

PICmicro MID-RANGE MCU FAMILY

17.4.2 General Call Address Support

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

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

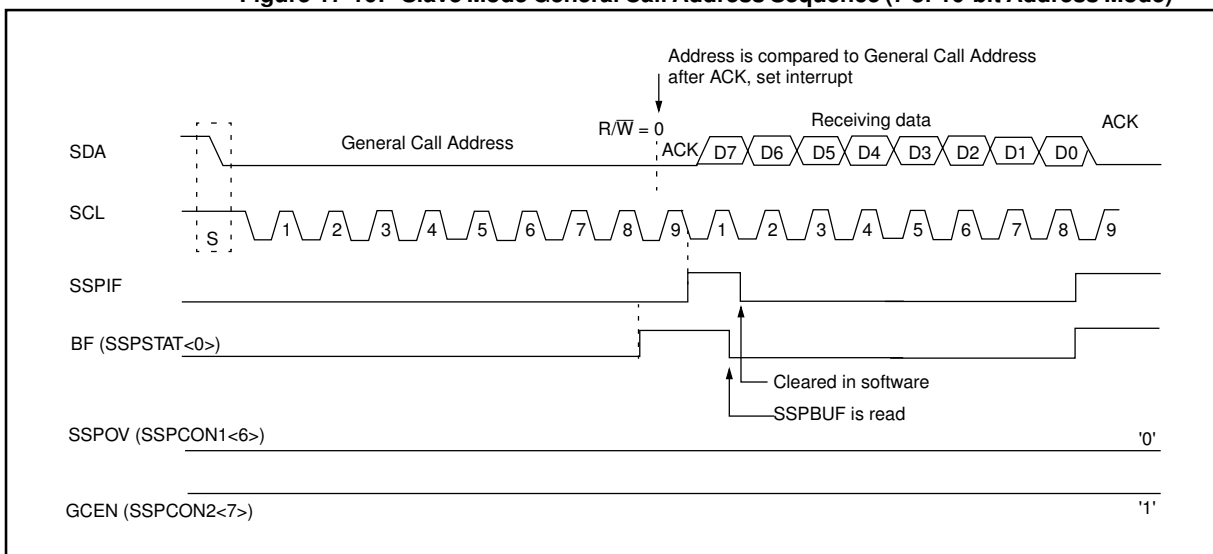
The general call address is recognized when the General Call Enable bit (GCEN) is enabled (SSPCON2<7> set). Following a start-bit detect, 8-bits are shifted into SSPSR and the address is compared against SSPADD, and is also compared to the general call address, fixed in hardware.

If the general call address matches, the SSPSR is transferred to the SSPBUF, the BF flag bit is set (eight bit), and on the falling edge of the ninth bit (\overline{ACK} bit) the SSPIF interrupt flag bit is set.

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

In 10-bit mode, the SSPADD is required to be updated for the second half of the address to match, and the UA bit is set (SSPSTAT<1>). If the general call address is sampled when the GCEN bit is set while the slave is configured in 10-bit address mode, then the second half of the address is not necessary, the UA bit will not be set, and the slave will begin receiving data after the acknowledge (Figure 17-16).

Figure 17-16: Slave Mode General Call Address Sequence (7 or 10-bit Address Mode)



In master mode the SCL and SDA lines are manipulated by the SSP hardware.

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

--



PICmicro MID-RANGE MCU FAMILY

17.4.7.1 I²C Master Mode Operation

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

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

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

The baud rate generator used for SPI mode operation is used to set the SCL clock frequency for either 100 kHz, 400 kHz, or 1 MHz I²C operation. The baud rate generator reload value is contained in the lower 7 bits of the SSPADD register. The baud rate generator will automatically begin counting on a write to the SSPBUF. Once the given operation is complete (i.e., transmission of the last data bit is followed by ACK) the internal clock will automatically stop counting and the SCL pin will remain in its last state.

A typical transmit sequence would go as follows:

- a) The user generates a Start Condition by setting the START enable bit, SEN (SSPCON2<0>).
- b) SSPIF is set. The SSP module will wait the required start time before any other operation takes place.
- c) The user loads the SSPBUF with the address to transmit.
- d) Address is shifted out the SDA pin until all 8 bits are transmitted.
- e) The SSP Module shifts in the ACK bit from the slave device, and writes its value into the SSPCON2 register (SSPCON2<6>).
- f) The SSP module generates an interrupt at the end of the ninth clock cycle by setting the SSPIF bit.
- g) The user loads the SSPBUF with eight bits of data.
- h) DATA is shifted out the SDA pin until all 8 bits are transmitted.
- i) The SSP Module shifts in the ACK bit from the slave device, and writes its value into the SSPCON2 register (SSPCON2<6>).
- j) The SSP module generates an interrupt at the end of the ninth clock cycle by setting the SSPIF bit.
- k) The user generates a STOP condition by setting the STOP enable bit, PEN (SSPCON2<2>).
- l) Interrupt is generated once the stop condition is complete.

PICmicro MID-RANGE MCU FAMILY

17.4.9 I²C Master Mode Start Condition Timing

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

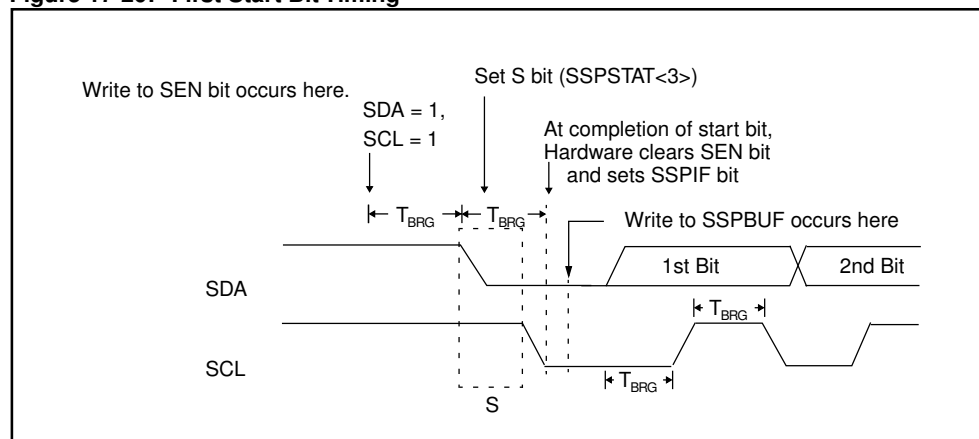
Note: If at the beginning of START condition the SDA and SCL pins are already sampled low, or if during the START condition the SCL line is sampled low before the SDA line is driven low, a bus collision occurs, the Bus Collision Interrupt Flag, BCLIF, is set, the START condition is aborted, and the I²C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

17.4.9.1 WCOL Status Flag

If the user writes the SSPBUF when an 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 to the lower 5 bits of SSPCON2 is disabled until the START condition is complete.

Figure 17-20: First Start Bit Timing



PICmicro MID-RANGE MCU FAMILY

17.4.10 I²C Master Mode Repeated Start Condition Timing

A Repeated Start condition occurs when the RSEN bit (SSPCON2<1>) is programmed high and the I²C logic module is in the idle state. When the RSEN bit is set, the SCL pin is asserted low. When the SCL pin is sampled low, the baud rate generator is loaded with the contents of SSPADD<5:0>, and begins counting. The SDA pin is released (brought high) for one baud rate generator count (T_{BRG}). When the baud rate generator times out, if SDA is sampled high, the SCL pin will be de-asserted (brought high). When SCL is sampled high the baud rate generator is re-loaded with the contents of SSPADD<6:0> and begins counting. SDA and SCL must be sampled high for one T_{BRG} . This action is then followed by assertion of the SDA pin ($SDA = 0$) for one T_{BRG} while SCL is high. Following this, the RSEN bit (SSPCON2<1>) will be automatically cleared, and the baud rate generator is not reloaded, leaving the SDA pin held low. As soon as a start condition is detected on the SDA and SCL pins, the S bit (SSPSTAT<3>) will be set. The SSPIF bit will not be set until the baud rate generator has timed-out.

Note 1: If RSEN is programmed while any other event is in progress, it will not take effect.

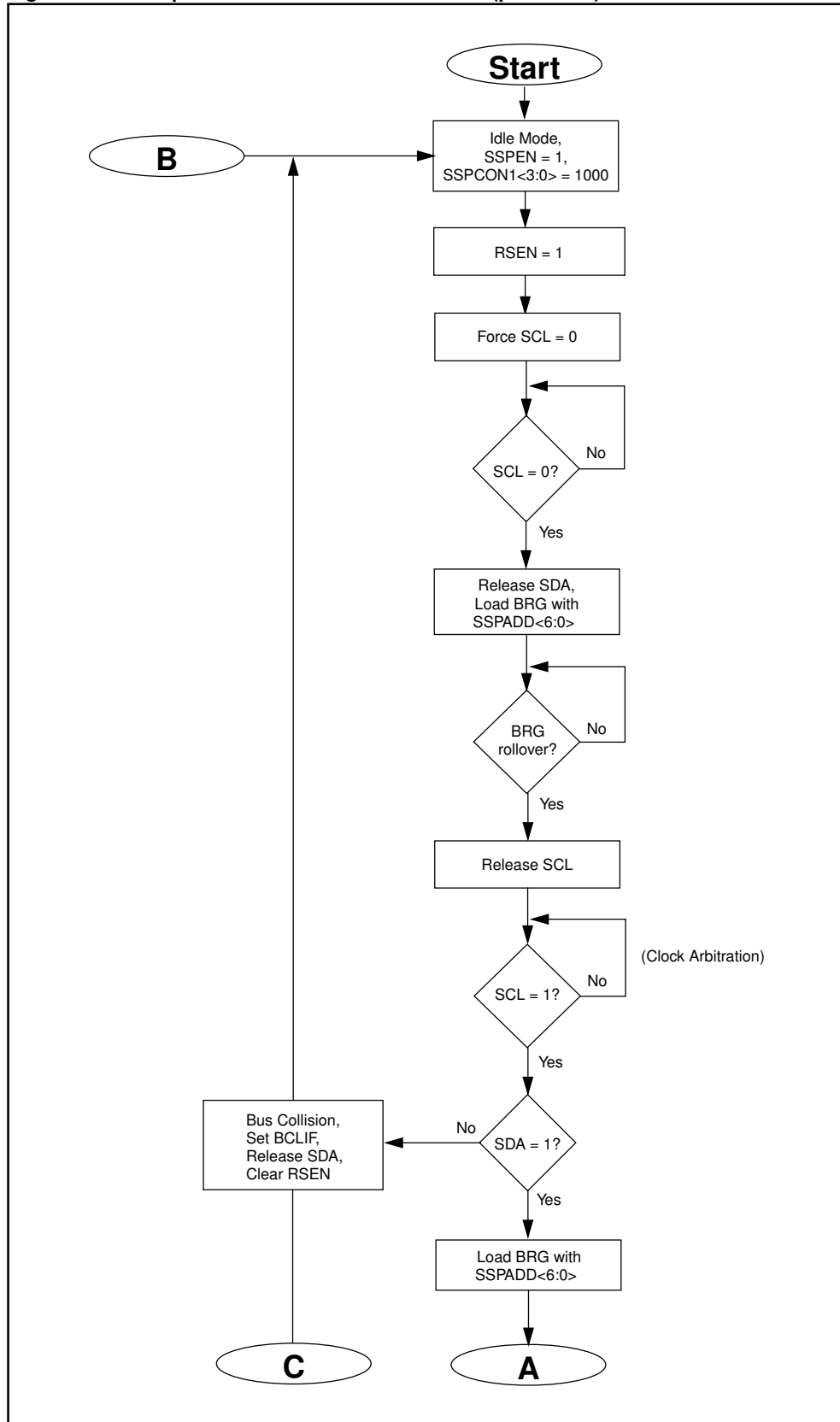
Note 2: A bus collision during the Repeated Start condition occurs if:

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

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

PICmicro MID-RANGE MCU FAMILY

Figure 17-23: Repeated Start Condition Flowchart (part 1 of 2)



PICmicro MID-RANGE MCU FAMILY

17.4.11 I²C Master Mode Transmission

Transmission of a data byte, a 7-bit address, or the either half of a 10-bit address is accomplished by simply writing a value to SSPBUF register. This action will set the buffer full flag 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 SDA pin after the falling edge of SCL is asserted (see data hold time specification [parameters 106](#)). SCL is held low for one baud rate generator roll over count (T_{BRG}). Data should be valid before SCL is released high (see Data setup time specification [parameters 107](#)). When the SCL pin is released high, it is held that way for T_{BRG} , the data on the SDA pin must remain stable for that duration and some hold time after the next falling edge of SCL. After the eighth bit is shifted out (the falling edge of the eighth clock), the BF flag is cleared and the master releases SDA allowing the slave device being addressed to respond with an ACK bit during the ninth bit time, if an address match occurs or if data was received properly. The status of \overline{ACK} is 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 SSPIF bit is set, and the master clock (baud rate generator) is suspended until the next data byte is loaded into the SSPBUF leaving SCL low and SDA unchanged ([Figure 17-26](#)).

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

17.4.11.1 BF Status Flag

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

17.4.11.2 WCOL Status Flag

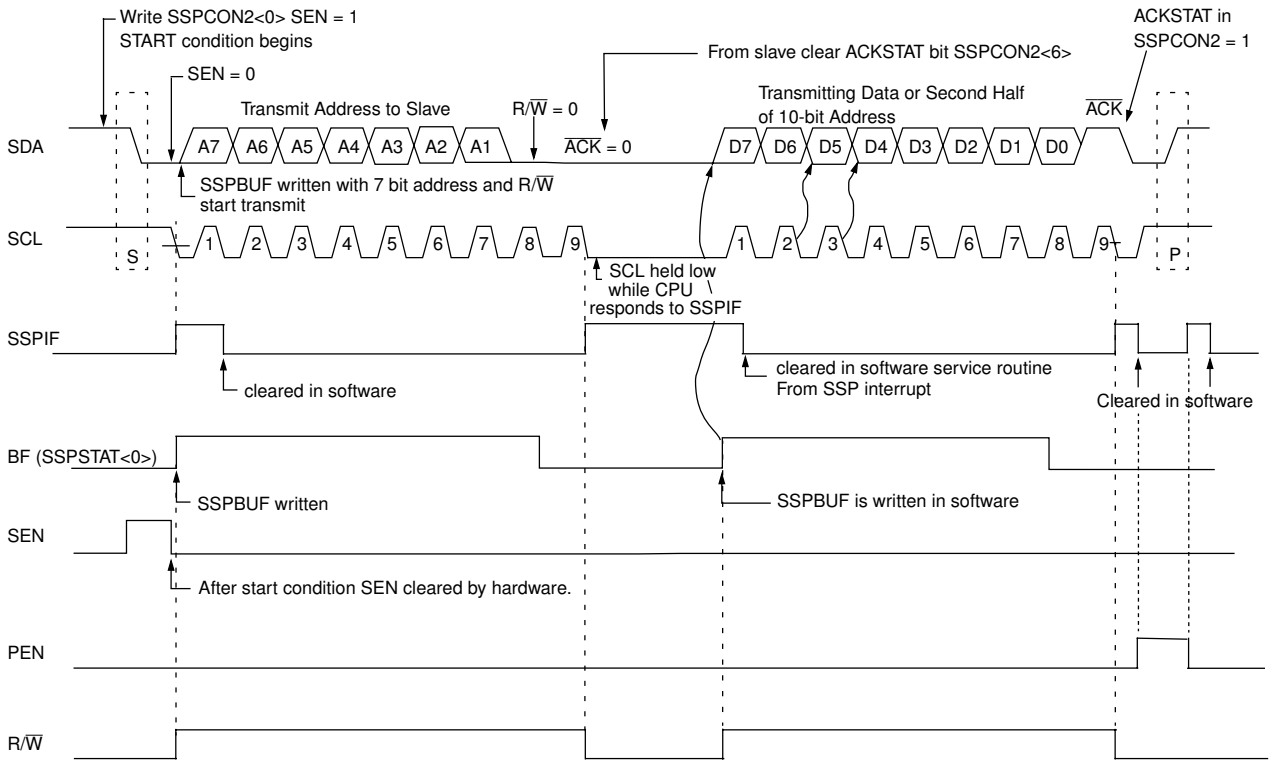
If the user writes the SSPBUF when a transmit is already in progress (i.e. SSPSR is still shifting out a data byte), then WCOL is set and the contents of the buffer are unchanged (the write doesn't occur).

WCOL must be cleared in software.

17.4.11.3 ACKSTAT Status Flag

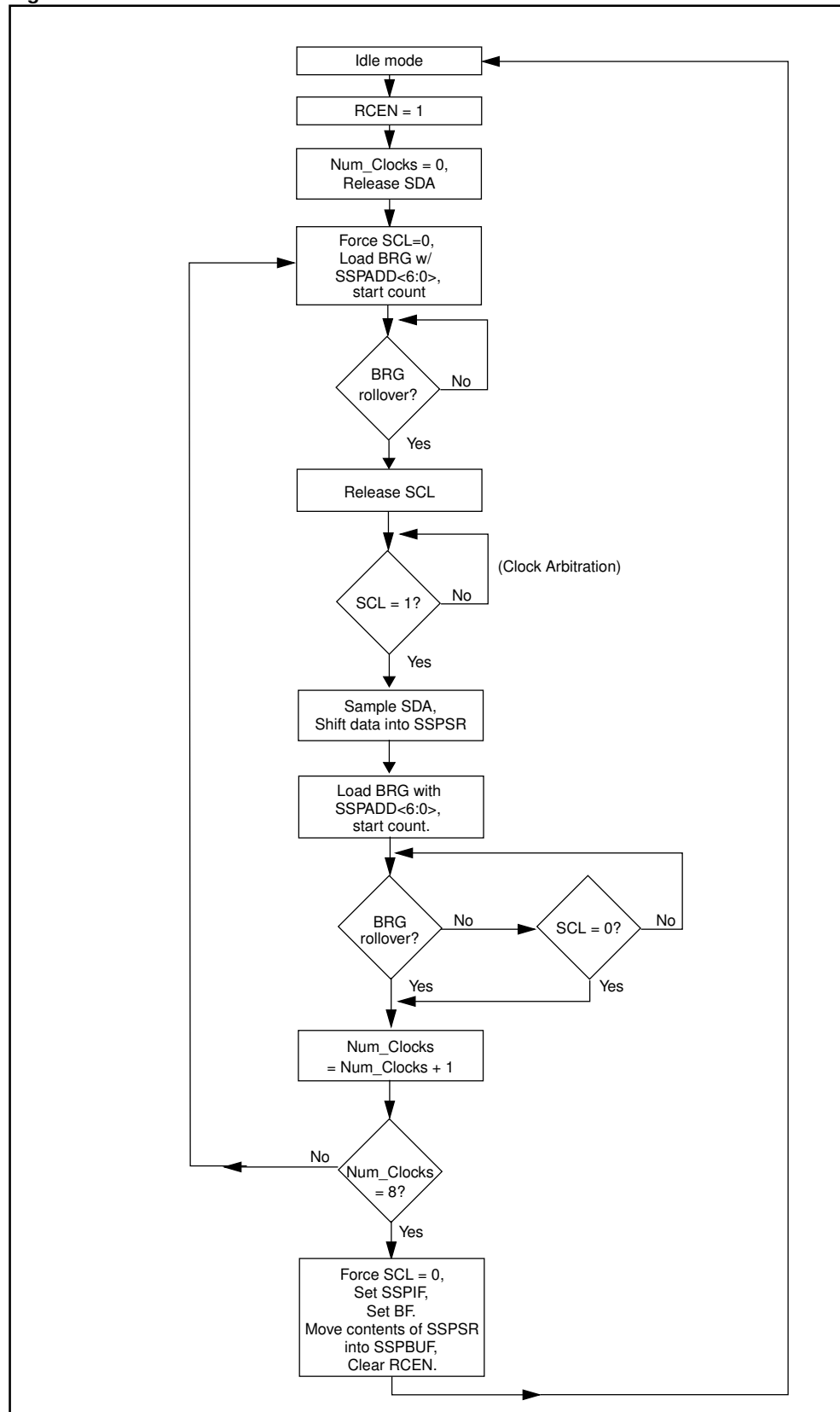
In transmit mode, the ACKSTAT bit (SSPCON2<6>) is cleared when the slave has sent an acknowledge ($\overline{ACK} = 0$), and is set when the slave does not acknowledge ($\overline{ACK} = 1$). A slave sends an acknowledge when it has recognized its address (including a general call), or when the slave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

Figure 17-26: ^{12}C Master Mode Waveform (Transmission, 7 or 10-bit Address,



PICmicro MID-RANGE MCU FAMILY

Figure 17-27: Master Receiver Flowchart



PICmicro MID-RANGE MCU FAMILY

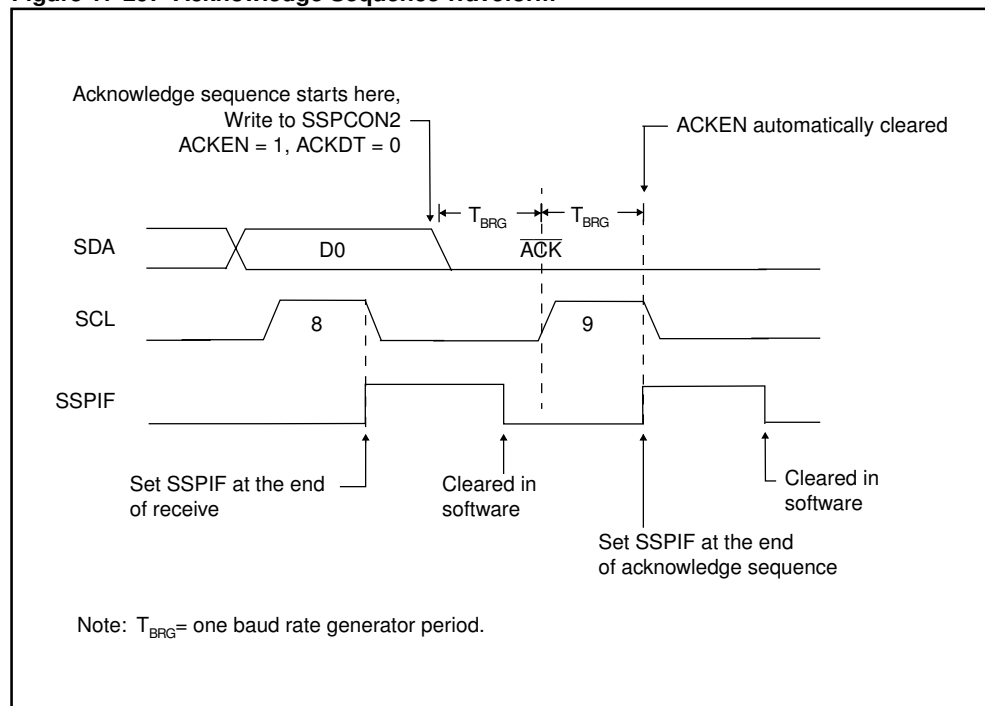
17.4.13 Acknowledge Sequence Timing

An acknowledge sequence is enabled by setting the acknowledge sequence enable bit, ACKEN (SSPCON2<4>). When this bit is set, the SCL pin is pulled low and the contents of the acknowledge data bit is presented on the SDA pin. If the user wishes to generate an acknowledge, 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 (T_{BRG}), and the SCL pin is de-asserted (pulled high). When the SCL pin is sampled high (clock arbitration), the baud rate generator counts for T_{BRG} . The SCL pin is then pulled low. Following this, the ACKEN bit is automatically cleared, the baud rate generator is turned off, and the SSP module then goes into IDLE mode (Figure 17-29).

17.4.13.1 WCOL Status Flag

If the user writes the SSPBUF 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 17-29: Acknowledge Sequence Waveform

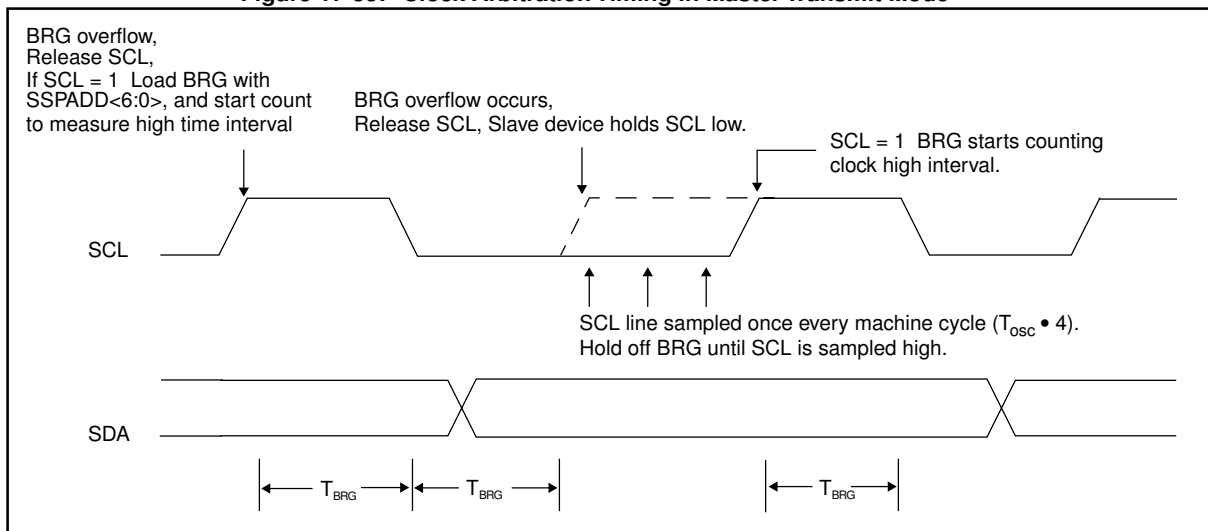


PICmicro MID-RANGE MCU FAMILY

17.4.15 Clock Arbitration

Clock arbitration occurs when the master, during any receive, transmit, or Repeated Start/stop condition de-asserts the SCL pin (SCL allowed to float high). When the SCL pin is allowed to float high, the baud rate generator (BRG) is suspended from counting until the SCL pin is actually sampled high. When the SCL pin is sampled high, the baud rate generator is reloaded with the contents of SSPADD<6:0> and begins counting. This ensures that the SCL high time will always be at least one BRG rollover count in the event that the clock is held low by an external device (Figure 17-33).

Figure 17-33: Clock Arbitration Timing in Master Transmit Mode



17.4.16 Sleep Operation

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

17.4.17 Effect of a Reset

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

PICmicro MID-RANGE MCU FAMILY

17.4.18.1 Bus Collision During a START Condition

During a START condition, a bus collision occurs if:

- a) SDA or SCL are sampled low at the beginning of the START condition ([Figure 17-35](#)).
- b) SCL is sampled low before SDA is asserted low ([Figure 17-36](#)).

During a START condition both the SDA and the SCL pins are monitored.

If:

the SDA pin is already low
or the SCL pin is already low,

then:

the START condition is aborted,
and the BCLIF flag is set,
and the SSP module is reset to its IDLE state ([Figure 17-35](#)).

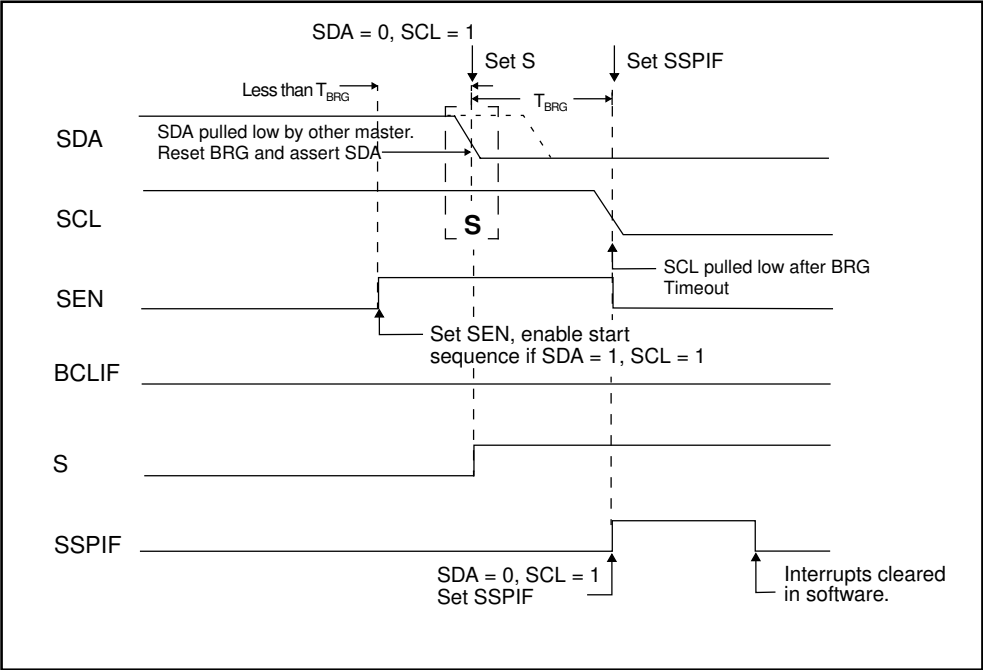
The START condition begins with the SDA and SCL pins de-asserted. When the SDA pin is sampled high, the baud rate generator is loaded from SSPADD<6:0> and counts down to 0. If the SCL pin is sampled low while SDA is high, a bus collision occurs, because it is assumed that another master is attempting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

If the SDA pin is sampled low during this count, the BRG is reset and the SDA line is asserted early ([Figure 17-37](#)). If however a '1' is sampled on the SDA pin, the SDA pin is asserted low at the end of the BRG count. The baud rate generator is then reloaded and counts down to 0, and during this time, if the SCL pins is sampled as '0', a bus collision does not occur. At the end of the BRG count the SCL pin is asserted low.

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

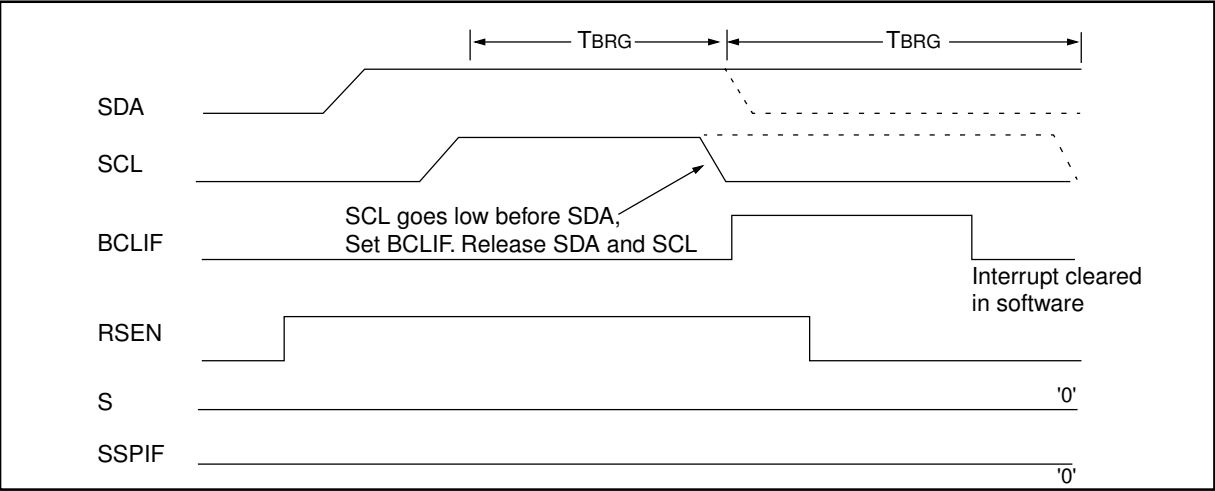
PICmicro MID-RANGE MCU FAMILY

Figure 17-37: BRG Reset Due to SDA Arbitration During Start Condition



PICmicro MID-RANGE MCU FAMILY

Figure 17-39: Bus Collision During Repeated Start Condition (Case 2)



PICmicro MID-RANGE MCU FAMILY

17.5 Connection Considerations for I²C Bus

For standard-mode I²C bus devices, the values of resistors **RP** and **RS** in Figure 17-42 depends on the following parameters:

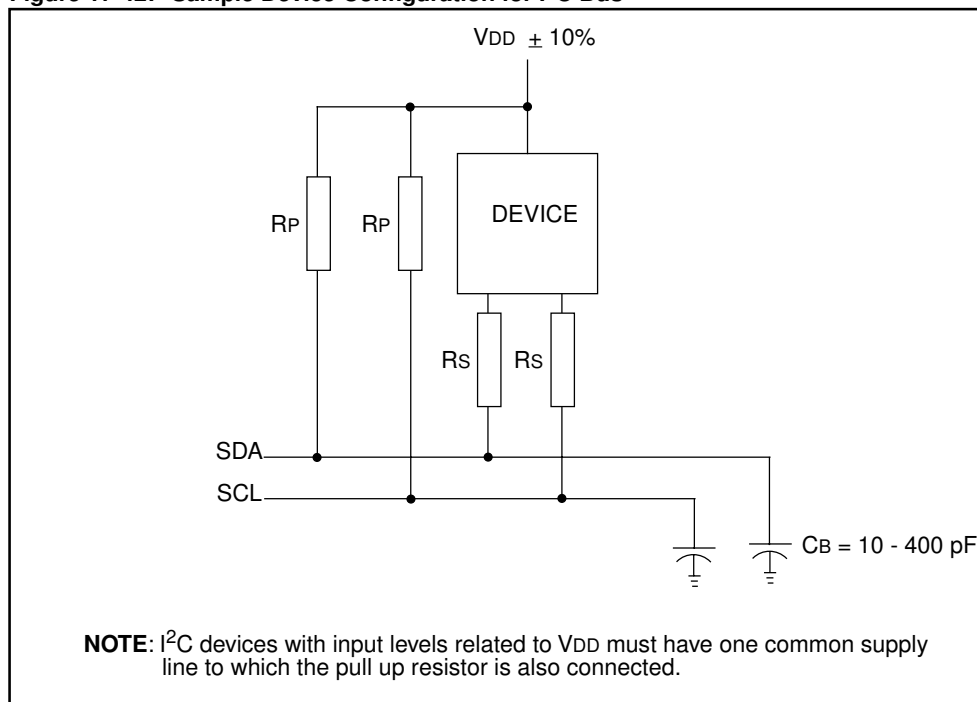
- Supply voltage
- Bus capacitance
- Number of connected devices (input current + leakage current)

The supply voltage limits the minimum value of resistor **RP** due to the specified minimum sink current of 3 mA at VOLMAX = 0.4V for the specified output stages. For example, with a supply voltage of VDD = 5V±10% and VOLMAX = 0.4V at 3 mA, RPMIN = (5.5-0.4)/0.003 = 1.7 kΩ. VDD as a function of **RP** is shown in Figure 17-42. The desired noise margin of 0.1VDD for the low level, limits the maximum value of **RS**. Series resistors are optional, and used to improve ESD susceptibility.

The bus capacitance is the total capacitance of wire, connections, and pins. This capacitance limits the maximum value of **RP** due to the specified rise time (Figure 17-42).

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

Figure 17-42: Sample Device Configuration for I²C Bus



PICmicro MID-RANGE MCU FAMILY

17.7 Design Tips

Question 1: *Using SPI mode, I do not seem able to talk to an SPI device.*

Answer 1:

Ensure that you are using the correct SPI mode for that device. This SPI supports all 4 SPI modes so you should be able to get it to function. Check the clock polarity and the clock phase.

Question 2: *Using I²C mode, I write data to the SSPBUF register, but the data did not transmit.*

Answer 2:

Ensure that you set the CKP bit to release the I²C clock.

PICmicro MID-RANGE MCU FAMILY

17.9 Revision History

Revision A

This is the initial released revision of the Master SSP module description.

PICmicro MID-RANGE MCU FAMILY

18.1 Introduction

The Universal Synchronous Asynchronous Receiver Transmitter (USART) module is one of the two serial I/O modules (other is the SSP module). The USART is also known as a Serial Communications Interface or SCI. The USART can be configured as a full duplex asynchronous system that can communicate with peripheral devices such as CRT terminals and personal computers, or it can be configured as a half duplex synchronous system that can communicate with peripheral devices such as A/D or D/A integrated circuits, Serial EEPROMs etc.

The USART can be configured in the following modes:

- Asynchronous (full duplex)
- Synchronous - Master (half duplex)
- Synchronous - Slave (half duplex)

The SPEN bit (RCSTA<7>), and the TRIS bits, have to be set in order to configure the TX/CK and RX/DT pins for the USART.

PICmicro MID-RANGE MCU FAMILY

Register 18-2: RCSTA: Receive Status and Control Register

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R-0	R-0	R-0
SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D
bit 7							bit 0

- bit 7 **SPEN:** Serial Port Enable bit
1 = Serial port enabled (Configures RX/DT and TX/CK pins as serial port pins)
0 = Serial port disabled
- bit 6 **RX9:** 9-bit Receive Enable bit
1 = Selects 9-bit reception
0 = Selects 8-bit reception
- bit 5 **SREN:** Single Receive Enable bit
Asynchronous mode
Don't care
Synchronous mode - master
1 = Enables single receive
0 = Disables single receive
This bit is cleared after reception is complete.
Synchronous mode - slave
Unused in this mode
- bit 4 **CREN:** Continuous Receive Enable bit
Asynchronous mode
1 = Enables continuous receive
0 = Disables continuous receive
Synchronous mode
1 = Enables continuous receive until enable bit CREN is cleared (CREN overrides SREN)
0 = Disables continuous receive
- bit 3 **Unimplemented:** Read as '0'
- bit 2 **FERR:** Framing Error bit
1 = Framing error (Can be updated by reading RCREG register and receive next valid byte)
0 = No framing error
- bit 1 **OERR:** Overrun Error bit
1 = Overrun error (Can be cleared by clearing bit CREN)
0 = No overrun error
- bit 0 **RX9D:** 9th bit of received data, can be parity bit.

Legend

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

- n = Value at POR reset

PICmicro MID-RANGE MCU FAMILY

Table 18-3: Baud Rates for Synchronous Mode

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	NA	-	-	NA	-	-	NA	-	-	NA	-	-
2.4	NA	-	-	NA	-	-	NA	-	-	NA	-	-
9.6	NA	-	-	NA	-	-	9.766	+1.73	255	9.622	+0.23	185
19.2	19.53	+1.73	255	19.23	+0.16	207	19.23	+0.16	129	19.24	+0.23	92
76.8	76.92	+0.16	64	76.92	+0.16	51	75.76	-1.36	32	77.82	+1.32	22
96	96.15	+0.16	51	95.24	-0.79	41	96.15	+0.16	25	94.20	-1.88	18
300	294.1	-1.96	16	307.69	+2.56	12	312.5	+4.17	7	298.3	-0.57	5
500	500	0	9	500	0	7	500	0	4	NA	-	-
HIGH	5000	-	0	4000	-	0	2500	-	0	1789.8	-	0
LOW	19.53	-	255	15.625	-	255	9.766	-	255	6.991	-	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	NA	-	-	NA	-	-	NA	-	-	NA	-	-	0.303	+1.14	26
1.2	NA	-	-	NA	-	-	NA	-	-	1.202	+0.16	207	1.170	-2.48	6
2.4	NA	-	-	NA	-	-	NA	-	-	2.404	+0.16	103	NA	-	-
9.6	9.6	0	131	9.615	+0.16	103	9.622	+0.23	92	9.615	+0.16	25	NA	-	-
19.2	19.2	0	65	19.231	+0.16	51	19.04	-0.83	46	19.24	+0.16	12	NA	-	-
76.8	79.2	+3.13	15	76.923	+0.16	12	74.57	-2.90	11	83.34	+8.51	2	NA	-	-
96	97.48	+1.54	12	1000	+4.17	9	99.43	+3.57	8	NA	-	-	NA	-	-
300	316.8	+5.60	3	NA	-	-	298.3	-0.57	2	NA	-	-	NA	-	-
500	NA	-	-	NA	-	-	NA	-	-	NA	-	-	NA	-	-
HIGH	1267	-	0	100	-	0	894.9	-	0	250	-	0	8.192	-	0
LOW	4.950	-	255	3.906	-	255	3.496	-	255	0.9766	-	255	0.032	-	255

PICmicro MID-RANGE MCU FAMILY

18.4 USART Asynchronous Mode

In this mode, the USART uses standard nonreturn-to-zero (NRZ) format (one start bit, eight or nine data bits and one stop bit). The most common data format is 8-bits. An on-chip dedicated 8-bit baud rate generator can be used to derive standard baud rate frequencies from the oscillator. The USART transmits and receives the LSb first. The USART's transmitter and receiver are functionally independent but use the same data format and baud rate. The baud rate generator produces a clock either x16 or x64 of the bit shift rate, depending on the BRGH bit (TXSTA<2>). Parity is not supported by the hardware, but can be implemented in software (stored as the ninth data bit). Asynchronous mode is stopped during SLEEP.

Asynchronous mode is selected by clearing the SYNC bit (TXSTA<4>).

The USART Asynchronous module consists of the following important elements:

- Baud Rate Generator
- Sampling Circuit
- Asynchronous Transmitter
- Asynchronous Receiver

18.4.1 USART Asynchronous Transmitter

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, TXREG. The TXREG register is loaded with data in software. The TSR register is not loaded until the STOP bit has been transmitted from the previous load. As soon as the STOP bit is transmitted, the TSR is loaded with new data from the TXREG register (if available). Once the TXREG register transfers the data to the TSR register (occurs in one Tcy), the TXREG register is empty and the TXIF flag bit is set. This interrupt can be enabled/disabled by setting/clearing the TXIE enable 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 indicated the status of the TXREG register, the TRMT bit (TXSTA<1>) shows the status of the TSR register. The TRMT status bit is a read only bit which is set when the TSR register 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.

Note 1: The TSR register is not mapped in data memory so it is not available to the user.

Note 2: When the TXEN bit is set, the TXIF flag bit will also be set since the transmit buffer is not yet full (still can move transmit data to the TXREG register).

Transmission is enabled by setting the TXEN enable bit (TXSTA<5>). The actual transmission will not occur until the TXREG register has been loaded with data and the baud rate generator (BRG) has produced a shift clock ([Figure 18-1](#)). The transmission can also be started by first loading the TXREG register and then setting the TXEN enable bit. 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. A back-to-back transfer is thus possible ([Figure 18-3](#)). Clearing the TXEN enable bit during a transmission will cause the transmission to be aborted and will reset the transmitter. As a result the TX/CK pin will revert to hi-impedance.

In order to select 9-bit transmission, transmit bit, TX9 (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 register can result in an immediate transfer of the data to the TSR register (if the TSR is empty). In such a case, an incorrect ninth data bit maybe loaded in the TSR register.

PICmicro MID-RANGE MCU FAMILY

Figure 18-3: Asynchronous Master Transmission (Back to Back)

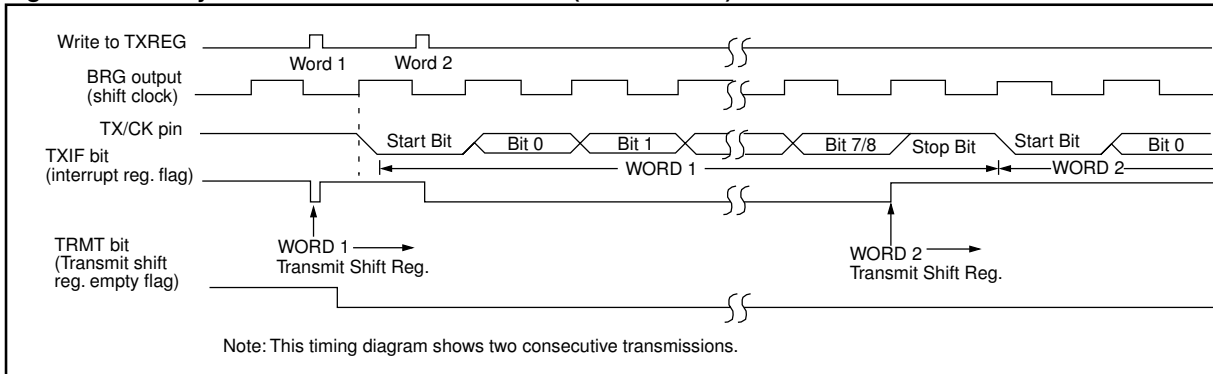


Table 18-6: Registers Associated with Asynchronous Transmission

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 locations read as '0'.

Shaded cells are not used for Asynchronous Transmission.

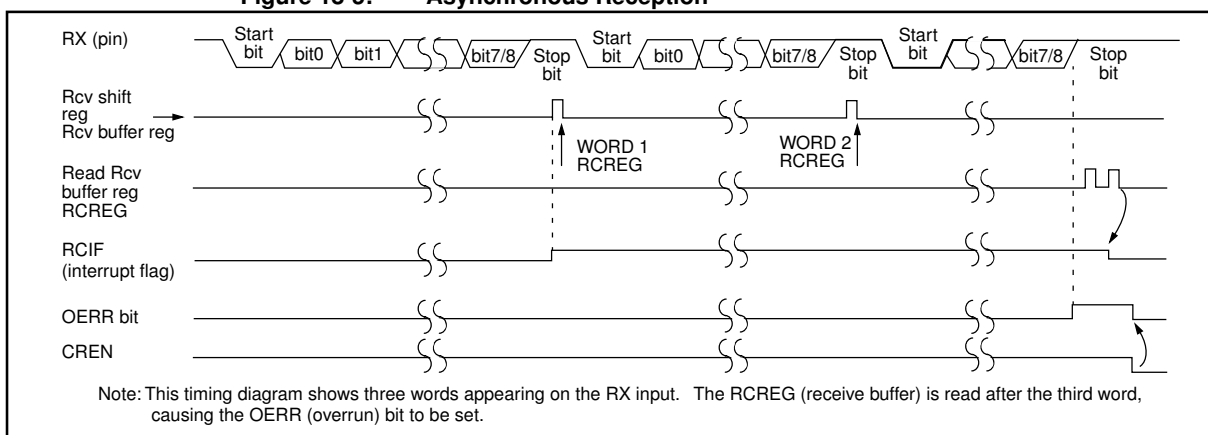
Note 1: The position of this bit is device dependent.

PICmicro MID-RANGE MCU FAMILY

Steps to follow when setting up an Asynchronous Reception:

1. Initialize the SPBRG register for the appropriate baud rate. If a high speed baud rate is desired, set bit BRGH. (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 RCIE, GIE and PEIE bits.
4. If 9-bit reception is desired, then set the RX9 bit.
5. Enable the reception by setting the CREN bit.
6. The RCIF flag bit will be set when reception is complete and an interrupt will be generated if the RCIE bit was set.
7. Read the RSTA register to get the ninth bit (if enabled) and determine if any error occurred during reception.
8. Read the 8-bit received data by reading the RCREG register.
9. If any error occurred, clear the error by clearing the CREN bit.

Figure 18-5: Asynchronous Reception



PICmicro MID-RANGE MCU FAMILY

Figure 18-8: RX Pin Sampling Scheme (BRGH = 1)

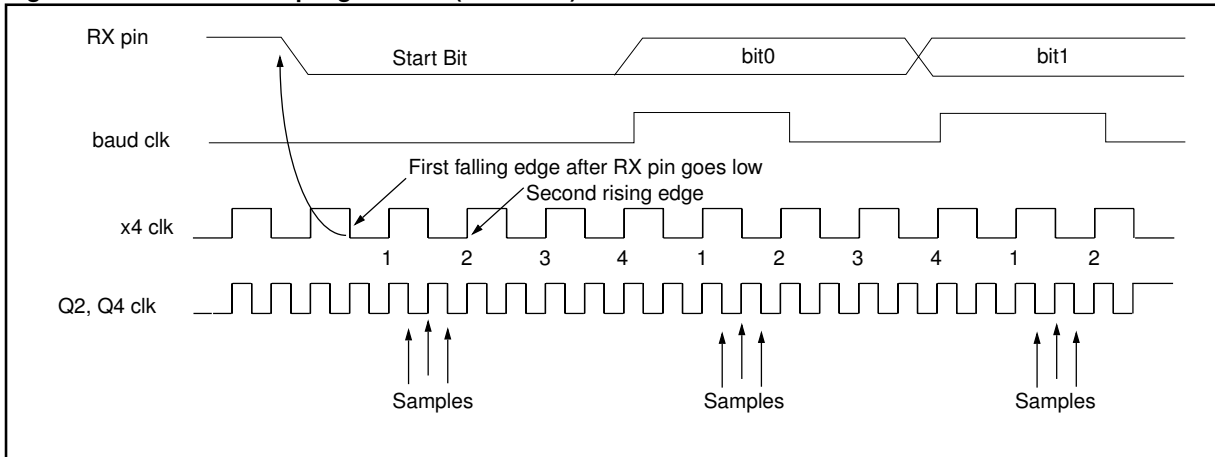


Figure 18-9: RX Pin Sampling Scheme (BRGH = 1)

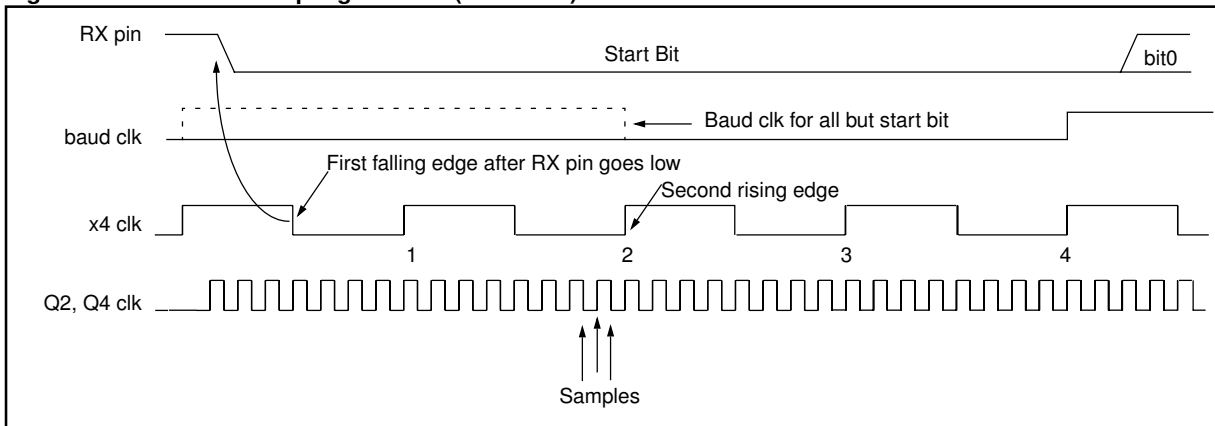


Table 18-7: Registers Associated with Asynchronous Reception

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 locations read as '0'.

Shaded cells are not used for Asynchronous Reception.

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

PICmicro MID-RANGE MCU FAMILY

Steps to follow when setting up a Synchronous Master Transmission:

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. If interrupts are desired, then set the TXIE bit.
4. If 9-bit transmission is desired, then set the TX9 bit.
5. Enable the transmission by setting the TXEN bit.
6. If 9-bit transmission is selected, the ninth bit should be loaded in the TX9D bit.
7. Start transmission by loading data to the TXREG register.

Table 18-8: Registers Associated with Synchronous Master 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 Master Transmission.

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

Figure 18-10: Synchronous Transmission

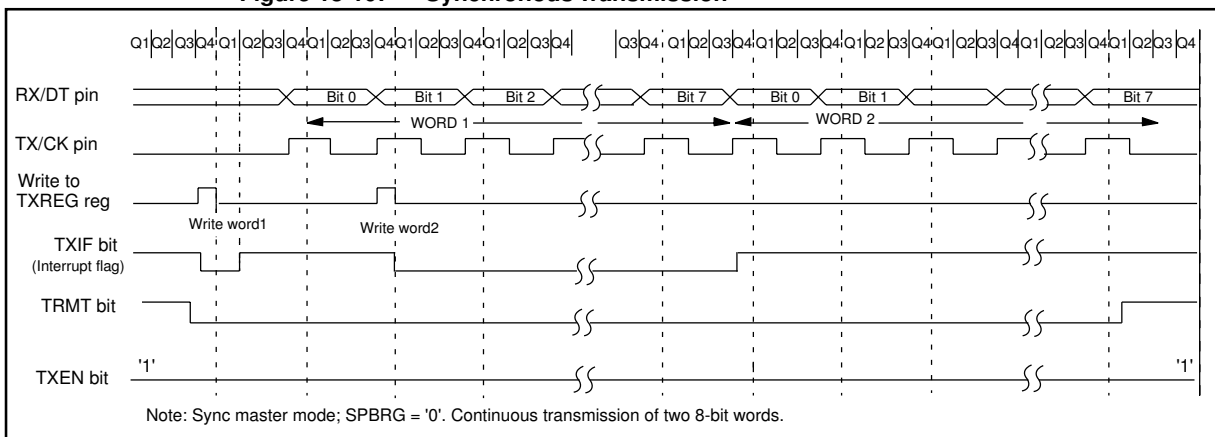
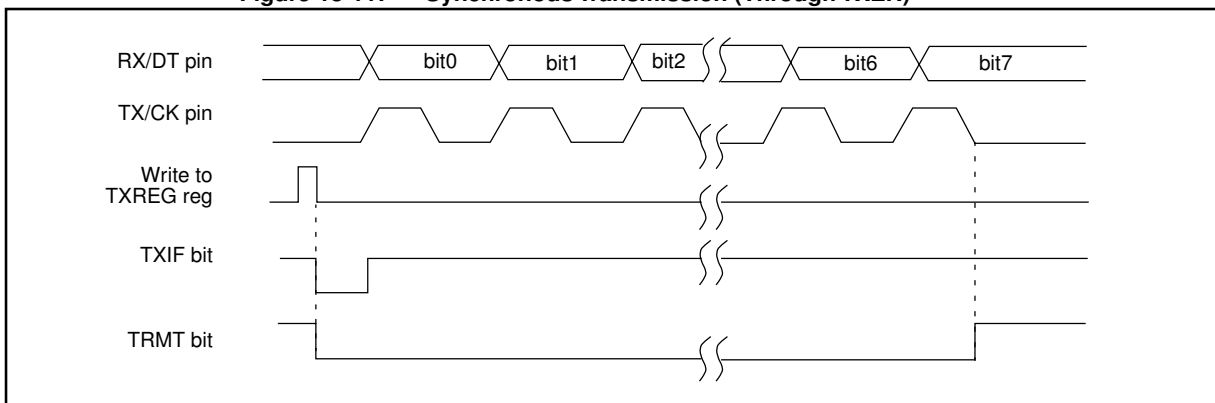
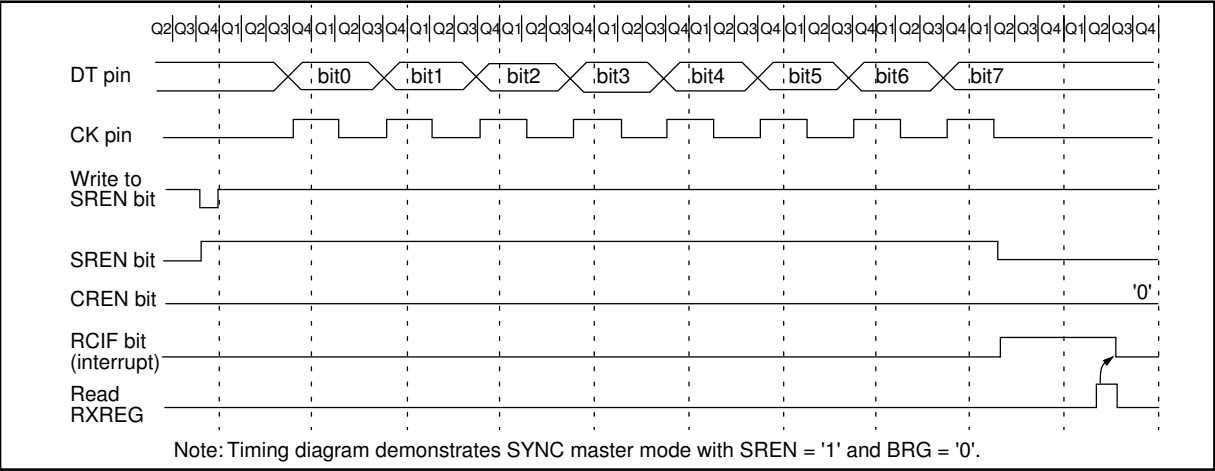


Figure 18-11: Synchronous Transmission (Through TXEN)



PICmicro MID-RANGE MCU FAMILY

Figure 18-12: Synchronous Reception (Master Mode, SREN)



PICmicro MID-RANGE MCU FAMILY

18.6.2 USART Synchronous Slave Reception

The operation of the synchronous master and slave modes is identical except in the case of the SLEEP mode. Also, bit SREN is a don't care in slave mode.

If receive is enabled, by setting the CREN bit, prior to the *SLEEP* instruction, then a word may be received during SLEEP. On completely receiving the word, the RSR register will transfer the data to the RCREG register and if the RCIE enable bit is set, the interrupt generated will wake the chip from SLEEP. If the global interrupt is enabled, the program will branch to the interrupt vector (0004h).

Steps to follow when setting up a Synchronous Slave Reception:

1. Enable the synchronous master serial port by setting the SYNC and SPEN bits and clearing the CSRC bit.
2. If interrupts are desired, then set the RCIE enable bit.
3. If 9-bit reception is desired, then set the RX9 bit.
4. To enable reception, set the CREN enable bit.
5. The RCIF bit will be set when reception is complete and an interrupt will be generated, if the RCIE bit was set.
6. Read the RCSTA register to get the ninth bit (if enabled) and determine if any error occurred during reception.
7. Read the 8-bit received data by reading the RCREG register.
8. If any error occurred, clear the error by clearing the CREN bit.

Table 18-11: Registers Associated with Synchronous Slave 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 Slave Reception.

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

PICmicro MID-RANGE MCU FAMILY

18.8 Design Tips

Question 1: *Using the Asynchronous mode I am getting a lot of transmission errors.*

Answer 1:

The most common reasons are

1. You are using the high speed mode (BRGH is set) on one of the devices which has an errata for this mode (PIC16C65/65A/73/73A/74/74A).
2. You have incorrectly calculated the value to load in to the SPBRG register
3. The sum of the baud errors for the transmitter and receiver is too high.

PICmicro MID-RANGE MCU FAMILY

18.10 Revision History

Revision A

This is the initial released revision of the USART module description.

PICmicro MID-RANGE MCU FAMILY

19.1 Introduction

The Voltage Reference module is typically used in conjunction with the Comparator module. The comparator module's inputs do not require very large drive, and therefore the drive capability of the Voltage Reference is limited.

The Voltage Reference is a 16-tap resistor ladder network that provides a selectable voltage reference. The resistor ladder is segmented to provide two ranges of VREF values and has a power-down function to conserve power when the reference is not being used. The VRCON register controls the operation of the reference as shown in Figure 19-1. The block diagram is given in Figure 19-1. Within each range, the 16 steps are monotonic (i.e. each increasing code will result in an increasing output).

Figure 19-1: Voltage Reference Block Diagram

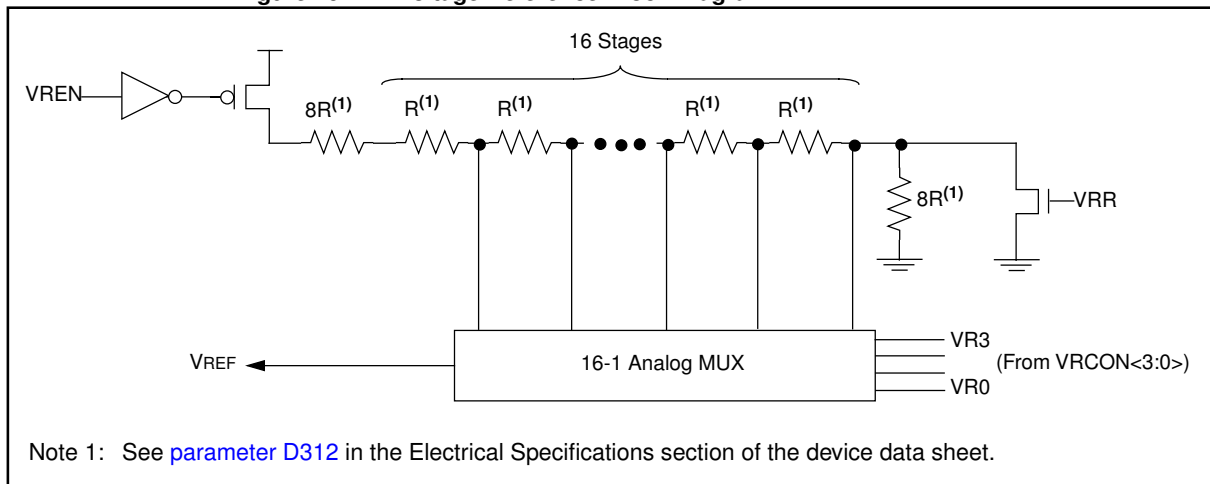


Table 19-1: Typical Voltage Reference with VDD = 5.0V

VR3:VR0	VREF	
	VRR = 1	VRR = 0
0000	0.00 V	1.25 V
0001	0.21 V	1.41 V
0010	0.42 V	1.56 V
0011	0.63 V	1.72 V
0100	0.83 V	1.88 V
0101	1.04 V	2.03 V
0110	1.25 V	2.19 V
0111	1.46 V	2.34 V
1000	1.67 V	2.50 V
1001	1.88 V	2.66 V
1010	2.08 V	2.81 V
1011	2.29 V	2.97 V
1100	2.50 V	3.13 V
1101	2.71 V	3.28 V
1110	2.92 V	3.44 V
1111	3.13 V	3.59 V

PICmicro MID-RANGE MCU FAMILY

19.3 Configuring the Voltage Reference

The Voltage Reference can output 16 distinct voltage levels for each range.

The equations used to calculate the output of the Voltage Reference are as follows:

$$\text{if } VRR = 1: V_{REF} = (VR3:VR0/24) \times V_{DD}$$

$$\text{if } VRR = 0: V_{REF} = (V_{DD} \times 1/4) + (VR3:VR0/32) \times V_{DD}$$

The settling time of the Voltage Reference must be considered when changing the V_{REF} output. [Example 19-1](#) shows an example of how to configure the Voltage Reference for an output voltage of 1.25V with $V_{DD} = 5.0V$.

Generally the V_{REF} and V_{DD} of the system will be known and you need to determine the value to load into $VR3:VR0$. [Equation 19-1](#) shows how to calculate the $VR3:VR0$ value. There will be some error since $VR3:VR0$ can only be an integer, and the V_{REF} and V_{DD} levels must be chosen so that the result is not greater than 15.

Equation 19-1: Calculating $VR3:VR0$

When $VRR = 1$

$$VR3:VR0 = \frac{V_{REF}}{V_{DD}} \times 24$$

When $VRR = 0$

$$VR3:VR0 = \frac{V_{REF} - V_{DD}/4}{V_{DD}} \times 32$$

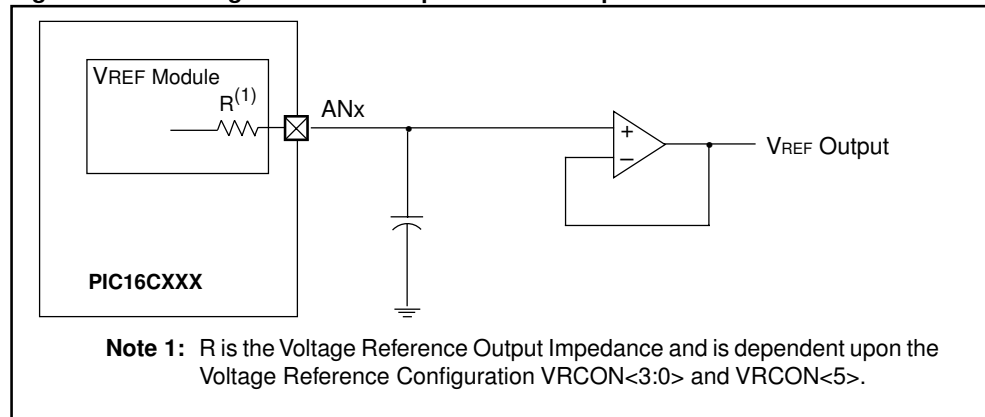
PICmicro MID-RANGE MCU FAMILY

19.7 Connection Considerations

The Voltage Reference Module operates independently of the comparator module. The output of the reference generator may be connected to the VREF pin if the corresponding TRIS bit is set and the VROE bit (VRCON<6>) is set. Enabling the Voltage Reference output onto the VREF pin with an input signal present will increase current consumption. Configuring the VREF as a digital output with VREF enabled will also increase current consumption.

The VREF pin can be used as a simple D/A output with limited drive capability. Due to the limited drive capability, a buffer must be used in conjunction with the Voltage Reference output for external connections to VREF. Figure 19-2 shows an example buffering technique.

Figure 19-2: Voltage Reference Output Buffer Example



PICmicro MID-RANGE MCU FAMILY

19.9 Design Tips

Question 1: *My VREF is not what I expect.*

Answer 1:

Any variation of the device VDD will translate directly onto the VREF pin. Also ensure that you have correctly calculated (specified) the VDD divider which generates the VREF.

Question 2: *I am connecting VREF into a low impedance circuit, and the VREF is not at the expected level.*

Answer 2:

The Voltage Reference module is not intended to drive large loads. A buffer must be used between the PICmicro's VREF pin and the load.

PICmicro MID-RANGE MCU FAMILY

19.11 Revision History

Revision A

This is the initial released revision of the Voltage Reference description.

PICmicro MID-RANGE MCU FAMILY

20.1 Introduction

The comparator module contains two analog comparators. The inputs to the comparators are multiplexed with the I/O pins. The on-chip Voltage Reference (see the “[Voltage Reference](#)” section) can also be an input to the comparators.

The CMCON register, shown in [Figure 20-1](#), controls the comparator input and output multiplexers. A block diagram of the comparator is shown in [Figure 20-1](#).

PICmicro MID-RANGE MCU FAMILY

20.3 Comparator Configuration

There are eight modes of operation for the comparators. The CMCON register is used to select the mode. [Figure 20-1](#) shows the eight possible modes. The TRIS register controls the data direction of the comparator I/O pins for each mode. If the comparator mode is changed, the comparator output level may not be valid for the new mode for the delay specified in the electrical specifications of the device.

Note: Comparator interrupts should be disabled during a comparator mode change, otherwise a false interrupt may occur.

PICmicro MID-RANGE MCU FAMILY

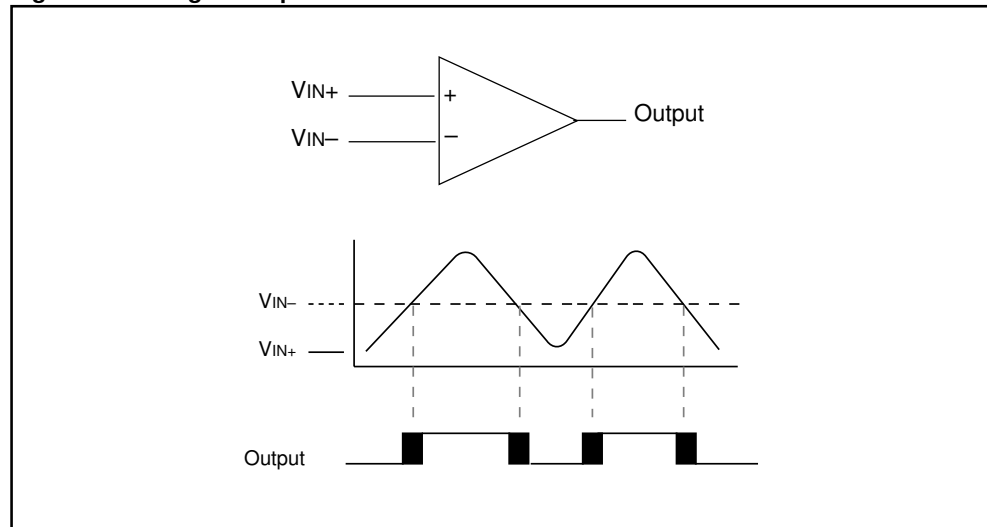
20.4 Comparator Operation

A single comparator is shown in [Figure 20-2](#) along with the relationship between the analog input levels and the digital output. When the analog input at V_{IN+} is less than the analog input V_{IN-} , the output of the comparator is a digital low level. When the analog input at V_{IN+} is greater than the analog input V_{IN-} , the output of the comparator is a digital high level. The shaded areas of the output of the comparator in [Figure 20-2](#) represent the uncertainty due to input offsets and response time.

20.5 Comparator Reference

An external or internal reference signal may be used depending on the comparator operating mode. The analog signal that is present at V_{IN-} is compared to the signal at V_{IN+} , and the digital output of the comparator is adjusted accordingly ([Figure 20-2](#)).

Figure 20-2: Single Comparator



PICmicro MID-RANGE MCU FAMILY

20.6 Comparator Response Time

Response time is the minimum time, after selecting a new reference voltage or input source, before the comparator output is guaranteed to have a valid level. If the internal reference is changed, the maximum settling time of the internal voltage reference must be considered when using the comparator outputs. Otherwise the maximum response time of the comparators should be used.

20.7 Comparator Outputs

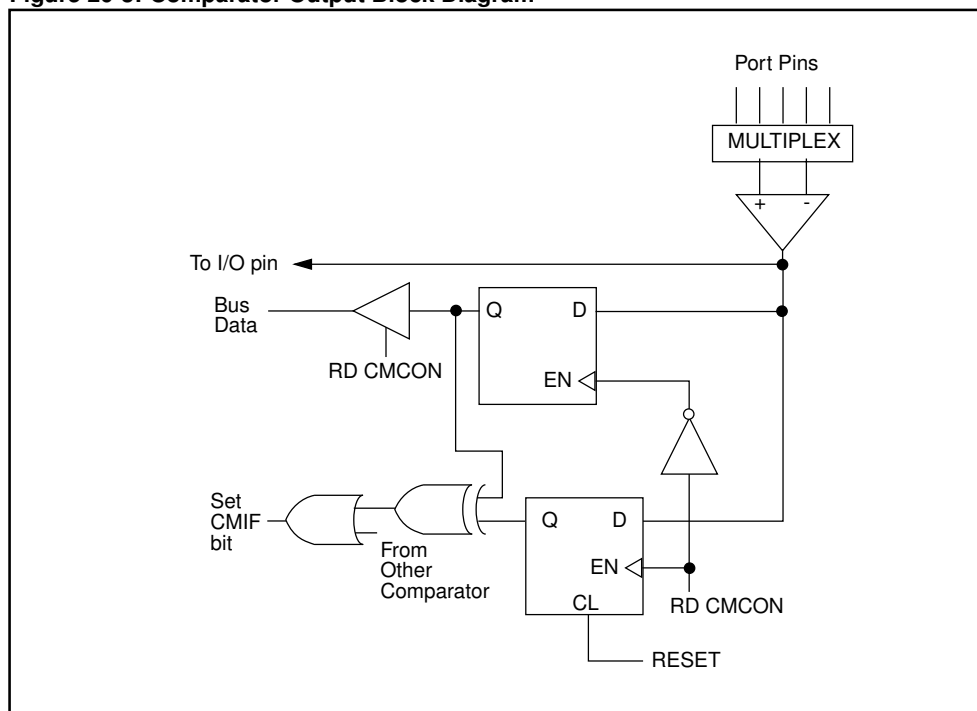
The comparator outputs are read through the CMCON register. These bits are read only. The comparator outputs may also be directly output to the I/O pins. When CM2:CM0 = 110, multiplexers in the output path of the I/O pins will switch and the output of each pin will be the unsynchronized output of the comparator. The uncertainty of each of the comparators is related to the input offset voltage and the response time given in the specifications. Figure 20-3 shows the comparator output block diagram.

The TRIS bits will still function as the output enable/disable for the I/O pins while in this mode.

Note 1: When reading the Port register, all pins configured as analog inputs will read as a '0'. Pins configured as digital inputs will convert an analog input according to the Schmitt Trigger input specification.

Note 2: Analog levels on any pin that is defined as a digital input may cause the input buffer to consume more current than is specified.

Figure 20-3: Comparator Output Block Diagram



PICmicro MID-RANGE MCU FAMILY

20.11 Analog Input Connection Considerations

A simplified circuit for an analog input is shown in [Figure 20-4](#). Since the analog pins are connected to a digital output, they have reverse biased diodes to VDD and VSS. The analog input therefore, must be between VSS and VDD. If the input voltage deviates from this range by more than 0.6V in either direction, one of the diodes is forward biased and a latch-up may occur. A maximum source impedance of 10 kΩ is recommended for the analog sources.

Figure 20-4: Analog Input Model

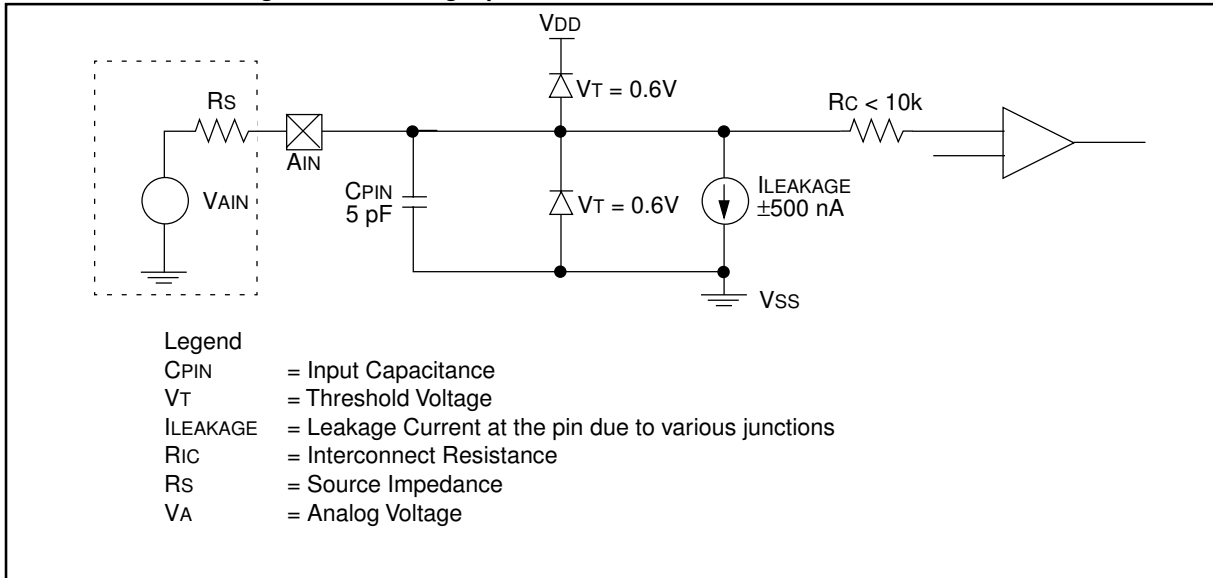


Table 20-1: Registers Associated with Comparator Module

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
CMCON	C2OUT	C1OUT	—	—	CIS	CM2	CM1	CM0	00-- 0000	00-- 0000
VRCON	VREN	VROE	VRR	—	VR3	VR2	VR1	VR0	000- 0000	000- 0000
INTCON	GIE	PEIE	TOIE	INTE	RBIE ⁽²⁾	TOIF	INTF	RBIF ⁽²⁾	0000 000x	0000 000x
PIR	CMIF ⁽¹⁾								0	0
PIE	CMIE ⁽¹⁾								0	0

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

Shaded cells are not used for Comparator Module.

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

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

PICmicro MID-RANGE MCU FAMILY

20.13 Design Tips

Question 1: *My program appears to lock up.*

Answer 1:

You may be getting stuck in an infinite loop with the comparator interrupt service routine if you did not follow the proper sequence to clear the CMIF flag bit. First you must read the CMCON register, and then you can clear the CMIF flag bit.

PICmicro MID-RANGE MCU FAMILY

20.15 Revision History

Revision A

This is the initial released revision of the Comparator module description.

PICmicro MID-RANGE MCU FAMILY

21.1 Introduction

The analog-to-digital (A/D) converter module has up to eight analog inputs.

The A/D allows conversion of an analog input signal to a corresponding 8-bit digital number. The output of the sample and hold is the input into the converter, which generates the result via successive approximation. The analog reference voltage is software selectable to either the device's positive supply voltage (V_{DD}) or the voltage level on the V_{REF} pin. The A/D converter has a unique feature of being able to operate while the device is in SLEEP mode.

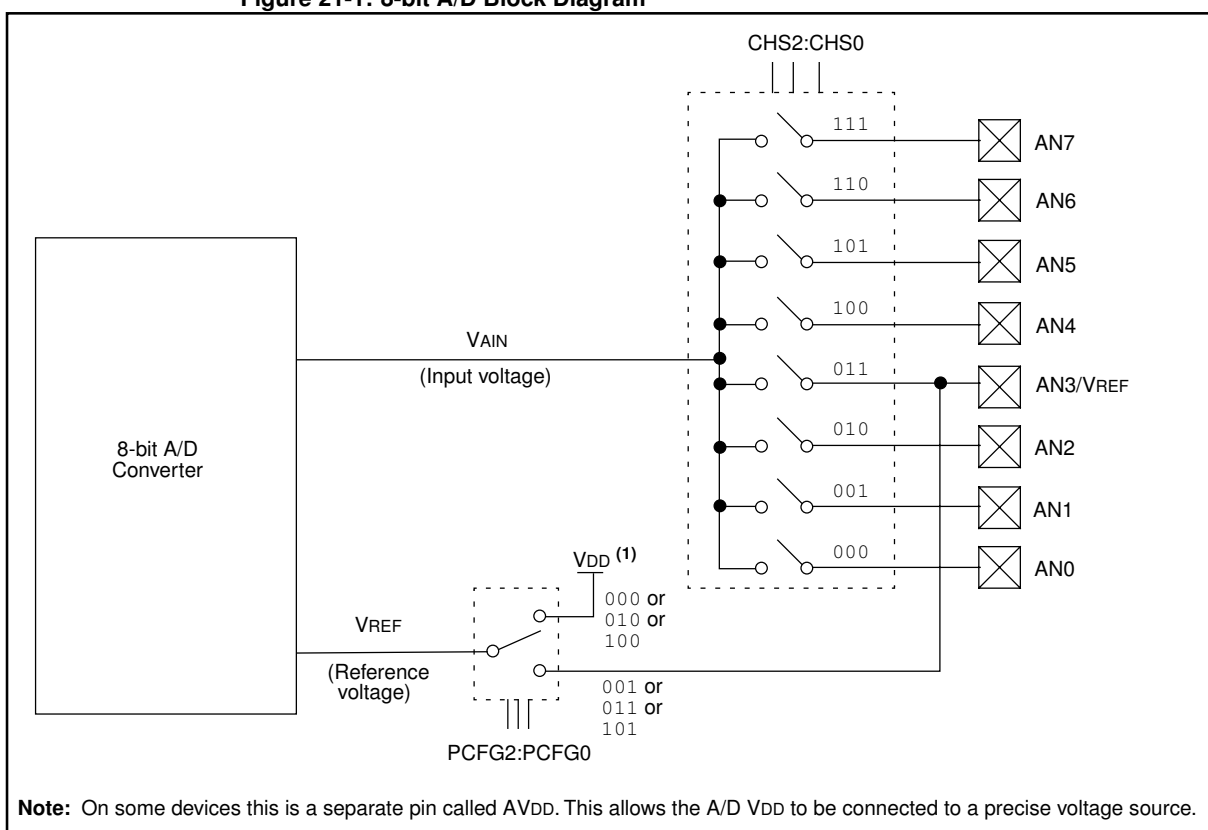
The A/D module has three registers. These registers are:

- A/D Result Register (ADRES)
- A/D Control Register0 (ADCON0)
- A/D Control Register1 (ADCON1)

The ADCON0 register, shown in [Figure 21-1](#), controls the operation of the A/D module. The ADCON1 register, shown in [Figure 21-2](#), configures the functions of the port pins. The I/O pins can be configured as analog inputs (one I/O can also be a voltage reference) or as digital I/O.

The block diagram of the A/D module is shown in [Figure 21-1](#).

Figure 21-1: 8-bit A/D Block Diagram



PICmicro MID-RANGE MCU FAMILY

Register 21-2: ADCON1 Register

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	PCFG2	PCFG1	PCFG0
bit 7					bit 0		

bit 7:3 **Unimplemented:** Read as '0'

bit 2:0 **PCFG2:PCFG0:** A/D Port Configuration Control bits

PCFG2:PCFG0	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0
000	A	A	A	A	A	A	A	A
001	A	A	A	A	VREF	A	A	A
010	D	D	D	A	A	A	A	A
011	D	D	A	A	VREF	A	A	A
100	D	D	D	D	A	D	A	A
101	D	D	D	D	VREF	D	A	A
11x	D	D	D	D	D	D	D	D

A = Analog input

D = Digital I/O

Note: When AN3 is selected as VREF, the A/D reference is the voltage on the AN3 pin. When AN3 is selected as an analog input (A), then the voltage reference for the A/D is the device VDD.

Legend

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

- n = Value at POR reset

Note 1: On any device reset, the Port pins multiplexed with analog functions (ANx) are forced to be an analog input.

PICmicro MID-RANGE MCU FAMILY

21.4 A/D Acquisition Requirements

For the A/D converter to meet its specified accuracy, the charge holding capacitor (**CHOLD**) must be allowed to fully charge to the input channel voltage level. The analog input model is shown in [Figure 21-3](#). The source impedance (**Rs**) and the internal sampling switch (**Rss**) impedance directly affect the time required to charge the capacitor CHOLD. The sampling switch (**Rss**) impedance varies over the device voltage (**VDD**) ([Figure 21-3](#)). **The maximum recommended impedance for analog sources is 10 kΩ.** After the analog input channel is selected (changed) the acquisition must be done before the conversion can be started.

To calculate the minimum acquisition time, [Equation 21-1](#) may be used. This equation assumes that 1/2 LSB error is used (512 steps for the A/D). The 1/2 LSB error is the maximum error allowed for the A/D to meet its specified resolution.

Equation 21-1: Acquisition Time

$$\begin{aligned} \text{TACQ} &= \text{Amplifier Settling Time} + \\ &\quad \text{Holding Capacitor Charging Time} + \\ &\quad \text{Temperature Coefficient} \\ &= \text{TAMP} + \text{TC} + \text{Tcoff} \end{aligned}$$

Equation 21-2: A/D Minimum Charging Time

$$\begin{aligned} \text{V}_{\text{HOLD}} &= (\text{V}_{\text{REF}} - (\text{V}_{\text{REF}}/512)) \cdot (1 - e^{(-\text{Tc}/\text{CHOLD}(\text{RIC} + \text{RSS} + \text{RS})))} \\ \text{or} \\ \text{Tc} &= -(51.2 \text{ pF})(1 \text{ k}\Omega + \text{RSS} + \text{RS}) \ln(1/511) \end{aligned}$$

[Example 21-1](#) shows the calculation of the minimum required acquisition time TACQ. This calculation is based on the following system assumptions.

Rs	=	10 kΩ	
Conversion Error	≤	1/2 LSB	
VDD	=	5V → Rss = 7 kΩ	(see graph in Figure 21-3)
Temperature	=	50°C (system max.)	
VHOLD	=	0V @ time = 0	

Example 21-1: Calculating the Minimum Required Acquisition Time

$$\begin{aligned} \text{TACQ} &= \text{TAMP} + \text{TC} + \text{Tcoff} \\ \text{TACQ} &= 5 \mu\text{s} + \text{Tc} + [(50^\circ\text{C} - 25^\circ\text{C})(0.05 \mu\text{s}/^\circ\text{C})] \\ \text{Tc} &= -\text{CHOLD} (\text{RIC} + \text{RSS} + \text{RS}) \ln(1/512) \\ &\quad -51.2 \text{ pF} (1 \text{ k}\Omega + 7 \text{ k}\Omega + 10 \text{ k}\Omega) \ln(0.0020) \\ &\quad -51.2 \text{ pF} (18 \text{ k}\Omega) \ln(0.0020) \\ &\quad -0.921 \mu\text{s} (-6.2146) \\ &\quad 5.724 \mu\text{s} \\ \text{TACQ} &= 5 \mu\text{s} + 5.724 \mu\text{s} + [(50^\circ\text{C} - 25^\circ\text{C})(0.05 \mu\text{s}/^\circ\text{C})] \\ &\quad 10.724 \mu\text{s} + 1.25 \mu\text{s} \\ &\quad 11.974 \mu\text{s} \end{aligned}$$

PICmicro MID-RANGE MCU FAMILY

21.5 Selecting the A/D Conversion Clock

The A/D conversion time per bit is defined as TAD. The A/D conversion requires 9.5 TAD per 8-bit conversion. The source of the A/D conversion clock is software selected. The four possible options for TAD are:

- 2TOSC
- 8TOSC
- 32TOSC
- Internal RC oscillator

For correct A/D conversions, the A/D conversion clock (TAD) must be selected to ensure a minimum TAD time of 1.6 μ s for all devices, as shown in [parameter 130](#) of the devices electrical specifications.

[Table 21-1](#) and [Table 21-2](#) show the resultant TAD times derived from the device operating frequencies and the A/D clock source selected.

Table 21-1: TAD vs. Device Operating Frequencies (for Standard, C, Devices)

AD Clock Source (TAD)		Device Frequency			
Operation	ADCS1:ADCS0	20 MHz	5 MHz	1.25 MHz	333.33 kHz
2TOSC	00	100 ns ⁽²⁾	400 ns ⁽²⁾	1.6 μ s	6 μ s
8TOSC	01	400 ns ⁽²⁾	1.6 μ s	6.4 μ s	24 μ s ⁽³⁾
32TOSC	10	1.6 μ s	6.4 μ s	25.6 μ s ⁽³⁾	96 μ s ⁽³⁾
RC	11	2 - 6 μ s ^(1,4)	2 - 6 μ s ^(1,4)	2 - 6 μ s ^(1,4)	2 - 6 μ s ⁽¹⁾

Legend: Shaded cells are outside of recommended range.

Note 1: The RC source has a typical TAD time of 4 μ s.

2: These values violate the minimum required TAD time.

3: For faster conversion times, the selection of another clock source is recommended.

4: For device frequencies above 1 MHz, the device must be in SLEEP for the entire conversion, or the A/D accuracy may be out of specification.

Table 21-2: TAD vs. Device Operating Frequencies (for Extended, LC, Devices)

AD Clock Source (TAD)		Device Frequency			
Operation	ADCS1:ADCS0	4 MHz	2 MHz	1.25 MHz	333.33 kHz
2TOSC	00	500 ns ⁽²⁾	1.0 μ s ⁽²⁾	1.6 μ s ⁽²⁾	6 μ s
8TOSC	01	2.0 μ s ⁽²⁾	4.0 μ s	6.4 μ s	24 μ s ⁽³⁾
32TOSC	10	8.0 μ s	16.0 μ s	25.6 μ s ⁽³⁾	96 μ s ⁽³⁾
RC	11	3 - 9 μ s ^(1,4)	3 - 9 μ s ^(1,4)	3 - 9 μ s ^(1,4)	3 - 9 μ s ⁽¹⁾

Legend: Shaded cells are outside of recommended range.

Note 1: The RC source has a typical TAD time of 6 μ s.

2: These values violate the minimum required TAD time.

3: For faster conversion times, the selection of another clock source is recommended.

4: For device frequencies above 1 MHz, the device must be in SLEEP for the entire conversion, or the A/D accuracy may be out of specification.

PICmicro MID-RANGE MCU FAMILY

21.7 A/D Conversions

Example 21-2 show how to perform an A/D conversion. The I/O pins are configured as analog inputs. The analog reference (VREF) is the device VDD. The A/D interrupt is enabled, and the A/D conversion clock is FRC. The conversion is performed on the AN0 channel.

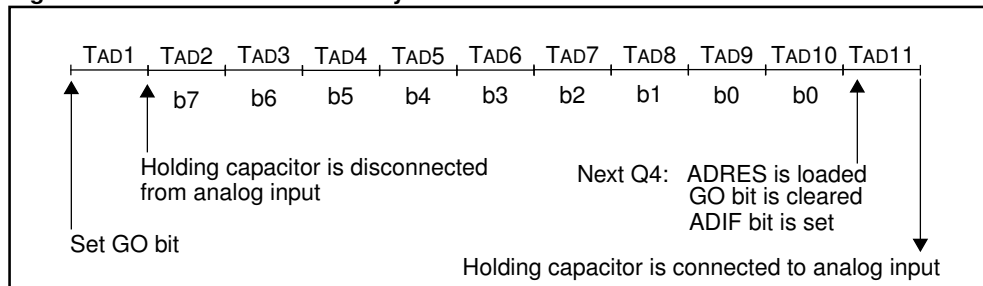
Note: The GO/DONE bit should **NOT** be set in the same instruction that turns on the A/D, due to the required acquisition time requirement.

Clearing the GO/DONE bit during a conversion will abort the current conversion. The ADRES register will NOT be updated with the partially completed A/D conversion sample. That is, the ADRES register will continue to contain the value of the last completed conversion (or the last value written to the ADRES register). After the A/D conversion is aborted, a 2TAD wait is required before the next acquisition is started. After this 2TAD wait, an acquisition is automatically started on the selected channel.

Example 21-2: Doing an A/D Conversion

```
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.
```

Figure 21-4: A/D Conversion TAD Cycles



PICmicro MID-RANGE MCU FAMILY

21.7.1 Faster Conversion - Lower Resolution Trade-off

Not all applications require a result with 8-bits of resolution, but may instead require a faster conversion time. The A/D module allows users to make the trade-off of conversion speed to resolution. Regardless of the resolution required, the acquisition time is the same. To speed up the conversion, the clock source of the A/D module may be switched so that the TAD time violates the minimum specified time (see the applicable electrical specification). Once the TAD time violates the minimum specified time, all the following A/D result bits are not valid (see A/D Conversion Timing in the Electrical Specifications section). The clock sources may only be switched between the three oscillator versions (cannot be switched from/to RC). The equation to determine the time before the oscillator can be switched is as follows:

$$\begin{aligned}\text{Conversion time} &= \text{TAD} + N \cdot \text{TAD} + (10 - N)(2\text{TOSC}) \\ \text{Where: } N &= \text{number of bits of resolution required.}\end{aligned}$$

Since the TAD is based from the device oscillator, the user must use some method (a timer, software loop, etc.) to determine when the A/D oscillator may be changed. [Example 21-3](#) shows a comparison of time required for a conversion with 4-bits of resolution, versus the 8-bit resolution conversion. The example is for devices operating at 20 MHz (The A/D clock is programmed for 32TOSC), and assumes that immediately after 5TAD, the A/D clock is programmed for 2TOSC.

The 2TOSC violates the minimum TAD time since the last 4-bits will not be converted to correct values.

Example 21-3: 4-bit vs. 8-bit Conversion Times

	Freq. (MHz) ⁽¹⁾	Resolution	
		4-bit	8-bit
TAD	20	1.6 μ s	1.6 μ s
TOSC	20	50 ns	50 ns
$\text{TAD} + N \cdot \text{TAD} + (10 - N)(2\text{TOSC})$	20	8.6 μ s	17.6 μ s

Note 1: A minimum TAD time of 1.6 μ s is required.

2: If the full 8-bit conversion is required, the A/D clock source should not be changed.

21.8 A/D Operation During Sleep

The A/D module can operate during SLEEP mode. This requires that the A/D clock source be set to RC (ADCS1:ADCS0 = 11). When the RC clock source is selected, the A/D module waits one instruction cycle before starting the conversion. This allows the SLEEP instruction to be executed, which eliminates all internal digital switching noise from the conversion. When the conversion is completed the GO/DONE bit will be cleared, and the result loaded into the ADRES register. If the A/D interrupt is enabled, the device will wake-up from SLEEP. If the A/D interrupt is not enabled, the A/D module will then be turned off (to conserve power), although the ADON bit will remain set.

When the A/D clock source is another clock option (not RC), a SLEEP instruction will cause the present conversion to be aborted and the A/D module to be turned off, though the ADON bit will remain set.

Turning off the A/D places the A/D module in its lowest current consumption state.

Note: For the A/D module to operate in SLEEP, the A/D clock source must be set to RC (ADCS1:ADCS0 = 11). To perform an A/D conversion in SLEEP, the GO/DONE bit must be set, followed by the SLEEP instruction.

PICmicro MID-RANGE MCU FAMILY

21.11 Use of the CCP Trigger

An A/D conversion may be started by the “special event trigger” of a CCP module. This requires that the CCPxM3:CCPxM0 bits (CCPxCON<3:0>) be programmed as 1011 and that the A/D module is enabled (ADON bit is set). When the trigger occurs, the GO/DONE bit will be set, starting the A/D conversion, and the Timer1 counter will be reset to zero. Timer1 is reset to automatically repeat the A/D acquisition period with minimal software overhead (moving the ADRES to the desired location). The appropriate analog input channel must be selected and the minimum acquisition done before the “special event trigger” sets the GO/DONE bit (starts a conversion).

If the A/D module is not enabled (ADON is cleared), then the “special event trigger” will be ignored by the A/D module, but will still reset the Timer1 counter.

21.12 Connection Considerations

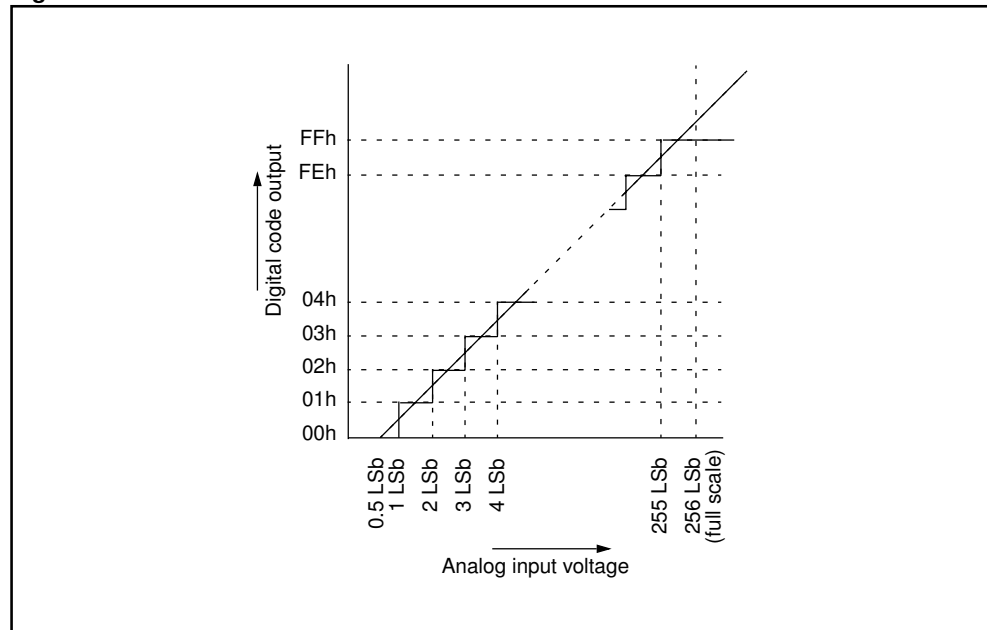
If the input voltage exceeds the rail values (VSS or VDD) by greater than 0.3V, then the accuracy of the conversion is out of specification.

An external RC filter can sometimes be added for anti-aliasing of the input signal. The R component should be selected to ensure that the total source impedance is kept under the 10 kΩ recommended specification. Any external components connected (via hi-impedance) to an analog input pin (capacitor, zener diode, etc.) should have very little leakage current at the pin.

21.13 Transfer Function

The ideal transfer function of the A/D converter is as follows: the first transition occurs when the analog input voltage (VAIN) is 1 LSb (or Analog VREF / 256) (Figure 21-6).

Figure 21-6: A/D Transfer Function



PICmicro MID-RANGE MCU FAMILY

21.15 Design Tips

Question 1: *I am using one of your PIC16C7X devices, and I find that the Analog to Digital Converter result is not always accurate. What can I do to improve accuracy?*

Answer 1:

1. Make sure you are meeting all of the timing specifications. If you are turning the A/D module off and on, there is a minimum delay you must wait before taking a sample, if you are changing input channels, there is a minimum delay you must wait for this as well, and finally there is T_{AD} , which is the time selected for each bit conversion. This is selected in ADCON0 and should be between 2 and 6 μ s. If T_{AD} is too short, the result may not be fully converted before the conversion is terminated, and if T_{AD} is made too long the voltage on the sampling capacitor can droop before the conversion is complete. These timing specifications are provided in the data book in a table or by way of a formula, and should be looked up for your specific part and circumstances.
2. Often the source impedance of the analog signal is high (greater than 1k ohms) so the current drawn from the source to charge the sample capacitor can affect accuracy. If the input signal does not change too quickly, try putting a 0.1 μ F capacitor on the analog input. This capacitor will charge to the analog voltage being sampled, and supply the instantaneous current needed to charge the 51.2 pf internal holding capacitor.
3. Finally, straight from the data book: "In systems where the device frequency is low, use of the A/D clock derived from the device oscillator is preferred...this reduces, to a large extent, the effects of digital switching noise." and "In systems where the device will enter SLEEP mode after start of A/D conversion, the RC clock source selection is required. This method gives the highest accuracy."

Question 2: *After starting an A/D conversion may I change the input channel (for my next conversion)?*

Answer 2:

After the holding capacitor is disconnected from the input channel, one T_{AD} after the GO bit is set, the input channel may be changed.

Question 3: *Do you know of a good reference on A/D's?*

Answer 3:

A very good reference for understanding A/D conversions is the "Analog-Digital Conversion Handbook" third edition, published by Prentice Hall (ISBN 0-13-03-2848-0).