Section 20. Serial Peripheral Interface (SPI™)

HIGHLIGHTS

This section of the manual contains the following major topics:

20.1 Introduction

The Serial Peripheral Interface (SPI) module is a synchronous serial interface useful for communicating with other peripheral or microcontroller devices. These peripheral devices may be Serial EEPROMs, shift registers, display drivers, A/D converters, etc. The SPI module is compatible with Motorola's SPI and SIOP interfaces.

Depending on the variant, the dsPIC30F family offers one or two SPI modules on a single device. SPI1 and SPI2 are functionally identical. The SPI2 module is available in many of the higher pin count packages (64-pin and higher), while the SPI1 module is available on all devices.

> **Note:** In this section, the SPI modules are referred together as SPIx or separately as SPI1 and SPI2. Special Function registers will follow a similar notation. For example, SPIxCON refers to the control register for the SPI1 or SPI2 module.

The SPI serial port consists of the following Special Function Registers (SFR):

- SPIxBUF: Address in SFR space that is used to buffer data to be transmitted and data that is received. This address is shared by the SPIxTXB and SPIxRXB registers.
- SPIxCON: A control register that configures the module for various modes of operation.
- SPIxSTAT: A status register that indicates various status conditions.

In addition, there is a 16-bit shift register, SPIxSR, that is not memory mapped. It is used for shifting data in and out of the SPI port.

The memory mapped SFR, SPIxBUF, is the SPI Data Receive/Transmit register. Internally, the SPIxBUF register actually comprises of two separate registers - SPIxTXB and SPIxRXB. The Receive Buffer register, SPIxRXB, and the Transmit Buffer register, SPIxTXB, are two unidirectional 16-bit registers. These registers share the SFR address named SPIxBUF. If a user writes data to be transmitted to the SPIxBUF address, internally the data gets written to the SPIxTXB register. Similarly, when the user reads the received data from SPIxBUF, internally the data is read from the SPIxRXB register. This double-buffering of transmit and receive operations allows continuous data transfers in the background. Transmission and reception occur simultaneously.

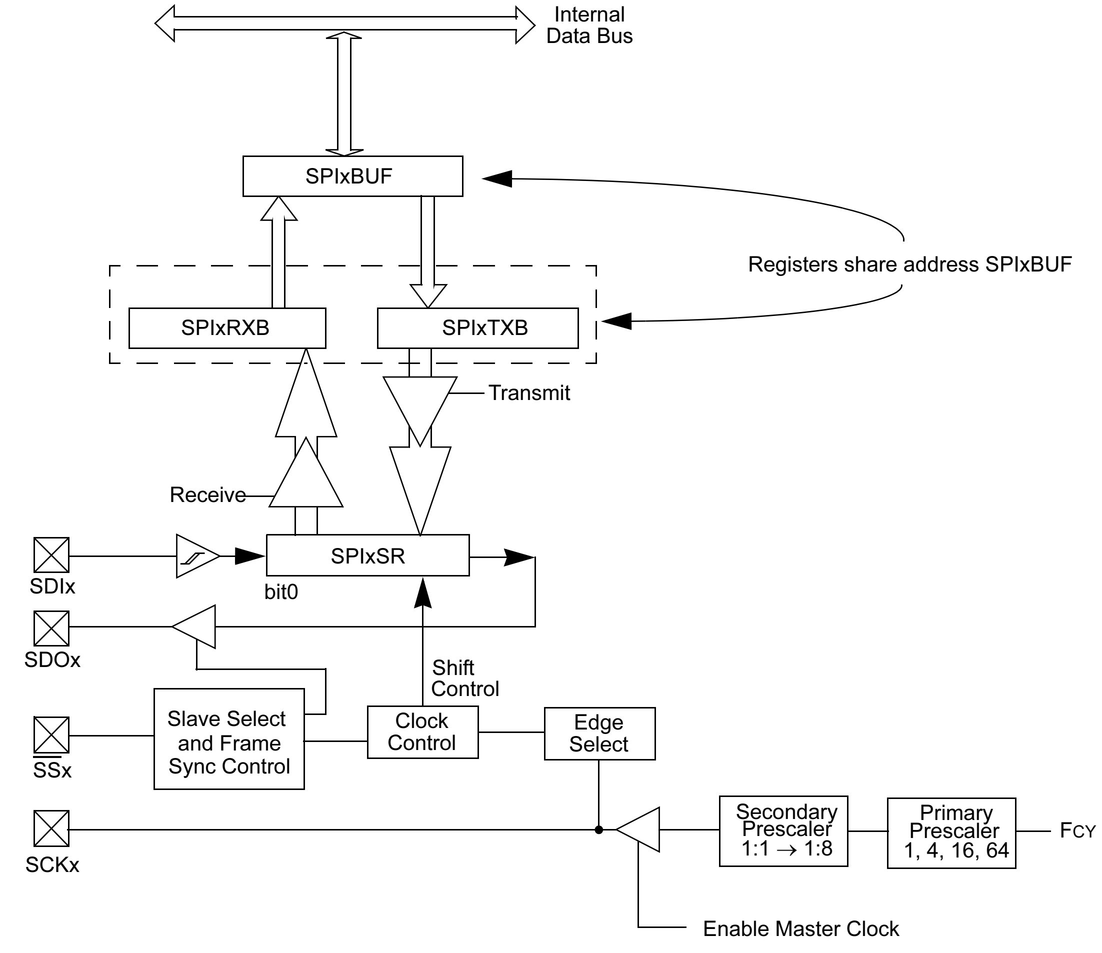
> **Note:** The user cannot write to the SPIxTXB register or read from the SPIxRXB register directly. All reads and writes are performed on the SPIxBUF register.

The SPI serial interface consists of the following four pins:

- SDIx: serial data input
- SDOx: serial data output
- SCKx: shift clock input or output
- $\overline{\text{SSx}}$: active low slave select or frame synchronization I/O pulse

> **Note:** The SPI module can be configured to operate using 3 or 4 pins. In the 3-pin mode, the $\overline{\text{SSx}}$ pin is not used.

Figure 20-1: SPI Module Block Diagram



Note: The SPIxTXB and SPIxRXB registers are memory mapped to the SPIxBUF register.

20.2 Status and Control Registers

Register 20-2: SPIxSTAT: SPI Status and Control Register

Upper Byte:							
R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
SPIEN	—	SPISIDL	—	—	—	—	—
bit 15							bit 8

Lower Byte:							
U-0	R/W-0 HS	U-0	U-0	U-0	U-0	R-0	R-0
—	SPIROV	—	—	—	—	SPITBF	SPIRBF
bit 7							bit 0

bit 15 **SPIEN:** SPI Enable bit
`1` = Enables module and configures SCKx, SDOx, SDIx and SSx as serial port pins
`0` = Disables module

bit 14 **Unimplemented:** Read as ‘0’

bit 13 **SPISIDL:** Stop in Idle Mode bit
`1` = Discontinue module operation when device enters Idle mode
`0` = Continue module operation in Idle mode

bit 12-7 **Unimplemented:** Read as ‘0’

bit 6 **SPIROV:** Receive Overflow Flag bit
`1` = A new byte/word is completely received and discarded. The user software has not read the previous data in the SPIxBUF register.
`0` = No overflow has occurred

bit 5-2 **Unimplemented:** Read as ‘0’

bit 1 **SPITBF:** SPI Transmit Buffer Full Status bit
`1` = Transmit not yet started, SPIxTXB is full
`0` = Transmit started, SPIxTXB is empty
Automatically set in hardware when CPU writes SPIxBUF location, loading SPIxTXB.
Automatically cleared in hardware when SPIx module transfers data from SPIxTXB to SPIxSR.

bit 0 **SPIRBF:** SPI Receive Buffer Full Status bit
`1` = Receive complete, SPIxRXB is full
`0` = Receive is not complete, SPIxRXB is empty
Automatically set in hardware when SPIx transfers data from SPIxSR to SPIxRXB.
Automatically cleared in hardware when core reads SPIxBUF location, reading SPIxRXB.

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
HC = Cleared by Hardware	HS = Set by Hardware		
-n = Value at Reset	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

Register 20-2: SPIxCON: SPI*x* Control Register

Upper Byte:							
U-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	FRMEN	SPIFSD	—	DISSDO	MODE16	SMP	CKE
bit 15							bit 8

Lower Byte:							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SSEN	CKP	MSTEN	SPRE<2:0>			PPRE<1:0>	
bit 7							bit 0

bit 15 **Unimplemented:** Read as ‘0’

bit 14 **FRMEN:** Framed SPI Support bit
1 = Framed SPI support enabled
0 = Framed SPI support disabled

bit 13 **SPIFSD:** Frame Sync Pulse Direction Control on $\overline{\text{SSx}}$ pin bit
1 = Frame sync pulse input (slave)
0 = Frame sync pulse output (master)

bit 12 **Unimplemented:** Read as ‘0’

bit 11 **DISSDO:** Disable SDOx pin bit
1 = SDOx pin is not used by module. Pin is controlled by associated port register.
0 = SDOx pin is controlled by the module

bit 10 **MODE16:** Word/Byte Communication Select bit
1 = Communication is word-wide (16 bits)
0 = Communication is byte-wide (8 bits)

bit 9 **SMP:** SPI Data Input Sample Phase bit
Master mode:
1 = Input data sampled at end of data output time
0 = Input data sampled at middle of data output time
Slave mode:
SMP must be cleared when SPI is used in Slave mode.

bit 8 **CKE:** SPI Clock Edge Select bit
1 = Serial output data changes on transition from active clock state to Idle clock state (see bit 6)
0 = Serial output data changes on transition from Idle clock state to active clock state (see bit 6)

Note: The CKE bit is not used in the Framed SPI modes. The user should program this bit to ‘0’ for the Framed SPI modes (FRMEN = 1).

bit 7 **SSEN:** Slave Select Enable (Slave mode) bit
1 = $\overline{\text{SS}}$ pin used for Slave mode
0 = $\overline{\text{SS}}$ pin not used by module. Pin controlled by port function.

bit 6 **CKP:** Clock Polarity Select bit
1 = Idle state for clock is a high level; active state is a low level
0 = Idle state for clock is a low level; active state is a high level

bit 5 **MSTEN:** Master Mode Enable bit
1 = Master mode
0 = Slave mode

Register 20-2: SPIxCON: SPIx Control Register (Continued)

bit 4-2 **SPRE<2:0>:** Secondary Prescale (Master Mode) bits
(Supported settings: 1:1, 2:1 through 8:1, all inclusive)
`111` = Secondary prescale 1:1
`110` = Secondary prescale 2:1
. . .
`000` = Secondary prescale 8:1

bit 1-0 **PPRE<1:0>:** Primary Prescale (Master Mode) bits
`11` = Primary prescale 1:1
`10` = Primary prescale 4:1
`01` = Primary prescale 16:1
`00` = Primary prescale 64:1

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

20.3 Modes of Operation

The SPI module has flexible Operating modes which are discussed in the following subsections:

- 8-bit and 16-bit Data Transmission/Reception
- Master and Slave Modes
- Framed SPI Modes

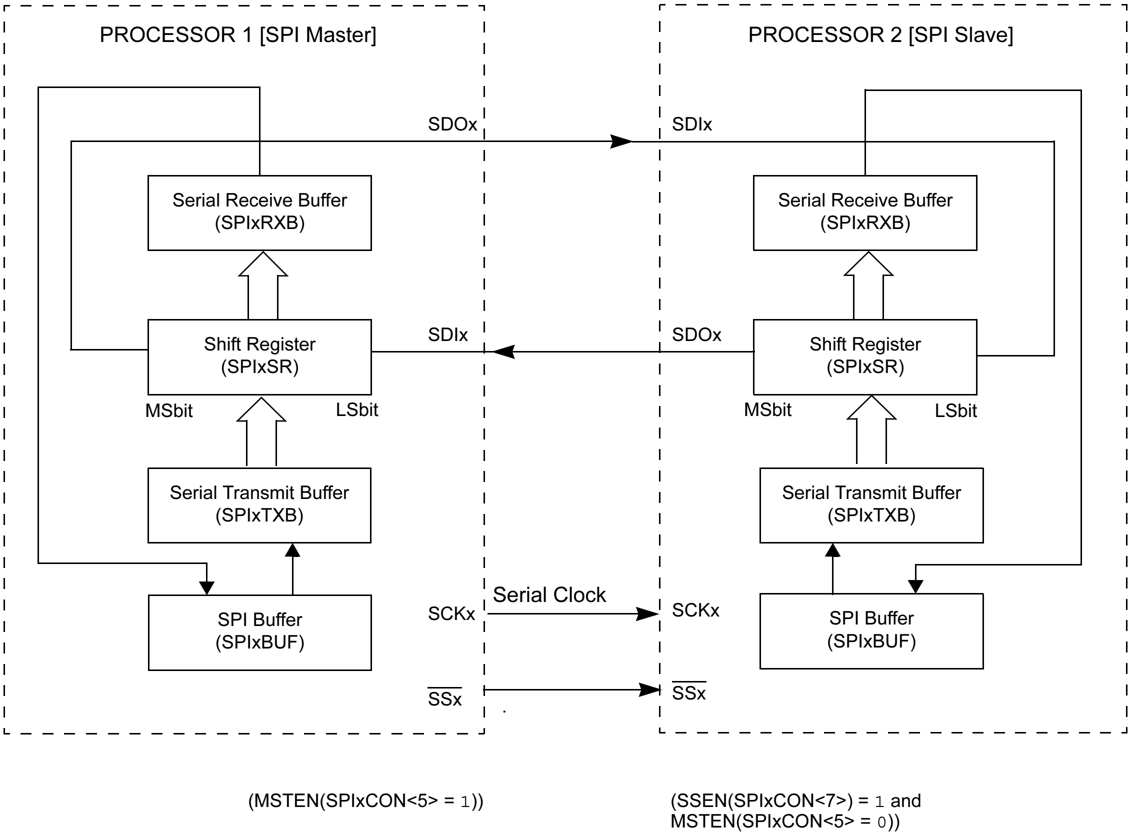
20.3.1 8-bit vs. 16-bit Operation

A control bit, MODE16 (SPIxCON<10>), allows the module to communicate in either 8-bit or 16-bit modes. The functionality will be the same for each mode except the number of bits that are received and transmitted. Additionally, the following should be noted in this context:

- The module is reset when the value of the MODE16 (SPIxCON<10>) bit is changed. Consequently, the bit should not be changed during normal operation.
- Data is transmitted out of bit 7 of the SPIxSR for 8-bit operation while it is transmitted and out of bit 15 of the SPIxSR for 16-bit operation. In both modes, data is shifted into bit ‘0’ of the SPIxSR.
- 8 clock pulses at the SCKx pin are required to shift in/out data in 8-bit mode, while 16 clock pulses are required in the 16-bit mode.

20.3.2 Master and Slave Modes

Figure 20-2: SPI Master/Slave Connection



Note 1: Using the $\overline{\text{SSx}}$ pin in Slave mode of operation is optional.

2: User must write transmit data to/read received data from SPIxBUF. The SPIxTXB and SPIxRXB registers are memory mapped to SPIxBUF.

20.3.2.1 Master Mode

The following steps should be taken to set up the SPI module for the Master mode of operation:

1. If using interrupts:
   - Clear the SPIxIF bit in the respective IFSn register.
   - Set the SPIxIE bit in the respective IECn register.
   - Write the SPIxIP bits in the respective IPCn register.
2. Write the desired settings to the SPIxCON register with MSTEN (SPIxCON<5>) = `1`.
3. Clear the SPIROV bit (SPIxSTAT<6>).
4. Enable SPI operation by setting the SPIEN bit (SPIxSTAT<15>).
5. Write the data to be transmitted to the SPIxBUF register. Transmission (and Reception) will start as soon as data is written to the SPIxBUF register.

In Master mode, the system clock is prescaled and then used as the serial clock. The prescaling is based on the settings in the PPRE<1:0> (SPIxCON<1:0>) and SPRE<1:0> (SPIxCON<4:2>) bits. The serial clock is output via the SCKx pin to slave devices. Clock pulses are only generated when there is data to be transmitted. For further information, refer to **Section 20.4 “SPI Master Mode Clock Frequency”**.

The CKP and CKE bits determine on which edge of the clock, data transmission occurs.

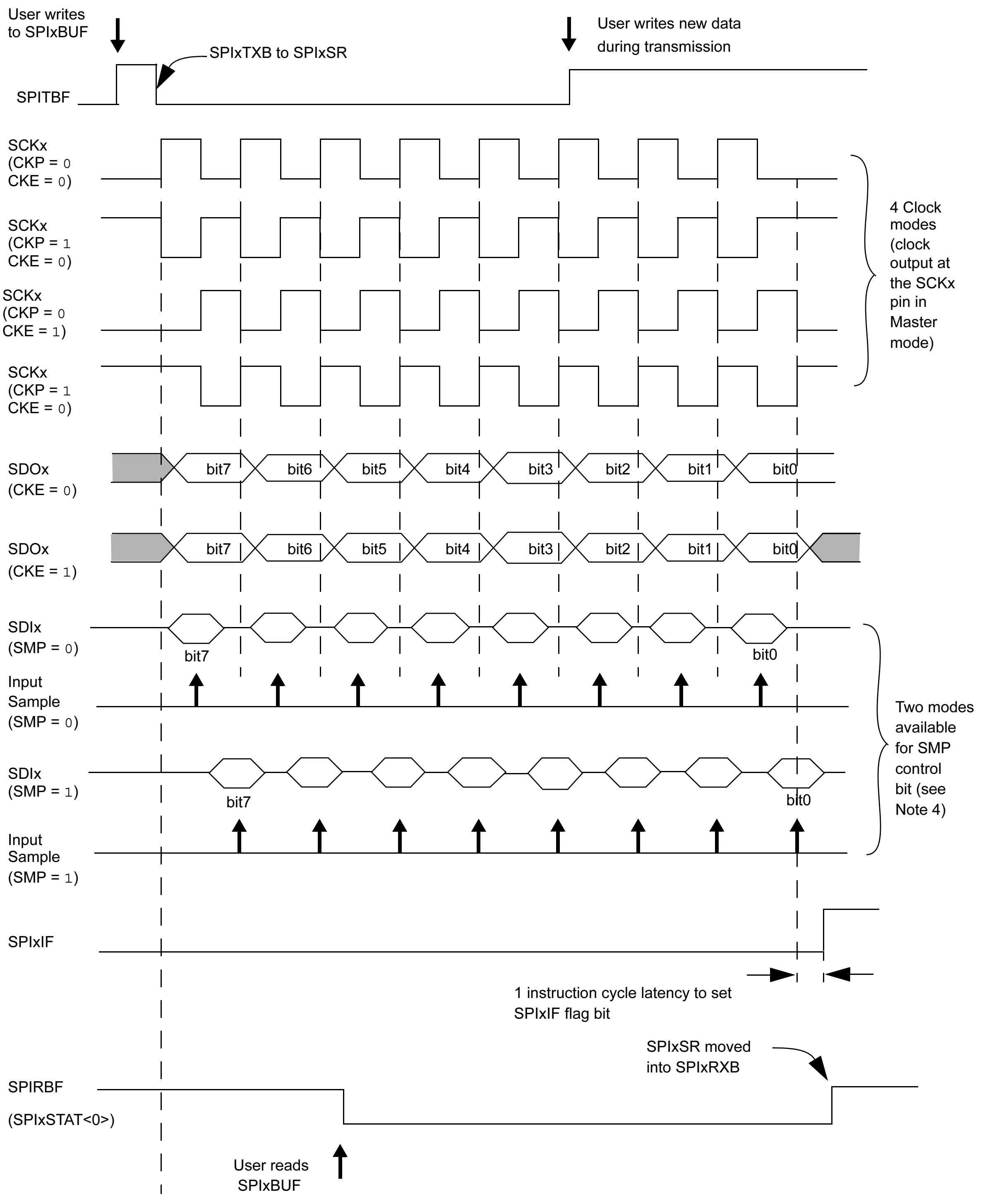
Both data to be transmitted and data that is received are respectively written into or read from the SPIxBUF register.

The following describes the SPI module operation in Master mode:

1. Once the module is set up for Master mode of operation and enabled, data to be transmitted is written to the SPIxBUF register. The SPITBF (SPIxSTAT<1>) bit is set.
2. The contents of SPIxTXB are moved to the shift register, SPIxSR, and the SPITBF bit is cleared by the module.
3. A series of 8/16 clock pulses shifts out 8/16 bits of transmit data from the SPIxSR to the SDOx pin and simultaneously shifts in the data at the SDIx pin into the SPIxSR.
4. When the transfer is complete, the following events will occur:
   - The interrupt flag bit, SPIxIF, is set. SPI interrupts can be enabled by setting the interrupt enable bit SPIxIE. The SPIxIF flag is not cleared automatically by the hardware.
   - Also, when the ongoing transmit and receive operation is completed, the contents of the SPIxSR are moved to the SPIxRXB register.
   - The SPIRBF (SPIxSTAT<0>) bit is set by the module, indicating that the receive buffer is full. Once the SPIxBUF register is read by the user code, the hardware clears the SPIRBF bit.
5. If the SPIRBF bit is set (receive buffer is full) when the SPI module needs to transfer data from SPIxSR to SPIxRXB, the module will set the SPIROV (SPIxSTAT<6>) bit, indicating an overflow condition.
6. Data to be transmitted can be written to SPIxBUF by the user software at any time as long as the SPITBF (SPIxSTAT<1>) bit is clear. The write can occur while SPIxSR is shifting out the previously written data, allowing continuous transmission.

Note: The SPIxSR register cannot be written into directly by the user. All writes to the SPIxSR register are performed through the SPIxBUF register.

Figure 20-3: SPI Master Mode Operation


User writes to SPIxBUF
SPIxTXB to SPIxSR
User writes new data during transmission
SPITBF
SCKx (CKP = 0 CKE = 0)
SCKx (CKP = 1 CKE = 0)
SCKx (CKP = 0 CKE = 1)
SCKx (CKP = 1 CKE = 0)
4 Clock modes (clock output at the SCKx pin in Master mode)
SDOx (CKE = 0)
bit7 bit6 bit5 bit4 bit3 bit2 bit1 bit0
SDOx (CKE = 1)
bit7 bit6 bit5 bit4 bit3 bit2 bit1 bit0
SDIx (SMP = 0)
bit7
bit0
Input Sample (SMP = 0)
SDIx (SMP = 1)
bit7
bit0
Input Sample (SMP = 1)
Two modes available for SMP control bit (see Note 4)
SPIxIF
1 instruction cycle latency to set SPIxIF flag bit
SPIxSR moved into SPIxRXB
SPIRBF
(SPIxSTAT<0>)
User reads SPIxBUF

Note 1: Four SPI Clock modes shown to demonstrate CKP (SPIxCON<6>) and CKE (SPIxCON<8>) bit functionality only. Only one of the four modes can be chosen for operation.

2: SDI and input sample shown for two different values of the SMP (SPIxCON<9>) bit, for demonstration purposes only. Only one of the two configurations of the SMP bit can be chosen during operation.

3: If there are no pending transmissions, SPIxTXB is transferred to SPIxSR as soon as the user writes to SPIxBUF.

4: Operation for 8-bit mode shown. The 16-bit mode is similar.

20.3.2.2 Slave Mode

The following steps should be taken to set up the SPI module for the Slave mode of operation:

1. Clear the SPIxBUF register.
2. If using interrupts:
   - Clear the SPIxIF bit in the respective IFSn register.
   - Set the SPIxIE bit in the respective IECn register.
   - Write the SPIxIP bits in the respective IPCn register.
3. Write the desired settings to the SPIxCON register with MSTEN (SPIxCON<5>) = `0`.
4. Clear the SMP bit.
5. If the CKE bit is set, then the SSEN bit must be set, thus enabling the $\overline{\text{SSx}}$ pin.
6. Clear the SPIROV bit (SPIxSTAT<6>) and,
7. Enable SPI operation by setting the SPIEN bit (SPIxSTAT<15>).

In Slave mode, data is transmitted and received as the external clock pulses appear on the SCKx pin. The CKP (SPIxCON<6>) and CKE (SPIxCON<8>) bits determine on which edge of the clock data transmission occurs.

Both data to be transmitted and data that is received are respectively written into or read from the SPIxBUF register.

The rest of the operation of the module is identical to that in the Master mode.

A few additional features provided in the Slave mode are:

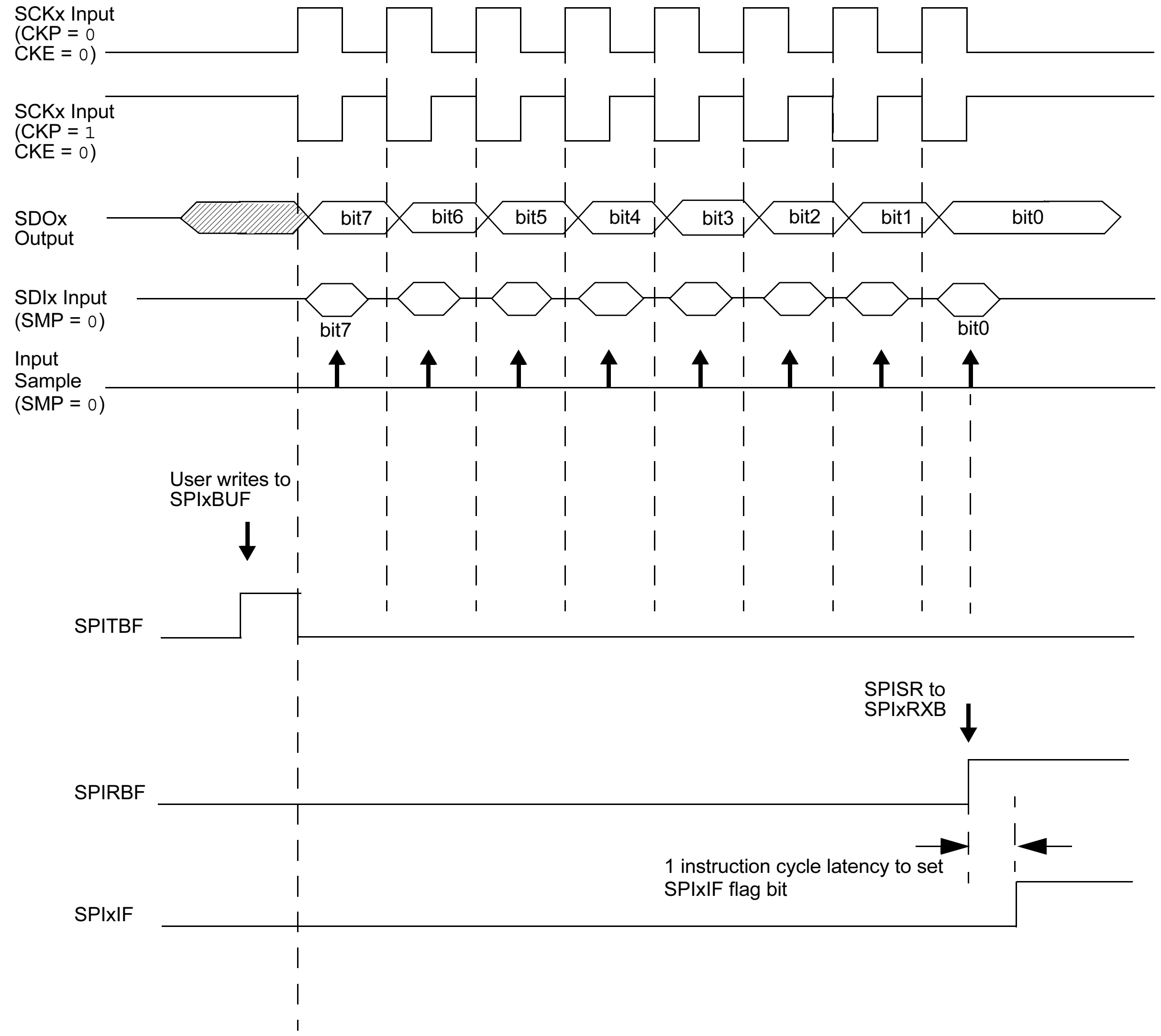
Slave Select Synchronization: The $\overline{\text{SSx}}$ pin allows a Synchronous Slave mode. If the SSEN (SPIxCON<7>) bit is set, transmission and reception is enabled in Slave mode only if the $\overline{\text{SSx}}$ pin is driven to a low state. The port output or other peripheral outputs must not be driven in order to allow the $\overline{\text{SSx}}$ pin to function as an input. If the SSEN bit is set and the $\overline{\text{SSx}}$ pin is driven high, the SDOx pin is no longer driven and will tri-state even if the module is in the middle of a transmission. An aborted transmission will be retried the next time the $\overline{\text{SSx}}$ pin is driven low using the data held in the SPIxTXB register. If the SSEN bit is not set, the $\overline{\text{SSx}}$ pin does not affect the module operation in Slave mode.

SPITBF Status Flag Operation: The function of the SPITBF (SPIxSTAT<1>) bit is different in the Slave mode of operation. The following describes the function of the SPITBF for various settings of the Slave mode of operation:

1. If SSEN (SPIxCON<7>) is cleared, the SPITBF is set when the SPIxBUF is loaded by the user code. It is cleared when the module transfers SPIxTXB to SPIxSR. This is similar to the SPITBF bit function in Master mode.
2. If SSEN (SPIxCON<7>) is set, the SPITBF is set when the SPIxBUF is loaded by the user code. However, it is cleared only when the SPIx module completes data transmission. A transmission will be aborted when the $\overline{\text{SSx}}$ pin goes high and may be retried at a later time. Each data word is held in SPIxTXB until all bits are transmitted to the receiver.

> **Note:** To meet module timing requirements, the $\overline{\text{SSx}}$ pin must be enabled in Slave mode when CKE = `1`. (Refer to Figure 20-6 for details.)

Figure 20-4: SPI Slave Mode Operation: Slave Select Pin Disabled

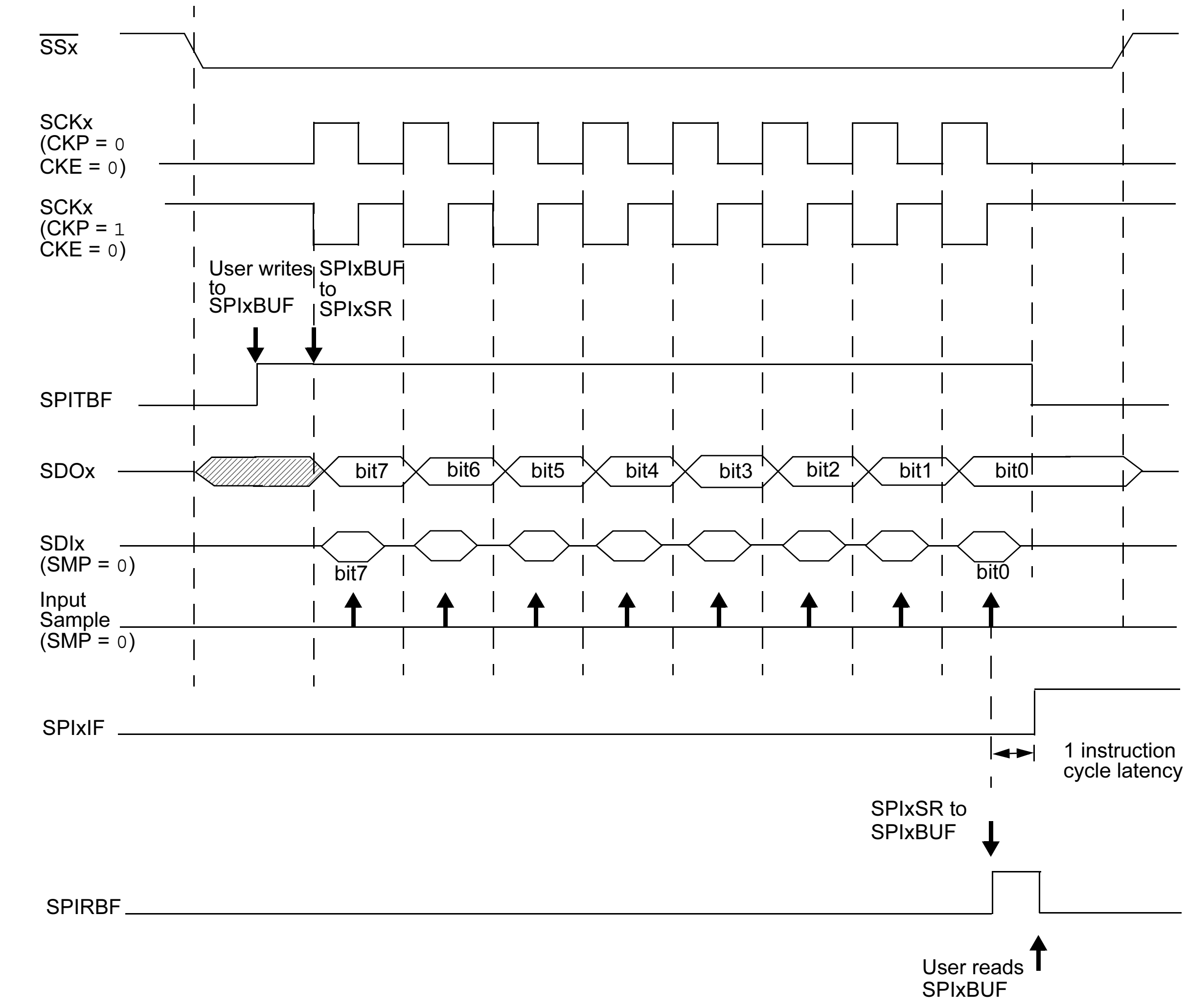


Note 1: Two SPI Clock modes shown only to demonstrate CKP (SPIxCON<6>) and CKE (SPIxCON<8>) bit functionality. Any combination of CKP and CKE bits can be chosen for module operation.

2: If there are no pending transmissions or a transmission in progress, SPIxBUF is transferred to SPIxSR as soon as the user writes to SPIxBUF.

3: Operation for 8-bit mode shown. The 16-bit mode is similar.

Figure 20-5: SPI Slave Mode Operation with Slave Select Pin Enabled

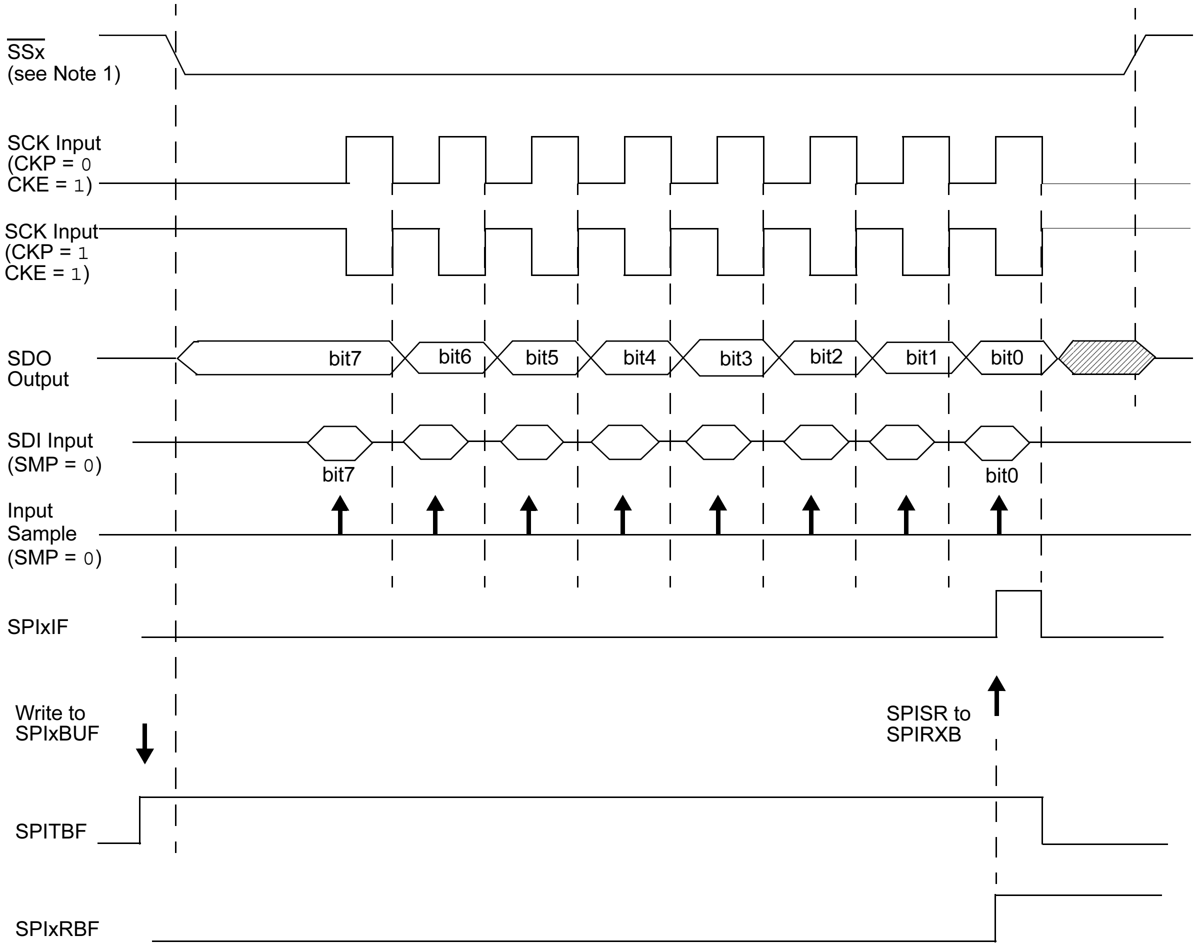


Note 1: When the SSEN (SPIxCON<7>) bit is set to '1', the $\overline{\text{SSx}}$ pin must be driven low so as to enable transmission and reception in Slave mode.

2: Transmit data is held in SPIxTXB and SPITBF remains set until all bits are transmitted.

3: Operation for 8-bit mode shown. The 16-bit mode is similar.

Figure 20-6: SPI Mode Timing (Slave Mode w/CKE = 1)



Note 1: The SSx pin must be used for Slave mode operation when CKE = 1.

2: When the SSEN (SPIxCON<7>) bit is set to '1', the SSx pin must be driven low so as to enable transmission and reception in Slave mode.

3: Transmit data is held in SPIxTXB and SPITBF remains set until all bits are transmitted.

4: Operation for 8-bit mode shown. The 16-bit mode is similar.

20.3.3 SPI Error Handling

When a new data word has been shifted into SPIxSR and the previous contents of SPIxRXB have not been read by the user software, the SPIROV bit (SPIxSTAT<6>) will be set. The module will not transfer the received data from SPIxSR to SPIxRXB. Further data reception is disabled until the SPIROV bit is cleared. The SPIROV bit is not cleared automatically by the module and must be cleared by the user software.

20.3.4 SPI Receive Only Operation

Setting the control bit, DISSDO (SPIxCON<11>), disables transmission at the SDOx pin. This allows the SPIx module to be configured for a Receive Only mode of operation. The SDOx pin will be controlled by the respective port function if the DISSDO bit is set.

The DISSDO function is applicable to all SPI Operating modes.

20.3.5 Framed SPI Modes

The module supports a very basic framed SPI protocol while operating in either Master or Slave modes. The following features are provided in the SPI module to support Framed SPI modes:

- The control bit, FRMEN (SPIxCON<14>), enables Framed SPI modes and causes the $\overline{\text{SSx}}$ pin to be used as a frame synchronization pulse input or output pin. The state of the SSEN (SPIxCON<7>) is ignored.
- The control bit, SPIFSD (SPIxCON<13>), determines whether the $\overline{\text{SSx}}$ pin is an input or an output (i.e., whether the module receives or generates the frame synchronization pulse).
- The frame synchronization pulse is an active high pulse for a single SPI clock cycle.

The following two framed SPI modes are supported by the SPI module:

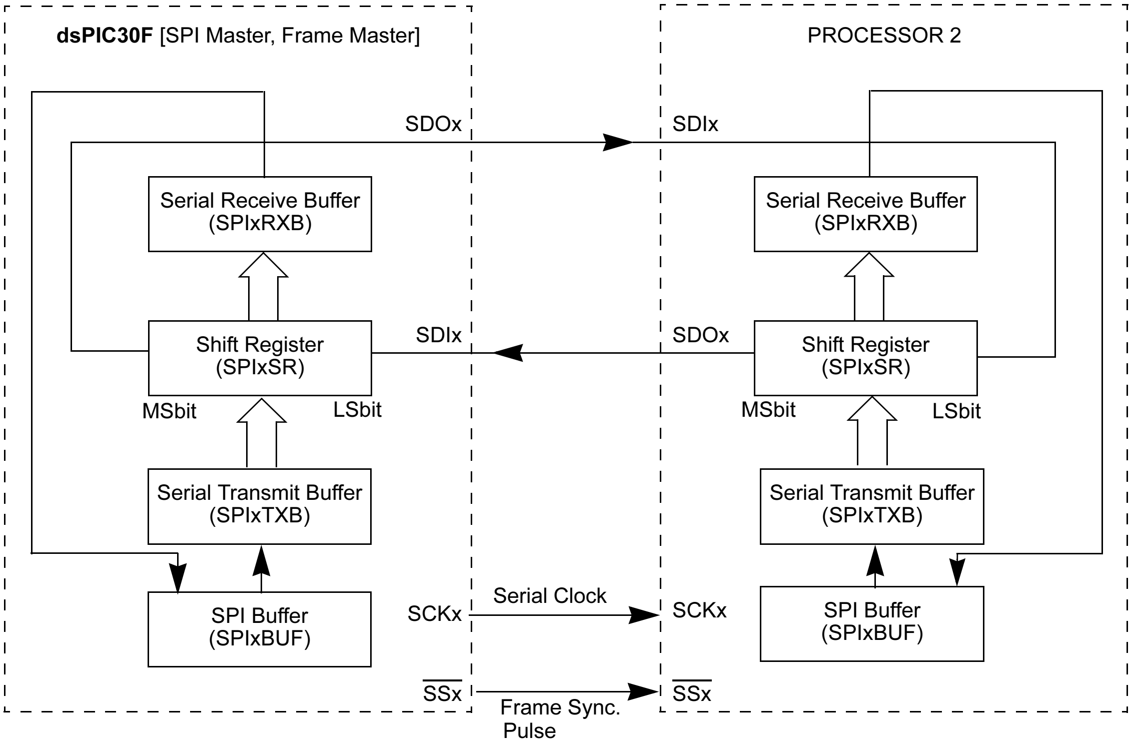
- Frame Master Mode: The SPI module generates the frame synchronization pulse and provides this pulse to other devices at the $\overline{\text{SSx}}$ pin.
- Frame Slave Mode: The SPI module uses a frame synchronization pulse received at the $\overline{\text{SSx}}$ pin.

The Framed SPI modes are supported in conjunction with the Master and Slave modes. Thus, the following four framed SPI configurations are available to the user:

- SPI Master Mode and Frame Master Mode
- SPI Master Mode and Frame Slave Mode
- SPI Slave Mode and Frame Master Mode
- SPI Slave Mode and Frame Slave Mode

These four modes determine whether or not the SPIx module generates the serial clock and the frame synchronization pulse.

Figure 20-7: SPI Master, Frame Master Connection Diagram



Note 1: In Framed SPI modes, the $\overline{SSx}$ pin is used to transmit/receive the frame synchronization pulse.
2: Framed SPI modes require the use of all four pins (i.e., using the $\overline{SSx}$ pin is not optional).
3: The SPIxTXB and SPIxRXB registers are memory mapped to the SPIxBUF register.

20.3.5.1 SCKx in Framed SPI Modes

When FRMEN (SPIxCON<14>) = 1 and MSTEN (SPIxCON<5>) = 1, the SCKx pin becomes an output and the SPI clock at SCKx becomes a free running clock.

When FRMEN = 1 and MSTEN = 0, the SCKx pin becomes an input. The source clock provided to the SCKx pin is assumed to be a free running clock.

The polarity of the clock is selected by the CKP (SPIxCON<6>) bit. The CKE (SPIxCON<8>) bit is not used for the Framed SPI modes and should be programmed to '0' by the user software.

When CKP = 0, the frame sync pulse output and the SDOx data output change on the rising edge of the clock pulses at the SCKx pin. Input data is sampled at the SDIx input pin on the falling edge of the serial clock.

When CKP = 1, the frame sync pulse output and the SDOx data output change on the falling edge of the clock pulses at the SCKx pin. Input data is sampled at the SDIx input pin on the rising edge of the serial clock.

20.3.5.2 SPIx Buffers in Framed SPI Modes

When SPIFSD (SPIxCON<13>) = 0, the SPIx module is in the Frame Master mode of operation. In this mode, the frame sync pulse is initiated by the module when the user software writes the transmit data to SPIxBUF location (thus writing the SPIxTXB register with transmit data). At the end of the frame sync pulse, the SPIxTXB is transferred to the SPIxSR and data transmission/reception begins.

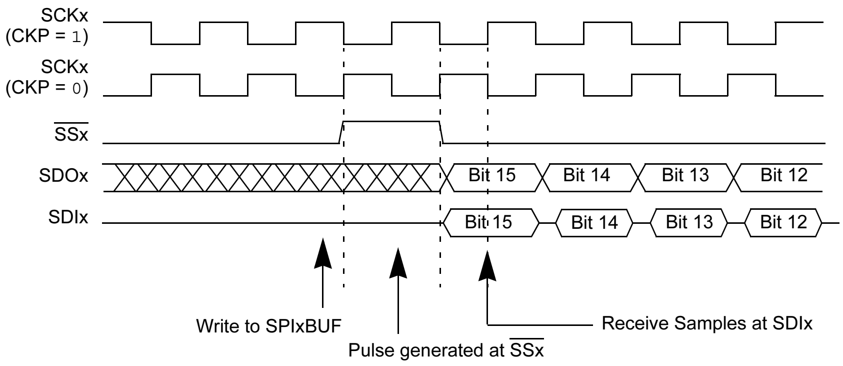
When SPIFSD (SPIxCON<13>) = 1, the module is in Frame Slave mode. In this mode, the frame sync pulse is generated by an external source. When the module samples the frame sync pulse, it will transfer the contents of the SPIxTXB register to the SPIxSR and data transmission/reception begins. The user must make sure that the correct data is loaded into the SPIxBUF for transmission before the frame sync pulse is received.

Note: Receiving a frame sync pulse will start a transmission, regardless of whether data was written to SPIxBUF. If no write was performed, the old contents of SPIxTXB will be transmitted.

20.3.5.3 SPI Master Mode and Frame Master Mode

This Framed SPI mode is enabled by setting the MSTEN (SPIxCON<5>) and FRMEN (SPIxCON<14>) bits to ‘1’ and the SPIFSD (SPIxCON<13>) bit to ‘0’. In this mode, the serial clock will be output continuously at the SCKx pin, regardless of whether the module is transmitting. When the SPIxBUF is written, the SSx pin will be driven high on the next transmit edge of the SCKx clock. The SSx pin will be high for one SCKx clock cycle. The module will start transmitting data on the next transmit edge of the SCKx, as shown in Figure 20-8. A connection diagram indicating signal directions for this Operating mode is shown in Figure 20-7.

Figure 20-8: SPI Master, Frame Master



20.3.5.4 SPI Master Mode and Frame Slave Mode

This Framed SPI mode is enabled by setting the MSTEN, FRMEN and the SPIFSD bits to ‘1’. The $\overline{\mathrm{SSx}}$ pin is an input, and it is sampled on the sample edge of the SPI clock. When it is sampled high, data will be transmitted on the subsequent transmit edge of the SPI clock, as shown in Figure 20-9. The interrupt flag, SPIxIF, is set when the transmission is complete. The user must make sure that the correct data is loaded into the SPIxBUF for transmission before the signal is received at the $\overline{\mathrm{SSx}}$ pin. A connection diagram indicating signal directions for this Operating mode is shown in Figure 20-10.

Figure 20-9: SPI Master, Frame Slave

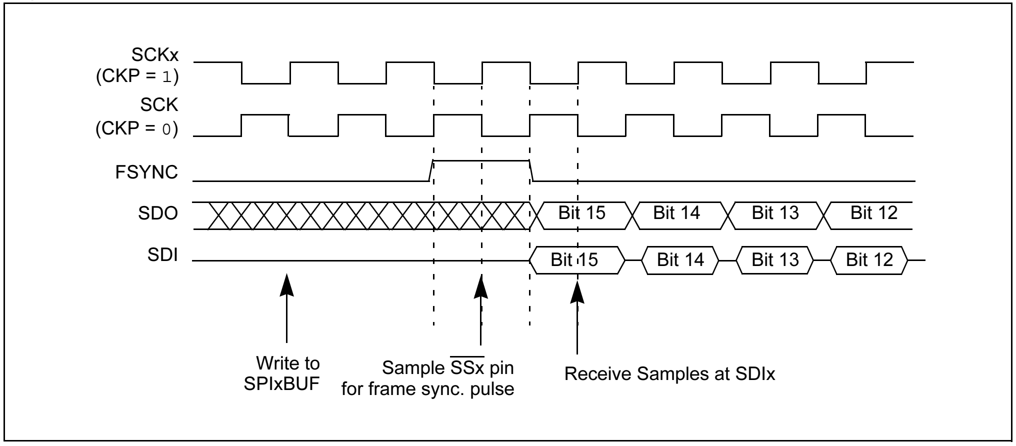
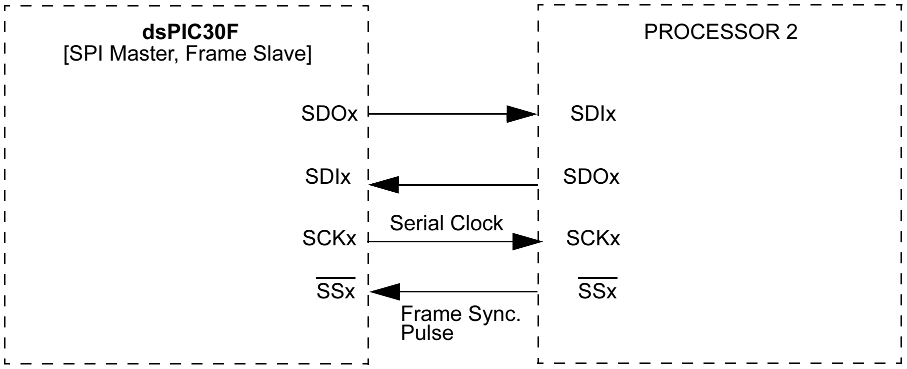


Figure 20-10: SPI Master, Frame Slave Connection Diagram


dsPIC30F
[SPI Master, Frame Slave]
SDOx
SDIx
SCKx
$\overline{\mathrm{SSx}}$
PROCESSOR 2
SDIx
SDOx
SCKx
$\overline{\mathrm{SSx}}$
Serial Clock
Frame Sync. Pulse

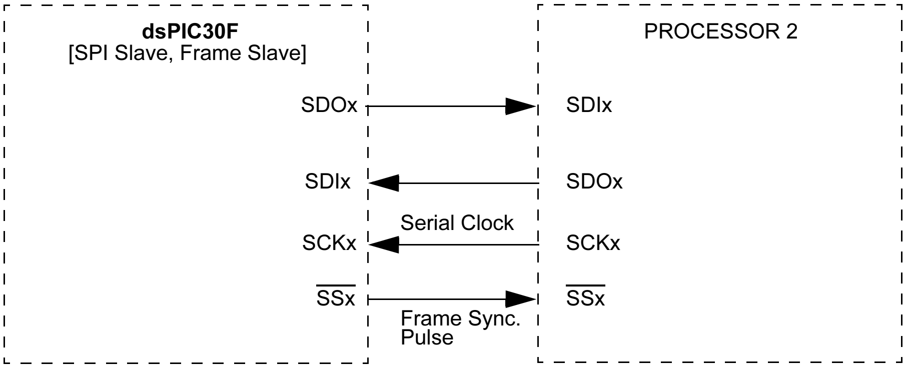
Note 1: In Framed SPI modes, the $\overline{\mathrm{SSx}}$ pin is used to transmit/receive the frame synchronization pulse.

2: Framed SPI modes require the use of all four pins (i.e., Using the $\overline{\mathrm{SSx}}$ pin is not optional).

20.3.5.5 SPI Slave Mode and Frame Master Mode

This framed SPI mode is enabled by setting the MSTEN (SPIxCON<5>) bit to ‘0’, the FRMEN (SPIxCON<14>) bit to ‘1’ and the SPIFSD (SPIxCON<13>) bit to ‘0’. The input SPI clock will be continuous in Slave mode. The $\overline{\text{SSx}}$ pin will be an output when the SPIFSD bit is low. Therefore, when the SPIBUF is written, the module will drive the $\overline{\text{SSx}}$ pin high on the next transmit edge of the SPI clock. The $\overline{\text{SSx}}$ pin will be driven high for one SPI clock cycle. Data will start transmitting on the next SPI clock transmit edge. A connection diagram indicating signal directions for this Operating mode is shown in Figure 20-11.

Figure 20-11: SPI Slave, Frame Master Connection Diagram



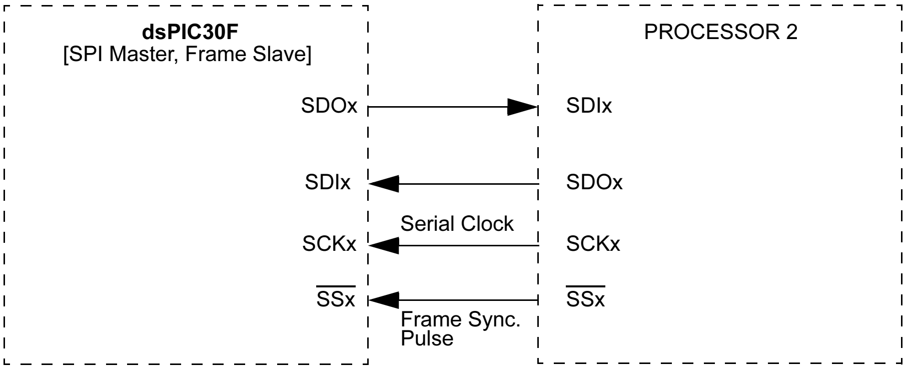
Note 1: In Framed SPI modes, the $\overline{\text{SSx}}$ pin is used to transmit/receive the frame synchronization pulse.

2: Framed SPI modes require the use of all four pins (i.e., Using the $\overline{\text{SSx}}$ pin is not optional).

20.3.5.6 SPI Slave Mode and Frame Slave Mode

This Framed SPI mode is enabled by setting the MSTEN (SPIxCON<5>) bit to ‘0’, the FRMEN bit (SPIxCON<14>) to ‘1’ and the SPIFSD (SPIxCON<13>) bit to ‘1’. Therefore, both the SCKx and $\overline{\text{SSx}}$ pins will be inputs. The $\overline{\text{SSx}}$ pin will be sampled on the sample edge of the SPI clock. When $\overline{\text{SSx}}$ is sampled high, data will be transmitted on the next transmit edge of SCKx. A connection diagram indicating signal directions for this Operating mode is shown in Figure 20-12.

Figure 20-12: SPI Slave, Frame Slave Connection Diagram



Note 1: In Framed SPI modes, the $\overline{\text{SSx}}$ pin is used to transmit/receive the frame synchronization pulse.

2: Framed SPI modes require the use of all four pins (i.e., Using the $\overline{\text{SSx}}$ pin is not optional).

20.4 SPI Master Mode Clock Frequency

In the Master mode, the clock provided to the SPI module is the instruction cycle (T_{CY}). This clock will then be prescaled by the primary prescaler (specified by PPRE<1:0> (SPIxCON<1:0>)), and the secondary prescaler (specified by SPRE<2:0> (SPIxCON<4:2>)). The prescaled instruction clock becomes the serial clock and is provided to external devices via the SCKx pin.

> **Note:** Note that the SCKx signal clock is not free running for normal SPI modes. It will only run for 8 or 16 pulses when the SPIxBUF is loaded with data. It will however, be continuous for Framed modes.

Equation 20-1 can be used to calculate the SCKx clock frequency as a function of the primary and secondary prescaler settings.

Equation 20-1:

$$F_{SCK} = \frac{F_{CY}}{\text{Primary Prescaler * Secondary Prescaler}}$$

Some sample SPI clock frequencies (in kHz) are shown in the table below:

Table 20-1: Sample SCKx Frequencies

F_{CY} = 30 MHz		**Secondary Prescaler Settings**				
		1:1	2:1	4:1	6:1	8:1
Primary Prescaler Settings	1:1	30000	15000	7500	5000	3750
	4:1	7500	3750	1875	1250	938
	16:1	1875	938	469	313	234
	64:1	469	234	117	78	59
F_{CY} = 5 MHz						
Primary Prescaler Settings	1:1	5000	2500	1250	833	625
	4:1	1250	625	313	208	156
	16:1	313	156	78	52	39
	64:1	78	39	20	13	10

Note: SCKx frequencies shown in kHz.

> **Note:** Not all clock rates are supported. For further information, refer to the SPI timing specifications in the specific device data sheet.

20.5 Operation in Power Save Modes

The dsPIC30FXXXX family of devices has three Power modes:

- Operational mode: The core and peripherals are running.
- Power Save modes: These are invoked by the execution of the `PWRSAV` instruction. There are two Power Save modes supported in the dsPIC30F family of devices. These are specified in the `PWRSAV` instruction via a parameter. The two modes are:
  - Sleep mode: Device clock source and entire device is shut down. This is achieved by the following instruction.

    ```
    ;include device p30fxxxx.inc file
    PWRSAV #SLEEP_MODE
    ```

  - Idle mode: Device clock is operational, CPU and selected peripherals are shut down.

    ```
    ;include device p30fxxxx.inc file
    PWRSAV #IDLE_MODE
    ```

20.5.1 Sleep Mode

When the device enters Sleep mode, the system clock is disabled.

20.5.1.1 Master Mode Operation

The following are a consequence of entering Sleep mode when the SPIx module is configured for master operation:

- The baud rate generator in the SPIx module stops and is reset.
- If the SPIx module enters Sleep mode in the middle of a transmission/reception, then the transmission/reception is aborted. Since there is no automatic way to prevent an entry into Sleep mode if a transmission or reception is pending, the user software must synchronize entry into Sleep with SPI module operation to avoid aborted transmissions.
- The transmitter and receiver will stop in Sleep. The transmitter or receiver does not continue with a partially completed transmission at wake-up.

20.5.1.2 Slave Mode Operation

Since the clock pulses at SCKx are externally provided for Slave mode, the module will continue to function in Sleep mode. It will complete any transactions during the transition into Sleep. On completion of a transaction, the SPIRBF flag is set. Consequently, the SPIxIF bit will be set. If SPI interrupts are enabled (SPIxIE = `1`), the device will wake from Sleep. If the SPI interrupt priority level is greater than the present CPU priority level, code execution will resume at the SPIx interrupt vector location. Otherwise, code execution will continue with the instruction following the `PWRSAV` instruction that previously invoked Sleep mode. The module is not reset on entering Sleep mode if it is operating as a slave device.

Register contents are not affected when the SPIx module is going into or coming out of Sleep mode.

20.5.2 Idle Mode

When the device enters Idle mode, the system clock sources remain functional. The SPISIDL bit (SPIxSTAT<13>) selects whether the module will stop or continue functioning on Idle.

- If SPISIDL = `1`, the SPI module will stop communication on entering Idle mode. It will operate in the same manner as it does in Sleep mode.
- If SPISID = `0` (default selection), the module will continue operation in Idle mode.

Table 20-2: Pins Associated with the SPI Modules

Pin Name	Pin Type	Buffer Type	Description
SCK1	I/O	CMOS	SPI1 module Clock Input or Output
SCK2	I/O	CMOS	SPI2 module Clock Input or Output
SDI1	I	CMOS	SPI1 module Data Receive pin
SDI2	I	CMOS	SPI2 module Data Receive pin
SDO1	O	CMOS	SPI1 module Data Transmit pin
SDO2	O	CMOS	SPI2 module Data Transmit pin
$\overline{\text{SS1}}$	I/O	CMOS	SPI1 module Slave Select Control pin 1) Used to enable transmit/receive in Slave mode, if SSEN (SPI1CON<7>) has been set to ‘1’ 2) Used as Frame Sync I/O Pulse when FRMEN and SPIFSD (SPI1CON<14:13>) are set to ‘11’ or ‘10’.
$\overline{\text{SS2}}$	I/O	CMOS	SPI2 module Slave Select Control pin 1) Used to enable transmit/receive in Slave mode, if SSEN (SPI2CON<7>) has been set to ‘1’ 2) Used as Frame Sync I/O Pulse when FRMEN and SPIFSD (SPI2CON<14:13>) are set to ‘11’ or ‘10’.

Legend: CMOS = CMOS compatible input or output, ST = Schmitt Trigger input with CMOS levels, I = Input, O = Output

20.6 Special Function Registers Associated with SPI Modules

Table 20-3: SPI1 Register Map

SFR Name	Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset State
SPI1STAT	0220	SPIEN	—	SPISIDL	—	—	—	—	—	—	SPIROV	—	—	—	—	SPITBF	SPIRBF	0000 0000 0000 0000
SPI1CON	0222	—	FRMEN	SPIFSD	—	DISSDO	MODE16	SMP	CKE	SSEN	CKP	MSTEN	SPRE2	SPRE1	SPRE0	PPRE1	PPRE0	0000 0000 0000 0000
SPI1BUF	0224	Transmit and Receive Buffer Address shared by SPI1TXB and SPI1RXB registers																0000 0000 0000 0000

Table 20-4: SPI2 Register Map

SFR Name	Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset State
SPI2STAT	0226	SPIEN	—	SPISIDL	—	—	—	—	—	—	SPIROV	—	—	—	—	SPITBF	SPIRBF	0000 0000 0000 0000
SPI2CON	0228	—	FRMEN	SPIFSD	—	DISSDO	MODE16	SMP	CKE	SSEN	CKP	MSTEN	SPRE2	SPRE1	SPRE0	PPRE1	PPRE0	0000 0000 0000 0000
SPI2BUF	022A	Transmit and Receive Buffer Address shared by SPI2TXB and SPI2RXB registers																0000 0000 0000 0000

Table 20-5: SPI Module Related Interrupt Registers

SFR Name	Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset State
INTCON1	0080	NSTDIS	—	—	—	—	OVATE	OVBTE	COVTE	—	—	—	SWTRAP	OVRFLOW	ADDRERR	STKERR	—	0000 0000 0000 0000
INTCON2	0082	ALTIVT	DISI	—	—	—	—	LEV8F	—	—	—	—	INT4EP	INT3EP	INT2EP	INT1EP	INT0EP	0000 0000 0000 0000
IFS0	0084	CNIF	MI2CIF	SI2CIF	NVMIF	ADIF	U1TXIF	U1RXIF	SPI1IF	T3IF	T2IF	OC2IF	IC2IF	T1IF	OC1IF	IC1IF	INT0	0000 0000 0000 0000
IFS1	0086	IC6IF	IC5IF	IC4IF	IC3IF	C1IF	SPI2IF	U2TXIF	U2RXIF	INT2IF	T5IF	T4IF	OC4IF	OC3IF	IC8IF	IC7IF	INT1IF	0000 0000 0000 0000
IEC0	008C	CNIE	MI2CIE	SI2CIE	NVMIE	ADIE	U1TXIE	U1RXIE	SPI1IE	T3IE	T2IE	OC2IE	IC2IE	T1IE	OC1IE	IC1IE	INT0IE	0000 0000 0000 0000
IEC1	008E	IC6IE	IC5IE	IC4IE	IC3IE	C1IE	SPI2IE	U2TXIE	U2RXIE	INT2IE	T5IE	T4IE	OC4IE	OC3IE	IC8IE	IC7IE	INT1IE	0000 0000 0000 0000
IPC2	0098	—	ADIP<2:0>			—	U1TXIP<2:0>			—	U1RXIP<2:0>			—	SPI1IP<2:0>			0100 0100 0100 0100
IPC6	00A0	—	C1IP<2:0>			—	SPI2IP<2:0>			—	U2TXIP<2:0>			—	U2RXIP<2:0>			0100 0100 0100 0100

20.7 Related Application Notes

This section lists application notes that are related to this section of the manual. These application notes may not be written specifically for the dsPIC30F Product Family, but the concepts are pertinent and could be used with modification and possible limitations. The current application notes related to the Serial Peripheral Interface (SPI) module are:

Title	Application Note #
Interfacing Microchip's MCP41XXX/MCP42XXX Digital Potentiometers to a PICmicro® Microcontroller	AN746
Interfacing Microchip's MCP3201 Analog-to-Digital Converter to the PICmicro® Microcontroller	AN719

Note: Please visit the Microchip web site (www.microchip.com) for additional Application Notes and code examples for the dsPIC30F Family of devices.

20.8 Revision History

Revision A

This is the initial released revision of this document.

Revision B

This revision reflects editorial and technical content changes for the dsPIC30F Serial Peripheral Interface (SPI) module.

Revision C

There were no technical content revisions to this section of the manual, however, this section was updated to reflect Revision C throughout the manual.