

PIC16F87X

9.0 MASTER SYNCHRONOUS SERIAL PORT (MSSP) MODULE

The Master Synchronous Serial Port (MSSP) module is a 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 MSSP module can operate in one of two modes:

- Serial Peripheral Interface (SPI)
- Inter-Integrated Circuit (I²C)

Figure 9-1 shows a block diagram for the SPI mode, while Figure 9-5 and Figure 9-9 show the block diagrams for the two different I²C modes of operation.

The Application Note AN734, "Using the PICmicro[®] SSP for Slave I²C™ Communication" describes the slave operation of the MSSP module on the PIC16F87X devices. AN735, "Using the PICmicro[®] MSSP Module for I²C™ Communications" describes the master operation of the MSSP module on the PIC16F87X devices.

PIC16F87X

REGISTER 9-1: SSPSTAT: SYNC SERIAL PORT STATUS REGISTER (ADDRESS: 94h)

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
SMP	CKE	D/A	P	S	R/W	UA	BF
bit 7							bit 0

bit 7	<p>SMP: Sample bit</p> <p><u>SPI Master mode:</u> 1 = Input data sampled at end of data output time 0 = Input data sampled at middle of data output time</p> <p><u>SPI Slave mode:</u> SMP must be cleared when SPI is used in slave mode</p> <p><u>In I²C Master or Slave mode:</u> 1 = Slew rate control disabled for standard speed mode (100 kHz and 1 MHz) 0 = Slew rate control enabled for high speed mode (400 kHz)</p>
bit 6	<p>CKE: SPI Clock Edge Select (Figure 9-2, Figure 9-3 and Figure 9-4)</p> <p><u>SPI mode:</u> For CKP = 0 1 = Data transmitted on rising edge of SCK 0 = Data transmitted on falling edge of SCK For CKP = 1 1 = Data transmitted on falling edge of SCK 0 = Data transmitted on rising edge of SCK</p> <p><u>In I²C Master or Slave mode:</u> 1 = Input levels conform to SMBus spec 0 = Input levels conform to I²C specs</p>
bit 5	<p>D/A: Data/Address bit (I²C mode only) 1 = Indicates that the last byte received or transmitted was data 0 = Indicates that the last byte received or transmitted was address</p>
bit 4	<p>P: STOP bit (I²C mode only. This bit is cleared when the MSSP module is disabled, SSPEN is cleared.) 1 = Indicates that a STOP bit has been detected last (this bit is '0' on RESET) 0 = STOP bit was not detected last</p>
bit 3	<p>S: START bit (I²C mode only. This bit is cleared when the MSSP module is disabled, SSPEN is cleared.) 1 = Indicates that a START bit has been detected last (this bit is '0' on RESET) 0 = START bit was not detected last</p>
bit 2	<p>R/W: Read/Write bit Information (I²C mode only) This bit holds the R/W bit information following the last address match. This bit is only valid from the address match to the next START bit, STOP bit or not ACK bit.</p> <p><u>In I²C Slave mode:</u> 1 = Read 0 = Write</p> <p><u>In I²C Master mode:</u> 1 = Transmit is in progress 0 = Transmit is not in progress Logical OR of this bit with SEN, RSEN, PEN, RCEN, or ACKEN will indicate if the MSSP is in IDLE mode.</p>
bit 1	<p>UA: Update Address (10-bit I²C mode only) 1 = Indicates that the user needs to update the address in the SSPADD register 0 = Address does not need to be updated</p>
bit	<p>BF: Buffer Full Status bit</p> <p><u>Receive (SPI and I²C modes):</u> 1 = Receive complete, SSPBUF is full 0 = Receive not complete, SSPBUF is empty</p> <p><u>Transmit (I²C mode only):</u> 1 = Data transmit in progress (does not include the ACK and STOP bits), SSPBUF is full 0 = Data transmit complete (does not include the ACK and STOP bits), SSPBUF is empty</p>

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

PIC16F87X

REGISTER 9-2: SSPCON: SYNC SERIAL PORT CONTROL REGISTER (ADDRESS 14h)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0
bit 7							bit 0

- bit 7 **WCOL:** Write Collision Detect bit
Master mode:
 1 = A write to SSPBUF was attempted while the I2C conditions were not valid
 0 = No collision
Slave mode:
 1 = SSPBUF register is written while still transmitting the previous word (must be cleared in software)
 0 = No collision
- bit 6 **SSPOV:** Receive Overflow Indicator bit
In SPI mode:
 1 = A new byte is received while SSPBUF holds previous data. Data in SSPSR is lost on overflow. In Slave mode, the user must read the SSPBUF, even if only transmitting data, to avoid overflows. In Master mode, the overflow bit is not set, since each operation is initiated by writing to the SSPBUF register. (Must be cleared in software.)
 0 = No overflow
In I²C mode:
 1 = A byte is received while the SSPBUF is holding the previous byte. SSPOV is a "don't care" in Transmit mode. (Must be cleared in software.)
 0 = No overflow
- bit 5 **SSPEN:** Synchronous Serial Port Enable bit
In SPI mode:
 When enabled, these pins must be properly configured as input or output
 1 = Enables serial port and configures SCK, SDO, SDI, and SS as the source of the serial port pins
 0 = Disables serial port and configures these pins as I/O port pins
In I²C mode:
 When enabled, these pins must be properly configured as input or output
 1 = Enables the serial port and configures the SDA and SCL pins as the source of the serial port pins
 0 = Disables serial port and configures these pins as I/O port pins
- bit 4 **CKP:** Clock Polarity Select bit
In SPI mode:
 1 = Idle state for clock is a high level
 0 = Idle state for clock is a low level
In I²C Slave mode:
 SCK release control
 1 = Enable clock
 0 = Holds clock low (clock stretch). (Used to ensure data setup time.)
In I²C Master mode:
 Unused in this mode
- bit 3-0 **SSPM3:SSPM0:** Synchronous Serial Port Mode Select bits
 0000 = SPI Master mode, clock = Fosc/4
 0001 = SPI Master mode, clock = Fosc/16
 0010 = SPI Master mode, clock = Fosc/64
 0011 = SPI Master mode, clock = TMR2 output/2
 0100 = SPI Slave mode, clock = SCK pin. \overline{SS} pin control enabled.
 0101 = SPI Slave mode, clock = SCK pin. \overline{SS} pin control disabled. \overline{SS} can be used as I/O pin.
 0110 = I²C Slave mode, 7-bit address
 0111 = I²C Slave mode, 10-bit address
 1000 = I²C Master mode, clock = Fosc / (4 * (SSPADD+1))
 1011 = I²C Firmware Controlled Master mode (slave idle)
 1110 = I²C Firmware Controlled Master mode, 7-bit address with START and STOP bit interrupts enabled
 1111 = I²C Firmware Controlled Master mode, 10-bit address with START and STOP bit interrupts enabled
 1001, 1010, 1100, 1101 = Reserved

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

PIC16F87X

REGISTER 9-3: SSPCON2: SYNC SERIAL PORT CONTROL REGISTER2 (ADDRESS 91h)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN
bit 7							bit 0

- bit 7 **GCEN:** General Call Enable bit (In I²C Slave mode only)
 1 = Enable interrupt when a general call address (0000h) is received in the SSPSR
 0 = General call address disabled
- bit 6 **ACKSTAT:** Acknowledge Status bit (In I²C Master mode only)
In Master Transmit mode:
 1 = Acknowledge was not received from slave
 0 = Acknowledge was received from slave
- bit 5 **ACKDT:** Acknowledge Data bit (In I²C Master mode only)
In Master Receive mode:
 Value that will be transmitted when the user initiates an Acknowledge sequence at the end of a receive.
 1 = Not Acknowledge
 0 = Acknowledge
- bit 4 **ACKEN:** Acknowledge Sequence Enable bit (In I²C Master mode only)
In Master Receive mode:
 1 = Initiate Acknowledge sequence on SDA and SCL pins and transmit ACKDT data bit. Automatically cleared by hardware.
 0 = Acknowledge sequence idle
- bit 3 **RCEN:** Receive Enable bit (In I²C Master mode only)
 1 = Enables Receive mode for I²C
 0 = Receive idle
- bit 2 **PEN:** STOP Condition Enable bit (In I²C Master mode only)
SCK Release Control:
 1 = Initiate STOP condition on SDA and SCL pins. Automatically cleared by hardware.
 0 = STOP condition idle
- bit 1 **RSEN:** Repeated START Condition Enable bit (In I²C Master mode only)
 1 = Initiate Repeated START condition on SDA and SCL pins. Automatically cleared by hardware.
 0 = Repeated START condition idle
- bit 0 **SEN:** START Condition Enable bit (In I²C Master mode only)
 1 = Initiate START condition on SDA and SCL pins. Automatically cleared by hardware.
 0 = START condition idle

Note: For bits ACKEN, RCEN, PEN, RSEN, SEN: If the I²C module is not in the IDLE mode, this bit may not be set (no spooling), and the SSPBUF may not be written (or writes to the SSPBUF are disabled).

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

PIC16F87X

9.1 SPI Mode

The SPI mode allows 8 bits of data to be synchronously transmitted and received simultaneously. All four modes of SPI are supported. To accomplish communication, typically three pins are used:

- Serial Data Out (SDO)
- Serial Data In (SDI)
- Serial Clock (SCK)

Additionally, a fourth pin may be used when in a Slave mode of operation:

- Slave Select (\overline{SS})

When initializing the SPI, several options need to be specified. This is done by programming the appropriate control bits (SSPCON<5:0> and SSPSTAT<7:6>). These control bits allow the following to be specified:

- Master mode (SCK is the clock output)
- Slave mode (SCK is the clock input)
- Clock Polarity (Idle state of SCK)
- Data input sample phase (middle or end of data output time)
- Clock edge (output data on rising/falling edge of SCK)
- Clock Rate (Master mode only)
- Slave Select mode (Slave mode only)

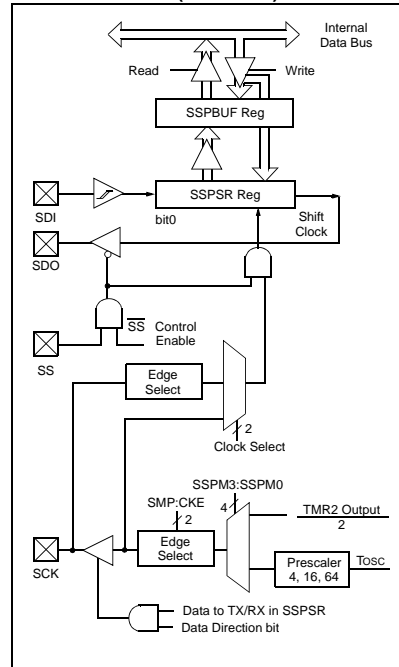
Figure 9-4 shows the block diagram of the MSSP module when in SPI mode.

To enable the serial port, MSSP Enable bit, SSPEN (SSPCON<5>) must be set. To reset or reconfigure SPI mode, clear bit SSPEN, re-initialize the SSPCON registers, and then set bit SSPEN. This configures the SDI, SDO, SCK and \overline{SS} pins as serial port pins. For the pins to behave as the serial port function, some must have their data direction bits (in the TRIS register) appropriately programmed. That is:

- SDI is automatically controlled by the SPI module
- SDO must have TRISC<5> cleared
- SCK (Master mode) must have TRISC<3> cleared
- SCK (Slave mode) must have TRISC<3> set
- \overline{SS} must have TRISA<5> set and register ADCON1 (see Section 11.0: A/D Module) must be set in a way that pin RA5 is configured as a digital I/O

Any serial port function that is not desired may be overridden by programming the corresponding data direction (TRIS) register to the opposite value.

FIGURE 9-1: MSSP BLOCK DIAGRAM (SPI MODE)



PIC16F87X

9.1.1 MASTER MODE

The master can initiate the data transfer at any time because it controls the SCK. The master determines when the slave (Processor 2, Figure 9-5) is to broadcast data by the software protocol.

In Master mode, the data is transmitted/received as soon as the SSPBUF register is written to. If the SPI module is only going to receive, the SDO output could be disabled (programmed as an input). The SSPSR register will continue to shift in the signal present on the SDI pin at the programmed clock rate. As each byte is received, it will be loaded into the SSPBUF register as if a normal received byte (interrupts and status bits appropriately set). This could be useful in receiver applications as a "line activity monitor".

The clock polarity is selected by appropriately programming bit CKP (SSPCON<4>). This then, would give waveforms for SPI communication as shown in

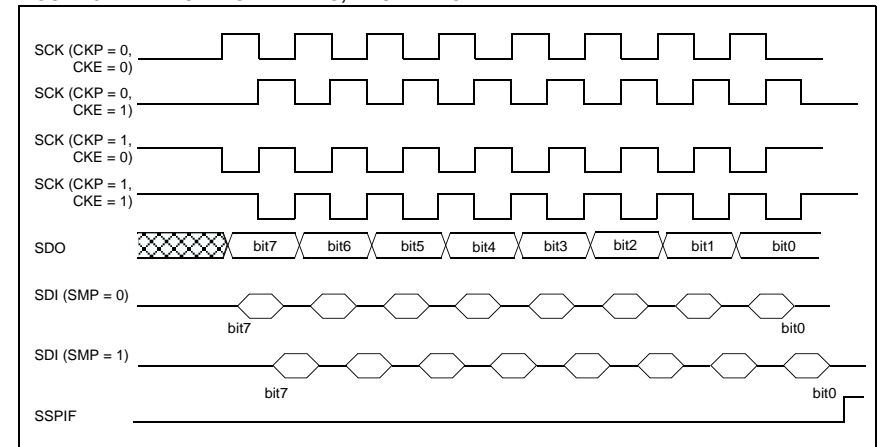
Figure 9-6, Figure 9-8 and Figure 9-9, where the MSb is transmitted first. In Master mode, the SPI clock rate (bit rate) is user programmable to be one of the following:

- $F_{osc}/4$ (or T_{cy})
- $F_{osc}/16$ (or $4 \cdot T_{cy}$)
- $F_{osc}/64$ (or $16 \cdot T_{cy}$)
- Timer2 output/2

This allows a maximum bit clock frequency (at 20 MHz) of 5.0 MHz.

Figure 9-6 shows the waveforms for Master mode. When $CKE = 1$, the SDO data is valid before there is a clock edge on SCK. The change of the input sample is shown based on the state of the SMP bit. The time when the SSPBUF is loaded with the received data is shown.

FIGURE 9-2: SPI MODE TIMING, MASTER MODE



PIC16F87X

9.1.2 SLAVE MODE

In Slave mode, the data is transmitted and received as the external clock pulses appear on SCK. When the last bit is latched, the interrupt flag bit SSPIF (PIR1<3>) is set.

While in Slave mode, the external clock is supplied by the external clock source on the SCK pin. This external clock must meet the minimum high and low times as specified in the electrical specifications.

While in SLEEP mode, the slave can transmit/receive data. When a byte is received, the device will wake-up from SLEEP.

Note 1: When the SPI module is in Slave mode with \overline{SS} pin control enabled (SSPCON<3:0> = 0100), the SPI module will reset if the \overline{SS} pin is set to VDD.

2: If the SPI is used in Slave mode with CKE = '1', then \overline{SS} pin control must be enabled.

FIGURE 9-3: SPI MODE TIMING (SLAVE MODE WITH CKE = 0)

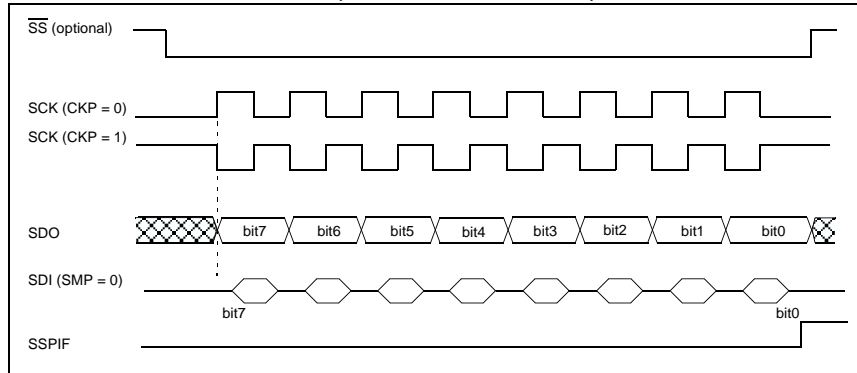
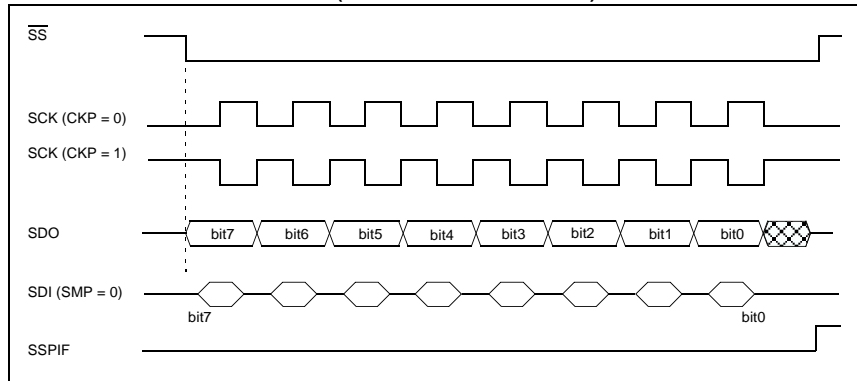


FIGURE 9-4: SPI MODE TIMING (SLAVE MODE WITH CKE = 1)



PIC16F87X

TABLE 9-1: REGISTERS ASSOCIATED WITH SPI OPERATION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on: MCLR, WDT
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIE ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
13h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxx xxxx	uuuu uuuu
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
94h	SSPSTAT	SMP	CKE	D/A	P	S	R/W	UA	BF	0000 0000	0000 0000

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used by the SSP in SPI mode.

Note 1: These bits are reserved on PIC16F873/876 devices; always maintain these bits clear.

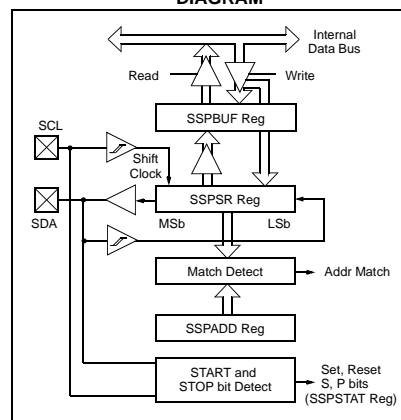
9.2 MSSP I²C Operation

The MSSP module in I²C mode, fully implements all master and slave functions (including general call support) and provides interrupts on START and STOP bits in hardware, to determine a free bus (multi-master function). The MSSP module implements the standard mode specifications, as well as 7-bit and 10-bit addressing.

Refer to Application Note AN578, "Use of the SSP Module in the I²C Multi-Master Environment."

A "glitch" filter is on the SCL and SDA pins when the pin is an input. This filter operates in both the 100 kHz and 400 kHz modes. In the 100 kHz mode, when these pins are an output, there is a slow rate control of the pin that is independent of device frequency.

FIGURE 9-5: I²C SLAVE MODE BLOCK DIAGRAM



Two pins are used for data transfer. These are the SCL pin, which is the clock, and the SDA pin, which is the data. The SDA and SCL pins are automatically configured when the I²C mode is enabled. The SSP module functions are enabled by setting SSP Enable bit SSPEN (SSPCON<5>).

The MSSP module has six registers for I²C operation. They are the:

- SSP Control Register (SSPCON)
- SSP Control Register2 (SSPCON2)
- SSP Status Register (SSPSTAT)
- Serial Receive/Transmit Buffer (SSPBUF)
- SSP Shift Register (SSPSR) - Not directly accessible
- SSP Address Register (SSPADD)

The SSPCON register allows control of the I²C operation. Four mode selection bits (SSPCON<3:0>) allow one of the following I²C modes to be selected:

- I²C Slave mode (7-bit address)
- I²C Slave mode (10-bit address)
- I²C Master mode, clock = OSC/4 (SSPADD +1)
- I²C firmware modes (provided for compatibility to other mid-range products)

Before selecting any I²C mode, the SCL and SDA pins must be programmed to inputs by setting the appropriate TRIS bits. Selecting an I²C mode by setting the SSPEN bit, enables the SCL and SDA pins to be used as the clock and data lines in I²C mode. Pull-up resistors must be provided externally to the SCL and SDA pins for the proper operation of the I²C module.

The CKE bit (SSPSTAT<6:7>) sets the levels of the SDA and SCL pins in either Master or Slave mode. When CKE = 1, the levels will conform to the SMBus specification. When CKE = 0, the levels will conform to the I²C specification.

The SSPSTAT register gives the status of the data transfer. This information includes detection of a START (S) or STOP (P) bit, specifies if the received byte was data or address, if the next byte is the completion of 10-bit address, and if this will be a read or write data transfer.

SSPBUF is the register to which the transfer data is written to, or read from. The SSPSR register shifts the data in or out of the device. In receive operations, the SSPBUF and SSPSR create a doubled buffered receiver. This allows reception of the next byte to begin before reading the last byte of received data. When the complete byte is received, it is transferred to the SSPBUF register and flag bit SSPIF is set. If another complete byte is received before the SSPBUF register is read, a receiver overflow has occurred and bit SSPOV (SSPCON<6>) is set and the byte in the SSPSR is lost.

The SSPADD register holds the slave address. In 10-bit mode, the user needs to write the high byte of the address (1111 0 A9 A8 0). Following the high byte address match, the low byte of the address needs to be loaded (A7:A0).

9.2.1 SLAVE MODE

In Slave mode, the SCL and SDA pins must be configured as inputs. The MSSP module will override the input state with the output data, when required (slave-transmitter).

When an address is matched, or the data transfer after an address match is received, the hardware automatically will generate the Acknowledge (ACK) pulse, and then load the SSPBUF register with the received value currently in the SSPSR register.

There are certain conditions that will cause the MSSP module not to give this ACK pulse. These are if either (or both):

- The buffer full bit BF (SSPSTAT<0>) was set before the transfer was received.
- The overflow bit SSPOV (SSPCON<6>) was set before the transfer was received.

If the BF bit is set, the SSPSR register value is not loaded into the SSPBUF, but bit SSPIF and SSPOV are set. Table 9-2 shows what happens when a data transfer byte is received, given the status of bits BF and SSPOV. The shaded cells show the condition where user software did not properly clear the overflow condition. Flag bit BF is cleared by reading the SSPBUF register, while bit SSPOV is cleared through software.

The SCL clock input must have a minimum high and low time for proper operation. The high and low times of the I²C specification, as well as the requirement of the MSSP module, is shown in timing parameter #100 and parameter #101 of the electrical specifications.

9.2.1.1 Addressing

Once the MSSP module has been enabled, it waits for a START condition to occur. Following the START condition, the 8-bits are shifted into the SSPSR register. All incoming bits are sampled with the rising edge of the clock (SCL) line. The value of register SSPSR<7:1> is compared to the value of the SSPADD register. The address is compared on the falling edge of the eighth clock (SCL) pulse. If the addresses match, and the BF and SSPOV bits are clear, the following events occur:

- The SSPSR register value is loaded into the SSPBUF register on the falling edge of the 8th SCL pulse.
- The buffer full bit, BF, is set on the falling edge of the 8th SCL pulse.
- An ACK pulse is generated.
- SSP interrupt flag bit, SSPIF (PIR1<3>), is set (interrupt is generated if enabled) on the falling edge of the 9th SCL pulse.

In 10-bit address mode, two address bytes need to be received by the slave. The five Most Significant bits (MSBs) of the first address byte specify if this is a 10-bit address. Bit R/W (SSPSTAT<2>) must specify a write so the slave device will receive the second address byte.

For a 10-bit address, the first byte would equal '1111 0 A9 A8 0', where A9 and A8 are the two MSBs of the address. The sequence of events for a 10-bit address is as follows, with steps 7-9 for slave-transmitter:

1. Receive first (high) byte of Address (bits SSPIF, BF and UA (SSPSTAT<1>) are set).
2. Update the SSPADD register with the second (low) byte of Address (clears bit UA and releases the SCL line).
3. Read the SSPBUF register (clears bit BF) and clear flag bit SSPIF.
4. Receive second (low) byte of Address (bits SSPIF, BF and UA are set).
5. Update the SSPADD register with the first (high) byte of Address. This will clear bit UA and release the SCL line.
6. Read the SSPBUF register (clears bit BF) and clear flag bit SSPIF.
7. Receive Repeated Start condition.
8. Receive first (high) byte of Address (bits SSPIF and BF are set).
9. Read the SSPBUF register (clears bit BF) and clear flag bit SSPIF.

Note: Following the Repeated START condition (step 7) in 10-bit mode, the user only needs to match the first 7-bit address. The user does not update the SSPADD for the second half of the address.

9.2.1.2 Slave Reception

When the R/W bit of the address byte is clear and an address match occurs, the R/W bit of the SSPSTAT register is cleared. The received address is loaded into the SSPBUF register.

When the address byte overflow condition exists, then no Acknowledge (ACK) pulse is given. An overflow condition is defined as either bit BF (SSPSTAT<0>) is set, or bit SSPOV (SSPCON<6>) is set. This is an error condition due to user firmware.

An SSP interrupt is generated for each data transfer byte. Flag bit SSPIF (PIR1<3>) must be cleared in software. The SSPSTAT register is used to determine the status of the received byte.

Note: The SSPBUF will be loaded if the SSPOV bit is set and the BF flag is cleared. If a read of the SSPBUF was performed, but the user did not clear the state of the SSPOV bit before the next receive occurred, the ACK is not sent and the SSPBUF is updated.

TABLE 9-2: DATA TRANSFER RECEIVED BYTE ACTIONS

Status Bits as Data Transfer is Received		SSPSR → SSPBUF	Generate ACK Pulse	Set bit SSPIF (SSP Interrupt occurs if enabled)
BF	SSPOV			
0	0	Yes	Yes	Yes
1	0	No	No	Yes
1	1	No	No	Yes
0	1	Yes	No	Yes

Note: Shaded cells show the conditions where the user software did not properly clear the overflow condition.

9.2.1.3 Slave Transmission

When the R/W bit of the incoming address byte is set and an address match occurs, the R/W bit of the SSPSTAT register is set. The received address is loaded into the SSPBUF register. The ACK pulse will be sent on the ninth bit, and the SCL pin is held low. The transmit data must be loaded into the SSPBUF register, which also loads the SSPSR register. Then, the SCL pin should be enabled by setting bit CKP (SSPCON<4>). The master must monitor the SCL pin prior to asserting another clock pulse. The slave devices may be holding off the master by stretching the clock. The eight data bits are shifted out on the falling edge of the SCL input. This ensures that the SDA signal is valid during the SCL high time (Figure 9-7).

An SSP interrupt is generated for each data transfer byte. The SSPIF flag bit must be cleared in software and the SSPSTAT register is used to determine the status of the byte transfer. The SSPIF flag bit is set on the falling edge of the ninth clock pulse.

As a slave-transmitter, the ACK pulse from the master receiver is latched on the rising edge of the ninth SCL input pulse. If the SDA line is high (not ACK), then the data transfer is complete. When the not ACK is latched by the slave, the slave logic is reset and the slave then monitors for another occurrence of the START bit. If the SDA line was low (ACK), the transmit data must be loaded into the SSPBUF register, which also loads the SSPSR register. Then the SCL pin should be enabled by setting the CKP bit.

FIGURE 9-6: I²C WAVEFORMS FOR RECEPTION (7-BIT ADDRESS)

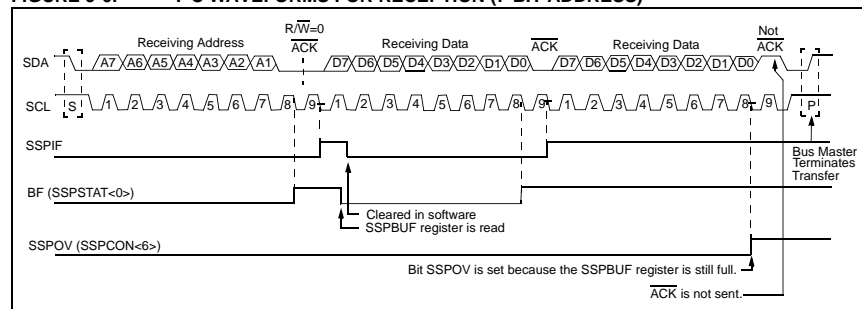
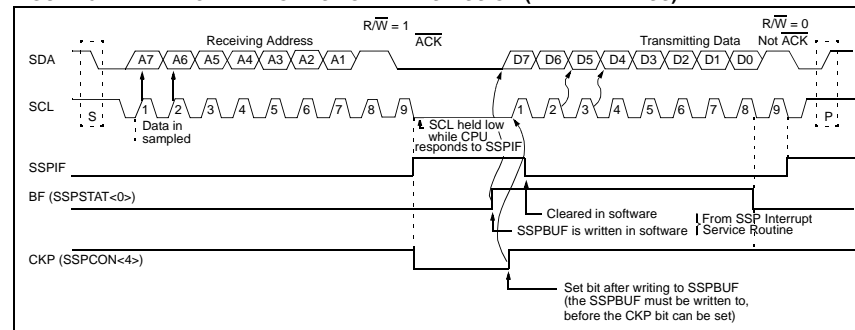


FIGURE 9-7: I²C WAVEFORMS FOR TRANSMISSION (7-BIT ADDRESS)



9.2.2 GENERAL CALL ADDRESS SUPPORT

The addressing procedure for the I²C bus is such that the first byte after the START condition usually determines which device will be the slave addressed by the master. The exception is the general call address, which can address all devices. When this address is used, all devices should, in theory, respond with an acknowledge.

The general call address is one of eight addresses reserved for specific purposes by the I²C protocol. It consists of all 0's with R/W = 0.

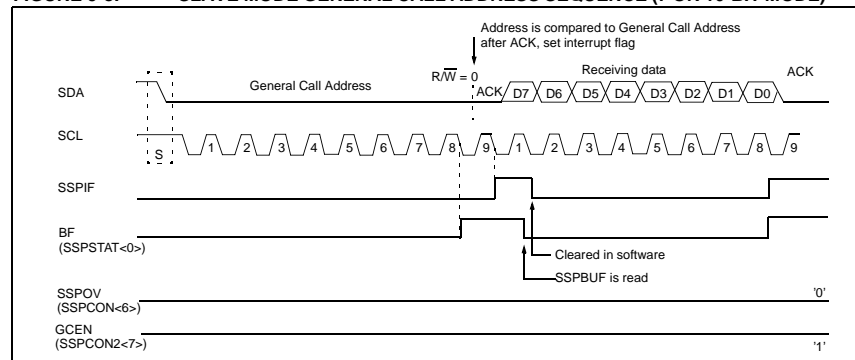
The general call address is recognized when the General Call Enable bit (GCEN) is enabled (SSPCON2<7> is set). Following a START bit detect, 8 bits are shifted into SSPSR and the address is compared against SSPADD. It is also compared to the general call address and fixed in hardware.

If the general call address matches, the SSPSR is transferred to the SSPBUF, the BF flag is set (eighth bit), and on the falling edge of the ninth bit (ACK bit), the SSPIF flag is set.

When the interrupt is serviced, the source for the interrupt can be checked by reading the contents of the SSPBUF to determine if the address was device specific, or a general call address.

In 10-bit mode, the SSPADD is required to be updated for the second half of the address to match, and the UA bit is set (SSPSTAT<1>). If the general call address is sampled when GCEN is set, while the slave is configured in 10-bit address mode, then the second half of the address is not necessary, the UA bit will not be set, and the slave will begin receiving data after the Acknowledge (Figure 9-8).

FIGURE 9-8: SLAVE MODE GENERAL CALL ADDRESS SEQUENCE (7 OR 10-BIT MODE)



PIC16F87X

9.2.3 SLEEP OPERATION

While in SLEEP mode, the I²C module can receive addresses or data. When an address match or complete byte transfer occurs, wake the processor from SLEEP (if the SSP interrupt is enabled).

9.2.4 EFFECTS OF A RESET

A RESET disables the SSP module and terminates the current transfer.

TABLE 9-3: REGISTERS ASSOCIATED WITH I²C OPERATION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on: MCLR, WDT
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
0Dh	PIR2	—	(2)	—	EEIF	BCLIF	—	—	CCP2IF	-x-0 0--0	-x-0 0--0
8Dh	PIE2	—	(2)	—	EEIE	BCLIE	—	—	CCP2IE	-x-0 0--0	-x-0 0--0
13h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxx xxxx	uuuu uuuu
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
91h	SSPCON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	0000 0000	0000 0000
93h	SSPAD	I ² C Slave Address/Master Baud Rate Register								0000 0000	0000 0000
94h	SSPSTAT	SMP	CKE	D/A	P	S	R/W	UA	BF	0000 0000	0000 0000

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used by the SSP I²C mode.

Note 1: These bits are reserved on PIC16F873/876 devices; always maintain these bits clear.

Note 2: These bits are reserved on these devices; always maintain these bits clear.

PIC16F87X

9.2.5 MASTER MODE

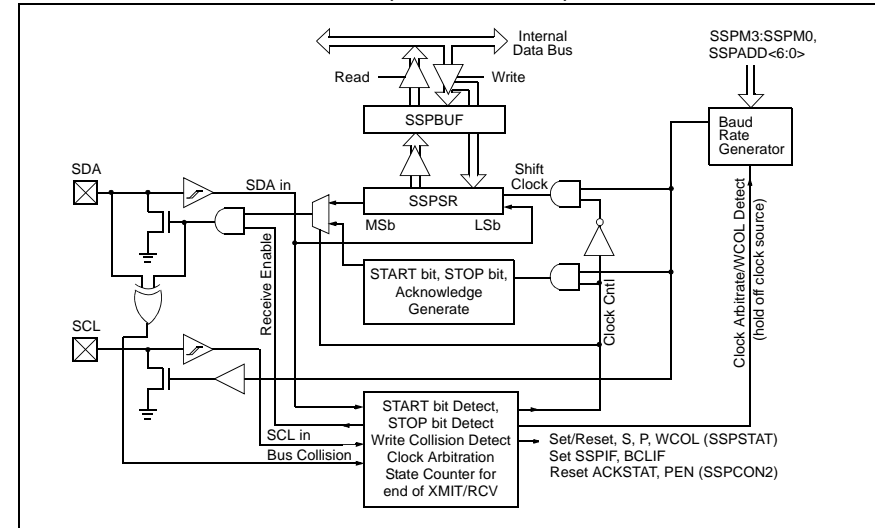
Master mode of operation is supported by interrupt generation on the detection of the START and STOP conditions. The STOP (P) and START (S) bits are cleared from a RESET, or when the MSSP module is disabled. Control of the I²C bus may be taken when the P bit is set, or the bus is idle, with both the S and P bits clear.

In Master mode, the SCL and SDA lines are manipulated by the MSSP hardware.

The following events will cause the SSP Interrupt Flag bit, SSPIF, to be set (an SSP interrupt will occur if enabled):

- START condition
- STOP condition
- Data transfer byte transmitted/received
- Acknowledge transmit
- Repeated START

FIGURE 9-9: SSP BLOCK DIAGRAM (I²C MASTER MODE)



9.2.6 MULTI-MASTER MODE

In Multi-Master mode, the interrupt generation on the detection of the START and STOP conditions allows the determination of when the bus is free. The STOP (P) and START (S) bits are cleared from a RESET or when the MSSP module is disabled. Control of the I²C bus may be taken when bit P (SSPSTAT<4>) is set, or the bus is idle with both the S and P bits clear. When the bus is busy, enabling the SSP Interrupt will generate the interrupt when the STOP condition occurs.

In Multi-Master operation, the SDA line must be monitored for arbitration to see if the signal level is the expected output level. This check is performed in hardware, with the result placed in the BCLIF bit.

The states where arbitration can be lost are:

- Address Transfer
- Data Transfer
- A START Condition
- A Repeated START Condition
- An Acknowledge Condition

9.2.7 I²C MASTER MODE SUPPORT

Master mode is enabled by setting and clearing the appropriate SSPM bits in SSPCON and by setting the SSPEN bit. Once Master mode is enabled, the user has six options:

- Assert a START condition on SDA and SCL.
- Assert a Repeated START condition on SDA and SCL.
- Write to the SSPBUF register initiating transmission of data/address.
- Generate a STOP condition on SDA and SCL.
- Configure the I²C port to receive data.
- Generate an Acknowledge condition at the end of a received byte of data.

Note: The MSSP Module, when configured in I²C Master mode, does not allow queuing of events. For instance, the user is not allowed to initiate a START condition and immediately write the SSPBUF register to initiate transmission before the START condition is complete. In this case, the SSPBUF will not be written to and the WCOL bit will be set, indicating that a write to the SSPBUF did not occur.

9.2.7.1 I²C Master Mode Operation

The master device generates all of the serial clock pulses and the START and STOP conditions. A transfer is ended with a STOP condition or with a Repeated START condition. Since the Repeated START condition is also the beginning of the next serial transfer, the I²C bus will not be released.

In Master Transmitter mode, serial data is output through SDA, while SCL outputs the serial clock. The first byte transmitted contains the slave address of the receiving device (7 bits) and the Read/Write (R/W) bit. In this case, the R/W bit will be logic '0'. Serial data is transmitted 8 bits at a time. After each byte is transmitted, an Acknowledge bit is received. START and STOP conditions are output to indicate the beginning and the end of a serial transfer.

In Master Receive mode, the first byte transmitted contains the slave address of the transmitting device (7 bits) and the R/W bit. In this case, the R/W bit will be logic '1'. Thus, the first byte transmitted is a 7-bit slave address followed by a '1' to indicate receive bit. Serial data is received via SDA, while SCL outputs the serial clock. Serial data is received 8 bits at a time. After each byte is received, an Acknowledge bit is transmitted. START and STOP conditions indicate the beginning and end of transmission.

The baud rate generator used for SPI mode operation is now used to set the SCL clock frequency for either 100 kHz, 400 kHz, or 1 MHz I²C operation. The baud rate generator reload value is contained in the lower 7 bits of the SSPADD register. The baud rate generator will automatically begin counting on a write to the

SSPBUF. Once the given operation is complete (i.e., transmission of the last data bit is followed by ACK), the internal clock will automatically stop counting and the SCL pin will remain in its last state.

A typical transmit sequence would go as follows:

- a) User generates a START condition by setting the START enable bit (SEN) in SSPCON2.
- b) SSPIF is set. The module will wait the required start time before any other operation takes place.
- c) User loads SSPBUF with address to transmit.
- d) Address is shifted out the SDA pin until all 8 bits are transmitted.
- e) MSSP module shifts in the ACK bit from the slave device and writes its value into the SSPCON2 register (SSPCON2<6>).
- f) MSSP module generates an interrupt at the end of the ninth clock cycle by setting SSPIF.
- g) User loads SSPBUF with eight bits of data.
- h) DATA is shifted out the SDA pin until all 8 bits are transmitted.
- i) MSSP module shifts in the ACK bit from the slave device, and writes its value into the SSPCON2 register (SSPCON2<6>).
- j) MSSP module generates an interrupt at the end of the ninth clock cycle by setting the SSPIF bit.
- k) User generates a STOP condition by setting the STOP enable bit, PEN, in SSPCON2.
- l) Interrupt is generated once the STOP condition is complete.

9.2.8 BAUD RATE GENERATOR

In I²C Master mode, the reload value for the BRG is located in the lower 7 bits of the SSPADD register (Figure 9-10). When the BRG is loaded with this value, the BRG counts down to 0 and stops until another reload has taken place. The BRG count is decremented twice per instruction cycle (T_{cy}), on the Q2 and Q4 clock.

In I²C Master mode, the BRG is reloaded automatically. If clock arbitration is taking place, the BRG will be reloaded when the SCL pin is sampled high (Figure 9-11).

Note: Baud Rate = $F_{osc} / (4 * (SSPADD + 1))$

FIGURE 9-10: BAUD RATE GENERATOR BLOCK DIAGRAM

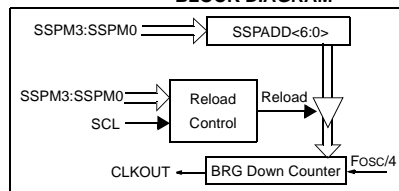
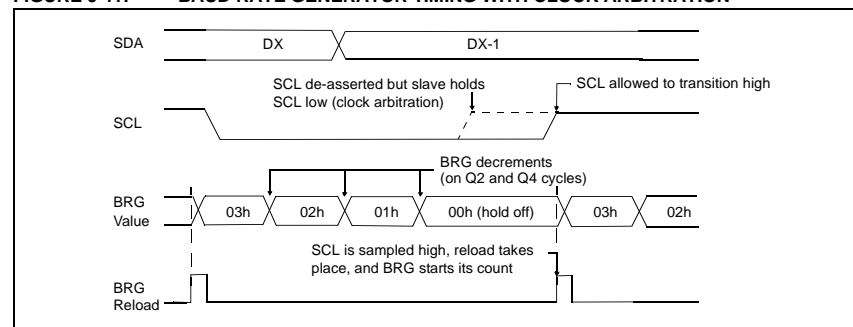


FIGURE 9-11: BAUD RATE GENERATOR TIMING WITH CLOCK ARBITRATION



9.2.9 I²C MASTER MODE START CONDITION TIMING

To initiate a START condition, the user sets the START condition enable bit, SEN (SSPCON2<0>). If the SDA and SCL pins are sampled high, the baud rate generator is reloaded with the contents of SSPADD<6:0> and starts its count. If SCL and SDA are both sampled high when the baud rate generator times out (T_{BRG}), the SDA pin is driven low. The action of the SDA being driven low while SCL is high is the START condition, and causes the S bit (SSPSTAT<3>) to be set. Following this, the baud rate generator is reloaded with the contents of SSPADD<6:0> and resumes its count. When the baud rate generator times out (T_{BRG}), the SEN bit (SSPCON2<0>) will be automatically cleared by hardware. The baud rate generator is suspended, leaving the SDA line held low, and the START condition is complete.

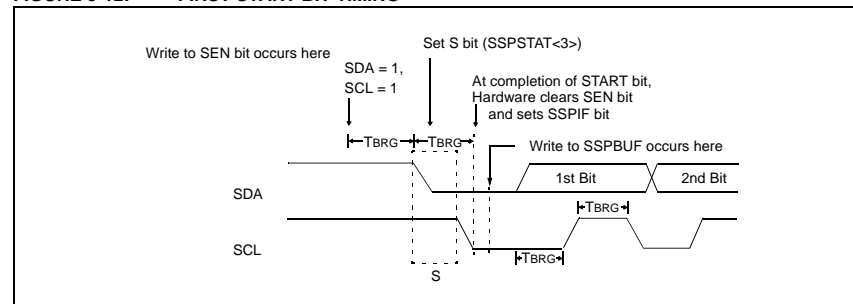
Note: If, at the beginning of START condition, the SDA and SCL pins are already sampled low, or if during the START condition the SCL line is sampled low before the SDA line is driven low, a bus collision occurs, the Bus Collision Interrupt Flag (BCLIF) is set, the START condition is aborted, and the I²C module is reset into its IDLE state.

9.2.9.1 WCOL Status Flag

If the user writes the SSPBUF when a START sequence is in progress, then WCOL is set and the contents of the buffer are unchanged (the write doesn't occur).

Note: Because queuing of events is not allowed, writing to the lower 5 bits of SSPCON2 is disabled until the START condition is complete.

FIGURE 9-12: FIRST START BIT TIMING



9.2.10 I²C MASTER MODE REPEATED START CONDITION TIMING

A Repeated START condition occurs when the RSEN bit (SSPCON2<1>) is programmed high and the I²C module is in the IDLE state. When the RSEN bit is set, the SCL pin is asserted low. When the SCL pin is sampled low, the baud rate generator is loaded with the contents of SSPADD<6:0> and begins counting. The SDA pin is released (brought high) for one baud rate generator count (TBRG). When the baud rate generator times out, if SDA is sampled high, the SCL pin will be de-asserted (brought high). When SCL is sampled high the baud rate generator is reloaded with the contents of SSPADD<6:0> and begins counting. SDA and SCL must be sampled high for one TBRG. This action is then followed by assertion of the SDA pin (SDA is low) for one TBRG, while SCL is high. Following this, the RSEN bit in the SSPCON2 register will be automatically cleared and the baud rate generator will not be reloaded, leaving the SDA pin held low. As soon as a START condition is detected on the SDA and SCL pins, the S bit (SSPSTAT<3>) will be set. The SSPIF bit will not be set until the baud rate generator has timed out.

- Note 1:** If RSEN is programmed while any other event is in progress, it will not take effect.
- Note 2:** A bus collision during the Repeated START condition occurs if:
- SDA is sampled low when SCL goes from low to high.
 - SCL goes low before SDA is asserted low. This may indicate that another master is attempting to transmit a data "1".

Immediately following the SSPIF bit getting set, the user may write the SSPBUF with the 7-bit address in 7-bit mode, or the default first address in 10-bit mode. After the first eight bits are transmitted and an ACK is received, the user may then transmit an additional eight bits of address (10-bit mode), or eight bits of data (7-bit mode).

9.2.10.1 WCOL Status Flag

If the user writes the SSPBUF when a Repeated START sequence is in progress, then WCOL is set and the contents of the buffer are unchanged (the write doesn't occur).

Note: Because queuing of events is not allowed, writing of the lower 5 bits of SSPCON2 is disabled until the Repeated START condition is complete.

9.2.11 I²C MASTER MODE TRANSMISSION

Transmission of a data byte, a 7-bit address, or either half of a 10-bit address, is accomplished by simply writing a value to SSPBUF register. This action will set the Buffer Full flag (BF) and allow the baud rate generator to begin counting and start the next transmission. Each bit of address/data will be shifted out onto the SDA pin after the falling edge of SCL is asserted (see data hold time spec). SCL is held low for one baud rate generator rollover count (TBRG). Data should be valid before SCL is released high (see data setup time spec). When the SCL pin is released high, it is held that way for TBRG. The data on the SDA pin must remain stable for that duration and some hold time after the next falling edge of SCL. After the eighth bit is shifted out (the falling edge of the eighth clock), the BF flag is cleared and the master releases SDA allowing the slave device being addressed to respond with an ACK bit during the ninth bit time, if an address match occurs or if data was received properly. The status of \overline{ACK} is read into the ACKDT on the falling edge of the ninth clock. If the master receives an Acknowledge, the Acknowledge Status bit (ACKSTAT) is cleared. If not, the bit is set. After the ninth clock, the SSPIF is set and the master clock (baud rate generator) is suspended until the next data byte is loaded into the SSPBUF, leaving SCL low and SDA unchanged (Figure 9-14).

After the write to the SSPBUF, each bit of address will be shifted out on the falling edge of SCL, until all seven address bits and the R/W bit are completed. On the falling edge of the eighth clock, the master will de-assert the SDA pin, allowing the slave to respond with an Acknowledge. On the falling edge of the ninth clock, the master will sample the SDA pin to see if the address was recognized by a slave. The status of the ACK bit is loaded into the ACKSTAT status bit (SSPCON2<6>). Following the falling edge of the ninth clock transmission of the address, the SSPIF is set, the BF flag is cleared, and the baud rate generator is turned off until another write to the SSPBUF takes place, holding SCL low and allowing SDA to float.

9.2.11.1 BF Status Flag

In Transmit mode, the BF bit (SSPSTAT<0>) is set when the CPU writes to SSPBUF and is cleared when all 8 bits are shifted out.

9.2.11.2 WCOL Status Flag

If the user writes the SSPBUF when a transmit is already in progress (i.e., SSPSR is still shifting out a data byte), then WCOL is set and the contents of the buffer are unchanged (the write doesn't occur).

WCOL must be cleared in software.

9.2.11.3 ACKSTAT Status Flag

In Transmit mode, the ACKSTAT bit (SSPCON2<6>) is cleared when the slave has sent an Acknowledge ($\overline{ACK} = 0$), and is set when the slave does not Acknowledge ($\overline{ACK} = 1$). A slave sends an Acknowledge when it has recognized its address (including a general call), or when the slave has properly received its data.

FIGURE 9-13: REPEAT START CONDITION WAVEFORM

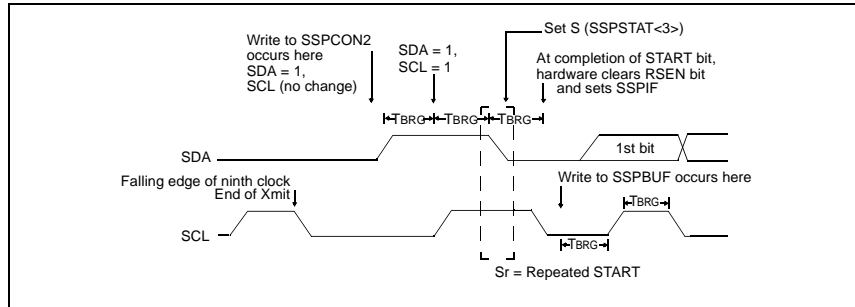
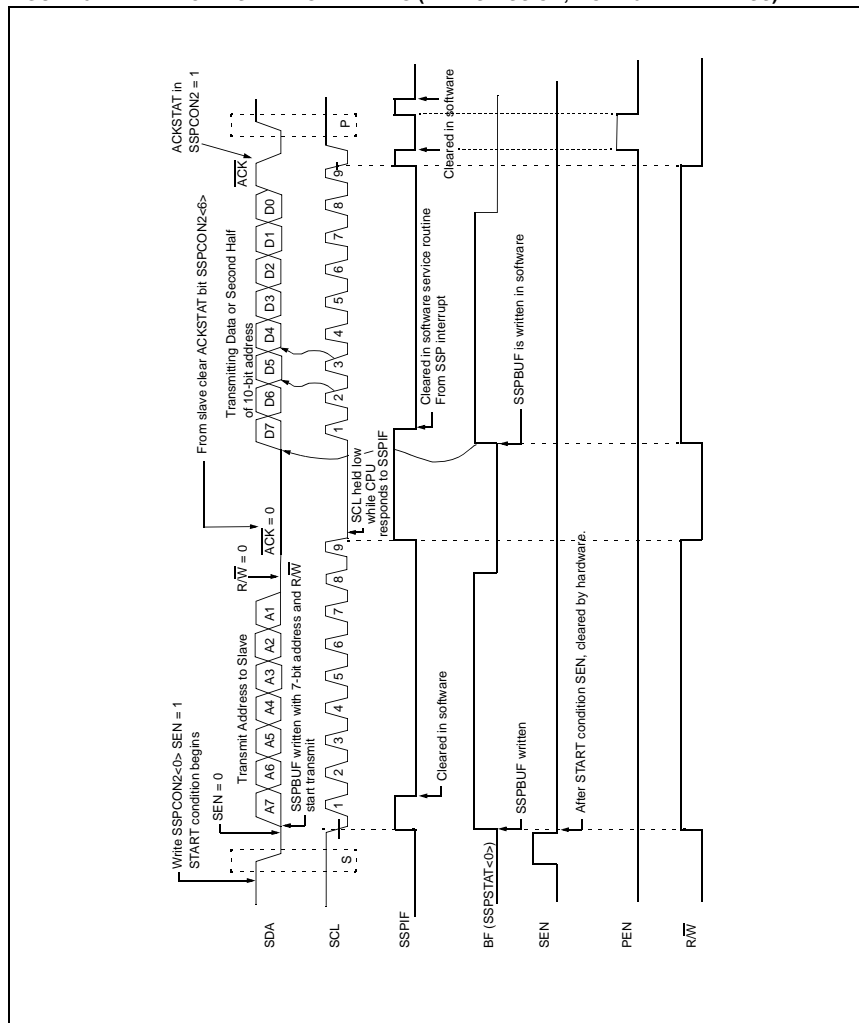


FIGURE 9-14: I²C MASTER MODE TIMING (TRANSMISSION, 7 OR 10-BIT ADDRESS)



9.2.12 I²C MASTER MODE RECEPTION

Master mode reception is enabled by programming the Receive Enable bit, RCEN (SSPCON2<3>).

Note: The SSP module must be in an IDLE state before the RCEN bit is set, or the RCEN bit will be disregarded.

The baud rate generator begins counting, and on each rollover, the state of the SCL pin changes (high to low/low to high), and data is shifted into the SSPSR. After the falling edge of the eighth clock, the receive enable flag is automatically cleared, the contents of the SSPSR are loaded into the SSPBUF, the BF flag is set, the SSPIF is set, and the baud rate generator is suspended from counting, holding SCL low. The SSP is now in IDLE state, awaiting the next command. When the buffer is read by the CPU, the BF flag is automatically cleared. The user can then send an Acknowledge bit at the end of reception, by setting the Acknowledge Sequence Enable bit, ACKEN (SSPCON2<4>).

9.2.12.1 BF Status Flag

In receive operation, BF is set when an address or data byte is loaded into SSPBUF from SSPSR. It is cleared when SSPBUF is read.

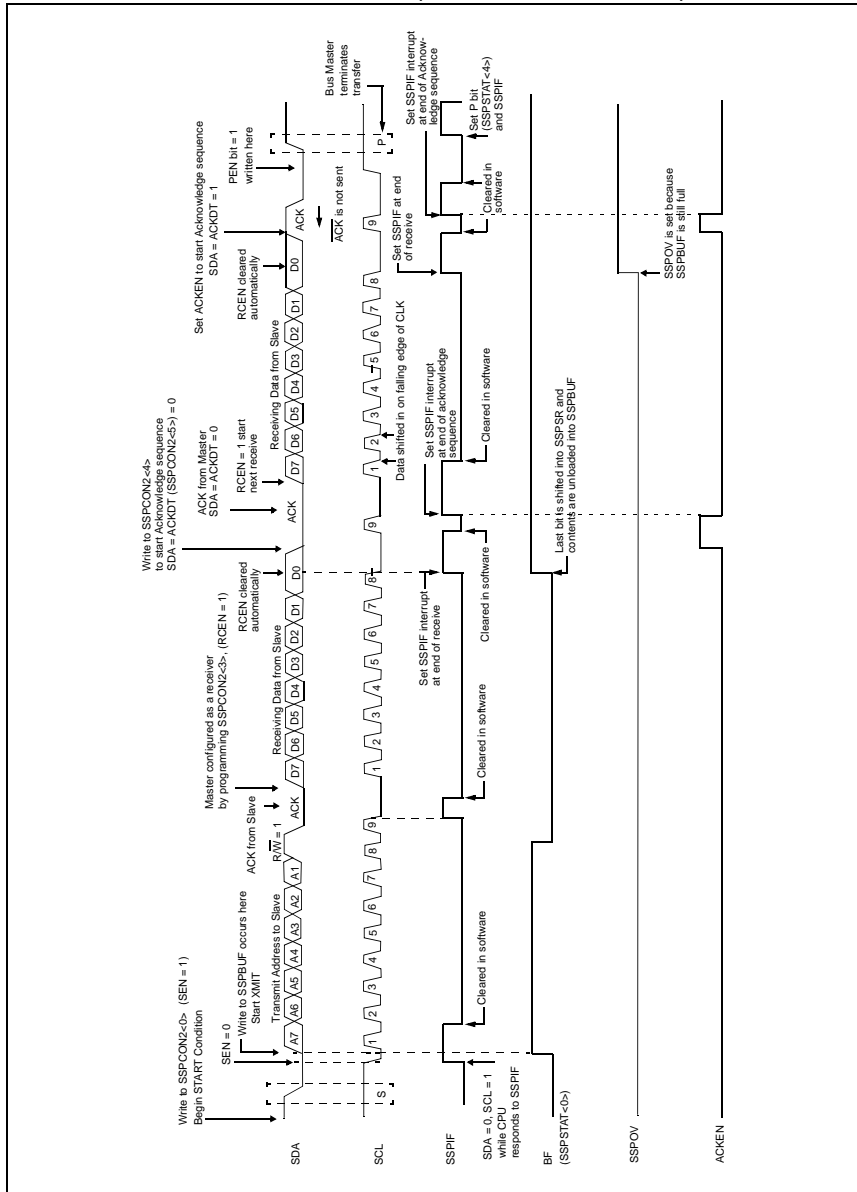
9.2.12.2 SSPOV Status Flag

In receive operation, SSPOV is set when 8 bits are received into the SSPSR, and the BF flag is already set from a previous reception.

9.2.12.3 WCOL Status Flag

If the user writes the SSPBUF when a receive is already in progress (i.e., SSPSR is still shifting in a data byte), then WCOL is set and the contents of the buffer are unchanged (the write doesn't occur).

FIGURE 9-15: I²C MASTER MODE TIMING (RECEPTION, 7-BIT ADDRESS)



9.2.13 ACKNOWLEDGE SEQUENCE TIMING

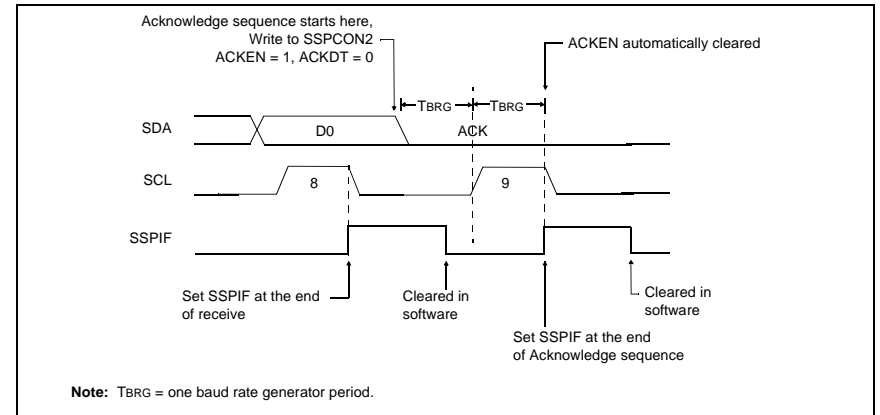
An Acknowledge sequence is enabled by setting the Acknowledge Sequence Enable bit, ACKEN (SSPCON2<4>). When this bit is set, the SCL pin is pulled low and the contents of the Acknowledge data bit is presented on the SDA pin. If the user wishes to generate an Acknowledge, the ACKDT bit should be cleared. If not, the user should set the ACKDT bit before starting an Acknowledge sequence. The baud rate generator then counts for one rollover period (TBRG), and the SCL pin is de-asserted high. When the SCL pin is sampled high (clock arbitration), the baud

rate generator counts for TBRG. The SCL pin is then pulled low. Following this, the ACKEN bit is automatically cleared, the baud rate generator is turned off, and the SSP module then goes into IDLE mode (Figure 9-16).

9.2.13.1 WCOL Status Flag

If the user writes the SSPBUF when an Acknowledge sequence is in progress, the WCOL is set and the contents of the buffer are unchanged (the write doesn't occur).

FIGURE 9-16: ACKNOWLEDGE SEQUENCE WAVEFORM



PIC16F87X

9.2.14 STOP CONDITION TIMING

A STOP bit is asserted on the SDA pin at the end of a receive/transmit by setting the Stop Sequence Enable bit, PEN (SSPCON2<2>). At the end of a receive/transmit, the SCL line is held low after the falling edge of the ninth clock. When the PEN bit is set, the master will assert the SDA line low. When the SDA line is sampled low, the baud rate generator is reloaded and counts down to 0. When the baud rate generator times out, the SCL pin will be brought high, and one TBRG (baud rate generator rollover count) later, the SDA pin will be de-asserted. When the SDA pin is sampled high

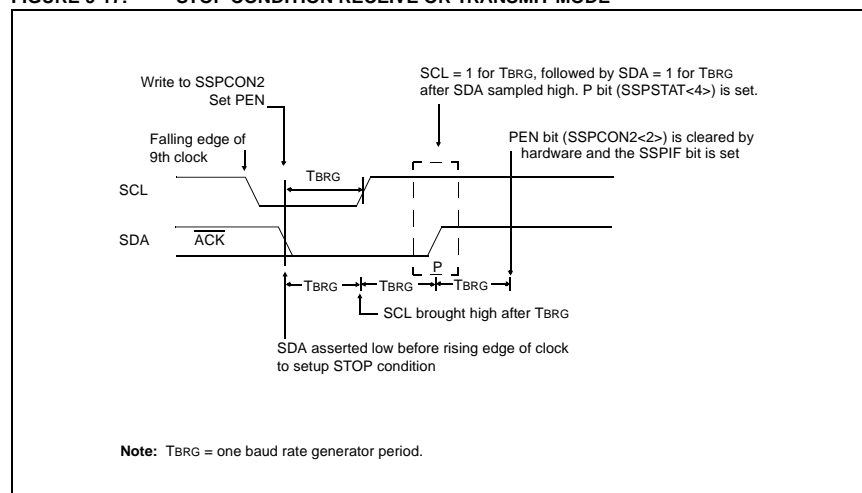
while SCL is high, the P bit (SSPSTAT<4>) is set. A TBRG later, the PEN bit is cleared and the SSPIF bit is set (Figure 9-17).

Whenever the firmware decides to take control of the bus, it will first determine if the bus is busy by checking the S and P bits in the SSPSTAT register. If the bus is busy, then the CPU can be interrupted (notified) when a STOP bit is detected (i.e., bus is free).

9.2.14.1 WCOL Status Flag

If the user writes the SSPBUF when a STOP sequence is in progress, then WCOL is set and the contents of the buffer are unchanged (the write doesn't occur).

FIGURE 9-17: STOP CONDITION RECEIVE OR TRANSMIT MODE



PIC16F87X

9.2.15 CLOCK ARBITRATION

Clock arbitration occurs when the master, during any receive, transmit, or Repeated START/STOP condition, de-asserts the SCL pin (SCL allowed to float high). When the SCL pin is allowed to float high, the baud rate generator (BRG) is suspended from counting until the SCL pin is actually sampled high. When the SCL pin is sampled high, the baud rate generator is reloaded with the contents of SSPADD<6:0> and begins counting. This ensures that the SCL high time will always be at least one BRG rollover count in the event that the clock is held low by an external device (Figure 9-18).

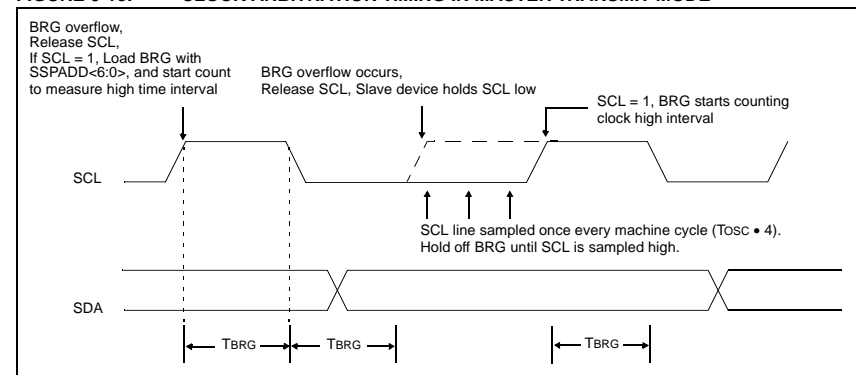
9.2.16 SLEEP OPERATION

While in SLEEP mode, the I²C module can receive addresses or data, and when an address match or complete byte transfer occurs, wake the processor from SLEEP (if the SSP interrupt is enabled).

9.2.17 EFFECTS OF A RESET

A RESET disables the SSP module and terminates the current transfer.

FIGURE 9-18: CLOCK ARBITRATION TIMING IN MASTER TRANSMIT MODE



PIC16F87X

9.2.18 MULTI-MASTER COMMUNICATION, BUS COLLISION, AND BUS ARBITRATION

Multi-Master mode support is achieved by bus arbitration. When the master outputs address/data bits onto the SDA pin, arbitration takes place when the master outputs a '1' on SDA, by letting SDA float high and another master asserts a '0'. When the SCL pin floats high, data should be stable. If the expected data on SDA is a '1' and the data sampled on the SDA pin = '0', a bus collision has taken place. The master will set the Bus Collision Interrupt Flag, BCLIF and reset the I²C port to its IDLE state (Figure 9-19).

If a transmit was in progress when the bus collision occurred, the transmission is halted, the BF flag is cleared, the SDA and SCL lines are de-asserted, and the SSPBUF can be written to. When the user services the bus collision Interrupt Service Routine, and if the I²C bus is free, the user can resume communication by asserting a START condition.

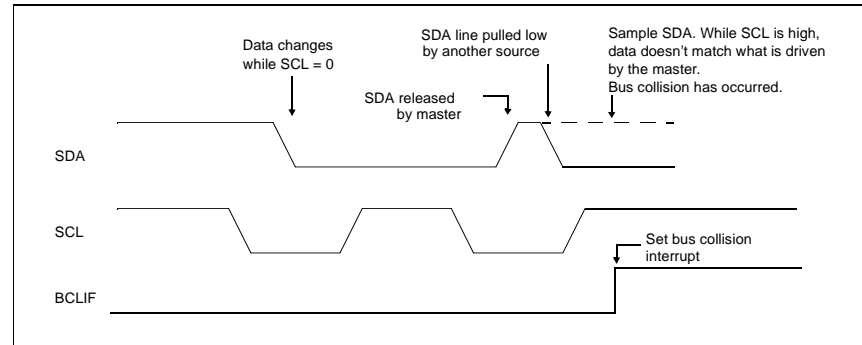
If a START, Repeated START, STOP, or Acknowledge condition was in progress when the bus collision occurred, the condition is aborted, the SDA and SCL lines are de-asserted, and the respective control bits in the SSPCON2 register are cleared. When the user services the bus collision Interrupt Service Routine, and if the I²C bus is free, the user can resume communication by asserting a START condition.

The master will continue to monitor the SDA and SCL pins and if a STOP condition occurs, the SSPIF bit will be set.

A write to the SSPBUF will start the transmission of data at the first data bit, regardless of where the transmitter left off when the bus collision occurred.

In Multi-Master mode, the interrupt generation on the detection of START and STOP conditions allows the determination of when the bus is free. Control of the I²C bus can be taken when the P bit is set in the SSPSTAT register, or the bus is idle and the S and P bits are cleared.

FIGURE 9-19: BUS COLLISION TIMING FOR TRANSMIT AND ACKNOWLEDGE



PIC16F87X

9.2.18.1 Bus Collision During a START Condition

During a START condition, a bus collision occurs if:

- SDA or SCL are sampled low at the beginning of the START condition (Figure 9-20).
- SCL is sampled low before SDA is asserted low (Figure 9-21).

During a START condition, both the SDA and the SCL pins are monitored. If either the SDA pin or the SCL pin is already low, then these events all occur:

- the START condition is aborted,
- and the BCLIF flag is set,
- and the SSP module is reset to its IDLE state (Figure 9-20).

The START condition begins with the SDA and SCL pins de-asserted. When the SDA pin is sampled high, the baud rate generator is loaded from SSPADD<6:0> and counts down to 0. If the SCL pin is sampled low while SDA is high, a bus collision occurs, because it is assumed that another master is attempting to drive a data '1' during the START condition.

If the SDA pin is sampled low during this count, the BRG is reset and the SDA line is asserted early (Figure 9-22). If, however, a '1' is sampled on the SDA pin, the SDA pin is asserted low at the end of the BRG count. The baud rate generator is then reloaded and counts down to 0. During this time, if the SCL pins are sampled as '0', a bus collision does not occur. At the end of the BRG count, the SCL pin is asserted low.

Note: The reason that bus collision is not a factor during a START condition is that no two bus masters can assert a START condition at the exact same time. Therefore, one master will always assert SDA before the other. This condition does not cause a bus collision, because the two masters must be allowed to arbitrate the first address following the START condition. If the address is the same, arbitration must be allowed to continue into the data portion, Repeated START, or STOP conditions.

FIGURE 9-20: BUS COLLISION DURING START CONDITION (SDA ONLY)

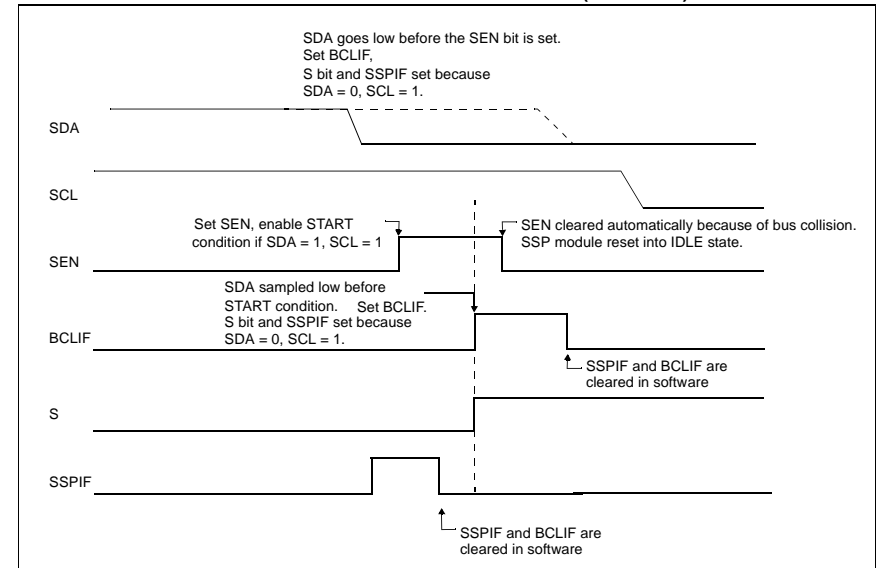


FIGURE 9-21: BUS COLLISION DURING START CONDITION (SCL = 0)

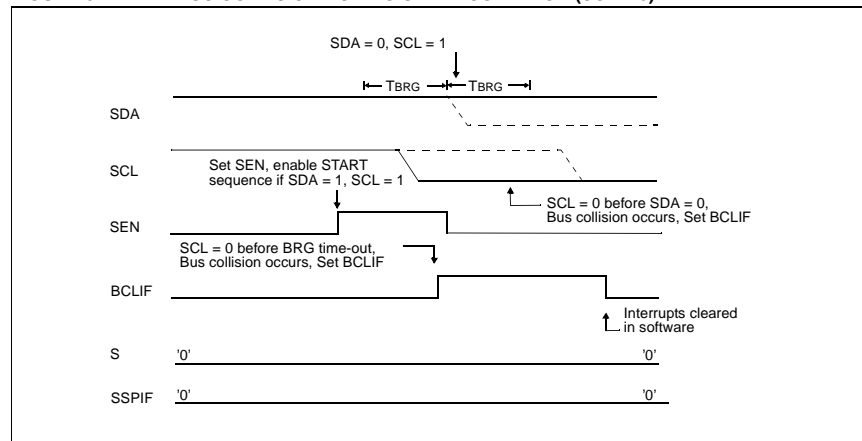
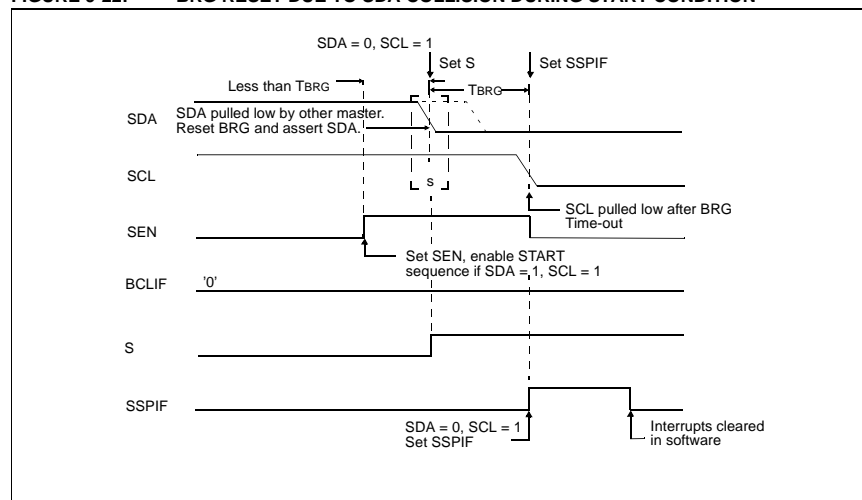


FIGURE 9-22: BRG RESET DUE TO SDA COLLISION DURING START CONDITION



9.2.18.2 Bus Collision During a Repeated START Condition

During a Repeated START condition, a bus collision occurs if:

- A low level is sampled on SDA when SCL goes from low level to high level.
- SCL goes low before SDA is asserted low, indicating that another master is attempting to transmit a data '1'.

When the user de-asserts SDA and the pin is allowed to float high, the BRG is loaded with SSPADD<6:0> and counts down to 0. The SCL pin is then de-asserted, and when sampled high, the SDA pin is sampled. If SDA is low, a bus collision has occurred (i.e., another master is attempting to transmit a data '0'). If, however,

SDA is sampled high, the BRG is reloaded and begins counting. If SDA goes from high to low before the BRG times out, no bus collision occurs, because no two masters can assert SDA at exactly the same time.

If, however, SCL goes from high to low before the BRG times out and SDA has not already been asserted, a bus collision occurs. In this case, another master is attempting to transmit a data '1' during the Repeated START condition.

If at the end of the BRG time-out, both SCL and SDA are still high, the SDA pin is driven low, the BRG is reloaded and begins counting. At the end of the count, regardless of the status of the SCL pin, the SCL pin is driven low and the Repeated START condition is complete (Figure 9-23).

FIGURE 9-23: BUS COLLISION DURING A REPEATED START CONDITION (CASE 1)

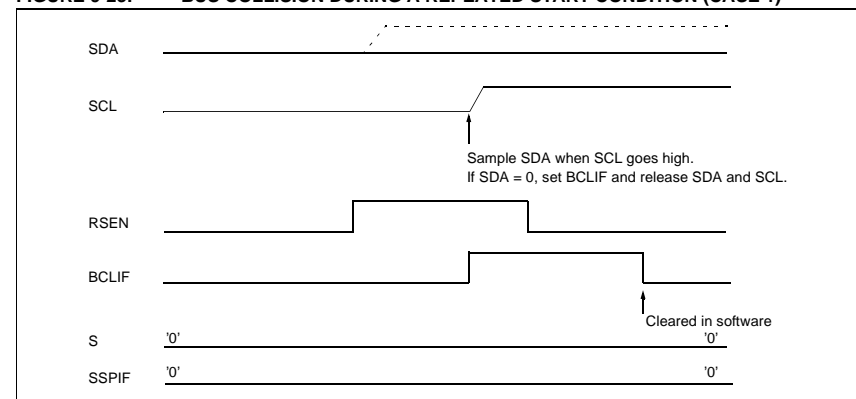
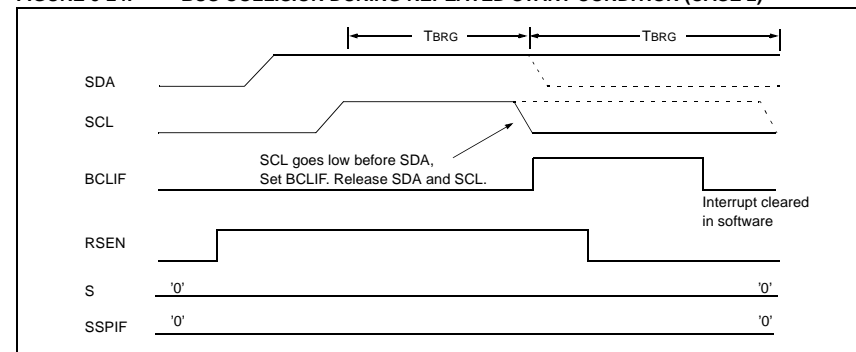


FIGURE 9-24: BUS COLLISION DURING REPEATED START CONDITION (CASE 2)



PIC16F87X

9.2.18.3 Bus Collision During a STOP Condition

Bus collision occurs during a STOP condition if:

- After the SDA pin has been de-asserted and allowed to float high, SDA is sampled low after the BRG has timed out.
- After the SCL pin is de-asserted, SCL is sampled low before SDA goes high.

The STOP condition begins with SDA asserted low. When SDA is sampled low, the SCL pin is allowed to float. When the pin is sampled high (clock arbitration), the baud rate generator is loaded with SSPADD<6:0> and counts down to 0. After the BRG times out, SDA is sampled. If SDA is sampled low, a bus collision has occurred. This is due to another master attempting to drive a data '0'. If the SCL pin is sampled low before SDA is allowed to float high, a bus collision occurs. This is a case of another master attempting to drive a data '0' (Figure 9-25).

FIGURE 9-25: BUS COLLISION DURING A STOP CONDITION (CASE 1)

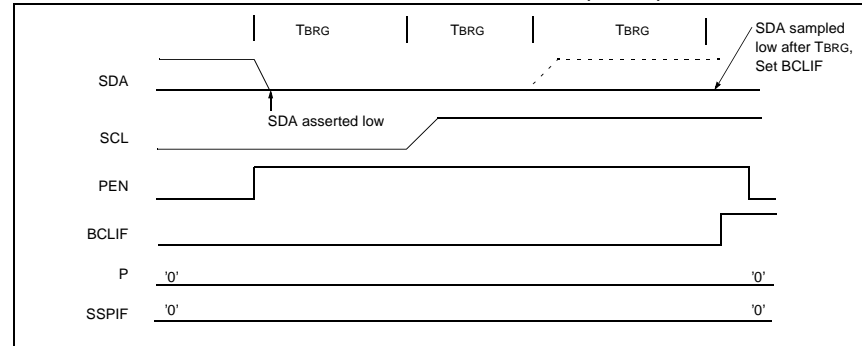
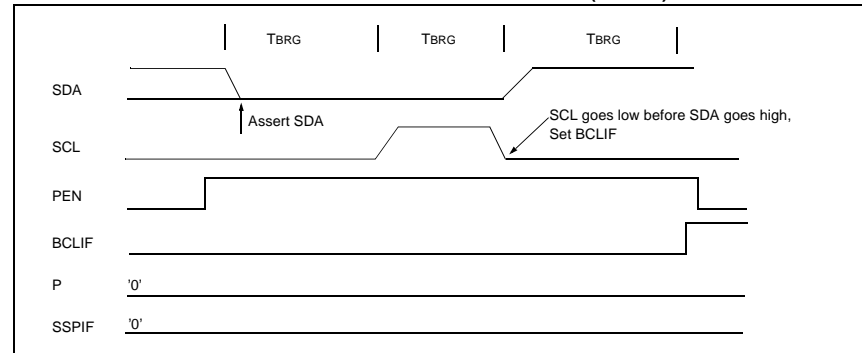


FIGURE 9-26: BUS COLLISION DURING A STOP CONDITION (CASE 2)



PIC16F87X

9.3 Connection Considerations for I²C Bus

For standard-mode I²C bus devices, the values of resistors R_p and R_s in Figure 9-27 depend on the following parameters:

- Supply voltage
- Bus capacitance
- Number of connected devices (input current + leakage current)

The supply voltage limits the minimum value of resistor R_p due to the specified minimum sink current of 3 mA at $V_{OL\ max} = 0.4V$, for the specified output stages. For

example, with a supply voltage of $V_{DD} = 5V \pm 10\%$ and $V_{OL\ max} = 0.4V$ at 3 mA, $R_{p\ min} = (5.5-0.4)/0.003 = 1.7\ k\Omega$. V_{DD} as a function of R_p is shown in Figure 9-27. The desired noise margin of $0.1V_{DD}$ for the low level limits the maximum value of R_s . Series resistors are optional and used to improve ESD susceptibility.

The bus capacitance is the total capacitance of wire, connections, and pins. This capacitance limits the maximum value of R_p due to the specified rise time (Figure 9-27).

The SMP bit is the slew rate control enabled bit. This bit is in the SSPSTAT register, and controls the slew rate of the I/O pins when in I²C mode (master or slave).

FIGURE 9-27: SAMPLE DEVICE CONFIGURATION FOR I²C BUS

