

XMC7000 Serial memory interface (SMIF)

Customer training workshop (CTW)

January 2026

public

Table of contents

1 Target products 3

2 Introduction to XMC7200 - 8M 6

3 Introduction to XMC7100 - 4M 9

4 Serial memory interface (SMIF) 11

5 Block components 16

6 Revision history 51

Copyright © Infineon Technologies AG 2026. All rights reserved.

Table of contents

1 **Target products** **3**

2 Introduction to XMC7200 - 8M 6

3 Introduction to XMC7100 - 4M 9

4 Serial memory interface (SMIF) 11

5 Block components 16

6 Revision history 51

Copyright © Infineon Technologies AG 2026. All rights reserved.

Target products

Target product list for this training material:

Family category	Series	Code flash memory size
XMC7200	XMC7200-8M	Up to 8384 KB
XMC7100	XMC7100-4M	Up to 4160 KB

Copyright © Infineon Technologies AG 2026. All rights reserved.

XMC7200 - 8M family



public

Table of contents

1 Target products 3

2 **Introduction to XMC7200 - 8M** **6**

3 Introduction to XMC7100 - 4M 9

4 Serial memory interface (SMIF) 11

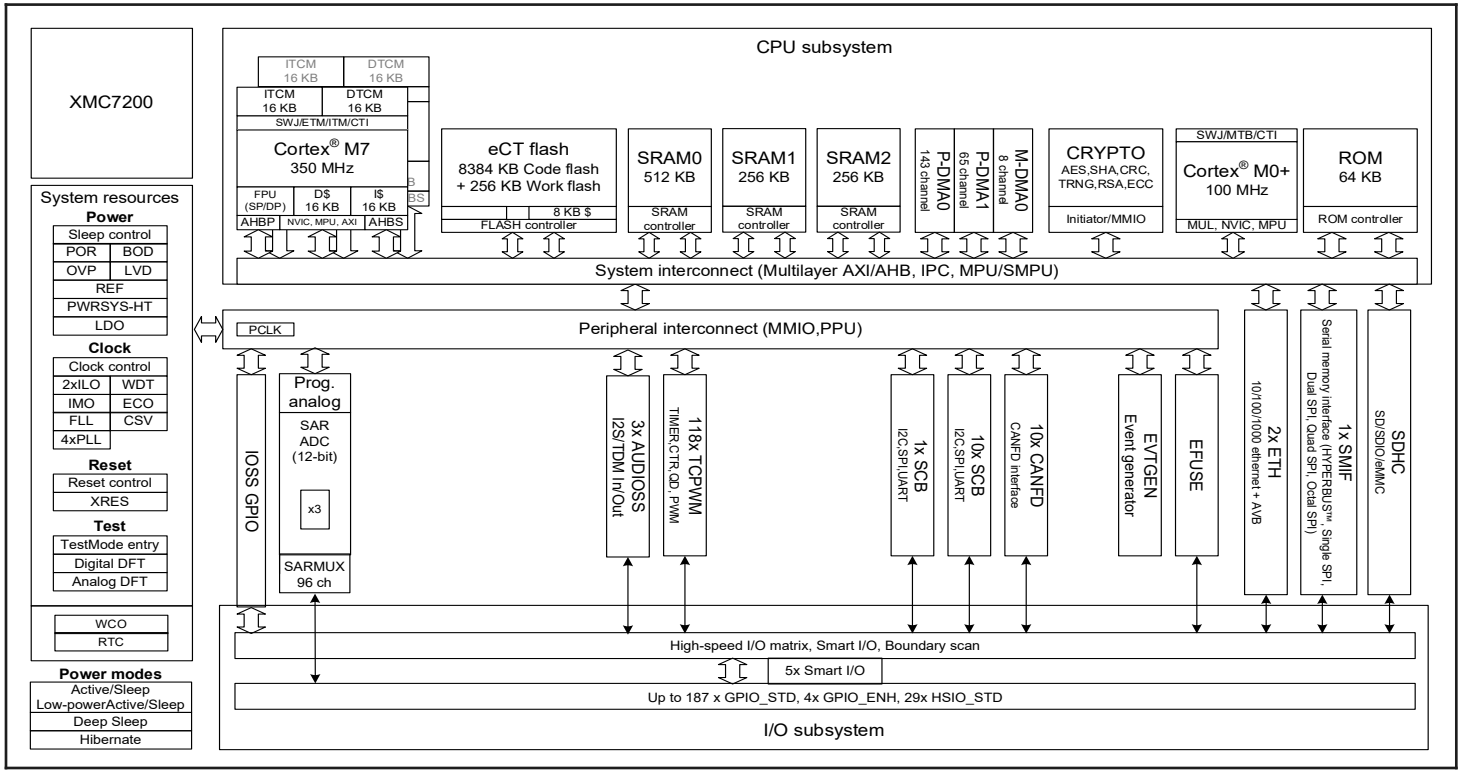
5 Block components 16

6 Revision history 51

Copyright © Infineon Technologies AG 2026. All rights reserved.

Introduction to XMC7200 - 8M

SMIF is located in the peripheral blocks.



Hint bar

See the architecture reference manual serial memory interface (chapter 30), for additional details.

Copyright © Infineon Technologies AG 2026. All rights reserved.

XMC7100 - 4M family



public

Table of contents

1 Target products 3

2 Introduction to XMC7200 - 8M 6

3 **Introduction to XMC7100 - 4M** **9**

4 Serial memory interface (SMIF) 11

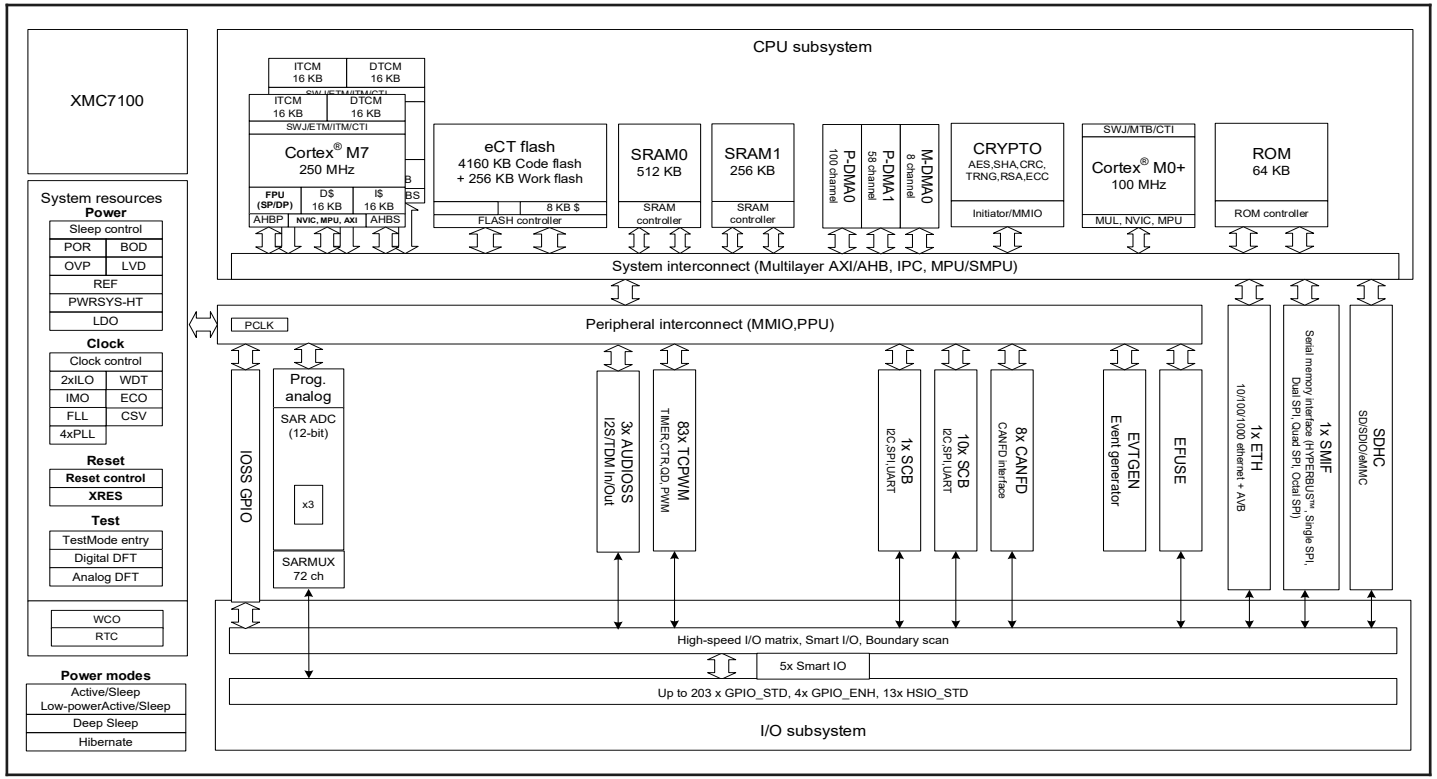
5 Block components 16

6 Revision history 51

Copyright © Infineon Technologies AG 2026. All rights reserved.

Introduction to XMC7100 - 4M

SMIF is located in the peripheral blocks.



Hint bar

See the architecture reference manual, serial memory interface (chapter 30), for additional details.

Copyright © Infineon Technologies AG 2026. All rights reserved.

Table of contents

1 Target products 3

2 Introduction to XMC7200 - 8M 6

3 Introduction to XMC7100 - 4M 9

4 **Serial memory interface (SMIF)** **11**

5 Block components 16

6 Revision history 51

Copyright © Infineon Technologies AG 2026. All rights reserved.

Serial memory interface (SMIF)



public

Overview

SMIF provides an interface to memories with SPI and HYPERBUS™ IFs.

Features

- SPI or HYPERBUS™ master functionality only
- SPI protocol
  - Mode 0 only, with configurable Master-In Slave-Out (MISO) sampling timing
  - Single/dual/quad/octal SPI transfer mode
  - Dual-quad SPI mode (two devices sharing one address range)
  - Single data rate (SDR) and double data rate (DDR) transfer
- Two operation modes
  - Memory-mapped I/O (MMIO) operation mode
  - Execute-in-place (XIP) operation mode
    - Read and write access
    - Automatic transition access from dedicated internal memory address to SPI protocol
    - On-the-fly encryption and decryption

Hint bar

- See the architecture reference manual, Features (section 30.1), for additional details
- SPI mode 0: Data is driven on a falling edge of SPIHB_CLK
- Data is captured on the rising edge of SPIHB_CLK

Copyright © Infineon Technologies AG 2026. All rights reserved.

Overview (contd.)

Features

- Memory device
  - Device capacity in the range of 64 KB to 4 GB (memory size: 2^N)
    - (XMC7000: 128 MB area in XIP mode)
  - Support for configurable external device capacities
  - Up to two external memory devices sharing one address range (for example, dual-quad mode)
  - Up to four chip selects available
    - (XMC7000: 2 pcs)
- Memory interface logic
  - Support stalling of transfers to address the back pressure on FIFOs
  - Supports an asynchronous SPI/HYPERBUS™ transmit and receive interface clock
  - Support read-write-data-strobe (RWDS)
  - Support data signal connections between flexible external SPI memory devices
  - Independent SPI interface transmitter clock from PLL/FLL

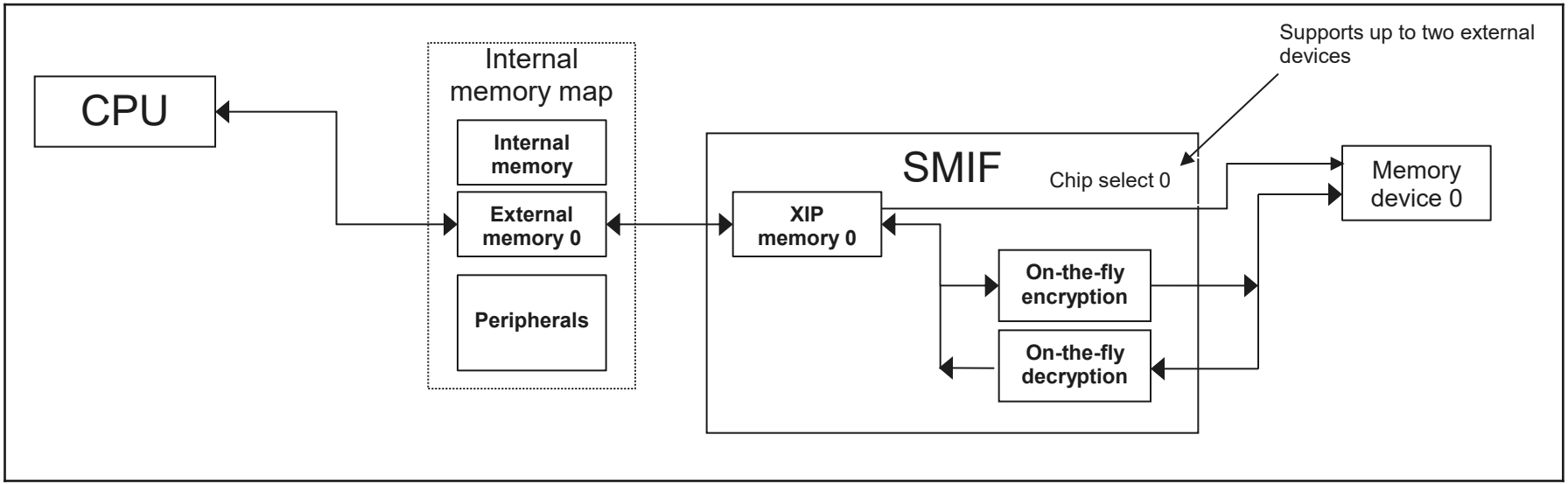
Hint bar

See the architecture reference manual, Features (section 30.1), for additional details.

Copyright © Infineon Technologies AG 2026. All rights reserved.

Example of using SMIF

- External memory for code and data space
- Directly mapped to internal memory map (XIP operation mode)
- Encrypted/decrypted code or data in an external space



Copyright © Infineon Technologies AG 2026. All rights reserved.

Table of contents

1 Target products 3

2 Introduction to XMC7200 - 8M 6

3 Introduction to XMC7100 - 4M 9

4 Serial memory interface (SMIF) 11

5 **Block components** **16**

6 Revision history 51

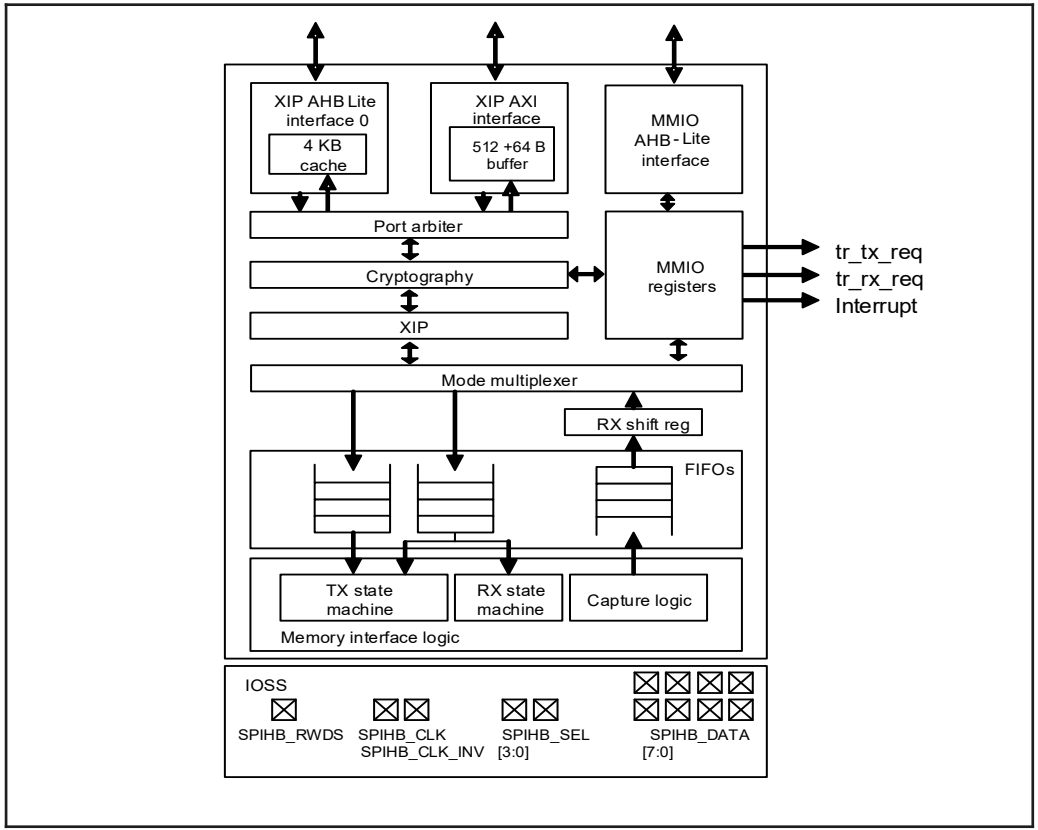
Copyright © Infineon Technologies AG 2026. All rights reserved.

Block components

public

Block components (1/33)

- **Bus interface**
  - MMIO AHB-Lite interface
  - XIP AXI interface
  - XIP AHB-Lite interface



Copyright © Infineon Technologies AG 2026. All rights reserved.

Block components (2/33)

Bus interface

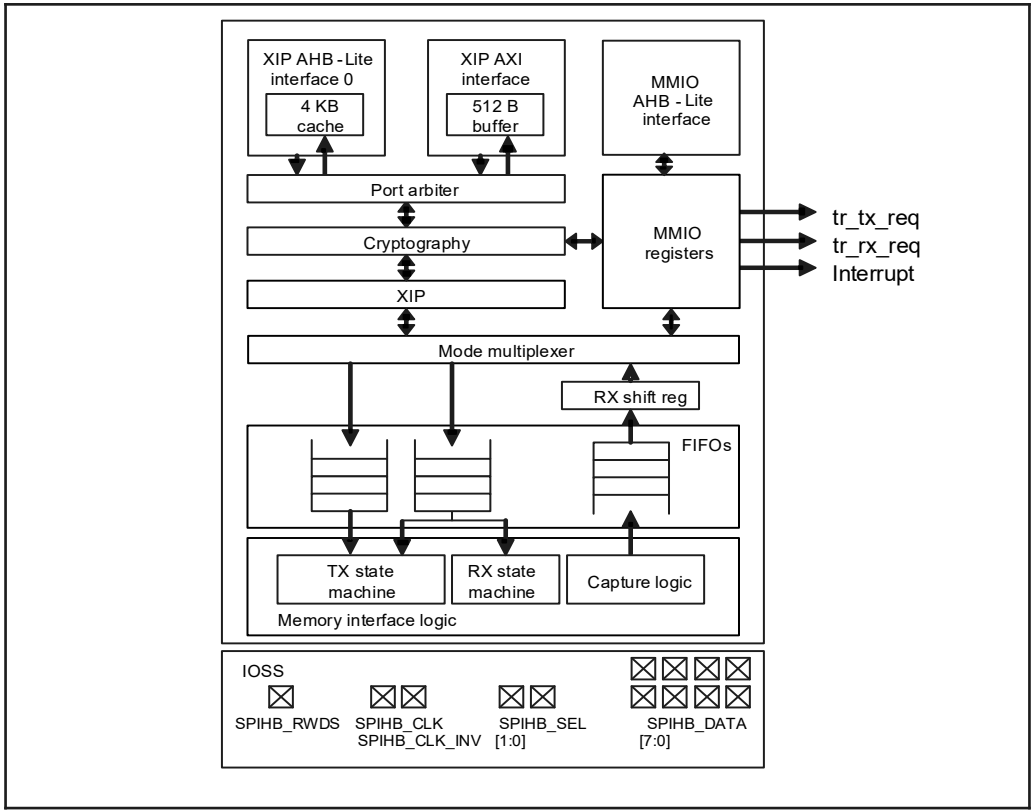
- All bus interfaces provide access to external memories
- Any of the three bus interfaces can access external memories at any time
  - MMIO AHB-Lite interface
    - Access to the MMIO registers
    - Supports MMIO operation mode
  - XIP AXI interface
    - AXI interface for CPUSS fast domain[1]
    - Supports XIP operation mode
    - Fixed 512B buffer[2]
  - XIP AHB-Lite interface
    - AHB-Lite interface for CPUSS slow domain[3]
    - Supports XIP operation mode
    - 4 KB read-only cache[3]
      - The ‘hit’ read transfers are processed by the cache
      - Four-way set associative with an LRU replacement scheme

[1] The CPUSS fast domain component is Cortex®-M7 CPUs.
[2] Note that the cache and the AXI interface buffer are not retained in Deep Sleep power mode.
[3] The CPUSS slow domain components are Cortex®-M0+ Crypto and P-DMA.

Copyright © Infineon Technologies AG 2026. All rights reserved.

Block components (3/33)

- FIFOs
- Two operation modes
  - MMIO operation mode
  - XIP operation mode
  - Mode is switched by XIP_MODE[1]
  - Both modes are mutually exclusive
- Continuous transfer merging



[1] The CPUSS fast domain component is rhe Cortex®-M7 CPUs.

Copyright © Infineon Technologies AG 2026. All rights reserved.

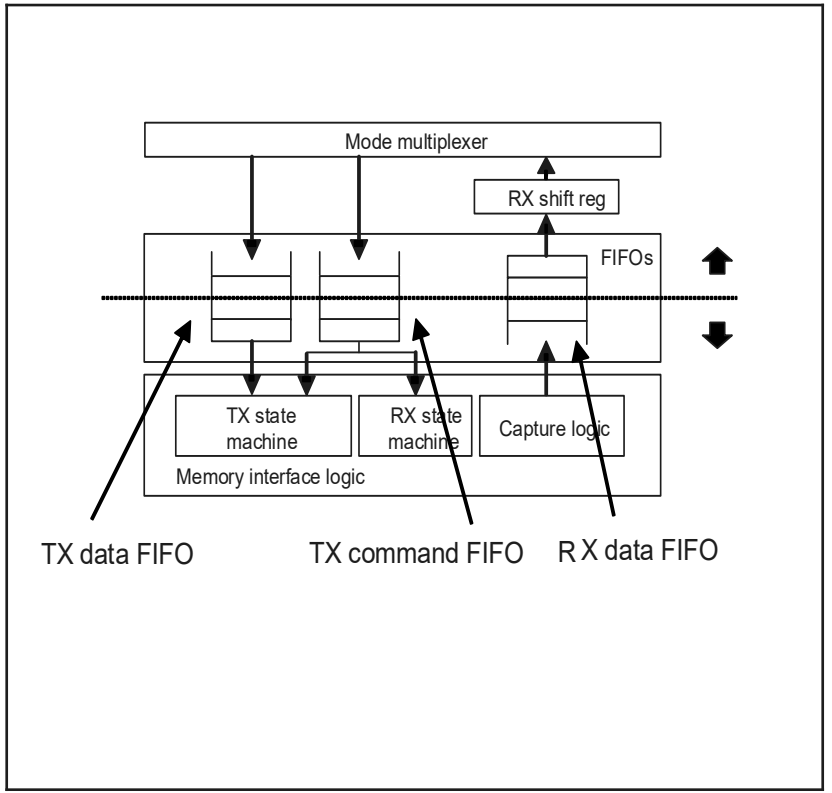
Block components (4/33)

FIFOs

SMIF has a TX command FIFO, a TX data FIFO, and a RX data FIFO:

- Provide an asynchronous clock domain transfer between clk_mem and clk_if_tx/clk_if_rx
- Software[1] controls the FIFOs in MMIO operation mode
- MMIO registers provide access to FIFOs

FIFO	Description
TX command FIFO	– Transmit memory commands to the memory interface logic for memory transfer – Support five types of commands
TX data FIFO	Transmit write data to the memory interface logic
RX data FIFO	Receive read data from the memory interface logic



Copyright © Infineon Technologies AG 2026. All rights reserved.

Block components (5/33)

FIFOs

TX command FIFO supports five command types:

- TX command FIFO is controlled by the SMIF_TX_CMD_FIFO_WR.DATA27[26:0][1] register
- DATA27[26:24] specifies the command and DATA[23:0] sets the command specification depending on command type

Command: DATA27[26:24]	Specification: DATA27[23:0]
TX: 0	– Width of the data transfer (single, dual, quad, or octal data transfer) – Data transfer mode (SDR or DDR) – External device select (multiple devices can be selected simultaneously) – The command is for the last phase of memory transfer – Transfer of 1-2 bytes **Note:** A memory transfer must start with a TX command.
TX_COUNT: 1	This command is used to transmit data from TX data FIFO to external memories – Value of memory data to be transmitted – Width of the data transfer – Data transfer mode (SDR or DDR) – Specifies if this command is for the last phase of the memory transfer

[1] See the reference manual for additional details.

Copyright © Infineon Technologies AG 2026. All rights reserved.

Block components (6/33)

FIFOs

The TX command FIFO supports five command types:

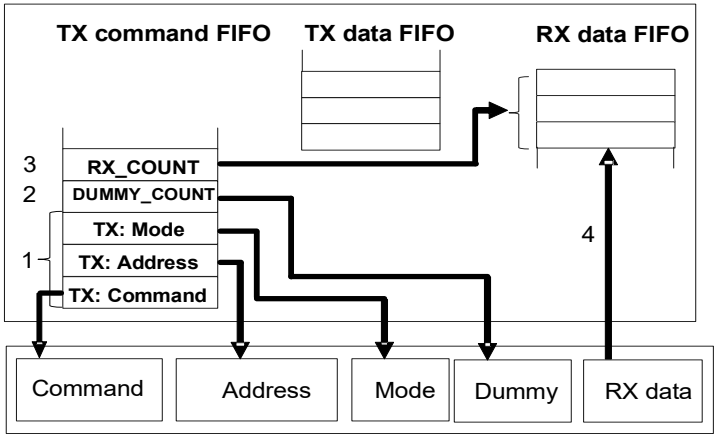
- TX command FIFO is controlled by the SMIF_TX_CMD_FIFO_WR.DATA27[26:0][1] register
- DATA27[26:24] specifies the command and DATA[23:0] sets the command specification depending on command type

Command: DATA27[26:24]	Specification: DATA27[23:0]
RX_COUNT: 2	This command is used by RX data FIFO to receive data from external memories – Value of the memory data received – Width of the data transfer – Data transfer mode (SDR or DDR) – Specifies if this command is for the last phase of the memory transfer
DUMMY_COUNT: 3	– Number of dummy cycles (used to implement turnaround time) – If the variable latency mode for HYPERRAM™ is enabled causing double the number of dummy cycles, this command never constitutes the last phase of the memory transfer
DESELECT: 4	– Finish a transfer and deselect the memory device – This command always constitutes the last phase of the memory transfer

Copyright © Infineon Technologies AG 2026. All rights reserved.

Block components (7/33)

Read command



1. Set TX command FIFO in the order of Command, Address, and Mode using the ‘TX’ command
2. Set dummy cycle to the TX command FIFO using the ‘DUMMY_COUNT’ command
3. Set number of reception data to the TX command FIFO using the ‘RX_COUNT’ command
4. Receive the number of data set by RX_COUNT in the RX data FIFO

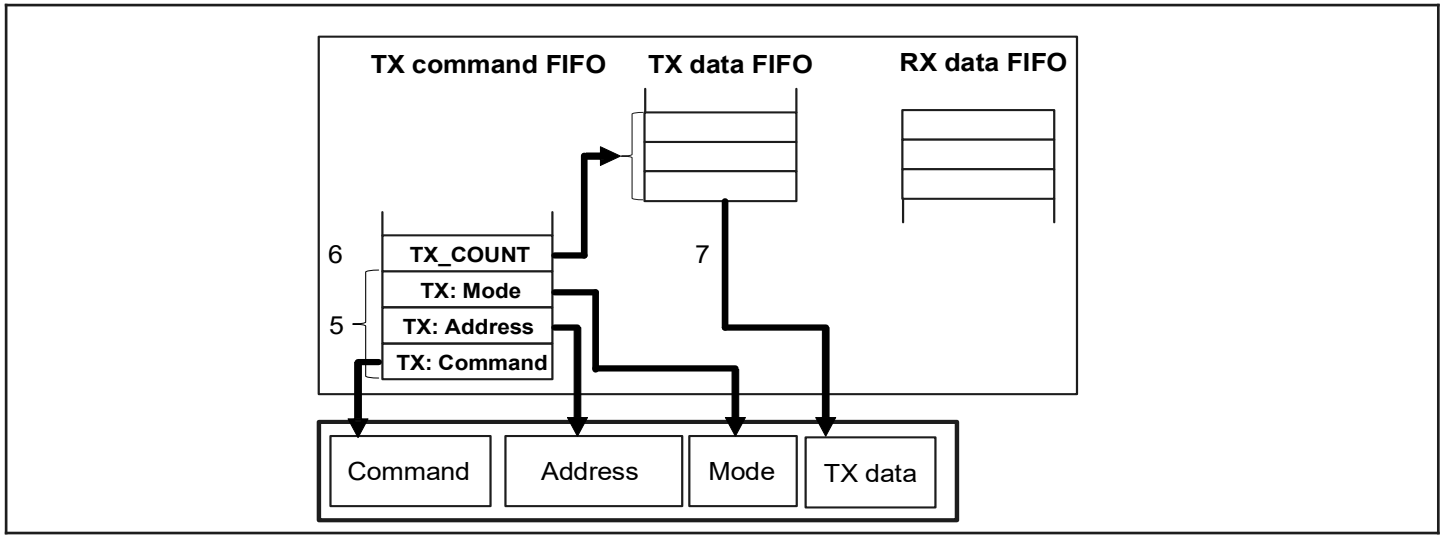
Hint bar

- See the architecture reference manual, Features (section 30.1), for additional details
- Mode:
  It can eliminate the overhead of SIO instructions when repeating the same type of read command

Copyright © Infineon Technologies AG 2026. All rights reserved.

Block components (8/33)

Write command



5. Set TX command FIFO in the order of Command, Address, and Mode using the ‘TX’ command
6. Set the number of transmission data to the TX command FIFO using the ‘TX_COUNT’ command
7. Transmit the number of data set by TX_COUNT from the TX data FIFO

Hint bar

- See the architecture reference manual section 30.1 Features for additional details
- Mode:
  It can eliminate the overhead of SIO instructions when repeating the same type of read command

Copyright © Infineon Technologies AG 2026. All rights reserved.

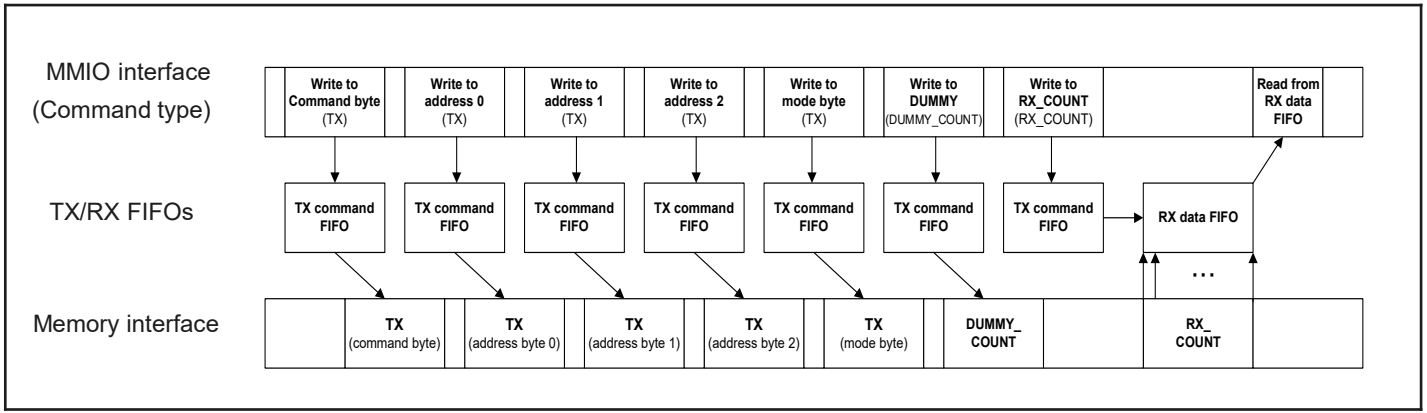
Block components (9/33)

Operation mode

MMIO operation modes

- Active by writing ‘0’ to XIP_MODE[1]
- Supports access through software using FIFOs
- Provides the flexibility to implement any SPI device transfer

Data read example



Hint bar

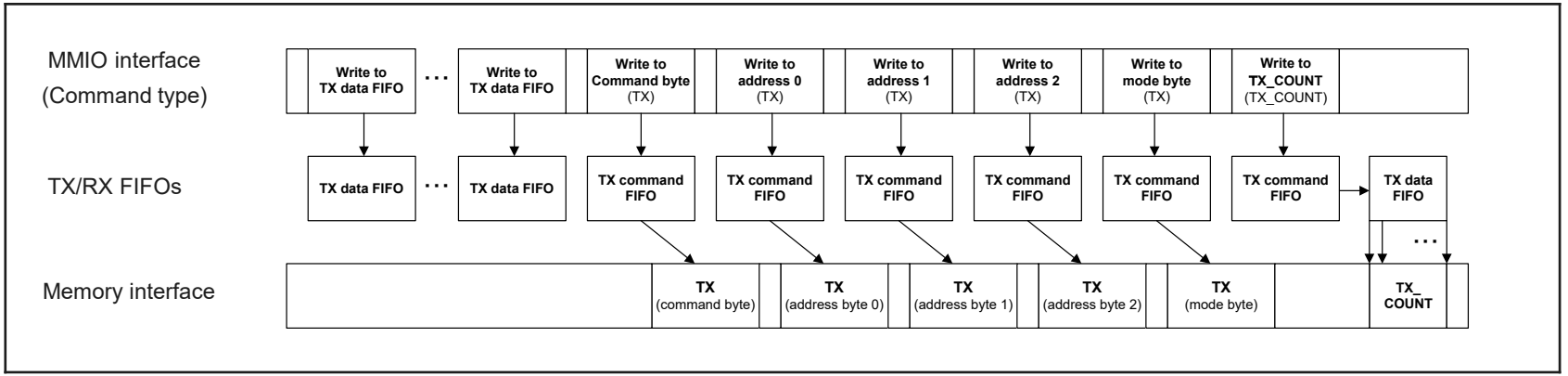
The software must ensure that it generates correct memory transfers.

[1] XIP_MODE default value is ‘0’ in the MMIO mode.

Copyright © Infineon Technologies AG 2026. All rights reserved.

Block components (10/33)

Data write example



Copyright © Infineon Technologies AG 2026. All rights reserved.

Block components (11/33)

Operation mode

XIP operation mode

- Allows mapping each external memory into an internal memory address space
- Active by writing ‘1’ to XIP_MODE
- Hardware generates memory transfers automatically without software intervention
- Supports read and write access
- FIFOs are controlled by hardware
- Read/write commands are specified by MMIO registers[1] for each device

[1] See the registers reference manual (SMIF_DEVICE_RD/WR_CMD_CTL, _RD/WR_ADDR_CTL, _RD/WR_MODE_CTL, RD/WR_DUMMY_CTL, RD/WR_DATA_CTL) for additional details.

Hint bar

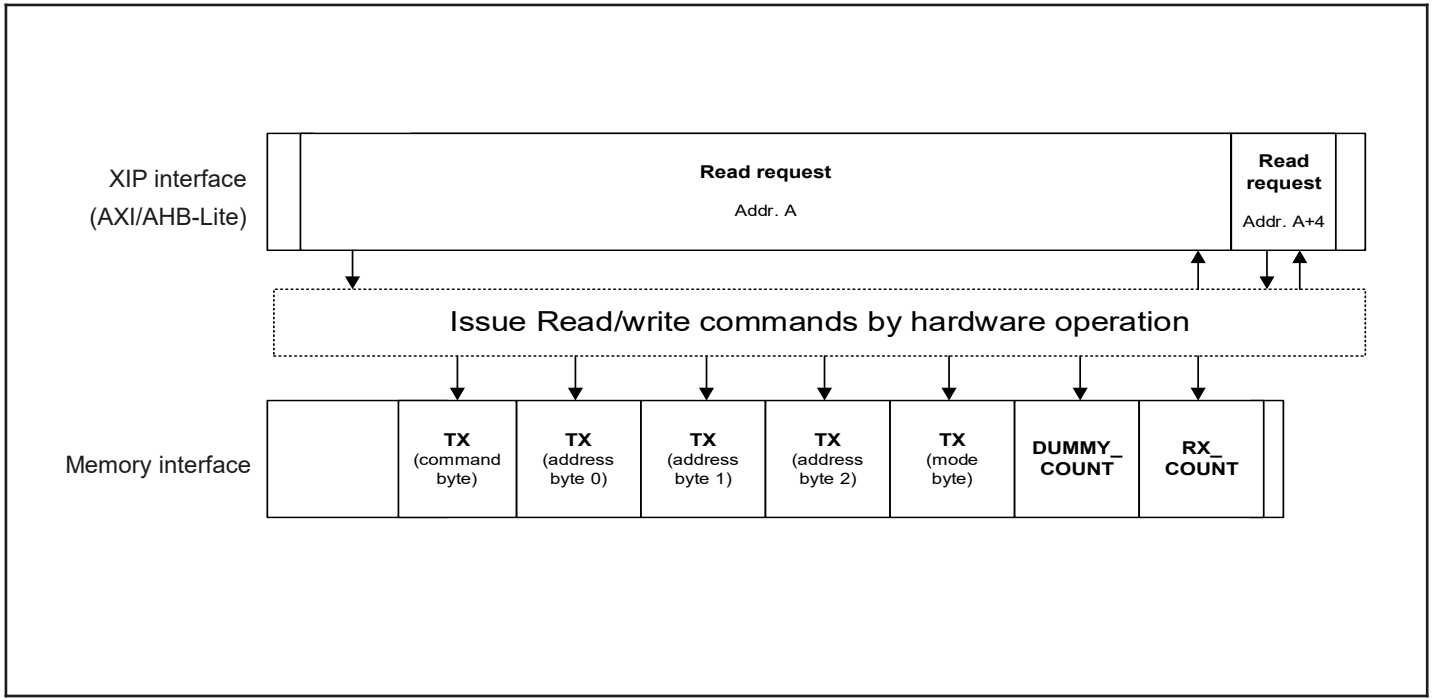
Read/write commands specification includes:

- Presence and value of SPI or HYPERBUS™ command byte
- Number of address bytes
- Presence and value of the mode byte
- Number of dummy cycles
- Data transfer widths and data transfer mode (SDR or DDR)

Copyright © Infineon Technologies AG 2026. All rights reserved.

Block components (12/33)

Data read example



Hint bar

Read/write commands specification includes:

- Presence and value of SPI or HYPERBUS™ command byte
- Number of address bytes
- Presence and value of the mode byte
- Number of dummy cycles
- Data transfer widths and data transfer mode (SDR or DDR)

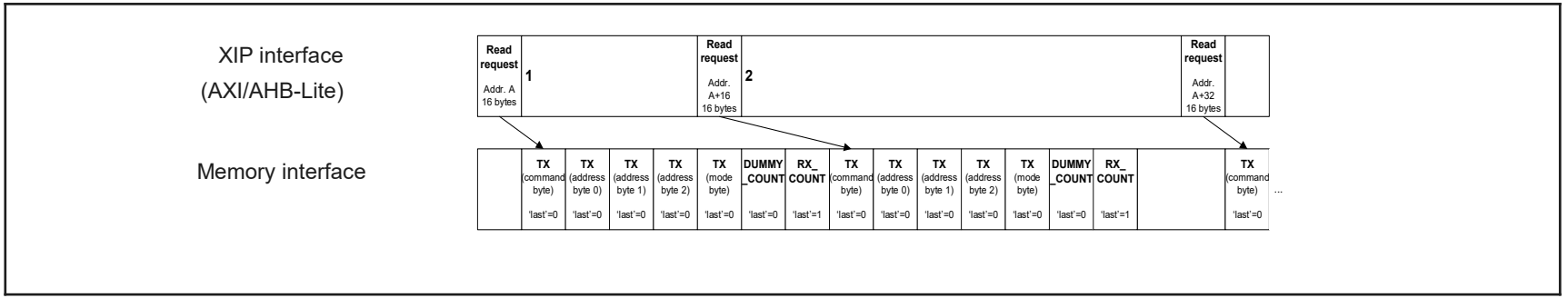
Copyright © Infineon Technologies AG 2026. All rights reserved.

Block components (13/33)

Continuous transfer merging

- The transfer can be merged to a single transfer at the memory interface[1] (MMIO and XIP modes)
  - Background: Longer transactions split on the AXI bus due to arbitration and crypto support
  - Avoid the overhead of multiple commands, addresses, modes, and dummy (latency) cycles

Read without continuous merging



1. 16-byte read transfer from address A involves command, address, and dummy cycle
2. Read transfers from continuous addresses (A+16) also involves command, address, and dummy cycles

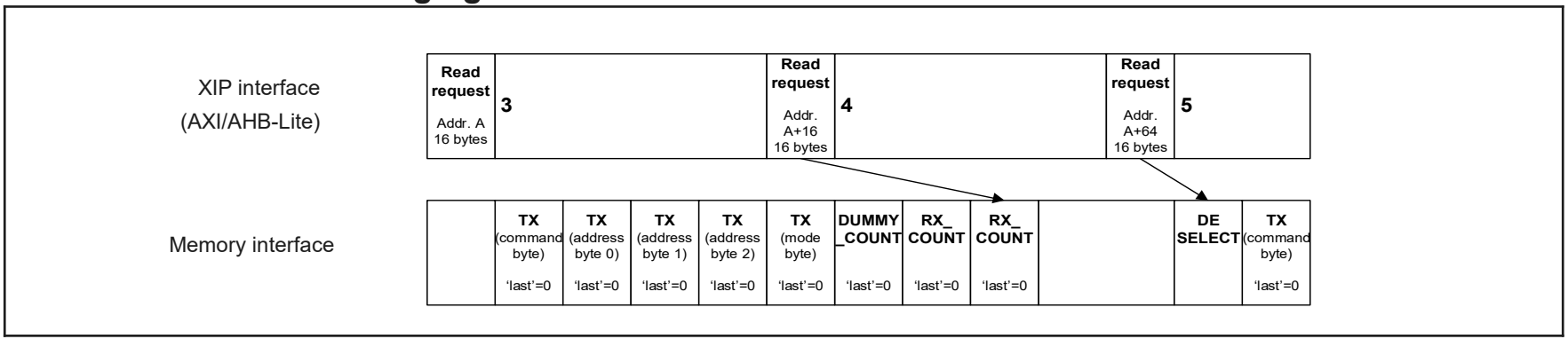
[1] See the registers reference manual (SMIF_CTL.MERGE_EN) for additional details.

Copyright © Infineon Technologies AG 2026. All rights reserved.

Block components (14/33)

Continuous transfer merging

Read with continuous merging



3. 16-byte read transfer from address A involves command, address, and dummy cycle
4. Read transfers from continuous addresses (A+16) do not require command, address, and dummy cycles
5. When a new transfer is requested, or the start address is not a continuation (A+64), a DESELECT command is generated. The next read transfer requires a command, address, and dummy cycle

Advantage

– Improves performance of multiple continuous transfers

[1] See the registers reference manual (SMIF_CTL.MERGE_EN) for additional details.

Copyright © Infineon Technologies AG 2026. All rights reserved.

Block components (15/33)

Bus error generation

External memory accesses cause a bus error under the following conditions:

Interface	Precondition	Bus error factor
MMIO (AHB-Lite)	– SMIF_CTL.BLOCK = 0[1]	Software attempts to write an entry of a full TX command FIFO
		Software attempts to write more bytes than the available entries in the TX data FIFO
		Software attempts to read more bytes than the available entries in the RX data FIFO
XIP (AXI/AHB-Lite)	– SMIF_CTL.ENABLED = 0 [1] – Write or read access	SMIF is disabled
	– SMIF_CTL.XIP_MODE = 0 [1] – Write or read access	SMIF is not in XIP mode
	– XIP mode (SMIF_CTL.XIP_MODE = 1) – Write or read access	Transfer request is not in a memory region
	– XIP mode (SMIF_CTL.XIP_MODE = 1) – SMIF_DEVICE_CTL.WR_EN = 0 [2]	Write transfer to a region that does not support writes
	– XIP mode (SMIF_CTL.XIP_MODE = 1) – Write access – Dual-quad SPI	Transfer address is not a multiple of 2
		Transfer size is not a multiple of 2
	– XIP mode (SMIF_CTL.XIP_MODE = 1) – Write access – Octal SPI DDR mode or HYPERBUS™ mode Memory write byte masking is not supported	Transfer address is not a multiple of 2
		Transfer size is not a multiple of 2
	– SMIF_CTL.INT_CLOCK_DL_ENABLED = 1	No delay line tap resulting in a matching data learning pattern

[1] See the reference manual (SMIF_CTL) for additional details.
[2] See the reference manual (SMIF_DEVICE_CTL) for additional details.

Copyright © Infineon Technologies AG 2026. All rights reserved.

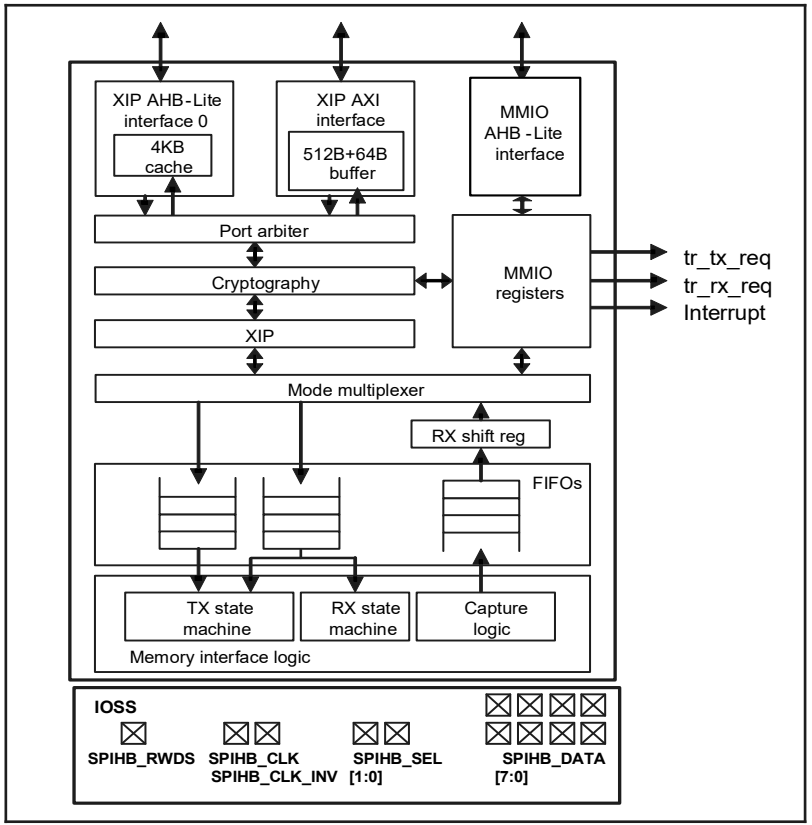
Block components (16/33)

SMIF external device connection

- Single, dual, quad, and octal data transfer
- Maximum memory interface clock
  - Depends on memory and I/O type

-		Memory type		
Products	I/O Type	SDR	DDR	HYPER BUS™
XMC7000	GPIO_STD[1]	32 MHz	32 MHz	32 MHz
	HSIO_STD[2]	100 MHz	90 MHz	100 MHz

- Can connect up to four devices
- Independent SPIHB_CLK_INV outputs
  - Less latency, and no overlapping of SPIHB_CLK



Copyright © Infineon Technologies AG 2026. All rights reserved.

Block components (17/33)

External device connection (1/7)

SPI connection

- Support for single, dual, quad, and octal SPI protocol
- Data signal type configuration in the MMIO register[1]
- Can be set independently for each memory device (chip select)

[1] See the reference manual (SMIF_DEVICE_CTL) for additional details.

Copyright © Infineon Technologies AG 2026. All rights reserved.

Block components (18/33)

External device connection (1/7)

Data signal connections

DATA_SEL[1:0]	Single SPI (Full duplex)	Dual SPI (Half duplex)	Quad SPI (Half duplex)	Octal SPI (Half duplex)
0	SPIHB_DATA [0] = SI SPIHB_DATA [1] = SO	SPIHB_DATA [0] = IO0 SPIHB_DATA [1] = IO1	SPIHB_DATA [0] = IO0 : SPIHB_DATA [3] = IO3	SPIHB_DATA [0] = IO0 : SPIHB_DATA [7] = IO7
1	SPIHB_DATA [2] = SI SPIHB_DATA [3] = SO	SPIHB_DATA [2] = IO0 SPIHB_DATA [3] = IO1	Illegal setting	Illegal setting
2	SPIHB_DATA [4] = SI SPIHB_DATA [5] = SO	SPIHB_DATA [4] = IO0 SPIHB_DATA [5] = IO1	SPIHB_DATA [4] = IO0 : SPIHB_DATA [7] = IO3	Illegal setting
3	SPIHB_DATA [6] = SI SPIHB_DATA [7] = SO	SPIHB_DATA [6] = IO0 SPIHB_DATA [7] = IO1	Illegal setting	Illegal setting

[1] See the reference manual (SMIF_DEVICE_CTL) for additional details.

Copyright © Infineon Technologies AG 2026. All rights reserved.

Block components (19/33)

External device connection (2/7)

SPI connection

– The following diagrams show examples of a single SPI connection

Features

– Data signal connections are configured by DATA_SEL

– Chip select signal setting is configured depending on the operation mode

Operation mode	Setting register	Chip select	Note
MMIO	TX command FIFO DATA[23:20][1]	DATA[21]: SPIHB0_SEL[1] DATA[20]: SPIHB0_SEL[0]	–
		DATA[23]: SPIHB1_SEL[1] DATA[22]: SPIHB1_SEL[0]	Supports XMC7000
XIP	SMIF0_DEVICE0 [2]	SPIHB0_SEL[0]	–
	SMIF0_DEVICE1 [2]	SPIHB0_SEL[1]	
	SMIF1_DEVICE0 [2]	SPIHB1_SEL[0]	Supports XMC7000
	SMIF1_DEVICE1 [2]	SPIHB1_SEL[1]	

[1] See the registers reference manual (SMIF_TX_CMD_FIFO_WR) for additional details. Multi-chip selects can be set at the same time in dual-quad mode.
[2] See the registers reference manual (SMIF_DEVICE) for additional details.

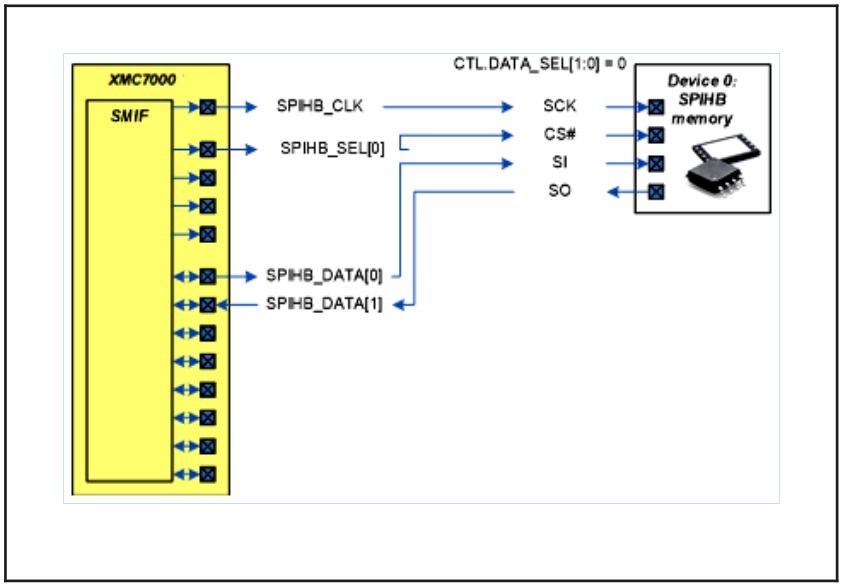
Copyright © Infineon Technologies AG 2026. All rights reserved.

Block components (20/33)

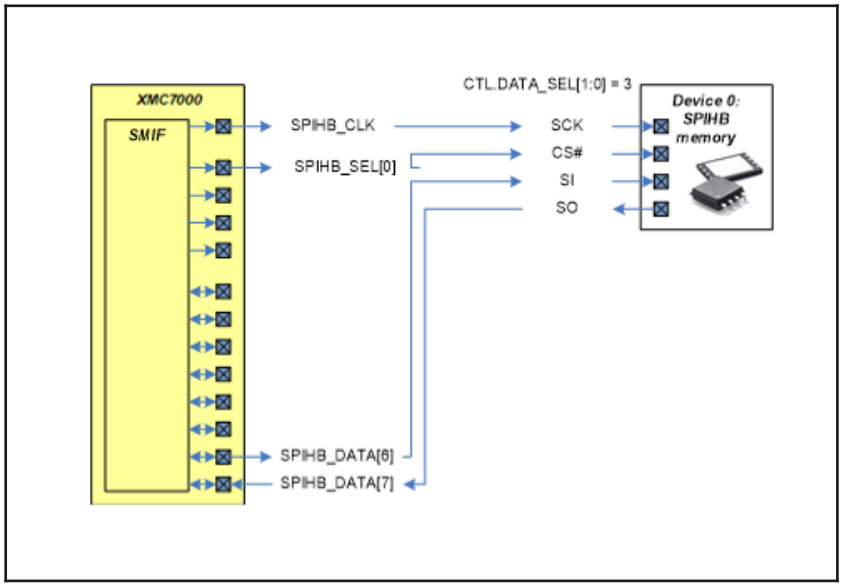
External device connection (3/7)

Single SPI connection examples

Example 1



Example 2



Copyright © Infineon Technologies AG 2026. All rights reserved.

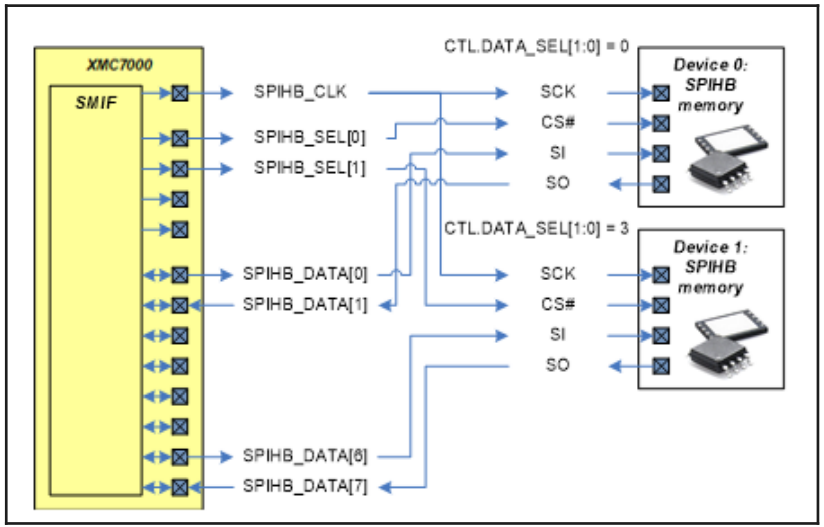
Block components (21/33)

External device connection (4/7)

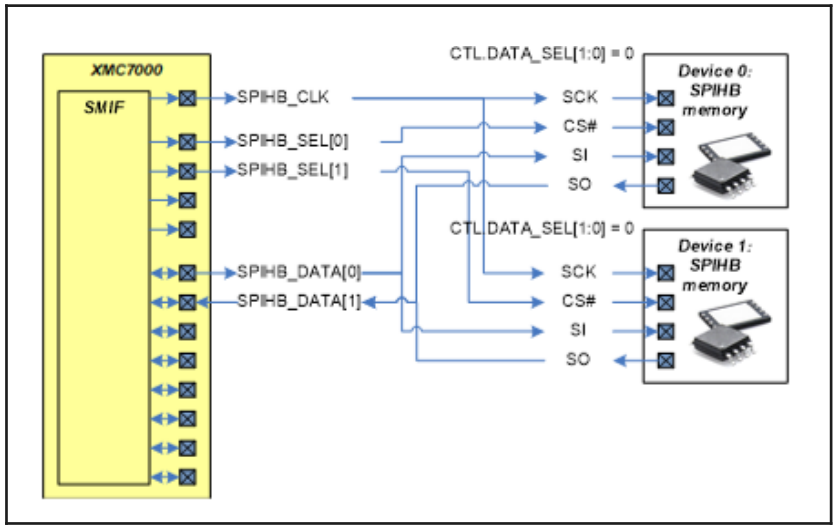
SPI connection

– The following diagrams provide examples of two single SPI device connections

Single SPI with separate data signals



Single SPI with shared data signals



Copyright © Infineon Technologies AG 2026. All rights reserved.

Block components (22/33)

External device connection (5/7)

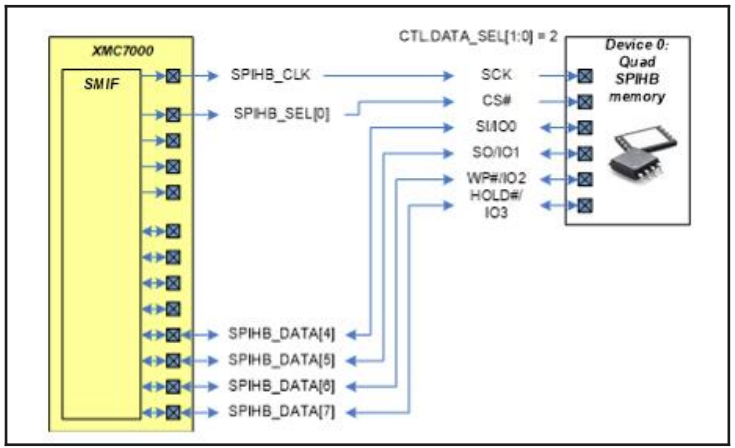
SPI connection

– The following diagrams show examples of quad and dual-quad SPI connections

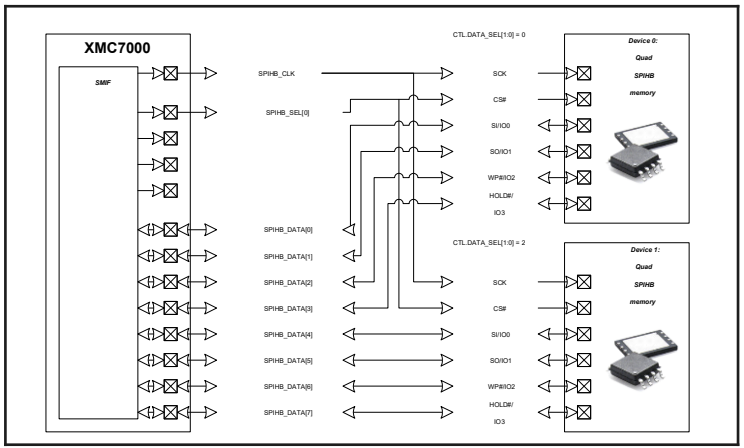
Features

– Quad SPI can also use SPIHB_DATA [4:7] by the DATA_SEL configuration

Quad SPI connection



Dual-quad SPI connection (Pseudo octal)



Copyright © Infineon Technologies AG 2026. All rights reserved.

Block components (23/33)

External device connection (6/7)

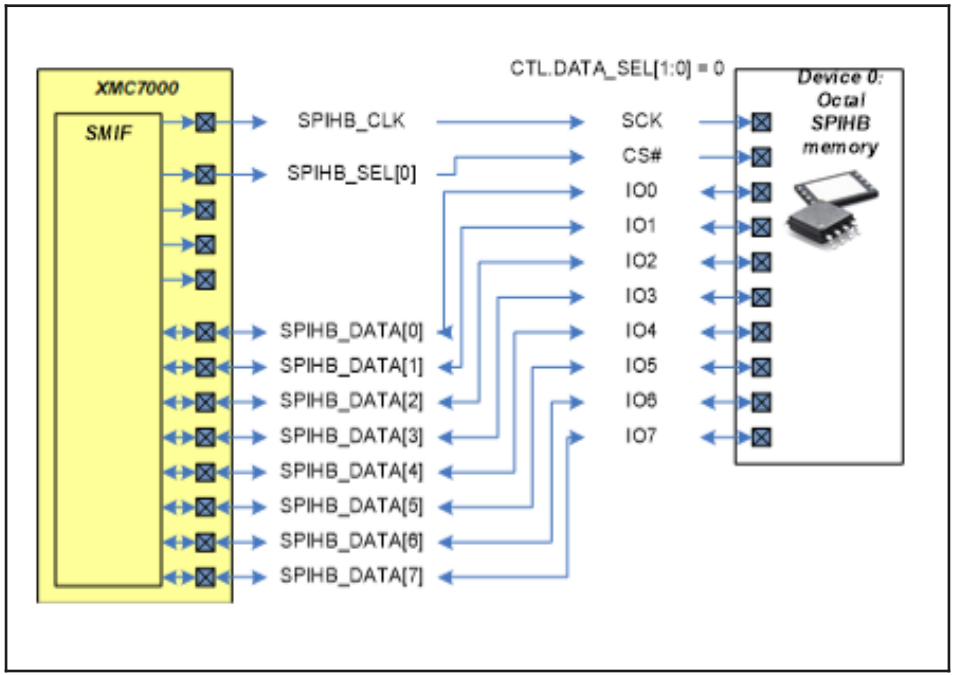
SPI connection

– The following diagram shows an example of an octal SPI connection

Features

– The only valid setting for DATA_SEL is ‘0’

Octal SPI connection



Copyright © Infineon Technologies AG 2026. All rights reserved.

Block components (24/33)

External device connection (7/7)

HYPERBUS™ connection

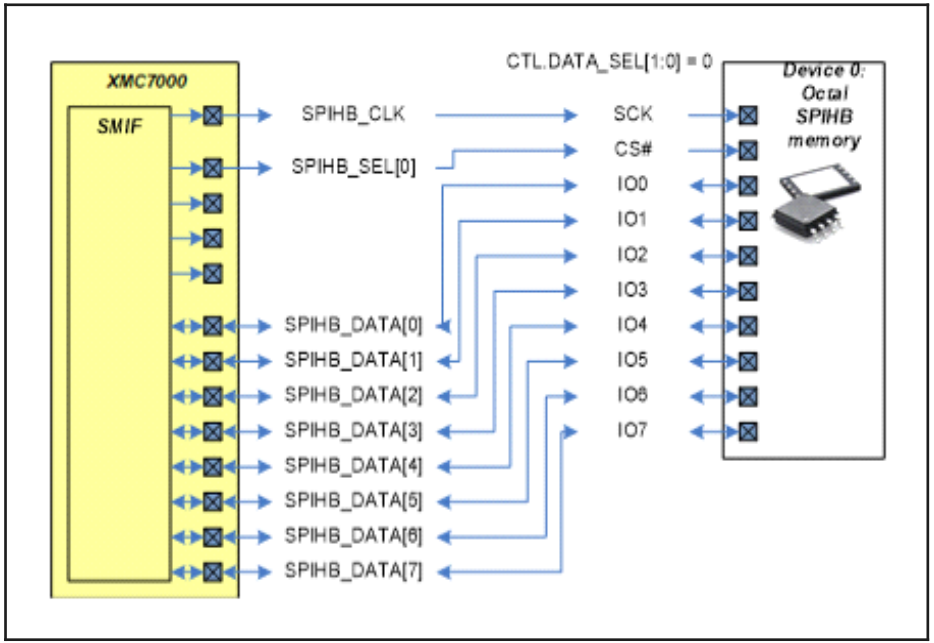
– The following diagram shows an example of the HYPERBUS™ connection

Features

– Supports read-write-data-strobe (RWDS)

– Only valid setting for DATA_SEL is ‘0’

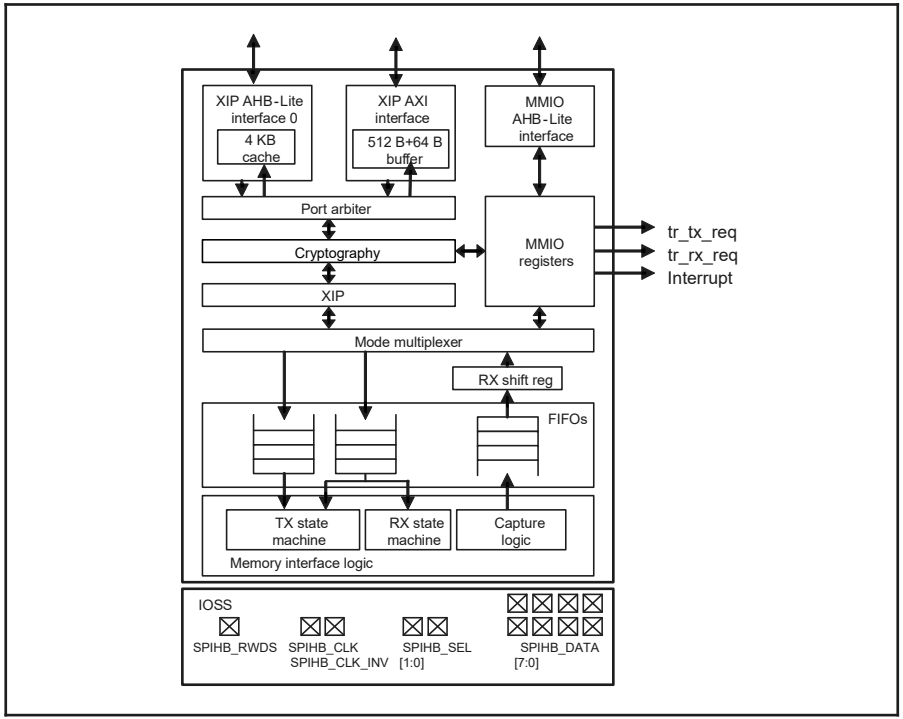
HYPERBUS™ connection



Copyright © Infineon Technologies AG 2026. All rights reserved.

Block component (25/33)

– Cryptography



Copyright © Infineon Technologies AG 2026. All rights reserved.

Block component (26/33)

Cryptography

Read and write data transfers are available for encryption and decryption:

- Encryption and decryption are based on the AES-128 forward block cipher
  - The key is programmed into the SMIF_CRYPTO_KEY3-0 registers[1]
- In MMIO mode
  - Supports offline encryption and decryption
  - Provides plaintext in SMIF_CRYPTO_INPUT3-0[1]
  - Provides ciphertext in SMIF_CRYPTO_OUTPUT3-0[1]
- In XIP mode
  - Supports on-the-fly encryption for write data
  - Supports on-the-fly decryption for read data
  - Uses XIP address for plaintext and extends to SMIF_CRYPTO_INPUT3-0

Hint bar

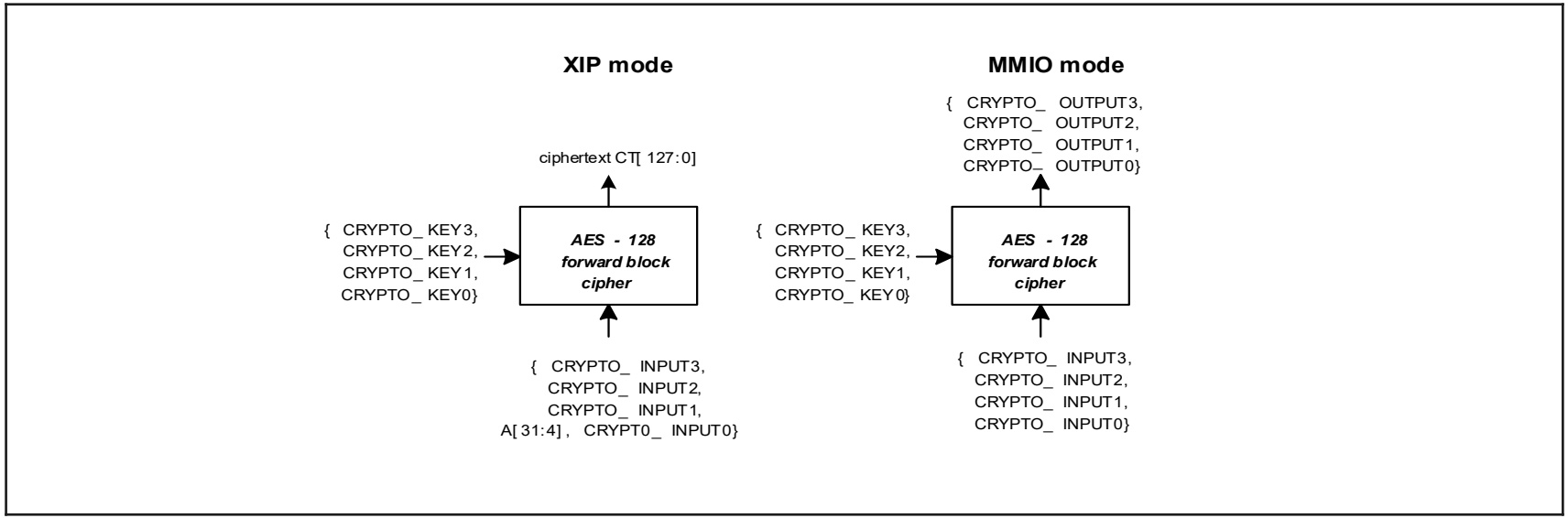
SMIF_CRYPTO_KEY registers are write-only. Software read returns ‘0’.

[1] See the reference manual for additional details.

Copyright © Infineon Technologies AG 2026. All rights reserved.

Block component (27/33)

Cryptography



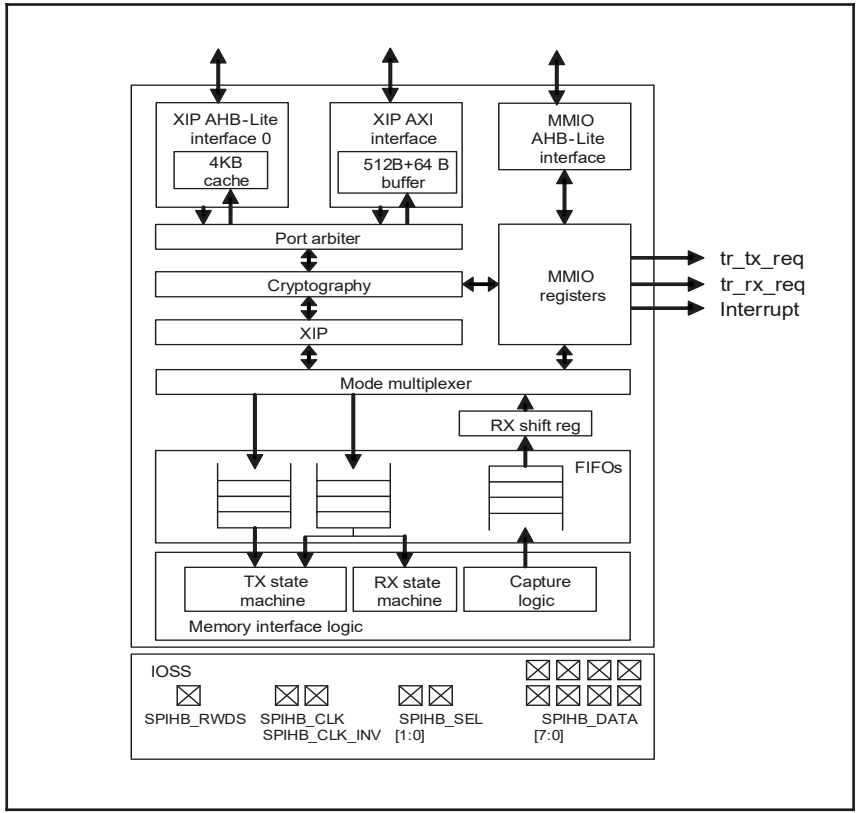
Advantage

– Storing data encrypted in external memory devices prevents leakage of sensitive data

Copyright © Infineon Technologies AG 2026. All rights reserved.

Block components (28/33)

- Trigger and interrupt
  - tr_tx_req
  - tr_rx_req
  - Interrupt



Copyright © Infineon Technologies AG 2026. All rights reserved.

Block components (29/33)

Trigger and interrupt

SMIF has the following two triggers:

Trigger	Condition
tr_tx_req	– Active when the number of used TX data FIFO entries is smaller than or equal to the specified number[1] – Activated in MMIO operation mode
tr_rx_req	– Active when the number of used RX data FIFO entries is greater than the specified number[2] – Activated in MMIO operation mode

[1] See the registers reference manual (SMIF_TX_DATA_MMIO_FIFO_CTL and SMIF_TX_DATA_MMIO_FIFO_STATUS) for additional details.
[2] See the registers reference manual (SMIF_RX_DATA_MMIO_FIFO_CTL and SMIF_RX_DATA_MMIO_FIFO_STATUS) for additional details.

Copyright © Infineon Technologies AG 2026. All rights reserved.

Block components (30/33)

Trigger and interrupt

SMIF has one interrupt:

Interrupt cause	Condition
TR_TX_REQ	– Active when the tr_tx_req trigger is activated – Activated in MMIO operation mode
TR_RX_REQ	– Active when the tr_rx_req trigger is activated – Activated in MMIO operation mode
XIP_ALIGNMENT_ERROR[1]	– The XIP AHB-Lite/AXI bus transfer address is not a multiple of ‘2’ or – The requested XIP AHB-Lite/AXI bus transfer size is not a multiple of ‘2’ In the above conditions 1. a write transfer is requested and 2. Dual-Quad SPI mode is selected or Octal SPI DDR mode or HYPERBUS™ mode is selected without memory write byte masking – Activated in XIP operation mode
TX_CMD_FIFO_OVERFLOW	– Write transfer to TX command FIFO2 with not enough free entries available – Activated in MMIO operation mode

[1] See the registers reference manual (SMIF_INTR) for additional details.
[2] See the registers reference manual (SMIF_TX_CMD_FIFO_WR) for additional details.

Copyright © Infineon Technologies AG 2026. All rights reserved.

Block components (31/33)

Trigger and interrupt

SMIF has one interrupt:

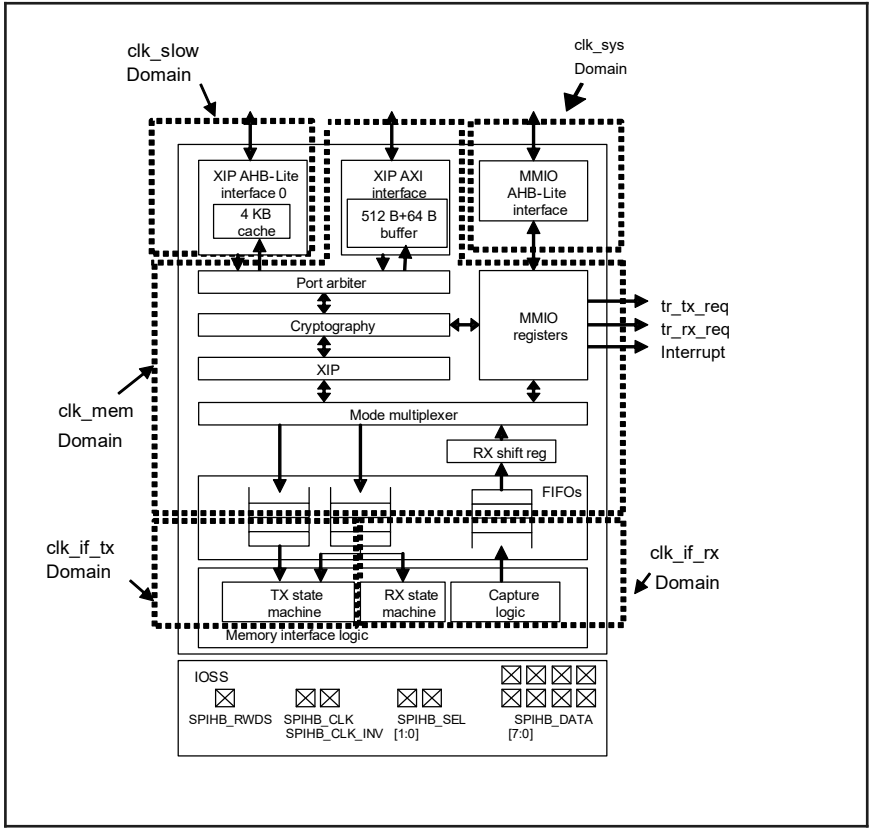
Interrupt cause	Condition
TX_DATA_FIFO_OVERFLOW	– Write transfer to TX data FIFO[3] with not enough free entries available – Activated in MMIO operation mode
RX_DATA_FIFO_OVERFLOW	– Read transfer from RX data FIFO[4] with not enough entries available – Activated in MMIO operation mode
CRC_ERROR	– CRC Error – A read transfer data CRC check failed
FS_STATUS_ERROR	– Functional Safety Status Error – A read transfer Functional Safety Status check failed

[3] See the registers reference manual (SMIF_TX_DATA_FIFO_WR1,WR2,WR4) for additional details.
[4] See the registers reference manual (SMIF_RX_DATA_FIFO_RD1,RD2,RD4) for additional details.

Copyright © Infineon Technologies AG 2026. All rights reserved.

Block components (32/33)

- Clock domains
  - CLK_MEM
  - CLK_SLOW
  - CLK_SYS
  - clk_if_tx/rx



Copyright © Infineon Technologies AG 2026. All rights reserved.

Block components (33/33)

Clock

The following clocks are used in SMIF:

Source	Used block	Function
clk_mem	SMIF Internal Block FIFOs	Used for the SMIF block; it is sourced from the mem clock The frequency range is up to 200 MHz
	AXI Master Interface	– Used by the XIP AXI master interface; it is sourced from the mem clock – The frequency range is up to 200 MHz
clk_slow	AHB Slave Interface	– Used by the XIP AHB-Lite master interface; it is sourced from the slow clock – The frequency range is up to 100 MHz
clk_sys	AHB Slave Interface	– Used by the MMIO AHB-Lite slave interface; it is clocked by the PERI group clock – The frequency range is up to 100 MHz
clk_if_tx/rx	Memory Interface FIFOs	– Used for memory interface of transmit and receive; it is clocked by the CLK_HF6 – clk_if_tx is the source clock for SPIHB_CLK

Hint bar

See the architecture reference manual, Clocking system (chapter 18), for additional details.

Copyright © Infineon Technologies AG 2026. All rights reserved.

Table of contents

1 Target products 3

2 Introduction to XMC7200 - 8M 6

3 Introduction to XMC7100 - 4M 9

4 Serial memory interface (SMIF) 11

5 Block components 16

6 **Revision history** **51**

Copyright © Infineon Technologies AG 2026. All rights reserved.

Revision history

Document revision	Date	Description of change
**	2022-10-19	Initial release.
*A	2026-01-04	Migrated to the latest empower training presentation template.

Copyright © Infineon Technologies AG 2026. All rights reserved.


Infineon