

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](#)"

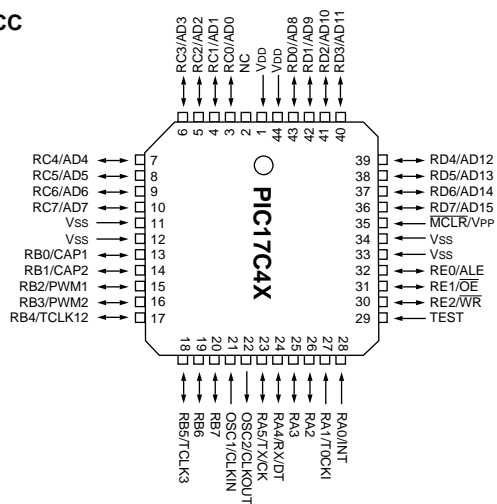
#### Details

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-QFP
Supplier Device Package	44-MQFP (10x10)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/microchip-technology/pic17c43t-25i-pq">https://www.e-xfl.com/product-detail/microchip-technology/pic17c43t-25i-pq</a>

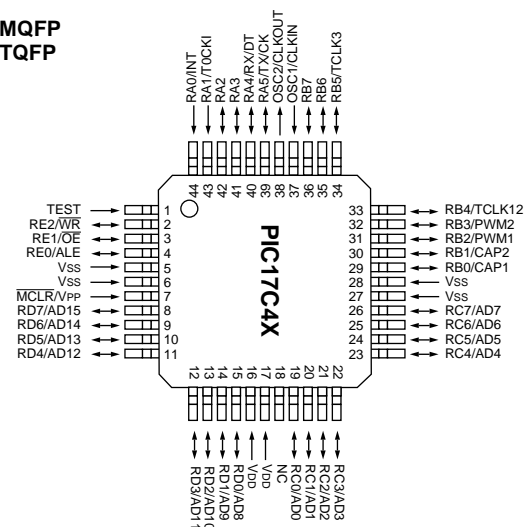
# PIC17C4X

## Pin Diagrams Cont'd

PLCC



MQFP  
TQFP



All devices are available in all package types, listed in Section 21.0, with the following exceptions:

- ROM devices are not available in Windowed Cerdip Packages
- TQFP is not available for the PIC17C42.

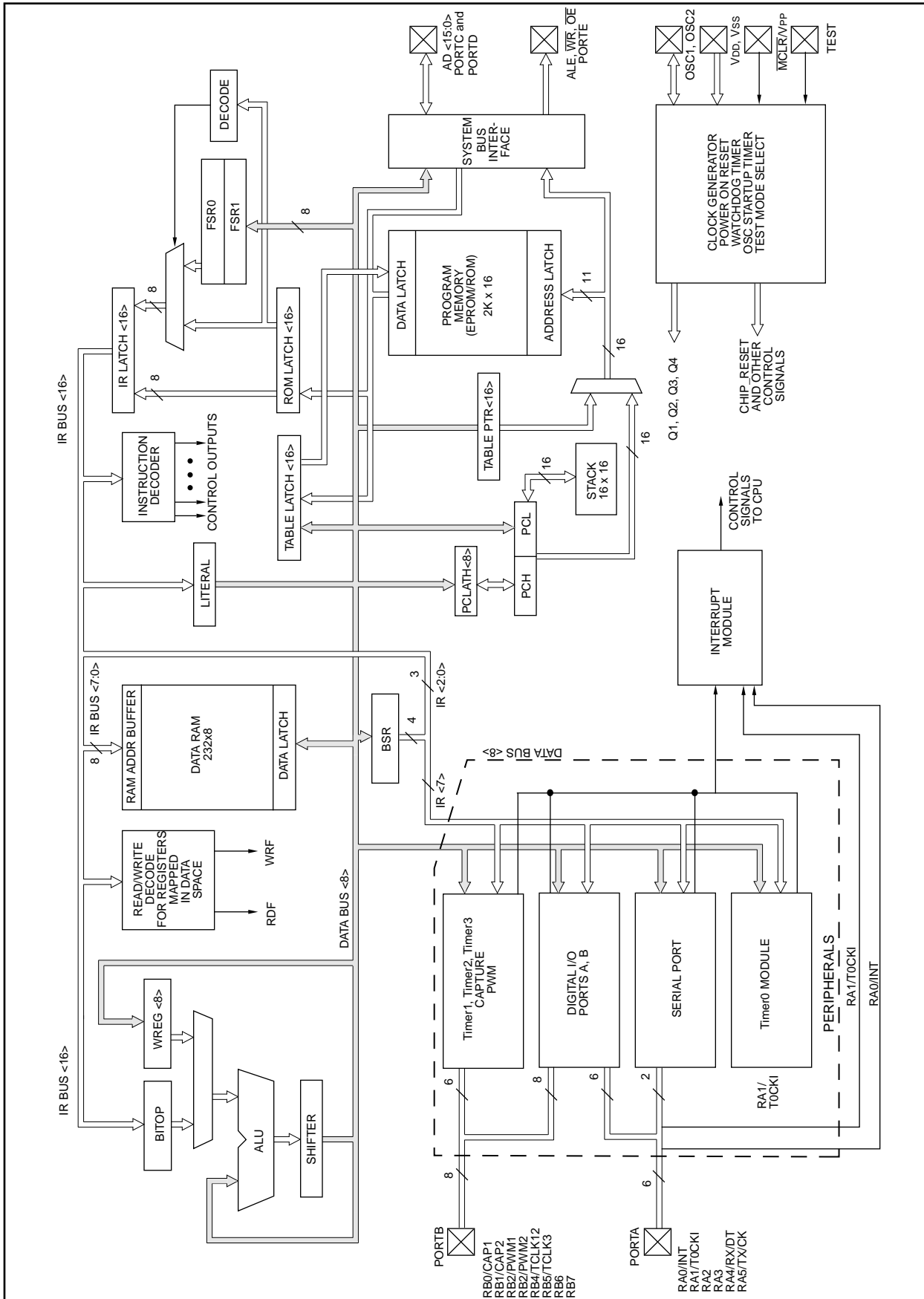
# PIC17C4X

---

NOTES:

# PIC17C4X

FIGURE 3-1: PIC17C42 BLOCK DIAGRAM



## 5.1 Interrupt Status Register (INTSTA)

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

## 6.8 Bank Select Register (BSR)

The BSR is used to switch between banks in the data memory area (Figure 6-13). In the PIC17C42, PIC17CR42, and PIC17C42A only the lower nibble is implemented. While in the PIC17C43, PIC17CR43, and PIC17C44 devices, the entire byte is implemented. The lower nibble is used to select the peripheral register bank. The upper nibble is used to select the general purpose memory bank.

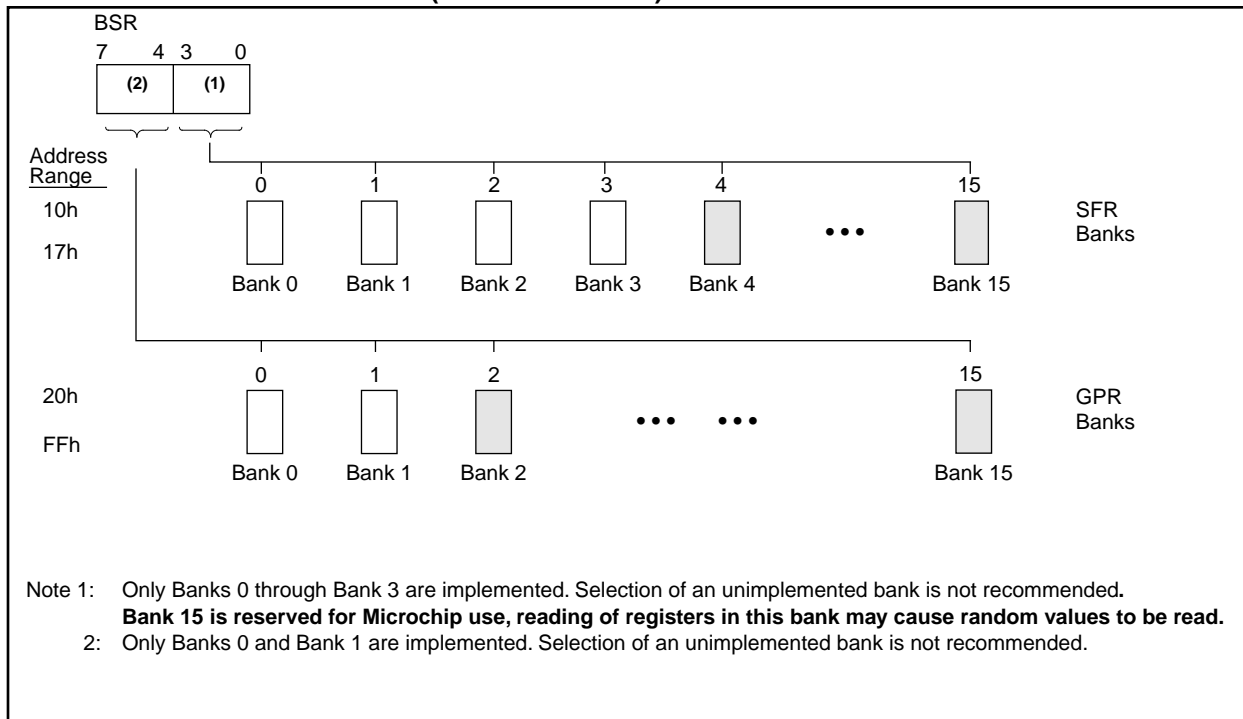
All the Special Function Registers (SFRs) are mapped into the data memory space. In order to accommodate the large number of registers, a banking scheme has been used. A segment of the SFRs, from address 10h to address 17h, is banked. The lower nibble of the bank select register (BSR) selects the currently active "peripheral bank." Effort has been made to group the peripheral registers of related functionality in one bank. However, it will still be necessary to switch from bank to bank in order to address all peripherals related to a single task. To assist this, a `MOVLB` bank instruction is in the instruction set.

For the PIC17C43, PIC17CR43, and PIC17C44 devices, the need for a large general purpose memory space dictated a general purpose RAM banking scheme. The upper nibble of the BSR selects the currently active general purpose RAM bank. To assist this, a `MOVLB` bank instruction has been provided in the instruction set.

If the currently selected bank is not implemented (such as Bank 13), any read will read all '0's. Any write is completed to the bit bucket and the ALU status bits will be set/cleared as appropriate.

**Note:** Registers in Bank 15 in the Special Function Register area, are reserved for Microchip use. Reading of registers in this bank may cause random values to be read.

**FIGURE 6-13: BSR OPERATION (PIC17C43/R43/44)**



## 7.1 Table Writes to Internal Memory

A table write operation to internal memory causes a long write operation. The long write is necessary for programming the internal EPROM. Instruction execution is halted while in a long write cycle. The long write will be terminated by any enabled interrupt. To ensure that the EPROM location has been well programmed, a minimum programming time is required (see specification #D114 ). Having only one interrupt enabled to terminate the long write ensures that no unintentional interrupts will prematurely terminate the long write.

The sequence of events for programming an internal program memory location should be:

1. Disable all interrupt sources, except the source to terminate EPROM program write.
2. Raise  $\overline{\text{MCLR}}$ /VPP pin to the programming voltage.
3. Clear the WDT.
4. Do the table write. The interrupt will terminate the long write.
5. Verify the memory location (table read).

**Note:** Programming requirements must be met. See timing specification in electrical specifications for the desired device. Violating these specifications (including temperature) may result in EPROM locations that are not fully programmed and may lose their state over time.

### 7.1.1 TERMINATING LONG WRITES

An interrupt source or reset are the only events that terminate a long write operation. Terminating the long write from an interrupt source requires that the interrupt enable and flag bits are set. The GLINTD bit only enables the vectoring to the interrupt address.

If the T0CKI, RA0/INT, or TMR0 interrupt source is used to terminate the long write; the interrupt flag, of the highest priority enabled interrupt, will terminate the long write and automatically be cleared.

**Note 1:** If an interrupt is pending, the TABLWT is aborted (an NOP is executed). The highest priority pending interrupt, from the T0CKI, RA0/INT, or TMR0 sources that is enabled, has its flag cleared.

**Note 2:** If the interrupt is not being used for the program write timing, the interrupt should be disabled. This will ensure that the interrupt is not lost, nor will it terminate the long write prematurely.

If a peripheral interrupt source is used to terminate the long write, the interrupt enable and flag bits must be set. The interrupt flag will not be automatically cleared upon the vectoring to the interrupt vector address.

If the GLINTD bit is cleared prior to the long write, when the long write is terminated, the program will branch to the interrupt vector.

If the GLINTD bit is set prior to the long write, when the long write is terminated, the program will not vector to the interrupt address.

**TABLE 7-1: INTERRUPT - TABLE WRITE INTERACTION**

Interrupt Source	GLINTD	Enable Bit	Flag Bit	Action
RA0/INT, TMR0, T0CKI	0	1	1	Terminate long table write (to internal program memory), branch to interrupt vector (branch clears flag bit).
	0	1	0	None
	1	0	x	None
	1	1	1	Terminate table write, do not branch to interrupt vector (flag is automatically cleared).
Peripheral	0	1	1	Terminate table write, branch to interrupt vector.
	0	1	0	None
	1	0	x	None
	1	1	1	Terminate table write, do not branch to interrupt vector (flag is set).

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         ;

```

## 9.3 PORTC and DDRC Registers

PORTC is an 8-bit bi-directional port. The corresponding data direction register is DDRC. A '1' in DDRC configures the corresponding port pin as an input. A '0' in the DDRC register configures the corresponding port pin as an output. Reading PORTC reads the status of the pins, whereas writing to it will write to the port latch. PORTC is multiplexed with the system bus. When operating as the system bus, PORTC is the low order byte of the address/data bus (AD7:AD0). The timing for the system bus is shown in the Electrical Characteristics section.

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

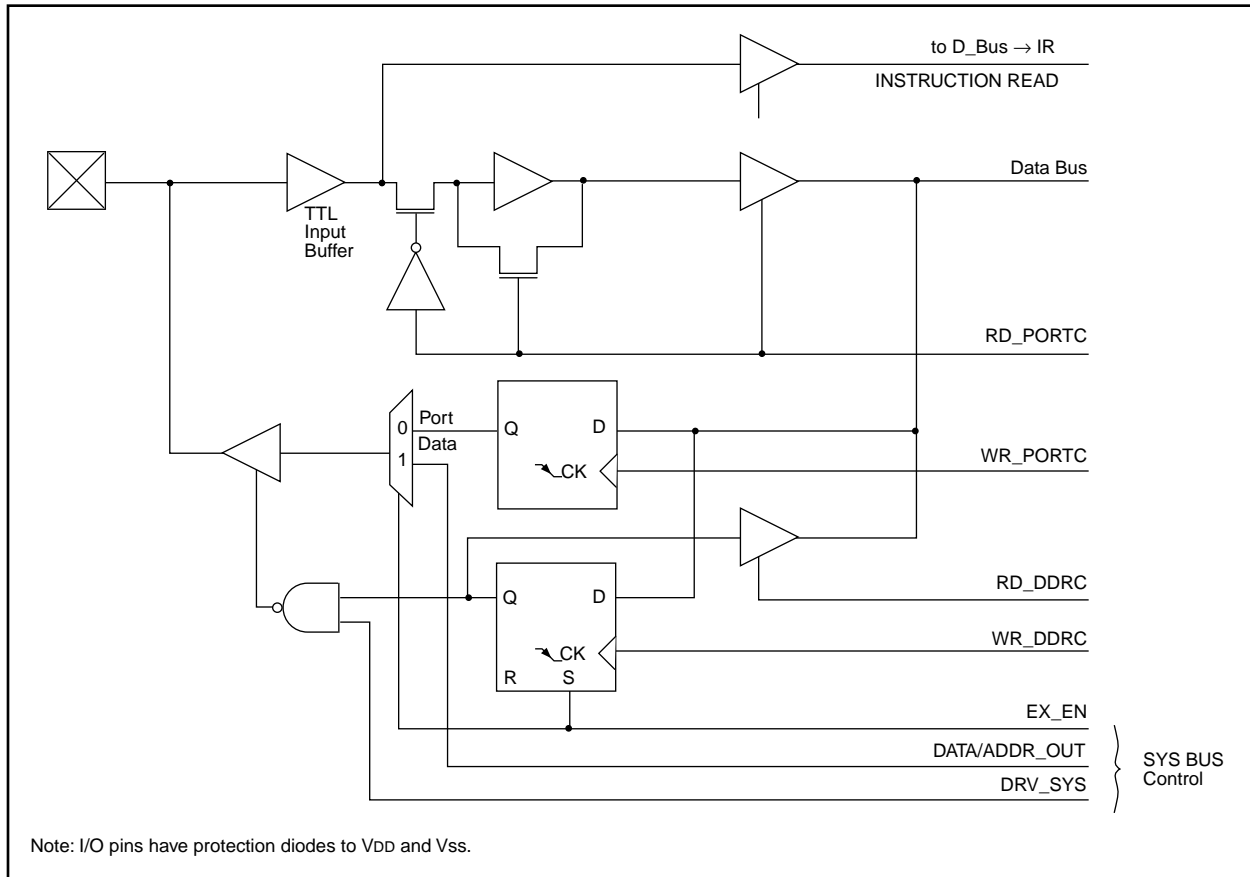
Example 9-2 shows the instruction sequence to initialize PORTC. The Bank Select Register (BSR) must be selected to Bank 1 for the port to be initialized.

### EXAMPLE 9-2: INITIALIZING PORTC

```

MOVLB 1           ; Select Bank 1
CLRWF PORTC       ; Initialize PORTC data
                  ; latches before setting
                  ; the data direction
                  ; register
MOVLW 0xCF        ; Value used to initialize
                  ; data direction
MOVWF DDRC        ; Set RC<3:0> as inputs
                  ; RC<5:4> as outputs
                  ; RC<7:6> as inputs
    
```

FIGURE 9-6: BLOCK DIAGRAM OF RC<7:0> PORT PINS



## 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 (FPWM) 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 (FPWM).

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 TCY (one instruction cycle). This will cause jitter in the duty cycle as well as the period of the PWM output.

This jitter will be  $\pm TCY$ , 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 (Fosc).

## 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.

MOVLR		Move Literal to high nibble in BSR							
Syntax:	[ <i>label</i> ] MOVLR k								
Operands:	0 ≤ k ≤ 15								
Operation:	k → (BSR<7:4>)								
Status Affected:	None								
Encoding:	<table><tr><td>1011</td><td>101x</td><td>kkkk</td><td>uuuu</td></tr></table>					1011	101x	kkkk	uuuu
1011	101x	kkkk	uuuu						
Description:	The 4-bit literal 'k' is loaded into the most significant 4-bits of the Bank Select Register (BSR). Only the high 4-bits of the Bank Select Register are affected. The lower half of the BSR is unchanged. The assembler will encode the "u" fields as 0.								
Words:	1								
Cycles:	1								
Q Cycle Activity:									
	Q1	Q2	Q3	Q4					
	Decode	Read literal 'k:u'	Execute	Write literal 'k' to BSR<7:4>					

**Example:** MOVLR 5

Before Instruction

BSR register = 0x22

After Instruction

BSR register = 0x52

**Note:** This instruction is not available in the PIC17C42 device.

MOVLW		Move Literal to WREG						
Syntax:	[ <i>label</i> ] MOVLW k							
Operands:	$0 \leq k \leq 255$							
Operation:	$k \rightarrow (\text{WREG})$							
Status Affected:	None							
Encoding:	<table border="1"><tr><td>1011</td><td>0000</td><td>kkkk</td><td>kkkk</td></tr></table>				1011	0000	kkkk	kkkk
1011	0000	kkkk	kkkk					
Description:	The eight bit literal 'k' is loaded into WREG.							
Words:	1							
Cycles:	1							
Q Cycle Activity:								
	Q1	Q2	Q3	Q4				
	Decode	Read literal 'k'	Execute	Write to WREG				

**Example:** MOVLW 0x5A

After Instruction

WREG = 0x5A

# PIC17C4X

Applicable Devices 42 R42 42A 43 R43 44

**TABLE 17-1: CROSS REFERENCE OF DEVICE SPECS FOR OSCILLATOR CONFIGURATIONS AND FREQUENCIES OF OPERATION (COMMERCIAL DEVICES)**

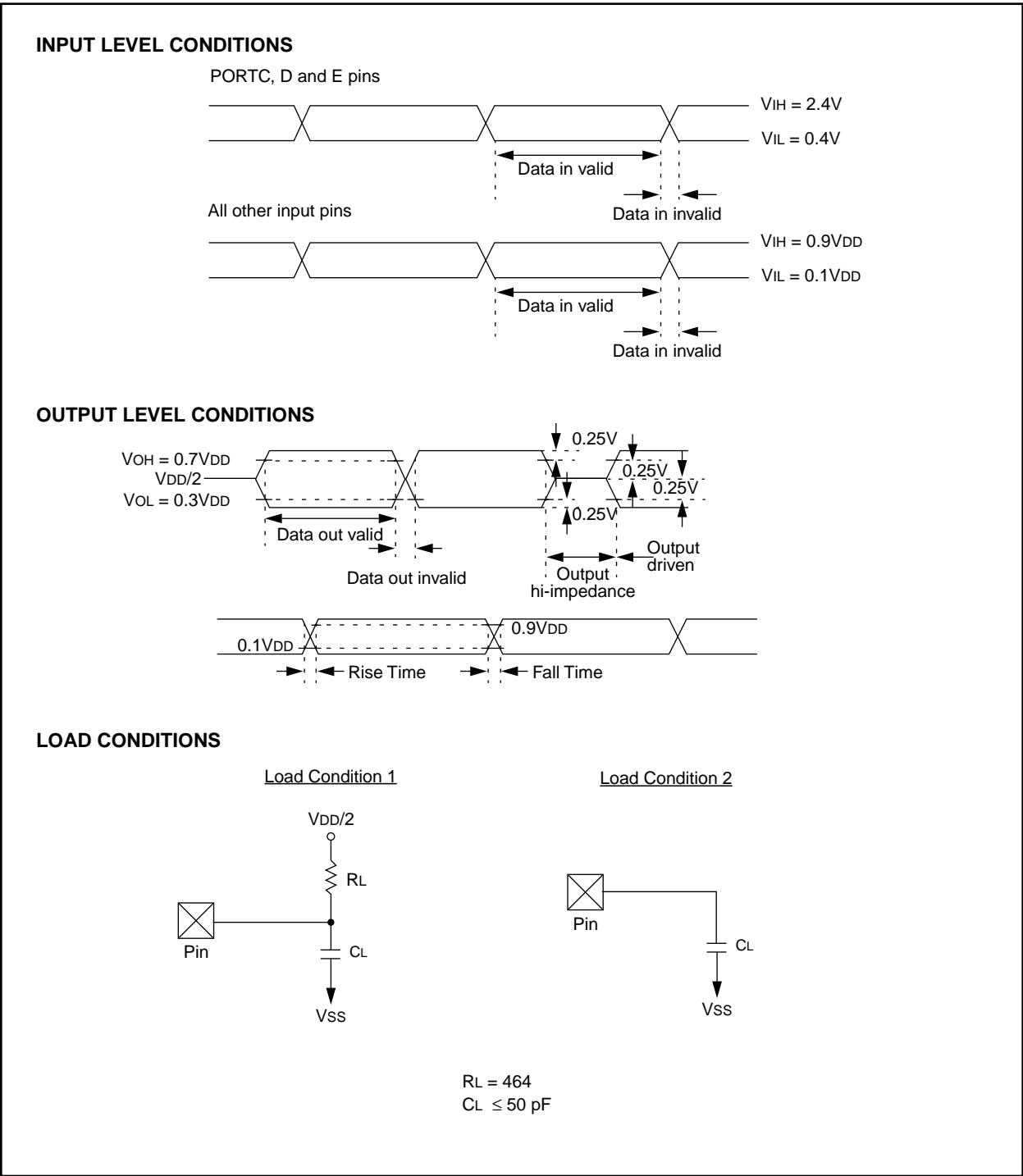
OSC	PIC17C42-16	PIC17C42-25
RC	VDD: 4.5V to 5.5V IDD: 6 mA max. IPD: 5 $\mu$ A max. at 5.5V (WDT disabled) Freq: 4 MHz max.	VDD: 4.5V to 5.5V IDD: 6 mA max. IPD: 5 $\mu$ A max. at 5.5V (WDT disabled) Freq: 4 MHz max.
XT	VDD: 4.5V to 5.5V IDD: 24 mA max. IPD: 5 $\mu$ A max. at 5.5V (WDT disabled) Freq: 16 MHz max.	VDD: 4.5V to 5.5V IDD: 38 mA max. IPD: 5 $\mu$ A max. at 5.5V (WDT disabled) Freq: 25 MHz max.
EC	VDD: 4.5V to 5.5V IDD: 24 mA max. IPD: 5 $\mu$ A max. at 5.5V (WDT disabled) Freq: 16 MHz max.	VDD: 4.5V to 5.5V IDD: 38 mA max. IPD: 5 $\mu$ A max. at 5.5V (WDT disabled) Freq: 25 MHz max.
LF	VDD: 4.5V to 5.5V IDD: 150 $\mu$ A max. at 32 kHz (WDT enabled) IPD: 5 $\mu$ A max. at 5.5V (WDT disabled) Freq: 2 MHz max.	VDD: 4.5V to 5.5V IDD: 150 $\mu$ A max. at 32 kHz (WDT enabled) IPD: 5 $\mu$ A max. at 5.5V (WDT disabled) Freq: 2 MHz max.

# PIC17C4X

Applicable Devices 42 R42 42A 43 R43 44

FIGURE 17-1: PARAMETER MEASUREMENT INFORMATION

All timings are measure between high and low measurement points as indicated in the figures below.



# PIC17C4X

Applicable Devices 42 R42 42A 43 R43 44

## 19.2 DC CHARACTERISTICS: PIC17LC42A/43/LC44 (Commercial, Industrial) PIC17LCR42/43 (Commercial, Industrial)

DC CHARACTERISTICS		Standard Operating Conditions (unless otherwise stated)					
		Operating temperature					
							-40°C ≤ TA ≤ +85°C for industrial and 0°C ≤ TA ≤ +70°C for commercial
Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
D001	VDD	Supply Voltage	2.5	–	6.0	V	
D002	VDR	RAM Data Retention Voltage (Note 1)	1.5 *	–	–	V	Device in SLEEP mode
D003	VPOR	VDD start voltage to ensure internal Power-on Reset signal	–	VSS	–	V	See section on Power-on Reset for details
D004	SVDD	VDD rise rate to ensure internal Power-on Reset signal	0.060 *	–	–	mV/ms	See section on Power-on Reset for details
D010 D011 D014	IDD	Supply Current (Note 2)	– – –	3 6 95	6 12 * 150	mA mA μA	FOSC = 4 MHz (Note 4) FOSC = 8 MHz FOSC = 32 kHz, WDT disabled (EC osc configuration)
D020 D021	IPD	Power-down Current (Note 3)	– –	10 < 1	40 5	μA μA	VDD = 5.5V, WDT enabled VDD = 5.5V, WDT disabled

\* These parameters are characterized but not tested.

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

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

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

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

OSC1=external square wave, from rail to rail; all I/O pins tristated, pulled to VDD or VSS, T0CKI = VDD, MCLR = VDD; WDT enabled/disabled as specified.

Current consumed from the oscillator and I/O's driving external capacitive or resistive loads needs to be considered.

For the RC oscillator, the current through the external pull-up resistor (R) can be estimated as:  $V_{DD} / (2 \cdot R)$ .

For capacitive loads, the current can be estimated (for an individual I/O pin) as  $(C_L \cdot V_{DD}) \cdot f$

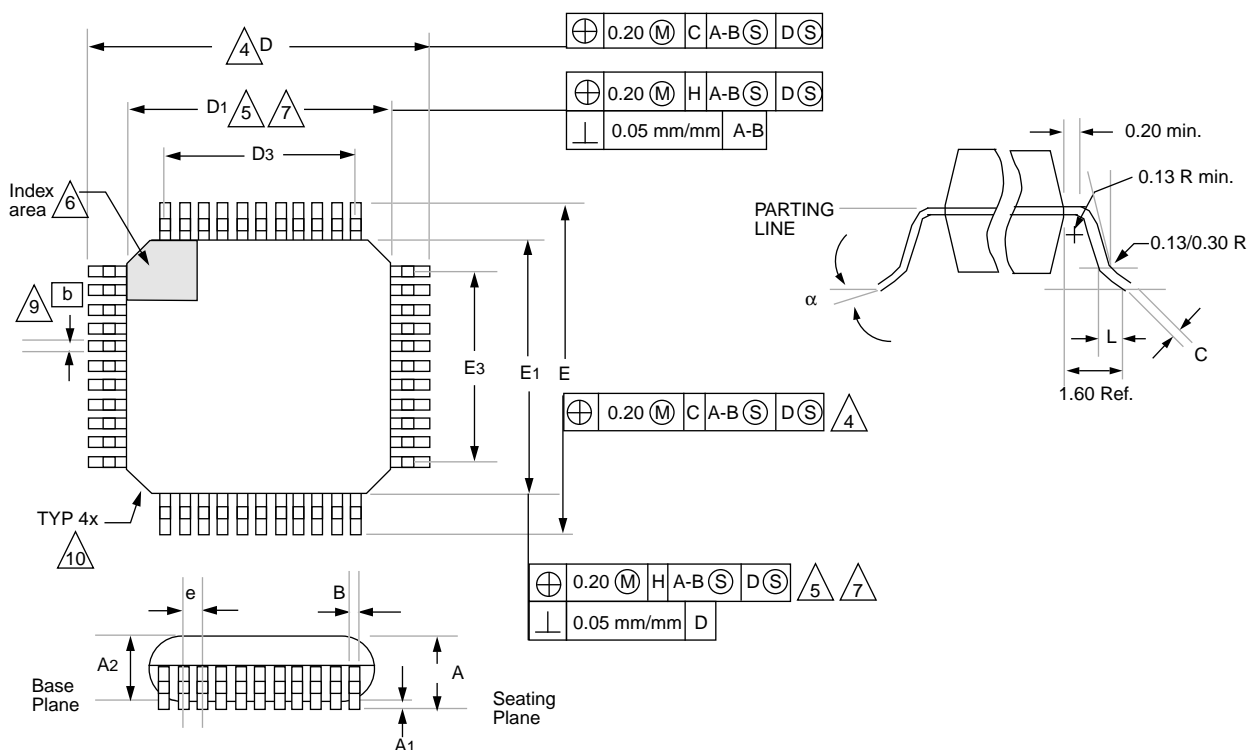
$C_L$  = Total capacitive load on the I/O pin;  $f$  = average frequency the I/O pin switches.

The capacitive currents are most significant when the device is configured for external execution (includes extended microcontroller mode).

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

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

## 21.4 44-Lead Plastic Surface Mount (MQFP 10x10 mm Body 1.6/0.15 mm Lead Form)



Package Group: Plastic MQFP						
Symbol	Millimeters			Inches		
	Min	Max	Notes	Min	Max	Notes
$\alpha$	0°	7°		0°	7°	
A	2.000	2.350		0.078	0.093	
A1	0.050	0.250		0.002	0.010	
A2	1.950	2.100		0.768	0.083	
b	0.300	0.450	Typical	0.011	0.018	Typical
C	0.150	0.180		0.006	0.007	
D	12.950	13.450		0.510	0.530	
D1	9.900	10.100		0.390	0.398	
D3	8.000	8.000	Reference	0.315	0.315	Reference
E	12.950	13.450		0.510	0.530	
E1	9.900	10.100		0.390	0.398	
E3	8.000	8.000	Reference	0.315	0.315	Reference
e	0.800	0.800		0.031	0.032	
L	0.730	1.030		0.028	0.041	
N	44	44		44	44	
CP	0.102	—		0.004	—	

# PIC17C4X

Delay From External Clock Edge .....	68
Development Support .....	143
Development Tools .....	143
Device Drawings	
44-Lead Plastic Surface Mount (MQFP	
10x10 mm Body 1.6/0.15 mm Lead Form) .....	209
DIGIT BORROW .....	9
Digit Carry (DC) .....	9
Duty Cycle .....	75

## E

### Electrical Characteristics

PIC17C42	
Absolute Maximum Ratings .....	147
Capture Timing .....	159
CLKOUT and I/O Timing .....	156
DC Characteristics .....	149
External Clock Timing .....	155
Memory Interface Read Timing .....	162
Memory Interface Write Timing .....	161
PWM Timing .....	159
RESET, Watchdog Timer, Oscillator Start-up	
Timer and Power-up Timer .....	157
Timer0 Clock Timings .....	158
Timer1, Timer2 and Timer3 Clock Timing .....	158
USART Module, Synchronous Receive .....	160
USART Module, Synchronous Transmission .....	160
PIC17C43/44	
Absolute Maximum Ratings .....	175
Capture Timing .....	188
CLKOUT and I/O Timing .....	185
DC Characteristics .....	177
External Clock Timing .....	184
Memory Interface Read Timing .....	191
Memory Interface Write Timing .....	190
Parameter Measurement Information .....	183
RESET, Watchdog Timer, Oscillator Start-up	
Timer and Power-up Timer Timing .....	186
Timer0 Clock Timing .....	187
Timer1, Timer2 and Timer3 Clock Timing .....	187
Timing Parameter Symbolology .....	182
USART Module Synchronous Receive	
Timing .....	189
USART Module Synchronous Transmission	
Timing .....	189
EPROM Memory Access Time Order Suffix .....	31
Extended Microcontroller .....	29
Extended Microcontroller Mode .....	31
External Memory Interface .....	31
External Program Memory Waveforms .....	31

## F

Family of Devices .....	6
PIC14000 .....	213
PIC16C5X .....	214
PIC16CXXX .....	215
PIC16C6X .....	216
PIC16C7X .....	217
PIC16C8X .....	218
PIC16C9XX .....	219
PIC17CXX .....	220
FERR .....	84, 91
FOSC0 .....	99

FOSC1 .....	99
FS0 .....	36
FS1 .....	36
FS2 .....	36
FS3 .....	36
FSR0 .....	34, 40
FSR1 .....	34, 40
Fuzzy Logic Dev. System ( <i>fuzzyTECH</i> <sup>®</sup> -MP) .....	143, 145

## G

General Format for Instructions .....	108
General Purpose RAM .....	29
General Purpose RAM Bank .....	42
General Purpose Register (GPR) .....	32
GLINTD .....	25, 37, 78, 105
GOTO .....	122
GPR (General Purpose Register) .....	32
Graphs	
IOH vs. VOH, VDD = 3V .....	170, 200
IOH vs. VOH, VDD = 5V .....	171, 201
IOL vs. VOL, VDD = 3V .....	171, 201
IOL vs. VOL, VDD = 5V .....	172, 202
Maximum IDD vs. Frequency	
(External Clock 125°C to -40°C) .....	167, 197
Maximum IPD vs. VDD Watchdog Disabled .....	168, 198
Maximum IPD vs. VDD Watchdog Enabled .....	169, 199
RC Oscillator Frequency vs.	
VDD (Cext = 100 pF) .....	164, 194
RC Oscillator Frequency vs.	
VDD (Cext = 22 pF) .....	164, 194
RC Oscillator Frequency vs.	
VDD (Cext = 300 pF) .....	165, 195
Transconductance of LF Oscillator vs. VDD .....	166, 196
Transconductance of XT Oscillator vs. VDD .....	166, 196
Typical IDD vs. Frequency	
(External Clock 25°C) .....	167, 197
Typical IPD vs. VDD Watchdog Disabled 25°C .....	168, 198
Typical IPD vs. VDD Watchdog Enabled 25°C .....	169, 199
Typical RC Oscillator vs. Temperature .....	163, 193
VTH (Input Threshold Voltage) of I/O Pins vs.	
VDD .....	172, 202
VTH (Input Threshold Voltage) of OSC1 Input	
(In XT, HS, and LP Modes) vs. VDD .....	173, 203
VTH, VIL of MCLR, T0CKI and OSC1	
(In RC Mode) vs. VDD .....	173, 203
WDT Timer Time-Out Period vs. VDD .....	170, 200

## H

Hardware Multiplier .....	49
---------------------------	----

## I

I/O Ports	
Bi-directional .....	64
I/O Ports .....	53
Programming Considerations .....	64
Read-Modify-Write Instructions .....	64
Successive Operations .....	64
INCF .....	123
INCFSNZ .....	124
INCFSZ .....	123
INDF0 .....	34, 40
INDF1 .....	34, 40

# PIC17C4X

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

## N

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

## O

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

## P

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

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

## R

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

Receive Status and Control Register .....	83
Register File Map .....	33
Registers	
ALUSTA .....	27, 36
BRG .....	86
BSR .....	27
CPUSTA .....	37
File Map .....	33
FSR0 .....	40
FSR1 .....	40
INDF0 .....	40
INDF1 .....	40
INTSTA .....	22
PIE .....	23
PIR .....	24
RCSTA .....	84
Special Function Table .....	34
T0STA .....	38, 67
TCON1 .....	71
TCON2 .....	72
TMR1 .....	81
TMR2 .....	81
TMR3 .....	81
TXSTA .....	83
WREG .....	27
Reset	
Section .....	15
Status Bits and Their Significance .....	16
Time-Out in Various Situations .....	16
Time-Out Sequence .....	16
RETFIE .....	131
RETLW .....	131
RETURN .....	132
RLCF .....	132
RLNCF .....	133
RRCF .....	133
RRNCF .....	134
RX Pin Sampling Scheme .....	91
RX9 .....	84
RX9D .....	84

## S

Sampling .....	91
Saving STATUS and WREG in RAM .....	27
SETF .....	134
SFR .....	108
SFR (Special Function Registers) .....	29, 32
SFR As Source/Destination .....	108
Signed Math .....	9
SLEEP .....	99, 105, 135
Software Simulator (MPSIM) .....	145
SPBRG .....	19, 34, 92, 96, 98
Special Features of the CPU .....	99
Special Function Registers .....	29, 32, 34, 108
SPEN .....	84
SREN .....	84
Stack	
Operation .....	39
Pointer .....	39
Stack .....	29
STKAL .....	39
STKAV .....	37
SUBLW .....	135
SUBWF .....	136
SUBWFB .....	136

SWAPF .....	137
SYNC .....	83
Synchronous Master Mode .....	93
Synchronous Master Reception .....	95
Synchronous Master Transmission .....	93
Synchronous Slave Mode .....	97

## T

T0CKI Pin .....	26
T0CKIE .....	22
T0CKIF .....	22
T0CS .....	38, 67
T0IE .....	22
T0IF .....	22
T0SE .....	38, 67
T0STA .....	34, 38
T16 .....	71
Table Latch .....	40
Table Pointer .....	40
Table Read	
Example .....	48
Section .....	43
Table Reads Section .....	48
TABLRD Operation .....	44
Timing .....	48
TLRD .....	48
TLRD Operation .....	44
Table Write	
Code .....	46
Interaction .....	45
Section .....	43
TABLWT Operation .....	43
Terminating Long Writes .....	45
Timing .....	46
TLWT Operation .....	43
To External Memory .....	46
To Internal Memory .....	45
TABLRD .....	44, 137, 138
TABLWT .....	43, 138, 139
TBLATH .....	40
TBLATL .....	40
TBLPTRH .....	34, 40
TBLPTRL .....	34, 40
TCLK12 .....	71
TCLK3 .....	71
TCON1 .....	20, 35
TCON2 .....	20, 35
Terminating Long Writes .....	45
Time-Out Sequence .....	16
Timer Resources .....	65
Timer0 .....	67
Timer1	
16-bit Mode .....	74
Clock Source Select .....	71
On bit .....	72
Section .....	71, 73
Timer2	
16-bit Mode .....	74
Clock Source Select .....	71
On bit .....	72
Section .....	71, 73
Timer3	
Clock Source Select .....	71
On bit .....	72
Section .....	71, 77

Figure 6-12:	Program Counter using The CALL and GOTO Instructions.....	41	Figure 14-3:	Crystal Operation, Overtone Crystals (XT OSC Configuration) .....	101
Figure 6-13:	BSR Operation (PIC17C43/R43/44) .....	42	Figure 14-4:	External Clock Input Operation (EC OSC Configuration) .....	101
Figure 7-1:	TLWT Instruction Operation.....	43	Figure 14-5:	External Parallel Resonant Crystal Oscillator Circuit .....	102
Figure 7-2:	TABLWT Instruction Operation.....	43	Figure 14-6:	External Series Resonant Crystal Oscillator Circuit .....	102
Figure 7-3:	TLRD Instruction Operation .....	44	Figure 14-7:	RC Oscillator Mode .....	102
Figure 7-4:	TABLRD Instruction Operation .....	44	Figure 14-8:	Watchdog Timer Block Diagram.....	104
Figure 7-5:	TABLWT Write Timing (External Memory) .....	46	Figure 14-9:	Wake-up From Sleep Through Interrupt... ..	105
Figure 7-6:	Consecutive TABLWT Write Timing (External Memory) .....	47	Figure 15-1:	General Format for Instructions .....	108
Figure 7-7:	TABLRD Timing .....	48	Figure 15-2:	Q Cycle Activity .....	109
Figure 7-8:	TABLRD Timing (Consecutive TABLRD Instructions) .....	48	Figure 17-1:	Parameter Measurement Information.....	154
Figure 9-1:	RA0 and RA1 Block Diagram .....	53	Figure 17-2:	External Clock Timing .....	155
Figure 9-2:	RA2 and RA3 Block Diagram .....	54	Figure 17-3:	CLKOUT and I/O Timing .....	156
Figure 9-3:	RA4 and RA5 Block Diagram .....	54	Figure 17-4:	Reset, Watchdog Timer, Oscillator Start-Up Timer and Power-Up Timer Timing .....	157
Figure 9-4:	Block Diagram of RB<7:4> and RB<1:0> Port Pins .....	55	Figure 17-5:	Timer0 Clock Timings.....	158
Figure 9-5:	Block Diagram of RB3 and RB2 Port Pins..	56	Figure 17-6:	Timer1, Timer2, And Timer3 Clock Timings.....	158
Figure 9-6:	Block Diagram of RC<7:0> Port Pins .....	58	Figure 17-7:	Capture Timings .....	159
Figure 9-7:	PORTD Block Diagram (in I/O Port Mode) .....	60	Figure 17-8:	PWM Timings .....	159
Figure 9-8:	PORTE Block Diagram (in I/O Port Mode) .....	62	Figure 17-9:	USART Module: Synchronous Transmission (Master/Slave) Timing .....	160
Figure 9-9:	Successive I/O Operation .....	64	Figure 17-10:	USART Module: Synchronous Receive (Master/Slave) Timing .....	160
Figure 11-1:	T0STA Register (Address: 05h, Unbanked) .....	67	Figure 17-11:	Memory Interface Write Timing .....	161
Figure 11-2:	Timer0 Module Block Diagram .....	68	Figure 17-12:	Memory Interface Read Timing .....	162
Figure 11-3:	TMR0 Timing with External Clock (Increment on Falling Edge) .....	68	Figure 18-1:	Typical RC Oscillator Frequency vs. Temperature .....	163
Figure 11-4:	TMR0 Timing: Write High or Low Byte .....	69	Figure 18-2:	Typical RC Oscillator Frequency vs. VDD .....	164
Figure 11-5:	TMR0 Read/Write in Timer Mode .....	70	Figure 18-3:	Typical RC Oscillator Frequency vs. VDD .....	164
Figure 12-1:	TCON1 Register (Address: 16h, Bank 3) ...	71	Figure 18-4:	Typical RC Oscillator Frequency vs. VDD .....	165
Figure 12-2:	TCON2 Register (Address: 17h, Bank 3) ...	72	Figure 18-5:	Transconductance (gm) of LF Oscillator vs. VDD .....	166
Figure 12-3:	Timer1 and Timer2 in Two 8-bit Timer/Counter Mode .....	73	Figure 18-6:	Transconductance (gm) of XT Oscillator vs. VDD .....	166
Figure 12-4:	TMR1 and TMR2 in 16-bit Timer/Counter Mode .....	74	Figure 18-7:	Typical IDD vs. Frequency (External Clock 25°C) .....	167
Figure 12-5:	Simplified PWM Block Diagram .....	75	Figure 18-8:	Maximum IDD vs. Frequency (External Clock 125°C to -40°C).....	167
Figure 12-6:	PWM Output .....	75	Figure 18-9:	Typical IPD vs. VDD Watchdog Disabled 25°C .....	168
Figure 12-7:	Timer3 with One Capture and One Period Register Block Diagram.....	78	Figure 18-10:	Maximum IPD vs. VDD Watchdog Disabled .....	168
Figure 12-8:	Timer3 with Two Capture Registers Block Diagram .....	79	Figure 18-11:	Typical IPD vs. VDD Watchdog Enabled 25°C .....	169
Figure 12-9:	TMR1, TMR2, and TMR3 Operation in External Clock Mode.....	80	Figure 18-12:	Maximum IPD vs. VDD Watchdog Enabled .....	169
Figure 12-10:	TMR1, TMR2, and TMR3 Operation in Timer Mode.....	81	Figure 18-13:	WDT Timer Time-Out Period vs. VDD .....	170
Figure 13-1:	TXSTA Register (Address: 15h, Bank 0) ....	83	Figure 18-14:	IOH vs. VOH, VDD = 3V.....	170
Figure 13-2:	RCSTA Register (Address: 13h, Bank 0) ...	84	Figure 18-15:	IOH vs. VOH, VDD = 5V.....	171
Figure 13-3:	USART Transmit.....	85	Figure 18-16:	IOL vs. VOL, VDD = 3V.....	171
Figure 13-4:	USART Receive.....	85	Figure 18-17:	IOL vs. VOL, VDD = 5V.....	172
Figure 13-5:	Asynchronous Master Transmission.....	90	Figure 18-18:	VTH (Input Threshold Voltage) of I/O Pins (TTL) vs. VDD .....	172
Figure 13-6:	Asynchronous Master Transmission (Back to Back) .....	90	Figure 18-19:	VTH, VIL of I/O Pins (Schmitt Trigger) vs. VDD .....	173
Figure 13-7:	RX Pin Sampling Scheme .....	91	Figure 18-20:	VTH (Input Threshold Voltage) of OSC1 Input (In XT and LF Modes) vs. VDD .....	173
Figure 13-8:	Asynchronous Reception.....	92	Figure 19-1:	Parameter Measurement Information.....	183
Figure 13-9:	Synchronous Transmission .....	94			
Figure 13-10:	Synchronous Transmission (Through TXEN) .....	94			
Figure 13-11:	Synchronous Reception (Master Mode, SREN).....	95			
Figure 14-1:	Configuration Word.....	99			
Figure 14-2:	Crystal or Ceramic Resonator Operation (XT or LF OSC Configuration) .....	100			

---

**Note the following details of the code protection feature on PICmicro® MCUs.**

- The PICmicro family meets the specifications contained in the Microchip Data Sheet.
- Microchip believes that its family of PICmicro microcontrollers is one of the most secure products of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the PICmicro microcontroller in a manner outside the operating specifications contained in the data sheet. The person doing so may be engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as “unbreakable”.
- Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our product.

If you have any further questions about this matter, please contact the local sales office nearest to you.

---

Information contained in this publication regarding device applications and the like is intended through suggestion only and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. No representation or warranty is given and no liability is assumed by Microchip Technology Incorporated with respect to the accuracy or use of such information, or infringement of patents or other intellectual property rights arising from such use or otherwise. Use of Microchip's products as critical components in life support systems is not authorized except with express written approval by Microchip. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights.

#### **Trademarks**


The Microchip name and logo, the Microchip logo, FilterLab, KEELOQ, microID, MPLAB, PIC, PICmicro, PICMASTER, PICSTART, PRO MATE, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

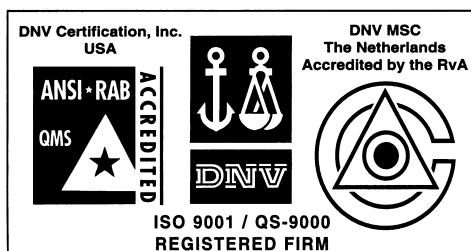
dsPIC, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, microPort, Migratable Memory, MPASM, MPLIB, MPLINK, MPSIM, MXDEV, PICC, PICDEM, PICDEM.net, rPIC, Select Mode and Total Endurance are trademarks of Microchip Technology Incorporated in the U.S.A.

Serialized Quick Turn Programming (SQTP) is a service mark of Microchip Technology Incorporated in the U.S.A.

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

© 2002, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

 Printed on recycled paper.



*Microchip received QS-9000 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona in July 1999. The Company's quality system processes and procedures are QS-9000 compliant for its PICmicro® 8-bit MCUs, KEELOQ® code hopping devices, Serial EEPROMs and microperipheral products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001 certified.*