



Welcome to [E-XFL.COM](https://www.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	Obsolete
Core Processor	PIC
Core Size	8-Bit
Speed	4MHz
Connectivity	I ² C, SPI
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	33
Program Memory Size	3.5KB (2K x 14)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	2.5V ~ 6V
Data Converters	-
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-TQFP
Supplier Device Package	44-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lc64at-04i-pt

3.0 ARCHITECTURAL OVERVIEW

The high performance of the PIC16CXX family can be attributed to a number of architectural features commonly found in RISC microprocessors. To begin with, the PIC16CXX uses a Harvard architecture, in which, program and data are accessed from separate memories using separate buses. This improves bandwidth over traditional von Neumann architecture where program and data may be fetched from the same memory using the same bus. Separating program and data buses further allows instructions to be sized differently than 8-bit wide data words. Instruction opcodes are 14-bits wide making it possible to have all single word instructions. A 14-bit wide program memory access bus fetches a 14-bit instruction in a single cycle. A two-stage pipeline overlaps fetch and execution of instructions (Example 3-1). Consequently, all instructions execute in a single cycle (200 ns @ 20 MHz) except for program branches.

The PIC16C61 addresses 1K x 14 of program memory. The PIC16C62/62A/R62/64/64A/R64 address 2K x 14 of program memory, and the PIC16C63/R63/65/65A/R65 devices address 4K x 14 of program memory. The PIC16C66/67 address 8K x 14 program memory. All program memory is internal.

The PIC16CXX can directly or indirectly address its register files or data memory. All special function registers including the program counter are mapped in the data memory. The PIC16CXX has an orthogonal (symmetrical) instruction set that makes it possible to carry out any operation on any register using any addressing mode. This symmetrical nature and lack of "special optimal situations" makes programming with the PIC16CXX simple yet efficient, thus significantly reducing the learning curve.

The PIC16CXX device contains an 8-bit ALU and working register (W). The ALU is a general purpose arithmetic unit. It performs arithmetic and Boolean functions between data in the working register and any register file.

The ALU is 8-bits wide and capable of addition, subtraction, shift, and logical operations. Unless otherwise mentioned, arithmetic operations are two's complement in nature. In two-operand instructions, typically one operand is the working register (W register), the other operand is a file register or an immediate constant. In single operand instructions, the operand is either the W register or a file register.

The W register is an 8-bit working register used for ALU operations. It is not an addressable register.

Depending upon the instruction executed, the ALU may affect the values of the Carry (C), Digit Carry (DC), and Zero (Z) bits in the STATUS register. Bits C and DC operate as a borrow and digit borrow out bit, respectively, in subtraction. See the `SUBLW` and `SUBWF` instructions for examples.

PIC16C6X

TABLE 4-6: SPECIAL FUNCTION REGISTERS FOR THE PIC16C66/67

Address	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 ⁽³⁾
Bank 0											
00h ⁽¹⁾	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	0000 0000
01h	TMR0	Timer0 module's register								xxxx xxxx	uuuu uuuu
02h ⁽¹⁾	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	0000 0000
03h ⁽¹⁾	STATUS	IRP	RP1	RP0	\overline{TO}	\overline{PD}	Z	DC	C	0001 1xxx	000q quuu
04h ⁽¹⁾	FSR	Indirect data memory address pointer								xxxx xxxx	uuuu uuuu
05h	PORTA	—	—	PORTA Data Latch when written: PORTA pins when read						- -xx xxxx	- -uu uuuu
06h	PORTB	PORTB Data Latch when written: PORTB pins when read								xxxx xxxx	uuuu uuuu
07h	PORTC	PORTC Data Latch when written: PORTC pins when read								xxxx xxxx	uuuu uuuu
08h ⁽⁵⁾	PORTD	PORTD Data Latch when written: PORTD pins when read								xxxx xxxx	uuuu uuuu
09h ⁽⁵⁾	PORTE	—	—	—	—	—	RE2	RE1	RE0	---- -xxx	---- -uuu
0Ah ^(1,2)	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	---0 0000
0Bh ⁽¹⁾	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF ⁽⁶⁾	(4)	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
0Dh	PIR2	—	—	—	—	—	—	—	CCP2IF	---- --0	---- --0
0Eh	TMR1L	Holding register for the Least Significant Byte of the 16-bit TMR1 register								xxxx xxxx	uuuu uuuu
0Fh	TMR1H	Holding register for the Most Significant Byte of the 16-bit TMR1 register								xxxx xxxx	uuuu uuuu
10h	T1CON	—	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	--00 0000	--uu uuuu
11h	TMR2	Timer2 module's register								0000 0000	0000 0000
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000
13h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxx xxxx	uuuu uuuu
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
15h	CCPR1L	Capture/Compare/PWM1 (LSB)								xxxx xxxx	uuuu uuuu
16h	CCPR1H	Capture/Compare/PWM1 (MSB)								xxxx xxxx	uuuu uuuu
17h	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	--00 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00x
19h	TXREG	USART Transmit Data Register								0000 0000	0000 0000
1Ah	RCREG	USART Receive Data Register								0000 0000	0000 0000
1Bh	CCPR2L	Capture/Compare/PWM2 (LSB)								xxxx xxxx	uuuu uuuu
1Ch	CCPR2H	Capture/Compare/PWM2 (MSB)								xxxx xxxx	uuuu uuuu
1Dh	CCP2CON	—	—	CCP2X	CCP2Y	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	--00 0000
1Eh-1Fh	—	Unimplemented								—	—

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented location read as '0'.
Shaded locations are unimplemented, read as '0'.

Note 1: These registers can be addressed from any bank.

2: The upper byte of the Program Counter (PC) is not directly accessible. PCLATH is a holding register for the PC whose contents are transferred to the upper byte of the program counter. (PC<12:8>)

3: Other (non power-up) resets include external reset through MCLR and the Watchdog Timer reset.

4: PIE1<6> and PIR1<6> are reserved on the PIC16C66/67, always maintain these bits clear.

5: PORTD, PORTE, TRISD, and TRISE are not implemented on the PIC16C66, read as '0'.

6: PSPIF (PIR1<7>) and PSPIE (PIE1<7>) are reserved on the PIC16C66, maintain these bits clear.

4.2.2.8 PCON REGISTER

Applicable Devices													
61	62	62A	R62	63	R63	64	64A	R64	65	65A	R65	66	67

The Power Control register (PCON) contains a flag bit to allow differentiation between a Power-on Reset to an external $\overline{\text{MCLR}}$ reset or WDT reset. Those devices with brown-out detection circuitry contain an additional bit to differentiate a Brown-out Reset condition from a Power-on Reset condition.

Note: $\overline{\text{BOR}}$ is unknown on Power-on Reset. It must then be set by the user and checked on subsequent resets to see if $\overline{\text{BOR}}$ is clear, indicating a brown-out has occurred. The $\overline{\text{BOR}}$ status bit is a "don't care" and is not necessarily predictable if the brown-out circuit is disabled (by clearing the BODEN bit in the Configuration word).

FIGURE 4-22: PCON REGISTER FOR PIC16C62/64/65 (ADDRESS 8Eh)

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-q
—	—	—	—	—	—	POR	—
bit7						bit0	

R = Readable bit
W = Writable bit
U = Unimplemented bit, read as '0'
- n = Value at POR reset
q = value depends on conditions

bit 7-2: **Unimplemented:** Read as '0'

bit 1: **POR:** Power-on Reset Status bit
1 = No Power-on Reset occurred
0 = A Power-on Reset occurred (must be set in software after a Power-on Reset occurs)

bit 0: **Reserved**
This bit should be set upon a Power-on Reset by user software and maintained as set. Use of this bit as a general purpose read/write bit is not recommended, since this may affect upward compatibility with future products.

R = Readable bit
W = Writable bit
U = Unimplemented bit, read as '0'
- n = Value at POR reset
q = value depends on conditions

FIGURE 4-23: PCON REGISTER FOR PIC16C62A/R62/63/R63/64A/R64/65A/R65/66/67 (ADDRESS 8Eh)

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-q
—	—	—	—	—	—	POR	BOR
bit7						bit0	

R = Readable bit
W = Writable bit
U = Unimplemented bit, read as '0'
- n = Value at POR reset
q = value depends on conditions

bit 7-2: **Unimplemented:** Read as '0'

bit 1: **POR:** Power-on Reset Status bit
1 = No Power-on Reset occurred
0 = A Power-on Reset occurred (must be set in software after a Power-on Reset occurs)

bit 0: **BOR:** Brown-out Reset Status bit
1 = No Brown-out Reset occurred
0 = A Brown-out Reset occurred (must be set in software after a Brown-out Reset occurs)

R = Readable bit
W = Writable bit
U = Unimplemented bit, read as '0'
- n = Value at POR reset
q = value depends on conditions

Example 4-1 shows the calling of a subroutine in page 1 of the program memory. This example assumes that the PCLATH is saved and restored by the interrupt service routine (if interrupts are used).

EXAMPLE 4-1: CALL OF A SUBROUTINE IN PAGE 1 FROM PAGE 0

```
ORG 0x500
BSF    PCLATH,3    ;Select page 1 (800h-FFFh)
BCF    PCLATH,4    ;Only on >4K devices
CALL   SUB1_P1     ;Call subroutine in
:               ;page 1 (800h-FFFh)
:
:
ORG 0x900
SUB1_P1:           ;called subroutine
:               ;page 1 (800h-FFFh)
:
RETURN          ;return to Call subroutine
:               ;in page 0 (000h-7FFh)
```

4.5 Indirect Addressing, INDF and FSR Registers

Applicable Devices											
61	62	62A	R62	63	R63	64	64A	R64	65	65A	R65
66	67										

The INDF register is not a physical register. Addressing the INDF register will cause indirect addressing.

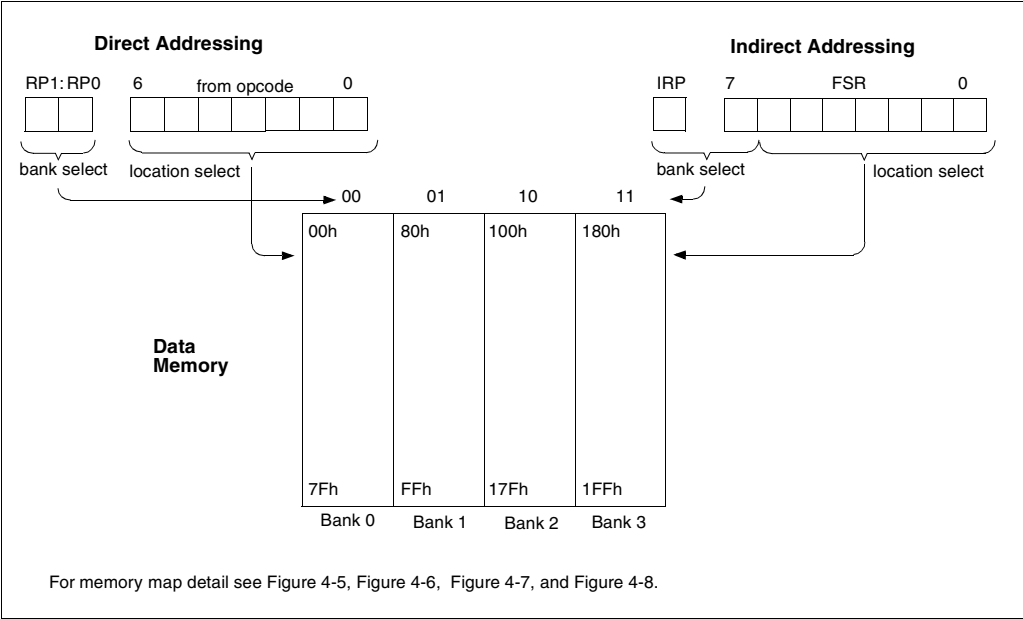
Indirect addressing is possible by using the INDF register. Any instruction using the INDF register actually accesses the register pointed to by the File Select Register, FSR. Reading the INDF register itself indirectly (FSR = '0') will produce 00h. Writing to the INDF register indirectly results in a no-operation (although status bits may be affected). An effective 9-bit address is obtained by concatenating the 8-bit FSR register and the IRP bit (STATUS<7>), as shown in Figure 4-25.

A simple program to clear RAM location 20h-2Fh using indirect addressing is shown in Example 4-2.

EXAMPLE 4-2: INDIRECT ADDRESSING

```
        movlw 0x20    ;initialize pointer
        movwf FSR     ; to RAM
NEXT     clrf  INDF    ;clear INDF register
        incf  FSR,F    ;inc pointer
        btfss FSR,4    ;all done?
        goto  NEXT    ;NO, clear next
CONTINUE
        :              ;YES, continue
```

FIGURE 4-25: DIRECT/INDIRECT ADDRESSING



7.0 TIMER0 MODULE

Applicable Devices															
61	62	62A	R62	63	R63	64	64A	R64	65	65A	R65	66	67		

The Timer0 module has the following features:

- 8-bit timer/counter register, TMR0
 - Read and write capability
 - Interrupt on overflow from FFh to 00h
- 8-bit software programmable prescaler
- Internal or external clock select
 - Edge select for external clock

Figure 7-1 is a simplified block diagram of the Timer0 module.

Timer mode is selected by clearing bit T0CS (OPTION<5>). In timer mode, the Timer0 module will increment every instruction cycle (without prescaler). If TMR0 register is written, the increment is inhibited for the following two instruction cycles (Figure 7-2 and Figure 7-3). The user can work around this by writing an adjusted value to the TMR0 register.

Counter mode is selected by setting bit T0CS. In this mode, Timer0 will increment either on every rising or falling edge of pin RA4/T0CKI. The incrementing edge is determined by the source edge select bit T0SE

(OPTION<4>). Clearing bit T0SE selects the rising edge. Restrictions on the external clock input are discussed in detail in Section 7.2.

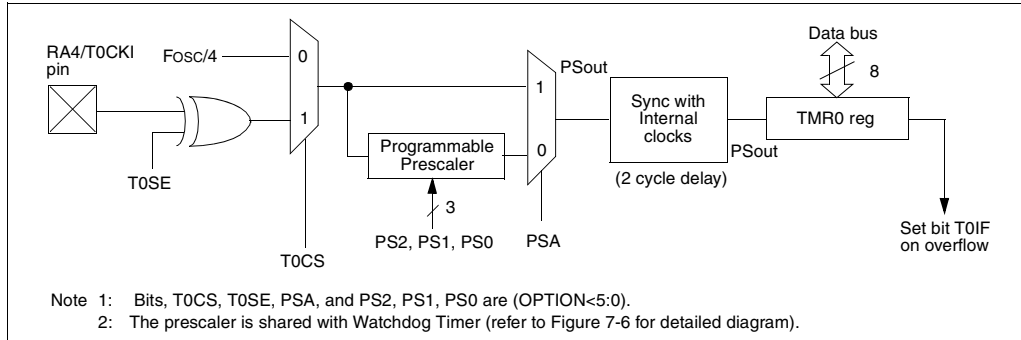
The prescaler is mutually exclusively shared between the Timer0 module and the Watchdog Timer. The prescaler assignment is controlled in software by control bit PSA (OPTION<3>). Clearing bit PSA will assign the prescaler to the Timer0 module. The prescaler is not readable or writable. When the prescaler is assigned to the Timer0 module, prescale values of 1:2, 1:4, ..., 1:256 are selectable. Section 7.3 details the operation of the prescaler.

7.1 TMR0 Interrupt

Applicable Devices															
61	62	62A	R62	63	R63	64	64A	R64	65	65A	R65	66	67		

The TMR0 interrupt is generated when the register (TMR0) overflows from FFh to 00h. This overflow sets interrupt flag bit T0IF (INTCON<2>). The interrupt can be masked by clearing enable bit T0IE (INTCON<5>). Flag bit T0IF must be cleared in software by the Timer0 interrupt service routine before re-enabling this interrupt. The TMR0 interrupt cannot wake the processor from SLEEP since the timer is shut off during SLEEP. Figure 7-4 displays the Timer0 interrupt timing.

FIGURE 7-1: TIMER0 BLOCK DIAGRAM



PIC16C6X

8.1 Timer1 Operation in Timer Mode

Applicable Devices

61	62	62A	R62	63	R63	64	64A	R64	65	65A	R65	66	67
----	----	-----	-----	----	-----	----	-----	-----	----	-----	-----	----	----

Timer mode is selected by clearing bit TMR1CS (T1CON<1>). In this mode, the input clock to the timer is $F_{osc}/4$. The synchronize control bit T1SYNC (T1CON<2>) has no effect since the internal clock is always in sync.

8.2 Timer1 Operation in Synchronized Counter Mode

Applicable Devices

61	62	62A	R62	63	R63	64	64A	R64	65	65A	R65	66	67
----	----	-----	-----	----	-----	----	-----	-----	----	-----	-----	----	----

Counter mode is selected by setting bit TMR1CS. In this mode the timer increments on every rising edge of clock input on T1OSI when enable bit T1OSCEN is set or pin with T1CKI when bit T1OSCEN is cleared.

Note: The T1OSI function is multiplexed to different pins, depending on the device. See the pinout descriptions to see which pin has the T1OSI function.

If T1SYNC is cleared, then the external clock input is synchronized with internal phase clocks. The synchronization is done after the prescaler stage. The prescaler stage is an asynchronous ripple counter.

In this configuration, during SLEEP mode, Timer1 will not increment even if an external clock is present, since the synchronization circuit is shut off. The prescaler, however, will continue to increment.

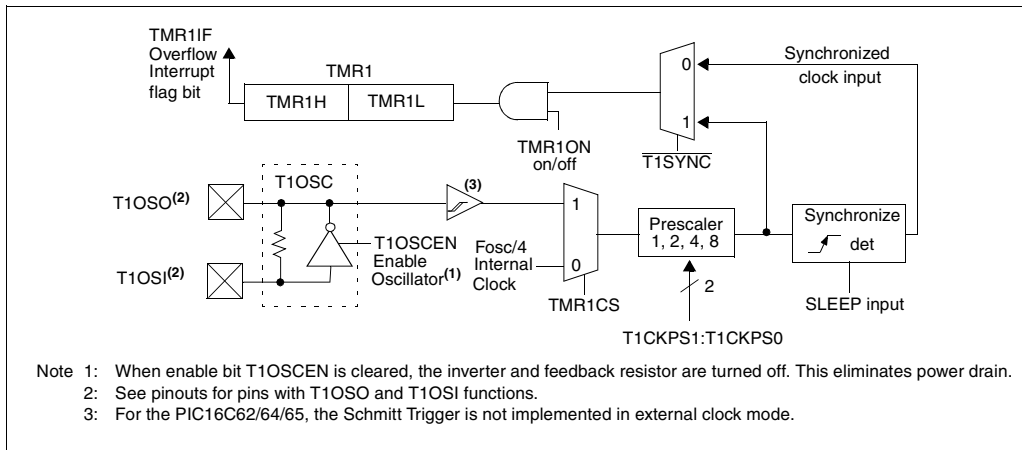
8.2.1 EXTERNAL CLOCK INPUT TIMING FOR SYNCHRONIZED COUNTER MODE

When an external clock input is used for Timer1 in synchronized counter mode, it must meet certain requirements. The external clock requirement is due to internal phase clock (Tosc) synchronization. Also, there is a delay in the actual incrementing of TMR1 after synchronization.

When the prescaler is 1:1, the external clock input is the same as the prescaler output. The synchronization of T1CKI with the internal phase clocks is accomplished by sampling the prescaler output on the Q2 and Q4 cycles of the internal phase clocks. Therefore, it is necessary for T1CKI to be high for at least $2T_{osc}$ (and a small RC delay of 20 ns) and low for at least $2T_{osc}$ (and a small RC delay of 20 ns). Refer to appropriate electrical specification section, parameters 45, 46, and 47.

When a prescaler other than 1:1 is used, the external clock input is divided by the asynchronous ripple-counter type prescaler so that the prescaler output is symmetrical. In order for the external clock to meet the sampling requirement, the ripple counter must be taken into account. Therefore, it is necessary for T1CKI to have a period of at least $4T_{osc}$ (and a small RC delay of 40 ns) divided by the prescaler value. The only requirement on T1CKI high and low time is that they do not violate the minimum pulse width requirements of 10 ns). Refer to applicable electrical specification section, parameters 40, 42, 45, 46, and 47.

FIGURE 8-2: TIMER1 BLOCK DIAGRAM



11.2.1 OPERATION OF SSP MODULE IN SPI MODE

Applicable Devices

61	62	62A	R62	63	R63	64	64A	R64	65	65A	R65	66	67
----	----	-----	-----	----	-----	----	-----	-----	----	-----	-----	----	----

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

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

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

- Slave Select (\overline{SS})

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

- Master Mode (SCK is the clock output)
- Slave Mode (SCK is the clock input)
- Clock Polarity (Output/Input data on the Rising/Falling edge of SCK)
- Clock Rate (Master mode only)
- Slave Select Mode (Slave mode only)

The SSP consists of a transmit/receive Shift Register (SSPSR) and a Buffer register (SSPBUF). The SSPSR shifts the data in and out of the device, MSb first. The SSPBUF holds the data that was written to the SSPSR, until the received data is ready. Once the 8-bits of data have been received, that byte is moved to the SSPBUF register. Then the Buffer Full bit, BF (SSPSTAT<0>) and flag bit SSPIF are set. This double buffering of the received data (SSPBUF) allows the next byte to start reception before reading the data that was just received. Any write to the SSPBUF register during transmission/reception of data will be ignored, and the write collision detect bit, WCOL (SSPCON<7>) will be set. User software must clear bit WCOL so that it can be determined if the following write(s) to the SSPBUF completed successfully. When the application software is expecting to receive valid data, the SSPBUF register should be read before the next byte of data to transfer is written to the SSPBUF register. The Buffer Full bit BF (SSPSTAT<0>) indicates when the SSPBUF register has been loaded with the received data (transmission is complete). When the SSPBUF is read, bit BF is cleared. This data may be irrelevant if the SPI is only a transmitter. Generally the SSP Interrupt is used to determine when the transmission/reception has completed. The SSPBUF register must be read and/or written. If the interrupt method is not going to be used, then software polling can be done to ensure that a write collision does not occur. Example 11-1 shows the loading of the SSPBUF (SSPSR) register for data transmission. The shaded instruction is only required if the received data is meaningful.

EXAMPLE 11-1: LOADING THE SSPBUF (SSPSR) REGISTER

```

BSF    STATUS, RP0    ;Specify Bank 1
LOOP  BTFS    SSPSTAT, BF    ;Has data been
                                ;received
                                ;(transmit
                                ;complete)?

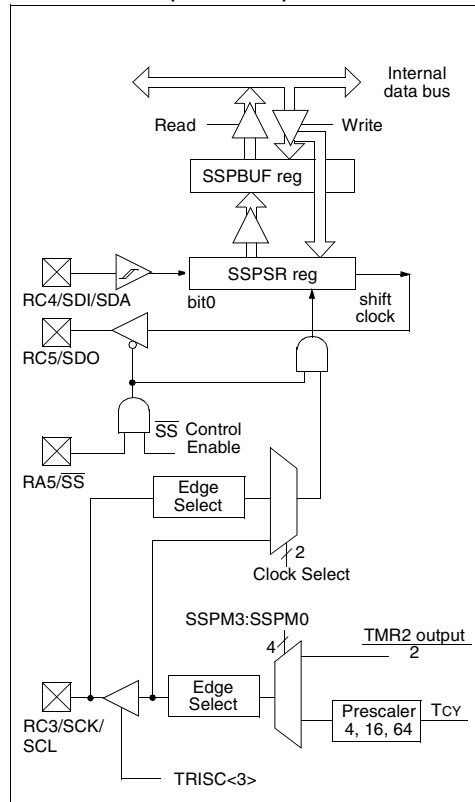
GOTO   LOOP            ;No

BCF    STATUS, RP0    ;Specify Bank 0
MOVF   SSPBUF, W       ;W reg = contents
                                ;of SSPBUF
MOVWF  RXDATA          ;Save in user RAM
MOVF   TXDATA, W       ;W reg = contents
                                ; of TXDATA
MOVWF  SSPBUF          ;New data to xmit

```

The block diagram of the SSP module, when in SPI mode (Figure 11-3), shows that the SSPSR register is not directly readable or writable, and can only be accessed from addressing the SSPBUF register. Additionally, the SSP status register (SSPSTAT) indicates the various status conditions.

FIGURE 11-3: SSP BLOCK DIAGRAM (SPI MODE)



12.4 USART Synchronous Slave Mode

Applicable Devices

61	62	62A	R62	63	R63	64	64A	R64	65	65A	R65	66	67
----	----	-----	-----	----	-----	----	-----	-----	----	-----	-----	----	----

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

12.4.1 USART SYNCHRONOUS SLAVE TRANSMIT

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

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

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

Steps to follow when setting up Synchronous Slave Transmission:

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

12.4.2 USART SYNCHRONOUS SLAVE RECEPTION

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

If receive is enabled by setting bit CREN 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 enable bit RCIE 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:

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

PIC16C6X

NOTES:

13.2 Oscillator Configurations

Applicable Devices

61	62	62A	R62	63	R63	64	64A	R64	65	65A	R65	66	67
----	----	-----	-----	----	-----	----	-----	-----	----	-----	-----	----	----

13.2.1 OSCILLATOR TYPES

The PIC16CXX can be operated in four different oscillator modes. The user can program two configuration bits (FOSC1 and FOSC0) to select one of these four modes:

- LP Low Power Crystal
- XT Crystal/Resonator
- HS High Speed Crystal/Resonator
- RC Resistor/Capacitor

13.2.2 CRYSTAL OSCILLATOR/CERAMIC RESONATORS

In LP, XT, or HS modes a crystal or ceramic resonator is connected to the OSC1/CLKIN and OSC2/CLKOUT pins to establish oscillation (Figure 13-4). The PIC16CXX oscillator design requires the use of a parallel cut crystal. Use of a series cut crystal may give a frequency out of the crystal manufacturers specifications. When in LP, XT, or HS modes, the device can have an external clock source to drive the OSC1/CLKIN pin (Figure 13-5).

FIGURE 13-4: CRYSTAL/CERAMIC RESONATOR OPERATION (HS, XT OR LP OSC CONFIGURATION)

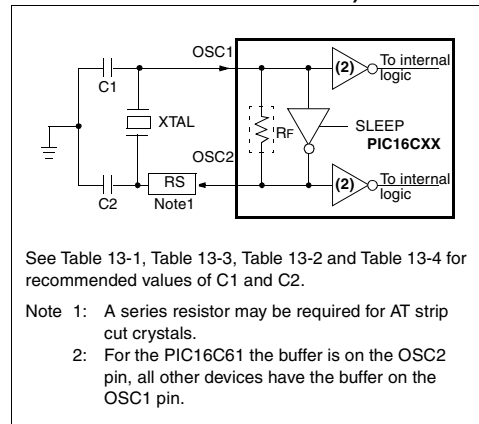
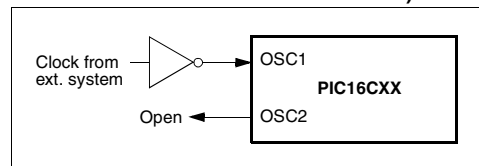
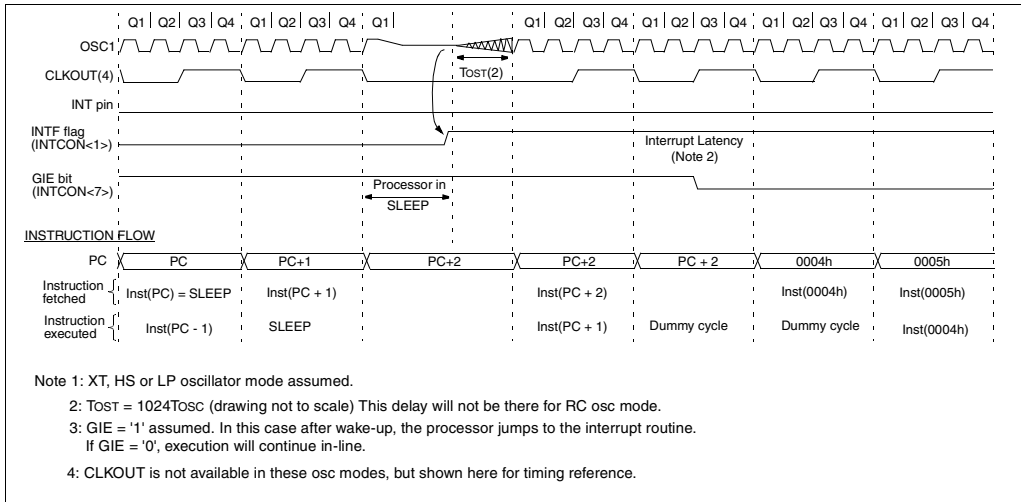


FIGURE 13-5: EXTERNAL CLOCK INPUT OPERATION (HS, XT OR LP OSC CONFIGURATION)



PIC16C6X

FIGURE 13-22: WAKE-UP FROM SLEEP THROUGH INTERRUPT



13.9 Program Verification/Code Protection

Applicable Devices

61 62 62A R62 63 R63 64 64A R64 65 65A R65 66 67

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

Note: Microchip does not recommend code protecting windowed devices.

13.10 ID Locations

Applicable Devices

61 62 62A R62 63 R63 64 64A R64 65 65A R65 66 67

Four memory locations (2000h - 2003h) are designated as ID locations where the user can store checksum or other code-identification numbers. These locations are not accessible during normal execution but are readable and writable during program/verify. It is recommended that only the 4 least significant bits of the ID location are used.

For ROM devices, these values are submitted along with the ROM code.

13.11 In-Circuit Serial Programming

Applicable Devices

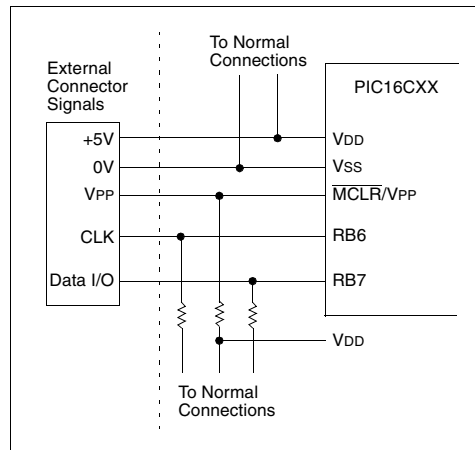
61 62 62A R62 63 R63 64 64A R64 65 65A R65 66 67

The PIC16CXX 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.

The device is placed into a program/verify mode by holding pins RB6 and RB7 low while raising the MCLR (VPP) pin from V_{IL} to V_{IH} (see programming specification). RB6 becomes the programming clock and RB7 becomes the programming data. Both RB6 and RB7 are Schmitt Trigger inputs in this mode.

After reset, to place the device in program/verify mode, the program counter (PC) is at location 00h. A 6-bit command is then supplied to the device. Depending on the command, 14-bits of program data are then supplied to or from the device, depending if the command was a load or a read. For complete details of serial programming, please refer to the PIC16C6X/7X Programming Specifications (Literature #DS30228).

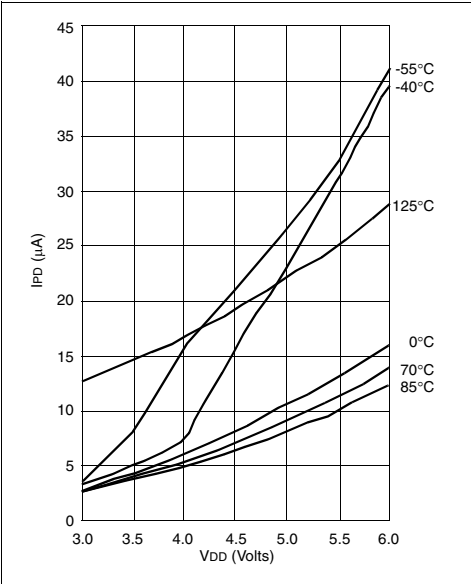
FIGURE 13-23: TYPICAL IN-CIRCUIT SERIAL PROGRAMMING CONNECTION



PIC16C6X

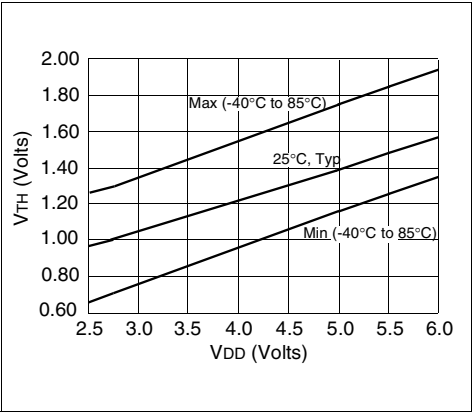
Applicable Devices 61 62 62A R62 63 R63 64 64A R64 65 65A R65 66 67

FIGURE 16-8: MAXIMUM IPD VS. VDD
WATCHDOG ENABLED*



*IPD, with Watchdog Timer enabled, has two components: The leakage current which increases with higher temperature and the operating current of the Watchdog Timer logic which increases with lower temperature. At -40°C, the latter dominates explaining the apparently anomalous behavior.

FIGURE 16-9: VTH (INPUT THRESHOLD
VOLTAGE) OF I/O PINS vs.
VDD



Data based on matrix samples. See first page of this section for details.

PIC16C6X

Applicable Devices 61 62 62A R62 63 R63 64 64A R64 65 65A R65 66 67

18.1 DC Characteristics: PIC16C62A/R62/64A/R64-04 (Commercial, Industrial, Extended) PIC16C62A/R62/64A/R64-10 (Commercial, Industrial, Extended) PIC16C62A/R62/64A/R64-20 (Commercial, Industrial, Extended)

DC CHARACTERISTICS		Standard Operating Conditions (unless otherwise stated)						Conditions
		Operating temperature -40°C ≤ TA ≤ +125°C for extended, -40°C ≤ TA ≤ +85°C for industrial and 0°C ≤ TA ≤ +70°C for commercial						
Param No.	Characteristic	Sym	Min	Typ†	Max	Units		
D001 D001A	Supply Voltage	VDD	4.0 4.5	- -	6.0 5.5	V V		
D002*	RAM Data Retention Voltage (Note 1)	VDR	-	1.5	-	V		
D003	VDD start voltage to ensure internal Power-on Reset signal	VPOR	-	VSS	-	V		
D004*	VDD rise rate to ensure internal Power-on Reset signal	SVDD	0.05	-	-	V/ms		
D005	Brown-out Reset Voltage	BVDD	3.7 3.7	4.0 4.0	4.3 4.4	V V		
D010 D013	Supply Current (Note 2, 5)	IDD	- -	2.7 10	5 20	mA mA		
D015*	Brown-out Reset Current (Note 6)	Δ IBOR	-	350	425	μA		
D020 D021 D021A D021B	Power-down Current (Note 3, 5)	IPD	- - - -	10.5 1.5 1.5 2.5	42 16 19 19	μA μA μA μA		
D023*	Brown-out Reset Current (Note 6)	Δ IBOR	-	350	425	μA		

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: This is the limit to which VDD can be lowered without losing RAM data.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors such as I/O pin loading and switching rate, oscillator type, internal code execution pattern, and temperature also have an impact on the current consumption.

The test conditions for all IDD measurements in active operation mode are:

OSC1 = external square wave, from rail to rail; all I/O pins tristated, pulled to VDD

MCLR = VDD; WDT enabled/disabled as specified.

3: The power-down current in SLEEP mode does not depend on the oscillator type. Power-down current is measured with the part in SLEEP mode, with all I/O pins in hi-impedance state and tied to VDD and VSS.

4: For RC osc configuration, current through Rext is not included. The current through the resistor can be estimated by the formula $I_r = VDD/2R_{ext}$ (mA) with Rext in kOhm.

5: Timer1 oscillator (when enabled) adds approximately 20 μA to the specification. This value is from characterization and is for design guidance only. This is not tested.

6: The Δ current is the additional current consumed when this peripheral is enabled. This current should be added to the base IDD or IPD measurement.

18.5 Timing Diagrams and Specifications

FIGURE 18-2: EXTERNAL CLOCK TIMING

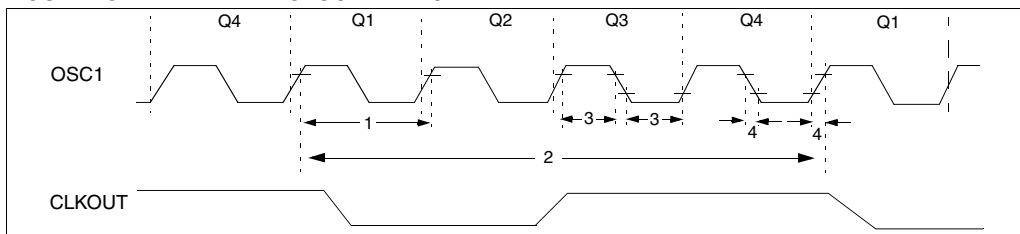


TABLE 18-2: EXTERNAL CLOCK TIMING REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
	Fosc	External CLKIN Frequency (Note 1)	DC	—	4	MHz	XT and RC osc mode
			DC	—	4	MHz	HS osc mode (-04)
			DC	—	10	MHz	HS osc mode (-10)
			DC	—	20	MHz	HS osc mode (-20)
			DC	—	200	kHz	LP osc mode
		Oscillator Frequency (Note 1)	DC	—	4	MHz	RC osc mode
			0.1	—	4	MHz	XT osc mode
			4	—	20	MHz	HS osc mode
			5	—	200	kHz	LP osc mode
			5	—	—	μs	LP osc mode
1	Tosc	External CLKIN Period (Note 1)	250	—	—	ns	XT and RC osc mode
			250	—	—	ns	HS osc mode (-04)
			100	—	—	ns	HS osc mode (-10)
			50	—	—	ns	HS osc mode (-20)
			5	—	—	μs	LP osc mode
		Oscillator Period (Note 1)	250	—	—	ns	RC osc mode
			250	—	10,000	ns	XT osc mode
			250	—	250	ns	HS osc mode (-04)
			100	—	250	ns	HS osc mode (-10)
			50	—	250	ns	HS osc mode (-20)
2	Tcy	Instruction Cycle Time (Note 1)	200	Tcy	DC	ns	Tcy = 4/Fosc
			200	—	—	ns	XT oscillator
3	TosL, TosH	External Clock in (OSC1) High or Low Time	2.5	—	—	μs	LP oscillator
			15	—	—	ns	HS oscillator
			—	—	—	ns	HS oscillator
4	TosR, TosF	External Clock in (OSC1) Rise or Fall Time	—	—	25	ns	XT oscillator
			—	—	50	ns	LP oscillator
			—	—	15	ns	HS oscillator

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: Instruction cycle period (Tcy) equals four times the input oscillator time-base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at "min." values with an external clock applied to the OSC1/CLKIN pin. When an external clock input is used, the "Max." cycle time limit is "DC" (no clock) for all devices.

PIC16C6X

Applicable Devices 61 62 62A R62 63 R63 64 64A R64 65 65A R65 66 67

FIGURE 18-3: CLKOUT AND I/O TIMING

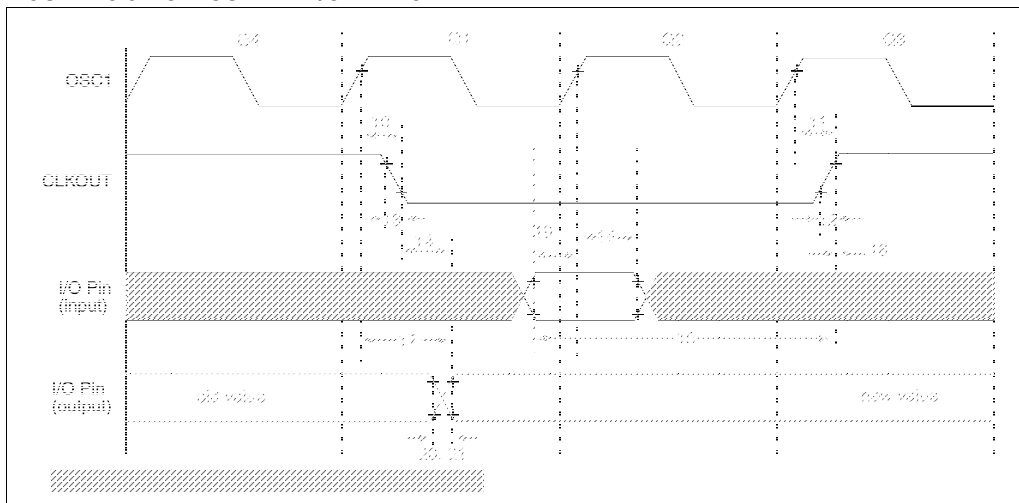


TABLE 18-3: CLKOUT AND I/O TIMING REQUIREMENTS

Parameters	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
10*	TosH2ckL	OSC1↑ to CLKOUT↓	—	75	200	ns	Note 1
11*	TosH2ckH	OSC1↑ to CLKOUT↑	—	75	200	ns	Note 1
12*	TckR	CLKOUT rise time	—	35	100	ns	Note 1
13*	TckF	CLKOUT fall time	—	35	100	ns	Note 1
14*	TckL2ioV	CLKOUT ↓ to Port out valid	—	—	0.5T _{CY} + 20	ns	Note 1
15*	TioV2ckH	Port in valid before CLKOUT ↑	T _{osc} + 200	—	—	ns	Note 1
16*	TckH2ioL	Port in hold after CLKOUT ↑	0	—	—	ns	Note 1
17*	TosH2ioV	OSC1↑ (Q1 cycle) to Port out valid	—	50	150	ns	
18*	TosH2ioL	OSC1↑ (Q2 cycle) to Port input invalid (I/O in hold time)	PIC16C62A/ R62/64A/R64	100	—	—	ns
			PIC16LC62A/ R62/64A/R64	200	—	—	ns
19*	TioV2osH	Port input valid to OSC1↑ (I/O in setup time)	0	—	—	ns	
20*	TioR	Port output rise time	PIC16C62A/ R62/64A/R64	—	10	40	ns
			PIC16LC62A/ R62/64A/R64	—	—	80	ns
21*	TioF	Port output fall time	PIC16C62A/ R62/64A/R64	—	10	40	ns
			PIC16LC62A/ R62/64A/R64	—	—	80	ns
22††*	Tinp	RB0/INT pin high or low time	T _{CY}	—	—	ns	
23††*	Trbp	RB7:RB4 change int high or low time	T _{CY}	—	—	ns	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

†† These parameters are asynchronous events not related to any internal clock edge.

Note 1: Measurements are taken in RC Mode where CLKOUT output is 4 x T_{OSC}.

PIC16C6X

Applicable Devices 61 62 62A R62 63 R63 64 64A R64 65 65A R65 66 67

21.4 Timing Parameter Symbology

The timing parameter symbols have been created following one of the following formats:

1. TppS2ppS
2. TppS
3. TCC:ST (I²C specifications only)
4. Ts (I²C specifications only)

T		T	
F	Frequency	T	Time

Lowercase letters (pp) and their meanings:

pp		osc	OSC1
cc	CCP1	rd	\overline{RD}
ck	CLKOUT	rw	\overline{RD} or \overline{WR}
cs	\overline{CS}	sc	SCK
di	SDI	ss	\overline{SS}
do	SDO	t0	T0CKI
dt	Data in	t1	T1CKI
io	I/O port	wr	\overline{WR}
mc	\overline{MCLR}		

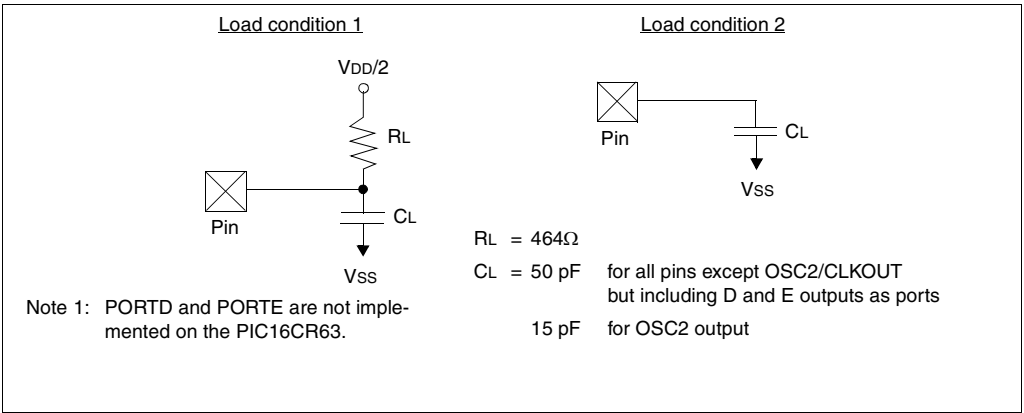
Uppercase letters and their meanings:

S		P	Period
F	Fall	R	Rise
H	High	V	Valid
I	Invalid (Hi-impedance)	Z	Hi-impedance
L	Low		
I²C only		High	High
AA	output access	Low	Low
BUF	Bus free		

TCC:ST (I²C specifications only)

CC		SU	Setup
HD	Hold		
ST		STO	STOP condition
DAT	DATA input hold		
STA	START condition		

FIGURE 21-1: LOAD CONDITIONS FOR DEVICE TIMING SPECIFICATIONS



Applicable Devices 61 62 62A R62 63 R63 64 64A R64 65 65A R65 66 67

FIGURE 21-7: CAPTURE/COMPARE/PWM TIMINGS (CCP1 AND CCP2)

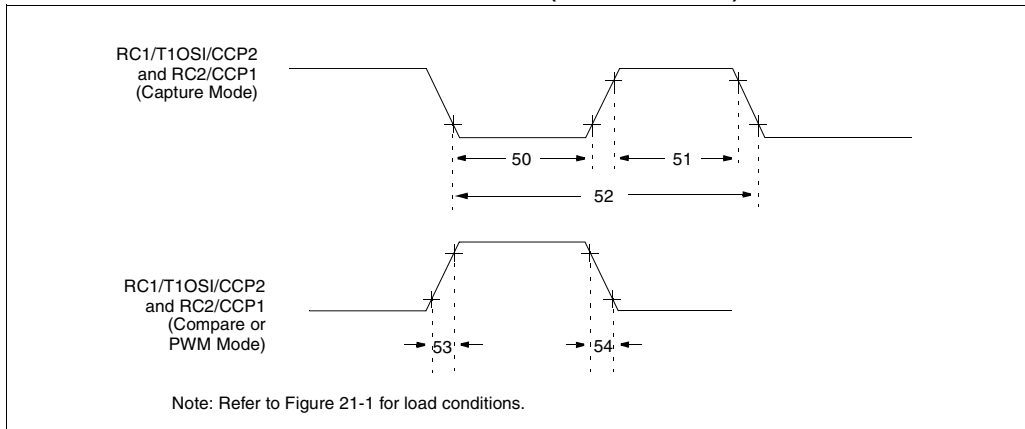


TABLE 21-6: CAPTURE/COMPARE/PWM REQUIREMENTS (CCP1 AND CCP2)

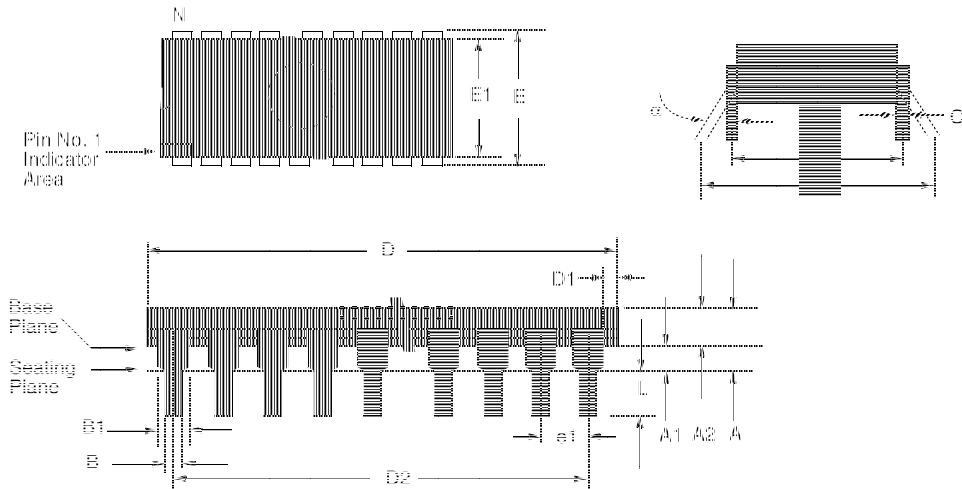
Param No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
50*	TccL	CCP1 and CCP2 input low time	No Prescaler	$0.5T_{CY} + 20$	—	—	ns	
			With Prescaler	PIC16CR63/R65	10	—	ns	
				PIC16LCR63/R65	20	—	ns	
51*	TccH	CCP1 and CCP2 input high time	No Prescaler	$0.5T_{CY} + 20$	—	—	ns	
			With Prescaler	PIC16CR63/R65	10	—	ns	
				PIC16LCR63/R65	20	—	ns	
52*	TccP	CCP1 and CCP2 input period		$\frac{3T_{CY} + 40}{N}$	—	—	ns	N = prescale value (1, 4, or 16)
53*	TccR	CCP1 and CCP2 output rise time		PIC16CR63/R65	10	25	ns	
				PIC16LCR63/R65	25	45	ns	
54*	TccF	CCP1 and CCP2 output fall time		PIC16CR63/R65	10	25	ns	
				PIC16LCR63/R65	25	45	ns	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

24.7 28-Lead Ceramic Cerdip Dual In-line with Window (300 mil) (JW)

Note: For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



Package Group: Ceramic Cerdip Dual In-Line (CDP)						
Symbol	Millimeters			Inches		
	Min	Max	Notes	Min	Max	Notes
α	0°	10°		0°	10°	
A	3.30	5.84		.130	0.230	
A1	0.38	—		0.015	—	
A2	2.92	4.95		0.115	0.195	
B	0.35	0.58		0.014	0.023	
B1	1.14	1.78	Typical	0.045	0.070	Typical
C	0.20	0.38	Typical	0.008	0.015	Typical
D	34.54	37.72		1.360	1.485	
D2	32.97	33.07	Reference	1.298	1.302	Reference
E	7.62	8.25		0.300	0.325	
E1	6.10	7.87		0.240	0.310	
e	2.54	2.54	Typical	0.100	0.100	Typical
eA	7.62	7.62	Reference	0.300	0.300	Reference
eB	—	11.43		—	0.450	
L	2.92	5.08		0.115	0.200	
N	28	28		28	28	
D1	0.13	—		0.005	—	

PIC16C6X

OSC1/CLKIN.....	16	Map.....	19, 20
OSC2/CLKOUT.....	16	Organization.....	19
PORTA.....	52	Paging.....	48
PORTB.....	54	Section.....	19
PORTC.....	55	Programming While In-circuit.....	142
PORTD.....	57	PS2:PS0.....	36
PORTE.....	59	PSA.....	36
RA4/T0CKI.....	16, 52	PSPIE.....	39
RA5/SS.....	16, 52	PSPIF.....	43
RB0/INT.....	16, 54	Pull-ups.....	53
RB6.....	142	PUSH.....	48
RB7.....	142	PWM.....	
RC0/T1OSI/T1CKI.....	55	Block Diagram.....	80
RC0/T1OSO/T1CKI.....	16, 55	Calculations.....	81
RC1/T1OSI.....	55	Mode.....	80
RC1/T1OSI/CCP2.....	16, 55	Output Timing.....	80
RC1/T1OSO.....	55	PWM Least Significant bits.....	78
RC2/CCP1.....	16, 55, 56		
RC3/SCK/SCL.....	16, 55, 56	Q	
RC4/SDI/SDA.....	16, 55, 56	Quadrature Clocks.....	18
RC5/SDO.....	16, 55, 56	Quick-Turnaround-Production.....	7
RC6/TX/CK.....	16, 55, 56, 105–120		
RC7/RX/DT.....	16, 55, 56, 105–120	R	
RD7/PSP7:RD0/PSP0.....	17, 57	R \overline{W} bit.....	84, 89, 96, 100, 101, 102
RE0/ \overline{RD}	17, 59, 61	RA0 pin.....	51
RE1/ \overline{WR}	17, 59, 61	RA1 pin.....	51
RE2/ \overline{CS}	17, 59, 61	RA2 pin.....	51
SCK.....	86–88	RA3 pin.....	51
SDI.....	86–88	RA4/T0CKI pin.....	51
SDO.....	86–88	RA5 pin.....	51
SS.....	86–88	RB Port Change Interrupt Enable bit, RBIE.....	37
V \overline{DD}	17	RB Port Change Interrupt Flag bit, RBIF.....	37
V \overline{SS}	17	RB0.....	54
PIR1.....	24, 26, 28, 30, 32, 34	RB0/INT.....	138
PIR2.....	24, 26, 28, 30, 32, 34	RB0/INT External Interrupt Enable bit, INTE.....	37
POP.....	48	RB0/INT External Interrupt Flag bit, INTF.....	37
POR.....	47, 131	RB1.....	54
POR Time-Out Sequence on Power-Up.....	134	RB2.....	54
Port RB Interrupt.....	53	RB3.....	54
PORTA.....	24, 26, 28, 30, 32, 34, 51	RB4.....	53
PORTB.....	24, 26, 28, 30, 32, 34, 53	RB5.....	53
PORTB Interrupt on Change.....	138	RB6.....	53
PORTB Pull-up Enable bit, RBPU.....	36	RB7.....	53
PORTC.....	24, 26, 28, 30, 32, 34, 55	RBIE.....	37
PORTD.....	24, 26, 28, 30, 32, 34, 57	RBIF.....	37
PORTE.....	24, 26, 28, 30, 32, 34, 58	RBPU.....	36, 53
Ports.....		RC Oscillator.....	130
Bi-directional.....	60	RCIE.....	39
I/O Programming Considerations.....	60	RCIF.....	42
PORTA.....	16	RCREG.....	24, 26, 28, 30, 32, 34
PORTB.....	16	RCSTA.....	24, 26, 28, 30, 32, 34, 106
PORTC.....	16	RCV_MODE.....	104
PORTD.....	17	Read Only Memory.....	7
PORTE.....	17	Read/Write bit Information, R \overline{W}	84, 89
Successive Operations on an I/O Port.....	60	Receive and Control Register.....	106
Power/Control Status Register, PCON.....	130	Receive Overflow Detect bit, SSPOV.....	85
Power-down bit.....	35	Receive Overflow Indicator bit, SSPOV.....	90
Power-down Mode.....	141	Register Bank Select bit, Indirect.....	35
Power-on Reset (POR).....	129	Register Bank Select bits. Direct.....	35
Power-on Reset Status bit, POR.....	47		
Power-up Timer (PWRT).....	123, 129		
PR2.....	25, 27, 29, 31, 33, 34		
Prescaler.....	68		
Prescaler Assignment bit, PSA.....	36		
Prescaler Rate Select bits, PS2:PS0.....	36		
PRO MATE Universal Programmer.....	159		
Program Memory.....			

TMR0	24, 26, 28, 30, 32, 34
TMR0 Clock Source Select bit, T0CS	36
TMR0 Interrupt	65
TMR0 Overflow Interrupt Enable bit, T0IE	37
TMR0 Overflow Interrupt Flag bit, T0IF	37
TMR0 Prescale Selection Table	36
TMR0 Source Edge Select bit, T0SE	36
TMR1 Overflow Interrupt Enable bit, TMR1IE	38
TMR1 Overflow Interrupt Flag bit, TMR1IF	41
TMR1CS	71
TMR1H	24, 26, 28, 30, 32, 34
TMR1IE	38
TMR1IF	41
TMR1L	24, 26, 28, 30, 32, 34
TMR1ON	71
TMR2	24, 26, 28, 30, 32, 34
TMR2 Register	75
TMR2 to PR2 Match Interrupt Enable bit, TMR2IE	38
TMR2 to PR2 Match Interrupt Flag bit, TMR2IF	41
TMR2IE	38
TMR2IF	41
TMR2ON	75
TO	35, 131
TOUTPS3:TOUTPS0	75
Transmit Enable bit, TXEN	105
Transmit Shift Register Status bit, TRMT	105
Transmit Status and Control Register	105
TRISA	25, 27, 29, 31, 33, 34, 51
TRISB	25, 27, 29, 31, 33, 34, 53
TRISC	25, 27, 29, 31, 33, 34, 55, 94
TRISD	25, 27, 29, 31, 33, 34, 57
TRISE	25, 27, 29, 31, 33, 34, 58
TRMT	105
TX9	105
TX9D	105
TXEN	105
TXIE	39
TXIF	42
TXREG	24, 26, 28, 30, 32, 34
TXSTA	25, 27, 29, 31, 33, 34, 105

U

UA	84, 89
Universal Synchronous Asynchronous Receiver Transmitter (USART)	
Asynchronous Mode	
Setting Up Transmission	113
Timing Diagram, Master Transmission	113
Transmitter	112
Asynchronous Receiver	
Setting Up Reception	115
Timing Diagram	114
Asynchronous Receiver Mode	
Block Diagram	114
Section	114
Section	105
Synchronous Master Mode	
Reception	118
Section	116
Setting Up Reception	118
Setting Up Transmission	116
Timing Diagram, Reception	119
Timing Diagram, Transmission	117
Transmission	116

Synchronous Slave Mode

Reception	120
Section	120
Setting Up Reception	120
Setting Up Transmission	120
Transmit	120
Transmit Block Diagram	112
Update Address bit, UA	84, 89
USART Receive Interrupt Enable bit, RCIE	39
USART Receive Interrupt Flag bit, RCIF	42
USART Transmit Interrupt Enable bit, TXIE	39
USART Transmit Interrupt Flag bit, TXIF	42
UV Erasable Devices	7

W

Wake-up from Sleep	141
Wake-up on Key Depression	53
Wake-up Using Interrupts	141
Watchdog Timer (WDT)	
Block Diagram	140
Period	140
Programming Considerations	140
Section	140
WCOL	85, 90
Weak Internal Pull-ups	53
Write Collision Detect bit, WCOL	85, 90

X

XMIT_MODE	104
XT	130

Z

Z	35
Zero bit	9, 35