Section 58. Master Synchronous Serial Port (MSSP)

HIGHLIGHTS

This section of the manual contains the following major topics:

58.1 OVERVIEW

The Master Synchronous Serial Port (MSSP) module is an 8-bit 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 (I^2C™)
  - Full Master mode
  - Slave mode (with General Call Address)

The SPI interface supports these modes in hardware:

- Master mode
- Slave mode
- Daisy-Chaining Operation in Slave mode
- Synchronized Slave Operation

The I^2C interface supports the following modes in hardware:

- Master mode
- Multi-Master mode
- Slave mode with 10-Bit and 7-Bit Addressing and Address Masking
- Byte NACKing
- Selectable Address and Data Hold and Interrupt Masking

PIC24F devices that use the MSSP module may incorporate more than one instance of the module. Each module functions independently of the others.

58.2 REGISTERS

Each MSSP module has a total of six associated registers that control and monitor operation. These include a STATUS register (SSPxSTAT), three control registers (SSPxCON1, SSPxCON2 and SSPxCON3) and two registers to configure I^2C operations (SSPxADD and SSPxMSK).

The use of these registers and their individual bits differ significantly depending on whether the MSSP module is operated in SPI or I^2C mode. For clarity, the registers that change the most in bit names and/or functions are shown separately for each mode.

Each MSSP module uses three control registers for SPI mode operation. These are:

- MSSP Status Register (SSPxSTAT) (Register 58-1)
- MSSP Control Register 1 (SSPxCON1) (Register 58-2)
- MSSP Control Register 3 (SSPxCON3) (Register 58-3)

The SSPxCON1 and SSPxCON3 registers are readable and writable. The lower 6 bits of the SSPxSTAT are read-only. The upper two bits of the SSPxSTAT are read/write.

The MSSP module uses six control registers for I^2C operation. These are:

- MSSP Status Register (SSPxSTAT) (Register 58-4)
- MSSP Control Register 1 (SSPxCON1) (Register 58-5)
- MSSP Control Register 2 (SSPxCON2) (Register 58-6)
- MSSP Control Register 3 (SSPxCON3) (Register 58-7)
- MSSP Address Register (SSPxADD) (Register 58-8)
- MSSP 7-Bit Address Mask Register (SSPxMSK) (Register 58-9)

In addition, each MSSP module has two data registers: the MSSP Shift Register (SSPxSR) and the Serial Receive/Transmit Buffer register (SSPxBUF). SSPxSR is the shift register used for shifting data in or out; it is not memory mapped and therefore, cannot be directly accessed.

SSPxBUF is the buffer register to which data bytes are written to or read from. In receive operations, SSPxSR and SSPxBUF together, create a double-buffered receiver. When SSPxSR receives a complete byte, it is transferred to SSPxBUF and the SSPxIF interrupt flag is set.

Register 58-1: SSPxSTAT: MSSPx Status Register (SPI Mode)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
SMP	CKE(1)	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF
bit 7							bit 0

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

bit 15-8 **Unimplemented:** Read as '0'

bit 7 **SMP:** Sample bit
SPI 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
SPI Slave mode:
SMP must be cleared when SPI is used in Slave mode.

bit 6 **CKE:** SPI Clock Select bit(1)
1 = Transmit occurs on transition from active to Idle clock state
0 = Transmit occurs on transition from Idle to active clock state

bit 5 **D/$\overline{A}$:** Data/$\overline{\text{Address}}$ bit
Used in I^{2}C™ mode only.

bit 4 **P:** Stop bit
Used in I^{2}C mode only. This bit is cleared when the MSSP module is disabled; SSPEN is cleared.

bit 3 **S:** Start bit
Used in I^{2}C mode only.

bit 2 **R/$\overline{W}$:** Read/$\overline{\text{Write}}$ Information bit
Used in I^{2}C mode only.

bit 1 **UA:** Update Address bit
Used in I^{2}C mode only.

bit 0 **BF:** Buffer Full Status bit (Receive mode only)
1 = Receive is complete, SSPxBUF is full
0 = Receive is not complete, SSPxBUF is empty

Note 1: The polarity of the clock state is set by the CKP bit (SSPxCON1<4>).

Register 58-2: SSPxCON1: MSSPx Control Register 1 (SPI Mode)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

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(1)	SSPEN(2)	CKP	SSPM3(3)	SSPM2(3)	SSPM1(3)	SSPM0(3)
bit 7							bit 0

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

bit 15-8 **Unimplemented:** Read as '0'

bit 7 **WCOL:** Write Collision Detect bit
1 = The SSPxBUF register is written while it is still transmitting the previous word (must be cleared in software)
0 = No collision

bit 6 **SSPOV:** Receive Overflow Indicator bit(1)
SPI Slave mode:
1 = A new byte is received while the SSPxBUF register is still holding the previous data. In case of overflow, the data in SSPxSR is lost. Overflow can only occur in Slave mode. The user must read the SSPxBUF, even if only transmitting data, to avoid setting overflow (must be cleared in software).
0 = No overflow

bit 5 **SSPEN:** Master Synchronous Serial Port Enable bit(2)
1 = Enables serial port and configures SCKx, SDOx, SDIx and $\overline{SSx}$ as serial port pins
0 = Disables serial port and configures these pins as I/O port pins

bit 4 **CKP:** Clock Polarity Select bit
1 = Idle state for clock is a high level
0 = Idle state for clock is a low level

bit 3-0 **SSPM<3:0>:** Master Synchronous Serial Port Mode Select bits(3)
1010 = SPI Master mode, clock = FOSC/(2 * ([SSPxADD] + 1))
0101 = SPI Slave mode, clock = SCKx pin, $\overline{SSx}$ pin control is disabled, $\overline{SSx}$ can be used as an I/O pin
0100 = SPI Slave mode, clock = SCKx pin, $\overline{SSx}$ pin control is enabled
0011 = SPI Master mode, clock = TMR2 output/2
0010 = SPI Master mode, clock = FOSC/32
0001 = SPI Master mode, clock = FOSC/8
0000 = SPI Master mode, clock = FOSC/2

Note 1: In Master mode, the overflow bit is not set since each new reception (and transmission) is initiated by writing to the SSPxBUF register.
2: When enabled, these pins must be properly configured as input or output.
3: Bit combinations not specifically listed here are either reserved or implemented in I^2C™ mode only.

Register 58-3: SSPxCON3: MSSPx Control Register 3 (SPI Mode)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ACKTIM	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN
bit 7							bit 0

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

bit 15-8 **Unimplemented:** Read as '0'

bit 7 **ACKTIM:** Acknowledge Time Status bit (I^2C™ mode only)
Unused in SPI mode.

bit 6 **PCIE**: Stop Condition Interrupt Enable bit (I^2C mode only)
Unused in SPI mode.

bit 5 **SCIE**: Start Condition Interrupt Enable bit (I^2C mode only)
Unused in SPI mode.

bit 4 **BOEN:** Buffer Overwrite Enable bit
In SPI Slave mode:
1 = SSPxBUF updates every time that a new data byte is shifted in, ignoring the BF bit
0 = If a new byte is received with the BF bit of the SSPxSTAT register already set, the SSPOV bit of the SSPxCON1 register is set and the buffer is not updated

bit 3 **SDAHT:** SDAx Hold Time Selection bit (I^2C mode only)
Unused in SPI mode.

bit 2 **SBCDE:** Slave Mode Bus Collision Detect Enable bit (I^2C Slave mode only)
Unused in SPI mode.

bit 1 **AHEN:** Address Hold Enable bit (I^2C Slave mode only)
Unused in SPI mode.

bit 0 **DHEN:** Data Hold Enable bit (I^2C Slave mode only)
Unused in SPI mode.

Register 58-4: SSPxSTAT: MSSPx Status Register (I^2C™ Mode)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
SMP	CKE	D/$\overline{A}$	P(1)	S(1)	R/$\overline{W}$(2,3)	UA	BF
bit 7							bit 0

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

bit 15-8 **Unimplemented:** Read as '0'

bit 7 **SMP:** Slew Rate Control bit

In Master or Slave mode:
1 = Slew rate control is disabled for Standard Speed mode (100 kHz and 1 MHz)
0 = Slew rate control is enabled for High-Speed mode (400 kHz)

bit 6 **CKE:** SMBus Select bit

In Master or Slave mode:
1 = Enables SMBus specific inputs
0 = Disables SMBus specific inputs

bit 5 **D/$\overline{A}$:** Data/$\overline{\text{Address}}$ bit

In Master mode:
Reserved.

In Slave mode:
1 = Indicates that the last byte received or transmitted was data
0 = Indicates that the last byte received or transmitted was address

bit 4 **P:** Stop bit(1)

1 = Indicates that a Stop bit has been detected last
0 = Stop bit was not detected last

bit 3 **S:** Start bit(1)

1 = Indicates that a Start bit has been detected last
0 = Start bit was not detected last

bit 2 **R/$\overline{W}$:** Read/$\overline{\text{Write}}$ Information bit(2,3)

In Slave mode:
1 = Read
0 = Write

In Master mode:
1 = Transmit is in progress
0 = Transmit is not in progress

Note 1: This bit is cleared on Reset and when SSPEN is cleared.

2: This bit holds the R/$\overline{W}$ 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 $\overline{\text{ACK}}$ bit.

3: ORing this bit with SEN, RSEN, PEN, RCEN or ACKEN will indicate if the MSSP is in Active mode.

Register 58-4: SSPxSTAT: MSSPx Status Register (I^2C™ Mode) (Continued)

bit 1 **UA:** Update Address bit (10-Bit Slave mode only)
1 = Indicates that the user needs to update the address in the SSPxADD register
0 = Address does not need to be updated

bit 0 **BF:** Buffer Full Status bit
In Transmit mode:
1 = SSPxBUF is full
0 = SSPxBUF is empty
In Receive mode:
1 = SSPxBUF is full (does not include the $\overline{\text{ACK}}$ and Stop bits)
0 = SSPxBUF is empty (does not include the $\overline{\text{ACK}}$ and Stop bits)

Note 1: This bit is cleared on Reset and when SSPEN is cleared.
2: This bit holds the R/$\overline{\text{W}}$ 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 $\overline{\text{ACK}}$ bit.
3: ORing this bit with SEN, RSEN, PEN, RCEN or ACKEN will indicate if the MSSP is in Active mode.

Register 58-5: SSPxCON1: MSSPx Control Register 1 (I²C™ Mode)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

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(1)	CKP	SSPM3(2)	SSPM2(2)	SSPM1(2)	SSPM0(2)
bit 7							bit 0

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

bit 15-8 **Unimplemented:** Read as '0'

bit 7 **WCOL:** Write Collision Detect bit

In Master Transmit mode:
1 = A write to the SSPxBUF register was attempted while the I²C conditions were not valid for a transmission to be started (must be cleared in software)
0 = No collision

In Slave Transmit mode:
1 = The SSPxBUF register is written while it is still transmitting the previous word (must be cleared in software)
0 = No collision

In Receive mode (Master or Slave modes):
This is a "don't care" bit.

bit 6 **SSPOV:** Receive Overflow Indicator bit

In Receive mode:
1 = A byte is received while the SSPxBUF register is still holding the previous byte (must be cleared in software)
0 = No overflow

In Transmit mode:
This is a "don't care" bit in Transmit mode.

bit 5 **SSPEN:** Master Synchronous Serial Port Enable bit(1)

1 = Enables the serial port and configures the SDAx and SCLx pins as the serial port pins
0 = Disables serial port and configures these pins as I/O port pins

bit 4 **CKP:** SCKx Release Control bit

In Slave mode:
1 = Releases clock
0 = Holds clock low (clock stretch), used to ensure data setup time

In Master mode:
Unused in this mode.

bit 3-0 **SSPM<3:0>:** Master Synchronous Serial Port Mode Select bits(2)

1111 = I²C Slave mode, 10-bit addressing with Start and Stop bit interrupts is enabled
1110 = I²C Slave mode, 7-bit addressing with Start and Stop bit interrupts is enabled
1011 = I²C Firmware Controlled Master mode (Slave Idle)
1000 = I²C Master mode, Clock = FOSC/(2 * ([SSPxADD] + 1))(3)
0111 = I²C Slave mode, 10-bit addressing
0110 = I²C Slave mode, 7-bit addressing

Note 1: When enabled, the SDAx and SCLx pins must be configured as inputs.
2: Bit combinations not specifically listed here are either reserved or implemented in SPI mode only.
3: This mode is only available when 7-Bit Masking mode is selected (SSPxMSK Configuration bit is '1').

Register 58-6: SSPxCON2: MSSPx Control Register 2 (I^2C™ Mode)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

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(1)	ACKEN(2)	RCEN(2)	PEN(2)	RSEN(2)	SEN(2)
bit 7							bit 0

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

bit 15-8 **Unimplemented:** Read as '0'

bit 7 **GCEN:** General Call Enable bit (Slave mode only)
1 = Enables interrupt when a General Call Address (0000h) is received in the SSPxSR
0 = General Call Address is disabled

bit 6 **ACKSTAT:** Acknowledge Status bit (Master Transmit mode only)
1 = Acknowledge was not received from slave
0 = Acknowledge was received from slave

bit 5 **ACKDT:** Acknowledge Data bit (Master Receive mode only)(1)
1 = Not Acknowledge
0 = Acknowledge

bit 4 **ACKEN:** Acknowledge Sequence Enable bit (Master mode only)(2)
1 = Initiates Acknowledge sequence on the SDAx and SCLx pins and transmit ACKDT data bit; automatically cleared by hardware
0 = Acknowledge sequence is Idle

bit 3 **RCEN:** Receive Enable bit (Master Receive mode only)(2)
1 = Enables Receive mode for I^2C
0 = Receive is Idle

bit 2 **PEN:** Stop Condition Enable bit (Master mode only)(2)
1 = Initiates Stop condition on SDAx and SCLx pins; automatically cleared by hardware.
0 = Stop condition is Idle

bit 1 **RSEN:** Repeated Start Condition Enable bit (Master mode only)(2)
1 = Initiates Repeated Start condition on the SDAx and SCLx pins; automatically cleared by hardware
0 = Repeated Start condition is Idle

bit 0 **SEN:** Start Condition Enable bit(2)
Master Mode:
1 = Initiates Start condition on SDAx and SCLx pins; automatically cleared by hardware
0 = Start condition is Idle
Slave Mode:
1 = Clock stretching is enabled for both slave transmit and slave receive (stretch is enabled)
0 = Clock stretching is disabled

Note 1: Value that will be transmitted when the user initiates an Acknowledge sequence at the end of a receive.
2: If the I^2C module is active, these bits may not be set (no spooling) and the SSPxBUF may not be written (or writes to the SSPxBUF are disabled).

Register 58-7: SSPxCON3: MSSPx Control Register 3 (I^2C™ Mode)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ACKTIM(1)	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN
bit 7							bit 0

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

bit 15-8 **Unimplemented:** Read as '0'

bit 7 **ACKTIM:** Acknowledge Time Status bit(1)
1 = Indicates the I^2C bus is in an Acknowledge sequence, set on the 8th falling edge of the SCLx clock
0 = Not an Acknowledge sequence; cleared on the 9th rising edge of the SCLx clock

bit 6 **PCIE**: Stop Condition Interrupt Enable bit
1 = Enables interrupt on detection of Stop condition
0 = Stop detection interrupts are disabled(2)

bit 5 **SCIE**: Start Condition Interrupt Enable bit
1 = Enables interrupt on detection of Start or Restart conditions
0 = Start detection interrupts are disabled(2)

bit 4 **BOEN:** Buffer Overwrite Enable bit
I^2C Master mode:
This bit is ignored.
I^2C Slave mode:
1 = SSPxBUF is updated and an $\overline{ACK}$ is generated for a received address/data byte, ignoring the state of the SSPOV bit only if the BF bit = 0
0 = SSPxBUF is only updated when SSPOV is clear

bit 3 **SDAHT:** SDAx Hold Time Selection bit
1 = Minimum of 300 ns hold time on SDAx after the falling edge of SCLx
0 = Minimum of 100 ns hold time on SDAx after the falling edge of SCLx

bit 2 **SBCDE:** Slave Mode Bus Collision Detect Enable bit (Slave mode only)
1 = Enables slave bus collision interrupts
0 = Slave bus collision interrupts are disabled

bit 1 **AHEN:** Address Hold Enable bit (Slave mode only)
1 = Following the 8th falling edge of SCLx for a matching received address byte; CKP bit of the SSPxCON1 register will be cleared and the SCLx will be held low
0 = Address holding is disabled

bit 0 **DHEN:** Data Hold Enable bit (Slave mode only)
1 = Following the 8th falling edge of SCLx for a received data byte; slave hardware clears the CKP bit of the SSPxCON1 register and SCLx is held low
0 = Data holding is disabled

Note 1: The ACKTIM status bit is active only when the AHEN bit or DHEN bit is set.
2: This bit has no effect in Slave modes for which Start and Stop condition detection is explicitly listed as enabled.

Register 58-8: SSPxADD: MSSPx Slave Address/Baud Rate Generator Register

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADD7	ADD6	ADD5	ADD4	ADD3	ADD2	ADD1	ADD0
bit 7							bit 0

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

bit 15-8 **Unimplemented:** Read as '0'

bit 7-0 **ADD<7:0>:** Slave Address/Baud Rate Generator Value bits

SPI Master and I^2C Master modes:
Reloads the value for the Baud Rate Generator; clock period is (([SSPxADD] + 1) * 2)/FOSC.

I^2C Slave modes:
Represents 7 or 8 bits of the slave address, depending on the addressing mode used.

7-Bit mode: Address is ADD<7:1>; ADD<0> is ignored.

10-Bit LSb mode: ADD<7:0> are the Least Significant bits of the address.

10-Bit MSb mode: ADD<2:1> are the two Most Significant bits of the address; ADD<7:3> are always '11110' as a specification requirement; ADD<0> is ignored.

Register 58-9: SSPxMSK: I^2C™ Slave Address Mask Register

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0(1)
bit 7							bit 0

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

bit 15-8 **Unimplemented:** Read as '0'

bit 7-0 **MSK<7:0>:** Slave Address Mask Select bit(1)

1 = Masking of corresponding bit of SSPxADD is enabled
0 = Masking of corresponding bit of SSPxADD is disabled

Note 1: MSK0 is not used as a mask bit in 7-bit addressing.

58.3 BAUD RATE GENERATOR

Each MSSP module has a Baud Rate Generator (BRG) that can serve as a clock source in both SPI and I^2C Master modes. In I^2C Master mode, the BRG is the only available clock source.

The clock speed, FCLOCK, is determined by the system clock frequency (FOSC) and the BRG reload value. The reload value is stored in the SSPxADD register (Figure 58-1). When a write occurs to SSPxBUF, the Baud Rate Generator automatically counts down to 00h and stops until a reload from SSPxADD has taken place. The BRG count is decremented, twice per instruction cycle (TCY), on the Q2 and Q4 clocks. The relationship between the reload value, system clock and BRG clock is FCLOCK = FOSC/(2 * ([SSPxADD] + 1)).

Once the given operation is complete (i.e., transmission of the last data bit is followed by $\overline{ACK}$), the internal clock automatically stops counting and the SCLx pin remains in its last state.

Table 58-1 demonstrates clock rates based on instruction cycles and the BRG value loaded into SSPxADD.

58.3.1 Baud Rate and Module Interdependence

Because the MSSP modules are independent, each module can use BRG as its clock source and operate in SPI or I^2C Master mode at different baud rates. This is done by using different BRG reload values for each module.

Because this mode derives its basic clock source from the system clock, any changes to the clock will affect all modules in the same proportion. It may be possible to change one or both baud rates back to a previous value by changing the BRG reload value.

Figure 58-1: Baud Rate Generator Block Diagram

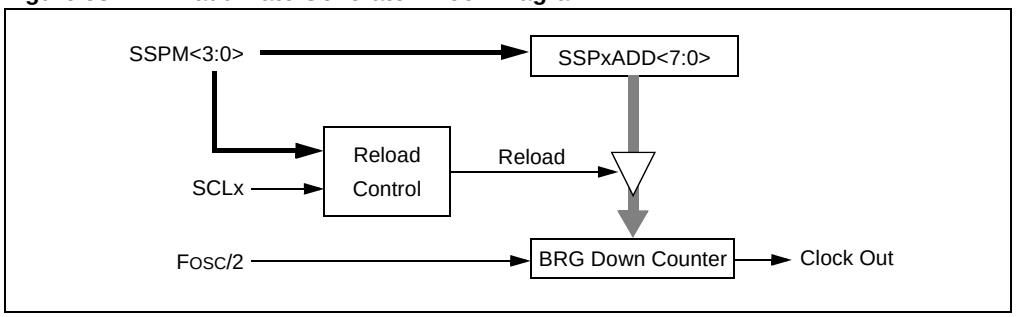


Table 58-1: Clock Rates with the BRG

FOSC	FCY	BRG Value (SSPxADD)	FCLOCK (2 Rollovers of BRG)
32 MHz	16 MHz	0Fh	1 MHz(1)
32 MHz	16 MHz	27h	400 kHz(1)
32 MHz	16 MHz	9Fh	100 kHz
16 MHz	8 MHz	07h	1 MHz(1)
16 MHz	8 MHz	13h	400 kHz(1)
16 MHz	8 MHz	4Fh	100 kHz
8 MHz	4 MHz	03h	1 MHz(1)
8 MHz	4 MHz	09h	400 kHz(1)
8 MHz	4 MHz	27h	100 kHz
2 MHz	1 MHz	02h(2)	333.3 kHz
2 MHz	1 MHz	09h	100 kHz

Note 1: The I^2C™ interface does not conform to the 400 kHz I^2C specification (which applies to rates greater than 100 kHz) in all details, but may be used with care where higher rates are required by the application.

2: BRG values of 00h through 02h are not supported in I^2C mode.

58.4 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:

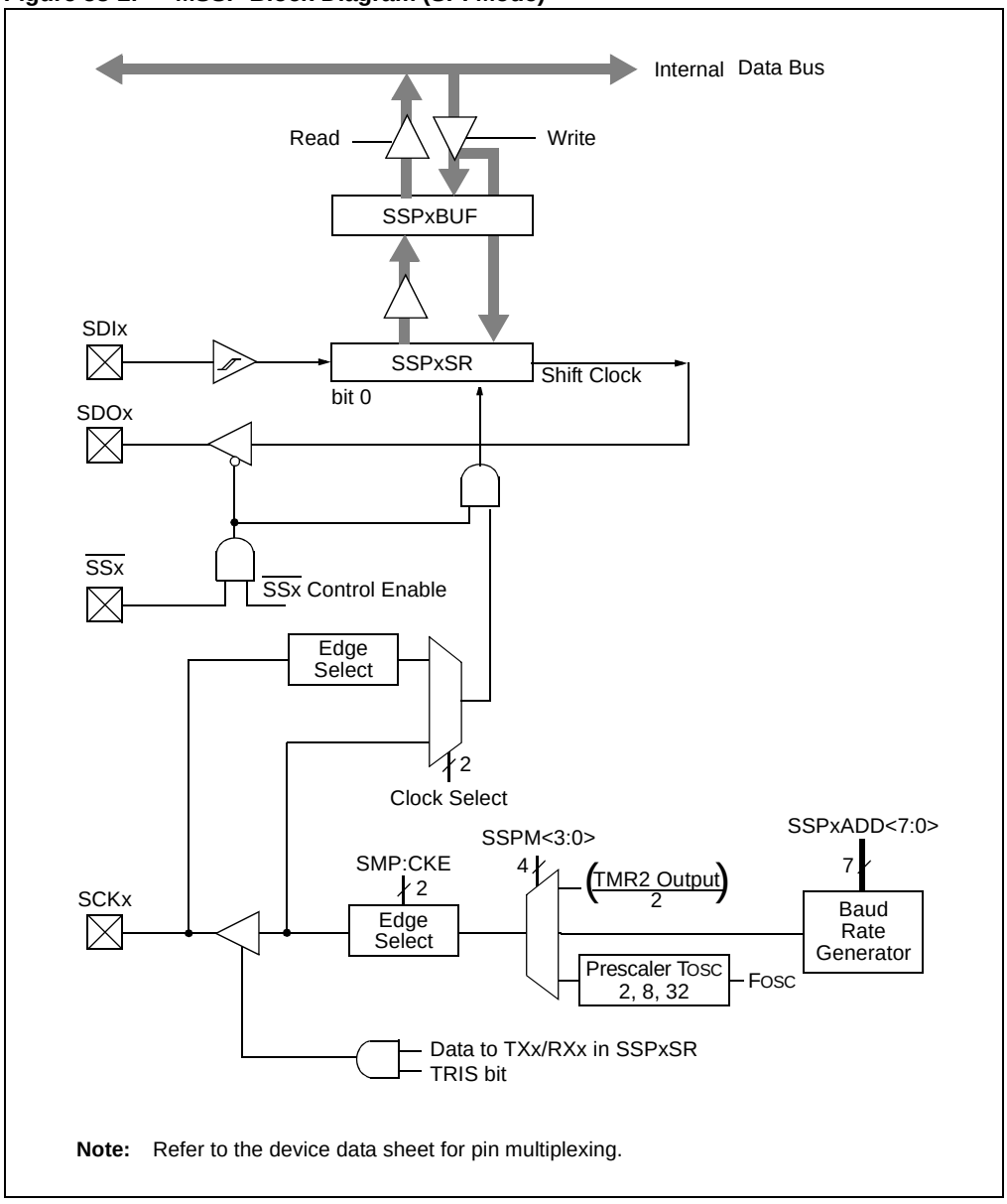
- Serial Data Out (SDOx)
- Serial Data In (SDIx)
- Serial Clock (SCKx)

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

- Slave Select ($\overline{SSx}$)

Figure 58-2 shows the block diagram of the MSSP module when operating in SPI mode.

Figure 58-2: MSSP Block Diagram (SPI Mode)



Note: Refer to the device data sheet for pin multiplexing.

58.4.1 Operation

When initializing the SPI, several options need to be specified. This is done by programming the appropriate control bits (SSPxCON1<5:0> and SSPxSTAT<7:6>). These control bits allow the following to be specified:

- Master mode (SCKx is the clock output)
- Slave mode (SCKx is the clock input)
- Clock Polarity (Idle state of SCKx)
- Data Input Sample Phase (middle or end of data output time)
- Clock Edge (output data on rising/falling edge of SCKx)
- Clock Rate (Master mode only)
- Slave Select mode (Slave mode only)

Each MSSP module consists of a Transmit/Receive Shift register (SSPxSR) and a Buffer register (SSPxBUF). The SSPxSR shifts the data in and out of the device, MSb first. The SSPxBUF holds the data that was written to the SSPxSR until the received data is ready. Once the 8 bits of data have been received, that byte is moved to the SSPxBUF register. Then, the Buffer Full detect bit, BF (SSPxSTAT<0>) and the interrupt flag bit, SSPxIF, are set. This double-buffering of the received data (SSPxBUF) allows the next byte to start reception before reading the data that was just received. Any write to the SSPxBUF register during transmission/reception of data will be ignored and the Write Collision Detect bit, WCOL (SSPxCON1<7>), will be set. User software must clear the WCOL bit so that it can be determined if the following write(s) to the SSPxBUF register completed successfully.

When the application software is expecting to receive valid data, the SSPxBUF should be read before the next byte of data to transfer is written to the SSPxBUF. The Buffer Full bit, BF (SSPxSTAT<0>), indicates when SSPxBUF has been loaded with the received data (transmission is complete). When the SSPxBUF is read, the BF bit is cleared. This data may be irrelevant if the SPI is only a transmitter. Generally, the MSSP interrupt is used to determine when the transmission/reception has completed. If the interrupt method is not going to be used, then software polling can be done to ensure that a write collision does not occur. Example 58-1 shows the loading of the SSPxBUF (SSPxSR) for data transmission.

The SSPxSR is not directly readable or writable and can only be accessed by addressing the SSPxBUF register. Additionally, the SSPxSTAT register indicates the various status conditions.

Example 58-1: Loading the SSP1BUF (SSP1SR) Register (in Assembly Language)

```
LOOP:
    BTSS    SSP1STAT, #BF      ;Has data been received (transmit complete)?
    BRA     LOOP               ;No
    MOV     SSP1BUF, W1        ;W1 register = contents of SSP1BUF
    MOV     W1, RXDATA         ;Save in user RAM, if data is meaningful
    MOV     TXDATA, W1         ;W1 register = contents of TXDATA
    MOV     W1, SSP1BUF        ;SSP1BUF = new data to transmit in W1
```

Example 58-2: Loading the SSP1BUF (SSP1SR) Register (in 'C')

```
while(SSP1STATbits.BF == 0);  // wait until data received
                              // (transmit complete)
rxdata = SSP1BUF;             // read SSP1BUF and save in user RAM
SSP1BUF = txdata;             // write byte to SSP1BUF,
                              // starts next SPI transfer
```


58.4.2 Enabling SPI I/O

To enable the serial port, the MSSP Enable bit, SSPEN (SSPxCON1<5>), must be set. This configures the SDIx, SDOx, SCKx and $\overline{SS}$x pins as SPI port pins.

To reset or reconfigure SPI mode:

1. Clear the SSPEN bit.
2. Re-initialize the SSPxCON registers.

In SPI Master mode, the MSSP module will assert control over any pins associated with the SDOx and SCKx outputs. This does not automatically disable other digital functions associated with the pin and may result in the module driving the digital I/O port inputs. To prevent this, the MSSP module outputs must be disconnected from its output pins while the module is in SPI Master mode.

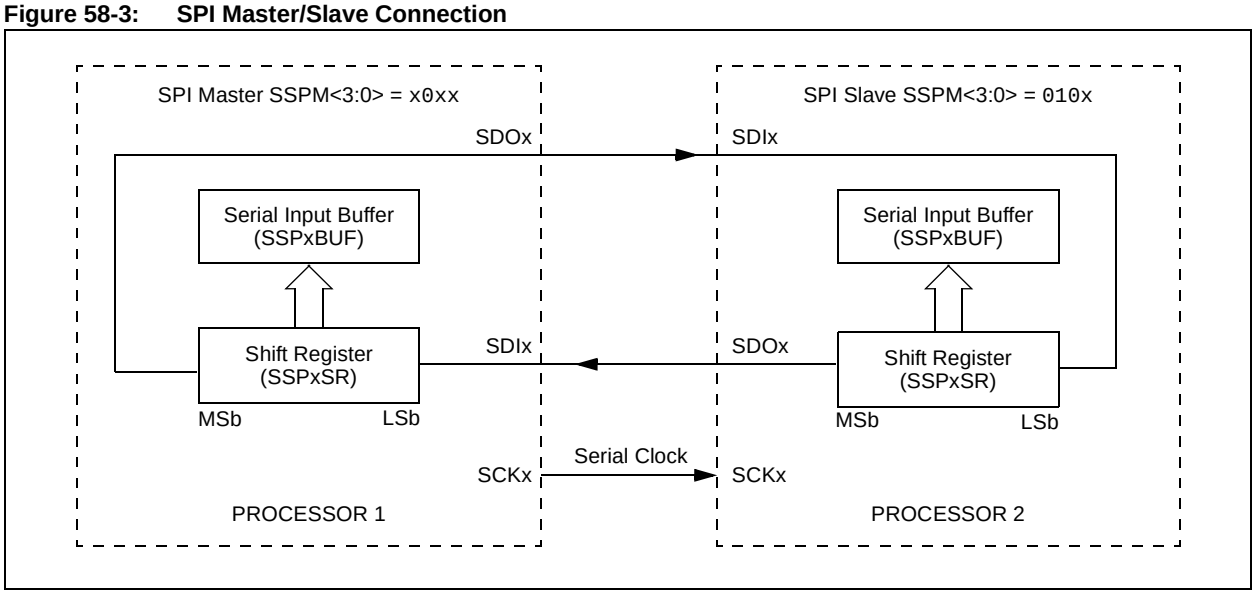
The SPI Master mode outputs may be disabled through one or more pad control bits in a Pad Configuration register, or by using the Peripheral Pin Select (PPS) feature (if available). Refer to the specific device data sheet for the available configuration options.

58.4.3 Typical Connections

Figure 58-3 shows a typical connection between two microcontrollers. The master controller (Processor 1) initiates the data transfer by sending the SCKx signal. Data is shifted out of both shift registers on their programmed clock edge and latched on the opposite edge of the clock. Both processors should be programmed to the same Clock Polarity (CKP), then both controllers would send and receive data at the same time. Whether the data is meaningful (or dummy data) depends on the application software. This leads to three scenarios for data transmission:

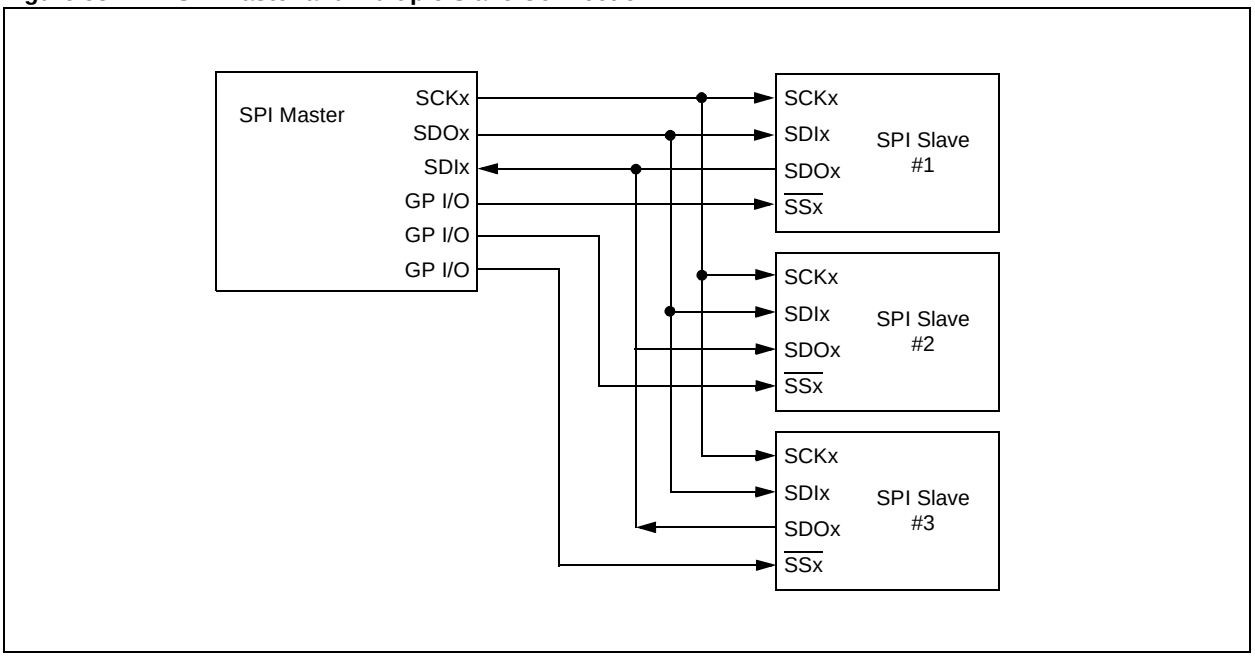
- Master sends data – Slave sends dummy data
- Master sends data – Slave sends data
- Master sends dummy data – Slave sends data

Figure 58-3: SPI Master/Slave Connection



One MSSP module may also function as the master device with multiple slave modules on the same bus. In these cases, multiple Slave Select ($\overline{SS}x$) signals are used to indicate which module is being addressed, with the master processor using general purpose I/O pins to drive the $\overline{SS}x$ pins for individual slave devices (Figure 58-4). Data and clock signals function in the same way as they do with master and single slave connections.

Figure 58-4: SPI Master and Multiple Slave Connection

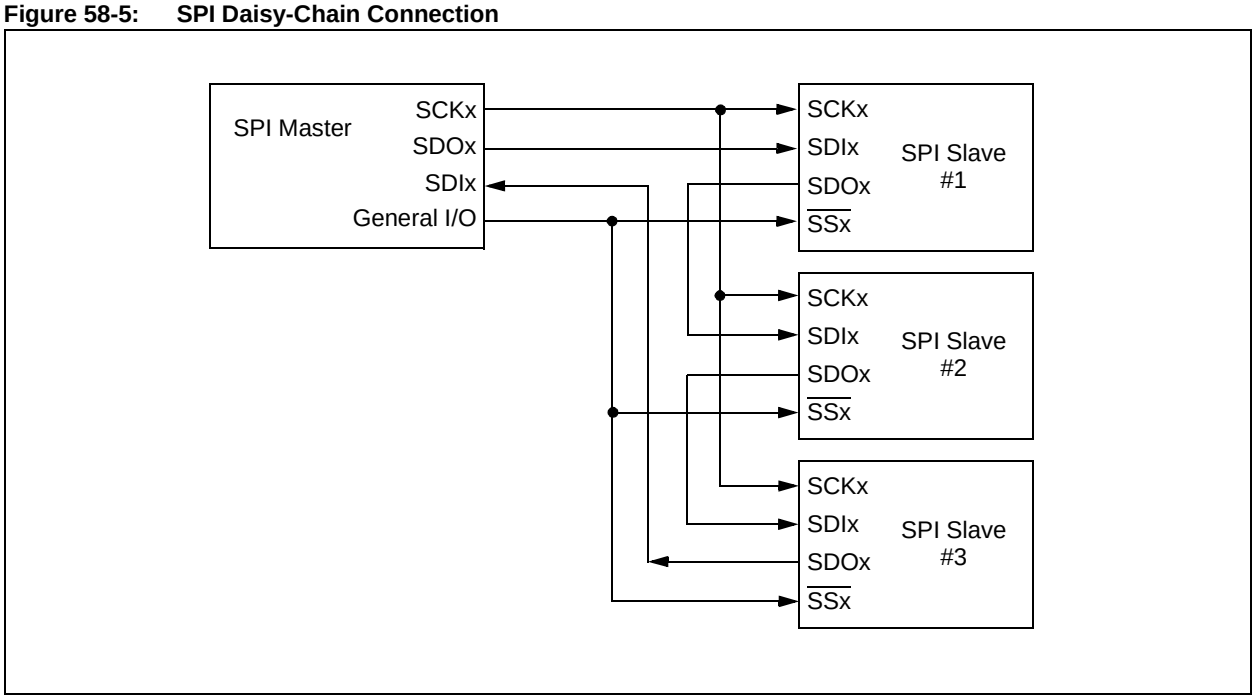


58.4.3.1 DAISY-CHAIN CONFIGURATION

When more than one SPI slave is present, the SPI bus can also be connected in a daisy-chain configuration. The first slave output is connected to the second slave input, the second slave output is connected to the third slave input, and so on. The final slave output is connected to the master input. Each slave sends out, during a second group of clock pulses, an exact copy of what was received during the first group of clock pulses. The whole chain acts as one large communication shift register. The daisy-chain feature only requires a single Slave Select line from the master device. Figure 58-5 shows the block diagram of a typical daisy-chain connection when operating in SPI mode.

In a daisy-chain configuration, only the most recent byte on the bus is required by the slave. The BOEN bit (SSPxCON3<4>) allows the user to ignore all but the last received byte. SSPOV is still set when a new byte is received and BF = 1, but hardware continues to write the most recent byte to SSPxBUF. Setting the BOEN bit enables writes to the SSPxBUF register, even if the previous byte has not been read. This allows the software to ignore data that may not apply to it.

Figure 58-5: SPI Daisy-Chain Connection



58.4.4 Master Mode

The master can initiate the data transfer at any time because it controls the SCKx. The master determines when the slave (Processor 2 in Figure 58-3) is to broadcast data by the software protocol.

In Master mode, the data is transmitted/received as soon as the SSPxBUF register is written to. If the SPI is only going to receive, the SDOx output could be disabled (programmed as an input). The SSPxSR register will continue to shift in the signal present on the SDIx pin at the programmed clock rate. As each byte is received, it will be loaded into the SSPxBUF register as a normal received byte (interrupts and status bits are appropriately set). This could be useful in receiver applications as a “Line Activity Monitor” mode.

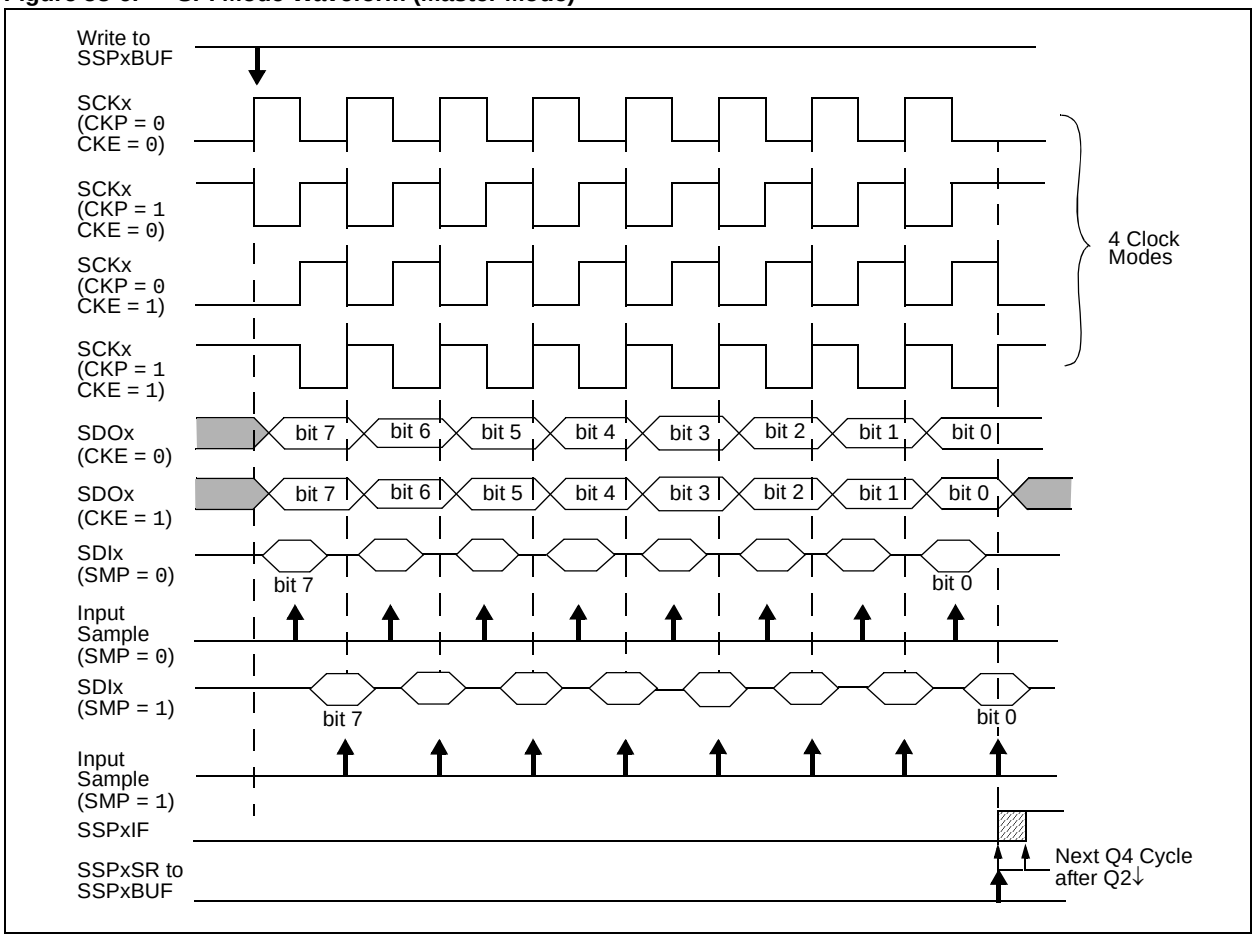
The clock polarity is selected by appropriately programming the CKP bit (SSPxCON1<4>). This then, would give waveforms for SPI communication, as shown in Figure 58-6, Figure 58-8 and Figure 58-9, where the MSB is transmitted first. In Master mode, the SPI clock rate (bit rate) is user-programmable to be one of the following:

- FOSC/(2 • [SSPxADD + 1])
- FOSC/2 (or TCY)
- FOSC/8 (or 4 • TCY)
- FOSC/32 (or 16 • TCY)
- Timer2 output/2

This allows a maximum data rate (at 32 MHz) of 16 Mbps.

Figure 58-6 shows the waveforms for Master mode. When the CKE bit is set, the SDOx data is valid before there is a clock edge on SCKx. The change of the input sample is shown based on the state of the SMP bit. The time when the SSPxBUF is loaded with the received data is shown.

Figure 58-6: SPI Mode Waveform (Master Mode)



58.4.5 Slave Mode

In Slave mode, the data is transmitted and received as the external clock pulses appear on SCKx. When the last bit is latched, the SSPxIF interrupt flag bit is set.

While in Slave mode, the external clock is supplied by the external clock source on the SCKx pin. This external clock must meet the minimum high and low times, as specified in the electrical characteristics in the specific device data sheet.

While in Sleep mode, the slave can transmit/receive data. When a byte is received, the device can be configured to wake-up from Sleep.

58.4.6 Slave Select Synchronization

The $\overline{SSx}$ pin allows a Synchronous Slave mode. The SPI must be in Slave mode with the $\overline{SSx}$ pin control enabled (SSPxCON1<3:0> = 04h). When the $\overline{SSx}$ pin is low, transmission and reception are enabled and the SDOx pin is driven. When the $\overline{SSx}$ pin goes high, the SDOx pin is no longer driven, even if in the middle of a transmitted byte and becomes a floating output. External pull-up/pull-down resistors may be desirable depending on the application.

Note 1:	When the SPI is in Slave mode with $\overline{SSx}$ pin control enabled (SSPxCON1<3:0> = 0100), the SPI module will reset if the $\overline{SSx}$ pin is set to VDD.
2:	If the SPI is used in Slave mode with CKE set, then the $\overline{SSx}$ pin control must be enabled.

When the SPI module resets, the bit counter is forced to '0'. This can be done by either forcing the $\overline{SSx}$ pin to a high level or clearing the SSPEN bit.

To emulate two-wire communication, the SDOx pin can be connected to the SDIx pin. When the SPI needs to operate as a receiver, the SDOx pin can be disabled and configured as an input. This disables transmissions from the SDOx pin. The SDIx pin can always be left as an input (SDIx function) since it cannot create a bus conflict.

Figure 58-7: Slave Synchronization Waveform

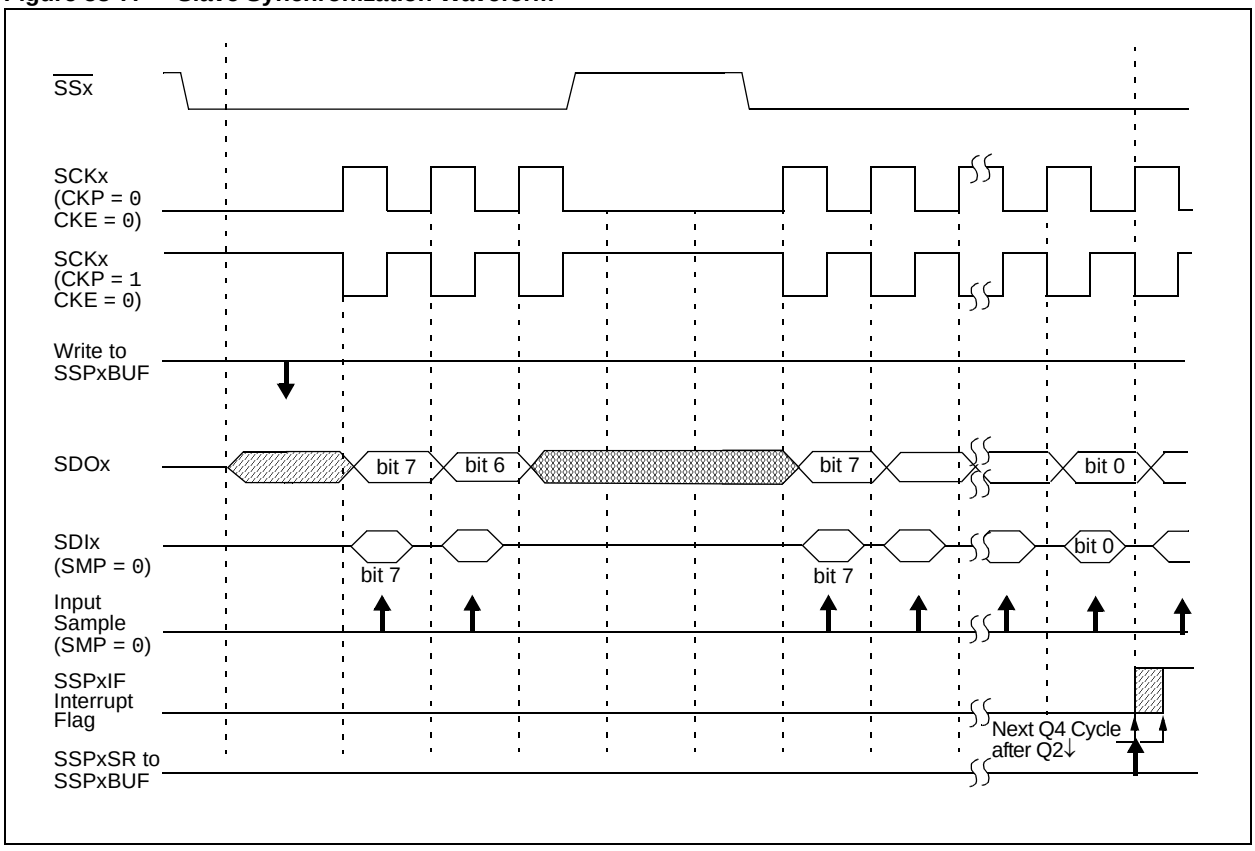


Figure 58-8: SPI Mode Waveform (Slave Mode with CKE = 0)

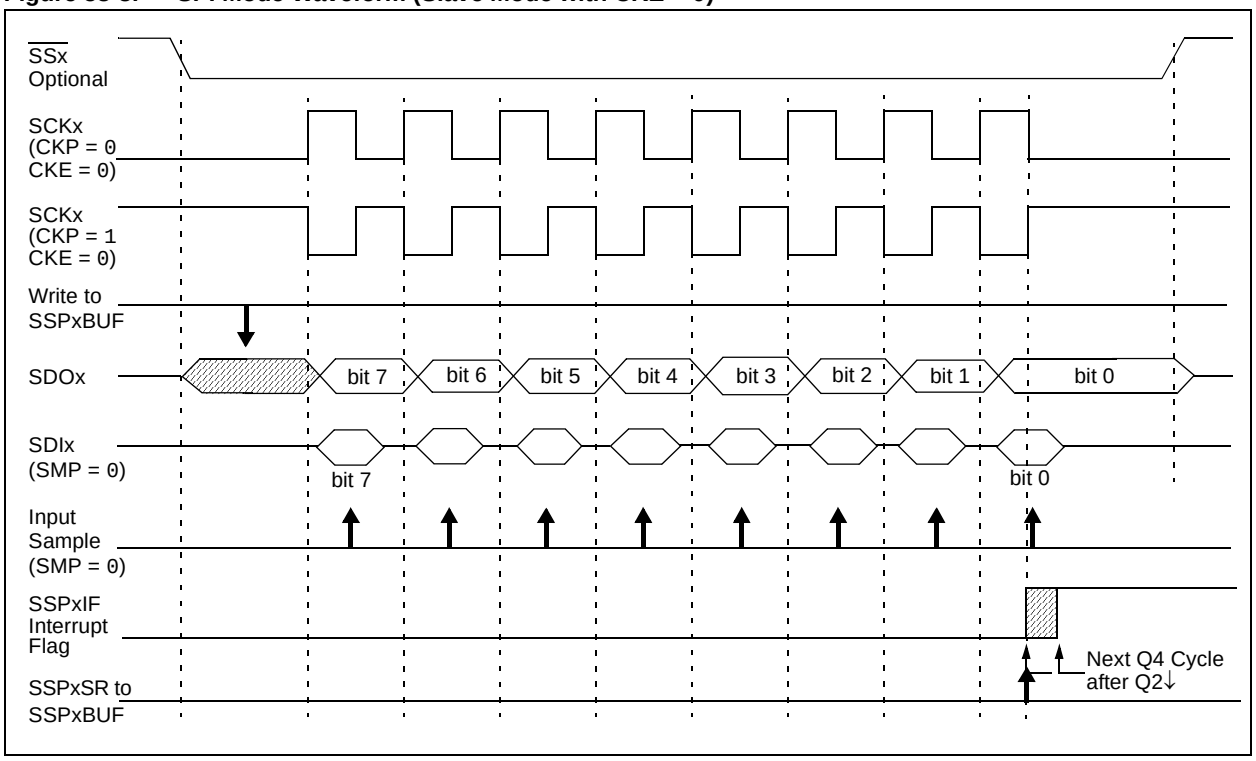
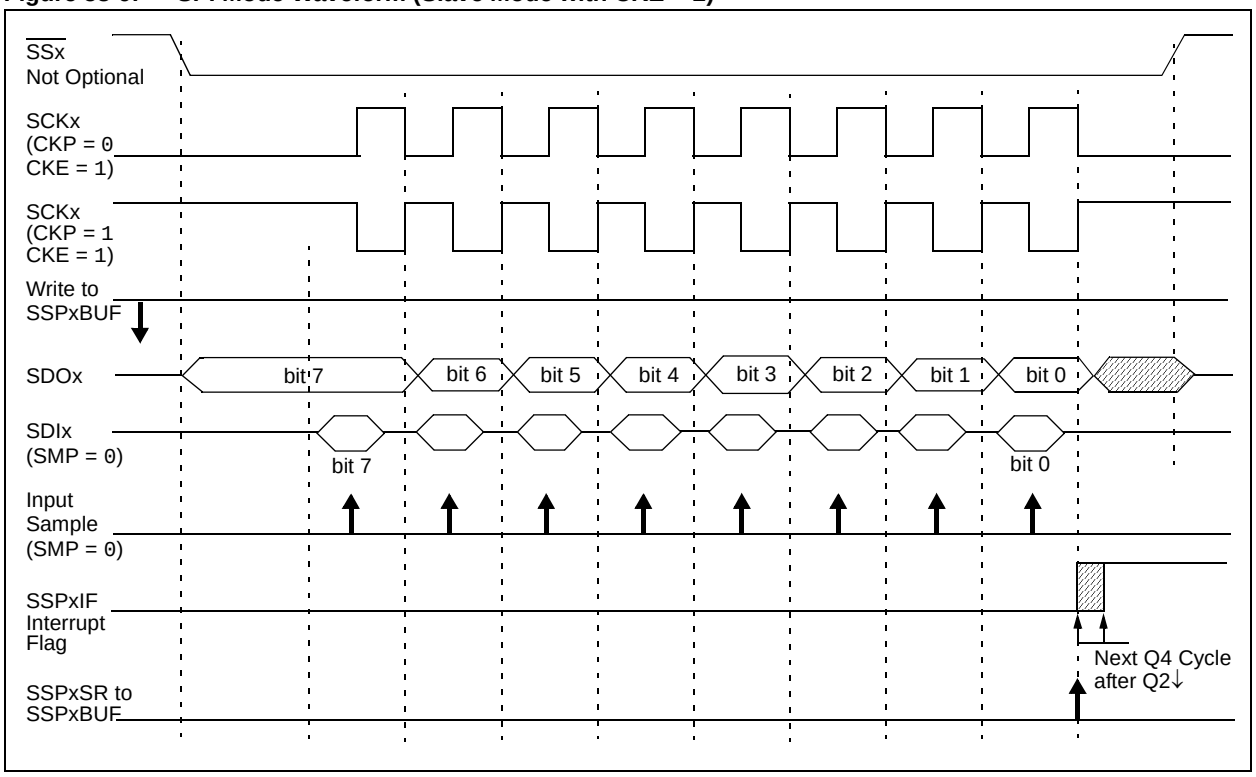


Figure 58-9: SPI Mode Waveform (Slave Mode with CKE = 1)



58.4.7 Bus Mode Compatibility

Table 58-2 shows the compatibility between the standard SPI modes, and the states of the CKP and CKE control bits. In addition, the SMP bit (SSPxCON1<7>) controls if the data is sampled in the middle or at the end of the output time.

Table 58-2: SPI Bus Modes

Standard SPI Mode Terminology	Control Bits State	
	CKP	CKE
0, 0	0	1
0, 1	0	0
1, 0	1	1
1, 1	1	0

58.4.8 SPI Clock Speed and Module Interactions

Because the MSSP modules are independent, they can operate simultaneously in SPI Master mode at different data rates. Setting the SSPM<3:0> bits of the SSPxCON1 register determines the rate for the corresponding module.

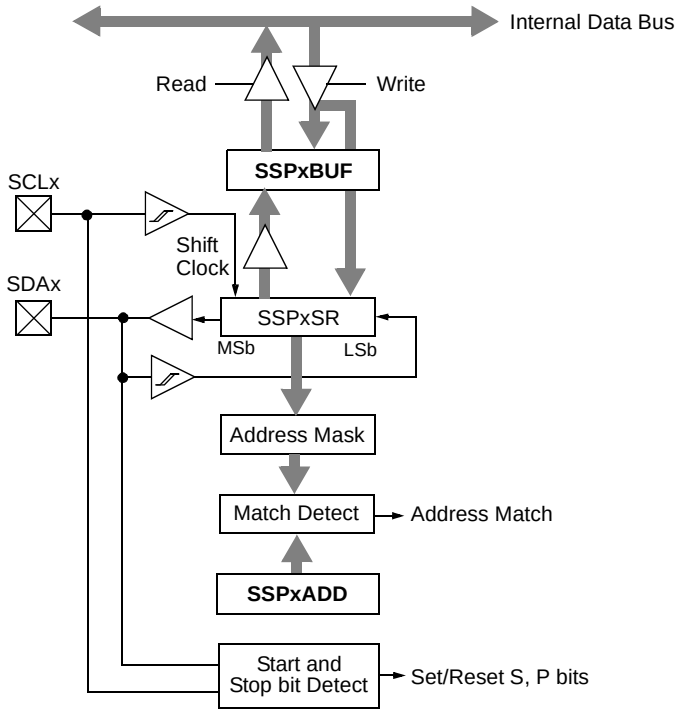
An exception is when all of the modules use Timer2 as a time base in Master mode. In this instance, any changes to the Timer2 module's operation will affect all MSSP modules equally. If different bit rates are required for each module, the user should select one of the other time base options for one of the modules.

58.5 I²C MODE

The MSSP module in I²C mode fully implements all master and slave functions (including General Call Address 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.

Two pins are used for data transfer: Serial Clock (SCLx) and Serial Data (SDAx). The user must configure these pins as inputs by setting the associated TRIS bits.

Figure 58-10: MSSP Block Diagram (I²C™ Mode)



Note: Only port I/O names are used in this diagram for the sake of brevity. Refer to the text for a full list of multiplexed functions.

58.5.1 Operation

The MSSP module functions are enabled by setting the MSSP Enable bit, SSPEN (SSPxCON1<5>).

The SSPxCON1 register allows control of the I²C operation. Four mode selection bits (SSPxCON1<3:0>) allow one of the following I²C modes to be selected:

- I²C Master mode
- I²C Slave mode (7-bit addressing)
- I²C Slave mode (10-bit addressing)
- I²C Slave mode (7-bit addressing) with Start and Stop bit interrupts enabled
- I²C Slave mode (10-bit addressing) with Start and Stop bit interrupts enabled
- I²C Firmware Controlled Master mode, slave is Idle

A selection of any I²C mode, with the SSPEN bit set, forces the SCLx and SDAx pins to be open-drain, provided these pins are programmed as inputs by setting the appropriate TRISx bits. To ensure proper operation of the module, pull-up resistors must be provided externally to the SCLx and SDAx pins.

58.5.1.1 SDAx HOLD TIME

Hold time is the time that data on SDAx is held valid after the falling edge of SCLx. By default, the module is specified for a hold time of a minimum of 100 ns. A longer hold time on SDAx (300 ns minimum) can be selected by setting the SDAHT bit (SSPxCON3<3>). Longer minimum hold times may be necessary on buses with large capacitance.

58.5.1.2 START/STOP CONDITION INTERRUPTS

The SCIE and PCIE bits (SSPxCON3<6,5>) are used to enable the generation of Start and Stop interrupts in Slave modes that do not typically support this function. These bits have no effect on Slave or Master modes when an interrupt-on-Start and Stop detect are already enabled.

58.5.2 Slave Mode

In Slave mode, the SCLx and SDAx pins must be configured as inputs by setting their corresponding TRIS bits. The MSSP module will override the input state with the output data when required (slave-transmitter).

The I^2C Slave mode hardware will always generate an interrupt on an address match. Address masking will allow the hardware to generate an interrupt for more than one address (up to 31 in 7-bit addressing and up to 63 in 10-bit addressing). Through the mode select bits, the user can also choose to interrupt-on-Start and Stop bits.

When an address is matched, or the data transfer after an address match is received, the hardware will automatically generate the Acknowledge ($\overline{ACK}$) pulse and load the SSPxBUF register with the received value currently in the SSPxSR register.

Any combination of the following conditions will cause the MSSP module not to give this $\overline{ACK}$ pulse:

- The Buffer Full bit, BF (SSPxSTAT<0>), was set before the transfer was received.
- The overflow bit, SSPOV (SSPxCON1<6>), was set before the transfer was received.

In this case, the SSPxSR register value is not loaded into the SSPxBUF, but SSPxIF is set. The BF bit is cleared by reading the SSPxBUF register, while bit, SSPOV, is cleared through software.

The SCLx clock input must have a minimum high and low for proper operation. Refer to the specific device data sheet for I^2C timing specifications and requirements.

58.5.2.1 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 SSPxSR register. All incoming bits are sampled with the rising edge of the clock (SCLx) line. The value of the register bits, SSPxSR<7:1>, is compared to the value of the SSPxADD register. The address is compared on the falling edge of the eighth clock (SCLx) pulse. If the addresses match and the BF and SSPOV bits are clear, the following events occur:

1. The SSPxSR register value is loaded into the SSPxBUF register.
2. The Buffer Full bit (BF) is set.
3. An $\overline{ACK}$ pulse is generated.
4. The SSPxIF interrupt flag bit is set (and an interrupt is generated, if enabled) on the falling edge of the ninth SCLx pulse.

In 10-Bit Addressing 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. The R/$\overline{W}$ bit (SSPxSTAT<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, '11110 A9 A8 0', where 'A9' and 'A8' are the two MSbs of the address.

The sequence of events for 10-bit addressing is as follows, with Steps 7 through 9 for the slave-transmitter:

1. Receive first (high) byte of address (bits, SSPxIF, BF and UA, are set on an address match).
2. Update the SSPxADD register with the second (low) byte of address (clears bit, UA, and releases the SCLx line).
3. Read the SSPxBUF register (clears bit, BF) and clear the SSPxIF interrupt flag.
4. Receive second (low) byte of address (bits, SSPxIF, BF and UA, are set).
5. Update the SSPxADD register with the first (high) byte of the address. If the match releases the SCLx line, this will clear bit, UA.
6. Read the SSPxBUF register (clears bit, BF) and clear SSPxIF.
7. Receive Repeated Start condition.
8. Receive first (high) byte of the address (bits, SSPxIF and BF, are set).
9. Read the SSPxBUF register (clears bit, BF) and clear SSPxIF.

58.5.2.2 ADDRESS MASKING

Masking an address bit causes that bit to become a “don't care”. When one address bit is masked, two addresses will be Acknowledged and cause an interrupt. It is possible to mask more than one address bit at a time, which greatly expands the number of addresses Acknowledged.

The I^2C Slave behaves the same way, whether address masking is used or not. However, when address masking is used, the I^2C slave can Acknowledge multiple addresses and cause interrupts. When this occurs, it is necessary to determine which address caused the interrupt by checking SSPxBUF.

Address masking supports both 7-bit and 10-bit addresses, using a mask of 7 or 8 bits to define a range of addresses that can be Acknowledged, using the lowest bits of the incoming address. This allows the module to Acknowledge up to 127 different addresses with 7-bit addressing or 255 with 10-bit addressing (see Example 58-1).

The SSPxMSK register (Register 58-9) is used to store the address mask value. Clearing a bit in SSPxMSK causes the corresponding address bit to be masked; setting the bit requires a match in that position. SSPxMSK resets to all ‘1’s upon any Reset condition and therefore, has no effect on the standard MSSP operation until written with a mask value.

With 7-bit addressing, SSPxMSK<7:1> mask the corresponding address bits in the SSPxADD register. For any SSPxMSK bits that are active (SSPxMSK<n> = 0), the corresponding SSPxADD address bit is ignored (SSPxADD<n> = x). For the module to issue an address Acknowledge, it is sufficient to match only on addresses that do not have an active address mask.

With 10-bit addressing, bits, SSPxMSK<7:0>, mask the corresponding address bits in the SSPxADD register. For any SSPxMSK bits that are active (= 0), the corresponding SSPxADD address bit is ignored (SSPxADD<n> = x).

Note:	The two Most Significant bits of the address are not affected by address masking.

Example 58-3: Address Masking Examples

7-Bit Addressing:

SSPxADD<7:1> = 1010 0000

SSPxMSK<7:1> = 1111 001

Addresses Acknowledged = A8h, A6h, A4h, A0h

10-Bit Addressing:

SSPxADD<7:0> = 1010 0000 (The two MSbs are ignored in this example since they are not affected.)

SSPxMSK<5:1> = 1111 0011

Addresses Acknowledged = A8h, A6h, A4h, A0h

58.5.2.3 RECEPTION

When the R/$\overline{W}$ bit of a matching received address byte is clear, the R/$\overline{W}$ bit of the SSPxSTAT register is cleared. The received address is loaded into the SSPxBUF register and Acknowledged.

When the overflow condition exists for a received address, then an Acknowledge ($\overline{ACK}$) is given. An overflow condition is defined as either the BF bit of the SSPxSTAT register is set or the SSPOV bit of the SSPxCON1 register is set. The BOEN bit (SSPxCON3<4>) selects if the SSPOV bit controls whether or not the $\overline{ACK}$ is generated; setting the bit allows the module to ignore SSPOV.

An MSSP interrupt is generated for each transferred data byte. The SSPxIF flag bit must be cleared by software.

When the SEN bit (SSPxCON2<0>) is set, SCLx is held low (clock stretch) following each received byte. The clock must be released by setting the CKP bit (SSPxCON1<4>), except for some incidents in 10-bit mode. See **Section 58.5.2.6 "Clock Stretching"** for more detail.

58.5.2.3.1 7-Bit Addressing Reception

This section describes a standard sequence of events for the MSSPx module, configured as an I^2C slave in 7-Bit Addressing mode. All decisions are made by hardware or software and their effect on reception. Figure 58-11 and Figure 58-12 are used as visual references for this description.

This is a step-by-step process of what typically must be done to accomplish I^2C communication.

1. A Start bit is detected.
2. The S bit (SSPxSTAT<3>) is set. If interrupt-on-Start detect is enabled, SSPxIF is also set.
3. The matching address with the R/$\overline{W}$ bit cleared is received.
4. The slave pulls SDAx low, sending an $\overline{ACK}$ to the master and sets the SSPxIF bit.
5. Software clears the SSPxIF bit.
6. Software reads the received address from SSPxBUF, clearing the BF flag.
7. If SEN = 1, slave software sets the CKP bit to release the SCLx line.
8. The master clocks out a data byte.
9. Slave drives SDAx low, sending an $\overline{ACK}$ to the master, and sets the SSPxIF bit.
10. Software clears SSPxIF.
11. Software reads the received byte from SSPxBUF, clearing BF.
12. Steps 8-12 are repeated for all received bytes from the master.
13. Master sends Stop condition, setting the P bit (SSPxSTAT<4>) and the bus goes Idle.

Figure 58-11: I²C™ Slave, 7-Bit Addressing, Reception (SEN = 0, AHEN = 0, DHEN = 0)

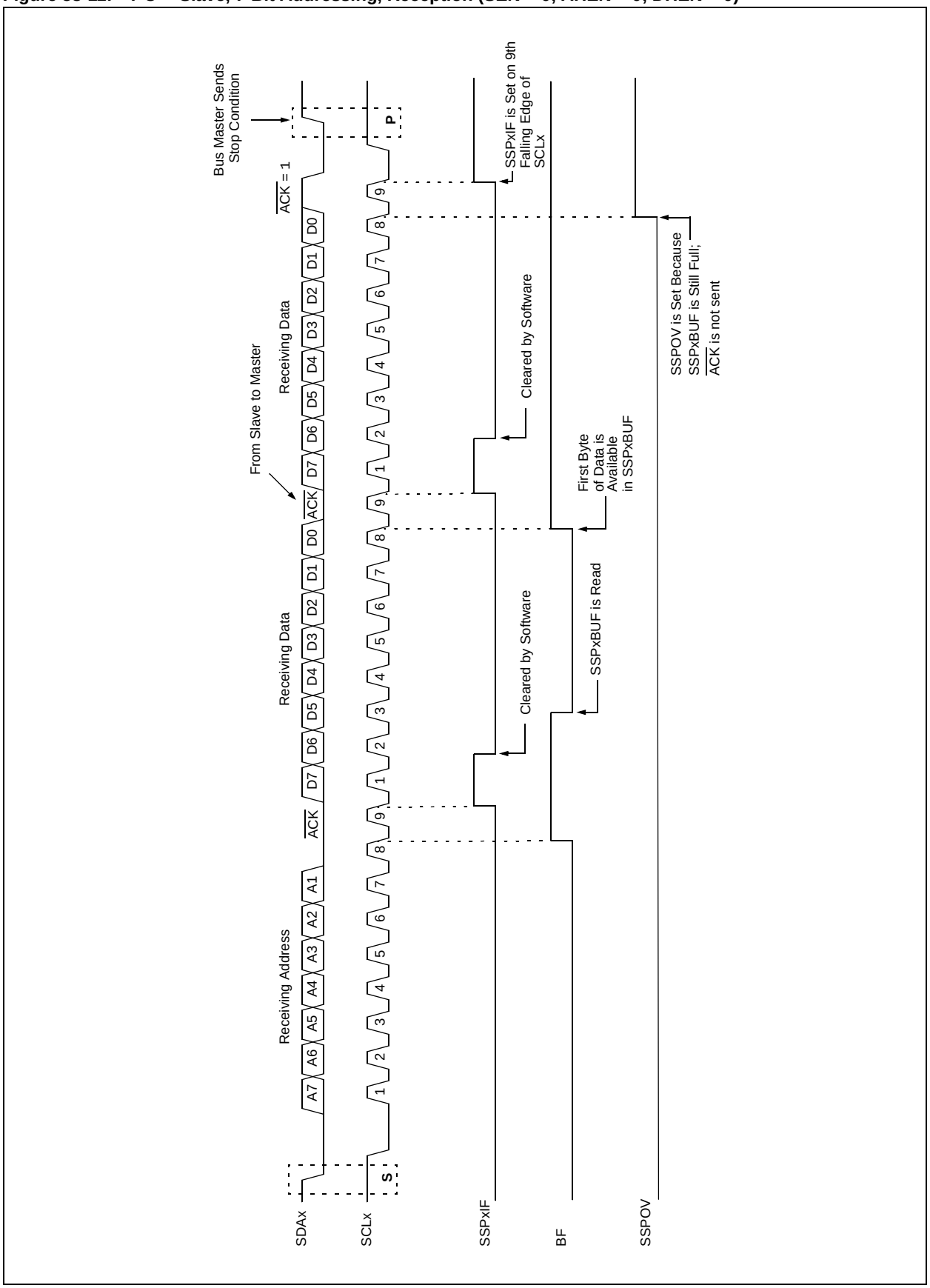
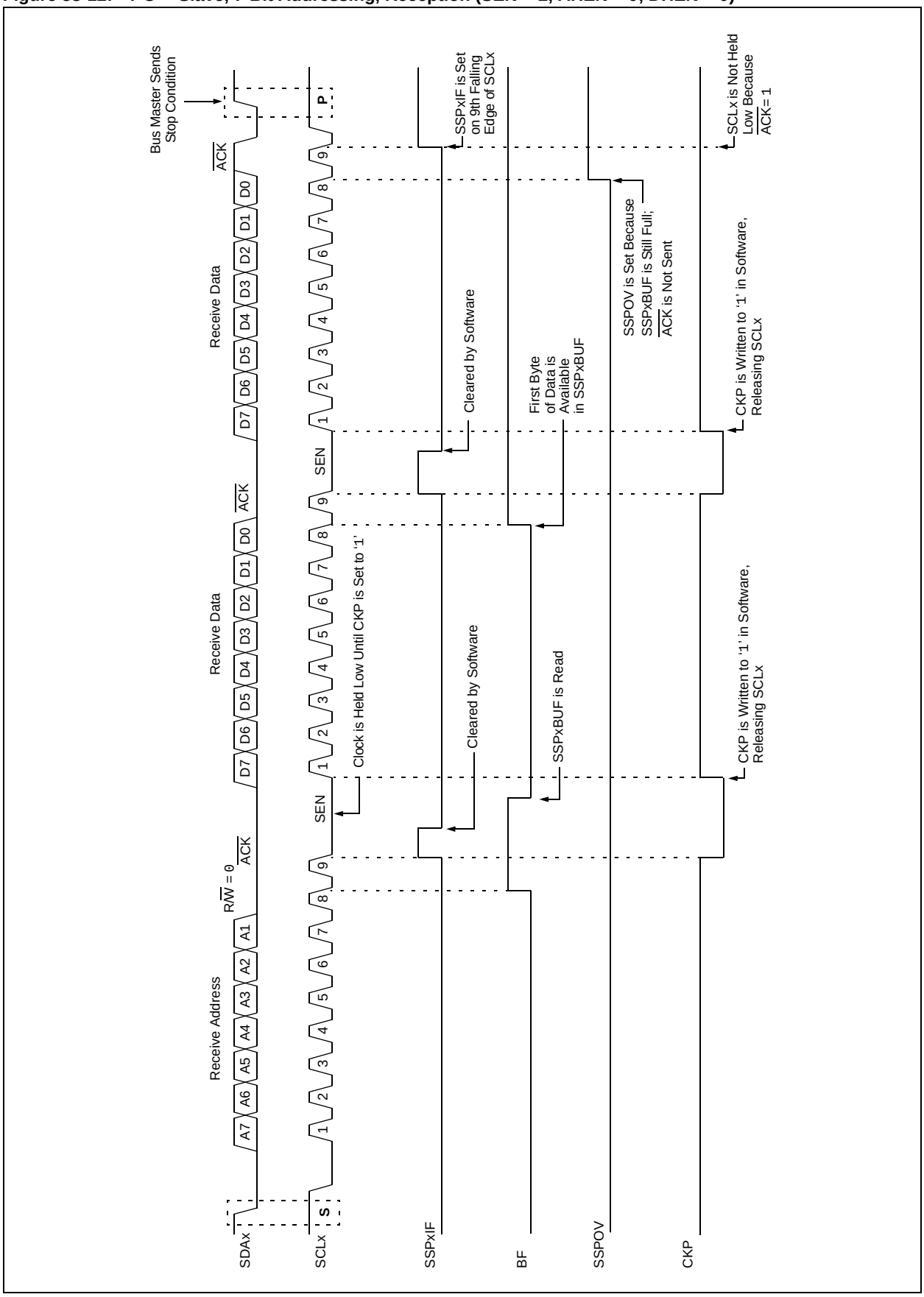


Figure 58-12: I²C™ Slave, 7-Bit Addressing, Reception (SEN = 1, AHEN = 0, DHEN = 0)



58.5.2.3.2 7-Bit Reception with Address and Data Hold Enabled

When the AHEN and DHEN bits are set, the slave device reception operates the same as without these options, except extra interrupts and clock stretching are added after the 8th falling edge of SCLx. These additional interrupts allow the application on the slave device to decide whether it wants to $\overline{ACK}$ the receive address or data byte, rather than the hardware. This functionality provides support for PMBus™ applications.

This list describes the steps that need to be taken by slave software to use these options for I^2C communication. Figure 58-13 displays a module using both address and data holding. Figure 58-14 includes the operation with the SEN bit also set.

1. The S bit (SSPxSTAT<3>) is set. If an interrupt-on-Start detect is enabled, SSPxIF is also set.
2. Matching address with the R/$\overline{W}$ bit clear is clocked in. SSPxIF is set and the CKP bit is cleared after the 8th falling edge of SCLx.
3. Slave clears the SSPxIF.
4. Slave can look at the ACKTIM bit (SSPxCON3<7>) to determine if the SSPxIF was after or before the $\overline{ACK}$.
5. Slave reads the address value from SSPxBUF, clearing the BF flag.
6. Slave sets $\overline{ACK}$ value clocked out to the master by setting ACKDT.
7. Slave releases the clock by setting CKP.
8. SSPxIF is set after an $\overline{ACK}$, not after a NACK.
9. If SEN = 1, the slave hardware will stretch the clock after the $\overline{ACK}$.
10. Slave clears SSPxIF.

Note:	SSPxIF is still set after the 9th falling edge of SCLx, even if there is no clock stretching and BF has been cleared. Only if a NACK is sent to the master is the SSPxIF bit not set.

11. SSPxIF is set and CKP is cleared after the 8th falling edge of SCLx for a received data byte.
12. Slave looks at the ACKTIM bit to determine the source of the interrupt.
13. Slave reads the received data from SSPxBUF, clearing BF.
14. Steps 7-14 are the same for each received data byte.
15. Communication is ended by either the slave sending an $\overline{ACK}$ = 1, or the master sending a Stop condition. If a Stop is sent and the interrupt-on-Stop detect is disabled, the slave will only know by polling the P bit.

Figure 58-13: I²C™ Slave, 7-Bit Addressing, Reception (SEN = 0, AHEN = 1, DHEN = 1)

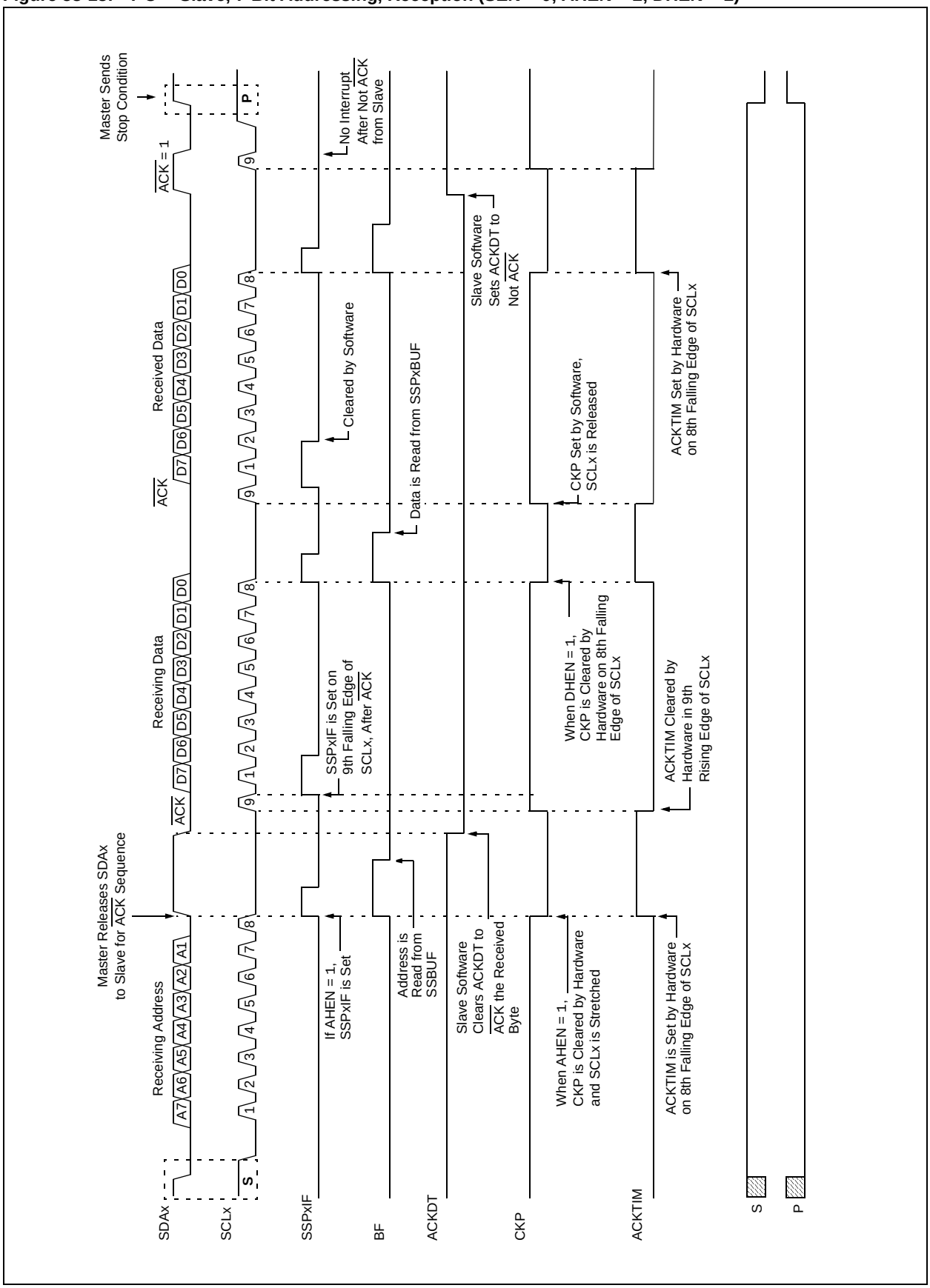
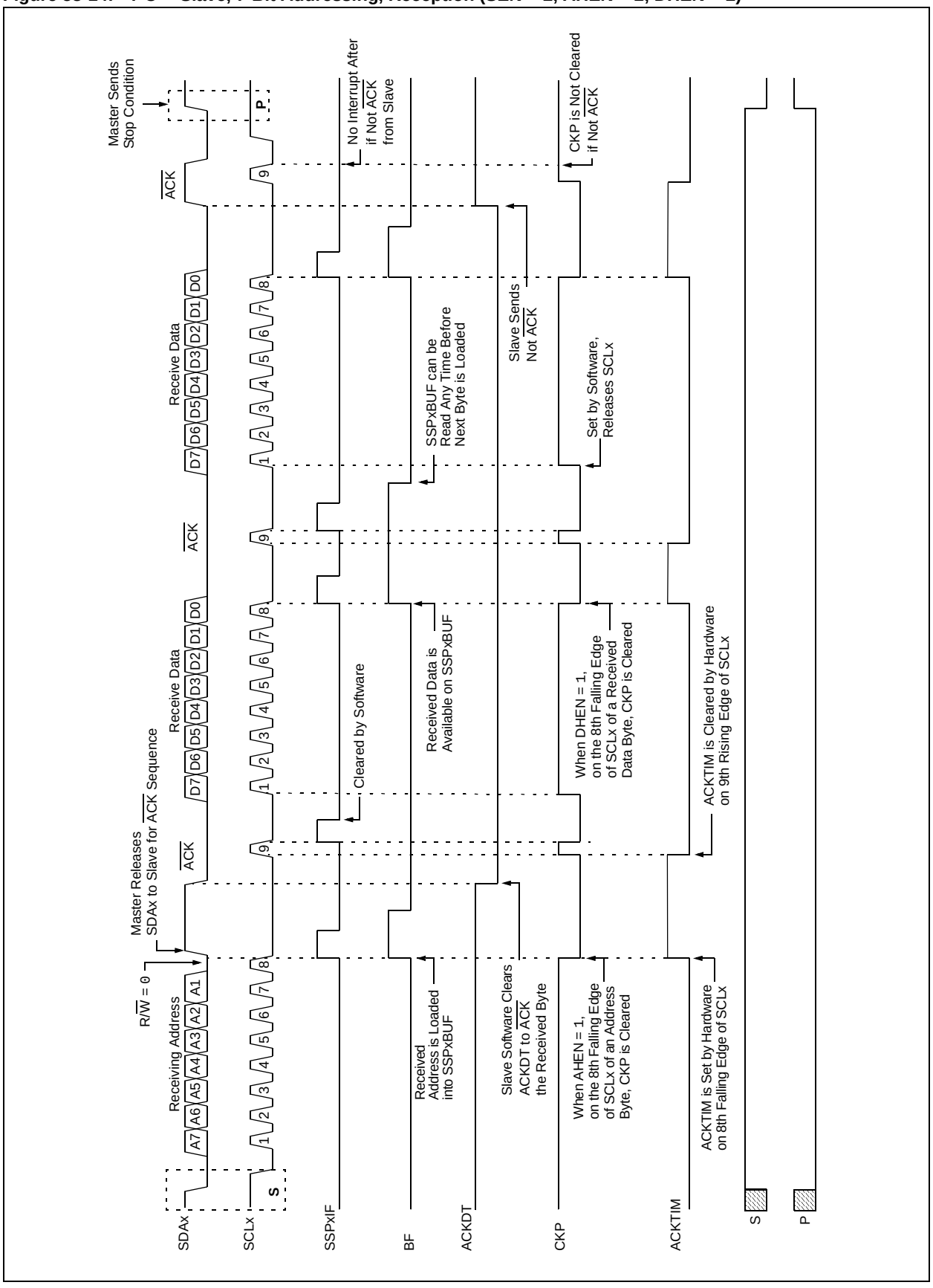


Figure 58-14: I²C™ Slave, 7-Bit Addressing, Reception (SEN = 1, AHEN = 1, DHEN = 1)



58.5.2.3.3 Slave Mode 10-Bit Addressing Reception

Figure 58-15 shows a standard sequence of events for the MSSP module, configured as an I^2C slave in 10-Bit Addressing mode. The step-by-step process of what must be done by slave software to accomplish I^2C communication is as follows:

1. Bus starts Idle.
2. Master sends Start condition. The S bit (SSPxSTAT<3>) is set. If interrupt-on-Start detect is enabled, SSPxIF is also set.
3. Master sends matching high address with the R/$\overline{W}$ bit clear; the UA bit (SSPxSTAT<>) is set.
4. Slave sends $\overline{ACK}$ and SSPxIF is set.
5. Software clears the SSPxIF bit.
6. Software reads received address from SSPxBUF, clearing the BF flag.
7. Slave loads low address into SSPxADD, releasing SCLx.
8. Master sends matching low address byte to the slave; the UA bit is set.

Note:	Updates to the SSPxADD register are not allowed until after the $\overline{ACK}$ sequence.

9. Slave sends $\overline{ACK}$ and SSPxIF is set.

Note:	If the low address does not match, SSPxIF and UA are still set so that the slave software can set SSPxADD back to the high address. BF is not set because there is no match; CKP is unaffected.

10. Slave clears SSPxIF.
11. Slave reads the received matching address from SSPxBUF, clearing BF.
12. Slave loads high address into SSPxADD.
13. Master clocks a data byte to the slave and clocks out the slave's $\overline{ACK}$ on the 9th SCLx pulse; SSPxIF is set.
14. If the SEN bit of SSPxCON2 is set, CKP is cleared by hardware and the clock is stretched.
15. Slave clears SSPxIF.
16. Slave reads the received byte from SSPxBUF, clearing BF.
17. If SEN is set, the slave sets CKP to release the SCLx.
18. Steps 13-17 repeat for each received byte.
19. Master sends Stop to end the transmission.

58.5.2.3.4 10-Bit Addressing Reception with Address or Data Hold

Reception using 10-bit addressing, with AHEN or DHEN set, is the same as with 7-bit modes. The only difference is the need to update the SSPxADD register using the UA bit. All functionality, specifically when the CKP bit is cleared and the SCLx line is held low, are the same. Figure 58-16 shows the sequence for slave reception in 10-Bit Addressing mode with AHEN set.

Figure 58-15: I²C™ Slave, 10-Bit Addressing, Reception (SEN = 1, AHEN = 0, DHEN = 0)

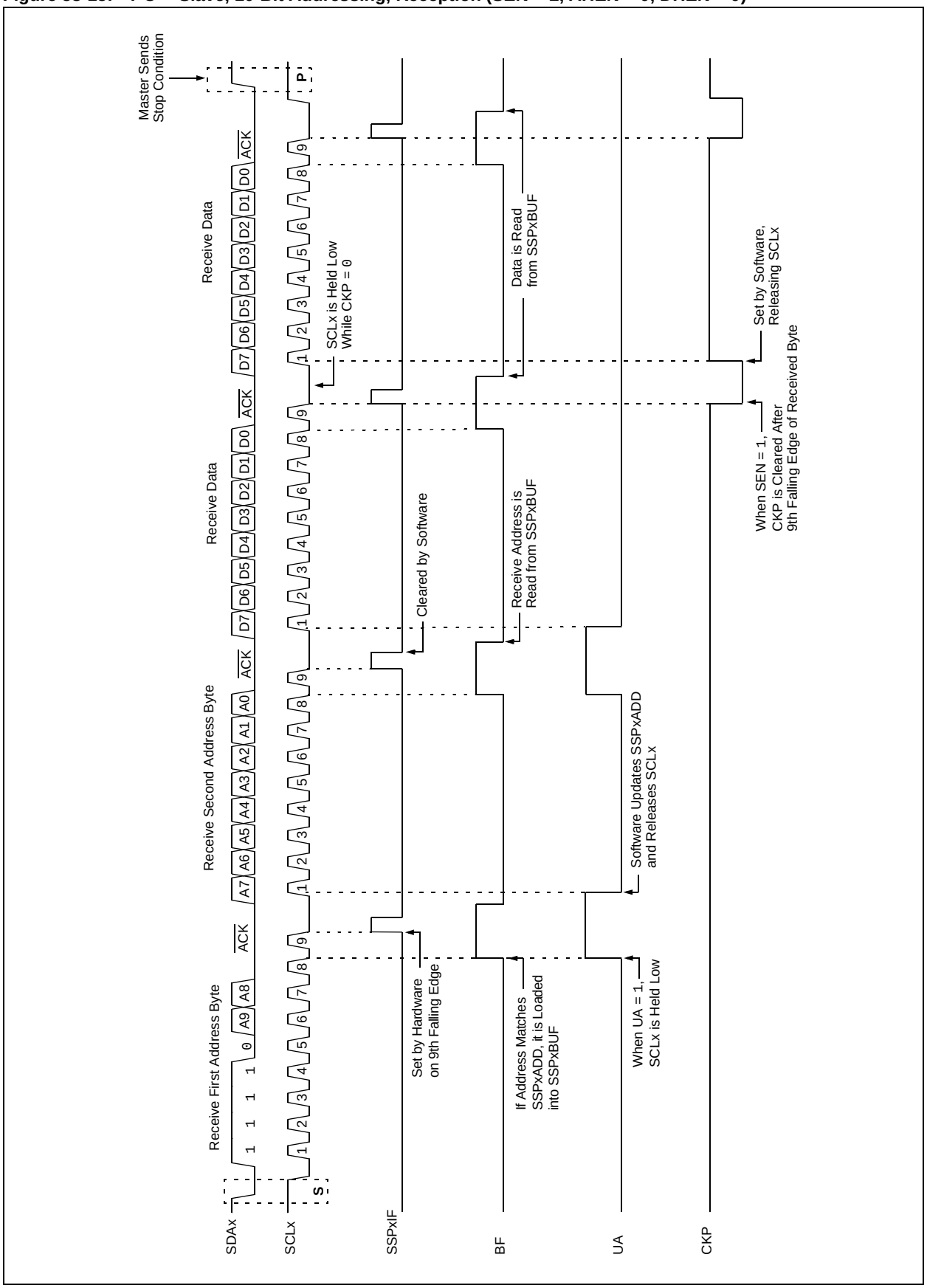
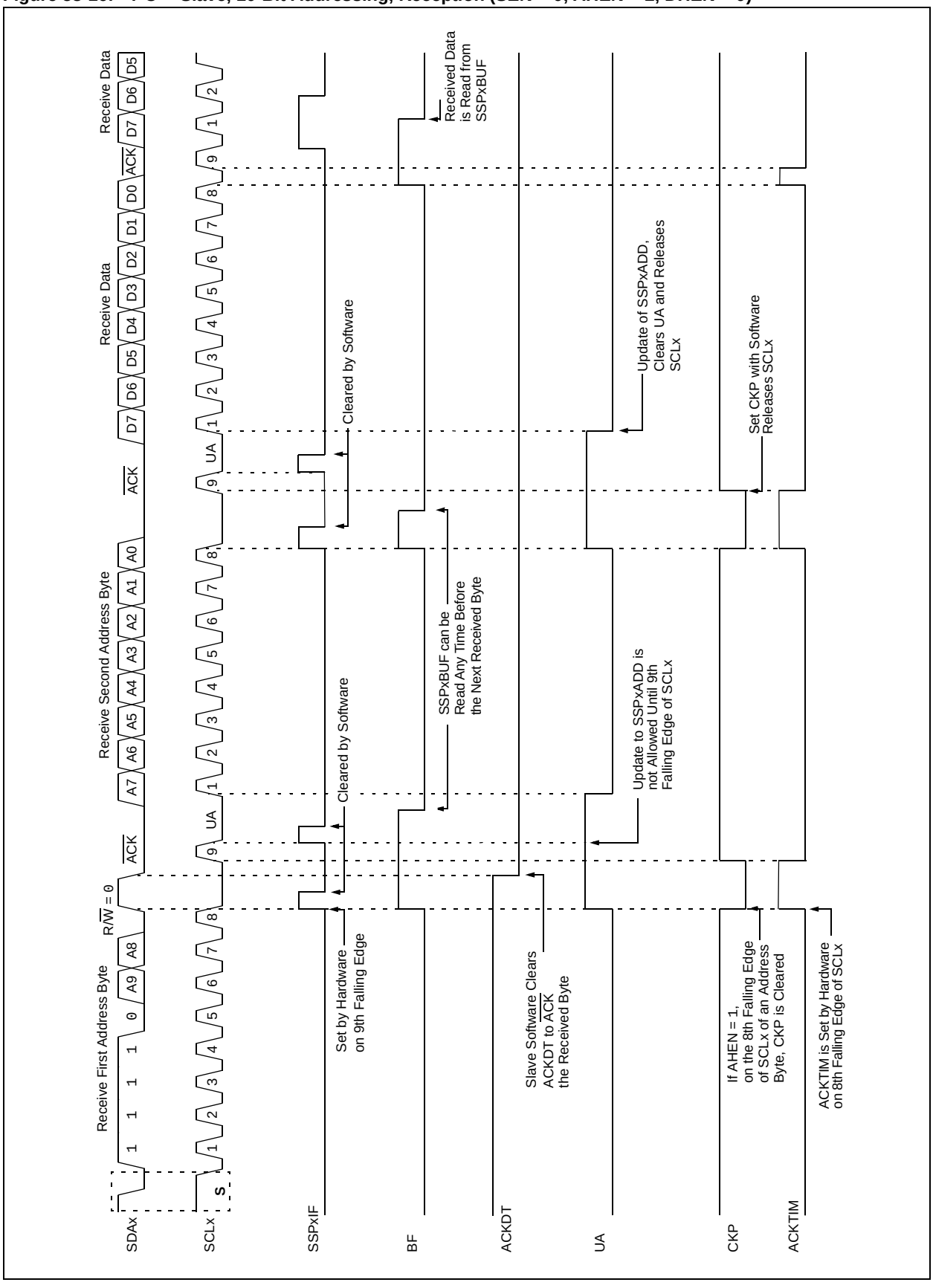


Figure 58-16: I²C™ Slave, 10-Bit Addressing, Reception (SEN = 0, AHEN = 1, DHEN = 0)



58.5.2.4 TRANSMISSION

When the R/$\overline{W}$ bit of the incoming address byte is set and an address match occurs, the R/$\overline{W}$ bit of the SSPxSTAT register is set. The received address is loaded into the SSPxBUF register. The $\overline{ACK}$ pulse will be sent on the ninth bit and the SCLx pin is held low, regardless of SEN (see **Section 58.5.2.6 "Clock Stretching"** for more detail). By stretching the clock, the master will be unable to assert another clock pulse until the slave is done preparing the transmit data. The transmit data must be loaded into the SSPxBUF register, which also loads the SSPxSR register. Then, the SCLx pin should be enabled by setting bit, CKP (SSPxCON1<4>). The eight data bits are shifted out on the falling edge of the SCLx input. This ensures that the SDAx signal is valid during the SCLx high time (Figure 58-17).

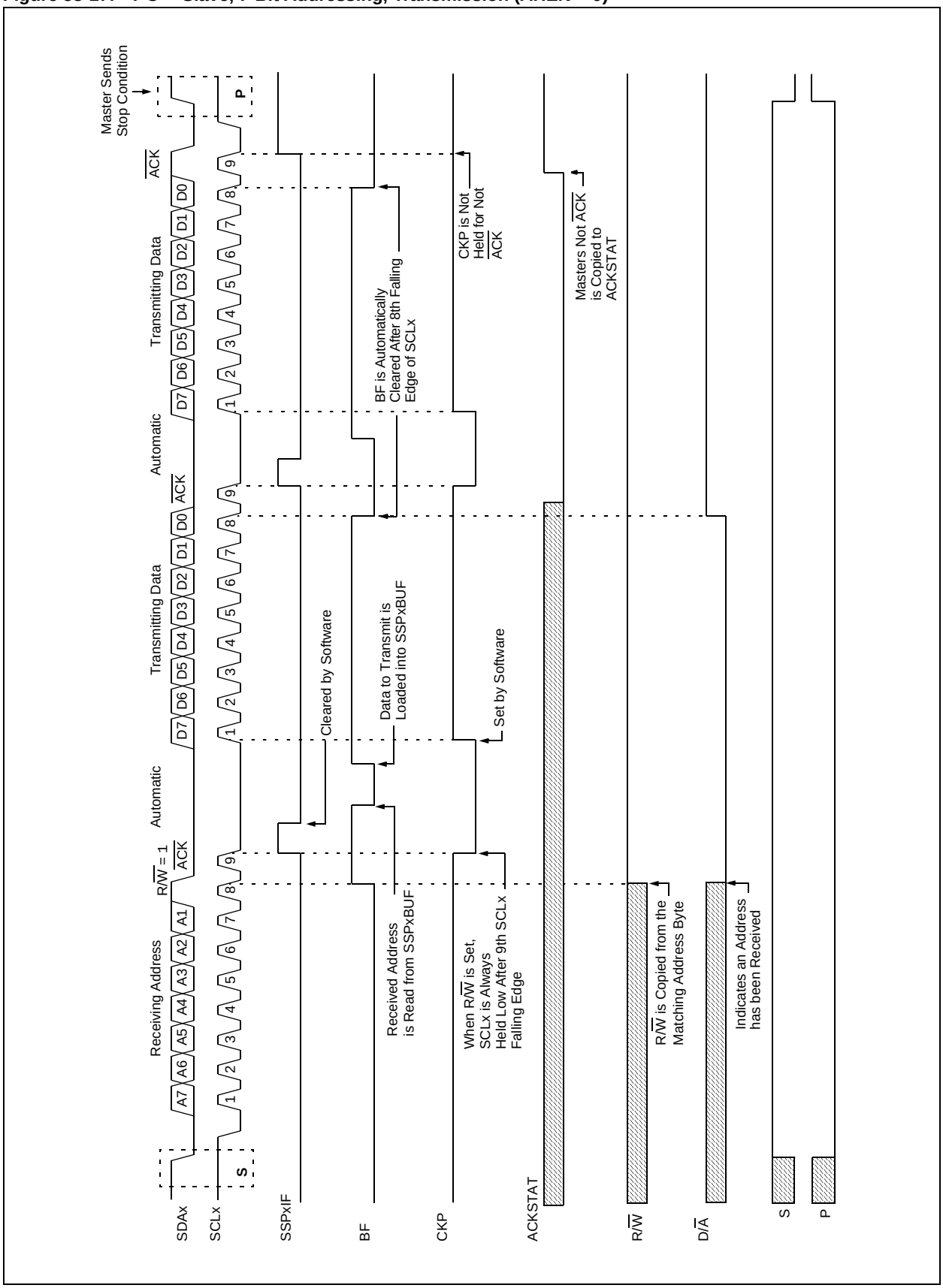
The $\overline{ACK}$ pulse from the master-receiver is latched on the rising edge of the ninth SCLx input pulse. If the SDAx line is high (not $\overline{ACK}$), then the data transfer is complete. In this case, when the $\overline{ACK}$ is latched by the slave, the slave logic is reset (resets SSPxSTAT register) and the slave monitors for another occurrence of the Start bit. If the SDAx line was low ($\overline{ACK}$), the next transmit data must be loaded into the SSPxBUF register. Again, the SCLx pin must be enabled by setting bit, CKP.

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

58.5.2.5 SLAVE MODE BUS COLLISION

A slave receives a read request and begins shifting data out on the SDAx line. If a bus collision is detected and the SBCDE bit (SSPxCON3<2>) is set, the BCLxIF (Bus Collision Interrupt Flag) is set. Once a bus collision is detected, the slave goes Idle and waits to be addressed again. User software can use the BCLxIF bit to handle a slave bus collision.

Figure 58-17: I²C™ Slave, 7-Bit Addressing, Transmission (AHEN = 0)



58.5.2.5.1 7-Bit Transmission with Address Hold Enabled

Setting the AHEN bit (SSPxCON3<1>) enables additional clock stretching and interrupt generation after the 8th falling edge of a received matching address. Once a matching address has been clocked in, CKP is cleared and the SSPxIF interrupt flag is set. Figure 58-18 shows a standard waveform of a 7-bit addressing slave transmission with AHEN enabled.

1. Bus starts Idle.
2. Master sends the Start condition; the S bit of SSPxSTAT is set. If interrupt-on-Start detect is enabled, the SSPxIF interrupt is also set.
3. Master sends a matching address with the R/$\overline{W}$ bit set. After the 8th falling edge of the SCLx line, the CKP bit is cleared and an SSPxIF interrupt is generated.
4. Slave software clears SSPxIF.
5. Slave software reads the ACKTIM bit (SSPxCON3<7>), and the R/$\overline{W}$ and D/$\overline{A}$ bits (SSPxSTAT<2,5>) to determine the source of the interrupt.
6. Slave reads the address value from the SSPxBUF register, clearing the BF bit.
7. Slave software decides from this information if it wishes to $\overline{ACK}$ or not $\overline{ACK}$, and sets the ACKDT bit (SSPxCON2<6>) accordingly.
8. Slave sets the CKP bit, releasing SCLx.
9. Master clocks in the $\overline{ACK}$ value from the slave.
10. Slave hardware automatically clears the CKP bit and sets SSPxIF after the $\overline{ACK}$, if the R/$\overline{W}$ bit is set.
11. Slave software clears SSPxIF.
12. Slave loads the value to transmit to the master into SSPxBUF, setting the BF bit.

Note: SSPxBUF cannot be loaded until after the $\overline{ACK}$.

13. Slave sets the CKP bit, releasing the clock.
14. Master clocks out the data from the slave and sends an $\overline{ACK}$ value on the 9th SCLx pulse.
15. Slave hardware copies the $\overline{ACK}$ value into the ACKSTAT bit.
16. Steps 10-15 are repeated for each byte transmitted to the master from the slave.
17. If the master sends a not $\overline{ACK}$, the slave releases the bus, allowing the master to send a Stop and end the communication.

Note: The master must send a not $\overline{ACK}$ on the last byte to ensure that the slave releases the SCLx line to receive a Stop.

58.5.2.5.2 10-Bit Addressing Transmission

The transmission using 10-bit addressing, with or without address and data holds, parallels the 7-bit procedures. Figure 58-19 shows the sequence for slave transmission with 10-bit addressing.

Figure 58-18: I²C™ Slave, 7-Bit Addressing, Transmission (AHEN = 1)

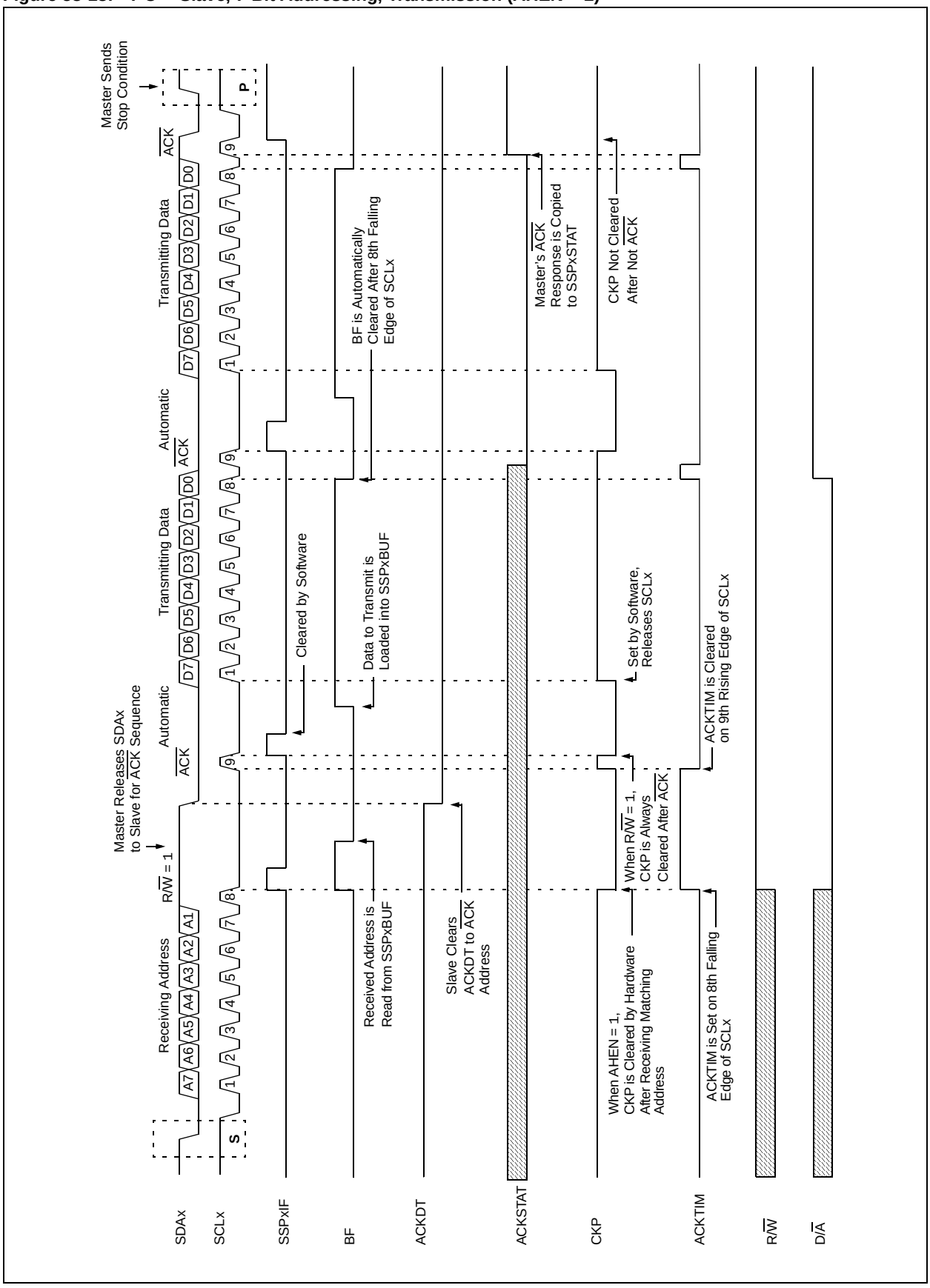
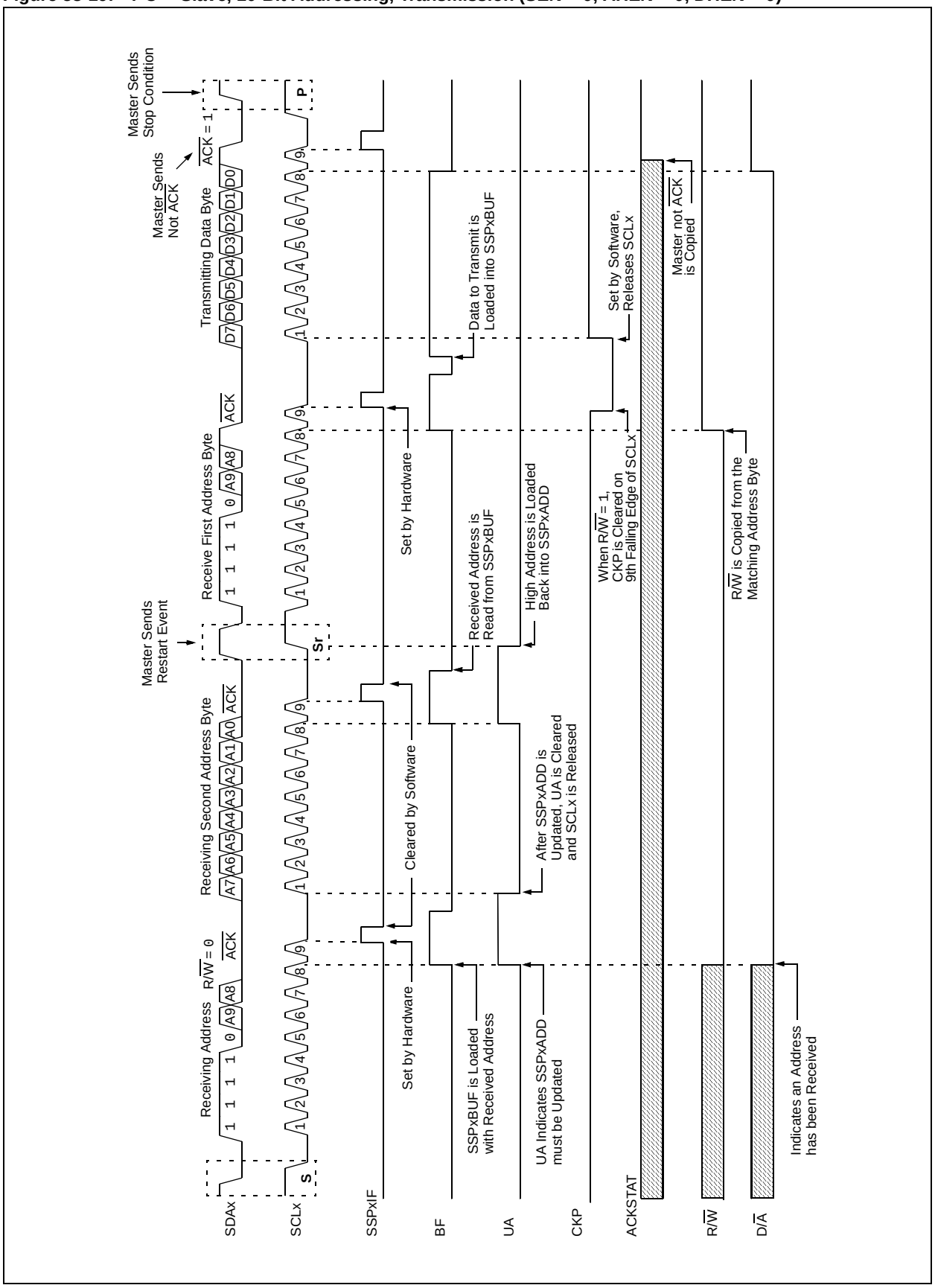


Figure 58-19: I²C™ Slave, 10-Bit Addressing, Transmission (SEN = 0, AHEN = 0, DHEN = 0)



58.5.2.6 CLOCK STRETCHING

Both 7-bit and 10-bit Slave modes implement automatic clock stretching during a transmit sequence.

In Slave modes, the SEN bit (SSPxCON2<0>) allows clock stretching to be enabled during receives. Setting SEN will cause the SCLx pin to be held low at the end of each data receive sequence.

58.5.2.6.1 Clock Stretching for 7-Bit Slave Receive Mode (SEN = 1)

In 7-Bit Slave Receive mode, on the falling edge of the ninth clock at the end of the $\overline{\text{ACK}}$ sequence, if the BF bit is set, the CKP bit in the SSPxCON1 register is automatically cleared, forcing the SCLx output to be held low. The CKP being cleared to '0' will assert the SCLx line low. The CKP bit must be set in the user's ISR (Interrupt Service Routine) before reception is allowed to continue. By holding the SCLx line low, the user has time to service the ISR and read the contents of the SSPxBUF before the master device can initiate another receive sequence. This will prevent buffer overruns from occurring (see Figure 58-12).

Note 1:	If the user reads the contents of the SSPxBUF before the falling edge of the ninth clock, thus clearing the BF bit, the CKP bit will not be cleared and clock stretching will not occur.
2:	The CKP bit can be set in software, regardless of the state of the BF bit. The user should be careful to clear the BF bit in the ISR before the next receive sequence in order to prevent an overflow condition.

58.5.2.6.2 Clock Stretching for 10-Bit Slave Receive Mode (SEN = 1)

In 10-Bit Slave Receive mode, during the address sequence, clock stretching automatically takes place but CKP is not cleared. During this time, if the UA bit is set after the ninth clock, clock stretching is initiated. The UA bit is set after receiving the upper byte of the 10-bit addressing and following the receive of the second byte of the 10-bit addressing with the R/$\overline{\text{W}}$ bit cleared to '0'. The release of the clock line occurs upon updating SSPxADD. Clock stretching will occur on each data receive sequence, as described in 7-bit mode (see Figure 58-15).

Note:	If the user polls the UA bit and clears it by updating the SSPxADD register, before the falling edge of the ninth clock occurs, and if the user hasn't cleared the BF bit by reading the SSPxBUF register before that time, then the CKP bit will still NOT be asserted low. Clock stretching, on the basis of the state of the BF bit, only occurs during a data sequence, not an addressing sequence.

58.5.2.6.3 Clock Stretching for 7-Bit Slave Transmit Mode

The 7-Bit Slave Transmit mode implements clock stretching by clearing the CKP bit, after the falling edge of the ninth clock, if the BF bit is clear. This occurs regardless of the state of the SEN bit.

The user's ISR must set the CKP bit before transmission is allowed to continue. By holding the SCLx line low, the user has time to service the ISR and load the contents of the SSPxBUF before the master device can initiate another transmit sequence (see Figure 58-11).

Note 1:	If the user loads the contents of SSPxBUF, setting the BF bit before the falling edge of the ninth clock, the CKP bit will not be cleared and clock stretching will not occur.
2:	The CKP bit can be set in software regardless of the state of the BF bit.

58.5.2.6.4 Clock Stretching for 10-Bit Slave Transmit Mode

In 10-Bit Slave Transmit mode, clock stretching is controlled during the first two address sequences by the state of the UA bit, just as it is in 10-Bit Slave Receive mode. The first two addresses are followed by a third address sequence, which contains the high-order bits of the 10-bit address and the R/$\overline{\text{W}}$ bit set to '1'. After the third address sequence is performed, the UA bit is not set, the module is now configured in Transmit mode and clock stretching is controlled by the BF flag, as in 7-Bit Slave Transmit mode (see Figure 58-18).

58.5.2.7 BYTE NACKING

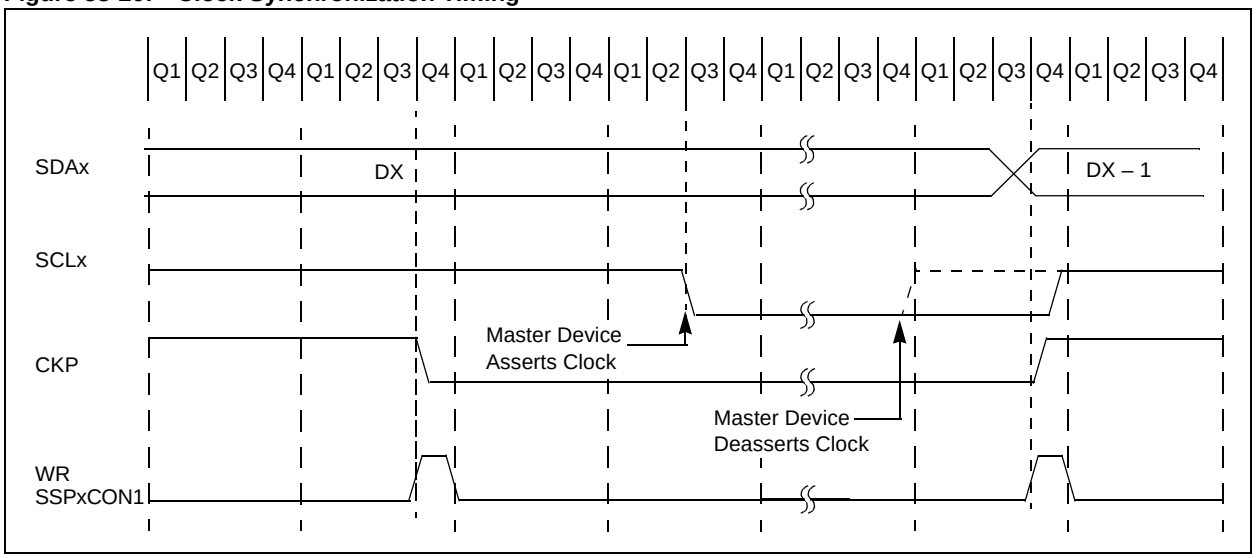
When the AHEN bit (SSPxCON3<1>) is set, the CKP bit is cleared by hardware after the 8th falling edge of SCLx, for a received matching address byte. When the DHEN bit (SSPxCON3<0>) is set, CKP is cleared after the 8th falling edge of SCLx for received data.

Stretching after the 8th falling edge of SCLx allows the slave to look at the received address or data and decide if it wants to $\overline{\text{ACK}}$ the received data.

58.5.2.8 CLOCK SYNCHRONIZATION AND THE CKP BIT

When the CKP bit is cleared, the SCLx output is forced to '0'. However, clearing the CKP bit will not assert the SCLx output low until the SCLx output is already sampled low. Therefore, the CKP bit will not assert the SCLx line until an external I^2C master device has already asserted the SCLx line. The SCLx output will remain low until the CKP bit is set and all other devices on the I^2C bus have deasserted SCLx. This ensures that a write to the CKP bit will not violate the minimum high time requirement for SCLx (see Figure 58-20).

Figure 58-20: Clock Synchronization Timing



58.5.2.9 GENERAL CALL ADDRESS SUPPORT

The addressing procedure for the I^2C 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 I^2C protocol; it consists of all '0's with R/$\overline{W}$ = 0.

The General Call Address is recognized when the General Call Enable bit, GCEN, is enabled (SSPxCON2<7> set). Following a Start bit detect, eight bits are shifted into the SSPxSR and the address is compared against the SSPxADD. It is also compared to the General Call Address and fixed in hardware.

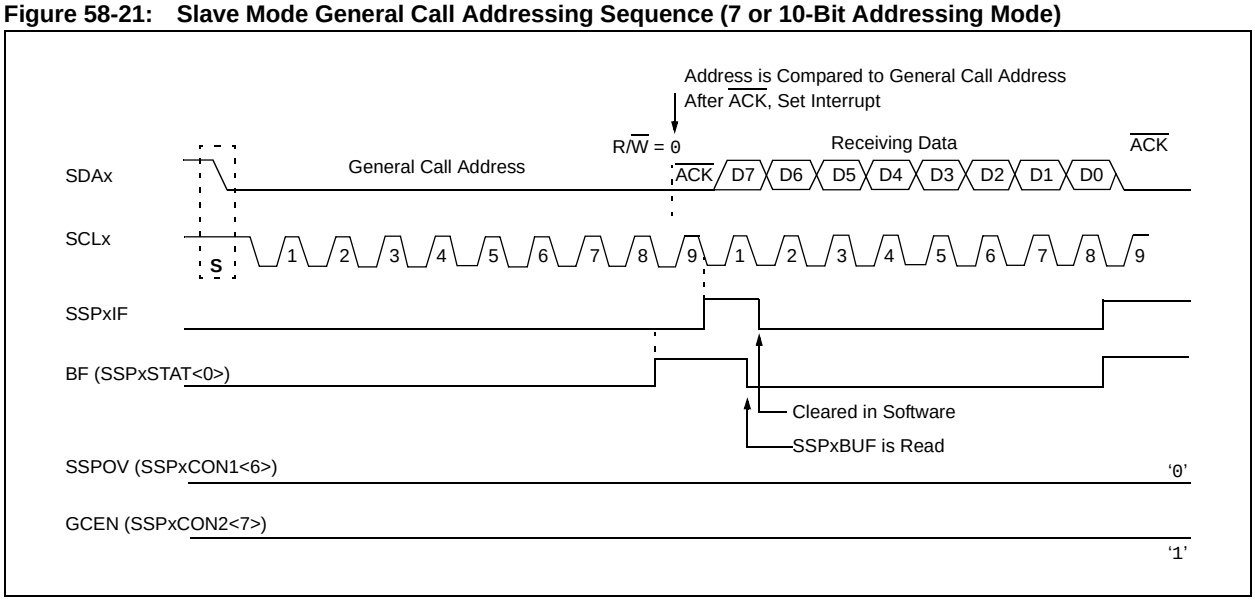
If the General Call Address matches, the SSPxSR is transferred to the SSPxBUF, the BF flag bit is set (eighth bit) and on the falling edge of the ninth bit ($\overline{ACK}$ bit), the SSPxIF interrupt flag bit is set.

When the interrupt is serviced, the source for the interrupt can be checked by reading the contents of the SSPxBUF. The value can be used to determine if the address was device-specific or a General Call Address.

In 10-bit mode, the SSPxADD is required to be updated for the second half of the address to match and the UA bit is set (SSPxSTAT<1>). If the General Call Address is sampled when the GCEN bit is set, while the slave is configured in 10-Bit Addressing 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 58-21).

If the AHEN bit (SSPxCON3<>) is set, just as with any other address reception, the slave hardware stretches the clock after the 8th falling edge of SCLx. The slave must then set its ACKDT value and release the clock with communication progressing as it would normally.

Figure 58-21: Slave Mode General Call Addressing Sequence (7 or 10-Bit Addressing Mode)



58.5.3 Master Mode

Master mode is enabled by setting and clearing the appropriate SSPM bits in SSPxCON1 and by setting the SSPEN bit. In Master mode, the SCLx and SDAx lines are manipulated by the MSSP hardware if the TRIS bits are set.

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 I^2C bus may be taken when the P bit is set, or the bus is Idle, with both the S and P bits clear.

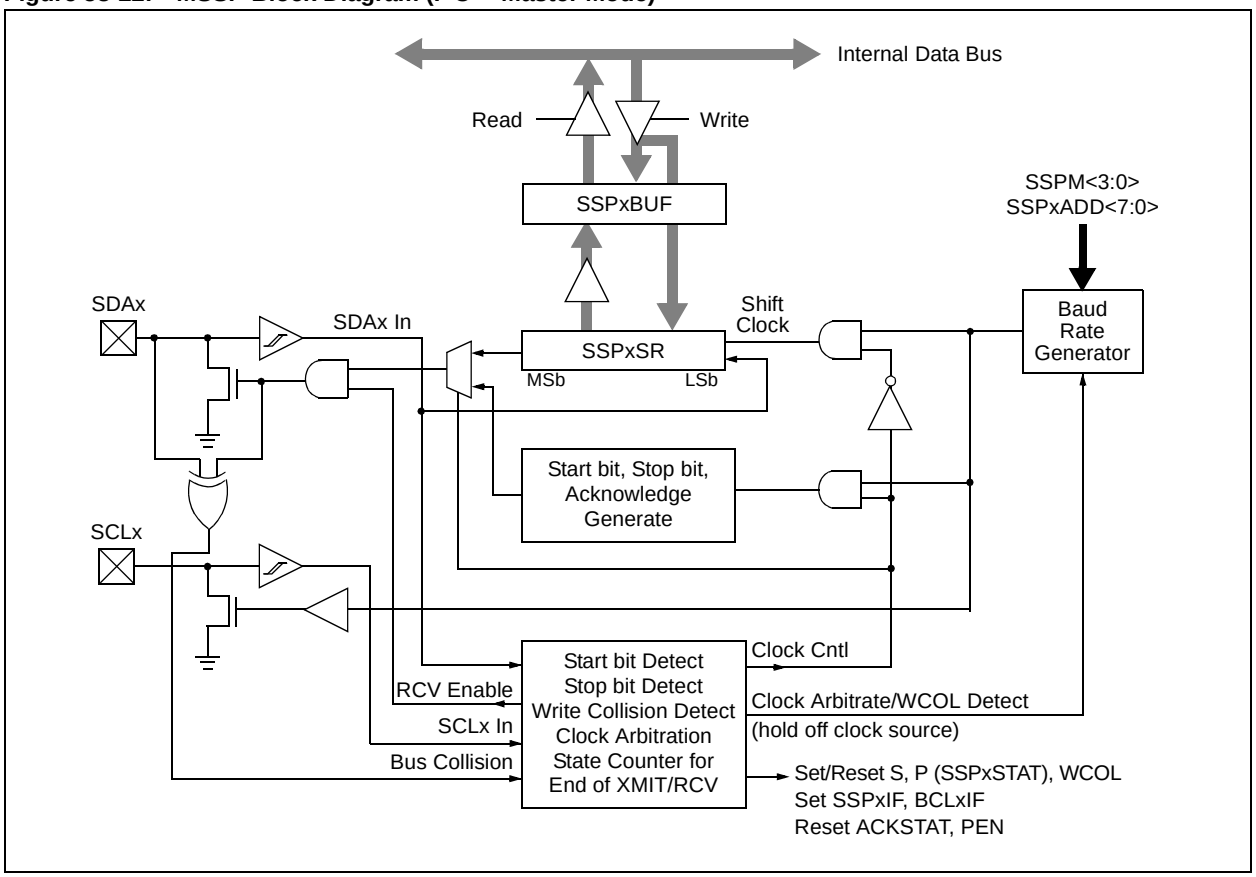
In Firmware Controlled Master mode, user code conducts all I^2C bus operations based on Start and Stop bit conditions.

Once Master mode is enabled, the user has six options:

1. Assert a Start condition on SDAx and SCLx.
2. Assert a Repeated Start condition on SDAx and SCLx.
3. Write to the SSPxBUF register, initiating transmission of data/address.
4. Configure the I^2C port to receive data.
5. Generate an Acknowledge condition at the end of a received byte of data.
6. Generate a Stop condition on SDAx and SCLx.

Note: The MSSP module, when configured in I^2C Master mode, does not allow queueing of events. For instance, the user is not allowed to initiate a Start condition and immediately write the SSPxBUF register to initiate transmission before the Start condition is complete. In this case, the SSPxBUF will not be written to and the WCOL bit will be set, indicating that a write to the SSPxBUF did not occur.

Figure 58-22: MSSP Block Diagram (I^2C™ Master Mode)



The following events will cause the MSSP Interrupt Flag bit, SSPxIF, to be set (and an MSSP interrupt, if enabled):

- Start condition
- Stop condition
- Data transfer byte transmitted/received
- Acknowledge transmit
- Repeated Start

58.5.3.1 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 SDAx, while SCLx outputs the serial clock. The first byte transmitted contains the slave address of the receiving device (7 bits) and the Read/Write (R/$\overline{W}$) bit. In this case, the R/$\overline{W}$ 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/$\overline{W}$ bit. In this case, the R/$\overline{W}$ bit will be logic '1'. Thus, the first byte transmitted is a 7-bit slave address, followed by a '1' to indicate the receive bit. Serial data is received via SDAx, while SCLx 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 the SPI mode operation is used to set the SCLx clock frequency for either 100 kHz, 400 kHz or 1 MHz I²C operation. See **Section 58.3 "Baud Rate Generator"** for more details.

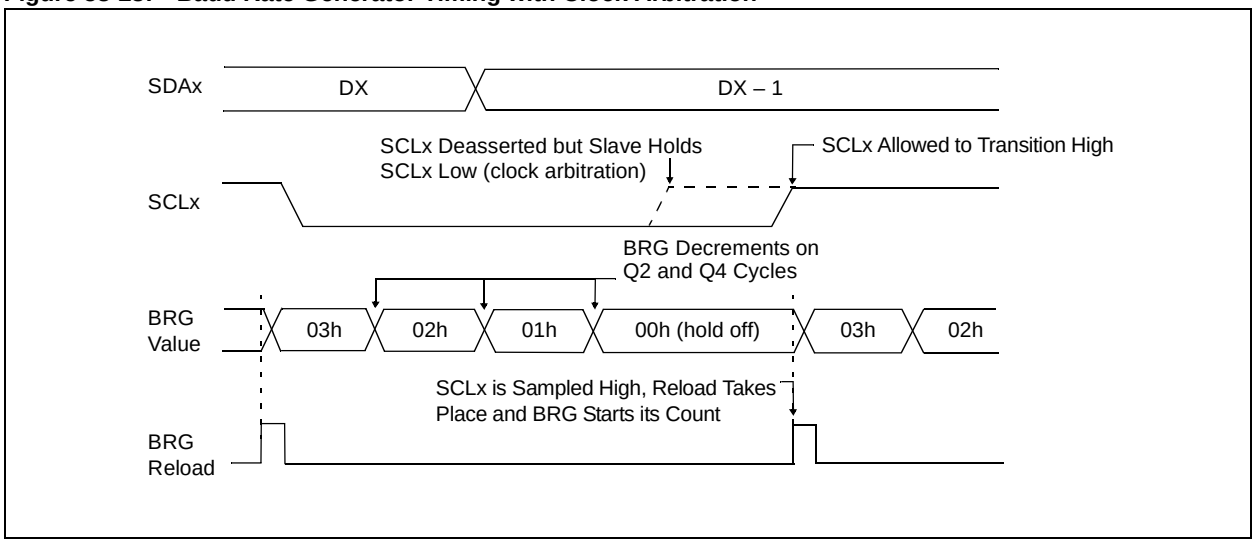
A typical transmit sequence would go as follows:

1. The user generates a Start condition by setting the Start Enable bit, SEN (SSPxCON2<0>).
2. SSPxIF is set. The MSSP module will wait the required start time before any other operation takes place.
3. The user loads the SSPxBUF with the slave address to transmit.
4. Address is shifted out the SDAx pin until all 8 bits are transmitted.
5. The MSSP module shifts in the $\overline{ACK}$ bit from the slave device and writes its value into the SSPxCON2 register (SSPxCON2<6>).
6. The MSSP module generates an interrupt at the end of the ninth clock cycle by setting the SSPxIF bit.
7. The user loads the SSPxBUF with eight bits of data.
8. Data is shifted out the SDAx pin until all 8 bits are transmitted.
9. The MSSP module shifts in the $\overline{ACK}$ bit from the slave device and writes its value into the SSPxCON2 register (SSPxCON2<6>).
10. The MSSP module generates an interrupt at the end of the ninth clock cycle by setting the SSPxIF bit.
11. The user generates a Stop condition by setting the Stop Enable bit, PEN (SSPxCON2<2>).
12. Interrupt is generated once the Stop condition is complete.

58.5.3.2 CLOCK ARBITRATION

Clock arbitration occurs when the master, during any receive, transmit or Repeated Start/Stop condition, deasserts the SCLx pin (SCLx allowed to float high). When the SCLx pin is allowed to float high, the Baud Rate Generator (BRG) is suspended from counting until the SCLx pin is actually sampled high. When the SCLx pin is sampled high, the Baud Rate Generator is reloaded with the contents of SSPxADD<6:0> and begins counting. This ensures that the SCLx 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 58-23).

Figure 58-23: Baud Rate Generator Timing with Clock Arbitration



58.5.3.3 I²C MASTER MODE START CONDITION TIMING

To initiate a Start condition, the user sets the Start Enable bit, SEN (SSPxCON2<0>). If the SDAx and SCLx pins are sampled high, the Baud Rate Generator is reloaded with the contents of SSPxADD<6:0> and starts its count. If SCLx and SDAx are both sampled high when the Baud Rate Generator times out (TBRG), the SDAx pin is driven low. The action of the SDAx being driven low while SCLx is high is the Start condition and causes the S bit (SSPxSTAT<3>) to be set. Following this, the Baud Rate Generator is reloaded with the contents of SSPxADD<6:0> and resumes its count. When the Baud Rate Generator times out (TBRG), the SEN bit (SSPxCON2<0>) will be automatically cleared by hardware; the Baud Rate Generator is suspended, leaving the SDAx line held low and the Start condition is complete.

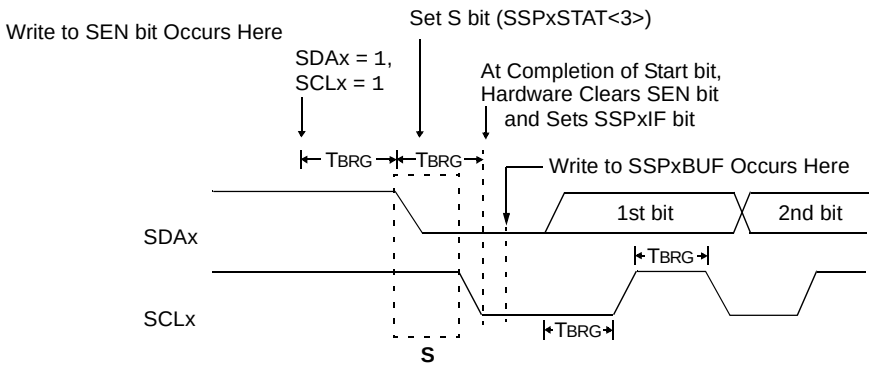
Note:	If, at the beginning of the Start condition, the SDAx and SCLx pins are already sampled low, or if during the Start condition, the SCLx line is sampled low before the SDAx line is driven low, a bus collision occurs. The Bus Collision Interrupt Flag, BCLxIF, is set, the Start condition is aborted and the I²C module is reset into its Idle state.

58.5.3.3.1 WCOL Status Flag

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

Note:	Because queueing of events is not allowed, writing to the lower 5 bits of SSPxCON2 is disabled until the Start condition is complete.

Figure 58-24: First Start Bit Timing



58.5.3.4 I²C MASTER MODE REPEATED START CONDITION TIMING

A Repeated Start condition occurs when the RSEN bit (SSPxCON2<1>) is programmed high and the I²C logic module is in the Idle state. When the RSEN bit is set, the SCLx pin is asserted low. When the SCLx pin is sampled low, the Baud Rate Generator is loaded with the contents of SSPxADD<5:0> and begins counting. The SDAx pin is released (brought high) for one Baud Rate Generator count (TBRG). When the Baud Rate Generator times out, if SDAx is sampled high, the SCLx pin will be deasserted (brought high). When SCLx is sampled high, the Baud Rate Generator is reloaded with the contents of SSPxADD<6:0> and begins counting. SDAx and SCLx must be sampled high for one TBRG. This action is then followed by assertion of the SDAx pin (SDAx = 0) for one TBRG while SCLx is high. Following this, the RSEN bit (SSPxCON2<1>) will be automatically cleared and the Baud Rate Generator will not be reloaded, leaving the SDAx pin held low. As soon as a Start condition is detected on the SDAx and SCLx pins, the S bit (SSPxSTAT<3>) will be set. The SSPxIF 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.

2: A bus collision during the Repeated Start condition occurs if:

- SDAx is sampled low when SCLx goes from low-to-high.
- SCLx goes low before SDAx is asserted low. This may indicate that another master is attempting to transmit a data '1'.

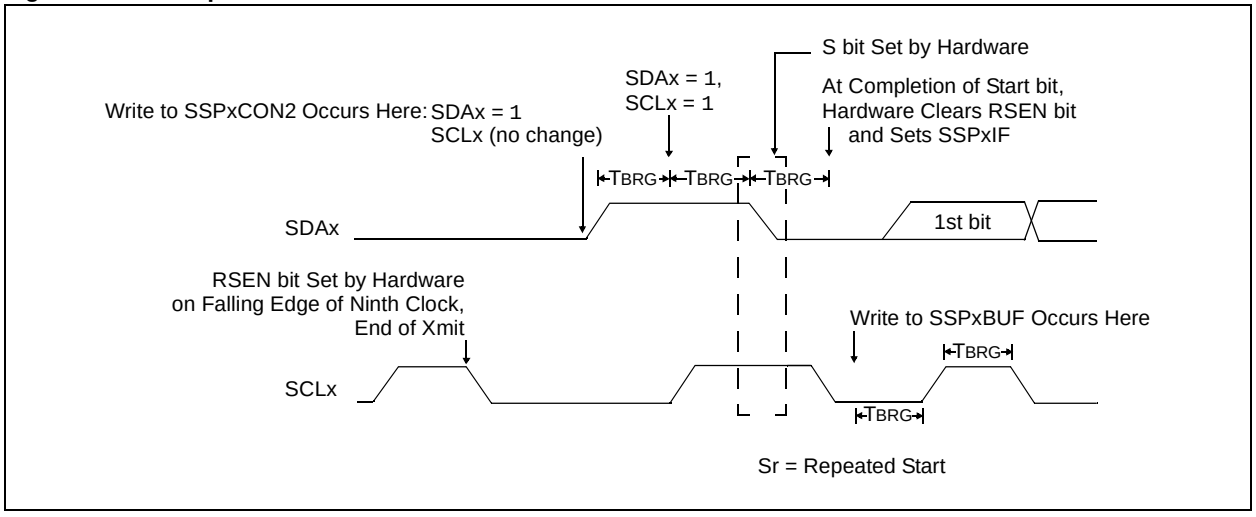
Immediately following the SSPxIF bit getting set, the user may write the SSPxBUF with the 7-bit address in 7-bit mode or the default first address in 10-bit mode. After the first eight bits are transmitted and an $\overline{ACK}$ is received, the user may then transmit an additional eight bits of address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

58.5.3.4.1 WCOL Status Flag

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

Note: Because queueing of events is not allowed, writing of the lower 5 bits of SSPxCON2 is disabled until the Repeated Start condition is complete.

Figure 58-25: Repeated Start Condition Waveform



58.5.3.5 I²C MASTER MODE TRANSMISSION

Transmission of a data byte, a 7-bit address, or the other half of a 10-bit address, is accomplished by simply writing a value to the SSPxBUF register. This action will set the Buffer Full flag bit, 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 SDAx pin after the falling edge of SCLx is asserted (see data hold time specification Parameter 106). SCLx is held low for one Baud Rate Generator rollover count (TBRG). Data should be valid before SCLx is released high (see data setup time specification Parameter 107). When the SCLx pin is released high, it is held that way for TBRG. The data on the SDAx pin must remain stable for that duration and some hold time after the next falling edge of SCLx. After the eighth bit is shifted out (the falling edge of the eighth clock), the BF flag is cleared and the master releases SDAx. This allows the slave device being addressed to respond with an $\overline{ACK}$ bit during the ninth bit time if an address match occurred, or if data was received properly. The status of $\overline{ACK}$ is written into the ACKDT bit 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 SSPxIF bit is set and the master clock (Baud Rate Generator) is suspended until the next data byte is loaded into the SSPxBUF, leaving SCLx low and SDAx unchanged (Figure 58-26).

After the write to the SSPxBUF, each bit of the address will be shifted out on the falling edge of SCLx until all seven address bits and the R/$\overline{W}$ bit are completed. On the falling edge of the eighth clock, the master will deassert the SDAx pin, allowing the slave to respond with an Acknowledge. On the falling edge of the ninth clock, the master will sample the SDAx pin to see if the address was recognized by a slave. The status of the $\overline{ACK}$ bit is loaded into the ACKSTAT status bit (SSPxCON2<6>). Following the falling edge of the ninth clock transmission of the address, the SSPxIF is set, the BF flag is cleared and the Baud Rate Generator is turned off until another write to the SSPxBUF takes place, holding SCLx low and allowing SDAx to float.

58.5.3.5.1 BF Status Flag

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

58.5.3.5.2 WCOL Status Flag

If the user writes the SSPxBUF when a transmit is already in progress (i.e., SSPxSR is still shifting out a data byte), the WCOL bit is set and the contents of the buffer are unchanged (the write doesn't occur) after 2 TCY after the SSPxBUF write. If SSPxBUF is rewritten within 2 TCY, the WCOL bit is set and SSPxBUF is updated. This may result in a corrupted transfer.

The user should verify that the WCOL bit is clear after each write to SSPxBUF to ensure the transfer is correct. In all cases, WCOL must be cleared in software.

58.5.3.5.3 ACKSTAT Status Flag

In Transmit mode, the ACKSTAT bit (SSPxCON2<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 Address), or when the slave has properly received its data.

58.5.3.6 I^2C MASTER MODE RECEPTION

Master mode reception is enabled by programming the Receive Enable bit, RCEN (SSPxCON2<3>).

Note:	The MSSP module must be in an inactive 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 SCLx pin changes (high-to-low/low-to-high) and data is shifted into the SSPxSR. After the falling edge of the eighth clock, the receive enable flag is automatically cleared, the contents of the SSPxSR are loaded into the SSPxBUF, the BF flag bit is set, the SSPxIF flag bit is set and the Baud Rate Generator is suspended from counting, holding SCLx low. The MSSP is now in Idle state awaiting the next command. When the buffer is read by the CPU, the BF flag bit is automatically cleared. The user can then send an Acknowledge bit at the end of reception by setting the Acknowledge Sequence Enable bit, ACKEN (SSPxCON2<4>).

58.5.3.6.1 BF Status Flag

In receive operation, the BF bit is set when an address or data byte is loaded into SSPxBUF from SSPxSR; it is cleared when the SSPxBUF register is read.

58.5.3.6.2 SSPOV Status Flag

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

58.5.3.6.3 WCOL Status Flag

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

Figure 58-26: I²C™ Master Mode Waveform (Transmission, 7 or 10-Bit Addressing)

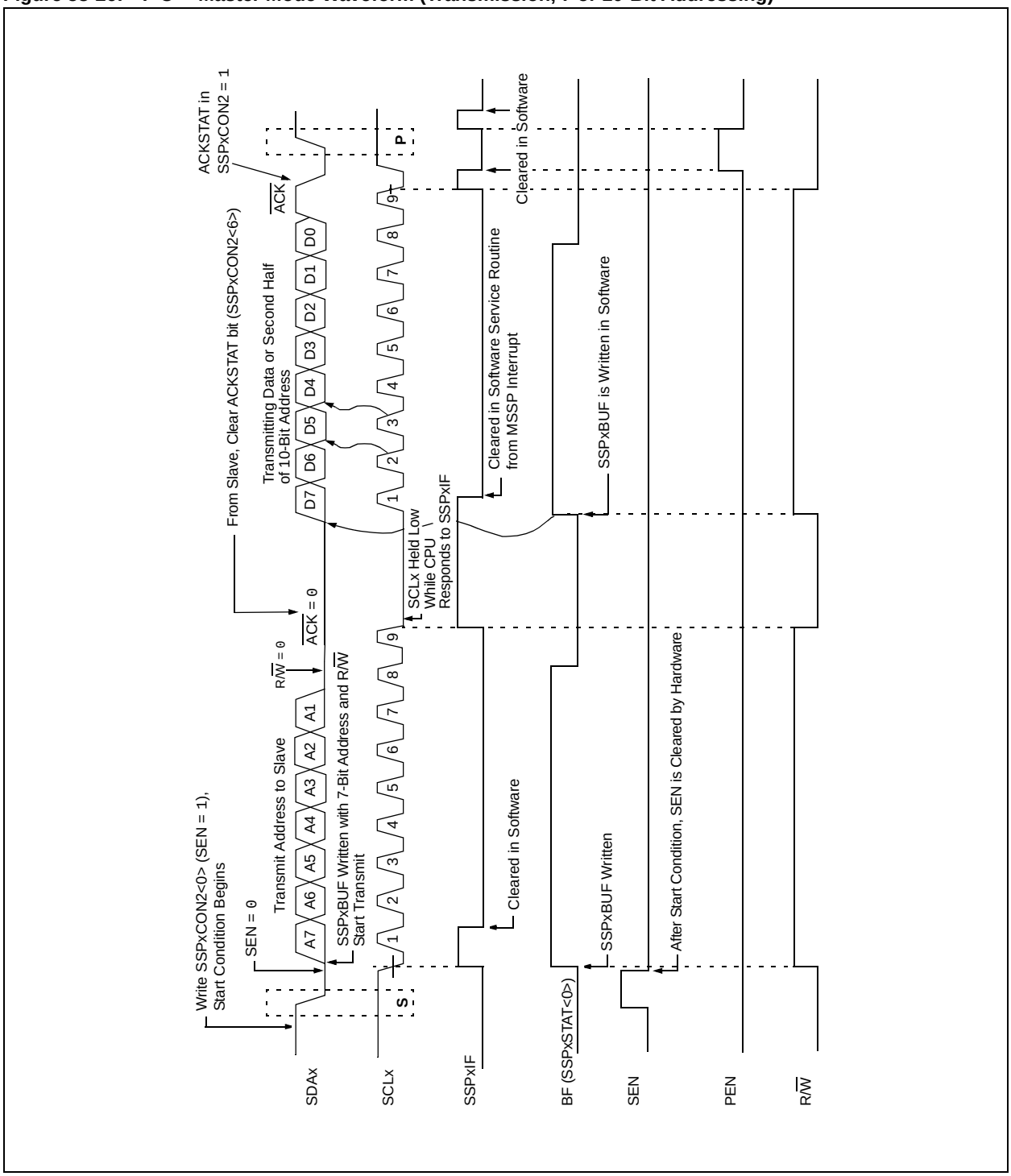
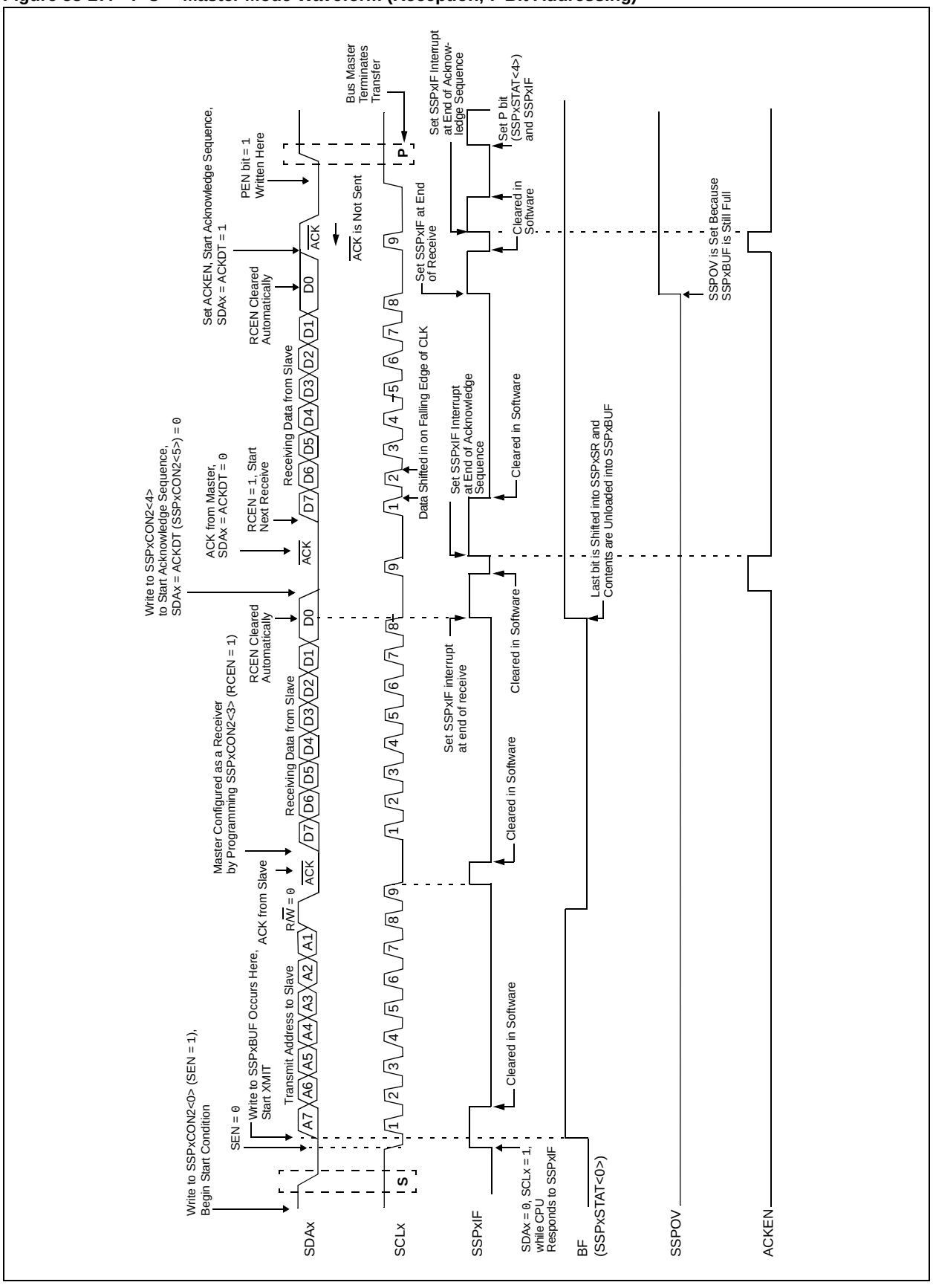


Figure 58-27: I²C™ Master Mode Waveform (Reception, 7-Bit Addressing)



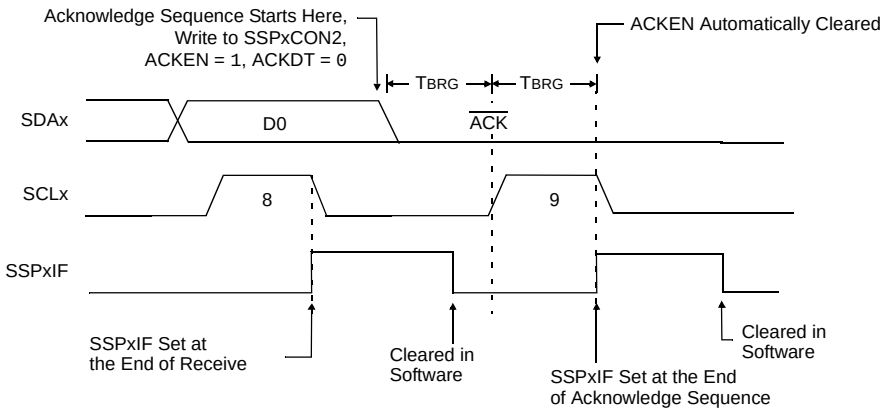
58.5.3.7 ACKNOWLEDGE SEQUENCE TIMING

An Acknowledge sequence is enabled by setting the Acknowledge Sequence Enable bit, ACKEN (SSPxCON2<4>). When this bit is set, the SCLx pin is pulled low and the contents of the Acknowledge data bit are presented on the SDAx pin. If the user wishes to generate an Acknowledge, then 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 SCLx pin is deasserted (pulled high). When the SCLx pin is sampled high (clock arbitration), the Baud Rate Generator counts for TBRG. The SCLx pin is then pulled low. Following this, the ACKEN bit is automatically cleared, the Baud Rate Generator is turned off and the MSSP module then goes into an inactive state (Figure 58-28).

58.5.3.7.1 WCOL Status Flag

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

Figure 58-28: Acknowledge Sequence Waveform



Note: TBRG = One Baud Rate Generator Period.

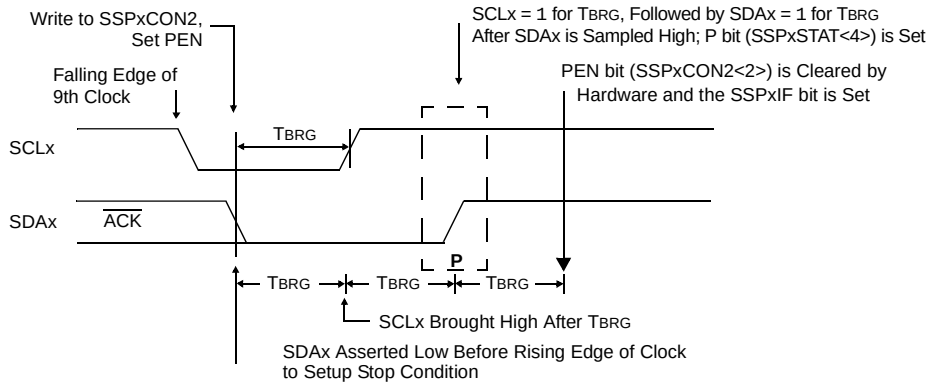
58.5.3.8 STOP CONDITION TIMING

A Stop bit is asserted on the SDAx pin at the end of a receive/transmit by setting the Stop Sequence Enable bit, PEN (SSPxCON2<2>). At the end of a receive/transmit, the SCLx line is held low after the falling edge of the ninth clock. When the PEN bit is set, the master will assert the SDAx line low. When the SDAx line is sampled low, the Baud Rate Generator is reloaded and counts down to '0'. When the Baud Rate Generator times out, the SCLx pin will be brought high and one TBRG (Baud Rate Generator rollover count) later, the SDAx pin will be deasserted. When the SDAx pin is sampled high while SCLx is high, the P bit (SSPxSTAT<4>) is set. A TBRG later, the PEN bit is cleared and the SSPxIF bit is set (Figure 58-29).

58.5.3.8.1 WCOL Status Flag

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

Figure 58-29: Stop Condition Receive or Transmit Mode



Note: TBRG = One Baud Rate Generator Period.

58.5.4 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 I^2C bus may be taken when the P bit (SSPxSTAT<4>) is set, or the bus is Idle, with both the S and P bits clear. When the bus is busy, enabling the MSSP interrupt will generate the interrupt when the Stop condition occurs.

In multi-master operation, the SDAx 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 BCLxIF bit.

The states where arbitration can be lost are:

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

58.5.4.1 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 SDAx pin, arbitration takes place when the master outputs a '1' on SDAx, by letting SDAx float high and another master asserts a '0'. When the SCLx pin floats high, data should be stable. If the expected data on SDAx is a '1' and the data sampled on the SDAx pin = 0, then a bus collision has taken place. The master will set the Bus Collision Interrupt Flag, BCLxIF, and reset the I^2C port to its Idle state (Figure 58-30).

If a transmit was in progress when the bus collision occurred, the transmission is Halted, the BF flag is cleared, the SDAx and SCLx lines are deasserted and the SSPxBUF can be written to. When the user services the bus collision Interrupt Service Routine (ISR), and if the I^2C bus is free, the user can resume communication by asserting a Start condition.

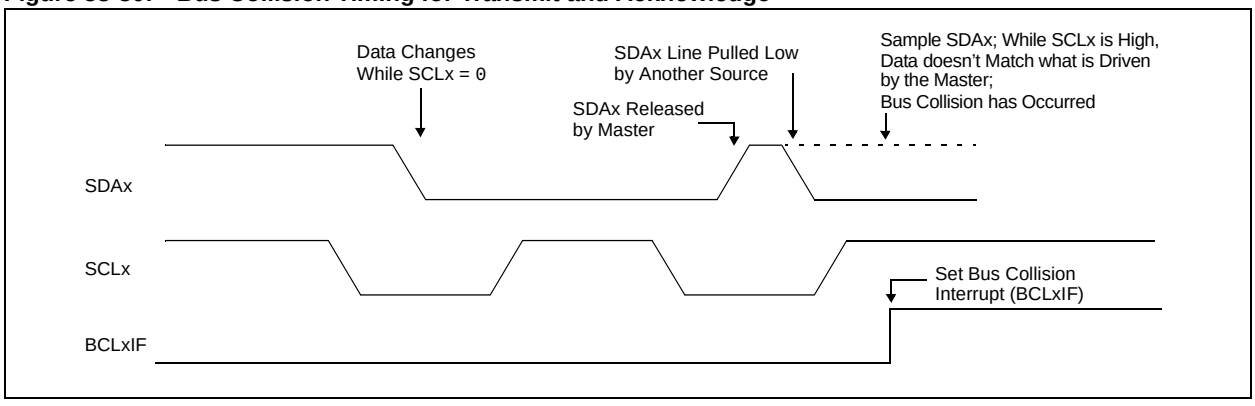
If a Start, Repeated Start, Stop or Acknowledge condition was in progress when the bus collision occurred, the condition is aborted, the SDAx and SCLx lines are deasserted and the respective control bits in the SSPxCON2 register are cleared. When the user services the bus collision Interrupt Service Routine and if the I^2C bus is free, the user can resume communication by asserting a Start condition.

The master will continue to monitor the SDAx and SCLx pins. If a Stop condition occurs, the SSPxIF bit will be set.

A write to the SSPxBUF 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 I^2C bus can be taken when the P bit is set in the SSPxSTAT register or the bus is Idle, and the S and P bits are cleared.

Figure 58-30: Bus Collision Timing for Transmit and Acknowledge



58.5.4.2 BUS COLLISION DURING A START CONDITION

During a Start condition, a bus collision occurs if:

- SDAx or SCLx are sampled low at the beginning of the Start condition (Figure 58-31).
- SCLx is sampled low before SDAx is asserted low (Figure 58-32).

During a Start condition, both the SDAx and the SCLx pins are monitored. If the SDAx pin is already low, or the SCLx pin is already low, then all of the following occur:

- The Start condition is aborted
- The BCLxIF flag is set
- The MSSP module is reset to its inactive state (Figure 58-31)

The Start condition begins with the SDAx and SCLx pins deasserted. When the SDAx pin is sampled high, the Baud Rate Generator is loaded from SSPxADD<6:0> and counts down to '0'. If the SCLx pin is sampled low while SDAx 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 SDAx pin is sampled low during this count, the BRG is reset and the SDAx line is asserted early (Figure 58-33). If, however, a '1' is sampled on the SDAx pin, the SDAx pin is asserted low at the end of the BRG count. The Baud Rate Generator is then reloaded and counts down to '0'. If the SCLx pin is sampled as '0' during this time, a bus collision does not occur. At the end of the BRG count, the SCLx pin is asserted low.

Note:	The reason that a 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 SDAx 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 58-31: Bus Collision During Start Condition (SDAx Only)

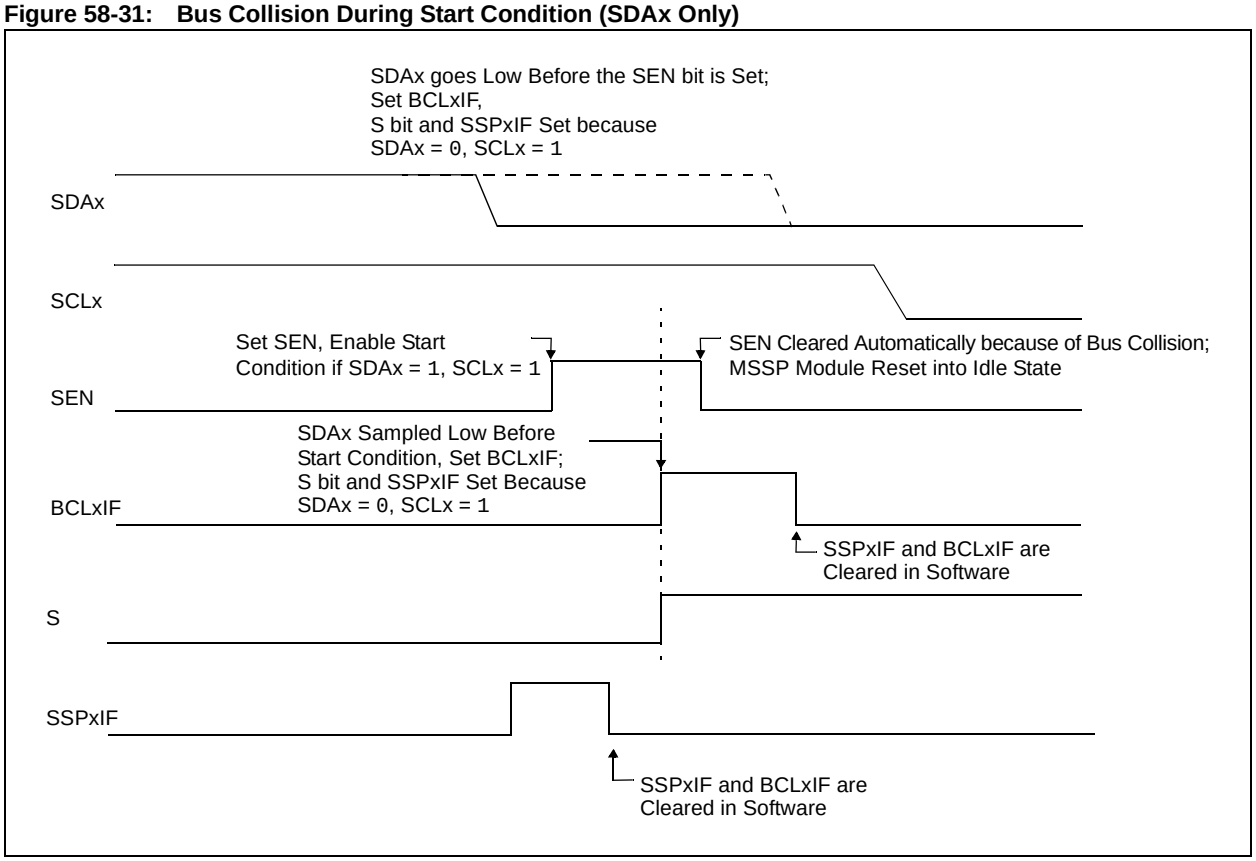


Figure 58-32: Bus Collision During Start Condition (SCLx = 0)

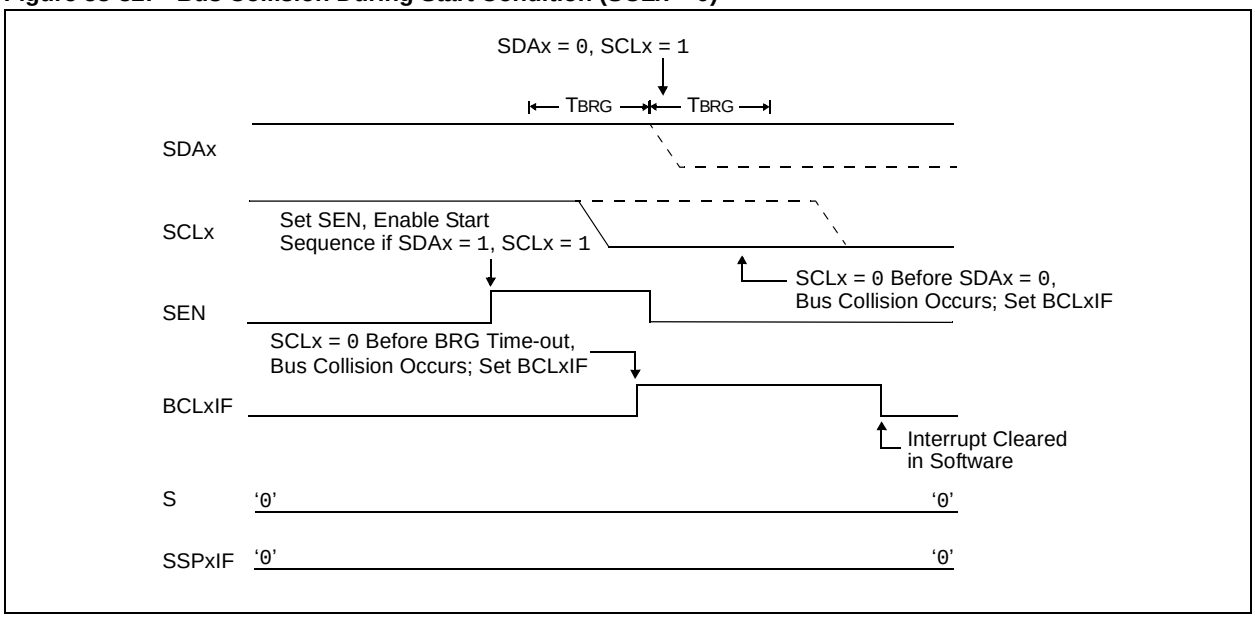
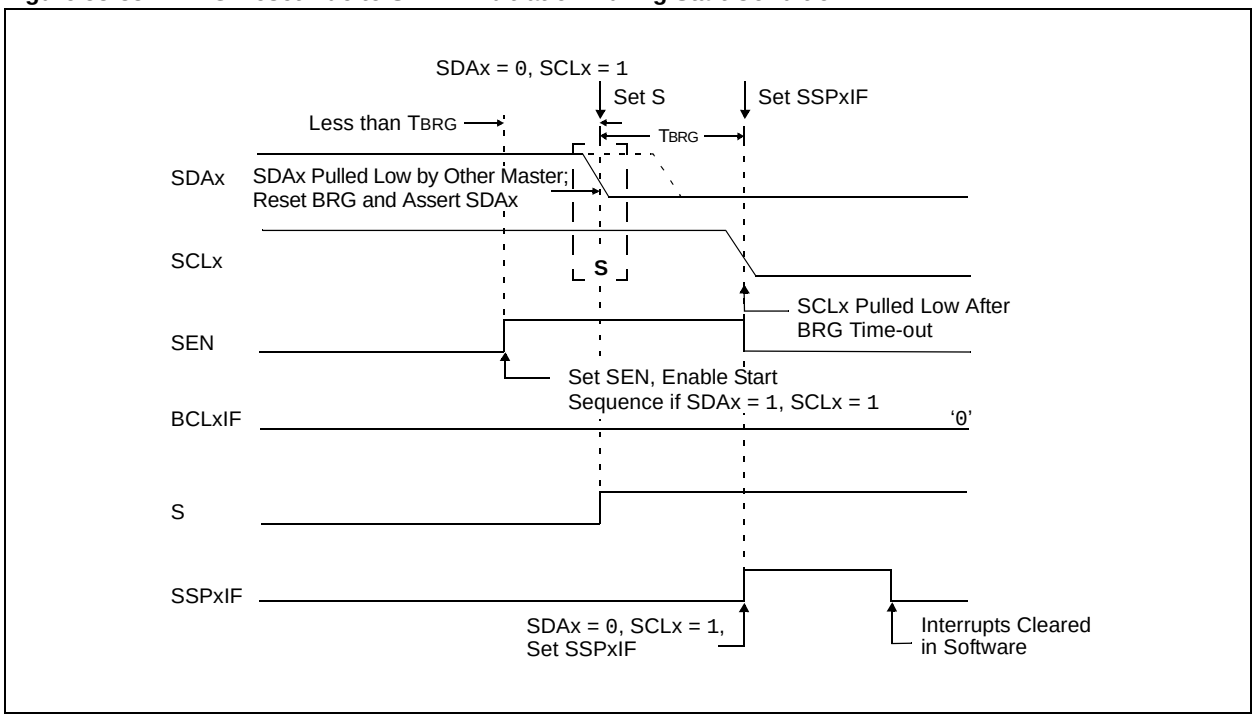


Figure 58-33: BRG Reset Due to SDAx Arbitration During Start Condition



58.5.4.3 BUS COLLISION DURING A REPEATED START CONDITION

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

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

When the user deasserts SDAx and the pin is allowed to float high, the BRG is loaded with SSPxADD<6:0> and counts down to '0'. The SCLx pin is then deasserted and when sampled high, the SDAx pin is sampled.

If SDAx is low, a bus collision has occurred (i.e., another master is attempting to transmit a data '0', Figure 58-34). If SDAx is sampled high, the BRG is reloaded and begins counting. If SDAx goes from high-to-low before the BRG times out, no bus collision occurs because no two masters can assert SDAx at exactly the same time.

If SCLx goes from high-to-low before the BRG times out and SDAx 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 (see Figure 58-35).

If, at the end of the BRG time-out, both SCLx and SDAx are still high, the SDAx pin is driven low and the BRG is reloaded and begins counting. At the end of the count, regardless of the status of the SCLx pin, the SCLx pin is driven low and the Repeated Start condition is complete.

Figure 58-34: Bus Collision During a Repeated Start Condition (Case 1)

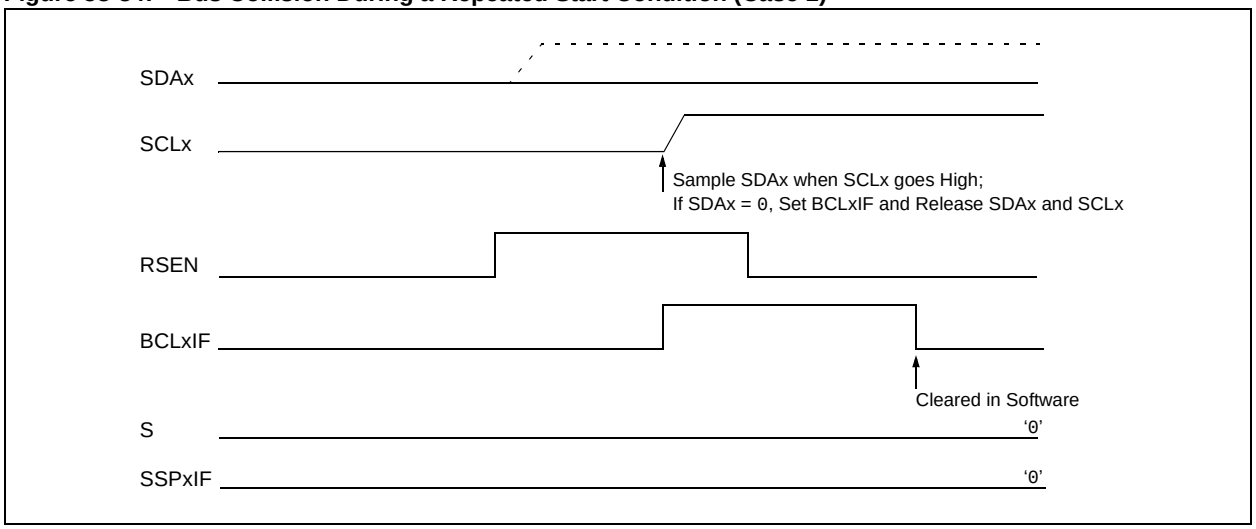
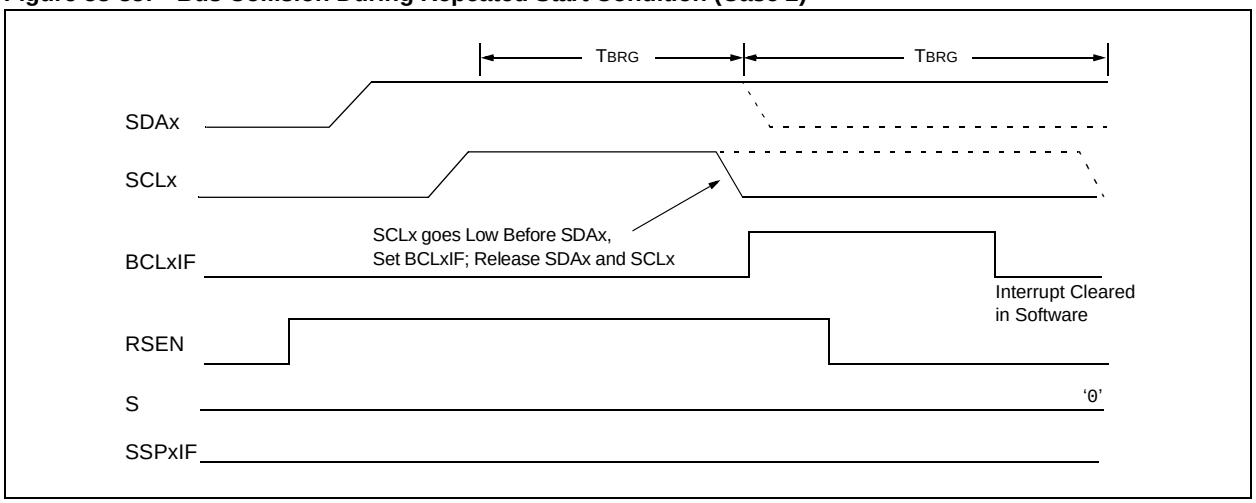


Figure 58-35: Bus Collision During Repeated Start Condition (Case 2)



58.5.4.4 BUS COLLISION DURING A STOP CONDITION

Bus collision occurs during a Stop condition if:

- After the SDAx pin has been deasserted and allowed to float high, SDAx is sampled low after the BRG has timed out.
- After the SCLx pin is deasserted, SCLx is sampled low before SDAx goes high.

The Stop condition begins with SDAx asserted low. When SDAx is sampled low, the SCLx pin is allowed to float. When the pin is sampled high (clock arbitration), the Baud Rate Generator is loaded with SSPxADD<6:0> and counts down to '0'. After the BRG times out, SDAx is sampled. If SDAx is sampled low, a bus collision has occurred. This is due to another master attempting to drive a data '0' (Figure 58-36). If the SCLx pin is sampled low before SDAx is allowed to float high, a bus collision occurs. This is another case of another master attempting to drive a data '0' (Figure 58-37).

Figure 58-36: Bus Collision During a Stop Condition (Case 1)

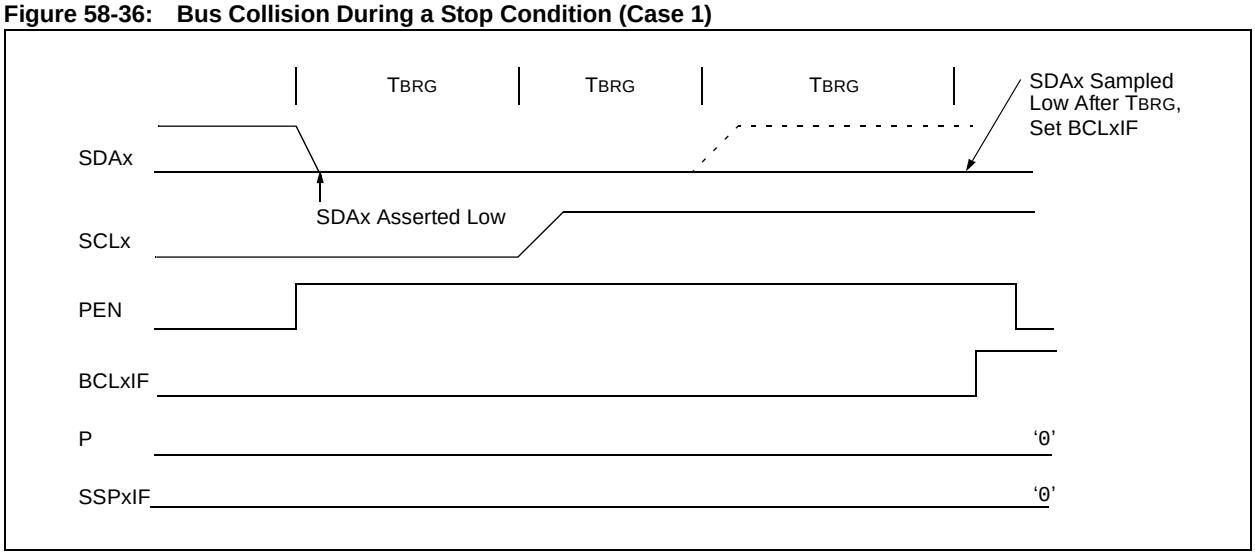
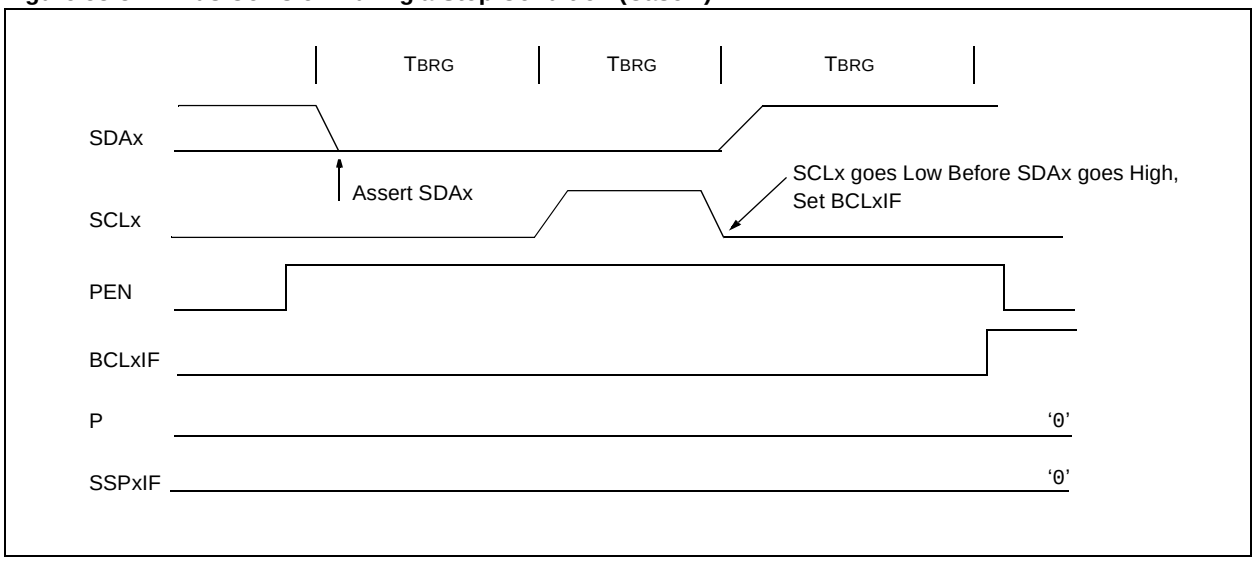


Figure 58-37: Bus Collision During a Stop Condition (Case 2)



58.6 OPERATION DURING SLEEP/IDLE MODES

58.6.1 Sleep Mode

In Master mode, all module clocks are Halted and the transmission/reception will remain in that state, during Sleep mode, until the devices wake. After the device returns to Run mode, the module will resume transmitting and receiving data.

In Slave mode with MSSP interrupts enabled, the controller can wake from Sleep mode after the master completes sending data. If an exit from Sleep mode is not desired, MSSP interrupts should be disabled.

In SPI Slave mode, the SPI Transmit/Receive Shift register operates asynchronously to the device, using the SCKx clock input. This allows the device to be placed in any power-managed mode and data to be shifted through the SPI Transmit/Receive Shift register. When all 8 bits have been received, the MSSP interrupt flag bit will be set, and if enabled, will wake the device.

In I^2C mode, the module can receive addresses or data. If an address match or complete byte transfer occurs, the module can wake the processor from Sleep (if the MSSP interrupt is enabled). There is no automatic method to prevent Sleep entry if a transmission or reception is active or pending. The application must synchronize Sleep entry with I^2C operation to avoid aborted messages.

58.6.2 Idle Mode

In Idle modes, the system clock continues to be provided to the peripherals. This allows the peripherals to continue operating normally while the controller core is shut down for power savings.

If MSSP interrupts are enabled, they can wake the controller from Idle mode when the master completes sending data. If an exit from Idle mode is not desired, MSSP interrupts should be disabled.

58.7 EFFECTS OF A RESET

In any mode, a device Reset disables the MSSP module and terminates the current transfer.

58.8 REGISTER MAPS

A summary of the registers associated with the PIC24F MSSP module is provided in Table 58-3.

Table 58-3: Representative PIC24F MSSPx Register Map

File Name	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
SSPxBUF	—	—	—	—	—	—	—	—	SSP Receive Buffer/Transmit Register								00xx
SSPxCON1	—	—	—	—	—	—	—	—	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000
SSPxCON2	—	—	—	—	—	—	—	—	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	0000
SSPxCON3	—	—	—	—	—	—	—	—	ACKTIM	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN	0000
SSPxSTAT	—	—	—	—	—	—	—	—	SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF	0000
SSPxADD	—	—	—	—	—	—	—	—	Address Register (I^2C™ Slave Mode)/Baud Rate Reload Register (SPI and I^2C Master Modes)								0000
SSPxMSK	—	—	—	—	—	—	—	—	SSP Address Mask Register (I^2C Slave Mode)								00FF

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

58.9 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 PIC24F device family, but the concepts are pertinent and could be used with modification and possible limitations. The current application notes related to the Master Synchronous Serial Port (MSSP) of PIC24F devices are:

Title	Application Note #
No related application notes at this time.	

Note: Please visit the Microchip web site (www.microchip.com) for additional application notes and code examples for the PIC24F family of devices.

58.10 REVISION HISTORY

Revision A (October 2011)

This is the original version of the document.

NOTES:

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Trademarks

Printed on recycled paper.

ISBN: 978-1-61341-691-4

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
ISO/TS 16949:2009

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://www.microchip.com/support
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Cleveland
Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Farmington Hills, MI
Tel: 248-538-2250
Fax: 248-538-2260

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

Santa Clara
Santa Clara, CA
Tel: 408-961-6444
Fax: 408-961-6445

Toronto
Mississauga, Ontario, Canada
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Chongqing
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

China - Hangzhou
Tel: 86-571-2819-3187
Fax: 86-571-2819-3189

China - Hong Kong SAR
Tel: 852-2401-1200
Fax: 852-2401-3431

China - Nanjing
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8203-2660
Fax: 86-755-8203-1760

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune
Tel: 91-20-2566-1512
Fax: 91-20-2566-1513

Japan - Yokohama
Tel: 81-45-471- 6166
Fax: 81-45-471-6122

Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-5778-366
Fax: 886-3-5770-955

Taiwan - Kaohsiung
Tel: 886-7-536-4818
Fax: 886-7-330-9305

Taiwan - Taipei
Tel: 886-2-2500-6610
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

UK - Wokingham
Tel: 44-118-921-5869
Fax: 44-118-921-5820

08/02/11