

Multichannel Serial Port Interface (McSPI)

This chapter describes the McSPI of the device.

Topic	Page
24.1 Introduction	3994
24.2 Integration	3995
24.3 Functional Description	3997
24.4 McSPI Registers	4032

24.1 Introduction

This document is intended to provide programmers with a functional presentation of the Master/Slave Multichannel Serial Port Interface (McSPI) module. It also provides a register description and a module configuration example.

McSPI is a general-purpose receive/transmit master/slave controller that can interface with up to four slave external devices or one single external master. It allows a duplex, synchronous, serial communication between a CPU and SPI compliant external devices (Slaves and Masters).

24.1.1 McSPI Features

The general features of the SPI controller are:

- Buffered receive/transmit data register per channel (1 word deep)
- Multiple SPI word access with one channel using a FIFO
- Two DMA requests per channel, one interrupt line
- Single interrupt line, for multiple interrupt source events
- Serial link interface supports:
 - Full duplex / Half duplex
 - Multi-channel master or single channel slave operations
 - Programmable 1-32 bit transmit/receive shift operations.
 - Wide selection of SPI word lengths continuous from 4 to 32 bits
- Up to four SPI channels
- SPI word Transmit / Receive slot assignment based on round robin arbitration
- SPI configuration per channel (clock definition, enable polarity and word width)
- Clock generation supports:
 - Programmable master clock generation (operating from fixed 48-MHz functional clock input)
 - Selectable clock phase and clock polarity per chip select.

24.1.2 Unsupported McSPI Features

This device supports only two chip selects per module. Module wakeup during slave mode operation is not supported, as noted in *McSPI Clock and Reset Management*.

Table 24-1. Unsupported McSPI Features

Feature	Reason
Chip selects 2 and 3	Not pinned out
Slave mode wakeup	SWAKEUP not connected
Retention during power down	Module not synthesized with retention enabled

24.2 Integration

This device includes two instantiations of McSPI: SPI0 and SPI1. The McSPI module is a general-purpose receive/transmit master/slave controller that can interface with either up to four slave external devices or one single external master. [Figure 24-1](#) shows the example of a system with multiple external slave SPI compatible devices and [Figure 24-2](#) shows the example of a system with an external master.

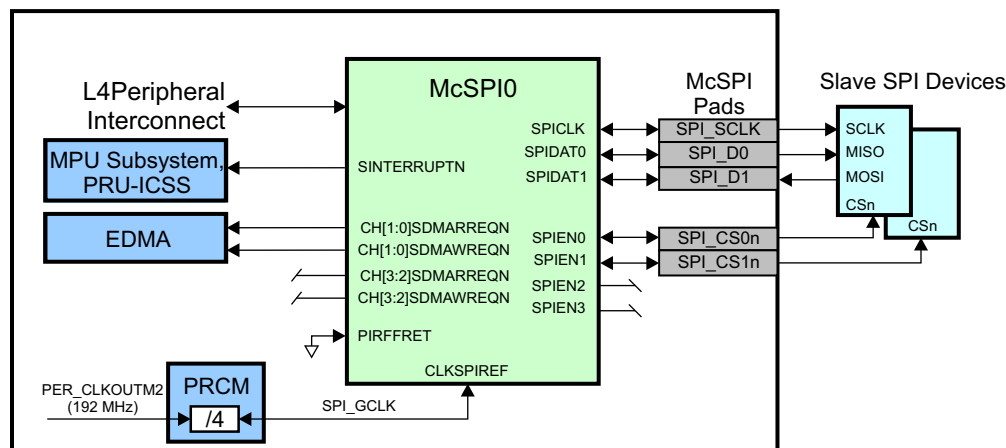


Figure 24-1. SPI Master Application

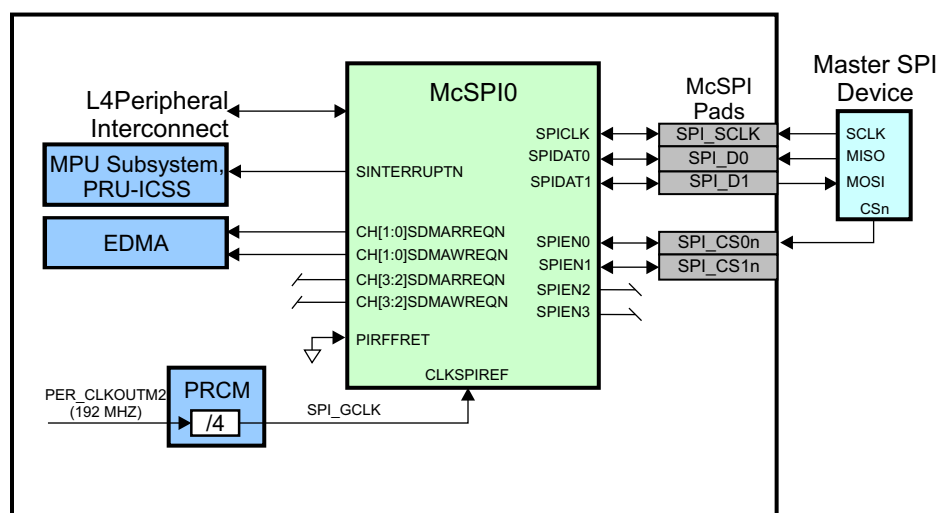


Figure 24-2. SPI Slave Application

24.2.1 McSPI Connectivity Attributes

The general connectivity attributes for the McSPI module are shown in [Table 24-2](#).

Table 24-2. McSPI Connectivity Attributes

Attributes	Type
Power Domain	Peripheral Domain
Clock Domain	PD_PER_L4LS_GCLK (Interface/OCP) PD_PER_SPI_GCLK (Func)
Reset Signals	PER_DOM_RST_N
Idle/Wakeup Signals	Smart Idle
Interrupt Requests	1 interrupt to MPU subsystem and PRU-ICSS (McSPI0INT) 1 interrupt to MPU subsystem only (McSPI1INT)
DMA Requests	4 DMA requests per instance to EDMA <ul style="list-style-type: none"> 1 RX request for CS0 (SPIREVT0) 1 TX request for CS0 (SPIXEVT0) 1 RX request for CS1 (SPIREVT1) 1 TX request for CS1 (SPIXEVT1)
Physical Address	L4 Peripheral slave port

24.2.2 McSPI Clock and Reset Management

The SPI module clocks can be woken up in two manners: by the SPI module itself using the SWAKEUP signal (refer to the module functional spec for detailed conditions), or directly from an external SPI master device by detecting an active low level on its chip select input pin (CS0n) using a GPIO attached to that device pin. Neither of these methods is supported on the device.

Table 24-3. McSPI Clock Signals

Clock Signal	Max Freq	Reference / Source	Comments
CLK Interface clock	100 MHz	CORE_CLKOUTM4 / 2	pd_per_l4ls_gclk From PRCM
CLKSPIREF Functional clock	48 MHz	PER_CLKOUTM2 / 4	pd_per_spi_gclk From PRCM

24.2.3 McSPI Pin List

The McSPI interface pins are summarized in [Table 24-4](#).

Table 24-4. McSPI Pin List

Pin	Type	Description
SPIx_SCLK	I/O ⁽¹⁾	SPI serial clock (output when master, input when slave)
SPIx_D0	I/O	Can be configured as either input or output (MOSI or MISO)
SPIx_D1	I/O	Can be configured as either input or output (MOSI or MISO)
SPIx_CS0	I/O	SPI chip select 0 output when master, input when slave (active low)
SPIx_CS1	O	SPI chip select 1 output when master, input when slave (active low)

⁽¹⁾ This output signal is also used as a re-timing input. The associated CONF_<module>_<pin>_RXACTIVE bit for the output clock must be set to 1 to enable the clock input back to the module.

24.3 Functional Description

24.3.1 SPI Transmission

This section describes the transmissions supported by McSPI. The SPI protocol is a synchronous protocol that allows a master device to initiate serial communication with a slave device. Data is exchanged between these devices. A slave select line (SPIEN) can be used to allow selection of an individual slave SPI device. Slave devices that are not selected do not interfere with SPI bus activities. Connected to multiple external devices, McSPI exchanges data with a single SPI device at a time through two main modes:

- Two data pins interface mode. (See [Section 24.3.1.1](#))
- Single data pin interface mode (recommended for half-duplex transmission). (See [Section 24.3.1.2](#))

The flexibility of McSPI allows exchanging data with several formats through programmable parameters described in [Section 24.3.1.3](#).

24.3.1.1 Two Data Pins Interface Mode

The two data pins interface mode, allows a full duplex SPI transmission where data is transmitted (shifted out serially) and received (shifted in serially) simultaneously on separate data lines SPIDAT [0] and SPIDAT [1]. Data leaving the master exits on transmit serial data line also known as MOSI: MasterOutSlaveIn. Data leaving the slave exits on the receive data line also known as MISO: MasterInSlaveOut.

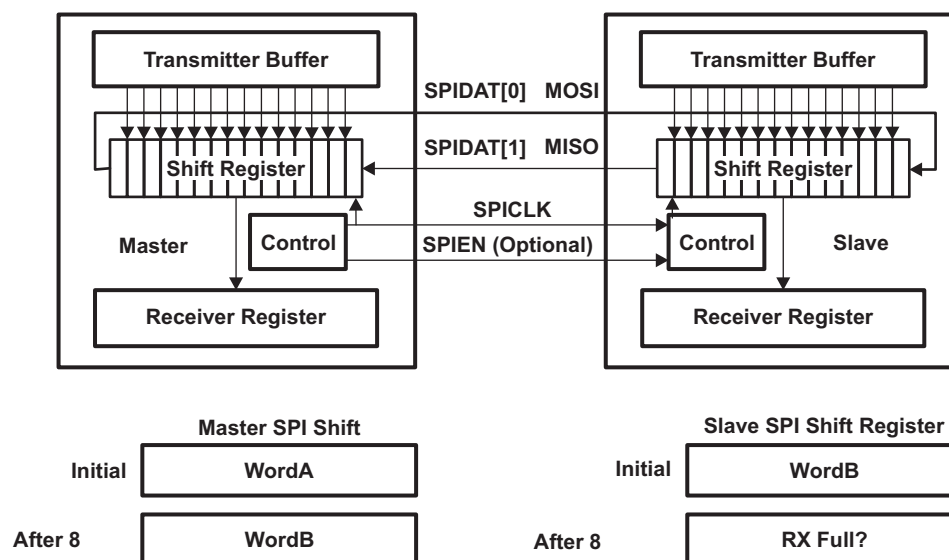
McSPI has a unified SPI port control: SPIDAT [1:0] can be independently configured as receive or transmit lines. The user has the responsibility to program which data line to use and in which direction (receive or transmit), according to the external slave/master connection.

The serial clock (SPICLK) synchronizes shifting and sampling of the information on the two serial data lines (SPIDAT [1:0]). Each time a bit is transferred out from the Master, one bit is transferred in from Slave.

Figure 24-3 shows an example of a full duplex system with a Master device on the left and a Slave device on the right. After 8 cycles of the serial clock SPICLK, the WordA has been transferred from the master to the slave. At the same time, the 8-bit WordB has been transferred from the slave to the master.

When referring to the master device, the control block transmits the clock SPICLK and the enable signal SPIEN (optional, see [Section 24.4.1.7](#), *McSPI_MODULECTRL*).

Figure 24-3. SPI Full-Duplex Transmission



24.3.1.2 Single Data Pin Interface Mode

In single data pin interface mode, under software control, a single data line is used to alternatively transmit and receive data (Half duplex transmission).

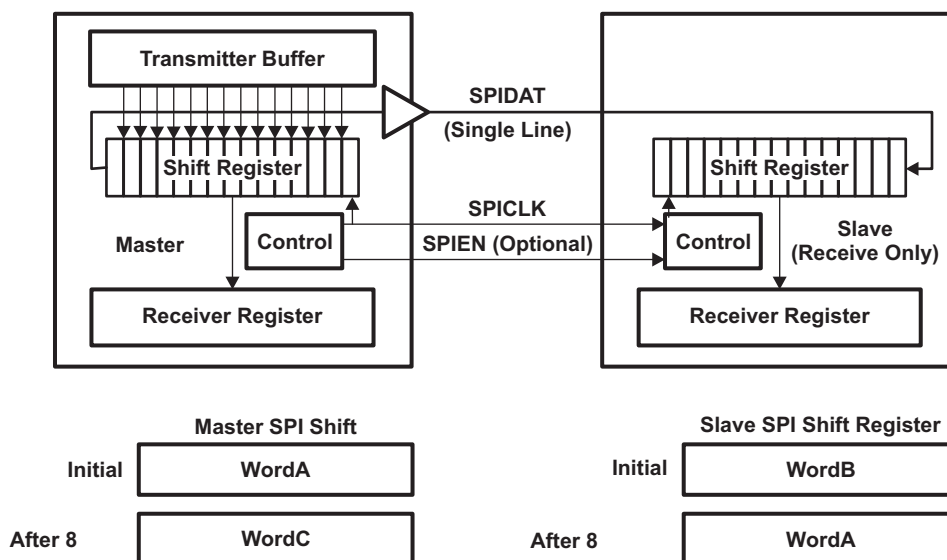
McSPI has a unified SPI port control: SPIDAT [1:0] can be independently configured as receive or transmit lines. The user has the responsibility to program which data line to use and in which direction (receive or transmit), according to the external slave/master connection.

As for a full duplex transmission, the serial clock (SPICLK) synchronizes shifting and sampling of the information on the single serial data line.

24.3.1.2.1 Example With a Receive-Only Slave

Figure 24-4 shows a half duplex system with a Master device on the left and a receive-only Slave device on the right. Each time a bit is transferred out from the Master, one bit is transferred in the Slave. After 8 cycles of the serial clock SPICLK, the 8-bit WordA has been transferred from the master to the slave.

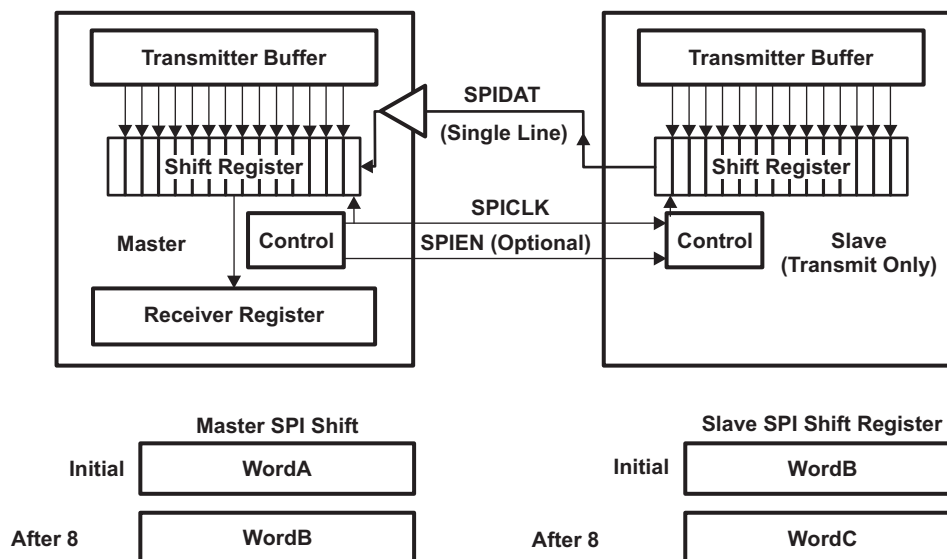
Figure 24-4. SPI Half-Duplex Transmission (Receive-only Slave)



24.3.1.2.2 Example With a Transmit-Only Slave

Figure 24-5 shows a half duplex system with a Master device on the left and a transmit-only Slave device on the right. Each time a bit is transferred out from the Slave, one bit is transferred in the Master. After 8 cycles of the serial clock SPICLK, the 8-bit WordA has been transferred from the slave to the master.

Figure 24-5. SPI Half-Duplex Transmission (Transmit-Only Slave)



24.3.1.3 Transfer Formats

This section describes the transfer formats supported by McSPI.

The flexibility of McSPI allows setting the parameters of the SPI transfer:

- SPI word length
- SPI enable generation programmable
- SPI enable assertion
- SPI enable polarity
- SPI clock frequency
- SPI clock phase
- SPI clock polarity

The consistency between SPI word length, clock phase and clock polarity of the master SPI device and the communicating slave device remains under software responsibility.

24.3.1.3.1 Programmable Word Length

McSPI supports any SPI word from 4 to 32 bits long.

The SPI word length can be changed between transmissions to allow a master device to communicate with peripheral slaves having different requirements.

24.3.1.3.2 Programmable SPI Enable Generation

McSPI is able to generate or not the SPI enable, if management of chip select is de-asserted a point to point connection is mandatory. Only a single master of slave device can be connected to the SPI bus.

24.3.1.3.3 Programmable SPI Enable (SPIEN)

The polarity of the SPIEN signals is programmable. SPIEN signals can be active high or low.

The assertion of the SPIEN signals is programmable: SPIEN signals can be manually asserted or can be automatically asserted.

Two consecutive words for two different slave devices may go along with active SPIEN signals with different polarity.

24.3.1.3.4 Programmable SPI Clock (SPICLK)

The phase and the polarity of the SPI serial clock are programmable when McSPI is a SPI master device or a SPI slave device. The baud rate of the SPI serial clock is programmable when McSPI is a SPI master.

When McSPI is operating as a slave, the serial clock SPICLK is an input from the master.

24.3.1.3.5 Bit Rate

In Master Mode, an internal reference clock CLKSPIREF is used as an input of a programmable divider to generate bit rate of the serial clock SPICLK. Granularity of this clock divider can be changed.

24.3.1.3.6 Polarity and Phase

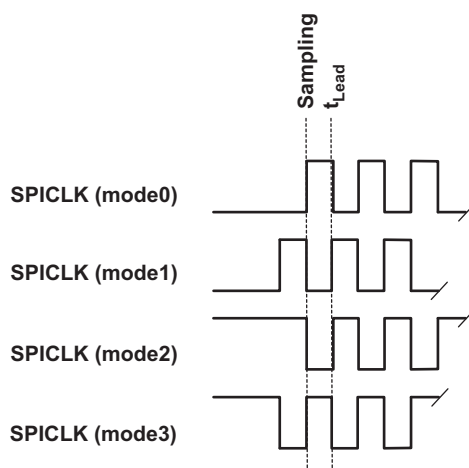
McSPI supports four sub-modes of the SPI format transfer that depend on the polarity (POL) and the phase (PHA) of the SPI serial clock (SPICLK). [Table 24-5](#) and [Figure 24-6](#) show a summary of the four sub-modes. Software selects one of four combinations of serial clock phase and polarity.

Two consecutive SPI words for two different slave devices may go along with active SPICLK signal with different phase and polarity.

Table 24-5. Phase and Polarity Combinations

Polarity (POL)	Phase (PHA)	SPI Mode	Comments
0	0	mode0	SPICLK active high and sampling occurs on the rising edge.
0	1	mode1	SPICLK active high and sampling occurs on the falling edge.
1	0	mode2	SPICLK active low and sampling occurs on the falling edge.
1	1	mode3	SPICLK active low and sampling occurs on the rising edge.

Figure 24-6. Phase and Polarity Combinations



24.3.1.3.7 Transfer Format With PHA = 0

This section describes the concept of a SPI transmission with the SPI mode0 and the SPI mode2.

In the transfer format with PHA = 0, SPIEN is activated a half cycle of SPICLK ahead of the first SPICLK edge.

In both master and slave modes, McSPI drives the data lines at the time of SPIEN is asserted.

Each data frame is transmitted starting with the MSB. At the extremity of both SPI data lines, the first bit of SPI word is valid a half-cycle of SPICLK after the SPIEN assertion.

Therefore, the first edge of the SPICLK line is used by the master to sample the first data bit sent by the slave. On the same edge, the first data bit sent by the master is sampled by the slave.

On the next SPICLK edge, the received data bit is shifted into the shift register, and a new data bit is transmitted on the serial data line.

This process continues for a total of pulses on the SPICLK line defined by the SPI word length programmed in the master device, with data being latched on odd numbered edges and shifted on even numbered edges.

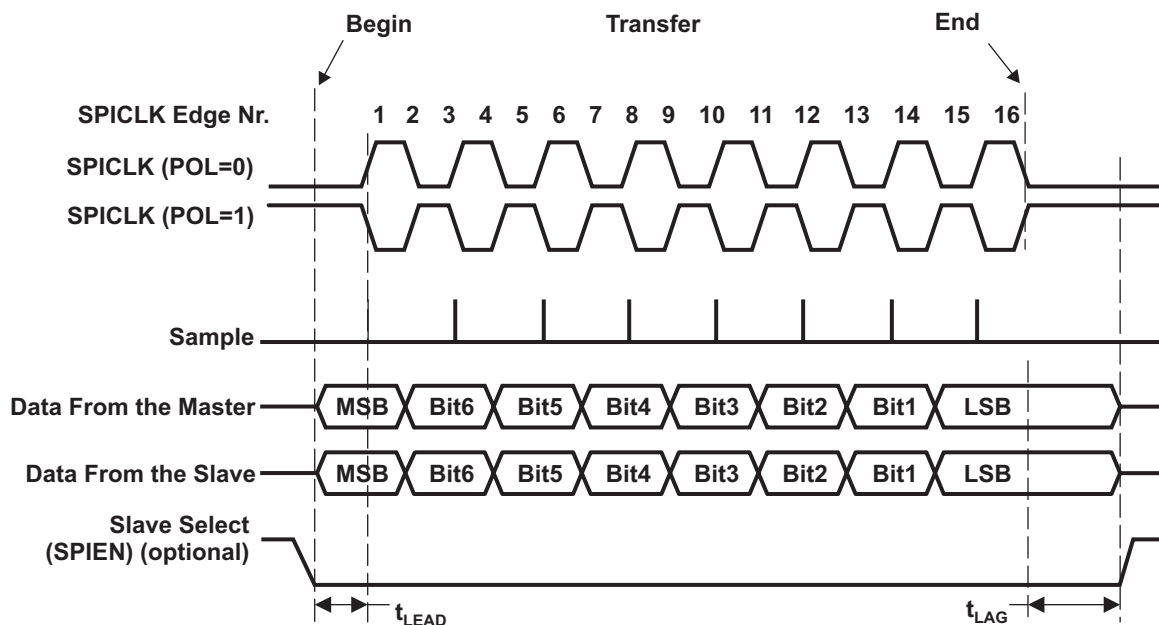
Figure 24-7 is a timing diagram of a SPI transfer for the SPI mode0 and the SPI mode2, when McSPI is master or slave, with the frequency of SPICLK equals to the frequency of CLKSPIREF. It should not be used as a replacement for SPI timing information and requirements detailed in the data manual.

When McSPI is in slave mode, if the SPIEN line is not de-asserted between successive transmissions then the content of the Transmitter register is not transmitted, instead the last received SPI word is transmitted.

In master mode, the SPIEN line must be negated and reasserted between each successive SPI word. This is because the slave select pin freezes the data in its shift register and does not allow it to be altered if PHA bit equals 0.

In 3-pin mode without using the SPIEN signal, the controller provides the same waveform but with SPIEN forced to low state. In slave mode SPIEN is useless

Figure 24-7. Full Duplex Single Transfer Format with PHA = 0



24.3.1.3.8 Transfer Format With PHA = 1

This section describes SPI full duplex transmission with the SPI mode1 and the SPI mode3.

In the transfer format with PHA = 1, SPIEN is activated a delay (t_{Lead}) ahead of the first SPICLK edge.

In both master and slave modes, McSPI drives the data lines on the first SPICLK edge.

Each data frame is transmitted starting with the MSB. At the extremity of both SPI data lines, the first bit of SPI word is valid on the next SPICLK edge, a half-cycle later of SPICLK. It is the sampling edge for both the master and slave.

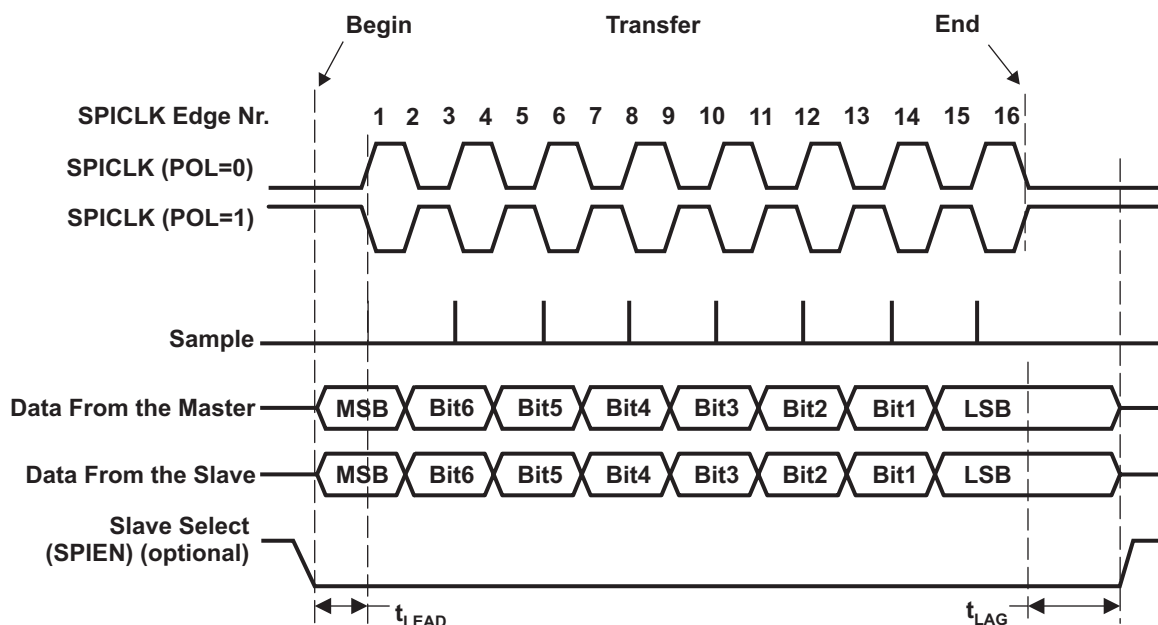
When the third edge occurs, the received data bit is shifted into the shift register. The next data bit of the master is provided to the serial input pin of the slave.

This process continues for a total of pulses on the SPICLK line defined by the word length programmed in the master device, with data being latched on even numbered edges and shifted on odd numbered edges.

Figure 24-8 is a timing diagram of a SPI transfer for the SPI mode1 and the SPI mode3, when McSPI is master or slave, with the frequency of SPICLK equals to the frequency of CLKSPIREF. It should not be used as a replacement for SPI timing information and requirements detailed in the data manual.

The SPIEN line may remain active between successive transfers. In 3-pin mode without using the SPIEN signal, the controller provides the same waveform but with SPIEN forced to low state. In slave mode SPIEN is useless.

Figure 24-8. Full Duplex Single Transfer Format With PHA = 1



24.3.2 Master Mode

McSPI is in master mode when the bit MS of the register MCSPI_MODULCTRL is cleared.

In master mode McSPI supports multi-channel communication with up to 4 independent SPI communication channel contexts. McSPI initiates a data transfer on the data lines (SPIDAT [1;0]) and generates clock (SPICLK) and control signals (SPIEN) to a single SPI slave device at a time.

24.3.2.1 Dedicated Resources Per Channel

In the following sections, the letter “I” indicates the channel number that can be 0, 1, 2 or 3. Each channel has the following dedicated resources:

- Its own channel enable, programmable with the bit EN of the register MCSPI_CH(I)CTRL. Disabling the channel, outside data word transmission, remains under user responsibility.
- Its own transmitter register MCSPI_TX on top of the common shift register. If the transmitter register is empty, the status bit TXS of the register MCSPI_CH(I)STAT is set.
- Its own receiver register MCSPI_RX on top of the common shift register. If the receiver register is full, the status bit RXS of the register MCSPI_CH(I)STAT is set.
- A fixed SPI ENABLE line allocation (SPIEN[i] port for channel “I”), SPI enable management is optional.
- Its own communication configuration with the following parameters via the register (I)CONF
 - Transmit/Receive modes, programmable with the bit TRM.
 - Interface mode (Two data pins or Single data pin) and data pins assignment, both programmable with the bits IS and DPE.
 - SPI word length, programmable with the bits WL.
 - SPIEN polarity, programmable with the bit EPOL.
 - SPIEN kept active between words, programmable with the bit FORCE.
 - Turbo mode, programmable with the bit TURBO.
 - SPICLK frequency, programmable with the bit CLKD, the granularity of clock division can be changed using CLKG bit, the clock ratio is then concatenated with MCSPI_CHCTRL[EXTCLK] value.
 - SPICLK polarity, programmable with the bit POL.
 - SPICLK phase, programmable with the bit PHA.
 - Start bit polarity, programmable with the bit SBPOL.
 - Use a FIFO Buffer or not (see the following note), programmable with FFER and FFEW, depending on transfer mode, (MCSPI_CH(I)CONF[TRM]).
- Two DMA requests events, read and write, to synchronize read/write accesses of the DMA controller with the activity of McSPI. The DMA requests are enabled with the bits DMAR and DMAW.
- Three interrupts events

Note: When more than one channel has an FIFO enable bit field (FFER or FFEW) set, the FIFO will not be used on any channel. Software must ensure that only one enabled channel is configured to use the FIFO buffer.

The transfers will use the latest loaded parameters of the register MCSPI_CH(I)CONF.

The configuration parameters SPIEN polarity, Turbo mode, SPICLK phase and SPICLK polarity can be loaded in the MCSPI_CH(I)CONF register only when the channel is disabled. The user has the responsibility to change the other parameters of the MCSPI_CH(I)CONF register when no transfer occurs on the SPI interface.

24.3.2.2 Interrupt Events in Master Mode

In master mode, the interrupt events related to the transmitter register state are TX_empty and TX_underflow. The interrupt event related to the receiver register state is RX_full.

24.3.2.2.1 TX_empty

The event TX_empty is activated when a channel is enabled and its transmitter register becomes empty (transient event). Enabling channel automatically raises this event, except for the Master receive only mode. (See [Section 24.3.2.5](#)). When the FIFO buffer is enabled (MCSPI_CH(I)CONF[FFEW] set to 1), the TX_empty is asserted as soon as there is enough space in the buffer to write a number of bytes defined by MCSPI_XFERLEVEL[AEL].

Transmitter register must be loaded to remove the source of the interrupt and the TX_empty interrupt status bit must be cleared for interrupt line de-assertion (if event enabled as interrupt source) . (See [Section 24.3.4](#)).

When FIFO is enabled, no new TX_empty event will be asserted as soon as CPU has not performed the number of write into transmitter register defined by MCSPI_XFERLEVEL[AEL]. It is the responsibility of CPU to perform the right number of writes.

24.3.2.2.2 TX_underflow

The event TX_underflow is activated when the channel is enabled and if the transmitter register or FIFO is empty (not updated with new data) at the time of shift register assignment.

The TX_underflow is a harmless warning in master mode.

To avoid having TX_underflow event at the beginning of a transmission, the event TX_underflow is not activated when no data has been loaded into the transmitter register since channel has been enabled.

To avoid having a TX_underflow event, the Transmit Register (MCSPI_TX(i)) should be loaded as infrequently as possible.

TX_underflow interrupt status bit must be cleared for interrupt line de-assertion (if event enable as interrupt source).

Note: When more than one channel has an FIFO enable bit field (FFER or FFEW) set, the FIFO will not be used on any channel. Software must ensure that only one enabled channel is configured to use the FIFO buffer.

24.3.2.2.3 RX_full

The event RX_full is activated when channel is enabled and receiver register becomes filled (transient event). When FIFO buffer is enabled (MCSPI_CH(I)CONF[FFER] set to 1), the RX_full is asserted when the number of bytes in the buffer equals the level defined by MCSPI_XFERLEVEL[AFL].

Receiver register must be read to remove source of interrupt and RX_full interrupt status bit must be cleared for interrupt line de-assertion (if event enabled as interrupt source).

When the FIFO is enabled, no new RX_FULL event will be asserted once the CPU has read the number of bytes defined by MCSPI_XFERLEVEL[AFL]. It is the responsibility of the CPU to perform the correct number of read operations.

24.3.2.2.4 End of Word Count

The event end of word (EOW) count is activated when channel is enabled and configured to use the built-in FIFO. This interrupt is raised when the controller had performed the number of transfer defined in MCSPI_XFERLEVEL[WCNT] register. If the value was programmed to 0000h, the counter is not enabled and this interrupt is not generated.

The EOW count interrupt also indicates that the SPI transfer has halted on the channel using the FIFO buffer.

The EOW interrupt status bit must be cleared for interrupt line de-assertion (if event enable as interrupt source).

24.3.2.3 Master Transmit and Receive Mode

This mode is programmable per channel (bit TRM of register MCSPI_CH(I)CONF).

The channel access to the shift registers, for transmission/reception, is based on its transmitter and receiver register state and round robin arbitration.

The channel that meets the rules below is included in the round robin list of already active channels scheduled for transmission and/or reception. The arbiter skips the channel that does not meet the rules and search for the next following enabled channel, in rotation.

Rule 1: Only enabled channels (bit EN of the register MCSPI_CH(I)CTRL), can be scheduled for transmission and/or reception.

Rule 2: An enabled channel can be scheduled if its transmitter register is not empty (bit TXS of the register MCSPI_CH(I)STAT) or its FIFO is not empty when the buffer is used for the corresponding channel (bit FFE of the register MCSPI_CH(I)STAT) at the time of shift register assignment. If the transmitter register or FIFO is empty, at the time of shift register assignment, the event TX_underflow is activated and the next enabled channel with new data to transmit is scheduled. (See also transmit only mode).

Rule 3: An enabled channel can be scheduled if its receive register is not full (bit RXS of the register MCSPI_CH(I)STAT) or its FIFO is not full when the buffer is used for the corresponding channel (bit FFF of the register MCSPI_CH(I)STAT) at the time of shift register assignment. (See also receive only mode). Therefore the receiver register or FIFO cannot be overwritten. The RX_overflow bit, in the MCSPI_IRQSTATUS register is never set in this mode.

On completion of SPI word transfer (bit EOT of the register MCSPI_CH(I)STAT is set) the updated transmitter register for the next scheduled channel is loaded into the shift register. This bit is meaningless when using the Buffer for this channel. The serialization (transmit and receive) starts according to the channel communication configuration. On serialization completion the received data is transferred to the channel receive register.

The built-in FIFO is available in this mode and if configured in one data direction, transmit or receive, then the FIFO is seen as a unique 64-byte buffer. If configured in both data directions, transmit and receive, then the FIFO is split into two separate 32-byte buffers with their own address space management. In this last case, the definition of AEL and AFL levels is based on 32 bytes and is under CPU responsibility.

24.3.2.4 Master Transmit-Only Mode

This mode eliminates the need for the CPU to read the receiver register (minimizing data movement) when only transmission is meaningful.

The master transmit only mode is programmable per channel (bits TRM of the register MCSPI_CH(I)CONF).

In master transmit only mode, transmission starts after data is loaded into the transmitter register.

Rule 1 and **Rule 2**, defined above, are applicable in this mode.

Rule 3, defined above, is not applicable: In master transmit only mode, the receiver register or FIFO state "full" does not prevent transmission, and the receiver register is always overwritten with the new SPI word. This event in the receiver register is not significant when only transmission is meaningful. So, the RX_overflow bit, in the MCSPI_IRQSTATUS register is never set in this mode.

The McSPI module automatically disables the RX_full interrupt status. The corresponding interrupt request and DMA Read request are not generated in master transmit only mode.

The status of the serialization completion is given by the bit EOT of the register MCSPI_CH(I)STAT. This bit is meaningless when using the Buffer for this channel.

The built-in FIFO is available in this mode and can be configured with FFEW bit field in the MCSPI_CH(I)CONF register, then the FIFO is seen as a unique 64-byte buffer.

24.3.2.5 Master Receive-Only Mode

This mode eliminates the need for the CPU to refill the transmitter register (minimizing data movement) when only reception is meaningful.

The master receive mode is programmable per channel (bits TRM of the register MCSPI_CH(I)CONF).

The master receive only mode enables channel scheduling only on empty state of the receiver register.

Rule 1 and **Rule 3**, defined above, are applicable in this mode.

Rule 2, defined above, is not applicable: In master receive only mode, after the first loading of the transmitter register of the enabled channel, the transmitter register state is maintained as full. The content of the transmitter register is always loaded into the shift register, at the time of shift register assignment. So, after the first loading of the transmitter register, the bits TX_empty and TX_underflow, in the MCSPI_IRQSTATUS register are never set in this mode.

The status of the serialization completion is given by the bit EOT of the register MCSPI_CH(I)STAT. The bit RX_full in the MCSPI_IRQSTATUS register is set when a received data is loaded from the shift register to the receiver register. This bit is meaningless when using the Buffer for this channel.

The built-in FIFO is available in this mode and can be configured with FFER bit field in the MCSPI_CH(I)CONF register, then the FIFO is seen as a unique 64-byte buffer.

24.3.2.6 Single-Channel Master Mode

When the SPI is configured as a master device with a single enabled channel, the assertion of the SPIM_CSX signal can be controlled in two different ways:

- In 3 pin mode : MCSPI_MODULCTRL[1] PIN34 and MCSPI_MODULCTRL[0] SINGLE bit are set to 1, the controller transmit SPI word as soon as transmit register or FIFO is not empty.
- In 4 pin mode : MCSPI_MODULCTRL[1] PIN34 bit is cleared to 0 and MCSPI_MODULCTRL[0] SINGLE bit is set to 1, SPIEN assertion/deassertion controlled by Software. (See [Section 24.3.2.6.1](#)) using the MCSPI_CHxCONF[20] FORCE bit.

24.3.2.6.1 Programming Tips When Switching to Another Channel

When a single channel is enabled and data transfer is ongoing:

- Wait for completion of the SPI word transfer (bit EOT of the register MCSPI_CH(I)STAT is set) before disabling the current channel and enabling a different channel.
- Disable the current channel first, and then enable the other channel.

24.3.2.6.2 Keep SPIEN Active Mode (Force SPIEN)

Continuous transfers are manually allowed by keeping the SPIEN signal active for successive SPI words transfer. Several sequences (configuration/enable/disable of the channel) can be run without deactivating the SPIEN line. This mode is supported by all channels and any master sequence can be used (transmit-receive, transmit-only, receive-only).

Keeping the SPIEN active mode is supported when:

- A single channel is used (bit MCSPI_MODULCTRL[Single] is set to 1).
- Transfer parameters of the transfer are loaded in the configuration register (MCSPI_CH(I)CONF) in the appropriate channel.

The state of the SPIEN signal is programmable.

- Writing 1 into the bit FORCE of the register MCSPI_CH(I)CONF drives high the SPIEN line when MCSPI_CHCONF(I)[EPOL] is set to zero, and drives it low when MCSPI_CHCONF(I)[EPOL] is set.
- Writing 0 into the bit FORCE of the register MCSPI_CH(I)CONF drives low the SPIEN line when MCSPI_CHCONF(I)[EPOL] is set to zero, and drives it high when MCSPI_CHCONF(I)[EPOL] is set.
- A single channel is enabled (MCSPI_CH(I)CTRL[En] set to 1) . The first enabled channel activates the SPIEN line.

Once the channel is enabled, the SPIEN signal is activated with the programmed polarity.

As in multi-channel master mode, the start of the transfer depends on the status of the transmitter register, the status of the receiver register and the mode defined by the bits TRM in the configuration register (transmit only, receive only or transmit and receive) of the enabled channel.

The status of the serialization completion of each SPI word is given by the bit EOT of the register MCSPI_CH(I)STAT. The bit RX_full in the MCSPI_IRQSTATUS register is set when a received data is loaded from the shift register to the receiver register.

A change in the configuration parameters is propagated directly on the SPI interface. If the SPIEN signal is activated the user must insure that the configuration is changed only between SPI words, in order to avoid corrupting the current transfer.

NOTE: The SPIEN polarity, the SPICLK phase and SPICLK polarity must not be modified when the SPIEN signal is activated. The Transmit/Receive mode, programmable with the bit TRM can be modified only when the channel is disabled. The channel can be disabled and enabled while the SPIEN signal is activated.

The delay between SPI words that requires the connected SPI slave device to switch from one configuration (Transmit only for instance) to another (receive only for instance) must be handled under software responsibility.

At the end of the last SPI word, the channel must be deactivated (MCSPI_CH(I)CTRL[En] is cleared to 0) and the SPIEN can be forced to its inactive state (MCSPI_CH(I)CONF[Force]).

Figure 24-9 and Figure 24-10 show successive transfers with SPIEN kept active low with a different configuration for each SPI word in respectively single data pin interface mode and two data pins interface mode. The arrows indicate when the channel is disabled before a change in the configuration parameters and enabled again.

Figure 24-9. Continuous Transfers With SPIEN Maintained Active (Single-Data-Pin Interface Mode)

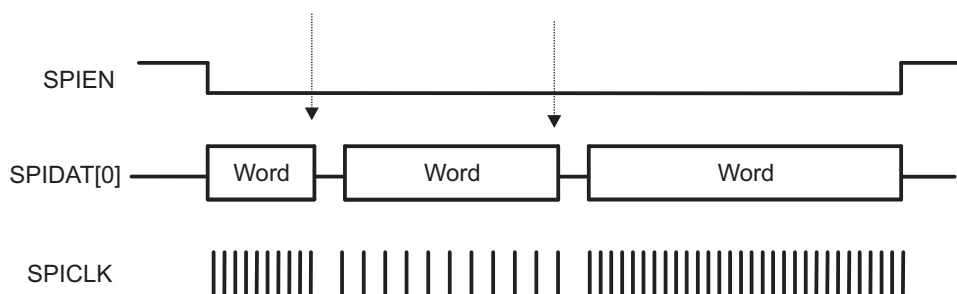
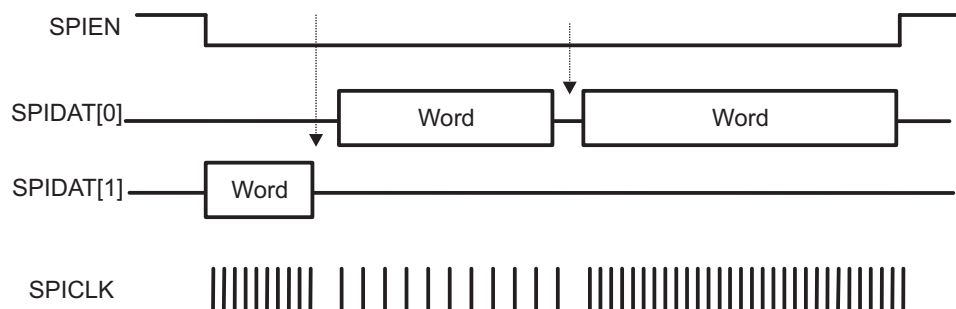


Figure 24-10. Continuous Transfers With SPIEN Maintained Active (Dual-Data-Pin Interface Mode)



NOTE: The turbo mode is also supported for the Keep SPIEN active mode when the following conditions are met:

- A single channel will be explicitly used (bit MCSPI_MODULCTRL[Single] is set to 1).
- The turbo mode is enabled in the configuration of the channel (bit Turbo of the register (i)CONF).

24.3.2.6.3 Turbo Mode

The purpose of the Turbo mode is to improve the throughput of the SPI interface when a single channel is enabled, by allowing transfers until the shift register and the receiver register are full.

This mode is programmable per channel (bit Turbo of the register (I)CONF). When several channels are enabled, the bit Turbo of the registers MCSPI_CH(I)CONF has no effect, and the channel access to the shift registers remains as described in [Section 24.3.2.3](#).

In Turbo mode, **Rule 1** and **Rule 2** defined in [Section 24.3.2.3](#) are applicable but Rule 3 is not applicable. An enabled channel can be scheduled if its receive register is full (bit RXS of the register MCSPI_CH(I)STAT) at the time of shift register assignment until the shift register is full.

In Turbo mode, **Rule 1** and **Rule 2** defined in [Section 24.3.2.3](#) are applicable but Rule 3 is not applicable. An enabled channel can be scheduled if its receive register is full (bit RXS of the register MCSPI_CH(I)STAT) at the time of shift register assignment until the shift register is full.

The receiver register cannot be overwritten in turbo mode. In consequence the RX_overflow bit, in MCSPI_IRQSTATUS register is never set in this mode.

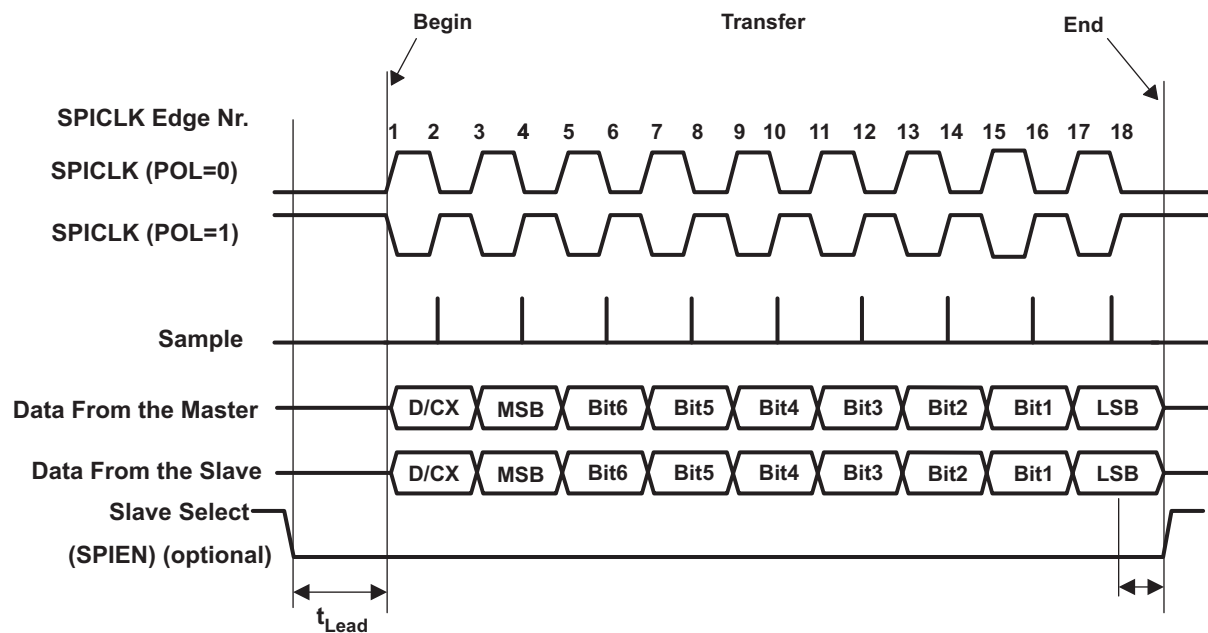
24.3.2.7 Start Bit Mode

The purpose of the start bit mode is to add an extended bit before the SPI word transmission specified by word length WL. This feature is only available in master mode.

This mode is programmable per channel (bit Start bit enable SBE of the register MCSPI_CH(I)CONF).

The polarity of the extended bit is programmable per channel and it indicates whether the next SPI word must be handled as a command when SBPOL is cleared to 0 or as a data or a parameter when SBPOL is set to 1. Moreover start bit polarity SBPOL can be changed dynamically during start bit mode transfer without disabling the channel for reconfiguration, in this case you have the responsibility to configure the SBPOL bit before writing the SPI word to be transmitted in TX register.

The start bit mode could be used at the same time as turbo mode and/or manual chip select mode. In this case only one channel could be used, no round-robin arbitration is possible.

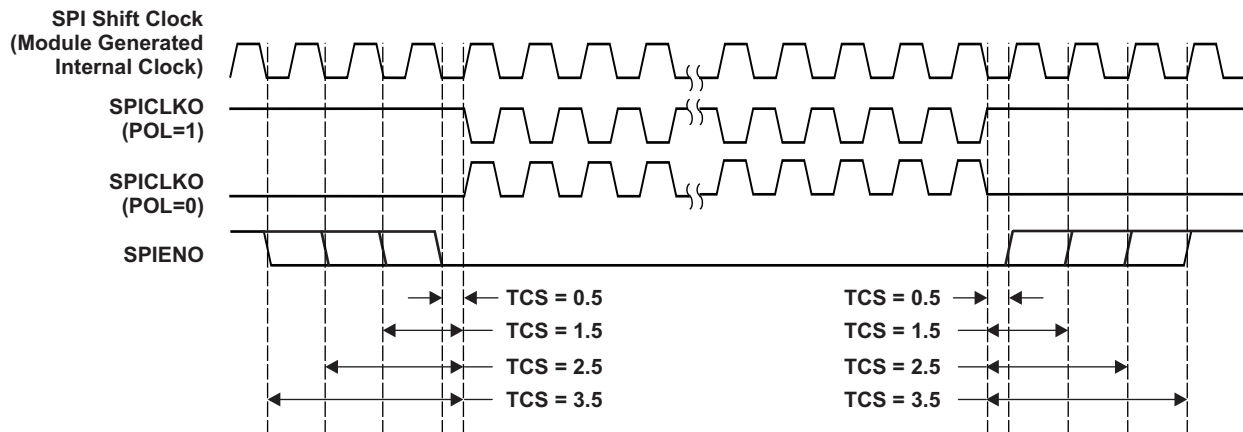
Figure 24-11. Extended SPI Transfer With Start Bit PHA = 1


24.3.2.8 Chip-Select Timing Control

The chip select timing control is only available in master mode with automatic chip select generation (FORCE bit field is cleared to 0), to add a programmable delay between chip select assertion and first clock edge or chip select removal and last clock edge. The option is available only in 4 pin mode MCSPI_MODULCTRL[1] PIN34 is cleared to 0.

This mode is programmable per channel (bit TCS of the register MCSPI_CH(I)CONF). [Figure 24-12](#) shows the chip-select SPIEN timing control.

Figure 24-12. Chip-Select SPIEN Timing Controls



NOTE: Because of the design implementation for transfers using a clock divider ratio set to 1 (clock bypassed), a half cycle must be added to the value between chip-select assertion and the first clock edge with PHA = 1 or between chip-select removal and the last clock edge with PHA = 0.

With an odd clock divider ratio which occurs when granularity is one clock cycle, that means that MCSPI_CH(I)CONF[CLKG] is set to 1 and MCSPI_CH(I)CONF[CLKD] has an even value, the clock duty cycle is not 50%, then one of the high level or low level duration is selected to be added to TCS delay.

[Table 24-6](#) summarizes all delays between chip select and first (setup) or last (hold) clock edge.

In 3-pin mode this option is useless, the chip select SPIEN is forced to low state.

Table 24-6. Chip Select ↔ Clock Edge Delay Depending on Configuration

Clock Ratio F_{ratio}	Clock Phase PHA	Chip Select ↔ Clock Edge Delay	
		Setup	Hold
1	0	$T_{ref} \times (TCS + \frac{1}{2})$	$T_{ref} \times (TCS + 1)$
	1	$T_{ref} \times (TCS + 1)$	$T_{ref} \times (TCS + \frac{1}{2})$
Even ≥ 2	x	$T_{ref} \times F_{ratio} \times (TCS + \frac{1}{2})$	$T_{ref} \times F_{ratio} \times (TCS + \frac{1}{2})$
Odd ≥ 3 (only with MCSPI_CH(I)CONF[CLKG] set to 1)	0	$T_{ref} \times [(F_{ratio} \times TCS) + (F_{ratio} + \frac{1}{2})]$	$T_{ref} \times [(F_{ratio} \times TCS) + (F_{ratio} + \frac{1}{2})]$
	1	$T_{ref} \times [(F_{ratio} \times TCS) + (F_{ratio} - \frac{1}{2})]$	$T_{ref} \times [(F_{ratio} \times TCS) + (F_{ratio} - \frac{1}{2})]$

T_{ref} = CLKSPIREF period in ns. F_{ratio} = SPI clock division ratio

The clock divider ratio depends on divider granularity MCSPI_CH(I)CONF[CLKG]:

- MCSPI_CH(I)CONF[CLKG] = 0 : granularity is power of two.
 $F_{ratio} = 2^{MCSPI_CH(I)CONF[CLKD]}$
- MCSPI_CH(I)CONF[CLKG] = 1 : granularity is power of two.

$$F_{\text{ratio}} = \text{MCSPI_CH(I)CNTRL[EXTCLK].MCSPI_CH(I)CONF[CLKD]} + 1$$

24.3.2.9 Clock Ratio Granularity

By default the clock division ratio is defined by the register MCSPI_CH(I)CONF[CLKD] with power of two granularity leading to a clock division in range 1 to 32768, in this case the duty cycle is always 50%. With bit MCSPI_CH(I)CONF[CLKG] the clock division granularity can be changed to one clock cycle, in that case the register MCSPI_CH(I)CTRL[EXTCLK] is concatenated with MCSPI_CH(I)CONF[CLKD] to give a 12-bit width division ratio in range 1 to 4096.

When granularity is one clock cycle (MCSPI_CH(I)CONF[CLKG] set to 1), for odd value of clock ratio the clock duty cycle is kept to 50-50 using falling edge of clock reference CLKSPIREF.

Table 24-7. CLKSPIO High/Low Time Computation

Clock Ratio F_{ratio}	CLKSPIO High Time	CLKSPIO Low Time
1	T_{high_ref}	T_{low_ref}
Even ≥ 2	$t_{ref} \times (F_{ratio}/2)$	$t_{ref} \times (F_{ratio}/2)$
Odd ≥ 3	$t_{ref} \times (F_{ratio}/2)$	$t_{ref} \times (F_{ratio}/2)$

T_{ref} = CLKSPIREF period in ns. T_{high_ref} = CLKSPIREF high Time period in ns. T_{low_ref} = CLKSPIREF low Time period in ns. F_{ratio} = SPI clock division ratio

$F_{ratio} = \text{MCSPI_CH(I)CTRL[EXTCLK].MCSPI_CH(I)CONF[CLKD]} + 1$

For odd ratio value the duty cycle is calculated as below:

$\text{Duty_cycle} = \frac{1}{2}$

Granularity examples: With a clock source frequency of 48 MHz:

Table 24-8. Clock Granularity Examples

MCSPI_CH (I)CTRL	MCSPI_CH (I)CONF	MCSPI_CH (I)CONF	F_{ratio}	MCSPI_CH (I)CONF	MCSPI_CH (I)CONF	Thigh (ns)	Tlow (ns)	Tperiod (ns)	Duty Cycle	Fout (MHz)
EXTCLK	CLKD	CLKG		PHA	POL					
X	0	0	1	X	X	10.4	10.4	20.8	50-50	48
X	1	0	2	X	X	20.8	20.8	41.6	50-50	24
X	2	0	4	X	X	41.6	41.6	83.2	50-50	12
X	3	0	8	X	X	83.2	83.2	166.4	50-50	6
0	0	1	1	X	X	10.4	10.4	20.8	50-50	48
0	1	1	2	X	X	20.8	20.8	41.6	50-50	24
0	2	1	3	1	0	31.2	31.2	62.4	50-50	16
0	2	1	3	1	1	31.2	31.2	62.4	50-50	16
0	3	1	4	X	X	41.6	41.6	83.2	50-50	12
5	0	1	81	1	0	842.4	842.4	1684.8	50-50	0.592
5	7	1	88	X	X	915.2	915.2	1830.4	50-50	0.545

24.3.2.10 FIFO Buffer Management

The McSPI controller has a built-in 64-byte buffer in order to unload DMA or interrupt handler and improve data throughput.

This buffer can be used by only one channel and is selected by setting MCSPI_CH(I)CONF[FFER] and/or MCSPI_CH(I)CONF[FFEW] to 1.

If several channels are selected and several FIFO enable bit fields set to 1, the controller forces the buffer to be disabled for all channels. It is the responsibility of the driver to enable the buffer for only one channel.

The buffer can be used in the modes defined below:

- Master or Slave mode.
- Transmit only, Receive only or Transmit/Receive mode.
- Single channel or turbo mode, or in normal round robin mode. In round robin mode the buffer is used by only one channel.
- All word length MCSPI_CH(I)CONF[WL] are supported.

Two levels AEL and AFL located in MCSPI_XFERLEVEL register rule the buffer management. The granularity of these levels is one byte, then it is not aligned with SPI word length. It is the responsibility of the driver to set these values as a multiple of SPI word length defined in MCSPI_CH(I)CONF[WL]. The number of byte written in the FIFO depends on word length (see [Table 24-9](#)).

Table 24-9. FIFO Writes, Word Length Relationship

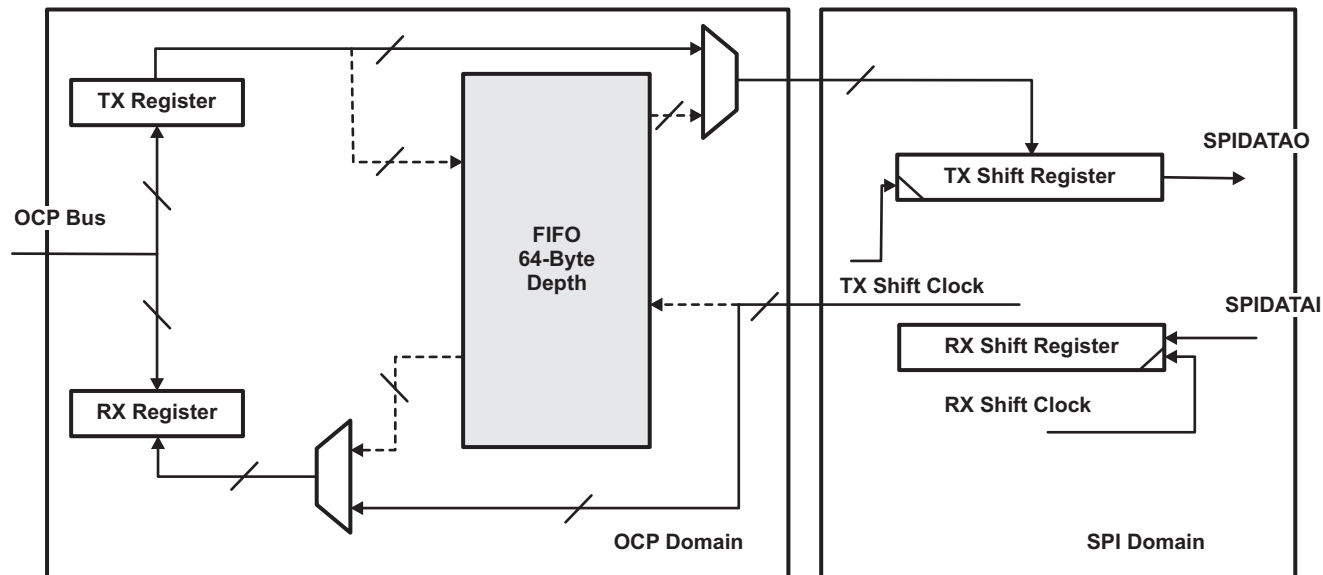
	SPI Word Length WL		
	$3 \leq WL \leq 7$	$8 \leq WL \leq 15$	$16 \leq WL \leq 31$
Number of byte written in the FIFO	1 byte	2 bytes	4 byte

24.3.2.10.1 Split FIFO

The FIFO can be split into two part when module is configured in transmit/receive mode MCSPI_CH(I)CONF[TRM] is cleared to 0 and MCSPI_CH(I)CONF[FFER] and MCSPI_CH(I)CONF[FFEW] asserted. Then system can access a 32-byte depth FIFO per direction.

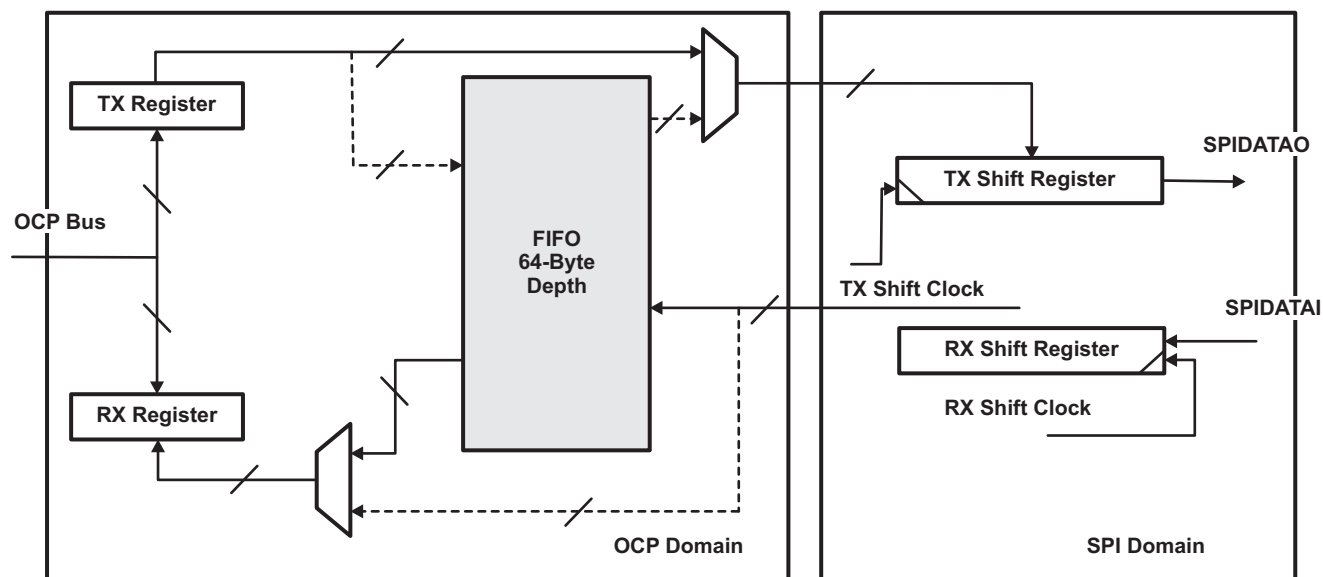
The FIFO buffer pointers are reset when the corresponding channel is enabled or FIFO configuration changes.

Figure 24-13. Transmit/Receive Mode With No FIFO Used



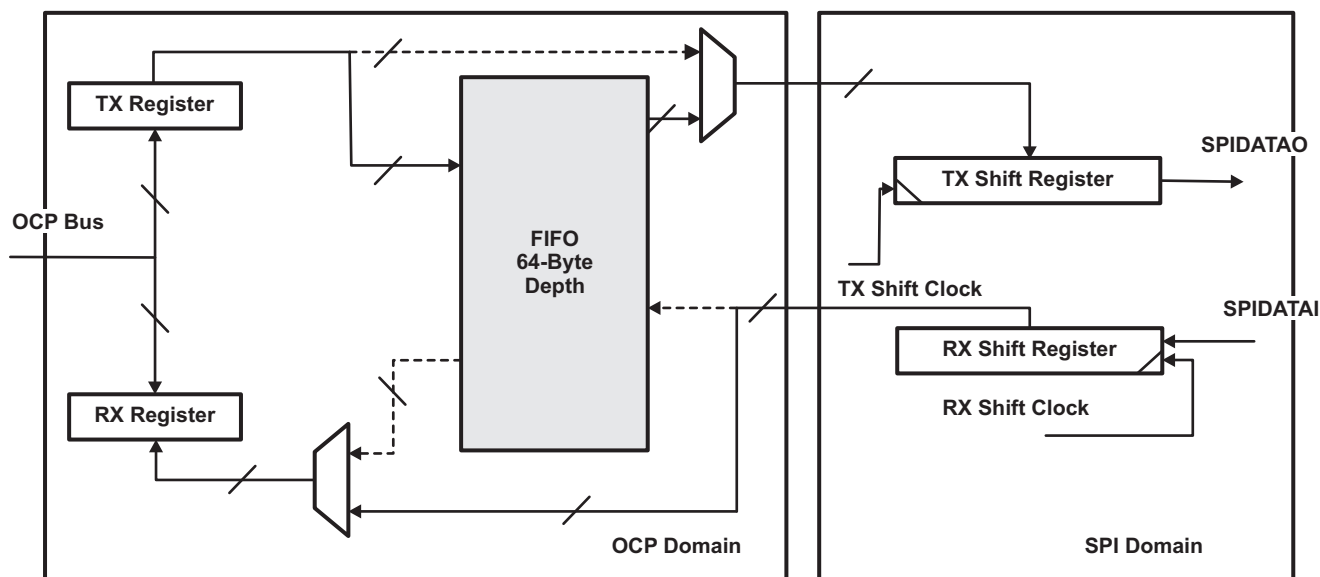
Configuration:
 MCSPI_CH(i)CONF[TRM]=0x0 Transmit/receive mode
 MCSPI_CH(i)CONF[FFRE]=0x0 FIFO disabled on receive path
 MCSPI_CH(i)CONF[FFWE]=0x0 FIFO disabled on transmit path

Figure 24-14. Transmit/Receive Mode With Only Receive FIFO Enabled



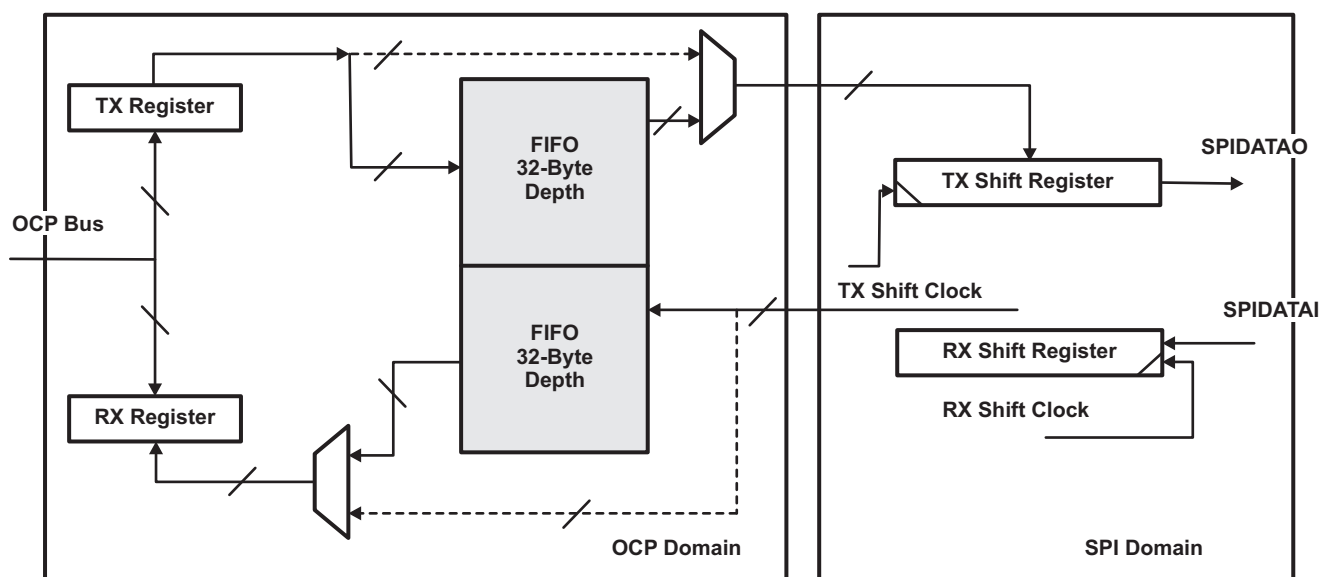
Configuration:
 MCSPI_CH(i)CONF[TRM]=0x0 Transmit/receive mode
 MCSPI_CH(i)CONF[FFRE]=0x1 FIFO enabled on receive path
 MCSPI_CH(i)CONF[FFWE]=0x0 FIFO disabled on transmit path

Figure 24-15. Transmit/Receive Mode With Only Transmit FIFO Used



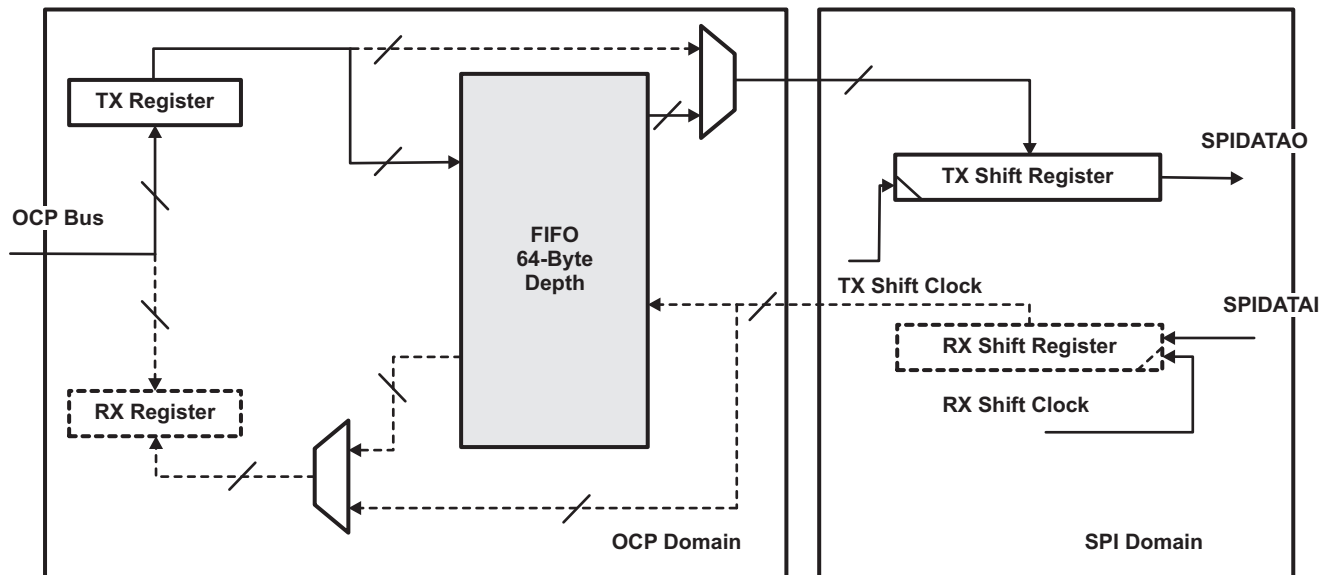
Configuration:
 MCSPI_CH(i)CONF[TRM]=0x0 Transmit/receive mode
 MCSPI_CH(i)CONF[FFRE]=0x0 FIFO disabled on receive path
 MCSPI_CH(i)CONF[FFWE]=0x1 FIFO enabled on transmit path

Figure 24-16. Transmit/Receive Mode With Both FIFO Direction Used



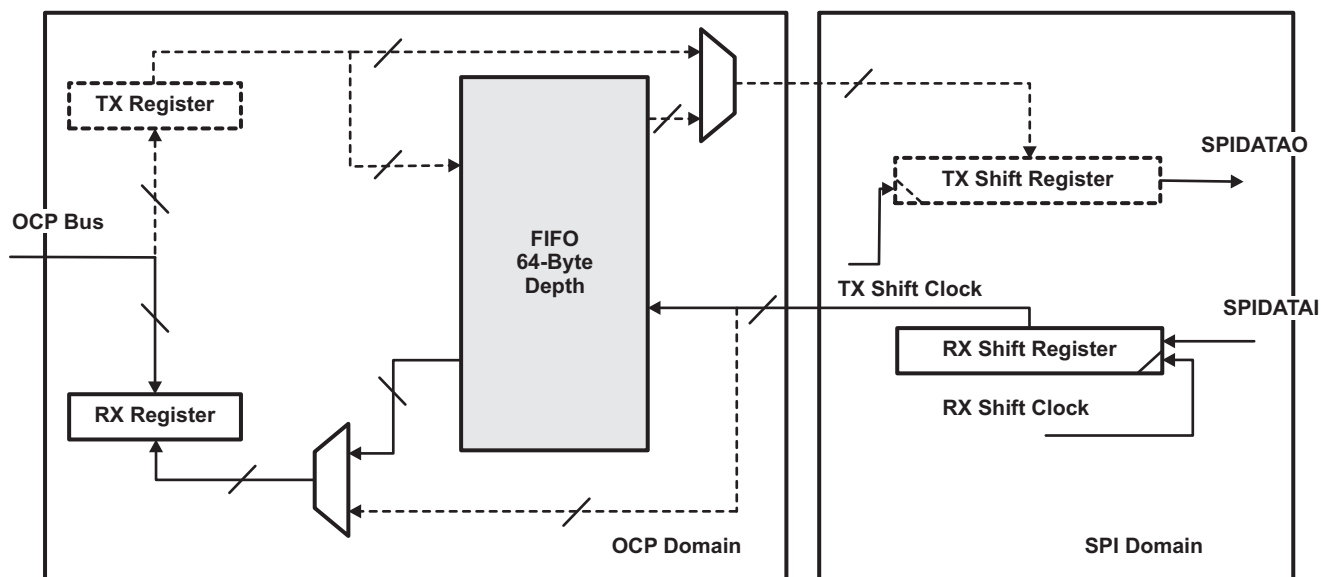
Configuration:
 MCSPI_CH(i)CONF[TRM]=0x0 Transmit/receive mode
 MCSPI_CH(i)CONF[FFRE]=0x1 FIFO enabled on receive path
 MCSPI_CH(i)CONF[FFWE]=0x0 FIFO disabled on transmit path

Figure 24-17. Transmit-Only Mode With FIFO Used



Configuration:
 MCSPI_CH(i)CONF[TRM]=0x2 Transmit only mode
 MCSPI_CH(i)CONF[FFRE]=0x1 FIFO enabled on transmit path
 MCSPI_CH(i)CONF[FFWE] not applicable

Figure 24-18. Receive-Only Mode With FIFO Used



Configuration:
 MCSPI_CH(i)CONF[TRM]=012 Receive only mode
 MCSPI_CH(i)CONF[FFRE]=0x1 FIFO enabled on receive path
 MCSPI_CH(i)CONF[FFWE] not applicable

24.3.2.10.2 Buffer Almost Full

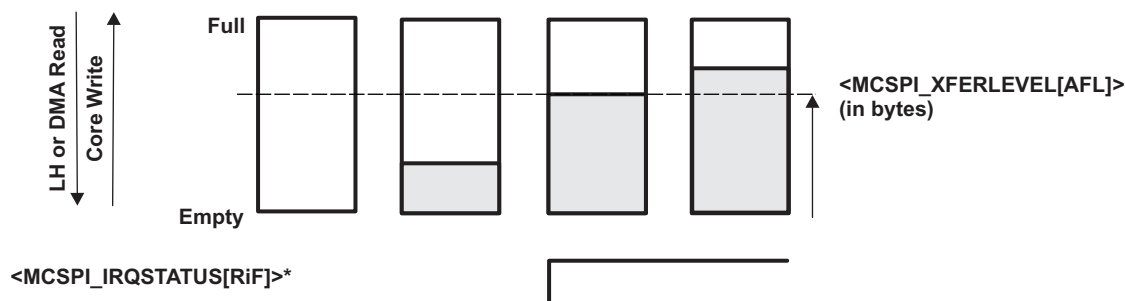
The bit field MCSPI_XFERLEVEL[AFL] is needed when the buffer is used to receive SPI word from a slave (MCSPI_CH(I)CONF[FFER] must be set to 1). It defines the almost full buffer status.

When FIFO pointer reaches this level an interrupt or a DMA request is sent to the CPU to enable system to read AFL+1 bytes from receive register. Be careful AFL+1 must correspond to a multiple value of MCSPI_CH(I)CONF[WL].

When DMA is used, the request is de-asserted after the first receive register read.

No new request will be asserted until the system has performed the correct number of read operations from the buffer.

Figure 24-19. Buffer Almost Full Level (AFL)



* non-DMA mode only. In DMA mode, the DMA RX request is asserted to its active level under identical conditions.

NOTE: SPI_IRQSTATUS register bits are not available in DMA mode. In DMA mode, the SPI_m_DMA_RX_n request is asserted on the same conditions as the SPI_IRQSTATUS RX_n_FULL flag.

24.3.2.10.3 Buffer Almost Empty

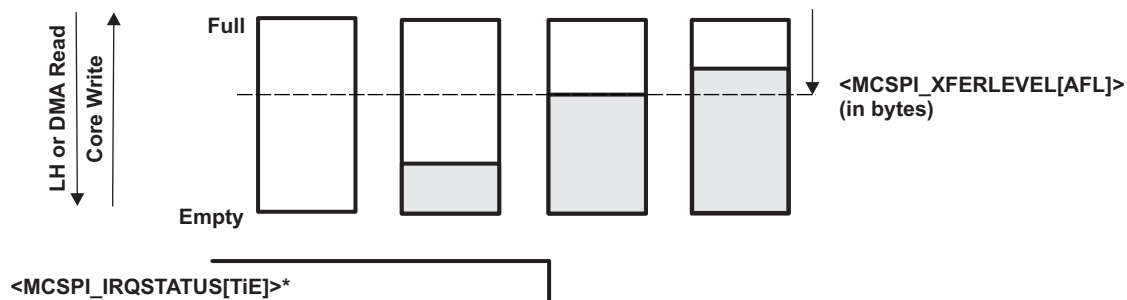
The bitfield MCSPI_XFERLEVEL[AEL] is needed when the buffer is used to transmit SPI word to a slave (MCSPI_CH(I)CONF[FFEW] must be set to 1). It defines the almost empty buffer status.

When FIFO pointer has not reached this level an interrupt or a DMA request is sent to the CPU to enable system to write AEL+1 bytes to transmit register. Be careful AEL+1 must correspond to a multiple value of MCSPI_CH(I)CONF[WL].

When DMA is used, the request is de-asserted after the first transmit register write.

No new request will be asserted until the system has performed the correct number of write operations.

Figure 24-20. Buffer Almost Empty Level (AEL)



* non-DMA mode only. In DMA mode, the DMA TX request is asserted to its active level under identical conditions.

24.3.2.10.4 End of Transfer Management

When the FIFO buffer is enabled for a channel, the user should configure the MCSPI_XFERLEVEL register, the AEL and AFL levels, and, especially, the WCNT bit field to define the number of SPI word to be transferred using the FIFO. This should be done before enabling the channel.

This counter allows the controller to stop the transfer correctly after a defined number of SPI words have been transferred. If WCNT is cleared to 0, the counter is not used and the user must stop the transfer manually by disabling the channel, in this case the user doesn't know how many SPI transfers have been done. For receive transfer, software shall poll the corresponding FFE bit field and read the Receive register to empty the FIFO buffer.

When End Of Word count interrupt is generated, the user can disable the channel and poll on MCSPI_CH(I)STAT[FFE] register to know if SPI word is still there in FIFO buffer and read last words.

24.3.2.10.5 Multiple SPI Word Access

The CPU has the ability to perform multiple SPI word access to the receive or transmit registers within a single 32-bit OCP access by setting the bit field MCSPI_MODULCTRL[MOA] to '1' under specific conditions:

- The channel selected has the FIFO enable.
- Only FIFO sense enabled support the kind of access.
- The bit field MCSPI_MODULCTRL[MOA] is set to 1
- Only 32-bit OCP access and data width can be performed to receive or transmit registers, for other kind of access the CPU must de-assert MCSPI_MODULCTRL[MOA] bit fields.
- The Level MCSPI_XFERLEVEL[AEL] and MCSPI_XFERLEVEL[AFL] must be 32-bit aligned , it means that $AEL[0] = AEL[1] = 1$ or $AFL[0] = AFL[1] = 1$.
- If MCSPI_XFERLEVEL[WCNT] is used it must be configured according to SPI word length.
- The word length of SPI words allows to perform multiple SPI access, that means that $MCSPI_CH(l)CONF[WL] < 16$

Number of SPI word access depending on SPI word length:

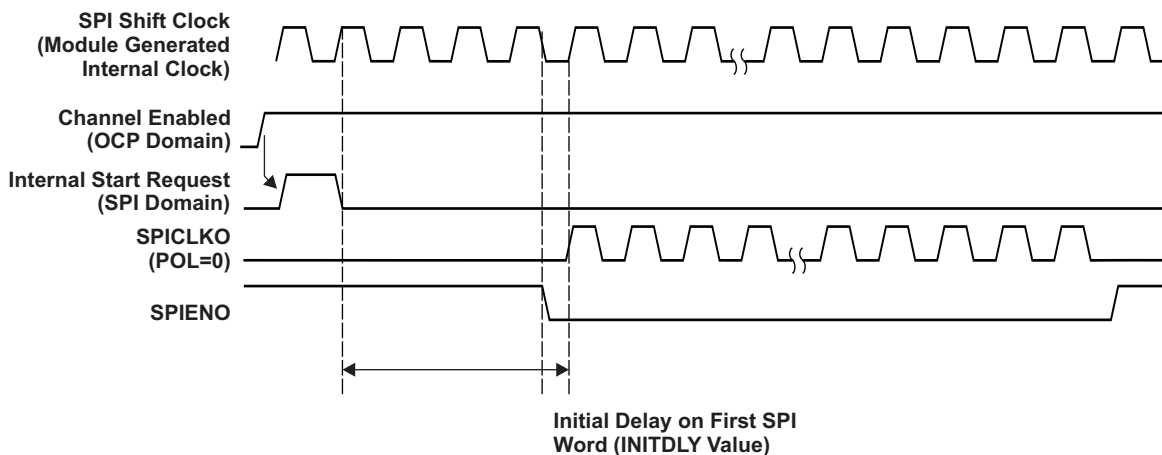
- $3 \leq WL \leq 7$, SPI word length smaller or equal to byte length, four SPI words accessed per 32-bit OCP read/write. If word count is used (MCSPI_XFERLEVEL[WCNT]), set the bit field to $WCNT[0]=WCNT[1]=0$
- $8 \leq WL \leq 15$, SPI word length greater than byte or equal to 16-bit length, two SPI words accessed per 32-bit OCP read/write. If word count is used (MCSPI_XFERLEVEL[WCNT]), set the bit field to $WCNT[0] = 0$.
- $16 \leq WL$ multiple SPI word access not applicable.

24.3.2.11 First SPI Word Delayed

The McSpi controller has the ability to delay the first SPI word transfer to give time for system to complete some parallel processes or fill the FIFO in order to improve transfer bandwidth. This delay is applied only on first SPI word after SPI channel enabled and first write in Transmit register. It is based on output clock frequency.

This option is meaningful in master mode and single channel mode MCSPI_MODULECTRL[SINGLE] asserted.

Figure 24-21. Master Single Channel Initial Delay



Few delay values are available: No delay, 4/8/16/32 Spi cycles.

Its accuracy is half cycle in clock bypass mode and depends on clock polarity and phase.

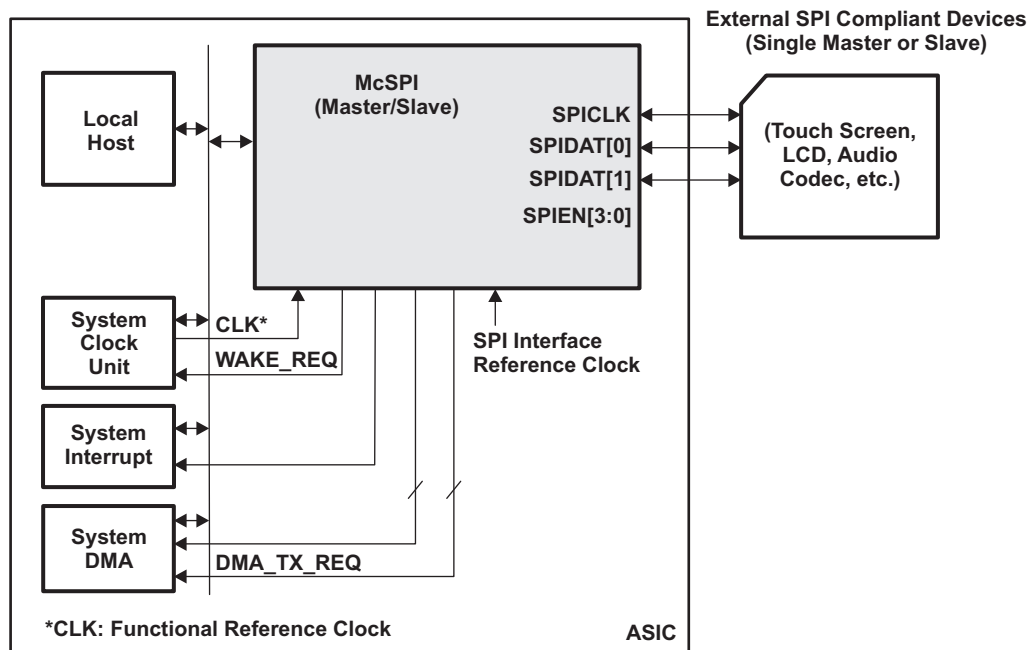
24.3.2.12 3- or 4-Pin Mode

External SPI bus interface can be configured to use a restricted set of pin using the bit field MCSPI_MODULCTRL[PIN34] and depending on targeted application:

- If MCSPI_MODULCTRL[PIN34] is cleared to 0 (default value) the controller is in 4-pin mode using the SPI pins CLKSPI, SOMI, SIMO and chip enable CS.
- If MCSPI_MODULCTRL[PIN34] is set to 1 the controller is in 3-pin mode using the SPI pins CLKSPI, SOMI and SIMO.

In 3-pin mode it is mandatory to put the controller in single channel master mode (MCSPI_MODULECTRL[SINGLE] asserted) and to connect only one SPI device on the bus.

Figure 24-22. 3-Pin Mode System Overview



In 3-pin mode all options related to chip select management are useless:

- MCSPI_CHxCONF[EPOL]
- MCSPI_CHxCONF[TCS0]
- MCSPI_CHxCONF[FORCE]

The chip select pin SPIEN is forced to '0' in this mode.

24.3.3 Slave Mode

McSPI is in slave mode when the bit MS of the register MCSPI_MODULCTRL is set.

In slave mode, McSPI can be connected to up to 4 external SPI master devices. McSPI handles transactions with a single SPI master device at a time.

In slave mode, McSPI initiates data transfer on the data lines (SPIDAT[1;0]) when it receives an SPI clock (SPICLK) from the external SPI master device.

The controller is able to work with or without a chip select SPIEN depending on MCSPI_MODULCTRL[PIN34] bit setting. It also supports transfers without a dead cycle between two successive words.

24.3.3.1 Dedicated Resources

In slave mode, enabling a channel that is not channel 0 has no effect. Only channel 0 can be enabled. The channel 0, in slave mode has the following resources:

- Its own channel enable, programmable with the bit EN of the register MCSPI_CH0CTRL. This channel should be enabled before transmission and reception. Disabling the channel, outside data word transmission, remains under user responsibility.
- Any of the 4 ports SPIEN[3:0] can be used as a slave SPI device enable. This is programmable with the bits SPIENSLV of the register MCSPI_CH0CONF.
- Its own transmitter register MCSPI_TX on top of the common shift register. If the transmitter register is empty, the status bit TXS of the register MCSPI_CH0STAT is set. When McSPI is selected by an external master (active signal on the SPIEN port assigned to channel 0), the transmitter register content of channel0 is always loaded in shift register whether it has been updated or not. The transmitter register should be loaded before McSPI is selected by a master.
- Its own receiver register MCSPI_RX on top of the common shift register. If the receiver register is full, the status bit RXS of the register MCSPI_CH0STAT is set.

NOTE: The transmitter register and receiver registers of the other channels are not used. Read from or Write in the registers of a channel other than 0 has no effect.

- Its own communication configuration with the following parameters via the register MCSPI_CH0CONF:
 - Transmit/Receive modes, programmable with the bit TRM.
 - Interface mode (Two data pins or Single data pin) and data pins assignment, both programmable with the bits IS and DPE.
 - SPI word length, programmable with the bits WL.
 - SPIEN polarity, programmable with the bit EPOL.
 - SPICLK polarity, programmable with the bit POL.
 - SPICLK phase, programmable with the bit PHA.
 - Use a FIFO buffer or not, programmable with FFER and FFEW, depending on transfer mode TRM.

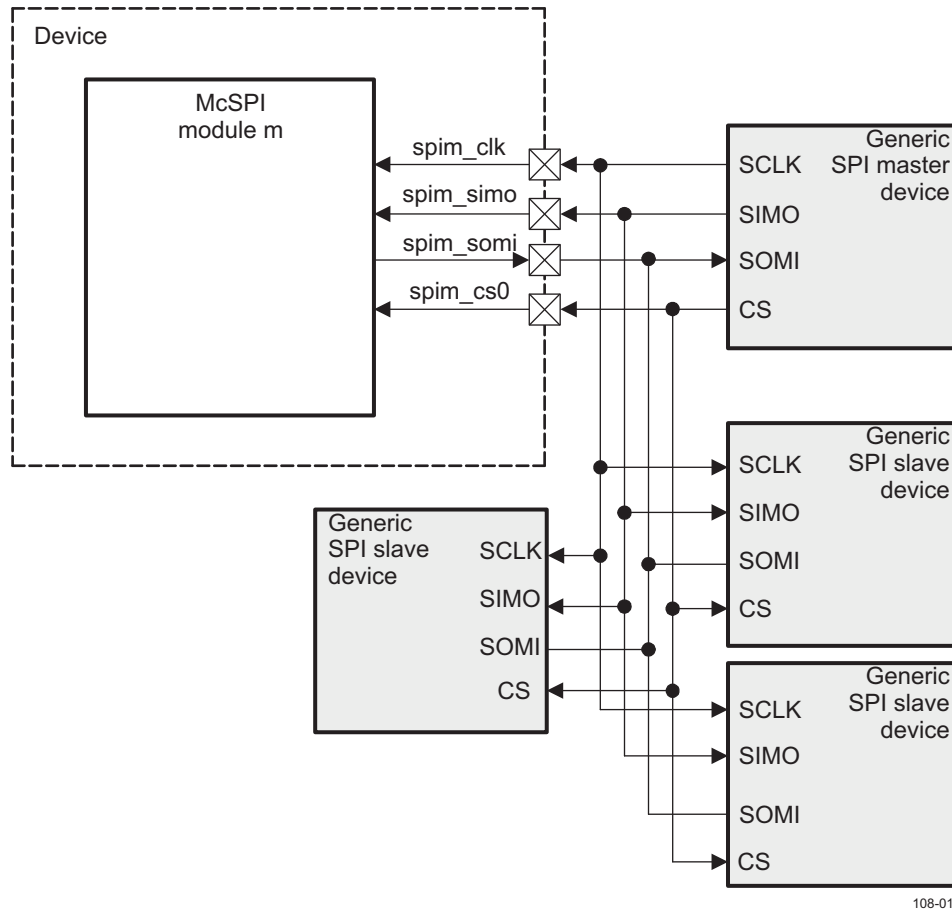
The SPICLK frequency of a transfer is controlled by the external SPI master connected to McSPI. The bits CLKD0 of the register 0CONF are not used in slave mode.

NOTE: The configuration of the channel can be loaded in the MCSPI_CH0CONF register only when the channel is disabled.

- Two DMA requests events, read and write, to synchronize read/write accesses of the DMA controller with the activity of McSPI. The DMA requests are enabled with the bits DMAR and DMAW of the register 0CONF.
- Four interrupts events.

Figure 24-23 shows an example of four slaves wired on a single master device.

Figure 24-23. Example of SPI Slave with One Master and Multiple Slave Devices on Channel 0



24.3.3.2 Interrupt Events in Slave Mode

The interrupt events related to the transmitter register state are TX_empty and TX_underflow. The interrupt events related to the receiver register state are RX_full and RX_overflow.

24.3.3.2.1 TX_EMPTY

The event TX_empty is activated when the channel is enabled and its transmitter register becomes empty. Enabling channel automatically raises this event. When FIFO buffer is enabled (MCSPi_CH(l)CONF[FFEW] set to 1), the TX_empty is asserted as soon as there is enough space in buffer to write a number of byte defined by MCSPi_XFERLEVEL[AEL].

Transmitter register must be load to remove source of interrupt and TX_empty interrupt status bit must be cleared for interrupt line de-assertion (if event enable as interrupt source).

When FIFO is enabled, no new TX_empty event will be asserted unless the host performs the number of writes to the transmitter register defined by MCSPi_XFERLEVEL[AEL]. It is the responsibility of the Local Host to perform the right number of writes.

24.3.3.2.2 TX_UNDERFLOW

The event TX_underflow is activated when channel is enabled and if the transmitter register or FIFO (if use of buffer is enabled) is empty (not updated with new data) when an external master device starts a data transfer with McSPI (transmit and receive).

When the FIFO is enabled, the data read while the underflow flag is set will not be the last word written to the FIFO.

The TX_underflow indicates an error (data loss) in slave mode.

To avoid having TX_underflow event at the beginning of a transmission, the event TX_underflow is not activated when no data has been loaded into the transmitter register since channel has been enabled.

TX_underflow interrupt status bit must be cleared for interrupt line de-assertion (if event enable as interrupt source).

24.3.3.2.3 RX_FULL

The event RX_FULL is activated when channel is enabled and receiver becomes filled (transient event). When FIFO buffer is enabled (MCSPI_CH(I)CONF[FFER] set to 1), the RX_FULL is asserted as soon as there is a number of bytes holds in buffer to read defined by MCSPI_XFERLEVEL[AFL].

Receiver register must be read to remove source of interrupt and RX_full interrupt status bit must be cleared for interrupt line de-assertion (if event enable as interrupt source).

When FIFO is enabled, no new RX_FULL event will be asserted unless the host has performed the number of reads from the receive register defined by MCSPI_XFERLEVEL[AFL]. It is the responsibility of Local Host to perform the right number of reads.

24.3.3.2.4 RX_OVERFLOW

The RX0_OVERFLOW event is activated in slave mode in either transmit-and-receive or receive-only mode, when a channel is enabled and the SPI_RXn register or FIFO is full when a new SPI word is received. The SPI_RXn register is always overwritten with the new SPI word. If the FIFO is enabled, data within the FIFO is overwritten, it must be considered as corrupted. The RX0_OVERFLOW event should not appear in slave mode using the FIFO.

The RX0_OVERFLOW indicates an error (data loss) in slave mode.

The SPI_IRQSTATUS[3] RX0_OVERFLOW interrupt status bit must be cleared for interrupt line deassertion (if the event is enabled as the interrupt source).

24.3.3.2.5 End of Word Count

The event end of word (EOW) count is activated when channel is enabled and configured to use the built-in FIFO. This interrupt is raised when the controller had performed the number of transfer defined in MCSPI_XFERLEVEL[WCNT] register. If the value was programmed to 0000h, the counter is not enabled and this interrupt is not generated.

The EOW count interrupt also indicates that the SPI transfer has halted on the channel using the FIFO buffer.

The EOW interrupt status bit must be cleared for interrupt line de-assertion (if event enable as interrupt source).

24.3.3.3 Slave Transmit-and-Receive Mode

The slave transmit and receive mode is programmable (TRM bit cleared to 0 in the register MCSPI_CH(I)CONF).

After the channel is enabled, transmission and reception proceeds with interrupt and DMA request events.

In slave transmit and receive mode, transmitter register should be loaded before McSPI is selected by an external SPI master device.

Transmitter register or FIFO (if enabled) content is always loaded into the shift register whether it has been updated or not. The event TX_underflow is activated accordingly, and does not prevent transmission.

On completion of SPI word transfer (bit EOT of the register MCSPI_CH(I)STAT is set) the received data is transferred to the channel receive register. This bit is meaningless when using the Buffer for this channel.

The built-in FIFO is available in this mode and can be configured in one data direction, transmit or receive, then the FIFO is seen as a unique 64-byte buffer. It can also be configured in both data directions, transmit and receive, then the FIFO is split into two separate 32-byte buffers with their own address space management.

24.3.3.4 Slave Receive-Only Mode

The slave receive mode is programmable (MCSPI_CH(i)CONF[TRM set to 01).

In receive-only mode, the transmitter register should be loaded before McSPI is selected by an external SPI master device. Transmitter register or FIFO (if enabled) content is always loaded into the shift register whether it has been updated or not. The event TX_underflow is activated accordingly, and does not prevent transmission.

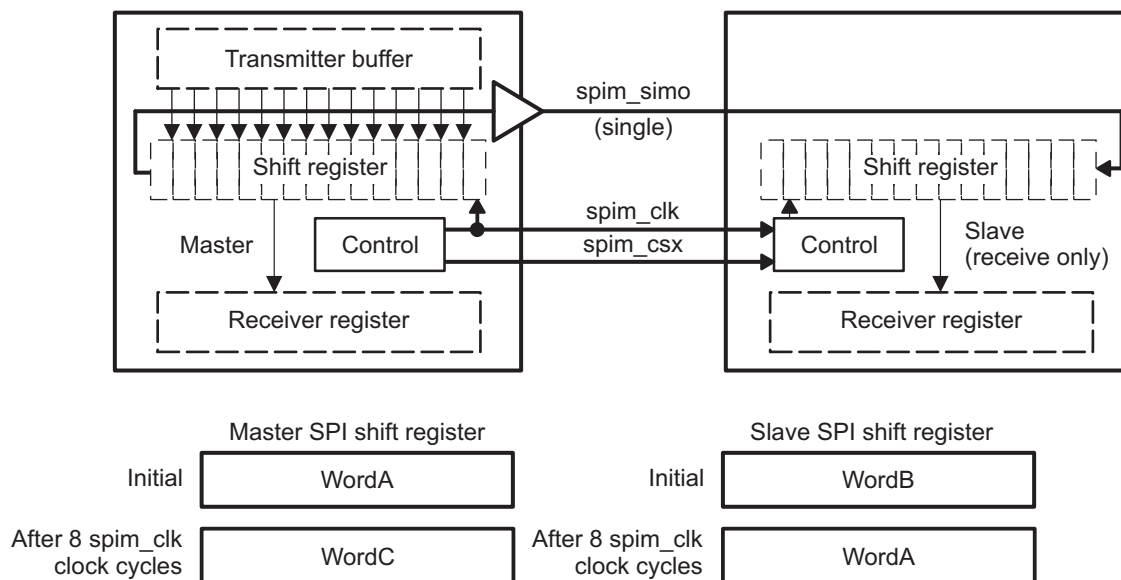
When an SPI word transfer completes (the MCSPI_CH(I)STAT[EOT] bit (with $I = 0$) is set to 1), the received data is transferred to the channel receive register.

To use McSPI as a slave receive-only device with MCSPI_CH(I)CONF[TRM]=00, the user has the responsibility to disable the TX_empty and TX_underflow interrupts and DMA write requests due to the transmitter register state.

On completion of SPI word transfer (bit EOT of the register MCSPI_CH(I)STAT is set) the received data is transferred to the channel receive register. This bit is meaningless when using the Buffer for this channel. The built-in FIFO is available in this mode and can be configured with FFER bit field in the MCSPI_CH(I)CONF register, then the FIFO is seen as a unique 64-byte buffer.

Figure 24-24 shows an example of a half-duplex system with a master device on the left and a receive-only slave device on the right. Each time one bit transfers out from the master, one bit transfers in to the slave. If WordA is 8 bits, then after eight cycles of the serial clock spim_clk, WordA transfers from the master to the slave.

Figure 24-24. SPI Half-Duplex Transmission (Receive-Only Slave)



108-032

24.3.3.5 Slave Transmit-Only Mode

The slave transmit-only mode is programmable (MCSPI_CH(i)CONF[TRM set to 10). This mode eliminates the need for the CPU to read the receiver register (minimizing data movement) when only transmission is meaningful.

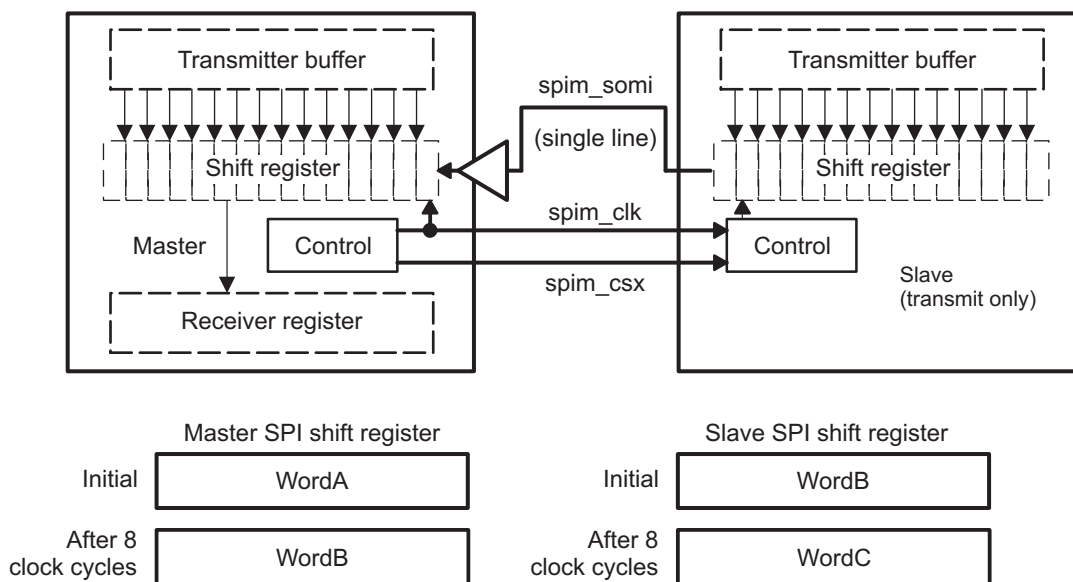
To use McSPI as a slave transmit-only device with MCSPI_CH(I)CONF[TRM]=10, the user should disable the RX_full and RX_overflow interrupts and DMA read requests due to the receiver register state.

On completion of SPI word transfer the bit EOT of the register MCSPI_CH(I)STAT is set. This bit is meaningless when using the Buffer for this channel.

The built-in FIFO is available in this mode and can be configured with FFER bit field in the MCSPI_CH(I)CONF register, then the FIFO is seen as a unique 64-byte buffer.

Figure 24-25 shows a half-duplex system with a master device on the left and a transmit-only slave device on the right. Each time a bit transfers out from the slave device, one bit transfers in the master. If WordB is 8-bits, then after eight cycles of the serial clock spim_clk, WordB transfers from the slave to the master.

Figure 24-25. SPI Half-Duplex Transmission (Transmit-Only Slave)



108-031

24.3.4 Interrupts

According to its transmitter register state and its receiver register state each channel can issue interrupt events if they are enabled.

The interrupt events are listed in the [Section 24.3.2.2](#) and in [Section 24.3.3.2](#).

Each interrupt event has a status bit, in the MCSPI_IRQSTATUS register, which indicates service is required, and an interrupt enable bit, in the MCSPI_IRQENABLE register, which enables the status to generate hardware interrupt requests.

When an interrupt occurs and it is later masked (IRQENABLE), the interrupt line is not asserted again even if the interrupt source has not been serviced.

McSPI supports interrupt driven operation and polling.

24.3.4.1 Interrupt-Driven Operation

Alternatively, an interrupt enable bit, in the MCSPI_IRQENABLE register, can be set to enable each of the events to generate interrupt requests when the corresponding event occurs. Status bits are automatically set by hardware logic conditions.

When an event occurs (the single interrupt line is asserted), the CPU must:

- Read the MCSPI_IRQSTATUS register to identify which event occurred,
- Read the receiver register that corresponds to the event in order to remove the source of an RX_full event, or write into the transmitter register that corresponds to the event in order to remove the source of a TX_empty event. No action is needed to remove the source of the events TX_underflow and RX_overflow.
- Write a 1 into the corresponding bit of MCSPI_IRQSTATUS register to clear the interrupt status, and release the interrupt line.

The interrupt status bit should always be reset after the channel is enabled and before the event is enabled as an interrupt source.

24.3.4.2 Polling

When the interrupt capability of an event is disabled in the MCSPI_IRQENABLE register, the interrupt line is not asserted and:

- The status bits in the MCSPI_IRQSTATUS register can be polled by software to detect when the corresponding event occurs.
- Once the expected event occurs, CPU must read the receiver register that corresponds to the event in order to remove the source of an RX_full event, or write into the transmitter register that corresponds to the event in order to remove the source of a TX_empty event. No action is needed to remove the source of the events TX_underflow and RX_overflow.
- Writing a 1 into the corresponding bit of MCSPI_IRQSTATUS register clears the interrupt status and does not affect the interrupt line state.

24.3.5 DMA Requests

McSPI can be interfaced with a DMA controller. At system level, the advantage is to free the local host of the data transfers.

According to its transmitter register state, its receiver register state or FIFO level (if use of buffer for the channel) each channel can issue DMA requests if they are enabled.

The DMA requests need to be disabled in order to get TX and RX interrupts, in order to define either the end of the transfer or the transfer of the last words for the modes listed below:

- Master Transmit On
- Master normal receive only mode
- Master turbo receive only m
- Slave Transmit Only

There are two DMA request lines per channel. The management of DMA requests differ according to use of FIFO buffer or not:

24.3.5.1 FIFO Buffer Disabled

The DMA Read request line is asserted when the channel is enabled and a new data is available in the receive register of the channel. DMA Read request can be individually masked with the bit DMAR of the register MCSPI_CH(I)CONF. The DMA Read request line is de-asserted on read completion of the receive register of the channel.

The DMA Write request line is asserted when the channel is enabled and the transmitter register of the channel is empty. DMA Write request can be individually masked with the bit DMAW of the register MCSPI_CH(I)CONF. The DMA Write request line is de-asserted on load completion of the transmitter register of the channel.

Only one SPI word can be transmitted/received per OCP bus access to write/read the transmit or receive register.

24.3.5.2 FIFO Buffer Enabled

The DMA Read request line is asserted when the channel is enabled and a number of bytes defined in MCSPI_XFERLEVEL[AFL] bit field is hold in FIFO buffer for the receive register of the channel. DMA Read request can be individually masked with the bit DMAR of the register MCSPI_CH(I)CONF. The DMA Read request line is de-asserted on the first SPI word read completion of the receive register of the channel. No new DMA request will be asserted again as soon as user has not performed the right number of read accesses defined by MCSPI_XFERLEVEL[AFL] it is under user responsibility.

The DMA Write request line is asserted when the channel is enabled and the number of bytes hold in FIFO buffer is below the level defined by the MCSPI_XFERLEVEL[AEL] bit field. DMA Write request can be individually masked with the bit DMAW of the register MCSPI_CH(I)CONF. The DMA Write request line is de-asserted on load completion of the first SPI word in the transmitter register of the channel. No new DMA request will be asserted again as soon as user has not performed the right number of write accesses defined by MCSPI_XFERLEVEL[AEL] it is under user responsibility.

Only one SPI word can be transmitted/received per OCP bus access to write/read the transmit or receive FIFO.

24.3.5.3 DMA 256-Bit Aligned Addresses

The controller has two registers, MCSPI_DAFTX and MCSPI_DAFRX, used only with an enabled channel which manages the FIFO to be compliant the a DMA handler providing only 256-bit aligned addresses.

This features is activated when the bit field MCSPI_MODULCTRL[FDDA] is set to '1' and only one enabled channel have its bit field MCSPI_CH(I)CONF[FFEW] or MCSPI_CH(I)CONF[FFER] enabled.

In this case the registers MCSPI_TX(I) and MCSPI_RX(I) are not used and data is managed through registers MCSPI_DAFTX and MCSPI_DAFRX.

24.3.6 Emulation Mode

The MReqDebug input differentiates a regular access of a processor (application access), from an emulator access.

Application access: MReqDebug = 0

In functional mode, the consequences of a read of a receiver register MCSPI_RX(i) are the following:

- The source of an RXi_Full event in the MCSPI_IRQSTATUS register is removed, if it was enabled in the MCSPI_IRQENABLE register.
- The RXiS status bit in the MCSPI_IRQSTATUS register is cleared.
- In master mode, depending on the round robin arbitration, and the transmitter register state, the channel may access to the shift register for transmission/reception.

Emulator access: MReqDebug = 1

In emulation mode, McSPI behavior is the same as in functional mode but a read of a receiver register MCSPI_RX(i) is not intrusive:

- MCSPI_RX(i) is still considered as not read. When the FIFO buffer is enabled, pointers are not updated.
- The source of an RXi_Full event in the MCSPI_IRQSTATUS register is not removed. The RXiS status bit in the MCSPI_CH(i)STAT register is held steady.

In emulation mode, as in functional mode, based on the ongoing data transfers, the status bits of the MCSPI_CH(i)STAT register may be optionally updated, the interrupt and DMA request lines may be optionally asserted.

24.3.7 Power Saving Management

Independently of the module operational modes (Transmit and/or Receive), two modes of operations are defined from a power management perspective: normal and idle modes.

The two modes are temporally fully exclusive.

24.3.7.1 Normal Mode

Both the OCP clock and SPI clock (CLKSPIREF) provided to McSPI must be active for both master and slave modes. The auto-gating of the module OCP clock and SPI clock occurs when the following conditions are met:

- The bit Autoidle of the register MCSPI_SYSCONFIG is set.
- In master mode, there is no data to transmit or receive in all channels.
- In slave mode, the SPI is not selected by the external SPI master device and no OCP accesses.

Autogating of the module OCP clock and SPI clock stops when the following conditions are met:

- In master mode, an OCP access occurs.
- In slave mode, an OCP access occurs or McSPI is selected by an external SPI master device.

24.3.7.2 Idle Mode

The OCP clock and SPI clock provided to McSPI may be switched off on system power manager request and switched back on module request.

McSPI is compliant with the power management handshaking protocol: idle request from the system power manager, idle acknowledgement from McSPI.

The idle acknowledgement in response to an idle request from the system power manager varies according to a programmable mode in the MCSPI_SYSCONFIG register: No idle mode, force idle mode, and smart idle mode.

- When programmed for no idle mode (the bit SIdleMode of the register MCSPI_SYSCONFIG is set to "01"), the module ignores the system power manager request, and behaves normally, as if the request was not asserted.
- When programmed for smart idle mode (the bit SIdleMode of the register MCSPI_SYSCONFIG is set to "10"), the module acknowledges the system power manager request according to its internal state.
- When programmed for force idle mode (the bit SIdleMode of the register MCSPI_SYSCONFIG is set to "00"), the module acknowledges the system power manager request unconditionally.

The OCP clock will be optionally switched off, during the smart idle mode period, if the bit ClockActivity of the register MCSPI_SYSCONFIG is set.

The SPI clock will be optionally switched off, during the smart idle mode period, if the second bit ClockActivity of the register MCSPI_SYSCONFIG is set.

McSPI assumes that both clocks may be switched off whatever the value set in the field ClockActivity of the register MCSPI_SYSCONFIG.

24.3.7.2.1 Transitions from Normal Mode to Smart-Idle Mode

The module detects an idle request when the synchronous signal IdleReq is asserted.

When IdleReq is asserted, any access to the module will generate an error as long as the OCP clock is alive.

When configured as a slave device, McSPI responds to the idle request by asserting the SIdleAck signal (idle acknowledgement) only after completion of the current transfer (SPIEN slave selection signal deasserted by the external master) and if interrupt or DMA request lines are not asserted.

As a master device, McSPI responds to the idle request by asserting the SIdleAck signal (idle acknowledgement) only after completion of all the channel data transfers and if interrupt or DMA request lines are not asserted.

As long as SIdleAck is not asserted, if an event occurs, the module can still generate an interrupt or a DMA request after IdleReq assertion. In this case, the module ignores the idle request and SIdleAck will not get asserted: The system power manager will abort the power mode transition procedure. It is then the responsibility of the system to de-assert IdleReq before attempting to access the module.

When SIdleAck is asserted, the module does not assert any new interrupt or DMA request.

24.3.7.2.2 Transition From Smart-Idle Mode to Normal mode

McSPI detects the end of the idle period when the idle request signal (IdleReq) is deasserted.

Upon IdleReq de-assertion, the module switches back to normal mode and de-asserts SIdleAck signal. The module is fully operational.

24.3.7.2.3 Force-Idle Mode

Force-idle mode is enabled as follows:

- The bit SIdleMode of the register MCSPI_SYSCONFIG is cleared to “00” (Force Idle).
The force idle mode is an idle mode where McSPI responds unconditionally to the idle request by asserting the SIdleAck signal and by deasserting unconditionally the interrupt and DMA request lines if asserted.
The transition from normal mode to idle mode does not affect the interrupt event bits of the MCSPI_IRQSTATUS register.
In force-idle mode, the module is supposed to be disabled at that time, so the interrupt and DMA request lines are likely deasserted. OCP clock and SPI clock provided to McSPI can be switched off.
An idle request during an SPI data transfer can lead to an unexpected and unpredictable result, and is under software responsibility.
Any access to the module in force idle mode will generate an error as long as the OCP clock is alive and IdleReq is asserted.

The module exits the force idle mode when:

- The idle request signal (IdleReq) is de-asserted.
Upon IdleReq de-assertion, the module switches back to normal mode and de-asserts SIdleAck signal. The module is fully operational. The interrupt and DMA request lines are optionally asserted a clock cycle later.

24.3.8 System Test Mode

McSPI is in system test mode (SYSTEST) when the bit System_Test of the register MCSPI_MODULCTRL is set.

The SYSTEST mode is used to check in a very simple manner the correctness of the system interconnect either internally to interrupt handler, or power manager, or externally to SPI I/Os.

I/O verification can be performed in SYSTEST mode by toggling the outputs and capturing the logic state of the inputs. (See MCSPI_SYST register definition in [Section 24.4.1.6](#))

24.3.9 Reset

24.3.9.1 Internal Reset Monitoring

The module is reset by the hardware when an active-low reset signal, synchronous to the OCP interface clock is asserted on the input pin RESETN.

This hardware reset signal has a global reset action on the module. All configuration registers and all state machines are reset, in all clock domains.

Additionally, the module can be reset by software through the bit SoftReset of the register MCSPI_SYSCONFIG. This bit has exactly the same action on the module logic as the hardware RESETN signal. The register MCSPI_SYSCONFIG is not sensitive to software reset. The SoftReset control bit is active high. The bit is automatically reset to 0 by the hardware.

A global ResetDone status bit is provided in the status register MCSPI_SYSSTATUS. This bit is set to 1 when all the different clock domains resets (OCP domain and SPI domains) have been released (logical AND).

The global ResetDone status bit can be monitored by the software to check if the module is ready-to-use following a reset (either hardware or software).

The clock CLKSPIREF must be provided to the module, in order to allow the ResetDone status bit to be set.

When used in slave mode, the clock CLKSPIREF is needed only during the reset phase. The clock CLKSPIREF can be switched off after the ResetDone status is set.

24.3.9.2 Reset Values of Registers

The reset values of registers and signals are described in [Section 24.4.1](#).

24.3.10 Access to Data Registers

This section details the supported data accesses (read or write) from/to the data receiver registers MCSPI_RX(i) and data transmitter registers MCSPI_TX(i).

Supported access:

McSPI supports only one SPI word per register (receiver or transmitter) and does not support successive 8-bit or 16-bit accesses for a single SPI word.

The SPI word received is always right justified on LSbit of the 32bit register MCSPI_RX(i), and the SPI word to transmit is always right justified on LSbit of the 32bit register MCSPI_TX(i).

The upper bits, above SPI word length, are ignored and the content of the data registers is not reset between the SPI data transfers.

The coherence between the number of bits of the SPI Word, the number of bits of the access and the enabled byte remains under the user's responsibility. Only aligned accesses are supported.

In Master mode, data should not be written in the transmit register when the channel is disabled.

24.3.11 Programming Aid

24.3.11.1 Module Initialization

- Hard or soft reset.
- Read MCSPI_SYSSTATUS.
- Check if reset is done.
- Module configuration: (a) Write into MCSPI_MODULCTRL (b) Write into MCSPI_SYSCONFIG.
- Before the ResetDone bit is set, the clocks CLK and CLKSPIREF must be provided to the module.
- To avoid hazardous behavior, it is advised to reset the module before changing from MASTER mode to SLAVE mode or from SLAVE mode to MASTER mode.

24.3.11.2 Common Transfer Sequence

McSPI module allows the transfer of one or several words, according to different modes:

- MASTER, MASTER Turbo, SLAVE
- TRANSMIT - RECEIVE, TRANSMIT ONLY, RECEIVE ONLY
- Write and Read requests: Interrupts, DMA
- SPIEN lines assertion/deassertion: automatic, manual

For all these flows, the host process contains the main process and the interrupt routines. The interrupt routines are called on the interrupt signals or by an internal call if the module is used in polling mode.

In multi-channel master mode, the flows of different channels can be run simultaneously.

24.3.11.3 Main Program

- Interrupt Initialization: (a) Reset status bits in MCSPI_IRQSTATUS (b) Enable interrupts in MCSPI_IRQENA.
- Channel Configuration: Write MCSPI_CH(i)CONF.
- Start the channel: Write 0000 0001h in MCSPI_CH(i)CTRL.
- First write request: TX empty - Generate DMA write event/ polling TX empty flag by CPU to write First transmit word into MCSPI_TX(i).
- End of transfer: Stop the channel by writing 0000 0000h in MCSPI_CH(i)CTRL

The end of transfer depends on the transfer mode.

In multi-channel master mode, be careful not to overwrite the bits of other channels when initializing MCSPI_IRQSTATUS and MCSPI_IRQENABLE.

24.3.12 Interrupt and DMA Events

McSPI has two DMA requests (Rx and Tx) per channel. It also has one interrupt line for all the interrupt requests.

24.4 McSPI Registers

24.4.1 SPI Registers

[Table 24-10](#) lists the McSPI registers.

Table 24-10. SPI Registers

Offset Address	Acronym	Register Name	Section
000h	MCSPI_REVISION	McSPI revision register	Section 24.4.1.1
110h	MCSPI_SYSCONFIG	McSPI system configuration register	Section 24.4.1.2
114h	MCSPI_SYSSTATUS	McSPI system status register	Section 24.4.1.3
118h	MCSPI_IRQSTATUS	McSPI interrupt status register	Section 24.4.1.4
11Ch	MCSPI_IRQENABLE	McSPI interrupt enable register	Section 24.4.1.5
124h	MCSPI_SYST	McSPI system register	Section 24.4.1.6
128h	MCSPI_MODULCTRL	McSPI module control register	Section 24.4.1.7
12Ch	MCSPI_CH0CONF	McSPI channel <i>i</i> configuration register	Section 24.4.1.8
130h	MCSPI_CH0STAT	McSPI channel <i>i</i> status register	Section 24.4.1.9
134h	MCSPI_CH0CTRL	McSPI channel <i>i</i> control register	Section 24.4.1.10
138h	MCSPI_TX0	McSPI channel <i>i</i> FIFO transmit buffer register	Section 24.4.1.11
13Ch	MCSPI_RX0	McSPI channel <i>i</i> FIFO receive buffer register	Section 24.4.1.12
140h	MCSPI_CH1CONF	McSPI channel <i>i</i> configuration register	Section 24.4.1.8
144h	MCSPI_CH1STAT	McSPI channel <i>i</i> status register	Section 24.4.1.9
148h	MCSPI_CH1CTRL	McSPI channel <i>i</i> control register	Section 24.4.1.10
14Ch	MCSPI_TX1	McSPI channel <i>i</i> FIFO transmit buffer register	Section 24.4.1.11
150h	MCSPI_RX1	McSPI channel <i>i</i> FIFO receive buffer register	Section 24.4.1.12
154h	MCSPI_CH2CONF	McSPI channel <i>i</i> configuration register	Section 24.4.1.8
158h	MCSPI_CH2STAT	McSPI channel <i>i</i> status register	Section 24.4.1.9
15Ch	MCSPI_CH2CTRL	McSPI channel <i>i</i> control register	Section 24.4.1.10
160h	MCSPI_TX2	McSPI channel <i>i</i> FIFO transmit buffer register	Section 24.4.1.11
164h	MCSPI_RX2	McSPI channel <i>i</i> FIFO receive buffer register	Section 24.4.1.12
168h	MCSPI_CH3CONF	McSPI channel <i>i</i> configuration register	Section 24.4.1.8
16Ch	MCSPI_CH3STAT	McSPI channel <i>i</i> status register register	Section 24.4.1.9
170h	MCSPI_CH3CTRL	McSPI channel <i>i</i> control register	Section 24.4.1.10

Table 24-10. SPI Registers (continued)

Offset Address	Acronym	Register Name	Section
174h	MCSPi_TX3	McSPI channel <i>i</i> FIFO transmit buffer register	Section 24.4.1.11
178h	MCSPi_RX3	McSPI channel <i>i</i> FIFO receive buffer register	Section 24.4.1.12
17Ch	MCSPi_XFERLEVEL	McSPI transfer levels register	Section 24.4.1.13
180h	MCSPi_DAFTX	McSPI DMA address aligned FIFO transmitter register	
1A0h	MCSPi_DAFRX	McSPI DMA address aligned FIFO receiver register	Section 24.4.1.15

24.4.1.2 McSPI System Configuration Register (MCSPI_SYSCONFIG)

The McSPI system configuration register (MCSPI_SYSCONFIG) allows control of various parameters of the module interface. It is not sensitive to software reset. The MCSPI_SYSCONFIG is shown in [Figure 24-27](#) and described in [Table 24-12](#).

Figure 24-27. McSPI System Configuration Register (MCSPI_SYSCONFIG)

31	Reserved																16
R/W-0																	
15	Reserved										10	9	8				
R/W-0										CLOCKACTIVITY							
R/W-0										R/W-0							
7	Reserved				5	4	3	2	1	0							
R/W-0					SIDLEMODE				Reserved		SOFTRESET		AUTOIDLE				
R/W-0					R/W-2h				R		R/W-0		R/W-1				

LEGEND: R/W = Read/Write; -n = value after reset

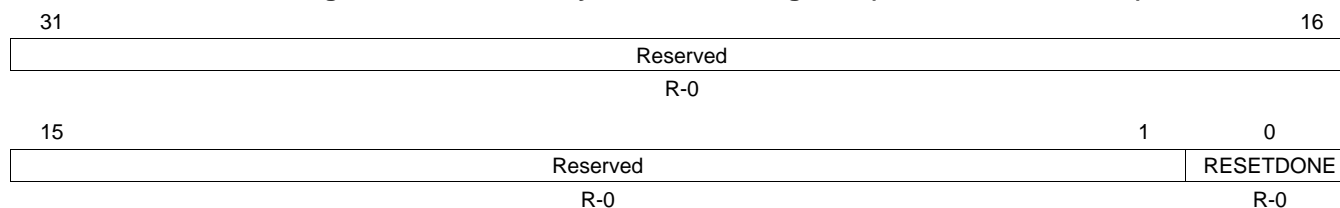
Table 24-12. McSPI System Configuration Register (MCSPI_SYSCONFIG) Field Descriptions

Bit	Field	Value	Description
31-10	Reserved	0	Reads returns 0
9-8	CLOCKACTIVITY	0 1h 2h 3h	Clocks activity during wake-up mode period. OCP and Functional clocks may be switched off. OCP clock is maintained. Functional clock may be switched-off. Functional clock is maintained. OCP clock may be switched-off. OCP and Functional clocks are maintained.
7-5	Reserved	0	Reads returns 0
4-3	SIDLEMODE	0 1h 2h 3h	Power management If an idle request is detected, the McSPI acknowledges it unconditionally and goes in Inactive mode. Interrupt, DMA requests are unconditionally de-asserted. If an idle request is detected, the request is ignored and keeps on behaving normally. Smart-idle mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements. Reserved
2	Reserved	0	Reserved
1	SOFTRESET	0 1	Software reset. During reads it always returns 0. (write) Normal mode (write) Set this bit to 1 to trigger a module reset. The bit is automatically reset by the hardware.
0	AUTOIDLE	0 1	Internal OCP Clock gating strategy OCP clock is free-running Automatic OCP clock gating strategy is applied, based on the OCP interface activity

24.4.1.3 McSPI System Status Register (MCSPI_SYSSTATUS)

The McSPI system status register (MCSPI_SYSSTATUS) provides status information about the module excluding the interrupt status information. The MCSIPI_SYSSTATUS is shown in [Figure 24-28](#) and described in [Table 24-13](#).

Figure 24-28. McSPI System Status Register (MCSPI_SYSSTATUS)



LEGEND: R = Read only; -n = value after reset

Table 24-13. McSPI System Status Register (MCSPI_SYSSTATUS) Field Descriptions

Bit	Field	Value	Description
31-1	Reserved	0	Reserved for module specific status information. Read returns 0.
0	RESETDONE	0	Internal Reset Monitoring
		0	Internal module reset is on-going
		1	Reset completed

24.4.1.4 McSPI Interrupt Status Register (MCSPI_IRQSTATUS)

The McSPI interrupt status register (MCSPI_IRQSTATUS) regroups all the status of the module internal events that can generate an interrupt. The MCSPI_IRQSTATUS is shown in [Figure 24-29](#) and described in [Table 24-14](#).

NOTE: In SYSTEST mode, the bits of this register have no meaning and always read 0.

Figure 24-29. McSPI Interrupt Status Register (MCSPI_IRQSTATUS)

31																		18		17		16	
Reserved																		EOW		Rsvd			
R/W-0																		R/W-0		R-0			
15		14		13		12		11		10		9		8									
Rsvd		RX3_FULL		TX3_UNDERFLOW		TX3_EMPTY		Reserved		RX2_FULL		TX2_UNDERFLOW		TX2_EMPTY									
R/W-0		R/W-0		R/W-0		R/W-0		R/W-0		R/W-0		R/W-0		R/W-0									
7		6		5		4		3		2		1		0									
Rsvd		RX1_FULL		TX1_UNDERFLOW		TX1_EMPTY		RX0_OVERFLOW		RX0_FULL		TX0_UNDERFLOW		TX0_EMPTY									
R/W-0		R/W-0		R/W-0		R/W-0		R/W-0		R/W-0		R/W-0		R/W-0									

LEGEND: R/W = Read/Write; -n = value after reset

Table 24-14. McSPI Interrupt Status Register (MCSPI_IRQSTATUS) Field Descriptions

Bit	Field	Value	Description
31-18	Reserved	0	Reads returns 0
17	EOW	Write 0 Read 0 Write 1 Read 1	End of word (EOW) count event when a channel is enabled using the FIFO buffer and the channel has sent the number of McSPI words defined by the MCSPI_XFERLEVEL[WCNT]. Event status bit is unchanged. Event false. Event status bit is reset. Event is pending.
16	Reserved	0	Reserved
15	Reserved	0	Reads returns 0
14	RX3_FULL	Write 0 Read 0 Write 1 Read 1	Receiver register is full or almost full. Only when Channel 3 is enabled. This bit indicate FIFO almost full status when built-in FIFO is used for receive register (MCSPI_CH3CONF[FFE3R] is set). Event status bit is unchanged. Event false. Event status bit is reset. Event is pending.
13	TX3_UNDERFLOW	Write 0 Read 0 Write 1 Read 1	Transmitter register underflow. Only when Channel 3 is enabled. The transmitter register is empty (not updated by Host or DMA with new data) before its time slot assignment. Exception: No TX_underflow event when no data has been loaded into the transmitter register since channel has been enabled. Event status bit is unchanged. Event false. Event status bit is reset. Event is pending.

Table 24-14. McSPI Interrupt Status Register (MCSPI_IRQSTATUS) Field Descriptions (continued)

Bit	Field	Value	Description
12	TX3_EMPTY	Write 0 Event status bit is unchanged. Read 0 Event false. Write 1 Event status bit is reset. Read 1 Event is pending.	Transmitter register is empty or almost empty. This bit indicate FIFO almost full status when built-in FIFO is used for transmit register (MCSPI_CH3CONF[FFE3W] is set). Note: Enabling the channel automatically raises this event.
11	Reserved	0	Reads returns 0
10	RX2_FULL	Write 0 Event status bit is unchanged. Read 0 Event false. Write 1 Event status bit is reset. Read 1 Event is pending.	Receiver register full or almost full. Channel 2 This bit indicate FIFO almost full status when built-in FIFO is used for receive register (MCSPI_CH3CONF[FFE2R] is set).
9	TX2_UNDERFLOW	Write 0 Event status bit is unchanged. Read 0 Event false. Write 1 Event status bit is reset. Read 1 Event is pending.	Transmitter register underflow. Channel 2
8	TX2_EMPTY	Write 0 Event status bit is unchanged. Read 0 Event false. Write 1 Event status bit is reset. Read 1 Event is pending.	Transmitter register empty or almost empty. Channel 2. This bit indicate FIFO almost full status when built-in FIFO is used for transmit register (MCSPI_CH3CONF[FFE2W] is set).
7	Reserved	0	Reads returns 0
6	RX1_FULL	Write 0 Event status bit is unchanged. Read 0 Event false. Write 1 Event status bit is reset. Read 1 Event is pending.	Receiver register full or almost full. Channel 1. This bit indicate FIFO almost full status when built-in FIFO is use for receive register (MCSPI_CH3CONF[FFE1R] is set).
5	TX1_UNDERFLOW	Write 0 Event status bit is unchanged. Read 0 Event false. Write 1 Event status bit is reset. Read 1 Event is pending.	Transmitter register underflow. Channel 1.
4	TX1_EMPTY	Write 0 Event status bit is unchanged. Read 0 Event false. Write 1 Event status bit is reset. Read 1 Event is pending.	Transmitter register empty or almost empty. Channel 1. This bit indicate FIFO almost full status when built-in FIFO is use for transmit register (MCSPI_CH3CONF[FFE1W] is set).
3	RX0_OVERFLOW	Write 0 Event status bit is unchanged. Read 0 Event false. Write 1 Event status bit is reset. Read 1 Event is pending.	Receiver register overflow (slave mode only). Channel 0.

Table 24-14. McSPI Interrupt Status Register (MCSPI_IRQSTATUS) Field Descriptions (continued)

Bit	Field	Value	Description
2	RX0_FULL	Write 0 Read 0 Write 1 Read 1	Receiver register full or almost full. Channel 0. Receiver register full or almost full. Channel 0 Event status bit is unchanged. Event false. Event status bit is reset. Event is pending.
1	TX0_UNDERFLOW	Write 0 Read 0 Write 1 Read 1	Transmitter register underflow. Channel 0. Event status bit is unchanged. Event false. Event status bit is reset. Event is pending.
0	TX0_EMPTY	Write 0 Read 0 Write 1 Read 1	Transmitter register empty or almost empty. Channel 0. This bit indicate FIFO almost full status when built-in FIFO is use for transmit register (MCSPI_CH3CONF[FFE0W] is set). Event status bit is unchanged. Event false. Event status bit is reset. Event is pending.

24.4.1.5 McSPI Interrupt Enable Register (MCSPI_IRQENABLE)

This McSPI interrupt enable register (MCSPI_IRQENABLE) enables/disables the module internal sources of interrupt, on an event-by-event basis. The MCSPI_IRQENABLE is shown in [Figure 24-30](#) and described in [Table 24-15](#).

Figure 24-30. McSPI Interrupt Enable Register (MCSPI_IRQENABLE)

31																18		17	16	
Reserved																		EOWKE	Rsvd	
R/W-0																R/W-0		R-0		
15		14		13		12		11		10		9		8						
Rsvd	RX3_FULL_ENABLE	TX3_UNDERFLOW_ENABLE		TX3_EMPTY_ENABLE		Reserved		RX2_FULL_ENABLE		TX2_UNDERFLOW_ENABLE		TX2_EMPTY_ENABLE								
R/W-0		R/W-0		R/W-0		R/W-0		R/W-0		R/W-0		R/W-0								
7		6		5		4		3		2		1		0						
Rsvd	RX1_FULL_ENABLE	TX1_UNDERFLOW_ENABLE		TX1_EMPTY_ENABLE		RX0_OVERFLOW_ENABLE		RX0_FULL_ENABLE		TX0_UNDERFLOW_ENABLE		TX0_EMPTY_ENABLE								
R/W-0		R/W-0		R/W-0		R/W-0		R/W-0		R/W-0		R/W-0								

LEGEND: R/W = Read/Write; -n = value after reset

Table 24-15. McSPI Interrupt Enable Register (MCSPI_IRQENABLE) Field Descriptions

Bit	Field	Value	Description
31-18	Reserved	0	Reads return 0
17	EOWKE	0 1	End of word count interrupt enable. Interrupt is disabled. Interrupt is enabled.
16	Reserved		Reserved
15	Reserved	0	Reads return 0
14	RX3_FULL_ENABLE	0 1	McSPI_RX3 receiver register full or almost full interrupt enable (channel 3). Interrupt is disabled. Interrupt is enabled.
13	TX3_UNDERFLOW_ENABLE	0 1	McSPI_TX3 transmitter register underflow interrupt enable (channel 3). Interrupt is disabled. Interrupt is enabled.
12	TX3_EMPTY_ENABLE	0 1	McSPI_TX3 transmitter register empty or almost empty interrupt enable (channel 3). Interrupt is disabled. Interrupt is enabled.
11	Reserved	0	Reads return 0
10	RX2_FULL_ENABLE	0 1	McSPI_RX2 receiver register full or almost full interrupt enable (channel 2). Interrupt is disabled. Interrupt is enabled.
9	TX2_UNDERFLOW_ENABLE	0 1	McSPI_TX2 transmitter register underflow interrupt enable (channel 2). Interrupt is disabled. Interrupt is enabled.
8	TX2_EMPTY_ENABLE	0 1	McSPI_TX2 transmitter register empty or almost empty interrupt enable (channel 2). Interrupt is disabled. Interrupt is enabled.
7	Reserved	0	Reads return 0

Table 24-15. McSPI Interrupt Enable Register (MCSPI_IRQENABLE) Field Descriptions (continued)

Bit	Field	Value	Description
6	RX1_FULL_ENABLE	0	MCSPi_RX1 receiver register full or almost full interrupt enable (channel 1) Interrupt is disabled.
		1	Interrupt is enabled.
5	TX1_UNDERFLOW_ENABLE	0	MCSPi_TX1 transmitter register underflow interrupt enable (channel 1). Interrupt is disabled.
		1	Interrupt is enabled.
4	TX1_EMPTY_ENABLE	0	MCSPi_TX1 transmitter register empty or almost empty interrupt enable (channel 1). Interrupt is disabled.
		1	Interrupt is enabled.
3	RX0_OVERFLOW_ENABLE	0	MCSPi_RX0 receiver register overflow interrupt enable (channel 0). Interrupt is disabled.
		1	Interrupt is enabled.
2	RX0_FULL_ENABLE	0	MCSPi_RX0 receiver register full or almost full interrupt enable (channel 0). Interrupt is disabled.
		1	Interrupt is enabled.
1	TX0_UNDERFLOW_ENABLE	0	MCSPi_TX0 transmitter register underflow interrupt enable (channel 0). Interrupt is disabled.
		1	Interrupt is enabled.
0	TX0_EMPTY_ENABLE	0	MCSPi_TX0 transmitter register empty or almost empty interrupt enable (channel 0). Interrupt is disabled.
		1	Interrupt is enabled.

24.4.1.6 McSPI System Register (MCSPI_SYST)

This McSPI system register (MCSPI_SYST) is used to configure the system interconnect either internally to the peripheral bus or externally to the device I/O pads, when the module is configured in the system test (SYSTEST) mode. The MCSPI_SYST is shown in [Figure 24-31](#) and described in [Table 24-16](#).

Figure 24-31. McSPI System Register (MCSPI_SYST)

Reserved															
R/W-0															
Reserved				SSB		SPIENDIR		SPIDATDIR1		SPIDATDIR0					
R/W-0				R/W-0		R/W-0		R/W-0		R/W-0					
Reserved		SPICLK		SPIDAT_1		SPIDAT_0		SPIEN_3		SPIEN_2		SPIEN_1		SPIEN_0	
R-0		R/W-0		R/W-0		R/W-0		R/W-0		R/W-0		R/W-0		R/W-0	

LEGEND: R/W = Read/Write; -n = value after reset

Table 24-16. McSPI System Register (MCSPI_SYST) Field Descriptions

Bit	Field	Value	Description
31-12	Reserved	0	Reads returns 0
11	SSB	0 1	Set status bit. This bit must be cleared prior attempting to clear a status bit of the MCSPI_IRQSTATUS register. No action. Writing 0 does not clear already set status bits. This bit must be cleared prior attempting to clear a status bit of the <MCSPI_IRQSTATUS> register. Writing 1 sets to 1 all status bits contained in the MCSPI_IRQSTATUS register. Writing 1 into this bit sets to 1 all status bits contained in the <MCSPI_IRQSTATUS> register.
10	SPIENDIR	0 1	Sets the direction of the SPIEN[3:0] lines and SPICLK line. Output (as in master mode). Input (as in slave mode).
9	SPIDATDIR1	0 1	Sets the direction of the SPIDAT[1]. Output Input
8	SPIDATDIR0	0 1	Sets the direction of the SPIDAT[0]. Output Input
7	Reserved	0	Reserved
6	SPICLK		SPICLK line (signal data value) If MCSPI_SYST[SPIENDIR] = 1 (input mode direction), this bit returns the value on the CLKSPI line (high or low), and a write into this bit has no effect. If MCSPI_SYST[SPIENDIR] = 0 (output mode direction), the CLKSPI line is driven high or low according to the value written into this register.
5	SPIDAT_1		SPIDAT[1] line (signal data value) If MCSPI_SYST[SPIDATDIR1] = 0 (output mode direction), the SPIDAT[1] line is driven high or low according to the value written into this register. If MCSPI_SYST[SPIDATDIR1] = 1 (input mode direction), this bit returns the value on the SPIDAT[1] line (high or low), and a write into this bit has no effect.
4	SPIDAT_0		SPIDAT[0] line (signal data value) If MCSPI_SYST[SPIDATDIR0] = 0 (output mode direction), the SPIDAT[0] line is driven high or low according to the value written into this register. If MCSPI_SYST[SPIDATDIR0] = 1 (input mode direction), this bit returns the value on the SPIDAT[0] line (high or low), and a write into this bit has no effect.

Table 24-16. McSPI System Register (MCSPI_SYST) Field Descriptions (continued)

Bit	Field	Value	Description
3	SPIEN_3		<p>SPIEN[3] line (signal data value)</p> <p>If MCSPI_SYST[SPIENDIR] = 0 (output mode direction), the SPIENT[3] line is driven high or low according to the value written into this register.</p> <p>If MCSPI_SYST[SPIENDIR] = 1 (input mode direction), this bit returns the value on the SPIEN[3] line (high or low), and a write into this bit has no effect.</p>
2	SPIEN_2		<p>SPIEN[2] line (signal data value)</p> <p>If MCSPI_SYST[SPIENDIR] = 0 (output mode direction), the SPIENT[2] line is driven high or low according to the value written into this register.</p> <p>If MCSPI_SYST[SPIENDIR] = 1 (input mode direction), this bit returns the value on the SPIEN[2] line (high or low), and a write into this bit has no effect.</p>
1	SPIEN_1		<p>SPIEN[1] line (signal data value)</p> <p>If MCSPI_SYST[SPIENDIR] = 0 (output mode direction), the SPIENT[1] line is driven high or low according to the value written into this register.</p> <p>If MCSPI_SYST[SPIENDIR] = 1 (input mode direction), this bit returns the value on the SPIEN[1] line (high or low), and a write into this bit has no effect.</p>
0	SPIEN_0		<p>SPIEN[0] line (signal data value)</p> <p>If MCSPI_SYST[SPIENDIR] = 0 (output mode direction), the SPIENT[0] line is driven high or low according to the value written into this register.</p> <p>If MCSPI_SYST[SPIENDIR] = 1 (input mode direction), this bit returns the value on the SPIEN[0] line (high or low), and a write into this bit has no effect.</p>

24.4.1.7 McSPI Module Control Register (MCSPI_MODULCTRL)

This McSPI module control register (MCSPI_MODULCTRL) is used to configure the serial port interface. The MCSPI_MODULCTRL is shown in [Figure 24-32](#) and described in [Table 24-17](#).

Figure 24-32. McSPI Module Control Register (MCSPI_MODULCTRL)

31	Reserved															16
R/W-0																
15	Reserved													9	8	
R/W-0													FDAA			
R/W-0																
7	6	4		3		2		1		0						
MOA	INITDLY			SYSTEM_TEST		MS		PIN34		SINGLE						
R/W-0		R/W-0			R/W-0		R/W-0		R/W-0		R/W-0					

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 24-17. McSPI Module Control Register(MCSPI_MODULCTRL) Field Descriptions

Bit	Field	Value	Description
31-9	Reserved	0	
8	FDAA	0 1	FIFO DMA Address 256-bit aligned. This register is used when a FIFO is managed by the module and DMA connected to the controller provides only 256 bit aligned address. If this bit is set the enabled channel which uses the FIFO has its datas managed through MCSPI_DAFTX and MCSPI_DAFRX registers instead of MCSPI_TX(i) and MCSPI_RX(i) registers. FIFO data managed by MCSPI_TX(i) and MCSPI_RX(i) registers. FIFO data managed by MCSPI_DAFTX and MCSPI_DAFRX registers.
7	MOA	0 1	Multiple word ocp access. This register can only be used when a channel is enabled using a FIFO. It allows the system to perform multiple SPI word access for a single 32-bit OCP word access. This is possible for WL < 16. Multiple word access disabled Multiple word access enabled with FIFO
6-4	INITDLY	0 1h 2h 3h 4h 5h-Fh	Initial SPI delay for first transfer. This register is an option only available in SINGLE master mode, The controller waits for a delay to transmit the first SPI word after channel enabled and corresponding TX register filled. This Delay is based on SPI output frequency clock, No clock output provided to the boundary and chip select is not active in 4 pin mode within this period. No delay for first SPI transfer The controller wait 4 SPI bus clock The controller wait 8 SPI bus clock The controller wait 16 SPI bus clock The controller wait 32 SPI bus clock Reserved
3	SYSTEM_TEST	0 1	Enables the system test mode Functional mode System test mode (SYSTEST)
2	MS	0 1	Master/ Slave Master - The module generates the SPICLK and SPIEN[3:0] Slave - The module receives the SPICLK and SPIEN[3:0]
1	PIN34	0 1	Pin mode selection. This register is used to configure the SPI pin mode, in master or slave mode. If asserted the controller only use SIMO,SOMI and SPICLK clock pin for SPI transfers. SPIEN is used as a chip select. SPIEN is not used. In this mode all related option to chip select have no meaning.

Table 24-17. McSPI Module Control Register(MCSPI_MODULCTRL) Field Descriptions (continued)

Bit	Field	Value	Description
0	SINGLE		Single channel / Multi Channel (master mode only)
		0	More than one channel will be used in master mode.
		1	Only one channel will be used in master mode. This bit must be set in Force SPIEN mode.

24.4.1.8 McSPI Channel (i) Configuration Register (MCSPI_CH(i)CONF)

The McSPI channel *i* configuration register (MCSPI_CH(i)CONF) is used to configure channel *i*. The (MCSPI_CH(i)CONF) is shown in [Figure 24-33](#) and described in [Table 24-18](#).

Figure 24-33. McSPI Channel (i) Configuration Register (MCSPI_CH(i)CONF)

31	30	29	28	27	26	25	24
Reserved	CLKG	FFER	FFEW	TCS	SBPOL		
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
23	22	21	20	19	18	17	16
SBE	SPIENSLV	FORCE	TURBO	IS	DPE1	DPE0	
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-1	R/W-0
15	14	13	12	11			8
DMAR	DMAW	TRM		WL			
R/W-0	R/W-0	R/W-0		R/W-0			
7	6	5		2	1		0
WL	EPOL	CLKD		POL	PHA		
R/W-0	R/W-0	R/W-0		R/W-0	R/W-0		R/W-0

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 24-18. McSPI Channel (i) Configuration Register (MCSPI_CH(i)CONF) Field Descriptions

Bit	Field	Value	Description
31-30	Reserved	0	Read returns 0
29	CLKG	0 1	Clock divider granularity. This register defines the granularity of channel clock divider: power of two or one clock cycle granularity. When this bit is set the register MCSPI_CHCTRL[EXTCLK] must be configured to reach a maximum of 4096 clock divider ratio. Then The clock divider ratio is a concatenation of MCSPI_CHCONF[CLKD] and MCSPI_CHCTRL[EXTCLK] values. 0 Clock granularity of power of 2 1 1 clock cycle granularity
28	FFER	0 1	FIFO enabled for receive. Only one channel can have this bit set. 0 The buffer is not used to receive data. 1 The buffer is used to receive data.
27	FFEW	0 1	FIFO enabled for transmit. Only one channel can have this bit set. 0 The buffer is not used to transmit data. 1 The buffer is used to transmit data.
26-25	TCS	0 1h 2h 3h	Chip select time control. This 2-bits field defines the number of interface clock cycles between CS toggling and first or last edge of SPI clock. 0 0.5 clock cycles 1h 1.5 clock cycles 2h 2.5 clock cycles 3h 3.5 clock cycles
24	SBPOL	0 1	Start bit polarity. 0 Start bit polarity is held to 0 during SPI transfer. 1 Start bit polarity is held to 1 during SPI transfer.
23	SBE	0 1	Start bit enable for SPI transfer. 0 Default SPI transfer length as specified by WL bit field. 1 Start bit D/CX added before SPI transfer. Polarity is defined by MCSPI_CH(i)CONF[SBPOL].

Table 24-18. McSPI Channel (i) Configuration Register (MCSPI_CH(i)CONF) Field Descriptions (continued)

Bit	Field	Value	Description
22-21	SPIENSLV		Channel 0 only and slave mode only: SPI slave select signal detection. Reserved bits (read returns 0) for other cases.
		0	Detection enabled only on SPIEN[0]
		1h	Detection enabled only on SPIEN[1]
		2h	Detection enabled only on SPIEN[2]
		3h	Detection enabled only on SPIEN[3]
20	FORCE	0	Manual SPIEN assertion to keep SPIEN active between SPI words. (single channel master mode only) Writing 0 into this bit drives the SPIEN line when MCSPI_CHCONF(i)[EPOL]=0, and drives it high when MCSPI_CHCONF(i)[EPOL]=1.
		1	Writing 1 into this bit drives the SPIEN line when MCSPI_CHCONF(i)[EPOL]=0, and drives it low when MCSPI_CHCONF(i)[EPOL]=1
19	TURBO		Turbo mode.
		0	Turbo is deactivated (recommended for single SPI word transfer).
		1	Turbo is activated to maximize the throughput for multi-SPI word transfers.
18	IS		Input select
		0	Data line 0 (SPIDAT[0]) selected for reception.
		1	Data line 1 (SPIDAT[1]) selected for reception.
17	DPE1		Transmission enable for data line 1 (SPIDATAGZEN[1])
		0	Data line 1 (SPIDAT[1]) selected for transmission
		1	No transmission on data line 1 (SPIDAT[1])
16	DPE0		Transmission enable for data line 0 (SPIDATAGZEN[0])
		0	Data line 0 (SPIDAT[0]) selected for transmission
		1	No transmission on data line 0 (SPIDAT[0])
15	DMAR		DMA read request. The DMA read request line is asserted when the channel is enabled and new data is available in the receive register of the channel. The DMA read request line is deasserted on read completion of the receive register of the channel.
		0	DMA read request is disabled.
		1	DMA read request is enabled.
14	DMAW		DMA write request. The DMA write request line is asserted when the channel is enabled and the MCSPI_TXn register of the channel is empty. The DMA write request line is deasserted on load completion of the MCSPI_TXn register of the channel.
		0	DMA write request is disabled.
		1	DMA write request is enabled.
13-12	TRM		Transmit/receive modes.
		0	Transmit and receive mode
		1h	Receive-only mode
		2h	Transmit-only mode
		3h	Reserved

Table 24-18. McSPI Channel (i) Configuration Register (MCSPI_CH(i)CONF) Field Descriptions (continued)

Bit	Field	Value	Description
11-7	WL		SPI word length.
		0	Reserved
		1h	Reserved
		2h	Reserved
		3h	The SPI word is 4-bits long.
		4h	The SPI word is 5-bits long
		5h	The SPI word is 6-bits long
		6h	The SPI word is 7-bits long
		7h	The SPI word is 8-bits long
		8h	The SPI word is 9-bits long
		9h	The SPI word is 10-bits long
		Ah	The SPI word is 11-bits long
		Bh	The SPI word is 12-bits long
		Ch	The SPI word is 13-bits long
		Dh	The SPI word is 14-bits long
		Eh	The SPI word is 15-bits long
		Fh	The SPI word is 16-bits long
		10h	The SPI word is 17-bits long
		11h	The SPI word is 18-bits long
		12h	The SPI word is 19-bits long
		13h	The SPI word is 20-bits long
		14h	The SPI word is 21-bits long
		15h	The SPI word is 22-bits long
		16h	The SPI word is 23-bits long
		17h	The SPI word is 24-bits long
		18h	The SPI word is 25-bits long
		19h	The SPI word is 26-bits long
		1Ah	The SPI word is 27-bits long
		1Bh	The SPI word is 28-bits long
		1Ch	The SPI word is 29-bits long
		1Dh	The SPI word is 30-bits long
		1Eh	The SPI word is 31-bits long
		1Fh	The SPI word is 32-bits long
6	EPOL		SPIEN polarity
		0	SPIEN is held high during the active state.
		1	SPIEN is held low during the active state.

Table 24-18. McSPI Channel (i) Configuration Register (MCSPI_CH(i)CONF) Field Descriptions (continued)

Bit	Field	Value	Description
5-2	CLKD		Frequency divider for SPICLK. (only when the module is a Master SPI device). A programmable clock divider divides the SPI reference clock (CLKSPIREF) with a 4-bit value, and results in a new clock SPICLK available to shift-in and shift-out data. By default the clock divider ratio has a power of two granularity when MCSPI_CHCONF[CLKG] is cleared. Otherwise this register is the 4 LSB bit of a 12-bit register concatenated with clock divider extension MCSPI_CHCTRL[EXTCLK] register. The value description below defines the clock ratio when MCSPI_CHCONF[CLKG] is cleared to 0.
		0	Divide by 1.
		1h	2
		2h	4
		3h	8
		4h	16
		5h	32
		6h	64
		7h	128
		8h	256
		9h	512
		Ah	1024
		Bh	2048
		Ch	4096
		Dh	8192
		Eh	16384
		Fh	32768
1	POL		SPICLK polarity
		0	SPICLK is held high during the active state
		1	SPICLK is held low during the active state
0	PHA		SPICLK phase
		0	Data are latched on odd numbered edges of SPICLK
		1	Data are latched on even numbered edges of SPICLK

Table 24-19. Data Lines Configurations

ISi	DPEi1	DPEi0	TRMi		
			Transmit and Receive	Receive Only	Transmit Only
0	0	0	Supported	Supported	Supported
0	0	1	Supported	Supported	Supported
0	1	0	Supported	Supported	Supported
0	1	1	Not supported (unpredictable result)	Supported	Not supported (unpredictable result)
1	0	0	Supported	Supported	Supported
1	0	1	Supported	Supported	Supported
1	1	0	Supported	Supported	Supported
1	1	1	Not supported (unpredictable result)	Supported	Not supported (unpredictable result)

24.4.1.9 McSPI Channel (i) Status Register (MCSPI_CH(i)STAT)

The McSPI channel *i* status register register (MCSPI_CH(i)STAT) provides status information about the McSPI channel *i* FIFO transmit buffer register (MCSPI_TX*n*) and the McSPI channel *i* FIFO receive buffer register (MCSPI_RX*n*) of channel *i*. The (MCSPI_CH(i)STAT) is shown in [Figure 24-34](#) and described in [Table 24-20](#).

Figure 24-34. McSPI Channel (i) Status Register (MCSPI_CH(i)STAT)

31							8								
Reserved															
R-0															
7		6		5		4		3		2		1		0	
Reserved		RXFFF		RXFFE		TXFFF		TXFFE		EOT		TXS		RXS	
R-0		R-0		R-0		R-0		R-0		R-0		R-0		R-0	

LEGEND: R = Read only; -*n* = value after reset

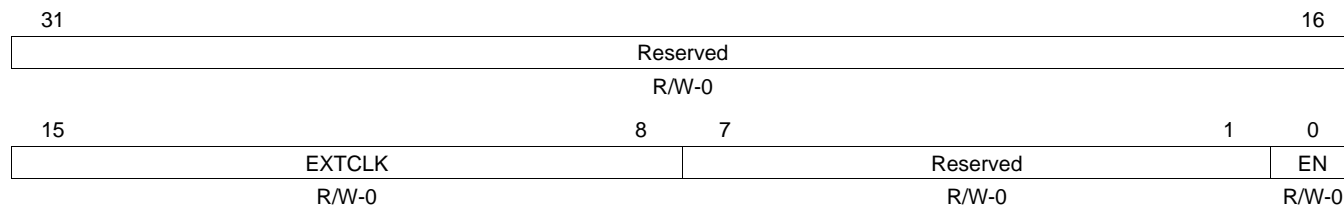
Table 24-20. McSPI Channel (i) Status Register (MCSPI_CH(i)STAT) Field Descriptions

Bit	Field	Value	Description
31-7	Reserved	0	Read returns 0
6	RXFFF	0	Channel <i>i</i> FIFO receive buffer full status.
		0	FIFO receive buffer is not full.
		1	FIFO receive buffer is full.
5	RXFFE	0	Channel <i>i</i> FIFO receive buffer empty status.
		0	FIFO receive buffer is not empty.
		1	FIFO receive buffer is empty.
4	TXFFF	0	Channel <i>i</i> FIFO transmit buffer full status.
		0	FIFO transmit buffer is not full.
		1	FIFO transmit buffer is full.
3	TXFFE	0	Channel <i>i</i> FIFO transmit buffer empty status.
		0	FIFO transmit buffer is not empty.
		1	FIFO transmit buffer is empty.
2	EOT	0	Channel <i>i</i> end-of-transfer status. The definitions of beginning and end of transfer vary with master versus slave and the transfer format (transmit/receive mode, turbo mode).
		0	This flag is automatically cleared when the shift register is loaded with the data from the transmitter register (beginning of transfer).
		1	This flag is automatically set to one at the end of an SPI transfer.
1	TXS	0	Channel <i>i</i> transmitter register status. The bit is cleared when the host writes the most significant byte of the SPI word in the MCSPI_TX(i) register. The bit is set when enabling the channel <i>i</i> , and also when the SPI word is transferred from the MCSPI_TX(i) register to the shift register.
		0	Register is full.
		1	Register is empty.
0	RXS	0	Channel <i>i</i> receiver register status. The bit is cleared when enabling the channel <i>i</i> , and also when the host reads the most significant byte of the received SPI word from the MCSPI_RX(i) register. The bit is set when the received SPI word is transferred from the shift register to the MCSPI_RX(i) register.
		0	Register is empty.
		1	Register is full.

24.4.1.10 McSPI Channel (i) Control Register (MCSPI_CH(I)CTRL)

The McSPI channel *i* control register (MCSPI_CH*i*CTRL) is used to enable channel *i*. The MCSPI_CH*i*CTRL is shown in [Figure 24-35](#) and described in [Table 24-21](#).

Figure 24-35. McSPI Channel (i) Control Register (MCSPI_CH(I)CTRL)



LEGEND: R/W = Read/Write; -*n* = value after reset

Table 24-21. McSPI Channel (i) Control Register (MCSPI_CH(I)CTRL) Field Descriptions

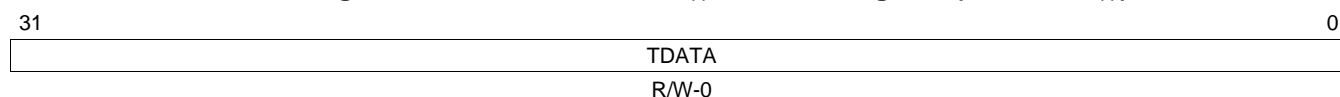
Bit	Field	Value	Description
31-16	Reserved	0	Reserved
15-8	EXTCLK	0	Clock ratio extension. Used to concatenate with the CLKD bit field in MCSPI_CH <i>n</i> CONF for clock ratio only when granularity is 1 clock cycle (CLKG bit in MCSPI_CH <i>n</i> CONF set to 1). Then the maximum value reached is a 4096 clock divider ratio.
		0	Clock ratio is CLKD + 1
		1h	Clock ratio is CLKD + 1 + 16
	
		FFh	Clock ratio is CLKD + 1 + 4080
7-1	Reserved	0	Reserved
0	EN	0	Channel <i>n</i> enable.
		0	Channel <i>n</i> is not active.
		1	Channel <i>n</i> is active.

24.4.1.11 McSPI Channel *i* Transmit Register (MCSPI_TX(*i*))

The McSPI channel *i* transmit register (MCSPI_TX(*i*)) contains a single McSPI word to transmit on the serial link. The (MCSPI_TX(*i*)) is shown in [Figure 24-36](#) and described in [Table 24-22](#).

- Little endian host access SPI 8 bit word on 0; big endian host accesses on 3h.
- The SPI words are transferred with MSB first.

Figure 24-36. McSPI Channel *i* Transmit Register (MCSPI_TX(*i*))



LEGEND: R/W = Read/Write; -*n* = value after reset

Table 24-22. McSPI Channel *i* Transmit Register (MCSPI_TX(*i*)) Field Descriptions

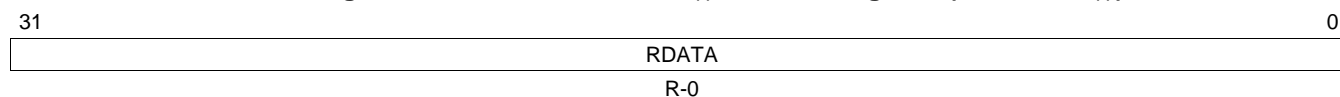
Bit	Field	Value	Description
31-0	TDATA	0-FFFF FFFFh	Channel <i>i</i> data to transmit.

24.4.1.12 McSPI Channel *i* Receive Register (MCSPI_RX(*i*))

The McSPI channel *i* FIFO receive buffer register (MCSPI_RX(*i*)) contains a single McSPI word received through the serial link. The (MCSPI_RX(*i*)) is shown in [Figure 24-37](#) and described in [Table 24-23](#).

- Little endian host access SPI 8 bit word on 0; big endian host accesses on 3h.

Figure 24-37. McSPI Channel *i* Receive Register (MCSPI_RX(*i*))



LEGEND: R = Read only; -*n* = value after reset

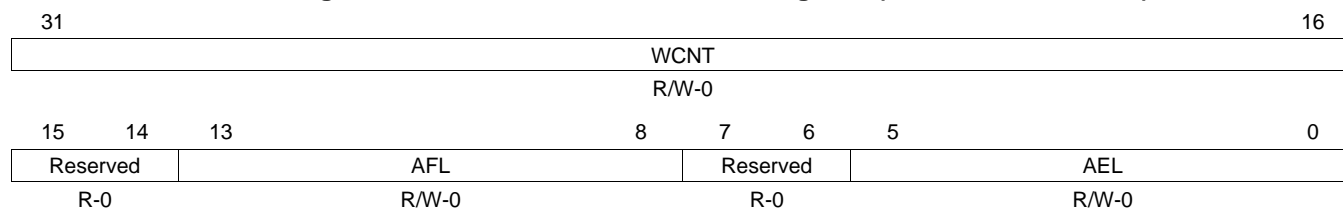
Table 24-23. McSPI Channel *i* Receive Register (MCSPI_RX(*i*)) Field Descriptions

Bit	Field	Value	Description
31-0	RDATA	0-FFFF FFFFh	Channel <i>i</i> received data.

24.4.1.13 McSPI Transfer Levels Register (MCSPI_XFERLEVEL)

The McSPI transfer levels register (MCSPI_XFERLEVEL) provides the transfer levels needed while using the FIFO buffer during transfer. The MCSIPI_XFERLEVEL is shown in [Figure 24-38](#) and described in [Table 24-24](#).

Figure 24-38. McSPI Transfer Levels Register (MCSPI_XFERLEVEL)



LEGEND: R/W = Read/Write; -n = value after reset

Table 24-24. McSPI Transfer Levels Register (MCSPI_XFERLEVEL) Field Descriptions

Bit	Field	Value	Description
31-16	WCNT	<div>0</div> <div>1</div> <div>...</div> <div>FFFEh</div> <div>FFFFh</div>	<div>SPI word counter. Holds the programmable value of the number of SPI words to be transferred on the channel that is using the FIFO buffer. When the transfer has started, a read back of this register returns the current SPI word transfer index.</div> <div>Counter not used</div> <div>1 SPI word</div> <div>...</div> <div>65534 SPI word</div> <div>65535 SPI word</div>
15-14	Reserved	0	Reserved.
13-8	AFL	<div>0</div> <div>1</div> <div>...</div> <div>3Fh</div>	<div>Buffer almost full. Holds the programmable almost full level value used to determine almost full buffer condition. If you want an interrupt or a DMA read request to be issued during a receive operation when the data buffer holds at least <i>n</i> bytes, then the buffer MCSIPI_MODULCTRL[AFL] must be set with <i>n</i> - 1.</div> <div>1 byte</div> <div>2 bytes</div> <div>...</div> <div>64 bytes</div>
7-6	Reserved	0	Reserved.
5-0	AEL	<div>0</div> <div>1</div> <div>...</div> <div>3Fh</div>	<div>Buffer almost empty. Holds the programmable almost empty level value used to determine almost empty buffer condition. If you want an interrupt or a DMA write request to be issued during a transmit operation when the data buffer is able to receive <i>n</i> bytes, then the buffer MCSIPI_MODULCTRL[AEL] must be set with <i>n</i> - 1.</div> <div>1 byte</div> <div>2 bytes</div> <div>...</div> <div>64 bytes</div>

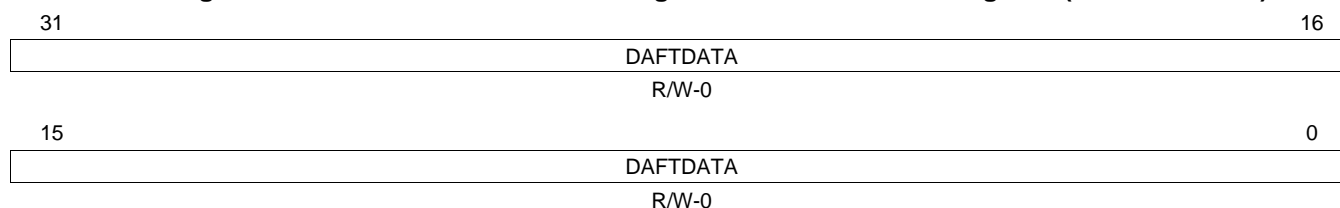
24.4.1.14 McSPI DMA Address Aligned FIFO Transmitter Register (MCSPI_DAFTX)

The McSPI DMA address aligned FIFO transmitter register (MCSPI_DAFTX) contains the SPI words to transmit on the serial link when FIFO is used and the DMA address is aligned on 256 bit. This register is an image of one of the MCSPI_TX(i) registers corresponding to the channel which have its FIFO enabled. The MCSPI_DAFTX register is shown in [Figure 24-39](#) and described in [Table 24-25](#).

NOTE: See Chapter Access to data registers for the list of supported accesses.

The SPI words are transferred with MSB first.

Figure 24-39. McSPI DMA Address Aligned FIFO Transmitter Register (MCSPI_DAFTX)



LEGEND: R/W = Read/Write; -n = value after reset

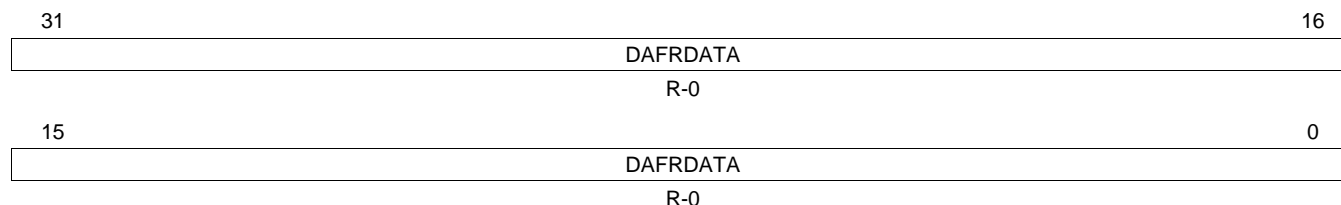
Table 24-25. McSPI DMA Address Aligned FIFO Transmitter Register (MCSPI_DAFTX) Field Descriptions

Bit	Field	Value	Description
31-0	DAFTDATA		FIFO Data to transmit with DMA 256 bit aligned address. This register is used only when MCSPI_MODULCTRL[FDAA] is set to '1', and only one of the MCSPI_CH(i)CONF[FEW] of enabled channels is set. Without these conditions, any access to this register will return a null value.

24.4.1.15 McSPI DMA Address Aligned FIFO Receiver Register (MCSPI_DAFRX)

The McSPI DMA address aligned FIFO receiver register (MCSPI_DAFRX) contains the SPI words to received on the serial link when FIFO used and DMA address is aligned on 256 bit. This register is an image of one of MCSPI_RX(i) register corresponding to the channel which have its FIFO enabled. The MCSPI_DAFRX register is shown in [Figure 24-40](#) and described in [Table 24-26](#).

Figure 24-40. McSPI DMA Address Aligned FIFO Receiver Register (MCSPI_DAFRX)



LEGEND: R/W = Read/Write; -n = value after reset

Table 24-26. McSPI DMA Address Aligned FIFO Receiver Register (MCSPI_DAFRX) Field Descriptions

Bit	Field	Value	Description
31-0	DAFRDATA		FIFO Received Data with DMA 256 bit aligned address. This register is used only when MCSPI_MODULCTRL[FDAA] is set to '1', and only one of the MCSPI_CH(i)CONF[FFER] of enabled channels is set. Without these conditions, any access to this register will return a null value.