STM32U535xx and STM32U545xx device errata

Applicability

This document applies to the part numbers of STM32U535xx and STM32U545xx devices and the device variants as stated in this page.

It gives a summary and a description of the device errata, with respect to the device datasheet and reference manual RM0456.

Deviation of the real device behavior from the intended device behavior is considered to be a device limitation. Deviation of the description in the reference manual or the datasheet from the intended device behavior is considered to be a documentation erratum. The term *"errata"* applies both to limitations and documentation errata.

Table 1. Device summary

Reference	Part numbers
STM32U535xx	STM32U535CB, STM32U535CC, STM32U535CE, STM32U535JE, STM32U535NC, STM32U535NE, STM32U535RB, STM32U535RC, STM32U535RE, STM32U535VC, STM32U535VE
STM32U545xx	STM32U545CE, STM32U545JE, STM32U545NE, STM32U545RE, STM32U545VE

Table 2. Device variants

Reference	Silicon revision codes	
	Device marking(1)	REV_ID(2)
STM32U535xx and STM32U545xx	Z	0x1001
	Y	0x1003

1. *Refer to the device datasheet for how to identify this code on different types of package.*
2. *REV_ID[5:0] bitfield of DBGMCU_IDCODE fuse.*

Note: DBGMCU_IDCODE register is not automatically shadowed with OTP content, so a fuse read sequence must be issued to get the register updated once (clear after reading). Refer to product reference manual - BSEC section "Operations on fuses".

1 Summary of device errata

The following table gives a quick reference to the STM32U535xx and STM32U545xx device limitations and their status:

A = limitation present, workaround available

N = limitation present, no workaround available

P = limitation present, partial workaround available

"-" = limitation absent

Applicability of a workaround may depend on specific conditions of target application. Adoption of a workaround may cause restrictions to target application. Workaround for a limitation is deemed partial if it only reduces the rate of occurrence and/or consequences of the limitation, or if it is fully effective for only a subset of instances on the device or in only a subset of operating modes, of the function concerned.

Table 3. Summary of device limitations

Function	Section	Limitation	Status	
			Rev. Z	Rev. Y
Core	2.1.1	Access permission faults are prioritized over unaligned Device memory faults	N	N
System	2.2.1	LSE crystal oscillator may be disturbed by transitions on PC13	N	N
	2.2.2	The PWR_S3WU interrupt is generated for internal wake-up sources (WUSELx = 11)	A	A
	2.2.4	Flash programming can remain stuck in case of programming sequence error	A	A
	2.2.5	HardFault on wake-up from Stop mode may occur in debug mode	N	N
	2.2.6	Full JTAG configuration without NJTRST pin cannot be used	A	A
	2.2.7	Too low MSI frequency upon exit from Standby or Stop 3 mode	A	A
	2.2.8	PWR_BDCR1 is not write-protected by DBP	N	N
	2.2.9	MSIK clock cannot be stopped when used as kernel clock by MDF1 or ADF1	A	A
	2.2.10	Incorrect backup domain reset with V_{BAT} and V_{DD} supplied by different power sources	A	A
	2.2.11	EXTI LOCK bit does not lock privilege configuration	N	N
	2.2.12	Device may be locked upon system reset under Stop 2 mode	P	-
	2.2.14	PVD_IN I/Os analog switches are not functional when supplied by V_{DDA}	A	-
OCTOSPI	2.3.1	Memory-mapped write error response when DQS output is disabled	P	P
	2.3.2	Byte possibly dropped during an SDR read in clock mode 3 when a transfer gets automatically split	A	A
	2.3.3	Deadlock or write-data corruption after spurious write to a misaligned address in OCTOSPI_AR register	N	N
	2.3.4	Read data corruption after a few bytes are skipped when crossing a four-byte boundary	A	A
	2.3.5	At least six cycles memory latency must be set when DQS is used for HyperBus™ memories	A	A
	2.3.6	Data write discarded in memory-mapped mode if a write to a misaligned address is directly followed by a request to the same address	A	A
	2.3.7	Setting the ABORT bit does not generate an error on the AHB bus for undefined-length incremental burst transfers	P	P

Function	Section	Limitation	Status	
			Rev. Z	Rev. Y
OCTOSPI	2.3.8	Read data corruption when a wrap transaction is followed by a linear read to the same MSB address	N	N
	2.3.9	Transactions are limited to 8 Mbytes in OctaRAM™ memories	N	N
ADC1	2.4.1	ADC_AWDy_OUT reset by non-guarded channels	A	A
	2.4.2	Injected data stored in the wrong ADC_JDRx registers	A	A
ADC	2.4.3	ADC I/Os analog switches are not functional when supplied by V_{DDA}	A	-
ADC4	2.6.1	ADC4 conversion error when used simultaneously with ADC1	P	P
VREFBUF	2.7.1	V_{REFBUF_OUT} voltage overshoots in Range 4, Stop 1 or Stop 2 mode	A	A
COMP	2.8.1	COMP I/Os analog switches are not functional when supplied by V_{DDA}	A	-
OPAMP	2.9.1	OPAMP I/Os analog switches are not functional when supplied by V_{DDA}	A	-
TSC	2.10.1	TSC I/Os analog switches are not functional when supplied by V_{DDA}	A	-
LPTIM	2.12.1	Device may remain stuck in LPTIM interrupt when entering Stop mode	A	A
	2.12.2	ARRM and CMPM flags are not set when APB clock is slower than kernel clock	P	P
	2.12.3	Interrupt status flag is cleared by hardware upon writing its corresponding bit in LPTIM_DIER register	N	N
IWDG	2.13.1	IWDG is stopped when BDRST is set	P	P
RTC and TAMP	2.14.1	Alarm flag may be repeatedly set when the core is stopped in debug	N	N
	2.14.2	Binary mode: SSR is not reloaded with 0xFFFF FFFF when SSCLR = 1	A	A
	2.14.3	Parasitic tamper detection when debugger is used in RDP Level 0	A	A
I2C	2.15.1	Wrong data sampling when data setup time ($t_{SU;DAT}$) is shorter than one I2C kernel clock period	P	P
	2.15.2	Spurious bus error detection in master mode	A	A
USART	2.16.1	Data corruption due to noisy receive line	A	A
	2.16.2	Wrong data received by SPI slave receiver in autonomous mode with CPOL = 1	A	A
	2.16.3	Received data may be corrupted upon clearing the ABREN bit	A	A
	2.16.4	Noise error flag set while ONEBIT is set	N	N
LPUART	2.17.1	Possible LPUART transmitter issue when using low BRR[15:0] value	P	P
SPI	2.18.1	Possible corruption of last-received data depending on CRCSIZE setting	A	A
	2.18.2	MODF flag cannot generate interrupt	A	A
	2.18.3	RDY output failure at high serial clock frequency	N	N
	2.18.4	Master communication suspension fails in Autonomous mode	N	N
	2.18.5	SPE may not be cleared upon MODF event	A	A
	2.18.6	SPI slave stalls with masters not providing extra SCK periods upon *Not ready* signalling	A	A
	2.18.7	Truncation of SPI output signals after EOT event	A	A
SDMMC	2.19.1	Command response and receive data end bits not checked	N	N
FDCAN	2.20.1	Desynchronization under specific condition with edge filtering enabled	A	A
	2.20.2	Tx FIFO messages inverted under specific buffer usage and priority setting	A	A
USB	2.21.1	Buffer description table update completes after CTR interrupt triggers	A	A

The following table gives a quick reference to the documentation errata.

Table 4. Summary of device documentation errata

Function	Section	Documentation erratum
System	2.2.3	VDDUSB is internally connected to VDD in WLCSP56 packages
	2.2.13	SRAM ECC error flags and addresses are updated only if interrupt is enabled
ADC1	2.5.3	14-bit ADC offset error and integral linearity error values are increased in datasheets
SAES	2.11.1	Data transfer from TAMP_BKPxR to key registers must be done only in ascending order when KEYSEL[2:0] is set to 010 or 100

2 Description of device errata

The following sections describe the errata of the applicable devices with Arm® core and provide workarounds if available. They are grouped by device functions.

Note: Arm is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

2.1 Core

Reference manual and errata notice for the Arm® Cortex®-M33 core revision r0p4 is available from http://infocenter.arm.com. Only applicable information from the Arm errata notice is replicated in this document.

2.1.1 Access permission faults are prioritized over unaligned Device memory faults

Description

A load or store which causes an unaligned access to Device memory will result in an UNALIGNED UsageFault exception. However, if the region is not accessible because of the MPU access permissions (as specified in MPU_RBAR.AP), then the resulting MemManage fault will be prioritized over the UsageFault.

The failure occurs when the MPU is enabled and:

- A load/store access occurs to an address which is not aligned to the data type specified in the instruction.
- The memory access hits one region only.
- The region attributes (specified in the MAIR register) mark the location as Device memory.
- The region access permissions prevent the access (that is, unprivileged or write not allowed).

The MemManage fault caused by the access permission violation will be prioritized over the UNALIGNED UsageFault exception because of the memory attributes.

Workaround

None. However, it is expected that no existing software is relying on this behavior since it was permitted in Armv7-M.

2.2 System

2.2.1 LSE crystal oscillator may be disturbed by transitions on PC13

Description

On LQFP and UFQFPN packages, the LSE crystal oscillator clock frequency can be incorrect when PC13 is toggling in input or output (for example when used for RTC_OUT1).

The external clock input (LSE bypass) is not impacted by this limitation.

The WLCSP and UFBGA packages are not impacted by this limitation.

Workaround

None.

Avoid toggling PC13 when LSE is used on LQFP and UFQFPN packages.

2.2.2 The PWR_S3WU interrupt is generated for internal wake-up sources (WUSELx = 11)

Description

According to some reference manual versions, the PWR_S3WU interrupt is not generated for internal wake-up sources (when WUSELx = 11 in PWR_WUCR3 register). However, upon wake-up from Stop 3 mode with WUSELx = 11, two interrupts are generated: the PWR_S3WU and the internal wakeup source (RTC or TAMP) interrupts.

Note: The PWR_S3WU flag (WUFx in PWR_WUSR) is automatically cleared when clearing the internal wake-up source flag.

Workaround

Set the PWR_S3WU interrupt with a lower priority than the internal source interrupt (RTC or TAMP). Consequently, upon wake-up from Stop 3 mode, the RTC or TAMP interrupt is serviced first. Then, the PWR_S3WU interrupt is entered without the corresponding flag (WUFx in PWR_WUSR) being set. This interrupt service routine can be returned without clearing any flag.

2.2.3 VDDUSB is internally connected to VDD in WLCSP56 packages

Description

The VDDUSB is internally connected to VDD in the following these devices:

- STM32U535NEY6QTR
- STM32U545NEY6QTR
- STM32U535NCY6QTR

As a consequence, if USB is required, V_{DD} must be in the [3.0 V; 3.6 V] range. The latest revision of the datasheet (Revision 4) includes the following updates:

- Figure x: WLCSP56_SMPS ballout now correctly shows VDD instead of VDDUSB.
- Table x: STM32U535xx/545xx pin/ball definitions now shows pin number A1 moved from VDDUSB to VDD line.

Workaround

None.

2.2.4 Flash programming can remain stuck in case of programming sequence error

Description

If an operation is started while the write buffer is waiting for the next data (STRT or OPTSTRT is set while WDW is already set), the PGSERR flag is set and the programming sequence is expected to be aborted. However, the WDW bit remains set and the flash programming operation remains stuck, waiting for the second word to be written.

Workaround

Write the second word to the write buffer to either start the programming (if the programming sequence is correct) or raise a new error, which resets the WDW flag.

2.2.5 HardFault on wake-up from Stop mode may occur in debug mode

Description

A HardFault may occur at wake-up from Stop mode when the following conditions are met:

- Device is in debug mode.
- DBG_STOP bit is set in DBGMCU_CR.
- A wake-up event/interrupt from an SRD peripheral (except EXTI) occurs in a timing window of four clock cycles during Stop mode entry sequence. SRD peripherals are the ones connected to AHB3 and APB3.

Workaround

None.

2.2.6 Full JTAG configuration without NJTRST pin cannot be used

Description

When using the JTAG debug port in Debug mode, the connection with the debugger is lost if the NJTRST pin (PB4) is used as a GPIO or for an alternate function other than NJTRST. Only the 4-wire JTAG port configuration is impacted.

Workaround

Use the SWD debug port instead of the full 4-wire JTAG port.

2.2.7 Too low MSI frequency upon exit from Standby or Stop 3 mode

Description

The MSI clock frequency can be up to 25% lower than expected upon exit from Standby or Stop 3 mode, for a maximum of 200 µs.

Workaround

In case accuracy is needed, wait for 200 µs after exiting Standby or Stop 3 mode before using the MSI.

2.2.8 PWR_BDCR1 is not write-protected by DBP

Description

When the DBP bit of PWR_DBPR register is cleared, the write access to backup domain content is expected to be disabled. However, PWR_BDCR1 register is not write-protected when DBP = 0.

Workaround

None.

2.2.9 MSIK clock cannot be stopped when used as kernel clock by MDF1 or ADF1

Description

MSIK can be used as kernel clock by the MDF1 and ADF1 by configuring the MDF1SEL and ADF1SEL bitfields of RCC_CCIPRx registers.

Once selected as kernel clock source for MDF1 or ADF1 peripherals, MSIK can no longer be disabled using the MSIKON bits of the RCC_CR register. This is because MDF1 and ADF1 request their kernel clock by default after reset.

If the application requests an entry in Stop 0, Stop 1, or Stop 2 mode, the selected MSIK kernel clock remains enabled leading to an overconsumption in this mode. The Stop 3 mode is not concerned by this limitation. MDF1 is not concerned for Stop 2 mode.

Workaround

Before disabling the MSIK clocks, configure MDF1SEL and ADF1SEL to HCLK.

In case the application wants to enter Stop 0 or Stop 1 mode, and the MDF1 and ADF1 are not used during Stop mode, then configure MDF1SEL and ADF1SEL bitfields to HCLK before entering Stop mode. Upon Stop mode exit, restore the required MSIK kernel clock with the MDF1SEL and ADF1SEL bitfields. In case the application wants to enter Stop 2 mode and the ADF1 is not used during Stop mode, the same procedure is needed only for ADF1SEL.

In case the application wants to enter Stop 0, Stop 1, or Stop 2 mode and the MDF1 or ADF1 peripheral is used during Stop mode (Stop 0 and Stop 1 modes only for MDF1), then simply configure the MDF1SEL and ADF1SEL bitfields to required MSIK kernel clock. These peripherals request their kernel clock when they need it.

2.2.10 Incorrect backup domain reset with V_{BAT} and V_{DD} supplied by different power sources

Description

The backup domain reset may be missed upon backup domain power-on subsequent to a V_{BAT} power-off, in V_{BAT} mode, if the V_{BAT} voltage during the power-off phase drops to a few mV window before starting to rise again. In this critical window, the flip-flops are no longer able to safely retain the information, and the backup domain reset has not yet been triggered. This window is located in the range between 100 mV and 700 mV, with the exact position depending mainly on the device and on the temperature. The issue only occurs when MONEN = 0 in the PWR_BDCR1 register (backup domain supply and temperature monitoring are disabled).

This missed reset results in unpredictable values of the backup domain registers, which may cause a spurious behavior such as driving the LSCO output pin on PA2, raising an unexpected tamper event preventing the SRAM2 or PKA access, or influencing any of the backup domain functions.

Workaround

Apply one of the following measures to avoid the incorrect backup domain reset:

- If the extra consumption is acceptable, enable backup domain supply and temperature monitoring (MONEN = 1 in PWR_BDCR1). The maximum extra consumption is 1.2 µA from V_{BAT} when V_{DD} is above V_{BOR0}, and around 3.6 µA from V_{BAT} in V_{BAT} mode.
- In the application, let the V_{BAT} supply voltage fall to a level below 100 mV for more than 200 ms, before a new power-on.

If the two previous workarounds are not applicable and the boot follows a backup domain power-on reset:

- Erase the backup domain by software.

In order to discriminate the backup domain power-on reset from a power-on reset or exit from Shutdown mode, at least one backup register (called, for example, BackupTestRegister) must be previously programmed with a BKP_REG_VAL value with 16 bits set and 16 bits cleared. Robustness of this workaround can be significantly improved by using a CRC rather than registers. The registers are subject to backup domain reset.

The workaround consists in calculating the CRC of the backup domain registers: RCC_BDCR and RTC/TAMP registers, excluding bits modified by hardware.

The CRC result can be stored in the backup register instead of a fixed value. This value needs to be updated for each modification of values covered by CRC, such as by using CRC peripheral.

At the very beginning of the boot code, insert the following software sequence:

1. Check if the BORRSTF flag of RCC_CSR is set (the reset is caused by a power-on).
2. If true, check that the BackupTestRegister content is different from BKP_REG_VAL, or the CRC recalculation is different from stored results, accordingly the chosen workaround implementation.
3. If true and if no tamper flag is set (when tamper detection is enabled), the reset is caused by a backup domain power-on. Apply the following sequence:
   a. Enable the PWR clock in RCC, by setting the PWREN bit of RCC_AHB3ENR.
   b. Enable backup domain access, by setting the DBP bit of PWR_DBPR.
   c. Reset the backup domain by:
      i. Writing 0x0001 0000 to RCC_BDCR, which sets the BDRST bit and clears other register bits that might not be reset
      ii. Reading RCC_BDCR, to make the reset time long enough
      iii. Writing 0x0000 0000 to RCC_BDCR, to clear the BDRST bit
   d. Clear BORRSTF, by setting the RMVF bit of RCC_CSR.

2.2.11 EXTI LOCK bit does not lock privilege configuration

Description

Both security and privilege configuration of the extended interrupts and event controller (EXTI) must be locked when the LOCK bit of the EXTI lock register (EXTI_LOCKR) is set. The EXTI security configuration register (EXTI_SECCFGR1) is locked as expected, but the EXTI privilege configuration register (EXTI_PRIVCFGR1) can still be modified when the LOCK bit is set.

Workaround

None.

2.2.12 Device may be locked upon system reset under Stop 2 mode

Description

A system reset occurs when the MCU is in Stop 2 mode on LDO regulator, with at least, one RAM in power-down (SRAMx, Caches, or peripheral SRAMs). This may lock the device in a high consumption state (dozens of mA). The IWDG, the external NRST pin, and the BOR thresholds 1 to 4 are the reset sources that create this limitation.

Only a power-on sequence recovers from this state: V_{DD} must drop below V_{BOR0} (brownout reset minimum threshold value), then raise to its expected value.

This limitation is not present:

- upon wake-up from Stop 2 mode with other wake-up sources (GPIO or peripheral interrupt).
- if the device is supplied by the SMPS regulator during Stop 2 mode.
- in Stop 0, Stop 1, and Stop 3 modes.

Workaround

Keep all SRAMs powered on during Stop 2 mode that is all xRAMxPD/PDS bits in PWR_CRx registers must be kept at their reset state.

2.2.13 SRAM ECC error flags and addresses are updated only if interrupt is enabled

Description

In the reference manual RM0456, up to version 4, it is stated that:

- If a single error is detected, SEDC and CSEDC flags are set in RAMCFG_MxISR and RAMCFG_MxICR respectively. The ECC single error address is updated in RAMCFG_MxSEAR.
- If a double error is detected, DED and CDED flags are set in RAMCFG_MxISR and RAMCFG_MxICR respectively. The ECC double error address is updated in RAMCFG_MxDEAR.

However, the real behavior is that:

- If a single error is detected, SEDC and CSEDC flags are set, and the associated ECC single error address is updated only if the single error interrupt is enabled (SEIE bit of RAMCFG_MxIER is set).
- If a double error is detected, DED and CDED flags are set, and the associated ECC double error address is updated only if double error interrupt or NMI is enabled (DEIE bit or ECCNMI bit of RAMCFG_MxIER is set).

This is a documentation error rather than a device limitation.

Workaround

When the application needs to get the ECC error flags and addresses status without servicing the associated interrupts, the RAMCFG SEIE/DEIE interrupts must be enabled with the associated NVIC RAMCFG vector 5 disabled.

2.2.14 PVD_IN I/Os analog switches are not functional when supplied by V_{DDA}

Description

The PVD_IN IO analog switches are not functional when supplied by V_{DDA} (BOOSTEN = ANASWVDD = 0 in SYSCFG_CFGR1 register) or when supplied by booster voltage (BOOSTEN = 1 in SYSCFG_CFGR1 register). Consequently, PVD_IN does not work properly.

Note: There is no PVD limitation when PVD_IN is not used.

Workaround

For all V_{DD} and V_{DDA} voltage settings:

1. Disable the booster (BOOSTEN = 0 in SYSCFG_CFGR1 register).
2. Supply the analog switch by V_{DD} (ANASWVDD = 1 in SYSCFG_CFGR1 register).

2.3 OCTOSPI

2.3.1 Memory-mapped write error response when DQS output is disabled

Description

If the DQSE control bit of the OCTOSPI_WCCR register is cleared for memories without DQS pin, it results in an error response for every memory-mapped write request.

Workaround

When doing memory-mapped writes, set the DQSE bit of the OCTOSPI_WCCR register, even for memories that have no DQS pin.

2.3.2 Byte possibly dropped during an SDR read in clock mode 3 when a transfer gets automatically split

Description

When reading a continuous stream of data from sequential addresses in a serial memory, the OCTOSPI can interrupt the transfer and automatically restart it at the next address when features generating transfer splits (CSBOUND, REFRESH, TIMEOUT or MAXTRAN) are active. Thus, a single continuous transfer can effectively be split into multiple smaller transfers.

When the OCTOSPI is configured to use clock mode 3 (CKMODE bit of the OCTOSPI_DCR1 register set) and a continuous stream of data is read in SDR mode (DDTR bit of the OCTOSPI_CCR register cleared), the last byte sent by the memory before an automatic split gets dropped, thus causing all the subsequent bytes to be seen one address earlier.

Workaround

Use clock mode 0 (CKMODE bit of the OCTOSPI_DCR1 register cleared) when in SDR mode.

2.3.3 Deadlock or write-data corruption after spurious write to a misaligned address in OCTOSPI_AR register

Description

Upon writing a misaligned address to OCTOSPI_AR just before switching to memory-mapped mode (without first triggering the indirect write operation), with the OCTOSPI configured as follows:

- FMODE = 00 in OCTOSPI_CR (indirect write mode)
- DQSE = 1 in OCTOSPI_CCR (DQS active)

then, the OCTOSPI may be deadlocked on the first memory-mapped request or the first memory-mapped write to memory (and any sequential writes after it) may be corrupted.

An address is misaligned if:

- the address is odd and the OCTOSPI is configured to send two bytes of data to the memory every cycle (octal-DTR mode or dual-quad-DTR mode), or
- the address is not a multiple of four when the OCTOSPI is configured to send four bytes of data to the memory (16-bit DTR mode or dual-octal DTR mode).

If the OCTOSPI_AR register is reprogrammed with an aligned address (without triggering the indirect write between the two writes to OCTOSPI register), the data sent to the memory during the indirect write operation are also corrupted.

Workaround

None.

2.3.4 Read data corruption after a few bytes are skipped when crossing a four-byte boundary

Description

A memory-mapped read is corrupted when the following sequence occurs:

1. An 8- or 16-bit read is performed in a four-byte window, and the last byte of this window is not read.
2. Any read in the next four-byte window is done before the completion of the previous read memory prefetch.

OCTOSPI immediately responds to this read request, even before the data are read from memory, thus resulting in incorrect data read.

If the next read operations are issued to consecutive addresses, then the read data are also corrupted.

This limitation is not present when the SDMMC accesses the OCTOSPI since the SDMMC performs only 32-bit read accesses.

Workaround

Apply one of the following measures:

- Perform only 32-bit memory-mapped read accesses. For CPU read accesses, enable ICACHE and/or DCACHE and configure the OCTOSPI memory as cacheable (CACHE only performs 32-bit read accesses).
  For system DMA, use only 32-bit data size for read accesses.
- If this is not possible and an 8-bit or 16-bit read is done at the beginning of a four-byte window, ensure that the next access is not a read from the next four-byte window or that the second access occurs after the data at the skipped addresses are prefetched from memory.

2.3.5 At least six cycles memory latency must be set when DQS is used for HyperBus™ memories

Description

For HyperBus™ memories, the TACC[7:0] bitfield of the OCTOSPI_HLCR register enables the setting of the memory latency in number of clock cycles. These dummy cycles are inserted between the address and the data phases during read operations.

When the DQS signal is used for HyperBus™ memories, and the number of latency clock cycles programmed in TACC[7:0] is lower than six, a deadlock occurs during read operations.

Workaround

Configure the memory and the octo-SPI controller to have at least six clock cycles of latency.

2.3.6 Data write discarded in memory-mapped mode if a write to a misaligned address is directly followed by a request to the same address

Description

In memory-mapped mode with DQS enabled, a data write is discarded if the write targets a misaligned address and is directly followed by a request (cycle by cycle on AHB) to the same address.

An address is misaligned if the address is odd and the OCTOSPI is configured to send two bytes of data to the memory on every cycle that is targeted by one of the following modes:

- Octal DTR
- Dual-quad DTR

Workaround

Use one of the following measures:

- Configure the OCTOSPI to issue commands to aligned addresses, that is to an even address when two bytes are transferred during each clock cycle.
- Avoid consecutive back-to-back (AHB cycle by cycle) accesses to the memory after a write to a memory mapped at the same address. Instead, insert a NOP (no operation) or a software delay between the two accesses.

2.3.7 Setting the ABORT bit does not generate an error on the AHB bus for undefined-length incremental burst transfers

Description

An AHB error is expected to be generated when the ABORT bit of the OCTOSPI_CR register is set while a request is ongoing.

Instead, the controller does not trigger any AHB error if the ongoing request is an undefined-length incremental burst AHB transfer.

An AHB error is generated for all other transfer types.

Workaround

When possible, wait for the end of the transfer before setting the ABORT bit.

2.3.8 Read data corruption when a wrap transaction is followed by a linear read to the same MSB address

Description

If a wrap transaction is followed by a linear read having the same MSB start address as the wrap (), then the linear read is wrongly considered as a sequential transaction to the previous one, taking back the prefetched data and causing data corruption.

Notice that for a wrap transaction, the prefetch starts after the last address of the wrap window.

Workaround

As prefetch cannot be disabled, there is no workaround. However, the issue is seldom encountered since wrap operations are mostly initiated by the internal cache to refresh its cacheline. All the other masters must avoid retrieving data by using a linear read access to the same MSB address as the wrap, which has been just completed.

2.3.9 Transactions are limited to 8 Mbytes in OctaRAM™ memories

Description

When the controller is configured in Macronix OctaRAM™ mode, by setting the MTYP[2:0] bitfield of the OCTOSPI_DCR1 register to 011, only 13 bits of row address are decoded and sent to the memory, meaning that only 8 K of 1-Kbyte blocks can be accessed (8 Mbytes).

Workaround

None.

This limitation is not present for PSRAMs or HyperRAM™ memories.

2.4 ADC

2.4.1 ADC_AWDy_OUT reset by non-guarded channels

Description

ADC_AWDy_OUT is set when a guarded conversion of a regular or injected channel is outside the programmed thresholds. It is reset after the end of the next guarded conversion that is inside the programmed thresholds.

However, the ADC_AWDy_OUT signal is also reset at the end of conversion of non-guarded channels, both regular and injected.

Workaround

When ADC_AWDy_OUT is enabled, it is recommended to use only the ADC channels that are guarded by a watchdog.

If ADC_AWDy_OUT is used with ADC channels that are not guarded by a watchdog, take only ADC_AWDy_OUT rising edge into account.

2.4.2 Injected data stored in the wrong ADC_JDRx registers

Description

When the AHB clock frequency is higher than the ADC clock frequency after the prescaler is applied (ratio > 10), if a JADSTP command is issued to stop the injected conversion (JADSTP bit set to 1 in ADC_CR register) at the end of an injected conversion, exactly when the data are available, then the injected data are stored in ADC_JDR1 register instead of ADC_JDR2/3/4 registers.

Workaround

Before setting JADSTP bit, check that the JEOS flag is set in ADC_ISR register (end of injected channel sequence).

2.4.3 ADC I/Os analog switches are not functional when supplied by V_{DDA}

Description

The ADC IO analog switches are not functional when supplied by V_{DDA} (BOOSTEN = ANASWVDD = 0 in SYSCFG_CFGR1 register), or when supplied by booster voltage (BOOSTEN = 1 in SYSCFG_CFGR1 register).

Consequently, the ADC IO switch impedance is higher than expected, and the maximum external input impedance (R_{AIN}) for ADC1 and ADC4 tables are wrong in the datasheets.

The analog switches booster is not functional, consequently the datasheet "Analog switches booster characteristics" table is not applicable to silicon rev. Z.

Workaround

For all V_{DD} and V_{DDA} voltage settings:

1. Disable the booster (BOOSTEN = 0 in SYSCFG_CFGR1 register).
2. Supply the analog switch by V_{DD} (ANASWVDD = 1 in SYSCFG_CFGR1 register).
3. When V_{DD} < 2.4V, increase the ADC sampling time compared to values provided in datasheets.

Table 5 shows the maximum external input impedance values for ADC1.

Table 5. Maximum ADC1 R_{AIN}

1. Guaranteed by design
2. Values without external capacitor

Resolution	RAIN (Ω)	New sampling time [ns]	New sampling cycle at 5 MHz	New sampling cycle at 55 MHz
14bits 2 LSB gabarit	47	337	5	20

Resolution	RAIN (Ω)	New sampling time [ns]	New sampling cycle at 5 MHz	New sampling cycle at 55 MHz
14bits 2 LSB gabarit	68	342	5	20
	100	399	5	36
12 bits	47	320	5	20
	68	318	5	20
	100	327	5	20
	150	334	5	20
	220	340	5	20
	330	342	5	20
	470	389	5	36
10 bits	47	304	5	20
	68	307	5	20
	100	309	5	20
	150	310	5	20
	220	319	5	20
	330	325	5	20
	470	348	5	20
	680	370	5	36
	1 000	403	5	36
	1 500	461	5	36
	2 200	599	5	36
8 bits	47	292	5	20
	68	293	5	20
	100	294	5	20
	150	298	5	20
	220	298	5	20
	330	308	5	20
	470	306	5	20
	680	334	5	20
	1 000	364	5	36
	1 500	395	5	36
	2 200	461	5	36
	3 300	543	5	36
	4 700	664	5	68
	6 800	882	5	68
	10 000	1 362	12	391

Table 6 shows the maximum external input impedance values for ADC4.

Table 6. Maximum ADC4 R_{AIN}

1. Guaranteed by design
2. Values without external capacitor

Resolution	RAIN (Ω)	New sampling time [ns]	New sampling cycle at 35 MHz	New sampling cycle at 55 MHz
12 bits	47	658	40	40
	68	685	40	40
	100	725	40	80
	150	793	40	80
	220	885	40	80
	330	1 031	40	80
	470	1 220	80	80
	680	1 498	80	815
	1 000	1 925	80	815
	1 500	2 585	815	815
	2 200	4 460	815	815
	3 300	5 448	815	815
	4 700	7 593	815	815
	6 800	10 791	815	815
	10 000	15 506	815	NA
	15 000	22 556	815	NA
	22 000	31 761	NA	NA
	33 000	44 836	NA	NA
10 bits	47	203	8	13
	68	211	8	13
	100	220	13	13
	150	248	13	20
	220	261	13	20
	330	301	13	20
	470	350	13	20
	680	456	20	40
	1 000	585	40	40
	1 500	787	40	80
	2 200	1 069	40	80
	3 300	1 510	80	815
	4 700	2 067	80	815
	6 800	2 892	815	815
	10 000	4 126	815	815
	15 000	5 992	815	815
	22 000	8 760	815	815
	33 000	12 922	815	815
	47 000	17 801	815	NA
8 bits	47	105	8	8
	68	106	8	8
	100	109	8	8
	150	118	8	8

Resolution	RAIN (Ω)	New sampling time [ns]	New sampling cycle at 35 MHz	New sampling cycle at 55 MHz
8 bits	220	128	8	8
	330	145	8	13
	470	165	8	13
	680	200	8	13
	1 000	249	13	20
	1 500	330	13	20
	2 200	442	20	40
	3 300	620	40	40
	4 700	844	40	80
	6 800	1 180	80	80
	10 000	1 681	80	815
	15 000	2 456	815	815
	22 000	3 510	815	815
	33 000	5 109	815	815
	47 000	7 018	815	815
6 bits	47	74	4	8
	68	73	4	8
	100	74	4	8
	150	77	4	8
	220	78	4	8
	330	82	4	8
	470	95	4	8
	680	113	8	8
	1 000	141	8	13
	1 500	181	8	13
	2 200	243	13	20
	3 300	335	13	20
	4 700	456	20	40
	6 800	636	40	40
	10 000	912	40	80
	22 000	1 922	80	815
	33 000	2 830	815	815
	47 000	3 954	815	815

2.5 ADC1

2.5.1 ADC_AWDy_OUT reset by non-guarded channels

Description

ADC_AWDy_OUT is set when a guarded conversion of a regular or injected channel is outside the programmed thresholds. It is reset after the end of the next guarded conversion that is inside the programmed thresholds.

However, the ADC_AWDy_OUT signal is also reset at the end of conversion of non-guarded channels, both regular and injected.

Workaround

When ADC_AWDy_OUT is enabled, it is recommended to use only the ADC channels that are guarded by a watchdog.

If ADC_AWDy_OUT is used with ADC channels that are not guarded by a watchdog, take only ADC_AWDy_OUT rising edge into account.

2.5.2 Injected data stored in the wrong ADC_JDRx registers

Description

When the AHB clock frequency is higher than the ADC clock frequency after the prescaler is applied (ratio > 10), if a JADSTP command is issued to stop the injected conversion (JADSTP bit set to 1 in ADC_CR register) at the end of an injected conversion, exactly when the data are available, then the injected data are stored in ADC_JDR1 register instead of ADC_JDR2/3/4 registers.

Workaround

Before setting JADSTP bit, check that the JEOS flag is set in ADC_ISR register (end of injected channel sequence).

2.5.3 14-bit ADC offset error and integral linearity error values are increased in datasheets

Description

STM32U535 and STM32U545 datasheets revision 1 (DS14216 and DS14217) specify that:

- The offset error maximum value is ±5 LSB in singled ended mode, while the correct value is ±12 LSB.
- The integral linearity error maximum value is ±5 LSB in singled ended mode, while the correct value is ±7 LSB.

This is a documentation error rather than a device limitation. This error has no impact on the ADC accuracy.

Workaround

None.

2.6 ADC4

2.6.1 ADC4 conversion error when used simultaneously with ADC1

Description

The ADC4 conversion is disturbed when carried out simultaneously with an ADC1 conversion and results in an ADC4 conversion error. The error is due to a V_{REF+} disturbance during an ADC1 sampling phase, and occurs regardless of which V_{REF+} is provided: either by external reference, or by VREFBUF.

Workaround

ADC1 and ADC4 analog converter clocks must be identical (same clock source, same frequency, and same phase) to avoid any disturbance. This is possible only when both ADC prescalers are programmed without any division factor. The bitfields of the both registers listed below must be set to 0b0000:

- ADC1 prescaler: PRESC[3:0] in the ADC12_CCR register
- ADC4 prescaler: PRESC[3:0] in the ADC4_CCR register

ADC1 and ADC4 analog converter clock frequencies must not exceed 55 MHz and the ADC4 clock duty cycle must be set to between 45% and 55%. Selecting AHB clock as the ADC kernel clock limits the whole AHB to 55 MHz. Other independent clock sources can be used to avoid this drawback. As a result, the triggers from timers, that are clocked by AHB clock, have an uncertainty due to trigger synchronization delay from timers AHB clock to ADC asynchronous kernel clock. In case PLL output pll2_r_ck is used as ADC kernel clock, the PLL division (controlled by the PLL2R[6:0] bitfield in the RCC_ RCC_PLL2_DIVR register) must be set to an even division (division by 2, 4, and so on) to guarantee a 50% ratio.

Note: The use of ADC kernel clock division factor is possible if ADC1 and ADC4 do not convert simultaneously.

2.7 VREFBUF

2.7.1 V_{REFBUF_OUT} voltage overshoots in Range 4, Stop 1 or Stop 2 mode

Description

The V_{REFBUF_OUT} output voltage overshoots when started:

- while the regulator is in Range 4,
- while the device is in Stop 1 or Stop 2 mode.

Workaround

Modify the regulator voltage range to Range 1, 2, or 3 before enabling the voltage reference buffer.

2.8 COMP

2.8.1 COMP I/Os analog switches are not functional when supplied by V_{DDA}

Description

The COMP IO analog switches are not functional when supplied by V_{DDA} (BOOSTEN = ANASWVDD = 0 in SYSCFG_CFGR1 register) or when supplied by booster voltage (BOOSTEN = 1 in SYSCFG_CFGR1 register). Consequently, COMP does not work properly.

Workaround

For all V_{DD} and V_{DDA} voltage settings:

1. Disable the booster (BOOSTEN = 0 in SYSCFG_CFGR1 register).
2. Supply the analog switch by V_{DD} (ANASWVDD = 1 in SYSCFG_CFGR1 register).

2.9 OPAMP

2.9.1 OPAMP I/Os analog switches are not functional when supplied by V_{DDA}

Description

The OPAMP IO analog switches are not functional when supplied by V_{DDA} (BOOSTEN = ANASWVDD = 0 in SYSCFG_CFGR1 register) or when supplied by booster voltage (BOOSTEN = 1 in SYSCFG_CFGR1 register). Consequently, OPAMP does not work properly.

Workaround

For all V_{DD} and V_{DDA} voltage settings:

1. Disable the booster (BOOSTEN = 0 in SYSCFG_CFGR1 register).
2. Supply the analog switch by V_{DD} (ANASWVDD = 1 in SYSCFG_CFGR1 register).

2.10 TSC

2.10.1 TSC I/Os analog switches are not functional when supplied by V_{DDA}

Description

The TSC IO analog switches are not functional when supplied by V_{DDA} (BOOSTEN = ANASWVDD = 0 in SYSCFG_CFGR1 register), or when supplied by booster voltage (BOOSTEN = 1 in SYSCFG_CFGR1 register). Consequently, TSC does not work properly.

Note: When ASV = 0 in PWR_SVMCR register, the IO analog switches are supplied by V_{DD}, so the limitation is not present.

Workaround

When ASV = 1, for all V_{DD} and V_{DDA} voltage settings:

1. Disable the booster (BOOSTEN = 0 in SYSCFG_CFGR1 register).
2. Supply the analog switch by V_{DD} (ANASWVDD = 1 in SYSCFG_CFGR1 register).

2.11 SAES

2.11.1 Data transfer from TAMP_BKPxR to key registers must be done only in ascending order when KEYSEL[2:0] is set to 010 or 100

Description

The KEYSEL[2:0] bitfield of the SAES_CR register defines the source of the key information to use in the SAES cryptographic core:

- When KEYSEL[2:0] is set to 010, the boot hardware key (BHK), stored in tamper-resistant secure backup registers, is entirely transferred into the key registers upon a secure application performing a single read of all TAMP_BKPxR registers (x = 0 to 3 for KEYSIZE = 0, x = 0 to 7 for KEYSIZE = 1).
- When KEYSEL[2:0] is set to 100, the XOR combination of DHUK and BHK is entirely transferred into the key registers upon a secure application performing a single read of all TAMP_BKPxR registers (x = 0 to 3 for KEYSIZE = 0, x = 0 to 7 for KEYSIZE = 1).

Some revisions of the reference manual may wrongly specify that the read operation can be performed either in ascending or descending order, while it must be performed always in **ascending** order.

This is a documentation issue rather than a product limitation.

Workaround

No application workaround is required, provided that the read operation to the TAMP_BKPxR registers is always done in ascending order.

2.12 LPTIM

2.12.1 Device may remain stuck in LPTIM interrupt when entering Stop mode

Description

This limitation occurs when disabling the low-power timer (LPTIM).

When the user application clears the ENABLE bit in the LPTIM_CR register within a small time window around one LPTIM interrupt occurrence, then the LPTIM interrupt signal used to wake up the device from Stop mode may be frozen in active state. Consequently, when trying to enter Stop mode, this limitation prevents the device from entering low-power mode and the firmware remains stuck in the LPTIM interrupt routine.

This limitation applies to all Stop modes and to all instances of the LPTIM. Note that the occurrence of this issue is very low.

Workaround

In order to disable a low power timer (LPTIMx) peripheral, do not clear its ENABLE bit in its respective LPTIM_CR register. Instead, reset the whole LPTIMx peripheral via the RCC controller by setting and resetting its respective LPTIMxRST bit in the relevant RCC register.

2.12.2 ARRM and CMPM flags are not set when APB clock is slower than kernel clock

Description

When LPTIM is configured in one shot mode and APB clock is lower than kernel clock, there is a chance that ARRM and CMPM flags are not set at the end of the counting cycle defined by the repetition value REP[7:0]. This issue can only occur when the repetition counter is configured with an odd repetition value.

Workaround

To avoid this issue the following formula must be respected:

{ARR, CMP} ≥ KER_CLK / (2* APB_CLK),

where APB_CLK is the LPTIM APB clock frequency, and KER_CLK is the LPTIM kernel clock frequency. ARR and CMP are expressed in decimal value.

Example: The following example illustrates a configuration where the issue can occur:

- APB clock source (MSI) = 1 MHz , Kernel clock source (HSI) = 16 MHz
- Repetition counter is set with REP[7:0] = 0x3 (odd value)

The above example is subject to issue, unless the user respects:

{CMP, ARR} ≥ 16 MHz / (2 * 1 MHz)

→ ARR must be ≥ 8 and CMP must be ≥ 8

Note: REP set to 0x3 means that effective repetition is REP+1 (= 4) but the user must consider the parity of the value loaded in LPTIM_RCR register (=3, odd) to assess the risk of issue.

2.12.3 Interrupt status flag is cleared by hardware upon writing its corresponding bit in LPTIM_DIER register

Description

When any interrupt bit of the LPTIM_DIER register is modified, the corresponding flag of the LPTIM_ISR register is cleared by hardware.

Workaround

None.

2.13 IWDG

2.13.1 IWDG is stopped when BDRST is set

Description

IWDG, once started, is expected to stop only in case of system reset. However, the LSI (IWDG clock) is stopped when BDRST is set in the RCC_BDCR register.

In addition, the BDRST bit is not protected against non-secure access when LSI or IWDG is secure.

Workaround

If a Backup domain reset must be done, set BDRST and clear it right after to minimize the duration LSI is stopped.

BDRST can be protected against non-secure access by configuring at least one function of RTC or TAMP as secure.

2.14 RTC and TAMP

2.14.1 Alarm flag may be repeatedly set when the core is stopped in debug

Description

When the core is stopped in debug mode, the clock is supplied to subsecond RTC alarm downcounter even when the device is configured to stop the RTC in debug.

As a consequence, when the subsecond counter is used for alarm condition (the MASKSS[3:0] bitfield of the RTC_ALRMASSR and/or RTC_ALRMBSSR register set to a non-zero value) and the alarm condition is met just before entering a breakpoint or printf, the ALRAF and/or ALRBF flag of the RTC_SR register is repeatedly set by hardware during the breakpoint or printf, which makes any attempt to clear the flag(s) ineffective.

Workaround

None.

2.14.2 Binary mode: SSR is not reloaded with 0xFFFF FFFF when SSCLR = 1

Description

When SSCLR bit of the RTC_ALRMxSSR register is set when in binary mode, SSR is reloaded with 0xFFFF FFFF at the end of the ck_apre cycle when RTC_SSR is set to RTC_ALRxBINR (*x* stands for either A or B)

RTC_SSR is not reloaded with 0xFFFF FFFF if RTC_ALRxBINR is modified while RTC_SSR is set to RTC_ALRxBINR. Rather, SSR continues to decrement.

Workaround

The workarounds are described for alarm A, and can be applied in the same manner for alarm B. Two workarounds are proposed, the second one requires to use the second alarm.

- Wait for one ck_apre cycle after an alarm A event before changing the RTC_ALRABINR register value.
- Do not reprogram RTC_ALRABINR following the alarm A event itself. Instead, use alarm B configured with RTC_ALRBBINR set to 0xFFFF FFFF, and reprogram RTC_ALRABINR after the alarm B event. This ensures that one ck_apre cycle elapses following the alarm A event.

2.14.3 Parasitic tamper detection when debugger is used in RDP Level 0

Description

The internal tamper 6 flag (ITAMP6F) can be unexpectedly set in the TAMP status register (TAMP_SR) when a debugger is connected in RDP Level 0, in case a switch to V_{BAT} occurs (V_{DD} is below the BOR0 threshold).

Workaround

Keep internal tamper 6 flag disabled as long as debug is needed, and enable it once development phase is complete. The tamper flag cannot be set if no debug access is done.

2.15 I2C

2.15.1 Wrong data sampling when data setup time ($t_{SU;DAT}$) is shorter than one I2C kernel clock period

Description

The I^2C-bus specification and user manual specify a minimum data setup time ($t_{SU;DAT}$) as:

- 250 ns in Standard mode
- 100 ns in Fast mode
- 50 ns in Fast mode Plus

The device does not correctly sample the I^2C-bus SDA line when $t_{SU;DAT}$ is smaller than one I2C kernel clock (I^2C-bus peripheral clock) period: the previous SDA value is sampled instead of the current one. This can result in a wrong receipt of slave address, data byte, or acknowledge bit.

Workaround

Increase the I2C kernel clock frequency to get I2C kernel clock period within the transmitter minimum data setup time. Alternatively, increase transmitter's minimum data setup time. If the transmitter setup time minimum value corresponds to the minimum value provided in the I^2C-bus standard, the minimum I2CCLK frequencies are as follows:

- In Standard mode, if the transmitter minimum setup time is 250 ns, the I2CCLK frequency must be at least 4 MHz.
- In Fast mode, if the transmitter minimum setup time is 100 ns, the I2CCLK frequency must be at least 10 MHz.
- In Fast-mode Plus, if the transmitter minimum setup time is 50 ns, the I2CCLK frequency must be at least 20 MHz.

2.15.2 Spurious bus error detection in master mode

Description

In master mode, a bus error can be detected spuriously, with the consequence of setting the BERR flag of the I2C_SR register and generating bus error interrupt if such interrupt is enabled. Detection of bus error has no effect on the I^2C-bus transfer in master mode and any such transfer continues normally.

Workaround

If a bus error interrupt is generated in master mode, the BERR flag must be cleared by software. No other action is required and the ongoing transfer can be handled normally.

2.16 USART

2.16.1 Data corruption due to noisy receive line

Description

In all modes, except synchronous slave mode, the received data may be corrupted if a glitch to zero shorter than the half-bit occurs on the receive line within the second half of the stop bit.

Workaround

Apply one of the following measures:

- Either use a noiseless receive line, or
- add a filter to remove the glitches if the receive line is noisy.

2.16.2 Wrong data received by SPI slave receiver in autonomous mode with CPOL = 1

Description

The SPI slave receiver device receives wrong data when all the following conditions are met:

- The USART is used in SPI master transmitter mode
- The autonomous mode is used
- The CPOL bit of the USART_CR2 register is set

Workaround

When the autonomous mode is used, do not set the CPOL bit in USART_CR2.

2.16.3 Received data may be corrupted upon clearing the ABREN bit

Description

The USART receiver may miss data or receive corrupted data when the auto baud rate feature is disabled by software (ABREN bit cleared in the USART_CR2 register) after an auto baud rate detection, while a reception is ongoing.

Workaround

Do not clear the ABREN bit.

2.16.4 Noise error flag set while ONEBIT is set

Description

When the ONEBIT bit is set in the USART_CR3 register (one sample bit method is used), the noise error (NE) flag must remain cleared. Instead, this flag is set upon noise detection on the START bit.

Workaround

None.

Note: Having noise on the START bit is contradictory with the fact that the one sample bit method is used in a noise free environment.

2.17 LPUART

2.17.1 Possible LPUART transmitter issue when using low BRR[15:0] value

Description

The LPUART transmitter bit length sequence is not reset between consecutive bytes, which could result in a jitter that cannot be handled by the receiver device. As a result, depending on the receiver device bit sampling sequence, a desynchronization between the LPUART transmitter and the receiver device may occur resulting in data corruption on the receiver side.

This happens when the ratio between the LPUART kernel clock and the baud rate programmed in the LPUART_BRR register (BRR[15:0]) is not an integer, and is in the three to four range. A typical example is when the 32.768 kHz clock is used as kernel clock and the baud rate is equal to 9600 baud, resulting in a ratio of 3.41.

Workaround

Apply one of the following measures:

- On the transmitter side, increase the ratio between the LPUART kernel clock and the baud rate. To do so:
  - Increase the LPUART kernel clock frequency, or
  - Decrease the baud rate.
- On the receiver side, generate the baud rate by using a higher frequency and applying oversampling techniques if supported.

2.18 SPI

2.18.1 Possible corruption of last-received data depending on CRCSIZE setting

Description

With the CRC calculation disabled (CRCEN = 0), the transfer size bitfield set to a value greater than zero (TSIZE[15:0] > 0), and the length of CRC frame set to less than 8 bits (CRCSIZE[4:0] < 00111), the last data received in the RxFIFO may be corrupted.

Workaround

Keep the CRCSIZE[4:0] bitfield at its default setting (00111) during the data reception if CRCEN = 0 and TSIZE[15:0] > 0.

2.18.2 MODF flag cannot generate interrupt

Description

Mode fault detection results in disabling SPI. With the MODFIE bit of the SPI_IER register set, the mode fault flag (MODF) going high is expected to trigger an interrupt. However, disabling SPI unduly blocks this interrupt request.

Workaround

To detect a mode fault event, poll the MODF flag by software.

2.18.3 RDY output failure at high serial clock frequency

Description

When acting as slave with RDY alternate function enabled through setting the RDIOM bit of the SPI_CFG2 register, the device may fail to indicate its *Not ready* status in time through the RDY output signal to suspend communication. This may then lead to data overrun and/or underrun on the device side. The failure occurs when the serial clock frequency exceeds:

- twice the APB clock frequency, with data sizes from 8 to 15 bits
- six times the APB clock frequency, with data sizes from 16 to 23 bits
- fourteen times the APB clock frequency, with data sizes from 24 to 32 bits

Workaround

None.

2.18.4 Master communication suspension fails in Autonomous mode

Description

The SPI peripheral is blocked regardless of the completion of the ongoing data frame transaction, and the SUPSF flag is never set, when:

- the master provides a communication triggered in Autonomous mode (TRIGEN=1 of the SPI_AUTOCR register), and
- the suspension of the ongoing transaction is applied by setting the CSUSP bit through the smart DMA in Stop mode.

Workaround

None.

Note: The user software must avoid any master suspension in Stop mode while the master operates in Autonomous mode and waits for EOT if TSIZE is greater than 0. If an endless transaction is applied (TSIZE = 0), the suspension is the only way to stop the ongoing transaction. Then to unblock the peripheral, the software must disable SPI then apply the hardware reset. Otherwise, the system cannot proceed to the next transaction.

2.18.5 SPE may not be cleared upon MODF event

Description

The failure described applies to multi-master topology when the device is configured to monitor the SS input signal by hardware (SSM = 0, SSOE = 0 of the SPI_CFG2 register).

If the software sets the SPE (SPI enable) bit of the SPI_CR1 register at the instant of the SS signal transiting to its active logical level, the resulting MODF event duly switches the SPI into slave mode, but it fails to clear the SPE bit and thus disable the SPI.

Note: The SS active logical level is the one that matches the SSIOP bit of the SPI_CFG2 register.

Workaround

Whenever MODF event fails to clear the SPE bit, do it by software.

2.18.6 SPI slave stalls with masters not providing extra SCK periods upon *Not ready* signalling

Description

In Stop mode, the device SPI operating as slave with the *Ready* signalling enabled (the RDIOM of the SPI_CFG2 register set) may stall and never retrieve the *Ready* state. This occurs when SCK stops immediately after *Not ready* status.

Note: STM32 devices supporting the Ready signaling and operating as SPI master provide some extra SCK periods upon detecting Not ready signal, thus allowing the SPI slaves to operate correctly.

Workaround

If in the application, there is an SPI master that stops SCK immediately upon *Not ready* signal, without providing some extra SCK periods, do not enable the *Ready* signalling.

2.18.7 Truncation of SPI output signals after EOT event

Description

After an EOT event signaling the end of a non-zero transfer size transaction (TSIZE > 0) upon sampling the last data bit, the software may disable the SPI peripheral. As expected, disabling SPI deactivates the SPI outputs (SCK, MOSI and SS when the SPI operates as a master, MISO when as a slave), by making them float or statically output their by-default levels, according to the AFCNTR bit of the SPI_CFG2 register.

With fast software execution (high PCLK frequency) and slow SPI (low SCK frequency), the SPI disable occurring too fast may result in truncating the SPI output signals. For example, the device operating as a master then generates an asymmetric last SCK pulse (with CPHA = 0), which may prevent the correct last data bit reception by the other node involved in the communication.

Workaround

Apply one of the following measures or their combination:

- Add a delay between the EOT event and SPI disable action.
- Decrease the ratio between PCLK and SCK frequencies.

2.19 SDMMC

2.19.1 Command response and receive data end bits not checked

Description

The command response and receive data end bits are not checked by the SDMMC. A reception with only a wrong end bit value is not detected. This does not cause a communication failure since the received command response or data is correct.

Workaround

None.

2.20 FDCAN

2.20.1 Desynchronization under specific condition with edge filtering enabled

Description

FDCAN may desynchronize and incorrectly receive the first bit of the frame if:

- the edge filtering is enabled (the EFBI bit of the FDCAN_CCCR register is set), and
- the end of the integration phase coincides with a falling edge detected on the FDCAN_Rx input pin

If this occurs, the CRC detects that the first bit of the received frame is incorrect, flags the received frame as faulty and responds with an error frame.

Note: This issue does not affect the reception of standard frames.

Workaround

Disable edge filtering or wait for frame retransmission.

2.20.2 Tx FIFO messages inverted under specific buffer usage and priority setting

Description

Two consecutive messages from the Tx FIFO may be inverted in the transmit sequence if:

- FDCAN uses both a dedicated Tx buffer and a Tx FIFO (the TFQM bit of the FDCAN_TXBC register is cleared), and
- the messages contained in the Tx buffer have a higher internal CAN priority than the messages in the Tx FIFO.

Workaround

Apply one of the following measures:

- Ensure that only one Tx FIFO element is pending for transmission at any time:
  The Tx FIFO elements may be filled at any time with messages to be transmitted, but their transmission requests are handled separately. Each time a Tx FIFO transmission has completed and the Tx FIFO gets empty (TFE bit of FDACN_IR set to 1) the next Tx FIFO element is requested.
- Use only a Tx FIFO:
  Send both messages from a Tx FIFO, including the message with the higher priority. This message has to wait until the preceding messages in the Tx FIFO have been sent.
- Use two dedicated Tx buffers (for example, use Tx buffer 4 and 5 instead of the Tx FIFO). The following pseudo-code replaces the function in charge of filling the Tx FIFO:

```
Write message to Tx Buffer 4
Transmit Loop:
    Request Tx Buffer 4 - write AR4 bit in FDCAN_TXBAR
    Write message to Tx Buffer 5
    Wait until transmission of Tx Buffer 4 complete (IR bit in FDCAN_IR),
    read TO4 bit in FDCAN_TXBTO
    Request Tx Buffer 5 - write AR5 bit of FDCAN_TXBAR
    Write message to Tx Buffer 4
    Wait until transmission of Tx Buffer 5 complete (IR bit in FDCAN_IR),
    read TO5 bit in FDCAN_TXBTO
```

2.21 USB

2.21.1 Buffer description table update completes after CTR interrupt triggers

Description

During OUT transfers, the correct transfer interrupt (CTR) is triggered a little before the last USB SRAM accesses have completed. If the software responds quickly to the interrupt, the full buffer contents may not be correct.

Workaround

Software should ensure that a small delay is included before accessing the SRAM contents. This delay should be 800 ns in Full Speed mode and 6.4 µs in Low Speed mode.

Important security notice

The STMicroelectronics group of companies (ST) places a high value on product security, which is why the ST product(s) identified in this documentation may be certified by various security certification bodies and/or may implement our own security measures as set forth herein. However, no level of security certification and/or built-in security measures can guarantee that ST products are resistant to all forms of attacks. As such, it is the responsibility of each of ST's customers to determine if the level of security provided in an ST product meets the customer needs both in relation to the ST product alone, as well as when combined with other components and/or software for the customer end product or application. In particular, take note that:

- ST products may have been certified by one or more security certification bodies, such as Platform Security Architecture (www.psacertified.org) and/or Security Evaluation standard for IoT Platforms (www.trustcb.com). For details concerning whether the ST product(s) referenced herein have received security certification along with the level and current status of such certification, either visit the relevant certification standards website or go to the relevant product page on www.st.com for the most up to date information. As the status and/or level of security certification for an ST product can change from time to time, customers should re-check security certification status/level as needed. If an ST product is not shown to be certified under a particular security standard, customers should not assume it is certified.
- Certification bodies have the right to evaluate, grant and revoke security certification in relation to ST products. These certification bodies are therefore independently responsible for granting or revoking security certification for an ST product, and ST does not take any responsibility for mistakes, evaluations, assessments, testing, or other activity carried out by the certification body with respect to any ST product.
- Industry-based cryptographic algorithms (such as AES, DES, or MD5) and other open standard technologies which may be used in conjunction with an ST product are based on standards which were not developed by ST. ST does not take responsibility for any flaws in such cryptographic algorithms or open technologies or for any methods which have been or may be developed to bypass, decrypt or crack such algorithms or technologies.
- While robust security testing may be done, no level of certification can absolutely guarantee protections against all attacks, including, for example, against advanced attacks which have not been tested for, against new or unidentified forms of attack, or against any form of attack when using an ST product outside of its specification or intended use, or in conjunction with other components or software which are used by customer to create their end product or application. ST is not responsible for resistance against such attacks. As such, regardless of the incorporated security features and/or any information or support that may be provided by ST, each customer is solely responsible for determining if the level of attacks tested for meets their needs, both in relation to the ST product alone and when incorporated into a customer end product or application.
- All security features of ST products (inclusive of any hardware, software, documentation, and the like), including but not limited to any enhanced security features added by ST, are provided on an "AS IS" BASIS. AS SUCH, TO THE EXTENT PERMITTED BY APPLICABLE LAW, ST DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, unless the applicable written and signed contract terms specifically provide otherwise.

Revision history

Table 7. **Document revision history**

Date	Version	Changes
16-Feb-2023	1	Initial release
28-Jun-2023	2	Added errata: • **System:** Incorrect backup domain reset with VBAT and VDD supplied by different power sources • EXTI LOCK bit does not lock privilege configuration • Device may be locked upon system reset under Stop 2 mode • SRAM ECC error flags and addresses are updated only if interrupt is enabled • The ADC I/Os analog switch voltage booster is not functional • **OCTOSPI:** At least six cycles memory latency must be set when DQS is used for HyperBus™ memories • Data write discarded in memory-mapped mode if a write to a misaligned address is directly followed by a request to the same address • **ADC1:** 14-bit ADC offset error and integral linearity error values are increased in datasheets • **USART:** Wrong data received by SPI slave receiver in Autonomous mode with CPOL = 1 • Received data may be corrupted upon clearing the ABREN bit • Noise error flag set while ONEBIT is set Renamed LPUART limitation: Wrong data received when the communication nodes are two LPUART instances into Possible LPUART transmitter issue when using low BRR[15:0] value Removed errata not applicable for this product: • **ADC1:** Unexpected regular conversion when two consecutive injected conversions are performed in Dual interleaved mode • Channel selection not reset in Bulb mode when regular conversions are stopped by DMA or ADSTP • ADC slave data may be shifted in Dual regular simultaneous mode • **TSC:** Inhibited acquisition in short transfer phase configuration • **TIM:** Consecutive compare event missed in specific conditions • Output compare clear not working with external counter reset
21-Dec-2023	3	Added errata: • **System:** Flash programming can remain stuck in case of programming sequence error • VDDUSB is internally connected to VDD in WLCSP56 packages • **OCTOSPI:** Setting the ABORT bit does not generate an error on the AHB bus for undefined-length incremental burst transfers • **SAES:** Data transfer from TAMP_BKPxR to key registers must be done only in ascending order when KEYSEL[2:0] is set to 010 or 100 • **USB:** Buffer description table update completes after CTR interrupt triggers Modified errata: • **System:** LSE crystal oscillator may be disturbed by transitions on PC13 • **OCTOSPI:** Read data corruption after a few bytes are skipped when crossing a four-byte boundary • **USART:** Data corruption due to noisy receive line Removed errata not applicable for this product: • **OCTOSPI:** Received data corrupted after arbitration ownership toggles when using clock mode 3 and no DQS for read direction • Deadlock can occur under specific conditions • **USB:** False wakeup detection for last K-state not terminated by EOP or reset before suspend • ESOF interrupt timing desynchronized after resume signaling • Incorrect CRC16 in the memory buffer • Possible packet memory overrun/underrun at low APB frequency
4-Apr-2024	4	Added Rev. Y.

Date	Version	Changes
		Added errata: • Read data corruption when a wrap transaction is followed by a linear read to the same MSB address • Transactions are limited to 8 Mbytes in OctaRAM™ memories • Command response and receive data end bits not checked • TSC I/Os analog switches are not functional when supplied by V_{DDA} • COMP I/Os analog switches are not functional when supplied by V_{DDA} • OPAMP I/Os analog switches are not functional when supplied by V_{DDA} • PVD_IN I/Os analog switches are not functional when supplied by V_{DDA} Modified errata: ADC I/Os analog switches are not functional when supplied by V_{DDA}

Contents