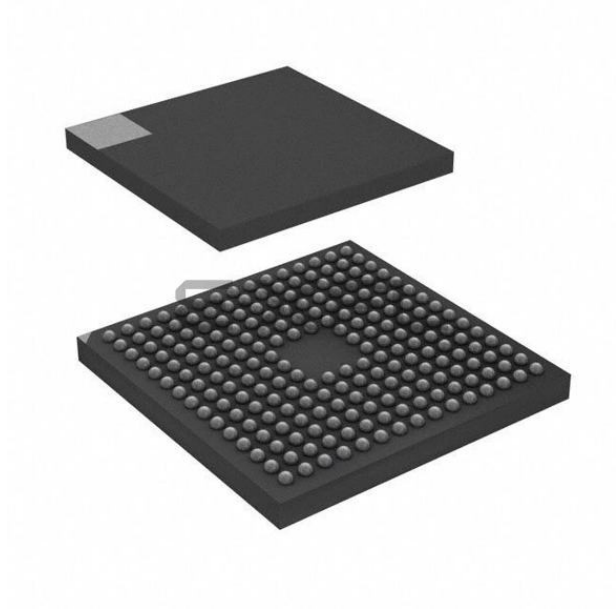


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	ARM® Cortex®-M7
Core Size	32-Bit Single-Core
Speed	216MHz
Connectivity	CANbus, Ethernet, HDMI-CEC, I²C, IrDA, LINbus, MMC/SD, SAI, SPDIFRX, SPI, UART/USART, USB OTG
Peripherals	Brown-out Detect/Reset, DMA, LCD, POR, PWM, WDT
Number of I/O	168
Program Memory Size	1MB (1M x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	320K x 8
Voltage - Supply (Vcc/Vdd)	1.7V ~ 3.6V
Data Converters	A/D 24x12b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	216-TFBGA
Supplier Device Package	216-TFBGA (13x13)
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/stm32f756ngh6

3 Embedded Flash memory (FLASH)

3.1 Introduction

The Flash memory interface manages Cortex®-M7 AXI and TCM accesses to the Flash memory. It implements the erase and program Flash memory operations and the read and write protection mechanisms.

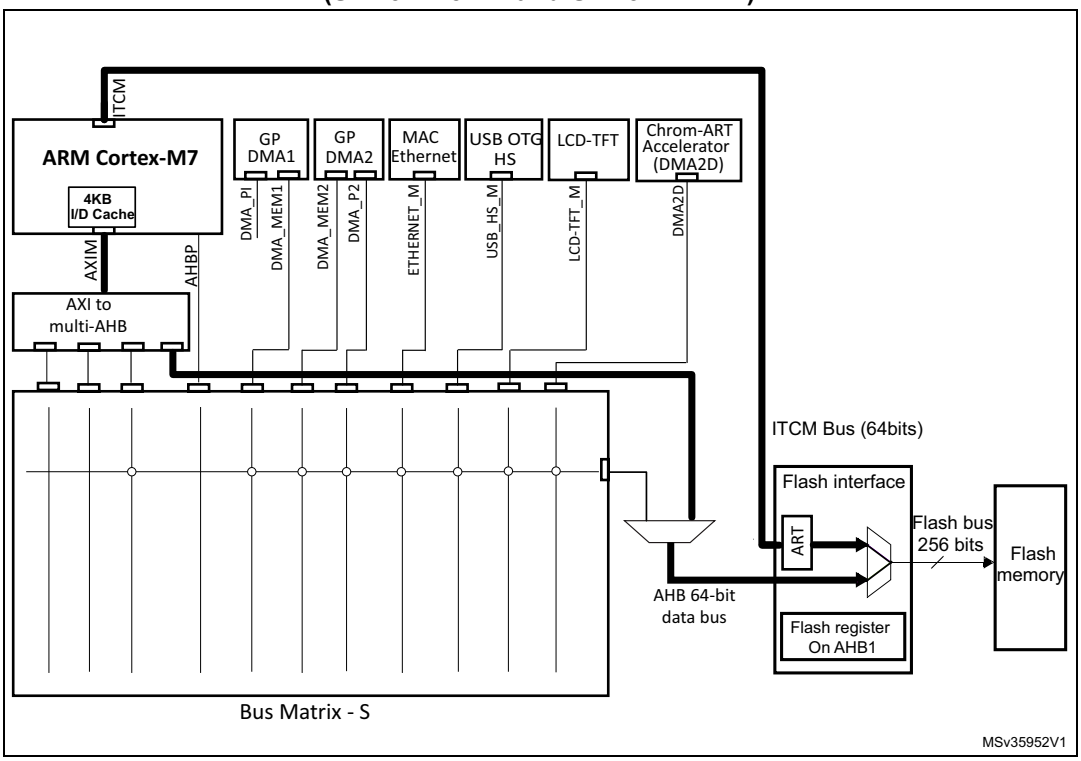
The Flash memory interface accelerates code execution with a system of instruction prefetch and cache lines on ITCM interface (ART Accelerator™).

3.2 Flash main features

- Flash memory read operations
- Flash memory program/erase operations
- Read / write protections
- 64 cache lines of 256 bits on ITCM interface (ART Accelerator™)
- Prefetch on TCM instruction code

Figure 2 shows the Flash memory interface connection inside the system architecture.

Figure 2. Flash memory interface connection inside system architecture (STM32F75xxx and STM32F74xxx)



Bit 16 **HSEON**: HSE clock enable

Set and cleared by software.

Cleared by hardware to stop the HSE oscillator when entering Stop or Standby mode. This bit cannot be reset if the HSE oscillator is used directly or indirectly as the system clock.

0: HSE oscillator OFF

1: HSE oscillator ON

Bits 15:8 **HSICAL[7:0]**: Internal high-speed clock calibration

These bits are initialized automatically at startup.

Bits 7:3 **HSITRIM[4:0]**: Internal high-speed clock trimming

These bits provide an additional user-programmable trimming value that is added to the HSICAL[7:0] bits. It can be programmed to adjust to variations in voltage and temperature that influence the frequency of the internal HSI RC.

Bit 2 Reserved, must be kept at reset value.

Bit 1 **HSIRDY**: Internal high-speed clock ready flag

Set by hardware to indicate that the HSI oscillator is stable. After the HSION bit is cleared, HSIRDY goes low after 6 HSI clock cycles.

0: HSI oscillator not ready

1: HSI oscillator ready

Bit 0 **HSION**: Internal high-speed clock enable

Set and cleared by software.

Set by hardware to force the HSI oscillator ON when leaving the Stop or Standby mode or in case of a failure of the HSE oscillator used directly or indirectly as the system clock. This bit cannot be cleared if the HSI is used directly or indirectly as the system clock.

0: HSI oscillator OFF

1: HSI oscillator ON

Bit 31 Reserved, must be kept at reset value.

Bits 30:28 **PLLI2SR[2:0]:** PLLI2S division factor for I2S clocks

These bits are set and cleared by software to control the I2S clock frequency. These bits should be written only if the PLLI2S is disabled. The factor must be chosen in accordance with the prescaler values inside the I2S peripherals, to reach 0.3% error when using standard crystals and 0% error with audio crystals. For more information about I2S clock frequency and precision, refer to *Section 32.7.4: Start-up description* in the I2S chapter.

Caution: The I2Ss requires a frequency lower than or equal to 192 MHz to work correctly.

I2S clock frequency = VCO frequency / PLLR with 2 ≤ PLLR ≤ 7
000: PLLR = 0, wrong configuration
001: PLLR = 1, wrong configuration
010: PLLR = 2
...
111: PLLR = 7

Bits 27:24 **PLLI2SQ[3:0]**: PLLI2S division factor for SAIs clock

These bits are set and cleared by software to control the SAIs clock frequency.
They should be written when the PLLI2S is disabled.
SAI clock frequency = VCO frequency / PLLI2SQ with 2 <= PLLI2SIQ <= 15
0000: PLLI2SQ = 0, wrong configuration
0001: PLLI2SQ = 1, wrong configuration
0010: PLLI2SQ = 2
0011: PLLI2SQ = 3
0100: PLLI2SQ = 4
0101: PLLI2SQ = 5
...
1111: PLLI2SQ = 15

Bits 23:18 Reserved, must be kept at reset value.

Bits 17:16 **PLLI2SP[1:0]**: PLLI2S division factor for SPDIFRX clock

These bits are set and cleared by software to control the SPDIFRX clock. These bits can be written only if the PLLI2S is disabled.
The factor must be chosen in accordance with the prescaler values inside the SPDIF to reach an audio clock close to 44 kHz or 48 kHz according to the SPDIF mode
SPDIF clock frequency = VCO frequency / PLLI2SP with PLLI2S P = 2, 4, 6, or 8
00: PLLI2SP = 2
01: PLLI2SP = 4
10: PLLI2SP = 6
11: PLLI2SP = 8

Bit 2 **DMDIS**: Direct mode disable

This bit is set and cleared by software. It can be set by hardware.

0: Direct mode enabled

1: Direct mode disabled

This bit is protected and can be written only if EN is ‘0’.

This bit is set by hardware if the memory-to-memory mode is selected (DIR bit in DMA_SxCR are “10”) and the EN bit in the DMA_SxCR register is ‘1’ because the direct mode is not allowed in the memory-to-memory configuration.

Bits 1:0 **FTH[1:0]**: FIFO threshold selection

These bits are set and cleared by software.

00: 1/4 full FIFO

01: 1/2 full FIFO

10: 3/4 full FIFO

11: full FIFO

These bits are not used in the direct mode when the DMIS value is zero.

These bits are protected and can be written only if EN is ‘0’.

9.5.2 DMA2D Interrupt Status Register (DMA2D_ISR)

Address offset: 0x0004

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CEIF	CTCIF	CAEIF	TWIF	TCIF	TEIF
										r	r	r	r	r	r

Bits 31:6 Reserved, must be kept at reset value

Bit 5 **CEIF**: Configuration error interrupt flag

This bit is set when the START bit of DMA2D_CR, DMA2DFGPFCCR or DMA2D_BGPFCCR is set and a wrong configuration has been programmed.

Bit 4 **CTCIF**: CLUT transfer complete interrupt flag

This bit is set when the CLUT copy from a system memory area to the internal DMA2D memory is complete.

Bit 3 **CAEIF**: CLUT access error interrupt flag

This bit is set when the CPU accesses the CLUT while the CLUT is being automatically copied from a system memory to the internal DMA2D.

Bit 2 **TWIF**: Transfer watermark interrupt flag

This bit is set when the last pixel of the watermarked line has been transferred.

Bit 1 **TCIF**: Transfer complete interrupt flag

This bit is set when a DMA2D transfer operation is complete (data transfer only).

Bit 0 **TEIF**: Transfer error interrupt flag

This bit is set when an error occurs during a DMA transfer (data transfer or automatic CLUT loading).

9.5.19 DMA2D line watermark register (DMA2D_LWR)

Address offset: 0x0048

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LW[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:16 Reserved, must be kept at reset value

Bits 15:0 **LW[15:0]**: Line watermark

These bits allow to configure the line watermark for interrupt generation.
An interrupt is raised when the last pixel of the watermarked line has been transferred.
These bits can only be written when data transfers are disabled. Once the transfer has started, they are read-only.

9.5.20 DMA2D AHB master timer configuration register (DMA2D_AMTCR)

Address offset: 0x004C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DT[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	Res.	EN
rw	rw	rw	rw	rw	rw	rw	rw								rw

Bits 31:16 Reserved

Bits 15:8 **DT[7: 0]**: Dead Time

Dead time value in the AHB clock cycle inserted between two consecutive accesses on the AHB master port. These bits represent the minimum guaranteed number of cycles between two consecutive AHB accesses.

Bits 7:1 Reserved

Bit 0 **EN**: Enable

Enables the dead time functionality.

PSRAM/SRAM, non-multiplexed I/Os

Table 59. Non-multiplexed I/Os PSRAM/SRAM

FMC signal name	I/O	Function
CLK	O	Clock (only for PSRAM synchronous access)
A[25:0]	O	Address bus
D[31:0]	I/O	Data bidirectional bus
NE[x]	O	Chip Select, x = 1..4 (called NCE by PSRAM (Cellular RAM i.e. CRAM))
NOE	O	Output enable
NWE	O	Write enable
NL(= NADV)	O	Address valid only for PSRAM input (memory signal name: NADV)
NWAIT	I	PSRAM wait input signal to the FMC
NBL[3:0]	O	Byte lane output. Byte 0 to Byte 3 control (Upper and lower byte enable)

The maximum capacity is 512 Mbits.

PSRAM, 16-bit multiplexed I/Os

Table 60. 16-Bit multiplexed I/O PSRAM

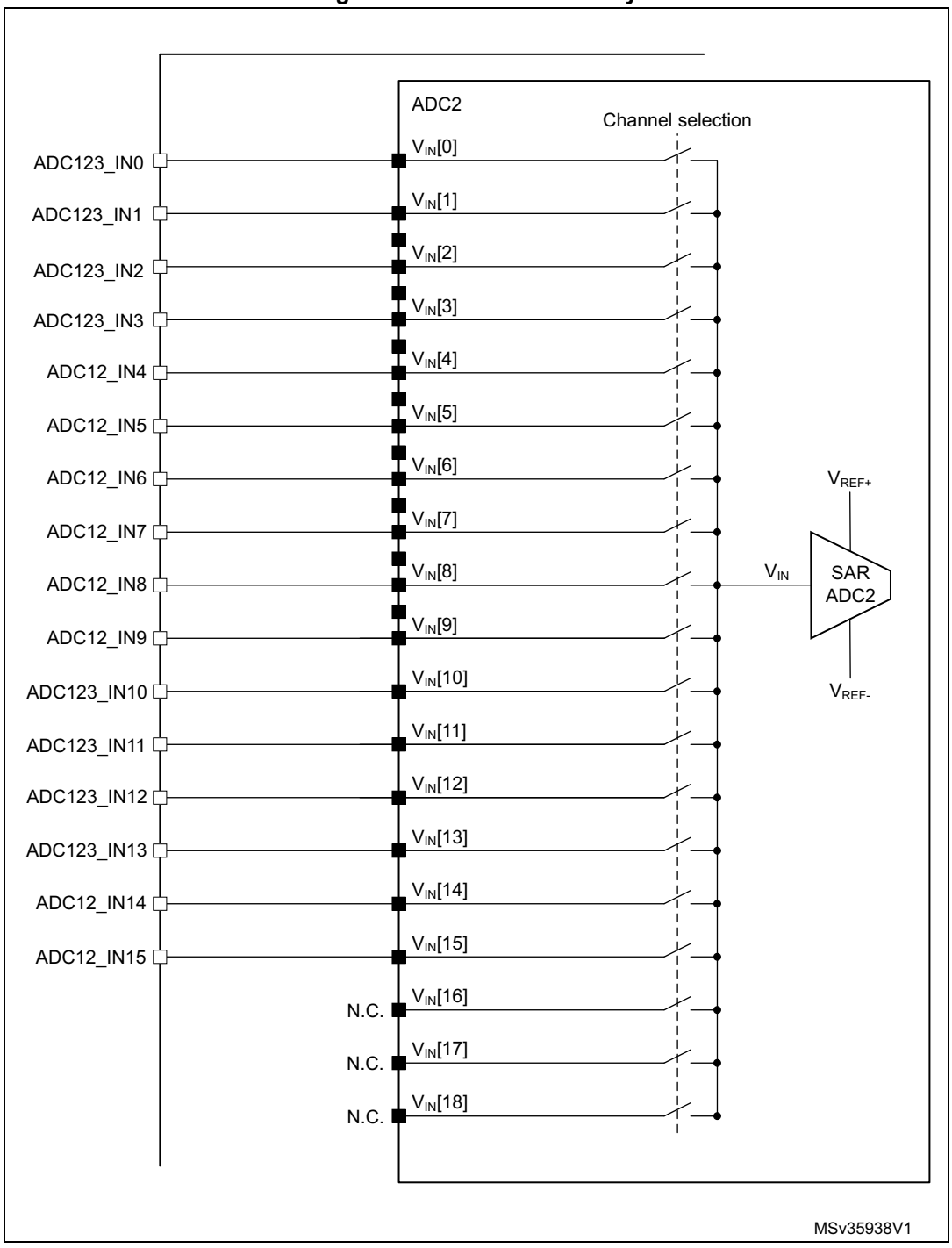
FMC signal name	I/O	Function
CLK	O	Clock (for synchronous access)
A[25:16]	O	Address bus
AD[15:0]	I/O	16-bit multiplexed, bidirectional address/data bus (the 16-bit address A[15:0] and data D[15:0] are multiplexed on the databus)
NE[x]	O	Chip Select, x = 1..4 (called NCE by PSRAM (Cellular RAM i.e. CRAM))
NOE	O	Output enable
NWE	O	Write enable
NL(= NADV)	O	Address valid PSRAM input (memory signal name: NADV)
NWAIT	I	PSRAM wait input signal to the FMC
NBL[1:0]	O	Byte lane output. Byte 0 and Byte 1 control (upper and lower byte enable)

The maximum capacity is 512 Mbits (26 address lines).

13.5.2 Supported memories and transactions

Table 61 below shows an example of the supported devices, access modes and transactions when the memory data bus is 16-bit wide for NOR Flash memory, PSRAM and SRAM. The transactions not allowed (or not supported) by the FMC are shown in gray in this example.

Figure 71. ADC2 connectivity



Bit 19 **JSTRT3:** Injected channel Start flag of ADC3
This bit is a copy of the JSTRT bit in the ADC3_SR register.

Bit 18 **JEOC3:** Injected channel end of conversion of ADC3
This bit is a copy of the JEOC bit in the ADC3_SR register.

Bit 17 **EOC3:** End of conversion of ADC3
This bit is a copy of the EOC bit in the ADC3_SR register.

Bit 16 **AWD3:** Analog watchdog flag of ADC3
This bit is a copy of the AWD bit in the ADC3_SR register.

Bits 15:14 Reserved, must be kept at reset value.

Bit 13 **OVR2:** Overrun flag of ADC2
This bit is a copy of the OVR bit in the ADC2_SR register.

Bit 12 **STRT2:** Regular channel Start flag of ADC2
This bit is a copy of the STRT bit in the ADC2_SR register.

Bit 11 **JSTRT2:** Injected channel Start flag of ADC2
This bit is a copy of the JSTRT bit in the ADC2_SR register.

Bit 10 **JEOC2:** Injected channel end of conversion of ADC2
This bit is a copy of the JEOC bit in the ADC2_SR register.

Bit 9 **EOC2:** End of conversion of ADC2
This bit is a copy of the EOC bit in the ADC2_SR register.

Bit 8 **AWD2:** Analog watchdog flag of ADC2
This bit is a copy of the AWD bit in the ADC2_SR register.

Bits 7:6 Reserved, must be kept at reset value.

Bit 5 **OVR1:** Overrun flag of ADC1
This bit is a copy of the OVR bit in the ADC1_SR register.

Bit 4 **STRT1:** Regular channel Start flag of ADC1
This bit is a copy of the STRT bit in the ADC1_SR register.

Bit 3 **JSTRT1:** Injected channel Start flag of ADC1
This bit is a copy of the JSTRT bit in the ADC1_SR register.

Bit 2 **JEOC1:** Injected channel end of conversion of ADC1
This bit is a copy of the JEOC bit in the ADC1_SR register.

Bit 1 **EOC1:** End of conversion of ADC1
This bit is a copy of the EOC bit in the ADC1_SR register.

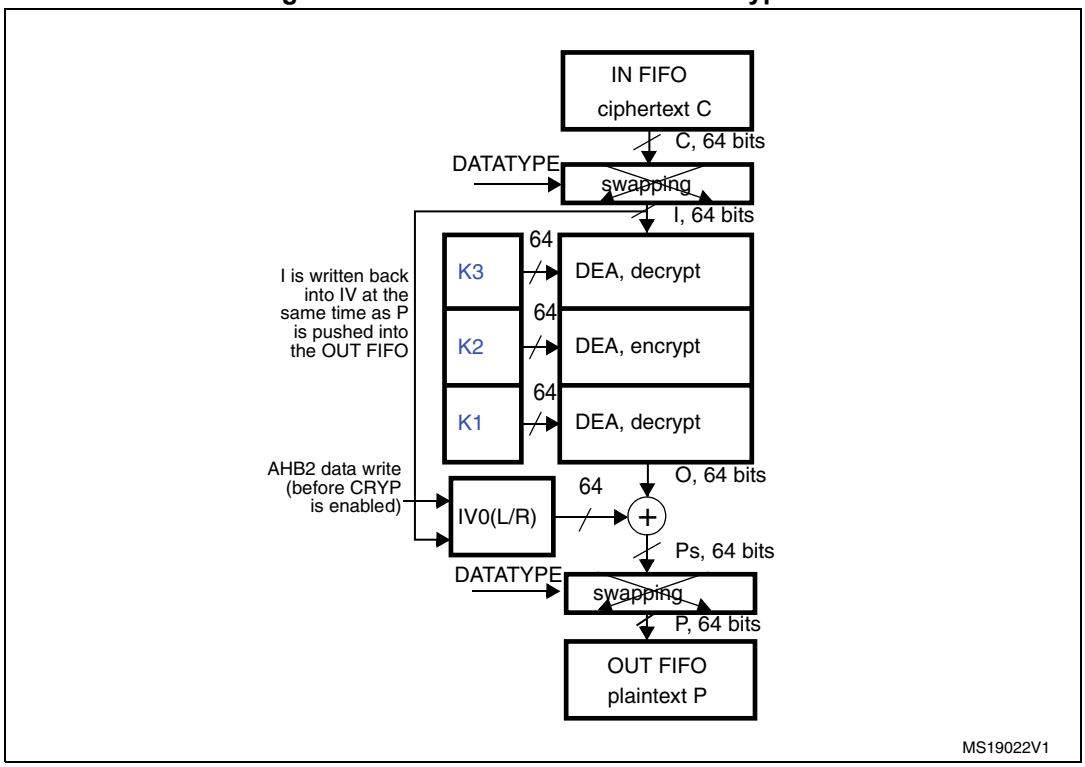
Bit 0 **AWD1:** Analog watchdog flag of ADC1
This bit is a copy of the AWD bit in the ADC1_SR register.

15.13.16 ADC common control register (ADC_CCR)

Address offset: 0x04 (this offset address is relative to ADC1 base address + 0x300)

Reset value: 0x0000 0000

Figure 120. DES/TDES-CBC mode decryption



1. K: key; C: cipher text; I: input block; O: output block; Ps: plain text before swapping (when decoding) or after swapping (when encoding); P: plain text; IV: initialization vectors.

20.3.3 AES cryptographic core

The AES cryptographic core consists of three components:

- The AES algorithm (AEA: advanced encryption algorithm)
- Multiple keys
- Initialization vector(s) or Nonce

The AES utilizes keys of 3 possible lengths: 128, 192 or 256 bits and, depending on the operation mode used, zero or one 128-bit initialization vector (IV).

The basic processing involved in the AES is as follows: an input block of 128 bits is read from the input FIFO and sent to the AEA to be encrypted using the key (K0...3). The key format depends on the key size:

- If Key size = 128: Key = [K3 K2]
- If Key size = 192: Key = [K3 K2 K1]
- If Key size = 256: Key = [K3 K2 K1 K0]

where Kx=[KxR KxL],R=right, L=left

According to the mode implemented, the resultant output block is used to calculate the ciphertext.

FIPS PUB 197 (November 26, 2001) provides a thorough explanation of the processing involved in the four operation modes supplied by the AES core: AES-ECB encryption, AES-ECB decryption, AES-CBC encryption and AES-CBC decryption.This reference manual only gives a brief explanation of each mode.

Bit 24 **BK2E**: Break 2 enable

This bit enables the complete break 2 protection (including all sources connected to bk_acth and BKIN sources, as per *Figure 252: Break and Break2 circuitry overview*).

0: Break2 function disabled

1: Break2 function enabled

Note: The BRKIN2 must only be used with OSSR = OSSI = 1.

Note: This bit cannot be modified when LOCK level 1 has been programmed (LOCK bits in TIMx_BDTR register).

Note: Any write operation to this bit takes a delay of 1 APB clock cycle to become effective.

Bits 23:20 **BK2F[3:0]**: Break 2 filter

This bit-field defines the frequency used to sample BRK2 input and the length of the digital filter applied to BRK2. The digital filter is made of an event counter in which N consecutive events are needed to validate a transition on the output:

0000: No filter, BRK2 acts asynchronously

0001: $f_{SAMPLING}=f_{CK\_INT}$, N=2

0010: $f_{SAMPLING}=f_{CK\_INT}$, N=4

0011: $f_{SAMPLING}=f_{CK\_INT}$, N=8

0100: $f_{SAMPLING}=f_{DTS}/2$, N=6

0101: $f_{SAMPLING}=f_{DTS}/2$, N=8

0110: $f_{SAMPLING}=f_{DTS}/4$, N=6

0111: $f_{SAMPLING}=f_{DTS}/4$, N=8

1000: $f_{SAMPLING}=f_{DTS}/8$, N=6

1001: $f_{SAMPLING}=f_{DTS}/8$, N=8

1010: $f_{SAMPLING}=f_{DTS}/16$, N=5

1011: $f_{SAMPLING}=f_{DTS}/16$, N=6

1100: $f_{SAMPLING}=f_{DTS}/16$, N=8

1101: $f_{SAMPLING}=f_{DTS}/32$, N=5

1110: $f_{SAMPLING}=f_{DTS}/32$, N=6

1111: $f_{SAMPLING}=f_{DTS}/32$, N=8

Note: This bit cannot be modified when LOCK level 1 has been programmed (LOCK bits in TIMx_BDTR register).

Bits 19:16 **BKF[3:0]**: Break filter

This bit-field defines the frequency used to sample BRK input and the length of the digital filter applied to BRK. The digital filter is made of an event counter in which N consecutive events are needed to validate a transition on the output:

0000: No filter, BRK acts asynchronously

0001: $f_{SAMPLING}=f_{CK\_INT}$, N=2

0010: $f_{SAMPLING}=f_{CK\_INT}$, N=4

0011: $f_{SAMPLING}=f_{CK\_INT}$, N=8

0100: $f_{SAMPLING}=f_{DTS}/2$, N=6

0101: $f_{SAMPLING}=f_{DTS}/2$, N=8

0110: $f_{SAMPLING}=f_{DTS}/4$, N=6

0111: $f_{SAMPLING}=f_{DTS}/4$, N=8

1000: $f_{SAMPLING}=f_{DTS}/8$, N=6

1001: $f_{SAMPLING}=f_{DTS}/8$, N=8

1010: $f_{SAMPLING}=f_{DTS}/16$, N=5

1011: $f_{SAMPLING}=f_{DTS}/16$, N=6

1100: $f_{SAMPLING}=f_{DTS}/16$, N=8

1101: $f_{SAMPLING}=f_{DTS}/32$, N=5

1110: $f_{SAMPLING}=f_{DTS}/32$, N=6

1111: $f_{SAMPLING}=f_{DTS}/32$, N=8

Note: This bit cannot be modified when LOCK level 1 has been programmed (LOCK bits in TIMx_BDTR register).

USARTDIV is an unsigned fixed point number that is coded on the USART_BRR register.

- When OVER8 = 0, BRR = USARTDIV.
- When OVER8 = 1
  - BRR[2:0] = USARTDIV[3:0] shifted 1 bit to the right.
  - BRR[3] must be kept cleared.
  - BRR[15:4] = USARTDIV[15:4]

Note: The baud counters are updated to the new value in the baud registers after a write operation to USART_BRR. Hence the baud rate register value should not be changed during communication.

In case of oversampling by 16 or 8, USARTDIV must be greater than or equal to 0d16.

How to derive USARTDIV from USART_BRR register values

Example 1

To obtain 9600 baud with f_{CK} = 8 MHz.

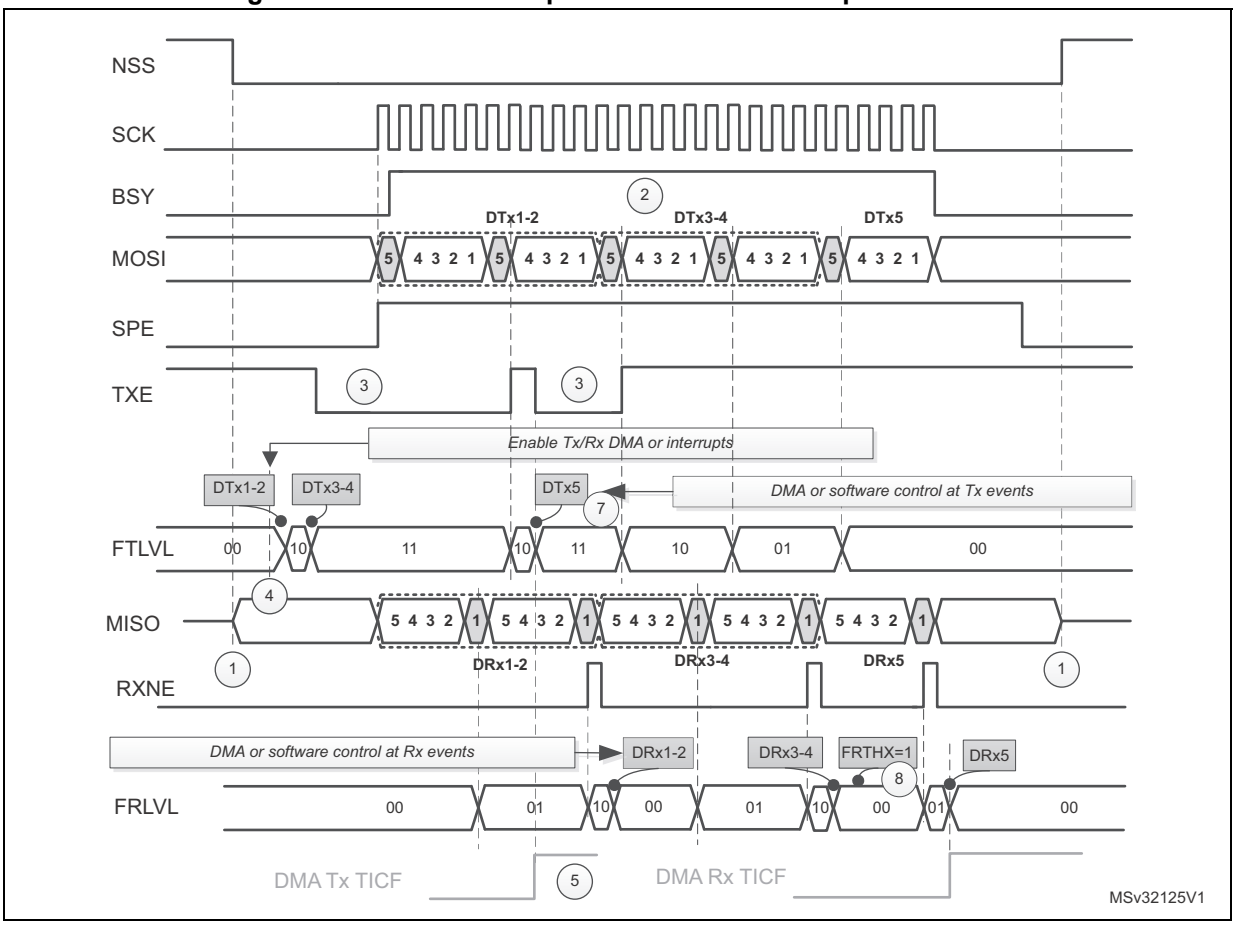
- In case of oversampling by 16:
  USARTDIV = 8 000 000/9600
  BRR = USARTDIV = 833d = 0341h
- In case of oversampling by 8:
  USARTDIV = 2 * 8 000 000/9600
  USARTDIV = 1666,66 (1667d = 683h)
  BRR[3:0] = 3h >> 1 = 1h
  BRR = 0x681

Example 2

To obtain 921.6 Kbaud with f_{CK} = 48 MHz.

- In case of oversampling by 16:
  USARTDIV = 48 000 000/921 600
  BRR = USARTDIV = 52d = 34h
- In case of oversampling by 8:
  USARTDIV = 2 * 48 000 000/921 600
  USARTDIV = 104 (104d = 68h)
  BRR[3:0] = USARTDIV[3:0] >> 1 = 8h >> 1 = 4h
  BRR = 0x64

Figure 354. Master full-duplex communication in packed mode



Assumptions for master full-duplex communication in packed mode example:

- Data size = 5 bit
- Read/write FIFO is performed mostly by 16-bit access
- FRXTH=0

If DMA is used:

- Number of Tx frames to be transacted by DMA is set to 3
- Number of Rx frames to be transacted by DMA is set to 3
- PSIZE for both Tx and Rx DMA channel is set to 16-bit
- LDMA_TX=1 and LDMA_RX=1

See also *: Communication diagrams on page 1066* for details about common assumptions and notes.

Table 205. Transmit FIFO status flags

Flag	Description
TXFIFOF	Set to high when all 32 transmit FIFO words contain valid data.
TXFIFOE	Set to high when the transmit FIFO does not contain valid data.
TXFIFOHE	Set to high when 8 or more transmit FIFO words are empty. This flag can be used as a DMA request.
TXDAVL	Set to high when the transmit FIFO contains valid data. This flag is the inverse of the TXFIFOE flag.
TXUNDERR	Set to high when an underrun error occurs. This flag is cleared by writing to the SDMMC Clear register. *Note: In case of TXUNDERR, and DMA is used to fill SDMMC FIFO, user software should disable DMA stream, and then write DMAEN bit in SDMMC_DCTRL with '0' (to disable DMA request generation).*

- Receive FIFO

  When the data path subunit receives a word of data, it drives the data on the write databus. The write pointer is incremented after the write operation completes. On the read side, the contents of the FIFO word pointed to by the current value of the read pointer is driven onto the read databus. If the receive FIFO is disabled, all status flags are deasserted, and the read and write pointers are reset. The data path subunit asserts RXACT when it receives data. *Table 206* lists the receive FIFO status flags. The receive FIFO is accessible via 32 sequential addresses.

Table 206. Receive FIFO status flags

Flag	Description
RXFIFOF	Set to high when all 32 receive FIFO words contain valid data
RXFIFOE	Set to high when the receive FIFO does not contain valid data.
RXFIFOHF	Set to high when 8 or more receive FIFO words contain valid data. This flag can be used as a DMA request.
RXDAVL	Set to high when the receive FIFO is not empty. This flag is the inverse of the RXFIFOE flag.
RXOVERR	Set to high when an overrun error occurs. This flag is cleared by writing to the SDMMC Clear register. *Note: In case of RXOVERR, and DMA is used to read SDMMC FIFO, user software should disable DMA stream, and then write DMAEN bit in SDMMC_DCTRL with '0' (to disable DMA request generation).*

35.3.2 SDMMC APB2 interface

The APB2 interface generates the interrupt and DMA requests, and accesses the SDMMC adapter registers and the data FIFO. It consists of a data path, register decoder, and interrupt/DMA logic.

SDMMC interrupts

The interrupt logic generates an interrupt request signal that is asserted when at least one of the selected status flags is high. A mask register is provided to allow selection of the conditions that will generate an interrupt. A status flag generates the interrupt request if a corresponding mask flag is set.

SDMMC/DMA interface

SDMMC APB interface controls all subunit to perform transfers between the host and card

Example of read procedure using DMA

Send CMD17 (READ_BLOCK) as follows:

a) Program the SDMMC data length register (SDMMC data timer register should be already programmed before the card identification process)

b) Program DMA channel (please refer to *DMA configuration for SDMMC controller*)

c) Program the SDMMC data control register: DTEN with ‘1’ (SDMMC card host enabled to send data); DTDIR with ‘1’ (from card to controller); DTMODE with ‘0’ (block data transfer); DMAEN with ‘1’ (DMA enabled); DBLOCKSIZE with 0x9 (512 bytes). Other fields are don’t care.

d) Program the SDMMC argument register with the address location of the card from where data is to be transferred

e) Program the SDMMC command register: CmdIndex with 17(READ_BLOCK); WaitResp with ‘1’ (SDMMC card host waits for a response); CPSMEN with ‘1’ (SDMMC card host enabled to send a command). Other fields are at their reset value.

f) Wait for SDMMC_STA[6] = CMDREND interrupt, (CMDREND is set if there is no error on command path).

g) Wait for SDMMC_STA[10] = DBCKEND, (DBCKEND is set in case of no errors until the CRC check is passed)

h) Wait until the FIFO is empty, when FIFO is empty the SDMMC_STA[5] = RXOVERR value has to be check to guarantee that read succeeded

Note: When FIFO overrun error occurs with last 1-4 bytes, it may happens that RXOVERR flag is set 2 APB clock cycles after DATAEND flag is set. To guarantee success of read operation RXOVERR must be cheked after FIFO is empty.

CAN filter mode register (CAN_FM1R)

Address offset: 0x204
Reset value: 0x0000 0000

This register can be written only when the filter initialization mode is set (FINIT=1) in the CAN_FMR register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	FBM27	FBM26	FBM25	FBM24	FBM23	FBM22	FBM21	FBM20	FBM19	FBM18	FBM17	FBM16
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FBM15	FBM14	FBM13	FBM12	FBM11	FBM10	FBM9	FBM8	FBM7	FBM6	FBM5	FBM4	FBM3	FBM2	FBM1	FBM0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Note: Please refer to Figure 433: Filter bank scale configuration - register organization on page 1263

Bits 31:28 Reserved, must be kept at reset value.

Bits 27:0 **FBM*x***: Filter mode
Mode of the registers of Filter x.
0: Two 32-bit registers of filter bank x are in Identifier Mask mode.
1: Two 32-bit registers of filter bank x are in Identifier List mode.

CAN filter scale register (CAN_FS1R)

Address offset: 0x20C
Reset value: 0x0000 0000

This register can be written only when the filter initialization mode is set (FINIT=1) in the CAN_FMR register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	FSC27	FSC26	FSC25	FSC24	FSC23	FSC22	FSC21	FSC20	FSC19	FSC18	FSC17	FSC16
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FSC15	FSC14	FSC13	FSC12	FSC11	FSC10	FSC9	FSC8	FSC7	FSC6	FSC5	FSC4	FSC3	FSC2	FSC1	FSC0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:28 Reserved, must be kept at reset value.

Bits 27:0 **FSCx**: Filter scale configuration
These bits define the scale configuration of Filters 27-0.
0: Dual 16-bit scale configuration
1: Single 32-bit scale configuration

Note: Please refer to Figure 433: Filter bank scale configuration - register organization on page 1263.

CAN filter FIFO assignment register (CAN_FFA1R)

Address offset: 0x214
Reset value: 0x0000 0000

configured for 96 bit times, the MAC follows the rule of deference specified in Section 4.2.3.2.1 of the IEEE 802.3 specification. The MAC resets its IFG counter if a carrier is detected during the first two-thirds (64-bit times for all IFG values) of the IFG interval. If the carrier is detected during the final one third of the IFG interval, the MAC continues the IFG count and enables the transmitter after the IFG interval. The MAC implements the truncated binary exponential backoff algorithm when it operates in Half-duplex mode.

Transmit flow control

When the Transmit Flow Control Enable bit (TFE bit in ETH_MACFCR) is set, the MAC generates Pause frames and transmits them as necessary, in Full-duplex mode. The Pause frame is appended with the calculated CRC, and is sent. Pause frame generation can be initiated in two ways.

A pause frame is sent either when the application sets the FCB bit in the ETH_MACFCR register or when the receive FIFO is full (packet buffer).

- If the application has requested flow control by setting the FCB bit in ETH_MACFCR, the MAC generates and transmits a single Pause frame. The value of the pause time in the generated frame contains the programmed pause time value in ETH_MACFCR. To extend the pause or end the pause prior to the time specified in the previously transmitted Pause frame, the application must request another Pause frame transmission after programming the Pause Time value (PT in ETH_MACFCR register) with the appropriate value.
- If the application has requested flow control when the receive FIFO is full, the MAC generates and transmits a Pause frame. The value of the pause time in the generated frame is the programmed pause time value in ETH_MACFCR. If the receive FIFO remains full at a configurable number of slot-times (PLT bits in ETH_MACFCR) before this Pause time runs out, a second Pause frame is transmitted. The process is repeated as long as the receive FIFO remains full. If this condition is no more satisfied prior to the sampling time, the MAC transmits a Pause frame with zero pause time to indicate to the remote end that the receive buffer is ready to receive new data frames.

Single-packet transmit operation

The general sequence of events for a transmit operation is as follows:

1. If the system has data to be transferred, the DMA controller fetches them from the memory through the AHB Master interface and starts forwarding them to the FIFO. It continues to receive the data until the end of frame is transferred.
2. When the threshold level is crossed or a full packet of data is received into the FIFO, the frame data are popped and driven to the MAC core. The DMA continues to transfer data from the FIFO until a complete packet has been transferred to the MAC. Upon completion of the frame, the DMA controller is notified by the status coming from the MAC.

Transmit operation—Two packets in the buffer

1. Because the DMA must update the descriptor status before releasing it to the Host, there can be at the most two frames inside a transmit FIFO. The second frame is fetched by the DMA and put into the FIFO only if the OSF (operate on second frame) bit is set. If this bit is not set, the next frame is fetched from the memory only after the MAC has completely processed the frame and the DMA has released the descriptors.
2. If the OSF bit is set, the DMA starts fetching the second frame immediately after completing the transfer of the first frame to the FIFO. It does not wait for the status to be updated. In the meantime, the second frame is received into the FIFO while the first

Bit 5 **FT:** Frame type

When set, this bit indicates that the Receive frame is an Ethernet-type frame (the LT field is greater than or equal to 0x0600). When this bit is reset, it indicates that the received frame is an IEEE802.3 frame. This bit is not valid for Runt frames less than 14 bytes. When the normal descriptor format is used (ETH_DMABMR EDFE=0), FT can take on special meaning as specified in *Table 254*.

Bit 4 **RWT:** Receive watchdog timeout

When set, this bit indicates that the Receive watchdog timer has expired while receiving the current frame and the current frame is truncated after the watchdog timeout.

Bit 3 **RE:** Receive error

When set, this bit indicates that the RX_ERR signal is asserted while RX_DV is asserted during frame reception.

Bit 2 **DE:** Dribble bit error

When set, this bit indicates that the received frame has a non-integer multiple of bytes (odd nibbles). This bit is valid only in MII mode.

Bit 1 **CE:** CRC error

When set, this bit indicates that a cyclic redundancy check (CRC) error occurred on the received frame. This field is valid only when the last descriptor (RDES0[8]) is set.

Bit 0 **PCE/ESA**: Payload checksum error / extended status available

When set, it indicates that the TCP, UDP or ICMP checksum the core calculated does not match the received encapsulated TCP, UDP or ICMP segment's Checksum field. This bit is also set when the received number of payload bytes does not match the value indicated in the Length field of the encapsulated IPv4 or IPv6 datagram in the received Ethernet frame. This bit can take on special meaning as specified in *Table 254*.

If the enhanced descriptor format is enabled (EDFE=1, bit 7 in ETH_DMABMR), this bit takes on the ESA function (otherwise it is PCE). When ESA is set, it indicates that the extended status is available in descriptor word 4 (RDES4). ESA is valid only when the last descriptor bit (RDES0[8]) is set.

Bits 5, 7, and 0 reflect the conditions discussed in *Table 254*.

Bit 4 **BRESTP:** *Rx-Stop on Bit Rising Error*

The BRESTP bit is set and cleared by software.

0: BRE detection does not stop reception of the CEC message. Data bit is sampled at 1.05 ms.

1: BRE detection stops message reception

Bit 3 **RXTOL:** Rx-Tolerance

The RXTOL bit is set and cleared by software.

0: Standard tolerance margin:

- Start-Bit, +/- 200 µs rise, +/- 200 µs fall.
- Data-Bit: +/- 200 µs rise. +/- 350 µs fall.

1: Extended Tolerance

- Start-Bit: +/- 400 µs rise, +/- 400 µs fall
- Data-Bit: +/-300 µs rise, +/- 500 µs fall

Bits 2:0 **SFT:** *Signal Free Time*

SFT bits are set by software. In the SFT=0x0 configuration the number of nominal data bit periods waited before transmission is ruled by hardware according to the transmission history. In all the other configurations the SFT number is determined by software.

- 0x0
  - 2.5 Data-Bit periods if CEC is the last bus initiator with unsuccessful transmission (ARBLST=1, TXERR=1, TXUDR=1 or TXACKE= 1)
  - 4 Data-Bit periods if CEC is the new bus initiator
  - 6 Data-Bit periods if CEC is the last bus initiator with successful transmission (TXEOM=1)
- 0x1: 0.5 nominal data bit periods
- 0x2: 1.5 nominal data bit periods
- 0x3: 2.5 nominal data bit periods
- 0x4: 3.5 nominal data bit periods
- 0x5: 4.5 nominal data bit periods
- 0x6: 5.5 nominal data bit periods
- 0x7: 6.5 nominal data bit periods

Note: Note that the SW-DP state machine is inactive until the target reads this ID code.

- The SW-DP state machine is in RESET STATE either after power-on reset, or after the DP has switched from JTAG to SWD or after the line is high for more than 50 cycles
- The SW-DP state machine is in IDLE STATE if the line is low for at least two cycles after RESET state.
- After RESET state, it is **mandatory** to first enter into an IDLE state AND to perform a READ access of the DP-SW ID CODE register. Otherwise, the target will issue a FAULT acknowledge response on another transactions.

Further details of the SW-DP state machine can be found in the *Cortex®-M7 with FPU TRM* and the *CoreSight Components Technical Reference Manual*.

40.8.4 DP and AP read/write accesses

- Read accesses to the DP are not posted: the target response can be immediate (if ACK=OK) or can be delayed (if ACK=WAIT).
- Read accesses to the AP are posted. This means that the result of the access is returned on the next transfer. If the next access to be done is NOT an AP access, then the DP-RDBUFF register must be read to obtain the result.
  The READOK flag of the DP-CTRL/STAT register is updated on every AP read access or RDBUFF read request to know if the AP read access was successful.
- The SW-DP implements a write buffer (for both DP or AP writes), that enables it to accept a write operation even when other transactions are still outstanding. If the write buffer is full, the target acknowledge response is “WAIT”. With the exception of IDCODE read or CTRL/STAT read or ABORT write which are accepted even if the write buffer is full.
- Because of the asynchronous clock domains SWCLK and HCLK, two extra SWCLK cycles are needed after a write transaction (after the parity bit) to make the write effective internally. These cycles should be applied while driving the line low (IDLE state)
  This is particularly important when writing the CTRL/STAT for a power-up request. If the next transaction (requiring a power-up) occurs immediately, it will fail.

40.8.5 SW-DP registers

Access to these registers are initiated when APnDP=0

Table 269. SW-DP registers

A(3:2)	R/W	CTRLSEL bit of SELECT register	Register	Notes
00	Read	-	IDCODE	The manufacturer code is not set to ST code. 0x5BA02477 (identifies the SW-DP)
00	Write	-	ABORT	-