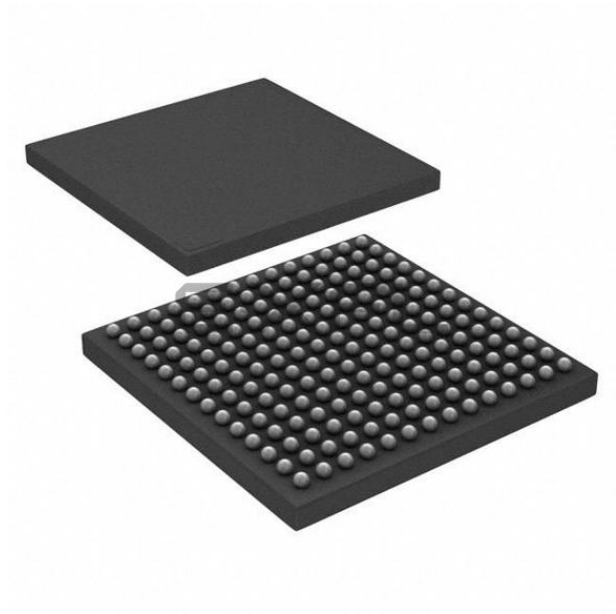


Welcome to E-XFL.COM

Understanding Embedded - Microprocessors

Embedded microprocessors are specialized computing chips designed to perform specific tasks within an embedded system. Unlike general-purpose microprocessors found in personal computers, embedded microprocessors are tailored for dedicated functions within larger systems, offering optimized performance, efficiency, and reliability. These microprocessors are integral to the operation of countless electronic devices, providing the computational power necessary for controlling processes, handling data, and managing communications.

Applications of Embedded - Microprocessors

Embedded microprocessors are utilized across a broad spectrum of applications, making them indispensable in

Details	
Product Status	Active
Core Processor	ARM® Cortex®-A5
Number of Cores/Bus Width	1 Core, 32-Bit
Speed	500MHz
Co-Processors/DSP	Multimedia; NEON™ MPE
RAM Controllers	LPDDR1, LPDDR2, LPDDR3, DDR2, DDR3, DDR3L, QSPI
Graphics Acceleration	Yes
Display & Interface Controllers	Keyboard, LCD, Touchscreen
Ethernet	10/100Mbps (1)
SATA	-
USB	USB 2.0 + HSIC
Voltage - I/O	3.3V
Operating Temperature	-40°C ~ 85°C (TA)
Security Features	ARM TZ, Boot Security, Cryptography, RTIC, Secure Fusebox, Secure JTAG, Secure Memory, Secure RTC
Package / Case	196-TFBGA, CSBGA
Supplier Device Package	196-TFBGA (11x11)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsama5d21a-cur

5. Safety and Security Features

5.1 Design for Safety and IEC60730 Class B Certification

5.1.1 Background Information

The IEC 60730 standard encompasses all aspects of appliance design. Annex H of the standard covers the aspects most relevant to microcontrollers. It details the tests and diagnostics which are intended to ensure safe operation of embedded control hardware and software. IEC 60730 defines three classifications for electronic control functions:

- Class A - Control functions which are not intended to be relied upon for safety of the equipment
- Class B - Control functions intended to prevent unsafe operation of the controlled equipment
- Class C - Control functions intended to prevent special hazards such as explosions

Specific design techniques have been used in the SAMA5D2 to ease compliance with the IEC 60730 Class B Certification and to resolve general-purpose safety concerns. This allows reduced software development and code size as well as savings on external hardware circuitry, since built-in self-tests are already embedded in the MPU. Table 5-1 gives the list of peripherals which incorporate these techniques, and details whether these features are applicable for the IEC 60730 Class B Certification or for general-purpose safety considerations.

5.2 Design for Security

The SAMA5D2 embeds peripherals with security features to prevent counterfeiting, to secure external communication, and to authenticate the system.

Table 5-2 provides the list of peripherals and an overview of their security function. For more information, see the sections on each peripheral.

Table 16-2: SAMA5D2 MRL A and MRL B Parts (Continued)

Device Name
ATSAMA5D26B
ATSAMA5D27B
ATSAMA5D28B

- Definitions (MRL A, MRL B)

SPI x-y-z protocol:

- Command opcode is sent on x I/O data line(s) with x in {1, 2, 4}
- Address is sent on y I/O data line(s) with y in {1, 2, 4}
- Data are sent or received on z I/O data lin(s) with z in {1, 2, 4}

Relevant combinations are:

SPI 1-1-1: legacy SPI protocol using MOSI/IO0 and MISO/IO1 lines

SPI 1-1-2: SPI Dual Output using IO0 and IO1 lines

SPI 1-2-2: SPI Dual I/O using IO0 and IO1 lines

SPI 2-2-2: SPI Dual Command using IO0 and IO1 lines

SPI 1-1-4: SPI Quad Output using IO0, IO1, IO2 and IO3 lines

SPI 1-4-4: SPI Quad I/O using IO0, IO1, IO2 and IO3 lines

SPI 4-4-4: SPI Quad Command using IO0, IO1, IO2 and IO3 lines

- Supported QSPI Memory Manufacturers (MRL A, MRL B)

The ROM code only supports the three following manufacturers (manufacturer ID):

- Cypress (01h)
- Micron (20h)
- Macronix (C2h)

Other manufacturer IDs are **ignored**: The ROM code jumps to the next Non-Volatile Memory in the Boot Sequence.

- SPI Clock Frequency, Phase and Polarity (MRL A, MRL B)

The peripheral clock of each QSPI controller is gated from the Master Clock (MCK). The ROM code configures MCK and the QSPI Serial Clock (QSCK). See Table 16-6.

The QSPI controller is configured to use Clock Mode 0: Both CPHA and CPOL are cleared in QSPI_SCR.

CPOL = 0: The inactive state value of QSCK is logic level zero.

CPHA = 0: Data is captured on the leading edge of QSCK and changed on the following edge of QSCK.

- QSPI Memory Detection (MRL A, MRL B)

The ROM code probes the QSPI memory using JEDEC Read ID commands. However the opcode and the SPI protocol to be used to read the JEDEC ID of the QSPI memory depend on its Manufacturer and its current internal state.

- Cypress

  Cypress memories do not support the SPI 4-4-4 protocol. The command opcode is always sent on the single MOSI/IO1 data line. Hence when writing the 9Fh opcode on MOSI during the first 8 cycles, Cypress memories should always reply on MISO with their JEDEC ID during the following cycles.

- Micron

  Micron memories provide three modes of operation:

  - Extended SPI: standard SPI protocol upgraded with dual (SPI 1-1-2, SPI 1-2-2) and quad (SPI 1-1-4, SPI 1-4-4) operations
  - Dual I/O SPI: **all** commands use the SPI 2-2-2 protocol
  - Quad I/O SPI: **all** commands use the SPI 4-4-4 protocol

  The ROM code supports the Extended and Quad I/O SPI modes but not Dual I/O SPI.

  In Extended SPI mode, Micron memories replies to the regular Read JEDEC ID opcode using the protocol SPI 1-1-1: the 9Fh opcode is sent on MOSI using eight clock cycles then the JEDEC ID is read from MISO only.

33.22.24 PMC Peripheral Clock Enable Register 1

Name: PMC_PCER1

Address: 0xF0014100

Access: Write-only

31	30	29	28	27	26	25	24
PID63	PID62	PID61	PID60	PID59	PID58	PID57	PID56
23	22	21	20	19	18	17	16
PID55	PID54	PID53	PID52	PID51	PID50	PID49	PID48
15	14	13	12	11	10	9	8
PID47	PID46	PID45	PID44	PID43	PID42	PID41	PID40
7	6	5	4	3	2	1	0
PID39	PID38	PID37	PID36	PID35	PID34	PID33	PID32

This register can only be written if the WPEN bit is cleared in the PMC Write Protection Mode Register.

PIDx: Peripheral Clock x Enable

0: No effect.

1: Enables the corresponding peripheral clock.

Note 1: PID32 to PID63 refer to identifiers as defined in Section 11.2 “Peripheral Identifiers”.

2: Programming the control bits of the Peripheral ID that are not implemented has no effect on the behavior of the PMC.

34.7.21 Secure PIO Output Data Status Register

Name: S_PIO_ODSRx [x=0..3]

Address: 0xFC039018 [0], 0xFC039058 [1], 0xFC039098 [2], 0xFC0390D8 [3]

Access: Read/Write

31	30	29	28	27	26	25	24
P31	P30	P29	P28	P27	P26	P25	P24
23	22	21	20	19	18	17	16
P23	P22	P21	P20	P19	P18	P17	P16
15	14	13	12	11	10	9	8
P15	P14	P13	P12	P11	P10	P9	P8
7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0

Writing this register will only affect I/O lines enabled in the S_PIO_MSKRx.

P0–P31: Output Data Status

0: The data to be driven on the I/O line of the I/O group x is 0.

1: The data to be driven on the I/O line of the I/O group x is 1.

Table 37-20: Register Mapping (Continued)

Offset	Register	Name	Access	Reset
0x588	PMECC Error Location SIGMA 24 Register	HSMC_SIGMA24	Read/Write	0x0
0x58C	PMECC Error Location SIGMA 25 Register	HSMC_SIGMA25	Read/Write	0x0
0x590	PMECC Error Location SIGMA 26 Register	HSMC_SIGMA26	Read/Write	0x0
0x594	PMECC Error Location SIGMA 27 Register	HSMC_SIGMA27	Read/Write	0x0
0x598	PMECC Error Location SIGMA 28 Register	HSMC_SIGMA28	Read/Write	0x0
0x59C	PMECC Error Location SIGMA 29 Register	HSMC_SIGMA29	Read/Write	0x0
0x5A0	PMECC Error Location SIGMA 30 Register	HSMC_SIGMA30	Read/Write	0x0
0x5A4	PMECC Error Location SIGMA 31 Register	HSMC_SIGMA31	Read/Write	0x0
0x5A8	PMECC Error Location SIGMA 32 Register	HSMC_SIGMA32	Read/Write	0x0
0x5AC	PMECC Error Location 0 Register	HSMC_ERRLOC0	Read-only	0x0
...	...	...	...	...
0x628	PMECC Error Location 31 Register	HSMC_ERRLOC31	Read-only	0x0
0x62C–0x6FC	Reserved	–	–	–
0x14*CS_number+0x700	Setup Register	HSMC_SETUP	Read/Write	0x0101_0101
0x14*CS_number+0x704	Pulse Register	HSMC_PULSE	Read/Write	0x0101_0101
0x14*CS_number+0x708	Cycle Register	HSMC_CYCLE	Read/Write	0x0003_0003
0x14*CS_number+0x70C	Timings Register	HSMC_TIMINGS	Read/Write	0x0000_0000
0x14*CS_number+0x710	Mode Register	HSMC_MODE	Read/Write	0x0000_1003
0x7A0	Off Chip Memory Scrambling Register	HSMC_OCMS	Read/Write	0x0
0x7A4	Off Chip Memory Scrambling KEY1 Register	HSMC_KEY1	Write-once	0x0
0x7A8	Off Chip Memory Scrambling KEY2 Register	HSMC_KEY2	Write-once	0x0
0x7AC–0x7E0	Reserved	–	–	–
0x7E4	Write Protection Mode Register	HSMC_WPMR	Read/Write	0x0
0x7E8	Write Protection Status Register	HSMC_WPSR	Read-only	0x0
0x7EC–0x7FC	Reserved	–	–	–

37.20.24 PMECC Error Location Disable Register

Name: HSMC_ELDIS

Address: 0xF801450C

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	DIS

DIS: Disable Error Location Engine

0: No effect

1: Disable the Error location engine.

38.9.20 XDMAC Channel x [x = 0..15] Interrupt Mask Register

Name: XDMAC_CIMx [x = 0..15]

Address: 0xF0004058 (1)[0], 0xF0004098 (1)[1], 0xF00040D8 (1)[2], 0xF0004118 (1)[3], 0xF0004158 (1)[4], 0xF0004198 (1)[5], 0xF00041D8 (1)[6], 0xF0004218 (1)[7], 0xF0004258 (1)[8], 0xF0004298 (1)[9], 0xF00042D8 (1)[10], 0xF0004318 (1)[11], 0xF0004358 (1)[12], 0xF0004398 (1)[13], 0xF00043D8 (1)[14], 0xF0004418 (1)[15], 0xF0010058 (0)[0], 0xF0010098 (0)[1], 0xF00100D8 (0)[2], 0xF0010118 (0)[3], 0xF0010158 (0)[4], 0xF0010198 (0)[5], 0xF00101D8 (0)[6], 0xF0010218 (0)[7], 0xF0010258 (0)[8], 0xF0010298 (0)[9], 0xF00102D8 (0)[10], 0xF0010318 (0)[11], 0xF0010358 (0)[12], 0xF0010398 (0)[13], 0xF00103D8 (0)[14], 0xF0010418 (0)[15]

Access: Read-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–

23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–

15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–

7	6	5	4	3	2	1	0
–	ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM

BIM: End of Block Interrupt Mask Bit

0: Block interrupt is masked.

1: Block interrupt is activated.

LIM: End of Linked List Interrupt Mask Bit

0: End of linked list interrupt is masked.

1: End of linked list interrupt is activated.

DIM: End of Disable Interrupt Mask Bit

0: End of disable interrupt is masked.

1: End of disable interrupt is activated.

FIM: End of Flush Interrupt Mask Bit

0: End of flush interrupt is masked.

1: End of flush interrupt is activated.

RBEIM: Read Bus Error Interrupt Mask Bit

0: Bus error interrupt is masked.

1: Bus error interrupt is activated.

WBEIM: Write Bus Error Interrupt Mask Bit

0: Bus error interrupt is masked.

1: Bus error interrupt is activated.

ROIM: Request Overflow Error Interrupt Mask Bit

0: Request overflow interrupt is masked.

1: Request overflow interrupt is activated.

39.7.11 LCD Controller Interrupt Enable Register

Name: LCDC_LCDIER

Address: 0xF000002C

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–

23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–

15	14	13	12	11	10	9	8
–	–	PPIE	–	HEOIE	OVR2IE	OVR1IE	BASEIE

7	6	5	4	3	2	1	0
–	–	–	FIFOERRIE	–	DISPIE	DISIE	SOFIE

SOFIE: Start of Frame Interrupt Enable

0: No effect.

1: Enables the interrupt.

DISIE: LCD Disable Interrupt Enable

0: No effect.

1: Enables the interrupt.

DISPIE: Powerup/Powerdown Sequence Terminated Interrupt Enable

0: No effect.

1: Enables the interrupt.

FIFOERRIE: Output FIFO Error Interrupt Enable

0: No effect.

1: Enables the interrupt.

BASEIE: Base Layer Interrupt Enable

0: No effect.

1: Enables the interrupt.

OVR1IE: Overlay 1 Interrupt Enable

0: No effect.

1: Enables the interrupt.

OVR2IE: Overlay 2 Interrupt Enable

0: No effect.

1: Enables the interrupt.

HEOIE: High-End Overlay Interrupt Enable

0: No effect.

1: Enables the interrupt.

39.7.46 Overlay 1 Configuration Register 1

Name: LCDC_OVR1CFG1

Address: 0xF0000170

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–

23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–

15	14	13	12	11	10	9	8
–	–	–	–	–	–	CLUTMODE	

7	6	5	4	3	2	1	0
RGBMODE				–	–	–	CLUTEN

CLUTEN: Color Lookup Table Mode Enable

0: RGB mode is selected.

1: Color Lookup Table mode is selected.

RGBMODE: RGB Mode Input Selection

Value	Name	Description
0	12BPP_RGB_444	12 bpp RGB 444
1	16BPP_ARGB_4444	16 bpp ARGB 4444
2	16BPP_RGBA_4444	16 bpp RGBA 4444
3	16BPP_RGB_565	16 bpp RGB 565
4	16BPP_TRGB_1555	16 bpp TRGB 1555
5	18BPP_RGB_666	18 bpp RGB 666
6	18BPP_RGB_666PACKED	18 bpp RGB 666 PACKED
7	19BPP_TRGB_1666	19 bpp TRGB 1666
8	19BPP_TRGB_PACKED	19 bpp TRGB 1666 PACKED
9	24BPP_RGB_888	24 bpp RGB 888
10	24BPP_RGB_888_PACKED	24 bpp RGB 888 PACKED
11	25BPP_TRGB_1888	25 bpp TRGB 1888
12	32BPP_ARGB_8888	32 bpp ARGB 8888
13	32BPP_RGBA_8888	32 bpp RGBA 8888

REP: Use Replication logic to expand RGB color to 24 bits

0: When the selected pixel depth is less than 24 bpp the pixel is shifted and least significant bits are set to 0.

1: When the selected pixel depth is less than 24 bpp the pixel is shifted and the least significant bit replicates the msb.

DSTKEY: Destination Chroma Keying

0: Source Chroma keying is enabled.

1: Destination Chroma keying is used.

VIDPRI: Video Priority

0: OVR1 layer is above HEO layer.

1: OVR1 layer is below HEO layer.

GA: Blender Global Alpha

Global alpha blender for the current layer.

44.8.6 I2SC Interrupt Enable Register

Name: I2SC_IER

Address: 0xF8050014 (0), 0xFC04C014 (1)

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	TXUR	TXRDY	–	–	RXOR	RXRDY	–

RXRDY: Receiver Ready Interrupt Enable

0: Writing a ’0’ to this bit has no effect.

1: Writing a ’1’ to this bit sets the corresponding bit in I2SC_IMR.

RXOR: Receiver Overrun Interrupt Enable

0: Writing a ’0’ to this bit has no effect.

1: Writing a ’1’ to this bit sets the corresponding bit in I2SC_IMR.

TXRDY: Transmit Ready Interrupt Enable

0: Writing a ’0’ to this bit as no effect.

1: Writing a ’1’ to this bit sets the corresponding bit in I2SC_IMR.

TXUR: Transmit Underflow Interrupt Enable

0: Writing a ’0’ to this bit has no effect.

1: Writing a ’1’ to this bit sets the corresponding bit in I2SC_IMR.

SDA: SDA Line Value

0: SDA line sampled value is ‘0’.

1: SDA line sampled value is ‘1’.

46.7.24 TWIHS FIFO Interrupt Disable Register

Name: TWIHS_FIDR

Address: 0xF8028068 (0), 0xFC028068 (1)

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

TXFEF: TXFEF Interrupt Disable

TXFFF: TXFFF Interrupt Disable

TXFTHF: TXFTHF Interrupt Disable

RXFEF: RXFEF Interrupt Disable

RXFFF: RXFFF Interrupt Disable

RXFTHF: RXFTHF Interrupt Disable

TXFPTEF: TXFPTEF Interrupt Disable

RXFPTEF: RXFPTEF Interrupt Disable

The DMAC transfer type must be configured in bytes or halfwords when FIFOs operate in Single Data mode (the same applies when FIFOs are disabled).

47.7.11.7 USART Multiple Data Mode

Multiple Data mode minimizes the number of accesses by concatenating the data to send/read in one access.

When FLEX_US_FMR.TXRDYM > 0, the Transmit FIFO operates in Multiple Data mode.

When FLEX_US_FMR.RXRDYM > 0, the Receive FIFO operates in Multiple Data mode.

However, Multiple Data mode cannot be used for the following configurations:

- If FLEX_US_MR.MODE9 is set
- If FLEX_US_MR.USART_MODE is set to either LIN_MASTER or LIN_SLAVE

In Multiple Data mode, it is possible to write/read up to four data in one FLEX_US_THR/FLEX_US_RHR access.

The number of data to write/read is defined by the size of the register access. If the access is a byte-size register access, only one data is written/read, if the access is a halfword size register access, then two data are written/read and, finally, if the access is a word-size register access, four data are written/read.

Written/read data are always right-aligned, as described in Section 47.10.21 ”USART Receive Holding Register (FIFO Multi Data)” and Section 47.10.23 ”USART Transmit Holding Register (FIFO Multi Data)”.

As an example, if the Transmit FIFO is empty and there are six data to send, either of the following write accesses may be performed:

- six FLEX_US_THR-byte write accesses
- three FLEX_US_THR-halfword write accesses
- one FLEX_US_THR word write access and one FLEX_US_THR halfword write access

With a Receive FIFO containing six data, any of the following read accesses may be performed:

- six FLEX_US_RHR-byte read accesses
- three FLEX_US_RHR-halfword read accesses
- one FLEX_US_RHR-word read access and one FLEX_US_RHR-halfword read access
- TXRDY and RXRDY Configuration

In Multiple Data mode, it is possible to write one or more data in the same FLEX_US_THR/FLEX_US_RHR access. The TXRDY flag indicates if one or more data can be written in the FIFO depending on the configuration of FLEX_US_FMR.TXRDYM/RXRDYM.

As an example, if four data are written each time in FLEX_US_THR, it is useful to configure the TXRDYM field to the value ‘2’ so that the TXRDY flag is at ‘1’ only when at least four data can be written in the Transmit FIFO.

In the same way, if four data are read each time in FLEX_US_RHR, it is useful to configure the RXRDYM field to the value ‘2’ so that the RXRDY flag is at ‘1’ only when at least four unread data are in the Receive FIFO.

- DMAC

When FIFOs operate in Multiple Data mode, the DMAC transfer type must be configured in byte, halfword or word depending on the FLEX_US_FMR.TXRDYM/RXRDYM settings.

47.7.11.8 Transmit FIFO Lock

- LIN Mode:

If a frame is aborted using the Abort LIN Transmission bit (FLEX_US_CR.LINABT), a lock is set on the Transmit FIFO, preventing any new frame from being sent until it is cleared. This allows clearing the FIFO if needed, resetting DMAC channels, etc., without any risk.

The TXFLOCK bit in the USART FIFO Event Status Register (FLEX_US_FESR) is used to check the state of the Transmit FIFO lock.

The Transmit FIFO lock can be cleared by setting FLEX_US_CR.TXFLCLR to ‘1’.

47.7.11.9 FIFO Pointer Error

A FIFO overflow is reported in FLEX_US_FESR.

If the Transmit FIFO is full and a write access is performed on FLEX_US_THR, it generates a Transmit FIFO pointer error and sets FLEX_US_FESR.TXFPTEF.

In Multiple Data mode, if the number of data written in FLEX_US_THR (according to the register access size) is greater than the free space in the Transmit FIFO, a Transmit FIFO pointer error is generated and FLEX_US_FESR.TXFPTEF is set.

A FIFO underflow is reported in FLEX_US_FESR.

In Multiple Data mode, if the number of data read in FLEX_US_RHR (according to the register access size) is greater than the number of unread data in the Receive FIFO, a Receive FIFO pointer error is generated and FLEX_US_FESR.RXFPTEF is set.

47.8.7.6 SPI Single Data Mode

In Single Data mode, only one data is written every time FLEX_SPI_TDR is accessed, and only one data is read every time FLEX_SPI_RDR is accessed.

When FLEX_SPI_FMR.TXRDYM = 0, the Transmit FIFO operates in Single Data mode.

When FLEX_SPI_FMR.RXRDYM = 0, the Receive FIFO operates in Single Data mode.

If Master mode is used (FLEX_SPI_MR.MSTR=1), the Receive FIFO must operate in Single Data mode.

If Variable Peripheral Select mode is used (FLEX_SPI_MR.PS=1), the Transmit FIFO must operate in Single Data mode.

See Section 47.10.48 ”SPI Transmit Data Register” and Section 47.10.45 ”SPI Receive Data Register”.

- DMAC

When FIFOs operate in Single Data mode, the DMAC transfer type must be configured either in bytes, halfwords or words depending on FLEX_SPI_MR.PS bit value and FLEX_SPI_CSRx.BITS field value.

The same applies when FIFOs are disabled.

47.8.7.7 SPI Multiple Data Mode

Multiple Data mode minimizes the number of accesses by concatenating the data to send/read in one access.

When FLEX_SPI_FMR.TXRDYM > 0, the Transmit FIFO operates in Multiple Data mode.

When FLEX_SPI_FMR.RXRDYM > 0, the Receive FIFO operates in Multiple Data mode.

Multiple data can be read from the Receive FIFO only in Slave mode (FLEX_SPI_MR.MSTR=0).

The Transmit FIFO can be loaded with multiple data in the same access by configuring TXRDYM>0 and when FLEX_SPI_MR.PS=0.

In Multiple Data mode, up to two data can be written in one FLEX_SPI_TDR write access. It is also possible to read up to four data in one FLEX_SPI_RDR access if FLEX_SPI_CSRx.BITS is configured to ‘0’ (8-bit data size) and up to two data if FLEX_SPI_CSRx.BITS is configured to a value other than ‘0’ (more than 8-bit data size).

The number of data to write/read is defined by the size of the register access. If the access is a byte-size register access, only one data is written/read. If the access is a halfword size register access, then up to two data are read and only one data is written. Lastly, if the access is a word-size register access, then up to four data are read and up to two data are written.

Written/read data are always right-aligned, as described in Section 47.10.46 ”SPI Receive Data Register (FIFO Multiple Data, 8-bit)”, Section 47.10.47 ”SPI Receive Data Register (FIFO Multiple Data, 16-bit)” and Section 47.10.49 ”SPI Transmit Data Register (FIFO Multiple Data, 8- to 16-bit)”.

As an example, if the Transmit FIFO is empty and there are six data to send, either of the following write accesses may be performed:

- six FLEX_SPI_TDR-byte write accesses
- three FLEX_SPI_TDR-halfword write accesses

With a Receive FIFO containing six data, any of the following read accesses may be performed:

- six FLEX_SPI_RDR-byte read accesses
- three FLEX_SPI_RDR-halfword read accesses
- one FLEX_SPI_RDR-word read access and one FLEX_SPI_RDR-halfword read access
- TDRE and RDRF Configuration

In Multiple Data mode, it is possible to write one or more data in the same FLEX_SPI_TDR/RDR access. The TDRE flag indicates if one or more data can be written in the FIFO depending on the configuration of FLEX_SPI_FMR.TXRDYM/RXRDYM.

As an example, if two data are written each time in FLEX_SPI_TDR, it is useful to configure the TXRDYM field to the value ‘1’ so that the TDRE flag is at ‘1’ only when at least two data can be written in the Transmit FIFO.

Similarly, if four data are read each time in FLEX_SPI_RDR, it is useful to configure the RXRDYM field to the value ‘2’ so that the RDRF flag is at ‘1’ only when at least four unread data are in the Receive FIFO.

- DMAC

It is mandatory to configure DMAC channel size (byte, halfword or word) according to FLEX_SPI_FMR.TXRDYM/RXRDYM configuration. See Section 47.8.7.7 ”SPI Multiple Data Mode” for constraints.

47.8.7.8 FIFO Pointer Error

A FIFO overflow is reported in FLEX_SPI_SR.

If the Transmit FIFO is full and a write access is performed on FLEX_SPI_TDR, it generates a Transmit FIFO pointer error and sets FLEX_SPI_SR.TXFPTEF.

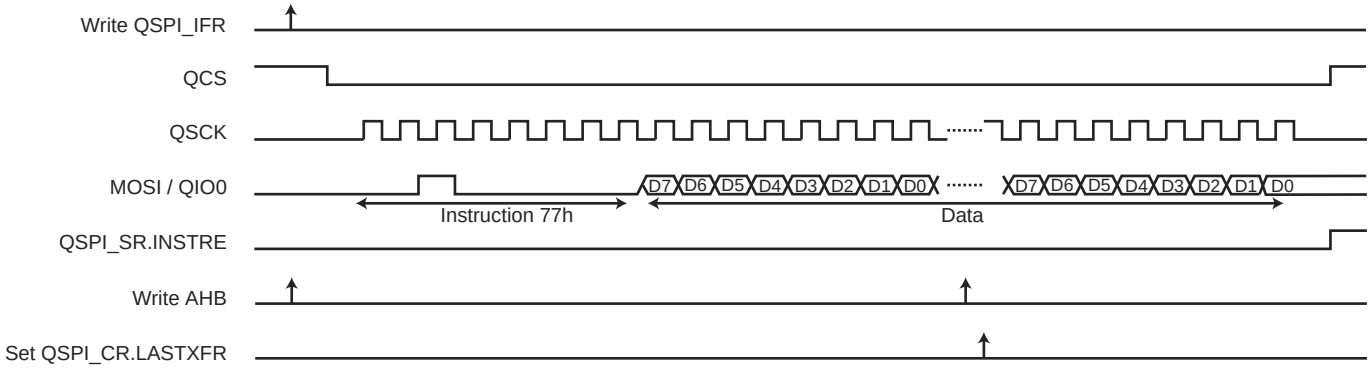
Example 4:

Instruction in Single-bit SPI, without address, without option, with data write in Single-bit SPI.

Command: SET BURST (77h)

- Write 0x0000_0077 in QSPI_ICR.
- Write 0x0000_2090 in QSPI_IFR.
- Read QSPI_IFR (dummy read) to synchronize system bus accesses.
- Write data in the system bus memory space (0x9000_00000-0x9800_00000/0XD000_0000--0XD800_0000). The address of system bus write accesses is not used.
- Write a ‘1’ to QSPI_CR.LASTXFR.
- Wait for QSPI_SR.INSTRE to rise.

Figure 50-14: Instruction Transmission Waveform 4



CMDIDX: Command Index Error

This bit is set to 1 if a Command Index error occurs in the command response.

This bit can only be set to 1 if SDMMC_EISTER.CMDIDX is set to 1. An interrupt can only be generated if SDMMC_EISIER.CMDIDX is set to 1.

Writing this bit to 1 clears this bit.

0: No error.

1: Error.

DATTEO: Data Timeout Error

This bit is set to 1 when detecting one of following timeout conditions.

- Busy timeout for R1b, R5b response type (see “Physical Layer Simplified Specification V3.01” and “SDIO Simplified Specification V3.00”).
- Busy timeout after Write CRC status.
- Write CRC Status timeout.
- Read data timeout

This bit can only be set to 1 if SDMMC_EISTER.DATTEO is set to 1. An interrupt can only be generated if SDMMC_EISIER.DATTEO is set to 1.

Writing this bit to 1 clears this bit.

0: No error.

1: Error.

DATCRC: Data CRC error

This bit is set to 1 when detecting a CRC error when transferring read data which uses the DAT line or when detecting that the Write CRC Status has a value other than “010”.

This bit can only be set to 1 if SDMMC_EISTER.DATCRC is set to 1. An interrupt can only be generated if SDMMC_EISIER.DATCRC is set to 1.

Writing this bit to 1 clears this bit.

0: No error.

1: Error.

DATEND: Data End Bit Error

This bit is set to 1 either when detecting 0 at the end bit position of read data which uses the DAT line or at the end bit position of the CRC Status.

This bit can only be set to 1 if SDMMC_EISTER.DATEND is set to 1. An interrupt can only be generated if SDMMC_EISIER.DATEND is set to 1.

Writing this bit to 1 clears this bit.

0: No error.

1: Error.

CURLIM: Current Limit Error

By setting SD Bus Power (SDBPWR) in SDMMC_PCR, the SDMMC is requested to supply power for the SD Bus. The SDMMC is protected from an illegal card by stopping power supply to the card, in which case this bit indicates a failure status. Reading 1 means the SDMMC is not supplying power to the card due to some failure. Reading 0 means that the SDMMC is supplying power and no error has occurred. The SDMMC may require some sampling time to detect the current limit.

This bit can only be set to 1 if SDMMC_EISTER.CURLIM is set to 1. An interrupt can only be generated if SDMMC_EISIER.CURLIM is set to 1.

Writing this bit to 1 clears this bit.

0: No error.

1: Error.

53.5.7.6 Extended Message ID Filter Element

Up to 64 filter elements can be configured for 29-bit extended IDs. When accessing an Extended Message ID Filter element, its address is the Filter List Extended Start Address MCAN_XIDFC.FLESA plus two times the index of the filter element (0…63).

Table 53-13: Extended Message ID Filter Element

	31	24	23	16	15	8	7	0
F0	EFEC [2:0]		EFID1[28:0]					
F1	EFT[1:0]	–	EFID2[28:0]					

- **F0 Bit 31:29 EFEC[2:0]: Extended Filter Element Configuration**

All enabled filter elements are used for acceptance filtering of extended frames. Acceptance filtering stops at the first matching enabled filter element or when the end of the filter list is reached. If EFEC = “100”, “101”, or “110”, a match sets the interrupt flag MCAN_IR.HPM and, if enabled, an interrupt is generated. In this case, register MCAN_HPMS is updated with the status of the priority match.

Value	Description
0	Disable filter element
1	Store in Rx FIFO 0 if filter matches
2	Store in Rx FIFO 1 if filter matches
3	Reject ID if filter matches
4	Set priority if filter matches
5	Set priority and store in FIFO 0 if filter matches
6	Set priority and store in FIFO 1 if filter matches
7	Store into Rx Buffer or as debug message, configuration of EFT[1:0] ignored

- **F0 Bits 28:0 EFID1[28:0]: Extended Filter ID 1**

First ID of extended ID filter element.

When filtering for Rx Buffers or for debug messages this field defines the ID of an extended message to be stored. The received identifiers must match exactly, only MCAN_XIDAM masking mechanism (see Extended Message ID Filtering) is used.

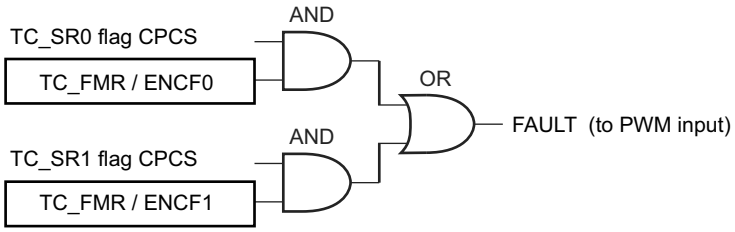
- **F1 Bits 31:30 EFT[1:0]: Extended Filter Type**

Value	Description
0	Range filter from EF1ID to EF2ID (EF2ID ≥ EF1ID)
1	Dual ID filter for EF1ID or EF2ID
2	Classic filter: EF1ID = filter, EF2ID = mask
3	Range filter from EF1ID to EF2ID (EF2ID ≥ EF1ID), MCAN_XIDAM mask not applied

It is possible to trigger the FAULT output of the TIMER1 with TC_SR0.CPCS and/or TC_SR1.CPCS. Each source can be independently enabled/disabled in the TC_FMR.

This can be useful to detect an overflow on speed and/or position when QDEC is processed and to act immediately by using the FAULT output.

Figure 54-24: Fault Output Generation



58.6.3 ICM Status Register

Name: ICM_SR

Address: 0xF8040008

Access: Read-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
RMDIS				RAWRMDIS			
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	ENABLE

ENABLE: ICM Enable Register

0: ICM is disabled.

1: ICM is activated.

RAWRMDIS: Region Monitoring Disabled Raw Status

0: Region *i* monitoring has been activated by writing a 1 in RMEN[*i*] of ICM_CTRL.

1: Region *i* monitoring has been deactivated by writing a 1 in RMDIS[*i*] of ICM_CTRL.

RMDIS: Region Monitoring Disabled Status

0: Region *i* is being monitored (occurs after integrity check value has been calculated and written to Hash area).

1: Region *i* monitoring is not being monitored.