STM32F405/407xx and STM32F415/417xx device errata

Applicability

This document applies to the part numbers of STM32F405/407xx and STM32F415/417xx 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 RM0090.

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
STM32F405xx	STM32F405OG, STM32F405OE, STM32F405RG, STM32F405VG, STM32F405ZG
STM32F407xx	STM32F407IG, STM32F407VG, STM32F407ZG, STM32F407ZE, STM32F407IE, STM32F407VE
STM32F415xx	STM32F415OG, STM32F415RG, STM32F415VG, STM32F415ZG
STM32F417xx	STM32F417VG, STM32F417IG, STM32F417ZG, STM32F417VE, STM32F417ZE, STM32F417IE

Table 2. Device variants

Reference	Silicon revision codes	
	Device marking(1)	REV_ID(2)
STM32F405xx, STM32F407xx, STM32F415xx, STM32F417xx	A	0x1000
	Z	0x1001
	1	0x1003
	2	0x1007
	Y	0x100F
	4	
	5	0x101F
	6	

1. *Refer to the device datasheet for how to identify this code on different types of package.*
2. *REV_ID[15:0] bitfield of DBGMCU_IDCODE register.*

STM32F405/407xx and STM32F415/417xx

Errata sheet

STM32F405/407xx 和 STM32F415/417xx 设备勘误表

适用性

本文档适用于STM32F405/407xx和STM32F415/417xx设备的零件号以及如本页所述的设备型号变体。

它提供了设备勘误表的摘要和描述，参照设备数据手册和参考手册 RM0090。

实际设备行为与预期设备行为之间的偏差被视为设备限制。参考手册或数据手册中的描述与预期设备行为之间的偏差被视为文档错误。术语 *“errata”* 同时适用于设备限制和文档错误。

表 1. 设备摘要

Reference	Part numbers
STM32F405xx	STM32F405OG, STM32F405OE, STM32F405RG, STM32F405VG, STM32F405ZG
STM32F407xx	STM32F407IG, STM32F407VG, STM32F407ZG, STM32F407ZE, STM32F407IE, STM32F407VE
STM32F415xx	STM32F415OG, STM32F415RG, STM32F415VG, STM32F415ZG
STM32F417xx	STM32F417VG, STM32F417IG, STM32F417ZG, STM32F417VE, STM32F417ZE, STM32F417IE

表2. 设备变体

Reference	Silicon revision codes	
	Device marking(1)	REV_ID(2)
STM32F405xx, STM32F407xx, STM32F415xx, STM32F417xx	A	0x1000
	Z	0x1001
	1	0x1003
	2	0x1007
	Y	0x100F
	4	
	5	0x101F
	6	

1. *Refer to the device datasheet for how to identify this code on different types of package.*
2. *REV_ID[15:0] bitfield of DBGMCU_IDCODE register.*

1 Summary of device errata

The following table gives a quick reference to the STM32F405/407xx and STM32F415/417xx 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. A	Rev. Z	Rev. 1	Rev. 2	Rev. Y	Rev. 4	Rev. 5	Rev. 6
Core	2.1.1	Interrupted loads to SP can cause erroneous behavior	A	A	A	A	A	A	A	A
	2.1.2	VDIV or VSQRT instructions might not complete correctly when very short ISRs are used	A	A	A	A	A	A	A	A
	2.1.3	Store immediate overlapping exception return operation might vector to incorrect interrupt	A	A	A	A	A	A	A	A
System	2.2.1	ART Accelerator prefetch queue instruction is not supported	N	-	-	-	-	-	-	-
	2.2.2	MCU device ID is incorrect	A	-	-	-	-	-	-	-
	2.2.3	Debugging Stop mode and SysTick timer	A	A	A	A	A	A	A	A
	2.2.4	Debugging Stop mode with WFE entry	A	A	A	A	A	A	A	A
	2.2.5	Debugging Sleep/Stop mode with WFE/WFI entry	A	A	A	A	A	A	A	A
	2.2.6	Wake-up sequence from Standby mode when using more than one wake-up source	A	A	A	A	A	A	A	A
	2.2.7	Full JTAG configuration without NJTRST pin cannot be used	A	A	A	A	A	A	A	A
	2.2.8	PDR_ON pin not available on LQFP100 package except for revision A devices	-	N	N	N	N	N	N	N
	2.2.9	Incorrect BOR option byte when consecutively programming BOR option byte	A	A	A	A	A	A	A	A
	2.2.10	Configuration of PH10 and PI10 as external interrupts is erroneous	N	N	N	N	N	N	N	N
	2.2.11	Slowing down APB clock during a DMA transfer	A	A	A	A	A	A	A	A
	2.2.12	MPU attribute to RTC and IWDG registers incorrectly managed	A	A	A	A	A	A	A	A
	2.2.13	Delay after an RCC peripheral clock enabling	A	A	A	A	A	A	A	A
	2.2.14	Battery charge monitoring lower than 2.4 V	P	P	P	P	P	P	P	P
	2.2.15	Internal noise impacting the ADC accuracy	A	A	A	A	A	A	A	A
	2.2.16	RDP level 2 and sector write protection configuration	A	A	A	A	A	A	A	A

1 设备勘误汇总

下表提供了STM32F405/407xx和STM32F415/417xx设备的限制条件及其状态的快速参考:

存在=限制，有解决方法
存在N=限制，无可用变通方案
P = 存在限制，有部分解决方法
“-”= 无限制
变通方法的适用性可能取决于目标应用程序的特定条件。采用变通方法可能导致对目标应用程序的限制。如果变通方法仅减少限制的发生率和/或后果，或仅在设备上的部分实例或特定操作模式下对相关功能完全有效，则该变通方法被视为部分有效。

表 3. 设备限制摘要

Function	Section	Limitation	Status							
			Rev. A	Rev. Z	Rev. 1	Rev. 2	Rev. Y	Rev. 4	Rev. 5	Rev. 6
Core	2.1.1	Interrupted loads to SP can cause erroneous behavior	A	A	A	A	A	A	A	A
	2.1.2	VDIV or VSQRT instructions might not complete correctly when very short ISRs are used	A	A	A	A	A	A	A	A
	2.1.3	Store immediate overlapping exception return operation might vector to incorrect interrupt	A	A	A	A	A	A	A	A
System	2.2.1	ART Accelerator prefetch queue instruction is not supported	N	-	-	-	-	-	-	-
	2.2.2	MCU device ID is incorrect	A	-	-	-	-	-	-	-
	2.2.3	Debugging Stop mode and SysTick timer	A	A	A	A	A	A	A	A
	2.2.4	Debugging Stop mode with WFE entry	A	A	A	A	A	A	A	A
	2.2.5	Debugging Sleep/Stop mode with WFE/WFI entry	A	A	A	A	A	A	A	A
	2.2.6	Wake-up sequence from Standby mode when using more than one wake-up source	A	A	A	A	A	A	A	A
	2.2.7	Full JTAG configuration without NJTRST pin cannot be used	A	A	A	A	A	A	A	A
	2.2.8	PDR_ON pin not available on LQFP100 package except for revision A devices	-	N	N	N	N	N	N	N
	2.2.9	Incorrect BOR option byte when consecutively programming BOR option byte	A	A	A	A	A	A	A	A
	2.2.10	Configuration of PH10 and PI10 as external interrupts is erroneous	N	N	N	N	N	N	N	N
	2.2.11	Slowing down APB clock during a DMA transfer	A	A	A	A	A	A	A	A
	2.2.12	MPU attribute to RTC and IWDG registers incorrectly managed	A	A	A	A	A	A	A	A
	2.2.13	Delay after an RCC peripheral clock enabling	A	A	A	A	A	A	A	A
	2.2.14	Battery charge monitoring lower than 2.4 V	P	P	P	P	P	P	P	P
	2.2.15	Internal noise impacting the ADC accuracy	A	A	A	A	A	A	A	A
	2.2.16	RDP level 2 and sector write protection configuration	A	A	A	A	A	A	A	A

Function	Section	Limitation	Status							
			Rev. A	Rev. Z	Rev. 1	Rev. 2	Rev. Y	Rev. 4	Rev. 5	Rev. 6
System	2.2.17	Possible delay in backup domain protection disabling/enabling after programming the DBP bit	A	A	A	A	A	A	A	A
	2.2.18	PC13 signal transitions disturb LSE	N	N	N	N	N	N	N	N
	2.2.19	In some specific cases, DMA2 data corruption occurs when managing AHB and APB2 peripherals in a concurrent way	A	A	A	A	A	A	A	A
	2.2.20	Corrupted content of the backup domain due to a missed power-on reset after this domain supply voltage drop	A	A	A	A	A	A	A	A
FSMC	2.3.1	Dummy read cycles inserted when reading synchronous memories	N	N	N	N	N	N	N	N
	2.3.2	FSMC synchronous mode and NWAIT signal disabled	A	A	A	A	A	A	A	A
	2.3.3	FSMC NOR/PSRAM controller: asynchronous read access on bank 2 to 4 returns wrong data when bank 1 is in synchronous mode (BURSTEN bit is set)	A	A	A	A	A	A	A	A
SDIO	2.4.1	Wrong data written during SDIO hardware flow control	N	N	N	N	N	N	N	N
	2.4.2	Wrong CCRCFAIL status after a response without CRC is received	A	A	A	A	A	A	A	A
	2.4.3	SDIO clock divider Bypass mode may not work properly	A	A	A	A	A	A	A	A
	2.4.4	Data corruption in SDIO clock dephasing (NEGEDGE) mode	N	N	N	N	N	N	N	N
	2.4.5	CE-ATA multiple write command and card busy signal management	A	A	A	A	A	A	A	A
	2.4.6	No underrun detection with wrong data transmission	A	A	A	A	A	A	A	A
ADC	2.5.1	ADC sequencer modification during conversion	A	A	A	A	A	A	A	A
DAC	2.6.1	DMA request not automatically cleared by clearing DMAEN	A	A	A	A	A	A	A	A
	2.6.2	DMA underrun flag not set when an internal trigger is detected on the clock cycle of the DMA request acknowledge	N	N	N	N	N	N	N	N
TIM	2.7.1	PWM re-enabled in automatic output enable mode despite of system break	P	P	P	P	P	P	P	P
	2.7.3	Consecutive compare event missed in specific conditions	N	N	N	N	N	N	N	N
	2.7.4	Output compare clear not working with external counter reset	P	P	P	P	P	P	P	P
IWDG	2.8.1	RVU flag not reset in Stop	A	A	A	A	A	A	A	A
	2.8.2	PVU flag not reset in Stop	A	A	A	A	A	A	A	A
	2.8.3	RVU flag not cleared at low APB clock frequency	A	A	A	A	A	A	A	A
	2.8.4	PVU flag not cleared at low APB clock frequency	A	A	A	A	A	A	A	A
RTC	2.9.1	Spurious tamper detection when disabling the tamper channel	N	N	N	N	N	N	N	N

Function	Section	Limitation	Status							
			Rev. A	Rev. Z	Rev. 1	Rev. 2	Rev. Y	Rev. 4	Rev. 5	Rev. 6
System	2.2.17	Possible delay in backup domain protection disabling/enabling after programming the DBP bit	A	A	A	A	A	A	A	A
	2.2.18	PC13 signal transitions disturb LSE	N	N	N	N	N	N	N	N
	2.2.19	In some specific cases, DMA2 data corruption occurs when managing AHB and APB2 peripherals in a concurrent way	A	A	A	A	A	A	A	A
	2.2.20	Corrupted content of the backup domain due to a missed power-on reset after this domain supply voltage drop	A	A	A	A	A	A	A	A
FSMC	2.3.1	Dummy read cycles inserted when reading synchronous memories	N	N	N	N	N	N	N	N
	2.3.2	FSMC synchronous mode and NWAIT signal disabled	A	A	A	A	A	A	A	A
	2.3.3	FSMC NOR/PSRAM controller: asynchronous read access on bank 2 to 4 returns wrong data when bank 1 is in synchronous mode (BURSTEN bit is set)	A	A	A	A	A	A	A	A
SDIO	2.4.1	Wrong data written during SDIO hardware flow control	N	N	N	N	N	N	N	N
	2.4.2	Wrong CCRCFAIL status after a response without CRC is received	A	A	A	A	A	A	A	A
	2.4.3	SDIO clock divider Bypass mode may not work properly	A	A	A	A	A	A	A	A
	2.4.4	Data corruption in SDIO clock dephasing (NEGEDGE) mode	N	N	N	N	N	N	N	N
	2.4.5	CE-ATA multiple write command and card busy signal management	A	A	A	A	A	A	A	A
	2.4.6	No underrun detection with wrong data transmission	A	A	A	A	A	A	A	A
ADC	2.5.1	ADC sequencer modification during conversion	A	A	A	A	A	A	A	A
DAC	2.6.1	DMA request not automatically cleared by clearing DMAEN	A	A	A	A	A	A	A	A
	2.6.2	DMA underrun flag not set when an internal trigger is detected on the clock cycle of the DMA request acknowledge	N	N	N	N	N	N	N	N
TIM	2.7.1	PWM re-enabled in automatic output enable mode despite of system break	P	P	P	P	P	P	P	P
	2.7.3	Consecutive compare event missed in specific conditions	N	N	N	N	N	N	N	N
	2.7.4	Output compare clear not working with external counter reset	P	P	P	P	P	P	P	P
IWDG	2.8.1	RVU flag not reset in Stop	A	A	A	A	A	A	A	A
	2.8.2	PVU flag not reset in Stop	A	A	A	A	A	A	A	A
	2.8.3	RVU flag not cleared at low APB clock frequency	A	A	A	A	A	A	A	A
	2.8.4	PVU flag not cleared at low APB clock frequency	A	A	A	A	A	A	A	A
RTC	2.9.1	Spurious tamper detection when disabling the tamper channel	N	N	N	N	N	N	N	N

Function	Section	Limitation	Status							
			Rev. A	Rev. Z	Rev. 1	Rev. 2	Rev. Y	Rev. 4	Rev. 5	Rev. 6
RTC	2.9.2	RTC calendar registers are not locked properly	A	A	A	A	A	A	A	A
	2.9.3	RTC interrupt can be masked by another RTC interrupt	A	A	A	A	A	A	A	A
	2.9.4	Calendar initialization may fail in case of consecutive INIT mode entry	A	A	A	A	A	A	A	A
	2.9.5	Alarm flag may be repeatedly set when the core is stopped in debug	N	N	N	N	N	N	N	N
	2.9.6	Detection of a tamper event occurring before enabling the tamper detection is not supported in edge detection mode	A	A	A	A	A	A	A	A
I2C	2.10.1	Spurious bus error detection in controller mode	A	A	A	A	A	A	A	A
	2.10.2	SMBus standard not fully supported	A	A	A	A	A	A	A	A
	2.10.3	Start cannot be generated after a misplaced Stop	A	A	A	A	A	A	A	A
	2.10.4	Mismatch on the “Setup time for a repeated Start condition” timing parameter	A	A	A	A	A	A	A	A
	2.10.5	Data valid time ($t_{VD;DAT}$) violated without the OVR flag being set	A	A	A	A	A	A	A	A
	2.10.6	Both SDA and SCL maximum rise times (t_r) violated when the VDD_I2C bus voltage is higher than ((V_{DD} + 0.3) / 0.7) V	A	A	A	A	A	A	A	A
USART	2.11.1	Idle frame is not detected if the receiver clock speed is deviated	N	N	N	N	N	N	N	N
	2.11.2	In full-duplex mode, the Parity Error (PE) flag can be cleared by writing to the data register	A	A	A	A	A	A	A	A
	2.11.3	Parity Error (PE) flag is not set when receiving in Mute mode using address mark detection	N	N	N	N	N	N	N	N
	2.11.4	Break frame is transmitted regardless of CTS input line status	N	N	N	N	N	N	N	N
	2.11.5	RTS signal abnormally driven low after a protocol violation	A	A	A	A	A	A	A	A
	2.11.6	Start bit detected too soon when sampling for NACK signal from the smartcard	N	N	N	N	N	N	N	N
	2.11.7	Break request can prevent the transmission complete flag (TC) from being set	A	A	A	A	A	A	A	A
	2.11.8	Guard time not respected when data are sent on TXE events	A	A	A	A	A	A	A	A
	2.11.9	RTS is active while RE or UE = 0	A	A	A	A	A	A	A	A
SPI/I2S	2.12.1	BSY bit may stay high when SPI is disabled	A	A	A	A	A	A	A	A
	2.12.2	Anticipated communication upon SPI transit from slave receiver to master	A	A	A	A	A	A	A	A
	2.12.3	Wrong CRC calculation when the polynomial is even	A	A	A	A	A	A	A	A
	2.12.4	Corrupted last bit of data and/or CRC received in Master mode with delayed SCK feedback	A	A	A	A	A	A	A	A
	2.12.5	BSY flag may stay high at the end of a data transfer in Slave mode	A	A	A	A	A	A	A	A

Function	Section	Limitation	Status							
			Rev. A	Rev. Z	Rev. 1	Rev. 2	Rev. Y	Rev. 4	Rev. 5	Rev. 6
RTC	2.9.2	RTC calendar registers are not locked properly	A	A	A	A	A	A	A	A
	2.9.3	RTC interrupt can be masked by another RTC interrupt	A	A	A	A	A	A	A	A
	2.9.4	Calendar initialization may fail in case of consecutive INIT mode entry	A	A	A	A	A	A	A	A
	2.9.5	Alarm flag may be repeatedly set when the core is stopped in debug	N	N	N	N	N	N	N	N
	2.9.6	Detection of a tamper event occurring before enabling the tamper detection is not supported in edge detection mode	A	A	A	A	A	A	A	A
I2C	2.10.1	Spurious bus error detection in controller mode	A	A	A	A	A	A	A	A
	2.10.2	SMBus standard not fully supported	A	A	A	A	A	A	A	A
	2.10.3	Start cannot be generated after a misplaced Stop	A	A	A	A	A	A	A	A
	2.10.4	Mismatch on the “Setup time for a repeated Start condition” timing parameter	A	A	A	A	A	A	A	A
	2.10.5	Data valid time ($t_{VD;DAT}$) violated without the OVR flag being set	A	A	A	A	A	A	A	A
	2.10.6	Both SDA and SCL maximum rise times (t_r) violated when the VDD_I2C bus voltage is higher than ((V_{DD} + 0.3) / 0.7) V	A	A	A	A	A	A	A	A
USART	2.11.1	Idle frame is not detected if the receiver clock speed is deviated	N	N	N	N	N	N	N	N
	2.11.2	In full-duplex mode, the Parity Error (PE) flag can be cleared by writing to the data register	A	A	A	A	A	A	A	A
	2.11.3	Parity Error (PE) flag is not set when receiving in Mute mode using address mark detection	N	N	N	N	N	N	N	N
	2.11.4	Break frame is transmitted regardless of CTS input line status	N	N	N	N	N	N	N	N
	2.11.5	RTS signal abnormally driven low after a protocol violation	A	A	A	A	A	A	A	A
	2.11.6	Start bit detected too soon when sampling for NACK signal from the smartcard	N	N	N	N	N	N	N	N
	2.11.7	Break request can prevent the transmission complete flag (TC) from being set	A	A	A	A	A	A	A	A
	2.11.8	Guard time not respected when data are sent on TXE events	A	A	A	A	A	A	A	A
	2.11.9	RTS is active while RE or UE = 0	A	A	A	A	A	A	A	A
SPI/I2S	2.12.1	BSY bit may stay high when SPI is disabled	A	A	A	A	A	A	A	A
	2.12.2	Anticipated communication upon SPI transit from slave receiver to master	A	A	A	A	A	A	A	A
	2.12.3	Wrong CRC calculation when the polynomial is even	A	A	A	A	A	A	A	A
	2.12.4	Corrupted last bit of data and/or CRC received in Master mode with delayed SCK feedback	A	A	A	A	A	A	A	A
	2.12.5	BSY flag may stay high at the end of a data transfer in Slave mode	A	A	A	A	A	A	A	A

Function	Section	Limitation	Status							
			Rev. A	Rev. Z	Rev. 1	Rev. 2	Rev. Y	Rev. 4	Rev. 5	Rev. 6
SPI/I2S	2.12.6	In I2S Slave mode, the WS level must be set by the external master when enabling the I2S	A	A	A	A	A	A	A	A
	2.12.7	I2S2 in full-duplex mode may not work properly when SCK and WS signals are mapped on PI1 and PI0, respectively	A	A	A	A	A	A	A	A
bxCAN	2.13.1	bxCAN time-triggered communication mode not supported	N	N	N	N	N	N	N	N
OTG_FS	2.14.1	Transmit data FIFO is corrupted when a write sequence to the FIFO is interrupted with accesses to certain OTG_FS registers	A	A	A	A	A	A	A	A
	2.14.2	Host packet transmission may hang when connecting through a hub to a low-speed device	N	N	N	N	N	N	N	N
	2.14.3	Data in RxFIFO is overwritten when all channels are disabled simultaneously	A	A	A	A	A	A	A	A
	2.14.4	OTG host blocks the receive channel when receiving IN packets and no TxFIFO is configured	A	A	A	A	A	A	A	A
	2.14.5	Host channel-halted interrupt not generated when the channel is disabled	A	A	A	A	A	A	A	A
	2.14.6	Wrong software-read OTG_FS_DCFG register values	A	A	A	A	A	A	A	A
OTG_HS	2.15.1	Transmit data FIFO is corrupted when a write sequence to the FIFO is interrupted with accesses to certain OTG_HS registers	A	A	A	A	A	A	A	A
	2.15.2	Host packet transmission may hang when connecting the full speed interface through a hub to a low-speed device	N	N	N	N	N	N	N	N
ETH	2.16.1	Incorrect L3 checksum is inserted in transmitted IPv6 packets without TCP, UDP or ICMP payloads	A	A	A	A	A	A	A	A
	2.16.2	The Ethernet MAC processes invalid extension headers in the received IPv6 frames	N	N	N	N	N	N	N	N
	2.16.3	MAC stuck in the idle state on receiving the TxFIFO flush command exactly one clock cycle after a transmission completes	P	P	P	P	P	P	P	P
	2.16.4	Transmit frame data corruption	A	A	A	A	A	A	A	A
	2.16.5	Incorrect status and corrupted frames when RxFIFO overflow occurs on the penultimate word of Rx frames	A	A	A	A	A	A	A	A
	2.16.6	Successive write operations to the same register might not be fully taken into account	A	A	A	A	A	A	A	A
	2.16.7	Incorrect remote wakeup on global unicast packet	P	P	P	P	P	P	P	P
	2.16.8	Overflow status bits of missed frame and buffer overflow counters are cleared without a read operation	A	A	A	A	A	A	A	A
	2.16.9	MAC may provide incorrect Rx status for the MAC control frames when receive checksum offload is enabled	A	A	A	A	A	A	A	A
	2.16.10	MAC may provide an inaccurate Rx status when receive checksum offload is enabled in cut-through mode	P	P	P	P	P	P	P	P

Function	Section	Limitation	Status							
			Rev. A	Rev. Z	Rev. 1	Rev. 2	Rev. Y	Rev. 4	Rev. 5	Rev. 6
SPI/I2S	2.12.6	In I2S Slave mode, the WS level must be set by the external master when enabling the I2S	A	A	A	A	A	A	A	A
	2.12.7	I2S2 in full-duplex mode may not work properly when SCK and WS signals are mapped on PI1 and PI0, respectively	A	A	A	A	A	A	A	A
bxCAN	2.13.1	bxCAN time-triggered communication mode not supported	N	N	N	N	N	N	N	N
OTG_FS	2.14.1	Transmit data FIFO is corrupted when a write sequence to the FIFO is interrupted with accesses to certain OTG_FS registers	A	A	A	A	A	A	A	A
	2.14.2	Host packet transmission may hang when connecting through a hub to a low-speed device	N	N	N	N	N	N	N	N
	2.14.3	Data in RxFIFO is overwritten when all channels are disabled simultaneously	A	A	A	A	A	A	A	A
	2.14.4	OTG host blocks the receive channel when receiving IN packets and no TxFIFO is configured	A	A	A	A	A	A	A	A
	2.14.5	Host channel-halted interrupt not generated when the channel is disabled	A	A	A	A	A	A	A	A
	2.14.6	Wrong software-read OTG_FS_DCFG register values	A	A	A	A	A	A	A	A
OTG_HS	2.15.1	Transmit data FIFO is corrupted when a write sequence to the FIFO is interrupted with accesses to certain OTG_HS registers	A	A	A	A	A	A	A	A
	2.15.2	Host packet transmission may hang when connecting the full speed interface through a hub to a low-speed device	N	N	N	N	N	N	N	N
ETH	2.16.1	Incorrect L3 checksum is inserted in transmitted IPv6 packets without TCP, UDP or ICMP payloads	A	A	A	A	A	A	A	A
	2.16.2	The Ethernet MAC processes invalid extension headers in the received IPv6 frames	N	N	N	N	N	N	N	N
	2.16.3	MAC stuck in the idle state on receiving the TxFIFO flush command exactly one clock cycle after a transmission completes	P	P	P	P	P	P	P	P
	2.16.4	Transmit frame data corruption	A	A	A	A	A	A	A	A
	2.16.5	Incorrect status and corrupted frames when RxFIFO overflow occurs on the penultimate word of Rx frames	A	A	A	A	A	A	A	A
	2.16.6	Successive write operations to the same register might not be fully taken into account	A	A	A	A	A	A	A	A
	2.16.7	Incorrect remote wakeup on global unicast packet	P	P	P	P	P	P	P	P
	2.16.8	Overflow status bits of missed frame and buffer overflow counters are cleared without a read operation	A	A	A	A	A	A	A	A
	2.16.9	MAC may provide incorrect Rx status for the MAC control frames when receive checksum offload is enabled	A	A	A	A	A	A	A	A
	2.16.10	MAC may provide an inaccurate Rx status when receive checksum offload is enabled in cut-through mode	P	P	P	P	P	P	P	P

Function	Section	Limitation	Status							
			Rev. A	Rev. Z	Rev. 1	Rev. 2	Rev. Y	Rev. 4	Rev. 5	Rev. 6
ETH	2.16.11	MAC may not drop received giant error frames	A	A	A	A	A	A	A	A

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

Table 4. Summary of device documentation errata

Function	Section	Documentation erratum
TIM	2.7.2	TRGO and TRGO2 trigger output failure

Function	Section	Limitation	Status							
			Rev. A	Rev. Z	Rev. 1	Rev. 2	Rev. Y	Rev. 4	Rev. 5	Rev. 6
ETH	2.16.11	MAC may not drop received giant error frames	A	A	A	A	A	A	A	A

以下表格提供了对文档勘误的快速参考。

表4. 设备文档勘误摘要

Function	Section	Documentation erratum
TIM	2.7.2	TRGO and TRGO2 trigger output failure

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®-M4F core revision r0p1 is available from http://infocenter.arm.com. Only applicable information from the Arm errata notice is replicated in this document.

2.1.1 Interrupted loads to SP can cause erroneous behavior

This limitation is registered under Arm ID number 752770 and classified into “Category B”. Its impact to the device is minor.

Description

If an interrupt occurs during the data-phase of a single word load to the stack-pointer (SP/R13), erroneous behavior can occur. In all cases, returning from the interrupt will result in the load instruction being executed an additional time. For all instructions performing an update to the base register, the base register will be erroneously updated on each execution, resulting in the stack-pointer being loaded from an incorrect memory location.

The affected instructions that can result in the load transaction being repeated are:

- LDR SP, [Rn],#imm
- LDR SP, [Rn,#imm]!
- LDR SP, [Rn,#imm]
- LDR SP, [Rn]
- LDR SP, [Rn,Rm]

The affected instructions that can result in the stack-pointer being loaded from an incorrect memory address are:

- LDR SP,[Rn],#imm
- LDR SP,[Rn,#imm]!

As compilers do not generate these particular instructions, the limitation is only likely to occur with hand-written assembly code.

Workaround

Both issues may be worked around by replacing the direct load to the stack-pointer, with an intermediate load to a general-purpose register followed by a move to the stack-pointer.

2.1.2 VDIV or VSQRT instructions might not complete correctly when very short ISRs are used

This limitation is registered under Arm ID number 776924 and classified into “Category B”. Its impact to the device is limited.

Description

The VDIV and VSQRT instructions take 14 cycles to execute. When an interrupt is taken a VDIV or VSQRT instruction is not terminated, and completes its execution while the interrupt stacking occurs. If lazy context save of floating point state is enabled then the automatic stacking of the floating point context does not occur until a floating point instruction is executed inside the interrupt service routine.

Lazy context save is enabled by default. When it is enabled, the minimum time for the first instruction in the interrupt service routine to start executing is 12 cycles. In certain timing conditions, and if there is only one or two instructions inside the interrupt service routine, then the VDIV or VSQRT instruction might not write its result to the register bank or to the FPSCR.

2 设备错误描述

以下各节描述了具有 Arm® 核心的适用设备的勘误表，并在有可用的解决方法时提供变通方案。它们按设备功能分类。

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

二点一 核心{v*}

Arm® Cortex®-M4F核心版本r0p1的参考手册和勘误表可在http:// infocenter.arm.com获取。本文件中仅包含适用于Arm勘误表的信息。

2.1.1 中断到SP的加载可能导致错误行为

此限制登记在Arm ID编号752770下，并归类为“类别B”。其对设备的影响较小。

描述

如果在将单字加载到栈指针（SP/R13）的数据阶段发生中断，可能会出现错误行为。在所有情况下，从中断返回会导致加载指令被再次执行。对于所有对基址寄存器进行更新的指令，每次执行时基址寄存器都会被错误地更新，导致栈指针从错误的内存位置加载。可能导致加载事务被重复执行的受影响指令包括：

- 加载到寄存器SP，[Rn]，#imm
- LDR 堆栈指针, [Rn, #imm]!
- 加载到堆栈指针，[Rn,#imm]
- LDR 堆栈指针, [Rn]
- LDR SP, [Rn,Rm]

受影响的指令可能导致堆栈指针被从错误的内存地址加载，这些指令是：

- LDR 堆栈指针,[Rn],#立即数
- LDR 堆栈指针, [Rn, #imm]!

由于编译器不会生成这些特定指令，因此，这种局限性只会出现在手写汇编代码中。

变通方法

这两个问题可以通过将直接加载到堆栈指针的操作替换为中间加载到通用寄存器，然后将其值移动到堆栈指针来解决。

2.1.2 当使用非常简短的中断服务程序时，VDIV 或 VSQRT 指令可能无法正确完成

此限制已登记在Arm ID编号776924下，并被归类为“B类”。其对设备的影响是有限的。

描述

VDIV 和 VSQRT 指令需要 14 个周期来执行。当发生中断时，VDIV 或 VSQRT 指令不会被终止，并在中断堆栈发生的过程中完成其执行。如果启用了浮点状态的延迟上下文保存，则浮点上下文的自动堆栈不会在中断服务程序中执行浮点指令之前发生。

延迟上下文保存默认启用。当它启用时，中断服务例程中第一条指令开始执行的最短时间是12个周期。在特定的时序条件下，如果中断服务例程中仅有一条或两条指令，那么VDIV或VSQRT指令可能不会将结果写入寄存器组或FPSCR。

The failure occurs when the following condition is met:

1. The floating point unit is enabled
2. Lazy context saving is not disabled
3. A VDIV or VSQRT is executed
4. The destination register for the VDIV or VSQRT is one of s0 - s15
5. An interrupt occurs and is taken
6. The interrupt service routine being executed does not contain a floating point instruction
7. Within 14 cycles after the VDIV or VSQRT is executed, an interrupt return is executed

A minimum of 12 of these 14 cycles are utilized for the context state stacking, which leaves 2 cycles for instructions inside the interrupt service routine, or 2 wait states applied to the entire stacking sequence (which means that it is not a constant wait state for every access).

In general, this means that if the memory system inserts wait states for stack transactions (that is, external memory is used for stack data), then this erratum cannot be observed.

The effect of this erratum is that the VDIV or VQSRT instruction does not complete correctly and the register bank and FPSCR are not updated, which means that these registers hold incorrect, out of date, data.

Workaround

A workaround is only required if the floating point unit is enabled. A workaround is not required if the stack is in external memory.

There are two possible workarounds:

- Disable lazy context save of floating point state by clearing LSPEN to 0 (bit 30 of the FPCCR at address 0xE000EF34).
- Ensure that every interrupt service routine contains more than 2 instructions in addition to the exception return instruction.

2.1.3 Store immediate overlapping exception return operation might vector to incorrect interrupt

This limitation is registered under Arm ID number 838869 and classified into “Category B (rare)”. Its impact to the device is minor.

Description

The core includes a write buffer that permits execution to continue while a store is waiting on the bus. Under specific timing conditions, during an exception return while this buffer is still in use by a store instruction, a late change in selection of the next interrupt to be taken might result in there being a mismatch between the interrupt acknowledged by the interrupt controller and the vector fetched by the processor.

The failure occurs when the following condition is met:

1. The handler for interrupt A is being executed.
2. Interrupt B, of the same or lower priority than interrupt A, is pending.
3. A store with immediate offset instruction is executed to a bufferable location.
   - STR/STRH/STRB <Rt>, [<Rn>,#imm]
   - STR/STRH/STRB <Rt>, [<Rn>,#imm]!
   - STR/STRH/STRB <Rt>, [<Rn>],#imm
4. Any number of additional data-processing instructions can be executed.
5. A BX instruction is executed that causes an exception return.
6. The store data has wait states applied to it such that the data is accepted at least two cycles after the BX is executed.
   - Minimally, this is two cycles if the store and the BX instruction have no additional instructions between them.
   - The number of wait states required to observe this erratum needs to be increased by the number of cycles between the store and the interrupt service routine exit instruction.
7. Before the bus accepts the buffered store data, another interrupt C is asserted which has the same or lower priority as A, but a greater priority than B.

故障发生当以下条件满足时：

1. 浮点单元已启用 2. 延迟上下文保存未被禁用 3. 执行 VDIV 或 VSQRT 4. VDIV 或 VSQRT 的目标寄存器是 s0 到 s15 中的一个 5. 发生中断并被处理 6. 正在执行的中断服务例程不包含浮点指令 7. 在执行 VDIV 或 VSQRT 后的14个周期内执行中断返回

上下文状态堆栈至少使用了这14个周期中的12个，剩下2个周期用于中断服务程序中的指令，或者在整个堆栈序列中应用2个等待状态（这意味着它不是每次访问的固定等待状态）。

通常，这意味着如果内存系统为堆栈事务插入等待状态（即，外部内存用于堆栈数据），则此错误无法被观察到。

此次勘误的影响是，VDIV 或 VQSRT 指令无法正确完成，寄存器组和 FPSCR 未被更新，这意味着这些寄存器保存了错误且过时的数据。

变通方法

如果浮点单元已启用，则仅需采用变通方法。如果堆栈位于外部存储器中，则无需采用变通方法。

有两种可能的解决方法：

- 通过将LSPEN清除为0以禁用浮点状态的延迟上下文保存（FPCCR寄存器在地址0xE000EF34处的第30位）。
- 确保每个中断服务程序包含多于2条指令以及异常返回指令。

2.1.3 存储立即重叠异常返回操作可能会引导至错误中断

此限制已注册在Arm ID号838869下，并归类为“Category B (rare)”。其对设备的影响较小。

描述

核心包含一个写缓冲区，允许在存储操作等待总线时继续执行。在特定的时序条件下，当该缓冲区仍在被存储指令使用时，在异常返回期间，对下一个要处理的中断的选择发生较晚的更改可能导致中断控制器确认的中断与处理器获取的向量之间出现不匹配。

失败发生在以下条件满足时：

1. 中断A的处理程序正在执行。 2. 优先级相同或更低的中断B正在待处理。

3. 一个带有立即偏移指令的存储操作被执行到可缓冲的位置。

   – 存储/存储半字/存储字 <Rt>, [<Rn>,#立即数] – 存储/存储半字/存储字 <Rt>, [<Rn>,#立即数]! – 存储/存储半字/存储字 <Rt>, [<Rn>], #立即数

4. 可以执行任意数量的附加数据处理指令。

5. 执行一条导致异常返回的BX指令 rn。

6. 存储数据应用了等待状态，使得数据在BX指令执行后至少两个周期才被接受。

   至少，这种情况需要两个周期，前提是存储指令和BX指令之间没有其他指令them. – 观察此勘误所需的等待状态数量需要增加，增加的数量等于存储指令与中断服务程序退出指令之间的周期数。

7. 在总线接受缓冲存储数据之前，另一个中断C被断言，其优先级与A相同或更低，但比B的优先级更高。

Example:

The processor should execute interrupt handler C, and on completion of handler C should execute the handler for B. If the conditions above are met, then this erratum results in the processor erroneously clearing the pending state of interrupt C, and then executing the handler for B twice. The first time the handler for B is executed it will be at interrupt C's priority level. If interrupt C is pended by a level-based interrupt which is cleared by C's handler then interrupt C will be pended again once the handler for B has completed and the handler for C will be executed.

As the STM32 interrupt C is level based, it eventually becomes pending again and is subsequently handled.

Workaround

For software not using the memory protection unit, this erratum can be worked around by setting DISDEFWBUF in the Auxiliary Control Register.

In all other cases, the erratum can be avoided by ensuring a DSB occurs between the store and the BX instruction. For exception handlers written in C, this can be achieved by inserting the appropriate set of intrinsics or inline assembly just before the end of the interrupt function, for example:

ARMCC:

```
...
__schedule_barrier();
__asm{DSB};
__schedule_barrier();
}
```

GCC:

```
...
__asm volatile ("dsb 0xf":::"memory");
}
```

2.2 System

2.2.1 ART Accelerator prefetch queue instruction is not supported

Description

The ART Accelerator prefetch queue instruction is not supported on revision A devices.

This limitation does not prevent the ART Accelerator from using the cache enable/disable capability and the selection of the number of wait states according to system frequency.

Workaround

- Revision A: None.
- Other revisions: fixed.

2.2.2 MCU device ID is incorrect

Description

On revision A devices, the STM32F40x and STM32F41x have the same MCU device ID as the STM32F20x and STM32F21x devices. Reading the revision identifier returns 0x2000 instead of 0x1000. The device ID and revision ID can be read from address 0xE004 2000.

示例:
处理器应执行中断处理程序 C，并在处理程序 C 执行完毕后执行中断 B 的处理程序。如果上述条件得到满足，则此错误会导致处理器错误地清除中断 C 的挂起状态，然后执行中断 B 的处理程序两次。第一次执行中断 B 的处理程序时，其执行级别为中断 C 的优先级。如果中断 C 是由一个基于级别的中断挂起，而该中断被 C 的处理程序清除，则在 B 的处理程序执行完毕后，当 C 的处理程序再次执行时，中断 C 会被重新挂起。

由于STM32中断C是电平触发的，它最终会再次变为待处理，并随后被处理。

变通方法

对于不使用内存保护单元的软件，此错误可以通过在辅助控制寄存器中设置DISDEFWBUF来解决。

在所有其他情况下，可以通过确保在存储操作和BX指令之间发生DSB来避免错误。对于用C语言编写的异常处理程序，可以在中断函数结束前插入适当的内联汇编或内联函数，例如：

ARMCC:

```
... __schedule_barrier(); __asm
{DSB}; __schedule_barrier();
}
```

GCC:

```
... __asm volatile ("dsb 0xf":::"memory"); }
```

2.2 系统

2.2.1节 ART Accelerator 预取队列指令不被支持

描述

ART Accelerator的预取队列指令不支持在A版本设备上。
此限制不会阻止ART Accelerator使用缓存启用/禁用功能以及根据系统频率选择等待状态的数量。

变通方法

- 修订版 A: 无.
- 其他修订：已修复。

2.2.2 MCU设备ID不正确

描述

在A版设备上，STM32F40x和STM32F41x的MCU设备ID与STM32F20x和STM32F21x设备相同。读取修订标识符返回的是0x2000而不是0x1000。设备ID和修订ID可以从地址0xE004 2000读取。

Workaround

- Revision A:
  To differentiate the STM32F4xxx from the STM32F2xxx series, read the MCU device ID and the core device:
  - For STM32F2xxx
    - MCU device ID = STM32F2xxx device ID
    - Core device = Cortex®-M3
  - For STM32F4xxx
    - MCU device ID = STM32F4xxx device ID
    - Core device = Cortex®-M4
- Other revisions: fixed.

2.2.3 Debugging Stop mode and SysTick timer

Description

If the SysTick timer interrupt is enabled during the Stop mode debug (DBG_STOP bit set in the DBGMCU_CR register), it wakes up the system from Stop mode.

Workaround

To debug the Stop mode, disable the SysTick timer interrupt.

2.2.4 Debugging Stop mode with WFE entry

Description

When the Stop debug mode is enabled (DBG_STOP bit set in the DBGMCU_CR register), the software debugging is allowed during Stop mode. However, if the application software uses the WFE instruction to enter Stop mode, after wake-up, some instructions may be missed if the WFE is followed by sequential instructions. This affects only Stop debug mode with WFE entry.

Workaround

To debug Stop mode with WFE entry, the WFE instruction must be inside a dedicated function with one instruction (NOP) between the execution of the WFE and the Bx LR. For example:

```
__asm void _WFE(void)
WFE
NOP
BX lr }
```

2.2.5 Debugging Sleep/Stop mode with WFE/WFI entry

Description

When the Sleep debug or Stop debug mode is enabled (DBG_SLEEP bit or DBG_STOP bit are set in the DBGMCU_CR register), software debugging is allowed during Sleep or Stop mode. After wake-up, some unreachable instructions can be executed if the following conditions are met:

- The application software disables the Prefetch queue,
- the number of wait states configured for the flash memory interface is higher than zero, and
- the linker places the WFE or WFI instruction on a 4-byte aligned addresses (0x080xx xxx4).

变通方法

- 修订版A：为了区分STM32F4xxx与STM32F2xxx系列，需要读取MCU设备ID和核心设备：

  – 针对STM32F2xxx ◦ MCU设备ID = STM32F2xxx设备ID ◦ 核心设备 = Cortex®-M3 – 针对STM32F4xxx ◦ MCU设备ID = STM32F4xxx设备ID ◦ 核心设备 = Cortex®-M4

- 其他修订：已修复。

2.2.3 调试停止模式和SysTick计时器

描述

如果在停止模式调试期间启用了SysTick定时器中断（DBGMCU_CR寄存器中的DBG_STOP位被设置），则会将系统从停止模式唤醒。

变通方法

为了调试停止模式，请禁用SysTick定时器中断。

2.2.4 调试停止模式与WFE入口

描述

当停止调试模式被启用（DBGMCU_CR寄存器中的DBG_STOP位被设置）时，软件调试可以在停止模式下进行。然而，如果应用程序软件使用WFE指令进入停止模式，则在唤醒后，如果WFE后紧跟顺序指令，可能会漏掉某些指令。这仅影响通过WFE指令进入的停止调试模式。

变通方法

为了调试带有WFE入口的停止模式，WFE指令必须位于一个专用函数中，并且在执行WFE指令和Bx LR指令之间插入一个NOP指令。例如：

```
__asm void 无等待函数(void)
WFE  NOP BX lr }
```

2.2.5 通过 WFE/WFI 入口对睡眠/停止模式进行调试

描述

当Sleep调试或Stop调试模式被启用（DBGMCU_CR寄存器中的DBG_SLEEP位或DBG_STOP位被设置）时，软件调试可以在Sleep或Stop模式下进行。唤醒后，如果满足以下条件，可以执行某些不可达指令：

- 应用程序软件禁用了Prefetch队列，
- 为闪存存储器接口配置的等待状态数量大于零，且
- 链接器将 WFE 或 WFI 指令放置在 4 字节对齐的地址 (0x080xx xxx4) 上。

Workaround

Apply one of the following measures:

- Add three NOPs after WFI/WFE instruction.
- Keep one AHB master active during Sleep (example keep DMA1 or DMA2 RCC clock enable bit set).
- Execute WFI/WFE instruction from routines inside the SRAM.

2.2.6 Wake-up sequence from Standby mode when using more than one wake-up source

Description

The various wake-up sources are logically OR-ed in front of the rising-edge detector that generates the wake-up flag (WUF). The WUF needs to be cleared before Standby mode entry, otherwise the MCU wakes up immediately. If one of the configured wake-up sources is kept high during the clearing of the WUF (by setting the CWUF bit), it may mask further wake-up events on the input of the edge detector. As a consequence, the MCU may not be able to wake up from Standby mode.

Workaround

To avoid this problem, apply the following sequence before entering Standby mode:

1. Disable all used wake-up sources.
2. Clear all related wake-up flags.
3. Reenable all used wake-up sources.
4. Enter Standby mode.

Note: Be aware that, when applying this workaround, if one of the wake-up sources is still kept high, the MCU enters Standby mode but then it wakes up immediately and generates a power reset.

2.2.7 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. Only the 4-wire JTAG port configuration is impacted.

Workaround

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

2.2.8 PDR_ON pin not available on LQFP100 package except for revision A devices

Description

On all device revisions except for revision A, the PDR_ON pin (pin 99), available on LQFP100 package, is replaced by VSS. As a consequence, the POR/PDR feature is always enabled.

Workaround

- Applications using revision A devices with PDR_ON pin connected to V_{DD} (POR/PDR feature enabled): Connect the former PDR_ON pin to V_{SS} on revision Z devices.
- Applications using revision A devices with PDR_ON pin connected to V_{SS} (POR/PDR feature disabled): No modification is required when migrating to other device revisions. However, it is no longer possible to supply the product from a 1.7 V V_{DD} on LQFP100 package since V_{DD} minimum value is 1.8 V when the POR/PDR feature is enabled.

变通方法

应用以下措施之一：

- 在WFI/WFE指令之后添加三个NOP。
- 在睡眠期间保持一个AHB主设备激活（例如保持DMA1或DMA2的RCC时钟使能位设置）.
- 从SRAM内部的例程中执行WFI/WFE指令。

2.2.6 从待机模式唤醒的唤醒序列当使用多个唤醒源时

描述

各种唤醒源在上升沿检测器之前进行逻辑或运算，该检测器生成唤醒标志（WUF）。在进入待机模式之前，必须清除唤醒标志（WUF），否则MCU会立即唤醒。如果在清除唤醒标志（WUF）的过程中（通过设置CWUF位），某个已配置的唤醒源保持高电平，则可能会在边沿检测器的输入端屏蔽后续的唤醒事件。因此，MCU可能无法从待机模式中唤醒。

变通方法

为了避免这个问题，请在进入待机模式之前应用以下序列：
1. 禁用所有已使用的唤醒源。2. 清除所
有相关的唤醒标志。3. 重新启用所有已
使用的唤醒源。4. 进入待机模式。

Note: Be aware that, when applying this workaround, if one of the wake-up sources is still kept high, the MCU enters Standby mode but then it wakes up immediately and generates a power reset.

2.2.7 没有NJTRST引脚的完整JTAG配置不可用

描述

当在调试模式下使用JTAG调试端口时，如果将NJTRST引脚（PB4）用作通用输入/输出引脚，则与调试器的连接将被断开。只有4线JTAG端口配置受到影响。

变通方法

使用SWD调试端口，而不是完整的4线JTAG端口。

2.2.8 PDR_ON 引脚 n不适用于LQFP100封装，除修订版外 n A类设备

描述

除版本A外，所有设备版本的PDR_ON引脚（引脚99），可在LQFP100封装中找到，被替换为VSS。因此，POR/PDR功能始终启用。

变通方法

- 使用A版本设备且PDR_ON引脚连接到VDD（POR/PDR功能已启用）的应用：将原PDR_ON引脚连接到Z版本设备的VSS。
- 使用A版本设备且PDR_ON引脚连接到VSS（POR/PDR功能禁用）的应用：迁移到其他设备版本时无需进行任何修改。然而，由于启用POR/PDR功能时VDD最小值为1.8 V，无法再通过LQFP100封装上的1.7 V VDD为产品供电。

2.2.9 Incorrect BOR option byte when consecutively programming BOR option byte

Description

When the AHB prescaler is greater than 2, and consecutive BOR option byte program operations are performed without resetting the device, then an incorrect value may be programmed in the BOR option byte.

Workaround

To program consecutive BOR option byte values, either configure the AHB prescaler to 1 or 2, or perform a system reset between each BOR option byte program operation.

2.2.10 Configuration of PH10 and PI10 as external interrupts is erroneous

Description

PH10 or PI10 is selected as the source for the EXTI10 external interrupt by setting bits EXTI10[3:0] of the SYSCFG_EXTICR3 register to 0x0111 or 0x1000, respectively. Instead, this operation wrongly enables PH2 and PI2 as external interrupt inputs.

As a result, it is not possible to use PH10/PI10 as interrupt sources if PH2/PI2 are not selected as interrupt source as well. This means that bits EXTI10[3:0] of the SYSCFG_EXTICR3 register and bits EXTI2[3:0] of the SYSCFG_EXTICR1 must be programmed to the same value:

- 0x0111 to select PH10/PH2
- 0x1000 to select PI10/PI2

Workaround

None.

2.2.11 Slowing down APB clock during a DMA transfer

Description

When the CPU modifies the APB clock (slows down the clock by changing the AHB/APB prescaler from 1 to 2, 1 to 4, 1 to 8 or 1 to 16) while the DMA is performing a write access to the same APB peripherals, the current DMA transfer is blocked. Only a system reset recovers.

Workaround

Before slowing down the APB clock, wait until the end of the DMA transfer on this APB.

2.2.12 MPU attribute to RTC and IWDG registers incorrectly managed

Description

If the MPU is used and the nonbufferable attribute is set to the RTC or IWDG memory map region, the CPU access to the RTC or IWDG registers may be treated as bufferable, provided there is no APB prescaler configured (AHB/APB prescaler is equal to 1).

Workaround

If the nonbufferable attribute is required for these registers, perform by software a read after the write to guaranty the completion of the write access.

2.2.13 Delay after an RCC peripheral clock enabling

Description

A delay may be observed between an RCC peripheral clock enable and the effective peripheral enabling. It must be taken into account in order to manage the peripheral read/write from/to registers.

2.2.9 连续编程BOR选项字节时，BOR选项字节错误

描述

当AHB预分频器大于2时，且连续的BOR选项字编程操作在不复位设备的情况下执行，则可能在BOR选项字中编程错误的值。

变通方法

要编程连续的BOR选项字值，可以将AHB预分频器配置为1或2，或在每次BOR选项字编程操作之间执行系统复位。

2.2.10 PH10和PI10配置为外部中断是错误的

描述

通过将 SYSCFG_EXTICR3 寄存器的 EXTI10[3:0] 位设置为 0x0111 或 0x1000，分别选择 PH10 或 PI10 作为 EXTI10 外部中断的源。相反，该操作错误地将 PH2 和 PI2 启用为外部中断输入。

因此，如果 PH2/PI2 未被选为中断源，则无法将 PH10/PI10 用作中断源。这意味着必须将 SYSCFG_EXTICR3 寄存器的 EXTI10[3:0] 位和 SYSCFG_EXTICR1 的 EXTI2[3:0] 位编程为相同值：

- 0x0111 用于选择 PH10/PH2
- 0x1000 用于选择 PI10/PI2

变通方法

无。

2.2.11 在DMA传输期间降低APB时钟

描述

当CPU修改APB时钟（通过将AHB/APB预分频器从1改为2，1到4，1到8或1到16）时，如果DMA正在对同一APB外设执行写访问，当前DMA传输会被阻塞。只有系统复位才能恢复。

变通方法

在降低APB时钟速度之前，等待此APB上的DMA传输结束。

2.2.12 MPU对RTC和IWDG寄存器的属性管理不正确

描述

如果使用了MPU，并且将非缓冲属性设置为RTC或IWDG内存映射区域，CPU对RTC或IWDG寄存器的访问可能被视为可缓冲的，前提是未配置APB预分频器（AHB/APB预分频器等于1）。

变通方法

如果这些寄存器需要非缓冲属性，则通过软件执行写后读操作以确保写访问的完成。

2.2.13 RCC外设时钟使能后的延迟

描述

可能会观察到RCC外设时钟使能与实际外设使能之间的延迟。必须考虑这一点，以管理外设的读/写操作。

This delay depends on the peripheral mapping:

- If the peripheral is mapped on the AHB: the delay may be equal to two AHB cycles.
- If the peripheral is mapped on the APB: the delay may be equal to 1 + (AHB/APB prescaler) cycles.

Workaround

Apply one of the following measures:

- Use the DSB instruction to stall the Arm® Cortex®-M4 CPU pipeline until the instruction has completed.
- Insert “n” NOPs between the RCC enable bit write and the peripheral register writes (n = 2 for AHB peripherals, n = 1 + AHB/APB prescaler for APB peripherals).
- Simply insert a dummy read operation from the corresponding register just after enabling the peripheral clock.

2.2.14 Battery charge monitoring lower than 2.4 V

Description

If V_{DD} = V_{DDA} is lower than or equal to 2.4 V, the V_{BAT} conversion correctness is not guaranteed in full temperature and voltage ranges. When V_{BAT} is set, the voltage divider bridge is enabled, and V_{BAT} / 2 is connected to the ADC input. In order to correctly monitor the battery charge, the input of the ADC must not be higher than V_{DDA} - 0.6 V. Thus, V_{BAT} / 2 < V_{DD} – 0.6 V implies that V_{DD} > 2.4 V.

Workaround

None. V_{DD} = V_{DDA} must be greater than 2.4 V.

2.2.15 Internal noise impacting the ADC accuracy

Description

An internal noise generated on V_{DD} supplies and propagated internally may impact the ADC accuracy. This noise is always present whatever the power mode of the MCU (Run or Sleep).

Workaround

Use the following sequence to adapt the accuracy level to the application requirements:

1. Configure the flash memory ART with prefetch OFF and data + instruction cache ON.
2. Use averaging and filtering algorithms on ADC output codes.

For more detailed workarounds, refer to the application note "*How to improve ADC accuracy when using STM32F2xx and STM32F4xx microcontrollers*" (AN4073).

2.2.16 RDP level 2 and sector write protection configuration

Description

When the MCU is protected with RDP level2, the configuration of the sector write protection remains changeable by the user code.

Workaround

Protect sensitive sectors and the FLASH_OPTCR register using the Arm® Cortex®-M MPU (memory protection unit) taking special care of ISR management.

2.2.17 Possible delay in backup domain protection disabling/enabling after programming the DBP bit

Description

Depending on the AHB/APB1 prescaler, a delay between DBP bit programming and the effective disabling/enabling of the backup domain protection can be observed and must be taken into account.

The higher the APB1 prescaler value, the higher the delay.

该延迟取决于外围设备映射：

- 如果外设映射到AHB：延迟可能等于两个AHB周期。
- 如果外设映射到APB：时延可能等于1 +（AHB/APB 预分频器）周期。

变通方法

应用以下措施之一：

- 使用DSB指令阻塞Arm® Cortex®-M4 CPU流水线，直到该指令完成。
- 在RCC使能位写入和外设寄存器写入之间插入n个NOP指令（n = 2用于AHB外设，n = 1 +用于AHB/APB预分频器的APB外设）。
- 在启用外设时钟后，立即从对应的寄存器执行一次虚拟读取操作。

2.2.14 电池充电监控低于2.4 {v*}

描述

如果 VDD = VDDA 小于或等于 2.4 V，则在完整的温度和电压范围内，VBAT 转换的准确性无法得到保证。当设置 VBAT 时，电压分压桥被启用，并且 VBAT / 2 连接到 ADC 输入。为了正确监测电池电量，ADC 输入必须不超过 VDDA - 0.6 V。因此，VBAT / 2 < VDD – 0.6 V 表示 VDD > 2.4 V。

变通方法

None. VDD = VDDA 必须大于 2.4伏

2.2.15 内部噪声影响ADC精度

描述

在{v*}电源上产生的内部噪声以及在内部传播的噪声可能影响ADC转换器精度。无论MCU的电源模式如何（运行或睡眠），这种噪声始终存在。

变通方法

使用以下步骤调整精度级别以满足应用需求：

1. 配置闪存内存 ART，预取关闭，数据 + 指令缓存开启。2. 在ADC输出码上应用平均和滤波算法。如需更多详细解决方案，请参阅应用笔记 "*How to improve ADC accuracy when using STM32F2xx and STM32F4xx microcontrollers*" (AN4073)。

2.2.16 RDP 二级和扇区写保护配置

描述

当微控制器通过RDP level2保护时，扇区写保护的配置仍可由用户代码更改。

变通方法

使用Arm® Cortex®-M内存保护单元（MPU）保护敏感扇区和FLASH_OPTCR寄存器，特别注意中断服务程序管理。

2.2.17 在编程DBP位之后，备份域保护的禁用/启用可能会出现延迟

描述

根据AHB/APB1预分频器的设置，在DBP位编程与备份域保护的有效禁用/启用之间可以观察到延迟，必须予以考虑。APB1预分频器的值越高，延迟越大。

Workaround

Apply one of the following measures:

- Insert a dummy read operation to the PWR_CR register just after programming the DBP bit.
- Wait for the end of the operation (reset through the BDRST bit or write to the backup domain) via a polling loop on targeted registers.

2.2.18 PC13 signal transitions disturb LSE

Description

PC13 toggling in input or output (for example when used for RTC_AF1) may cause an incorrect LSE crystal oscillator clock frequency.

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

Avoid toggling PC13 when LSE is used.

Workaround

None.

2.2.19 In some specific cases, DMA2 data corruption occurs when managing AHB and APB2 peripherals in a concurrent way

Description

When the DMA2 is managing concurrent requests of AHB and APB2 peripherals, the transfer on the AHB can be performed several times. Impacted peripheral are:

- QUADSPI: indirect mode read and write transfers
- FSMC: read and write operation with external device having FIFO
- GPIO: DMA2 transfers to GPIO registers (in memory-to-peripheral transfer mode). The transfers from the GPIOs register are not impacted.

The data corruption is due to multiple DMA2 accesses over the AHB peripheral port impacting peripherals embedding a FIFO.

For transfer to the internal SRAM through the DMA2 AHB peripheral port, the accesses can be performed several times but without data corruptions in cases of concurrent requests.

Workaround

- Use the DMA2 AHB memory port when reading/writing from/to QUADSPI and FSMC instead of DMA2 AHB default peripheral port.
- Use the DMA2 AHB memory port when writing to GPIOs instead of DMA2 AHB default peripheral port.

For more details about DMA controller features, refer to the section *Take benefits of DMA2 controller and system architecture flexibility* of the application note "*Using the STM32F2, STM32F4 and STM32F7 Series DMA controller*" (AN4031).

2.2.20 Corrupted content of the backup domain due to a missed power-on reset after this domain supply voltage drop

Description

The backup domain reset may be missed upon a power-on following a power-off, if its supply voltage drops during the power-off phase hitting a window, which is few mV wide before it starts 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.

This missed reset results in unpredictable values of the backup domain registers. This may cause a spurious behavior (such as driving the LSCO output pin on or influencing backup functions).

变通方法

应用以下措施之一:

- 在设置DBP位之后立即向PWR_CR寄存器插入一个虚拟读操作。
- 通过在目标寄存器上的轮询循环等待操作完成（通过BDRST位进行复位或向备份域写入）

2.2.18 PC13信号转换干扰LSE

描述

PC13在输入或输出模式下切换（例如用于RTC_AF1时）可能导致不正确的LSE晶体振荡器时钟频率。

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

当使用LSE时，避免切换PC13。

变通方法

无。

2.2.19 在某些特定情况下，当以并发方式管理AHB和APB2外设时，DMA2数据损坏会发生。

描述

当DMA2管理AHB和APB2外设的并发请求时，AHB上的传输可以执行多次。受影响的外设是:

- QUADSPI: 间接模式读写传输
- FSMC: 与具有FIFO的外部设备进行读写操作
- GPIO：DMA2将数据传输到GPIO寄存器（内存至外设传输模式）。来自GPIO寄存器的传输不会受到影响。

数据损坏是由于通过AHB外设端口的多次DMA2访问影响了嵌入FIFO的外设。

当通过DMA2 AHB外设端口向内部SRAM传输数据时，访问操作可以多次执行，但不会导致数据损坏。

变通方法

- 在从QUADSPI和FSMC读取/写入时，使用DMA2 AHB内存端口而不是DMA2 AHB默认的外设端口
- 在向GPIO写入时，使用DMA2 AHB内存端口，而不是DMA2 AHB默认的外设端口。

如需了解DMA控制器的更多信息，请参见应用笔记“*Using the STM32F2, STM32F4 and STM32F7 Series DMA controller*”中的第*Take benefits of DMA2 controller and system architecture flexibility*部分（AN4031）。

2.2.20 备份域中损坏的内容是由于在本域供电电压下降后未执行上电复位

描述

在断电后通电时可能会被遗漏，如果其供电电压在断电阶段下降时进入一个窗口，该窗口在再次上升前仅宽几毫伏。在此关键窗口期间，触发器无法安全地保留信息，且备份域复位尚未被触发。该窗口位于100 mV至700 mV之间，具体位置主要取决于设备和温度。

此未复位会导致备份域寄存器的不可预测的值。这可能引起误动作（例如驱动LSCO输出引脚开启或影响备份功能）。

Workaround

Apply one of the following measures:

- In the application, let the V_{DD} and V_{BAT} supply voltages fall to a level below 100 mV for more than 200 ms before a new power-on.
- If the above workaround cannot be applied, and the boot follows a power-on reset, erase the backup domain by software.

2.3 FSMC

2.3.1 Dummy read cycles inserted when reading synchronous memories

Description

When performing a burst read access from a synchronous memory, two dummy read accesses are performed at the end of the burst cycle whatever the type of burst access.

The extra data values read are not used by the FSMC and there is no functional failure.

Workaround

None.

2.3.2 FSMC synchronous mode and NWAIT signal disabled

Description

When the FSMC operates in synchronous mode with the NWAIT signal disabled, if the polarity (WAITPOL in the FSMC_BCRx register) of the NWAIT signal is identical to that of the NWAIT input signal level, the system hangs and no fault is generated.

Workaround

Do not set the PD6 port to AF12 (NWAIT input). Configure the NWAIT polarity to active high by setting the WAITPOL bit of the FSMC_BCRx register.

2.3.3 FSMC NOR/PSRAM controller: asynchronous read access on bank 2 to 4 returns wrong data when bank 1 is in synchronous mode (BURSTEN bit is set)

Description

If an interrupt occurs during a CPU AHB read access to one NOR/PSRAM bank (bank 2 to 4) which is enabled in asynchronous mode, while bank 1 of the NOR/PSRAM controller is configured in synchronous read mode (BURSTEN bit set), then the FSMC NOR/PSRAM controller returns wrong data. This limitation does not occur when using the DMA or when only bank 1 is used in synchronous mode.

Workaround

If multiple banks are enabled in mixed asynchronous and synchronous modes, use any NOR/PSRAM bank for synchronous read accesses, except for bank 1. As a consequence the continuous clock feature is not available.

2.4 SDIO

2.4.1 Wrong data written during SDIO hardware flow control

Description

When enabling the hardware flow control by setting bit 14 of the SDIO_CLKCR register, glitches may occur on the SDIOCLK output clock, resulting in wrong data to be written to the SD/MMC card or to the SDIO device. As a consequence, a CRC error is reported to the SD/SDIO MMC host interface (DCRCFAIL bit set in the SDIO_STA register).

变通方法

应用以下措施之一:

- 在应用中，使VDD和VBAT供电电压降至低于100 mV的水平，持续时间超过200毫秒在重新上电之前.
- 如果上述变通方法无法应用，且系统启动遵循上电复位，则通过软件擦除备份域。

2.3 有限状态机控制器

2.3.1 在读取同步存储器时插入的虚拟读取周期

描述

在进行突发读取访问时，无论突发访问的类型如何，都会在突发周期结束时执行两次虚拟读取访问。

读取的额外数据值 {v*} 未被FSMC使用，且没有功能故障。

变通方法

无。

2.3.2 FSMC 同步模式及 NWAIT 信号禁用

描述

当FSMC以同步模式运行且NWAIT信号被禁用时，如果NWAIT信号的极性（FSMC_BCRx寄存器中的WAITPOL）与NWAIT输入信号电平的极性相同，系统会挂起且不会产生任何故障。

变通方法

不要将PD6端口设置为AF12（NWAIT输入）。通过设置FSMC_BCRx寄存器的WAITPOL位，将NWAIT极性配置为高电平有效。

2.3.3 FSMC NOR/PSRAM控制器：当银行1处于同步模式（BURSTEN位已设置）时，对银行2至4的异步读取访问会返回错误数据。

描述

如果在CPU AHB读取访问某个NOR/PSRAM存储体（存储体2至4）时发生中断，该存储体已启用在异步模式，当NOR/PSRAM控制器的Bank 1配置为同步读取模式（BURSTEN位设置）时，FSMC NOR/PSRAM控制器会返回错误数据。此限制在使用DMA或仅在同步模式下使用Bank 1时不会出现。

变通方法

如果在混合异步和同步模式下启用了多个银行，请为同步读取访问使用任何NOR/PSRAM银行，但不包括银行1。因此，连续时钟功能不可用。

2.4 安全数字输入/输出

2.4.1 在SDIO硬件流控制期间写入错误数据

描述

当通过设置SDIO_CLKCR寄存器的位14启用硬件流控制时，可能会在SDIOCLK输出时钟上出现毛刺，导致错误的数据被写入SD/MMC卡或SDIO设备。因此，SD/SDIO MMC主机接口会报告CRC错误（SDIO_STA寄存器中的DCRCFAIL位被设置）。

Workaround

None.

Note: Do not use the hardware flow control. Overrun errors (Rx mode) and FIFO underrun (Tx mode) must be managed by the application software.

2.4.2 Wrong CCRCFAIL status after a response without CRC is received

Description

The CRC is calculated even if the response to a command does not contain any CRC field. As a consequence, after the SDIO command IO_SEND_OP_COND (CMD5) is sent, the CCRCFAIL bit of the SDIO_STA register is set.

Workaround

The CCRCFAIL bit in the SDIO_STA register must be ignored by the software. CCRCFAIL must be cleared by setting the CCRCFAILC bit of the SDIO_ICR register after receiving the response to the CMD5 command.

2.4.3 SDIO clock divider Bypass mode may not work properly

Description

In high speed communication mode, when SDIO_CK is equal to 48 MHz (PLL48_output = 48 MHz), the BYPASS bit is equal to 1 and the NEGEDGE bit is equal to 0 (respectively bit 10 and bit 13 of the SDIO_CLKCR register), the hold timing at the I/O pin is not aligned with the SD/MMC 2.0 specifications.

Workaround

When neither the USB nor the RNG is used, the PLL48_output (SDIOCLK) frequency can be raised to 75 MHz, enabling to reach 37.5 MHz on SDIO_CK in high speed mode. The BYPASS bit, the CLKDIV bit, and the NEGEDGE bit are equal to 0.

2.4.4 Data corruption in SDIO clock dephasing (NEGEDGE) mode

Description

Setting the NEGEDGE bit may lead to invalid data and command response read.

Workaround

None.

Avoid configuring the NEGEDGE bit to 1.

2.4.5 CE-ATA multiple write command and card busy signal management

Description

The CE-ATA card may inform the host that it is busy by driving the SDIO_D0 line low, two cycles after the transfer of a write command (RW_MULTIPLE_REGISTER or RW_MULTIPLE_BLOCK). When the card is in a busy state, the host must not send any data until the BUSY signal is de-asserted (SDIO_D0 released by the card).

This condition is not respected if the data state machine leaves the IDLE state (write operation programmed and started, DTEN = 1, DTDIR = 0 in the SDIO_DCTRL register, and TXFIFOE = 0 in the SDIO_STA register).

As a consequence, the write transfer fails and the data lines are corrupted.

Workaround

After sending the write command (RW_MULTIPLE_REGISTER or RW_MULTIPLE_BLOCK), the application must check that the card is not busy by polling the BSY bit of the ATA status register using the FAST_IO (CMD39) command before enabling the data state machine.

变通方法

无。

Note: Do not use the hardware flow control. Overrun errors (Rx mode) and FIFO underrun (Tx mode) must be managed by the application software.

2.4.2 在收到无CRC响应后出现错误的CCRCFAIL状态

描述

CRC 被计算，即使命令的响应不包含任何 CRC 字段。因此，在发送 SDIO 命令 IO_SEND_OP_COND (CMD5) 之后，SDIO_STA 寄存器中的 CCRCFAIL 位被置位。

变通方法

SDIO_STA寄存器中的CCRCFAIL位必须被软件忽略。在接收到CMD5命令的响应后，必须通过设置SDIO_ICR寄存器中的CCRCFAILC位来清除CCRCFAIL。

2.4.3 SDIO时钟分频器旁路模式可能无法正常工作

描述

在高速通信模式下，当SDIO_CK等于48 MHz（PLL48_output = 48 MHz）时，BYPASS位等于1，NEGEDGE位等于0（分别对应SDIO_CLKCR寄存器的位10和位13），I/O引脚上的保持时间不符合SD/MMC 2.0规范。

变通方法

当既不使用USB也不使用RNG时，PLL48_output (SDIOCLK)的频率可以提升至75 MHz，从而在高速模式下达到SDIO_CK的37.5 MHz。BYPASS位、CLKDIV位和NEGEDGE位均等于0。

2.4.4 SDIO时钟相位偏移（NEGEDGE）模式中的数据损坏

描述

设置 N EGEDGE位可能导致无效数据和命令响应 e读了。

变通方法

无。
避免将NEGEDGE位配置为1。

2.4.5 CE-ATA 多个写入命令和卡忙信号管理

描述

CE-ATA卡可能通过在写命令（RW_MULTIPLE_REGISTER或RW_MULTIPLE_BLOCK）传输后的两个周期内将SDIO_D0线路拉低，向主机通知其处于忙碌状态。当卡处于忙碌状态时，主机必须在BUSY信号解除（SDIO_D0由卡释放）之前不得发送任何数据。如果数据状态机离开空闲状态（写操作被编程并启动，SDIO_DCTRL寄存器中的DTEN = 1，DTDIR = 0，以及SDIO_STA寄存器中的TXFIFOE = 0），则该条件未被满足。因此，写传输失败，数据线路被破坏。

变通方法

发送写命令（RW_MULTIPLE_REGISTER 或 RW_MULTIPLE_BLOCK）后，应用程序必须通过使用 FAST_IO（CMD39）命令轮询 ATA 状态寄存器的 BSY 位，确认卡未处于忙碌状态，然后才能启用数据状态机。

2.4.6 No underrun detection with wrong data transmission

Description

In case there is an ongoing data transfer from the SDIO host to the SD card and the hardware flow control is disabled (bit 14 of the SDIO_CLKCR is not set), if an underrun condition occurs, the controller may transmit a corrupted data block (with a wrong data word) without detecting the underrun condition when the clock frequencies have the following relationship:

$[3 \times \text{period}(\text{PCLK2}) + 3 \times \text{period}(\text{SDIOCLK})] \geq (32/(\text{BusWidth})) \times \text{period}(\text{SDIO\_CK})$

Workaround

Avoid the above-mentioned clock frequency relationship, by applying one of the following measures:

- Increment the APB frequency, or
- decrease the transfer bandwidth, or
- reduce SDIO_CK frequency.

2.5 ADC

2.5.1 ADC sequencer modification during conversion

Description

When a software start-of-conversion is used as an ADC trigger, and if the ADC_SQRx or ADC_JSQRx register is modified during the conversion, the current conversion is reset and the ADC does not automatically restart the new conversion sequence. The hardware start-of-conversion trigger is not impacted and the ADC automatically restarts the new sequence when the next hardware trigger occurs.

Workaround

When a software start-of-conversion is used, apply the following sequence:

1. First set the SWSART bit in the ADC_CR2 register.
2. Then restart the new conversion sequence.

2.6 DAC

2.6.1 DMA request not automatically cleared by clearing DMAEN

Description

Upon an attempt to stop a DMA-to-DAC transfer, the DMA request is not automatically cleared by clearing the DAC channel bit of the DAC_CR register (DMAEN) or by disabling the DAC clock.

If the application stops the DAC operation while the DMA request is pending, the request remains pending while the DAC is reinitialized and restarted, with the risk that a spurious DMA request is serviced as soon as the DAC is enabled again.

Workaround

Apply the following sequence to stop the current DMA-to-DAC transfer and restart the DAC:

1. Check if DMAUDR bit is set in DAC_CR.
2. Clear the DAC channel DMAEN bit.
3. Disable the DAC clock.
4. Reconfigure the DAC, DMA and the triggers.
5. Restart the application.

二点四点六 在错误数据传输时无数据不足检测

描述

如果存在从SDIO主机到SD卡的正在进行的数据传输，并且硬件流控制已禁用（SDIO_CLKCR的第14位未设置），则当发生下溢条件时，控制器可能会在未检测到下溢条件的情况下传输损坏的数据块（包含错误的数据字），此时时钟频率具有以下关系：

$$[3 \times \text{period}(\text{PCLK2}) + 3 \times \text{period}(\text{SDIOCLK})] \geq (32/(\text{BusWidth})) \times \text{period}(\text{SDIO\_CK})$$

变通方法

避免上述时钟频率关联，通过应用以下任一措施：

- 增加APB频率，或
- 降低传输带宽，或
- 降低 SDIO_CK 频率。

2.5 模数转换器

2.5.1 在转换过程中对ADC顺序器的修改

描述

当使用软件转换开始作为ADC触发器时，如果在转换过程中修改了ADC_SQRx或ADC_JSQRx寄存器，当前转换将被重置，ADC不会自动重新启动新的转换序列。硬件转换开始触发器不受影响，当下一个硬件触发发生时，ADC会自动重新启动新的转换序列。

变通方法

当使用软件启动转换时，应按照以下步骤操作：1. 首先在ADC_CR2寄存器中设置SWSART位。2. 然后重新启动新的转换序列。

2.6 数模转换器

2.6.1 清除DMAEN不会自动清除DMA请求

描述

在尝试停止DMA到DAC传输时，DMA请求不会自动清除，通过清除DAC_CR寄存器中的DAC通道位（DMAEN）或禁用DAC时钟。

如果应用程序在DMA请求处于待处理状态时停止DAC操作，该请求将在DAC重新初始化并重新启动期间保持待处理状态，存在虚假的DMA请求一旦DAC再次启用就被处理的风险。

变通方法

应用以下序列以停止当前的DMA到DAC传输并重新启动DAC：

1. 检查DAC_CR中的DMAUDR位是否被置位。2. 清除DAC通道的DMAEN位。3. 禁用DAC时钟。4. 重新配置DAC、DMA和触发器。5. 重启应用程序。

2.6.2 DMA underrun flag not set when an internal trigger is detected on the clock cycle of the DMA request acknowledge

Description

When the DAC channel operates in DMA mode (DMAEN of DAC_CR register set), the DMA channel underrun flag (DMAUDR of DAC_SR register) fails to rise upon an internal trigger detection if that detection occurs during the same clock cycle as a DMA request acknowledge. As a result, the user application is not informed that an underrun error occurred.

This issue occurs when software and hardware triggers are used concurrently to trigger DMA transfers.

Workaround

None.

2.7 TIM

2.7.1 PWM re-enabled in automatic output enable mode despite of system break

Description

In automatic output enable mode (AOE bit set in TIMx_BDTR register), the break input can be used to do a cycle-by-cycle PWM control for a current mode regulation. A break signal (typically a comparator with a current threshold) disables the PWM output(s) and the PWM is re-armed on the next counter period.

However, a system break (typically coming from the CSS Clock security System) is supposed to stop definitively the PWM to avoid abnormal operation (for example with PWM frequency deviation).

In the current implementation, the timer system break input is not latched. As a consequence, a system break indeed disables the PWM output(s) when it occurs, but PWM output(s) is (are) re-armed on the following counter period.

Workaround

Preferably, implement control loops with the output clear enable function (OCxCE bit in the TIMx_CCMR1/CCMR2 register), leaving the use of break circuitry solely for internal and/or external fault protection (AOE bit reset).

2.7.2 TRGO and TRGO2 trigger output failure

Description

Some reference manual revisions may omit the following information.

The timers can be linked using ITRx inputs and TRGOx outputs. Additionally, the TRGOx outputs can be used as triggers for other peripherals (for example ADC). Since this circuitry is based on pulse generation, care must be taken when initializing master and slave peripherals or when using different master/slave clock frequencies:

- If the master timer generates a trigger output pulse on TRGOx prior to have the destination peripheral clock enabled, the triggering system may fail.
- If the frequency of the destination peripheral is modified on-the-fly (clock prescaler modification), the triggering system may fail.

As a conclusion, the clock of the slave timer or slave peripheral must be enabled prior to receiving events from the master timer, and must not be changed on-the-fly while triggers are being received from the master timer.

This is a documentation issue rather than a product limitation.

Workaround

No application workaround is required or applicable as long as the application handles the clock as indicated.

2.6.2 DMA 下溢标志未设置当在 DMA 请求确认的时钟周期上检测到内部触发信号时{v*}

描述

当DAC通道以DMA模式运行（DAC_CR寄存器的DMAEN位被置位时），如果内部触发检测发生在DMA请求确认的同一时钟周期内，则DMA通道下溢标志（DAC_SR寄存器的DMAUDR标志）无法在检测到内部触发时被置位。因此，用户应用程序不会被告知发生了下溢错误。

当软件和硬件触发器并发用于触发DMA传输时，会出现此问题。

变通方法

无。

二点七 提姆

2.7.1 PWM 重新启用在自动输出使能模式下，尽管系统故障

描述

在自动输出使能模式（TIMx_BDTR寄存器中AOE位被设置）下，断开输入可用于实现逐周期PWM控制，以进行电流模式调节。断开信号（通常是一个具有电流阈值的比较器）会禁用PWM输出（s），并在下一个计数周期重新启用PWM。

然而，系统中断（通常来自CSS时钟安全系统）应彻底停止PWM以避免异常操作（例如PWM频率偏差）。

在当前的实现中，定时器系统断开输入未被锁存。因此，当系统断开发生时，确实会禁用PWM输出（s），但在随后的计数周期中，PWM输出（s）会被重新启用。

变通方法

建议使用输出清除使能功能（TIMx_CCMR1/CCMR2寄存器中的OCxCE位）实现控制环路，将断路电路的使用仅限于内部和/或外部故障保护（AOE位复位）。

2.7.2 TRGO 和 TRGO2 触发输出故障

描述

某些参考手册的修订可能省略以下信息。{v*}

定时器可以通过ITRx输入和TRGOx输出进行连接。此外，TRGOx输出还可以用作其他外设（例如ADC）的触发信号。由于该电路基于脉冲生成，需要注意在初始化主从外设或使用不同的主从时钟频率时的情况：

- 如果主定时器在目标外设时钟使能之前生成触发输出脉冲 {v*}，触发系统可能失效。
- 如果目标外设的频率在实时修改（时钟预分频器修改）时，触发系统可能会失效。

综上所述，必须在接收来自主定时器的事件之前启用从设备定时器或从设备外设的时钟，并且在接收来自主定时器的触发信号时不得更改。

这是一个文档问题，而不是产品限制。

变通方法

只要应用程序按照指示处理时钟，就不需要任何变通方法。

2.7.3 Consecutive compare event missed in specific conditions

Description

Every match of the counter (CNT) value with the compare register (CCR) value is expected to trigger a compare event. However, if such matches occur in two consecutive counter clock cycles (as consequence of the CCR value change between the two cycles), the second compare event is missed for the following CCR value changes:

- in edge-aligned mode, from ARR to 0:
  - first compare event: CNT = CCR = ARR
  - second (missed) compare event: CNT = CCR = 0
- in center-aligned mode while up-counting, from ARR-1 to ARR (possibly a new ARR value if the period is also changed) at the crest (that is, when TIMx_RCR = 0):
  - first compare event: CNT = CCR = (ARR-1)
  - second (missed) compare event: CNT = CCR = ARR
- in center-aligned mode while down-counting, from 1 to 0 at the valley (that is, when TIMx_RCR = 0):
  - first compare event: CNT = CCR = 1
  - second (missed) compare event: CNT = CCR = 0

This typically corresponds to an abrupt change of compare value aiming at creating a timer clock single-cycle-wide pulse in toggle mode.

As a consequence:

- In toggle mode, the output only toggles once per counter period (squared waveform), whereas it is expected to toggle twice within two consecutive counter cycles (and so exhibit a short pulse per counter period).
- In center mode, the compare interrupt flag does note rise and the interrupt is not generated.

Note: The timer output operates as expected in modes other than the toggle mode.

Workaround

None.

2.7.4 Output compare clear not working with external counter reset

Description

The output compare clear event (ocref_clr) is not correctly generated when the timer is configured in the following slave modes: Reset mode, Combined reset + trigger mode, and Combined gated + reset mode.

The PWM output remains inactive during one extra PWM cycle if the following sequence occurs:

1. The output is cleared by the ocref_clr event.
2. The timer reset occurs before the programmed compare event.

Workaround

Apply one of the following measures:

- Use BKIN (or BKIN2 if available) input for clearing the output, selecting the Automatic output enable mode (AOE = 1).
- Mask the timer reset during the PWM ON time to prevent it from occurring before the compare event (for example with a spare timer compare channel open-drain output connected with the reset signal, pulling the timer reset line down).

2.7.3 连续比较事件在特定条件下未被检测到

描述

每个计数器（CNT）值与比较寄存器（CCR）值的匹配都预期会触发一个比较事件。然而，如果此类匹配发生在两个连续的计数器时钟周期（由于两个周期之间CCR值的变化），则后续的CCR值变化将错过第二个比较事件：

- 在边对齐模式中，从 ARR 到 0：– 首次比较事件：CNT = CCR = ARR – 第二次（漏掉的）比较事件：CNT = CCR = 0
- 在居中对齐模式下进行向上计数时，从ARR-1到ARR（如果周期也发生变化，可能是一个新的ARR值）在峰值（即当TIMx_RCR = 0时）：– 第一次比较事件：CNT = CCR = (ARR-1) – 第二次（漏掉的）比较事件：CNT = CCR = ARR
- 在中心对齐模式下进行向下计数时，从1到0的谷值（即当 TIMx_RCR = 0 时）：– 第一比较事件：CNT = CCR = 1 – 第二（未命中）比较事件：CNT = CCR = 0

这通常对应于比较值的突然变化，旨在在翻转模式下产生定时器时钟的单周期宽脉冲。

因此：

- 在翻转模式下，输出仅在每个计数周期内翻转一次（方波），而预期在两个连续的计数周期内翻转两次（从而在每个计数周期内产生一个短脉冲）。
- 在中心模式下，比较中断标志不会上升，且中断不会被生成。

Note: The timer output operates as expected in modes other than the toggle mode.

变通方法

无。

2.7.4 出 比较清除功能无法与外部计数器配合使用 r 重置

描述

当定时器配置为以下从模式时，输出比较清除事件（ocref_clr）未被正确生成：复位模式、组合复位 + 触发模式以及组合门控 + 复位模式。

如果以下序列发生，PWM输出将在额外的一个PWM周期内保持非活动状态：

1. 输出通过ocref_clr事件被清除。 2. 定时器重置发生在编程比较事件之前。

变通方法

应用以下措施之一：

- 使用 BKIN（如有 BKIN2 则使用 BKIN2）输入以清除输出，选择自动输出使能模式（AOE = 1）。
- 在PWM开启时间内屏蔽定时器复位，以防止其在比较事件之前发生（例如，通过将备用定时器比较通道的开漏输出连接到复位信号，将定时器复位线拉低）。

2.8 IWDG

2.8.1 RVU flag not reset in Stop

Description

Successful write to the IWDG_RLR register raises the RVU flag and prevents further write accesses to the register until the RVU flag is automatically cleared by hardware. However, if the device enters Stop mode while the RVU flag is set, the hardware never clears that flag, and writing to the IWDG_RLR register is no longer possible.

Workaround

Ensure that the RVU flag is cleared before entering Stop mode.

2.8.2 PVU flag not reset in Stop

Description

Successful write to the IWDG_PR register raises the PVU flag and prevents further write accesses to the register until the PVU flag is automatically cleared by hardware. However, if the device enters Stop mode while the PVU flag is set, the hardware never clears that flag, and writing to the IWDG_PR register is no longer possible.

Workaround

Ensure that the PVU flag is cleared before entering Stop mode.

2.8.3 RVU flag not cleared at low APB clock frequency

Description

Successful write to the IWDG_RLR register raises the RVU flag and prevents further write accesses to the register until the RVU flag is automatically cleared by hardware. However, at APB clock frequency lower than twice the IWDG clock frequency, the hardware never clears that flag, and writing to the IWDG_RLR register is no longer possible.

Workaround

Set the APB clock frequency higher than twice the IWDG clock frequency.

2.8.4 PVU flag not cleared at low APB clock frequency

Description

Successful write to the IWDG_PR register raises the PVU flag and prevents further write accesses to the register until the PVU flag is automatically cleared by hardware. However, at APB clock frequency lower than twice the IWDG clock frequency, the hardware never clears that flag, and writing to the IWDG_PR register is no longer possible.

Workaround

Set the APB clock frequency higher than twice the IWDG clock frequency.

2.9 RTC

2.9.1 Spurious tamper detection when disabling the tamper channel

Description

If the tamper detection is configured for detecting on the falling edge event (TAMPFLT = 00 and TAMPxTRG = 1) and if the tamper event detection is disabled when the tamper pin is at high level, a false tamper event is detected.

2.8 独立看门狗

二.八.一 RVU标志在停止时未重置

描述

成功写入IWDG_RLR寄存器会设置RVU标志位，并阻止对该寄存器的进一步写入操作，直到RVU标志位被硬件自动清除为止。然而，如果设备在RVU标志位被设置时进入停止模式，则硬件将不会清除该标志位，且无法再对该寄存器进行写入操作。

变通方法

在进入停止模式之前，确保RVU标志被清除。

2.8.2 在停止时PVU标志未重置

描述

向IWDG_PR寄存器成功写入会设置PVU标志，并阻止对该寄存器的进一步写入，直到硬件自动清除该标志。然而，如果在PVU标志被设置时设备进入停止模式，硬件将永远不会清除该标志，且无法再向IWDG_PR寄存器写入。

变通方法

在进入停止模式之前，确保清除PVU标志。

2.8.3 RVU标志位在低APB时钟频率下未被清除

描述

向IWDG_RLR寄存器成功写入会设置RVU标志，并阻止对寄存器的进一步写入，直到该标志由硬件自动清除。然而，当APB时钟频率低于两倍的IWDG时钟频率时，硬件永远不会清除该标志，且无法再向IWDG_RLR寄存器写入。

变通方法

集合APB时钟频率高于两倍的IWDG时钟频率 频率.

2.8.4 在低APB时钟频率下，PVU标志未被清除

描述

向IWDG_PR寄存器成功写入会设置PVU标志，并阻止对该寄存器的进一步写入操作，直到该标志由硬件自动清除。然而，当APB时钟频率低于IWDG时钟频率的两倍时，硬件永远不会清除该标志，且无法再向IWDG_PR寄存器写入。

变通方法

将APB时钟频率设置为高于IWDG时钟频率的两倍。

2.9 实时通信

2.9.1 在禁用防篡改通道时的虚假篡改检测

描述

如果配置了防篡改检测以在下降沿事件（TAMPFLT = 00 和 TAMPxTRG = 1）上触发，并且如果防篡改引脚处于高电平时禁用防篡改事件检测，则会检测到虚假的防篡改事件。

Workaround

None.

2.9.2 RTC calendar registers are not locked properly

Description

When reading the calendar registers with BYPSHAD = 0, the RTC_TR and RTC_DR registers may not be locked after reading the RTC_SSR register. This happens if the read operation is initiated one APB clock period before the shadow registers are updated. This can result in a non-consistency of the three registers. Similarly, the RTC_DR register can be updated after reading the RTC_TR register instead of being locked.

Workaround

Apply one of the following measures:

- Use BYPSHAD = 1 mode (bypass shadow registers), or
- If BYPSHAD = 0, read SSR again after reading SSR/TR/DR to confirm that SSR is still the same, otherwise read the values again.

2.9.3 RTC interrupt can be masked by another RTC interrupt

Description

One RTC interrupt request can mask another RTC interrupt request if they share the same EXTI configurable line. For example, interrupt requests from Alarm A and Alarm B or those from tamper and timestamp events are OR-ed to the same EXTI line (refer to the *EXTI line connections* table in the *Extended interrupt and event controller (EXTI)* section of the reference manual).

The following code example and figure illustrate the failure mechanism: The Alarm A event is lost (fails to generate interrupt) as it occurs in the failure window, that is, after checking the Alarm A event flag but before the effective clear of the EXTI interrupt flag by hardware. The effective clear of the EXTI interrupt flag is delayed with respect to the software instruction to clear it.

Alarm interrupt service routine:

```
void RTC_Alarm_IRQHandler(void)
{
    CLEAR_ALARM_EXTI(); /* Clear the EXTI line flag for RTC alarms*/
    If(ALRAF) /* Check if Alarm A triggered ISR */
    {
        CLEAR_FLAG(ALRAF); /* Clear the Alarm A interrupt pending bit */
        PROCESS_AlarmAEvent(); /* Process Alarm A event */
    }
    If(ALRBF) /* Check if Alarm B triggered ISR */
    {
        CLEAR_FLAG(ALRBF); /* Clear the Alarm B interrupt pending bit */
        PROCESS_AlarmBEvent(); /* Process Alarm B event */
    }
}
```

变通方法

无。

2.9.2 RTC日历寄存器未正确锁定

描述

使用 BYPSHAD = 0 读取日历寄存器时，在读取 RTC_SSR 寄存器后，RTC_TR 和 RTC_DR 寄存器可能未被锁定。这发生在影子寄存器更新前一个 APB 时钟周期启动读取操作的情况下。这可能导致三个寄存器的不一致性。同样，RTC_DR 寄存器可能在读取 RTC_TR 寄存器后被更新，而不是被锁定。

变通方法

应用以下措施之一:

- 使用 BYPSHAD = 1 模式（绕过影子寄存器），或
- 如果 BYPSHAD = 0，则在读取 SSR/TR/DR 后再次读取 SSR 以确认其值是否保持不变，否则再次读取这些值。

2.9.3 实时时钟中断可以被另一个实时时钟中断屏蔽

描述

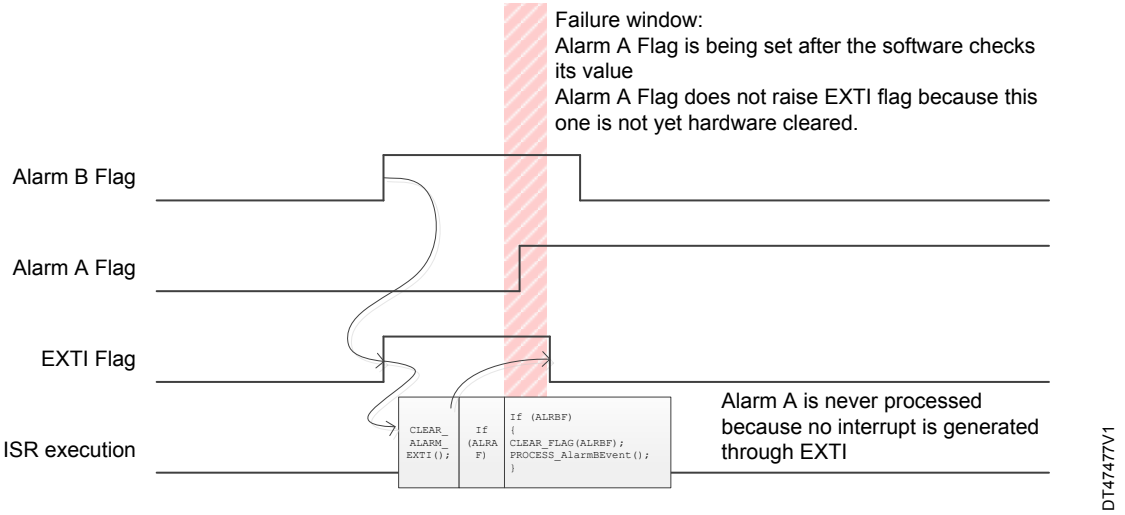
一个RTC中断请求如果与另一个RTC中断请求共享相同的EXTI可配置线，则可以屏蔽另一个RTC中断请求。例如，来自报警A和报警B的中断请求，或来自篡改和时间戳事件的中断请求，被OR合并到同一EXTI线（参见参考手册中*Extended interrupt and event controller (EXTI)*部分的*EXTI line connections*表）。

以下代码示例和图示说明了故障机制：Alarm A事件在发生时被丢失（未能生成中断），即在检查Alarm A事件标志后但在硬件有效清除EXTI中断标志之前。EXTI中断标志的有效清除相对于软件清除指令被延迟。

报警中断服务程序:

```
void RTC_Alarm_IRQHandler(void) {    CLEAR_ALARM_EXTI(); /* 清除 RTC 报警的 EXTI 线标志 */
If(ALRAF) /* 检查 Alarm A 是否触发了 ISR */    {        CLEAR_FLAG(ALRAF); /* 清除 Alarm A 中
断挂起位 */        PROCESS_AlarmAEvent(); /* 处理 Alarm A 事件 */    }    If(ALRBF) /* 检查 Alarm
B 是否触发了 ISR */    {        CLEAR_FLAG(ALRBF); /* 清除 Alarm B 中断挂起位 */        PROCESS
_AlarmBEvent(); /* 处理 Alarm B 事件 */    } }
```

Figure 1. Masked RTC interrupt



Workaround

In the interrupt service routine, apply three consecutive event flag ckecks - source one, source two, and source one again, as in the following code example:

```
void RTC_Alarm_IRQHandler(void)
{
    CLEAR_ALARM_EXTI(); /* Clear the EXTI's line Flag for RTC Alarm */
    If(ALRAF) /* Check if AlarmA triggered ISR */
    {
        CLEAR_FLAG(ALRAF); /* Clear the AlarmA interrupt pending bit */
        PROCESS_AlarmAEvent(); /* Process AlarmA Event */
    }
    If(ALRBF) /* Check if AlarmB triggered ISR */
    {
        CLEAR_FLAG(ALRBF); /* Clear the AlarmB interrupt pending bit */
        PROCESS_AlarmBEvent(); /* Process AlarmB Event */
    }
    If(ALRAF) /* Check if AlarmA triggered ISR */
    {
        CLEAR_FLAG(ALRAF); /* Clear the AlarmA interrupt pending bit */
        PROCESS_AlarmAEvent(); /* Process AlarmA Event */
    }
}
```

2.9.4 Calendar initialization may fail in case of consecutive INIT mode entry

Description

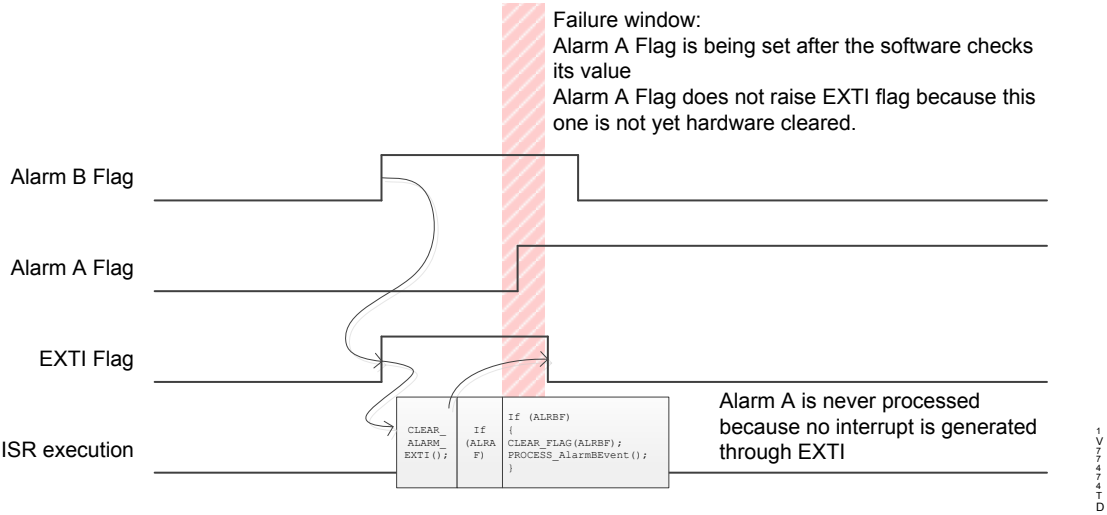
If the INIT bit of the RTC_ISR register is set between one and two RTCCLK cycles after being cleared, the INITF flag is set immediately instead of waiting for synchronization delay (which should be between one and two RTCCLK cycles), and the initialization of registers may fail.

Depending on the INIT bit clearing and setting instants versus the RTCCLK edges, it can happen that, after being immediately set, the INITF flag is cleared during one RTCCLK period then set again. As writes to calendar registers are ignored when INITF is low, a write during this critical period might result in the corruption of one or more calendar registers.

Workaround

After existing the initialization mode, clear the BYPSHAD bit (if set) then wait for RSF to rise, before entering the initialization mode again.

图1. 屏蔽的RTC中断



变通方法

在中断服务程序中，应用三个连续的事件标志检查 - 源一、源二和源一再次，如以下代码示例所示：

```
void RTC_Alarm_IRQHandler(void)
{
    CLEAR_ALARM_EXTI(); /* Clear the EXTI's line Flag for RTC Alarm */
    If(ALRAF) /* Check if AlarmA triggered ISR */
    {
        CLEAR_FLAG(ALRAF); /* Clear the AlarmA interrupt pending bit */
        PROCESS_AlarmAEvent(); /* Process AlarmA Event */
    }
    If(ALRBF) /* Check if AlarmB triggered ISR */
    {
        CLEAR_FLAG(ALRBF); /* Clear the AlarmB interrupt pending bit */
        PROCESS_AlarmBEvent(); /* Process AlarmB Event */
    }
    If(ALRAF) /* Check if AlarmA triggered ISR */
    {
        CLEAR_FLAG(ALRAF); /* Clear the AlarmA interrupt pending bit */
        PROCESS_AlarmAEvent(); /* Process AlarmA Event */
    }
}
```

2.9.4 在连续进入INIT模式的情况下，日历初始化可能会失败

描述

如果在INIT位被清除后的一到两个RTCCLK周期内设置该位，则INITF标志会立即被设置，而不是等待同步延迟（应为一到两个RTCCLK周期），可能导致寄存器初始化失败。

根据 INIT 位清除和设置时刻与 RTCCLK 边沿的关系，可能发生的情况是，在被立即置位，INITF标志在一次RTCCLK周期内被清除，然后再次置位。由于在INITF为低电平时，对日历寄存器的写入会被忽略，因此在此关键时期内进行的写入可能导致一个或多个日历寄存器的损坏。

变通方法

退出初始化模式后，清除BYPSSHAD位（如果已设置），然后等待RSF上升，再重新进入初始化模式。

Note: It is recommended to write all registers in a single initialization session to avoid accumulating synchronization delays.

2.9.5 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.

2.9.6 Detection of a tamper event occurring before enabling the tamper detection is not supported in edge detection mode

Description

When the tamper detection is enabled in edge detection mode (TAMPFLT = 00):

- When TAMPxTRG = 0 (rising edge detection): if the tamper input is already high before enabling the tamper detection, the tamper event may or may not be detected when enabling the tamper detection. The probability to detect it increases with the APB frequency.
- When TAMPxTRG = 1 (falling edge detection): if the tamper input is already low before enabling the tamper detection, the tamper event is not detected when enabling the tamper detection.

Workaround

Check the I/O state by software in the GPIO registers, just after enabling the tamper detection and before writing sensitive values in the backup registers. This ensures that no active edge occurred before enabling the tamper event detection.

2.10 I2C

2.10.1 Spurious bus error detection in controller mode

Description

In controller 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 controller mode and any such transfer continues normally.

Workaround

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

2.10.2 SMBus standard not fully supported

Description

The I2C peripheral is not fully compliant with the SMBus v2.0 standard since it does not support the capability to NACK an invalid byte/command.

Note: It is recommended to write all registers in a single initialization session to avoid accumulating synchronization delays.

2.9.5 当核心在调试中停止时，警报标志可能被重复设置

描述

当核心在调试模式下停止时，即使设备被配置为在调试模式下停止RTC，时钟仍会被供应给子秒级RTC报警计数器。

因此，当子秒计数器用于报警条件（RTC_ALRMASSR 和/或 RTC_ALRMBSSR 寄存器的 MASKSS[3:0] 位字段设置为非零值）且报警条件在进入断点或 printf 之前满足时，RTC_SR 寄存器的 ALRAF 和/或 ALRBF 标志会在断点或 printf 过程中被硬件重复设置，这使得任何尝试清除标志（s）的操作都无效。

变通方法

无。

2.9.6 在启用篡改检测之前发生的篡改事件的检测不支持边缘检测模式。

描述

当在边缘检测模式（TAMPFLT = 00）中启用防篡改检测时：

- 当 TAMPxTRG = 0（上升沿检测）：如果在启用防篡改检测之前，防篡改输入已经为高电平，则启用防篡改检测时，防篡改事件可能被检测到也可能未被检测到。检测到它的概率随着APB频率的增加而增加。
- 当TAMPxTRG = 1（下降沿检测）时：如果在启用干扰之前干扰输入已经处于低电平检测，当启用防篡改检测时，篡改事件未被检测到。

变通方法

在启用防篡改检测且在向备份寄存器写入敏感值之前，通过软件检查GPIO寄存器中的I/O状态。这确保在启用防篡改事件检测之前未发生有效边沿。

2.10 I²C

2.10.1 控制器模式下的虚假总线错误检测

描述

在控制器模式下，总线错误可能被错误地检测，其结果是设置I2C_SR寄存器中的BERR标志并生成总线错误中断（如果该中断已启用）。检测到总线错误对控制器模式下的I2C总线传输没有影响，任何此类传输将继续正常进行。

变通方法

如果在控制器模式下发生总线错误中断，则必须通过软件清除BERR标志。无需其他操作，正在进行的传输可以正常处理。

2.10.2 SMBus标准未完全支持

描述

I2C外设不完全符合SMBus v2.0标准，因为它不支持对无效字节/命令进行NACK的功能。

Workaround

A higher-level mechanism must be used to verify that a write operation is being performed correctly at the target device, such as:

- the SMBAL pin if it is supported by the host
- the alert response address (ARA) protocol
- the host-notify protocol

2.10.3 Start cannot be generated after a misplaced Stop

Description

If a controller generates a misplaced Stop on the bus (bus error) while the microcontroller I2C peripheral attempts to switch to Controller mode by setting the START bit, the Start condition is not properly generated.

Workaround

In the I^2C standard, it is allowed to send a Stop only at the end of the full byte (8 bits + acknowledge), so this scenario is not allowed. Other derived protocols such as CBUS allow it, but they are not supported by the I2C peripheral.

A software workaround consists in asserting the software reset using the SWRST bit of the I2C_CR1 control register.

2.10.4 Mismatch on the “Setup time for a repeated Start condition” timing parameter

Description

In case of repeated Start, the “Setup time for a repeated Start condition” (named Tsu;sta in the I^2C specification) can be slightly violated when the I2C operates in Controller standard mode at a frequency between 88 kHz and 100 kHz.

The issue can occur only in the following configuration:

- In Controller mode
- In Standard mode at a frequency between 88 kHz and 100 kHz (no limitation in Fast mode)
- SCL rise time:
  - If the target does not stretch the clock and the SCL rise time is more than 300 ns (if the SCL rise time is less than 300 ns, the issue does not occur).
  - If the target stretches the clock.

The setup time can be violated independently of the APB peripheral frequency.

Workaround

Reduce the frequency down to 88 kHz or use the I^2C Fast mode, if it is supported by the target.

2.10.5 Data valid time ($t_{VD;DAT}$) violated without the OVR flag being set

Description

The data valid time ($t_{VD;DAT}$, $t_{VD;ACK}$) described by the I^2C standard can be violated (as well as the maximum data hold time of the current data ($t_{HD;DAT}$)) under the conditions described below. This violation cannot be detected because the OVR flag is not set (no transmit buffer underrun is detected).

This limitation can occur only under the following conditions:

- in Target transmit mode
- with clock stretching disabled (NOSTRETCH = 1)
- if the software is late to write to the DR data register, but not late enough to set the OVR flag (the data register is written before)

变通方法

必须使用一种更高层次的机制来验证写操作是否在目标设备上正确执行，例如：

- SMBAL引脚如果由主机支持
- 警报响应地址（ARA）协议
- 主机通知协议

2.10.3 Start不能在错误放置的Stop之后生成。

描述

如果控制器在总线（总线错误）上生成了一个错误的停止信号，同时微控制器的I2C外设尝试通过设置START位切换到控制器模式，那么起始条件未能正确生成。{v*}

变通方法

在I2C标准中，仅允许在完整字节（8位 + 应答）结束时发送停止条件，因此该场景不允许。其他衍生协议如CBUS允许这样做，但I2C外设不支持这些协议。

一种软件解决方法是通过使用I2C_CR1控制寄存器中的SWRST位来发起软件复位。

2.10.4 在“重复启动条件的建立时间”定时参数上不匹配

描述

在重复启动的情况下，当I2C在88 kHz到100 kHz之间的频率下以控制器标准模式运行时，‘重复启动条件的建立时间’（在I2C规范中称为Tsu;sta）可能会略微违反。

该问题只能在以下配置中发生：

- 在控制器模式下
- 在标准模式下，频率范围在88千赫兹至100千赫兹之间（快速模式下无限制）
- SCL上升时间：– 如果目标不拉伸时钟且SCL上升时间超过300 ns（如果SCL上升时间小于300 ns，则问题不会发生）。– 如果目标拉伸时钟。

设置　　时间可以独立于APB外设频率被违反　　流利度

变通方法

将频率降至88 kHz，或在目标设备支持的情况下使用I2C快速模式。

二点一零五 数据 有效时间 (tVD;DAT) 被违反，没有OVR标志　　英集合

描述

I2C标准描述的数据有效时间（tVD;DAT, tVD;ACK）可能在以下条件下被违反（以及当前数据的最大数据保持时间（tHD;DAT））。此违规无法被检测到，因为OVR标志未被置位（未检测到发送缓冲区下溢）。

此限制只能在以下条件下发生：

- 在目标发送模式中 {v*}
- 禁用时钟伸缩（NOSTRETCH = 1）
- 如果软件写入DR数据寄存器的时间较晚，但尚未晚到足以设置OVR标志（数据寄存器在此之前已被写入）

Workaround

If the controller device allows it, use the clock stretching mechanism by clearing the bit NOSTRETCH of the I2C_CR1 register.

If the controller device does not allow it, ensure that the software is fast enough when polling the TXE or ADDR flag to immediately write to the DR data register. For instance, use an interrupt on the TXE or ADDR flag and boost its priority to the higher level.

2.10.6 Both SDA and SCL maximum rise times (t_r) violated when the VDD_I2C bus voltage is higher than ((V_{DD} + 0.3) / 0.7) V

Description

When an external legacy I^2C bus voltage (V_{DD_I2C}) is set to 5 V while the MCU is powered from V_{DD}, the internal 5-Volt tolerant circuitry is activated as soon the input voltage (V_{IN}) reaches the V_{DD} + diode threshold level. An additional internal large capacitance then prevents the external pull-up resistor (R_P) from rising the SDA and SCL signals within the maximum timing (t_r), which is 300 ns in Fast mode and 1000 ns in Standard mode.

The rise time (t_r) is measured from V_{IL} and V_{IH} with levels set at 0.3 V_{DD_I2C} and 0.7 V_{DD_I2C}.

Workaround

The external V_{DD_I2C} bus voltage must be limited to a maximum value of ((V_{DD} + 0.3) / 0.7) V. As a result, when the MCU is powered from V_{DD} = 3.3 V, V_{DD_I2C}must not exceed 5.14 V to be compliant with I^2C specifications.

2.11 USART

2.11.1 Idle frame is not detected if the receiver clock speed is deviated

Description

If the USART receives an idle frame followed by a character, and the clock of the transmitter device is faster than the USART receiver clock, the USART receive signal falls too early when receiving the character start bit, with the result that the idle frame is not detected (the IDLE flag is not set).

Workaround

None.

2.11.2 In full-duplex mode, the Parity Error (PE) flag can be cleared by writing to the data register

Description

In full-duplex mode, when the Parity Error flag is set by the receiver at the end of a reception, it may be cleared while transmitting by reading the USART_SR register to check the TXE or TC flags and writing data to the data register. Consequently, the software receiver can read the PE flag as '0' even if a parity error occurred.

Workaround

The Parity Error flag should be checked after the end of reception and before transmission.

2.11.3 Parity Error (PE) flag is not set when receiving in Mute mode using address mark detection

Description

If the USART receiver is in Mute mode, and is configured to exit from Mute mode using the address mark detection, when the USART receiver recognizes a valid address with a parity error, it exits from Mute mode without setting the Parity Error flag.

变通方法

如果控制器设备允许，可通过清除I2C_CR1寄存器中的NOSTRETCH位来使用时钟扩展机制。

如果控制器设备不允许这样做，请确保在轮询TXE或ADDR标志时软件足够快速，以立即写入DR数据寄存器。例如，在TXE或ADDR标志上使用中断并将其优先级提升到更高优先级。

2.10.6 当VDD_I2C总线电压高于((VDD + 0.3)/0.7) V时，SDA和SCL的最大上升时间（tr）被违反。

描述

当外部传统I2C总线电压（VDD_I2C）设置为5 V，而微控制器（MCU）由VDD供电时，内部5伏容限电路在输入电压（VIN）达到VDD + 二极管阈值电平时立即被激活。额外的内部大电容随后防止外部上拉电阻（RP）在最大时间（tr）内使SDA和SCL信号上升，其中在快速模式下为300 ns，在标准模式下为1000 ns。上升时间（tr）是从VIL和VIH测量的，其电平分别设置为0.3 VDD_I2C和0.7 VDD_I2C。

变通方法

外部VDD_I2C总线电压必须限制在最大值为 ((VDD + 0.3) / 0.7) V。因此，当MCU由VDD = 3.3 V供电时，VDD_I2C不得超过5.14 V以符合I2C规范。

2.11 通用同步异步收发传输器

2.11.1 如果接收器时钟速度出现偏差，则空闲帧无法被检测到

描述

如果USART接收到一个空闲帧后接收到一个字符，并且发送设备的时钟比USART接收器的时钟快，那么在接收字符起始位时，USART接收信号会过早下降，导致空闲帧未被检测到（IDLE标志未被设置）。

变通方法

无。

2.11.2 在全双工模式下，可以通过向数据寄存器写入来清除奇偶校验错误（PE）标志。

描述

在全双工模式下，当接收器在接收结束后设置奇偶校验错误标志时，可以通过读取USART_SR寄存器检查TXE或TC标志，并向数据寄存器写入数据来清除该标志。因此，即使发生了奇偶校验错误，软件接收器仍可将PE标志读取为'0'。

变通方法

在接收结束后和发送前，应检查奇偶校验错误标志。

2.11.3 当在使用地址标记检测的静音模式下接收时，奇偶校验错误（PE）标志未被设置

描述

如果USART接收器处于静默模式，并且配置为通过地址标记检测退出静默模式，当USART接收器检测到带有奇偶校验错误的有效地址时，它退出静默模式而不设置奇偶校验错误标志。

Workaround

None.

2.11.4 Break frame is transmitted regardless of CTS input line status

Description

When the CTS hardware flow control is enabled (CTSE = 1) and the send break bit (SBK) is set, the transmitter sends a break frame at the end of the current transmission regardless of CTS input line status. Consequently, if an external receiver device is not ready to accept a frame, the transmitted break frame is lost.

Workaround

None.

2.11.5 RTS signal abnormally driven low after a protocol violation

Description

When RTS hardware flow control is enabled, the RTS signal goes high when data is received. If this data was not read and new data is sent to the USART (protocol violation), the RTS signal goes back to low level at the end of this new data.

Consequently, the sender gets the wrong information that the USART is ready to receive further data.

On the USART side, an overrun is detected, which indicates that data has been lost.

Workaround

A workaround is required only if the other USART device violates the communication protocol, which is not the case in most applications.

Two workarounds can be used:

- After data reception and before reading the data in the data register, the software takes over the control of the RTS signal as a GPIO, and holds it high as long as needed. If the USART device is not ready, the software holds the RTS pin high, and releases it when the device is ready to receive new data.
- Make sure the time required by the software to read the received data is always lower than the duration of the second data reception. For example, this can be ensured by handling all the receptions in DMA mode.

2.11.6 Start bit detected too soon when sampling for NACK signal from the smartcard

Description

According to ISO/IEC 7816-3 standard, when a character parity error is detected, the receiver shall transmit a NACK error signal 10.5 ± 0.2 ETUs after the character START bit falling edge. In this case, the transmitter is able to detect correctly the NACK signal until 11 ± 0.2 ETUs after the character START bit falling edge. In Smartcard mode, the USART peripheral monitors the NACK signal during the receiver time frame (10.5 ± 0.2 ETUs), while it should wait for it during the transmitter one (11 ± 0.2 ETUs). In real cases, this would not be a problem as the card itself needs to respect a 10.7 ETU period when sending the NACK signal. However, this may be an issue to undertake a certification.

Workaround

None.

变通方法

无。

2.11.4 布雷k帧无论在CTS输入线路的情况下都会被传输 状态

描述

当CTS硬件流控制被启用（CTSE = 1）且发送断位（SBK）被设置时，发送器会在当前传输结束时发送断帧，无论CTS输入线路状态如何。因此，如果外部接收设备未准备好接收帧，发送的断帧将被丢失。

变通方法

无。

2.11.5 转发信号在协议违规后异常被拉低 翻译

描述

当启用RTS硬件流控制时，接收到数据时，RTS信号变为高电平。如果此数据未被读取且新数据被发送到USART（协议违规），则在新数据结束时，RTS信号恢复为低电平。

因此，发送方会获得错误的信息，即USART已准备好接收更多数据。
在USART端检测到溢出，这表明数据已丢失。

变通方法

只有在其他USART设备违反通信协议的情况下才需要变通方法，这在大多数应用中并不存在。

有两种解决方法可以使用：

- 在数据接收后且在读取数据寄存器中的数据之前，软件将RTS信号作为GPIO接管控制，并保持其高电平，直到需要为止。如果USART设备未准备好，软件保持RTS引脚高电平，并在设备准备好接收新数据时释放它。
- 确保软件读取接收到的数据所需的时间始终低于第二次数据接收的持续时间。例如，可以通过DMA模式处理所有接收操作来实现这一点。

2.11.6 在采样智能卡的NACK信号时，过早检测到起始位

描述

根据ISO/IEC 7816-3标准，当检测到字符奇偶校验错误时，接收器应在字符START位下降沿后10.5 ± 0.2 ETUs传输NACK错误信号。在这种情况下，发送端能够正确检测到NACK信号，直到字符START位下降沿后11 ± 0.2 ETUs。在智能卡模式下，USART外设在接收器时间窗口（10.5 ± 0.2 ETUs）期间监控NACK信号，而在发送端时间窗口（11 ± 0.2 ETUs）期间则应等待该信号。在实际情况下，这不会成为问题，因为卡片本身在发送NACK信号时需要遵守10.7 ETU的时间段。然而，这可能成为进行认证的问题。

变通方法

无。

2.11.7 Break request can prevent the transmission complete flag (TC) from being set

Description

After the end of transmission of a data (D1), the transmission complete (TC) flag is not set if the following conditions are met:

- CTS hardware flow control is enabled,
- D1 is being transmitted,
- a break transfer is requested before the end of D1 transfer,
- CTS is de-asserted before the end of D1 data transfer.

Workaround

If the application needs to detect the end of a data transfer, check that the TC flag is set, and issue a break request.

2.11.8 Guard time not respected when data are sent on TXE events

Description

In Smartcard mode, when sending a data on TXE event, the programmed guard time is not respected, that is the data written in the data register is transferred to the bus without waiting the completion of the guard-time duration corresponding to the previous transmitted data.

Workaround

Since in Smartcard mode the TC flag is set at the end of the guard time duration, wait until TC is set, then write the data.

2.11.9 RTS is active while RE or UE = 0

Description

The RTS line is driven low as soon as the RTSE bit is set, even if the USART is disabled (UE = 0) or if the receiver is disabled (RE = 0) that is not ready to receive data.

Workaround

After setting the UE and RE bits, configure the I/O used for RTS as an alternate function.

2.12 SPI/I2S

2.12.1 BSY bit may stay high when SPI is disabled

Description

The BSY flag may remain high upon disabling the SPI while operating in:

- master transmit mode and the TXE flag is low (data register full).
- master receive-only mode (simplex receive or half-duplex bidirectional receive phase) and an SCK strobing edge has not occurred since the transition of the RXNE flag from low to high.
- slave mode and NSS signal is removed during the communication.

Workaround

When the SPI operates in:

- master transmit mode, disable the SPI when TXE = 1 and BSY = 0.
- master receive-only mode, ignore the BSY flag.
- slave mode, do not remove the NSS signal during the communication.

2.11 0.7 中断请求可以防止传输完成标志（TC）被设置

描述

在数据（D1）传输结束后，如果满足以下条件，传输完成（TC）标志不会被设置:

- CTS硬件流控制已启用，
- D1正在被传输，
- 请求中断传输在D1传输结束前，
- 在D1数据传输结束前，CTS被取消。

变通方法

如果应用程序需要检测数据传输的结束，请检查TC标志是否已设置，并发出中断请求。

2.11.8 当数据在TXE事件上发送时，保护时间未被遵守

描述

在智能卡模式下，当在TXE事件上发送数据时，编程的守卫时间未被遵守，即数据寄存器中写入的数据在未完成前一次传输数据对应的守卫时间持续时间的情况下，直接传输到总线上。

变通方法

由于在智能卡模式下，TC标志在保护时间持续时间结束时被置位，因此需要等待直到TC被置位，然后写入数据。

2.11.9 当RE或UE = 0时，RTS处于活动状态

描述

一旦RTSE位被设置，RTS线路就被拉低，即使USART被禁用（UE = 0）或接收器被禁用（RE = 0）且未准备好接收数据。

变通方法

设置之后 th UE和RE位，将用于RTS的I/O配置为alternate 激活函数.

2.12 串行外设接口/集成电路互连音频

二点一二一 当SPI被禁用时，BSY位可能保持高电平

描述

的 在禁用SPI时，BSY标志可能会保持高电平，即使在操作过程中正在输入:

- 主发送模式，且TXE标志位为低（数据寄存器满）。
- 主接收模式（单工接收或半双工双向接收阶段）且自RXNE标志从低到高转换以来，SCK触发边沿尚未发生。
- 从机模式和NSS信号在通信期间被移除。

变通方法

当SPI工作在:

- 主发送模式，当TXE = 1且BSY = 0时禁用SPI。
- 主设备只接收模式，忽略BSY标志。
- 从机模式，在通信期间不要移除NSS信号。

2.12.2 Anticipated communication upon SPI transit from slave receiver to master

Description

Regardless of the master mode configured, the communication clock starts upon setting the MSTR bit even though the SPI is disabled, if transiting from receive-only (RXONLY = 1) or half-duplex receive (BIDIMODE = 1 and BIDIOE = 0) slave mode to master mode.

Workaround

Apply one of the following measures:

- Before transiting to master mode, hardware-reset the SPI via the reset controller.
- Set the MSTR and SPE bits of the SPI configuration register simultaneously, which forces the immediate start of the communication clock. In transmitter configuration, load the data register in advance with the data to send.

2.12.3 Wrong CRC calculation when the polynomial is even

Description

When the CRC is enabled, the CRC calculation is wrong if the polynomial is even.

Workaround

Use odd polynomial.

2.12.4 Corrupted last bit of data and/or CRC received in Master mode with delayed SCK feedback

Description

When performing a receive transaction in I2S or SPI Master mode, the last bit of the transacted frame is not captured when the signal provided by an internal feedback loop from the SCK pin exceeds a critical delay. The lastly transacted bit of the stored data then keeps the value from the pattern received previously. As a consequence, the last receive data bit may be wrong, and/or the CRCERR flag can be unduly asserted in the SPI mode if any data under checksum, and/or just the CRC pattern is wrongly captured.

In SPI mode, data are synchronous with the APB clock. A delay of up to two APB clock periods can thus be tolerated for the internal feedback delay.

The I2S mode is more sensitive than the SPI mode, especially in the case where an odd I2S prescaler factor is set and the APB clock is the system clock divided by two. In this case, the internal feedback delay is lower than 1.5 APB clock period.

The main factors contributing to the delay increase are low V_{DD} level, high temperature, high SCK pin capacitive load, and low SCK I/O output speed. The SPI communication speed has no impact.

Workaround

The following workarounds can be adopted, jointly or individually:

- Decrease the APB clock speed.
- Configure the I/O pad of the SCK pin to be faster.

The following table gives the maximum allowable APB frequency (that still prevents the issue from occurring) versus GPIOx_OSPEEDR output speed for the SCK pin, with a 30 pF capacitive load.

Table 5. Maximum allowable APB frequency at 30 pF load

OSPEEDR [1:0] for SCK pin	Max. APB frequency for SPI mode (MHz)	Max. APB frequency for I2S mode (MHz)
11 (very high), 10 (high)	84	42
01 (medium)	75	35
00 (low)	25	16

2.12.2 预期的通信在SPI传输期间从从设备接收器到主设备

描述

无论配置的主模式是什么，通信时钟在设置MSTR位时启动，即使SPI处于禁用状态，如果从仅接收（RXONLY = 1）或半双工接收（BIDIMODE = 1 且 BIDIOE = 0）从机模式切换到主机模式。

变通方法

应用以下措施之一：

- 在切换到主模式之前，通过复位控制器对SPI进行硬件复位。
- 同时设置 {v*} 和 {v*} 位，这会强制通信时钟立即启动。在 {v*} 配置中，提前将待发送的数据加载到 {v*} 中。

2.12.3 当多项式为偶数时

描述

当CR C已启用，如果多项式，CRC计算错误。 是偶数

变通方法

使用奇多项式。

2.12.4 在主模式下带有延迟SCK反馈时接收到的损坏数据最后一位和/或循环冗余校验

描述

当在I2S或SPI主模式下执行接收事务时，如果由SCK引脚的内部反馈环提供的信号超过临界延迟，则事务帧的最后一位不会被捕获。存储数据的最后一位则保留之前接收到的模式的值。因此，最后接收的数据位可能错误，且/或在SPI模式下，如果任何校验数据，且/或仅CRC模式被错误捕获，则CRCERR标志可能被不当置位。

在SPI模式下，数据与APB时钟同步。因此，内部反馈延迟可以容忍最多两个APB时钟周期的延迟。

I2S模式比SPI模式更敏感，尤其是在设置奇数的I2S预分频系数且APB时钟为系统时钟除以二的情况下。在这种情况下，内部反馈延迟低于1.5个APB时钟周期。

导致延迟增加的主要因素包括低VDD电平、高温、高SCK引脚电容负载以及低SCK I/O输出速度。SPI通信速度没有影响。

变通方法

以下变通方法可以采用，联合或单独使用：

- 降低APB时钟速度。
- 配置SCK引脚的I/O垫以实现更快的速度。

下表给出了对于SCK引脚的GPIOx_OSPEEDR输出速度与最大允许的APB频率（仍可防止问题发生）相对，带30 pF电容负载。

表5. 最大允许APB频率在30 pF负载下 德

OSPEEDR [1:0] for SCK pin	Max. APB frequency for SPI mode (MHz)	Max. APB frequency for I2S mode (MHz)
11 (very high), 10 (high)	84	42
01 (medium)	75	35
00 (low)	25	16

2.12.5 BSY flag may stay high at the end of a data transfer in Slave mode

Description

The BSY flag may sporadically remain high at the end of a data transfer in Slave mode. The issue appears when an accidental synchronization happens between the internal CPU clock and the external SCK clock provided by the master.

This is related to the end of data transfer detection while the SPI is enabled in Slave mode.

As a consequence, the end of the data transaction may be not recognized when the software needs to monitor it (for example at the end of a session before entering the low-power mode or before the direction of the data line has to be changed at half duplex bidirectional mode). The BSY flag is unreliable to detect the end of any data sequence transaction.

Workaround

When the NSS hardware management is applied and the NSS signal is provided by the master, the end of a transaction can be detected by the NSS polling by the slave:

- If the SPI receiving mode is enabled, the end of a transaction with the master can be detected by the corresponding RXNE event signaling the last data transfer completion.
- In SPI transmit mode, the user can check the BSY under timeout corresponding to the time necessary to complete the last data frame transaction. The timeout must be measured from TXE event signaling the last data frame transaction start (it is raised once the second bit transaction is ongoing). Either BSY becomes low normally or the timeout expires when the synchronization issue happens.

When the above workarounds are not applicable, the following sequence can be used to prevent the synchronization issue during SPI transmit mode:

1. Write the last data to the data register.
2. Poll TXE until it becomes high to ensure the data transfer has started.
3. Disable SPI by clearing SPE while the last data transfer is still ongoing.
4. Poll the BSY bit until it becomes low.
5. The BSY flag works correctly and can be used to recognize the end of the transaction.

Note: This workaround can be used only when the CPU has enough performance to disable the SPI after a TXE event is detected, while the data frame transfer is still ongoing. It is impossible to achieve it when the ratio between CPU and SPI clock is low, and the data frame is short. In this specific case, the timeout can be measured from TXE, while calculating the fixed number of CPU clock periods corresponding to the time necessary to complete the data frame transaction.

2.12.6 In I2S Slave mode, the WS level must be set by the external master when enabling the I2S

Description

In Slave mode, the WS signal level is used only to start the communication. If the I2S (in Slave mode) is enabled while the master is already sending the clock and the WS signal level is low (for the I2S protocol) or is high (for the LSB or MSB-justified mode), the slave starts communicating data immediately. In this case, the master and slave are desynchronized throughout the whole communication.

Workaround

Make sure the I2S peripheral is enabled when the external master sets the WS line at:

- High level when the I2S protocol is selected.
- Low level when the LSB or MSB-justified mode is selected

2.12.7 I2S2 in full-duplex mode may not work properly when SCK and WS signals are mapped on PI1 and PI0, respectively

Description

When the SCK and WS signals are used to support the I2S full-duplex through the GPIO port PI1 and PI0, respectively, the I2S2 peripheral may not be able to provide internally the SCK and WS signals to the I2S2_ext interface. In this case, the I2S2_ext interface is not able to send/receive data.

2.12.5 BSY标志可能在数据传输结束后保持高电平

描述

BSY标志可能在从机模式的数据传输结束时偶尔保持高电平。该问题出现在内部CPU时钟与主设备提供的外部SCK时钟之间发生意外同步时。

这与SPI在从机模式下启用时的数据传输结束检测相关。
因此，当软件需要监控数据事务结束时，可能无法识别该结束点（例如在进入低功耗模式前的会话结束时，或在半双工双向模式下需要改变数据线方向之前）。BSY标志无法可靠检测任何数据序列传输的结束。

变通方法

当应用NSS硬件管理且NSS信号由主设备提供时，从设备可以通过NSS轮询检测到事务的结束：

- 当SPI接收模式被启用时，可以通过相应的RXNE事件检测与主设备的数据传输结束，该事件指示最后的数据传输完成。
- 在SPI发送模式中，用户可以在对应于完成最后一个数据帧事务所需的时间内检查BSY。超时必须从TXE事件开始测量，该事件标志着最后一个数据帧事务的开始（当第二个位事务正在进行时触发）。当发生同步问题时，BSY正常变为低电平或超时到期。{v*}

当上述变通方法不可行时，可以使用以下步骤以防止在SPI传输模式下的同步问题：

1. 将最后的数据写入数据寄存器。2. 轮询 TXE 直到其变为高电平，以确保数据传输已启动。3. 在最后的数据传输仍在进行时，通过清除 SPE 来禁用 SPI。4. 轮询 BSY 位直到其变为低电平。5. BSY 标志工作正常，可用于识别事务的结束。

Note: This workaround can be used only when the CPU has enough performance to disable the SPI after a TXE event is detected, while the data frame transfer is still ongoing. It is impossible to achieve it when the ratio between CPU and SPI clock is low, and the data frame is short. In this specific case, the timeout can be measured from TXE, while calculating the fixed number of CPU clock periods corresponding to the time necessary to complete the data frame transaction.

2.12.6 在I2S从机模式下，当启用I2S时，WS电平必须由外部主设备设置。

描述

在从机模式下，WS信号电平仅用于启动通信。如果在从机模式下启用I2S（当主设备正在发送时钟信号且WS信号电平为低（适用于I2S协议）或为高（适用于LSB或MSB对齐模式）时，从设备立即开始传输数据。在这种情况下，主设备和从设备在整个通信过程中不同步。

变通方法

请确保当外部主设备设置WS线时，I2S外设已启用：

- 当选择I2S协议时，处于高电平。
- 当选择LSB或MSB对齐模式时，低电平

2.12.7 当 SCK 和 WS 信号分别映射到 PI1 和 PI0 时，I2S2 在全双工模式下可能无法正常工作。

描述

当SCK和WS信号通过GPIO端口PI1和PI0分别用于支持I2S全双工时，I2S2外设可能无法将SCK和WS信号内部提供给I2S2_ext接口。在这种情况下，I2S2_ext接口无法进行数据的收发。

Workaround

Use other mapped pins for SCK and WS signals:

- I2S2 CK signal: PB10 or PB13 pin.
- I2S2 WS signal: PB12 or PB9 pin.

2.13 bxCAN

2.13.1 bxCAN time-triggered communication mode not supported

Description

The time-triggered communication mode described in the reference manual is not supported. As a result, timestamp values are not available. The TTCM bit of the CAN_MCR register must be kept cleared (time-triggered communication mode disabled).

Workaround

None.

2.14 OTG_FS

2.14.1 Transmit data FIFO is corrupted when a write sequence to the FIFO is interrupted with accesses to certain OTG_FS registers

Description

When the USB on-the-go full-speed peripheral is in Device mode, interrupting transmit FIFO write sequence with read or write accesses to OTG_FS endpoint-specific registers (those ending in *0* or *x*) leads to corruption of the next data written to the transmit FIFO.

Workaround

Ensure that the transmit FIFO write sequence is not interrupted with accesses to the OTG_FS registers.

2.14.2 Host packet transmission may hang when connecting through a hub to a low-speed device

Description

When the USB on-the-go full-speed peripheral connects to a low-speed device via a hub, the transmitter internal state machine may hang. This leads, after a timeout expiry, to a port disconnect interrupt.

Workaround

None. However, increasing the capacitance on the data lines may reduce the occurrence.

2.14.3 Data in RxFIFO is overwritten when all channels are disabled simultaneously

Description

If the available RxFIFO is just large enough to host one packet plus its data status, and it is currently occupied by the last received data plus its status, and, at the same time, the application requests that more IN channels are disabled, the OTG_FS peripheral does not first check for available space before inserting the disabled status of the IN channels. It just inserts them by overwriting the existing data payload.

变通方法

使用其他已映射的引脚用于 SCK 和 WS 信号：

- I2S2 时钟信号：PB10 或 PB13 引脚。
- I2S2 从机信号：PB12 或 PB9 引脚。

2.13 bxCAN

2.13.1 bxCAN 时间触发的通信模式不支持

描述

参考手册中描述的时间触发通信模式不被支持。因此，时间戳值不可用。CAN_MCR寄存器的TTCM位必须保持清除状态（时间触发通信模式禁用）。

变通方法

无。

2.14 OTG_FS

2.14.1 当向FIFO写入的序列被中断且在访问某些OTG_FS寄存器时，发送数据FIFO会损坏

描述

当USB On-The-Go全速外设处于设备模式时，通过读取或写入OTG_FS端点专用寄存器（以*0*或*x*结尾的）中断发送FIFO的写序列会导致后续写入发送FIFO的数据损坏。

变通方法

确保在访问OTG_FS寄存器时，传输FIFO写入序列不会被中断。

2.14.2 主机数据包传输可能在通过集线器连接到低速设备时卡住

描述

当USB On-The-Go全速外设通过集线器连接到低速设备时，发送器内部状态机可能卡死。这会导致在超时后触发端口断开中断。

变通方法

无。然而，增加数据线上的电容可能会减少发生率。

2.14.3 当所有通道同时被禁用时，RxFIFO中的数据会被覆盖

描述

如果可用的RxFIFO刚好足以容纳一个数据包及其状态数据，并且当前正被最后接收到的数据及其状态所占用，同时应用程序请求禁用更多的IN通道，则OTG_FS外设在插入IN通道的禁用状态之前不会首先检查可用空间。它只是通过覆盖现有数据负载来插入这些状态。

Workaround

Use one of the following measures:

- Configure the RxFIFO to host a minimum of 2 × MPSIZ + 2 × data status entries.
- Check the RXFLVL bit (RxFIFO nonempty) of the OTG_FS_GINTSTS register before disabling each IN channel. If this bit is cleared, then disable an IN channel at a time. Each time the application disables an IN channel, it first has to check that the RXFLVL bit = 0 condition is true.

2.14.4 OTG host blocks the receive channel when receiving IN packets and no TxFIFO is configured

Description

When receiving data, the OTG_FS core erroneously checks for available TxFIFO space when it should only check for RxFIFO space. If the OTG_FS core cannot see any space allocated for data transmission, it blocks the reception channel, and no data is received.

Workaround

Set at least one TxFIFO equal to the maximum packet size. In this way, the host application, which intends to supports only IN traffic, also has to allocate some space for the TxFIFO.

Since a USB host is expected to support any kind of connected endpoint, it is good practice to always configure enough TxFIFO space for OUT endpoints.

2.14.5 Host channel-halted interrupt not generated when the channel is disabled

Description

When the application enables then immediately disables the host channel before the OTG_FS host has had time to begin the transfer sequence, the OTG_FS core, as a host, does not generate a channel-halted interrupt. The OTG_FS core continues to operate normally.

Workaround

Do not disable the host channel immediately after enabling it.

2.14.6 Wrong software-read OTG_FS_DCFG register values

Description

When the application writes to the DAD and PFIVL bitfields of the OTG_FS_DCFG register, and then reads the newly written bitfield values, the read values may not be correct.

However, the values written by the application are correctly retained by the core, and the normal operation of the device is not affected.

Workaround

Do not read the OTG_FS_DCFG register DAD and PFIVL bitfields just after programming them.

2.15 OTG_HS

2.15.1 Transmit data FIFO is corrupted when a write sequence to the FIFO is interrupted with accesses to certain OTG_HS registers

Description

When the USB on-the-go high-speed peripheral is in Device mode, interrupting transmit FIFO write sequence with read or write accesses to OTG_HS endpoint-specific registers (those ending in *0* or *x*) leads to corruption of the next data written to the transmit FIFO.

变通方法

使用以下任一方法:

- 配置RxFIFO以支持至少2 × MPSIZ + 2 × 数据状态条目。
- 在禁用每个IN通道之前，检查OTG_FS_GINTSTS寄存器中的RXFLVL位（RxFIFO非空）。如果该位被清除，则每次仅禁用一个IN通道。每次应用程序禁用一个IN通道时，必须首先检查RXFLVL位=0条件是否为真。

2.14. 4 OTG主机在接收IN数据包时阻塞接收通道，且未配置TxFIFO

描述

当接收数据时，OTG_FS核心错误地检查可用的TxFIFO空间，而它本应仅检查RxFIFO空间。如果OTG_FS核心无法检测到已分配用于数据传输的空间，它将阻塞接收通道，并且无法接收数据。

变通方法

将至少一个 TxFIFO 设置为最大数据包大小。通过这种方式，旨在仅支持 IN 通信的主机应用程序也必须为 TxFIFO 分配一些空间。

由于USB主机需要支持任何连接的端点，因此应始终为OUT端点配置足够的TxFIFO空间。

2.14.5 当通道被禁用时，主机不会生成通道停止中断。

描述

当应用程序启用主机通道后立即将其禁用，在OTG_FS主机有机会开始传输序列之前，OTG_FS核心作为主机不会生成通道停止中断。OTG_FS核心继续正常运行。

变通方法

不要在启用主机通道后立即禁用它。

2.14.6 错误的软件读取 OTG_FS_DCFG 寄存器值 是

描述

当应用程序向OTG_FS_DCFG寄存器的DAD和PFIVL位字段写入数据，并随后读取新写入的位字段值时，读取的值可能不正确。

然而，应用程序写入的值被核心正确保留，且设备的正常运行不会受到影响。

变通方法

不要阅读 O TG_FS_DCFG寄存器的DAD和PFIVL位字段紧接着pr 编程它们。

2.15 OTG高速模式

2.15.1 当向FIFO进行写入操作时，如果写入序列在访问某些OTG_HS寄存器时被中断，则传输数据FIFO将被破坏。

描述

当USB On-The-Go高速外设处于设备模式时，中断传输FIFO写入序列的读或写访问会导致下一个写入传输FIFO的数据被破坏。

Workaround

Ensure that the transmit FIFO write sequence is not interrupted with accesses to the OTG_HS registers. Note that enabling DMA mode guarantees this.

2.15.2 Host packet transmission may hang when connecting the full speed interface through a hub to a low-speed device

Description

When the USB on-the-go high-speed peripheral is used with the full speed interface (DM and DP pins, N.B. not available on all devices), and connects to a low-speed device via a hub, the transmitter internal state machine may hang. This leads, after a timeout expiry, to a port disconnect interrupt.

Workaround

None. However, increasing the capacitance on the data lines may reduce the occurrence.

2.16 ETH

2.16.1 Incorrect L3 checksum is inserted in transmitted IPv6 packets without TCP, UDP or ICMP payloads

Description

The application provides a frame-by-frame control to instruct the MAC to insert the layer 3 (L3) checksums for TCP, UDP and ICMP packets. When automatic checksum insertion is enabled and the input packet is an IPv6 packet without the TCP, UDP or ICMP payload, then the MAC may incorrectly insert a checksum into the packet. For IPv6 packets without a TCP, UDP or ICMP payload, the MAC core considers the next header (NH) field as the extension header and continues to parse the extension header. Sometimes, the payload data in such packets matches the NH field for TCP, UDP or ICMP and, as a result, the MAC core inserts a checksum.

Workaround

When the IPv6 packets have a TCP, UDP or ICMP payload, enable checksum insertion for transmit frames, or bypass checksum insertion by using the CIC bits of the TDES0 transmit descriptor word0.

2.16.2 The Ethernet MAC processes invalid extension headers in the received IPv6 frames

Description

In IPv6 frames, the extension headers which precede the actual IP payload may or may not be present. The Ethernet MAC processes the following extension headers defined in the IPv6 protocol: hop-by-hop options header, routing header and destination options header.

All extension headers, except the hop-by-hop extension header, can be present multiple times and in any order before the actual IP payload. The hop-by-hop extension header, if present, has to come immediately after the IPv6 main header.

The Ethernet MAC processes all extension headers whether valid or invalid including the hop-by-hop extension headers that are present after the first extension header. For this reason, the GMAC core accepts IPv6 frames with invalid hop-by-hop extension headers. As a consequence, it accepts any IP payload as valid IPv6 frames with TCP, UDP or ICMP payload, and then incorrectly update the receive status of the corresponding frame.

Workaround

None.

变通方法

确保在访问OTG_HS寄存器时，发送FIFO的写序列不会被中断。请注意，启用DMA模式可确保这一点。

2.15.2 当通过集线器将全速接口连接到低速设备时，主机数据包传输可能会卡住。

描述

当使用USB On-The-Go高速外设与全速接口（DM和DP引脚，注：并非所有设备都提供这些引脚）时，若通过集线器连接到低速设备，发送器内部状态机可能卡死。这会导致在超时后产生端口断开中断。

变通方法

无。然而，增加数据线上的电容可能会减少发生率。

2.16 以太坊

第2.16.1条 在传输的IPv6数据包中插入了错误的第三层校验和，且这些数据包不含TCP、UDP或ICMP有效载荷

描述

该应用提供逐帧控制以指示MAC插入TCP、UDP和ICMP数据包的第三层（L3）校验和。当启用自动校验和插入功能且输入数据包为不含TCP、UDP或ICMP负载的IPv6数据包时，MAC可能错误地将校验和插入到数据包中。对于不含TCP、UDP或ICMP负载的IPv6数据包，MAC核心将下一报头（NH）字段视为扩展报头并继续解析扩展报头。有时，此类数据包中的负载数据与TCP、UDP或ICMP的NH字段匹配，因此MAC核心会插入校验和。

变通方法

当IPv6数据包具有TCP、UDP或ICMP有效载荷时，启用发送帧的校验和插入，或通过使用TDES0发送描述符字0中的CIC位来绕过校验和插入。

2.16.2 以太网MAC处理接收的IPv6帧中的无效扩展头

描述

在IPv6帧中，位于实际IP有效载荷之前的扩展头部可能有也可能没有。以太网MAC处理以下IPv6协议中定义的扩展头部：逐跳选项头部、路由头部和目标选项头部。

所有扩展头，除了逐跳扩展头，都可以在实际IP负载之前多次出现且顺序任意。如果存在逐跳扩展头，它必须紧接在IPv6主头之后。

以太网MAC处理所有扩展头，无论其是否有效，包括位于第一个扩展头之后的跳跃式扩展头。因此，GMAC核心接受包含无效跳跃式扩展头的IPv6帧。结果，它将任何IP有效载荷视为有效的IPv6帧（带有TCP、UDP或ICMP有效载荷），然后错误地更新相应帧的接收状态。

变通方法

无。

2.16.3 MAC stuck in the idle state on receiving the TxFIFO flush command exactly one clock cycle after a transmission completes

Description

When the software issues a TxFIFO flush command, the transfer of frame data stops, even in the middle of a frame transfer. The TxFIFO read controller goes into the Idle state by clearing the TFRS [1:0] bit field of the ETH_MACDBGR register. It then resumes its normal operation.

However, if the TxFIFO read controller receives the TxFIFO flush command exactly one clock cycle after receiving the status from the MAC, the controller remains stuck in the Idle state and stops transmitting frames from the TxFIFO. The system only recovers from this state with a reset (for example a soft reset).

Workaround

Wait until the TxFIFO is empty before using the TxFIFO flush command.

2.16.4 Transmit frame data corruption

Description

Frame data may get corrupted when the TxFIFO repeatedly switches from non-empty to empty, and back to non-empty again for a very short period, without causing any underflow.

The issue occurs when switching back and forth between non-empty and empty happens when the rate the data is being written to the TxFIFO is almost equal to or a little slower than the rate at which the data is read.

This corruption cannot be detected by the receiver when the CRC is inserted by the MAC, as the corrupted data is used for the CRC computation.

Workaround

Use the transmit Store-and-Forward mode by setting the TSF bit of the ETH_DMAOMR register. In this mode, the data is transmitted only when the whole packet is available in the TxFIFO.

2.16.5 Incorrect status and corrupted frames when RxFIFO overflow occurs on the penultimate word of Rx frames

Description

When operating in Threshold mode, the RxFIFO may overflow when the received frame data is written faster than the speed at which the application reads it from the RxFIFO. The RxFIFO overflow is declared at the moment that a non-EOF word is received and the RxFIFO has only two locations available. The receiver descriptor overflow error (OE) bit of the RDES0 receive descriptor word0 is set to indicate that the receive frame is incomplete.

The problem occurs after the following events:

1. RxFIFO overflow is declared exactly on the penultimate word of the Rx Frame. The EOF word is received in the next clock cycle.
2. The EOF word has exactly one valid byte. This is possible only when the length of the packet, after CRC or PAD stripping (if enabled), is a multiple of 4 bytes plus 1 (for example, 5, 9, 13, 17).

After the above sequence, the frames status information is corrupted and the overflow error flag is not set. Furthermore, if the next frame arrives soon enough, the MAC might falsely interpret that there is space in the RxFIFO and overwrite unread data with the next frame, thus corrupting the existing frames.

The MAC recovers automatically after transferring a few corrupt or incorrect packets.

Workaround

Operate the RxFIFO in the Store-and-Forward mode.

二.十六.三　在传输完成后正好一个时钟周期，MAC在接收TxFIFO刷新命令时卡在空闲状态

描述

当软件发出TxFIFO刷新命令时，帧数据传输会停止，即使在帧传输过程中。TxFIFO读取控制器通过清除ETH_MACDBGR寄存器中的TFRS [1:0]位字段进入空闲状态。然后恢复其正常操作。

然而，如果TxFIFO读取控制器在从MAC接收到状态后恰好一个时钟周期接收到TxFIFO刷新命令，则控制器会卡在空闲状态并停止从TxFIFO传输帧。系统只能通过复位（例如软复位）从该状态中恢复。

变通方法

W等待直到TxFIFO为空，在使用TxFIFO刷新命令之前　　命令.

2.16.4　传输帧数据损坏

描述

当TxFIFO反复在非空和空之间切换，并在极短时间内再次切换回非空状态而不会导致任何下溢时，帧数据可能会损坏。
当在非空和空状态之间来回切换时，如果数据写入TxFIFO的速率几乎等于或略低于数据读取速率，则会出现问题。
当MAC插入CRC时，接收方无法检测到这种错误，因为被损坏的数据用于CRC计算。

变通方法

通过设置ETH_DMAOMR寄存器中的TSF位来使用传输存储转发模式。在此模式下，数据仅在TxFIFO中存在完整数据包时才被传输。

2.16.5　当RxFIFO溢出发生在Rx帧的倒数第二个字处时，会出现错误的状态和损坏的帧

描述

当处于阈值模式时，如果接收到的数据帧写入速度超过应用程序从RxFIFO读取的速度，RxFIFO可能会溢出。当接收到一个非EOF字且RxFIFO仅剩两个可用位置时，RxFIFO溢出被声明。RDES0接收描述符字0中的接收描述符溢出错误（OE）位被置位，以指示接收的数据帧不完整。

该问题发生在以下事件之后：
1. 接收FIFO溢出仅在接收帧的倒数第二个字上声明。帧结束字在下一个时钟周期接收。2. 帧结束字恰好有一个有效字节。这仅在数据包长度（在CRC或填充剥离（若启用）之后）为4字节的整数倍加1时可能发生（例如，5, 9, 13, 17）。

在上述序列之后，帧状态信息被破坏，且溢出错误标志未被设置。此外，如果下一帧及时到达，MAC可能会错误地认为RxFIFO中有空间，并用下一帧覆盖未读数据，从而破坏现有帧。在传输几个损坏或错误的数据包后，MAC会自动恢复。

变通方法

Operate the RxFIFO in the Store-and-Forward mode.

2.16.6 Successive write operations to the same register might not be fully taken into account

Description

A write to a register might not be fully taken into account if a previous write to the same register is performed within a time period of four TX_CLK/RX_CLK clock cycles. When this error occurs, reading the register returns the most recently written value, but the Ethernet MAC continues to operate as if the latest write operation never occurred.

Refer to the following table for the registers and bits impacted by this limitation.

Impacted registers and bits

Register name	Bit number	Bit name
DMA registers		
ETH_DMABMR	7	EDFE
ETH_DMAOMR	26	DTCEFD
	25	RSF
	20	FTF
	7	FEF
	6	FUGF
	4:3	RTC
GMAC registers		
ETH_MACCR	25	CSTF
	23	WD
	22	JD
	19:17	IFG
	16	CSD
	14	FES
	13	ROD
	12	LM
	11	DM
	10	IPCO
	9	RD
	7	APCS
	6:5	BL
	4	DC
	3	TE
	2	RE
ETH_MACFFR MAC	-	frame filter register
ETH_MACHTHR	31:0	Hash Table High Register
ETH_MACHTLR	31:0	Hash Table Low Register
ETH_MACFCR	31:16	PT
	7	ZQPD
	5:4	PLT
	3	UPFD
	2	RFCE
	1	TFCE

2.16.6 连续的写入操作到同一寄存器可能不会被完全考虑

描述

如果在四个TX_CLK/RX_CLK时钟周期内对同一寄存器执行了先前的写入操作，则对寄存器的写入可能不会被完全考虑。当此错误发生时，读取寄存器会返回最近一次写入的值，但以太网MAC会继续运行，仿佛最近的写入操作从未发生过。

参见 th　下表列出了受此限制影响的寄存器和位　引用
受影响的寄存器和位

Register name	Bit number	Bit name
DMA registers		
ETH_DMABMR	7	EDFE
ETH_DMAOMR	26	DTCEFD
	25	RSF
	20	FTF
	7	FEF
	6	FUGF
	4:3	RTC
GMAC registers		
ETH_MACCR	25	CSTF
	23	WD
	22	JD
	19:17	IFG
	16	CSD
	14	FES
	13	ROD
	12	LM
	11	DM
	10	IPCO
	9	RD
	7	APCS
	6:5	BL
	4	DC
	3	TE
	2	RE
ETH_MACFFR MAC	-	frame filter register
ETH_MACHTHR	31:0	Hash Table High Register
ETH_MACHTLR	31:0	Hash Table Low Register
ETH_MACFCR	31:16	PT
	7	ZQPD
	5:4	PLT
	3	UPFD
	2	RFCE
	1	TFCE

Register name	Bit number	Bit name
ETH_MACFCR	0	FCB/BPA
ETH_MACVLANTR	16	VLANTC
	15:0	VLANTI
ETH_MACRWUFFR	-	all remote wakeup registers
ETH_MACPMTCSR	31	WFFRPR
	9	GU
	2	WFE
	1	MPE
	0	PD
ETH_MACA0HR	-	MAC address 0 high register
ETH_MACA0LR	-	MAC address 0 low register
ETH_MACA1HR	-	MAC address 1 high register
ETH_MACA1LR	-	MAC address 1 low register
ETH_MACA2HR	-	MAC address 2 high register
ETH_MACA2LR	-	MAC address 2 low register
ETH_MACA3HR	-	MAC address 3 high register
ETH_MACA3LR	-	MAC address 3 low register
IEEE 1588 time stamp registers		
ETH_PTPTSCR	18	TSPFFMAE
	17:16	TSCNT
	15	TSSMRME
	14	TSSEME
	13	TSSIPV4FE
	12	TSSIPV6FE
	11	TSSPTPOEFE
	10	TSPTPPSV2E
	9	TSSSR
	8	TSSARFE
	5	TSARU
	3	TSSTU
	2	TSSTI
	1	TSFCU
	0	TSE

Workaround

Applty on of the following measures:

- Ensure a delay of four TX_CLK/RX_CLK clock cycles between the successive write operations to the same register.
- Make several successive write operations without delay, then read the register when all the operations are complete, and finally reprogram it after a delay of four TX_CLK/RX_CLK clock cycles.

Register name	Bit number	Bit name
ETH_MACFCR	0	FCB/BPA
ETH_MACVLANTR	16	VLANTC
	15:0	VLANTI
ETH_MACRWUFFR	-	all remote wakeup registers
ETH_MACPMTCSR	31	WFFRPR
	9	GU
	2	WFE
	1	MPE
	0	PD
ETH_MACA0HR	-	MAC address 0 high register
ETH_MACA0LR	-	MAC address 0 low register
ETH_MACA1HR	-	MAC address 1 high register
ETH_MACA1LR	-	MAC address 1 low register
ETH_MACA2HR	-	MAC address 2 high register
ETH_MACA2LR	-	MAC address 2 low register
ETH_MACA3HR	-	MAC address 3 high register
ETH_MACA3LR	-	MAC address 3 low register
IEEE 1588 time stamp registers		
ETH_PTPTSCR	18	TSPFFMAE
	17:16	TSCNT
	15	TSSMRME
	14	TSSEME
	13	TSSIPV4FE
	12	TSSIPV6FE
	11	TSSPTPOEFE
	10	TSPTPPSV2E
	9	TSSSR
	8	TSSARFE
	5	TSARU
	3	TSSTU
	2	TSSTI
	1	TSFCU
	0	TSE

变通方法

应用以下措施之一:

- 确保在对同一寄存器进行连续写操作之间，延迟四个 TX_CLK/RX_CLK 时钟周期。
- 连续执行多个写操作而不延迟，然后在所有操作完成时读取寄存器，最后在延迟四个 {v*} 时钟周期后重新编程。

2.16.7 Incorrect remote wakeup on global unicast packet

Description

The PMT remote wakeup block can be enabled to generate remote wakeup interrupt on receiving a global unicast packet, for example a unicast destination address (DA) that perfectly matches one of the MAC address registers enabled for DA.

However, the PMT remote wakeup block generates interrupts for any unicast packet that passes the DA filter (not necessarily for a unicast packet that passes DA perfect filter). For example, the address filter is set for inverse filtering or hash filtering, it gives a PASS for any packet whose DA is accepted by the inverse/hash filter. This results in power down exit on unintended received packets.

Workaround

Only enable DA perfect filter when global unicast based remote wakeup is enabled.

2.16.8 Overflow status bits of missed frame and buffer overflow counters are cleared without a read operation

Description

The direct memory access (DMA) controller maintains two counters to track the number of frames missed because of the following events:

- Rx descriptor not being available
- Rx FIFO buffer overflow during reception

The missed frame and buffer overflow counter of the ETH_DMAMFBOCR register indicate the current value of the missed frames and FIFO overflow frame counters. This register also has the overflow status bits: OFOC bit OMFC bit which indicate whether the rollover occurred for the respective counters and are set when the respective counters roll over. They should remain set until this register is read.

However, when the counter rollover occurs a second time after the status bit is set, the respective status bits are cleared. Therefore, the application may incorrectly detect that the rollover did not occur since the last read operation.

Workaround

The application should read the missed frame and buffer overflow counter register periodically (or after the overflow or rollover status bits are set) such that the counter rollover does not occur twice between read operations.

2.16.9 MAC may provide incorrect Rx status for the MAC control frames when receive checksum offload is enabled

Description

The MAC can be programmed to forward the MAC control frames (with length/type field 0x8808) to the application by setting the PCF bits of the ETH_MACFFR register. When the IPv4 checksum offload function is enabled by setting the IPCO bit of the ETH_MACCR register and clearing the EDFE bit of the ETH_DMABMR register, the MAC provides the encoded Rx status in the FT bit, IPHCE/TSV bit, and the PCE/ESA bit of the RDES0 receive descriptor word0. When a MAC control frame is received, the MAC should provide 0b011 in RDES0 receive descriptor word0 of Rx status. This indicates that an Ethernet type frame is received, which is neither IPv4 nor IPv6 (the checksum offload engine bypasses the checksum completely).

However, when a MAC control frame is received with IPv4 checksum offload function enabled without the extended status, the MAC provides incorrect Rx status (0b100) in RDES0 receive descriptor word0, indicating that an IPv4 or IPv6 type frame is received with no checksum error.

Workaround

Ignore these status bits, as reported in the RDES0 receive descriptor word0, about IP header error after it identifies the control packet from the received packet and processes it accordingly.

2.16.7 错误的远程唤醒在全局单播数据包上

描述

PMT远程唤醒模块可以启用，以在接收到全局单播数据包时生成远程唤醒中断，例如一个单播目标地址（DA）完全匹配已启用用于DA的MAC地址寄存器之一。

然而，PMT远程唤醒模块会为任何通过DA过滤器的单播数据包生成中断（并非一定针对通过DA完美过滤器的单播数据包）。例如，当地址过滤器设置为逆向过滤或哈希过滤时，它会对任何DA被逆向/哈希过滤器接受的数据包给出PASS。这会导致在意外接收的数据包上发生电源关闭退出。

变通方法

仅在基于全局单播的远程唤醒启用时启用DA完美过滤器。

二.十六.八 丢失帧和缓冲区溢出计数器的溢出状态位被清除，无需读取操作

描述

直接内存访问（DMA）控制器维护两个计数器，以记录丢失的帧数，由于以下事件：

- 接收描述符不可用
- 接收过程中FIFO缓冲区溢出

ETH_DMAMFBOCR寄存器的丢失帧和缓冲区溢出计数器指示当前的丢失帧和FIFO溢出帧计数器的值。该寄存器还包含溢出状态位：OFOC位、OMFC位，用于指示相应计数器是否发生溢出，并在相应计数器溢出时设置。这些状态位应保持设置状态，直到读取该寄存器。

然而，当状态位被置位后计数器溢出第二次发生时，相应的状态位被清除。因此，应用程序可能会错误地检测计数器溢出并未在上一次读取操作后发生。

变通方法

应用程序应定期读取丢失的帧和缓冲区溢出计数器寄存器（或在溢出或滚出状态位被设置后），以确保在两次读取操作之间不会发生计数器溢出。

2.16.9 MAC 可能为 MAC 控制帧提供错误的接收状态，当启用接收校验和卸载时

描述

MAC 可以被编程以将 MAC 控制帧（具有长度/类型字段 0x8808）转发到应用程序
通过设置ETH_MACFFR寄存器的PCF位。当通过设置ETH_MACCR寄存器的IPCO位并清除ETH_DMABMR寄存器的EDFE位来启用IPv4校验和卸载功能时，MAC会在RDES0接收描述符字0的FT位、IPHCE/TSV位以及PCE/ESA位中提供编码的接收状态。当接收到MAC控制帧时，MAC应在RDES0接收描述符字0的接收状态中提供0b011。这表明接收到的以太网类型帧既不是IPv4也不是IPv6（校验和卸载引擎完全绕过校验和）。

然而，当启用IPv4校验和卸载功能但未启用扩展状态时，MAC在RDES0接收描述符字0中提供错误的Rx状态（0b100），表明已接收无校验和错误的IPv4或IPv6类型帧。

变通方法

忽略这些状态位，如在RDES0接收描述符字0中报告的，关于在识别出控制数据包并进行相应处理后IP头部错误的情况。

2.16.10 MAC may provide an inaccurate Rx status when receive checksum offload is enabled in cut-through mode

Description

The MAC can be programmed in cut-through mode by resetting the DTCEFD bit of the ETH_DMAOMR register. When IPv4 checksum offload function is enabled by setting the IPCO bit of ETH_MACCR register and extended status is not enabled, the MAC provides the bit 25 (error summary), bit 14 (descriptor error), bit 11 (overflow error), bit 7 (IPC checksum error), bit 6 (late collision), bit 4 (watchdog error), bit 3 (receive error) and bit 0 (payload checksum error) of the RDES0: Receive descriptor Word0.

However, when a frame with a payload checksum error is received with the IPv4 checksum offload function enabled without the extended status, the MAC provides a correct error summary status but an incorrect Rx status on bit 0, indicating the IPv4 or IPv6 type frame is received with no checksum error. The source of the error is not specified, none of the individual status bits are set.

Workaround

Enable the Store-and-Forward mode by setting the RSF bit of the ETH_DMAOMR register. In this case the faulty frames are silently discarded as expected.

2.16.11 MAC may not drop received giant error frames

Description

The MAC considers a received frame with a length of more than 1522 bytes, as a giant error frame. When RxFIFO is operating in the Store-and-Forward mode and is programmed to drop error frames, by resetting the FEF bit of the ETH_DMAOMR, all error frames should be dropped in the FIFO layer. Due to the error frames being dropped in the FIFO layer, the DMA controller does not send them to the host.

However, the MAC does not drop the giant error frames and the DMA controller unnecessarily wastes system bandwidth transferring the giant frame to the host.

Workaround

The software driver must check the FL bits of the RDES0 receive descriptor word0, ignore or drop the frame and not forward it to the upper layer.

The frame length field is valid only when the LS bit of the RDES0 receive descriptor word0 register is set and the DE bit of the RDES0 receive descriptor word0 is reset.

2.16.10 媒体访问控制（MAC）在直通模式下启用接收校验和卸载时，可能会提供不准确的接收状态。

描述

MAC 可以通过重置 ETH_DMAOMR 寄存器的 DTCEFD 位来以直通模式编程。当通过设置 ETH_MACCR 寄存器的 IPCO 位启用 IPv4 校验和卸载功能且扩展状态未启用时，MAC 提供 RDES0：接收描述符 Word0 的位 25（错误汇总）、位 14（描述符错误）、位 11（溢出错误）、位 7（IPC 校验和错误）、位 6（晚冲突）、位 4（看门狗错误）、位 3（接收错误）和位 0（负载校验和错误）。

然而，当接收到一个带有有效载荷校验和错误的帧，且在未启用扩展状态的情况下启用了IPv4校验和卸载功能时，MAC提供正确的错误摘要状态，但在第0位上指示错误的接收状态，表明接收到的IPv4或IPv6类型帧无校验和错误。错误源未被指定，各个状态位均未被设置。

变通方法

通过设置ETH_DMAOMR寄存器的RSF位来启用存储转发模式。在这种情况下，故障帧如预期般被静默丢弃。

2.16.11 MAC层可能不会丢弃接收到的巨型错误帧

描述

MAC将接收到的长度超过1522字节的帧视为巨帧错误帧。当RxFIFO以存储转发模式运行且被配置为丢弃错误帧时，通过重置ETH_DMAOMR中的FEF位，所有错误帧应在FIFO层被丢弃。由于错误帧在FIFO层被丢弃，DMA控制器不会将它们发送到主机。

然而，MAC不丢弃巨型错误帧，而DMA控制器在将巨型帧传输到主机时无谓地浪费系统带宽。

变通方法

软件驱动程序必须检查RDES0接收描述符字0的FL位，忽略或丢弃该帧，并不将其转发至上层。

帧长度字段仅在RDES0接收描述符字0寄存器的LS位被置位且DE位被清除时有效。

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.

重要安全公告

意法半导体公司集团（ST）高度重视产品安全，因此本文件中标识的 ST 产品（ST）可能由各种安全认证机构和/或根据本文件所述实施我们自己的安全措施。然而，无论何种安全认证和/或内置安全措施都无法保证 ST 产品能够抵御所有形式的攻击。因此，意法半导体的每一位客户都有责任自行判断所提供的安全级别是否满足其需求，无论是单独使用 ST 产品，还是与其他组件和/或软件结合用于客户最终产品或应用时。特别是请注意：

- ST产品可能已通过一个或多个安全认证机构的认证，例如平台安全架构（www.psacertified.org）和/或物联网平台安全评估标准（www.trustcb.com）。如需了解此处提及的ST产品是否已获得安全认证及其认证等级和当前状态，请访问相关认证标准网站或前往www.st.com上的相关产品页面以获取最新信息。由于ST产品的安全认证状态和/或等级可能随时发生变化，客户应根据需要重新检查安全认证的状态/等级。如果某ST产品未显示已通过特定安全标准的认证，客户不应假设其已通过认证。

- 认证机构有权针对ST产品进行安全认证的评估、颁发和撤销。因此，这些认证机构独立负责对ST产品的安全认证进行颁发或撤销，ST不对认证机构针对任何ST产品进行的任何错误、评估、测试或其他活动承担任何责任。

- 基于行业标准的加密算法（如AES、DES或MD5）及其他可能与ST产品结合使用的公开标准技术，其基础是未由ST开发的标准。ST不对此类加密算法或公开技术本身的任何缺陷，以及任何已开发或可能开发用于绕过、解密或破解此类算法或技术的方法承担责任。

- 尽管可以进行全面的安全测试，但没有任何认证等级可以绝对保证对所有攻击的防护，包括但不限于未被测试的高级攻击、新或未识别的攻击形式，或在超出其规格或预期用途的情况下使用ST产品，或与其他组件或软件结合使用（这些组件或软件由客户用于创建其最终产品或应用程序）时的任何攻击形式。ST不对此类攻击的抵御负责。因此，无论ST提供的集成安全功能和/或任何信息或支持，每个客户都单独负责确定所测试的攻击等级是否符合其需求，无论单独使用ST产品还是集成到客户最终产品或应用程序中。

- ST产品的所有安全功能（包括任何硬件、软件、文档等），" {v*}
包括但不限于ST添加的任何增强的安全功能，均按“现状”基础提供。因此，在适用法律允许的范围内，ST不承担任何担保责任，明示或默示，包括但不限于隐含的适销性或适用性担保，除非适用的书面并签署的合同条款另有规定。

Revision history

Table 6. **Document revision history**

Date	Version	Changes
19-Sep-2011	1	Initial release.
12-Dec-2011	2	Replaced STM42F4xx by STM32F4xx on cover page. Added silicon revision Z. Modified link to Arm 32-bit Cortex-M4F errata notice in Section 1: Arm® 32-bit Cortex®-M4 with FPU limitations. Updated status of ART Accelerator prefetch queue and MCU device ID limitations for revision Z in Table 4: Summary of silicon limitations. Updated Section 2.1.1: ART Accelerator prefetch queue instruction is not supported and Section 2.1.2: MCU device ID is incorrect to make differentiate between revision A and revision Z devices. Added Section 2.1.6: Full JTAG configuration without NJTRST pin cannot be used, Section 2.1.7: PDR_ON pin not available on LQFP100 package for revision Z devices, Section 2.1.8: Incorrect BOR option byte when consecutively programming BOR option byte, and Section 2.1.9: Configuration of PH10 and PI10 as external interrupts is erroneous. Updated workaround for Section 2.8.5: nRTS signal abnormally riven low after a protocol violation. Added Section 2.13.2: Wrong CCRCFAIL status after a response without CRC is received and Section 2.3.1: RVU and PVU flags are not reset in Stop mode.
03-Aug-2012	3	Added Section : None., Section 2.1.11: Slowing down APB clock during a DMA transfer, Section 2.1.12: MPU attribute to RTC and IWDG registers could be managed incorrectly, Section 2.1.13: Delay after an RCC peripheral clock enabling, Section 2.1.14: Battery charge monitoring lower than 2.4 Volts and Appendix A: Revision code on device marking. Added Section 2.12.2: FSMC synchronous mode and NWAIT signal disabled. Added Section 2.13.3: SDIO clock divider BYPASS mode may not work properly, Section 2.13.4: Data corruption in SDIO clock dephasing (NEGEDGE) mode and Section 2.13.5: CE-ATA multiple write command and card busy signal management. Added Section 2.15: DAC peripheral limitations with Section 2.15.1: DMA underrun flag management and Section 2.15.2: DMA request not automatically cleared by DMAEN=0.
25-Apr-2013	4	Added Section 1.2: VDIV or VSQRT instructions might not complete correctly when very short ISRs are used. Removed the reference to ‘Cortex-M4F’ in the whole document. Updated Table 2: Device summary, Section 2.1.2: MCU device ID is incorrect. Added Section 2.1.5: Wakeup sequence from Standby mode when using more than one wakeup source. Updated Section 2.12.1: Dummy read cycles inserted when reading synchronous memories. Added Section 2.2: TIM limitations, Section 2.7.2: I2S2 in full-duplex mode may not work properly when SCK and WS signals are mapped on PI1 and PI0 respectively, Section 2.11.5: Successive write operations to the same register might not be fully taken into account and Section 2.12.3: FSMC NOR Flash/ PSRAM controller asynchronous access on bank 2 to 4 when bank 1 is in synchronous mode (CBURSTRW bit is set) , Section 2.13.6: No underrun detection with wrong data transmission and Section 2.14.1: ADC sequencer modification during conversion. Added Figure 6: WLCSP90 top package view.

修订历史 {v*}

表6. 文档修订历史

Date	Version	Changes
19-Sep-2011	1	Initial release.
12-Dec-2011	2	Replaced STM42F4xx by STM32F4xx on cover page. Added silicon revision Z. Modified link to Arm 32-bit Cortex-M4F errata notice in Section 1: Arm® 32-bit Cortex®-M4 with FPU limitations. Updated status of ART Accelerator prefetch queue and MCU device ID limitations for revision Z in Table 4: Summary of silicon limitations. Updated Section 2.1.1: ART Accelerator prefetch queue instruction is not supported and Section 2.1.2: MCU device ID is incorrect to make differentiate between revision A and revision Z devices. Added Section 2.1.6: Full JTAG configuration without NJTRST pin cannot be used, Section 2.1.7: PDR_ON pin not available on LQFP100 package for revision Z devices, Section 2.1.8: Incorrect BOR option byte when consecutively programming BOR option byte, and Section 2.1.9: Configuration of PH10 and PI10 as external interrupts is erroneous. Updated workaround for Section 2.8.5: nRTS signal abnormally riven low after a protocol violation. Added Section 2.13.2: Wrong CCRCFAIL status after a response without CRC is received and Section 2.3.1: RVU and PVU flags are not reset in Stop mode.
03-Aug-2012	3	Added Section : None., Section 2.1.11: Slowing down APB clock during a DMA transfer, Section 2.1.12: MPU attribute to RTC and IWDG registers could be managed incorrectly, Section 2.1.13: Delay after an RCC peripheral clock enabling, Section 2.1.14: Battery charge monitoring lower than 2.4 Volts and Appendix A: Revision code on device marking. Added Section 2.12.2: FSMC synchronous mode and NWAIT signal disabled. Added Section 2.13.3: SDIO clock divider BYPASS mode may not work properly, Section 2.13.4: Data corruption in SDIO clock dephasing (NEGEDGE) mode and Section 2.13.5: CE-ATA multiple write command and card busy signal management. Added Section 2.15: DAC peripheral limitations with Section 2.15.1: DMA underrun flag management and Section 2.15.2: DMA request not automatically cleared by DMAEN=0.
25-Apr-2013	4	Added Section 1.2: VDIV or VSQRT instructions might not complete correctly when very short ISRs are used. Removed the reference to ‘Cortex-M4F’ in the whole document. Updated Table 2: Device summary, Section 2.1.2: MCU device ID is incorrect. Added Section 2.1.5: Wakeup sequence from Standby mode when using more than one wakeup source. Updated Section 2.12.1: Dummy read cycles inserted when reading synchronous memories. Added Section 2.2: TIM limitations, Section 2.7.2: I2S2 in full-duplex mode may not work properly when SCK and WS signals are mapped on PI1 and PI0 respectively, Section 2.11.5: Successive write operations to the same register might not be fully taken into account and Section 2.12.3: FSMC NOR Flash/ PSRAM controller asynchronous access on bank 2 to 4 when bank 1 is in synchronous mode (CBURSTRW bit is set) , Section 2.13.6: No underrun detection with wrong data transmission and Section 2.14.1: ADC sequencer modification during conversion. Added Figure 6: WLCSP90 top package view.

Date	Version	Changes
11-Oct-2013	5	Added silicon revision 1. Added Section 2.5.5: Both SDA and SCL maximum rise time (tr) violated when VDD_I2C bus higher than ((VDD+0.3) / 0.7) V. Moved device marking to datasheets.
21-Jan-2015	6	Added: rev 2 and Y on Table 1: Device identification, Section 1.2: VDIV or VSQRT instructions might not complete correctly when very short ISRs are used. Section 2.4: RTC limitations: updated Section 2.8.6: Start bit detected too soon when sampling for NACK signal from the smartcard to Section 2.8.9: nRTS is active while RE or UE = 0 Section 2.9: bxCAN limitations: updated Table 4: Summary of silicon limitations.
14-Sep-2015	7	Added:Section 2.6: SPI peripheral limitations, Section 2.6.1: Wrong CRC calculation when the polynomial is even, Section 2.5.6: Spurious Bus Error detection in Master mode. Section 2.4.7: Wrong behavior related with MCU Stop mode when wakeup from Stop mode by I2C peripheral is disabled. Section 2.6.3: BSY bit may stay high at the end of a data transfer in Slave mode. Section 2.6.2: Corrupted last bit of data and/or CRC, received in Master mode with delayed SCK feedback. Updated:Section 2.1.10: DMA2 data corruption when managing AHB and APB peripherals in a concurrent way. Replaced Section 2.1.5: Debugging Stop mode with WFE entry with Section 2.1.3: Debugging Sleep/Stop mode with WFE/WFI entry.
24-Nov-2016	8	Added workaround in Section 2.1.13: Delay after an RCC peripheral clock enabling. Added Section 2.4.3: RTC calendar registers are not locked properly. Updated Section 2.6.2: Corrupted last bit of data and/or CRC, received in Master mode with delayed SCK feedback and added Section 2.5.3: Wrong CRC transmitted in Master mode with delayed SCK feedback. Updated Section 2.6.3: BSY bit may stay high at the end of a data transfer in Slave mode. Updated limitation description in Section 2.5.2: Start cannot be generated after a misplaced Stop. Added Section 2.7.3: Corrupted last bit of data and/or CRC, received in Master mode with delayed SCK feedback in Section 2.7: I2S peripheral limitations
04-Jul-2017	9	Removed Section Wrong CRC transmitted in Master mode with delayed SCK feedback and Section 2.4.7: Wrong behavior related with MCU Stop mode when wakeup from Stop mode by I2C peripheral is disabled. The I2C limitation does not apply to STM32F40x and STM32F41x microcontrollers. Updated Section 2.6.2: Corrupted last bit of data and/or CRC, received in Master mode with delayed SCK feedback
24-Apr-2019	10	Added revision code “4” in Table 1: Device identification. Updated Table 4: Summary of silicon limitations. Added Section 2.1.16: RDP level 2 and sector write protection configuration.
05-Dec-2019	11	Added Section 2.1.17: Possible delay in backup domain protection disabling/ enabling after programming the DBP bit.
08-Apr-2020	12	Added revision code “4” on document cover page.
04-Jan-2021	13	Removed list of revisions from Section : Silicon identification introduction. Added revision 5 and 6. Added Section 2.2: TIM limitations.

Date	Version	Changes
11-Oct-2013	5	Added silicon revision 1. Added Section 2.5.5: Both SDA and SCL maximum rise time (tr) violated when VDD_I2C bus higher than ((VDD+0.3) / 0.7) V. Moved device marking to datasheets.
21-Jan-2015	6	Added: rev 2 and Y on Table 1: Device identification, Section 1.2: VDIV or VSQRT instructions might not complete correctly when very short ISRs are used. Section 2.4: RTC limitations: updated Section 2.8.6: Start bit detected too soon when sampling for NACK signal from the smartcard to Section 2.8.9: nRTS is active while RE or UE = 0 Section 2.9: bxCAN limitations: updated Table 4: Summary of silicon limitations.
14-Sep-2015	7	Added:Section 2.6: SPI peripheral limitations, Section 2.6.1: Wrong CRC calculation when the polynomial is even, Section 2.5.6: Spurious Bus Error detection in Master mode. Section 2.4.7: Wrong behavior related with MCU Stop mode when wakeup from Stop mode by I2C peripheral is disabled. Section 2.6.3: BSY bit may stay high at the end of a data transfer in Slave mode. Section 2.6.2: Corrupted last bit of data and/or CRC, received in Master mode with delayed SCK feedback. Updated:Section 2.1.10: DMA2 data corruption when managing AHB and APB peripherals in a concurrent way. Replaced Section 2.1.5: Debugging Stop mode with WFE entry with Section 2.1.3: Debugging Sleep/Stop mode with WFE/WFI entry.
24-Nov-2016	8	Added workaround in Section 2.1.13: Delay after an RCC peripheral clock enabling. Added Section 2.4.3: RTC calendar registers are not locked properly. Updated Section 2.6.2: Corrupted last bit of data and/or CRC, received in Master mode with delayed SCK feedback and added Section 2.5.3: Wrong CRC transmitted in Master mode with delayed SCK feedback. Updated Section 2.6.3: BSY bit may stay high at the end of a data transfer in Slave mode. Updated limitation description in Section 2.5.2: Start cannot be generated after a misplaced Stop. Added Section 2.7.3: Corrupted last bit of data and/or CRC, received in Master mode with delayed SCK feedback in Section 2.7: I2S peripheral limitations
04-Jul-2017	9	Removed Section Wrong CRC transmitted in Master mode with delayed SCK feedback and Section 2.4.7: Wrong behavior related with MCU Stop mode when wakeup from Stop mode by I2C peripheral is disabled. The I2C limitation does not apply to STM32F40x and STM32F41x microcontrollers. Updated Section 2.6.2: Corrupted last bit of data and/or CRC, received in Master mode with delayed SCK feedback
24-Apr-2019	10	Added revision code “4” in Table 1: Device identification. Updated Table 4: Summary of silicon limitations. Added Section 2.1.16: RDP level 2 and sector write protection configuration.
05-Dec-2019	11	Added Section 2.1.17: Possible delay in backup domain protection disabling/ enabling after programming the DBP bit.
08-Apr-2020	12	Added revision code “4” on document cover page.
04-Jan-2021	13	Removed list of revisions from Section : Silicon identification introduction. Added revision 5 and 6. Added Section 2.2: TIM limitations.

Date	Version	Changes
21-Feb-2023	14	Core: added Store immediate overlapping exception return operation might vector to incorrect interrupt erratum. System: • Updated workaround of ART Accelerator prefetch queue instruction is not supported, MCU device ID is incorrect, Debugging Sleep/Stop mode with WFE/WFI entry, and Delay after an RCC peripheral clock enabling errata. • Updated Configuration of PH10 and PI10 as external interrupts is erroneous erratum description. • Updated In some specific cases, DMA2 data corruption occurs when managing AHB and APB2 peripherals in a concurrent way erratum. • Added PC13 signal transitions disturb LSE erratum. IWDG: added RVU flag not cleared at low APB clock frequency and PVU flag not cleared at low APB clock frequency errata. RTC: added RTC interrupt can be masked by another RTC interrupt, Calendar initialization may fail in case of consecutive INIT mode entry and Alarm flag may be repeatedly set when the core is stopped in debug errata. USART: • Changed nCTS and nRTS to CTS and RTS respectively. • Updated Break request can prevent the transmission complete flag (TC) from being set and Guard time not respected when data are sent on TXE events erratum workarounds. SPI/I2S: added BSY bit may stay high when SPI is disabled, and Anticipated communication upon SPI transit from slave receiver to master errata. SDIO: updated Wrong data written during SDIO hardware flow control erratum title. OTG_FS: added Transmit data FIFO is corrupted when a write sequence to the FIFO is interrupted with accesses to certain OTG_FS registers and Host packet transmission may hang when connecting through a hub to a low-speed device errata. Added OTG_HS errata. ETH: added Incorrect status and corrupted frames when RxFIFO overflow occurs on the penultimate word of Rx frames, Incorrect remote wakeup on global unicast packet, Overflow status bits of missed frame and buffer overflow counters are cleared without a read operation, MAC may provide incorrect Rx status for the MAC control frames when receive checksum offload is enabled, MAC may provide an inaccurate Rx status when receive checksum offload is enabled in cut-through mode and MAC may not drop received giant error frames errata. Added Section Important security notice.
12-Feb-2024	15	Updated errata: PC13 signal transitions disturb LSE
10-Jul-2024	16	Updated the order of functions in Table 3. Summary of device limitations.
27-Jan-2025	17	Master and slave terms in I2C replaced with controller and target, respectively. Added errata: Corrupted content of the backup domain due to a missed power-on reset after this domain supply voltage drop
24-Feb-2025	18	Corrupted content of the backup domain due to a missed power-on reset after this domain supply voltage drop updated.

Date	Version	Changes
21-Feb-2023	14	Core: added Store immediate overlapping exception return operation might vector to incorrect interrupt erratum. System: • Updated workaround of ART Accelerator prefetch queue instruction is not supported, MCU device ID is incorrect, Debugging Sleep/Stop mode with WFE/WFI entry, and Delay after an RCC peripheral clock enabling errata. • Updated Configuration of PH10 and PI10 as external interrupts is erroneous erratum description. • Updated In some specific cases, DMA2 data corruption occurs when managing AHB and APB2 peripherals in a concurrent way erratum. • Added PC13 signal transitions disturb LSE erratum. IWDG: added RVU flag not cleared at low APB clock frequency and PVU flag not cleared at low APB clock frequency errata. RTC: added RTC interrupt can be masked by another RTC interrupt, Calendar initialization may fail in case of consecutive INIT mode entry and Alarm flag may be repeatedly set when the core is stopped in debug errata. USART: • Changed nCTS and nRTS to CTS and RTS respectively. • Updated Break request can prevent the transmission complete flag (TC) from being set and Guard time not respected when data are sent on TXE events erratum workarounds. SPI/I2S: added BSY bit may stay high when SPI is disabled, and Anticipated communication upon SPI transit from slave receiver to master errata. SDIO: updated Wrong data written during SDIO hardware flow control erratum title. OTG_FS: added Transmit data FIFO is corrupted when a write sequence to the FIFO is interrupted with accesses to certain OTG_FS registers and Host packet transmission may hang when connecting through a hub to a low-speed device errata. Added OTG_HS errata. ETH: added Incorrect status and corrupted frames when RxFIFO overflow occurs on the penultimate word of Rx frames, Incorrect remote wakeup on global unicast packet, Overflow status bits of missed frame and buffer overflow counters are cleared without a read operation, MAC may provide incorrect Rx status for the MAC control frames when receive checksum offload is enabled, MAC may provide an inaccurate Rx status when receive checksum offload is enabled in cut-through mode and MAC may not drop received giant error frames errata. Added Section Important security notice.
12-Feb-2024	15	Updated errata: PC13 signal transitions disturb LSE
10-Jul-2024	16	Updated the order of functions in Table 3. Summary of device limitations.
27-Jan-2025	17	Master and slave terms in I2C replaced with controller and target, respectively. Added errata: Corrupted content of the backup domain due to a missed power-on reset after this domain supply voltage drop
24-Feb-2025	18	Corrupted content of the backup domain due to a missed power-on reset after this domain supply voltage drop updated.

Contents

目录