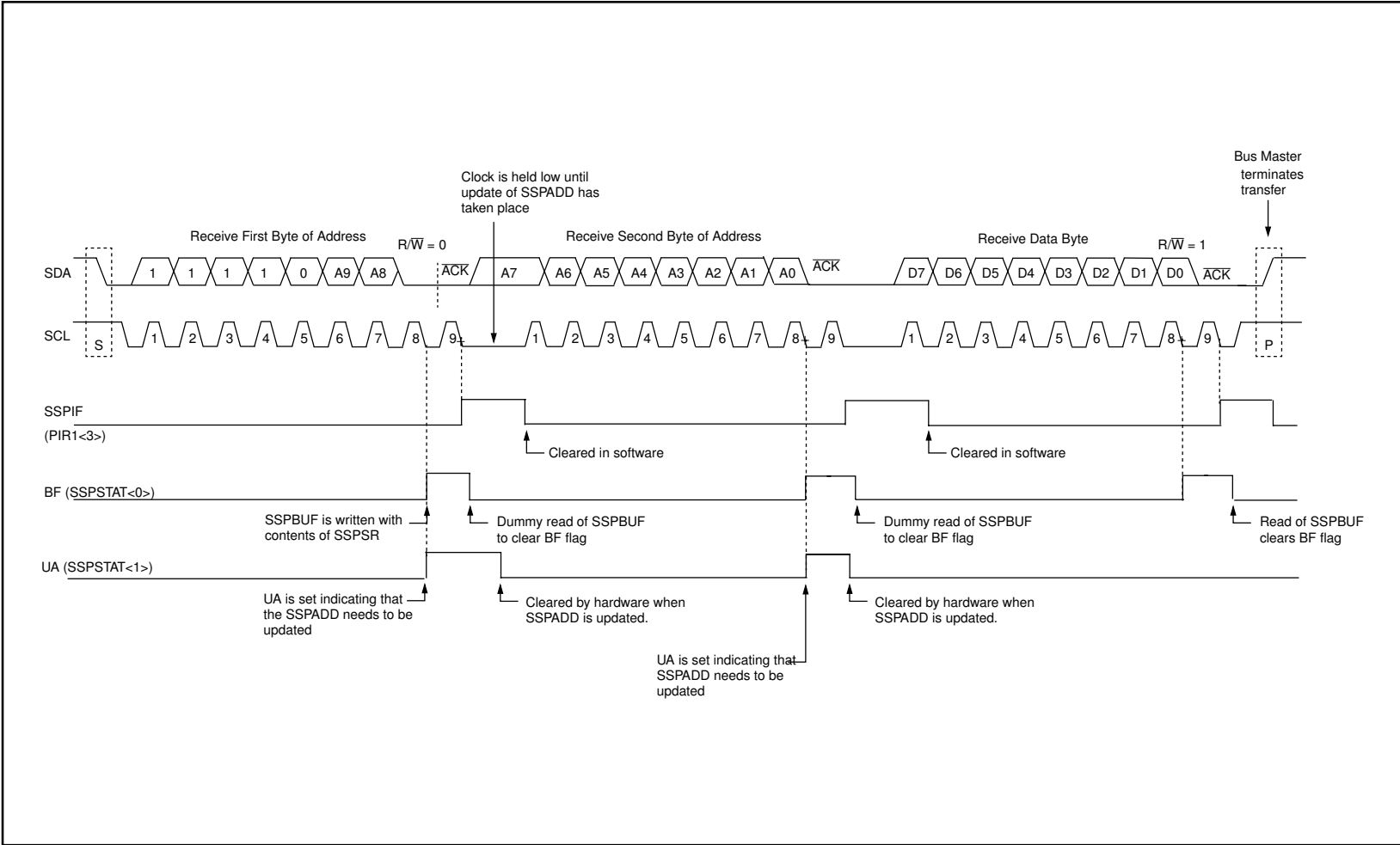


Figure 17-15: I²C Slave Mode Waveform (Reception 10-bit Address)



Section 17. MSSP

17.4.3 Sleep Operation

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

17.4.4 Effect of a Reset

A reset disables the MSSP module and terminates the current transfer.

Table 17-3: Registers Associated with I²C Operation

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other resets
INTCON	GIE	PEIE	TOIE	INTE	RBIE ⁽²⁾	TOIF	INTF	RBIF ⁽²⁾	0000 0000	0000 0000
PIR	SSPIF, BCLIF ⁽¹⁾								0, 0	0, 0
PIE	SSPIE, BCLIF ⁽¹⁾								0, 0	0, 0
SSPADD	Synchronous Serial Port (I ² C mode) Address Register (slave mode)/Baud Rate Generator (master mode)								0000 0000	0000 0000
SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxx xxxx	uuuu uuuu
SSPCON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
SSPCON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	0000 0000	0000 0000
SSPSTAT	SMP	CKE	D/Ā	P	S	R/Ā	UA	BF	0000 0000	0000 0000

Legend: x = unknown, u = unchanged, – = unimplemented read as '0'.

Shaded cells are not used by the SSP in I²C mode.

Note 1: The position of these bits is device dependent.

2: These bits may also be named GPIE and GPIF.

17.4.6 Multi-Master Mode

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

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

The states where arbitration can be lost are:

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

17.4.7 I²C Master Mode Support

Master Mode is enabled by setting and clearing the appropriate SSPM bits in SSPCON1 and by setting the SSPEN bit. Once master mode is enabled, the user has six options.

1. Assert a start condition on SDA and SCL.
2. Assert a Repeated Start condition on SDA and SCL.
3. Write to the SSPBUF register initiating transmission of data/address.
4. Generate a stop Condition on SDA and SCL.
5. Configure the I²C port to receive data.
6. Generate an acknowledge condition at the end of a received byte of data.

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

17.4.8 Baud Rate Generator

In I²C master mode, the reload value for the BRG is located in the lower 7 bits of the SSPADD register (Figure 17-18). When the BRG is loaded with this value, the BRG counts down to 0 and stops until another reload has taken place. In I²C master mode, the BRG is reloaded automatically. If Clock Arbitration is taking place for instance, the BRG will be reloaded when the SCL pin is sampled high (Figure 17-19).

Figure 17-18: Baud Rate Generator Block Diagram

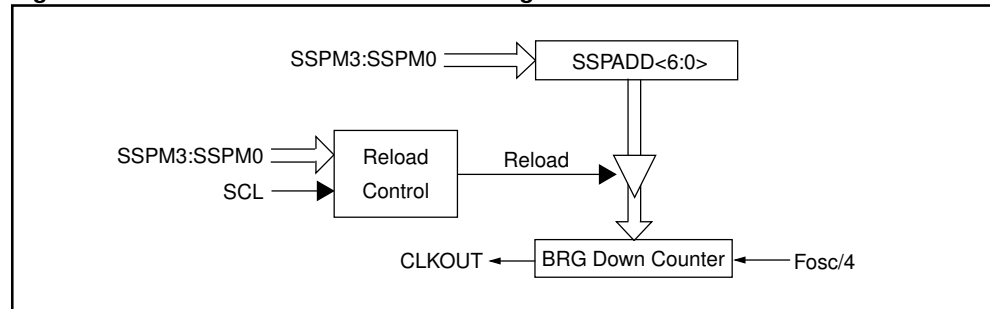


Figure 17-19: Baud Rate Generator Timing With Clock Arbitration

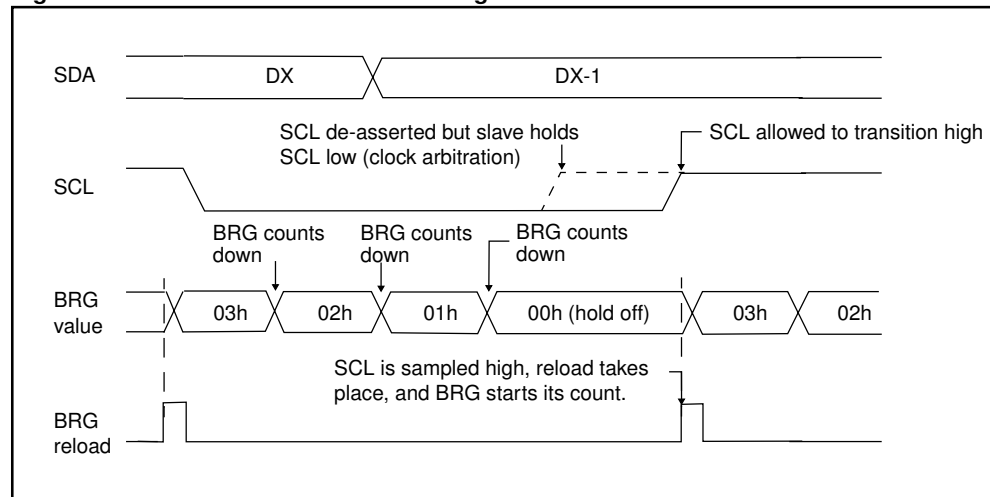
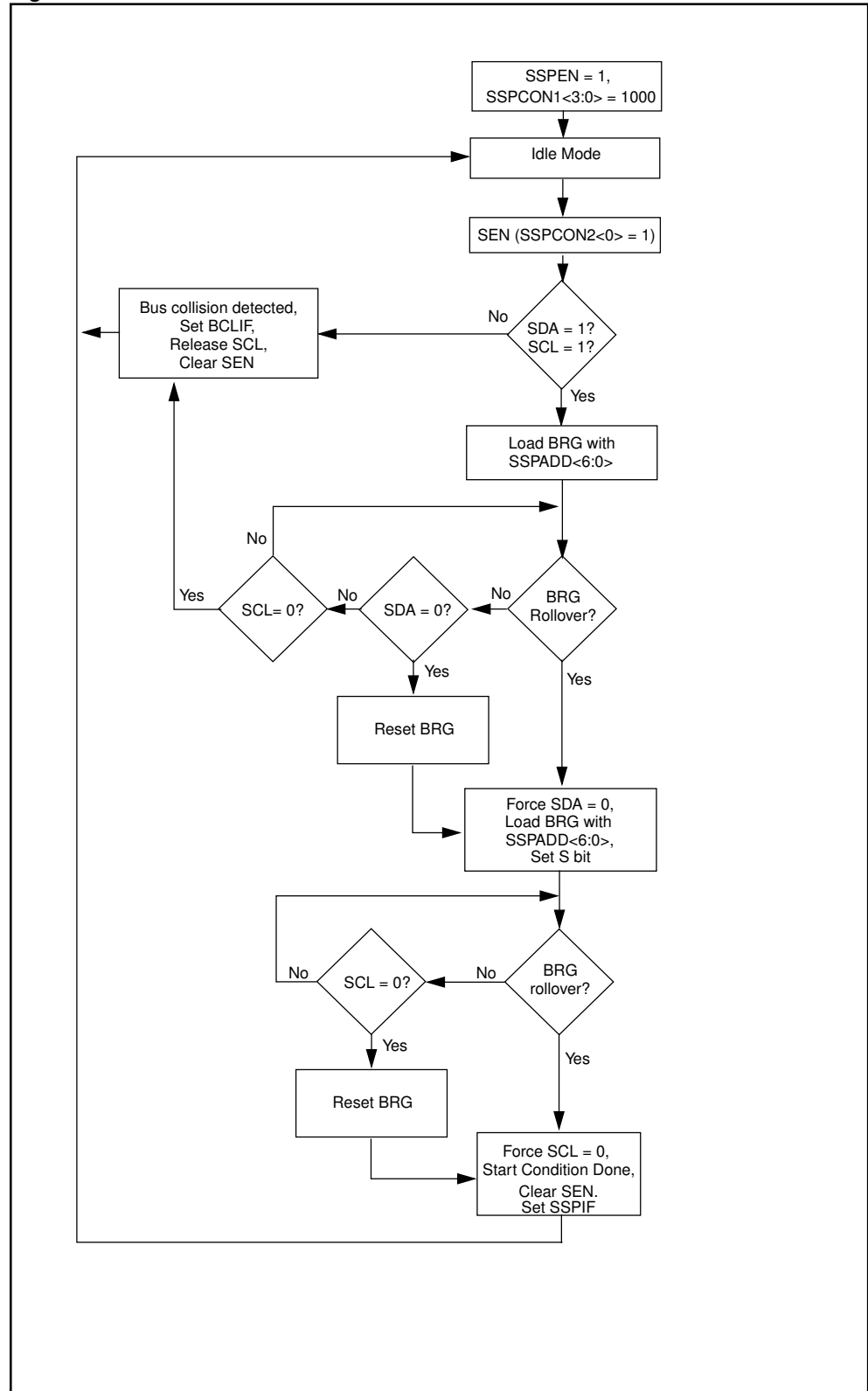


Figure 17-21: Start Condition Flowchart

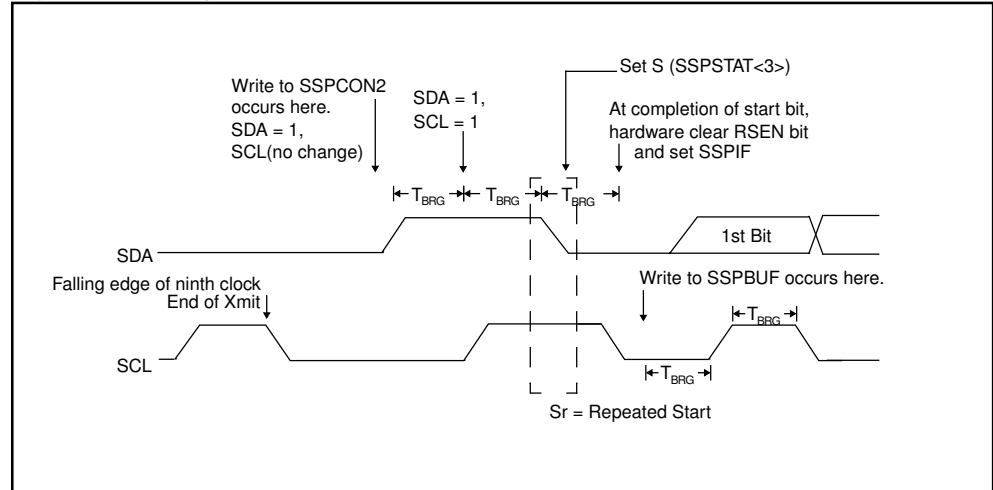


17.4.10.1 WCOL Status Flag

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

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

Figure 17-22: Repeat Start Condition Waveform



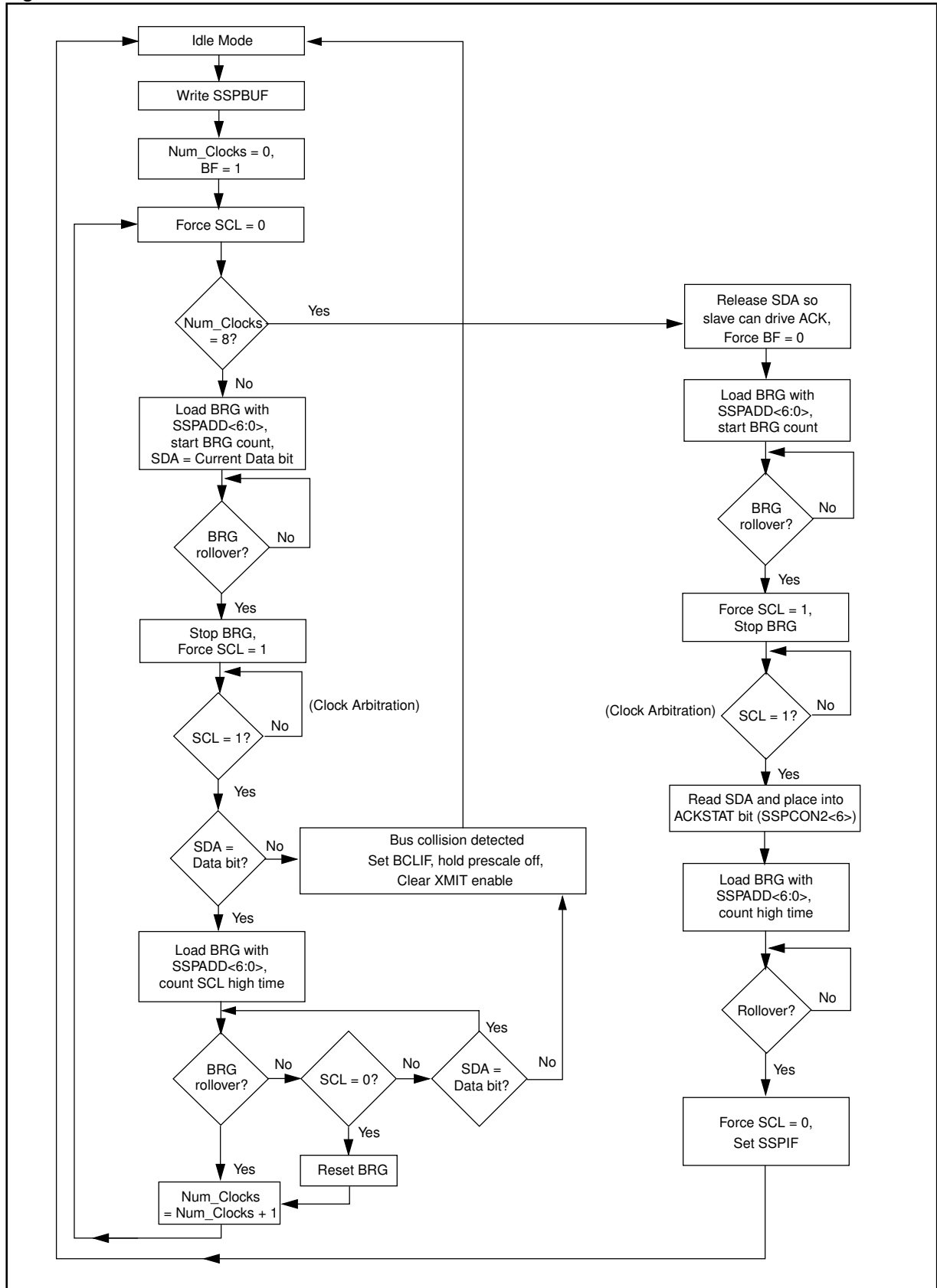
MSP

```

graph TD
    A([A]) --> D1{BRG rollover?}
    D1 -- Yes --> E[Force SDA = 0, Load BRG with SSPADD<6:0>]
    D1 -- No --> D2{SDA = 0?}
    D2 -- Yes --> F[Reset BRG]
    D2 -- No --> D3{SCL = 1?}
    D3 -- Yes --> A
    D3 -- No --> C([C])
    C --> D1
    E --> G[Set S]
    G --> D4{BRG rollover?}
    D4 -- Yes --> H[Force SCL = 0, Repeated Start condition done, Clear RSEN, Set SSPIF]
    D4 -- No --> D5{SCL = '0'?}
    D5 -- Yes --> I[Reset BRG]
    D5 -- No --> D4
    I --> H
    H --> B([B])
    B --> A
  
```

The flowchart illustrates the I2C Repeated Start Sequence. It begins at point A, where a decision is made on whether the BRG has rolled over. If yes, SDA is forced to 0 and the BRG is loaded with SSPADD<6:0>. If no, a decision is made on whether SDA is 0. If yes, the BRG is reset. If no, a decision is made on whether SCL is 1. If yes, the sequence loops back to A. If no, the sequence proceeds to point C. From point C, the sequence loops back to point A. From point E, the sequence proceeds to Set S. From Set S, a decision is made on whether the BRG has rolled over. If yes, SCL is forced to 0, the Repeated Start condition is done, RSEN is cleared, and SSPIF is set. If no, a decision is made on whether SCL is '0'. If yes, the BRG is reset. If no, the sequence loops back to the BRG rollover decision. From point H, the sequence proceeds to point B. From point B, the sequence loops back to point A.

Figure 17-25: Master Transmit Flowchart



17.4.12 I²C Master Mode Reception

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

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

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

17.4.12.1 BF Status Flag

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

17.4.12.2 SSPOV Status Flag

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

17.4.12.3 WCOL Status Flag

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

Figure 17-28: I²C Master Mode Waveform (Reception 7-Bit Address)

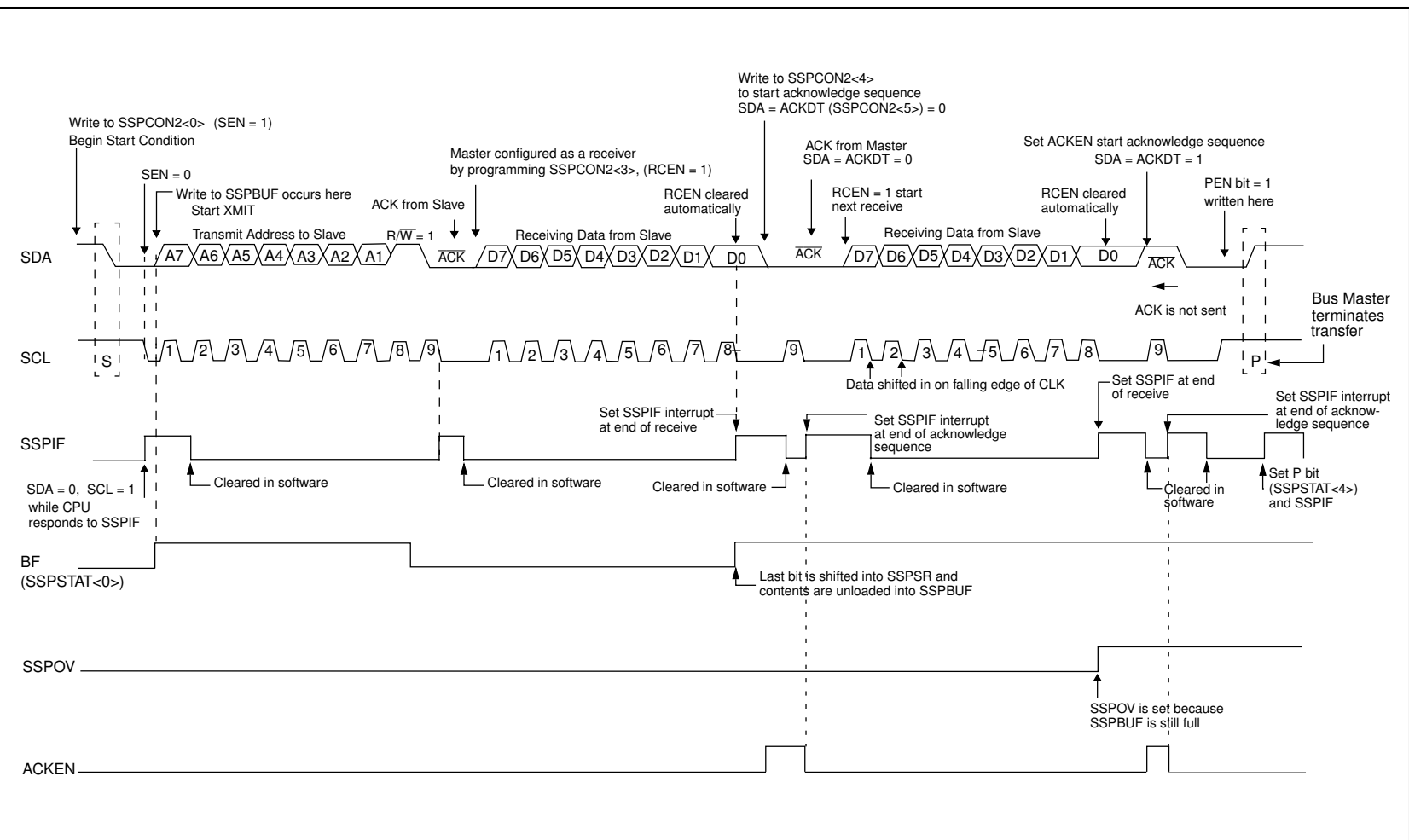


Figure 17-30: Acknowledge Flowchart

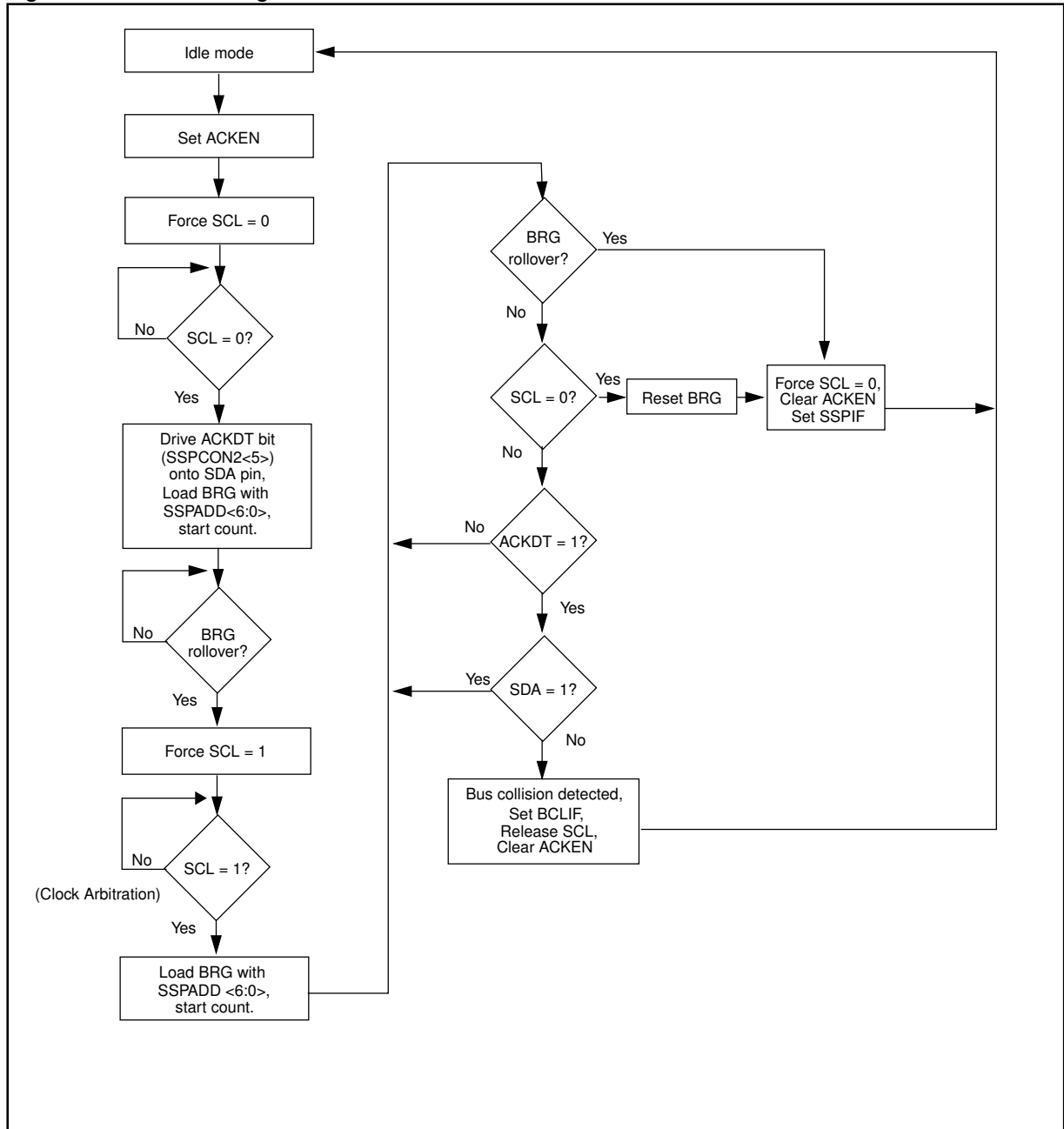
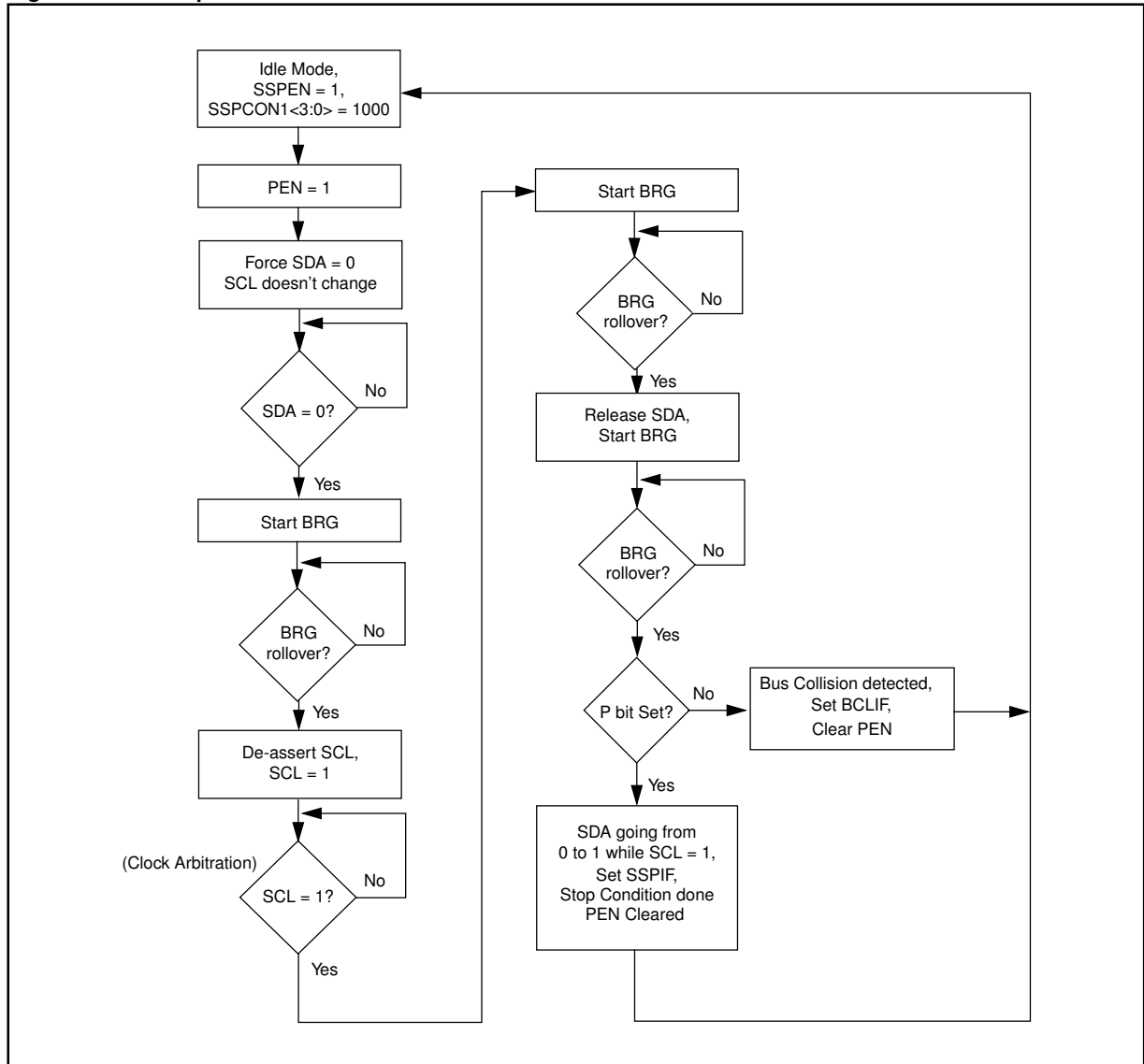


Figure 17-32: Stop Condition Flowchart



17.4.18 Multi-Master Communication, Bus Collision, and Bus Arbitration

Multi-Master mode support is achieved by bus arbitration. When the master outputs address/data bits onto the SDA pin, arbitration takes place when the master outputs a '1' on SDA by letting SDA float high and another master asserts a '0'. When the SCL pin floats high, data should be stable. If the expected data on SDA is a '1' and the data sampled on the SDA pin = '0', then a bus collision has taken place. The master will set the Bus Collision Interrupt Flag, BCLIF and reset the I²C port to its IDLE state. (Figure 17-34).

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

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

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

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

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

Figure 17-34: Bus Collision Timing for Transmit and Acknowledge

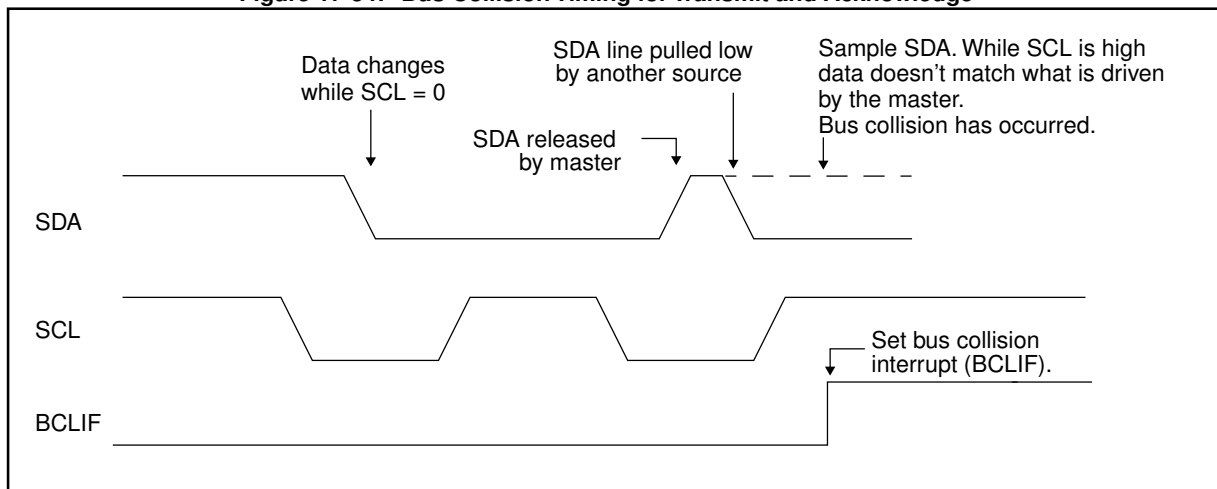


Figure 17-35: Bus Collision During Start Condition (SDA only)

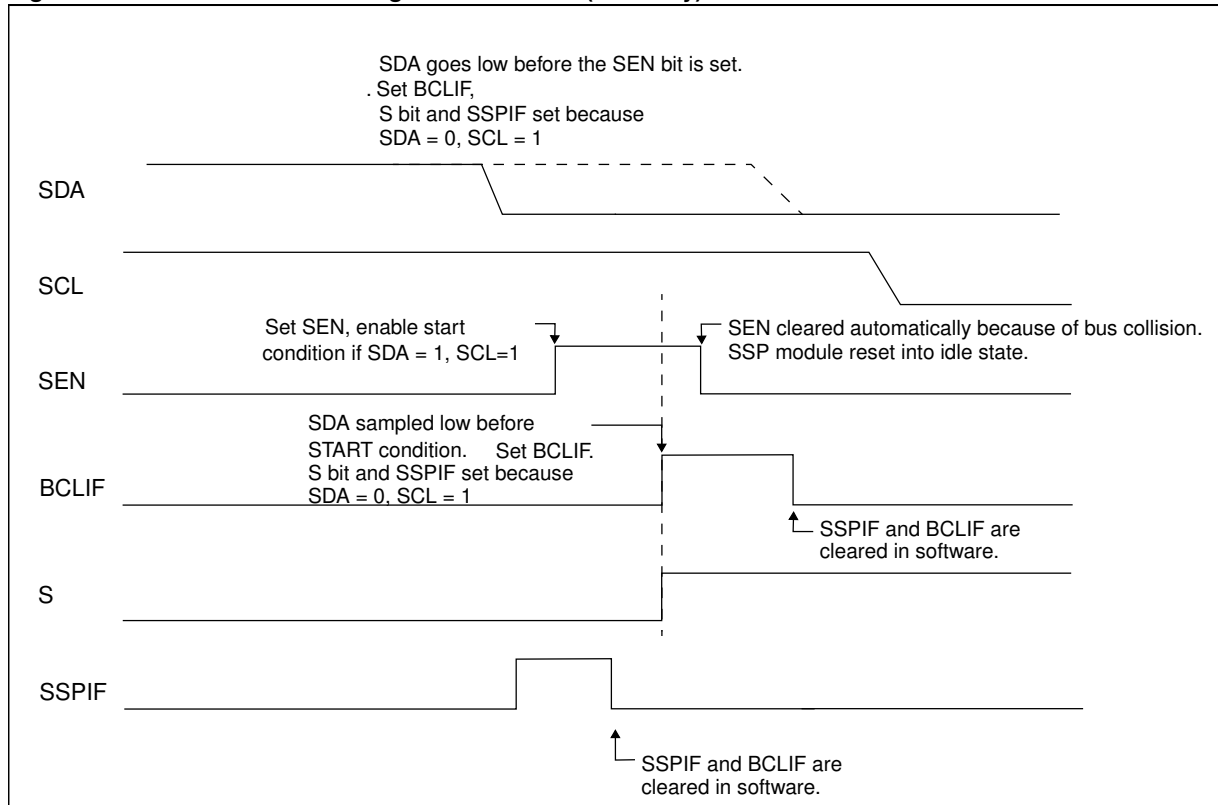
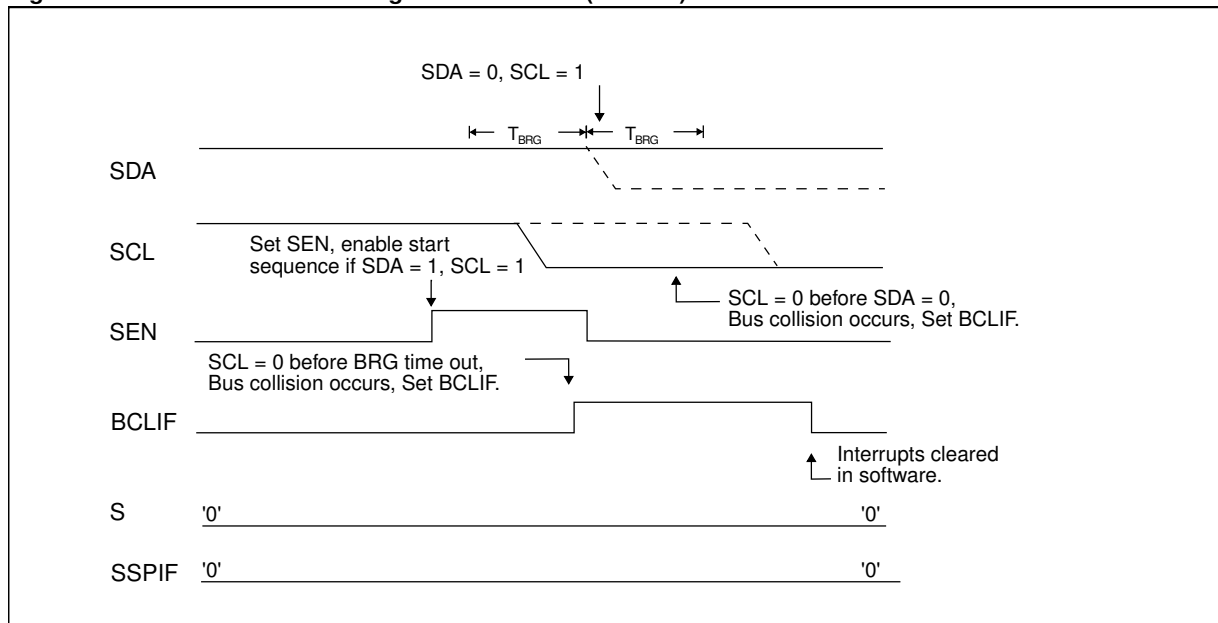


Figure 17-36: Bus Collision During Start Condition (SCL = 0)



17.4.18.2 Bus Collision During a Repeated Start Condition

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

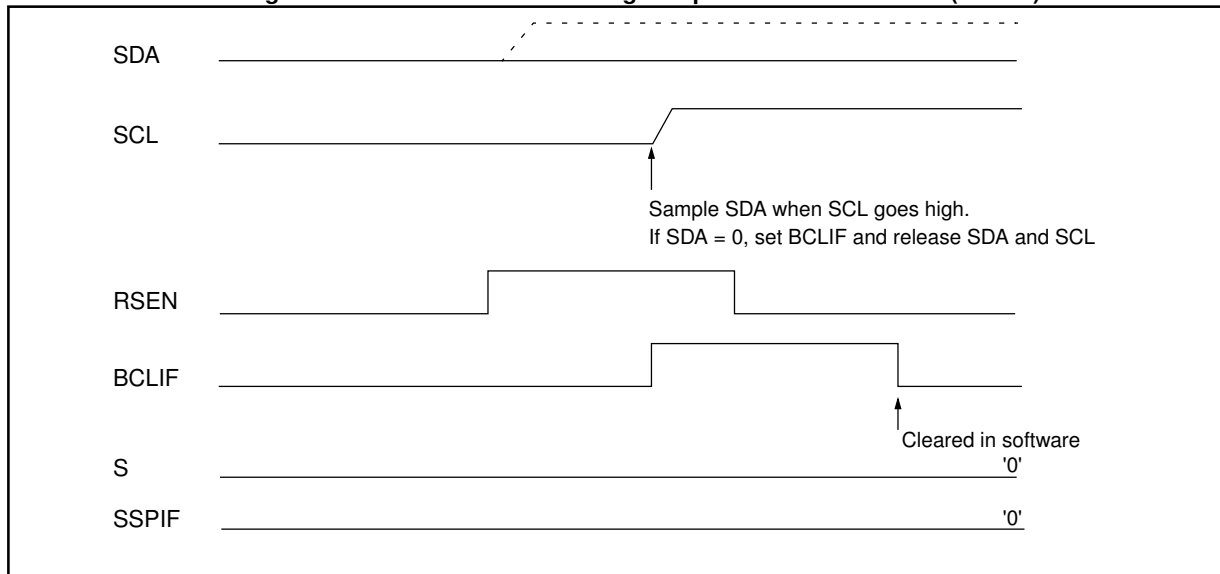
- A low level is sampled on SDA when SCL goes from low level to high level.
- SCL goes low before SDA is asserted low, indicating that another master is attempting to transmit a data '1'.

When the user de-asserts SDA and the pin is allowed to float high, the BRG is loaded with SSPADD<6:0>, and counts down to zero. The SCL pin is then de-asserted, and when sampled high, the SDA pin is sampled. If SDA is low, a bus collision has occurred (i.e. another master, [Figure 17-38](#), is attempting to transmit a data '0'). If, however, SDA is sampled high then the BRG is reloaded and begins counting. If SDA goes from high to low before the BRG times out, no bus collision occurs, because no two masters can assert SDA at exactly the same time.

If, however, SCL goes from high to low before the BRG times out and SDA has not already been asserted, then a bus collision occurs. In this case, another master is attempting to transmit a data '1' during the Repeated Start condition, [Figure 17-39](#).

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

Figure 17-38: Bus Collision During a Repeated Start Condition (Case 1)



17.4.18.3 Bus Collision During a STOP Condition

Bus collision occurs during a STOP condition if:

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

The STOP condition begins with SDA asserted low. When SDA is sampled low, the SCL pin is allowed to float. When the pin is sampled high (clock arbitration), the baud rate generator is loaded with SSPADD<6:0> and counts down to 0. After the BRG times out SDA is sampled. If SDA is sampled low, a bus collision has occurred. This is due to another master attempting to drive a data '0' (Figure 17-40). If the SCL pin is sampled low before SDA is allowed to float high, a bus collision occurs. This is another case of another master attempting to drive a data '0' (Figure 17-41).

Figure 17-40: Bus Collision During a STOP Condition (Case 1)

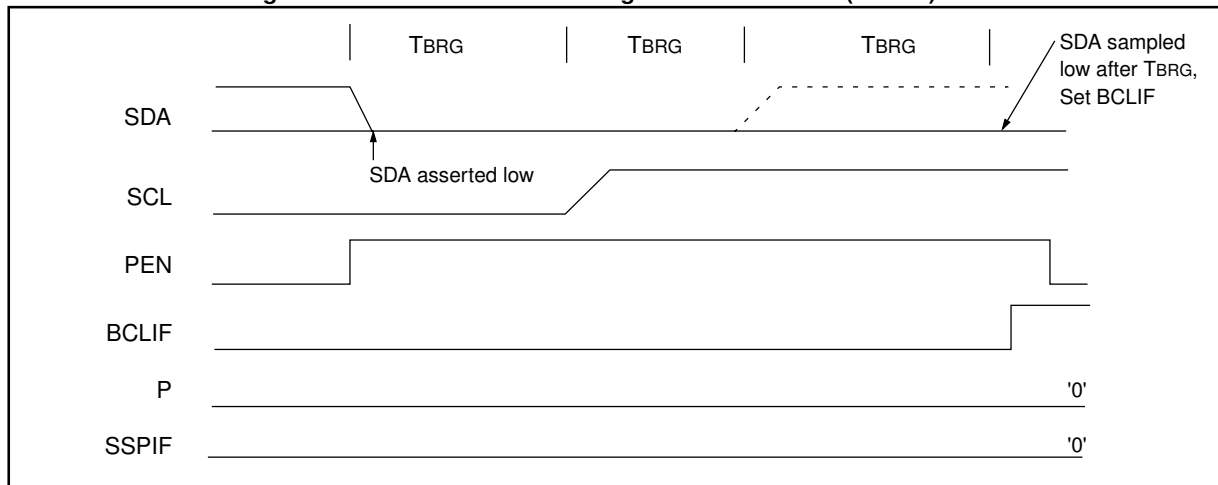
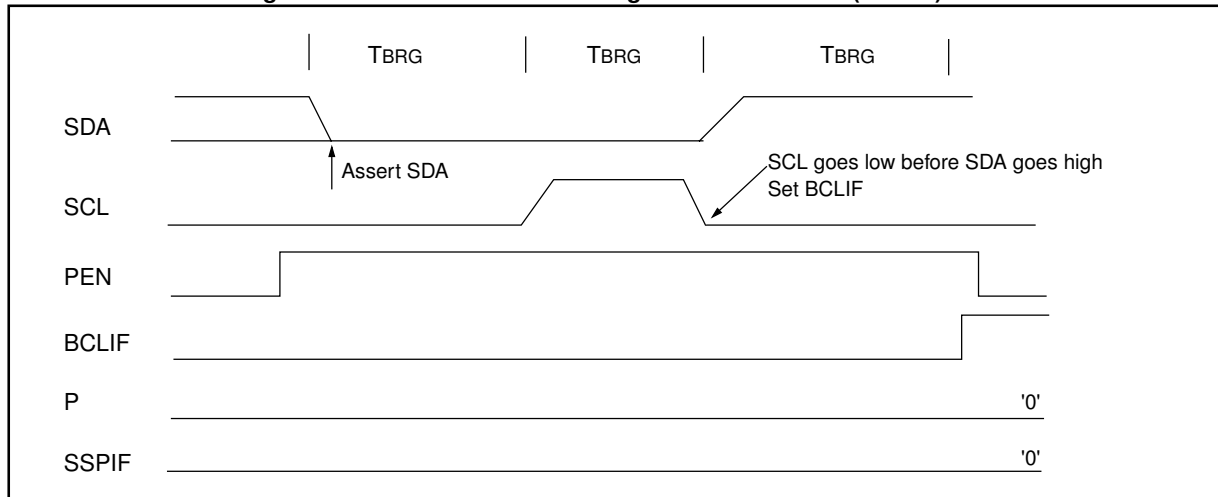


Figure 17-41: Bus Collision During a STOP Condition (Case 2)



17.6 Initialization

Example 17-2: SPI Master Mode Initialization

```

CLRf  STATUS      ; Bank 0
CLRf  SSPSTAT     ; SMP = 0, CKE = 0, and clear status bits
BSF   SSPSTAT, CKE ; CKE = 1
MOVLW 0x31        ; Set up SPI port, Master mode, CLK/16,
MOVWF SSPCON      ; Data xmit on falling edge (CKE=1 & CKP=1)
                ; Data sampled in middle (SMP=0 & Master mode)

BSF   STATUS, RP0 ; Bank 1
BSF   PIE, SSPIE  ; Enable SSP interrupt
BCF   STATUS, RP0 ; Bank 0
BSF   INTCON, GIE ; Enable, enabled interrupts
MOVLW DataByte    ; Data to be Transmitted
                ; Could move data from RAM location
MOVWF SSPBUF      ; Start Transmission
    
```

17.6.1 Master SSP Module / Basic SSP Module Compatibility

When changing from the SPI in the Basic SSP module, the SSPSTAT register contains two additional control bits. These bits are:

- SMP, SPI data input sample phase
- CKE, SPI Clock Edge Select

To be compatible with the SPI of the Master SSP module, these bits must be appropriately configured. If these bits are not at the states shown in [Table 17-4](#), improper SPI communication may occur.

Table 17-4: New bit States for Compatibility

Basic SSP Module	Master SSP Module		
CKP	CKP	CKE	SMP
1	1	0	0
0	0	0	0

17.8 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 Mid-Range MCU family (that is they may be written for the Base-Line, or High-End families), but the concepts are pertinent, and could be used (with modification and possible limitations). The current application notes related to the Master SSP module are:

Title	Application Note #
Use of the SSP Module in the I ² C Multi-Master Environment.	AN578
Using Microchip 93 Series Serial EEPROMs with Microcontroller SPI Ports	AN613
Interfacing PIC16C64/74 to Microchip SPI Serial EEPROM	AN647
Interfacing a Microchip PIC16C92x to Microchip SPI Serial EEPROM	AN668

Section 18. USART

HIGHLIGHTS

This section of the manual contains the following major topics:

18.1	Introduction	18-2
18.2	Control Registers	18-3
18.3	USART Baud Rate Generator (BRG).....	18-5
18.4	USART Asynchronous Mode	18-8
18.5	USART Synchronous Master Mode	18-15
18.6	USART Synchronous Slave Mode	18-19
18.7	Initialization	18-21
18.8	Design Tips	18-22
18.9	Related Application Notes.....	18-23
18.10	Revision History	18-24

Section 18. USART

18.2 Control Registers

Register 18-1: TXSTA: Transmit Status and Control Register

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D

bit 7

bit 0

- bit 7 **CSRC**: Clock Source Select bit
Asynchronous mode
Don't care
Synchronous mode
1 = Master mode (Clock generated internally from BRG)
0 = Slave mode (Clock from external source)
- bit 6 **TX9**: 9-bit Transmit Enable bit
1 = Selects 9-bit transmission
0 = Selects 8-bit transmission
- bit 5 **TXEN**: Transmit Enable bit
1 = Transmit enabled
0 = Transmit disabled
Note: SREN/CREN overrides TXEN in SYNC mode.
- bit 4 **SYNC**: USART Mode Select bit
1 = Synchronous mode
0 = Asynchronous mode
- bit 3 **Unimplemented**: Read as '0'
- bit 2 **BRGH**: High Baud Rate Select bit
Asynchronous mode
1 = High speed
0 = Low speed
Synchronous mode
Unused in this mode
- bit 1 **TRMT**: Transmit Shift Register Status bit
1 = TSR empty
0 = TSR full
- bit 0 **TX9D**: 9th bit of transmit data. Can be parity bit.

Legend

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

- n = Value at POR reset

Section 18. USART

18.3 USART Baud Rate Generator (BRG)

The BRG supports both the Asynchronous and Synchronous modes of the USART. It is a dedicated 8-bit baud rate generator. The SPBRG register controls the period of a free running 8-bit timer. In asynchronous mode bit BRGH (TXSTA<2>) also controls the baud rate. In synchronous mode bit BRGH is ignored. [Table 18-1](#) shows the formula for computation of the baud rate for different USART modes which only apply in master mode (internal clock).

Given the desired baud rate and Fosc, the nearest integer value for the SPBRG register can be calculated using the formula in [Table 18-1](#), where X equals the value in the SPBRG register (0 to 255). From this, the error in baud rate can be determined.

Table 18-1: Baud Rate Formula

SYNC	BRGH = 0 (Low Speed)	BRGH = 1 (High Speed)
0	(Asynchronous) Baud Rate = $F_{osc}/(64(X+1))$	Baud Rate = $F_{osc}/(16(X+1))$
1	(Synchronous) Baud Rate = $F_{osc}/(4(X+1))$	NA

X = value in SPBRG (0 to 255)

[Example 18-1](#) shows the calculation of the baud rate error for the following conditions:

Fosc = 16 MHz
 Desired Baud Rate = 9600
 BRGH = 0
 SYNC = 0

Example 18-1: Calculating Baud Rate Error

Desired Baud rate	=	$F_{osc} / (64 (X + 1))$
9600	=	$16000000 / (64 (X + 1))$
X	=	$\lfloor 25.042 \rfloor = 25$
Calculated Baud Rate	=	$16000000 / (64 (25 + 1))$
	=	9615
Error	=	$\frac{(\text{Calculated Baud Rate} - \text{Desired Baud Rate})}{\text{Desired Baud Rate}}$
	=	$(9615 - 9600) / 9600$
	=	0.16%

It may be advantageous to use the high baud rate (BRGH = 1) even for slower baud clocks. This is because the $F_{osc} / (16(X + 1))$ equation can reduce the baud rate error in some cases.

Writing a new value to the SPBRG register causes the BRG timer to be reset (or cleared). This ensures the BRG does not wait for a timer overflow before outputting the new baud rate.

Table 18-2: Registers Associated with Baud Rate Generator

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other resets
TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
RCSTA	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00x
SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, – = unimplemented read as '0'. Shaded cells are not used by the BRG.

Section 18. USART

Table 18-4: Baud Rates for Asynchronous Mode (BRGH = 0)

BAUD RATE (Kbps)	Fosc = 20 MHz			16 MHz			10 MHz			7.15909 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	NA	-	-	NA	-	-	NA	-	-	NA	-	-
1.2	1.221	+1.73	255	1.202	+0.16	207	1.202	+0.16	129	1.203	+0.23	92
2.4	2.404	+0.16	129	2.404	+0.16	103	2.404	+0.16	64	2.380	-0.83	46
9.6	9.469	-1.36	32	9.615	+0.16	25	9.766	+1.73	15	9.322	-2.90	11
19.2	19.53	+1.73	15	19.23	+0.16	12	19.53	+1.73	7	18.64	-2.90	5
76.8	78.13	+1.73	3	83.33	+8.51	2	78.13	+1.73	1	NA	-	-
96	104.2	+8.51	2	NA	-	-	NA	-	-	NA	-	-
300	312.5	+4.17	0	NA	-	-	NA	-	-	NA	-	-
500	NA	-	-	NA	-	-	NA	-	-	NA	-	-
HIGH	312.5	-	0	250	-	0	156.3	-	0	111.9	-	0
LOW	1.221	-	255	0.977	-	255	0.6104	-	255	0.437	-	255

BAUD RATE (Kbps)	Fosc = 5.0688 MHz			4 MHz			3.579545 MHz			1 MHz			32.768 kHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	0.31	+3.13	255	0.3005	-0.17	207	0.301	+0.23	185	0.300	+0.16	51	0.256	-14.67	1
1.2	1.2	0	65	1.202	+1.67	51	1.190	-0.83	46	1.202	+0.16	12	NA	-	-
2.4	2.4	0	32	2.404	+1.67	25	2.432	+1.32	22	2.232	-6.99	6	NA	-	-
9.6	9.9	+3.13	7	NA	-	-	9.322	-2.90	5	NA	-	-	NA	-	-
19.2	19.8	+3.13	3	NA	-	-	18.64	-2.90	2	NA	-	-	NA	-	-
76.8	79.2	+3.13	0	NA	-	-	NA	-	-	NA	-	-	NA	-	-
96	NA	-	-	NA	-	-	NA	-	-	NA	-	-	NA	-	-
300	NA	-	-	NA	-	-	NA	-	-	NA	-	-	NA	-	-
500	NA	-	-	NA	-	-	NA	-	-	NA	-	-	NA	-	-
HIGH	79.2	-	0	62.500	-	0	55.93	-	0	15.63	-	0	0.512	-	0
LOW	0.3094	-	255	3.906	-	255	0.2185	-	255	0.0610	-	255	0.0020	-	255

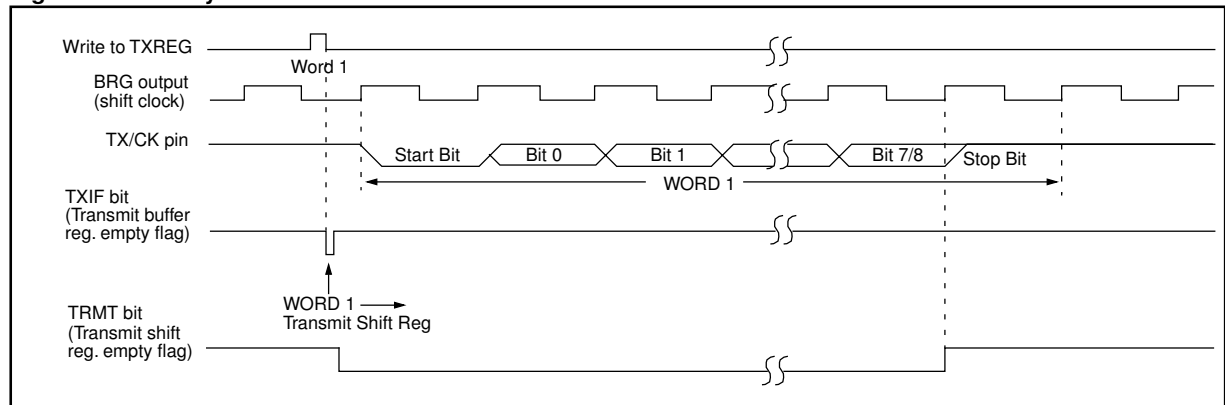
Table 18-5: Baud Rates for Asynchronous Mode (BRGH = 1)

BAUD RATE (Kbps)	Fosc = 20 MHz			16 MHz			10 MHz			7.15909 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
9.6	9.615	+0.16	129	9.615	+0.16	103	9.615	+0.16	64	9.520	-0.83	46
19.2	19.230	+0.16	64	19.230	+0.16	51	18.939	-1.36	32	19.454	+1.32	22
38.4	37.878	-1.36	32	38.461	+0.16	25	39.062	+1.7	15	37.286	-2.90	11
57.6	56.818	-1.36	21	58.823	+2.12	16	56.818	-1.36	10	55.930	-2.90	7
115.2	113.636	-1.36	10	111.111	-3.55	8	125	+8.51	4	111.860	-2.90	3
250	250	0	4	250	0	3	NA	-	-	NA	-	-
625	625	0	1	NA	-	-	625	0	0	NA	-	-
1250	1250	0	0	NA	-	-	NA	-	-	NA	-	-

BAUD RATE (Kbps)	Fosc = 5.0688 MHz			4 MHz			3.579545 MHz			1 MHz			32.768 kHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
9.6	9.6	0	32	NA	-	-	9.727	+1.32	22	8.928	-6.99	6	NA	-	-
19.2	18.645	-2.94	16	1.202	+0.17	207	18.643	-2.90	11	20.833	+8.51	2	NA	-	-
38.4	39.6	+3.12	7	2.403	+0.13	103	37.286	-2.90	5	31.25	-18.61	1	NA	-	-
57.6	52.8	-8.33	5	9.615	+0.16	25	55.930	-2.90	3	62.5	+8.51	0	NA	-	-
115.2	105.6	-8.33	2	19.231	+0.16	12	111.860	-2.90	1	NA	-	-	NA	-	-
250	NA	-	-	NA	-	-	223.721	-10.51	0	NA	-	-	NA	-	-
625	NA	-	-	NA	-	-	NA	-	-	NA	-	-	NA	-	-
1250	NA	-	-	NA	-	-	NA	-	-	NA	-	-	NA	-	-

USART

Figure 18-2: Asynchronous Master Transmission



1. Initialize the SPBRG register for the appropriate baud rate. If a high speed baud rate is desired, set the BRGH bit. (Subsection **18.3 “USART Baud Rate Generator (BRG)”**)
2. Enable the asynchronous serial port by clearing the SYNC bit and setting the SPEN bit.
3. If interrupts are desired, then set the TXIE, GIE and PEIE bits.
4. If 9-bit transmission is desired, then set the TX9 bit.
5. Enable the transmission by setting the TXEN bit, which will also set the TXIF bit.
6. If 9-bit transmission is selected, the ninth bit should be loaded in the TX9D bit.
7. Load data to the TXREG register (starts transmission).

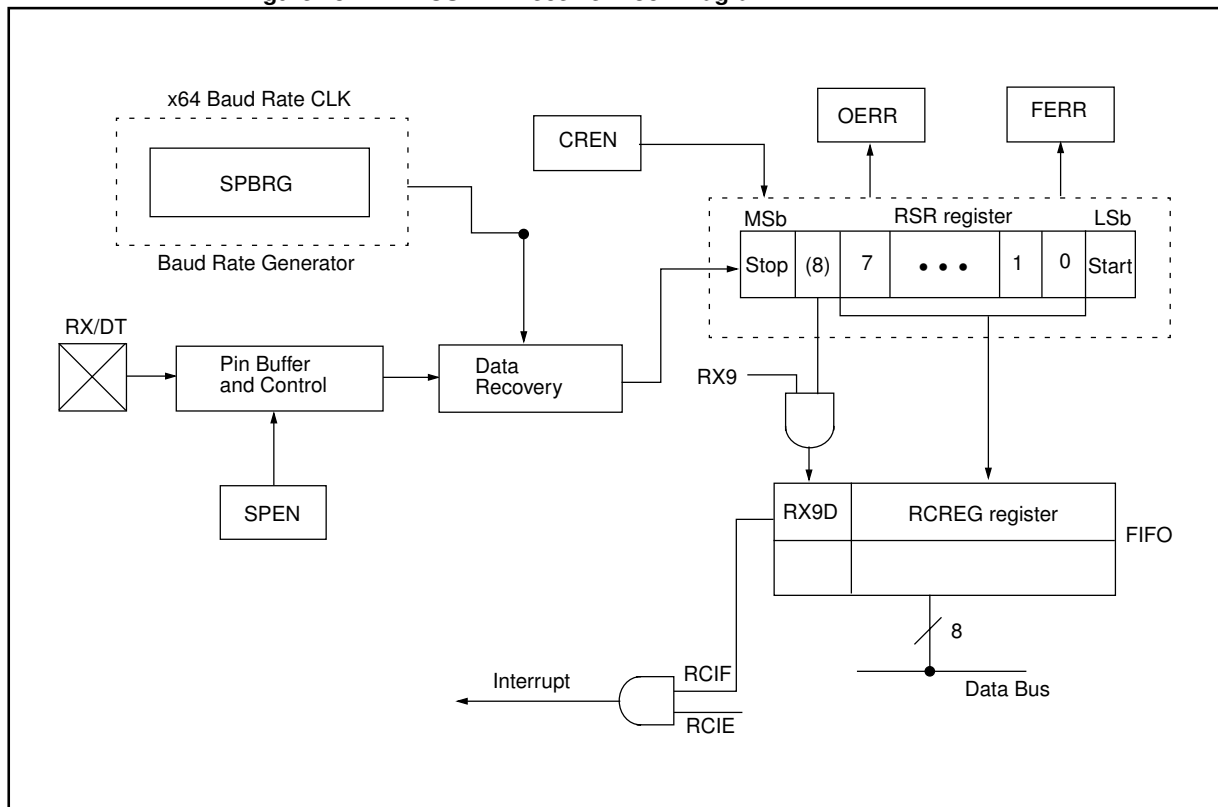
18.4.2 USART Asynchronous Receiver

The receiver block diagram is shown in Figure 18-4. The data is received on the RX/DT pin and drives the data recovery block. The data recovery block is actually a high speed shifter operating at x16 times the baud rate, whereas the main receive serial shifter operates at the bit rate or at FOSC.

Once Asynchronous mode is selected, reception is enabled by setting the CREN bit (RCSTA<4>).

The heart of the receiver is the receive (serial) shift register (RSR). After sampling the RX/TX pin for the STOP bit, the received data in the RSR is transferred to the RCREG register (if it is empty). If the transfer is complete, the RCIF flag bit is set. The actual interrupt can be enabled/disabled by setting/clearing the RCIE enable bit. The RCIF flag bit is a read only bit which is cleared by the hardware. It is cleared when the RCREG register has been read and is empty. The RCREG is a double buffered register, i.e. it is a two deep FIFO. It is possible for two bytes of data to be received and transferred to the RCREG FIFO and a third byte begin shifting to the RSR register. On the detection of the STOP bit of the third byte, if the RCREG register is still full then overrun error bit, OERR (RCSTA<1>), will be set. The word in the RSR will be lost. The RCREG register can be read twice to retrieve the two bytes in the FIFO. The OERR bit has to be cleared in software. This is done by resetting the receive logic (the CREN bit is cleared and then set). If the OERR bit is set, transfers from the RSR register to the RCREG register are inhibited, so it is essential to clear the OERR bit if it is set. Framing error bit, FERR (RCSTA<2>), is set if a stop bit is detected as a low level. The FERR bit and the 9th receive bit are buffered the same way as the receive data. Reading the RCREG will load the RX9D and FERR bits with new values, therefore it is essential for the user to read the RCSTA register before reading the next RCREG register in order not to lose the old (previous) information in the FERR and RX9D bits.

Figure 18-4: USART Receive Block Diagram

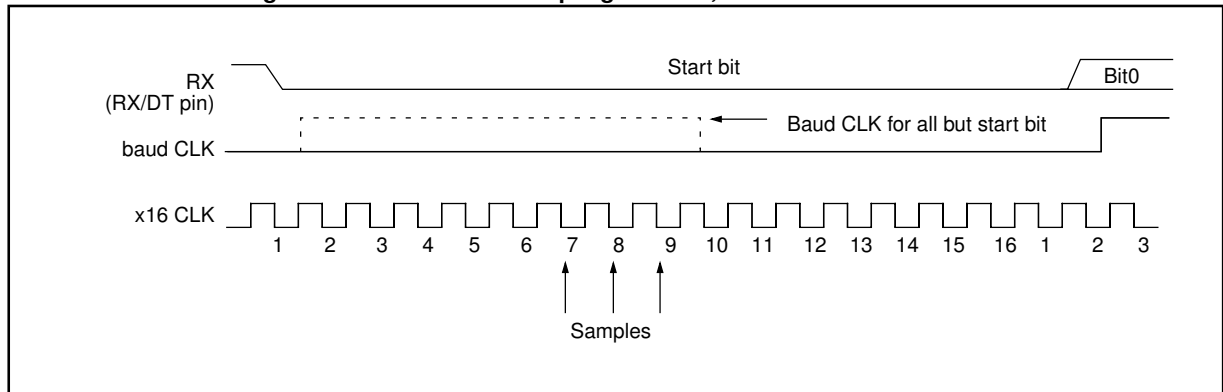


Section 18. USART

18.4.3 Sampling

The data on the RX/DT pin is sampled three times by a majority detect circuit to determine if a high or a low level is present at the RX pin. Figure 18-6 shows the waveform for the sampling circuit. The sampling operates the same regardless of the state of the BRGH bit, only the source of the x16 clock is different.

Figure 18-6: RX Pin Sampling Scheme, BRGH = 0 or BRGH = 1



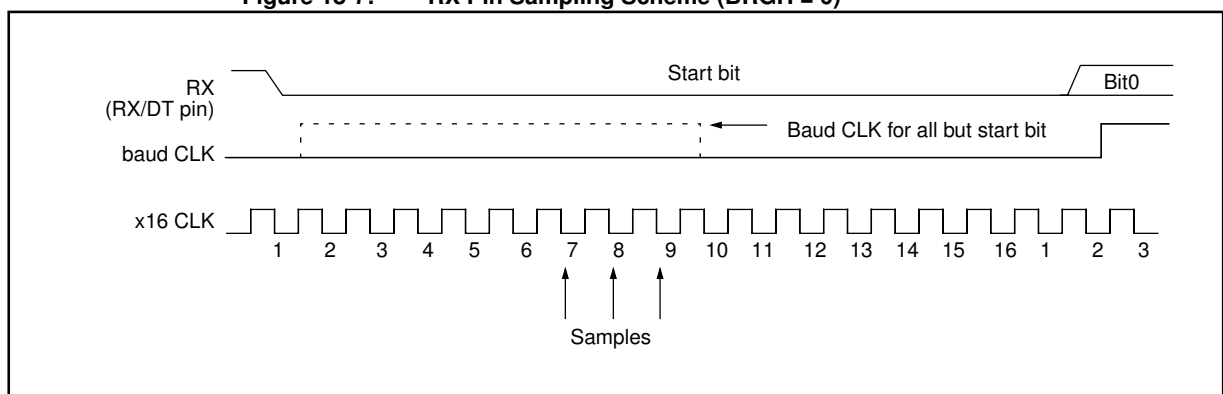
18.4.3.1 Device Exceptions

All new devices will use the sampling scheme shown in Figure 18-6. Devices that have an exception to the above sampling scheme are:

- PIC16C63
- PIC16C65
- PIC16C65A
- PIC16C73
- PIC16C73A
- PIC16C74
- PIC16C74A

These devices have a sampling circuitry that works as follows. If the BRGH bit (TXSTA<2>) is clear (i.e., at the low baud rates), the sampling is done on the seventh, eighth and ninth falling edges of a x16 clock (Figure 18-7). If bit BRGH is set (i.e., at the high baud rates), the sampling is done on the 3 clock edges preceding the second rising edge after the first falling edge of a x4 clock (Figure 18-8 and Figure 18-9).

Figure 18-7: RX Pin Sampling Scheme (BRGH = 0)



18.5 USART Synchronous Master Mode

In Synchronous Master mode, the data is transmitted in a half-duplex manner, i.e. transmission and reception do not occur at the same time. When transmitting data, the reception is inhibited and vice versa. Synchronous mode is entered by setting the SYNC bit (TXSTA<4>). In addition, the SPEN enable bit (RCSTA<7>) is set in order to configure the TX/CK and RX/DT I/O pins to CK (clock) and DT (data) lines respectively. The Master mode indicates that the processor transmits the master clock on the CK line. The Master mode is entered by setting the CSRC bit (TXSTA<7>).

18.5.1 USART Synchronous Master Transmission

The USART transmitter block diagram is shown in [Figure 18-1](#). The heart of the transmitter is the transmit (serial) shift register (TSR). The shift register obtains its data from the read/write transmit buffer register TXREG. The TXREG register is loaded with data in software. The TSR register is not loaded until the last bit has been transmitted from the previous load. As soon as the last bit is transmitted, the TSR is loaded with new data from the TXREG (if available). Once the TXREG register transfers the data to the TSR register (occurs in one Tcycle), the TXREG is empty and the TXIF interrupt flag bit is set. The interrupt can be enabled/disabled by setting/clearing enable the TXIE bit. The TXIF flag bit will be set regardless of the state of the TXIE enable bit and cannot be cleared in software. It will reset only when new data is loaded into the TXREG register. While the TXIF flag bit indicates the status of the TXREG register, the TRMT bit (TXSTA<1>) shows the status of the TSR register. The TRMT bit is a read only bit which is set when the TSR is empty. No interrupt logic is tied to this bit, so the user has to poll this bit in order to determine if the TSR register is empty. The TSR is not mapped in data memory so it is not available to the user.

Transmission is enabled by setting the TXEN bit (TXSTA<5>). The actual transmission will not occur until the TXREG register has been loaded with data. The first data bit will be shifted out on the next available rising edge of the clock on the CK line. Data out is stable at the falling edge of the synchronous clock ([Figure 18-10](#)). The transmission can also be started by first loading the TXREG register and then setting the TXEN bit. This is advantageous when slow baud rates are selected, since the BRG is kept in reset when the TXEN, CREN, and SREN bits are clear. Setting the TXEN bit will start the BRG, creating a shift clock immediately. Normally when transmission is first started, the TSR register is empty, so a transfer to the TXREG register will result in an immediate transfer to TSR resulting in an empty TXREG. Back-to-back transfers are possible.

Clearing the TXEN bit during a transmission will cause the transmission to be aborted and will reset the transmitter. The DT and CK pins will revert to hi-impedance. If either of the CREN or SREN bits are set during a transmission, the transmission is aborted and the DT pin reverts to a hi-impedance state (for a reception). The CK pin will remain an output if the CSRC bit is set (internal clock). The transmitter logic is not reset although it is disconnected from the pins. In order to reset the transmitter, the user has to clear the TXEN bit. If the SREN bit is set (to interrupt an on-going transmission and receive a single word), then after the single word is received, the SREN bit will be cleared and the serial port will revert back to transmitting since the TXEN bit is still set. The DT line will immediately switch from hi-impedance receive mode to transmit and start driving. To avoid this the TXEN bit should be cleared.

In order to select 9-bit transmission, the TX9 bit (TXSTA<6>) should be set and the ninth bit should be written to the TX9D bit (TXSTA<0>). The ninth bit must be written before writing the 8-bit data to the TXREG register. This is because a data write to the TXREG can result in an immediate transfer of the data to the TSR register (if the TSR is empty). If the TSR was empty and the TXREG was written before writing the "new" value to the TX9D bit, the "present" value of the TX9D bit is loaded.

18.5.2 USART Synchronous Master Reception

Once Synchronous mode is selected, reception is enabled by setting either of the SREN (RCSTA<5>) or CREN (RCSTA<4>) bits. Data is sampled on the RX/DT pin on the falling edge of the clock. If the SREN bit is set, then only a single word is received. If the CREN bit is set, the reception is continuous until the CREN bit is cleared. If both bits are set then the CREN bit takes precedence. After clocking the last serial data bit, the received data in the Receive Shift Register (RSR) is transferred to the RCREG register (if it is empty). When the transfer is complete, the RCIF interrupt flag bit is set. The actual interrupt can be enabled/disabled by setting/clearing the RCIE enable bit. The RCIF flag bit is a read only bit which is cleared by the hardware. In this case it is cleared when the RCREG register has been read and is empty. The RCREG is a double buffered register, i.e. it is a two deep FIFO. It is possible for two bytes of data to be received and transferred to the RCREG FIFO and a third byte to begin shifting into the RSR register. On the clocking of the last bit of the third byte, if the RCREG register is still full then overrun error bit, OERR (RCSTA<1>), is set and the word in the RSR is lost. The RCREG register can be read twice to retrieve the two bytes in the FIFO. The OERR bit has to be cleared in software (by clearing the CREN bit). If the OERR bit is set, transfers from the RSR to the RCREG are inhibited, so it is essential to clear the OERR bit if it is set. The 9th receive bit is buffered the same way as the receive data. Reading the RCREG register will load the RX9D bit with a new value, therefore it is essential for the user to read the RCSTA register before reading RCREG in order not to lose the old (previous) information in the RX9D bit.

Steps to follow when setting up a Synchronous Master Reception:

1. Initialize the SPBRG register for the appropriate baud rate. (Subsection [18.3 “USART Baud Rate Generator \(BRG\)”](#))
2. Enable the synchronous master serial port by setting the SYNC, SPEN, and CSRC bits.
3. Ensure that the CREN and SREN bits are clear.
4. If interrupts are desired, then set the RCIE bit.
5. If 9-bit reception is desired, then set the RX9 bit.
6. If a single reception is required, set the SREN bit. For continuous reception set the CREN bit.
7. The RCIF bit will be set when reception is complete and an interrupt will be generated if the RCIE bit was set.
8. Read the RCSTA register to get the ninth bit (if enabled) and determine if any error occurred during reception.
9. Read the 8-bit received data by reading the RCREG register.
10. If any error occurred, clear the error by clearing the CREN bit.

Table 18-9: Registers Associated with Synchronous Master Reception

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
PIR	RCIF ⁽¹⁾								0	0
RCSTA	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00x
RCREG	RX7	RX6	RX5	RX4	RX3	RX2	RX1	RX0	0000 0000	0000 0000
PIE	RCIE ⁽¹⁾								0	0
TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented read as '0'.

Shaded cells are not used for Synchronous Master Reception.

Note 1: The position of this bit is device dependent.

18.6 USART Synchronous Slave Mode

Synchronous slave mode differs from the Master mode in the fact that the shift clock is supplied externally at the TX/CK pin (instead of being supplied internally in master mode). This allows the device to transfer or receive data while in SLEEP mode. Slave mode is entered by clearing the CSRC bit (TXSTA<7>).

18.6.1 USART Synchronous Slave Transmit

The operation of the synchronous master and slave modes are identical except in the case of the SLEEP mode.

If two words are written to the TXREG and then the SLEEP instruction is executed, the following will occur:

- The first word will immediately transfer to the TSR register and transmit.
- The second word will remain in TXREG register.
- The TXIF flag bit will not be set.
- When the first word has been shifted out of TSR, the TXREG register will transfer the second word to the TSR and the TXIF flag bit will now be set.
- If the TXIE enable bit is set, the interrupt will wake the chip from SLEEP and if the global interrupt is enabled, the program will branch to the interrupt vector (0004h).

Steps to follow when setting up a Synchronous Slave Transmission:

- Enable the synchronous slave serial port by setting the SYNC and SPEN bits and clearing the CSRC bit.
- Clear the CREN and SREN bits.
- If interrupts are desired, then set the TXIE enable bit.
- If 9-bit transmission is desired, then set the TX9 bit.
- Enable the transmission by setting the TXEN enable bit.
- If 9-bit transmission is selected, the ninth bit should be loaded into the TX9D bit.
- Start transmission by loading data to the TXREG register.

Table 18-10: Registers Associated with Synchronous Slave Transmission

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
PIR	TXIF ⁽¹⁾								0	0
RCSTA	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00x
TXREG	TX7	TX6	TX5	TX4	TX3	TX2	TX1	TX0	0000 0000	0000 0000
PIE	TXIE ⁽¹⁾								0	0
TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented read as '0'.

Shaded cells are not used for Synchronous Slave Transmission.

Note 1: The position of this bit is device dependent.

18.7 Initialization

[Example 18-2](#) is an initialization routine for asynchronous Transmitter/Receiver mode. [Example 18-3](#) is for the synchronous mode. In both examples the data is 8-bits, and the value to load into the SPBRG register is dependent on the desired baud rate and the device frequency.

Example 18-2: Asynchronous Transmitter/Receiver

```
BSF    STATUS,RP0    ; Go to Bank1
MOVLW  <baudrate>    ; Set Baud rate
MOVWF  SPBRG
MOVLW  0x40           ; 8-bit transmit, transmitter enabled,
MOVWF  TXSTA          ; asynchronous mode, low speed mode
BSF    PIE1,TXIE      ; Enable transmit interrupts
BSF    PIE1,RCIE      ; Enable receive interrupts
BCF    STATUS,RP0    ; Go to Bank 0
MOVLW  0x90           ; 8-bit receive, receiver enabled,
MOVWF  RCSTA          ; serial port enabled
```

Example 18-3: Synchronous Transmitter/Receiver

```
BSF    STATUS,RP0    ; Go to Bank 1
MOVLW  <baudrate>    ; Set Baud Rate
MOVWF  SPBRG
MOVLW  0xB0           ; Synchronous Master, 8-bit transmit,
MOVWF  TXSTA          ; transmitter enabled, low speed mode
BSF    PIE1,TXIE      ; Enable transmit interrupts
BSF    PIE1,RCIE      ; Enable receive interrupts
BCF    STATUS,RP0    ; Go to Bank 0
MOVLW  0x90           ; 8-bit receive, receiver enabled,
MOVWF  RCSTA          ; continuous receive, serial port enabled
```

Section 18. USART

18.9 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 Mid-Range MCU family (that is they may be written for the Base-Line, or High-End families), but the concepts are pertinent, and could be used (with modification and possible limitations). The current application notes related to this section are:

Title	Application Note #
Serial Port Utilities	AN547
Servo Control of a DC Brushless Motor	AN543

Section 19. Voltage Reference

HIGHLIGHTS

This section of the manual contains the following major topics:

19.1	Introduction	19-2
19.2	Control Register	19-3
19.3	Configuring the Voltage Reference	19-4
19.4	Voltage Reference Accuracy/Error	19-5
19.5	Operation During Sleep	19-5
19.6	Effects of a Reset.....	19-5
19.7	Connection Considerations	19-6
19.8	Initialization	19-7
19.9	Design Tips	19-8
19.10	Related Application Notes.....	19-9
19.11	Revision History	19-10

Section 19. Voltage Reference

19.2 Control Register

Register 19-1: VRCON Register

R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
VREN	VROE	VRR	—	VR3	VR2	VR1	VR0
bit 7				bit 0			

bit 7 **VREN:** VREF Enable

1 = VREF circuit powered on

0 = VREF circuit powered down

bit 6 **VROE:** VREF Output Enable

1 = VREF is internally connected to Comparator module's VREF. This voltage level is also output on the VREF pin

0 = VREF is not connected to the comparator module. This voltage is disconnected from the VREF pin

bit 5 **VRR:** VREF Range selection

1 = 0V to 0.75 VDD, with VDD/24 step size

0 = 0.25 VDD to 0.75 VDD, with VDD/32 step size

bit 4 **Unimplemented:** Read as '0'

bit 3:0 **VR3:VR0:** VREF value selection $0 \leq VR3:VR0 \leq 15$

When VRR = 1:

$$VREF = (VR<3:0>/24) \cdot VDD$$

When VRR = 0:

$$VREF = 1/4 \cdot VDD + (VR3:VR0/32) \cdot VDD$$

Legend

R = Readable bit W = Writable bit

U = Unimplemented bit, read as '0'

- n = Value at POR reset

Section 19. Voltage Reference

19.4 Voltage Reference Accuracy/Error

The full range of VSS to VDD cannot be realized due to the construction of the module. The transistors on the top and bottom of the resistor ladder network ([Figure 19-1](#)) keep VREF from approaching VSS or VDD. The Voltage Reference is VDD derived and therefore, the VREF output changes with fluctuations in VDD. The absolute accuracy of the Voltage Reference can be found in the Device Data Sheets electrical specification [parameter D311](#).

19.5 Operation During Sleep

When the device wakes up from sleep through an interrupt or a Watchdog Timer time-out, the contents of the VRCON register are not affected. To minimize current consumption in SLEEP mode, the Voltage Reference should be disabled.

19.6 Effects of a Reset

A device reset disables the Voltage Reference by clearing the VREN bit (VRCON<7>). This reset also disconnects the reference from the VREF pin by clearing the VROE bit (VRCON<6>) and selects the high voltage range by clearing the VRR bit (VRCON<5>). The VREF value select bits, VRCON<3:0>, are also cleared.

Section 19. Voltage Reference

19.8 Initialization

[Example 19-1](#) shows the steps to configure the voltage reference module.

Example 19-1: Voltage Reference Configuration

MOVLW	0x02	; 4 Inputs Muxed to 2 comparators
MOVWF	CMCON	;
BSF	STATUS,RP0	; go to Bank1
MOVLW	0x07	; RA3:RA0 are outputs
MOVWF	TRISA	; outputs
MOVLW	0xA6	; enable VREF
MOVWF	VRCON	; low range set VR3:VR0 = 6
BCF	STATUS,RP0	; go to Bank0
CALL	DELAY10	; 10 μ s delay

Section 19. Voltage Reference

19.10 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 Mid-Range MCU family (that is they may be written for the Base-Line, or High-End families), but the concepts are pertinent, and could be used (with modification and possible limitations). The current application notes related to Voltage Reference are:

Title	Application Note #
Resistance and Capacitance Meter using a PIC16C622	AN611

Section 20. Comparator

HIGHLIGHTS

This section of the manual contains the following major topics:

20.1	Introduction	20-2
20.2	Control Register	20-3
20.3	Comparator Configuration.....	20-4
20.4	Comparator Operation	20-6
20.5	Comparator Reference.....	20-6
20.6	Comparator Response Time	20-8
20.7	Comparator Outputs	20-8
20.8	Comparator Interrupts.....	20-9
20.9	Comparator Operation During SLEEP	20-9
20.10	Effects of a RESET	20-9
20.11	Analog Input Connection Considerations.....	20-10
20.12	Initialization	20-11
20.13	Design Tips	20-12
20.14	Related Application Notes.....	20-13
20.15	Revision History	20-14

Section 20. Comparator

20.2 Control Register

Register 20-1: CMCON Register

R-0	R-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
C2OUT	C1OUT	—	—	CIS	CM2	CM1	CM0
bit 7				bit 0			

bit 7 **C2OUT**: Comparator2 Output Indicator bit
1 = C2 VIN+ > C2 VIN–
0 = C2 VIN+ < C2 VIN–

bit 6 **C1OUT**: Comparator1 Output Indicator bit
1 = C1 VIN+ > C1 VIN–
0 = C1 VIN+ < C1 VIN–

bit 5:4 **Unimplemented**: Read as '0'

bit 3 **CIS**: Comparator Input Switch bit

When CM2:CM0 = 001:

1 = C1 VIN– connects to AN3
0 = C1 VIN– connects to AN0

When CM2:CM0 = 010:

1 = C1 VIN– connects to AN3
C2 VIN– connects to AN2
0 = C1 VIN– connects to AN0
C2 VIN– connects to AN1

bit 2:0 **CM2:CM0**: Comparator Mode Select bits
See [Figure 20-1](#).

Legend

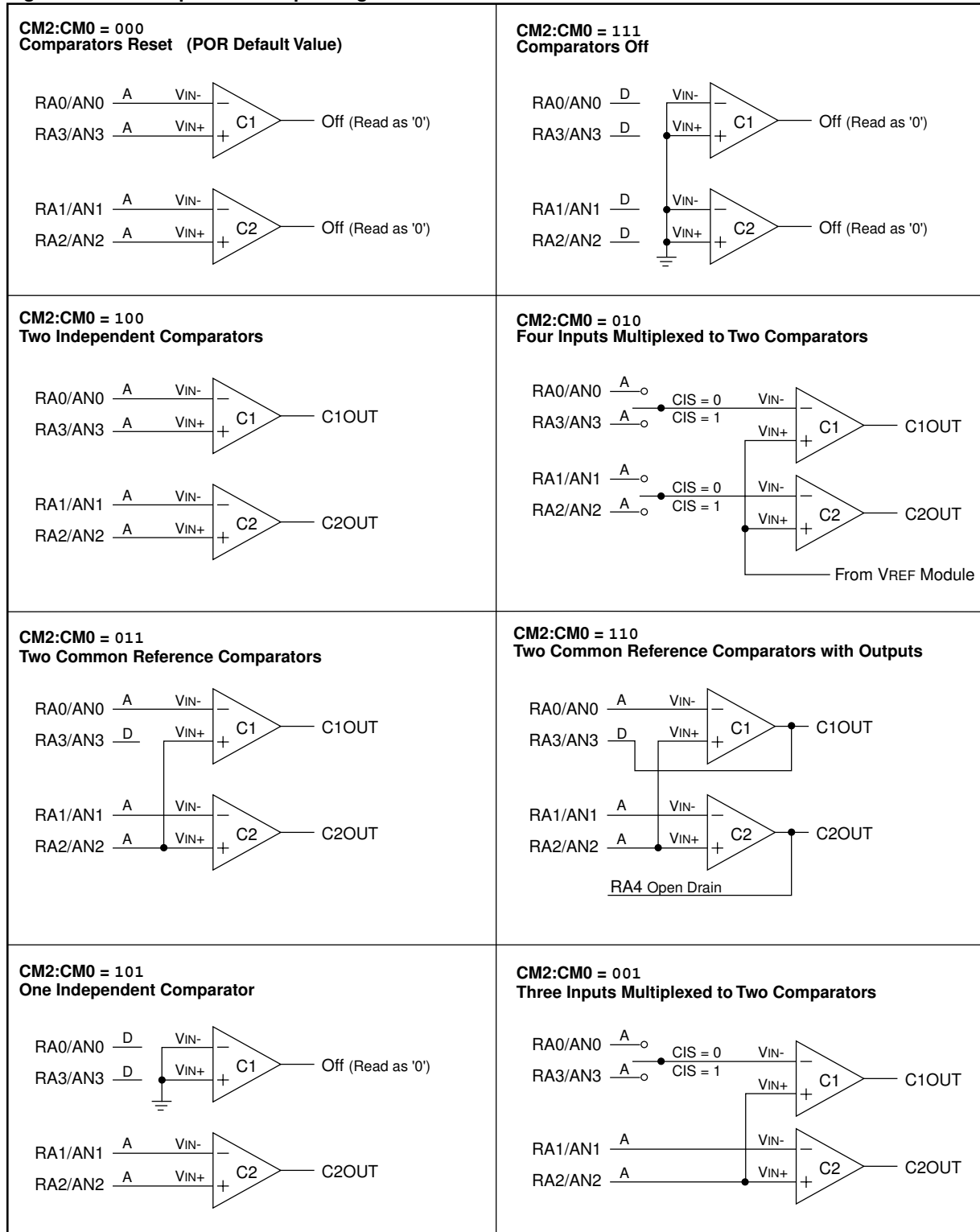
R = Readable bit W = Writable bit

U = Unimplemented bit, read as '0'

- n = Value at POR reset

Section 20. Comparator

Figure 20-1: Comparator I/O Operating Modes



A = Analog Input, port reads as zeros always.
D = Digital Input.
CIS (CMCON<3>) is the Comparator Input Switch.

Section 20. Comparator

20.5.1 External Reference Signal

When external voltage references are used, the comparator module can be configured to have the comparators operate from the same or different reference sources. The reference signal must be between V_{SS} and V_{DD} , and can be applied to either pin of the comparator(s).

20.5.2 Internal Reference Signal

The comparator module also allows the selection of an internally generated voltage reference for the comparators. The “[Voltage Reference](#)” section contains a detailed description of the Voltage Reference Module that provides this signal. The internal reference signal is used when the comparators are in mode $CM2:CM0 = 010$ ([Figure 20-1](#)). In this mode, the internal voltage reference is applied to the V_{IN+} input of both comparators.

The internal voltage reference may be used in any comparator mode. When used in this fashion the I/O/VREF pin may be used for I/O. The voltage reference is connected to the VREF pin.

Section 20. Comparator

20.8 Comparator Interrupts

The comparator interrupt flag is set whenever the comparators value changes relative to the last value loaded into CMxOUT bits. Software will need to maintain information about the status of the output bits, as read from CMCON<7:6>, to determine the actual change that has occurred. The CMIF bit, is the comparator interrupt flag. The CMIF bit must be cleared. Since it is also possible to set this bit, a simulated interrupt may be initiated.

The CMIE bit and the PEIE bit (INTCON<6>) must be set to enable the interrupt. In addition, the GIE bit must also be set. If any of these bits are clear, the interrupt is not enabled, though the CMIF bit will still be set if an interrupt condition occurs.

The user, in the interrupt service routine, can clear the interrupt in the following manner:

- a) Any read or write of the CMCON register. This will load the CMCON register with the new value with the CMxOUT bits.
- b) Clear the CMIF flag bit.

An interrupt condition will continue to set the CMIF flag bit. Reading CMCON will end the interrupt condition, and allow the CMIF flag bit to be cleared.

20.9 Comparator Operation During SLEEP

When a comparator is active and the device is placed in SLEEP mode, the comparator remains active and the interrupt is functional if enabled. This interrupt will wake up the device from SLEEP mode when enabled. While the comparator is powered-up, each comparator that is operational will consume additional current as shown in the comparator specifications. To minimize power consumption while in SLEEP mode, turn off the comparators, CM2:CM0 = 111, before entering sleep. If the device wakes-up from sleep, the contents of the CMCON register are not affected.

20.10 Effects of a RESET

A device reset forces the CMCON register to its reset state. This forces the comparator module to be in the comparator reset mode, CM2:CM0 = 000. This ensures that all potential inputs are analog inputs. Device current is minimized when analog inputs are present at reset time. The comparators will be powered-down during the reset interval.

Section 20. Comparator

20.12 Initialization

The code in [Example 20-1](#) depicts example steps required to configure the comparator module of the PIC16C62X devices. RA3 and RA4 are configured as digital output. RA0 and RA1 are configured as the V- inputs and RA2 as the V+ input to both comparators.

Example 20-1: Initializing Comparator Module (PIC16C62X)

```
FLAG_REG EQU 0X20
;
CLRF FLAG_REG ; Init flag register
CLRF PORTA ; Init PORTA
ANDLW 0xC0 ; Mask comparator bits
IORWF FLAG_REG,F ; Store bits in flag register
MOVLW 0x03 ; Init comparator mode
MOVWF CMCON ; CM<2:0> = 011
BSF STATUS,RP0 ; Select Bank1
MOVLW 0x07 ; Initialize data direction
MOVWF TRISA ; Set RA<2:0> as inputs, RA<4:3> as outputs,
; TRISA<7:5> always read '0'
BCF STATUS,RP0 ; Select Bank0
CALL DELAY 10 ; 10µs delay
MOVF CMCON,F ; Read CMCON to end change condition
BCF PIR1,CMIF ; Clear pending interrupts
BSF STATUS,RP0 ; Select Bank1
BSF PIE1,CMIE ; Enable comparator interrupts
BCF STATUS,RP0 ; Select Bank0
BSF INTCON,PEIE ; Enable peripheral interrupts
BSF INTCON,GIE ; Global interrupt enable
```

Section 20. Comparator

20.14 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 Mid-Range MCU family (that is they may be written for the Base-Line, or High-End families), but the concepts are pertinent, and could be used (with modification and possible limitations). The current application notes related to the comparator module are:

Title	Application Note #
Resistance and Capacitance Meter using a PIC16C622	AN611



Section 21. 8-bit A/D Converter

HIGHLIGHTS

This section of the manual contains the following major topics:

21.1	Introduction	21-2
21.2	Control Registers	21-3
21.3	Operation	21-5
21.4	A/D Acquisition Requirements	21-6
21.5	Selecting the A/D Conversion Clock	21-8
21.6	Configuring Analog Port Pins	21-9
21.7	A/D Conversions	21-10
21.8	A/D Operation During Sleep	21-12
21.9	A/D Accuracy/Error	21-13
21.10	Effects of a RESET	21-13
21.11	Use of the CCP Trigger	21-14
21.12	Connection Considerations	21-14
21.13	Transfer Function	21-14
21.14	Initialization	21-15
21.15	Design Tips	21-16
21.16	Related Application Notes	21-17
21.17	Revision History	21-18

Note: Please refer to Appendix C.3 or device Data Sheet to determine which devices use this module.
--

Section 21. 8-bit A/D Converter

21.2 Control Registers

Register 21-1: ADCON0 Register

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	Resv	ADON
bit 7							bit 0

bit 7:6 **ADCS1:ADCS0:** A/D Conversion Clock Select bits

00 = FOSC/2

01 = FOSC/8

10 = FOSC/32

11 = FRC (clock derived from the internal A/D RC oscillator)

bit 5:3 **CHS2:CHS0:** Analog Channel Select bits

000 = channel 0, (AN0)

001 = channel 1, (AN1)

010 = channel 2, (AN2)

011 = channel 3, (AN3)

100 = channel 4, (AN4)

101 = channel 5, (AN5)

110 = channel 6, (AN6)

111 = channel 7, (AN7)

Note: For devices that do not implement the full 8 A/D channels, the unimplemented selections are reserved. Do not select any unimplemented channels.

bit 2 **GO/DONE:** A/D Conversion Status bit

When ADON = 1

1 = A/D conversion in progress

(Setting this bit starts the A/D conversion. This bit is automatically cleared by hardware when the A/D conversion is complete)

0 = A/D conversion not in progress

bit 1 **Reserved:** Always maintain this bit cleared.

bit 0 **ADON:** A/D On bit

1 = A/D converter module is operating

0 = A/D converter module is shutoff and consumes no operating current

Legend

R = Readable bit W = Writable bit

U = Unimplemented bit, read as '0' - n = Value at POR reset

Section 21. 8-bit A/D Converter

21.3 Operation

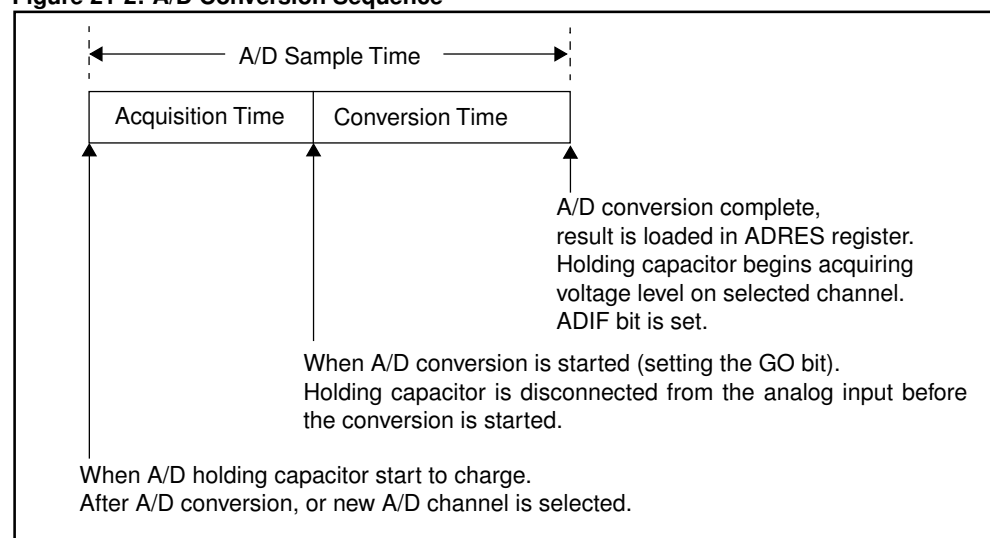
When the A/D conversion is complete, the result is loaded into the ADRES register, the GO/DONE bit (ADCON0<2>) is cleared, and A/D interrupt flag bit, ADIF, is set.

After the A/D module has been configured as desired, the selected channel must be acquired before the conversion is started. The analog input channels must have their corresponding TRIS bits selected as an input. To determine acquisition time, see Subsection 21.4 “A/D Acquisition Requirements.” After this acquisition time has elapsed the A/D conversion can be started. The following steps should be followed for doing an A/D conversion:

1. Configure the A/D module:
 - Configure analog pins / voltage reference / and digital I/O (ADCON1)
 - Select A/D input channel (ADCON0)
 - Select A/D conversion clock (ADCON0)
 - Turn on A/D module (ADCON0)
2. Configure A/D interrupt (if desired):
 - Clear the ADIF bit
 - Set the ADIE bit
 - Set the GIE bit
3. Wait the required acquisition time.
4. Start conversion:
 - Set the GO/DONE bit (ADCON0)
5. Wait for A/D conversion to complete, by either:
 - Polling for the GO/DONE bit to be clearedOR
 - Waiting for the A/D interrupt
6. Read A/D Result register (ADRES), clear the ADIF bit, if required.
7. For next conversion, go to step 1 or step 2 as required. The A/D conversion time per bit is defined as TAD. A minimum wait of 2TAD is required before next acquisition starts.

Figure 21-2 shows the conversion sequence, and the terms that are used. Acquisition time is the time that the A/D module's holding capacitor is connected to the external voltage level. Then there is the conversion time of 10 TAD, which is started when the GO bit is set. The sum of these two times is the sampling time. There is a minimum acquisition time to ensure that the holding capacitor is charged to a level that will give the desired accuracy for the A/D conversion.

Figure 21-2: A/D Conversion Sequence



Section 21. 8-bit A/D Converter

21

8-bit A/D
Converter

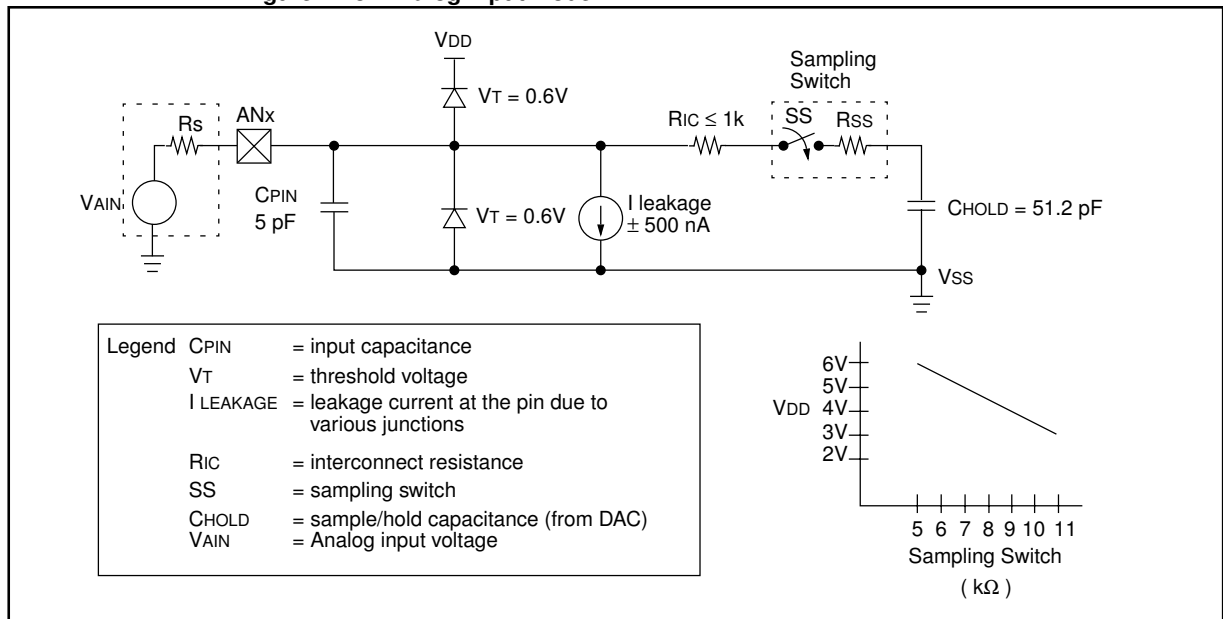
Note 1: The reference voltage (V_{REF}) has no effect on the equation, since it cancels itself out.

Note 2: The charge holding capacitor (C_{HOLD}) is not discharged after each conversion.

Note 3: The maximum recommended impedance for analog sources is $10\text{ k}\Omega$. This is required to meet the pin leakage specification.

Note 4: After a conversion has completed, a 2.0 TAD delay must complete before acquisition can begin again. During this time the holding capacitor is not connected to the selected A/D input channel.

Figure 21-3: Analog Input Model



Section 21. 8-bit A/D Converter

21.6 Configuring Analog Port Pins

ADCON1 and the corresponding TRIS registers control the operation of the A/D port pins. The port pins that are desired as analog inputs must have their corresponding TRIS bits set (input). If the TRIS bit is cleared (output), the digital output level (V_{OH} or V_{OL}) will be converted.

The A/D operation is independent of the state of the CHS2:CHS0 bits and the TRIS bits.

Note 1: When reading the port register, all pins configured as analog input channels will read as cleared (a low level). Pins configured as digital inputs, will convert an analog input. Analog levels on a digitally configured input will not affect the conversion accuracy.

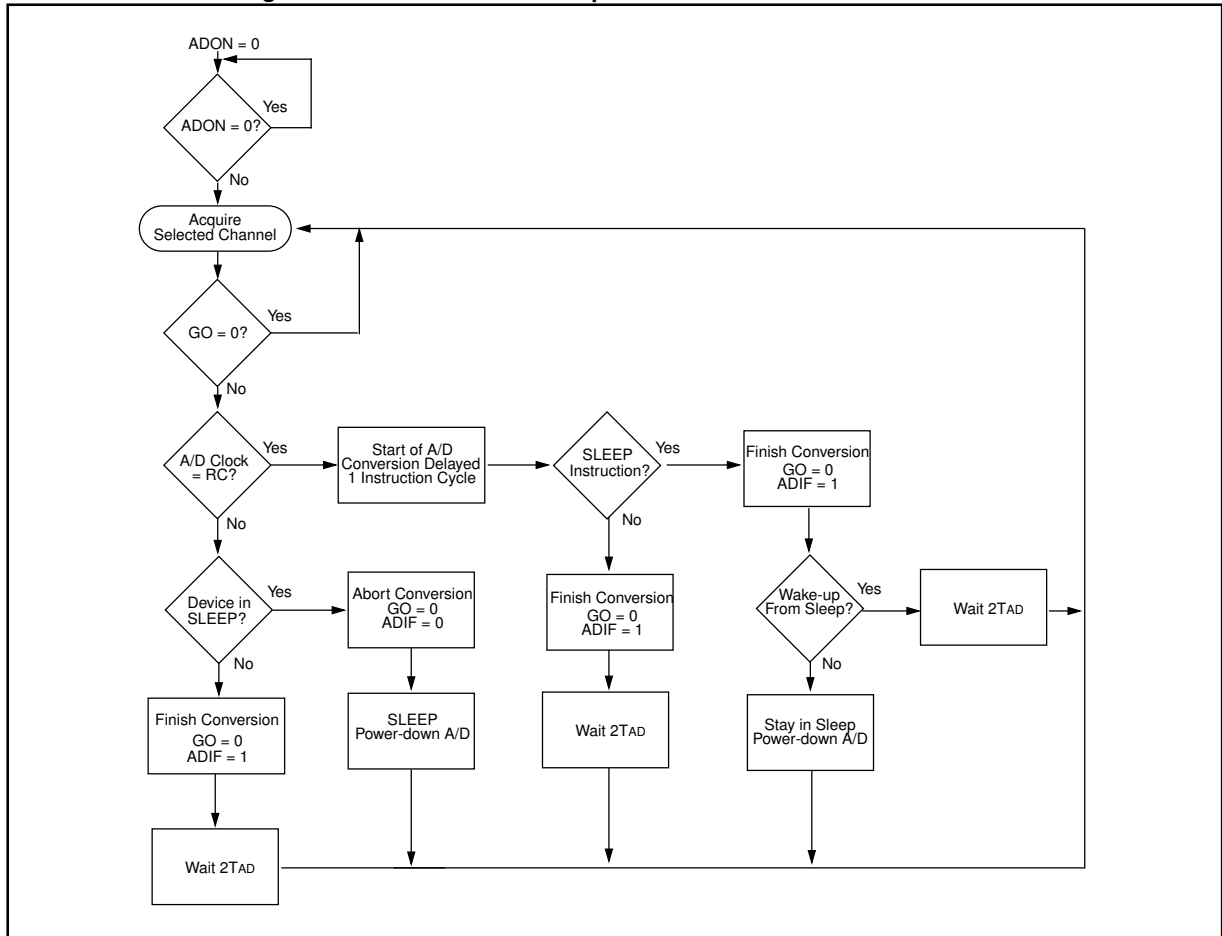
Note 2: Analog levels on any pin that is defined as a digital input (including the AN7:AN0 pins), may cause the input buffer to consume current that is out of the devices specification.

Section 21. 8-bit A/D Converter

21

8-bit A/D
Converter

Figure 21-5: Flowchart of A/D Operation



Section 21. 8-bit A/D Converter

21.9 A/D Accuracy/Error

In systems where the device frequency is low, use of the A/D RC clock is preferred. At moderate to high frequencies, TAD should be derived from the device oscillator.

The absolute accuracy specified for the A/D converter includes the sum of all contributions for quantization error, integral error, differential error, full scale error, offset error, and monotonicity. It is defined as the maximum deviation from an actual transition versus an ideal transition for any code. The absolute error of the A/D converter is specified at $< \pm 1$ LSb for $V_{DD} = V_{REF}$ (over the device's specified operating range). However, the accuracy of the A/D converter will degrade as V_{DD} diverges from V_{REF} .

For a given range of analog inputs, the output digital code will be the same. This is due to the quantization of the analog input to a digital code. Quantization error is typically $\pm 1/2$ LSb and is inherent in the analog to digital conversion process. The only way to reduce quantization error is to increase the resolution of the A/D converter.

Offset error measures the first actual transition of a code versus the first ideal transition of a code. Offset error shifts the entire transfer function. Offset error can be calibrated out of a system or introduced into a system through the interaction of the total leakage current and source impedance at the analog input.

Gain error measures the maximum deviation of the last actual transition and the last ideal transition adjusted for offset error. This error appears as a change in slope of the transfer function. The difference in gain error to full scale error is that full scale does not take offset error into account. Gain error can be calibrated out in software.

Linearity error refers to the uniformity of the code changes. Linearity errors cannot be calibrated out of the system. Integral non-linearity error measures the actual code transition versus the ideal code transition adjusted by the gain error for each code.

Differential non-linearity measures the maximum actual code width versus the ideal code width. This measure is unadjusted.

The maximum pin leakage current is specified in the Device Data Sheet electrical specification [parameter D060](#).

In systems where the device frequency is low, use of the A/D RC clock is preferred. At moderate to high frequencies, TAD should be derived from the device oscillator. TAD must not violate the minimum and should be minimized to reduce inaccuracies due to noise and sampling capacitor bleed off.

In systems where the device will enter SLEEP mode after the start of the A/D conversion, the RC clock source selection is required. In this mode, the digital noise from the modules in SLEEP are stopped. This method gives high accuracy.

21.10 Effects of a RESET

A device reset forces all registers to their reset state. This forces the A/D module to be turned off, and any conversion is aborted. The value that is in the ADRES register is not modified for a Power-on Reset. The ADRES register will contain unknown data after a Power-on Reset.

Section 21. 8-bit A/D Converter

21.14 Initialization

[Example 21-4](#) shows the initialization of the A/D module for the PIC16C74A

Example 21-4: A/D Initialization (for PIC16C74A)

```
BSF    STATUS, RP0    ; Select Bank1
CLRF   ADCON1         ; Configure A/D inputs
BSF    PIE1, ADIE      ; Enable A/D interrupts
BCF    STATUS, RP0    ; Select Bank0
MOVLW  0xC1           ; RC Clock, A/D is on, Channel 0 is selected
MOVWF  ADCON0         ;
BCF    PIR1, ADIF      ; Clear A/D interrupt flag bit
BSF    INTCON, PEIE    ; Enable peripheral interrupts
BSF    INTCON, GIE     ; Enable all interrupts
;
; Ensure that the required sampling time for the selected input
; channel has elapsed. Then the conversion may be started.
;
BSF    ADCON0, GO      ; Start A/D Conversion
:      ; The ADIF bit will be set and the GO/DONE
:      ; bit is cleared upon completion of the
:      ; A/D Conversion.
```