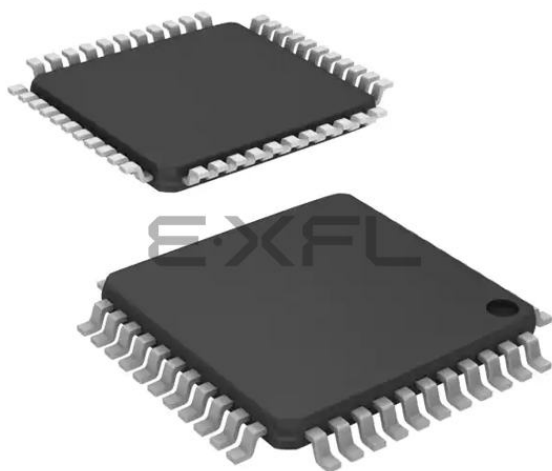




Welcome to [E-XFL.COM](https://www.e-xfl.com)

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	25MHz
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	-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/pic17c43t-25i-pt

3.1 Clocking Scheme/Instruction Cycle

The clock input (from OSC1) is internally divided by four to generate four non-overlapping quadrature clocks, namely Q1, Q2, Q3, and Q4. Internally, the program counter (PC) is incremented every Q1, and the instruction is fetched from the program memory and latched into the instruction register in Q4. The instruction is decoded and executed during the following Q1 through Q4. The clocks and instruction execution flow are shown in Figure 3-3.

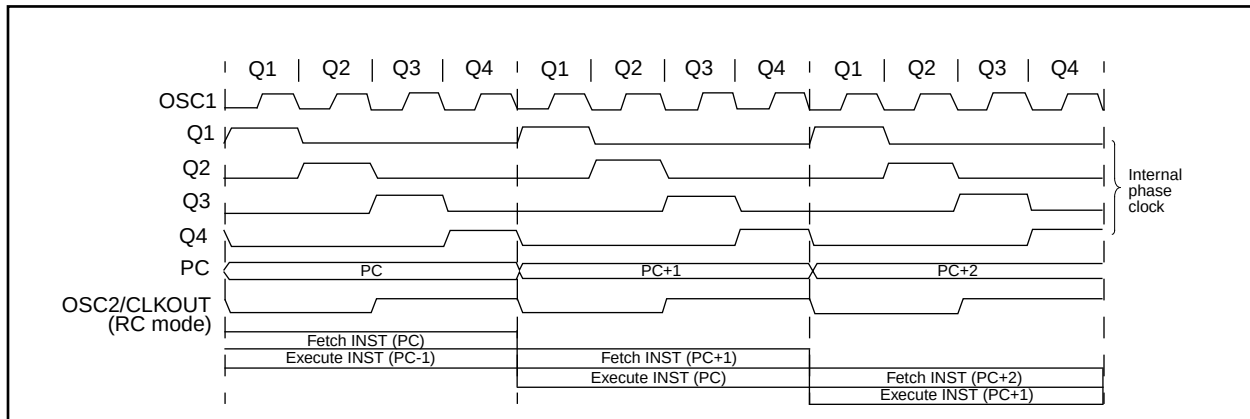
3.2 Instruction Flow/Pipelining

An "Instruction Cycle" consists of four Q cycles (Q1, Q2, Q3, and Q4). The instruction fetch and execute are pipelined such that fetch takes one instruction cycle while decode and execute takes another instruction cycle. However, due to the pipelining, each instruction effectively executes in one cycle. If an instruction causes the program counter to change (e.g. GOTO) then two cycles are required to complete the instruction (Example 3-2).

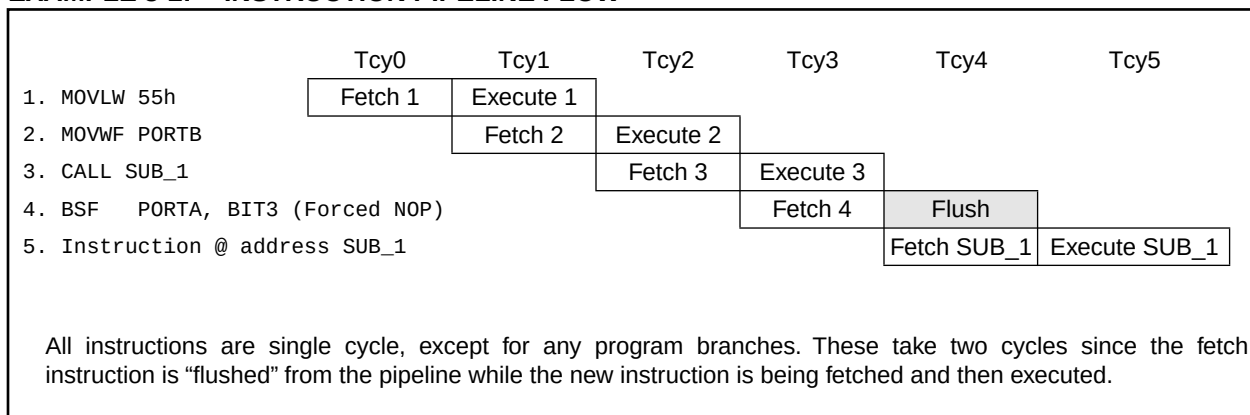
A fetch cycle begins with the program counter incrementing in Q1.

In the execution cycle, the fetched instruction is latched into the "Instruction Register (IR)" in cycle Q1. This instruction is then decoded and executed during the Q2, Q3, and Q4 cycles. Data memory is read during Q2 (operand read) and written during Q4 (destination write).

FIGURE 3-3: CLOCK/INSTRUCTION CYCLE



EXAMPLE 3-2: INSTRUCTION PIPELINE FLOW



5.3 Peripheral Interrupt Request Register (PIR)

This register contains the individual flag bits for the peripheral interrupts.

Note: These bits will be set by the specified condition, even if the corresponding interrupt enable bit is cleared (interrupt disabled), or the GLINTD bit is set (all interrupts disabled). Before enabling an interrupt, the user may wish to clear the interrupt flag to ensure that the program does not immediately branch to the peripheral interrupt service routine.

FIGURE 5-4: PIR REGISTER (ADDRESS: 16h, BANK 1)

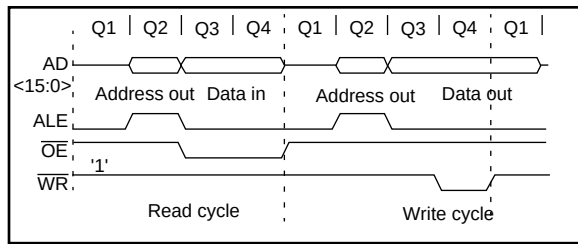
R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	R - 1	R - 0
RBIF	TMR3IF	TMR2IF	TMR1IF	CA2IF	CA1IF	TXIF	RCIF
bit7							bit0
bit 7: RBIF: PORTB Interrupt on Change Flag bit 1 = One of the PORTB inputs changed (Software must end the mismatch condition) 0 = None of the PORTB inputs have changed bit 6: TMR3IF: Timer3 Interrupt Flag bit If Capture1 is enabled ($CA1/\overline{PR3} = 1$) 1 = Timer3 overflowed 0 = Timer3 did not overflow If Capture1 is disabled ($CA1/\overline{PR3} = 0$) 1 = Timer3 value has rolled over to 0000h from equalling the period register (PR3H:PR3L) value 0 = Timer3 value has not rolled over to 0000h from equalling the period register (PR3H:PR3L) value bit 5: TMR2IF: Timer2 Interrupt Flag bit 1 = Timer2 value has rolled over to 0000h from equalling the period register (PR2) value 0 = Timer2 value has not rolled over to 0000h from equalling the period register (PR2) value bit 4: TMR1IF: Timer1 Interrupt Flag bit If Timer1 is in 8-bit mode ($T16 = 0$) 1 = Timer1 value has rolled over to 0000h from equalling the period register (PR) value 0 = Timer1 value has not rolled over to 0000h from equalling the period register (PR2) value If Timer1 is in 16-bit mode ($T16 = 1$) 1 = TMR1:TMR2 value has rolled over to 0000h from equalling the period register (PR1:PR2) value 0 = TMR1:TMR2 value has not rolled over to 0000h from equalling the period register (PR1:PR2) value bit 3: CA2IF: Capture2 Interrupt Flag bit 1 = Capture event occurred on RB1/CAP2 pin 0 = Capture event did not occur on RB1/CAP2 pin bit 2: CA1IF: Capture1 Interrupt Flag bit 1 = Capture event occurred on RB0/CAP1 pin 0 = Capture event did not occur on RB0/CAP1 pin bit 1: TXIF: USART Transmit Interrupt Flag bit 1 = Transmit buffer is empty 0 = Transmit buffer is full bit 0: RCIF: USART Receive Interrupt Flag bit 1 = Receive buffer is full 0 = Receive buffer is empty							

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

6.1.2 EXTERNAL MEMORY INTERFACE

When either microprocessor or extended microcontroller mode is selected, PORTC, PORTD and PORTE are configured as the system bus. PORTC and PORTD are the multiplexed address/data bus and PORTE is for the control signals. External components are needed to demultiplex the address and data. This can be done as shown in Figure 6-4. The waveforms of address and data are shown in Figure 6-3. For complete timings, please refer to the electrical specification section.

FIGURE 6-3: EXTERNAL PROGRAM MEMORY ACCESS WAVEFORMS



The system bus requires that there is no bus conflict (minimal leakage), so the output value (address) will be capacitively held at the desired value.

As the speed of the processor increases, external EPROM memory with faster access time must be used. Table 6-2 lists external memory speed requirements for a given PIC17C4X device frequency.

In extended microcontroller mode, when the device is executing out of internal memory, the control signals will continue to be active. That is, they indicate the action that is occurring in the internal memory. The external memory access is ignored.

This following selection is for use with Microchip EPROMs. For interfacing to other manufacturers memory, please refer to the electrical specifications of the desired PIC17C4X device, as well as the desired memory device to ensure compatibility.

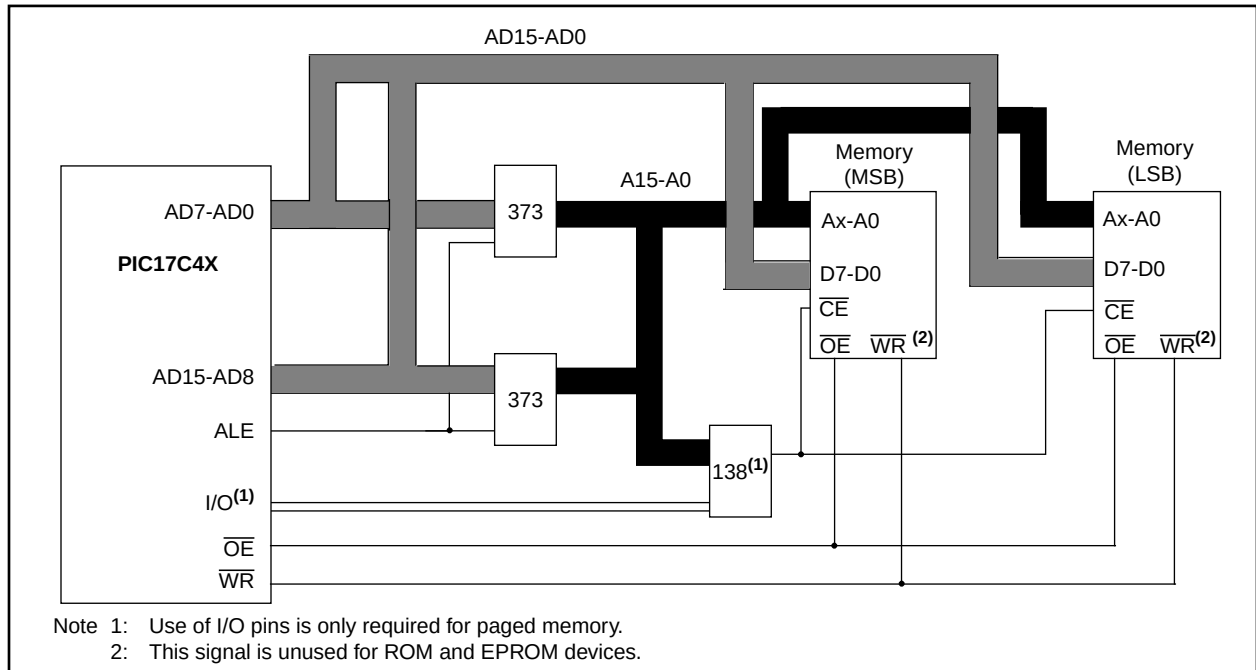
TABLE 6-2: EPROM MEMORY ACCESS TIME ORDERING SUFFIX

PIC17C4X Oscillator Frequency	Instruction Cycle Time (Tcy)	EPROM Suffix	
		PIC17C42	PIC17C43 PIC17C44
8 MHz	500 ns	-25	-25
16 MHz	250 ns	-12	-15
20 MHz	200 ns	-90	-10
25 MHz	160 ns	N.A.	-70
33 MHz	121 ns	N.A.	(1)

Note 1: The access times for this requires the use of fast SRAMS.

Note: The external memory interface is not supported for the LC devices.

FIGURE 6-4: TYPICAL EXTERNAL PROGRAM MEMORY CONNECTION DIAGRAM



6.2 Data Memory Organization

Data memory is partitioned into two areas. The first is the General Purpose Registers (GPR) area, while the second is the Special Function Registers (SFR) area. The SFRs control the operation of the device.

Portions of data memory are banked, this is for both areas. The GPR area is banked to allow greater than 232 bytes of general purpose RAM. SFRs are for the registers that control the peripheral functions. Banking requires the use of control bits for bank selection. These control bits are located in the Bank Select Register (BSR). If an access is made to a location outside this banked region, the BSR bits are ignored. Figure 6-5 shows the data memory map organization for the PIC17C42 and Figure 6-6 for all of the other PIC17C4X devices.

Instructions MOVPF and MOVFP provide the means to move values from the peripheral area ("P") to any location in the register file ("F"), and vice-versa. The definition of the "P" range is from 0h to 1Fh, while the "F" range is 0h to FFh. The "P" range has six more locations than peripheral registers (eight locations for the PIC17C42 device) which can be used as General Purpose Registers. This can be useful in some applications where variables need to be copied to other locations in the general purpose RAM (such as saving status information during an interrupt).

The entire data memory can be accessed either directly or indirectly through file select registers FSR0 and FSR1 (Section 6.4). Indirect addressing uses the appropriate control bits of the BSR for accesses into the banked areas of data memory. The BSR is explained in greater detail in Section 6.8.

6.2.1 GENERAL PURPOSE REGISTER (GPR)

All devices have some amount of GPR area. The GPRs are 8-bits wide. When the GPR area is greater than 232, it must be banked to allow access to the additional memory space.

Only the PIC17C43 and PIC17C44 devices have banked memory in the GPR area. To facilitate switching between these banks, the MOVLRL bank instruction has been added to the instruction set. GPRs are not initialized by a Power-on Reset and are unchanged on all other resets.

6.2.2 SPECIAL FUNCTION REGISTERS (SFR)

The SFRs are used by the CPU and peripheral functions to control the operation of the device (Figure 6-5 and Figure 6-6). These registers are static RAM.

The SFRs can be classified into two sets, those associated with the "core" function and those related to the peripheral functions. Those registers related to the "core" are described here, while those related to a peripheral feature are described in the section for each peripheral feature.

The peripheral registers are in the banked portion of memory, while the core registers are in the unbanked region. To facilitate switching between the peripheral banks, the MOVLLB bank instruction has been provided.

6.4.1 INDIRECT ADDRESSING REGISTERS

The PIC17C4X has four registers for indirect addressing. These registers are:

- INDF0 and FSR0
- INDF1 and FSR1

Registers INDF0 and INDF1 are not physically implemented. Reading or writing to these registers activates indirect addressing, with the value in the corresponding FSR register being the address of the data. The FSR is an 8-bit register and allows addressing anywhere in the 256-byte data memory address range. For banked memory, the bank of memory accessed is specified by the value in the BSR.

If file INDF0 (or INDF1) itself is read indirectly via an FSR, all '0's are read (Zero bit is set). Similarly, if INDF0 (or INDF1) is written to indirectly, the operation will be equivalent to a NOP, and the status bits are not affected.

6.4.2 INDIRECT ADDRESSING OPERATION

The indirect addressing capability has been enhanced over that of the PIC16CXX family. There are two control bits associated with each FSR register. These two bits configure the FSR register to:

- Auto-decrement the value (address) in the FSR after an indirect access
- Auto-increment the value (address) in the FSR after an indirect access
- No change to the value (address) in the FSR after an indirect access

These control bits are located in the ALUSTA register. The FSR1 register is controlled by the FS3:FS2 bits and FSR0 is controlled by the FS1:FS0 bits.

When using the auto-increment or auto-decrement features, the effect on the FSR is not reflected in the ALUSTA register. For example, if the indirect address causes the FSR to equal '0', the Z bit will not be set.

If the FSR register contains a value of 0h, an indirect read will read 0h (Zero bit is set) while an indirect write will be equivalent to a NOP (status bits are not affected).

Indirect addressing allows single cycle data transfers within the entire data space. This is possible with the use of the MOVPF and MOVFP instructions, where either 'p' or 'f' is specified as INDF0 (or INDF1).

If the source or destination of the indirect address is in banked memory, the location accessed will be determined by the value in the BSR.

A simple program to clear RAM from 20h - FFh is shown in Example 6-1.

EXAMPLE 6-1: INDIRECT ADDRESSING

```

MOVLW    0x20      ;
MOVWF     FSR0      ; FSR0 = 20h
BCF       ALUSTA, FS1 ; Increment FSR
BSF       ALUSTA, FS0 ; after access
BCF       ALUSTA, C   ; C = 0
MOVLW     END_RAM + 1 ;
LP CLRf     INDF0      ; Addr(FSR) = 0
CPFSEQ    FSR0      ; FSR0 = END_RAM+1?
GOTO      LP         ; NO, clear next
:         ; YES, All RAM is
:         ; cleared

```

6.5 Table Pointer (TBLPTRL and TBLPTRH)

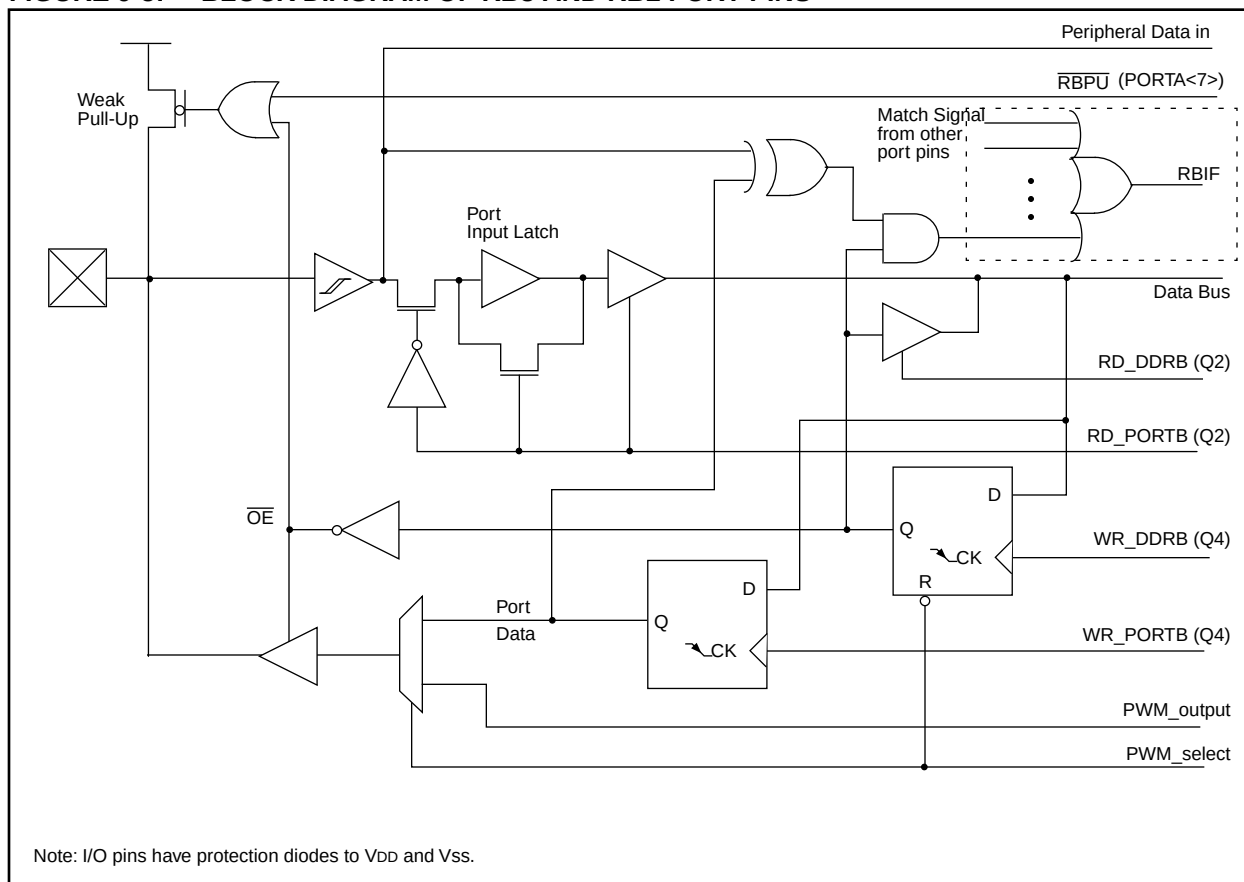
File registers TBLPTRL and TBLPTRH form a 16-bit pointer to address the 64K program memory space. The table pointer is used by instructions TABLWT and TABLRD.

The TABLRD and the TABLWT instructions allow transfer of data between program and data space. The table pointer serves as the 16-bit address of the data word within the program memory. For a more complete description of these registers and the operation of Table Reads and Table Writes, see Section 7.0.

6.6 Table Latch (TBLATH, TBLATL)

The table latch (TBLAT) is a 16-bit register, with TBLATH and TBLATL referring to the high and low bytes of the register. It is not mapped into data or program memory. The table latch is used as a temporary holding latch during data transfer between program and data memory (see descriptions of instructions TABLRD, TABLWT, TLRD and TLWT). For a more complete description of these registers and the operation of Table Reads and Table Writes, see Section 7.0.

FIGURE 9-5: BLOCK DIAGRAM OF RB3 AND RB2 PORT PINS



11.1 Timer0 Operation

When the T0CS (T0STA<5>) bit is set, TMR0 increments on the internal clock. When T0CS is clear, TMR0 increments on the external clock (RA1/T0CKI pin). The external clock edge can be configured in software. When the T0SE (T0STA<6>) bit is set, the timer will increment on the rising edge of the RA1/T0CKI pin. When T0SE is clear, the timer will increment on the falling edge of the RA1/T0CKI pin. The prescaler can be programmed to introduce a prescale of 1:1 to 1:256. The timer increments from 0000h to FFFFh and rolls over to 0000h. On overflow, the TMR0 Interrupt Flag bit (T0IF) is set. The TMR0 interrupt can be masked by clearing the corresponding TMR0 Interrupt Enable bit (T0IE). The TMR0 Interrupt Flag bit (T0IF) is automatically cleared when vectoring to the TMR0 interrupt vector.

11.2 Using Timer0 with External Clock

When the external clock input is used for Timer0, it is synchronized with the internal phase clocks. Figure 11-3 shows the synchronization of the external clock. This synchronization is done after the prescaler. The output of the prescaler (PSOUT) is sampled twice in every instruction cycle to detect a rising or a falling edge. The timing requirements for the external clock are detailed in the electrical specification section for the desired device.

11.2.1 DELAY FROM EXTERNAL CLOCK EDGE

Since the prescaler output is synchronized with the internal clocks, there is a small delay from the time the external clock edge occurs to the time TMR0 is actually incremented. Figure 11-3 shows that this delay is between $3T_{osc}$ and $7T_{osc}$. Thus, for example, measuring the interval between two edges (e.g. period) will be accurate within $\pm 4T_{osc}$ (± 121 ns @ 33 MHz).

FIGURE 11-2: TIMER0 MODULE BLOCK DIAGRAM

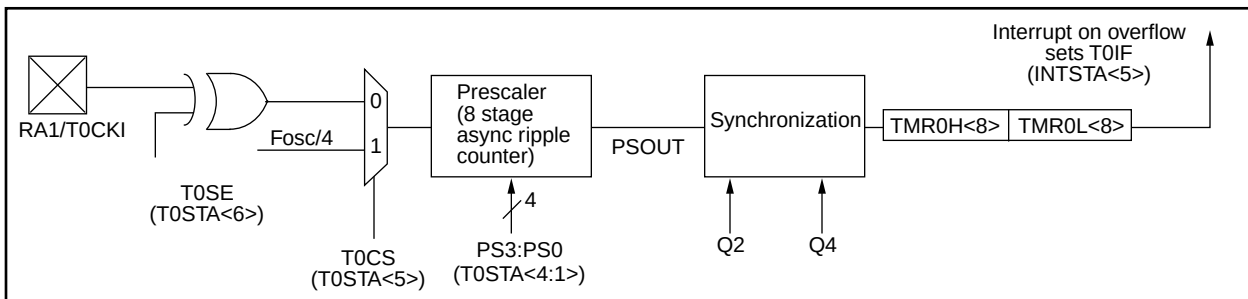
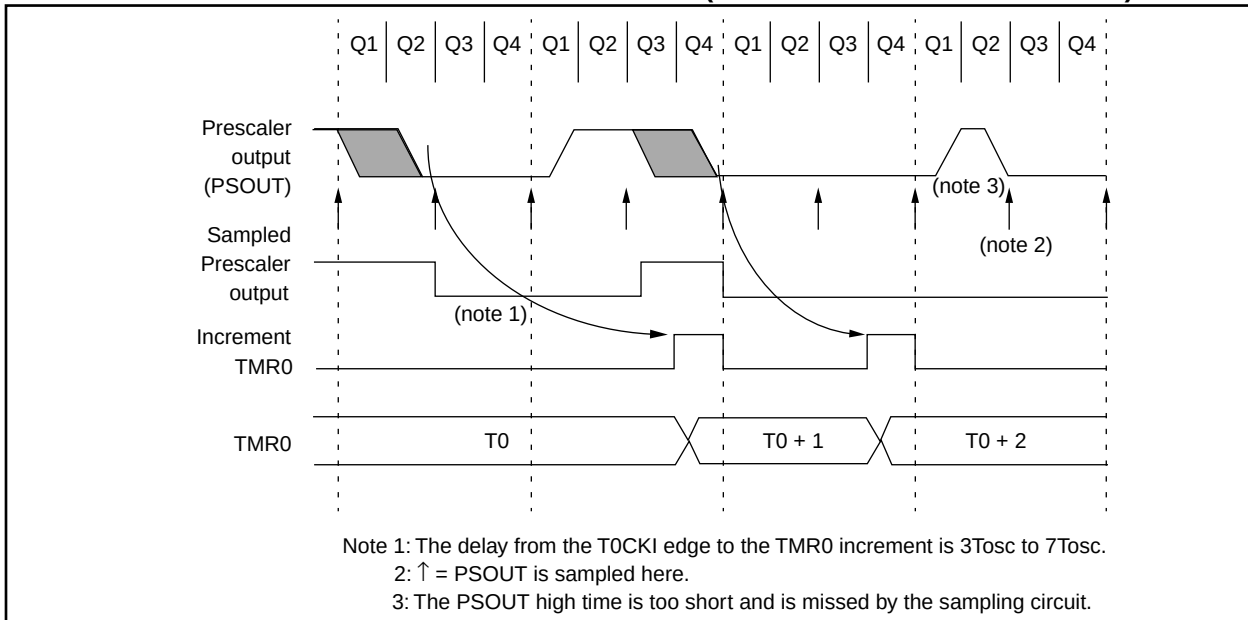


FIGURE 11-3: TMR0 TIMING WITH EXTERNAL CLOCK (INCREMENT ON FALLING EDGE)



12.0 TIMER1, TIMER2, TIMER3, PWMS AND CAPTURES

The PIC17C4X has a wealth of timers and time-based functions to ease the implementation of control applications. These time-base functions include two PWM outputs and two Capture inputs.

Timer1 and Timer2 are two 8-bit incrementing timers, each with a period register (PR1 and PR2 respectively) and separate overflow interrupt flags. Timer1 and Timer2 can operate either as timers (increment on internal Fosc/4 clock) or as counters (increment on falling edge of external clock on pin RB4/TCLK12). They are also software configurable to operate as a single 16-bit timer. These timers are also used as the time-base for the PWM (pulse width modulation) module.

Timer3 is a 16-bit timer/counter consisting of the TMR3H and TMR3L registers. This timer has four other associated registers. Two registers are used as a 16-bit period register or a 16-bit Capture1 register (PR3H/CA1H:PR3L/CA1L). The other two registers are strictly the Capture2 registers (CA2H:CA2L). Timer3 is the time-base for the two 16-bit captures.

TMR3 can be software configured to increment from the internal system clock or from an external signal on the RB5/TCLK3 pin.

Figure 12-1 and Figure 12-2 are the control registers for the operation of Timer1, Timer2, and Timer3, as well as PWM1, PWM2, Capture1, and Capture2.

FIGURE 12-1: TCON1 REGISTER (ADDRESS: 16h, BANK 3)

R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0
CA2ED1	CA2ED0	CA1ED1	CA1ED0	T16	TMR3CS	TMR2CS	TMR1CS
bit7							bit0
bit 7-6: CA2ED1:CA2ED0: Capture2 Mode Select bits 00 = Capture on every falling edge 01 = Capture on every rising edge 10 = Capture on every 4th rising edge 11 = Capture on every 16th rising edge bit 5-4: CA1ED1:CA1ED0: Capture1 Mode Select bits 00 = Capture on every falling edge 01 = Capture on every rising edge 10 = Capture on every 4th rising edge 11 = Capture on every 16th rising edge bit 3: T16: Timer1:Timer2 Mode Select bit 1 = Timer1 and Timer2 form a 16-bit timer 0 = Timer1 and Timer2 are two 8-bit timers bit 2: TMR3CS: Timer3 Clock Source Select bit 1 = TMR3 increments off the falling edge of the RB5/TCLK3 pin 0 = TMR3 increments off the internal clock bit 1: TMR2CS: Timer2 Clock Source Select bit 1 = TMR2 increments off the falling edge of the RB4/TCLK12 pin 0 = TMR2 increments off the internal clock bit 0: TMR1CS: Timer1 Clock Source Select bit 1 = TMR1 increments off the falling edge of the RB4/TCLK12 pin 0 = TMR1 increments off the internal clock							

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

12.1.3.1 PWM PERIODS

The period of the PWM1 output is determined by Timer1 and its period register (PR1). The period of the PWM2 output can be software configured to use either Timer1 or Timer2 as the time-base. When TM2PW2 bit (PW2DCL<5>) is clear, the time-base is determined by TMR1 and PR1. When TM2PW2 is set, the time-base is determined by Timer2 and PR2.

Running two different PWM outputs on two different timers allows different PWM periods. Running both PWMs from Timer1 allows the best use of resources by freeing Timer2 to operate as an 8-bit timer. Timer1 and Timer2 can not be used as a 16-bit timer if either PWM is being used.

The PWM periods can be calculated as follows:

$$\begin{aligned} \text{period of PWM1} &= [(PR1) + 1] \times 4T_{osc} \\ \text{period of PWM2} &= [(PR1) + 1] \times 4T_{osc} \quad \text{or} \\ &[(PR2) + 1] \times 4T_{osc} \end{aligned}$$

The duty cycle of PWMx is determined by the 10-bit value DCx<9:0>. The upper 8-bits are from register PWxDCH and the lower 2-bits are from PWxDCL<7:6> (PWxDCH:PWxDCL<7:6>). Table 12-3 shows the maximum PWM frequency (F_{PWM}) given the value in the period register.

The number of bits of resolution that the PWM can achieve depends on the operation frequency of the device as well as the PWM frequency (F_{PWM}).

Maximum PWM resolution (bits) for a given PWM frequency:

$$= \frac{\log \left(\frac{F_{osc}}{F_{PWM}} \right)}{\log (2)} \quad \text{bits}$$

The PWMx duty cycle is as follows:

$$\text{PWMx Duty Cycle} = (DCx) \times T_{osc}$$

where DCx represents the 10-bit value from PWxDCH:PWxDCL.

If DCx = 0, then the duty cycle is zero. If PRx = PWxDCH, then the PWM output will be low for one to four Q-clock (depending on the state of the PWxDCL<7:6> bits). For a Duty Cycle to be 100%, the PWxDCH value must be greater than the PRx value.

The duty cycle registers for both PWM outputs are double buffered. When the user writes to these registers, they are stored in master latches. When TMR1 (or TMR2) overflows and a new PWM period begins, the master latch values are transferred to the slave latches and the PWMx pin is forced high.

Note: For PW1DCH, PW1DCL, PW2DCH and PW2DCL registers, a write operation writes to the "master latches" while a read operation reads the "slave latches". As a result, the user may not read back what was just written to the duty cycle registers.

The user should also avoid any "read-modify-write" operations on the duty cycle registers, such as: ADDWF PW1DCH. This may cause duty cycle outputs that are unpredictable.

TABLE 12-3: PWM FREQUENCY vs. RESOLUTION AT 25 MHz

PWM Frequency	Frequency (kHz)				
	24.4	48.8	65.104	97.66	390.6
PRx Value	0xFF	0x7F	0x5F	0x3F	0x0F
High Resolution	10-bit	9-bit	8.5-bit	8-bit	6-bit
Standard Resolution	8-bit	7-bit	6.5-bit	6-bit	4-bit

12.1.3.2 PWM INTERRUPTS

The PWM module makes use of TMR1 or TMR2 interrupts. A timer interrupt is generated when TMR1 or TMR2 equals its period register and is cleared to zero. This interrupt also marks the beginning of a PWM cycle. The user can write new duty cycle values before the timer roll-over. The TMR1 interrupt is latched into the TMR1IF bit and the TMR2 interrupt is latched into the TMR2IF bit. These flags must be cleared in software.

12.1.3.3 EXTERNAL CLOCK SOURCE

The PWMs will operate regardless of the clock source of the timer. The use of an external clock has ramifications that must be understood. Because the external TCLK12 input is synchronized internally (sampled once per instruction cycle), the time TCLK12 changes to the time the timer increments will vary by as much as T_{cy} (one instruction cycle). This will cause jitter in the duty cycle as well as the period of the PWM output.

This jitter will be ±T_{cy}, unless the external clock is synchronized with the processor clock. Use of one of the PWM outputs as the clock source to the TCLKx input, will supply a synchronized clock.

In general, when using an external clock source for PWM, its frequency should be much less than the device frequency (F_{osc}).

13.3 USART Synchronous Master Mode

In Master Synchronous mode, the data is transmitted in a half-duplex manner; i.e. transmission and reception do not occur at the same time: when transmitting data, the reception is inhibited and vice versa. The synchronous mode is entered by setting the SYNC (TXSTA<4>) bit. In addition, the SPEN (RCSTA<7>) bit is set in order to configure the RA5 and RA4 I/O ports to CK (clock) and DT (data) lines respectively. The Master mode indicates that the processor transmits the master clock on the CK line. The Master mode is entered by setting the CSRC (TXSTA<7>) bit.

13.3.1 USART SYNCHRONOUS MASTER TRANSMISSION

The USART transmitter block diagram is shown in Figure 13-3. The heart of the transmitter is the transmit (serial) shift register (TSR). The shift register obtains its data from the read/write transmit buffer TXREG. TXREG is loaded with data in software. The TSR is not loaded until the last bit has been transmitted from the previous load. As soon as the last bit is transmitted, the TSR is loaded with new data from TXREG (if available). Once TXREG transfers the data to the TSR (occurs in one Tcy at the end of the current BRG cycle), TXREG is empty and the TXIF (PIR<1>) bit is set. This interrupt can be enabled/disabled by setting/clearing the TXIE bit (PIE<1>). TXIF will be set regardless of the state of bit TXIE and cannot be cleared in software. It will reset only when new data is loaded into TXREG. While TXIF indicates the status of TXREG, TRMT (TXSTA<1>) shows the status of the TSR. TRMT is a read only bit which is set when the TSR is empty. No interrupt logic is tied to this bit, so the user has to poll this bit in order to determine if the TSR is empty. The TSR is not mapped in data memory, so it is not available to the user.

Transmission is enabled by setting the TXEN (TXSTA<5>) bit. The actual transmission will not occur until TXREG has been loaded with data. The first data bit will be shifted out on the next available rising edge of the clock on the RA5/TX/CK pin. Data out is stable around the falling edge of the synchronous clock (Figure 13-10). The transmission can also be started by first loading TXREG and then setting TXEN. This is advantageous when slow baud rates are selected, since BRG is kept in RESET when the TXEN, CREN, and SREN bits are clear. Setting the TXEN bit will start the BRG, creating a shift clock immediately. Normally when transmission is first started, the TSR is empty, so a transfer to TXREG will result in an immediate transfer to the TSR, resulting in an empty TXREG. Back-to-back transfers are possible.

Clearing TXEN during a transmission will cause the transmission to be aborted and will reset the transmitter. The RA4/RX/DT and RA5/TX/CK pins will revert to hi-impedance. If either CREN or SREN are set during a transmission, the transmission is aborted and the

RA4/RX/DT pin reverts to a hi-impedance state (for a reception). The RA5/TX/CK pin will remain an output if the CSRC bit is set (internal clock). The transmitter logic is not reset, although it is disconnected from the pins. In order to reset the transmitter, the user has to clear the TXEN bit. If the SREN bit is set (to interrupt an ongoing transmission and receive a single word), then after the single word is received, SREN will be cleared and the serial port will revert back to transmitting, since the TXEN bit is still set. The DT line will immediately switch from hi-impedance receive mode to transmit and start driving. To avoid this, TXEN should be cleared.

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

Steps to follow when setting up a Synchronous Master Transmission:

1. Initialize the SPBRG register for the appropriate baud rate (see Baud Rate Generator Section for details).
2. Enable the synchronous master serial port by setting the SYNC, SPEN, and CSRC bits.
3. Ensure that the CREN and SREN bits are clear (these bits override transmission when set).
4. If interrupts are desired, then set the TXIE bit (the GLINTD bit must be clear and the PEIE bit must be set).
5. If 9-bit transmission is desired, then set the TX9 bit.
6. Start transmission by loading data to the TXREG register.
7. If 9-bit transmission is selected, the ninth bit should be loaded in TX9D.
8. Enable the transmission by setting TXEN.

Writing the transmit data to the TXREG, then enabling the transmit (setting TXEN) allows transmission to start sooner than doing these two events in the reverse order.

Note: To terminate a transmission, either clear the SPEN bit, or the TXEN bit. This will reset the transmit logic, so that it will be in the proper state when transmit is re-enabled.

TABLE 15-2: PIC17CXX INSTRUCTION SET

Mnemonic, Operands	Description	Cycles	16-bit Opcode		Status Affected	Notes
			MSb	LSb		
BYTE-ORIENTED FILE REGISTER OPERATIONS						
ADDWF	f,d	ADD WREG to f	1	0000 111d ffff ffff	OV,C,DC,Z	
ADDWFC	f,d	ADD WREG and Carry bit to f	1	0001 000d ffff ffff	OV,C,DC,Z	
ANDWF	f,d	AND WREG with f	1	0000 101d ffff ffff	Z	
CLRF	f,s	Clear f, or Clear f and Clear WREG	1	0010 100s ffff ffff	None	3
COMF	f,d	Complement f	1	0001 001d ffff ffff	Z	
CPFSEQ	f	Compare f with WREG, skip if f = WREG	1 (2)	0011 0001 ffff ffff	None	6,8
CPFSGT	f	Compare f with WREG, skip if f > WREG	1 (2)	0011 0010 ffff ffff	None	2,6,8
CPFSLT	f	Compare f with WREG, skip if f < WREG	1 (2)	0011 0000 ffff ffff	None	2,6,8
DAW	f,s	Decimal Adjust WREG Register	1	0010 111s ffff ffff	C	3
DECF	f,d	Decrement f	1	0000 011d ffff ffff	OV,C,DC,Z	
DECFSZ	f,d	Decrement f, skip if 0	1 (2)	0001 011d ffff ffff	None	6,8
DCFSNZ	f,d	Decrement f, skip if not 0	1 (2)	0010 011d ffff ffff	None	6,8
INCF	f,d	Increment f	1	0001 010d ffff ffff	OV,C,DC,Z	
INCFSZ	f,d	Increment f, skip if 0	1 (2)	0001 111d ffff ffff	None	6,8
INFSNZ	f,d	Increment f, skip if not 0	1 (2)	0010 010d ffff ffff	None	6,8
IORWF	f,d	Inclusive OR WREG with f	1	0000 100d ffff ffff	Z	
MOVFP	f,p	Move f to p	1	011p pppp ffff ffff	None	
MOVPF	p,f	Move p to f	1	010p pppp ffff ffff	Z	
MOVWF	f	Move WREG to f	1	0000 0001 ffff ffff	None	
MULWF	f	Multiply WREG with f	1	0011 0100 ffff ffff	None	9
NEGW	f,s	Negate WREG	1	0010 110s ffff ffff	OV,C,DC,Z	1,3
NOP	—	No Operation	1	0000 0000 0000 0000	None	
RLCF	f,d	Rotate left f through Carry	1	0001 101d ffff ffff	C	
RLNCF	f,d	Rotate left f (no carry)	1	0010 001d ffff ffff	None	
RRCF	f,d	Rotate right f through Carry	1	0001 100d ffff ffff	C	
RRNCF	f,d	Rotate right f (no carry)	1	0010 000d ffff ffff	None	
SETF	f,s	Set f	1	0010 101s ffff ffff	None	3
SUBWF	f,d	Subtract WREG from f	1	0000 010d ffff ffff	OV,C,DC,Z	1
SUBWFB	f,d	Subtract WREG from f with Borrow	1	0000 001d ffff ffff	OV,C,DC,Z	1
SWAPF	f,d	Swap f	1	0001 110d ffff ffff	None	
TABLRD	t,i,f	Table Read	2 (3)	1010 10ti ffff ffff	None	7

Legend: Refer to Table 15-1 for opcode field descriptions.

Note 1: 2's Complement method.

2: Unsigned arithmetic.

3: If s = '1', only the file is affected: If s = '0', both the WREG register and the file are affected; If only the Working register (WREG) is required to be affected, then f = WREG must be specified.

4: During an LCALL, the contents of PCLATH are loaded into the MSB of the PC and kkkk kkkk is loaded into the LSB of the PC (PCL)

5: Multiple cycle instruction for EPROM programming when table pointer selects internal EPROM. The instruction is terminated by an interrupt event. When writing to external program memory, it is a two-cycle instruction.

6: Two-cycle instruction when condition is true, else single cycle instruction.

7: Two-cycle instruction except for TABLRD to PCL (program counter low byte) in which case it takes 3 cycles.

8: A "skip" means that instruction fetched during execution of current instruction is not executed, instead an NOP is executed.

9: These instructions are not available on the PIC17C42.

PIC17C4X

Applicable Devices 42 R42 42A 43 R43 44

Standard Operating Conditions (unless otherwise stated)							
DC CHARACTERISTICS							
Operating temperature -40°C ≤ T _A ≤ +40°C							
Operating voltage V _{DD} range as described in Section 17.1							
Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
Internal Program Memory Programming Specs (Note 4)							
D110	V _{PP}	Voltage on $\overline{\text{MCLR}}/\text{VPP}$ pin	12.75	–	13.25	V	Note 5
D111	V _{DDP}	Supply voltage during programming	4.75	5.0	5.25	V	
D112	I _{PP}	Current into $\overline{\text{MCLR}}/\text{VPP}$ pin	–	25 ‡	50 ‡	mA	
D113	I _{DDP}	Supply current during programming	–	–	30 ‡	mA	
D114	T _{PROG}	Programming pulse width	10	100	1000	μs	Terminated via internal/external interrupt or a reset

* 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 for design guidance only and are not tested, nor characterized.

Note 1: In RC oscillator configuration, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended that the PIC17CXX devices be driven with external clock in RC mode.

2: The leakage current on the $\overline{\text{MCLR}}$ pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

3: Negative current is defined as coming out of the pin.

4: These specifications are for the programming of the on-chip program memory EPROM through the use of the table write instructions. The complete programming specifications can be found in: PIC17CXX Programming Specifications (Literature number DS30139).

5: The $\overline{\text{MCLR}}/\text{VPP}$ pin may be kept in this range at times other than programming, but is not recommended.

6: For TTL buffers, the better of the two specifications may be used.

Note: When using the Table Write for internal programming, the device temperature must be less than 40°C.

17.4 Timing Diagrams and Specifications

FIGURE 17-2: EXTERNAL CLOCK TIMING

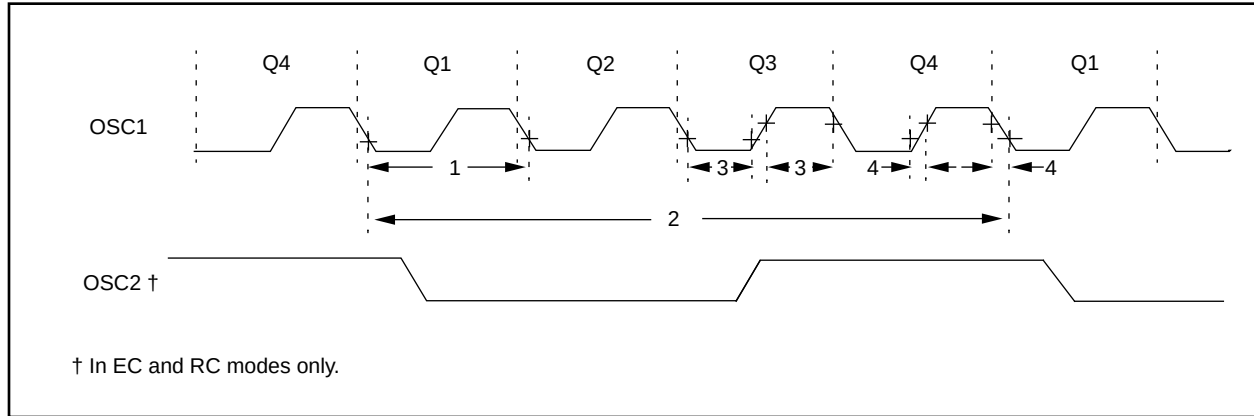


TABLE 17-2: EXTERNAL CLOCK TIMING REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
	Fosc	External CLKIN Frequency (Note 1)	DC	—	16	MHz	EC osc mode - PIC17C42-16
			DC	—	25	MHz	- PIC17C42-25
		Oscillator Frequency (Note 1)	DC	—	4	MHz	RC osc mode
			1	—	16	MHz	XT osc mode - PIC17C42-16
1	Tosc	External CLKIN Period (Note 1)	1	—	25	MHz	- PIC17C42-25
			DC	—	2	MHz	LF osc mode
		Oscillator Period (Note 1)	250	—	—	ns	RC osc mode
			62.5	—	1,000	ns	XT osc mode - PIC17C42-16
2	Tcy	Instruction Cycle Time (Note 1)	40	—	—	ns	- PIC17C42-25
			—	—	—	ns	LF osc mode
			—	—	—	ns	
			—	—	—	ns	
3	TosL, TosH	Clock in (OSC1) High or Low Time	10 ‡	—	—	ns	EC oscillator
4	TosR, TosF	Clock in (OSC1) Rise or Fall Time	—	—	5 ‡	ns	EC oscillator

† 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 for design guidance only and are not tested, nor characterized.

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 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 pin. When an external clock input is used, the "Max." cycle time limit is "DC" (no clock) for all devices.

PIC17C4X

Applicable Devices 42 R42 42A 43 R43 44

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

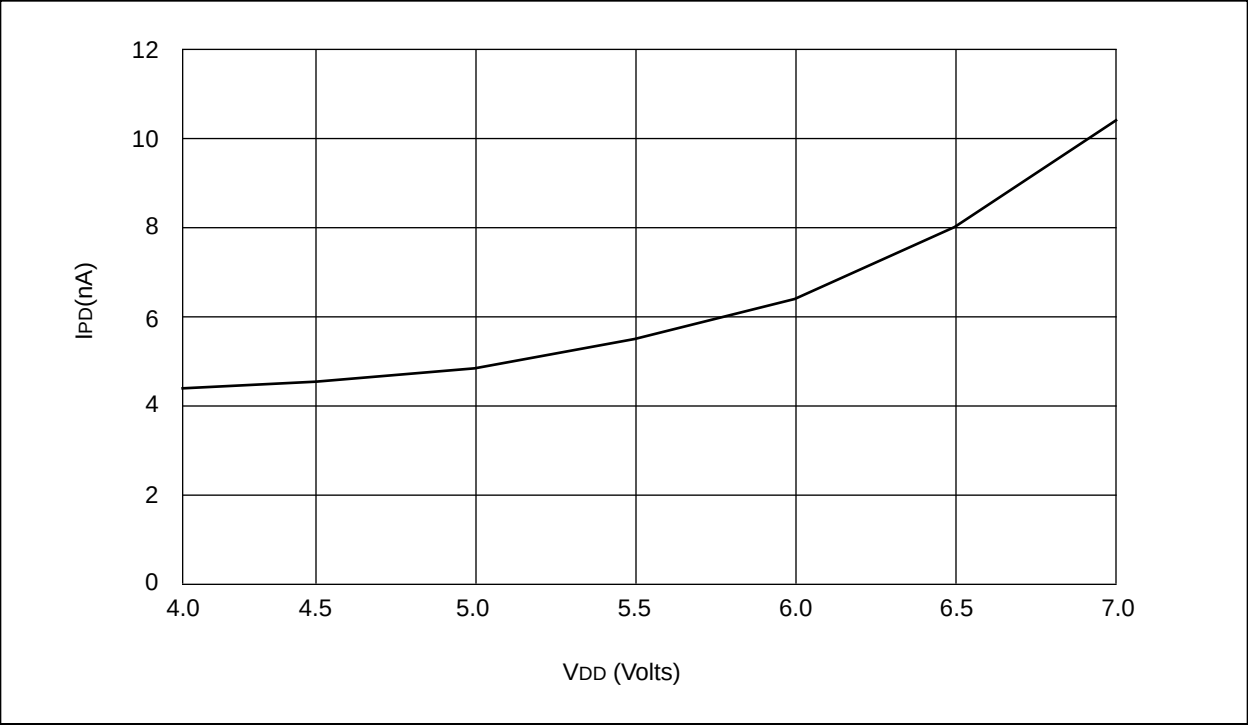
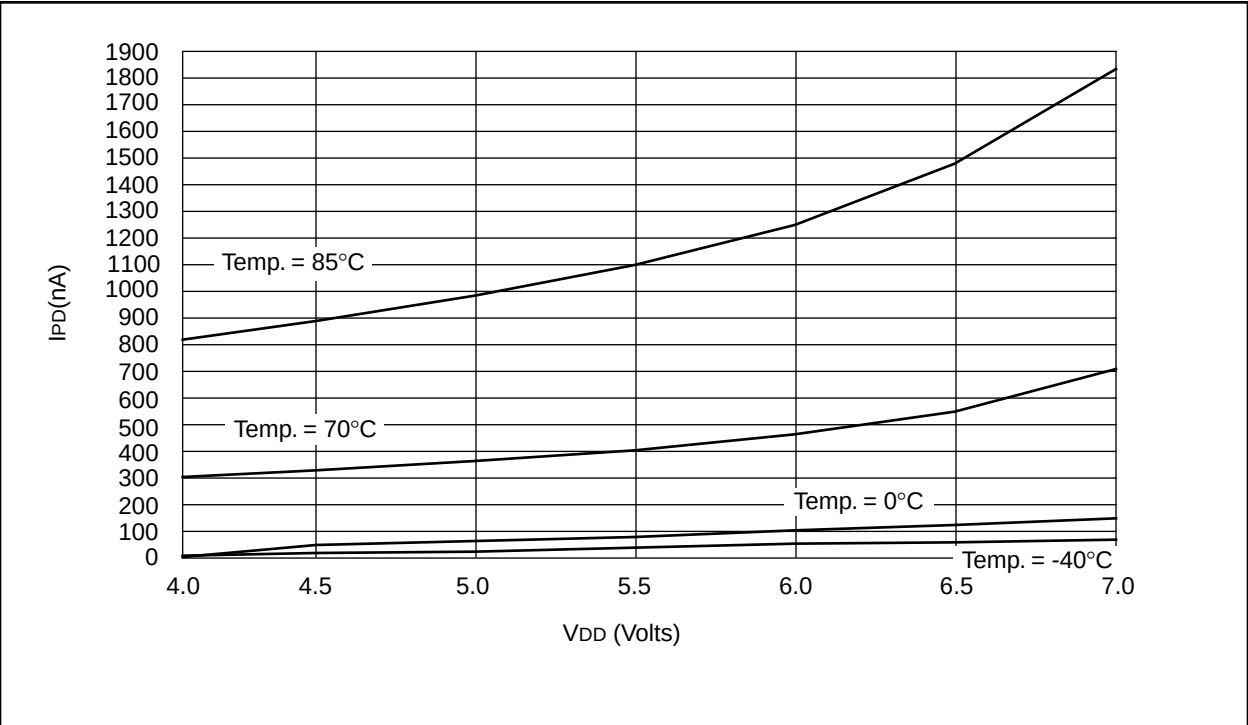


FIGURE 18-10: MAXIMUM I_{PD} vs. V_{DD} WATCHDOG DISABLED



PIC17C4X

Applicable Devices 42 R42 42A 43 R43 44

FIGURE 18-17: IOL vs. VOL, VDD = 5V

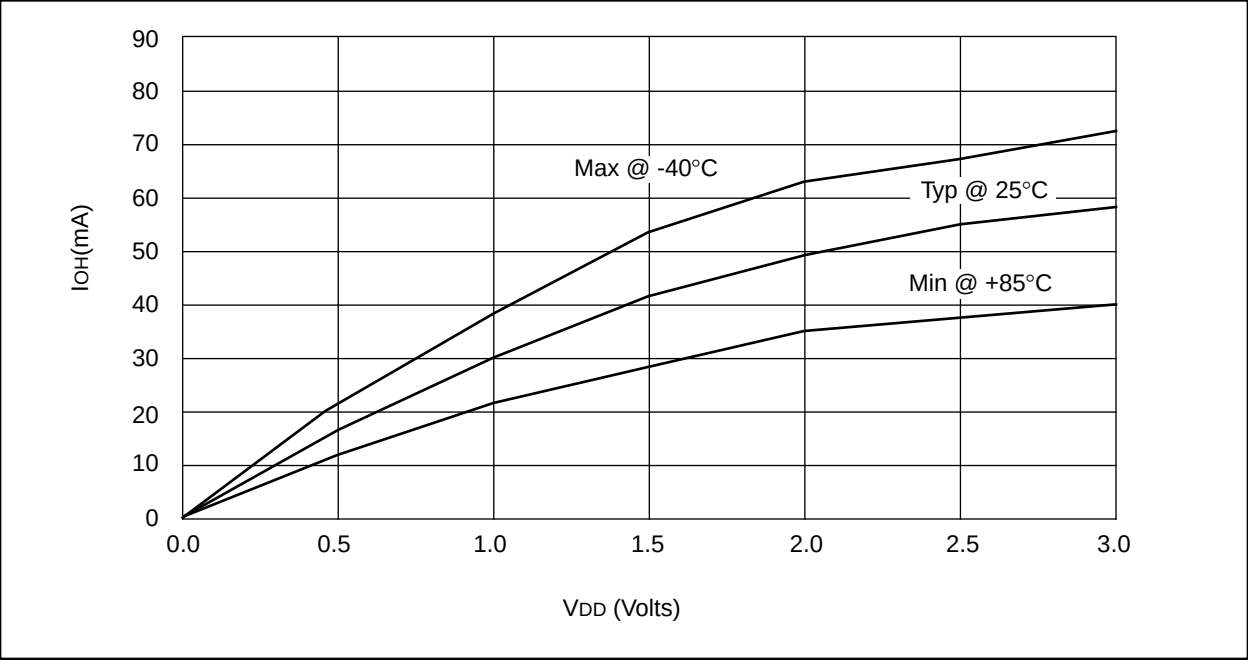
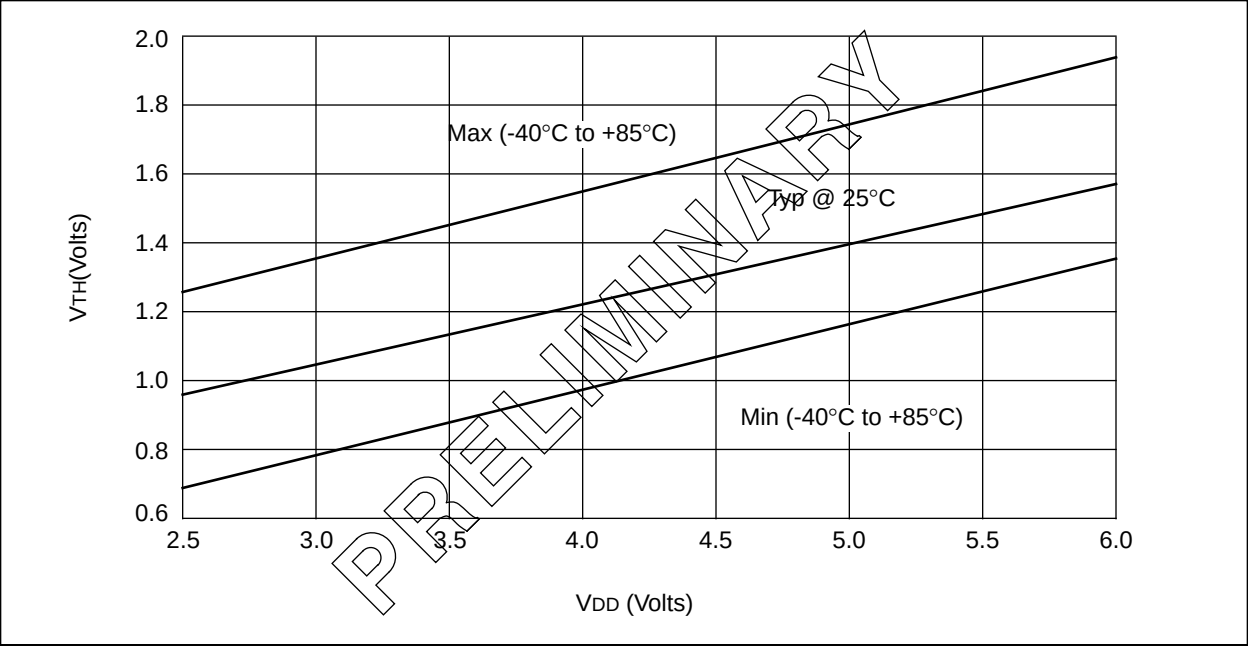


FIGURE 18-18: VTH (INPUT THRESHOLD VOLTAGE) OF I/O PINS (TTL) vs. VDD



PIC17C4X

NOTES:

20.0 PIC17CR42/42A/43/R43/44 DC AND AC CHARACTERISTICS

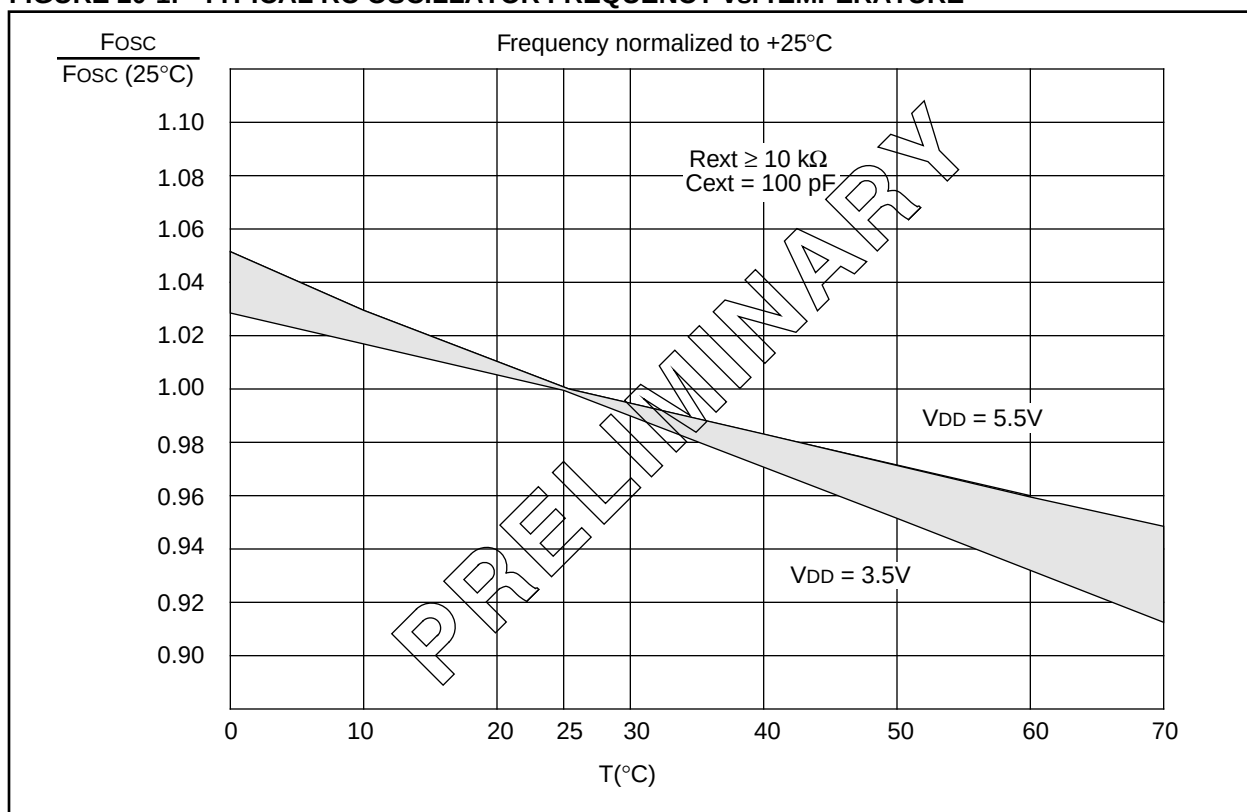
The graphs and tables provided in this section are for design guidance and are not tested nor guaranteed. In some graphs or tables the data presented is outside specified operating range (e.g. outside specified V_{DD} range). This is for information only and devices are ensured to operate properly only within the specified range.

The data presented in this section is a statistical summary of data collected on units from different lots over a period of time. "Typical" represents the mean of the distribution while "max" or "min" represents $(\text{mean} + 3\sigma)$ and $(\text{mean} - 3\sigma)$ respectively where σ is standard deviation.

TABLE 20-1: PIN CAPACITANCE PER PACKAGE TYPE

Pin Name	Typical Capacitance (pF)			
	40-pin DIP	44-pin PLCC	44-pin MQFP	44-pin TQFP
All pins, except $\overline{\text{MCLR}}$, V_{DD} , and V_{SS}	10	10	10	10
$\overline{\text{MCLR}}$ pin	20	20	20	20

FIGURE 20-1: TYPICAL RC OSCILLATOR FREQUENCY vs. TEMPERATURE



E.5 PIC16C7X Family of Devices

	Clock		Memory		Peripherals					Features		
	Maximum Frequency of Operation (MHz)	EPROM Program Memory (Kx4 words)	Data Memory (bytes)	Timer Modules(s)	Capture/Compare/PWM Modules(s)	Serial Ports (SPI/I ² C, USART)	A/D Converter (8-bit) Channels	I/O Pins	Voltage Range (Volts)	In-Circuit Serial Programming	Brown-out Reset	Packages
PIC16C710	20	512	36	TMR0	—	—	4	4	13	3.0-6.0	Yes	18-pin DIP, SOIC; 20-pin SSOP
PIC16C71	20	1K	36	TMR0	—	—	4	4	13	3.0-6.0	Yes	18-pin DIP, SOIC
PIC16C711	20	1K	68	TMR0	—	—	4	4	13	3.0-6.0	Yes	18-pin DIP, SOIC; 20-pin SSOP
PIC16C72	20	2K	128	TMR0, TMR1, TMR2	1 SPI/I ² C	—	5	8	22	2.5-6.0	Yes	28-pin SDIP, SOIC, SSOP
PIC16C73	20	4K	192	TMR0, TMR1, TMR2	2 SPI/I ² C, USART	—	5	11	22	3.0-6.0	Yes	28-pin SDIP, SOIC
PIC16C73A ⁽¹⁾	20	4K	192	TMR0, TMR1, TMR2	2 SPI/I ² C, USART	—	5	11	22	2.5-6.0	Yes	28-pin SDIP, SOIC
PIC16C74	20	4K	192	TMR0, TMR1, TMR2	2 SPI/I ² C, USART	Yes	8	12	33	3.0-6.0	Yes	40-pin DIP; 44-pin PLCC, MQFP
PIC16C74A ⁽¹⁾	20	4K	192	TMR0, TMR1, TMR2	2 SPI/I ² C, USART	Yes	8	12	33	2.5-6.0	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 PIC16C7X 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.

ON-LINE SUPPORT

Microchip provides two methods of on-line support. These are the Microchip BBS and the Microchip World Wide Web (WWW) site.

Use Microchip's Bulletin Board Service (BBS) to get current information and help about Microchip products. Microchip provides the BBS communication channel for you to use in extending your technical staff with micro-controller and memory experts.

To provide you with the most responsive service possible, the Microchip systems team monitors the BBS, posts the latest component data and software tool updates, provides technical help and embedded systems insights, and discusses how Microchip products provide project solutions.

The web site, like the BBS, is used by Microchip as a means to make files and information easily available to customers. To view the site, the user must have access to the Internet and a web browser, such as Netscape or Microsoft Explorer. Files are also available for FTP download from our FTP site.

Connecting to the Microchip InternetWeb Site

The Microchip web site is available by using your favorite Internet browser to attach to:

www.microchip.com

The file transfer site is available by using an FTP service to connect to:

[ftp.mchip.com/biz/mchip](ftp://mchip.com/biz/mchip)

The web site and file transfer site provide a variety of services. Users may download files for the latest Development Tools, Data Sheets, Application Notes, User's Guides, Articles and Sample Programs. A variety of Microchip specific business information is also available, including listings of Microchip sales offices, distributors and factory representatives. Other data available for consideration is:

- Latest Microchip Press Releases
- Technical Support Section with Frequently Asked Questions
- Design Tips
- Device Errata
- Job Postings
- Microchip Consultant Program Member Listing
- Links to other useful web sites related to Microchip Products

Connecting to the Microchip BBS

Connect worldwide to the Microchip BBS using either the Internet or the CompuServe® communications network.

Internet:

You can telnet or ftp to the Microchip BBS at the address:

[mchipbbs.microchip.com](telnet://mchipbbs.microchip.com)

CompuServe Communications Network:

When using the BBS via the Compuserve Network, in most cases, a local call is your only expense. The Microchip BBS connection does not use CompuServe membership services, therefore you do not need CompuServe membership to join Microchip's BBS. There is no charge for connecting to the Microchip BBS.

The procedure to connect will vary slightly from country to country. Please check with your local CompuServe agent for details if you have a problem. CompuServe service allow multiple users various baud rates depending on the local point of access.

The following connect procedure applies in most locations.

1. Set your modem to 8-bit, No parity, and One stop (8N1). This is not the normal CompuServe setting which is 7E1.
2. Dial your local CompuServe access number.
3. Depress the <Enter> key and a garbage string will appear because CompuServe is expecting a 7E1 setting.
4. Type +, depress the <Enter> key and "Host Name:" will appear.
5. Type MCHIPBBS, depress the <Enter> key and you will be connected to the Microchip BBS.

In the United States, to find the CompuServe phone number closest to you, set your modem to 7E1 and dial (800) 848-4480 for 300-2400 baud or (800) 331-7166 for 9600-14400 baud connection. After the system responds with "Host Name:", type NETWORK, depress the <Enter> key and follow CompuServe's directions.

For voice information (or calling from overseas), you may call (614) 723-1550 for your local CompuServe number.

Microchip regularly uses the Microchip BBS to distribute technical information, application notes, source code, errata sheets, bug reports, and interim patches for Microchip systems software products. For each SIG, a moderator monitors, scans, and approves or disapproves files submitted to the SIG. No executable files are accepted from the user community in general to limit the spread of computer viruses.

Systems Information and Upgrade Hot Line

The Systems Information and Upgrade Line provides system users a listing of the latest versions of all of Microchip's development systems software products. Plus, this line provides information on how customers can receive any currently available upgrade kits. The Hot Line Numbers are:

1-800-755-2345 for U.S. and most of Canada, and
1-602-786-7302 for the rest of the world.

960513

Trademarks: The Microchip name, logo, PIC, PICSTART, PICMASTER and PRO MATE are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries. FlexROM, MPLAB and fuzzyLAB, are trademarks and SQTP is a service mark of Microchip in the U.S.A.

fuzzyTECH is a registered trademark of Inform Software Corporation. IBM, IBM PC-AT are registered trademarks of International Business Machines Corp. Pentium is a trademark of Intel Corporation. Windows is a trademark and MS-DOS, Microsoft Windows are registered trademarks of Microsoft Corporation. CompuServe is a registered trademark of CompuServe Incorporated.

All other trademarks mentioned herein are the property of their respective companies.