Customer training workshop

32-bit Arm® Cortex®-M7 Microcontroller
XMC7000 – Serial Communication Block (SCB)

Target products

› Target product list for this training material:

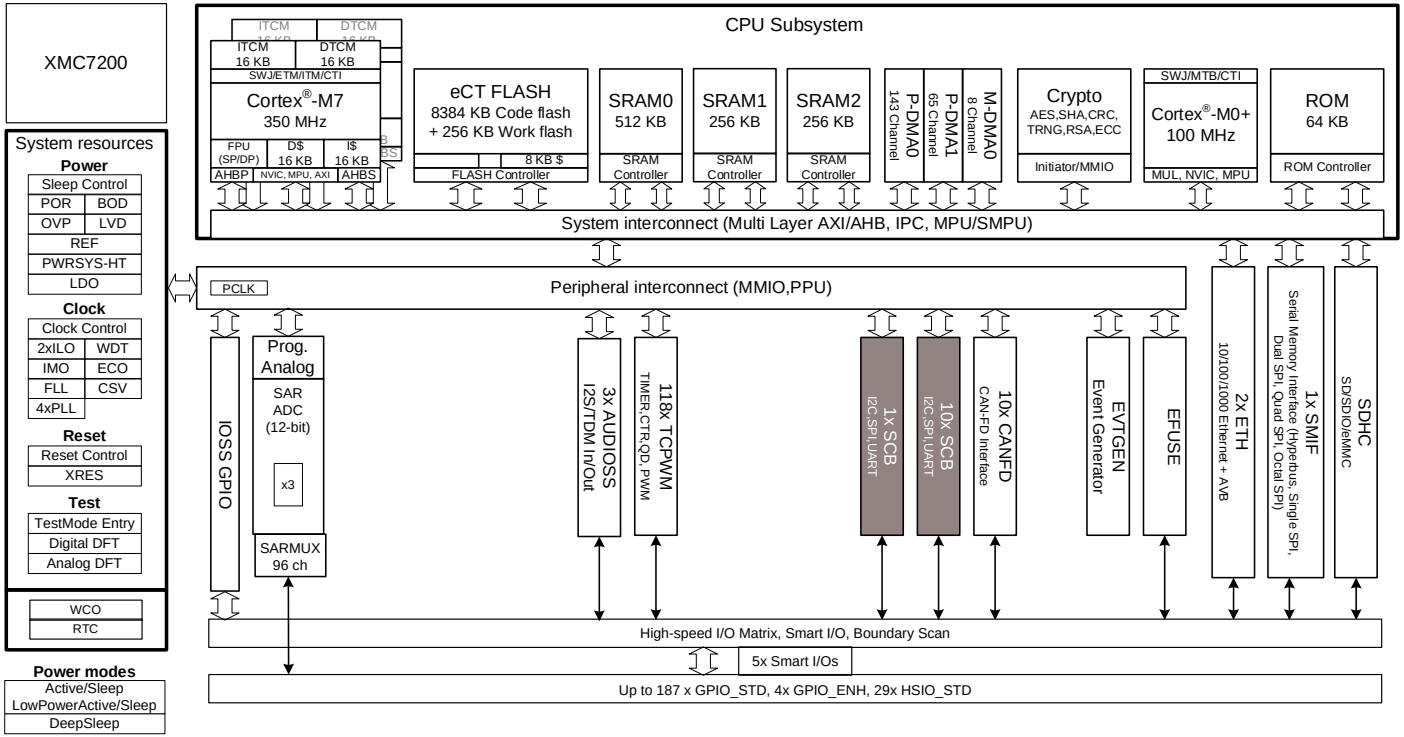
Family category	Series	Code flash memory size
XMC7200	XMC7200-8M	Up to 8384 KB
XMC7100	XMC7100-4M	Up to 4160 KB

Copyright © Infineon Technologies AG 2022. All rights reserved.

XMC7200 - 8M family

Introduction to XMC7200 - 8M

SCB is part of the peripheral blocks



Hint Bar

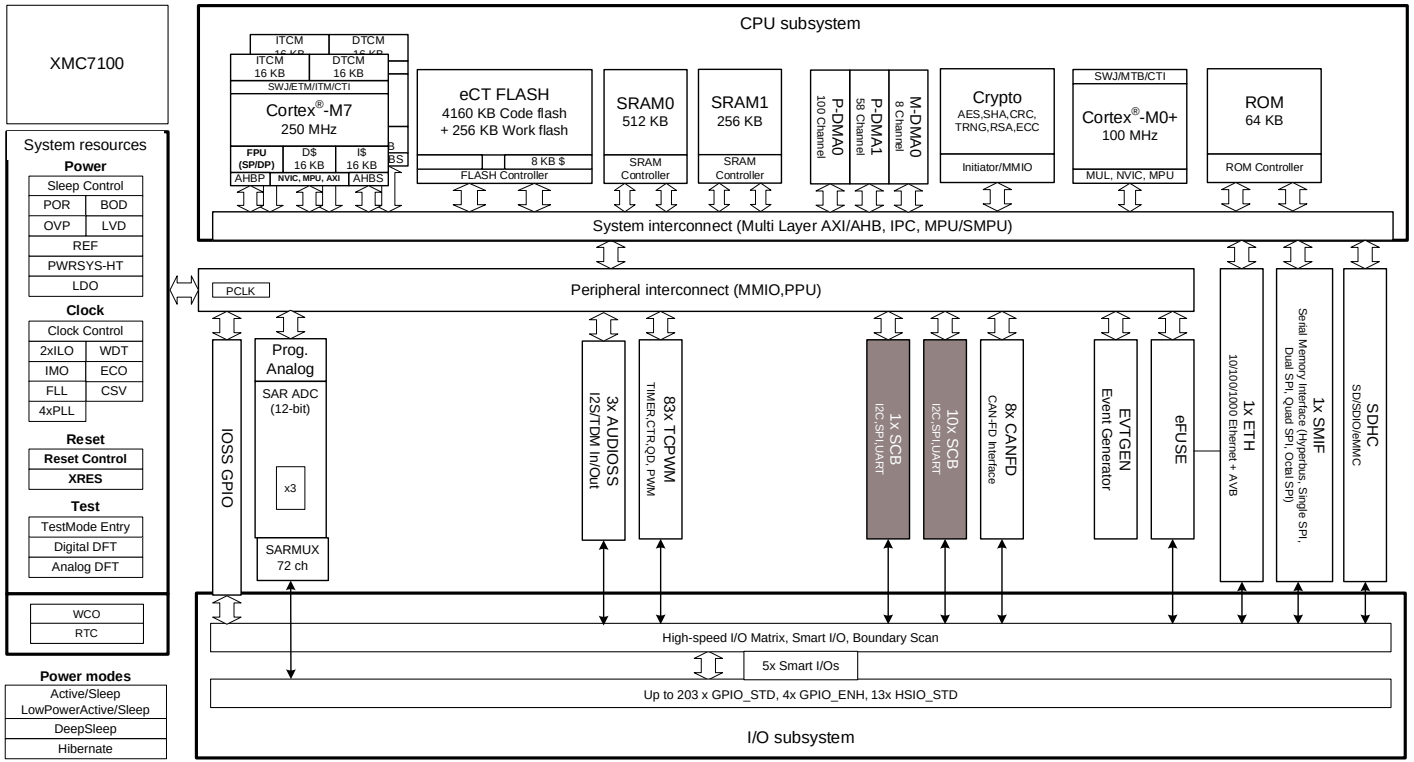
Review TRM chapter 23 for additional details

Copyright © Infineon Technologies AG 2022. All rights reserved.

XMC7100 - 4M family

Introduction to XMC7100 - 4M

SCB is part of the peripheral blocks



Hint Bar

Review TRM chapter 23 for additional details

Copyright © Infineon Technologies AG 2022. All rights reserved.

Serial communication block overview (1/3)

- Supports three serial interface protocols
  - Serial peripheral interface (SPI)
  - Universal asynchronous receiver/transmitter (UART)
  - Inter-integrated circuit (I^2C)
- SPI features
  - Supports master and slave functionality
  - Supports three SPI protocols (Motorola, Texas Instruments, National Semiconductor)
  - Supports up to four slave select lines
  - Programmable data frame size from 4 bits to 32 bits
  - Supports easy SPI (EZSPI) mode of operation

Hint Bar

Review TRM section 23 for additional details

Copyright © Infineon Technologies AG 2022. All rights reserved.

Serial communication block overview (2/3)

- UART features
  - Supports two protocols
    - Standard UART
    - Multi-processor mode
  - Programmable data frame size from 4 to 16 bits
  - Programmable number of STOP bits (can set half-bit periods between 1 and 4)
  - Programmable oversampling

Hint Bar

Review TRM section 23.5.1 for additional details

Copyright © Infineon Technologies AG 2022. All rights reserved.

Serial communication block overview (3/3)

- I^2C features
  - Master and slave mode
  - Four types of data-rate modes
    - Slow mode ,Standard mode, Fast mode, Fast mode plus
  - 7-bit slave addressing
  - Analog and digital glitch filter – up to 50 ns

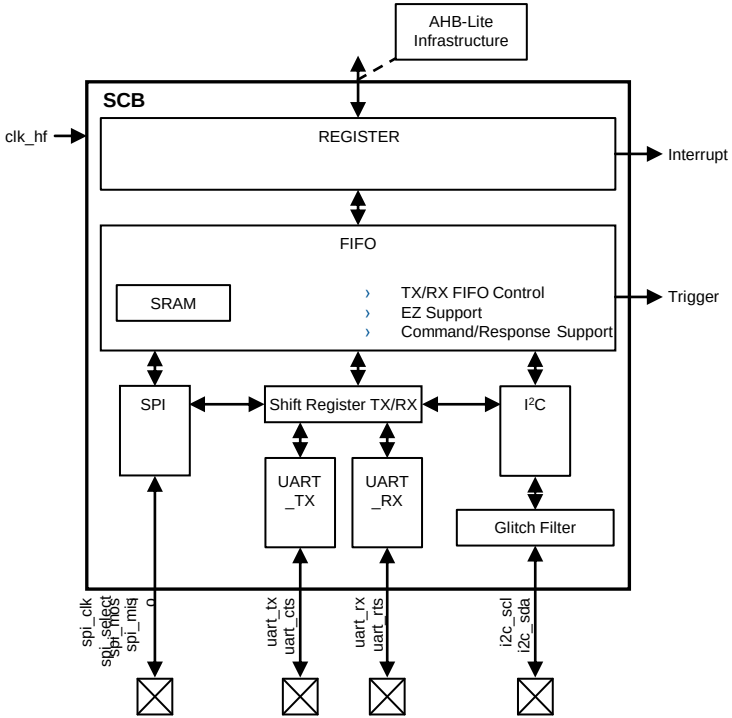
Hint Bar

Review TRM section 23.6.1 and 23.6.6.1 for additional details

Copyright © Infineon Technologies AG 2022. All rights reserved.

Serial communication block diagram

› Block diagram



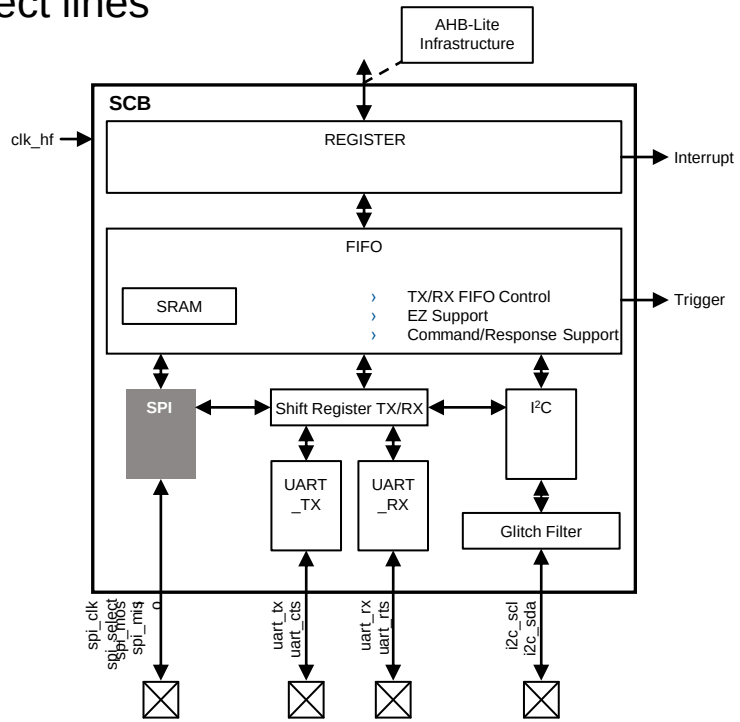
Hint Bar

Review TRM section 23.1 for additional details

Copyright © Infineon Technologies AG 2022. All rights reserved.

Serial communication block components

- SPI
  - Supports up to four slave select lines
  - Supports three SPI protocols
  - SPI Buffer mode
    - FIFO mode
    - EZSPI mode
    - Command-Response mode



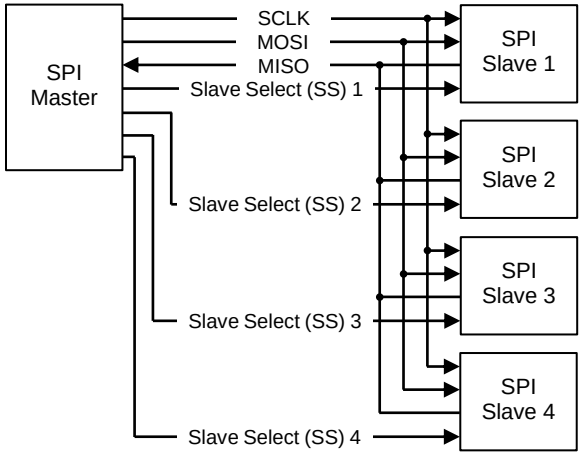
Hint Bar

Review TRM section 23.4 for additional details

Copyright © Infineon Technologies AG 2022. All rights reserved.

SPI Slave Select lines

- The standard SPI interface consists of four signals
  - SCLK: Serial clock (clock output from the master, input to the slave)
  - MOSI: Master-out-slave-in (data output from the master, input to the slave)
  - MISO: Master-in-slave-out (data input to the master, output from the slave)
  - Slave Select (SS): Typically an active-low signal (output from the master, input to the slave)
- Advantage
  - Can communicate with multiple external ICs using the same bus



Hint Bar

Review TRM section 23.4 for additional details

Copyright © Infineon Technologies AG 2022. All rights reserved.

SPI protocols

- Three SPI protocols[1]
  - The Motorola SPI[2] protocol has four clock modes
    - Data is driven and captured by clock polarity and clock phase
  - Texas Instruments SPI[2]
    - The Texas Instruments SPI protocol redefines the use of the SS signal to indicate the start of a data transfer
  - National Semiconductor SPI[3]
    - The National Semiconductor SPI protocol alternates transmission and reception

Hint Bar

Review TRM chapter 23.4 for additional details

[1] Refer to the Register TRM (SCBx_SPI_CTRL) for additional details
[2] Full Duplex
[3] Half Duplex

Copyright © Infineon Technologies AG 2022. All rights reserved.

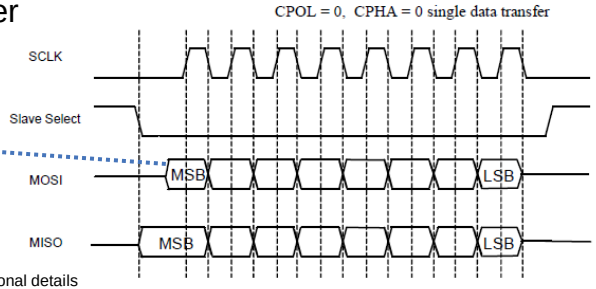
Motorola SPI mode

- The Motorola SPI protocol has four clock modes based on how data is driven and captured on the MOSI and MISO lines
- The modes are determined by clock polarity (CPOL) and clock phase (CPHA)

Mode	CPOL	CPHA	Description
0	0	0	Data is driven on a falling edge of SCLK. Data is captured on a rising edge of SCLK.
1	0	1	Data is driven on a rising edge of SCLK. Data is captured on a falling edge of SCLK.
2	1	0	Data is driven on a rising edge of SCLK. Data is captured on a falling edge of SCLK.
3	1	1	Data is driven on a falling edge of SCLK. Data is captured on a rising edge of SCLK.

- Motorola SPI data transfer example (Mode = 0)
  - A single 8-bit data transfer

Data Frame Size[1]
Programmable 4 to 32 bits



Legend:
CPOL: Clock Polarity
CPHA: Clock Phase
SCLK: SPI Interface Clock
MOSI: SPI Master-Out-Slave-In
MISO: SPI Master-In-Slave-Out

[1] Refer to the Register TRM (SCBx_TX_CTRL, SCBx_RX_CTRL) for additional details

Hint Bar

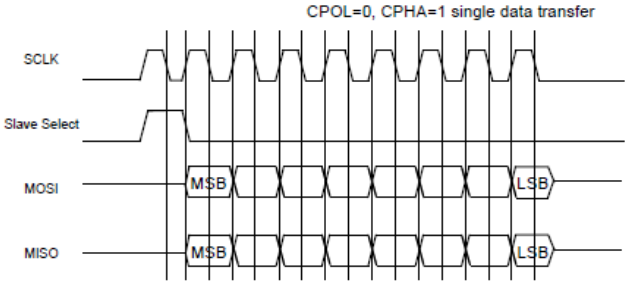
Review TRM section 23.4.3.1 for additional details

Copyright © Infineon Technologies AG 2022. All rights reserved.

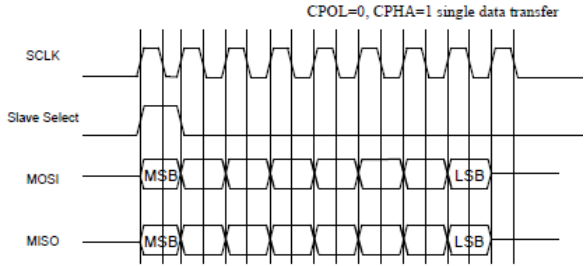
Texas Instruments SPI mode

- Texas Instruments SPI data transfer[1]
  - A single 8-bit data transfer
  - The Slave Select pulse either precedes the first data bit or coincides with the first data bit of a frame
    - Slave Select pulse preceding example

Legend:
CPOL: Clock Polarity
CPHA: Clock Phase
SCLK: SPI Interface Clock
MOSI: SPI Master-Out-Slave-In
MISO: SPI Master-In-Slave-Out



    - Slave Select pulse coinciding example



Hint Bar

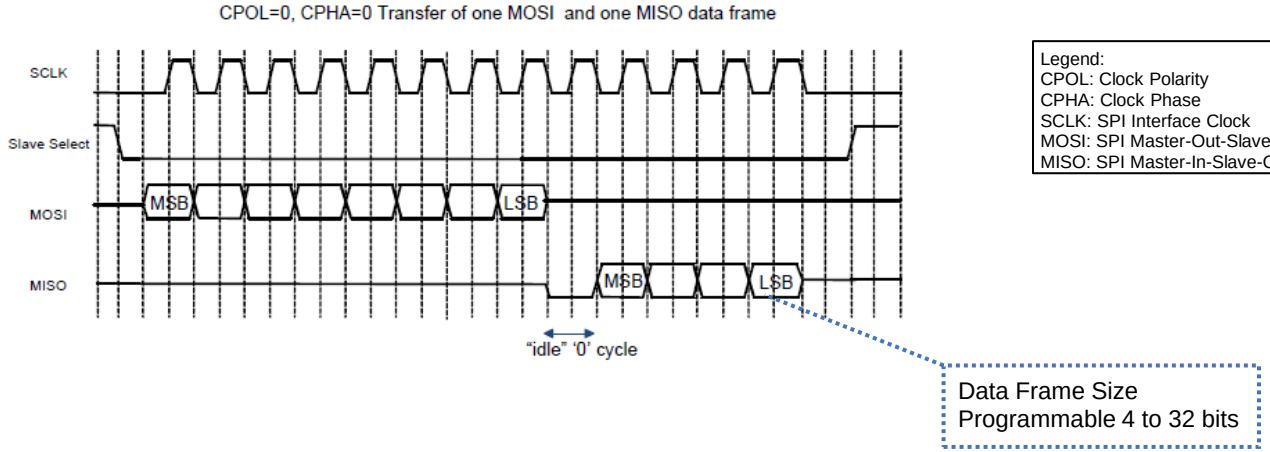
Review TRM section 23.4.3.2 for additional details

[1] Texas Instruments SPI mode uses only CPOL = 0 and CPHA = 1 settings

Copyright © Infineon Technologies AG 2022. All rights reserved.

National Semiconductor SPI mode

- National Semiconductor SPI data transfer[1]
  - The SPI protocol alternates transmission and reception
  - The transmission and reception data sizes are different
  - A single “idle” bit transfer period separates transmission from reception



Legend:
CPOL: Clock Polarity
CPHA: Clock Phase
SCLK: SPI Interface Clock
MOSI: SPI Master-Out-Slave-In
MISO: SPI Master-In-Slave-Out

Hint Bar

Review TRM section 23.4.3.3 for additional details

[1] National Semiconductor SPI mode uses only CPOL = 0 and CPHA = 0 settings

Copyright © Infineon Technologies AG 2022. All rights reserved.

SPI buffer modes (1/2)

- SPI can operate in three buffer modes[1]
  - FIFO mode
  - EZSPI mode
  - Command-Response mode
- Restrictions on using buffer mode
  - Master can only use FIFO mode
  - Slave can use all the three buffer modes listed above. However, the Command-Response mode is available only on DeepSleep-capable SCB
- SCB has an internal SRAM buffer
  - Capacity of 128 × 16 bits
  - Supports byte mode
  - Can be configured as 256 × 8 bits

Hint Bar

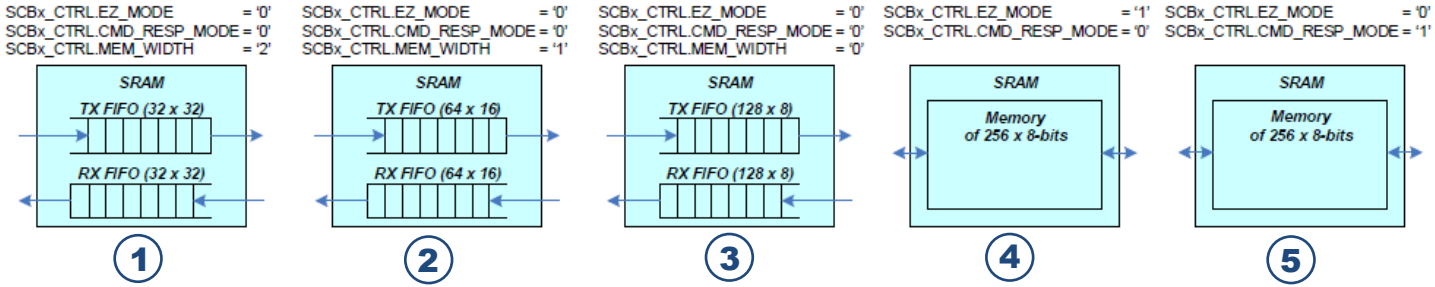
Review TRM section 23.4 for additional details

[1] Refer to the Register TRM (SCBx_CTRL.MODE) for additional details

Copyright © Infineon Technologies AG 2022. All rights reserved.

SPI buffer modes (2/2)

- Buffer mode can be configured in five types in dedicated SRAM[1]
  1. Two 32 deep FIFOs for up to 32-bit data elements
  2. Two 64 deep FIFOs for up to 16-bit data elements
  3. Two 128 deep FIFOs for up to 8-bit data elements
  4. One 256 Byte EZ memory buffer
  5. One 256 Byte CMD_RESP memory buffer



Hint Bar

Review TRM section 23.3.1 for additional details

[1] Refer to the Register TRM (SCBx_CTRL.EZ_MODE, SCBx_CTRL.CMD_RESP_MODE, SCBx_CTRL.MEM_WIDTH) for additional details

Copyright © Infineon Technologies AG 2022. All rights reserved.

FIFO mode

- FIFO mode includes TX and RX FIFOs
- Transmit and receive FIFO status[1] information indicates
  - Empty
  - Not empty
  - Full
  - Not full
- Provides “underflow” and “overflow” events
  - An “underflow” event is triggered by an attempt to read from an empty FIFO
  - An “overflow” event is triggered by an attempt to write to a full FIFO

Hint Bar

Review TRM section 23.4.4.1 for additional details

[1] Refer to the Register TRM (SCBx_INTR_TX, SCBx_INTR_RX) for additional details

Copyright © Infineon Technologies AG 2022. All rights reserved.

EZSPI mode

- The EZSPI[1] protocol is based on the Motorola SPI
  - The EZSPI mode is a method in which the master uses an 8-bit EZ address in the slave memory to write or read data
- It defines a single memory buffer with an 8-bit EZ address
- The EZ address indexes the buffer located on the slave device
- EZSPI has three types of transfers
  - EZ address written from the master to the slave
  - Data written from the master to an addressed slave memory location
  - Slave reads from an addressed slave memory location

Hint Bar

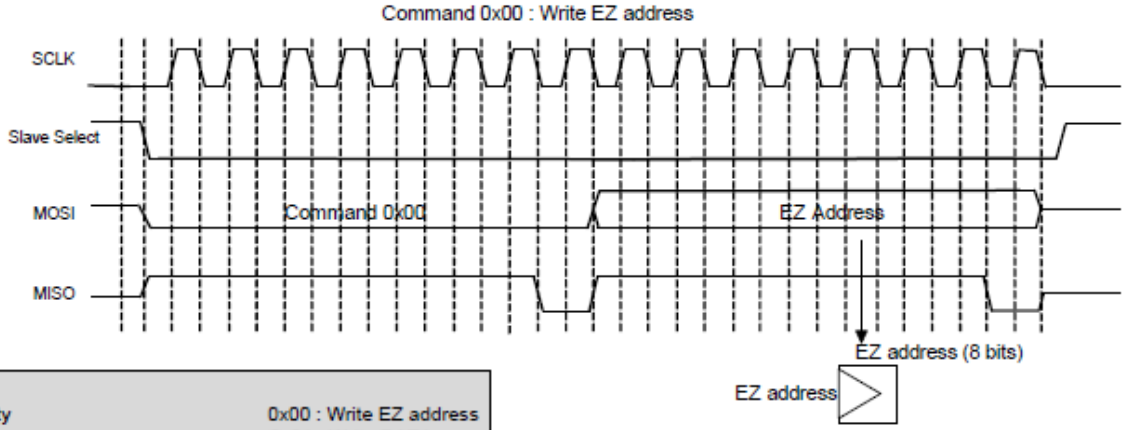
Review TRM section 23.4.4.2 for additional details

[1] Refer to the Register TRM (SCBx_CTRL.EZ_MODE) for additional details

Copyright © Infineon Technologies AG 2022. All rights reserved.

EZSPI mode (Address Write)

- EZ Address Write
  - An EZ address write starts with a command byte (0x00) on the MOSI line indicating the master's intent to write the EZ address



LEGEND :

CPOL : Clock Polarity	0x00 : Write EZ address
CPHA : Clock Phase	0x01 : Write DATA
SCLK : SPI Interface Clock	0x02 : Read DATA
MISO : SPI Master-In-Slave-Out	0xFE : "slave ready"
MOSI : SPI Master-Out-Slave-In	0xFF : "slave busy"

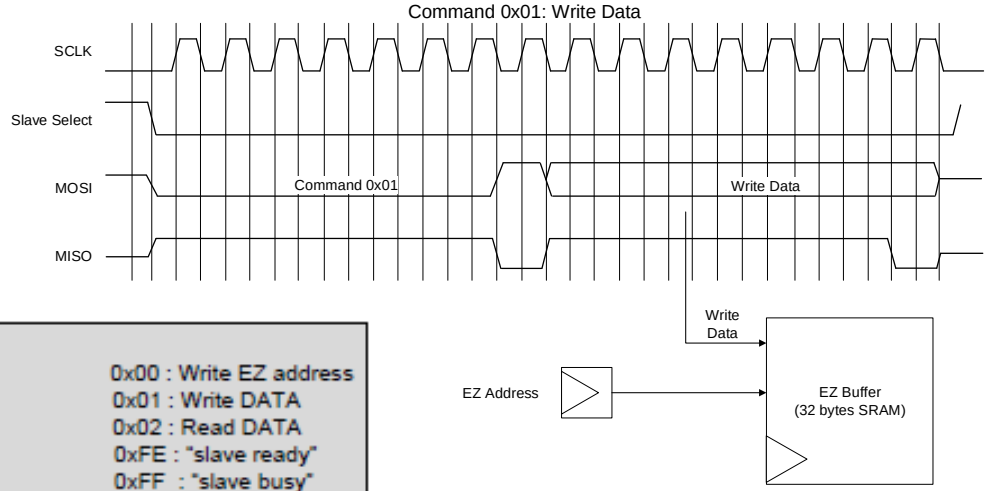
Hint Bar

Review TRM section 23.4.4.2 for additional details

Copyright © Infineon Technologies AG 2022. All rights reserved.

EZSPI mode (data write)

- Memory array write
  - A write to a memory array index starts with a command byte (0x01) on the MOSI line indicating the master's intent to write to the memory array
  - The EZ address is automatically incremented by the slave as bytes are written into the memory array



LEGEND :

CPOL : Clock Polarity	0x00 : Write EZ address
CPHA : Clock Phase	0x01 : Write DATA
SCLK : SPI Interface Clock	0x02 : Read DATA
MISO : SPI Master-In-Slave-Out	0xFE : "slave ready"
MOSI : SPI Master-Out-Slave-In	0xFF : "slave busy"

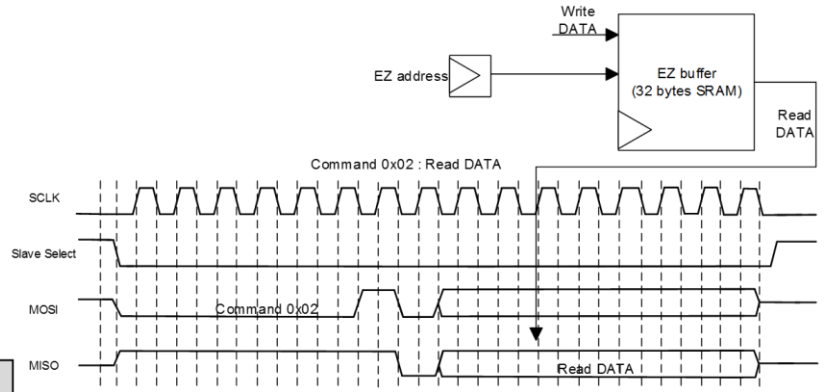
Hint Bar

Review TRM section 23.4.4.2 for additional details

Copyright © Infineon Technologies AG 2022. All rights reserved.

EZSPI mode (data read)

- Memory array read
  - A read from a memory array index starts with a command byte (0x02) on the MOSI line indicating the master's intent to read from the memory array



LEGEND :	
CPOL : Clock Polarity	0x00 : Write EZ address
CPHA : Clock Phase	0x01 : Write DATA
SCLK : SPI Interface Clock	0x02 : Read DATA
MISO : SPI Master-In-Slave-Out	0xFE : "slave ready"
MOSI : SPI Master-Out-Slave-In	0xFF : "slave busy"

- Advantage
  - Effective transmission and reception without using software to check the memory buffer address

Hint Bar

Review TRM section 23.4.4.2 for additional details

Copyright © Infineon Technologies AG 2022. All rights reserved.

Command-Response mode

- Only supports an SPI slave
- This mode has a single memory buffer, which is indexed using
  - Base read/write address
  - Current read/write address
- The command-response mode has two phases of operation
  - Write phase
  - Read phase
- The slave transmits either 0x62 (ready) or another value (busy)
- When disabled or reset, the slave transmits 0xFF (busy)

Hint Bar

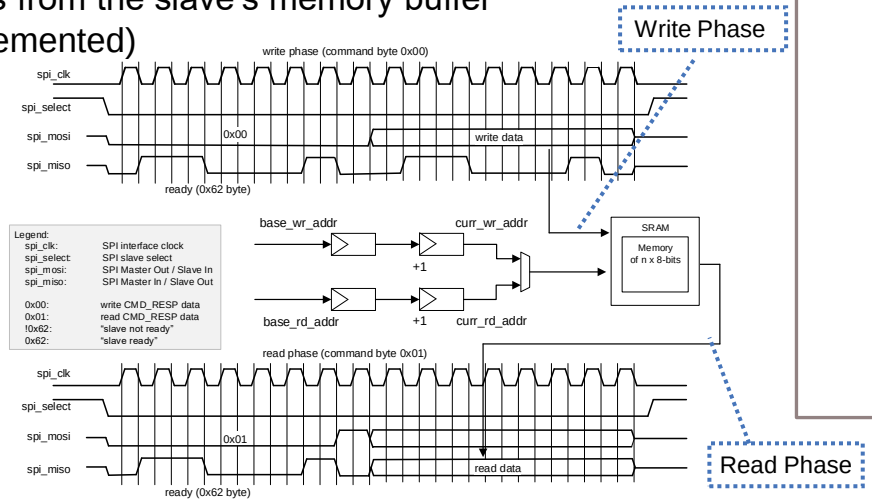
Review TRM section 23.4.4.3 for additional details

Copyright © Infineon Technologies AG 2022. All rights reserved.

Command-Response mode (Write/Read)

- The command-response write/read phases are
  - Write phase
    - Write phase begins with a selection byte (last bit set to ‘0’ indicating a write)
    - Master writes 8-bit data elements to the slave’s memory buffer (the current write address is incremented)
  - Read phase
    - Read phase begins with a selection byte (last bit set to ‘1’ indicating a read)
    - Master reads 8-bit data elements from the slave’s memory buffer (the current read address is incremented)

- Advantage
  - Effective transmission and reception without using software to check the memory buffer address



Hint Bar

Review TRM section 23.4.4.3 for additional details

Copyright © Infineon Technologies AG 2022. All rights reserved.

Interrupt events in SPI Mode

Event Type	Event
TX	SPI master transfer done
RX	SPI Bus Error
TX	TX FIFO is not full
TX	TX FIFO is empty
TX	TX FIFO overflows
TX	TX FIFO underflows
RX	RX FIFO is full
RX	RX FIFO is not empty
RX	RX FIFO overflows
RX	RX FIFO underflows

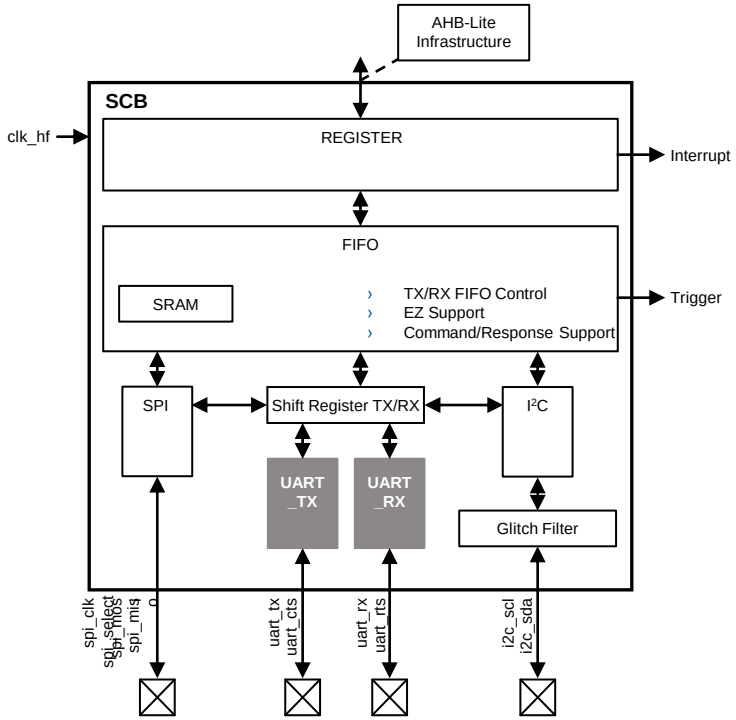
Hint Bar

Review TRM section 23.7.1 for additional details

Copyright © Infineon Technologies AG 2022. All rights reserved.

UART component serial communications block

- The UART in the SCB block supports the following:
  - Standard UART
  - Multi-processor mode
  - Start skipping function
  - Oversampling



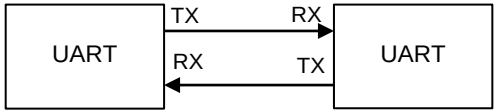
Hint Bar

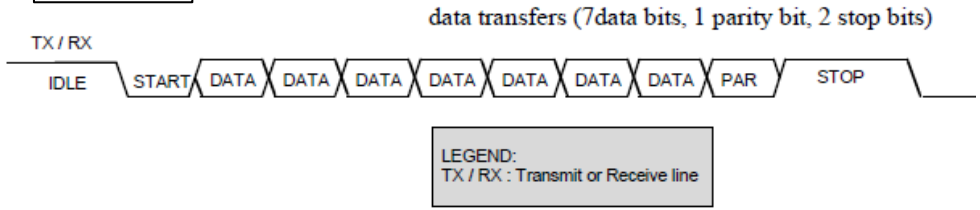
Review TRM section 23.5 for additional details

Copyright © Infineon Technologies AG 2022. All rights reserved.

Standard UART

- Standard UART
  - TX and RX connection
  - Buffer mode only uses FIFO mode
- Typical UART transfer frame components
  - Start bit
  - Data bits (programmable from 4 to16 bits)
  - Parity bit (optional)
  - Stop bit (can be set to half-bit periods between 1 and 4)





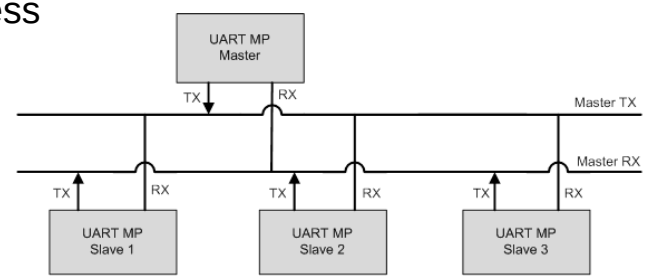
Hint Bar

Review TRM section 23.5.2 for additional details

Copyright © Infineon Technologies AG 2022. All rights reserved.

Multi-processor mode

- UART_MP (multi-processor) mode
  - Single-master-multi-slave connection
- UART_MP mode properties
  - Each slave is identified by a unique address
  - Uses a 9-bit data field
    - Ninth bit as address/data flag (MP bit)
  - Parity bit is disabled





- Advantage
  - Can communicate with multiple external ICs using the same bus

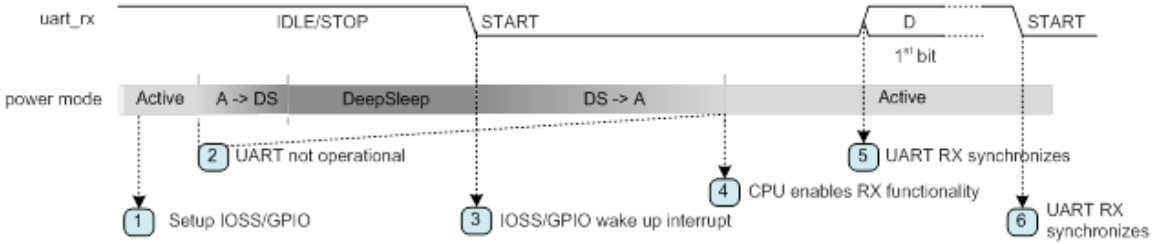
Hint Bar

Review TRM section 23.5.3.2 for additional details

Copyright © Infineon Technologies AG 2022. All rights reserved.

Start Skipping function

- Use Start Skipping to wake up from DeepSleep mode
  1. The GPIO is set as an interrupt to UART RX line transition ‘1’ to ‘0’ (START bit)
  2. The UART receiver is not functional during DeepSleep mode transition
  3. When the GPIO interrupt is activated, the system transits from DeepSleep to Active mode
  4. The CPU enables UART receive functionality
  5. The UART receiver synchronizes with the receipt data frame
  6. The UART receiver proceeds with normal operation



- Advantage
  - UART can receive data immediately upon MCU wakeup from DeepSleep

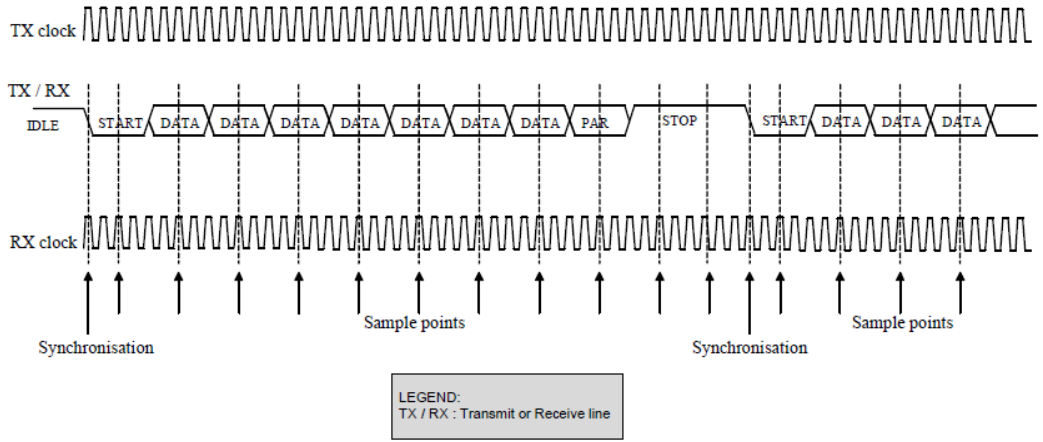
Hint Bar

Review TRM section 23.5.3.1 for additional details

Copyright © Infineon Technologies AG 2022. All rights reserved.

Oversampling

- The receiver oversamples the incoming signal
- The sampling clock can select from 8 to 16 clock cycles by software
- The value of the sample point[1] in the middle of the bit transfer period is used for transmitter and receiver clock synchronization



- Advantage
  - Can be used to synchronize the receiver with the transmitter clock

Hint Bar

Review TRM section 23.5.4 for additional details

[1] Alternatively, three samples around the middle of the bit transfer period are used for increased accuracy

Copyright © Infineon Technologies AG 2022. All rights reserved.

Interrupt

Interrupt events in UART mode

Event Type	Event
TX	TX done
RX	Frame error in received data frame
RX	Parity error in received data frame
RX	LIN baud rate detection is completed
RX	LIN break detection is successful
TX	TX FIFO is not full
TX	TX FIFO is empty
TX	TX FIFO overflows
TX	TX FIFO underflows
RX	RX FIFO is full
RX	RX FIFO is not empty
RX	RX FIFO overflows
RX	RX FIFO underflows

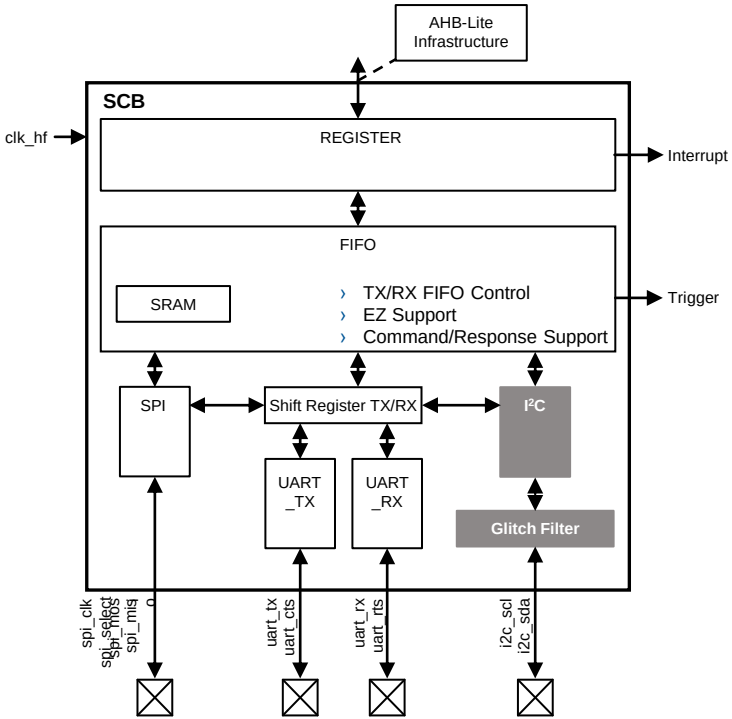
Hint Bar

Review TRM section 23.7.2 for additional details

Copyright © Infineon Technologies AG 2022. All rights reserved.

I²C component in serial communications block

- I²C supports the following:
  - Standard I²C
  - I²C Buffer mode
    - FIFO mode[1]
    - EZI2C mode
    - Command-Response mode
  - Glitch filtering



Hint Bar

Review TRM section 23.6 for additional details

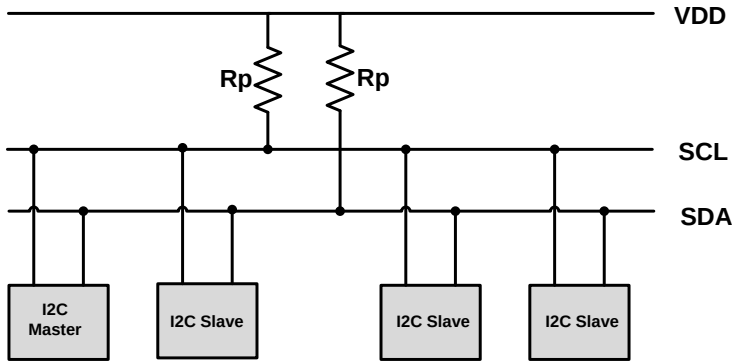
Refer to FIFO mode

[1] FIFO mode has the same features as SPI

Copyright © Infineon Technologies AG 2022. All rights reserved.

Standard I²C

- The standard I²C bus is a two-wire interface with these lines
  - Serial data (SDA)
  - Serial clock (SCL)
- I²C devices are connected to these lines through pull-up
- Masters and slaves can operate as either transmitter or receiver



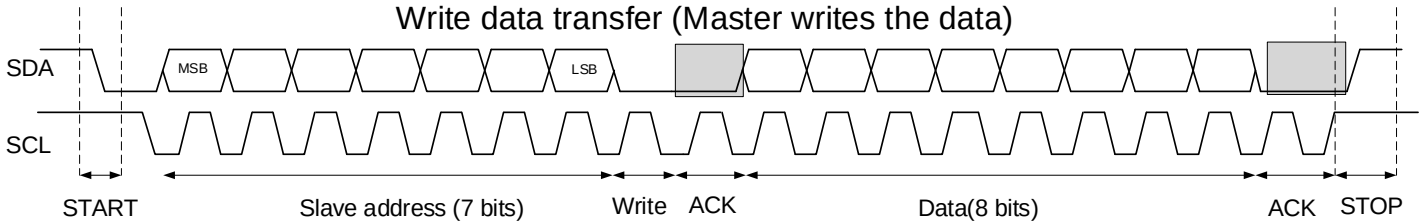
Hint Bar

Review TRM section 23.6 for additional details

Copyright © Infineon Technologies AG 2022. All rights reserved.

Write transfer

- The master sends data to the slave via a typical write transfer
  - Generate START condition
  - 7-bit I^2C slave address and a write indicator ‘0’ bit
  - The addressed slave transmits an acknowledgment
  - If the master receives an acknowledgment, it transmits data (8 bits) to the bus
  - When the transfer is complete, the master generates a STOP condition



LEGEND :
SDA: Serial Data Line
SCL: Serial Clock Line (always driven by the master)
Slave Transmit / Master Receive

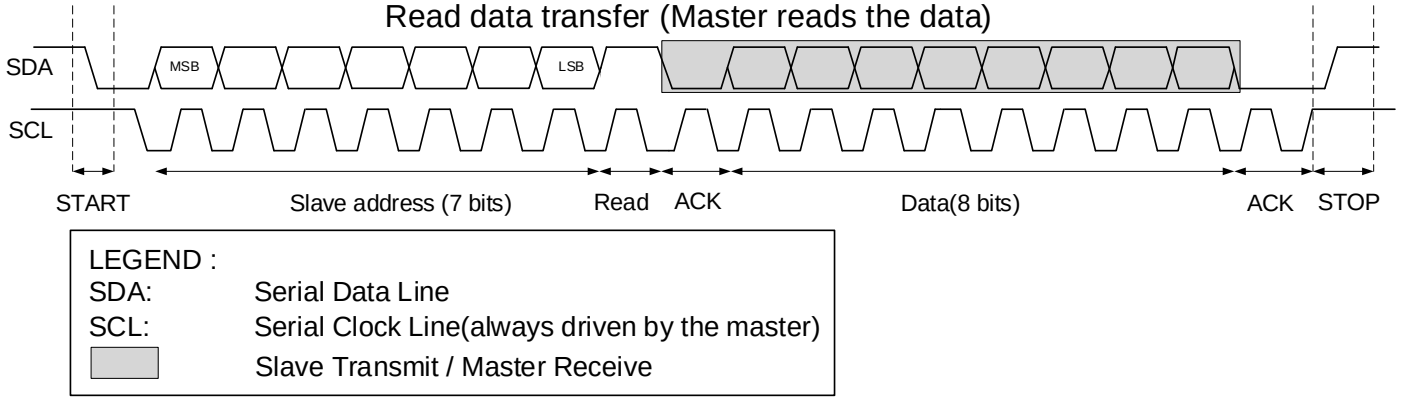
Hint Bar

Review TRM section 23.6.4.1 for additional details

Copyright © Infineon Technologies AG 2022. All rights reserved.

Read transfer

- The master receives data from the slave via a typical read transfer
  - Generate START condition
  - 7-bit I^2C slave address and a read indicator ‘1’ bit
  - If the slave acknowledges the acknowledgment signal from the master, it starts transmitting data after the acknowledgment signal
  - The master transmits an acknowledgment to confirm
  - When the transfer is complete, the master generates a STOP condition



Hint Bar

Review TRM section 23.6.4.2 for additional details

Copyright © Infineon Technologies AG 2022. All rights reserved.

I²C buffer modes (1/2)

- I²C can operate in three buffer modes[1]
  - FIFO mode[2]
  - EZI2C mode
  - Command-Response mode
- Restrictions on using buffer mode[3]
  - Master can only use FIFO mode
  - Slave can use all the three buffer modes. However, the Command-Response mode is available only on DeepSleep-capable SCB
- SCB has an internal SRAM buffer[3]
  - Capacity of 128 x 16 bits
  - Supports byte mode
  - Can be configured as 256 × 8 bits

Hint Bar

Review TRM section 23.6.5 for additional details

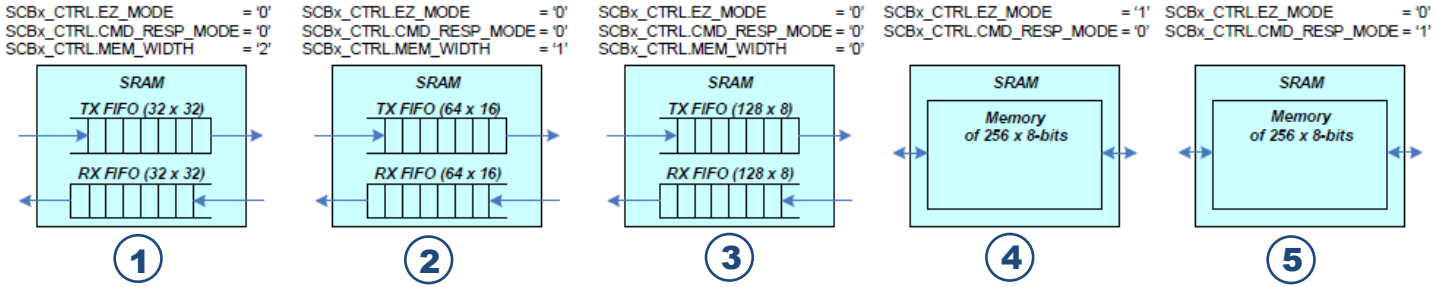
Refer to FIFO mode

[1] Refer to the Register TRM (SCBx_CTRL.MODE) for additional details
[2] FIFO mode has the same features as SPI
[3] These features are the same as the SPI Buffer mode

Copyright © Infineon Technologies AG 2022. All rights reserved.

I²C Buffer Modes (2/2)

- Buffer mode can be configured in five types in dedicated SRAM[1]
  1. Two 32 deep FIFOs for up to 32-bit data elements
  2. Two 64 deep FIFOs for up to 16-bit data elements
  3. Two 128 deep FIFOs for up to 8-bit data elements
  4. One 256 Byte EZ memory buffer
  5. One 256 Byte CMD_RESP memory buffer



Hint Bar

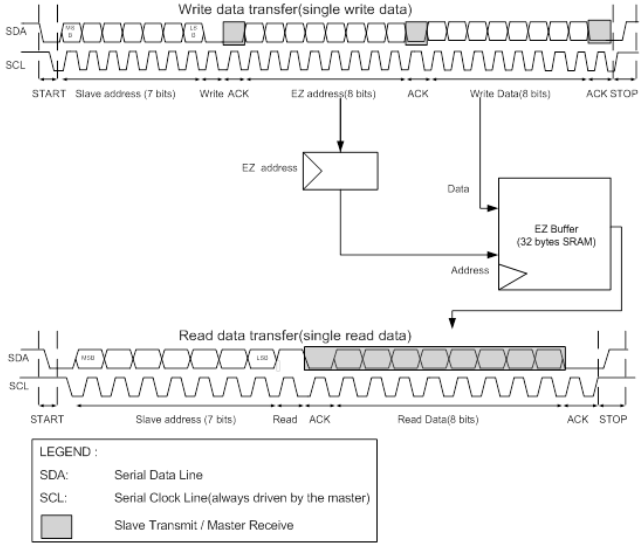
Review TRM section 23.3.1 for additional details

[1] Refer to the Register TRM (SCBx_CTRL.EZ_MODE, SCBx_CTRL.CMD_RESP_MODE, SCBx_CTRL.MEM_WIDTH) for additional details. These features are the same as the SPI Buffer mode.

Copyright © Infineon Technologies AG 2022. All rights reserved.

EZI2C mode

- EZI2C distinguishes three operation phases
  - Address phase[1]
    - The master transmits an 8-bit EZ address to the slave
  - Write phase[1]
    - The master writes 8-bit data elements to the slave's memory buffer
  - Read phase[1]
    - The master reads 8-bit data elements from the slave's memory buffer



- Advantage
  - Effective transmission and reception without using software to check the memory buffer address

Hint Bar

Review TRM section 23.6.5.2 for additional details

[1] These operations are the same as the EZ SPI mode

Copyright © Infineon Technologies AG 2022. All rights reserved.

Command-Response mode

- Only supports an I^2C slave
- This mode has a single memory buffer, which is indexed using[1]
  - Base read/write address
  - Current read/write address
- The command-response mode has two phases of operation
  - Write phase
  - Read phase

Hint Bar

Review TRM section 23.6.5.3 for additional details

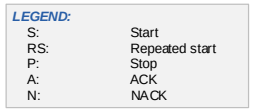
[1] The features are same as the SPI Command response mode

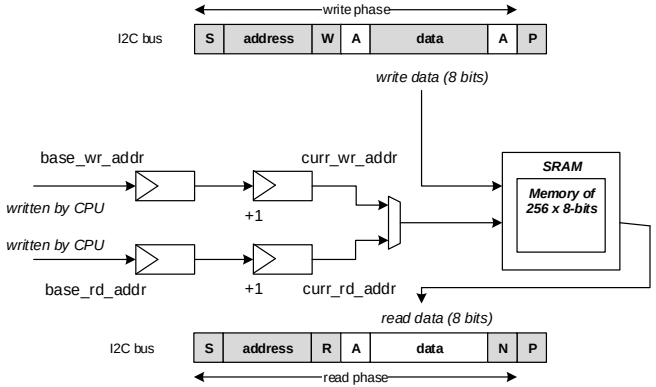
Copyright © Infineon Technologies AG 2022. All rights reserved.

Command-Response mode (Write/Read)

- The command-response write/read phases are
  - Write phase[1]
    - Begins with a START bit, slave address, and write bit (set to ‘0’ indicating a write)
    - The master writes 8-bit data elements to the slave’s memory buffer (the current write address is incremented)
  - Read phase[1]
    - Begins with a START bit, slave address, and read bit (set to ‘1’ indicating a read)
    - The master reads 8-bit data elements from the slave’s memory buffer (the current read address is incremented)

- Advantage
  - Effective transmission and reception without using software to check command





Hint Bar

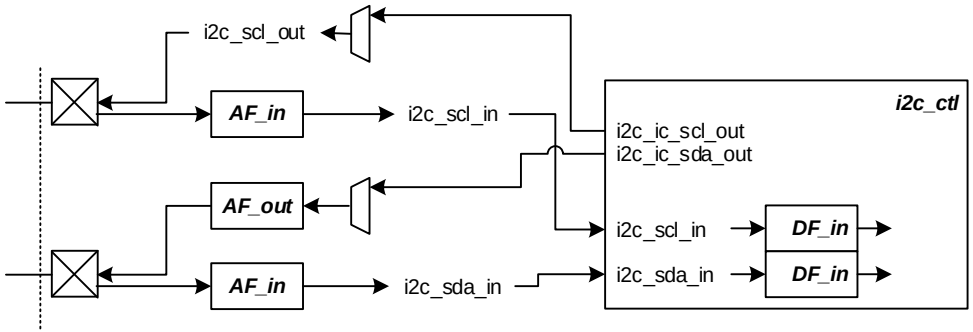
Review TRM section 23.6.5.3 for additional details

[1] The operation is the same as the SPI Command response mode

Copyright © Infineon Technologies AG 2022. All rights reserved.

Glitch filtering

- The XMC7000 SCB I²C has analog and digital glitch filters
- Analog glitch filters (AF_in, AF_out) are applied in the following:
  - i2c_scl_in
  - i2c_sda_in
  - i2c_sda_out
- Analog glitch filters are applied to filter glitches of up to 50 ns
- Digital glitch filters (DF_in) are applied in the following:
  - i2c_scl_in
  - i2c_sda_in
- Digital glitch filters are 3-tap median filters



Hint Bar

Review TRM section 23.6.6.1 for additional details

Copyright © Infineon Technologies AG 2022. All rights reserved.

Interrupt

› Interrupt events in I²C mode

Event Type	Event
TX	I²C master lost arbitration
RX	I²C master received NACK
RX	I²C master received ACK
TX	I²C master sent STOP
RX	I²C master bus error (unexpected stop/start condition detected)
TX	I²C slave lost arbitration
RX	I²C slave received NACK
RX	I²C slave received ACK
RX	I²C slave received STOP
RX	I²C slave received START

Event Type	Event
RX	I²C slave address matched
RX	I²C slave bus error (unexpected stop/start condition detected)
TX	TX FIFO is not full
TX	TX FIFO is empty
TX	TX FIFO overflows
TX	TX FIFO underflows
RX	RX FIFO is full
RX	RX FIFO is not empty
RX	RX FIFO overflows
RX	RX FIFO underflows

Hint Bar

Review TRM section 23.7.3 for additional details

Copyright © Infineon Technologies AG 2022. All rights reserved.



Part of your life. Part of tomorrow.

Revision History

Revision	ECN	Submission Date	Description of Change
**	7762132	2022-05-13	Initial release

Copyright © Infineon Technologies AG 2022. All rights reserved.