STM32N6xxxx device errata

Applicability

This document applies to the part numbers of STM32N6xxxx 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 RM0486.

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
STM32N6xxxx	STM32N645A0, STM32N645B0, STM32N645I0, STM32N645L0, STM32N645X0, STM32N645Z0, STM32N647A0, STM32N647B0, STM32N647I0, STM32N647L0, STM32N647X0, STM32N647Z0, STM32N655A0, STM32N655B0, STM32N655I0, STM32N655L0, STM32N655X0, STM32N655Z0, STM32N657A0, STM32N657B0, STM32N657I0, STM32N657L0, STM32N657X0, STM32N657Z0

Table 2. Device variants

Reference	Silicon revision codes	
	Device marking(1)	REV_ID(2)
STM32N6xxxx	Z	0x1000
	B	
	X	

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 STM32N6xxxx device limitations and their status:

A Limitation applicable. Workaround available

N Limitation applicable. No workaround available

P Limitation applicable. Partial workaround available

- Limitation not applicable.

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

Table 3. Summary of device limitations

Function	Section	Limitation	Status		
			Rev. Z	Rev. B	Rev. X
Core	2.1.1	The Send Event Arm® Cortex®-M55 instruction does not toggle I/O	N	N	N
	2.1.2	Enabling Arm® Cortex®-M55 data cache with SCB_EnableDCache() results in unpredictable debug behavior	N	-	-
	2.1.3	CoreSight™ STM-500 AXI stimulus port Master ID does not match expected value (Cortex®-M55 documentation)	N	N	N
System	2.2.1	APB*x* clock prescaling fails	P	P	P
	2.2.2	Do not clear BSECEN bit before entering Low-power mode	N	N	N
	2.2.3	Debugging connection lost when writing into RISAFx without enabling the related clock	A	A	A
	2.2.4	Overconsumption in Standby mode	N	-	-
	2.2.5	Incorrect DBGMCU_IDCODE register value	N	A	A
	2.2.6	Unable to enable VENC clock independently	A	A	A
	2.2.7	Boot process stuck when using HyperFlash™	P	P	P
	2.2.8	USB download stuck when using NA payload	N	-	-
	2.2.9	SecureBoot anti-rollback corner case error	N	-	-
	2.2.10	PG10 not maintained at 1 during blocking failure	N	-	-
	2.2.11	Monotonic counter only uses 32-bit fuse value in anti-rollback version	N	-	-
	2.2.12	FSBL2 boot fails due to SecureBoot after incorrect FSBL1 image version boot attempt	N	-	-
	2.2.13	FSBL2 execution fails after testing FSBL1 with incorrect public key and signature in SecureBoot	N	-	-
	2.2.14	Desynchronization of VENC and DCMIPP when streaming	A	-	-
	2.2.15	Bit 31 VSWRST: V_{switch} (VSW) domain software reset in RCC_BDCR register not functional	N	N	N
	2.2.16	CLAMP enable requested on path from VREFBUF to Pad_VREFP	N	A	A
	2.2.17	SSP is not supported	N	N	N
	2.2.18	Incorrect reset behavior when using NRST pin	A	A	-
	2.2.19	Access to SYSCFG_VDDIOxCCCR and SYSCFG_VDDIOxCCSR unduly depends on VDDIOxSV	A	A	A

Function	Section	Limitation	Status		
			Rev. Z	Rev. B	Rev. X
System	2.2.20	I/O compensation may deform output signal rise and fall edges	A	A	A
	2.2.21	STNoC AXI deadlock during GFXMMU read of 24 bpp frame buffer	N	N	N
	2.2.22	Boot ROM execution failure after system reset due to wrong clocking of peripherals	A	A	A
FMC	2.3.1	NOR flash memory/PSRAM incorrect bus turnaround timing	A	A	A
	2.3.2	Incorrect FMC_CLK clock period when CLKDIV value is changed on-the-fly in Continuous clock mode	A	A	A
XSPI	2.4.1	Deadlock can occur under certain conditions	A	A	A
	2.4.2	Indirect write mode limited to 256 Mbytes	N	N	N
	2.4.3	CALMAX bit not set when the PHY reaches the DLL maximum value	N	N	N
	2.4.4	Read data corruption or deadlock when a linear burst read is followed by a sequential misaligned wrap burst	A	A	A
	2.4.5	Deadlock or data corruption when DQS is enabled and the wrap request includes the last address of the memory	A	A	A
	2.4.6	Transaction issues when reaching the last memory address of a 256-Mbyte memory in memory mapped mode	A	A	A
	2.4.7	Transactions are limited to 8 Mbytes in OctaRAM™ memories	N	N	N
	2.4.8	Variable latency is not supported when a refresh collision occurs during a write access to some OctaRAM™ memories	P	P	P
	2.4.9	In automatic status-polling and multiplexed modes, the controller does not request the port if less than two bytes are sent per cycle when XSPI_DLR is cleared	A	A	A
	2.4.10	Possible deadlock when a request arrives during the disabling process	A	A	A
	2.4.11	A read mismatch occurs after disabling the controller during a read transaction	A	A	A
SDMMC	2.5.1	Command response and receive data end bits not checked	N	N	N
ADC	2.6.1	In injected simultaneous mode, stopping regular conversion may delay injected conversion	A	A	A
	2.6.2	In combined regular simultaneous plus alternate trigger mode, stopping injected conversion may delay regular conversion	A	A	A
	2.6.3	In certain dual ADC modes with regular oversampling continued mode enabled, the JEOC flag is not set at the end of the new injected conversion	A	A	A
	2.6.4	In certain dual ADC modes with DMA one-shot mode enabled, JADSTART and ADSTART may not be cleared if JADSTP and ADSTP are set at the same time	A	A	A
	2.6.5	Shifted master and slave sequence in interleaved discontinuous mode	A	A	A
	2.6.6	When the ADC clock is derived from the AHB clock, the injected conversion latency is not respected if the injected trigger coincides with the stopping of the regular conversion	A	A	A
	2.6.7	In certain dual modes, the fixed trigger latency for the injected conversions may not be respected	A	A	A
	2.6.8	In simultaneous regular mode, stopping an injected conversion may shift the next regular conversion master and slave timing by one clock cycle	A	A	A
LTDC	2.7.1	Layers cannot read YUV420 multibuffer data	N	N	N

Function	Section	Limitation	Status		
			Rev. Z	Rev. B	Rev. X
GPU2D	2.8.1	Occasional writing miss to frame buffer with slow memories	N	-	-
LPTIM	2.9.1	Device may remain stuck in LPTIM interrupt when entering Stop mode	A	A	A
	2.9.2	ARRM and CMPM flags are not set when APB clock is slower than kernel clock	A	A	A
	2.9.3	Interrupt status flag is cleared by hardware upon writing its corresponding bit in LPTIM_DIER register	N	N	N
	2.9.4	LPTIM4 remains active after user code starts in development mode	N	-	-
RTC	2.10.1	Alarm flag may be repeatedly set when the core is stopped in debug	N	N	N
I2C	2.11.1	Wrong data sampling when data setup time ($t_{SU;DAT}$) is shorter than one I2C kernel clock period	P	P	P
	2.11.2	Spurious bus error detection in controller mode	A	A	A
USART	2.12.1	Received data may be corrupted upon clearing the ABREN bit	A	A	A
	2.12.2	Noise error flag set while ONEBIT is set	N	N	N
LPUART	2.13.1	Possible LPUART transmitter issue when using low BRR[15:0] value	P	P	P
FDCAN	2.14.1	Desynchronization under specific condition with edge filtering enabled	A	A	A
	2.14.2	Tx FIFO messages inverted under specific buffer usage and priority setting	A	A	A
	2.14.3	DAR mode transmission failure due to lost arbitration	A	A	A
USBPHYC	2.15.1	External VBUS detection method malfunctioning	N	N	N
UCPD	2.16.1	Ordered set with multiple errors in a single K-code is reported as invalid	N	N	N
	2.16.2	Rp value drift causes incorrect source current capabiliy detection	N	N	-
ETH	2.17.1	Incorrect DMA transfer state in TEB[2:0] and REB[2:0] on bus error	A	A	A
	2.17.2	Transmission status is not updated in ETH_MACFPECSR for back-to-back frame preemption Verify or Respond mPackets	A	A	A
	2.17.3	Incorrect routing of Rx packet or incorrect splitting of header payload on preemption Rx queue overflow	P	P	P
	2.17.4	MAC fails to identify PTP SYNC and Follow_Up messages with peer delay reserved multicast address in 802.1AS mixed mode	A	A	A
	2.17.5	VLAN tag filter fail packet queuing feature is enabled without considering the RA bit of the ETH_MACPFR register	A	A	A
	2.17.6	Incorrect gate control list switching for intermediate cycles when CTR is less than the GCL execution time	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
System	2.2.23	Recovery from clock switching failure
	2.2.24	Incorrect definition of HSERDY flag in reference manual
	2.2.25	Incorrect definition of the RTCPRE divider formula in reference manual
	2.2.26	Incorrect description of the OTP16 BootROM_CONFIG_7 in reference manual

Function	Section	Documentation erratum
System	2.2.27	Incorrect HSE minimum values in OTP16 BootROM_CONFIG_7 and in RCC section of the reference manual
	2.2.28	Missing default value for rng_htcr_value in OPT19 BootROM_CONFIG_10
	2.2.29	Incorrect operational status of PVM monitoring for V_{DDIO2}, V_{DDIO3}, V_{DDIO4}, V_{DDIO5}, USB33, and V_{DDCORE} in reference manual or datasheet

2 Description of device errata

The following sections describe the errata of the applicable devices and provide a workaround where available. They are grouped by device functions.

The applicable devices are based on Arm® Cortex® core.

arm

Note: Arm and Cortex are registered trademarks of Arm Limited (or its subsidiaries or affiliates) in the US and/or elsewhere.

The Arm word and logo are trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. All rights reserved.

2.1 Core

Reference manual and errata notice for the Arm® Cortex®-M55 core revision r0p1 is available from http://infocenter.arm.com.

2.1.1 The Send Event Arm® Cortex®-M55 instruction does not toggle I/O

Description

When executing the `__SEV()` (Send Event) instruction, the associated pin defined for it does not toggle (EVENTOUT).

Workaround

None.

2.1.2 Enabling Arm® Cortex®-M55 data cache with SCB_EnableDCache() results in unpredictable debug behavior

Description

Issues caused by the usage of `SCB_EnableDCache()` have been reproduced, including the following:

- Semihosting for printf is not functional.
- Debug memory view does not accurately reflect CPU writes.
- Callstack is not fully visible.

Workaround

None.

2.1.3 CoreSight™ STM-500 AXI stimulus port Master ID does not match expected value (Cortex®-M55 documentation)

Description

When accessing the STM-500 from the HPDMA, the trace packet reports ID 0x41 (nonsecure) or 0X01 (secure).

When accessing the STM-500 from the Cortex®-M55 or from the AXI_AP, the trace packet reports ID 0x7F (nonsecure) or 0x3F (secure).

This allows the user to determine whether the trace is being performed from an internal or external source.

Workaround

None.

2.2 System

2.2.1 APB*x* clock prescaling fails

Description

Attempts to change the default APB*x* clock prescaler division ratio of 1 (via the PPRE*x*[2:0] bitfields in the RCC_CFGR2 register) fail. As a result, the APBx (`pclkx`) clocks always replicate the AHB clock signal.

Workaround

Keep the PPRE*x*[2:0] bitfields at their default value 000. To clock a peripheral with a signal other than `pclkx` replicating the AHB clock, select a clock source other than `pclkx` using the clock selector of the peripheral.

Note: Most peripherals offer clock sources other than `pclkx`.

For synchronous operation of multiple peripherals that cannot operate at AHB clock speed, select a clock source other than `pclkx` that is common to all peripherals involved (such as `per_ck`) to maintain synchronization.

2.2.2 Do not clear BSECEN bit before entering Low-power mode

Description

If the BSECEN bit (RCC_APB4HENR) is cleared, the Cortex®-M55 delivers cpu_sleep_in and cpu_deepsleep_in signals to RCC, causing the system to fail to enter Sleep, Stop, or Standby mode.

Workaround

None.

2.2.3 Debugging connection lost when writing into RISAFx without enabling the related clock

Description

When attempting to lock the RISAF3 (NPU master 0) configuration, the debugger connection is lost and cannot be restored. This issue occurs with all RISAFs when the related clock is not enabled.

Workaround

Enable the clock.

2.2.4 Overconsumption in Standby mode

Description

When using Standby mode, there is an issue with unwanted power consumption on $V_{DD18AON}$.

Workaround

None.

2.2.5 Incorrect DBGMCU_IDCODE register value

Description

The DBGMCU_IDCODE register value remains the same.

Workaround

For revision Z devices, no workaround is available.

For revision B devices, use the new register definition Boot ROM (Base address: 0x1800 1000, Address offset: 0x0004).

2.2.6 Unable to enable VENC clock independently

Description

The VENCEN bit in RCC_APB5ENR does not enable the VENC clock unless another IPEN bit is set in the same register.

Workaround

Manually enable the APB5 clock in RCC_BUSENR.

2.2.7 Boot process stuck when using HyperFlash™

Description

When the HyperFlash™ device is missing, the boot process becomes stuck and the BOOTROM does not enter the USB/UART loop or blocking failure mode.

Workaround

Ensure that the HyperFlash™ device is present when booting in HyperFlash™ mode.

2.2.8 USB download stuck when using NA payload

Description

During USB download, a checksum is performed on the downloaded data. However, when using the NA payload, the boot test tool does not perform a checksum on the full payload.

Workaround

None.

2.2.9 SecureBoot anti-rollback corner case error

Description

Test suites and certifications are failing due to a BSEC error.

Workaround

None.

2.2.10 PG10 not maintained at 1 during blocking failure

Description

The PG10 (BootFailedn) is not maintained at 1 during a blocking failure.

Workaround

None.

2.2.11 Monotonic counter only uses 32-bit fuse value in anti-rollback version

Description

It is not possible to program a fuse value for the OEM monotonic counter that exceeds 32 bits, even if a 64-bit value fuse (two fuses of 32 bits) is dedicated for this purpose.

Workaround

None.

2.2.12 FSBL2 boot fails due to SecureBoot after incorrect FSBL1 image version boot attempt

Description

Since the FSBL1 header contains an incorrect version number, the FSBL1 boot process must fail and the system must then boot using FSBL2. However, this is not occurring as expected.

Workaround

None.

2.2.13 FSBL2 execution fails after testing FSBL1 with incorrect public key and signature in SecureBoot

Description

Due to an incorrect public key or signature in FSBL1, FSBL2 is not being executed.

Workaround

None.

2.2.14 Desynchronization of VENC and DCMIPP when streaming

Description

DCMIPP and VENC can become desynchronized when VENC cannot keep up with the streaming pace of DCMIPP. This typically occurs when VENC memory accesses experience temporary bandwidth limitations or longer-than-expected latencies.

Workaround

To resynchronize, reset VENC cleanly by following these steps:

1. **Stop DCMIPP capture**
   This halts data flow from DCMIPP to VENC.
2. **Wait briefly**
   Allow VENC to finish encoding last one or two buffers received before stopping. After this, VENC becomes idle, waiting for new buffers and stops AXI accesses.
3. **Reset VENC**
   Perform reset only after VENC is idle to avoid AXI crashes caused by abrupt resets during AXI access.
4. **Reconfigure VENC**
5. **Restart VENC**
   VENC waits for DCMIPP to provide a group of lines to encode, expected at the top of a frame.
6. **Restart DCMIPP capture**
   DCMIPP waits for start-of-frame packet and then begins capturing. The first group of lines is the start of a new frame. DCMIPP and VENC are now synchronous again.

2.2.15 Bit 31 VSWRST: V_{switch} (VSW) domain software reset in RCC_BDCR register not functional

Description

When using the internal SMPS in Run mode and performing a backup domain reset, the V_{DDCORE} voltage drops and does not return to its nominal value even after the backup domain reset is released.

Workaround

None.

2.2.16 CLAMP enable requested on path from VREFBUF to Pad_VREFP

Description

This configuration can cause a short current through $V_{DDA18AON}$ when the ADC supply is not present.

Workaround

For revision Z devices, no workaround is available.
For revision B devices, a specific software part has been added in started Init file.c (patch for VREFP pull-down).

2.2.17 SSP is not supported

Description

Registers BSEC_DBGCR and BSEC_AP_UNLOCK can only be written to once per cold reset.

- Any changes made to these registers last until the next cold power-on reset.
- A power cycle must be performed after provisioning (when changing from CLOSED_UNLOCKED to CLOSED_LOCKED).

Workaround

None.

2.2.18 Incorrect reset behavior when using NRST pin

Description

The system may fail to perform a correct reset when using the NRST pin. This issue might require a full power cycle (power off and on) to restore functionality.

Workaround

When using the NRST pin, configure the reset mode to power-on reset as follows:

1. On the first system power-on reset, set the reset type to power-on reset by setting the CORE_RESET_TYPE bit in the SYSCFG_CM55RSTCR register.
2. Generate a software reset by setting the SYSRESETREQ bit in the AIRCR register of the SBC.

In the main application code, declare this global variable at any address in AXISRAM1:

```
volatile uint32_t * init_p = (volatile uint32_t *) 0x34001004;
```

At the start of the main function, insert:

```
/* Enable SYSCFG Clock */
RCC->APB4ENSR2 |= 1;  // SYSCFGEN: SYSCFG enable in RCC_APB4HENR register
if(*init_p != 0x11223344) //First power-on reset is detected
{
      *init_p=0x11223344; // change init_p variable with a known value
      SYSCFG->CM55RSTCR |= 1; //Setting power-on reset from NRST pin
      __ISB(); // Instruction Synchronization Barrier
      __DSB(); // Data Synchronization Barrier
      NVIC_SystemReset(); // Generate a system reset
  }
```

2.2.19 Access to SYSCFG_VDDIOxCCCR and SYSCFG_VDDIOxCCSR unduly depends on VDDIOxSV

Description

Unexpectedly, access to the SYSCFG_VDDIOxCCCR (read-write) and SYSCFG_VDDIOxCCSR (read-only) registers is effective only when the corresponding VDDIOxSV bits are set in the PWR_SVMCRx registers as follows:

- VDDIO2SV and VDDIO4SV for accessing SYSCFG_VDDIO2CCCR, SYSCFG_VDDIO2CCSR, SYSCFG_VDDIO4CCCR, and SYSCFG_VDDIO4CCSR
- VDDIO3SV and VDDIO5SV for accessing SYSCFG_VDDIO3CCCR, SYSCFG_VDDIO3CCSR, SYSCFG_VDDIO5CCCR, and SYSCFG_VDDIO5CCSR

Note: The VDDIO2SV and VDDIO3SV bits are in the PWR_SVMCR3 register, the VDDIO4SV bit is in PWR_SVMCR1, and the VDDIO5SV bit is in PWR_SVMCR2.

Workaround

Set the corresponding VDDIOxSV bits before accessing the SYSCFG_VDDIOxCCCR and SYSCFG_VDDIOxCCSR registers.

2.2.20 I/O compensation may deform output signal rise and fall edges

Description

The I/O compensation deforms output signal rise and fall edges when setting the SYSCFG_VDDCCCR or the SYSCFG_VDDIOxCCCR registers in either of the following ways:

- I/O compensation enabled in automatic mode (EN = 1, CS = 0)
- I/O compensation using default values (EN = 0, CS = 0, RAPSRC[3:0] = 0x7, RANSRC[3:0] = 0x8)

Either condition may cause the output duty cycle to be out of specification. The second condition may also increase output jitter at temperature points where the APSRC[3:0] and ANSRC[3:0] bitfields change.

Workaround

Disable the I/O compensation cell (EN = 0, CS = 1) on the relevant power domains (VDD, VDDIO2, VDDIO3, VDDIO4, and VDDIO5) and use fixed compensation values for the RAPSRC[3:0] and RANSRC[3:0] bitfields. Set the SYSCFG_VDDCCCR and SYSCFG_VDDIOxCCCR registers to 0x00000287 (CS = 1, RAPSRC[3:0] = 0x8, and RANSRC[3:0] = 0x7).

Execute the following code in the `SystemInit` function or immediately after:

```
/* PWR configuration */
__HAL_RCC_PWR_CLK_ENABLE();
HAL_PWREx_EnableVddIO2();
HAL_PWREx_EnableVddIO3();
HAL_PWREx_EnableVddIO4();
HAL_PWREx_EnableVddIO5();

/* SYSCFG configuration */
__HAL_RCC_SYSCFG_CLK_ENABLE();
HAL_SYSCFG_ConfigVDDCompensationCell(SYSCFG_IO_REGISTER_CODE, 0x7,0x8);
HAL_SYSCFG_ConfigVDDIOCompensationCell(SYSCFG_IO_VDDIO2_CELL, SYSCFG_IO_REGISTER_CODE, 0x7,0x8);
HAL_SYSCFG_ConfigVDDIOCompensationCell(SYSCFG_IO_VDDIO3_CELL, SYSCFG_IO_REGISTER_CODE, 0x7,0x8);
HAL_SYSCFG_ConfigVDDIOCompensationCell(SYSCFG_IO_VDDIO4_CELL, SYSCFG_IO_REGISTER_CODE, 0x7,0x8);
HAL_SYSCFG_ConfigVDDIOCompensationCell(SYSCFG_IO_VDDIO5_CELL, SYSCFG_IO_REGISTER_CODE, 0x7,0x8);
```

2.2.21 STNoC AXI deadlock during GFXMMU read of 24 bpp frame buffer

Description

A deadlock occurs in the STNoC AXI interconnect when GFXMMU reads and unpacks data from the 24 bpp frame buffer. This prevents use of the 24 bpp frame buffer and limits supported frame buffers to 16 bpp and 32 bpp.

Workaround

None.

2.2.22 Boot ROM execution failure after system reset due to wrong clocking of peripherals

Description

Upon a software reset, the RCC_CCIPRx registers duly revert to their default values, but the clock switches controlled by the PERxSEL bitfields in these registers may fail to output the corresponding clock signals.

Consequently, the boot ROM code execution may fail if the clock switch states before the reset differ from their default states for peripherals used during the ROM boot sequence.

Workaround

Before performing a software reset, ensure that the following clock switch control bitfields are set to their default values and that the selected clocks are properly supplied to their corresponding peripherals:

Register	Bitfield	Default value
RCC_CCIPR6	XSPI1SEL[1:0]	0x0
	OTGPHY1CKREFSEL	
RCC_CCIPR7	PERSEL[2:0]	
	RTCSEL[1:0]	0x2
RCC_CCIPR8	SDMMC1SEL[1:0]	0x0
	SDMMC2SEL[1:0]	
RCC_CCIPR13	USART1SEL[2:0]	
	USART2SEL[2:0]	
	UART4SEL[2:0]	
	UART5SEL[2:0]	

Important: To successfully switch from one clock signal to another, ensure that both clock signals are present at their clock switch inputs before updating the control bitfield value to reflect the new clock switch state.

2.2.23 Recovery from clock switching failure

Description

The *Clock switches and gating* subsection of the *Reset and clock control (RCC)* section in the reference manual states that switching from one clock (initial) to another (target) requires both clocks to be present at the clock switch inputs. If either clock is missing, the output provides no clock signal.

Some reference manual revisions incorrectly claim that recovery requires only providing the target clock or switching back to the initial clock. This is inaccurate.

The correct procedure is to provide clock signals at **both** clock switch inputs involved in the operation.

Workaround

This is a documentation inaccuracy issue. No workaround is applicable.

2.2.24 Incorrect definition of HSERDY flag in reference manual

Description

In reference manual versions prior to RM0486 revision 4, the HSERDY flag is incorrectly described.

The correct definition is that HSERDY is set once the initial clock cycles from the HSE oscillator are detected after the HSEON bit is set. This does not guarantee clock stability.

The stability of the HSE clock depends on the external oscillator. Proper HSE operation and frequency stability of the external clock must be verified by referring to the actual startup time of the external HSE oscillator, as specified in the datasheet.

Workaround

None.

2.2.25 Incorrect definition of the RTCPRE divider formula in reference manual

Description

In reference manual versions prior to RM0486 revision 4, the description of the RTC oscillator divider RTCPRE[5:0] and the associated formula are incorrect. The correct information is as follows:

The division ratio is non-linear and is defined as: division factor = hse_ck / (v + 1)², where v is the value of RTCPRE[5:0]:

- 0x00: hse_ck divided by 1
- 0x01: hse_ck divided by 4
- 0x02: hse_ck divided by 9
- 0x03: hse_ck divided by 16
- ...
- 0x3F: hse_ck divided by 4096

Workaround

None.

2.2.26 Incorrect description of the OTP16 BootROM_CONFIG_7 in reference manual

Description

In reference manual versions prior to RM0486 revision 4, the detailed descriptions of bit 0: disable_traces, bit 1: disable_hse_freq_detect, and bit 2: disable_hse_bypass_detect in OTP16 were incorrect and used inverted logic.

Workaround

None.

2.2.27 Incorrect HSE minimum values in OTP16 BootROM_CONFIG_7 and in RCC section of the reference manual

Description

In reference manual versions prior to RM0486 revision 4, all references to HSE values of 8, 10, 12, or 14 are incorrect.

The minimum HSE value on STM32N6 is 16 MHz.

Workaround

None.

2.2.28 Missing default value for rng_htcr_value in OPT19 BootROM_CONFIG_10

Description

In reference manual versions prior to RM0486 revision 4, the default value of the rng_htcr_value field in OPT19 – BootROM_CONFIG_10 is not documented correctly.

The correct default value is: rng_htcr_value = 6. This value corresponds to an rng_htcr register value of 0xAAC7 for the associated bitfield configuration.

This issue affects documentation only. The silicon behavior is correct.

Workaround

None.

2.2.29 Incorrect operational status of PVM monitoring for V_{DDIO2}, V_{DDIO3}, V_{DDIO4}, V_{DDIO5}, USB33, and V_{DDCORE} in reference manual or datasheet

Description

In the reference manual and datasheet, the Programmable Voltage Monitor (PVM) monitoring for the following power supplies is incorrectly described as operational:

- V_{DDIO2}
- V_{DDIO3}
- V_{DDIO4}
- V_{DDIO5}
- USB33 VAM
- V_{DDCORE}

Both documents wrongly indicate that the PVM functionality is available and functional for these power supplies. However, PVM monitoring is not operational for these supplies.

As a result, no PVM events, flags, or interrupts are gernerated for the listed voltage domains. Do not use software that relies on PVM status for these supplies for voltage supervision or safety monitoring.

Workaround

None.

2.3 FMC

2.3.1 NOR flash memory/PSRAM incorrect bus turnaround timing

Description

The delays between consecutive device accesses, programmed through the BUSTURN[3:0] bitfield of the FMC_BTRx and FMC_BWTRx registers, have no effect. Instead systematic delays are applied:

- t_{IDLE2}: 2 * fmc_ker_ck cycles between accesses to two NOR/PSRAM devices
- t_{IDLE1}: 1 * fmc_ker_ck cycle between accesses to a NOR/PSRAM and a NAND flash memory

Workaround

Extend the bus turnaround delays to satisfy bus turnaround constraints. Three cases need to be considered:

1. Two consecutive accesses to non-multiplexed NOR/PSRAM devices:
   Program t_{ADDSET} (NOR/PSRAM address setup phase) as needed (see Figure 1).
2. Access to a non-multiplexed NOR/PSRAM followed by an access either to a NOR/PSRAM device with multiplexed A/DQ signals or to a synchronous device:
   Decrease the FMC kernel clock frequency in order to meet the timing constraints (see Figure 2).
3. Access to a non-multiplexed NOR/PSRAM followed by an access to a NAND flash memory device
   Program t_{MEMSET} (NOR/PSRAM address setup phase) as needed (see Figure 3).

Figure 1. Bus turn timing recovery - asynchronous accesses to NOR/PSRAM devices (case 1)

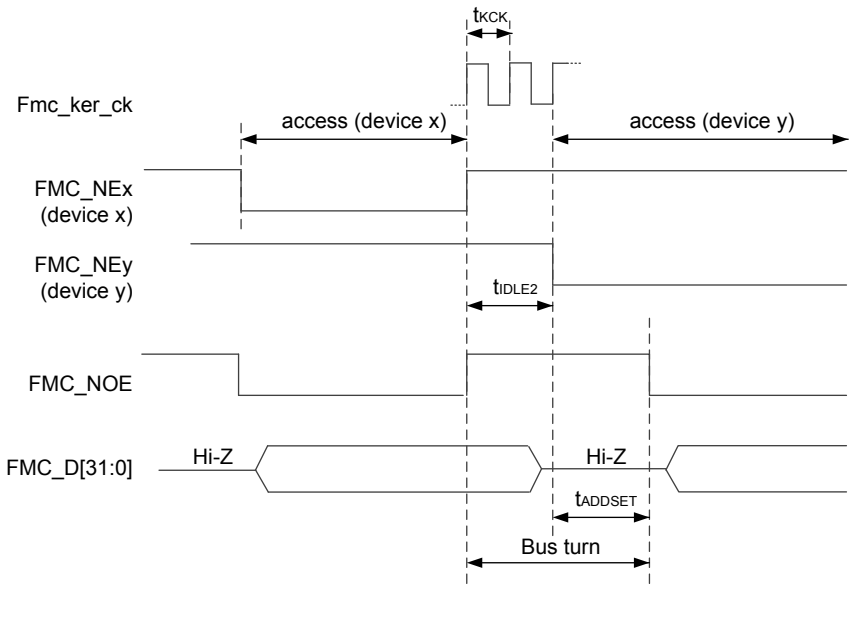

Figure 2. Bus turn timing recovery - access to NOR/PSRAM followed by access to multiplexed A/DQ synchronous device (case 2)

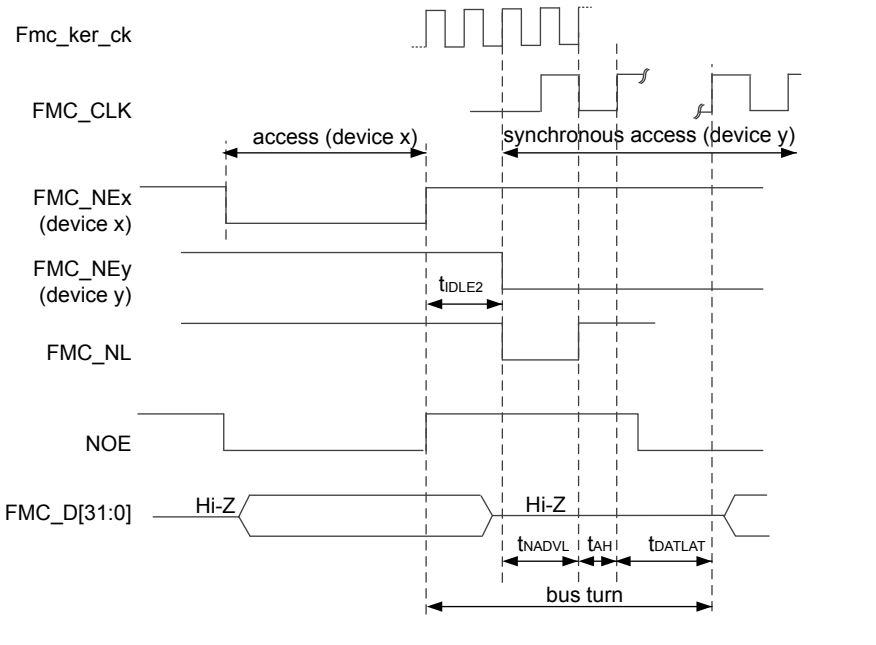

Figure 3. Bus turn timing recovery - access to NOR/PSRAM followed by access to NAND Flash device (case 3)

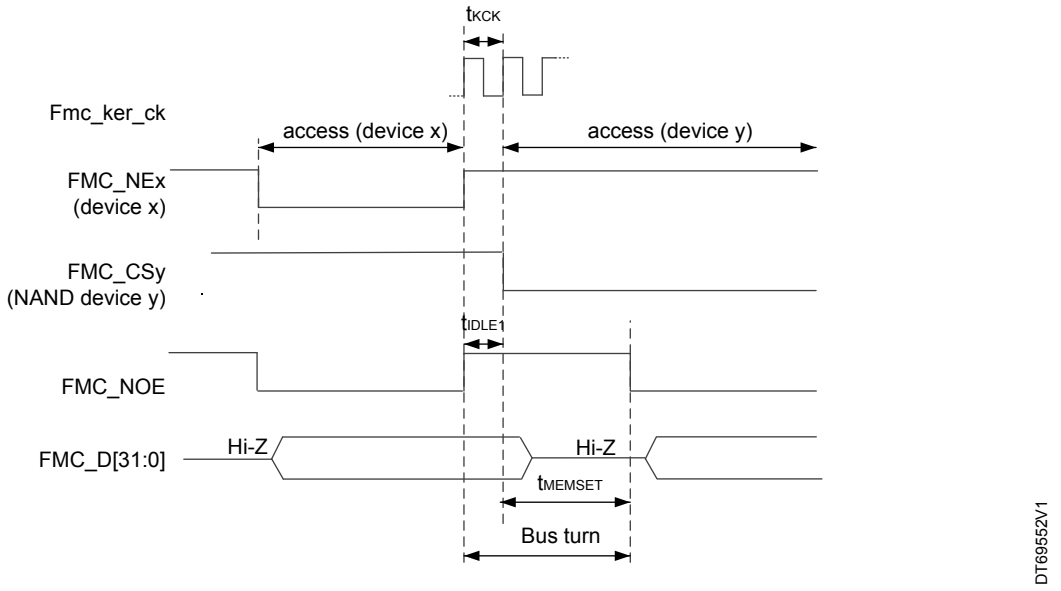

Table 5 gives the internal latencies that are not mentioned in the product datasheets. Refer to the datasheets for more details about the others timing values.

Table 5. Timing parameter description

Parameter	Description	Minimum value
t_{KCK}	FMC kernel clock period	See product datasheet
t_{IDLE1}	NEx high to CSy low (switching to a NAND flash memory device)	1 * t_{KCK}
t_{IDLE2}	NEx high to NEy low (NOR/PSRAM devices)	2 * t_{KCK}
t_{ADDSET}	NOR/PSRAM address setup phase	See Ref. Man.
t_{NADVL}	Address valid low pulse duration in synchronous mode	(CLKDIV+1) * t_{KCK}
t_{AH}	Address hold in synchronous mode	((CLKDIV+1) * t_{KCK}) / 2
t_{DATLAT}	NOR/PSRAM data latency in synchronous mode	See product datasheet
$t_{MEMxSET}$	NAND flash memory address setup phase	See product datasheet

2.3.2 Incorrect FMC_CLK clock period when CLKDIV value is changed on-the-fly in Continuous clock mode

Description

When the FMC operates in Continuous clock mode (CCLKEN is set in FMC_BCRx register), a new clock division factor is applied by changing the value of CLKDIV[3:0] in FMC_CFGR while the FMC is disabled (FMCEN cleared in FMC_BCRx), there is one FMC_CLK clock cycle during which the FMC_CLK period is not as expected: for example the clock low pulse duration matches the previous CLKDIV[3:0] value whereas the clock high pulse duration matches the new CLKDIV[3:0] value.

Workaround

Use the following sequence:

1. Stop the memory traffic for all devices.
2. Disable the FMC (refer to the disabling sequence described in the product reference manual).
3. Change CCLKEN for 1 to 0 in the FMC_BCRx register to stop the clock generation.
4. Program the desired CLKDIV[3:0] value in the FMC_CFGR register.
5. Change back CCLKEN from 0 to 1.
6. Enable the FMC.

2.4 XSPI

2.4.1 Deadlock can occur under certain conditions

Description

A deadlock can occur when all the following conditions are met:

- The product communicates through an I/O manager in multiplexed mode with an single external memory or an external combo featuring two memories, directly or through a high-speed interface.
- The external memory(ies) is(are) accessed in indirect mode or memory-mapped mode.

The deadlock can happen when the two following conditions occur at the same time:

- The Extended-SPI interface that currently owns the external bus (for example XSPI1) waits for a transfer to occur with the external memory, to complete its transfer on the internal interconnect matrix bus.
- A data transfer request on the internal interconnect matrix bus arrives to the other Extended-SPI interface (for example XSPI2).

This leads to an ownership conflict where:

- XSPI2 cannot get ownership of the external bus which is currently in use by XSPI1.
- XSPI1 cannot get ownership of the internal interconnect matrix bus which is currently in use by XSPI2.

Workaround

Apply one of the following measures:

- If any of the features generating automatic transfer split (MAXTRAN, REFRESH, CSBOUND, TIMEOUT) is set, XSPI1 splits its transfer at some point in time, releasing the bus. XSPI2 can then process its data, and when XSPI1 gets ownership back again, it resumes its transfer thanks to its embedded capability to restart at the address following the last address accessed. In this case, the deadlock is resolved.
  Limitation of the workaround: The automatic resume of the transfer does not work with certain flash memories in write direction only. These memories require an extra "write enable" command before resuming a write transfer. This "write enable" command is not generated by the XSPI.
- The application must ensure that it has sufficient room left in the XSPI internal FIFO for each and every transfer before launching it. The internal interconnect matrix bus activity no longer depends on what happens on external bus side, and the deadlock condition is avoided.

2.4.2 Indirect write mode limited to 256 Mbytes

Description

In indirect write mode, if the address is greater than 256 Mbytes, the indirect write is not performed at the targeted address, even if it is located inside the allowed memory space configured through the device size (DEVSIZE[4:0] of XSPI_DCR1). Actually, this write operation takes place within the 256-Mbyte memory space, thus corrupting the memory content.

Indirect read operations are not impacted.

Workaround

Indirect write operations have to be performed inside the first 256 Mbytes of the memory space.

2.4.3 CALMAX bit not set when the PHY reaches the DLL maximum value

Description

The CALMAX bit (bit 31 of the XSPI_CALFCR register) is expected to be set when the PHY reaches the maximum value of the DLL. However, this bit is wrongly cleared at the end of the calibration phase, meaning that it is always read at 0 when the calibration is complete, even if the maximum value of the calibration has been reached.

Workaround

None.

2.4.4 Read data corruption or deadlock when a linear burst read is followed by a sequential misaligned wrap burst

Description

When the prefetch is enabled, data corruption may occur when the following sequence occurs:

1. A linear burst read request has completed (with FIFO prefetch ongoing).
2. A misaligned wrap burst is requested with the first address of the wrap sequential to the last address of the linear read burst.

In this case, the data received come from the sequential prefetched data in the FIFO, but do not reflect the wrap roll-over to the beginning of the wrap window. For instance, for a wrap window from 0x1800 to 0x1820:

1. The linear read of 8 bytes is performed at 0x1810.
2. The misaligned wrap starts at 0x1818.

As a result, the wrap transaction returns values read from addresses 0x1818, 0x1820, 0x1828, and 0x1830, instead of 0x1818, 0x1800, 0x1808, and 0x1810.

In addition, the next sequential read request also returns corrupted data. This may even lead to a deadlock situation with the memory clock is permanently disabled.

Workaround

Apply one of the following measures:

- Disable the AXI prefetch by setting the NOPREF bit of the XSPI_CR register.
- Change the memory attribute to Device or disable the branch prediction in the memory protection unit.
- Access the memory from DCACHE or ICACHE.

2.4.5 Deadlock or data corruption when DQS is enabled and the wrap request includes the last address of the memory

Description

When DQS is enabled, if a wrap includes the last memory address, the wrap is not considered as complete even if the NCS is released by the controller.

This issue may results in data corruption and even cause a deadlock for the next transactions.

Workaround

Disregard a wrap including the last address of the memory. This can be done by managing the scatter file accordingly. In this case, the end address of the external memory to consider is:

```
@memory_last_address_to_use= @physical_memory_last_address - wrap_size.
```

2.4.6 Transaction issues when reaching the last memory address of a 256-Mbyte memory in memory mapped mode

Description

When accessing a 256-Mbyte memory in memory-mapped mode, the transaction may fail when it reaches the last address of the memory. The issue observed depends on the data direction (read/write from/to the memory):

- During a write access: the transaction is stopped before all the data have been sent.
- During a read access: a deadlock occurs because the AMBA® bus is still waiting for data.

Workaround

When accessing a 256-Mbyte memory in memory-mapped mode, avoid transactions ending at the last memory address.

2.4.7 Transactions are limited to 8 Mbytes in OctaRAM™ memories

Description

When the controller is configured in Macronix OctaRAM™ mode, by setting the MTYP[2:0] bitfield of the XSPI_DCR1 register to 011, only 13 bits of row address are decoded and sent to the memory, meaning that only 8 K of 1-Kbyte blocks can be accessed (8 Mbytes).

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

Workaround

None.

2.4.8 Variable latency is not supported when a refresh collision occurs during a write access to some OctaRAM™ memories

Description

When the memory type (MTYP[2:0] bitfield of the XSPI_CR register) is configured to 0b011 to target an OctaRAM™ memory, the host controller does not support the variable latency requested by the external memory if a refresh collision occurs during the write access. For example, some OctaRAM™ memories, such as ISSI memories, request extra latency cycles for write accesses during refresh collision. In this case, the controller does not sample the DQS input signal during the instruction phase, and cannot detect the extra latency requested by the external memory for the refresh operation.This results in data corruption.

Some OctaRAM™ memories do not request any additional latency for write access during refresh cycles. It is required only when the refresh occurs during a read access. In this case, no issue can be observed.

Workaround

When the application targets an OctaRAM™ memory that requests extra latency cycles for write access during refresh collision, force the fixed latency mode in the configuration register of the external memory. There is no constraint about read access, since both variable and fixed latency modes are supported.

2.4.9 In automatic status-polling and multiplexed modes, the controller does not request the port if less than two bytes are sent per cycle when XSPI_DLR is cleared

Description

Due to FIFO RX pointer mismatches, the controller configured in automatic status-polling mode (FMODE[1:0] = 10) may not be able to request the port ownership to serve the status-polling request to the external memory. As a result, the FIFO is wrongly detected full, thus blocking the request from the controller to the I/O manager.

The issue happens in the following conditions:

- The I/O manager is used in multiplexed mode (to connect two controllers sharing the same port).
- The I/O manager is connected to a PHY.
- The controller is configured in automatic status-polling mode.
- Less than two bytes are sent per CLK cycle.
- Data length register (XSPI_DLR) is cleared.

Workaround

Set the XSPI_DLR to configure the number of bytes to 2 (XSPI_DLR set to 0x0000 0001), and configure the XSPI polling status mask register (XSPI_PSMKR) to mask the second dummy byte

2.4.10 Possible deadlock when a request arrives during the disabling process

Description

If a new transaction request arrives during the disabling process (the ENABLE bit is cleared and the BUSY bit is still set), the following behaviors may be observed, potentially resulting in a deadlock situation:

- The CPU waits for an error response to the transaction, which never arrives.
- The controller waits for the acknowledgment of the new request, which does not arrive since the controller is in the process of being disabled.

Workaround

Before clearing the ENABLE bit, ensure that all current transactions are completed using synchronization barriers.

2.4.11 A read mismatch occurs after disabling the controller during a read transaction

Description

If the controller is disabled (the ENABLE bit is cleared) after a read request and before receiving data (the BUSY bit is still set), the FIFO read pointer may be outdated, resulting in a data mismatch after reenabling the controller.

Workaround

Before clearing the ENABLE bit, ensure that all transactions are completed using the synchronization barriers.

2.5 SDMMC

2.5.1 Command response and receive data end bits not checked

Description

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

Workaround

None.

2.6 ADC

2.6.1 In injected simultaneous mode, stopping regular conversion may delay injected conversion

Description

In dual ADC injected simultaneous mode, an injected conversion may be delayed by a few ADC clock cycles versus the corresponding injected trigger event. This occurs when the injected trigger event coincides with stopping, by application or DMA, an ongoing regular conversion.

Note: The dual ADC injected simultaneous mode is selected by setting the DUAL[4:0] bitfield of the ADCC_CCR register to 0x03 or to 0x05. To stop an ongoing regular conversion, the software sets the ADSTP bit of the ADC_CR register.

Workaround

- Design the software so as to avoid injected trigger events from coinciding with stopping, by application or DMA, an ongoing regular conversion.
- Avoid triggering an injected conversion when the application (ADSTP bit set) or the DMA stops a regular conversion.

2.6.2 In combined regular simultaneous plus alternate trigger mode, stopping injected conversion may delay regular conversion

Description

In dual ADC combined regular simultaneous plus alternate trigger mode, the resumption of an active regular conversion may be delayed by a few ADC clock cycles when an injected trigger event coincides with stopping, by application, an ongoing injected conversion.

Note: The dual ADC combined regular simultaneous plus alternate trigger mode is selected by setting the DUAL[4:0] bitfield of the ADCC_CCR register to 0x02. To stop an ongoing injected conversion, the software sets the JADSTP bit of the ADC_CR register.

Workaround

- Design the application so as to avoid injected trigger events from coinciding with stopping, by software or DMA, an ongoing regular conversion.
- Stop the regular conversion before stopping the injected conversion.

2.6.3 In certain dual ADC modes with regular oversampling continued mode enabled, the JEOC flag is not set at the end of the new injected conversion

Description

In dual ADC regular simultaneous or interleaved mode, when the regular oversampling is enabled in continued mode, and injected conversions are ongoing on both ADCs, if an injected conversion is stopped and a new conversion is started for the same ADC, the JEOC flag is not set at the end of the new injected conversion.

Note: The regular simultaneous and interleaved modes are selected by setting the DUAL[4:0] bitfield of the ADCC_CCR register to 0x06 and 0x07, respectively. The regular oversampling continued mode is enabled by setting the ROVSE bit and clearing the ROVSM bit of the ADC_CFGR2 register. To start a new injected conversion and stop the ongoing one, the software must set the JADSTART and JADSTP bits of the ADC_CR register, respectively. When set, the JEOC flag of the ADC_ISR register indicates the end of the injected channel conversion.

Workaround

Apply one of the following measures:

- Use one of the following modes instead of the regular simultaneous or the interleaved mode:
  - Combined regular simultaneous + injected simultaneous mode (DUAL[4:0] = 0x01)
  - Combined regular simultaneous + alternate trigger mode (DUAL[4:0] = 0x02)
  - Combined interleaved mode + injected simultaneous mode (DUAL[4:0] = 0x03)
- If an injected conversion is already ongoing on one ADC, avoid triggering a new injected conversion on the other ADC after stopping the ongoing conversion.
- Instead of the regular oversampling continued mode, use the resumed mode (ROVSM = 1).

2.6.4 In certain dual ADC modes with DMA one-shot mode enabled, JADSTART and ADSTART may not be cleared if JADSTP and ADSTP are set at the same time

Description

In certain dual ADC modes (DUAL[4:0] = 0x05, 0x06, 0x07, and 0x09 in the ADCC_CCR register), ADSTART and JADSTART bits may not be cleared, if the DMA is configured in one-shot mode, and the stopping of the ongoing injected conversion by software coincides with the stopping of the ongoing regular conversion by DMA.

Note: The DMA one-shot mode is enabled by setting the DMNGT[1:0] bitfield of the ADC_CFGR2 register to 0b01. To stop an injected conversion, the software must set the JADSTP bit of the ADC_CR register.

Workaround

Apply one of the following measures:

- Use one of the following modes instead of the injected simultaneous, regular simultaneous or the interleaved mode:
  - Combined regular simultaneous + injected simultaneous mode (DUAL[4:0] = 0x01)
  - Combined regular simultaneous + alternate trigger mode (DUAL[4:0] = 0x02)
  - Combined interleaved mode + injected simultaneous mode (DUAL[4:0] = 0x03)
- Avoid using DMA one-shot mode to manage regular conversions. Instead, use interrupt or polling.

2.6.5 Shifted master and slave sequence in interleaved discontinuous mode

Description

In dual ADC interleaved mode, when the discontinuous mode is enabled on regular channels, and the next trigger arrives before the end of the sequence, the master and slave interleave sequence may be shifted.

Note: The dual ADC interleaved modes are selected by setting the DUAL[4:0] bitfield of the ADCC_CCR register to 0x03 or 0x07. The discontinuous mode is enabled through the DISCEN bit of the ADC_CFGR1 register.

Workaround

Avoid triggering a new conversion when the ongoing conversion is not complete.

2.6.6 When the ADC clock is derived from the AHB clock, the injected conversion latency is not respected if the injected trigger coincides with the stopping of the regular conversion

Description

When the ADC clock is derived from the AHB clock, and the analog-to-digital conversion is triggered by a timer, the latency between the trigger and the start of the ADC sampling is fixed. When both injected and regular conversions are enabled, if the injected trigger coincides with the stopping of regular conversion, the latency of injected conversions becomes one clock cycle shorter than the expected latency.

Note: To stop an ongoing regular conversion, the software sets the ADSTP bit of the ADC_CR register.

Workaround

Apply one of the following measures:

- Avoid triggering injected conversions when the ADSTP bit is set.
- When an injected trigger is expected, keep the ADSTP bit cleared.

2.6.7 In certain dual modes, the fixed trigger latency for the injected conversions may not be respected

Description

If the application operates in regular simultaneous or interleaved mode (DUAL[4:0] = 0x06 or 0x07 in the ADCC_CCR register), the injected conversions can be activated either on the master or on the slave ADC. When a fixed trigger latency is configured for injected conversions, the regular conversions on both ADCs are interrupted by the injected trigger. In this case, the previous conversion trigger latency and the current conversion trigger latency may differ by one clock cycle.

Workaround

If the regular simultaneous mode or the interleaved mode is used (DUAL[4:0]=0x06 or 0x07), and the application needs injected conversions with a fixed trigger latency for both ADCs, apply one of the following measures:

- Select another dual mode, either DUAL[4:0] = 0x01 or 0x03.
- Make sure that no injected trigger event occurs when regular conversions are stopped, by setting the ADSTP bit in the ADC_CR register.

2.6.8 In simultaneous regular mode, stopping an injected conversion may shift the next regular conversion master and slave timing by one clock cycle

Description

In simultaneous regular mode (DUAL[4:0] = 0x06 in the ADCC_CCR register), injected conversions can be activated either on ADC1 or ADC2. When regular conversions are ongoing, if an injected conversion stop (JADSTP) occurs at the same time as an injected trigger event, then ADC1 and ADC2 timing may be shifted by one clock cycle.

Workaround

Apply one of the following measures:

- Avoid the following events from occurring simultaneously:
  - Triggering injected conversions
  - Stopping injected conversions
  - Enabling regular conversion in simultaneous regular mode (DUAL[4:0] = 0x06)
- Stop regular conversions before stopping injected conversions.
- Stop injected trigger source before stopping injected conversions.
- Configure DUAL[4:0] to 0x01 or 0x02.

2.7 LTDC

2.7.1 Layers cannot read YUV420 multibuffer data

Description

The YUV semiplanar and full-planar modes of the layers are not functional.

The YUV coplanar and the RGB modes can be used.

Workaround

None.

2.8 GPU2D

2.8.1 Occasional writing miss to frame buffer with slow memories

Description

The GPU operating slowly due to slow memories may occasionally fail to write isolated single-pixel data to the frame buffer.

Workaround

None

2.9 LPTIM

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

Description

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

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

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

Workaround

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

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

Description

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

Workaround

To avoid this issue, the following formula must be respected:

{ARR, CMP} ≥ KER_CLK / (2* APB_CLK),

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

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

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

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

{CMP, ARR} ≥ 16 MHz / (2 * 1 MHz)

→ ARR must be ≥ 8 and CMP must be ≥ 8

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

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

Description

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

Workaround

None.

2.9.4 LPTIM4 remains active after user code starts in development mode

Description

During a normal power-on reset, there are no issues because the disabling of the LPTIMER4 ROM function is executed while the debugger is connected. However, if there is a software or pad reset, the debugger takes control in the BOOTROM before the disabling of the LPTIMER4 ROM function has been executed.

Workaround

None.

2.10 RTC

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

Description

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

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

Workaround

None.

2.11 I2C

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

Description

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

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

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

Workaround

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

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

2.11.2 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.

2.12 USART

2.12.1 Received data may be corrupted upon clearing the ABREN bit

Description

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

Workaround

Do not clear the ABREN bit.

2.12.2 Noise error flag set while ONEBIT is set

Description

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

Workaround

None.

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

2.13 LPUART

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

Description

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

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

Workaround

Apply one of the following measures:

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

2.14 FDCAN

2.14.1 Desynchronization under specific condition with edge filtering enabled

Description

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

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

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

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

Workaround

Disable edge filtering or wait for frame retransmission.

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

Description

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

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

Workaround

Apply one of the following measures:

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

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

2.14.3 DAR mode transmission failure due to lost arbitration

Description

In DAR mode, the transmission may fail due to lost arbitration at the first two identifier bits.

Workaround

Upon failure, clear the corresponding Tx buffer transmission request bit TRPx of the FDCAN_TXBRP register and set the corresponding cancellation finished bit CFx of the FDCAN_TXBCF register, then restart the transmission.

2.15 USBPHYC

2.15.1 External VBUS detection method malfunctioning

Description

The control bit has been permanently tied to 1 (EXTERNAL-VBUS) at the PHY boundary, rendering the two balls (OTG1_VBUS and OTG2_VBUS) useless.

Workaround

None.

2.16 UCPD

2.16.1 Ordered set with multiple errors in a single K-code is reported as invalid

Description

The Power Delivery standard allows considering a received ordered set as valid even if it contains errors, provided that they only affect a single K-code of the ordered set.

In the reference manual, the RXSOP3OF4 flag is specified to signal errors affecting a single K-code, the RXERR flag to signal errors in multiple K-codes.

However, the behaviour does not conform with the reference manual. The RXSOP3OF4 flag is only raised in the case of a single error. The RXERR flag is raised in the case of multiple errors, regardless of whether they affect a single K-code or multiple K-codes. As a consequence, ordered sets with multiple errors in a single K-code are reported by the device as invalid although the Power Delivery standard allows considering them as valid.

Despite this non-conformity versus its reference manual, the device remains compliant with the Power Delivery standard.

Workaround

None.

2.16.2 Rp value drift causes incorrect source current capabiliy detection

Description

The UCPD*x*_CC line termination Rp value may change over time. This can cause incorrect detection of the source current capability. During a Power Delivery contract, Rp is used to avoid collisions. However, the system might mistakenly show Rp1.5A (sink transmit NoGo state or SinkTxNg) instead of Rp3.0A (sink transmit Ok state or SinkTxOk). This error can completely block sink-initiated atomic message sequences.

Workaround

None.

2.17 ETH

2.17.1 Incorrect DMA transfer state in TEB[2:0] and REB[2:0] on bus error

Description

When a bus error occurs during data transfer or descriptor access from Rx DMA or Tx DMA, the ETH controller updates the transfer state in the REB[2:0] (Rx DMA error) or TEB[2:0] (Tx DMA error) status bitfield of the corresponding DMA channel *x* status register (ETH_DMAC*x*SR)

The software uses the bus error indicated in REB[2:0] or TEB[2:0] to perform diagnostics, tracks the number of bus errors depending on the type of DMA transfer, and takes corrective actions other than a software reset.

However, when the error response to an Rx DMA write data transfer or an Rx or Tx DMA posted descriptor write transfer is received after the DMA state has already changed to another transfer type, the DMA updates the error response status in REB[2:0] or TEB[2:0] for the current transfer type, which is incorrect.

As a consequence, the software may diagnose or track incorrectly and counter-act inappropriately, which leads to frequent bus errors.

This issue has no functional impact because only the DMA transfer state is incorrect.

Workaround

Perform diagnostics on the entire system to identify the cause of the bus error. However, this may consume more instruction cycles and impact the software performance.

2.17.2 Transmission status is not updated in ETH_MACFPECSR for back-to-back frame preemption Verify or Respond mPackets

Description

The software can set the SVER and SRSP bits of the FPE control and status register (ETH_MACFPECSR) to request the MAC to transmit frame preemption Verify and Respond mPackets, respectively. Upon successful transmission of these frames, the MAC clears the SVER and SRSP bits and sets the TVER and TRSP bits of the ETH_MACFPECSR register.

However, the following issues arise:

- When Verify and Respond mPackets are triggered by the software within a very short period, the transmission status of the first triggered packet is not updated in the FPE control and status register (ETH_MACFPECSR).
- When both Verify and Respond mPackets are triggered simultaneously, the frame preemption Respond mPacket is transmitted first, causing its transmission status (TRSP bit) not to be updated in ETH_MACFPECSR.

Workaround

Do not trigger back-to-back frame preemption Verify and Respond mPackets. Instead, wait for the completion of the already triggered packet before triggering the next packet.

2.17.3 Incorrect routing of Rx packet or incorrect splitting of header payload on preemption Rx queue overflow

Description

The preempted packets are received by the MAC receiver. When the application bandwidth is not sufficient, a preemption Rx queue overflow occurs. Due to the issue, the header status associated with all subsequent preemption fragments or express packets is from the previous preemption fragment or express packet. This results in an incorrect header-payload splitting or Rx DMA routing for the subsequent packets. The application connected to the Rx DMA consequently drops packets or assembles them incorrectly. Data loss or retransmission impacts performance.

The issue is not observed when the bandwidth on the application side is sufficient to prevent the Rx queue overflow.

Workaround

Program greater size for the preemption Rx queues if heavy traffic is observed.

2.17.4 MAC fails to identify PTP SYNC and Follow_Up messages with peer delay reserved multicast address in 802.1AS mixed mode

Description

When the AV8021ASMEN bit of the timestamp control register (ETH_MACTSCR) is set, the MAC operates in the IEEE 802.1AS mixed mode. The delay request response and the peer delay mechanism are both used for PTP time correction. To capture the ingress timestamp, the MAC identifies the PTP SYNC and Follow_Up messages that contain a PTP peer delay reserved multicast destination address.

The mixed mode can be programmed by any of the following settings in the ETH_MACTSCR register:

- AV8021ASMEN = 1, SNAPTYPSEL[1:0] = 01 and TSEVNTENA = 0
- AV8021ASMEN = 1, SNAPTYPSEL[1:0] = 01, TSMSTRENA = 0, and TSEVNTENA = 1

However, due to the issue, when the MAC receives the PTP SYNC and Follow_Up messages with PTP peer delay reserved multicast destination address, it does not identify the packets as PTP packets.

The MAC identifies the received PTP SYNC and Follow_Up messages only if they contain a PTP primary multicast address.

Workaround

If the mixed mode is required, set the TSENALL bit of the ETH_MACTSCR register to enable the MAC to capture the timestamp for all the received packets. The software must identify the PTP SYNC and Follow_Up messages and associate the timestamp status provided by the MAC.

2.17.5 VLAN tag filter fail packet queuing feature is enabled without considering the RA bit of the ETH_MACPFR register

Description

When the VFFQE bit of the Rx queue control register (ETH_MACRXQCR) and the RA bit of the packet filtering control register (ETH_MACPFR) are both set, the tagged packets that fail the destination address filtering, the source address filtering, or the VLAN tag filtering are routed to the Rx queue selected through the VFFQ bit.

However, when the VFFQE bit is set, a tagged packet that fails the destination address filtering, the source address filtering, or the VLAN tag filtering, is incorrectly routed to the Rx queue selected through VFFQ, irrespective of the value of the RA bit. In addition, it may get routed to an unintended receive DMA if the static DMA mapping is enabled for the receive queue.

This issue is not observed when the VTFE bit in the ETH_MACPFR is set, as the received packet that fails the VLAN tag filtering is dropped in the MAC.

Workaround

Do not set the VFFQE bit when the RA bit is cleared.

2.17.6 Incorrect gate control list switching for intermediate cycles when CTR is less than the GCL execution time

Description

The EST (enhancements to scheduled traffic) scheduler switches to the next gate control list (GCL) after executing the current GCL, regardless of the difference between the start time of the next GCL and the time when the current GCL rows are completely executed.

If the GCL execution takes longer than the cycle time, the GCL is truncated at the cycle time, and the subsequent loop begins at BTR + N × cycle time, where N is an integer that represents the iteration number.

However, the GCL incorrectly updates the internal BTR twice, and skips the execution of the next GCL loop. This issue arises if one of the following conditions is met:

- CTR value < sum of time intervals of fully executed GCL rows, or
- CTR value > sum of time intervals of fully executed GCL rows, and
  CTR value < sum of time intervals of fully executed GCL rows + 8 PTP clock periods

Note: The CTR is the value initially configured by the software.

Workaround

Apply one of the following measures:

- CTR value > sum of time intervals of fully executed GCL rows + 8 PTP clock periods
- CTR = sum of time intervals of fully executed GCL rows

Important security notice

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

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

Revision history

Table 6. **Document revision history**

Date	Version	Changes
03-Dec-2024	1	Initial release.
25-Sep-2025	2	Added errata: • **System:** Incorrect reset behavior when using NRST pin • Access to SYSCFG_VDDIOxCCCR and SYSCFG_VDDIOxCCSR unduly depends on VDDIOxSV • I/O compensation may deform output signal rise and fall edges • STNoC AXI deadlock during GFXMMU read of 24 bpp frame buffer • **XSPI:** In automatic status-polling and multiplexed modes, the controller does not request the port if less than two bytes are sent per cycle when XSPI_DLR is cleared • Possible deadlock when a request arrives during the disabling process • A read mismatch occurs after disabling the controller during a read transaction • **ADC:** In certain dual modes, the fixed trigger latency for the injected conversions may not be respected • In simultaneous regular mode, stopping an injected conversion may shift the next regular conversion master and slave timing by one clock cycle • **LPTIM:** LPTIM4 remains active after user code starts in development mode • **USBPD:** Rp value drift causes incorrect source current capabiliy detection Modified errata: • **System:** APB*x* clock prescaling fails • Desynchronization of VENC and DCMIPP when streaming Removed errata: • **LTDC:** Ongoing AXI write never completes if disabling LTDC • **VENC:** VENC hardware self-reset after internal timeout is unstable
14-Nov-2025	3	Added errata: • Boot ROM execution failure after system reset due to wrong clocking of peripherals • Recovery from clock switching failure
26-May-2026	4	Updated: • Incorrect reset behavior when using NRST pin Added errata: • Incorrect definition of HSERDY flag in reference manual • Incorrect definition of the RTCPRE divider formula in reference manual • Incorrect description of the OTP16 BootROM_CONFIG_7 in reference manual • Incorrect HSE minimum values in OTP16 BootROM_CONFIG_7 and in RCC section of the reference manual • Missing default value for rng_htcr_value in OPT19 BootROM_CONFIG_10 • Incorrect operational status of PVM monitoring for V_{DDIO2}, V_{DDIO3}, V_{DDIO4}, V_{DDIO5}, USB33, and V_{DDCORE} in reference manual or datasheet
07-Aug-2026	5	Added silicon revision X.

Contents

IMPORTANT NOTICE – READ CAREFULLY

STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice.

In the event of any conflict between the provisions of this document and the provisions of any contractual arrangement in force between the purchasers and ST, the provisions of such contractual arrangement shall prevail.

The purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST’s terms and conditions of sale in place at the time of order acknowledgment.

The purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of the purchasers’ products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

If the purchasers identify an ST product that meets their functional and performance requirements but that is not designated for the purchasers' market segment, the purchasers shall contact ST for more information.

ST and the ST logo are trademarks of ST. For additional information about ST trademarks, refer to www.st.com/trademarks. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.