
Section 35. Serial Peripheral Interface (SPI) (Part II)

HIGHLIGHTS

This section of the manual contains the following major topics:

35.1	Introduction	35-2
35.2	SPI Registers	35-3
35.3	Modes of Operation	35-8
35.4	Master Mode Clock Frequency	35-21
35.5	Operation in Power-Saving Modes	35-22
35.6	Special Function Registers Associated with the SPI1 Module	35-23
35.7	Related Application Notes	35-24
35.8	Revision History	35-25

Section 35. Serial Peripheral Interface (SPI) (Part II)

35.2 SPI Registers

The SPI module consists of a 16-bit shift register, SPI1SR, used for shifting data in and out, and a buffer register, SPI1BUF. The control registers, SPI1CON1 and SPI1CON2, configure the module. Additionally, a status register, SPI1STAT, indicates various status conditions.

35.2.1 Status and Control Registers

The SPI1STAT, SPI1CON1 and SPI1CON2 registers provide the interface to control the module's operation. They are shown in detail in Register 35-1, Register 35-2 and Register 35-3.

- SPI1STAT: SPI1 Status and Control Register

This status register indicates various status conditions such as Receive Overflow, Transmit Buffer Full and Receive Buffer Full. This register is also used to specify the operation of the module during Idle mode and contains a bit that enables and disables the module.

- SPI1CON1: SPI1 Control Register 1

This control register specifies the clock prescaler, Master/Slave mode, Word/Byte communication, clock polarity and clock/data pin operation.

- SPI1CON2: SPI1 Control Register 2

This control register enables or disables the framed SPI operation. This register also specifies the frame synchronization pulse direction, polarity and edge selection.

35.2.2 SPI1BUF Register

SPI1BUF is the SPI1 Data Receive/Transmit register. The SPI1BUF register is actually comprised of two separate registers: the Transmit Buffer, SPI1TXB, and the Receive Buffer, SPI1RXB. These two unidirectional, 16-bit registers share the SFR address of SPI1BUF. If a user application writes data to be transmitted to the SPI1BUF address, internally the data is written to the SPI1TXB register. Similarly, when the user application reads the received data from SPI1BUF, internally the data is read from the SPI1RXB register.

This technique double buffers transmit and receive operations and allows continuous data transfers in the background. Transmission and reception occur simultaneously.

In addition, there is a 16-bit shift register, SPI1SR, that is not memory mapped. It is used for shifting data in and out of the SPI port.

dsPIC30F Family Reference Manual

Register 35-1: SPI1STAT: SPI1 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/C-0	U-0	U-0	U-0	U-0	R-0	R-0
—	SPIROV	—	—	—	—	SPITBF	SPIRBF
bit 7				bit 0			

- bit 15 **SPIEN:** SPI1 Enable bit
1 = Enables the module and configures SCK1, SDO1, SDI1 and $\overline{SS1}$ as serial port pins
0 = Disables the 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 was completely received and discarded. The user application has not read the previous data in the SPI1BUF register.
0 = No overflow has occurred
- bit 5-2 **Unimplemented:** Read as '0'
- bit 1 **SPITBF:** SPI1 Transmit Buffer Full Status bit
1 = Transmit has not yet started; SPI1TXB is full
0 = Transmit has started; SPI1TXB is empty
Automatically set in hardware when the CPU writes SPI1BUF location, loading SPI1TXB.
Automatically cleared in hardware when the SPI1 module transfers data from SPI1TXB to SPI1SR.
- bit 0 **SPIRBF:** SPI1 Receive Buffer Full Status bit
1 = Receive is complete; SPI1RXB is full
0 = Receive is not complete; SPI1RXB is empty
Automatically set in hardware when the SPI1 module transfers data from SPI1SR to SPI1RXB.
Automatically cleared in hardware when the core reads SPI1BUF location, reading SPI1RXB.

Legend:	C = Clearable bit		
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

Section 35. Serial Peripheral Interface (SPI) (Part II)

Register 35-2: SPI1CON1: SPI1 Control Register 1

Upper Byte:							
U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	DISSCK	DISSDO	MODE16	SMP ⁽¹⁾	CKE ⁽²⁾
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 15-13 **Unimplemented:** Read as '0'
- bit 12 **DISSCK:** Disable SCK1 Pin bit (SPI Master modes only)
1 = Internal SPI clock is disabled; pin functions as I/O
0 = Internal SPI clock is enabled
- bit 11 **DISSDO:** Disable SDO1 Pin bit
1 = SDO1 pin is not used by the module; pin functions as I/O
0 = SDO1 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:** SPI1 Data Input Sample Phase bit⁽¹⁾
Master mode:
1 = Input data is sampled at the end of data output time
0 = Input data is sampled at the middle of data output time
Slave mode:
SMP must be cleared when SPI1 is used in Slave mode
- bit 8 **CKE:** SPI1 Clock Edge Select bit⁽²⁾
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)
- bit 7 **SSEN:** Slave Select Enable bit (Slave mode)
1 = SS1 pin is used for Slave mode
0 = SS1 pin is not used by module; pin is 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
- bit 4-2 **SPRE<2:0>:** Secondary Prescale bits (Master mode)
111 = Secondary prescale 1:1
110 = Secondary prescale 2:1
•
•
•
000 = Secondary prescale 8:1

Note 1: The SMP bit must be set only after setting the MSTEN bit. The SMP bit remains cleared if MSTEN = 0.

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

Register 35-2: SPI1CON1: SPI1 Control Register 1 (Continued)

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

- Note 1:** SMP bit must be set only after setting the MSTEN bit. The SMP bit remains cleared if MSTEN = 0.
- 2:** The CKE bit is not used in the Framed SPI modes. The user application should program this bit to '0' for the Framed SPI modes (FRMEN = 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

Section 35. Serial Peripheral Interface (SPI) (Part II)

Register 35-3: SPI1CON2: SPI1 Control Register 2

Upper Byte:							
R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0
FRMEN	SPIFSD	FRMPOL	—	—	—	—	—
bit 15			bit 8				

Lower Byte:							
U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	U-0
—	—	—	—	—	—	FRMDLY	—
bit 7						bit 0	

- bit 15 **FRMEN**: Framed SPI1 Support bit
1 = Framed SPI1 support is enabled ($\overline{\text{SS1}}$ pin is used as frame sync pulse input/output)
0 = Framed SPI1 support is disabled
- bit 14 **SPIFSD**: Frame Sync Pulse Direction Control bit
1 = Frame sync pulse input (slave)
0 = Frame sync pulse output (master)
- bit 13 **FRMPOL**: Frame Sync Pulse Polarity bit
1 = Frame sync pulse is active-high
0 = Frame sync pulse is active-low
- bit 12-2 **Unimplemented**: Read as '0'
- bit 1 **FRMDLY**: Frame Sync Pulse Edge Select bit
1 = Frame sync pulse coincides with first bit clock
0 = Frame sync pulse precedes first bit clock
- bit 0 **Unimplemented**: This bit must not be set to '1' by the user application

Legend:

R = Readable bit
-n = Value at POR

W = Writable bit
'1' = Bit is set

U = Unimplemented bit, read as '0'
'0' = Bit is cleared
x = Bit is unknown

35.3 Modes of Operation

The SPI module has flexible operating modes, which are discussed in the following sections:

- 8-bit and 16-bit Data Transmission/Reception
- Master and Slave Modes
- Framed SPI Modes
- SPI1 Receive-only Operation
- SPI1 Error Handling

35.3.1 8-bit and 16-bit Data Transmission/Reception

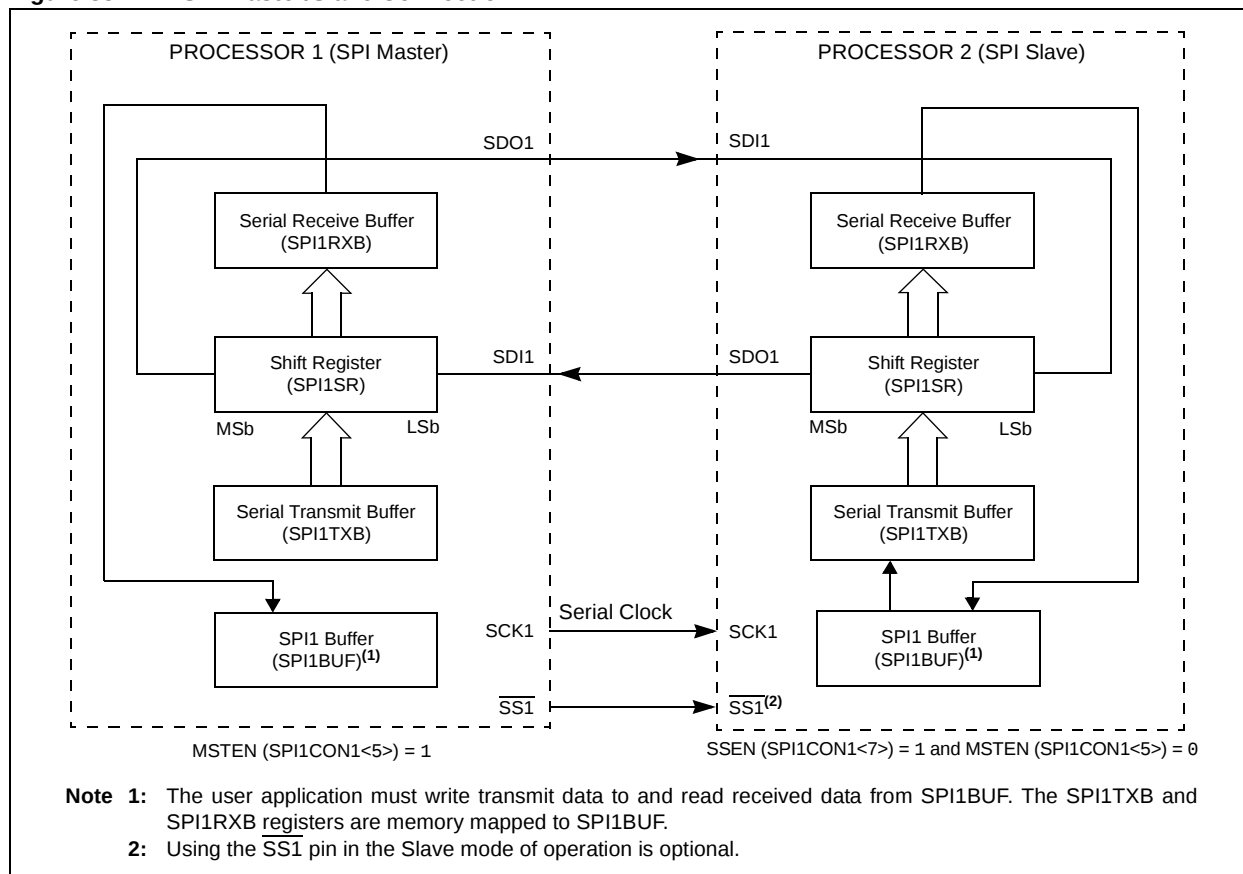
The MODE16 control bit (SPI1CON1<10>) allows the module to communicate in either 8-bit or 16-bit mode. The functionality will be the same for each mode, except for the number of bits that are received and transmitted. Additionally, the following items should be noted in this context:

- The module is reset when the value of the MODE16 bit is changed. Consequently, the bit should not be changed during normal operation.
- Data is transmitted out of bit 7 of the SPI1SR register for 8-bit operation, while it is transmitted out of bit 15 for 16-bit operation. In both modes, data is shifted into bit 0 of the SPI1SR register.
- When transmitting or receiving data, 8 clock pulses at the SCK1 pin are required to shift data in or out in 8-bit mode, while 16 clock pulses are required in 16-bit mode.

35.3.2 Master and Slave Modes

Data can be thought of as taking a direct path between the Most Significant bit (MSb) of one module's shift register and the Least Significant bit (LSb) of the other, and then into the appropriate Transmit or Receive Buffer. The module configured as the master module provides the serial clock and synchronization signals (as required) to the slave device. Figure 35-2 shows the connection of the Master and Slave modules.

Figure 35-2: SPI Master/Slave Connection



Section 35. Serial Peripheral Interface (SPI) (Part II)

35.3.2.1 Master Mode

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> (SPI1CON1<1:0>) and SPRE<2:0> (SPI1CON1<4:2>) bits. The serial clock is output via the SCK1 pin to the slave devices. The clock pulses are only generated when there is data to be transmitted. For further information, refer to **35.4 “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 received are, respectively, written into or read from the SPI1BUF register.

The following describes the SPI1 module operation in Master mode:

1. Once the module is set up for the Master mode of operation and enabled, data to be transmitted is written to the SPI1BUF register. The SPITBF (SPI1STAT<1>) bit is set.
2. The contents of SPI1TXB are moved to the shift register, SPI1SR, 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 SPI1SR to the SDO1 pin and simultaneously shifts in the data at the SDI1 pin into the SPI1SR.

When the transfer is complete, the following events occur:

- a) The interrupt flag bit, SPI1IF, is set. SPI1 interrupts can be enabled by setting the interrupt enable bit, SPI1IE. The SPI1IF flag is not cleared automatically by the hardware.
 - b) When the ongoing transmit and receive operation is completed, the contents of the SPI1SR register are moved to the SPI1RXB register.
 - c) The SPIRBF (SPI1STAT<0>) bit is set by the module, indicating that the receive buffer is full.
 - d) Once the SPI1BUF register is read by the user application, the hardware clears the SPIRBF bit.
4. If the SPIRBF bit is set (receive buffer is full) when the SPI1 module needs to transfer data from SPI1SR to SPI1RXB, the module will set the SPIROV (SPI1STAT<6>) bit, indicating an overflow condition.
 5. Data to be transmitted can be written to SPI1BUF by the user application at any time as long as the SPITBF (SPI1STAT<1>) bit is clear. The write can occur while SPI1SR is shifting out the previously written data, allowing continuous transmission.

Note: The SPI1SR register cannot be written into directly by the user application. All writes to the SPI1SR register are performed through the SPI1BUF register.

To set up the SPI1 module for the Master mode of operation, perform the following steps:

1. If using interrupts:
 - a) Clear the SPI1IF bit in the IFS0 register.
 - b) Set the SPI1IE bit in the IEC0 register.
 - c) Write the SPI1IP bits in the IPC2 register to set the interrupt priority.
2. Write the desired settings to the SPI1CON1 register with MSTEN (SPI1CON1<5>) = 1.
3. Clear the SPIROV bit (SPI1STAT<6>).
4. Enable SPI1 module operation by setting the SPIEN bit (SPI1STAT<15>).
5. Write the data to be transmitted to the SPI1BUF register. Transmission (and reception) will start as soon as data is written to the SPI1BUF register.

Example 35-1 shows a code snippet for configuring the SPI register for the Master mode.

Example 35-1: SPI Configuration – Master Mode

```
/* Following code snippet shows SPI register configuration for MASTER mode*/  
  
IFS0bits.SPI1IF = 0;           //Clear the Interrupt Flag  
IEC0bits.SPI1IE = 0;           //disable the Interrupt  
                                //SPI1CON1 Register Settings  
SPI1CON1bits.DISSCK = 0;       //Internal Serial Clock is Enabled.  
SPI1CON1bits.DISSDO = 0;       //SD01 pin is controlled by the module.  
SPI1CON1bits.MODE16 = 1;       //Communication is word-wide (16 bits).  
SPI1CON1bits.SMP = 0;          //Input Data is sampled at the middle of data  
                                //output time.  
SPI1CON1bits.CKE = 0;          //Serial output data changes on transition from  
                                //Idle clock state to active clock state  
SPI1CON1bits.CKP = 0;          //Idle state for clock is a low level; active  
                                //state is a high level  
SPI1CON1bits.MSTEN = 1;        //Master Mode Enabled  
SPI1STATbits.SPIEN = 1;        //Enable SPI Module  
SPI1BUF = 0x0000;              //Write data to be transmitted  
                                //Interrupt Controller Settings  
IFS0bits.SPI1IF = 0;           //Clear the Interrupt Flag  
IEC0bits.SPI1IE = 1;           //Enable the Interrupt
```

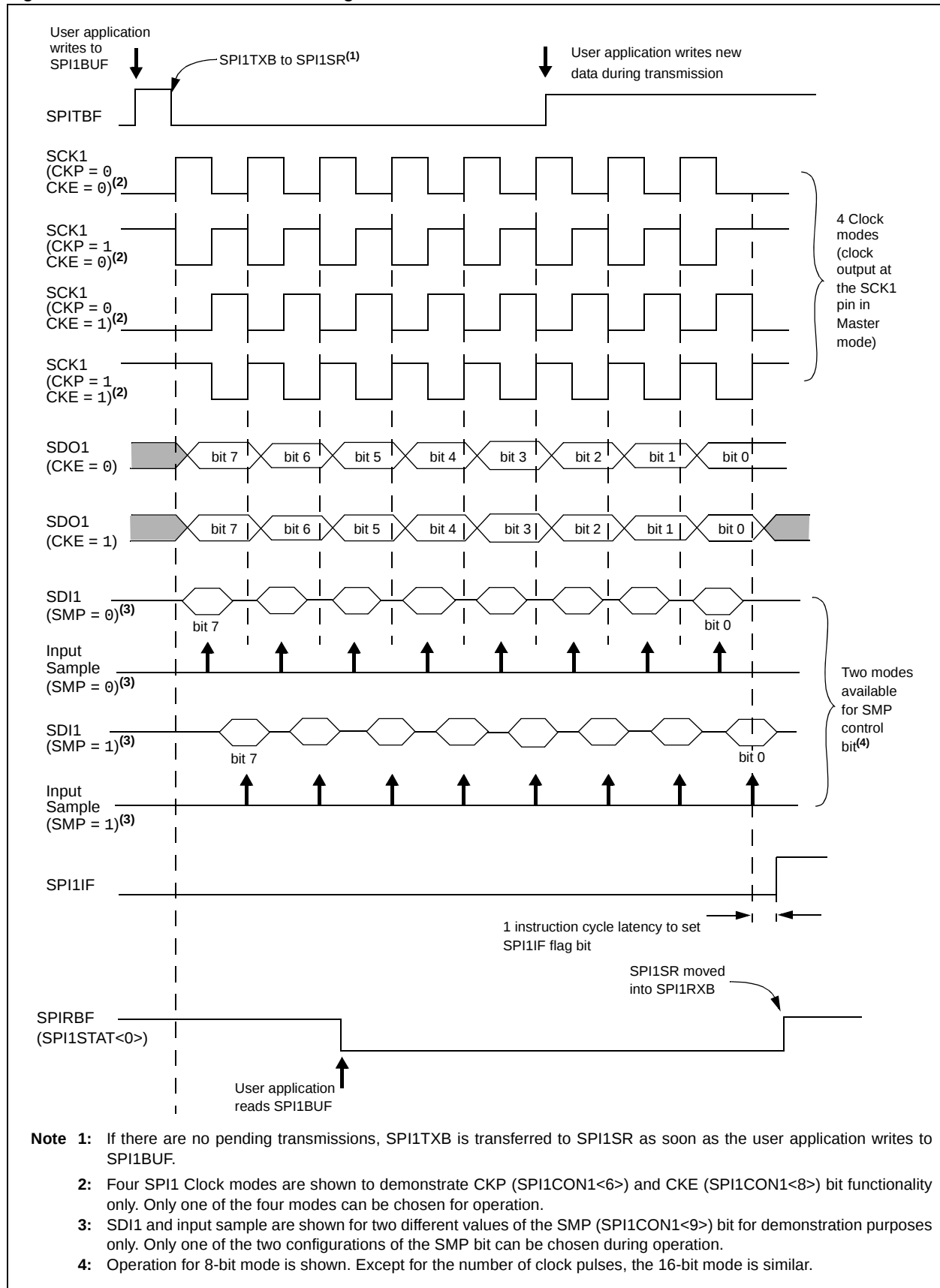
35.3.2.1.1 External Clocking in Master Mode

In Master mode, the module can also be configured to operate with an external data clock. SPI1 clock operation is controlled by the DISSCK bit (SPI1CON1<12>). When this bit is set, the internal data clock is disabled and data is transferred when external clock pulses are presented on the SCK1 pin. All other aspects of Master mode operation are the same as described above.

Note: The DISSCK bit is available only in SPI Master modes.

Section 35. Serial Peripheral Interface (SPI) (Part II)

Figure 35-3: SPI1 Master Mode Timing



35.3.2.2 Slave Mode

In Slave mode, data is transmitted and received as the external clock pulses appear on the SCK1 pin. The CKP (SPI1CON<6>) and CKE (SPI1CON<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 SPI1BUF register. The rest of the operation of the module is identical to that described above for the Master mode.

To set up the SPI1 module for the Slave mode of operation, perform the following steps:

1. Clear the SPI1BUF register.
2. If using interrupts:
 - a) Clear the SPI1IF bit in the IFS0 register.
 - b) Set the SPI1IE bit in the IEC0 register.
 - c) Write the SPI1IP bits in the IPC2 register to set the interrupt priority.
3. Write the desired settings to the SPI1CON1 and SPI1CON2 registers with MSTEN (SPI1CON1<5>) = 0.
4. Clear the SMP bit.
5. If the CKE bit is set, then the SSEN bit (SPI1CON1<7>) must be set to enable the $\overline{SS1}$ pin.
6. Clear the SPIROV bit (SPI1STAT<6>).
7. Enable SPI1 operation by setting the SPIEN bit (SPI1STAT<15>).

Example 35-2 shows a code snippet for configuring the SPI register for the Slave mode.

Example 35-2: SPI Configuration – Slave Mode

```
/* Following code snippet shows SPI register configuration for SLAVE mode*/

IFS0bits.SPI1IF = 0;           //Clear the Interrupt Flag
IEC0bits.SPI1IE = 0;           //Disable The Interrupt
                                // SPI1CON1 Register Settings

SPI1CON1bits.DISSCK = 0;       //Internal Serial Clock is Enabled.
SPI1CON1bits.DISSDO = 0;       //SD01 pin is controlled by the module.
SPI1CON1bits.MODE16 = 1;       //Communication is word-wide (16 bits).
SPI1CON1bits.SMP = 0;          //Input Data is sampled at the middle of data
                                //output time.
SPI1CON1bits.CKE = 0;          //Serial output data changes on transition
                                //from Idle clock state to active clock state
SPI1CON1bits.CKP = 0;          //Idle state for clock is a low level; active
                                //state is a high level
SPI1CON1bits.MSTEN = 0;        //Master Mode disabled
SPI1STATbits.SPIROV=0;         //No Receive Overflow Has Occurred
SPI1STATbits.SPIEN = 1;        //Enable SPI Module

                                //Interrupt Controller Settings
IFS0bits.SPI1IF = 0;           //Clear the Interrupt Flag
IEC0bits.SPI1IE = 1;           //Enable The Interrupt
```

35.3.2.2.1 Slave Select Synchronization

The $\overline{SS1}$ pin allows the Synchronous Slave mode. If the SSEN (SPI1CON1<7>) bit is set, transmission and reception are enabled in Slave mode only if the $\overline{SS1}$ pin is driven to a low state (see Figure 35-5). To ensure that the $\overline{SS1}$ pin can function correctly as an input, the port output or other peripheral outputs multiplexed with the $\overline{SS1}$ pin function must not be used. If the SSEN bit is set and the $\overline{SS1}$ pin is driven high, the SDO1 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, using the data held in the SPI1TXB register, the next time the $\overline{SS1}$ pin is driven low. If the SSEN bit is not set, the $\overline{SS1}$ pin does not affect the module operation in Slave mode.

Note: To meet module timing requirements, the $\overline{SS1}$ pin must be enabled in Slave mode when CKE = 1 (refer to Figure 35-6 for details).

Section 35. Serial Peripheral Interface (SPI) (Part II)

35.3.2.2.2 SPITBF Status Flag Operation

The SPITBF (SPI1STAT<1>) bit functions differently in the Slave mode of operation than in Master mode.

If SSEN (SPI1CON1<7>) is cleared, the SPITBF is set when the SPI1BUF is loaded by the user application. It is cleared when the module transfers SPI1TXB to SPI1SR. This is similar to the SPITBF bit function in Master mode.

If SSEN is set, the SPITBF is set when the SPI1BUF is loaded by the user application. However, it is cleared only when the SPI1 module completes data transmission. A transmission will be aborted when the SS1 pin goes high, but may be retried at a later time. Each data word is held in SPI1TXB until all bits are transmitted to the receiver.

Figure 35-4: SPI1 Slave Mode Timing (Slave Select Pin Disabled)⁽¹⁾

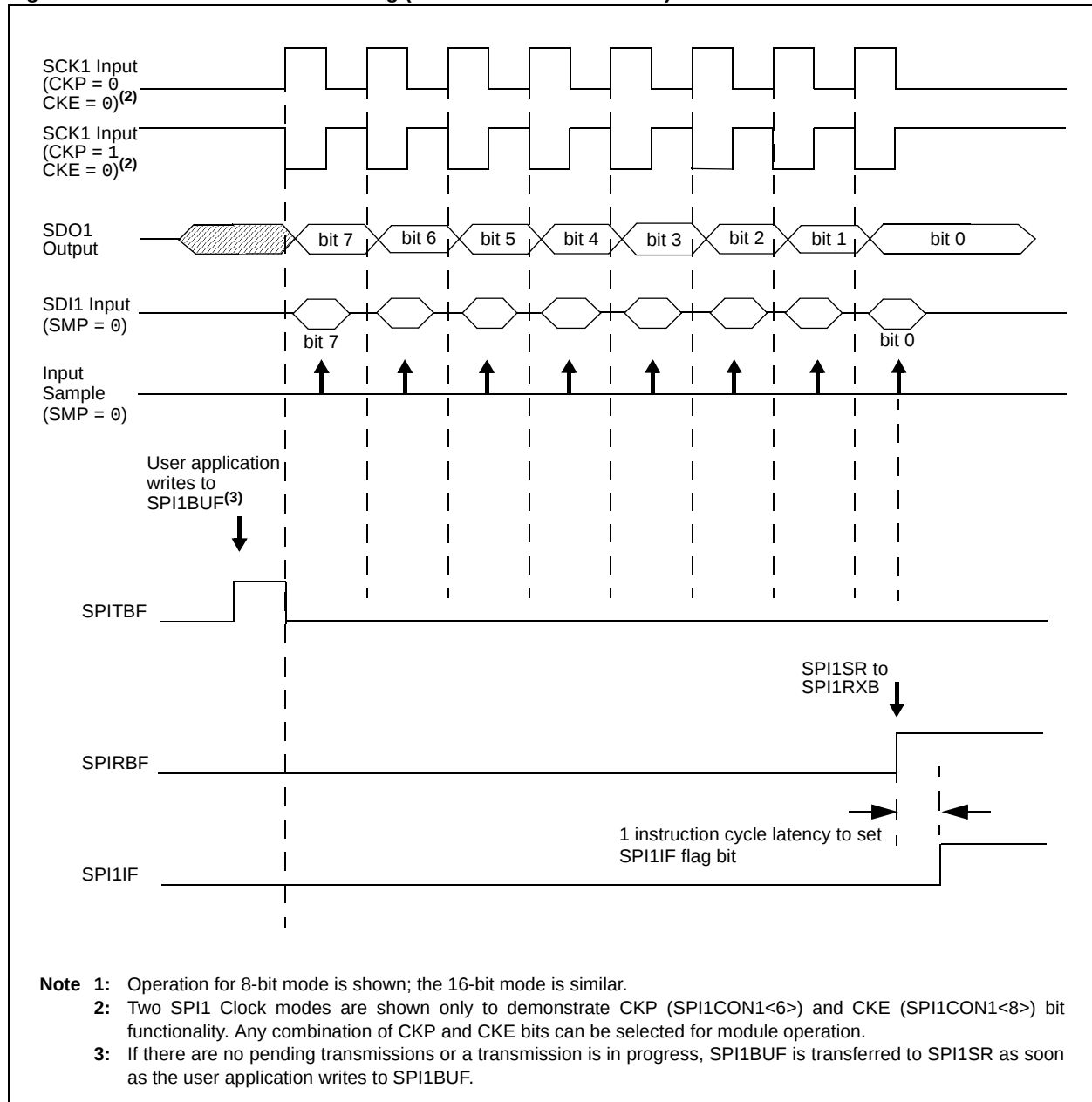
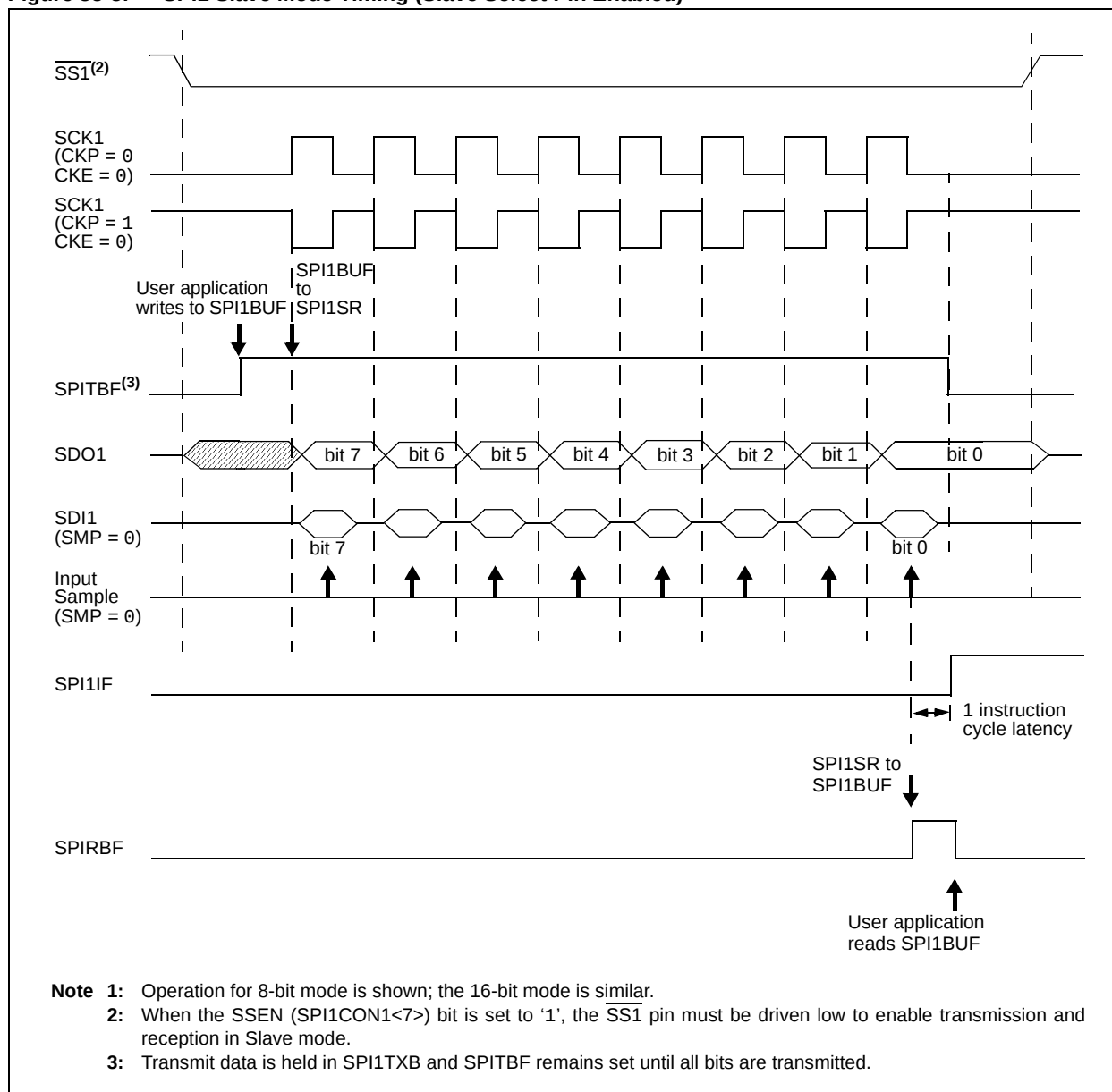
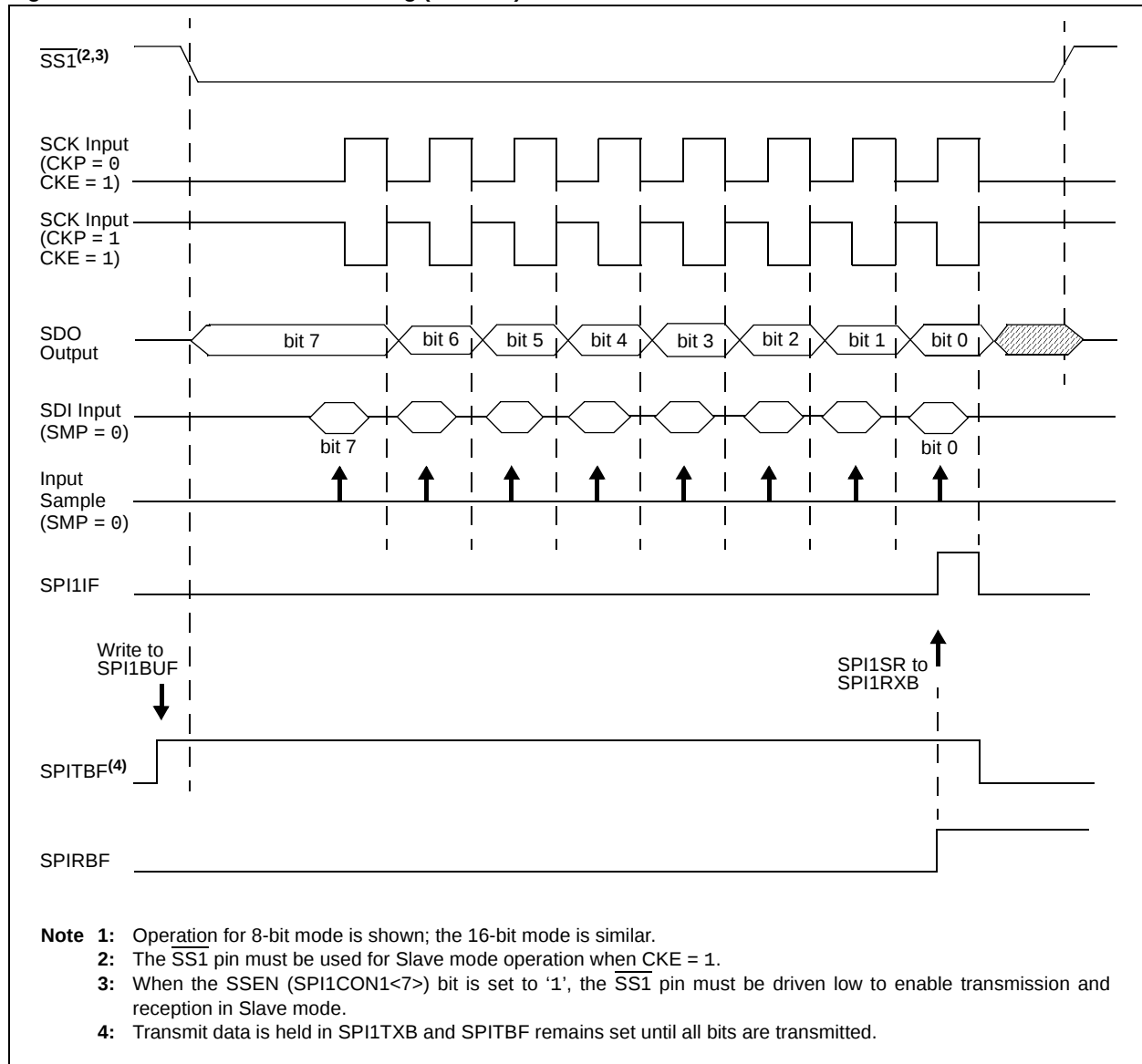


Figure 35-5: SPI1 Slave Mode Timing (Slave Select Pin Enabled)⁽¹⁾



Section 35. Serial Peripheral Interface (SPI) (Part II)

Figure 35-6: SPI1 Slave Mode Timing (CKE = 1)⁽¹⁾



35.3.3 Framed SPI Modes

The SPI1 module supports a basic framed SPI protocol while operating in either Master or Slave modes. The module uses four control bits to configure framed SPI operation:

- FRMEN (SPI1CON2<15>) enables the Framed SPI modes and causes the $\overline{SS1}$ pin to be used as a frame synchronization pulse input or output pin. The state of SSEN (SPI1CON1<7>) is ignored.
- SPIFSD (SPI1CON2<14>) determines whether the $\overline{SS1}$ pin is an input or an output (i.e., whether the module receives or generates the frame synchronization pulse).
- FRMPOL (SPI1CON2<13>) selects the polarity of the frame synchronization pulse (active-high or active-low) for a single SPI data frame.
- FRMDLY (SPI1CON2<1>) selects whether the synchronization pulse coincides with or precedes the first serial clock pulse.

The SPI1 module supports two framed modes of operation. In Framed Master mode, the SPI1 module generates the frame synchronization pulse and provides this pulse to other devices at the $\overline{SS1}$ pin. In Framed Slave mode, the SPI1 module uses a frame synchronization pulse received at the $\overline{SS1}$ pin.

Note: The use of the $\overline{SS1}$ and SCK1 pins is mandatory in all Framed SPI modes.

The following four Framed SPI modes are supported in conjunction with the unframed Master and Slave modes:

- SPI Master, Frame Master
- SPI Master, Frame Slave
- SPI Slave, Frame Master
- SPI Slave, Frame Slave

These modes determine whether or not the SPI1 module generates the serial clock and the frame synchronization pulse.

When $FRMEN$ ($SPI1CON2<15>$) = 1 and $MSTEN$ ($SPI1CON1<5>$) = 1, the SCK1 pin becomes an output and the SPI clock at SCK1 becomes a free running clock.

When $FRMEN$ = 1 and $MSTEN$ = 0, the SCK1 pin becomes an input. The source clock provided to the SCK1 pin is assumed to be a free running clock.

The polarity of the clock is selected by the CKP ($SPI1CON1<6>$) bit. The CKE ($SPI1CON1<8>$) bit is not used for the Framed SPI modes and should be programmed to '0' by the user application.

When CKP = 0, the frame synchronization pulse output and the SDO1 data output change on the rising edge of the clock pulses at the SCK1 pin. Input data is sampled at the SDI1 input pin on the falling edge of the serial clock.

When CKP = 1, the frame synchronization pulse output and the SDO1 data output change on the falling edge of the clock pulses at the SCK1 pin. Input data is sampled at the SDI1 input pin on the rising edge of the serial clock.

35.3.3.1 Frame Master and Frame Slave Modes

When $SPIFSD$ ($SPI1CON2<14>$) = 0, the SPI1 module is in the Frame Master mode of operation. In this mode, the frame sync pulse is initiated by the module when the user application writes the transmit data to the SPI1BUF location (thus writing the SPI1TXB register with transmit data). At the end of the frame synchronization pulse, the SPI1TXB is transferred to the SPI1SR and data transmission or reception begins.

When $SPIFSD$ = 1, the module is in Framed Slave mode. In this mode, the frame synchronization pulse is generated by an external source. When the module samples the frame synchronization pulse, it will transfer the contents of the SPI1TXB register to the SPI1SR and data transmission or reception begins. The user application must make sure that the correct transmission data is loaded into the SPI1BUF before the frame synchronization pulse is received.

Note: Receiving a frame synchronization pulse will start a transmission regardless of whether data was written to SPI1BUF. If no write was performed, the old contents of the SPI1TXB will be transmitted.

Section 35. Serial Peripheral Interface (SPI) (Part II)

35.3.3.2 SPI Master, Frame Master Mode

In the SPI Master/Frame Master mode, the SPI1 module generates both the clock and frame synchronization signals, as shown in Figure 35-7. This configuration is enabled by setting the MSTEN and FRMEN bits to '1' and the SPIFSD bit to '0'.

In this mode, the serial clock is output continuously at the SCK1 pin regardless of whether the module is transmitting. When SPI1BUF is written, the SS1 pin will be driven to its active state (as determined by the FRMPOL bit) on the appropriate transmit edge of the SCK1 clock and remain active for one data frame. If the FRMDLY control bit (SPI1CON2<1>) is cleared, the frame synchronization pulse precedes the data transmission, as shown in Figure 35-8. If FRMDLY is set, the frame synchronization pulse coincides with the beginning of the data transmission, as shown in Figure 35-9. The module starts transmitting data on the next transmit edge of the SCK1.

Figure 35-7: SPI Master, Frame Master Connection Diagram

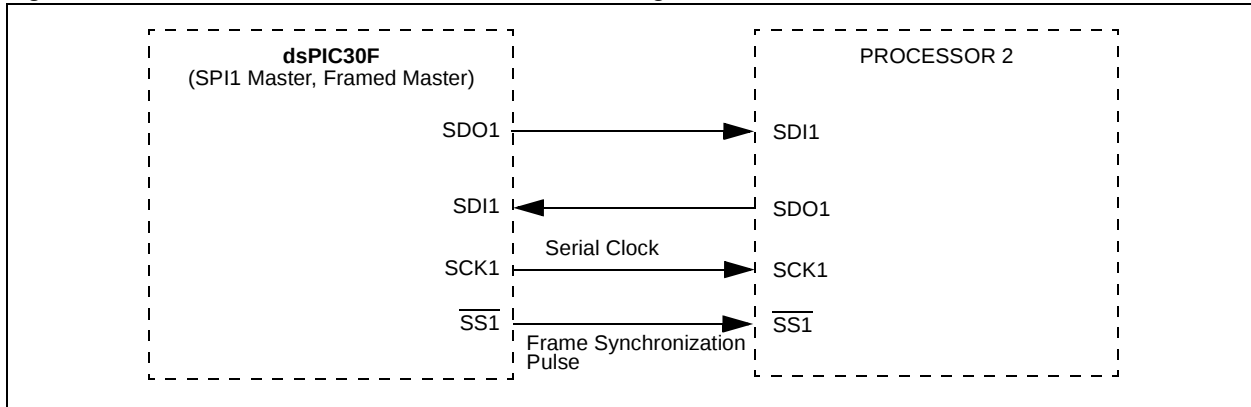


Figure 35-8: SPI Master, Frame Master Timing (FRMDLY = 0)

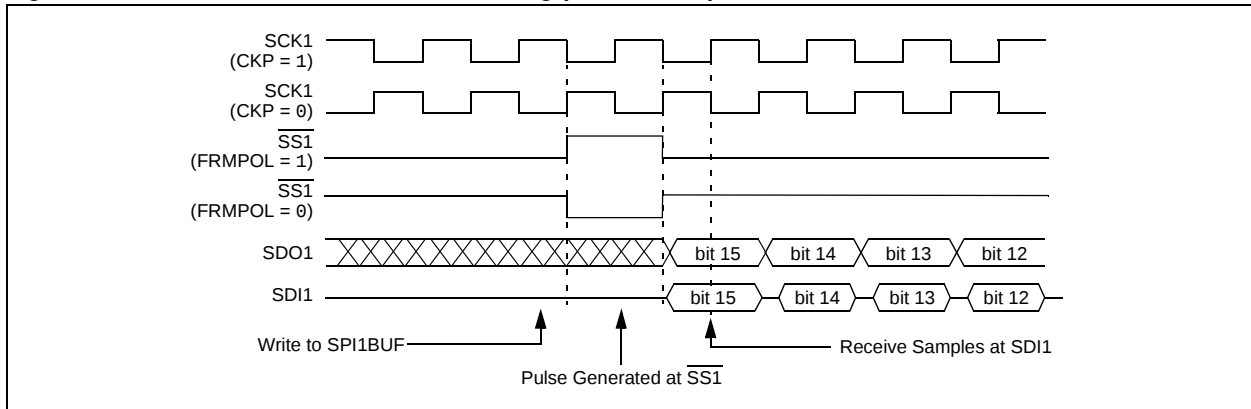
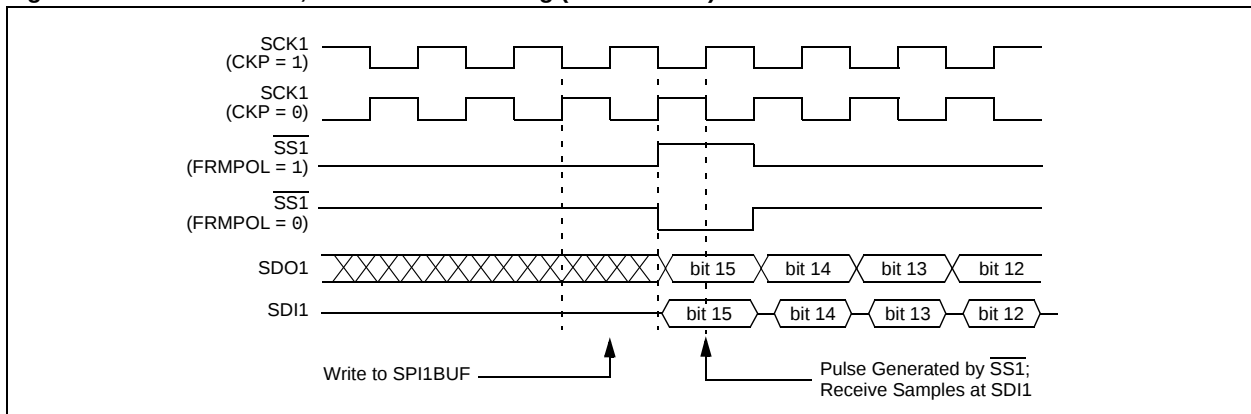


Figure 35-9: SPI Master, Frame Master Timing (FRMDLY = 1)



35.3.3.3 SPI Master, Frame Slave Mode

In the SPI Master/Frame Slave mode, the module generates the clock signal but uses the Slave module frame synchronization signal for data transmission (Figure 35-10). This mode is enabled by setting the MSTEN, FRMEN and SPIFSD bits to '1'.

In this mode, the $\overline{SS1}$ pin is an input and is sampled on the sample edge of the SPI1 clock. When it is sampled in its active state, data will be transmitted on the subsequent transmit edge of the SPI1 clock. The interrupt flag, SPI1IF, is set when the transmission is complete. The user application must make sure that the correct transmission data is loaded into the SPI1BUF before the signal is received at the $\overline{SS1}$ pin.

Figure 35-10: SPI Master, Frame Slave Connection Diagram

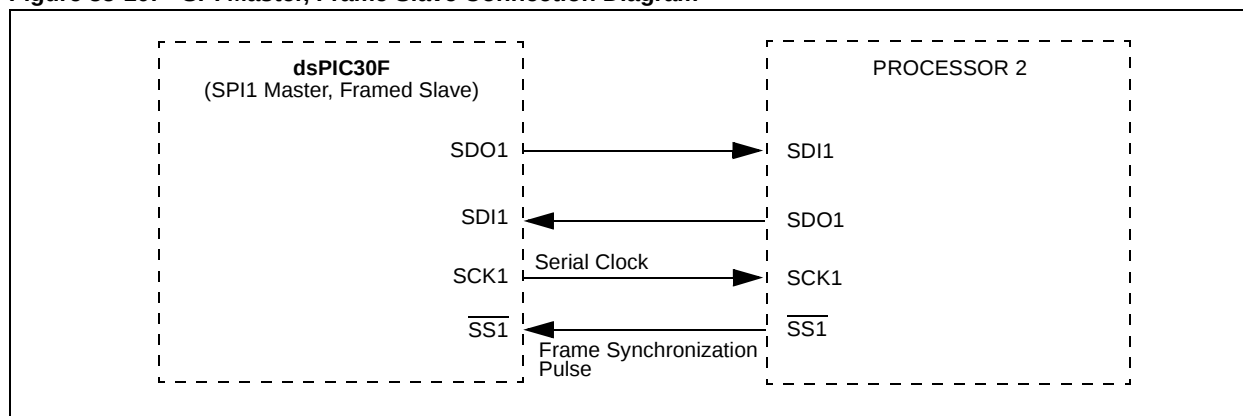


Figure 35-11: SPI Master, Frame Slave Timing (FRMDLY = 0)

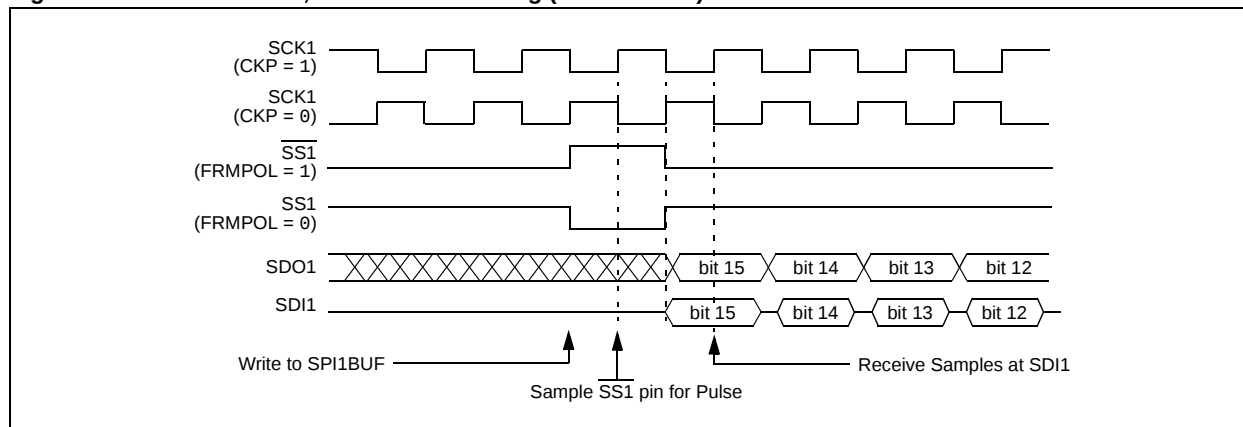
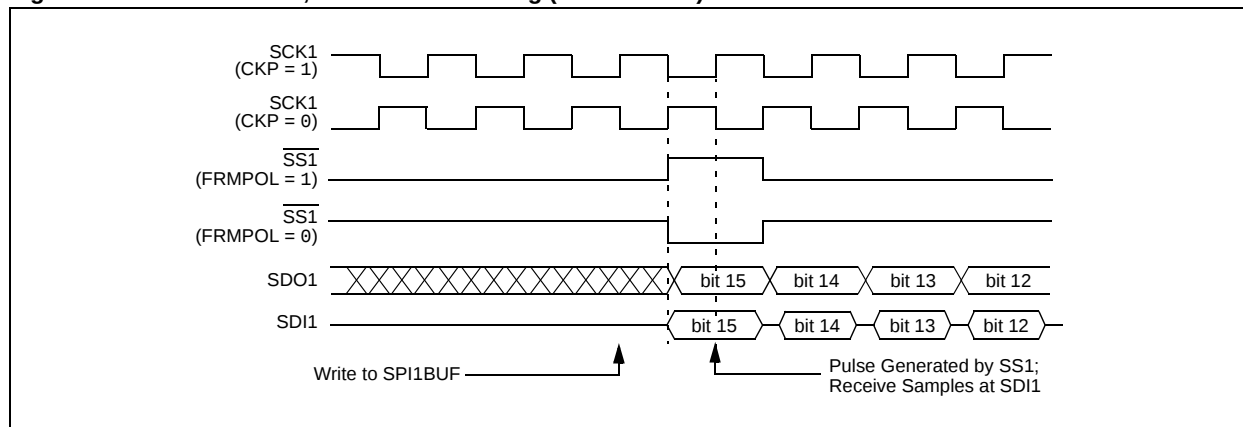


Figure 35-12: SPI Master, Frame Slave Timing (FRMDLY = 1)



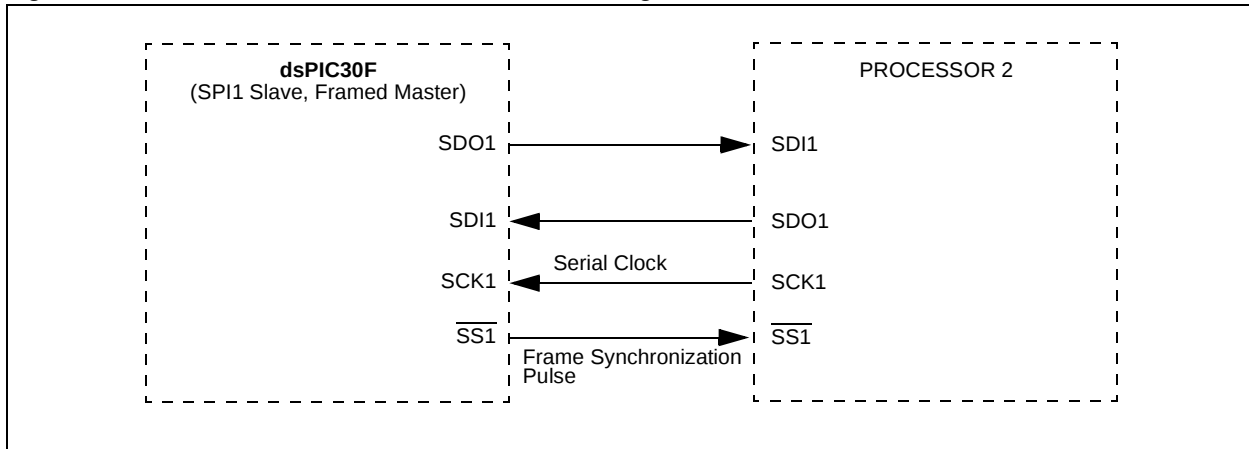
Section 35. Serial Peripheral Interface (SPI) (Part II)

35.3.3.4 SPI Slave, Frame Master Mode

In the SPI Slave/Frame Master mode, the module acts as the SPI slave and takes its clock from the other SPI module; however, it produces frame synchronization signals to control data transmission (Figure 35-13). This mode is enabled by setting the MSTEN bit to '0', the FRMEN bit to '1' and the SPIFSD bit to '0'.

The input SPI clock will be continuous in Slave mode. The $\overline{SS1}$ pin will be an output when the SPIFSD bit is low. Therefore, when the SPI1BUF is written, the module drives the $\overline{SS1}$ pin to the active state on the appropriate transmit edge of the SPI clock for one SPI clock cycle. Data will start transmitting on the appropriate SPI clock transmit edge.

Figure 35-13: SPI Slave, Frame Master Connection Diagram

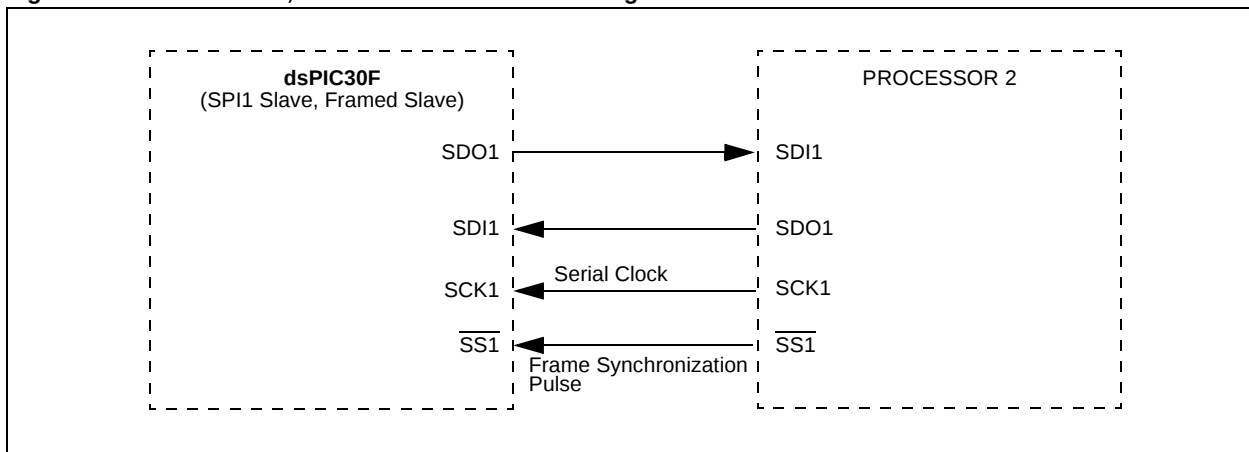


35.3.3.5 SPI Slave, Frame Slave Mode

In the SPI Slave/Frame Slave mode, the module obtains both its clock and frame synchronization signal from the Master module (Figure 35-14). This mode is enabled by setting MSTEN to '0', FRMEN to '1' and SPIFSD to '1'.

In this mode, both the SCK1 and $\overline{SS1}$ pins will be inputs. The $\overline{SS1}$ pin is sampled on the sample edge of the SPI clock. When $\overline{SS1}$ is sampled at its active state, data will be transmitted on the appropriate transmit edge of SCK1.

Figure 35-14: SPI Slave, Frame Slave Connection Diagram



35.3.4 SPI1 Receive-only Operation

Setting the DISSDO control bit (SPI1CON1<11>) disables transmission at the SDO1 pin. This allows the SPI1 module to be configured for a Receive-only mode of operation. The SDO1 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.

35.3.5 SPI1 Error Handling

If a new data word has been shifted into SPI1SR but the previous SPI1BUF contents have not been read, the SPIROV bit (SPI1STAT<6>) will be set. Any received data in SPI1SR will not be transferred, and further data reception is disabled until the SPIROV bit is cleared. The SPIROV bit is not cleared automatically by the module; it must be cleared by the user application.

The SPI1 Interrupt Flag, SPI1IF, is set whenever the SPIROV, SPIRBF (SPI1STAT<0>) or SPITBF (SPI1STAT<1>) bits are set. The interrupt flag cannot be cleared by hardware and must be reset in software. The actual SPI1 interrupt is generated only when the corresponding SPI1IE bit is set in the IEC0 Control register.

Section 35. Serial Peripheral Interface (SPI) (Part II)

35.4 Master Mode Clock Frequency

In the Master mode, the clock provided to the SPI1 module is the instruction cycle (Tcy). This clock will then be prescaled by the primary prescaler, specified by PPRE<1:0> (SPI1CON1<1:0>), and the secondary prescaler, specified by SPRE<2:0> (SPI1CON1<4:2>). The prescaled instruction clock becomes the serial clock and is provided to external devices through the SCK1 pin.

Note: Note that the SCK1 signal clock is not free running for normal SPI modes. It will only run for 8 or 16 pulses when the SPI1BUF is loaded with data. However, it is continuous for Framed modes.

Equation 35-1 can be used to calculate the SCK1 clock frequency as a function of the primary and secondary prescaler settings.

Equation 35-1: SPI Clock Frequency

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

Some sample SPI clock frequencies (in kHz) are shown in Table 35-1.

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

Table 35-1: Sample SCK1 Frequencies⁽¹⁾

Fcy = 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
Fcy = 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 1: SCK1 frequencies are shown in kHz.

35.5 Operation in Power-Saving Modes

The dsPIC30F SMPS and Digital Power Conversion device family has three power modes: the normal operational (Full-Power) mode and the two power-saving modes invoked by the PWRSAV instruction. Depending on the SPI mode selected, entering a power-saving mode may also affect the operation of the SPI1 module.

35.5.1 Sleep Mode

When the device enters Sleep mode, the system clock is disabled. The consequences of entering Sleep mode depend on which mode (Master or Slave) the SPI1 module is configured for at the time that Sleep mode is invoked.

35.5.1.1 Master Mode Operation

The consequences of entering Sleep mode when the SPI1 module is configured for Master operation are as follows:

- The Baud Rate Generator in the SPI1 module stops and is reset.
- The transmitter and receiver will stop in Sleep mode. The transmitter or receiver does not continue with a partially completed transmission at wake-up.
- If the SPI1 module enters Sleep mode in the middle of a transmission or reception, the transmission or 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 application must synchronize entry into Sleep mode with SPI1 module operation to avoid aborted transmissions.

35.5.1.2 Slave Mode Operation

Since the clock pulses at SCK1 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 mode. On completion of a transaction, the SPIRBF flag is set. Consequently, the SPI1IF bit will be set. If SPI1 interrupts are enabled (SPI1IE = 1), the device will wake from Sleep mode. If the SPI1 interrupt priority level is greater than the present CPU priority level, code execution will resume at the SPI1 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.

The register contents are not affected when the SPI1 module is going into or coming out of Sleep mode.

35.5.2 Idle Mode

When the device enters Idle mode, the system clock sources remain functional. The SPISIDL bit (SPI1STAT<13>) selects whether the module will stop or continue functioning on Idle mode.

If SPISIDL = 1, the SPI1 module will stop communication on entering Idle mode. It will operate in the same manner as it does in Sleep mode. If SPISIDL = 0 (default selection), the module will continue operation in Idle mode.

35.6 Special Function Registers Associated with the SPI1 Module

Table 35-2: 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	All Resets
SPI1STAT	0240	SPIEN	—	SPISIDL	—	—	—	—	—	—	SPIROV	—	—	—	—	SPITBF	SPIRBF	0000
SPI1CON1	0242	—	—	—	DISSCK	DISSDO	MODE16	SMP	CKE	SSEN	CKP	MSTEN	SPRE<2:0>			PPRE<1:0>		0000
SPI1CON2	0244	FRMEN	SPIFSD	FRMPOL	—	—	—	—	—	—	—	—	—	—	—	FRMDLY	—	0000
SPI1BUF	0246	SPI1 Transmit and Receive Buffer Register																0000

35.7 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 SMPS and Digital Power Conversion device 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 include the following:

Title	Application Note #
Interfacing SPI™ Serial EEPROMs to PIC18 Devices	AN1006
Interfacing Microchip's MCP41XXX and 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 SMPS and Digital Power Conversion device family.
--

35.8 Revision History

Revision A (February 2007)

This is the initial released revision of this document.

Revision B (September 2008)

This revision incorporates the following content updates:

- Registers:
 - SPI1CON1: SPI1 Control Register 1 (see Register 35-2) – Note 1 has been added in the register.
- Additional minor corrections such as language and formatting updates are incorporated in the entire document.

NOTES: