Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	40MHz
Connectivity	CANbus, I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	52
Program Memory Size	32KB (16K x 16)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	1.5K x 8
Voltage - Supply (Vcc/Vdd)	2.5V ~ 5.5V
Data Converters	A/D 12x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	68-LCC (J-Lead)
Supplier Device Package	68-PLCC (24.23x24.23)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lc658-i-l

Table of Contents

TABLE 1-2: PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number				Pin Type	Buffer Type	Description
	PIC18C658		PIC18C858				
	TQFP	PLCC	TQFP	PLCC			
							PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-ups on all inputs.
RB0/INT0	48	60	58	72			
RB0					I/O	TTL	Digital I/O
INT0					I	ST	External interrupt 0
RB1/INT1	47	59	57	71			
RB1					I/O	TTL	Digital I/O
INT1					I	ST	External interrupt 1
RB2/INT2	46	58	56	70			
RB2					I/O	TTL	Digital I/O
INT2					I	ST	External interrupt 2
RB3/INT3	45	57	55	69			
RB3					I/O	TTL	Digital I/O
INT3					I/O	ST	External interrupt 3
RB4	44	56	54	68	I/O	TTL	Digital I/O Interrupt on change pin
RB5	43	55	53	67	I/O	TTL	Digital I/O Interrupt-on-change pin
RB6	42	54	52	66	I/O	TTL	Digital I/O Interrupt-on-change pin
					I	ST	ICSP programming clock
RB7	37	48	47	60	I/O	TTL	Digital I/O Interrupt-on-change pin
					I/O	ST	ICSP programming data

Legend: TTL = TTL compatible input; ST = Schmitt Trigger input with CMOS levels; I = Input; P = Power; CMOS = CMOS compatible input or output; Analog = Analog input; O = Output; OD = Open Drain (no P diode to V_{DD})

2.0 OSCILLATOR CONFIGURATIONS

2.1 Oscillator Types

The PIC18CXX8 can be operated in one of eight oscillator modes, programmable by three configuration bits (FOSC2, FOSC1, and FOSC0).

1. LP Low Power Crystal
2. XT Crystal/Resonator
3. HS High Speed Crystal/Resonator
4. HS4 High Speed Crystal/Resonator with PLL enabled
5. RC External Resistor/Capacitor
6. RCIO External Resistor/Capacitor with I/O pin enabled
7. EC External Clock
8. ECIO External Clock with I/O pin enabled

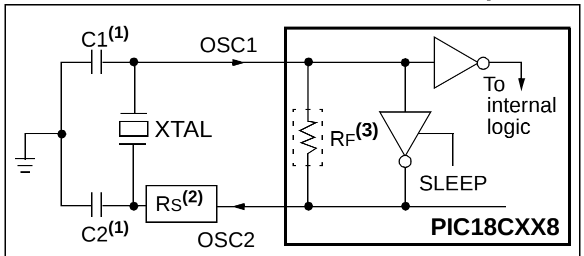
2.2 Crystal Oscillator/Ceramic Resonators

In XT, LP, HS or HS4 (PLL) oscillator modes, a crystal or ceramic resonator is connected to the OSC1 and OSC2 pins to establish oscillation. Figure 2-1 shows the pin connections. An external clock source may also be connected to the OSC1 pin, as shown in Figure 2-3 and Figure 2-4.

The PIC18CXX8 oscillator design requires the use of a parallel cut crystal.

Note:	Use of a series cut crystal may give a frequency out of the crystal manufacturer's specifications.

FIGURE 2-1: CRYSTAL/CERAMIC RESONATOR OPERATION (HS, XT OR LP OSC CONFIGURATION)



Note 1: See Table 2-1 and Table 2-2 for recommended values of C1 and C2.

2: A series resistor (RS) may be required for AT strip cut crystals.

3: RF varies with the crystal chosen.

Filename	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(3)
INDF2	Uses contents of FSR2 to address data memory - value of FSR2 not changed (not a physical register)								n/a	n/a
POSTINC2	Uses contents of FSR2 to address data memory - value of FSR2 post-incremented (not a physical register)								n/a	n/a
POSTDEC2	Uses contents of FSR2 to address data memory - value of FSR2 post-decremented (not a physical register)								n/a	n/a
PREINC2	Uses contents of FSR2 to address data memory - value of FSR2 pre-incremented (not a physical register)								n/a	n/a
PLUSW2	Uses contents of FSR2 to address data memory - value of FSR2 pre-incremented (not a physical register) - value of FSR2 offset by WREG								n/a	n/a
FSR2H	—	—	—	—	Indirect Data Memory Address Pointer 2 High				---- 0000	---- 0000
FSR2L	Indirect Data Memory Address Pointer 2 Low Byte								xxxx xxxx	uuuu uuuu
STATUS	—	—	—	N	OV	Z	DC	C	---x xxxx	---u uuuu
TMR0H	Timer0 register high byte								0000 0000	0000 0000
TMR0L	Timer0 register low byte								xxxx xxxx	uuuu uuuu
T0CON	TMR0ON	T08BIT	T0CS	T0SE	T0PS3	T0PS2	T0PS1	T0PS0	1111 1111	1111 1111
OSCCON	—	—	—	—	—	—	—	SCS	---- ---0	---- ---0
LVDCON	—	—	IRVST	LVDEN	LVDL3	LVDL2	LVDL1	LVDL0	--00 0101	--00 0101
WDTCON	—	—	—	—	—	—	—	SWDTEN	---- ---0	---- ---0
RCON	IPEN	$\overline{\text{LWRT}}$	—	$\overline{\text{RI}}$	$\overline{\text{TO}}$	$\overline{\text{PD}}$	$\overline{\text{POR}}$	$\overline{\text{BOR}}$	00-1 11qq	00-q qquu
TMR1H	Timer1 Register High Byte								xxxx xxxx	uuuu uuuu
TMR1L	Timer1 Register Low Byte								xxxx xxxx	uuuu uuuu
T1CON	RD16	—	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON	0-00 0000	u-uu uuuu
TMR2	Timer2 Register								0000 0000	0000 0000
PR2	Timer2 Period Register								1111 1111	1111 1111
T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000
SSPBUF	SSP Receive Buffer/Transmit Register								xxxx xxxx	uuuu uuuu
SSPADD	SSP Address Register in I²C Slave mode. SSP Baud Rate Reload Register in I²C Master mode.								0000 0000	0000 0000
SSPSTAT	SMP	CKE	D/$\overline{\text{A}}$	P	S	R/$\overline{\text{W}}$	UA	BF	0000 0000	0000 0000
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
ADRESH	A/D Result Register High Byte								xxxx xxxx	uuuu uuuu
ADRESL	A/D Result Register Low Byte								xxxx xxxx	uuuu uuuu
ADCON0	—	—	CHS3	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	ADON	--00 0000	--00 0000
ADCON1	—	—	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0	-000 0000	-000 0000
ADCON2	ADFM	—	—	—	—	ADCS2	ADCS1	ADCS0	0--- -000	0--- -000

Legend: x = unknown, u = unchanged, - = unimplemented, q = value depends on condition

Note 1: Bit 6 of PORTA, LATA and TRISA are enabled in ECIO and RCIO oscillator modes only. In all other oscillator modes, they are disabled and read '0'.

2: Bit 21 of the TBLPTRU allows access to the device configuration bits.

3: Other (non-power-up) RESETs include external RESET through $\overline{\text{MCLR}}$ and Watchdog Timer Reset.

4: These registers are reserved on PIC18C658.

Filename	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[(3)]
RXB0D7	RXB0D77	RXB0D76	RXB0D75	RXB0D74	RXB0D73	RXB0D72	RXB0D71	RXB0D70	xxxx xxxx	uuuu uuuu
RXB0D6	RXB0D67	RXB0D66	RXB0D65	RXB0D64	RXB0D63	RXB0D62	RXB0D61	RXB0D60	xxxx xxxx	uuuu uuuu
RXB0D5	RXB0D57	RXB0D56	RXB0D55	RXB0D54	RXB0D53	RXB0D52	RXB0D51	RXB0D50	xxxx xxxx	uuuu uuuu
RXB0D4	RXB0D47	RXB0D46	RXB0D45	RXB0D44	RXB0D43	RXB0D42	RXB0D41	RXB0D40	xxxx xxxx	uuuu uuuu
RXB0D3	RXB0D37	RXB0D36	RXB0D35	RXB0D34	RXB0D33	RXB0D32	RXB0D31	RXB0D30	xxxx xxxx	uuuu uuuu
RXB0D2	RXB0D27	RXB0D26	RXB0D25	RXB0D24	RXB0D23	RXB0D22	RXB0D21	RXB0D20	xxxx xxxx	uuuu uuuu
RXB0D1	RXB0D17	RXB0D16	RXB0D15	RXB0D14	RXB0D13	RXB0D12	RXB0D11	RXB0D10	xxxx xxxx	uuuu uuuu
RXB0D0	RXB0D07	RXB0D06	RXB0D05	RXB0D04	RXB0D03	RXB0D02	RXB0D0?	RXB0D00	xxxx xxxx	uuuu uuuu
RXB0DLC	—	RXRTR	RESB1	RESB0	DLC3	DLC2	DLC1	DLC0	0xxx xxxx	0uuu uuuu
RXB0EIDL	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	xxxx xxxx	uuuu uuuu
RXB0EIDH	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	xxxx xxxx	uuuu uuuu
RXB0SIDL	SID2	SID1	SID0	SRR	EXID	—	EID17	EID16	xxxx x-xx	uuuu u-uu
RXB0SIDH	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	xxxx xxxx	uuuu uuuu
RXB0CON	RXFUL	RXM1	RXM0	—	RXRTRRO	RXB0DBEN	JTOFF	FILHIT0	000- 0000	000- 0000
CANSTAT	OPMODE2	OPMODE1	OPMODE0	—	ICODE2	ICODE1	ICODE0	—	xxx- xxx-	uuu- uuu-
RXB1D7	RXB1D77	RXB1D76	RXB1D75	RXB1D74	RXB1D73	RXB1D72	RXB1D71	RXB1D70	xxxx xxxx	uuuu uuuu
RXB1D6	RXB1D67	RXB1D66	RXB1D65	RXB1D64	RXB1D63	RXB1D62	RXB1D61	RXB1D60	xxxx xxxx	uuuu uuuu
RXB1D5	RXB1D57	RXB1D56	RXB1D55	RXB1D54	RXB1D53	RXB1D52	RXB1D51	RXB1D50	xxxx xxxx	uuuu uuuu
RXB1D4	RXB1D47	RXB1D46	RXB1D45	RXB1D44	RXB1D43	RXB1D42	RXB1D41	RXB1D40	xxxx xxxx	uuuu uuuu
RXB1D3	RXB1D37	RXB1D36	RXB1D35	RXB1D34	RXB1D33	RXB1D32	RXB1D31	RXB1D30	xxxx xxxx	uuuu uuuu
RXB1D2	RXB1D27	RXB1D26	RXB1D25	RXB1D24	RXB1D23	RXB1D22	RXB1D21	RXB1D20	xxxx xxxx	uuuu uuuu
RXB1D1	RXB1D17	RXB1D16	RXB1D15	RXB1D14	RXB1D13	RXB1D12	RXB1D11	RXB1D10	xxxx xxxx	uuuu uuuu
RXB1D0	RXB1D07	RXB1D06	RXB1D05	RXB1D04	RXB1D03	RXB1D02	RXB1D01	RXB1D00	xxxx xxxx	uuuu uuuu
RXB1DLC	—	RXRTR	RESB1	RESB0	DLC3	DLC2	DLC1	DLC0	0xxx xxxx	0uuu uuuu
RXB1EIDL	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	xxxx xxxx	uuuu uuuu
RXB1EIDH	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	xxxx xxxx	uuuu uuuu
RXB1SIDL	SID2	SID1	SID0	SRR	EXID	—	EID17	EID16	xxxx x0xx	uuuu u0uu
RXB1SIDH	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	xxxx xxxx	uuuu uuuu
RXB1CON	RXFUL	RXM1	RXM0	—	RXRTRRO	FILHIT2	FILHIT1	FILHIT0	0000 0000	0000 0000
CANSTAT	OPMODE2	OPMODE1	OPMODE0	—	ICODE2	ICODE1	ICODE0	—	xxx- xxx-	uuu- uuu-

Legend: x = unknown, u = unchanged, - = unimplemented, q = value depends on condition

Note 1: Bit 6 of PORTA, LATA and TRISA are enabled in ECIO and RCIO oscillator modes only. In all other oscillator modes, they are disabled and read ‘0’.

2: Bit 21 of the TBLPTRU allows access to the device configuration bits.

3: Other (non-power-up) RESETs include external RESET through MCLR and Watchdog Timer Reset.

4: These registers are reserved on PIC18C658.

11.1 Timer1 Operation

Timer1 can operate in one of these modes:

- As a timer
- As a synchronous counter
- As an asynchronous counter

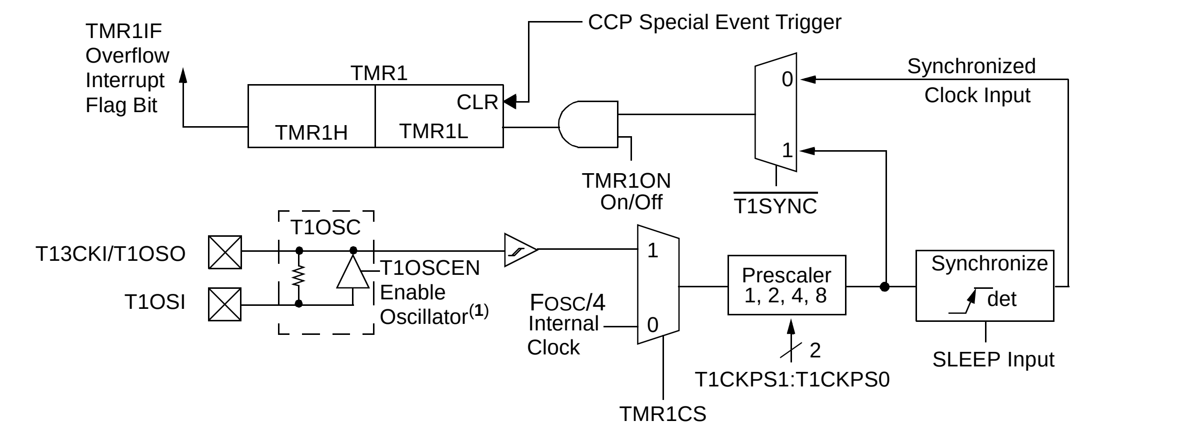
The operating mode is determined by the clock select bit, TMR1CS (T1CON register).

When TMR1CS is clear, Timer1 increments every instruction cycle. When TMR1CS is set, Timer1 increments on every rising edge of the external clock input or the Timer1 oscillator, if enabled.

When the Timer1 oscillator is enabled (T1OSCEN is set), the RC1/T1OSI and RC0/T1OSO/T1CKI pins become inputs. That is, the TRISC<1:0> value is ignored.

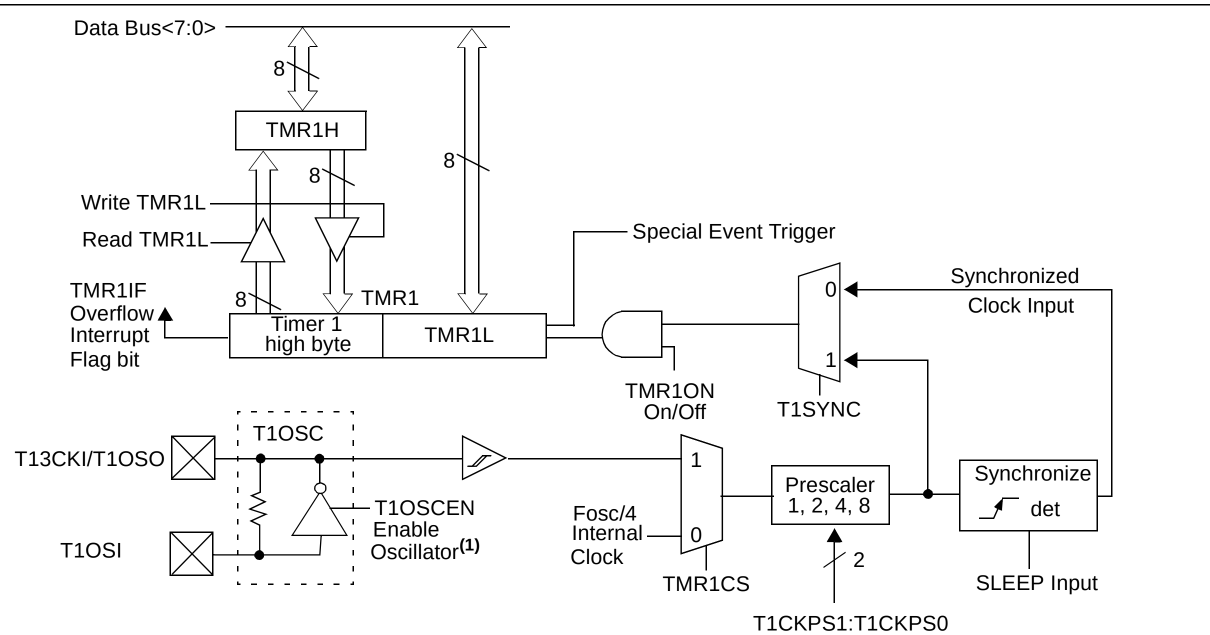
Timer1 also has an internal “RESET input”. This RESET can be generated by the CCP module (Section 14.0).

FIGURE 11-1: TIMER1 BLOCK DIAGRAM



Note 1: When enable bit T1OSCEN is cleared, the inverter and feedback resistor are turned off. This reduces power drain.

FIGURE 11-2: TIMER1 BLOCK DIAGRAM: 16-BIT READ/WRITE MODE



Note 1: When enable bit T1OSCEN is cleared, the inverter and feedback resistor are turned off. This reduces power drain.

15.2 Control Registers

The MSSP module has three associated registers. These include a status register and two control registers.

Register 15-1 shows the MSSP Status Register (SSPSTAT), Register 15-2 shows the MSSP Control Register 1 (SSPCON1), and Register 15-3 shows the MSSP Control Register 2 (SSPCON2).

REGISTER 15-1: SSPSTAT REGISTER

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

bit 7 **SMP:** Sample bit

SPI Master mode

1 = Input data sampled at end of data output time
0 = Input data sampled at middle of data output time

SPI Slave mode

SMP must be cleared when SPI is used in Slave mode

In I^2C Master or Slave mode:

1= Slew rate control disabled for standard speed mode (100 kHz and 1 MHz)
0= Slew rate control enabled for high speed mode (400 kHz)

bit 6 **CKE:** SPI Clock Edge Select

CKP = 0

1 = Data transmitted on rising edge of SCK
0 = Data transmitted on falling edge of SCK

CKP = 1

1 = Data transmitted on falling edge of SCK
0 = Data transmitted on rising edge of SCK

bit 5 **D/$\overline{\text{A}}$:** Data/$\overline{\text{Address}}$ bit (I^2C mode only)

1 = Indicates that the last byte received or transmitted was data
0 = Indicates that the last byte received or transmitted was address

bit 4 **P:** STOP bit
(I^2C mode only. This bit is cleared when the MSSP module is disabled, SSPEN is cleared.)

1 = Indicates that a STOP bit has been detected last (this bit is ’0’ on RESET)
0 = STOP bit was not detected last

bit 3 **S:** START bit
(I^2C mode only. This bit is cleared when the MSSP module is disabled, SSPEN is cleared.)

1 = Indicates that a START bit has been detected last (this bit is ’0’ on RESET)
0 = START bit was not detected last

bit 2 **R/$\overline{\text{W}}$:** Read/Write bit information (I^2C mode only)

This bit holds the R/W bit information following the last address match. This bit is only valid from the address match to the next START bit, STOP bit, or not $\overline{\text{ACK}}$ bit.

In I^2C Slave mode:

1 = Read
0 = Write

In I^2C Master mode:

1 = Transmit is in progress
0 = Transmit is not in progress
OR-ing this bit with SEN, RSEN, PEN, RCEN, or ACKEN will indicate if the MSSP is in IDLE mode.

bit 3 - 0 **SSPM3:SSPM0:** Synchronous Serial Port Mode Select bits

`0000` = SPI Master mode, clock = FOSC/4
`0001` = SPI Master mode, clock = FOSC/16
`0010` = SPI Master mode, clock = FOSC/64
`0011` = SPI Master mode, clock = TMR2 output/2
`0100` = SPI Slave mode, clock = SCK pin. $\overline{SS}$ pin control enabled.
`0101` = SPI Slave mode, clock = SCK pin. $\overline{SS}$ pin control disabled. $\overline{SS}$ can be used as I/O pin.
`0110` = I^2C Slave mode, 7-bit address
`0111` = I^2C Slave mode, 10-bit address
`1000` = I^2C Master mode, clock = FOSC / (4 * (SSPADD+1))
`1001` = Reserved
`1010` = Reserved
`1011` = I^2C firmware controlled Master mode (Slave idle)
`1100` = Reserved
`1101` = Reserved
`1110` = I^2C Slave mode, 7-bit address with START and STOP bit interrupts enabled
`1111` = I^2C Slave mode, 10-bit address with START and STOP bit interrupts enabled

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
- n = Value at POR	’1’ = Bit is set	’0’ = Bit is cleared	x = Bit is unknown

15.4.10 ACKNOWLEDGE SEQUENCE TIMING

An acknowledge sequence is enabled by setting the Acknowledge Sequence Enable bit ACKEN (SSPCON2 register). When this bit is set, the SCL pin is pulled low and the contents of the Acknowledge Data bit (ACKDT) 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 (TBRG) and the SCL pin is de-asserted (pulled high). When the SCL pin is sampled high (clock arbitration), the baud rate generator counts for TBRG. The SCL pin is then pulled low. Following this, the ACKEN bit is automatically cleared, the baud rate generator is turned off and the MSSP module then goes into IDLE mode (Figure 15-17).

15.4.10.1 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).

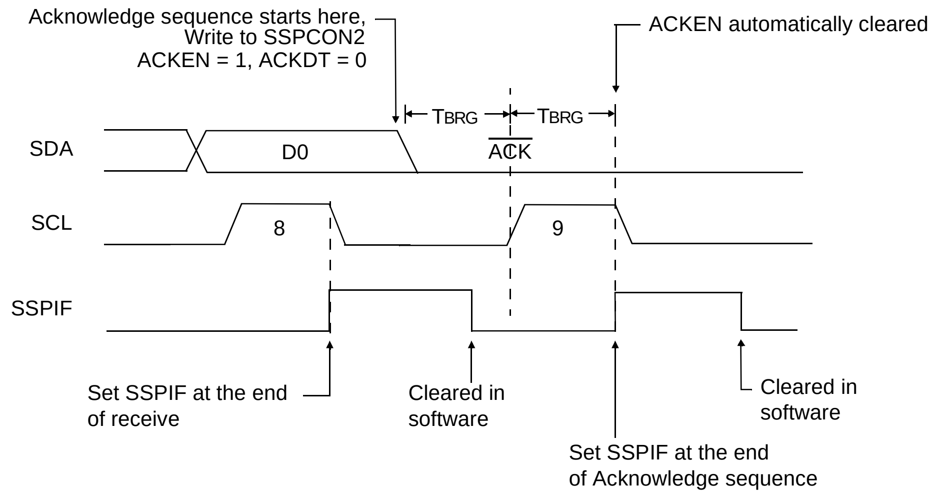
15.4.11 STOP CONDITION TIMING

A STOP bit is asserted on the SDA pin at the end of a receive/transmit by setting the Stop Sequence Enable bit, PEN (SSPCON2 register). At the end of a receive/transmit, the SCL line is held low after the falling edge of the ninth clock. When the PEN bit is set, the master will assert the SDA line low. When the SDA line is sampled low, the baud rate generator is reloaded and counts down to 0. When the baud rate generator times out, the SCL pin will be brought high, and one TBRG (baud rate generator rollover count) later, the SDA pin will be de-asserted. When the SDA pin is sampled high while SCL is high, the P bit (SSPSTAT register) is set. A TBRG later, the PEN bit is cleared and the SSPIF bit is set (Figure 15-18).

15.4.11.1 WCOL Status Flag

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

FIGURE 15-17: ACKNOWLEDGE SEQUENCE WAVEFORM



Note: TBRG = one baud rate generator period.

17.2.6 CAN MODULE I/O CONTROL REGISTER

This subsection describes the CAN Module I/O Control register.

REGISTER 17-32: CIOCON – CAN I/O CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0
TX1SRC	TX1EN	ENDRHI	CANCAP	—	—	—	—
bit 7							bit 0

bit 7 **TX1SRC**: CAN TX1 Pin Data Source
1 = CAN TX1 pin will output the CAN clock
0 = CAN TX1 pin will output $\overline{TXD}$

bit 6 **TX1EN:** CAN TX1 Pin Enable
1 = CAN TX1 pin will output $\overline{TXD}$ or CAN clock
0 = CAN TX1 pin will have digital I/O function

bit 5 **ENDRHI:** Enable Drive High
1 = CAN TX0, CAN TX1 pins will drive V_{DD} when recessive
0 = CAN TX0, CAN TX1 pins will tri-state when recessive

bit 4 **CANCAP:** CAN Message Receive Capture Enable
1 = Enable CAN capture
0 = Disable CAN capture

bit 3-0 **Unimplemented:** Read as ’0’

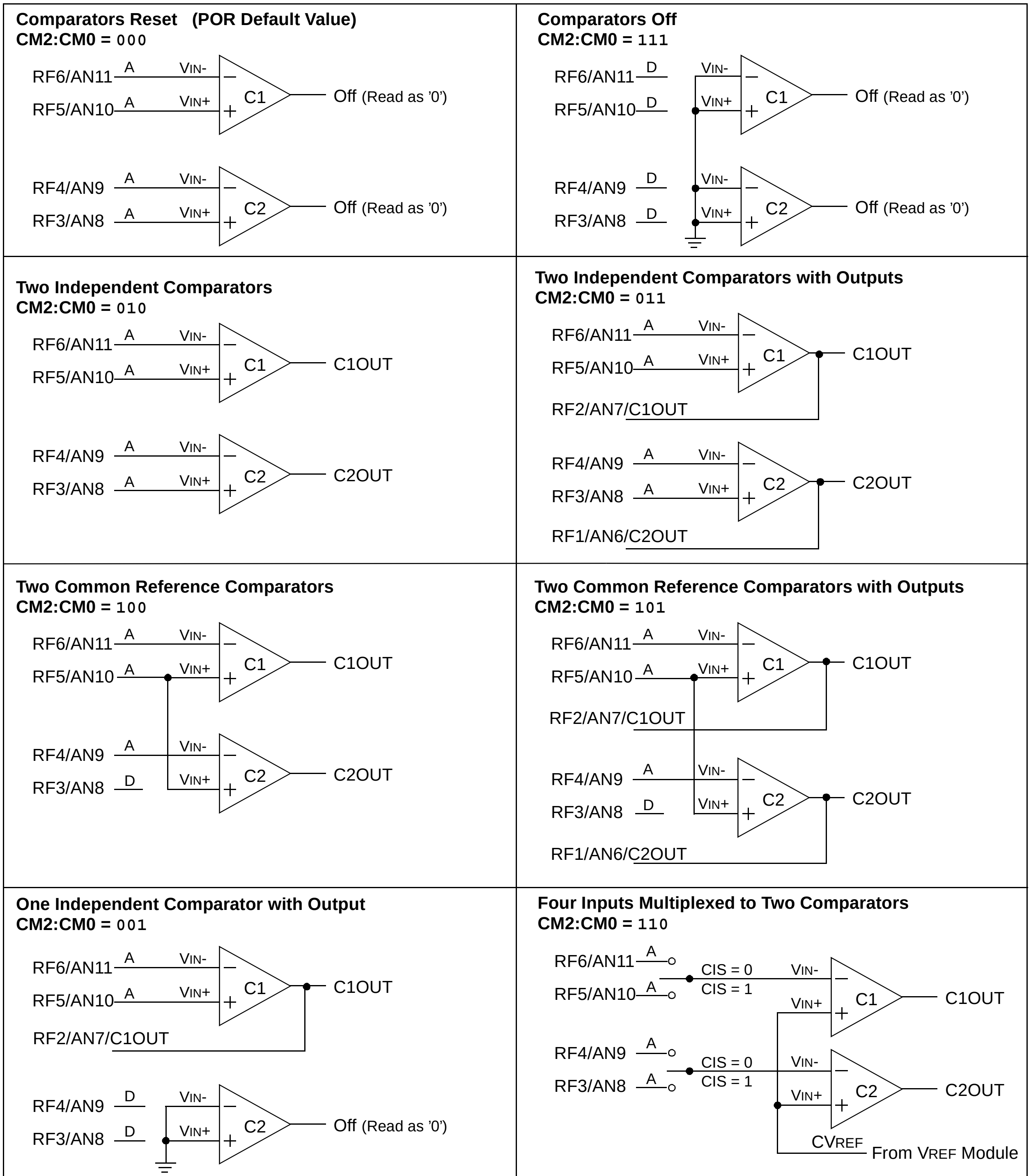
Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
- n = Value at POR	’1’ = Bit is set	’0’ = Bit is cleared	x = Bit is unknown

19.1 Comparator Configuration

There are eight modes of operation for the comparators. The CMCON register is used to select these modes. Figure 19-1 shows the eight possible modes. The TRISF register controls the data direction of the comparator pins for each mode. If the Comparator mode is changed, the comparator output level may not be valid for the specified mode change delay shown in Electrical Specifications (Section 25.0).

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

FIGURE 19-1: COMPARATOR I/O OPERATING MODES



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

20.0 COMPARATOR VOLTAGE REFERENCE MODULE

The Comparator 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 CV_{REF} values and has a power-down function to conserve power when the reference is not being used. The CVRCON register controls the operation of the reference as shown in Register 20-1. The block diagram is given in Figure 20-1.

The comparator reference supply voltage can come from either V_{DD} or V_{SS}, or the external $V_{REF}+$ and $V_{REF}-$ that are multiplexed with RA3 and RA2. The comparator reference supply voltage is controlled by the CVRSS bit.

20.1 Configuring the Comparator Voltage Reference

The Comparator Voltage Reference can output 16 distinct voltage levels for each range. The equations used to calculate the output of the Comparator Voltage Reference are as follows:

If CVRR = 1:
$CV_{REF} = (CV_R\langle 3:0\rangle/24) \times CV_{RSRC}$

If CVRR = 0:
$CV_{REF} = (CV_{DD} \times 1/4) + (CV_R\langle 3:0\rangle/32) \times CV_{RSRC}$

The settling time of the Comparator Voltage Reference must be considered when changing the CV_{REF} output (Section 25.0).

REGISTER 20-1: VRCON 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
VREN	VROE	VRR	VRSS	VR3	VR2	VR1	VR0
bit 7							bit 0

bit 7 **VREN**: Comparator Voltage Reference Enable
1 = CV_{REF} circuit powered on
0 = CV_{REF} circuit powered down

bit 6 **VROE**: Comparator V_{REF} Output Enable
1 = CV_{REF} voltage level is also output on the RF5/AN10/CV_{REF} pin
0 = CV_{REF} voltage is disconnected from the RF5/AN10/CV_{REF} pin

bit 5 **VRR**: Comparator V_{REF} Range Selection
1 = 0.00 CV_{RSRC} to 0.75 CV_{RSRC}, with $CV_{RSRC}/24$ step size
0 = 0.25 CV_{RSRC} to 0.75 CV_{RSRC}, with $CV_{RSRC}/32$ step size

bit 4 **VRSS**: Comparator V_{REF} Source Selection
1 = Comparator reference source $CV_{RSRC} = V_{REF}+ - V_{REF}-$
0 = Comparator reference source $CV_{RSRC} = V_{DD} - V_{SS}$

bit 3-0 **VR3:VR0:** Comparator V_{REF} Value Selection $0 \leq VR3:VR0 \leq 15$
When VRR = 1:
$CV_{REF} = (VR\langle 3:0\rangle/ 24) \bullet (CV_{RSRC})$
When VRR = 0:
$CV_{REF} = 1/4 \bullet (CV_{RSRC}) + (VR3:VR0/ 32) \bullet (CV_{RSRC})$

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
- n = Value at POR	’1’ = Bit is set	’0’ = Bit is cleared	x = Bit is unknown

21.2.1 REFERENCE VOLTAGE SET POINT

The Internal Reference Voltage of the LVD module may be used by other internal circuitry (the programmable Brown-out Reset). If these circuits are disabled (lower current consumption), the reference voltage circuit requires time to become stable before a low voltage condition can be reliably detected. This time is invariant of system clock speed. This start-up time is specified in electrical specification parameter #36. The low voltage interrupt flag will not be enabled until a stable reference voltage is reached. Refer to the waveform in Figure 21-3.

21.2.2 CURRENT CONSUMPTION

When the module is enabled, the LVD comparator and voltage divider are enabled and will consume static current. The voltage divider can be tapped from multiple places in the resistor array. Total current consumption, when enabled, is specified in electrical specification parameter #D022B.

21.3 External Analog Voltage Input

The LVD module has an additional feature that allows the user to supply the trip point voltage to the module from an external source (the LVDIN pin). The LVDIN pin is used as the trip point when the LVDL3:LVDL0 bits = '`1111`'. This state connects the LVDIN pin voltage to the comparator. The other comparator input is connected to an internal reference voltage source.

21.4 Operation During SLEEP

When enabled, the LVD circuitry continues to operate during SLEEP. If the device voltage crosses the trip point, the LVDIF bit will be set and the device will wake-up from SLEEP. Device execution will continue from the interrupt vector address if interrupts have been globally enabled.

21.5 Effects of a RESET

A device RESET forces all registers to their RESET state. This forces the LVD module to be turned off.

22.4 Program Verification/Code Protection

If the code protection bit(s) have not been programmed, the on-chip program memory can be read out for verification purposes.

Note:	Microchip Technology does not recommend code protecting windowed devices.

22.5 ID Locations

Five memory locations (200000h - 200004h) are designated as ID locations, where the user can store checksum or other code identification numbers. These locations are accessible during normal execution through the `TBLRD` instruction, or during program/verify. The ID locations can be read when the device is code protected.

22.6 In-Circuit Serial Programming

PIC18CXX8 microcontrollers can be serially programmed while in the end application circuit. This is simply done with two lines for clock and data, and three other lines for power, ground and the programming voltage. This allows customers to manufacture boards with unprogrammed devices, and then program the microcontroller just before shipping the product. This also allows the most recent firmware or a custom firmware to be programmed.

22.7 Device ID Bits

Device ID bits are located in program memory at 3FFFFEh and 3FFFFFh. The Device ID bits are used by programmers to retrieve part number and revision information about a device. These registers may also be accessed using a `TBLRD` instruction (Register 22-8 and Register 22-7).

REGISTER 22-7: DEVID1 ID REGISTER FOR THE PIC18CXX8 DEVICE (0x3FFFFE)

R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
DEV2	DEV1	DEV0	REV4	REV3	REV2	REV1	REV0
bit 7							bit 0

bit 7-5 **DEV2:DEV0**: Device ID bits
These bits are used with the DEV10:DEV3 bits in the Device ID register 2 to identify the part number

bit 4-0 **REV4:REV0**: Revision ID bits
These bits are used to indicate the revision of the device

Legend:	
R = Readable bit	P = Programmable bit
U = Unimplemented bit, read as ‘0’	- n = Unprogrammed Value (x = unknown)

REGISTER 22-8: DEVID2 ID REGISTER FOR THE PIC18CXX8 DEVICE (0x3FFFFF)

R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
DEV10	DEV9	DEV8	DEV7	DEV6	DEV5	DEV4	DEV3
bit 7							bit 0

bit 7-0 **DEV10:DEV3**: Device ID bits
These bits are used with the DEV2:DEV0 bits in the Device ID register 1 to identify the part number

Legend:	
R = Readable bit	P = Programmable bit
U = Unimplemented bit, read as ‘0’	- n = Unprogrammed Value (x = unknown)

RRNCF	**Rotate Right f (no carry)**
Syntax:	[*label*] RRNCF f [,d [,a]]
Operands:	$0 \leq f \leq 255$ $d \in [0,1]$ $a \in [0,1]$
Operation:	(f<n>) → dest<n-1>, (f<0>) → dest<7>
Status Affected:	N,Z
Encoding:	`0100` `00da` `ffff` `ffff`
Description:	The contents of register 'f' are rotated one bit to the right. If 'd' is 0, the result is placed in WREG. If 'd' is 1, the result is placed back in register 'f' (default). If 'a' is 0, the Access Bank will be selected, overriding the BSR value. If 'a' is 1, the Bank will be selected as per the BSR value.

Words:	1
Cycles:	1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write to destination

Example 1: `RRNCF REG`

Before Instruction
REG = `1101 0111`
N = ?
Z = ?

After Instruction
REG = `1110 1011`
N = 1
Z = 0

Example 2: `RRNCF REG, 0, 0`

Before Instruction
WREG = ?
REG = `1101 0111`
N = ?
Z = ?

After Instruction
WREG = `1110 1011`
REG = `1101 0111`
N = 1
Z = 0

SETF	**Set f**
Syntax:	[*label*] SETF f [,a]
Operands:	$0 \leq f \leq 255$ $a \in [0,1]$
Operation:	FFh → f
Status Affected:	None
Encoding:	`0110` `100a` `ffff` `ffff`
Description:	The contents of the specified register are set to FFh. If 'a' is 0, the Access Bank will be selected, overriding the BSR value. If 'a' is 1, the Bank will be selected as per the BSR value.
Words:	1
Cycles:	1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write register 'f'

Example: `SETF REG`

Before Instruction
REG = 0x5A

After Instruction
REG = 0xFF

SUBWF — Subtract WREG from f

Syntax:	[*label*] SUBWF f [,d [,a]]
Operands:	$0 \le f \le 255$ $d \in [0,1]$ $a \in [0,1]$
Operation:	(f) – (WREG) → dest
Status Affected:	N,OV, C, DC, Z
Encoding:	`0101` `11da` `ffff` `ffff`
Description:	Subtract WREG from register 'f' (2's complement method). If 'd' is 0, the result is stored in WREG. If 'd' is 1, the result is stored back in register 'f' (default). If 'a' is 0, the Access Bank will be selected, overriding the BSR value. If 'a' is 1, the Bank will be selected as per the BSR value.
Words:	1
Cycles:	1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write to destination

SUBWF — Subtract WREG from f (cont'd)

Example 1: `SUBWF REG`

Before Instruction

```
REG  = 3
WREG = 2
C    = ?
```

After Instruction

```
REG  = 1
WREG = 2
C    = 1   ; result is positive
Z    = 0
N    = 0
```

Example 2: `SUBWF REG, W`

Before Instruction

```
REG  = 2
WREG = 2
C    = ?
```

After Instruction

```
REG  = 2
WREG = 0
C    = 1   ; result is zero
Z    = 1
N    = 0
```

Example 3: `SUBWF REG`

Before Instruction

```
REG  = 1
WREG = 2
C    = ?
```

After Instruction

```
REG  = 0xFF  ;(2's complement)
WREG = 2
C    = 0   ; result is negative
Z    = 0
N    = 1
```

24.4 MPLINK Linker/MPLIB Librarian

The MPLINK object linker is a relocatable linker for the MPASM assembler and the MPLAB C17 and MPLAB C18 C compilers. It can link relocatable objects from assembly or C source files, along with pre-compiled libraries, using directives from a linker script.

The MPLIB object librarian is a librarian for pre-compiled code to be used with the MPLINK object linker. When a routine from a library is called from another source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications. The MPLIB object librarian manages the creation and modification of library files.

The MPLINK object linker features include:

- MPLINK object linker works with MPASM assembler and MPLAB C17 and MPLAB C18 C compilers.
- MPLINK object linker allows all memory areas to be defined as sections to provide link-time flexibility.

The MPLIB object librarian features include:

- MPLIB object librarian makes linking easier because single libraries can be included instead of many smaller files.
- MPLIB object librarian helps keep code maintainable by grouping related modules together.
- MPLIB object librarian commands allow libraries to be created and modules to be added, listed, replaced, deleted or extracted.

24.5 MPLAB SIM Software Simulator

The MPLAB SIM software simulator allows code development in a PC host environment by simulating the PICmicro series microcontrollers on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a file, or user-defined key press, to any of the pins. The execution can be performed in single step, execute until break, or trace mode.

The MPLAB SIM simulator fully supports symbolic debugging using the MPLAB C17 and the MPLAB C18 C compilers and the MPASM assembler. The software simulator offers the flexibility to develop and debug code outside of the laboratory environment, making it an excellent multi-project software development tool.

24.6 MPLAB ICE High Performance Universal In-Circuit Emulator with MPLAB IDE

The MPLAB ICE universal in-circuit emulator is intended to provide the product development engineer with a complete microcontroller design tool set for PICmicro microcontrollers (MCUs). Software control of the MPLAB ICE in-circuit emulator is provided by the MPLAB Integrated Development Environment (IDE), which allows editing, “make” and download and source debugging from a single environment.

Interchangeable processor modules allow the system to be easily reconfigured for emulation of different processors. The universal architecture of the MPLAB ICE in-circuit emulator allows expansion to support new PICmicro microcontrollers.

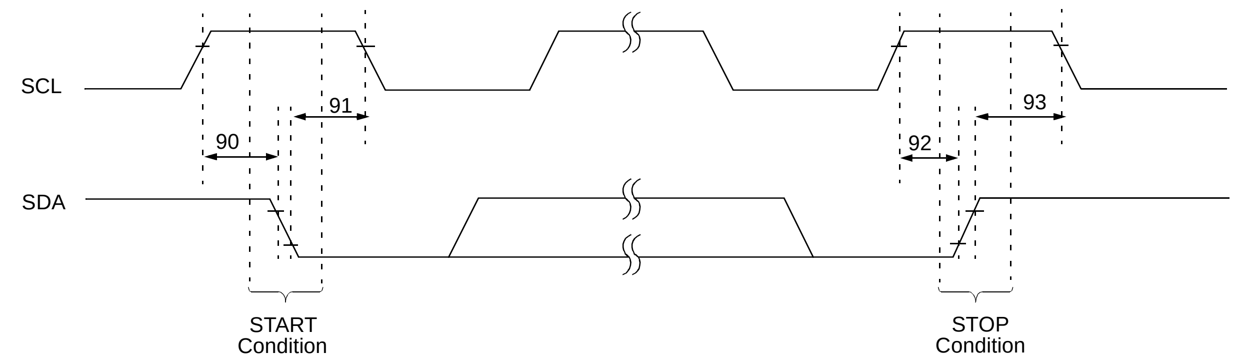
The MPLAB ICE in-circuit emulator system has been designed as a real-time emulation system, with advanced features that are generally found on more expensive development tools. The PC platform and Microsoft® Windows 3.x/95/98 environment were chosen to best make these features available to you, the end user.

The MPLAB ICE in-circuit emulator is available in two versions: MPLAB ICE 1000 and MPLAB ICE 2000. The MPLAB ICE 1000 is a basic, low cost emulator system with simple trace capabilities. The MPLAB ICE 2000 is a full featured emulator system with enhanced trace, trigger and data monitoring features. Both systems use the same processor modules and will operate across the full operating speed range of the PICmicro MCU.

24.7 ICEPIC In-Circuit Emulator

The ICEPIC low cost, in-circuit emulator is a solution for the Microchip Technology PIC16C5X, PIC16C6X, PIC16C7X and PIC16CXXX families of 8-bit One-Time-Programmable (OTP) microcontrollers. The modular system can support different subsets of PIC16C5X or PIC16CXXX products through the use of interchangeable personality modules, or daughter boards. The emulator is capable of emulating without target application circuitry being present.

FIGURE 25-18: MASTER SSP I²C BUS START/STOP BITS TIMING WAVEFORMS



Note: Refer to Figure 25-4 for load conditions.

TABLE 25-17: MASTER SSP I²C BUS START/STOP BITS REQUIREMENTS

Param. No.	Symbol	Characteristic		Min	Max	Units	Conditions
90	TSU:STA	START condition Setup time	100 kHz mode	2(TOSC)(BRG + 1)	—	ns	Only relevant for Repeated START condition
			400 kHz mode	2(TOSC)(BRG + 1)	—		
			1 MHz mode(1)	2(TOSC)(BRG + 1)	—		
91	THD:STA	START condition Hold time	100 kHz mode	2(TOSC)(BRG + 1)	—	ns	After this period, the first clock pulse is generated
			400 kHz mode	2(TOSC)(BRG + 1)	—		
			1 MHz mode(1)	2(TOSC)(BRG + 1)	—		
92	TSU:STO	STOP condition Setup time	100 kHz mode	2(TOSC)(BRG + 1)	—	ns	
			400 kHz mode	2(TOSC)(BRG + 1)	—		
			1 MHz mode(1)	2(TOSC)(BRG + 1)	—		
93	THD:STO	STOP condition Hold time	100 kHz mode	2(TOSC)(BRG + 1)	—	ns	
			400 kHz mode	2(TOSC)(BRG + 1)	—		
			1 MHz mode(1)	2(TOSC)(BRG + 1)	—		

Note 1: Maximum pin capacitance = 10 pF for all I²C pins.

W

X

READER RESPONSE

It is our intention to provide you with the best documentation possible to ensure successful use of your Microchip product. If you wish to provide your comments on organization, clarity, subject matter, and ways in which our documentation can better serve you, please FAX your comments to the Technical Publications Manager at (480) 792-4150.

Please list the following information, and use this outline to provide us with your comments about this Data Sheet.

To: Technical Publications Manager Total Pages Sent

RE: Reader Response

From: Name ________________

Company ________________

Address ________________

City / State / ZIP / Country ________________

Telephone: (_______) _________ - _________ FAX: (______) _________ - _________

Application (optional):

Would you like a reply?____Y ____N

Device: **PIC18CXX8** Literature Number: **DS30475A**

Questions:

1. What are the best features of this document?

2. How does this document meet your hardware and software development needs?

3. Do you find the organization of this data sheet easy to follow? If not, why?

4. What additions to the data sheet do you think would enhance the structure and subject?

5. What deletions from the data sheet could be made without affecting the overall usefulness?

6. Is there any incorrect or misleading information (what and where)?

7. How would you improve this document?

8. How would you improve our software, systems, and silicon products?