than 0.8*VDD_HV not to incur extra consumption.

Workaround:

None

1.11 ERR003247: MC_ME: STANDBY0/HALT0/STOP0 modes cannot be entered if the FlexCAN peripheral is active

Description:

If any FlexCAN module is enabled in current mode by the ME_RUN_PCx/ME_PCTLx registers of the MC_ME and also enabled at the FlexCAN module, (MCR.B.MDIS=0), STANDBY0/HALT0/STOP0 modes will not be entered if the target low power mode is disabling the module, the device mode transition hangs.

Workaround:

The FlexCAN module must be frozen (FLEXCANx_MCR[FRZ]=1) in DRUN or RUNx mode before entering STANDBY0/HALT0/STOP0 modes.

1.12 ERR003252: CTU: ADC1_X[0:3] channels cannot be triggered by CTU

Description:

CTU_EVTCFGRx[CHANNEL_VALUE] bitfield is 6 bit wide and can select ADC channels within the range 0-63. ADC1_X[0:3] indexed CH[64-95], cannot be selected.

Workaround:

None

1.13 ERR003288: Wakeup line functionality on PB[8], PB[9] not available in STANDBY

Description:

Wakeup line functionality is not available on ports PB[8], PB[9] during STANDBY mode. These pads are not supplied with ultra low power regulator, but are driven from main regulator which is switched off in STANDBY mode.

Workaround:

None

1.14 ERR003407: FlexCAN: CAN Transmitter Stall in case of no Remote Frame in response to Tx packet with RTR=1

Description:

FlexCAN does not transmit an expected message when the same node detects an incoming Remote Request message asking for any remote answer. The issue happens when two specific conditions occur:

1) The Message Buffer (MB) configured for remote answer (with code "a") is the last MB. The last MB is specified by Maximum MB field in the Module Configuration Register (MCR[MAXMB]).

2) The incoming Remote Request message does not match its ID against the last MB ID. While an incoming Remote Request message is being received, the FlexCAN also scans the transmit (Tx) MBs to select the one with the higher priority for the next bus arbitration. It is expected that by the Intermission field it ends up with a selected candidate (winner). The coincidence of conditions (1) and (2) above creates an internal corner case that cancels the Tx winner and therefore no message will be selected for transmission in the next frame. This gives the appearance that the FlexCAN transmitter is stalled or "stops transmitting". The problem can be detectable only if the message traffic ceases and the CAN bus enters into Idle state after the described sequence of events.

There is NO ISSUE if any of the conditions below holds:

a) The incoming message matches the remote answer MB with code "a".

b) The MB configured as remote answer with code "a" is not the last one.

c) Any MB (despite of being Tx or Rx) is reconfigured (by writing its CS field) just after the Intermission field.

d) A new incoming message sent by any external node starts just after the Intermission field

Workaround:

Do not configure the last MB as a Remote Answer (with code "a").

1.15 ERR003442: CMU monitor: FXOSC/FIRC and FMPLL/FIRC relation

Description:

Functional CMU monitoring can only be guaranteed when the following conditions are met:

- FXOSC frequency must be greater than (FIRC / 2^RCDIV) + 0.5MHz in order to guarantee correct FXOSC monitoring
- FMPLL frequency must be greater than (FIRC / 4) + 0.5MHz in order to guarantee correct FMPLL monitoring

Workaround:

Refer to description

1.16 ERR003446: CTU: The CTU (Cross Trigger Unit) CLR_FLAG in EVTCFGR register does not function as expected

Description:

If the CTU CLR_FLG is set and the CTU is idle, a PIT triggered request to the CTU does not result in the correct ADC channel number being latched. The previous ADC channel number is latched instead of the requested channel number.

Workaround:

There is no software workaround to allow the CLR_FLAG functionality to operate correctly. Do not program the CLR_FLAG bit to '1'.

1.17 ERR003449: DEBUG: Device may hang due to external or 'functional' reset while using debug handshaking mechanism

Description:

If the low-power mode debug handshake has been enabled and an external reset or a 'functional' reset occurs while the device is in a low-power mode, the device will not exit reset.

Workaround:

The NPC_PCR[LP_DBG_EN] bit must be cleared to ensure the correct reset sequence.

1.18 ERR003466: LINFlexD: Register bus aborts are not generated on illegal accesses to reserved addresses within the register address space of LINFlexD

Description:

Register bus aborts are not generated on illegal accesses to reserved addresses within the register address space of LINFlexD. This is applicable to LINFlex modules supporting master-only mode.

Workaround:

None

1.19 ERR003512: ECSM: ECSM_PFEDR displays incorrect endianness

Description:

The ECSM_PFEDR register reports ECC data using incorrect endiannes. For example, a flash location that contains the data 0xAABBCCDD would be reported as 0xDDCCBBAA at ECSM_PFEDR.

This 32-bit register contains the data associated with the faulting access of the last, properly-enabled flash ECC event. The register contains the data value taken directly from the data bus.

Workaround:

Software must correct endianness.

1.20 ERR003570: MC_ME: Possibility of Machine Check on Low-Power Mode Exit

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.

1.21 ERR004340: LINFlexD: Buffer overrun can not be detected in UART Rx FIFO mode

Description:

When the LINFlexD is configured in UART Receive (Rx) FIFO mode, the Buffer Overrun Flag (BOF) bit of the UART Mode Status Register (UARTSR) register is cleared in the subsequent clock cycle after being asserted. User software can not poll the BOF to detect an overflow. The LINFlexD Error Combined Interrupt can still be triggered by the buffer overrun. This interrupt is enabled by setting the Buffer Overrun Error Interrupt Enable (BOIE) bit in the LIN Interrupt enable register (LINIER). But the BOF bit will be cleared when the interrupt routine is entered, preventing the user to identify the source of error.

Workaround:

Buffer overrun errors in UART FIFO mode can be detected by enabling only this source in the LIN Error Combined interrupt.

1.22 ERR004405: SR bit of LINFlexD GCR register is not cleared automatically by hardware

Description:

After setting the SR bit of GCR (Global Control Register) to reset the LinFlexD controller, this bit is not cleared automatically by the hardware, keeping the peripheral in reset state

Workaround:

This bit should be cleared by software to perform further operations

2 Revision history

Table 2. Document revision history

Date	Revision	Changes
23-Mar-2012	1	Initial release.

Please Read Carefully:

Information in this document is provided solely in connection with ST products. STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, modifications or improvements, to this document, and the products and services described herein at any time, without notice.

All ST products are sold pursuant to ST’s terms and conditions of sale.

Purchasers are solely responsible for the choice, selection and use of the ST products and services described herein, and ST assumes no liability whatsoever relating to the choice, selection or use of the ST products and services described herein.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted under this document. If any part of this document refers to any third party products or services it shall not be deemed a license grant by ST for the use of such third party products or services, or any intellectual property contained therein or considered as a warranty covering the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

UNLESS OTHERWISE SET FORTH IN ST’S TERMS AND CONDITIONS OF SALE ST DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE USE AND/OR SALE OF ST PRODUCTS INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE (AND THEIR EQUIVALENTS UNDER THE LAWS OF ANY JURISDICTION), OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS EXPRESSLY APPROVED IN WRITING BY TWO AUTHORIZED ST REPRESENTATIVES, ST PRODUCTS ARE NOT RECOMMENDED, AUTHORIZED OR WARRANTED FOR USE IN MILITARY, AIR CRAFT, SPACE, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS, NOR IN PRODUCTS OR SYSTEMS WHERE FAILURE OR MALFUNCTION MAY RESULT IN PERSONAL INJURY, DEATH, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE. ST PRODUCTS WHICH ARE NOT SPECIFIED AS "AUTOMOTIVE GRADE" MAY ONLY BE USED IN AUTOMOTIVE APPLICATIONS AT USER’S OWN RISK.

Resale of ST products with provisions different from the statements and/or technical features set forth in this document shall immediately void any warranty granted by ST for the ST product or service described herein and shall not create or extend in any manner whatsoever, any liability of ST.

ST and the ST logo are trademarks or registered trademarks of ST in various countries.

Information in this document supersedes and replaces all information previously supplied.

The ST logo is a registered trademark of STMicroelectronics. All other names are the property of their respective owners.

STMicroelectronics group of companies

Australia - Belgium - Brazil - Canada - China - Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Philippines - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

www.st.com