SPI

The system features a double buffered Serial Peripheral Interface (SPI). You can configure it to work in all four SPI modes. The default is mode 0.

The SPI controls/monitors the following pins: **`MMISO`**, **`MMOSI`**, **`MSCK`**, **`SCSN`**, **`SMISO`**, **`SMOSI`** and **`SSCK.`**
.

The SPI Master function does not generate any chip select signal. The programmer typically uses another programmable digital I/O to act as chip selects for one or more external SPI Slave devices.

Features

- Double buffered FIFO.
- Full-duplex operation.
- Supports SPI modes 0 through 3.
- Configurable data order on xMISO/xMOSI.
- Four (Master) and six (Slave) interrupt sources.

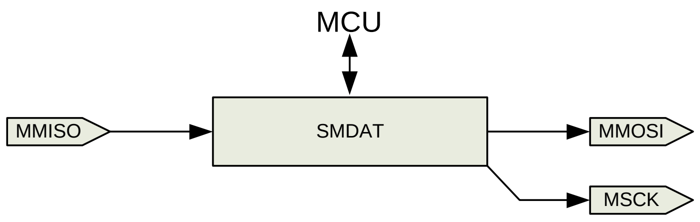

Figure 53. SPI Master.

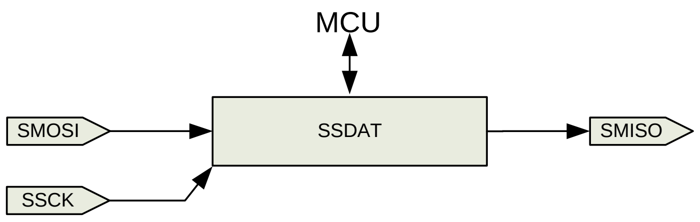

Figure 54. SPI Slave.

Functional Description

SPI Master

The following registers control the SPI Master:

Address (Hex)	Name/mnemonic	Bit	Reset value	Type	Description
0xFC	spiMasterConfig0 SPIMCON0	6:0	0x02	R/W	SPI Master configuration register 0.
	clockFrequency	6:4	010	R/W	Frequency on MSCK. (f_{ckCpu} is the MCU clock frequency.) 000 : 1/2 · f_{ckCpu} 001 : 1/4 · f_{ckCpu} 010 : 1/8 · f_{ckCpu} 011 : 1/16 · f_{ckCpu} 100 : 1/32 · f_{ckCpu} 101 : 1/64 · f_{ckCpu} 110 : 1/64 · f_{ckCpu} 111 : 1/64 · f_{ckCpu} · f_{ckCpu}
	dataOrder	3	0	R/W	Data order (bit wise per byte) on serial output and input (MMOSI and MMISO respectively). 1 : LSBit first, MSBit last. 0 : MSBit first, LSBit last.
	clockPolarity	2	0	R/W	Defines the SPI Master’s operating mode together with SPIMCON0.1. 1 : MSCK is active ‘low’. 0 : MSCK is active ‘high’.
	clockPhase	1	0	R/W	Defines the SPI Master’s operating mode together with SPIMCON0.2. 1 : Sample on trailing edge of MSCK, shift on leading edge. 0 : Sample on leading edge of MSCK, shift on trailing edge.
	spiMasterEnable	0	0	R/W	1 : SPI Master is enabled. The clock to the SPI Master core functionality is running. An SPI transfer can be initiated by the MCU via the 8051 SFR Bus (TX). 0 : SPI Master is disabled. The clock to the SPI Master core functionality stands still.
0xFD	spiMasterConfig1 SPIMCON1	3:0	0x0F	R/W	SPI Master configuration register 1.
	maskIrqRxFifoFull	3	1	R/W	1 : Disable interrupt when RX FIFO is full. 0 : Enable interrupt when RX FIFO is full.
	maskIrqRxDataReady	2	1	R/W	1 : Disable interrupt when data is available in RX FIFO. 0 : Enable interrupt when data is available in RX FIFO.

Address (Hex)	Name/mnemonic	Bit	Reset value	Type	Description
	maskIrqTxFifoE-mpty	1	1	R/W	1 : Disable interrupt when TX FIFO is empty. 0 : Enable interrupt when TX FIFO is empty.
	maskIrqTxFifo-Ready	0	1	R/W	1 : Disable interrupt when a location is available in TX FIFO. 0 : Enable interrupt when a location is available in TX FIFO.
0xFE	spiMasterStatus SPIMSTAT	3:0	0x03	R	SPI Master status register.
	rxFifoFull	3	0	R	Interrupt source. 1 : RX FIFO full. 0 : RX FIFO can accept more data from SPI. Cleared when the cause is removed.
	rxDataReady	2	0	R	Interrupt source. 1 : Data available in RX FIFO. 0 : No data in RX FIFO. Cleared when the cause is removed.
	txFifoEmpty	1	1	R	Interrupt source. 1 : TX FIFO empty. 0 : Data in TX FIFO. Cleared when the cause is removed.
	txFifoReady	0	1	R	Interrupt source. 1 : Location available in TX FIFO. 0 : TX FIFO full. Cleared when the cause is removed.
0xFF	spiMasterData SPIMDAT	7:0	0x00	R/W	SPI Master data register. Accesses TX (write) and RX (read) FIFO buffers, both two bytes deep.

Table 99. SPI Master registers

The SPI Master is configured through SPIMCON0 and SPIMCON1. It is enabled by setting SPIMCON0.0 to ‘1’. The SPI Master supports all four SPI modes, selected by SPIMCON0.2 and SPIMCON0.
The bit wise data order per byte on MMISO/MMOSI is defined by
SPIMCON0.3. MSCK can run on one of six predefined frequencies in the range of 1/2 to 1/64 of the MCU clock frequency, as defined by SPIMCON0.6 down to SPIMCON0.4.

SPIMDAT accesses both the TX (write) and the RX (read) FIFOs, which are two bytes deep. The FIFOs are dynamic and can be refilled according to the state of the status flags: “FIFO ready” means that the FIFO can accept data. “Data ready” means that the FIFO can provide data, minimum one byte.

Four different sources can generate interrupt, unless they are masked by their respective bits in SPIMCON1. SPIMSTAT reveals which sources are active.

SPI Slave

The following registers control the SPI Slave:

Address (Hex)	Name/mnemonic	Bit	Reset value	Type	Description
0xBC	spiSlaveConfig0 SPISCON0	7:0	0xF0	R/W	SPI Slave configuration register 0.
	maskIrqRxFifoFull	7	1	R/W	1 : Disable interrupt when RX FIFO is full. 0 : Enable interrupt when RX FIFO is full.
	maskIrqRxDataR-eady	6	1	R/W	1 : Disable interrupt when data is available in RX FIFO. 0 : Enable interrupt when data is available in RX FIFO.
	maskIrqTxFifoE-mpty	5	1	R/W	1 : Disable interrupt when TX FIFO is empty. 0 : Enable interrupt when TX FIFO is empty.
	maskIrqTxFifo-Ready	4	1	R/W	1 : Disable interrupt when a location is available in TX FIFO. 0 : Enable interrupt when a location is available in TX FIFO.
	dataOrder	3	0	R/W	Data order (bit wise per byte) on serial input and output (SMOSI and SMISO respectively). 1 : LSBit first, MSBit last. 0 : MSBit first, LSBit last.
	clockPolarity	2	0	R/W	Defines the SPI Slave's operating mode together with with SPISCON0.1 . 1 : SSCK is active 'low'. 0 : SSCK is active 'high'.
	clockPhase	1	0	R/W	Defines the SPI Slave's operating mode together with with SPISCON0.2 . 1 : Sample on trailing edge of SSCK, shift on leading edge. 0 : Sample on leading edge of SSCK, shift on trailing edge.
	spiSlaveEnable	0	0	R/W	1 : SPI Slave is enabled. The clock to the SPI Slave core functionality is running. An SPI trans-fer can be initiated by an SPI Master (RX). 0 : SPI Slave is disabled. The clock to the SPI Slave core functionality stands still.
0xBD	spiSlaveConfig1 SPISCON1	7:0	0x0F	R/W	SPI Slave configuration register 1.
	fifoDepth	7:2	0x03	R/W	Depth of RX (TX) FIFO (0 is interpreted as 1). Includes *one status byte + payload*. (TX FIFO does not contain the status byte; it is generated by the SPI Slave.)
	maskIrqScsnHigh	1	1	R/W	1 : Disable interrupt when SCSN goes 'high'. 0 : Enable interrupt when SCSN goes 'high'.
	maskIrqScsnLow	0	1	R/W	1 : Disable interrupt when SCSN goes 'low'. 0 : Enable interrupt when SCSN goes 'low'.
0xBE	spiSlaveStatus SPISSTAT	5:0	0x03	R	SPI Slave status register.

Address (Hex)	Name/mnemonic	Bit	Reset value	Type	Description
	scsnHigh	5	0	R	Interrupt source. 1 : Positive edge of SCSN detected. 0 : Positive edge of SCSN not detected. Cleared when read.
	scsnLow	4	0	R	Interrupt source. 1 : Negative edge of SCSN detected. 0 : Negative edge of SCSN not detected. Cleared when read.
	rxFifoFull	3	0	R	Interrupt source. 1 : RX FIFO full. 0 : RX FIFO can accept more data from SPI. Cleared when the cause is removed.
	rxDataReady	2	0	R	Interrupt source. 1 : Data available in RX FIFO. 0 : No data in RX FIFO. Cleared when the cause is removed.
	txFifoEmpty	1	0	R	Interrupt source. 1 : TX FIFO empty. 0 : Data in TX FIFO. Cleared when the cause is removed.
	txFifoReady	0	0	R	Interrupt source. 1 : Location available in TX FIFO. 0 : TX FIFO full. Cleared when the cause is removed.
0XBF	spiSlaveData SPISDAT	7:0	0x00	R/W	SPI Slave data register. Accesses the RX (read) /TX (write) FIFO buffer.
0xB7	spiSlaveRxData-Size SPISRDSZ	5:0	0x3F	R	SPI Slave RX data size register. Includes one status byte + payload.

Table 100. SPI Slave registers

The SPI Slave is configured through `SPISCON0` and `SPISCON1`. It is enabled by setting `SPISCON0.0` to ‘1’. The SPI Slave supports all four SPI modes, selected by `SPISCON0.2` and `SPISCON0.1`
. The bit wise data order per byte on SMISO/SMOSI is defined by `SPISCON0.3`. There
are six possible interrupt sources in the SPI Slave. Any one of them can be masked.

When an interrupt occurs, SPISSTAT provides information on what the source was.

The FIFO in the SPI Slave is shared by RX and TX data. When the phrase “RX FIFO” (“TX FIFO”) is used, it is merely for focusing on the current usage. The nature of SPI is full-duplex operations. This is not obstructed by the FIFO sharing, since a byte received just replaces a byte transmitted. There are limitations, though:

- The MCU and the SPI Master can only access the FIFO one at the time. The SPI Master is prioritized. This means that if SCSN goes ‘low’ while the MCU fills the TX FIFO, the TX data filling is disrupted. This is also signified by the TX FIFO flags being set to ‘0’.
- When the SPI Master has filled the RX FIFO, and the FIFO is full or SCSN goes ‘high’, the SPI Master can not access the FIFO again until the MCU has read all of the RX data, or flushed the FIFO.
- TX data is no longer available for the SPI Slave after transmission.

"FIFO ready" means that the FIFO can accept data. "Data ready" means that the FIFO can provide data, minimum one byte. "RX FIFO full" means that the RX FIFO is filled to its entire depth as defined by SPISCON1.7 down to SPISCON1.2. "TX FIFO full" means that that the TX FIFO is filled to a depth of minus one byte.

SPISDAT is used for accessing the RX/TX FIFO data buffer. SPISRDSZ provides the RX data size; status byte + payload, that is, the RX FIFO contains one status byte + the payload. The TX FIFO contains only the payload, since the SPI Slave itself generates the status byte to the SPI Master.

The status byte is always the first in an SPI transfer. The status byte from the SPI Master has no affect on the hardware, but should at the least be used for signifying to the MCU if the current RX data can be ignored, as would be the case for a status read out. The status bits from the SPI Master should then be, first to last bit on SSMOSI:

Optional 6:0;
Ignore RX data.

The status byte to the SPI Master is as follows, first to last bit on SSMISO:

Note: This is the sequence the status bit will follow no matter what value *SSCONF0.3* may have. That is, when *SSCONF0.3* = '0' the sequence represents MSB down to LSB, and when *SSCONF0.3* = '1' the sequence represents LSB up to MSB. This applies also to the status bits from the Master, on SSMOSI.

Reserved;
Reserved;
Reserved;
Reserved;
RX FIFO ready;
RX FIFO flushed;
TX data ready;
TX FIFO full.

The status byte to the SPI Master can be read by the SPI Master at any time, but may disrupt a TX FIFO filling by MCU.

Example on how to use the SPI Slave

After reset the SPI Slave is enabled for use by setting SPISCON0.0. The FIFO depth is set to three. For full-duplex operation, the TX payload of two bytes is filled into the FIFO, one byte less than the FIFO depth. It is possible to read the FIFO content if desirable. An interrupt request is sent to the SPI Master through a programmable digital I/O. The four status bits to the SPI Master are '1011'.

The SPI Master starts a transmission by setting SCSN 'low'. Two scenarios may occur:

1. The SPI Master wants to read the status byte and only one payload byte. It provides 16 clock cycles before setting SCSN 'high' again, and bit 0 in the status byte from the Master is set: Ignore RX data (firmware issue). Only the RX data ready flag is set. The MCU can then flush the FIFO by writing one byte to it. Note: The data in the flush write is not stored.

2. The SPI Master reads the status byte and both the data bytes, and transmits at the same time a similar packet to the SPI Slave. The two RX FIFO flags are both set, and the MCU reads one status byte and two data bytes.

The SPI Slave then has nothing to send, and asserts an interrupt request to the SPI Master through a programmable digital I/O without filling the TX FIFO first. In the first case above the four status bits to the SPI Master will be ‘1100’, and in the second case ‘1000’.

In the case of the RX FIFO ready status bit being ‘0’, the SPI Master must wait to initiate a new transmission.

If the TX data ready status flag is set, but TX FIFO full is not, there is no way for the SPI Master to know how many TX data bytes that are valid. In this case the SPI Slave and the SPI Master must agree to use ,for example, the second TX data byte for this information (firmware issue).

A synchronization problem can occur if the SPI Slave is powered down during a transmission. This is best solved at a higher level, for example, an additional interrupt request from the MCU on the chip containing SPI Slave, just before it is powered down.

SPI Timing

The four different SPI Modes are presented in Table 101. SPI modes, Figure 55. and Figure 56..

SPI mode	clockPolarity	clockPhase	Clock shift edge		Clock sample edge	
0	0	0	Trailing	Falling	Leading	Rising
1	0	1	Leading	Rising	Trailing	Falling
2	1	0	Trailing	Rising	Leading	Falling
3	1	1	Leading	Falling	Trailing	Rising

Table 101. SPI modes

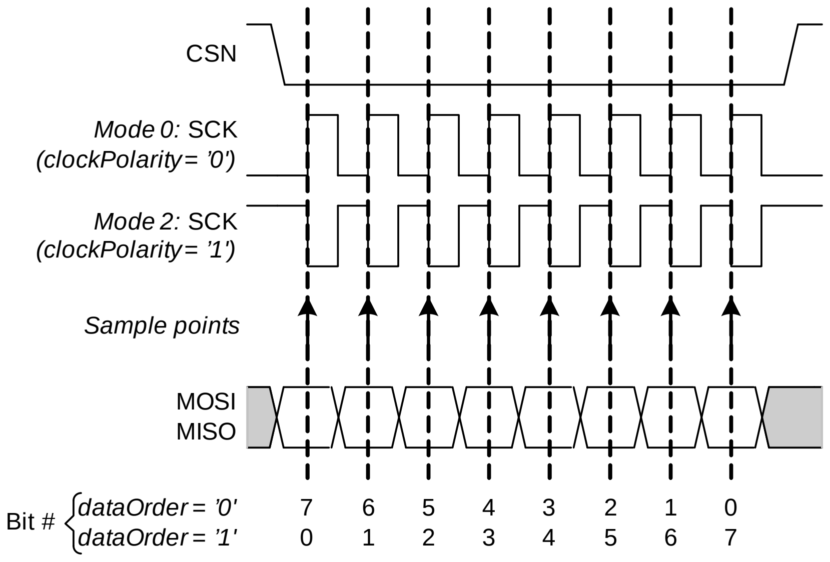

Figure 55. SPI Modes 0 and 2: clockPhase = ‘0’. One byte transmission.

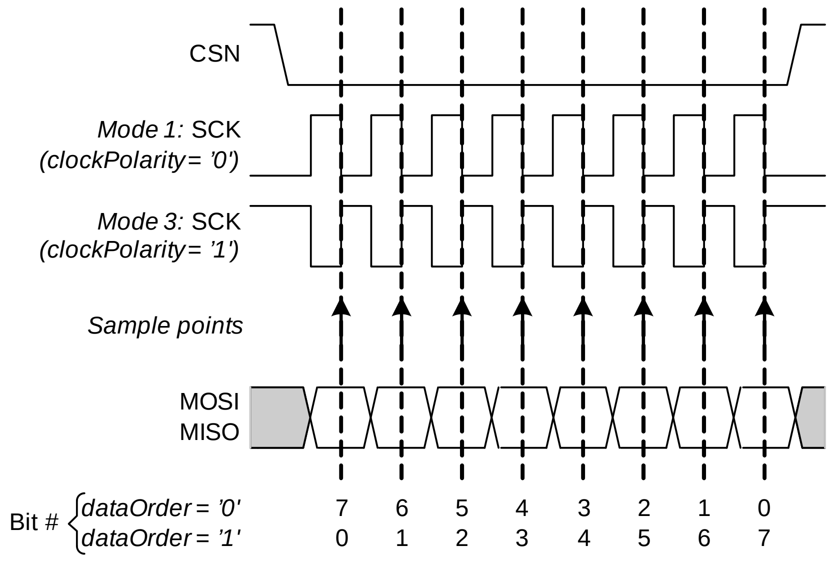

Figure 56. SPI Modes 1 and 3: clockPhase = '1'. One byte transmission.

SPI timing is given in Figure 57. and in Table 102. and Table 103.

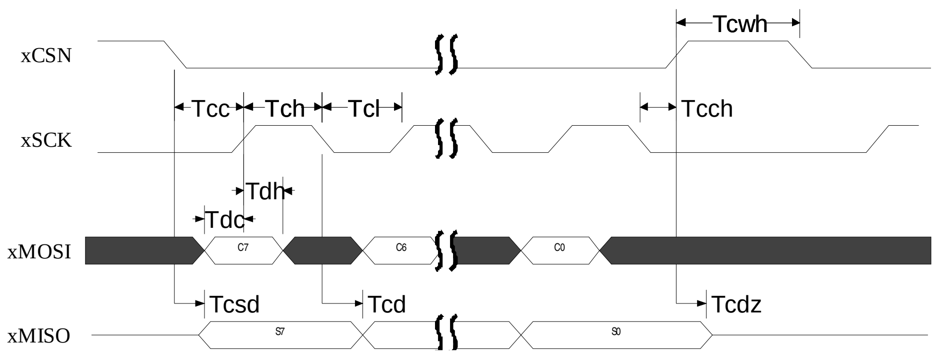

Figure 57. SPI timing diagram. One byte transmission.

Parameters	Symbol	Min	Max	Units
Data to SCK Setup	Tdc	2		ns
SCK to Data Hold	Tdh	2		ns
CSN to Data Valid	Tcsd		38	ns
SCK to Data Valid	Tcd		55	ns
SCK Low Time	Tcl	40		ns
SCK High Time	Tch	40		ns
SCK Frequency	Fsck	0	8	MHz
SCK Rise and Fall	Tr,Tf		100	ns
CSN to SCK Setup	Tcc	2		ns
SCK to CSN Hold	Tcch	2		ns
CSN Inactive time	Tcwh	50		ns
CSN to Output High Z	Tcdz		38	ns

Table 102. SPI timing parameters (C_{Load} = 5pF)

Parameters	Symbol	Min	Max	Units
Data to SCK Setup	Tdc	2		ns
SCK to Data Hold	Tdh	2		ns
CSN to Data Valid	Tcsd		42	ns
SCK to Data Valid	Tcd		58	ns
SCK Low Time	Tcl	40		ns
SCK High Time	Tch	40		ns
SCK Frequency	Fsck	0	8	MHz
SCK Rise and Fall	Tr,Tf		100	ns
CSN to SCK Setup	Tcc	2		ns
SCK to CSN Hold	Tcch	2		ns
CSN Inactive time	Tcwh	50		ns
CSN to Output High Z	Tcdz		42	ns

Table 103. SPI parameters (C_{Load} = 10pF)