



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	16MHz
Connectivity	UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	33
Program Memory Size	16KB (8K 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	Surface Mount
Package / Case	44-TQFP
Supplier Device Package	44-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic17c44t-16-pt

FIGURE 3-2: PIC17CR42/42A/43/R43/44 BLOCK DIAGRAM

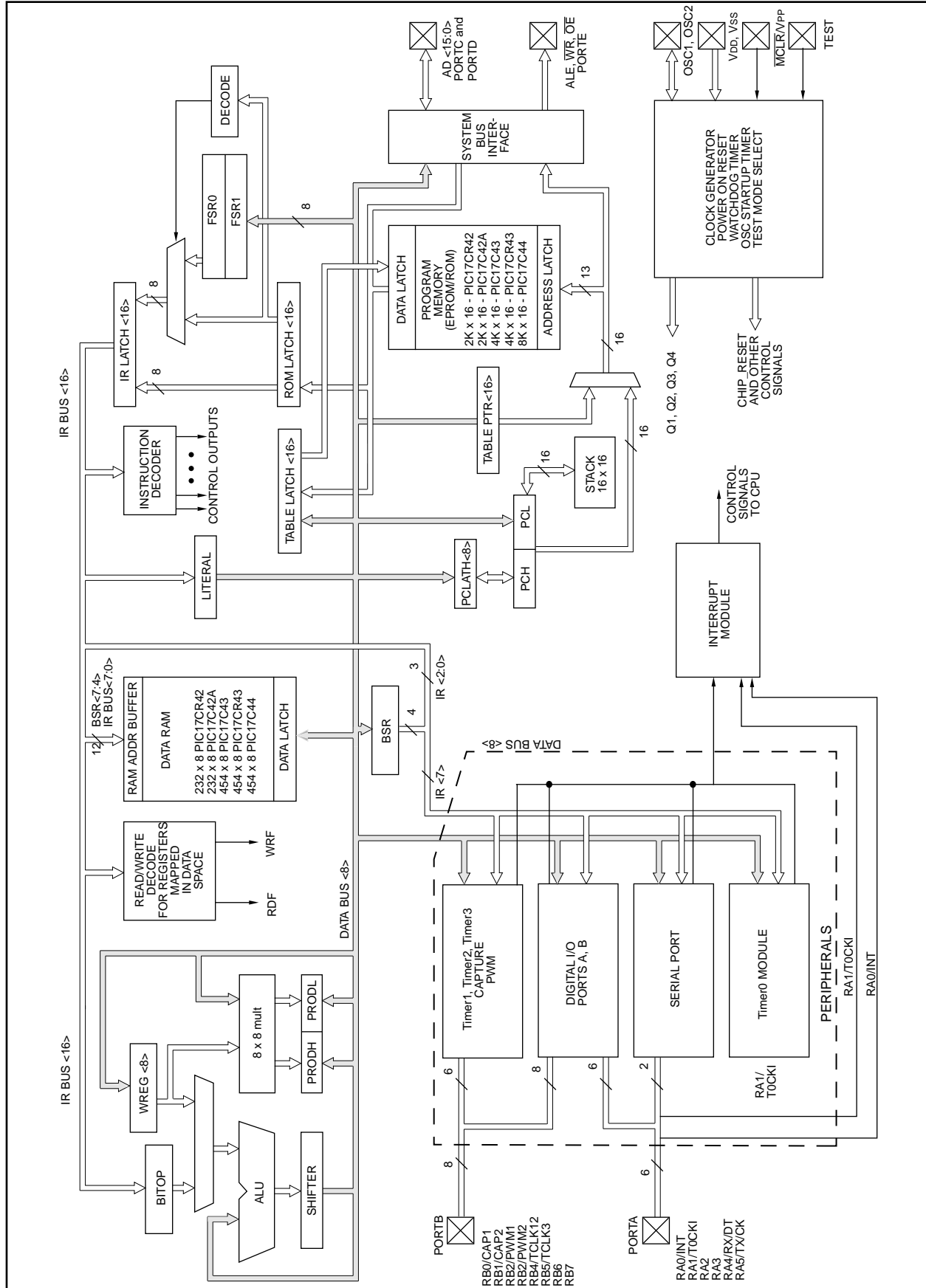


FIGURE 4-2: TIME-OUT SEQUENCE ON POWER-UP ($\overline{\text{MCLR}}$ TIED TO V_{DD})

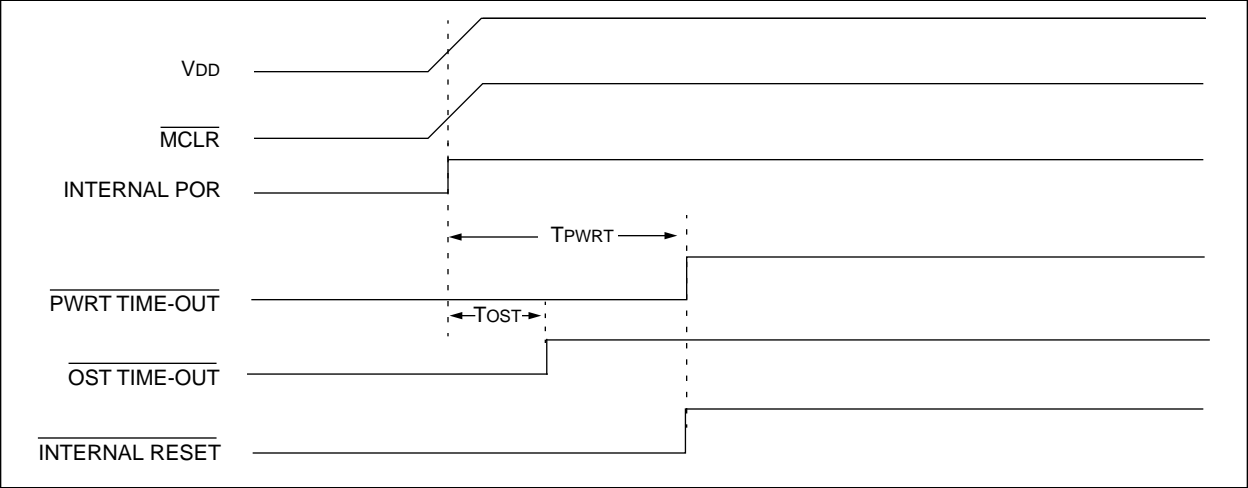


FIGURE 4-3: TIME-OUT SEQUENCE ON POWER-UP ($\overline{\text{MCLR}}$ NOT TIED TO V_{DD})

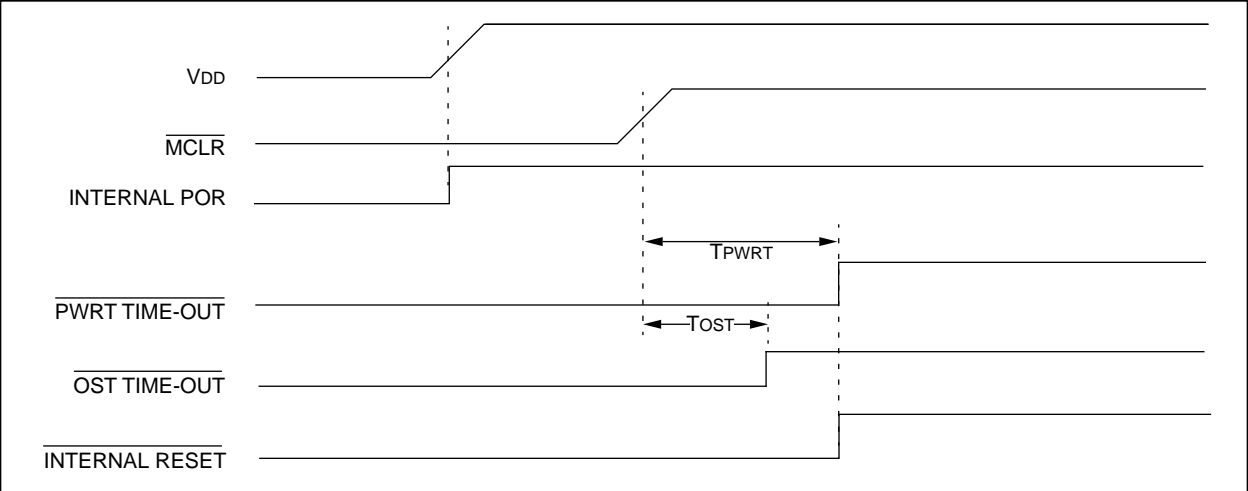
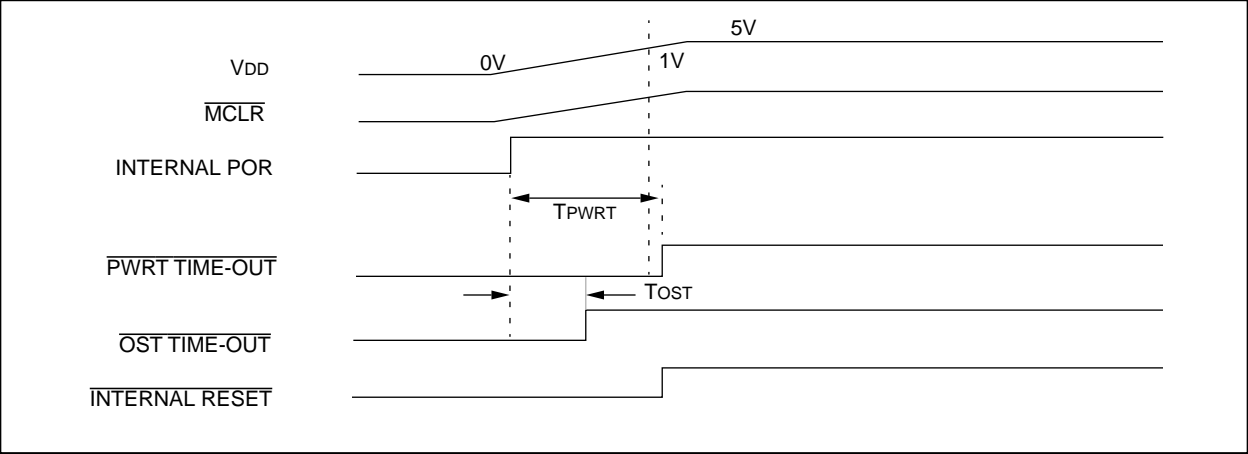


FIGURE 4-4: SLOW RISE TIME ($\overline{\text{MCLR}}$ TIED TO V_{DD})



5.2 Peripheral Interrupt Enable Register (PIE)

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

FIGURE 5-3: PIE REGISTER (ADDRESS: 17h, BANK 1)

R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0
RBIE	TMR3IE	TMR2IE	TMR1IE	CA2IE	CA1IE	TXIE	RCIE
bit7							bit0
<p>bit 7: RBIE: PORTB Interrupt on Change Enable bit 1 = Enable PORTB interrupt on change 0 = Disable PORTB interrupt on change</p> <p>bit 6: TMR3IE: Timer3 Interrupt Enable bit 1 = Enable Timer3 interrupt 0 = Disable Timer3 interrupt</p> <p>bit 5: TMR2IE: Timer2 Interrupt Enable bit 1 = Enable Timer2 interrupt 0 = Disable Timer2 interrupt</p> <p>bit 4: TMR1IE: Timer1 Interrupt Enable bit 1 = Enable Timer1 interrupt 0 = Disable Timer1 interrupt</p> <p>bit 3: CA2IE: Capture2 Interrupt Enable bit 1 = Enable Capture interrupt on RB1/CAP2 pin 0 = Disable Capture interrupt on RB1/CAP2 pin</p> <p>bit 2: CA1IE: Capture1 Interrupt Enable bit 1 = Enable Capture interrupt on RB2/CAP1 pin 0 = Disable Capture interrupt on RB2/CAP1 pin</p> <p>bit 1: TXIE: USART Transmit Interrupt Enable bit 1 = Enable Transmit buffer empty interrupt 0 = Disable Transmit buffer empty interrupt</p> <p>bit 0: RCIE: USART Receive Interrupt Enable bit 1 = Enable Receive buffer full interrupt 0 = Disable Receive buffer full interrupt</p>							

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

6.0 MEMORY ORGANIZATION

There are two memory blocks in the PIC17C4X; program memory and data memory. Each block has its own bus, so that access to each block can occur during the same oscillator cycle.

The data memory can further be broken down into General Purpose RAM and the Special Function Registers (SFRs). The operation of the SFRs that control the "core" are described here. The SFRs used to control the peripheral modules are described in the section discussing each individual peripheral module.

6.1 Program Memory Organization

PIC17C4X devices have a 16-bit program counter capable of addressing a 64K x 16 program memory space. The reset vector is at 0000h and the interrupt vectors are at 0008h, 0010h, 0018h, and 0020h (Figure 6-1).

6.1.1 PROGRAM MEMORY OPERATION

The PIC17C4X can operate in one of four possible program memory configurations. The configuration is selected by two configuration bits. The possible modes are:

- Microprocessor
- Microcontroller
- Extended Microcontroller
- Protected Microcontroller

The microcontroller and protected microcontroller modes only allow internal execution. Any access beyond the program memory reads unknown data. The protected microcontroller mode also enables the code protection feature.

The extended microcontroller mode accesses both the internal program memory as well as external program memory. Execution automatically switches between internal and external memory. The 16-bits of address allow a program memory range of 64K-words.

The microprocessor mode only accesses the external program memory. The on-chip program memory is ignored. The 16-bits of address allow a program memory range of 64K-words. Microprocessor mode is the default mode of an unprogrammed device.

The different modes allow different access to the configuration bits, test memory, and boot ROM. Table 6-1 lists which modes can access which areas in memory. Test Memory and Boot Memory are not required for normal operation of the device. Care should be taken to ensure that no unintended branches occur to these areas.

FIGURE 6-1: PROGRAM MEMORY MAP AND STACK

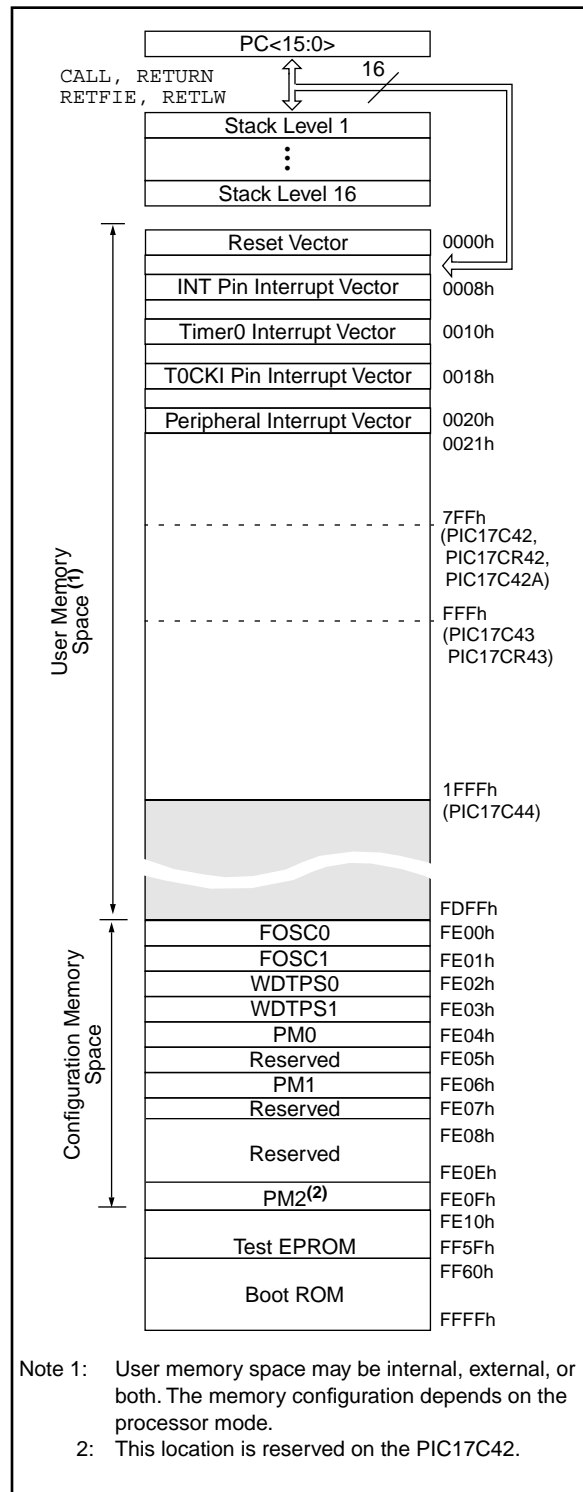


FIGURE 7-3: TLRD INSTRUCTION OPERATION

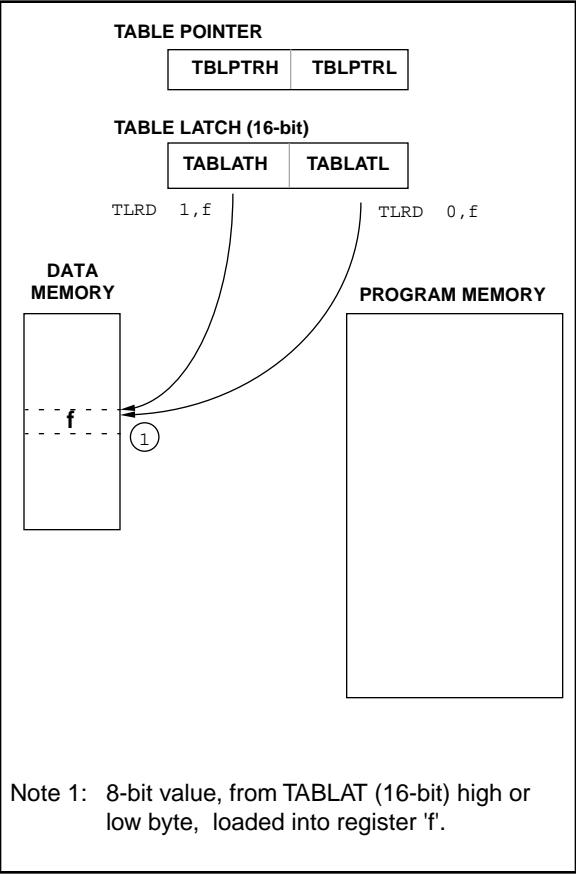


FIGURE 7-4: TABLRD INSTRUCTION OPERATION

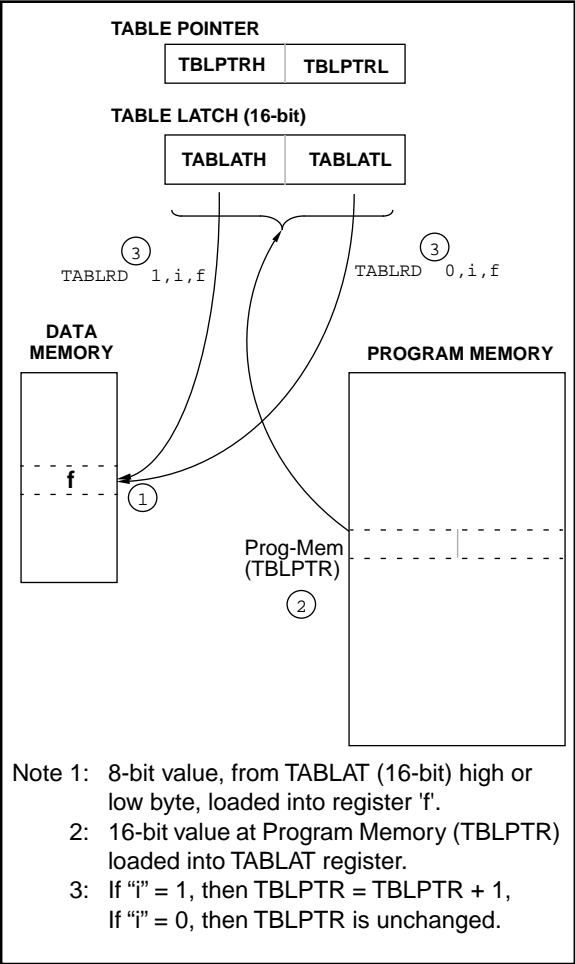
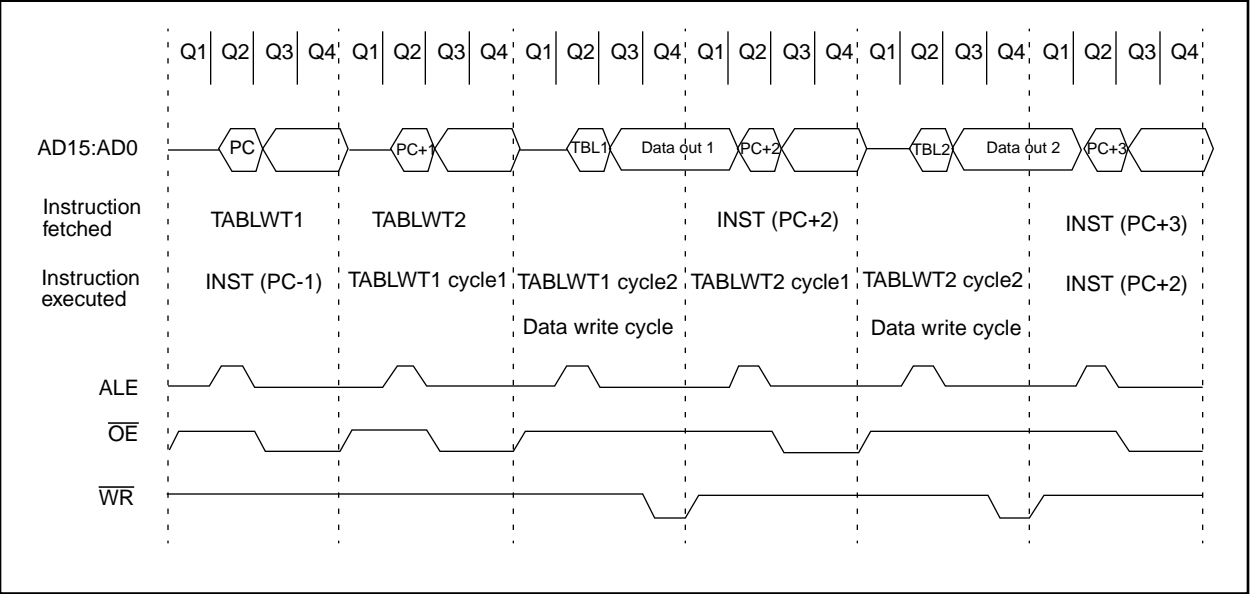


FIGURE 7-6: CONSECUTIVE TABLWT WRITE TIMING (EXTERNAL MEMORY)



Example 8-3 shows the sequence to do a 16 x 16 unsigned multiply. Equation 8-1 shows the algorithm that is used. The 32-bit result is stored in 4 registers RES3:RES0.

EQUATION 8-1: 16 x 16 UNSIGNED MULTIPLICATION ALGORITHM

$$\begin{aligned}
 \text{RES3:RES0} &= \text{ARG1H:ARG1L} * \text{ARG2H:ARG2L} \\
 &= (\text{ARG1H} * \text{ARG2H} * 2^{16}) + \\
 &\quad (\text{ARG1H} * \text{ARG2L} * 2^8) + \\
 &\quad (\text{ARG1L} * \text{ARG2H} * 2^8) + \\
 &\quad (\text{ARG1L} * \text{ARG2L})
 \end{aligned}$$

EXAMPLE 8-3: 16 x 16 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        ;

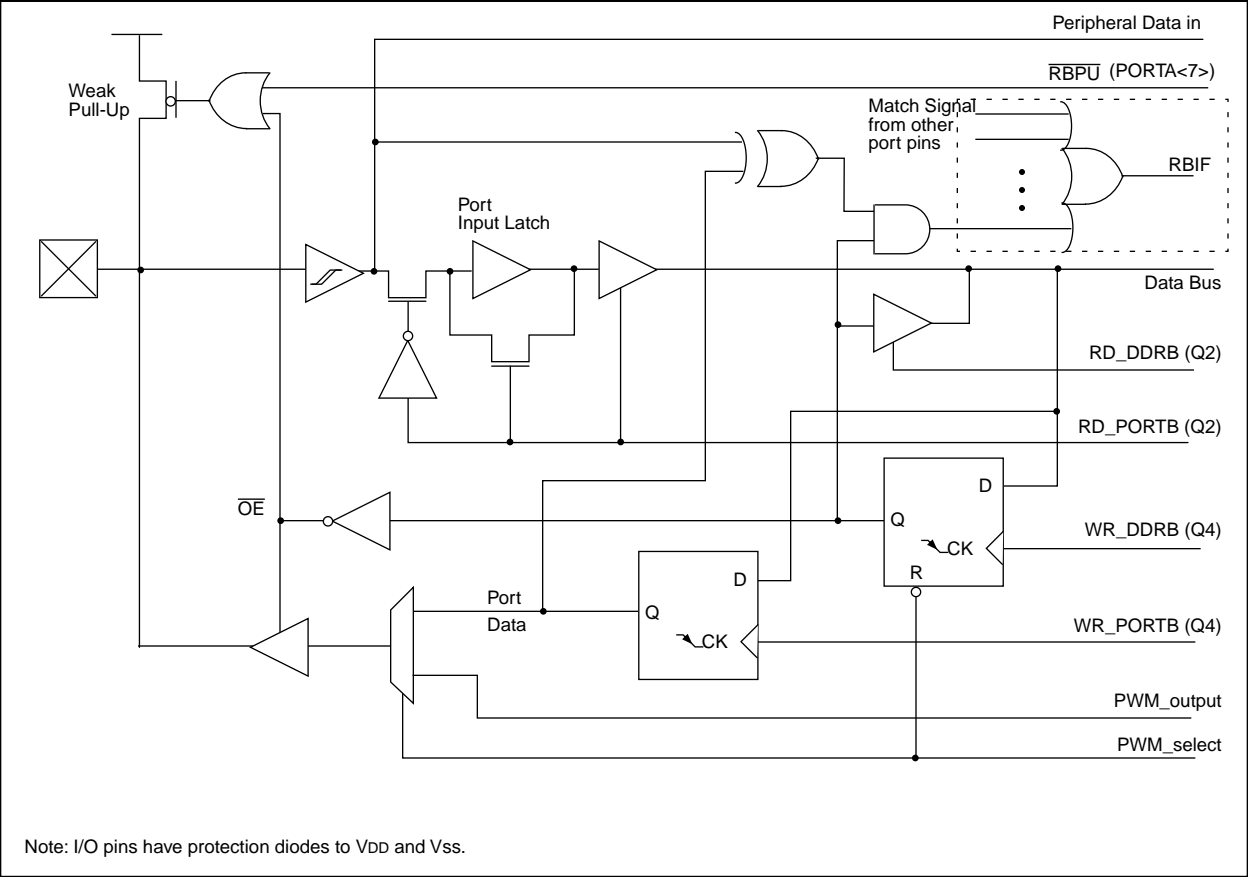
;

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        ;

```

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



12.2.3 EXTERNAL CLOCK INPUT FOR TIMER3

When TMR3CS is set, the 16-bit TMR3 increments on the falling edge of clock input TCLK3. The input on the RB5/TCLK3 pin is sampled and synchronized by the internal phase clocks twice every instruction cycle. This causes a delay from the time a falling edge appears on TCLK3 to the time TMR3 is actually incremented. For the external clock input timing requirements, see the Electrical Specification section. Figure 12-9 shows the timing diagram when operating from an external clock.

12.2.4 READING/WRITING TIMER3

Since Timer3 is a 16-bit timer and only 8-bits at a time can be read or written, care should be taken when reading or writing while the timer is running. The best method to read or write the timer is to stop the timer, perform any read or write operation, and then restart Timer3 (using the TMR3ON bit). However, if it is necessary to keep Timer3 free-running, care must be taken. For writing to the 16-bit TMR3, Example 12-2 may be used. For reading the 16-bit TMR3, Example 12-3 may be used. Interrupts must be disabled during this routine.

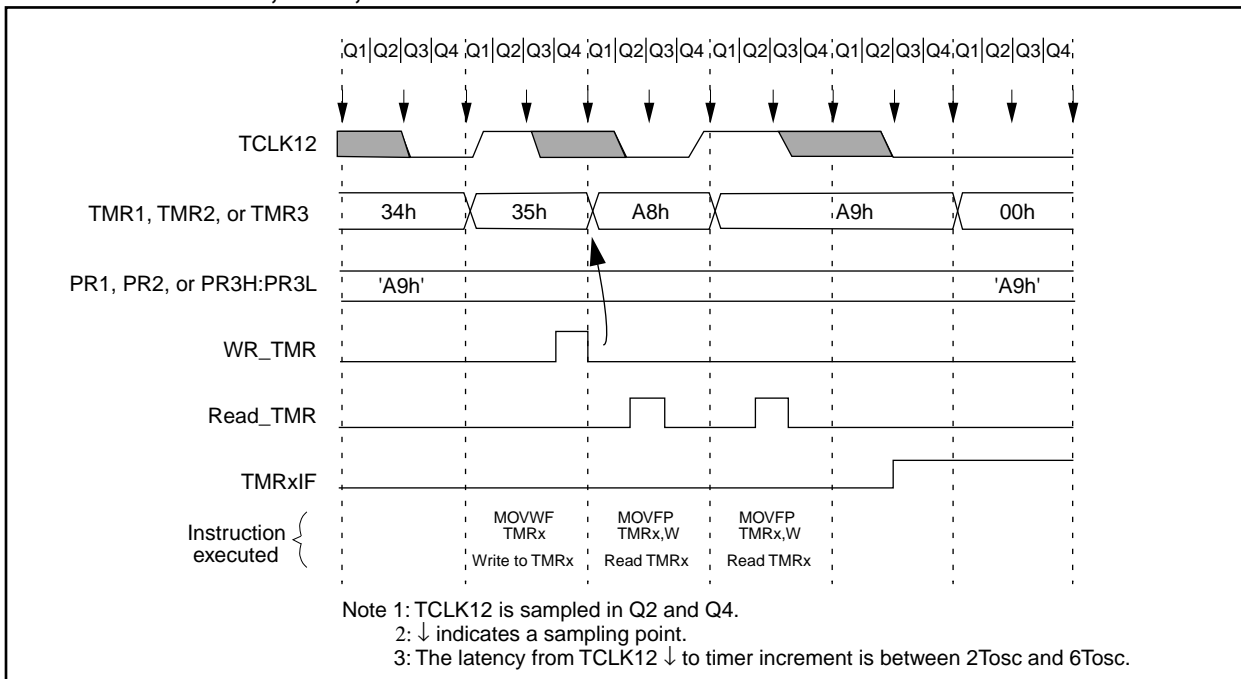
EXAMPLE 12-2: WRITING TO TMR3

```
BSF    CPUSTA, GLINTD ;Disable interrupt
MOVFP  RAM_L,  TMR3L  ;
MOVFP  RAM_H,  TMR3H  ;
BCF    CPUSTA, GLINTD ;Done,enable interrupt
```

EXAMPLE 12-3: READING FROM TMR3

```
MOVFP  TMR3L, TMPLO    ;read low tmr0
MOVFP  TMR3H, TMPHI    ;read high tmr0
MOVFP  TMPLO, WREG      ;tmplo -> wreg
CPFSLT TMR3L, WREG      ;tmr0l < wreg?
RETURN                    ;no then return
MOVFP  TMR3L, TMPLO    ;read low tmr0
MOVFP  TMR3H, TMPHI    ;read high tmr0
RETURN                    ;return
```

FIGURE 12-9: TMR1, TMR2, AND TMR3 OPERATION IN EXTERNAL CLOCK MODE



13.4 USART Synchronous Slave Mode

The synchronous slave mode differs from the master mode in the fact that the shift clock is supplied externally at the RA5/TX/CK pin (instead of being supplied internally in the master mode). This allows the device to transfer or receive data in the SLEEP mode. The slave mode is entered by clearing the CSRC (TXSTA<7>) bit.

13.4.1 USART SYNCHRONOUS SLAVE TRANSMIT

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

If two words are written to TXREG and then the SLEEP instruction executes, the following will occur. The first word will immediately transfer to the TSR and will transmit as the shift clock is supplied. The second word will remain in TXREG. TXIF will not be set. When the first word has been shifted out of TSR, TXREG will transfer the second word to the TSR and the TXIF flag will now be set. If TXIE is enabled, the interrupt will wake the chip from SLEEP and if the global interrupt is enabled, then the program will branch to interrupt vector (0020h).

Steps to follow when setting up a Synchronous Slave Transmission:

1. Enable the synchronous slave serial port by setting the SYNC and SPEN bits and clearing the CSRC bit.
2. Clear the CREN bit.
3. If interrupts are desired, then set the TXIE bit.
4. If 9-bit transmission is desired, then set the TX9 bit.
5. Start transmission by loading data to TXREG.
6. If 9-bit transmission is selected, the ninth bit should be loaded in TX9D.
7. 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.

13.4.2 USART SYNCHRONOUS SLAVE RECEPTION

Operation of the synchronous master and slave modes are identical except in the case of the SLEEP mode. Also, SREN is a don't care in slave mode.

If receive is enabled (CREN) prior to the SLEEP instruction, then a word may be received during SLEEP. On completely receiving the word, the RSR will transfer the data to RCREG (setting RCIF) and if the RCIE bit is set, the interrupt generated will wake the chip from SLEEP. If the global interrupt is enabled, the program will branch to the interrupt vector (0020h).

Steps to follow when setting up a Synchronous Slave Reception:

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

Note: To abort reception, either clear the SPEN bit, the SREN bit (when in single receive mode), or the CREN bit (when in continuous receive mode). This will reset the receive logic, so that it will be in the proper state when receive is re-enabled.

FIGURE 14-3: CRYSTAL OPERATION, OVERTONE CRYSTALS (XT OSC CONFIGURATION)

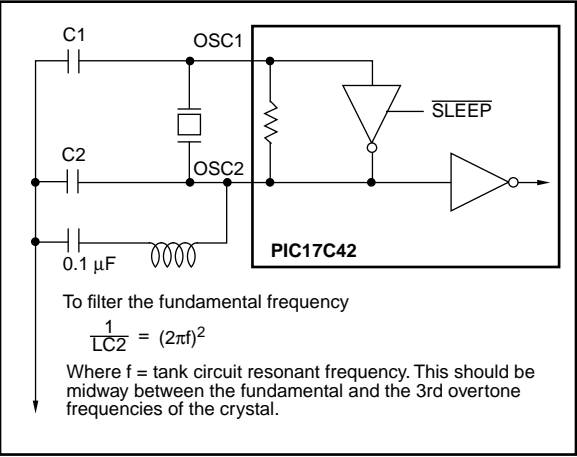


TABLE 14-2: CAPACITOR SELECTION FOR CERAMIC RESONATORS

Oscillator Type	Resonator Frequency	Capacitor Range C1 = C2
LF	455 kHz	15 - 68 pF
	2.0 MHz	10 - 33 pF
XT	4.0 MHz	22 - 68 pF
	8.0 MHz	33 - 100 pF
	16.0 MHz	33 - 100 pF

Higher capacitance increases the stability of the oscillator but also increases the start-up time. These values are for design guidance only. Since each resonator has its own characteristics, the user should consult the resonator manufacturer for appropriate values of external components.

Resonators Used:

455 kHz	Panasonic EFO-A455K04B	± 0.3%
2.0 MHz	Murata Erie CSA2.00MG	± 0.5%
4.0 MHz	Murata Erie CSA4.00MG	± 0.5%
8.0 MHz	Murata Erie CSA8.00MT	± 0.5%
16.0 MHz	Murata Erie CSA16.00MX	± 0.5%

Resonators used did not have built-in capacitors.

TABLE 14-3: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR

Osc Type	Freq	C1	C2
LF	32 kHz ⁽¹⁾	100-150 pF	100-150 pF
	1 MHz	10-33 pF	10-33 pF
	2 MHz	10-33 pF	10-33 pF
XT	2 MHz	47-100 pF	47-100 pF
	4 MHz	15-68 pF	15-68 pF
	8 MHz ⁽²⁾	15-47 pF	15-47 pF
	16 MHz	TBD	TBD
	25 MHz	15-47 pF	15-47 pF
	32 MHz ⁽³⁾	0 ⁽³⁾	0 ⁽³⁾

Higher capacitance increases the stability of the oscillator but also increases the start-up time and the oscillator current. These values are for design guidance only. Rs may be required in XT mode to avoid overdriving the crystals with low drive level specification. Since each crystal has its own characteristics, the user should consult the crystal manufacturer for appropriate values for external components.

Note 1: For VDD > 4.5V, C1 = C2 ≈ 30 pF is recommended.

2: Rs of 330Ω is required for a capacitor combination of 15/15 pF.

3: Only the capacitance of the board was present.

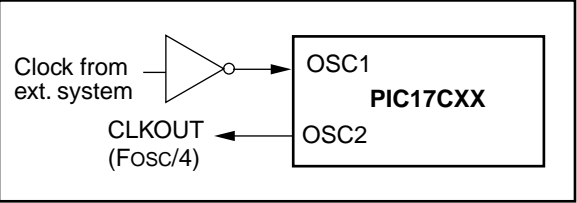
Crystals Used:

32.768 kHz	Epson C-001R32.768K-A	± 20 PPM
1.0 MHz	ECS-10-13-1	± 50 PPM
2.0 MHz	ECS-20-20-1	± 50 PPM
4.0 MHz	ECS-40-20-1	± 50 PPM
8.0 MHz	ECS ECS-80-S-4 ECS-80-18-1	± 50 PPM
16.0 MHz	ECS-160-20-1	TBD
25 MHz	CTS CTS25M	± 50 PPM
32 MHz	CRYSTEK HF-2	± 50 PPM

14.2.3 EXTERNAL CLOCK OSCILLATOR

In the EC oscillator mode, the OSC1 input can be driven by CMOS drivers. In this mode, the OSC1/CLKIN pin is hi-impedance and the OSC2/CLKOUT pin is the CLKOUT output (4 Tosc).

FIGURE 14-4: EXTERNAL CLOCK INPUT OPERATION (EC OSC CONFIGURATION)



ADDWFC		ADD WREG and Carry bit to f						
Syntax:	[<i>label</i>] ADDWFC f,d							
Operands:	0 ≤ f ≤ 255 d ∈ [0,1]							
Operation:	(WREG) + (f) + C → (dest)							
Status Affected:	OV, C, DC, Z							
Encoding:	<table border="1"><tr><td>0001</td><td>000d</td><td>ffff</td><td>ffff</td></tr></table>				0001	000d	ffff	ffff
0001	000d	ffff	ffff					
Description:	Add WREG, the Carry Flag and data memory location 'f'. If 'd' is 0, the result is placed in WREG. If 'd' is 1, the result is placed in data memory location 'f'.							
Words:	1							
Cycles:	1							
Q Cycle Activity:								
	Q1	Q2	Q3	Q4				
	Decode	Read register 'f'	Execute	Write to destination				

Example: ADDWFC REG 0

Before Instruction

Carry bit = 1
REG = 0x02
WREG = 0x4D

After Instruction

Carry bit = 0
REG = 0x02
WREG = 0x50

ANDLW		And Literal with WREG							
Syntax:	[<i>label</i>] ANDLW k								
Operands:	0 ≤ k ≤ 255								
Operation:	(WREG) .AND. (k) → (WREG)								
Status Affected:	Z								
Encoding:	<table border="1"><tr><td>1011</td><td>0101</td><td>kkkk</td><td>kkkk</td></tr></table>					1011	0101	kkkk	kkkk
1011	0101	kkkk	kkkk						
Description:	The contents of WREG are AND'd with the 8-bit literal 'k'. The result is placed in WREG.								
Words:	1								
Cycles:	1								
Q Cycle Activity:									
	Q1	Q2	Q3	Q4					
	Decode	Read literal 'k'	Execute	Write to WREG					

Example: ANDLW 0x5F

Before Instruction

WREG = 0xA3

After Instruction

WREG = 0x03

MOVPF Move p to f

Syntax: `[label] MOVPF p,f`

Operands: $0 \leq f \leq 255$
 $0 \leq p \leq 31$

Operation: $(p) \rightarrow (f)$

Status Affected: Z

Encoding:

010p	pppp	ffff	ffff
------	------	------	------

Description: Move data from data memory location 'p' to data memory location 'f'. Location 'f' can be anywhere in the 256 byte data space (00h to FFh) while 'p' can be 00h to 1Fh.

Either 'p' or 'f' can be WREG (a useful special situation).

MOVPF is particularly useful for transferring a peripheral register (e.g. the timer or an I/O port) to a data memory location. Both 'f' and 'p' can be indirectly addressed.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'p'	Execute	Write register 'f'

Example: `MOVPF REG1, REG2`

Before Instruction

REG1 = 0x11
 REG2 = 0x33

After Instruction

REG1 = 0x11
 REG2 = 0x11

MOVWF Move WREG to f

Syntax: `[label] MOVWF f`

Operands: $0 \leq f \leq 255$

Operation: $(WREG) \rightarrow (f)$

Status Affected: None

Encoding:

0000	0001	ffff	ffff
------	------	------	------

Description: Move data from WREG to register 'f'. Location 'f' can be anywhere in the 256 word data space.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example: `MOVWF REG`

Before Instruction

WREG = 0x4F
 REG = 0xFF

After Instruction

WREG = 0x4F
 REG = 0x4F

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 I²C 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.

TABLE 16-1: DEVELOPMENT TOOLS FROM MICROCHIP

Product	** MPLAB™ Integrated Development Environment	MPLAB™ C Compiler	MP-DriveWay Applications Code Generator	fuzzyTECH®-MP Explorer/Editor Fuzzy Logic Dev. Tool	*** PICMASTER®-CE In-Circuit Emulator	ICEPIC Low-Cost In-Circuit Emulator	****PRO MATE™ II Universal Microchip Programmer	PICSTART® Lite Ultra Low-Cost Dev. Kit	PICSTART® Plus Low-Cost Universal Dev. Kit
PIC12C508, 509	SW007002	SW006005	—	—	EM167015/ EM167101	—	DV007003	—	DV003001
PIC14000	SW007002	SW006005	—	—	EM147001/ EM147101	—	DV007003	—	DV003001
PIC16C52, 54, 54A, 55, 56, 57, 58A	SW007002	SW006005	SW006006	DV005001/ DV005002	EM167015/ EM167101	EM167201	DV007003	DV162003	DV003001
PIC16C554, 556, 558	SW007002	SW006005	—	DV005001/ DV005002	EM167033/ EM167113	—	DV007003	—	DV003001
PIC16C61	SW007002	SW006005	SW006006	DV005001/ DV005002	EM167021/ N/A	EM167205	DV007003	DV162003	DV003001
PIC16C62, 62A, 64, 64A	SW007002	SW006005	SW006006	DV005001/ DV005002	EM167025/ EM167103	EM167203	DV007003	DV162002	DV003001
PIC16C620, 621, 622	SW007002	SW006005	SW006006	DV005001/ DV005002	EM167023/ EM167109	EM167202	DV007003	DV162003	DV003001
PIC16C63, 65, 65A, 73, 73A, 74, 74A	SW007002	SW006005	SW006006	DV005001/ DV005002	EM167025/ EM167103	EM167204	DV007003	DV162002	DV003001
PIC16C642, 662*	SW007002	SW006005	—	—	EM167035/ EM167105	—	DV007003	DV162002	DV003001
PIC16C71	SW007002	SW006005	SW006006	DV005001/ DV005002	EM167027/ EM167105	EM167205	DV007003	DV162003	DV003001
PIC16C710, 711	SW007002	SW006005	SW006006	DV005001/ DV005002	EM167027/ EM167105	—	DV007003	DV162003	DV003001
PIC16C72	SW007002	SW006005	SW006006	—	EM167025/ EM167103	—	DV007003	DV162002	DV003001
PIC16F83	SW007002	SW006005	SW006006	DV005001/ DV005002	EM167029/ EM167107	—	DV007003	DV162003	DV003001
PIC16C84	SW007002	SW006005	SW006006	DV005001/ DV005002	EM167029/ EM167107	EM167206	DV007003	DV162003	DV003001
PIC16F84	SW007002	SW006005	SW006006	DV005001/ DV005002	EM167029/ EM167107	—	DV007003	DV162003	DV003001
PIC16C923, 924*	SW007002	SW006005	SW006006	DV005001/ DV005002	EM167031/ EM167111	—	DV007003	—	DV003001
PIC17C42, 42A, 43, 44	SW007002	SW006005	SW006006	DV005001/ DV005002	EM177007/ EM177107	—	DV007003	—	DV003001

*Contact Microchip Technology for availability date

**MPLAB Integrated Development Environment includes MPLAB-SIM Simulator and MPASM Assembler

***All PICMASTER and PICMASTER-CE ordering part numbers above include PRO MATE II programmer

****PRO MATE socket modules are ordered separately. See development systems ordering guide for specific ordering part numbers

Product	TRUEGAUGE® Development Kit	SEEVAL® Designers Kit	Hopping Code Security Programmer Kit	Hopping Code Security Eval/Demo Kit
All 2 wire and 3 wire Serial EEPROM's	N/A	DV243001	N/A	N/A
MTA11200B	DV114001	N/A	N/A	N/A
HCS200, 300, 301 *	N/A	N/A	PG306001	DM303001

PIC17C4X

Applicable Devices 42 R42 42A 43 R43 44

17.2 DC CHARACTERISTICS: PIC17C42-16 (Commercial, Industrial) PIC17C42-25 (Commercial, Industrial)

Standard Operating Conditions (unless otherwise stated)							
Operating temperature							
DC CHARACTERISTICS							
Operating voltage VDD range as described in Section 17.1							
Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
D030	VIL	Input Low Voltage					
D031		I/O ports					
D032		with TTL buffer	VSS	–	0.8	V	Note1
D033		with Schmitt Trigger buffer	VSS	–	0.2VDD	V	
D032		MCLR, OSC1 (in EC and RC mode)	VSS	–	0.2VDD	V	
D033		OSC1 (in XT, and LF mode)	–	0.5VDD	–	V	
D040	VIH	Input High Voltage					
D041		I/O ports					
D042		with TTL buffer	2.0	–	VDD	V	Note1
D043		with Schmitt Trigger buffer	0.8VDD	–	VDD	V	
D042		MCLR	0.8VDD	–	VDD	V	
D043		OSC1 (XT, and LF mode)	–	0.5VDD	–	V	
D050	VHYS	Hysteresis of Schmitt Trigger inputs	0.15VDD*	–	–	V	
D060	IIL	Input Leakage Current (Notes 2, 3)					
D060		I/O ports (except RA2, RA3)	–	–	±1	µA	VSS ≤ VPIN ≤ VDD, I/O Pin at hi-impedance PORTB weak pull-ups dis- abled
D061		MCLR	–	–	±2	µA	VPIN = VSS or VPIN = VDD
D062		RA2, RA3	–	–	±2	µA	VSS ≤ VRA2, VRA3 ≤ 12V
D063		OSC1, TEST	–	–	±1	µA	VSS ≤ VPIN ≤ VDD
D064		MCLR	–	–	10	µA	VMCLR = VPP = 12V (when not programming)
D070	IPURB	PORTB weak pull-up current	60	200	400	µA	VPIN = VSS, $\overline{\text{RBPU}} = 0$

* 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.

†† Design guidance to attain the AC timing specifications. These loads are not tested.

Note 1: In RC oscillator configuration, the OSC1 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 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 MCLR/Vpp pin may be kept in this range at times other than programming, but this is not recommended.

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

18.0 PIC17C42 DC AND AC CHARACTERISTICS

The graphs and tables provided in this section are for design guidance and are not tested or guaranteed. In some graphs or tables the data presented are 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 18-1: 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 18-1: TYPICAL RC OSCILLATOR FREQUENCY vs. TEMPERATURE

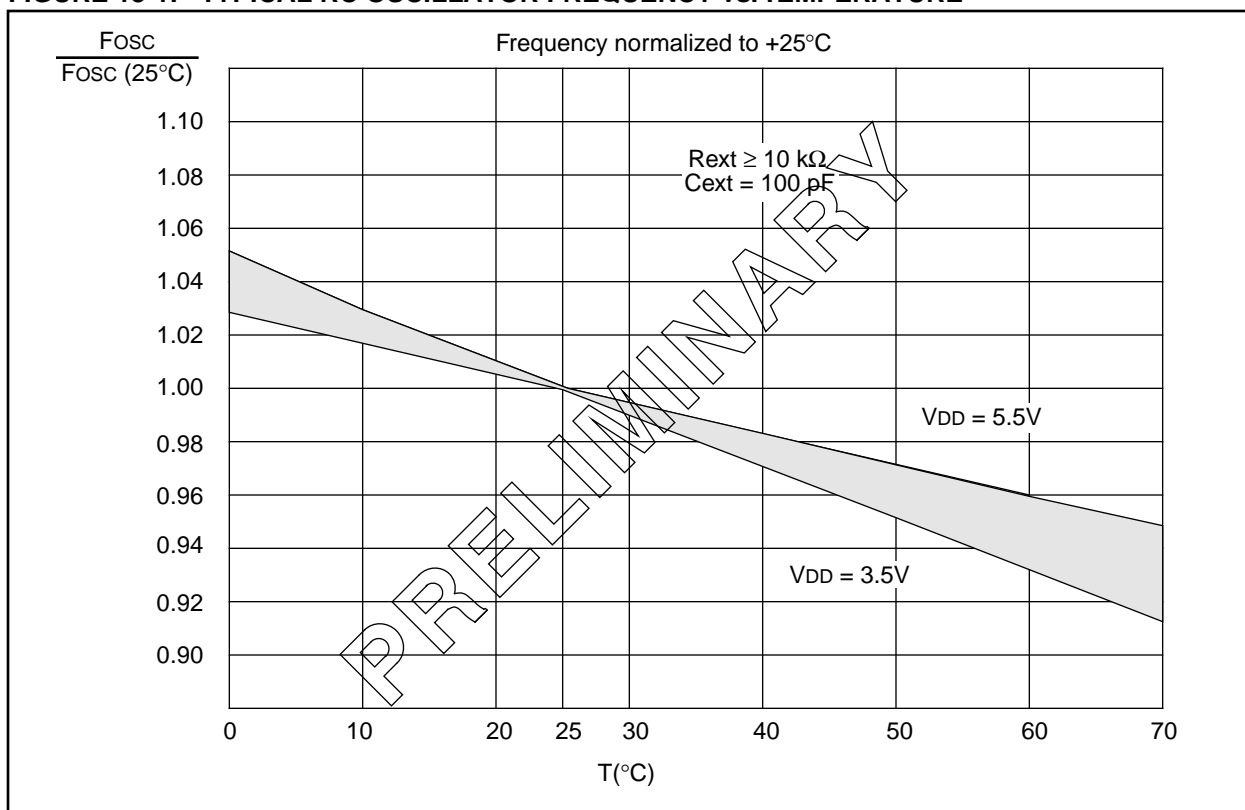


FIGURE 19-12: MEMORY INTERFACE READ TIMING (NOT SUPPORTED IN PIC17LC4X DEVICES)

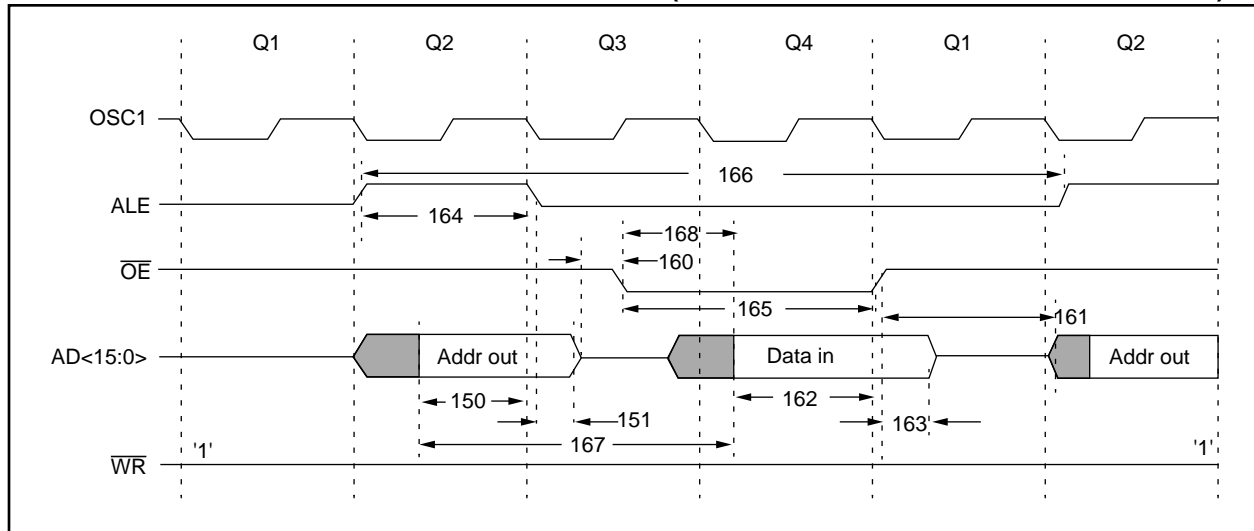


TABLE 19-12: MEMORY INTERFACE READ REQUIREMENTS (NOT SUPPORTED IN PIC17LC4X DEVICES)

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
150	TadV2aIL	AD15:AD0 (address) valid to ALE↓ (address setup time)	0.25Tcy - 10	—	—	ns	
151	TalL2adI	ALE↓ to address out invalid (address hold time)	5*	—	—	ns	
160	TadZ2oeL	AD15:AD0 hi-impedance to \overline{OE} ↓	0*	—	—	ns	
161	ToeH2adD	\overline{OE} ↑ to AD15:AD0 driven	0.25Tcy - 15	—	—	ns	
162	TadV2oeH	Data in valid before \overline{OE} ↑ (data setup time)	35	—	—	ns	
163	ToeH2adI	\overline{OE} ↑ to data in invalid (data hold time)	0	—	—	ns	
164	TalH	ALE pulse width	—	0.25Tcy §	—	ns	
165	ToeL	\overline{OE} pulse width	0.5Tcy - 35 §	—	—	ns	
166	TalH2alH	ALE↑ to ALE↑ (cycle time)	—	Tcy §	—	ns	
167	Tacc	Address access time	—	—	0.75Tcy - 30	ns	
168	Toe	Output enable access time (\overline{OE} low to Data Valid)	—	—	0.5Tcy - 45	ns	

* These parameters are characterized but not tested.

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

§ This specification ensured by design.

PIC17C4X

Applicable Devices 42 R42 42A 43 R43 44

FIGURE 20-2: TYPICAL RC OSCILLATOR FREQUENCY vs. VDD

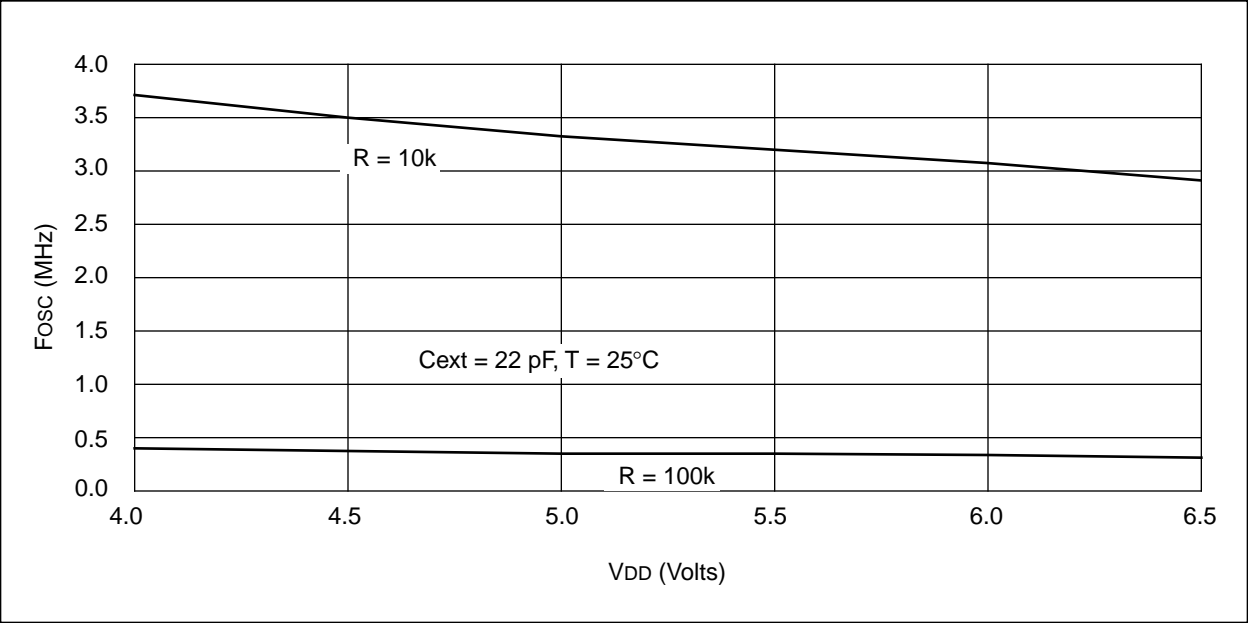
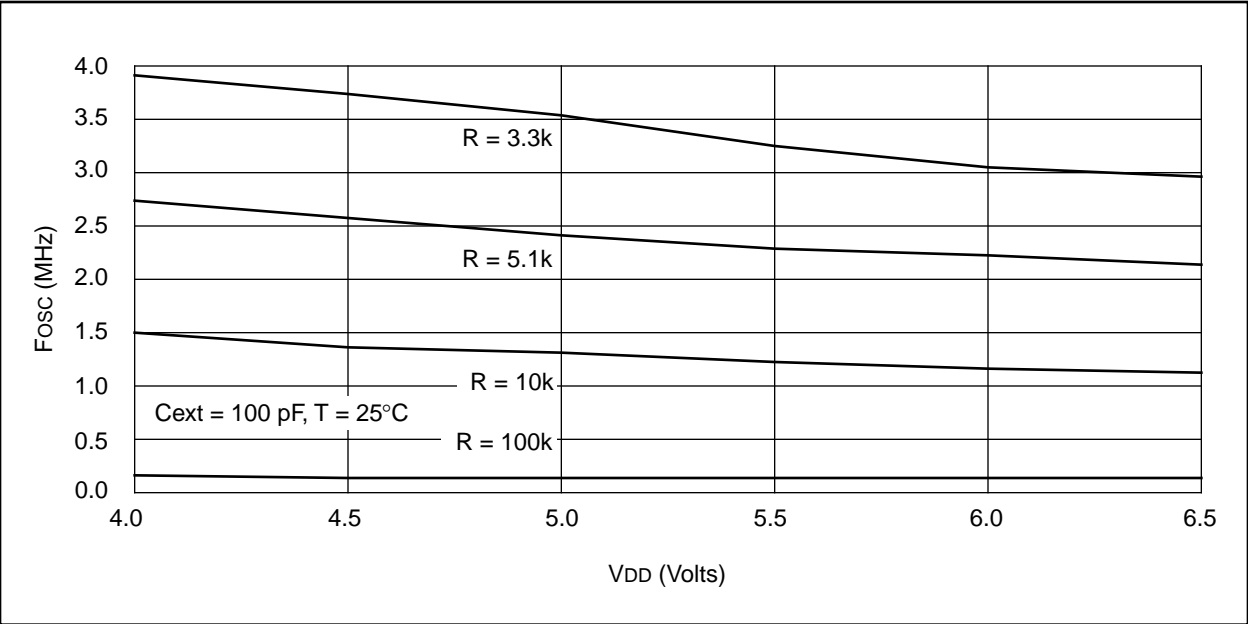
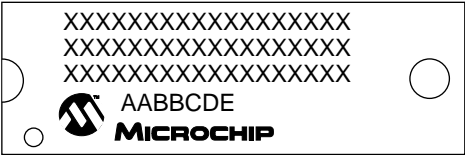


FIGURE 20-3: TYPICAL RC OSCILLATOR FREQUENCY vs. VDD



21.6 Package Marking Information

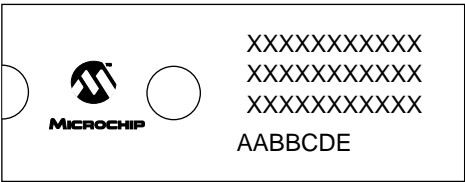
40-Lead PDIP/CERDIP



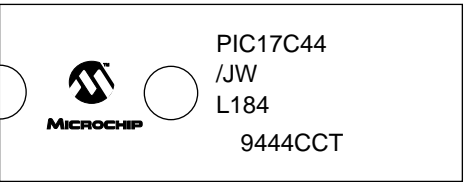
Example



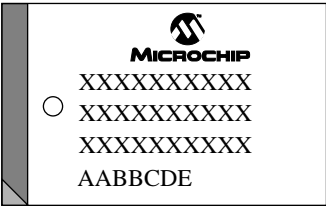
40 Lead CERDIP Windowed



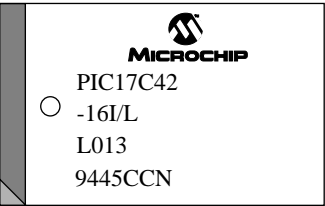
Example



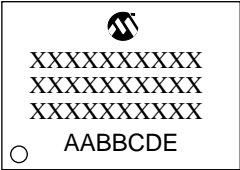
44-Lead PLCC



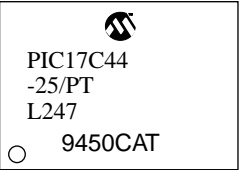
Example



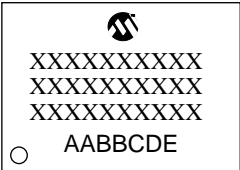
44-Lead MQFP



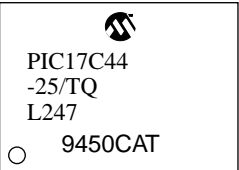
Example



44-Lead TQFP



Example



Legend:	MM...M	Microchip part number information
	XX...X	Customer specific information*
	AA	Year code (last 2 digits of calendar year)
	BB	Week code (week of January 1 is week '01')
	C	Facility code of the plant at which wafer is manufactured
		C = Chandler, Arizona, U.S.A.,
		S = Tempe, Arizona, U.S.A.
	D	Mask revision number
	E	Assembly code of the plant or country of origin in which part was assembled
Note:	In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line thus limiting the number of available characters for customer specific information.	

* Standard OTP marking consists of Microchip part number, year code, week code, facility code, mask rev#, and assembly code. For OTP marking beyond this, certain price adders apply. Please check with your Microchip Sales Office. For QTP devices, any special marking adders are included in QTP price.