

Welcome to [E-XFL.COM](http://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	16MHz
Connectivity	UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	33
Program Memory Size	8KB (4K x 16)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	454 x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 6V
Data Converters	-
Oscillator Type	External
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Through Hole
Package / Case	40-DIP (0.600", 15.24mm)
Supplier Device Package	40-PDIP
Purchase URL	<a href="https://www.e-xfl.com/product-detail/microchip-technology/pic17c43-16-p">https://www.e-xfl.com/product-detail/microchip-technology/pic17c43-16-p</a>

## 2.0 PIC17C4X DEVICE VARIETIES

A variety of frequency ranges and packaging options are available. Depending on application and production requirements, the proper device option can be selected using the information in the PIC17C4X Product Selection System section at the end of this data sheet. When placing orders, please use the "PIC17C4X Product Identification System" at the back of this data sheet to specify the correct part number.

For the PIC17C4X family of devices, there are four device "types" as indicated in the device number:

1. **C**, as in PIC17**C**42. These devices have EPROM type memory and operate over the standard voltage range.
2. **LC**, as in PIC17**LC**42. These devices have EPROM type memory, operate over an extended voltage range, and reduced frequency range.
3. **CR**, as in PIC17**CR**42. These devices have ROM type memory and operate over the standard voltage range.
4. **LCR**, as in PIC17**LCR**42. These devices have ROM type memory, operate over an extended voltage range, and reduced frequency range.

### 2.1 UV Erasable Devices

The UV erasable version, offered in Cerdip package, is optimal for prototype development and pilot programs.

The UV erasable version can be erased and reprogrammed to any of the configuration modes. Microchip's PRO MATE™ programmer supports programming of the PIC17C4X. Third party programmers also are available; refer to the *Third Party Guide* for a list of sources.

### 2.2 One-Time-Programmable (OTP) Devices

The availability of OTP devices is especially useful for customers expecting frequent code changes and updates.

The OTP devices, packaged in plastic packages, permit the user to program them once. In addition to the program memory, the configuration bits must also be programmed.

### 2.3 Quick-Turnaround-Production (QTP) Devices

Microchip offers a QTP Programming Service for factory production orders. This service is made available for users who choose not to program a medium to high quantity of units and whose code patterns have stabilized. The devices are identical to the OTP devices but with all EPROM locations and configuration options already programmed by the factory. Certain code and prototype verification procedures apply before production shipments are available. Please contact your local Microchip Technology sales office for more details.

### 2.4 Serialized Quick-Turnaround Production (SQTP<sup>SM</sup>) Devices

Microchip offers a unique programming service where a few user-defined locations in each device are programmed with different serial numbers. The serial numbers may be random, pseudo-random or sequential.

Serial programming allows each device to have a unique number which can serve as an entry-code, password or ID number.

ROM devices do not allow serialization information in the program memory space.

For information on submitting ROM code, please contact your regional sales office.

### 2.5 Read Only Memory (ROM) Devices

Microchip offers masked ROM versions of several of the highest volume parts, thus giving customers a low cost option for high volume, mature products.

For information on submitting ROM code, please contact your regional sales office.

## 5.1 Interrupt Status Register (INTSTA)

The Interrupt Status/Control register (INTSTA) records the individual interrupt requests in flag bits, and contains the individual interrupt enable bits (not for the peripherals).

The PEIF bit is a read only, bit wise OR of all the peripheral flag bits in the PIR register (Figure 5-4).

**Note:** T0IF, INTF, T0CKIF, or PEIF will be set by the specified condition, even if the corresponding interrupt enable bit is clear (interrupt disabled) or the GLINTD bit is set (all interrupts disabled).

Care should be taken when clearing any of the INTSTA register enable bits when interrupts are enabled (GLINTD is clear). If any of the INTSTA flag bits (T0IF, INTF, T0CKIF, or PEIF) are set in the same instruction cycle as the corresponding interrupt enable bit is cleared, the device will vector to the reset address (0x00).

When disabling any of the INTSTA enable bits, the GLINTD bit should be set (disabled).

**FIGURE 5-2: INTSTA REGISTER (ADDRESS: 07h, UNBANKED)**

R - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0
PEIF	T0CKIF	T0IF	INTF	PEIE	T0CKIE	T0IE	INTE
bit7							bit0

R = Readable bit  
W = Writable bit  
- n = Value at POR reset

bit 7: **PEIF:** Peripheral Interrupt Flag bit  
This bit is the OR of all peripheral interrupt flag bits AND'ed with their corresponding enable bits.  
1 = A peripheral interrupt is pending  
0 = No peripheral interrupt is pending

bit 6: **T0CKIF:** External Interrupt on T0CKI Pin Flag bit  
This bit is cleared by hardware, when the interrupt logic forces program execution to vector (18h).  
1 = The software specified edge occurred on the RA1/T0CKI pin  
0 = The software specified edge did not occur on the RA1/T0CKI pin

bit 5: **T0IF:** TMR0 Overflow Interrupt Flag bit  
This bit is cleared by hardware, when the interrupt logic forces program execution to vector (10h).  
1 = TMR0 overflowed  
0 = TMR0 did not overflow

bit 4: **INTF:** External Interrupt on INT Pin Flag bit  
This bit is cleared by hardware, when the interrupt logic forces program execution to vector (08h).  
1 = The software specified edge occurred on the RA0/INT pin  
0 = The software specified edge did not occur on the RA0/INT pin

bit 3: **PEIE:** Peripheral Interrupt Enable bit  
This bit enables all peripheral interrupts that have their corresponding enable bits set.  
1 = Enable peripheral interrupts  
0 = Disable peripheral interrupts

bit 2: **T0CKIE:** External Interrupt on T0CKI Pin Enable bit  
1 = Enable software specified edge interrupt on the RA1/T0CKI pin  
0 = Disable interrupt on the RA1/T0CKI pin

bit 1: **T0IE:** TMR0 Overflow Interrupt Enable bit  
1 = Enable TMR0 overflow interrupt  
0 = Disable TMR0 overflow interrupt

bit 0: **INTE:** External Interrupt on RA0/INT Pin Enable bit  
1 = Enable software specified edge interrupt on the RA0/INT pin  
0 = Disable software specified edge interrupt on the RA0/INT pin

## 5.5 RA0/INT Interrupt

The external interrupt on the RA0/INT pin is edge triggered. Either the rising edge, if INTEDG bit (T0STA<7>) is set, or the falling edge, if INTEDG bit is clear. When a valid edge appears on the RA0/INT pin, the INTF bit (INTSTA<4>) is set. This interrupt can be disabled by clearing the INTE control bit (INTSTA<0>). The INT interrupt can wake the processor from SLEEP. See Section 14.4 for details on SLEEP operation.

## 5.6 TMR0 Interrupt

An overflow (FFFFh → 0000h) in TMR0 will set the T0IF (INTSTA<5>) bit. The interrupt can be enabled/disabled by setting/clearing the T0IE control bit (INTSTA<1>). For operation of the Timer0 module, see Section 11.0.

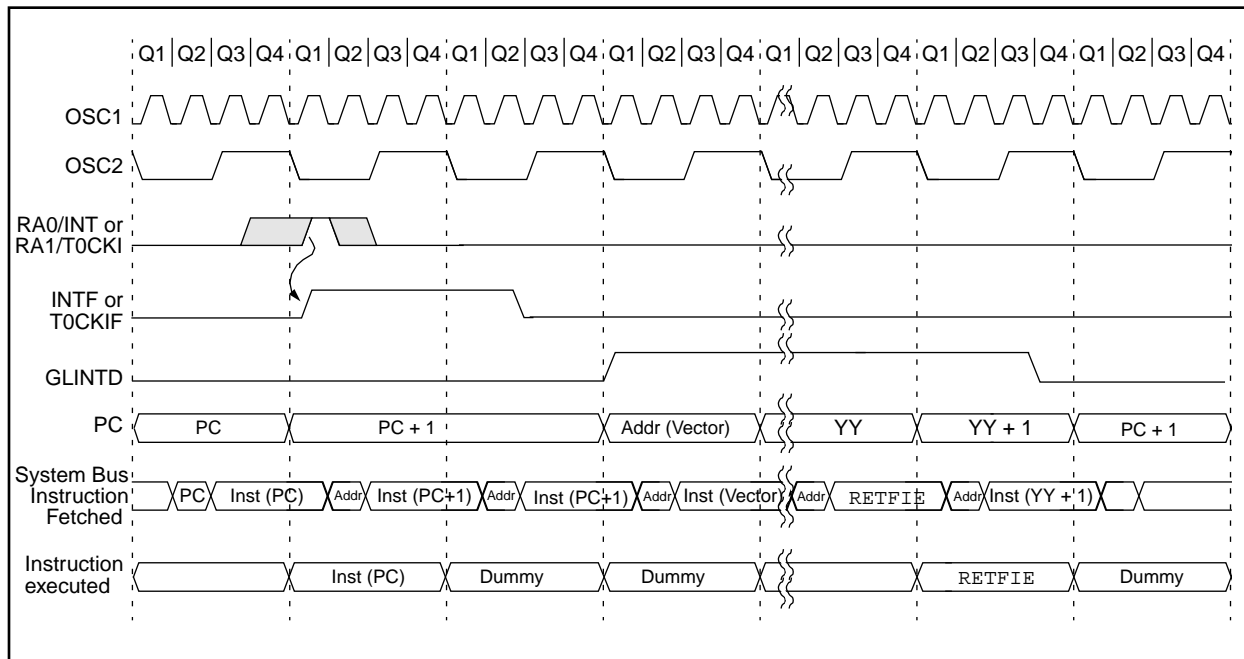
## 5.7 T0CKI Interrupt

The external interrupt on the RA1/T0CKI pin is edge triggered. Either the rising edge, if the T0SE bit (T0STA<6>) is set, or the falling edge, if the T0SE bit is clear. When a valid edge appears on the RA1/T0CKI pin, the T0CKIF bit (INTSTA<6>) is set. This interrupt can be disabled by clearing the T0CKIE control bit (INTSTA<2>). The T0CKI interrupt can wake up the processor from SLEEP. See Section 14.4 for details on SLEEP operation.

## 5.8 Peripheral Interrupt

The peripheral interrupt flag indicates that at least one of the peripheral interrupts occurred (PEIF is set). The PEIF bit is a read only bit, and is a bit wise OR of all the flag bits in the PIR register AND'ed with the corresponding enable bits in the PIE register. Some of the peripheral interrupts can wake the processor from SLEEP. See Section 14.4 for details on SLEEP operation.

**FIGURE 5-5: INT PIN / T0CKI PIN INTERRUPT TIMING**



Example 8-4 shows the sequence to do an 16 x 16 signed multiply. Equation 8-2 shows the algorithm that used. The 32-bit result is stored in four registers RES3:RES0. To account for the sign bits of the arguments, each argument pairs most significant bit (MSb) is tested and the appropriate subtractions are done.

## EQUATION 8-2: 16 x 16 SIGNED MULTIPLICATION ALGORITHM

RES3:RES0

$$\begin{aligned}
 &= \text{ARG1H:ARG1L} * \text{ARG2H:ARG2L} \\
 &= (\text{ARG1H} * \text{ARG2H} * 2^{16}) &+ \\
 &\quad (\text{ARG1H} * \text{ARG2L} * 2^8) &+ \\
 &\quad (\text{ARG1L} * \text{ARG2H} * 2^8) &+ \\
 &\quad (\text{ARG1L} * \text{ARG2L}) &+ \\
 &\quad (-1 * \text{ARG2H} <7> * \text{ARG1H:ARG1L} * 2^{16}) &+ \\
 &\quad (-1 * \text{ARG1H} <7> * \text{ARG2H:ARG2L} * 2^{16})
 \end{aligned}$$

## EXAMPLE 8-4: 16 x 16 SIGNED MULTIPLY ROUTINE

```

MOVFP ARG1L, WREG
MULWF ARG2L      ; ARG1L * ARG2L ->
                  ; PRODH:PRODL

MOVFP PRODH, RES1 ;
MOVFP PRODL, RES0 ;

;

MOVFP ARG1H, WREG
MULWF ARG2H      ; ARG1H * ARG2H ->
                  ; PRODH:PRODL

MOVFP PRODH, RES3 ;
MOVFP PRODL, RES2 ;

;

MOVFP ARG1L, WREG
MULWF ARG2H      ; ARG1L * ARG2H ->
                  ; PRODH:PRODL

MOVFP PRODL, WREG ;
ADDWF RES1, F    ; Add cross
MOVFP PRODH, WREG ; products
ADDWFC RES2, F   ;
CLRf WREG, F     ;
ADDWFC RES3, F   ;

;

MOVFP ARG1H, WREG ;
MULWF ARG2L      ; ARG1H * ARG2L ->
                  ; PRODH:PRODL

MOVFP PRODL, WREG ;
ADDWF RES1, F    ; Add cross
MOVFP PRODH, WREG ; products
ADDWFC RES2, F   ;
CLRf WREG, F     ;
ADDWFC RES3, F   ;

;

BTfSS ARG2H, 7   ; ARG2H:ARG2L neg?
GOTO SIGN_ARG1  ; no, check ARG1
MOVFP ARG1L, WREG ;
SUBWF RES2      ;
MOVFP ARG1H, WREG ;
SUBWFB RES3     ;

;

SIGN_ARG1
BTfSS ARG1H, 7   ; ARG1H:ARG1L neg?
GOTO CONT_CODE  ; no, done
MOVFP ARG2L, WREG ;
SUBWF RES2      ;
MOVFP ARG2H, WREG ;
SUBWFB RES3     ;

;

CONT_CODE
:

```

## 9.4.1 PORTE AND DDRE REGISTER

PORTE is a 3-bit bi-directional port. The corresponding data direction register is DDRE. A '1' in DDRE configures the corresponding port pin as an input. A '0' in the DDRE register configures the corresponding port pin as an output. Reading PORTE reads the status of the pins, whereas writing to it will write to the port latch. PORTE is multiplexed with the system bus. When operating as the system bus, PORTE contains the control signals for the address/data bus (AD15:AD0). These control signals are Address Latch Enable (ALE), Output Enable ( $\overline{OE}$ ), and Write ( $\overline{WR}$ ). The control signals  $\overline{OE}$  and  $\overline{WR}$  are active low signals. The timing for the system bus is shown in the Electrical Characteristics section.

**Note:** This port is configured as the system bus when the device's configuration bits are selected to Microprocessor or Extended Microcontroller modes. In the two other microcontroller modes, this port is a general purpose I/O.

Example 9-4 shows the instruction sequence to initialize PORTE. The Bank Select Register (BSR) must be selected to Bank 1 for the port to be initialized.

### EXAMPLE 9-4: INITIALIZING PORTE

```

MOVLB 1           ; Select Bank 1
CLRF  PORTE       ; Initialize PORTE data
                  ; latches before setting
                  ; the data direction
                  ; register
MOVLW 0x03        ; Value used to initialize
                  ; data direction
MOVWF DDRE        ; Set RE<1:0> as inputs
                  ; RE<2> as outputs
                  ; RE<7:3> are always
                  ; read as '0'
    
```

**FIGURE 9-8: PORTE BLOCK DIAGRAM (IN I/O PORT MODE)**

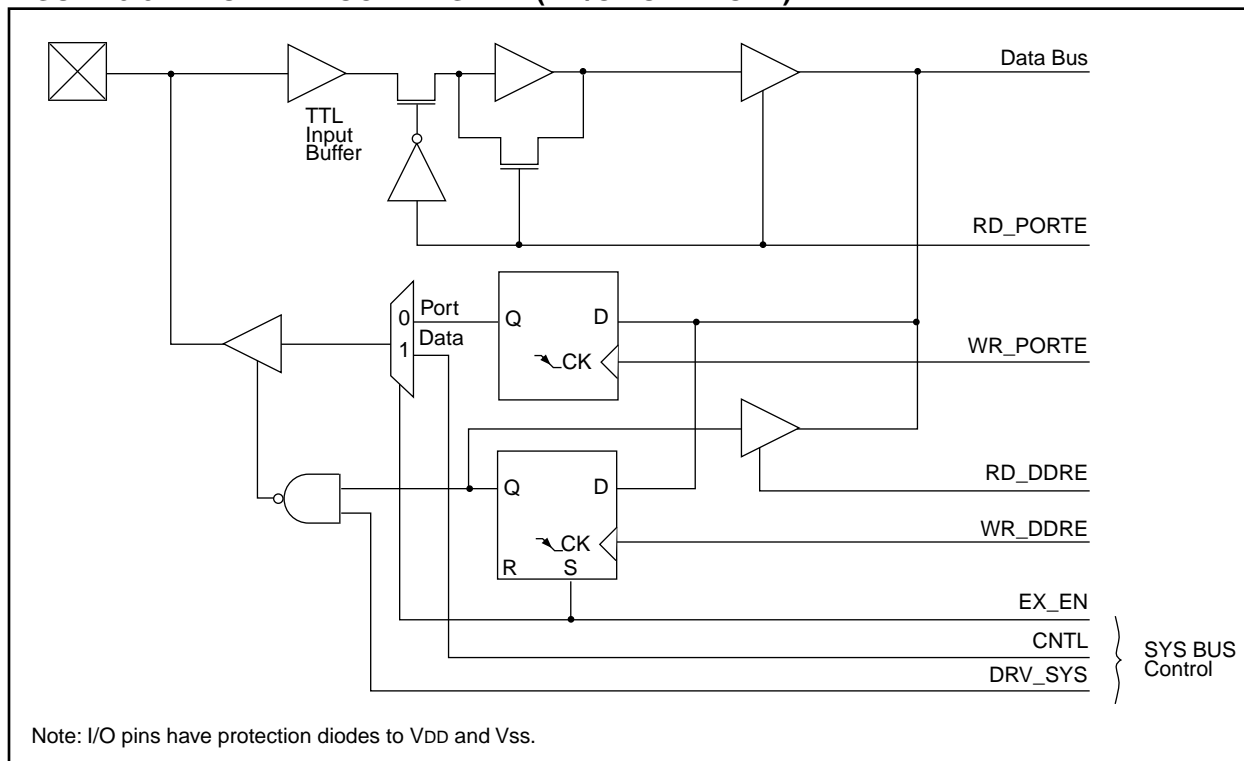


FIGURE 11-5: TMR0 READ/WRITE IN TIMER MODE

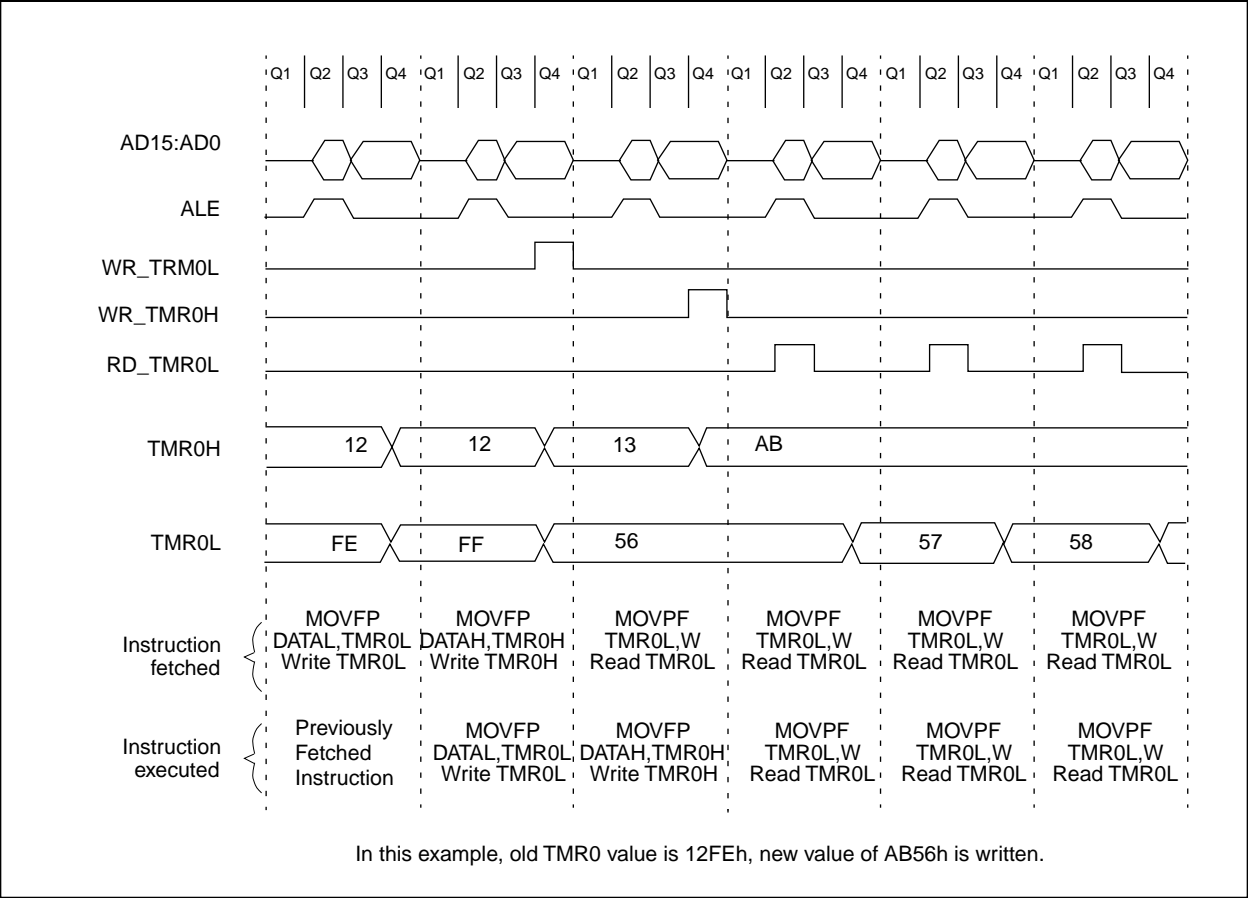


TABLE 11-1: REGISTERS/BITS ASSOCIATED WITH TIMER0

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-on Reset	Value on all other resets (Note1)
05h, Unbanked	T0STA	INTEDG	T0SE	T0CS	PS3	PS2	PS1	PS0	—	0000 000—	0000 000—
06h, Unbanked	CPUSTA	—	—	STKAV	GLINTD	$\overline{\text{TO}}$	$\overline{\text{PD}}$	—	—	--11 11--	--11 qq--
07h, Unbanked	INTSTA	PEIF	T0CKIF	T0IF	INTF	PEIE	T0CKIE	T0IE	INTE	0000 0000	0000 0000
0Bh, Unbanked	TMR0L	TMR0 register; low byte								xxxx xxxx	uuuu uuuu
0Ch, Unbanked	TMR0H	TMR0 register; high byte								xxxx xxxx	uuuu uuuu

Legend: x = unknown, u = unchanged, - = unimplemented read as a '0', q - value depends on condition, Shaded cells are not used by Timer0.  
Note 1: Other (non power-up) resets include: external reset through MCLR and the Watchdog Timer Reset.

**FIGURE 12-2: TCON2 REGISTER (ADDRESS: 17h, BANK 3)**

R - 0	R - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0
CA2OVF	CA1OVF	PWM2ON	PWM1ON	CA1/PR3	TMR3ON	TMR2ON	TMR1ON
bit7							bit0

R = Readable bit  
W = Writable bit  
-n = Value at POR reset

bit 7: **CA2OVF:** Capture2 Overflow Status bit  
This bit indicates that the capture value had not been read from the capture register pair (CA2H:CA2L) before the next capture event occurred. The capture register retains the oldest unread capture value (last capture before overflow). Subsequent capture events will not update the capture register with the Timer3 value until the capture register has been read (both bytes).  
1 = Overflow occurred on Capture2 register  
0 = No overflow occurred on Capture2 register

bit 6: **CA1OVF:** Capture1 Overflow Status bit  
This bit indicates that the capture value had not been read from the capture register pair (PR3H/CA2H:PR3L/CA2L) before the next capture event occurred. The capture register retains the oldest unread capture value (last capture before overflow). Subsequent capture events will not update the capture register with the TMR3 value until the capture register has been read (both bytes).  
1 = Overflow occurred on Capture1 register  
0 = No overflow occurred on Capture1 register

bit 5: **PWM2ON:** PWM2 On bit  
1 = PWM2 is enabled (The RB3/PWM2 pin ignores the state of the DDRB<3> bit)  
0 = PWM2 is disabled (The RB3/PWM2 pin uses the state of the DDRB<3> bit for data direction)

bit 4: **PWM1ON:** PWM1 On bit  
1 = PWM1 is enabled (The RB2/PWM1 pin ignores the state of the DDRB<2> bit)  
0 = PWM1 is disabled (The RB2/PWM1 pin uses the state of the DDRB<2> bit for data direction)

bit 3: **CA1/PR3:** CA1/PR3 Register Mode Select bit  
1 = Enables Capture1 (PR3H/CA1H:PR3L/CA1L is the Capture1 register. Timer3 runs without a period register)  
0 = Enables the Period register (PR3H/CA1H:PR3L/CA1L is the Period register for Timer3)

bit 2: **TMR3ON:** Timer3 On bit  
1 = Starts Timer3  
0 = Stops Timer3

bit 1: **TMR2ON:** Timer2 On bit  
This bit controls the incrementing of the Timer2 register. When Timer2:Timer1 form the 16-bit timer (T16 is set), TMR2ON must be set. This allows the MSB of the timer to increment.  
1 = Starts Timer2 (Must be enabled if the T16 bit (TCON1<3>) is set)  
0 = Stops Timer2

bit 0: **TMR1ON:** Timer1 On bit  
When T16 is set (in 16-bit Timer Mode)  
1 = Starts 16-bit Timer2:Timer1  
0 = Stops 16-bit Timer2:Timer1  
  
When T16 is clear (in 8-bit Timer Mode)  
1 = Starts 8-bit Timer1  
0 = Stops 8-bit Timer1



## 13.3.2 USART SYNCHRONOUS MASTER RECEPTION

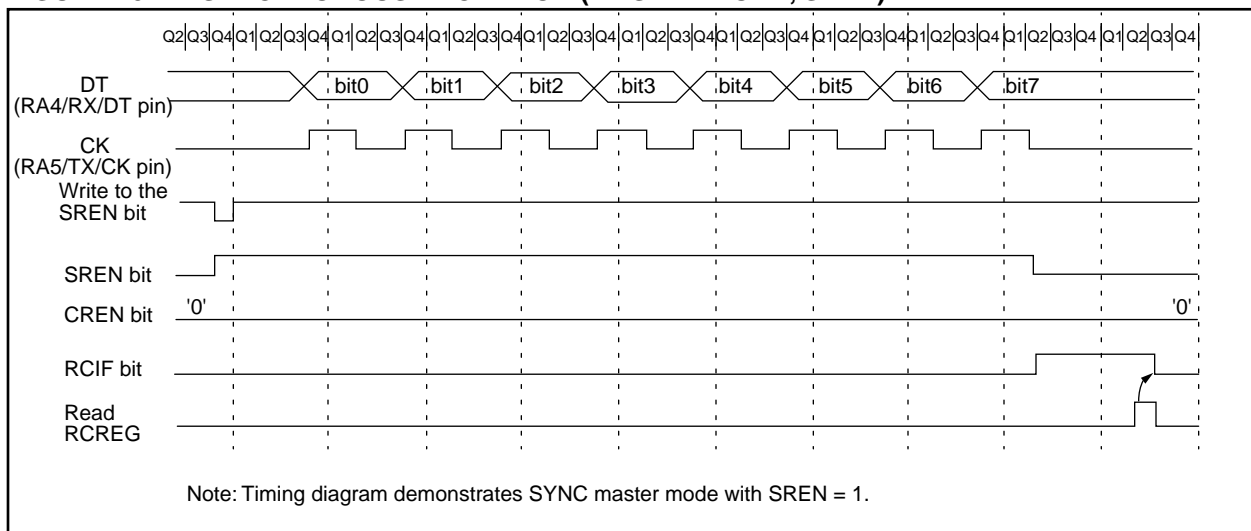
Once synchronous mode is selected, reception is enabled by setting either the SREN (RCSTA<5>) bit or the CREN (RCSTA<4>) bit. Data is sampled on the RA4/RX/DT pin on the falling edge of the clock. If SREN is set, then only a single word is received. If CREN is set, the reception is continuous until CREN is reset. If both bits are set, then CREN takes precedence. After clocking the last bit, the received data in the Receive Shift Register (RSR) is transferred to RCREG (if it is empty). If the transfer is complete, the interrupt bit RCIF (PIR<0>) is set. The actual interrupt can be enabled/disabled by setting/clearing the RCIE (PIE<0>) bit. RCIF is a read only bit which is RESET by the hardware. In this case it is reset when RCREG has been read and is empty. RCREG is a double buffered register; i.e., it is a two deep FIFO. It is possible for two bytes of data to be received and transferred to the RCREG FIFO and a third byte to begin shifting into the RSR. On the clocking of the last bit of the third byte, if RCREG is still full, then the overrun error bit OERR (RCSTA<1>) is set. The word in the RSR will be lost. RCREG can be read twice to retrieve the two bytes in the FIFO. The OERR bit has to be cleared in software. This is done by clearing the CREN bit. If OERR bit is set, transfers from RSR to RCREG are inhibited, so it is essential to clear OERR bit if it is set. The 9th receive bit is buffered the same way as the receive data. Reading the RCREG register will allow the RX9D and FERR bits to be loaded with values for the next received data; therefore, it is essential for the user to read the RCSTA register before reading RCREG in order not to lose the old FERR and RX9D information.

Steps to follow when setting up a Synchronous Master Reception:

1. Initialize the SPBRG register for the appropriate baud rate. See Section 13.1 for details.
2. Enable the synchronous master serial port by setting bits SYNC, SPEN, and CSRC.
3. If interrupts are desired, then set the RCIE bit.
4. If 9-bit reception is desired, then set the RX9 bit.
5. If a single reception is required, set bit SREN. For continuous reception set bit CREN.
6. The RCIF bit will be set when reception is complete and an interrupt will be generated if the RCIE bit was set.
7. Read RCSTA to get the ninth bit (if enabled) and determine if any error occurred during reception.
8. Read the 8-bit received data by reading RCREG.
9. If any error occurred, clear the error by clearing CREN.

**Note:** To terminate a reception, either clear the SREN and CREN bits, or the SPEN bit. This will reset the receive logic, so that it will be in the proper state when receive is re-enabled.

**FIGURE 13-11: SYNCHRONOUS RECEPTION (MASTER MODE, SREN)**



## 14.2.4 EXTERNAL CRYSTAL OSCILLATOR CIRCUIT

Either a prepackaged oscillator can be used or a simple oscillator circuit with TTL gates can be built. Prepackaged oscillators provide a wide operating range and better stability. A well-designed crystal oscillator will provide good performance with TTL gates. Two types of crystal oscillator circuits can be used: one with series resonance, or one with parallel resonance.

Figure 14-5 shows implementation of a parallel resonant oscillator circuit. The circuit is designed to use the fundamental frequency of the crystal. The 74AS04 inverter performs the 180-degree phase shift that a parallel oscillator requires. The 4.7 k $\Omega$  resistor provides the negative feedback for stability. The 10 k $\Omega$  potentiometer biases the 74AS04 in the linear region. This could be used for external oscillator designs.

**FIGURE 14-5: EXTERNAL PARALLEL RESONANT CRYSTAL OSCILLATOR CIRCUIT**

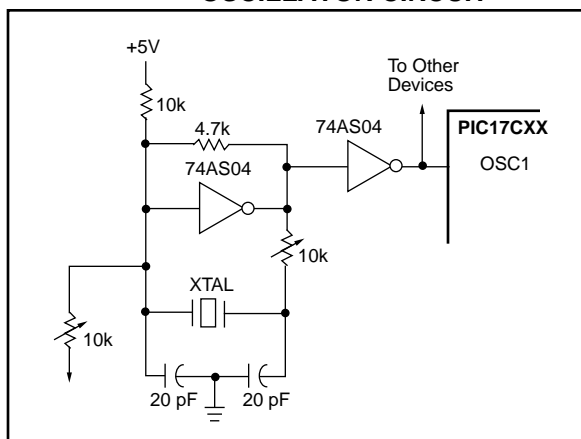
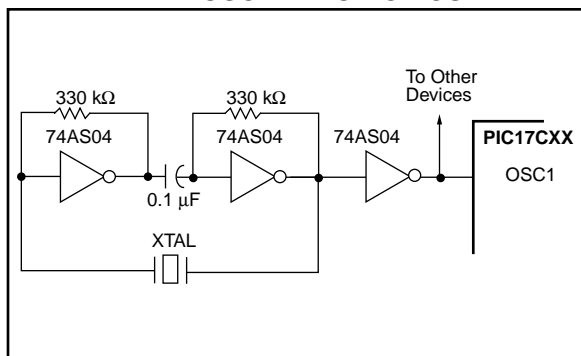


Figure 14-6 shows a series resonant oscillator circuit. This circuit is also designed to use the fundamental frequency of the crystal. The inverter performs a 180-degree phase shift in a series resonant oscillator circuit. The 330 k $\Omega$  resistors provide the negative feedback to bias the inverters in their linear region.

**FIGURE 14-6: EXTERNAL SERIES RESONANT CRYSTAL OSCILLATOR CIRCUIT**



## 14.2.5 RC OSCILLATOR

For timing insensitive applications, the RC device option offers additional cost savings. RC oscillator frequency is a function of the supply voltage, the resistor ( $R_{ext}$ ) and capacitor ( $C_{ext}$ ) values, and the operating temperature. In addition to this, oscillator frequency will vary from unit to unit due to normal process parameter variation. Furthermore, the difference in lead frame capacitance between package types will also affect oscillation frequency, especially for low  $C_{ext}$  values. The user also needs to take into account variation due to tolerance of external R and C components used. Figure 14-6 shows how the R/C combination is connected to the PIC17CXX. For  $R_{ext}$  values below 2.2 k $\Omega$ , the oscillator operation may become unstable, or stop completely. For very high  $R_{ext}$  values (e.g. 1 M $\Omega$ ), the oscillator becomes sensitive to noise, humidity and leakage. Thus, we recommend to keep  $R_{ext}$  between 3 k $\Omega$  and 100 k $\Omega$ .

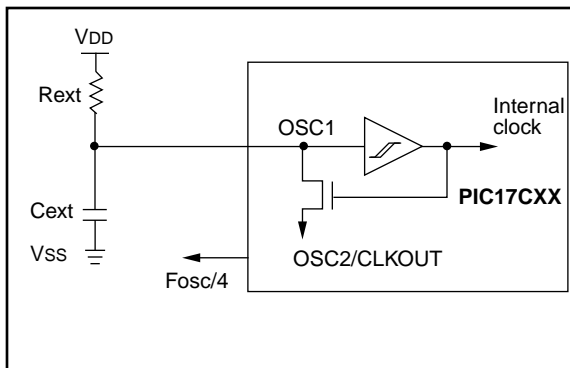
Although the oscillator will operate with no external capacitor ( $C_{ext} = 0$  pF), we recommend using values above 20 pF for noise and stability reasons. With little or no external capacitance, oscillation frequency can vary dramatically due to changes in external capacitances, such as PCB trace capacitance or package lead frame capacitance.

See Section 18.0 for RC frequency variation from part to part due to normal process variation. The variation is larger for larger R (since leakage current variation will affect RC frequency more for large R) and for smaller C (since variation of input capacitance will affect RC frequency more).

See Section 18.0 for variation of oscillator frequency due to  $V_{DD}$  for given  $R_{ext}/C_{ext}$  values as well as frequency variation due to operating temperature for given R, C, and  $V_{DD}$  values.

The oscillator frequency, divided by 4, is available on the OSC2/CLKOUT pin, and can be used for test purposes or to synchronize other logic (see Figure 3-2 for waveform).

**FIGURE 14-7: RC OSCILLATOR MODE**



## 14.4.2 MINIMIZING CURRENT CONSUMPTION

To minimize current consumption, all I/O pins should be either at VDD, or VSS, with no external circuitry drawing current from the I/O pin. I/O pins that are hi-impedance inputs should be pulled high or low externally to avoid switching currents caused by floating inputs. The T0CKI input should be at VDD or VSS. The contributions from on-chip pull-ups on PORTB should also be considered, and disabled when possible.

## 14.5 Code Protection

The code in the program memory can be protected by selecting the microcontroller in code protected mode (PM2:PM0 = '000').

**Note:** PM2 does not exist on the PIC17C42. To select code protected microcontroller mode, PM1:PM0 = '00'.

In this mode, instructions that are in the on-chip program memory space, can continue to read or write the program memory. An instruction that is executed outside of the internal program memory range will be inhibited from writing to or reading from program memory.

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

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

## 15.2 Q Cycle Activity

Each instruction cycle (Tcy) is comprised of four Q cycles (Q1-Q4). The Q cycles provide the timing/designation for the Decode, Read, Execute, Write etc., of each instruction cycle. The following diagram shows the relationship of the Q cycles to the instruction cycle.

The 4 Q cycles that make up an instruction cycle (Tcy) can be generalized as:

Q1: Instruction Decode Cycle or forced NOP

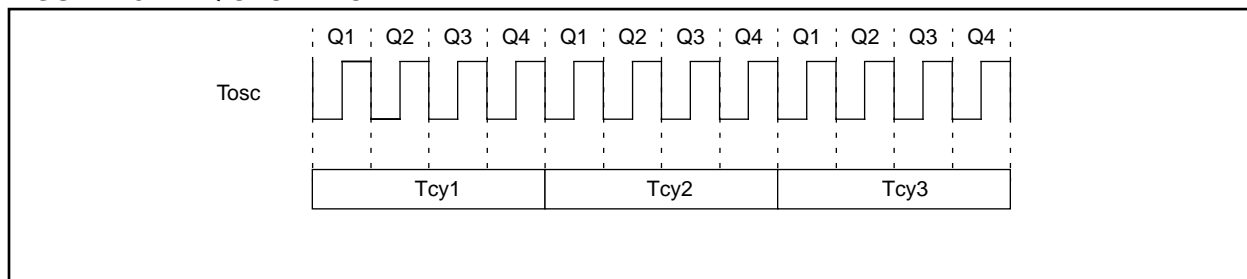
Q2: Instruction Read Cycle or NOP

Q3: Instruction Execute

Q4: Instruction Write Cycle or NOP

Each instruction will show the detailed Q cycle operation for the instruction.

**FIGURE 15-2: Q CYCLE ACTIVITY**



# PIC17C4X

## RETURN Return from Subroutine

Syntax: [ *label* ] RETURN

Operands: None

Operation: TOS → PC;

Status Affected: None

Encoding: 

0000	0000	0000	0010
------	------	------	------

Description: Return from subroutine. The stack is popped and the top of the stack (TOS) is loaded into the program counter.

Words: 1

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register PCL*	Execute	NOP
Forced NOP	NOP	Execute	NOP

\* Remember reading PCL causes PCLATH to be updated. This will be the high address of where the RETURN instruction is located.

**Example:** RETURN

After Interrupt  
PC = TOS

## RLCF Rotate Left f through Carry

Syntax: [ *label* ] RLCF f,d

Operands:  $0 \leq f \leq 255$   
 $d \in [0,1]$

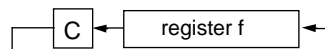
Operation:  $f\langle n \rangle \rightarrow d\langle n+1 \rangle$ ;  
 $f\langle 7 \rangle \rightarrow C$ ;  
 $C \rightarrow d\langle 0 \rangle$

Status Affected: C

Encoding: 

0001	101d	ffff	ffff
------	------	------	------

Description: The contents of register 'f' are rotated one bit to the left through the Carry Flag. If 'd' is 0 the result is placed in WREG. If 'd' is 1 the result is stored back in register 'f'.



Words: 1

Cycles: 1

Q Cycle Activity:

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

**Example:** RLCF REG, 0

Before Instruction

REG = 1110 0110  
C = 0

After Instruction

REG = 1110 0110  
WREG = 1100 1100  
C = 1

MPASM allow full symbolic debugging from the Microchip Universal Emulator System (PICMASTER).

MPASM has the following features to assist in developing software for specific use applications.

- Provides translation of Assembler source code to object code for all Microchip microcontrollers.
- Macro assembly capability.
- Produces all the files (Object, Listing, Symbol, and special) required for symbolic debug with Microchip's emulator systems.
- Supports Hex (default), Decimal and Octal source and listing formats.

MPASM provides a rich directive language to support programming of the PIC16/17. Directives are helpful in making the development of your assemble source code shorter and more maintainable.

## **16.11 Software Simulator (MPLAB-SIM)**

The MPLAB-SIM Software Simulator allows code development in a PC host environment. It allows the user to simulate the PIC16/17 series microcontrollers on an instruction level. On any given instruction, the user may examine or modify any of the data areas or provide external stimulus to any of the pins. The input/output radix can be set by the user and the execution can be performed in; single step, execute until break, or in a trace mode.

MPLAB-SIM fully supports symbolic debugging using MPLAB-C and MPASM. The Software Simulator offers the low cost flexibility to develop and debug code outside of the laboratory environment making it an excellent multi-project software development tool.

## **16.12 C Compiler (MPLAB-C)**

The MPLAB-C Code Development System is a complete 'C' compiler and integrated development environment for Microchip's PIC16/17 family of microcontrollers. The compiler provides powerful integration capabilities and ease of use not found with other compilers.

For easier source level debugging, the compiler provides symbol information that is compatible with the MPLAB IDE memory display (PICMASTER emulator software versions 1.13 and later).

## **16.13 Fuzzy Logic Development System (fuzzyTECH-MP)**

*fuzzyTECH-MP* fuzzy logic development tool is available in two versions - a low cost introductory version, MP Explorer, for designers to gain a comprehensive working knowledge of fuzzy logic system design; and a full-featured version, *fuzzyTECH-MP*, edition for implementing more complex systems.

Both versions include Microchip's *fuzzyLAB™* demonstration board for hands-on experience with fuzzy logic systems implementation.

## **16.14 MP-DriveWay™ – Application Code Generator**

MP-DriveWay is an easy-to-use Windows-based Application Code Generator. With MP-DriveWay you can visually configure all the peripherals in a PIC16/17 device and, with a click of the mouse, generate all the initialization and many functional code modules in C language. The output is fully compatible with Microchip's MPLAB-C C compiler. The code produced is highly modular and allows easy integration of your own code. MP-DriveWay is intelligent enough to maintain your code through subsequent code generation.

## **16.15 SEEVAL® Evaluation and Programming System**

The SEEVAL SEEPROM Designer's Kit supports all Microchip 2-wire and 3-wire Serial EEPROMs. The kit includes everything necessary to read, write, erase or program special features of any Microchip SEEPROM product including Smart Serials™ and secure serials. The Total Endurance™ Disk is included to aid in trade-off analysis and reliability calculations. The total kit can significantly reduce time-to-market and result in an optimized system.

## **16.16 TrueGauge® Intelligent Battery Management**

The TrueGauge development tool supports system development with the MTA11200B TrueGauge Intelligent Battery Management IC. System design verification can be accomplished before hardware prototypes are built. User interface is graphically-oriented and measured data can be saved in a file for exporting to Microsoft Excel.

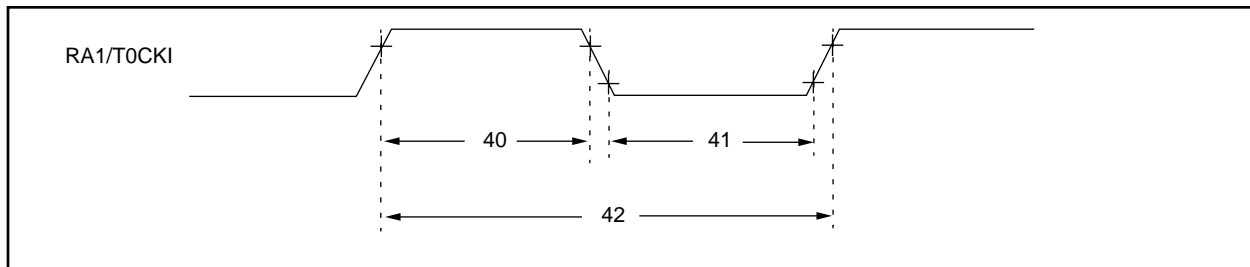
## **16.17 KEELOQ® Evaluation and Programming Tools**

KEELOQ evaluation and programming tools support Microchips HCS Secure Data Products. The HCS evaluation kit includes an LCD display to show changing codes, a decoder to decode transmissions, and a programming interface to program test transmitters.

# PIC17C4X

Applicable Devices 42 R42 42A 43 R43 44

**FIGURE 17-5: TIMER0 CLOCK TIMINGS**



**TABLE 17-5: TIMER0 CLOCK REQUIREMENTS**

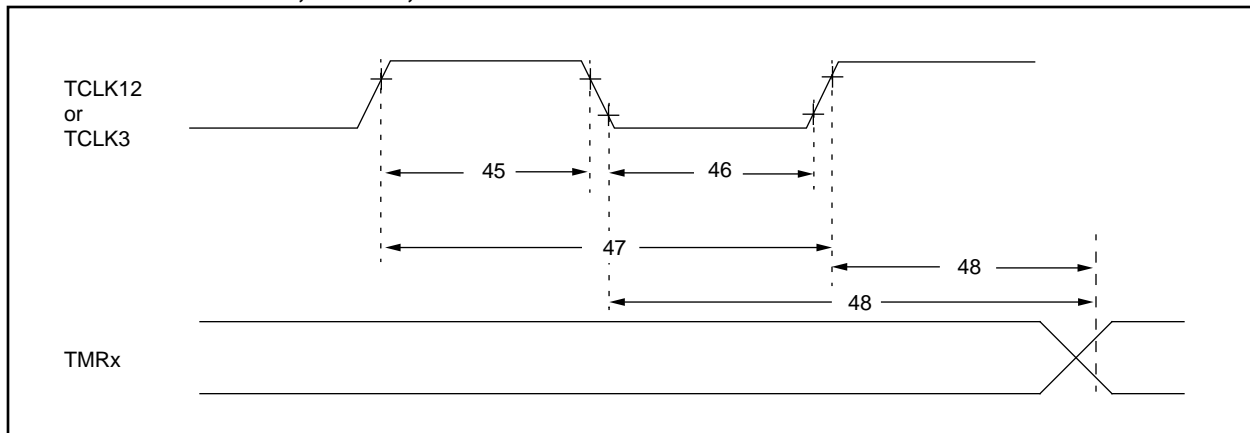
Parameter No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
40	Tt0H	T0CKI High Pulse Width	No Prescaler	0.5Tcy + 20 §	—	—	ns	
			With Prescaler	10*	—	—	ns	
41	Tt0L	T0CKI Low Pulse Width	No Prescaler	0.5Tcy + 20 §	—	—	ns	
			With Prescaler	10*	—	—	ns	
42	Tt0P	T0CKI Period		$\frac{Tcy + 40}{N}$ §	—	—	ns	N = prescale value (1, 2, 4, ..., 256)

\* 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.

§ This specification ensured by design.

**FIGURE 17-6: TIMER1, TIMER2, AND TIMER3 CLOCK TIMINGS**



**TABLE 17-6: TIMER1, TIMER2, AND TIMER3 CLOCK REQUIREMENTS**

Parameter No.	Sym	Characteristic	Min	Typ †	Max	Units	Conditions
45	Tt123H	TCLK12 and TCLK3 high time	0.5 Tcy + 20 §	—	—	ns	
46	Tt123L	TCLK12 and TCLK3 low time	0.5 Tcy + 20 §	—	—	ns	
47	Tt123P	TCLK12 and TCLK3 input period	$\frac{Tcy + 40}{N}$ §	—	—	ns	N = prescale value (1, 2, 4, 8)
48	TckE2tmrl	Delay from selected External Clock Edge to Timer increment	2Tosc §	—	6 Tosc §	—	

\* 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.

§ This specification ensured by design.

# PIC17C4X

Applicable Devices 42 R42 42A 43 R43 44

FIGURE 18-9: TYPICAL  $I_{PD}$  vs.  $V_{DD}$  WATCHDOG DISABLED 25°C

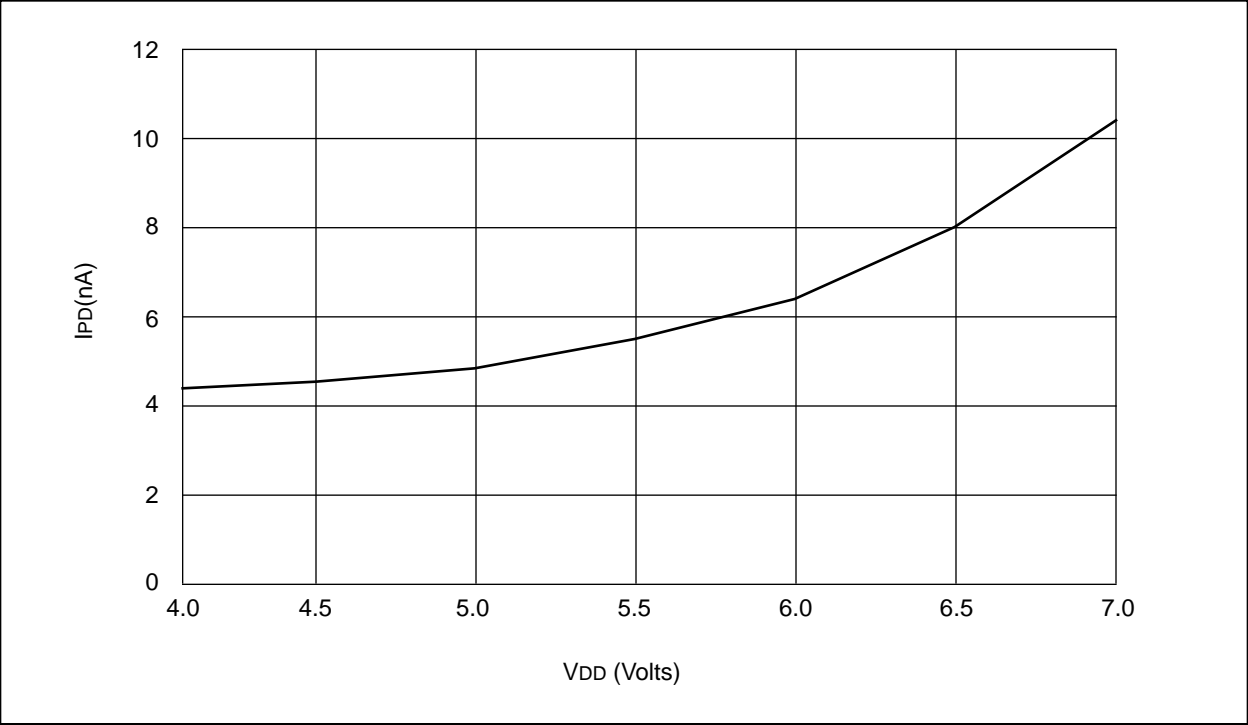
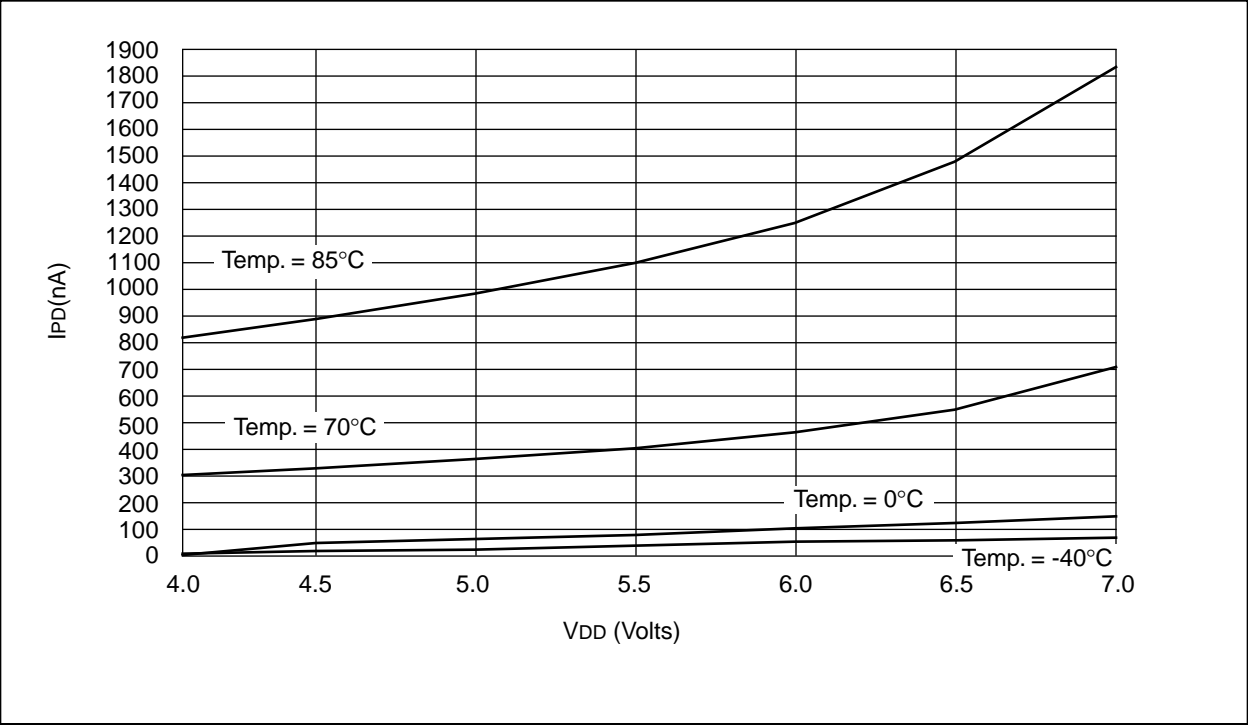
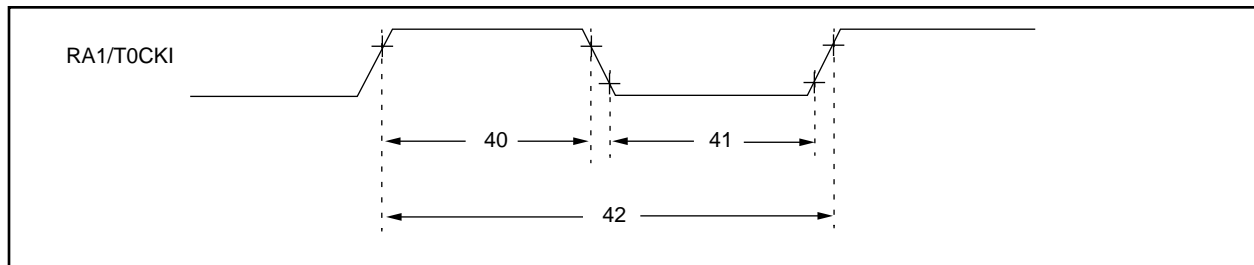


FIGURE 18-10: MAXIMUM  $I_{PD}$  vs.  $V_{DD}$  WATCHDOG DISABLED



**FIGURE 19-5: TIMER0 CLOCK TIMINGS**



**TABLE 19-5: TIMER0 CLOCK REQUIREMENTS**

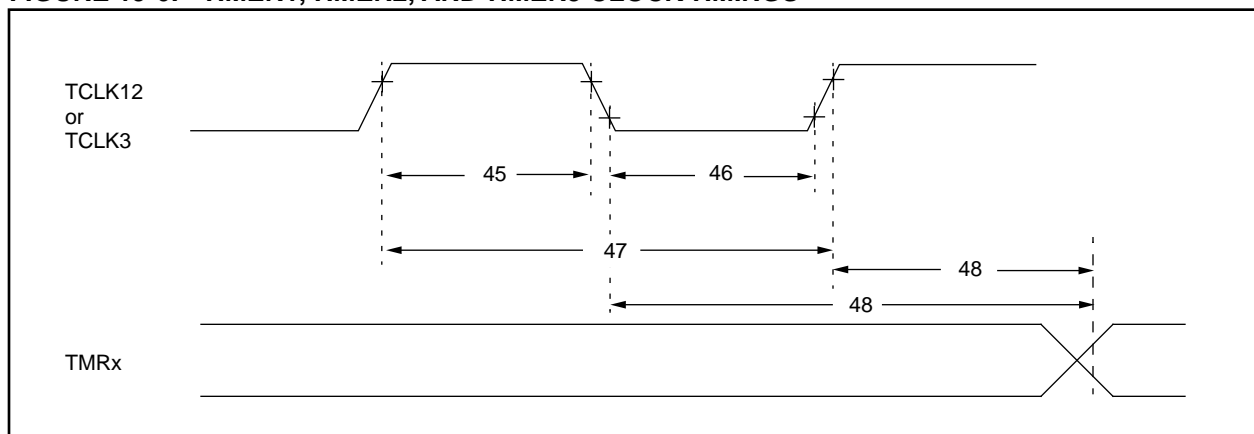
Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
40	Tt0H	T0CKI High Pulse Width	No Prescaler	0.5Tcy + 20 §	—	—	ns
		With Prescaler	10*	—	—	ns	
41	Tt0L	T0CKI Low Pulse Width	No Prescaler	0.5Tcy + 20 §	—	—	ns
		With Prescaler	10*	—	—	ns	
42	Tt0P	T0CKI Period	Greater of: 20 ns or $\frac{Tcy + 40 §}{N}$	—	—	ns	N = prescale value (1, 2, 4, ..., 256)

\* 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.

§ This specification ensured by design.

**FIGURE 19-6: TIMER1, TIMER2, AND TIMER3 CLOCK TIMINGS**



**TABLE 19-6: TIMER1, TIMER2, AND TIMER3 CLOCK REQUIREMENTS**

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
45	Tt123H	TCLK12 and TCLK3 high time	0.5Tcy + 20 §	—	—	ns	
46	Tt123L	TCLK12 and TCLK3 low time	0.5Tcy + 20 §	—	—	ns	
47	Tt123P	TCLK12 and TCLK3 input period	$\frac{Tcy + 40 §}{N}$	—	—	ns	N = prescale value (1, 2, 4, 8)
48	TckE2tmr1	Delay from selected External Clock Edge to Timer increment	2Tosc §		6Tosc §		

\* 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.

§ This specification ensured by design.

# PIC17C4X

## E.4 PIC16C6X Family of Devices

	Clock		Memory		Peripherals				Features																															
	Maximum Frequency of Operation (MHz)		Program Memory (K*14 words)		Timer Modules				Serial Ports (SPI/I <sup>2</sup> C, USART)				Parallel Slave Port				Interrupt Sources				I/O Pins				Voltage Range (Volts)				In-Circuit Serial Programming				Brown-out Reset				Packages			
EPROM		Data Memory (bytes)		Capture/Compare/PWM Modules				Serial Ports (SPI/I <sup>2</sup> C, USART)				Parallel Slave Port				Interrupt Sources				I/O Pins				Voltage Range (Volts)				In-Circuit Serial Programming				Brown-out Reset				Packages				
20	2K	—	128	TMR0, TMR1, TMR2	1	SPI/I <sup>2</sup> C	—	7	22	3.0-6.0	Yes	—	28-pin SDIP, SOIC, SSOP																											
PIC16C62	20	2K	—	128	TMR0, TMR1, TMR2	1	SPI/I <sup>2</sup> C	—	7	22	2.5-6.0	Yes	Yes	28-pin SDIP, SOIC, SSOP																										
PIC16CR62 <sup>(1)</sup>	20	—	2K	128	TMR0, TMR1, TMR2	1	SPI/I <sup>2</sup> C	—	7	22	2.5-6.0	Yes	Yes	28-pin SDIP, SOIC, SSOP																										
PIC16C63	20	4K	—	192	TMR0, TMR1, TMR2	2	SPI/I <sup>2</sup> C, USART	—	10	22	2.5-6.0	Yes	Yes	28-pin SDIP, SOIC																										
PIC16CR63 <sup>(1)</sup>	20	—	4K	192	TMR0, TMR1, TMR2	2	SPI/I <sup>2</sup> C, USART	—	10	22	2.5-6.0	Yes	Yes	28-pin SDIP, SOIC																										
PIC16C64	20	2K	—	128	TMR0, TMR1, TMR2	1	SPI/I <sup>2</sup> C	Yes	8	33	3.0-6.0	Yes	—	40-pin DIP; 44-pin PLCC, MQFP																										
PIC16C64A <sup>(1)</sup>	20	2K	—	128	TMR0, TMR1, TMR2	1	SPI/I <sup>2</sup> C	Yes	8	33	2.5-6.0	Yes	Yes	40-pin DIP; 44-pin PLCC, MQFP, TQFP																										
PIC16CR64 <sup>(1)</sup>	20	—	2K	128	TMR0, TMR1, TMR2	1	SPI/I <sup>2</sup> C	Yes	8	33	2.5-6.0	Yes	Yes	40-pin DIP; 44-pin PLCC, MQFP, TQFP																										
PIC16C65	20	4K	—	192	TMR0, TMR1, TMR2	2	SPI/I <sup>2</sup> C, USART	Yes	11	33	3.0-6.0	Yes	—	40-pin DIP; 44-pin PLCC, MQFP																										
PIC16C65A <sup>(1)</sup>	20	4K	—	192	TMR0, TMR1, TMR2	2	SPI/I <sup>2</sup> C, USART	Yes	11	33	2.5-6.0	Yes	Yes	40-pin DIP; 44-pin PLCC, MQFP, TQFP																										
PIC16CR65 <sup>(1)</sup>	20	—	4K	192	TMR0, TMR1, TMR2	2	SPI/I <sup>2</sup> C, USART	Yes	11	33	2.5-6.0	Yes	Yes	40-pin DIP; 44-pin PLCC, MQFP, TQFP																										

All PIC16/17 family devices have Power-on Reset, selectable Watchdog Timer, selectable code protect, and high I/O current capability.

All PIC16C6X family devices use serial programming with clock pin RB6 and data pin RB7.

Note 1: Please contact your local sales office for availability of these devices.

# PIC17C4X

---

NOTES:

# PIC17C4X

MP-C C Compiler .....	145
MPSIM Software Simulator .....	143, 145
MULLW .....	129
Multiply Examples	
16 x 16 Routine .....	50
16 x 16 Signed Routine .....	51
8 x 8 Routine .....	49
8 x 8 Signed Routine .....	49
MULWF .....	129

## N

NEGW .....	130
NOP .....	130

## O

OCERR .....	84
Opcode Field Descriptions .....	107
OSC Selection .....	99
Oscillator	
Configuration .....	100
Crystal .....	100
External Clock .....	101
External Crystal Circuit .....	102
External Parallel Resonant Crystal Circuit .....	102
External Series Resonant Crystal Circuit .....	102
RC .....	102
RC Frequencies .....	165, 195
Oscillator Start-up Time (Figure) .....	18
Oscillator Start-up Timer (OST) .....	15, 99
OST .....	15, 99
OV .....	9, 36
Overflow (OV) .....	9

## P

Package Marking Information .....	210
Packaging Information .....	205
Parameter Measurement Information .....	154
PC (Program Counter) .....	41
PCH .....	41
PCL .....	34, 41, 108
PCLATH .....	34, 41
$\overline{PD}$ .....	37, 105
PEIE .....	22, 78
PEIF .....	22
Peripheral Bank .....	42
Peripheral Interrupt Enable .....	23
Peripheral Interrupt Request (PIR) .....	24
PICDEM-1 Low-Cost PIC16/17 Demo Board .....	143, 144
PICDEM-2 Low-Cost PIC16CXX Demo Board .....	143, 144
PICDEM-3 Low-Cost PIC16C9XXX Demo Board .....	144
PICMASTER <sup>®</sup> RT In-Circuit Emulator .....	143
PICSTART <sup>®</sup> Low-Cost Development System .....	143
PIE .....	19, 34, 92, 96, 98
Pin Compatible Devices .....	221
PIR .....	19, 34, 92, 96, 98
PM0 .....	99, 106
PM1 .....	99, 106
POP .....	27, 39
POR .....	15, 99
PORTA .....	19, 34, 53
PORTB .....	19, 34, 55
PORTC .....	19, 34, 58

PORTD .....	19, 34, 60
PORTE .....	19, 34, 62
Power-down Mode .....	105
Power-on Reset (POR) .....	15, 99
Power-up Timer (PWRT) .....	15, 99
PR1 .....	20, 35
PR2 .....	20, 35
PR3/CA1H .....	20
PR3/CA1L .....	20
PR3H/CA1H .....	35
PR3L/CA1L .....	35
Prescaler Assignments .....	69
PRO MATE <sup>®</sup> Universal Programmer .....	143
PRODH .....	20
PRODL .....	20
Program Counter (PC) .....	41
Program Memory	
External Access Waveforms .....	31
External Connection Diagram .....	31
Map .....	29
Modes	
Extended Microcontroller .....	29
Microcontroller .....	29
Microprocessor .....	29
Protected Microcontroller .....	29
Operation .....	29
Organization .....	29
Transfers from Data Memory .....	43
Protected Microcontroller .....	29
PS0 .....	38, 67
PS1 .....	38, 67
PS2 .....	38, 67
PS3 .....	38, 67
PUSH .....	27, 39
PW1DCH .....	20, 35
PW1DCL .....	20, 35
PW2DCH .....	20, 35
PW2DCL .....	20, 35
PWM .....	71, 75
Duty Cycle .....	76
External Clock Source .....	76
Frequency vs. Resolution .....	76
Interrupts .....	76
Max Resolution/Frequency for External	
Clock Input .....	77
Output .....	75
Periods .....	76
PWM1 .....	72
PWM1ON .....	72, 75
PWM2 .....	72
PWM2ON .....	72, 75
PWRT .....	15, 99

## R

RA1/T0CKI pin .....	67
RBIE .....	23
RBIF .....	24
RBPJ .....	55
RC Oscillator .....	102
RC Oscillator Frequencies .....	165, 195
RCIE .....	23
RCIF .....	24
RCREG .....	19, 34, 91, 92, 96, 97
RCSTA .....	19, 34, 92, 96, 98
Reading 16-bit Value .....	69