



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	33MHz
Connectivity	UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	33
Program Memory Size	4KB (2K x 16)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	232 x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 6V
Data Converters	-
Oscillator Type	External
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	44-TQFP
Supplier Device Package	44-TQFP (10x10)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/microchip-technology/pic17c42at-33-pt">https://www.e-xfl.com/product-detail/microchip-technology/pic17c42at-33-pt</a>

Table of Contents

1.0 Overview ..... 5

2.0 PIC17C4X Device Varieties ..... 7

3.0 Architectural Overview ..... 9

4.0 Reset ..... 15

5.0 Interrupts ..... 21

6.0 Memory Organization ..... 29

7.0 Table Reads and Table Writes ..... 43

8.0 Hardware Multiplier ..... 49

9.0 I/O Ports ..... 53

10.0 Overview of Timer Resources ..... 65

11.0 Timer0 ..... 67

12.0 Timer1, Timer2, Timer3, PWMs and Captures ..... 71

13.0 Universal Synchronous Asynchronous Receiver Transmitter (USART) Module ..... 83

14.0 Special Features of the CPU ..... 99

15.0 Instruction Set Summary ..... 107

16.0 Development Support ..... 143

17.0 PIC17C42 Electrical Characteristics ..... 147

18.0 PIC17C42 DC and AC Characteristics ..... 163

19.0 PIC17CR42/42A/43/R43/44 Electrical Characteristics ..... 175

20.0 PIC17CR42/42A/43/R43/44 DC and AC Characteristics ..... 193

21.0 Packaging Information ..... 205

Appendix A: Modifications ..... 211

Appendix B: Compatibility ..... 211

Appendix C: What's New ..... 212

Appendix D: What's Changed ..... 212

Appendix E: PIC16/17 Microcontrollers ..... 213

Appendix F: Errata for PIC17C42 Silicon ..... 223

Index ..... 226

PIC17C4X Product Identification System ..... 237

For register and module descriptions in this data sheet, device legends show which devices apply to those sections. For example, the legend below shows that some features of only the PIC17C43, PIC17CR43, PIC17C44 are described in this section.

Applicable Devices				
42	R42	42A	43	R43 44

To Our Valued Customers

We constantly strive to improve the quality of all our products and documentation. We have spent an exceptional amount of time to ensure that these documents are correct. However, we realize that we may have missed a few things. If you find any information that is missing or appears in error from the previous version of the PIC17C4X Data Sheet (Literature Number DS30412B), please use the reader response form in the back of this data sheet to inform us. We appreciate your assistance in making this a better document.

To assist you in the use of this document, Appendix C contains a list of new information in this data sheet, while Appendix D contains information that has changed

# PIC17C4X

---

NOTES:

## 3.0 ARCHITECTURAL OVERVIEW

The high performance of the PIC17C4X can be attributed to a number of architectural features commonly found in RISC microprocessors. To begin with, the PIC17C4X uses a modified Harvard architecture. This architecture has the program and data accessed from separate memories. So the device has a program memory bus and a data memory bus. This improves bandwidth over traditional von Neumann architecture, where program and data are fetched from the same memory (accesses over the same bus). Separating program and data memory further allows instructions to be sized differently than the 8-bit wide data word. PIC17C4X opcodes are 16-bits wide, enabling single word instructions. The full 16-bit wide program memory bus fetches a 16-bit instruction in a single cycle. A two-stage pipeline overlaps fetch and execution of instructions. Consequently, all instructions execute in a single cycle (121 ns @ 33 MHz), except for program branches and two special instructions that transfer data between program and data memory.

The PIC17C4X can address up to 64K x 16 of program memory space.

The **PIC17C42** and **PIC17C42A** integrate 2K x 16 of EPROM program memory on-chip, while the **PIC17CR42** has 2K x 16 of ROM program memory on-chip.

The **PIC17C43** integrates 4K x 16 of EPROM program memory, while the **PIC17CR43** has 4K x 16 of ROM program memory.

The **PIC17C44** integrates 8K x 16 EPROM program memory.

Program execution can be internal only (microcontroller or protected microcontroller mode), external only (microprocessor mode) or both (extended microcontroller mode). Extended microcontroller mode does not allow code protection.

The PIC17CXX can directly or indirectly address its register files or data memory. All special function registers, including the Program Counter (PC) and Working Register (WREG), are mapped in the data memory. The PIC17CXX 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' make programming with the PIC17CXX simple yet efficient. In addition, the learning curve is reduced significantly.

One of the PIC17CXX family architectural enhancements from the PIC16CXX family allows two file registers to be used in some two operand instructions. This allows data to be moved directly between two registers without going through the WREG register. This increases performance and decreases program memory usage.

The PIC17CXX devices contain an 8-bit ALU and working register. 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.

The WREG register is an 8-bit working register used for ALU operations.

All PIC17C4X devices (except the PIC17C42) have an 8 x 8 hardware multiplier. This multiplier generates a 16-bit result in a single cycle.

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

Although the ALU does not perform signed arithmetic, the Overflow bit (OV) can be used to implement signed math. Signed arithmetic is comprised of a magnitude and a sign bit. The overflow bit indicates if the magnitude overflows and causes the sign bit to change state. Signed math can have greater than 7-bit values (magnitude), if more than one byte is used. The use of the overflow bit only operates on bit6 (MSb of magnitude) and bit7 (sign bit) of the value in the ALU. That is, the overflow bit is not useful if trying to implement signed math where the magnitude, for example, is 11-bits. If the signed math values are greater than 7-bits (15-, 24- or 31-bit), the algorithm must ensure that the low order bytes ignore the overflow status bit.

Care should be taken when adding and subtracting signed numbers to ensure that the correct operation is executed. Example 3-1 shows an item that must be taken into account when doing signed arithmetic on an ALU which operates as an unsigned machine.

### EXAMPLE 3-1: SIGNED MATH

Hex Value	Signed Value Math	Unsigned Value Math
FFh	-127	255
+ 01h	+ 1	+ 1
= ?	= -126 (FEh)	= 0 (00h); Carry bit = 1

Signed math requires the result in REG to be FEh (-126). This would be accomplished by subtracting one as opposed to adding one.

Simplified block diagrams are shown in Figure 3-1 and Figure 3-2. The descriptions of the device pins are listed in Table 3-1.

# PIC17C4X

**TABLE 3-1: PINOUT DESCRIPTIONS**

Name	DIP No.	PLCC No.	QFP No.	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	19	21	37	I	ST	Oscillator input in crystal/resonator or RC oscillator mode. External clock input in external clock mode.
OSC2/CLKOUT	20	22	38	O	—	Oscillator output. Connects to crystal or resonator in crystal oscillator mode. In RC oscillator or external clock modes OSC2 pin outputs CLKOUT which has one fourth the frequency of OSC1 and denotes the instruction cycle rate.
MCLR/VPP	32	35	7	I/P	ST	Master clear (reset) input/Programming Voltage (VPP) input. This is the active low reset input to the chip.
RA0/INT	26	28	44	I	ST	<p>PORTA is a bi-directional I/O Port except for RA0 and RA1 which are input only.</p> <p>RA0/INT can also be selected as an external interrupt input. Interrupt can be configured to be on positive or negative edge.</p> <p>RA1/T0CKI can also be selected as an external interrupt input, and the interrupt can be configured to be on positive or negative edge. RA1/T0CKI can also be selected to be the clock input to the Timer0 timer/counter.</p> <p>High voltage, high current, open drain input/output port pins.</p> <p>High voltage, high current, open drain input/output port pins.</p> <p>RA4/RX/DT can also be selected as the USART (SCI) Asynchronous Receive or USART (SCI) Synchronous Data.</p> <p>RA5/TX/CK can also be selected as the USART (SCI) Asynchronous Transmit or USART (SCI) Synchronous Clock.</p>
RA1/T0CKI	25	27	43	I	ST	
RA2	24	26	42	I/O	ST	
RA3	23	25	41	I/O	ST	
RA4/RX/DT	22	24	40	I/O	ST	
RA5/TX/CK	21	23	39	I/O	ST	
RB0/CAP1	11	13	29	I/O	ST	<p>PORTB is a bi-directional I/O Port with software configurable weak pull-ups.</p> <p>RB0/CAP1 can also be the CAP1 input pin.</p> <p>RB1/CAP2 can also be the CAP2 input pin.</p> <p>RB2/PWM1 can also be the PWM1 output pin.</p> <p>RB3/PWM2 can also be the PWM2 output pin.</p> <p>RB4/TCLK12 can also be the external clock input to Timer1 and Timer2.</p> <p>RB5/TCLK3 can also be the external clock input to Timer3.</p>
RB1/CAP2	12	14	30	I/O	ST	
RB2/PWM1	13	15	31	I/O	ST	
RB3/PWM2	14	16	32	I/O	ST	
RB4/TCLK12	15	17	33	I/O	ST	
RB5/TCLK3	16	18	34	I/O	ST	
RB6	17	19	35	I/O	ST	
RB7	18	20	36	I/O	ST	
RC0/AD0	2	3	19	I/O	TTL	<p>PORTC is a bi-directional I/O Port.</p> <p>This is also the lower half of the 16-bit wide system bus in microprocessor mode or extended microcontroller mode. In multiplexed system bus configuration, these pins are address output as well as data input or output.</p>
RC1/AD1	3	4	20	I/O	TTL	
RC2/AD2	4	5	21	I/O	TTL	
RC3/AD3	5	6	22	I/O	TTL	
RC4/AD4	6	7	23	I/O	TTL	
RC5/AD5	7	8	24	I/O	TTL	
RC6/AD6	8	9	25	I/O	TTL	
RC7/AD7	9	10	26	I/O	TTL	

Legend: I = Input only; O = Output only; I/O = Input/Output; P = Power; — = Not Used; TTL = TTL input; ST = Schmitt Trigger input.

## 4.0 RESET

The PIC17CXX differentiates between various kinds of reset:

- Power-on Reset (POR)
- $\overline{\text{MCLR}}$  reset during normal operation
- WDT Reset (normal operation)

Some registers are not affected in any reset condition; their status is unknown on POR and unchanged in any other reset. Most other registers are forced to a "reset state" on Power-on Reset (POR), on  $\overline{\text{MCLR}}$  or WDT Reset and on  $\overline{\text{MCLR}}$  reset during SLEEP. They are not affected by a WDT Reset during SLEEP, since this reset is viewed as the resumption of normal operation. The  $\overline{\text{TO}}$  and  $\overline{\text{PD}}$  bits are set or cleared differently in different reset situations as indicated in Table 4-3. These bits are used in software to determine the nature of reset. See Table 4-4 for a full description of reset states of all registers.

**Note:** While the device is in a reset state, the internal phase clock is held in the Q1 state. Any processor mode that allows external execution will force the RE0/ALE pin as a low output and the RE1/ $\overline{\text{OE}}$  and RE2/ $\overline{\text{WR}}$  pins as high outputs.

A simplified block diagram of the on-chip reset circuit is shown in Figure 4-1.

## 4.1 Power-on Reset (POR), Power-up Timer (PWRT), and Oscillator Start-up Timer (OST)

### 4.1.1 POWER-ON RESET (POR)

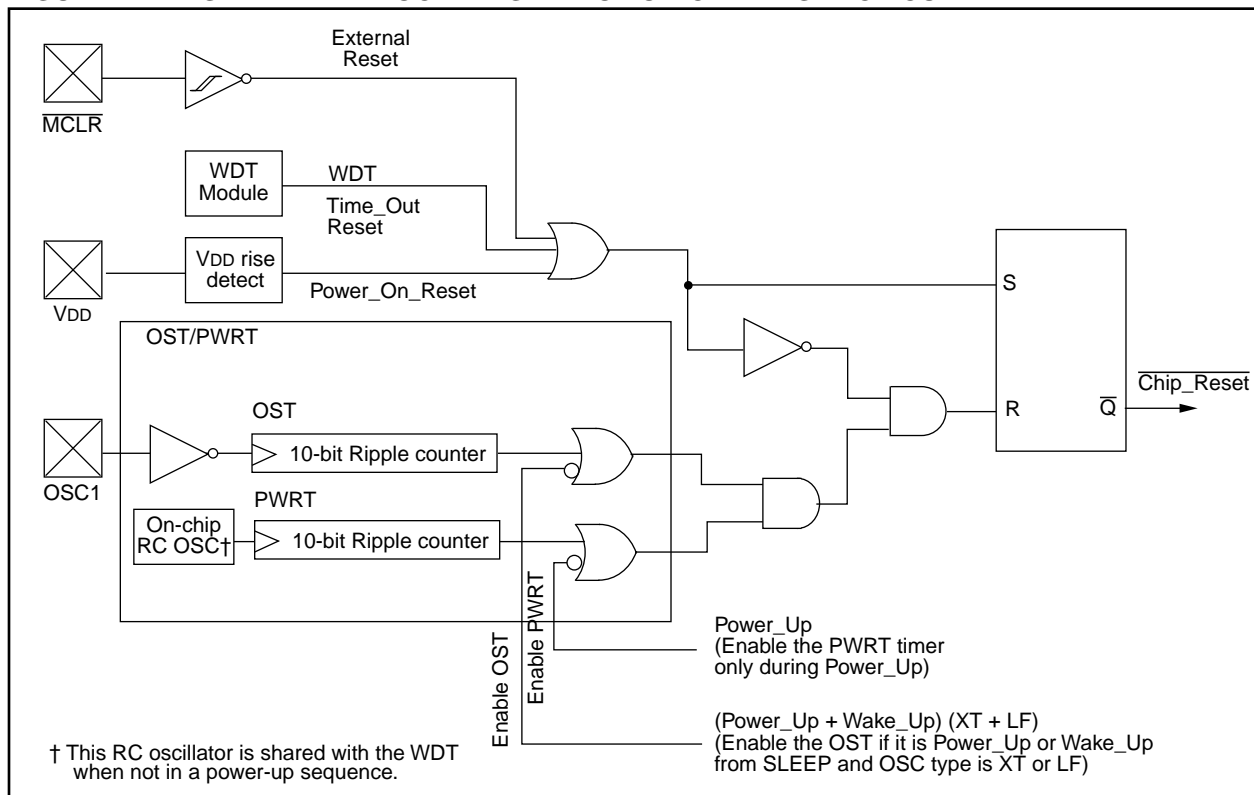
The Power-on Reset circuit holds the device in reset until  $V_{DD}$  is above the trip point (in the range of 1.4V - 2.3V). The PIC17C42 does not produce an internal reset when  $V_{DD}$  declines. All other devices will produce an internal reset for both rising and falling  $V_{DD}$ . To take advantage of the POR, just tie the  $\overline{\text{MCLR}}/\text{VPP}$  pin directly (or through a resistor) to  $V_{DD}$ . This will eliminate external RC components usually needed to create Power-on Reset. A minimum rise time for  $V_{DD}$  is required. See Electrical Specifications for details.

### 4.1.2 POWER-UP TIMER (PWRT)

The Power-up Timer provides a fixed 96 ms time-out (nominal) on power-up. This occurs from rising edge of the POR signal and after the first rising edge of  $\overline{\text{MCLR}}$  (detected high). The Power-up Timer operates on an internal RC oscillator. The chip is kept in RESET as long as the PWRT is active. In most cases the PWRT delay allows the  $V_{DD}$  to rise to an acceptable level.

The power-up time delay will vary from chip to chip and to  $V_{DD}$  and temperature. See DC parameters for details.

**FIGURE 4-1: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT**



**TABLE 6-3: SPECIAL FUNCTION REGISTERS (Cont'd)**

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 (3)
Bank 2											
10h	TMR1	Timer1								xxxx xxxx	uuuu uuuu
11h	TMR2	Timer2								xxxx xxxx	uuuu uuuu
12h	TMR3L	TMR3 register; low byte								xxxx xxxx	uuuu uuuu
13h	TMR3H	TMR3 register; high byte								xxxx xxxx	uuuu uuuu
14h	PR1	Timer1 period register								xxxx xxxx	uuuu uuuu
15h	PR2	Timer2 period register								xxxx xxxx	uuuu uuuu
16h	PR3L/CA1L	Timer3 period register, low byte/capture1 register; low byte								xxxx xxxx	uuuu uuuu
17h	PR3H/CA1H	Timer3 period register, high byte/capture1 register; high byte								xxxx xxxx	uuuu uuuu
Bank 3											
10h	PW1DCL	DC1	DC0	—	—	—	—	—	—	xx-- ----	uu-- ----
11h	PW2DCL	DC1	DC0	TM2PW2	—	—	—	—	—	xx0- ----	uu0- ----
12h	PW1DCH	DC9	DC8	DC7	DC6	DC5	DC4	DC3	DC2	xxxx xxxx	uuuu uuuu
13h	PW2DCH	DC9	DC8	DC7	DC6	DC5	DC4	DC3	DC2	xxxx xxxx	uuuu uuuu
14h	CA2L	Capture2 low byte								xxxx xxxx	uuuu uuuu
15h	CA2H	Capture2 high byte								xxxx xxxx	uuuu uuuu
16h	TCON1	CA2ED1	CA2ED0	CA1ED1	CA1ED0	T16	TMR3CS	TMR2CS	TMR1CS	0000 0000	0000 0000
17h	TCON2	CA2OVF	CA1OVF	PWM2ON	PWM1ON	CA1/PR3	TMR3ON	TMR2ON	TMR1ON	0000 0000	0000 0000
Unbanked											
18h <sup>(5)</sup>	PRODL	Low Byte of 16-bit Product (8 x 8 Hardware Multiply)								xxxx xxxx	uuuu uuuu
19h <sup>(5)</sup>	PRODH	High Byte of 16-bit Product (8 x 8 Hardware Multiply)								xxxx xxxx	uuuu uuuu

- Legend: x = unknown, u = unchanged, - = unimplemented read as '0', q - value depends on condition. Shaded cells are unimplemented, read as '0'.
- Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for PC<15:8> whose contents are updated from or transferred to the upper byte of the program counter.
- 2: The  $\overline{TO}$  and  $\overline{PD}$  status bits in CPUSTA are not affected by a MCLR reset.
- 3: Other (non power-up) resets include: external reset through MCLR and the Watchdog Timer Reset.
- 4: The following values are for both TBLPTRL and TBLPTRH:  
All PIC17C4X devices (Power-on Reset 0000 0000) and (All other resets 0000 0000)  
except the PIC17C42 (Power-on Reset xxxx xxxx) and (All other resets uuuu uuuu)
- 5: The PRODL and PRODH registers are not implemented on the PIC17C42.

## 6.2.2.1 ALU STATUS REGISTER (ALUSTA)

The ALUSTA register contains the status bits of the Arithmetic and Logic Unit and the mode control bits for the indirect addressing register.

As with all the other registers, the ALUSTA register can be the destination for any instruction. If the ALUSTA register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. These bits are set or cleared according to the device logic. Therefore, the result of an instruction with the ALUSTA register as destination may be different than intended.

For example, `CLRF ALUSTA` will clear the upper four bits and set the Z bit. This leaves the ALUSTA register as 0000u1uu (where u = unchanged).

It is recommended, therefore, that only `BCF`, `BSF`, `SWAPF` and `MOVWF` instructions be used to alter the ALUSTA register because these instructions do not affect any status bit. To see how other instructions affect the status bits, see the "Instruction Set Summary."

**Note 1:** The C and DC bits operate as a borrow out bit in subtraction. See the `SUBLW` and `SUBWF` instructions for examples.

**Note 2:** The overflow bit will be set if the 2's complement result exceeds +127 or is less than -128.

Arithmetic and Logic Unit (ALU) is capable of carrying out arithmetic or logical operations on two operands or a single operand. All single operand instructions operate either on the WREG register or a file register. For two operand instructions, one of the operands is the WREG register and the other one is either a file register or an 8-bit immediate constant.

**FIGURE 6-7: ALUSTA REGISTER (ADDRESS: 04h, UNBANKED)**

R/W - 1	R/W - 1	R/W - 1	R/W - 1	R/W - x	R/W - x	R/W - x	R/W - x
FS3	FS2	FS1	FS0	OV	Z	DC	C
bit7							bit0

R = Readable bit  
W = Writable bit  
-n = Value at POR reset  
(x = unknown)

bit 7-6: **FS3:FS2:** FSR1 Mode Select bits  
 00 = Post auto-decrement FSR1 value  
 01 = Post auto-increment FSR1 value  
 1x = FSR1 value does not change

bit 5-4: **FS1:FS0:** FSR0 Mode Select bits  
 00 = Post auto-decrement FSR0 value  
 01 = Post auto-increment FSR0 value  
 1x = FSR0 value does not change

bit 3: **OV:** Overflow bit  
 This bit is used for signed arithmetic (2's complement). It indicates an overflow of the 7-bit magnitude, which causes the sign bit (bit7) to change state.  
 1 = Overflow occurred for signed arithmetic, (in this arithmetic operation)  
 0 = No overflow occurred

bit 2: **Z:** Zero bit  
 1 = The result of an arithmetic or logic operation is zero  
 0 = The results of an arithmetic or logic operation is not zero

bit 1: **DC:** Digit carry/borrow bit  
 For `ADDWF` and `ADDLW` instructions.  
 1 = A carry-out from the 4th low order bit of the result occurred  
 0 = No carry-out from the 4th low order bit of the result  
 Note: For borrow the polarity is reversed.

bit 0: **C:** carry/borrow bit  
 For `ADDWF` and `ADDLW` instructions.  
 1 = A carry-out from the most significant bit of the result occurred  
 Note that a subtraction is executed by adding the two's complement of the second operand. For rotate (`RRCF`, `RLCF`) instructions, this bit is loaded with either the high or low order bit of the source register.  
 0 = No carry-out from the most significant bit of the result  
 Note: For borrow the polarity is reversed.



FIGURE 7-3: TLRD INSTRUCTION OPERATION

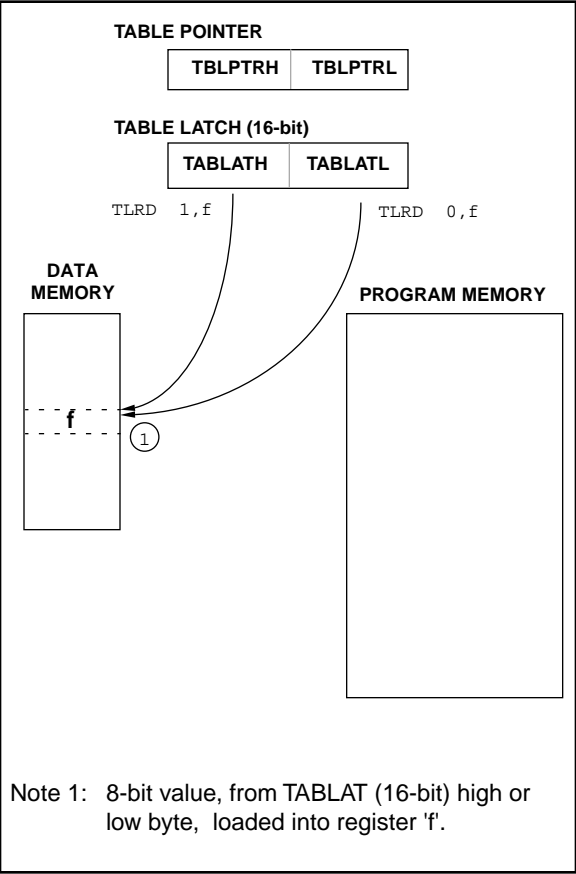
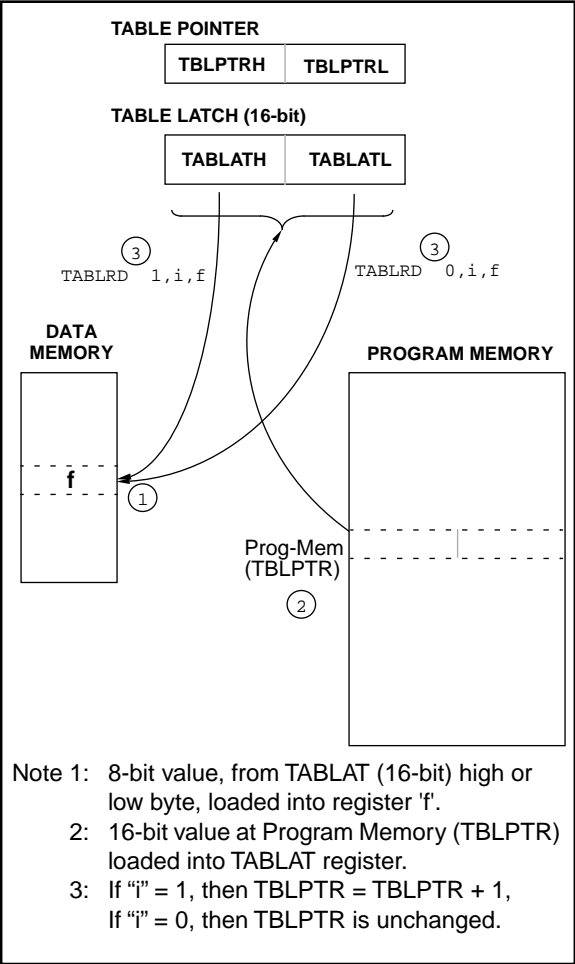


FIGURE 7-4: TABLRD INSTRUCTION OPERATION



Example 9-1 shows the instruction sequence to initialize PORTB. The Bank Select Register (BSR) must be selected to Bank 0 for the port to be initialized.

## EXAMPLE 9-1: INITIALIZING PORTB

```
MOVLB 0      ; Select Bank 0
CLRF  PORTB  ; Initialize PORTB by clearing
              ; output data latches
MOVLW 0xCF   ; Value used to initialize
              ; data direction
MOVWF DDRB   ; Set RB<3:0> as inputs
              ; RB<5:4> as outputs
              ; RB<7:6> as inputs
```

**TABLE 9-3: PORTB FUNCTIONS**

Name	Bit	Buffer Type	Function
RB0/CAP1	bit0	ST	Input/Output or the RB0/CAP1 input pin. Software programmable weak pull-up and interrupt on change features.
RB1/CAP2	bit1	ST	Input/Output or the RB1/CAP2 input pin. Software programmable weak pull-up and interrupt on change features.
RB2/PWM1	bit2	ST	Input/Output or the RB2/PWM1 output pin. Software programmable weak pull-up and interrupt on change features.
RB3/PWM2	bit3	ST	Input/Output or the RB3/PWM2 output pin. Software programmable weak pull-up and interrupt on change features.
RB4/TCLK12	bit4	ST	Input/Output or the external clock input to Timer1 and Timer2. Software programmable weak pull-up and interrupt on change features.
RB5/TCLK3	bit5	ST	Input/Output or the external clock input to Timer3. Software programmable weak pull-up and interrupt on change features.
RB6	bit6	ST	Input/Output pin. Software programmable weak pull-up and interrupt on change features.
RB7	bit7	ST	Input/Output pin. Software programmable weak pull-up and interrupt on change features.

Legend: ST = Schmitt Trigger input.

**TABLE 9-4: REGISTERS/BITS ASSOCIATED WITH PORTB**

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)
12h, Bank 0	PORTB	PORTB data latch								xxxx xxxx	uuuu uuuu
11h, Bank 0	DDRB	Data direction register for PORTB								1111 1111	1111 1111
10h, Bank 0	PORTA	RBPU	—	RA5	RA4	RA3	RA2	RA1/T0CKI	RA0/INT	0-xx xxxx	0-uu uuuu
06h, Unbanked	CPUSTA	—	—	STKAV	GLINTD	T0	PD	—	—	--11 11--	--11 qq--
07h, Unbanked	INTSTA	PEIF	T0CKIF	T0IF	INTF	PEIE	T0CKIE	T0IE	INTE	0000 0000	0000 0000
16h, Bank 1	PIR	RBIF	TMR3IF	TMR2IF	TMR1IF	CA2IF	CA1IF	TXIF	RCIF	0000 0010	0000 0010
17h, Bank 1	PIE	RBIE	TMR3IE	TMR2IE	TMR1IE	CA2IE	CA1IE	TXIE	RCIE	0000 0000	0000 0000
16h, Bank 3	TCON1	CA2ED1	CA2ED0	CA1ED1	CA1ED0	T16	TMR3CS	TMR2CS	TMR1CS	0000 0000	0000 0000
17h, Bank 3	TCON2	CA2OVF	CA1OVF	PWM2ON	PWM1ON	CA1/PR3	TMR3ON	TMR2ON	TMR1ON	0000 0000	0000 0000

Legend: x = unknown, u = unchanged, - = unimplemented read as '0', q = Value depends on condition.

Shaded cells are not used by PORTB.

Note 1: Other (non power-up) resets include: external reset through  $\overline{\text{MCLR}}$  and the Watchdog Timer Reset.

## 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.

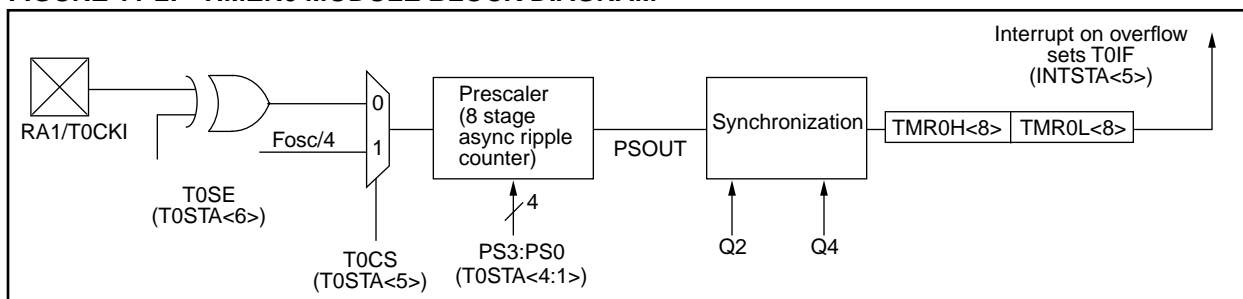
## 11.2 Using Timer0 with External Clock

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 3TOSC and 7TOSC. Thus, for example, measuring the interval between two edges (e.g. period) will be accurate within  $\pm 4TOSC$  ( $\pm 121$  ns @ 33 MHz).

**FIGURE 11-2: TIMER0 MODULE BLOCK DIAGRAM**



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

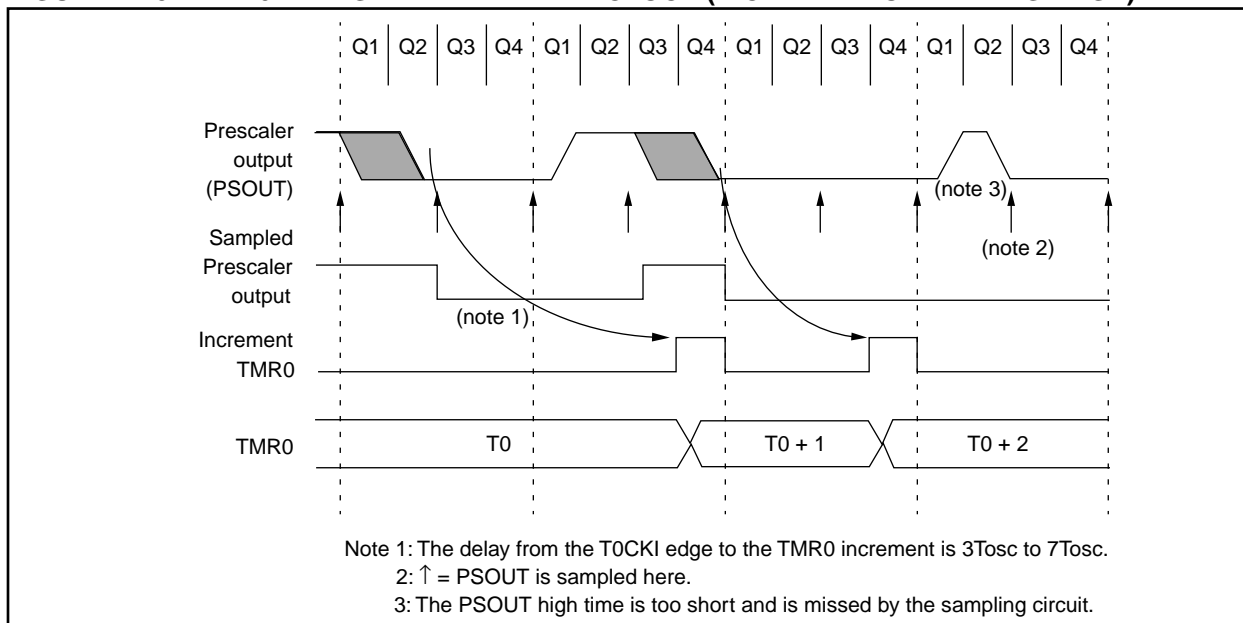


FIGURE 11-5: TMR0 READ/WRITE IN TIMER MODE

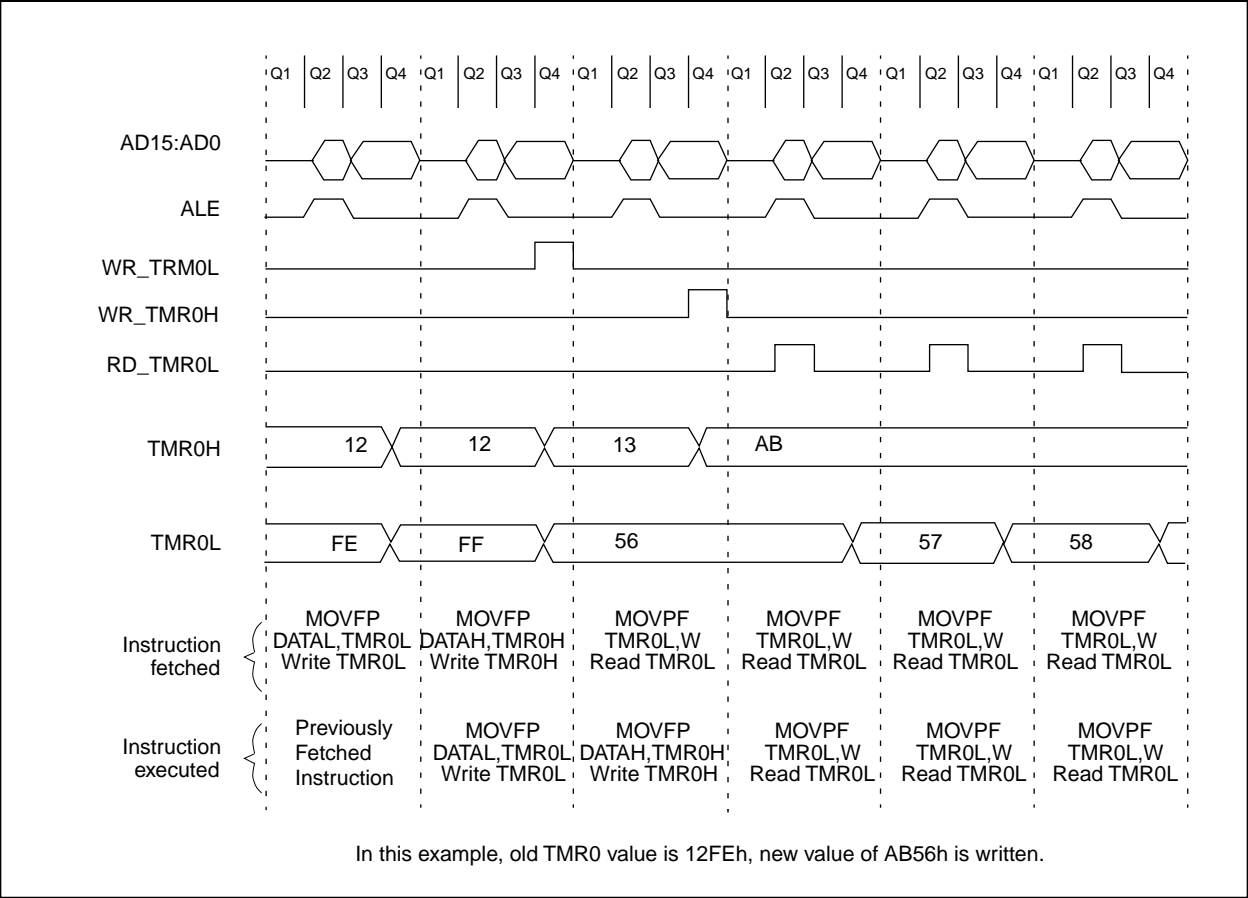


TABLE 11-1: REGISTERS/BITS ASSOCIATED WITH TIMER0

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.

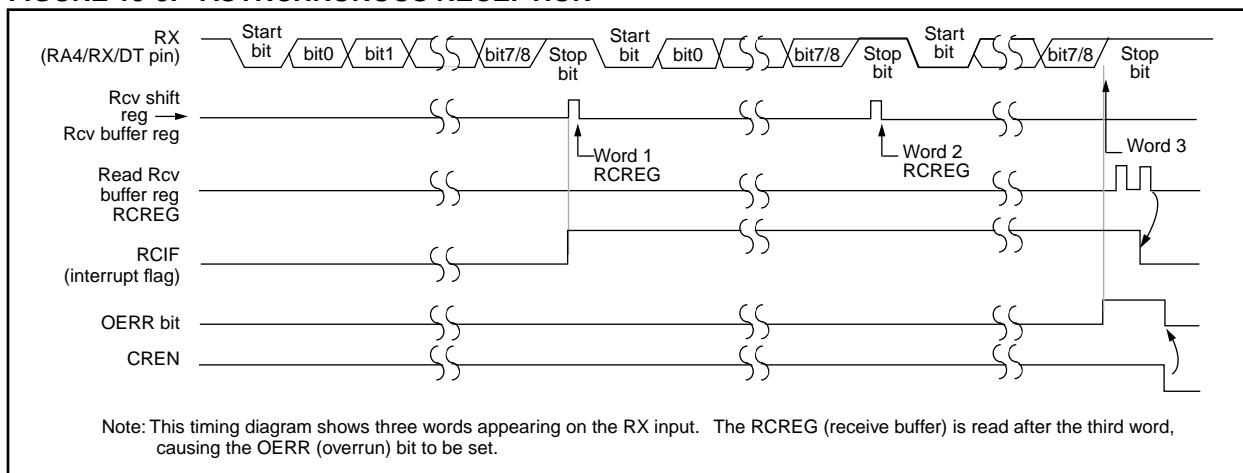
# PIC17C4X

Steps to follow when setting up an Asynchronous Reception:

1. Initialize the SPBRG register for the appropriate baud rate.
2. Enable the asynchronous serial port by clearing the SYNC bit and setting the SPEN bit.
3. If interrupts are desired, then set the RCIE bit.
4. If 9-bit reception is desired, then set the RX9 bit.
5. Enable the reception by setting the CREN bit.
6. The RCIF bit will be set when reception completes and an interrupt will be generated if the RCIE bit was set.
7. Read RCSTA to get the ninth bit (if enabled) and FERR bit to determine if any error occurred during reception.
8. Read RCREG for the 8-bit received data.
9. If an overrun error occurred, clear the error by clearing the OERR bit.

**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 13-8: ASYNCHRONOUS RECEPTION**



**TABLE 13-6: REGISTERS ASSOCIATED WITH ASYNCHRONOUS RECEPTION**

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)
16h, Bank 1	PIR	RBIF	TMR3IF	TMR2IF	TMR1IF	CA2IF	CA1IF	TXIF	RCIF	0000 0010	0000 0010
13h, Bank 0	RCSTA	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00u
14h, Bank 0	RCREG	RX7	RX6	RX5	RX4	RX3	RX2	RX1	RX0	xxxx xxxx	uuuu uuuu
17h, Bank 1	PIE	RBIE	TMR3IE	TMR2IE	TMR1IE	CA2IE	CA1IE	TXIE	RCIE	0000 0000	0000 0000
15h, Bank 0	TXSTA	CSRC	TX9	TXEN	SYNC	—	—	TRMT	TX9D	0000 --1x	0000 --1u
17h, Bank 0	SPBRG	Baud rate generator register								xxxx xxxx	uuuu uuuu

Legend: x = unknown, u = unchanged, - = unimplemented read as a '0', shaded cells are not used for asynchronous reception.

Note 1: Other (non power-up) resets include: external reset through MCLR and Watchdog Timer Reset.

**TABLE 13-8: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER RECEPTION**

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)
16h, Bank 1	PIR	RBIF	TMR3IF	TMR2IF	TMR1IF	CA2IF	CA1IF	TXIF	RCIF	0000 0010	0000 0010
13h, Bank 0	RCSTA	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00u
14h, Bank 0	RCREG	RX7	RX6	RX5	RX4	RX3	RX2	RX1	RX0	xxxx xxxx	uuuu uuuu
17h, Bank 1	PIE	RBIE	TMR3IE	TMR2IE	TMR1IE	CA2IE	CA1IE	TXIE	RCIE	0000 0000	0000 0000
15h, Bank 0	TXSTA	CSRC	TX9	TXEN	SYNC	—	—	TRMT	TX9D	0000 --1x	0000 --1u
17h, Bank 0	SPBRG	Baud rate generator register								xxxx xxxx	uuuu uuuu

Legend: x = unknown, u = unchanged, - = unimplemented read as a '0', shaded cells are not used for synchronous master reception.

Note 1: Other (non power-up) resets include: external reset through  $\overline{\text{MCLR}}$  and Watchdog Timer Reset.

## 16.6 PICDEM-1 Low-Cost PIC16/17 Demonstration Board

The PICDEM-1 is a simple board which demonstrates the capabilities of several of Microchip's microcontrollers. The microcontrollers supported are: PIC16C5X (PIC16C54 to PIC16C58A), PIC16C61, PIC16C62X, PIC16C71, PIC16C8X, PIC17C42, PIC17C43 and PIC17C44. All necessary hardware and software is included to run basic demo programs. The users can program the sample microcontrollers provided with the PICDEM-1 board, on a PRO MATE II or PICSTART-16B programmer, and easily test firmware. The user can also connect the PICDEM-1 board to the PICMASTER emulator and download the firmware to the emulator for testing. Additional prototype area is available for the user to build some additional hardware and connect it to the microcontroller socket(s). Some of the features include an RS-232 interface, a potentiometer for simulated analog input, push-button switches and eight LEDs connected to PORTB.

## 16.7 PICDEM-2 Low-Cost PIC16CXX Demonstration Board

The PICDEM-2 is a simple demonstration board that supports the PIC16C62, PIC16C64, PIC16C65, PIC16C73 and PIC16C74 microcontrollers. All the necessary hardware and software is included to run the basic demonstration programs. The user can program the sample microcontrollers provided with the PICDEM-2 board, on a PRO MATE II programmer or PICSTART-16C, and easily test firmware. The PICMASTER emulator may also be used with the PICDEM-2 board to test firmware. Additional prototype area has been provided to the user for adding additional hardware and connecting it to the microcontroller socket(s). Some of the features include a RS-232 interface, push-button switches, a potentiometer for simulated analog input, a Serial EEPROM to demonstrate usage of the I<sup>2</sup>C bus and separate headers for connection to an LCD module and a keypad.

## 16.8 PICDEM-3 Low-Cost PIC16CXXX Demonstration Board

The PICDEM-3 is a simple demonstration board that supports the PIC16C923 and PIC16C924 in the PLCC package. It will also support future 44-pin PLCC microcontrollers with a LCD Module. All the necessary hardware and software is included to run the basic demonstration programs. The user can program the sample microcontrollers provided with the PICDEM-3 board, on a PRO MATE II programmer or PICSTART Plus with an adapter socket, and easily test firmware. The PICMASTER emulator may also be used with the PICDEM-3 board to test firmware. Additional prototype area has been provided to the user for adding hardware and connecting it to the microcontroller socket(s). Some of the features

include an RS-232 interface, push-button switches, a potentiometer for simulated analog input, a thermistor and separate headers for connection to an external LCD module and a keypad. Also provided on the PICDEM-3 board is an LCD panel, with 4 commons and 12 segments, that is capable of displaying time, temperature and day of the week. The PICDEM-3 provides an additional RS-232 interface and Windows 3.1 software for showing the demultiplexed LCD signals on a PC. A simple serial interface allows the user to construct a hardware demultiplexer for the LCD signals. PICDEM-3 will be available in the 3rd quarter of 1996.

## 16.9 MPLAB Integrated Development Environment Software

The MPLAB IDE Software brings an ease of software development previously unseen in the 8-bit microcontroller market. MPLAB is a windows based application which contains:

- A full featured editor
- Three operating modes
  - editor
  - emulator
  - simulator
- A project manager
- Customizable tool bar and key mapping
- A status bar with project information
- Extensive on-line help

MPLAB allows you to:

- Edit your source files (either assembly or 'C')
- One touch assemble (or compile) and download to PIC16/17 tools (automatically updates all project information)
- Debug using:
  - source files
  - absolute listing file
- Transfer data dynamically via DDE (soon to be replaced by OLE)
- Run up to four emulators on the same PC

The ability to use MPLAB with Microchip's simulator allows a consistent platform and the ability to easily switch from the low cost simulator to the full featured emulator with minimal retraining due to development tools.

## 16.10 Assembler (MPASM)

The MPASM Universal Macro Assembler is a PC-hosted symbolic assembler. It supports all microcontroller series including the PIC12C5XX, PIC14000, PIC16C5X, PIC16CXXX, and PIC17CXX families.

MPASM offers full featured Macro capabilities, conditional assembly, and several source and listing formats. It generates various object code formats to support Microchip's development tools as well as third party programmers.

# PIC17C4X

Applicable Devices 42 R42 42A 43 R43 44

DC CHARACTERISTICS		Standard Operating Conditions (unless otherwise stated)					
		Operating temperature					
		-40°C ≤ TA ≤ +40°C					
		Operating voltage VDD range as described in Section 17.1					
Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
		<b>Internal Program Memory Programming Specs (Note 4)</b>					
D110	VPP	Voltage on $\overline{\text{MCLR}}/\text{VPP}$ pin	12.75	–	13.25	V	Note 5
D111	VDDP	Supply voltage during programming	4.75	5.0	5.25	V	
D112	IPP	Current into $\overline{\text{MCLR}}/\text{VPP}$ pin	–	25 ‡	50 ‡	mA	
D113	IDDP	Supply current during programming	–	–	30 ‡	mA	
D114	TPROG	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).

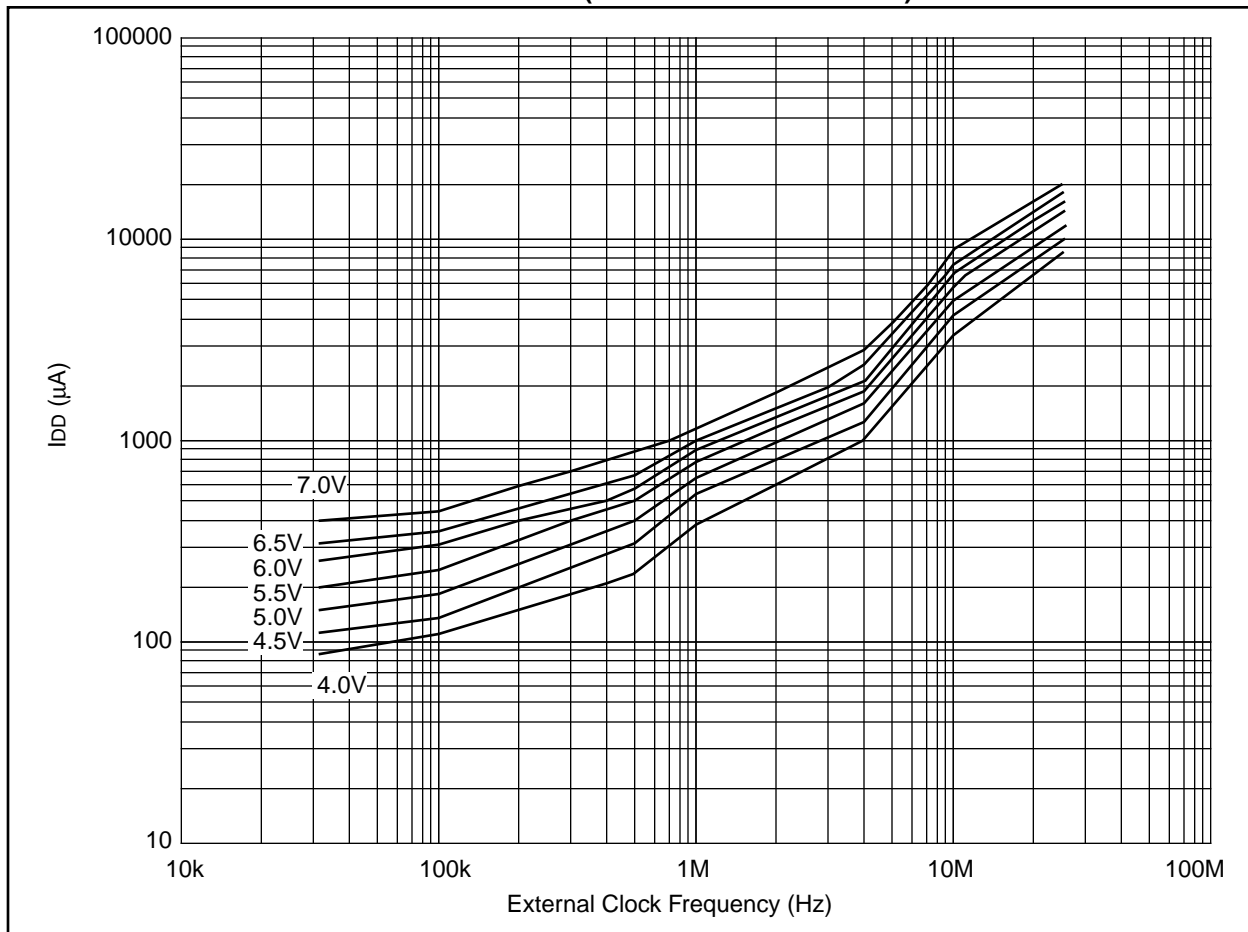
5: The  $\overline{\text{MCLR}}/\text{VPP}$  pin may be kept in this range at times other than programming, but is not recommended.

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

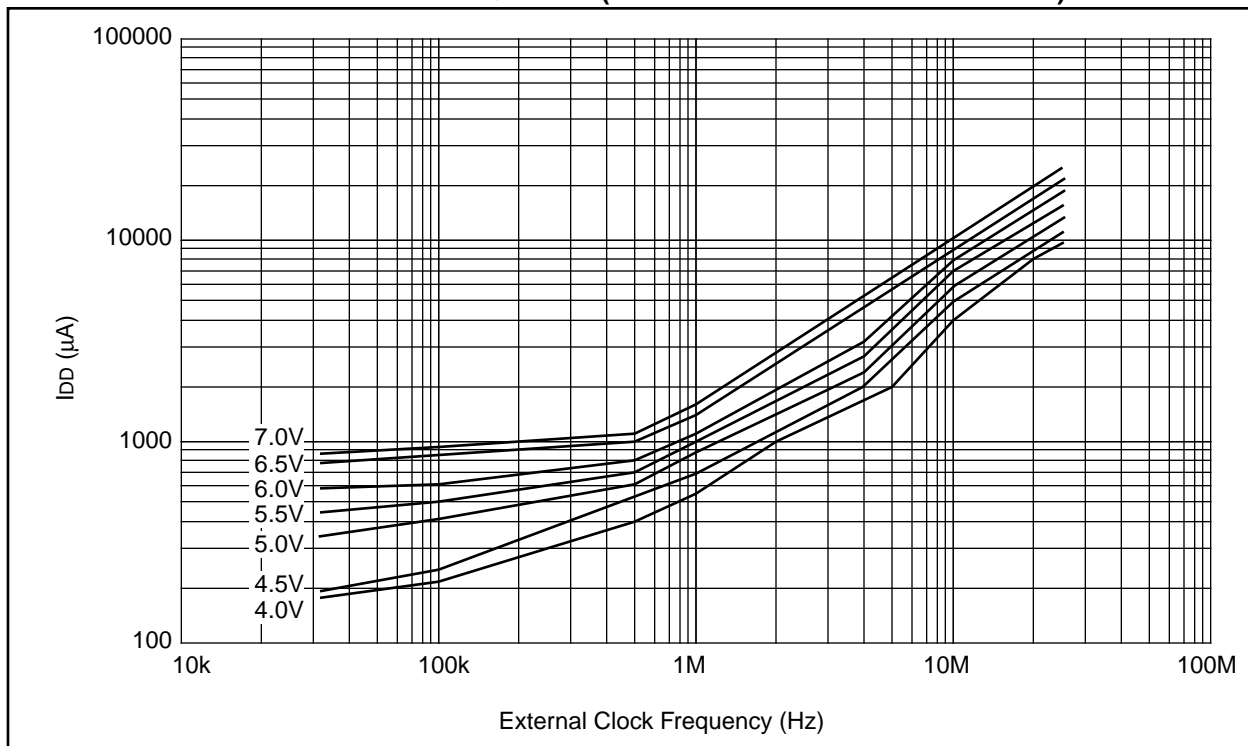
**Note:** When using the Table Write for internal programming, the device temperature must be less than 40°C.



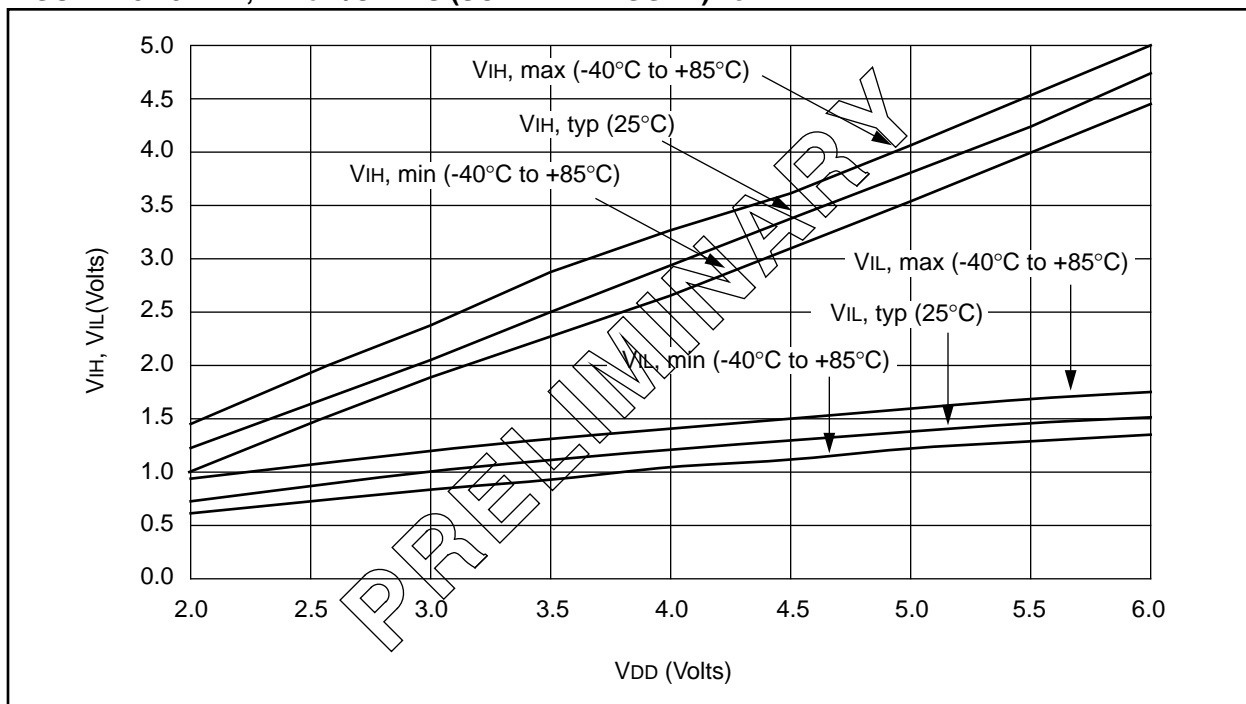
**FIGURE 18-7: TYPICAL  $I_{DD}$  vs. FREQUENCY (EXTERNAL CLOCK 25°C)**



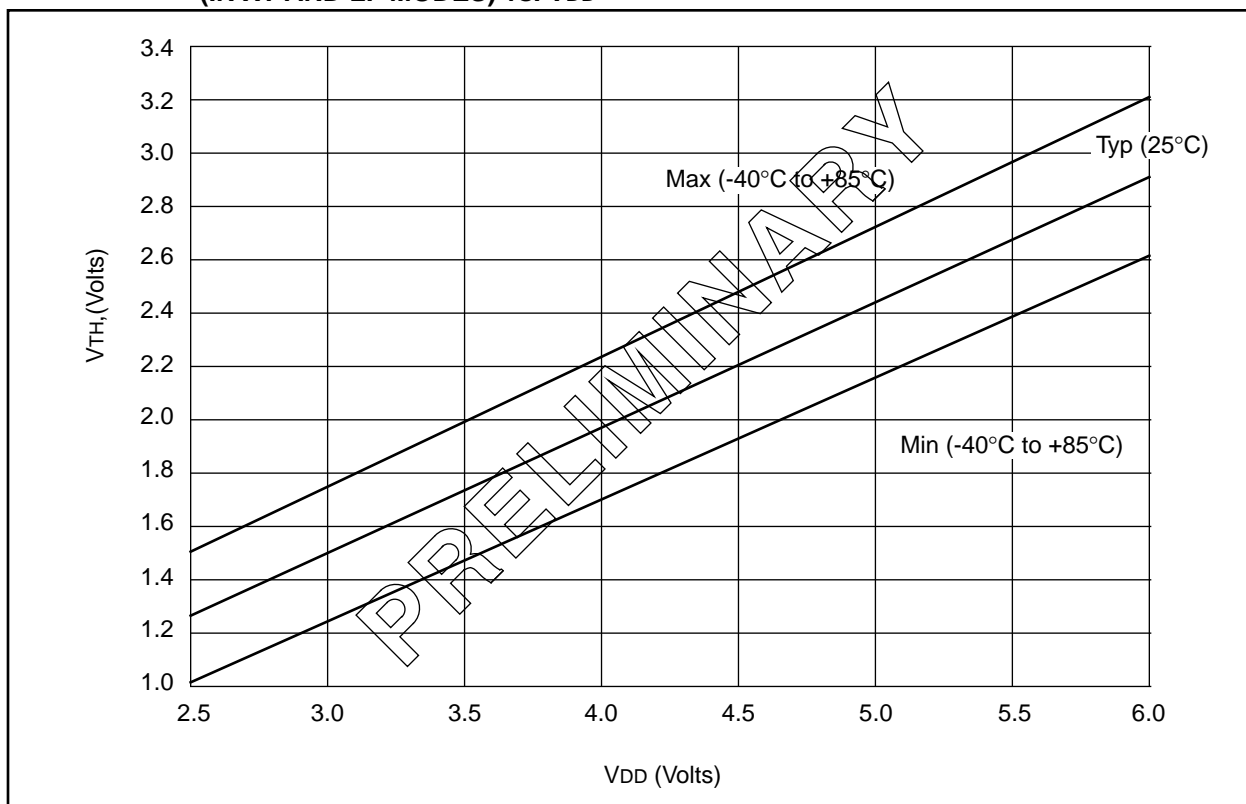
**FIGURE 18-8: MAXIMUM  $I_{DD}$  vs. FREQUENCY (EXTERNAL CLOCK 125°C TO -40°C)**



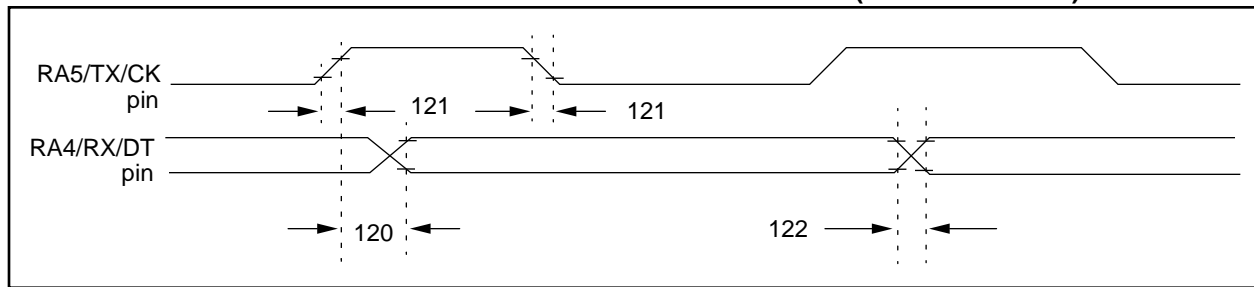
**FIGURE 18-19:  $V_{TH}$ ,  $V_{IL}$  of I/O PINS (SCHMITT TRIGGER) vs.  $V_{DD}$**



**FIGURE 18-20:  $V_{TH}$  (INPUT THRESHOLD VOLTAGE) OF OSC1 INPUT (IN XT AND LF MODES) vs.  $V_{DD}$**



**FIGURE 19-9: USART MODULE: SYNCHRONOUS TRANSMISSION (MASTER/SLAVE) TIMING**

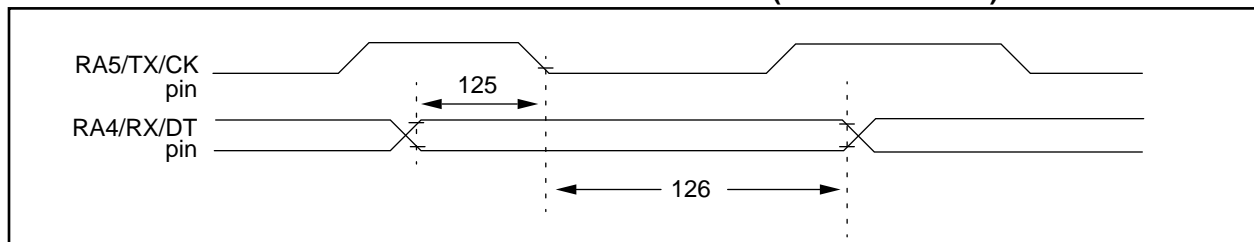


**TABLE 19-9: SYNCHRONOUS TRANSMISSION REQUIREMENTS**

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
120	TckH2dtV	SYNC XMIT (MASTER & SLAVE) Clock high to data out valid	PIC17CR42/42A/43/R43/44	—	—	50	ns
			PIC17LCR42/42A/43/R43/44	—	—	75	ns
121	TckRF	Clock out rise time and fall time (Master Mode)	PIC17CR42/42A/43/R43/44	—	—	25	ns
			PIC17LCR42/42A/43/R43/44	—	—	40	ns
122	TdtRF	Data out rise time and fall time	PIC17CR42/42A/43/R43/44	—	—	25	ns
			PIC17LCR42/42A/43/R43/44	—	—	40	ns

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

**FIGURE 19-10: USART MODULE: SYNCHRONOUS RECEIVE (MASTER/SLAVE) TIMING**



**TABLE 19-10: SYNCHRONOUS RECEIVE REQUIREMENTS**

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
125	TdtV2ckL	SYNC RCV (MASTER & SLAVE) Data hold before CK↓ (DT hold time)	15	—	—	ns	
126	TckL2dtl	Data hold after CK↓ (DT hold time)	15	—	—	ns	

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

# PIC17C4X

## E.2 PIC16C5X Family of Devices

	Clock		Memory		Peripherals		Features	
	Maximum Frequency of Operation (MHz)	Program Memory (x12 words)	ROM	RAM Data Memory (bytes)	Timer Module(s)	I/O Pins	Voltage Range (Volts)	Number of Instructions
PIC16C52	4	384	—	25	TMR0	12	2.5-6.25	33
PIC16C54	20	512	—	25	TMR0	12	2.5-6.25	33
PIC16C54A	20	512	—	25	TMR0	12	2.0-6.25	33
PIC16CR54A	20	—	512	25	TMR0	12	2.0-6.25	33
PIC16C55	20	512	—	24	TMR0	20	2.5-6.25	33
PIC16C56	20	1K	—	25	TMR0	12	2.5-6.25	33
PIC16C57	20	2K	—	72	TMR0	20	2.5-6.25	33
PIC16CR57B	20	—	2K	72	TMR0	20	2.5-6.25	33
PIC16C58A	20	2K	—	73	TMR0	12	2.0-6.25	33
PIC16CR58A	20	—	2K	73	TMR0	12	2.5-6.25	33

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

## APPENDIX F: ERRATA FOR PIC17C42 SILICON

The PIC17C42 devices that you have received have the following anomalies. At present there is no intention for future revisions to the present PIC17C42 silicon. If these cause issues for the application, it is recommended that you select the PIC17C42A device.

**Note:** New designs should use the PIC17C42A.

### Design considerations

The device must not be operated outside of the specified voltage range. An external reset circuit must be used to ensure the device is in reset when a brown-out occurs or the VDD rise time is too long. Failure to ensure that the device is in reset when device voltage is out of specification may cause the device to lock-up and ignore the MCLR pin.

1. When the Oscillator Start-Up Timer (OST) is enabled (in LF or XT oscillator modes), any interrupt that wakes the processor may cause a WDT reset. This occurs when the WDT is greater than or equal to 50% time-out period when the `SLEEP` instruction is executed. This will not occur in either the EC or RC oscillator modes.

#### Work-arounds

- a) Always ensure that the `CLRWDT` instruction is executed before the WDT increments past 50% of the WDT period. This will keep the "false" WDT reset from occurring.
- b) When using the WDT as a normal timer (WDT disabled), ensure that the WDT is less than or equal to 50% time-out period when the `SLEEP` instruction is executed. This can be done by monitoring the  $\overline{TO}$  bit for changing state from set to clear. Example 1 shows putting the PIC17C42 to sleep.

#### EXAMPLE F-1: PIC17C42 TO SLEEP

```

BTFSS    CPUSTA, TO    ; TO = 0?
CLRWDT                    ; YES, WDT = 0
LOOP     BTFSC    CPUSTA, TO    ; WDT rollover?
        GOTO     LOOP          ; NO, Wait
        SLEEP                    ; YES, goto Sleep
    
```

2. When the clock source of Timer1 or Timer2 is selected to external clock, the overflow interrupt flag will be set twice, once when the timer equals the period, and again when the timer value is reset to 0h. If the latency to clear TMRxIF is greater than the time to the next clock pulse, no problems will be noticed. If the latency is less than the time to the next timer clock pulse, the interrupt will be serviced twice.

#### Work-arounds

- a) Ensure that the timer has rolled over to 0h before clearing the flag bit.
- b) Clear the timer in software. Clearing the timer in software causes the period to be one count less than expected.