

# **High-Performance 8-Bit CMOS EPROM/ROM Microcontroller**

#### Devices included in this data sheet:

- PIC17CR42
- PIC17C42A
- PIC17C43
- PIC17CR43
- PIC17C44
- PIC17C42†

### **Microcontroller Core Features:**

- · Only 58 single word instructions to learn
- All single cycle instructions (121 ns) except for program branches and table reads/writes which are two-cycle
- Operating speed:



- DC 33 MHz clock input
- DC 121 ns instruction cycle

Dovino	Program N	lemory	Data Mamaru	
Device	EPROM ROM		Data Memory	
PIC17CR42	-	2K	232	
PIC17C42A	2K	-	232	
PIC17C43	4K	-	454	
PIC17CR43	-	4K	454	
PIC17C44	8K	-	454	
PIC17C42†	2K	-	232	



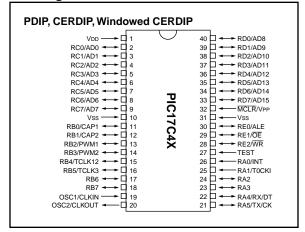
- Hardware Multiplier (Not available on the PIC17C42)
  - · Interrupt capability
  - · 16 levels deep hardware stack
  - · Direct, indirect and relative addressing modes
  - Internal/External program memory execution
  - 64K x 16 addressable program memory space

### **Peripheral Features:**

- · 33 I/O pins with individual direction control
- High current sink/source for direct LED drive
  - RA2 and RA3 are open drain, high voltage (12V), high current (60 mA), I/O
- Two capture inputs and two PWM outputs
  - Captures are 16-bit, max resolution 160 ns
  - PWM resolution is 1- to 10-bit
- TMR0: 16-bit timer/counter with 8-bit programmable prescaler
- TMR1: 8-bit timer/counter

†NOT recommended for new designs, use 17C42A.

### Pin Diagram



- TMR2: 8-bit timer/counter
- TMR3: 16-bit timer/counter
- Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI)

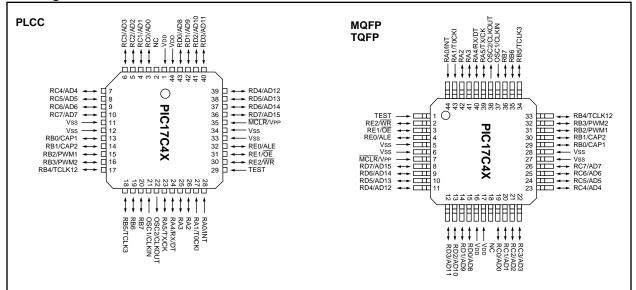
### **Special Microcontroller Features:**

- Power-on Reset (POR), Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- · Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Code-protection
- Power saving SLEEP mode
- · Selectable oscillator options

### **CMOS Technology:**

- · Low-power, high-speed CMOS EPROM/ROM technology
- · Fully static design
- Wide operating voltage range (2.5V to 6.0V)
- Commercial and Industrial Temperature Range
- · Low-power consumption
  - < 5 mA @ 5V, 4 MHz
  - 100 μA typical @ 4.5V, 32 kHz
  - < 1 μA typical standby current @ 5V

# Pin Diagrams Cont.'d



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.

# **Table of Contents**

1.0 Overview	
2.0 PIC17C4X Device Varieties	
3.0 Architectural Overview	<u></u>
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	
10.0 Overview of Timer Resources	65
11.0 Timer0	67
12.0 Timer1, Timer2, Timer3, PWMs and Captures	
13.0 Universal Synchronous Asynchronous Receiver Transmitter (USART) Module	
14.0 Special Features of the CPU	
15.0 Instruction Set Summary	107
16.0 Development Support	143
17.0 PIC17C42 Electrical Characteristics	
18.0 PIC17C42 DC and AC Characteristics	
19.0 PIC17CR42/42A/43/R43/44 Electrical Characteristics	
20.0 PIC17CR42/42A/43/R43/44 DC and AC Characteristics	
21.0 Packaging Information	
Appendix A: Modifications	
Appendix B: Compatibility	
Appendix C: What's New	
Appendix D: What's Changed	
Appendix E: PIC16/17 Microcontrollers	
Appendix F: Errata for PIC17C42 Silicon	
Index	
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, 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

**NOTES:** 

# 1.0 OVERVIEW

This data sheet covers the PIC17C4X group of the PIC17CXX family of microcontrollers. The following devices are discussed in this data sheet:

- PIC17C42
- PIC17CR42
- PIC17C42A
- PIC17C43
- PIC17CR43
- PIC17C44

The PIC17CR42, PIC17C42A, PIC17C43, PIC17CR43, and PIC17C44 devices include architectural enhancements over the PIC17C42. These enhancements will be discussed throughout this data sheet.

The PIC17C4X devices are 40/44-Pin, EPROM/ROM-based members of the versatile PIC17CXX family of low-cost, high-performance, CMOS, fully-static, 8-bit microcontrollers.

All PIC16/17 microcontrollers employ an advanced RISC architecture. The PIC17CXX has enhanced core features, 16-level deep stack, and multiple internal and external interrupt sources. The separate instruction and data buses of the Harvard architecture allow a 16-bit wide instruction word with a separate 8-bit wide data. The two stage instruction pipeline allows all instructions to execute in a single cycle, except for program branches (which require two cycles). A total of 55 instructions (reduced instruction set) are available in the PIC17C42 and 58 instructions in all the other devices. Additionally, a large register set gives some of the architectural innovations used to achieve a very high performance. For mathematical intensive applications all devices, except the PIC17C42, have a single cycle 8 x 8 Hardware Multiplier.

PIC17CXX microcontrollers typically achieve a 2:1 code compression and a 4:1 speed improvement over other 8-bit microcontrollers in their class.

PIC17C4X devices have up to 454 bytes of RAM and 33 I/O pins. In addition, the PIC17C4X adds several peripheral features useful in many high performance applications including:

- · Four timer/counters
- · Two capture inputs
- Two PWM outputs
- A Universal Synchronous Asynchronous Receiver Transmitter (USART)

These special features reduce external components, thus reducing cost, enhancing system reliability and reducing power consumption. There are four oscillator options, of which the single pin RC oscillator provides a low-cost solution, the LF oscillator is for low frequency crystals and minimizes power consumption, XT is a standard crystal, and the EC is for external clock input. The SLEEP (power-down) mode offers additional

power saving. The user can wake-up the chip from SLEEP through several external and internal interrupts and device resets.

There are four configuration options for the device operational modes:

- Microprocessor
- Microcontroller
- · Extended microcontroller
- Protected microcontroller

The microprocessor and extended microcontroller modes allow up to 64K-words of external program memory.

A highly reliable Watchdog Timer with its own on-chip RC oscillator provides protection against software malfunction.

Table 1-1 lists the features of the PIC17C4X devices.

A UV-erasable CERDIP-packaged version is ideal for code development while the cost-effective One-Time Programmable (OTP) version is suitable for production in any volume.

The PIC17C4X fits perfectly in applications ranging from precise motor control and industrial process control to automotive, instrumentation, and telecom applications. Other applications that require extremely fast execution of complex software programs or the flexibility of programming the software code as one of the last steps of the manufacturing process would also be well suited. The EPROM technology makes customization of application programs (with unique security codes, combinations, model numbers, parameter storage, etc.) fast and convenient. Small footprint package options make the PIC17C4X ideal for applications with space limitations that require high performance. High speed execution, powerful peripheral features, flexible I/O, and low power consumption all at low cost make the PIC17C4X ideal for a wide range of embedded control applications.

# 1.1 Family and Upward Compatibility

Those users familiar with the PIC16C5X and PIC16CXX families of microcontrollers will see the architectural enhancements that have been implemented. These enhancements allow the device to be more efficient in software and hardware requirements. Please refer to Appendix A for a detailed list of enhancements and modifications. Code written for PIC16C5X or PIC16CXX can be easily ported to PIC17CXX family of devices (Appendix B).

# 1.2 <u>Development Support</u>

The PIC17CXX family is supported by a full-featured macro assembler, a software simulator, an in-circuit emulator, a universal programmer, a "C" compiler, and fuzzy logic support tools.

TABLE 1-1: PIC17CXX FAMILY OF DEVICES

Features		PIC17C42	PIC17CR42	PIC17C42A	PIC17C43	PIC17CR43	PIC17C44
Maximum Frequency of O	peration	25 MHz	25 MHz		33 MHz	33 MHz	
Operating Voltage Range		4.5 - 5.5V	2.5 - 6.0V				
Program Memory x16	(EPROM)	2K	-	2K	4K	-	8K
	(ROM)	-	2K	=	=	4K	-
Data Memory (bytes)		232	232	232	454	454	454
Hardware Multiplier (8 x 8)	)	-	Yes	Yes	Yes	Yes	Yes
Timer0 (16-bit + 8-bit post	scaler)	Yes	Yes	Yes	Yes	Yes	Yes
Timer1 (8-bit)		Yes	Yes	Yes	Yes	Yes	Yes
Timer2 (8-bit)		Yes	Yes	Yes	Yes	Yes	Yes
Timer3 (16-bit)		Yes	Yes	Yes	Yes	Yes	Yes
Capture inputs (16-bit)		2	2	2	2 2		2
PWM outputs (up to 10-bit	i)	2	2	2	2	2 2	
USART/SCI		Yes	Yes	Yes	Yes	Yes	Yes
Power-on Reset		Yes	Yes	Yes	Yes Yes		Yes
Watchdog Timer		Yes	Yes	Yes	Yes	Yes	Yes
External Interrupts		Yes	Yes	Yes	Yes	Yes	Yes
Interrupt Sources		11	11	11	11	11	11
Program Memory Code Pr	rotect	Yes	Yes	Yes	Yes	Yes	Yes
I/O Pins		33	33	33	33	33	33
I/O High Current Capabil-	Source	25 mA					
ity	Sink	25 mA <sup>(1)</sup>					
Package Types		40-pin DIP					
		44-pin PLCC					
		44-pin MQFP					
			44-pin TQFP				

Note 1: Pins RA2 and RA3 can sink up to 60 mA.

# 2.0 PIC17C4X DEVICE VARIETIES

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

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

- C, as in PIC17C42. These devices have EPROM type memory and operate over the standard voltage range.
- LC, as in PIC17LC42. These devices have EPROM type memory, operate over an extended voltage range, and reduced frequency range.
- CR, as in PIC17CR42. These devices have ROM type memory and operate over the standard voltage range.
- LCR, as in PIC17LCR42. These devices have ROM type memory, operate over an extended voltage range, and reduced frequency range.

#### 2.1 UV Erasable Devices

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

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

# 2.2 <u>One-Time-Programmable (OTP)</u> Devices

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

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

# 2.3 Quick-Turnaround-Production (QTP) Devices

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

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

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

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

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

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

### 2.5 Read Only Memory (ROM) Devices

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

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

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 twostage 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.

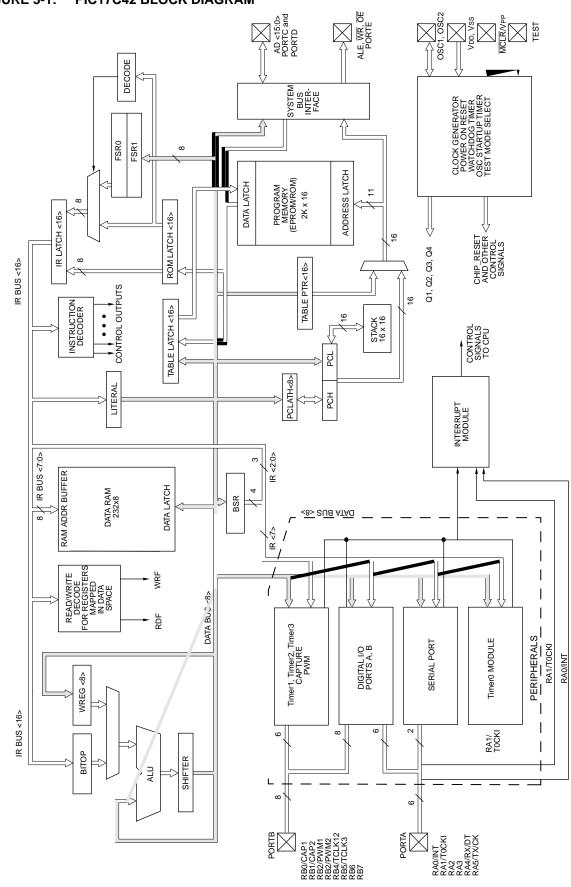
#### **EXAMPLE 3-1: SIGNED MATH**

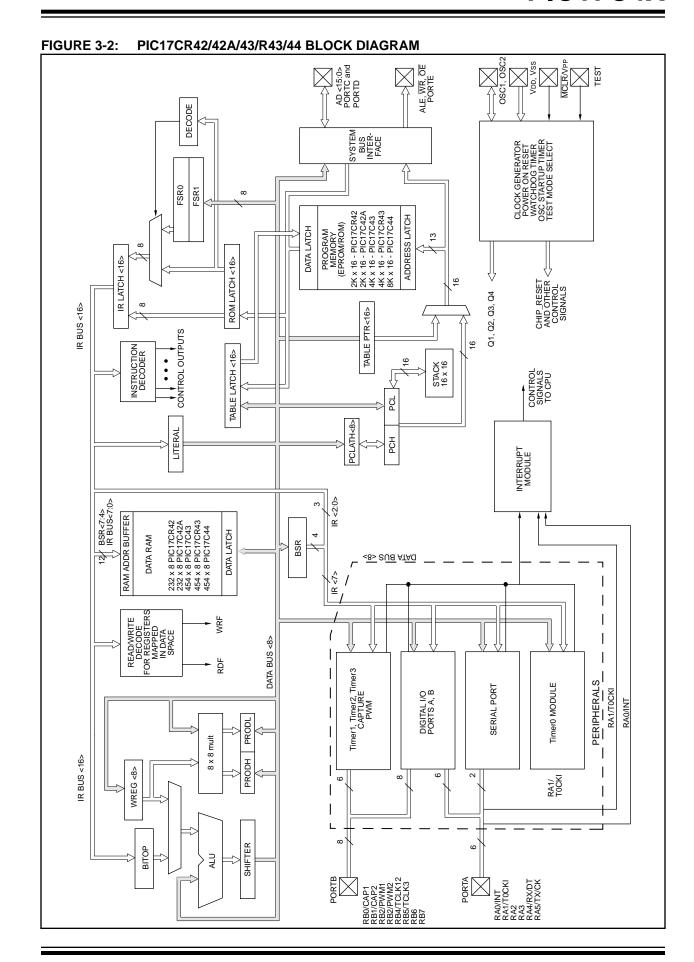
Hex Value	Signed Value Math	Unsigned Value Math
FFh + 01h = ?	-127 + 1 = -126 (FEh)	255 + 1 = 0 (00h); Carry bit = 1

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.

FIGURE 3-1: PIC17C42 BLOCK DIAGRAM





**TABLE 3-1: PINOUT DESCRIPTIONS** 

Name	DIP No.	PLCC No.	QFP No.	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	19	21	37	Туре	ST	Oscillator input in crystal/resonator or RC oscillator mode.
OSC I/CERIN	13	21	31	'	31	External clock input in external clock mode.
OSC2/CLKOUT	20	22	38	0	_	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.
						PORTA is a bi-directional I/O Port except for RA0 and RA1 which are input only.
RA0/INT	26	28	44	I	ST	RA0/INT can also be selected as an external interrupt input. Interrupt can be configured to be on positive or negative edge.
RA1/T0CKI	25	27	43	I	ST	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.
RA2	24	26	42	I/O	ST	High voltage, high current, open drain input/output port pins.
RA3	23	25	41	I/O	ST	High voltage, high current, open drain input/output port pins.
RA4/RX/DT	22	24	40	I/O	ST	RA4/RX/DT can also be selected as the USART (SCI) Asynchronous Receive or USART (SCI) Synchronous Data.
RA5/TX/CK	21	23	39	I/O	ST	RA5/TX/CK can also be selected as the USART (SCI) Asynchronous Transmit or USART (SCI) Synchronous Clock.
						PORTB is a bi-directional I/O Port with software configurable weak pull-ups.
RB0/CAP1	11	13	29	I/O	ST	RB0/CAP1 can also be the CAP1 input pin.
RB1/CAP2	12	14	30	I/O	ST	RB1/CAP2 can also be the CAP2 input pin.
RB2/PWM1	13	15	31	I/O	ST	RB2/PWM1 can also be the PWM1 output pin.
RB3/PWM2	14	16	32	I/O	ST	RB3/PWM2 can also be the PWM2 output pin.
RB4/TCLK12	15	17	33	I/O	ST	RB4/TCLK12 can also be the external clock input to Timer1 and Timer2.
RB5/TCLK3	16	18	34	I/O	ST	RB5/TCLK3 can also be the external clock input to Timer3.
RB6	17	19	35	I/O	ST	
RB7	18	20	36	I/O	ST	
						PORTC is a bi-directional I/O Port.
RC0/AD0	2	3	19	I/O	TTL	This is also the lower half of the 16-bit wide system bus
RC1/AD1	3	4	20	I/O	TTL	in microprocessor mode or extended microcontroller
RC2/AD2	4	5	21	I/O	TTL	mode. In multiplexed system bus configuration, these pins are address output as well as data input or output.
RC3/AD3	5	6	22	I/O	TTL	pino are address output as well as data input of output.
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.

**TABLE 3-1: PINOUT DESCRIPTIONS** 

Name	DIP No.	PLCC No.	QFP No.	I/O/P Type	Buffer Type	Description
						PORTD is a bi-directional I/O Port.
RD0/AD8	40	43	15	I/O	TTL	This is also the upper byte of the 16-bit system bus in
RD1/AD9	39	42	14	I/O	TTL	microprocessor mode or extended microprocessor mode
RD2/AD10	38	41	13	I/O	TTL	or extended microcontroller mode. In multiplexed system
RD3/AD11	37	40	12	I/O	TTL	bus configuration these pins are address output as well as data input or output.
RD4/AD12	36	39	11	I/O	TTL	as data input of output.
RD5/AD13	35	38	10	I/O	TTL	
RD6/AD14	34	37	9	I/O	TTL	
RD7/AD15	33	36	8	I/O	TTL	
						PORTE is a bi-directional I/O Port.
RE0/ALE	30	32	4	I/O	TTL	In microprocessor mode or extended microcontroller mode, it is the Address Latch Enable (ALE) output. Address should be latched on the falling edge of ALE output.
RE1/OE	29	31	3	I/O	TTL	In microprocessor or extended microcontroller mode, it is the Output Enable $(\overline{OE})$ control output (active low).
RE2/WR	28	30	2	I/O	TTL	In microprocessor or extended microcontroller mode, it is the Write Enable (WR) control output (active low).
TEST	27	29	1	I	ST	Test mode selection control input. Always tie to Vss for normal operation.
Vss	10,	11,	5, 6,	Р		Ground reference for logic and I/O pins.
	31	12, 33, 34	27, 28			
VDD	1	1, 44	16, 17	Р		Positive supply for logic and I/O pins.

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

#### **Clocking Scheme/Instruction Cycle** 3.1

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

#### 3.2 **Instruction Flow/Pipelining**

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

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

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

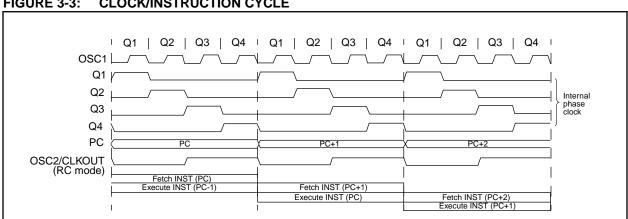
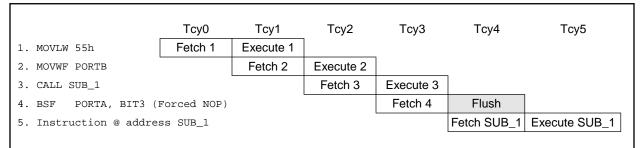


FIGURE 3-3: CLOCK/INSTRUCTION CYCLE

#### **EXAMPLE 3-2: INSTRUCTION PIPELINE FLOW**



All instructions are single cycle, except for any program branches. These take two cycles since the fetch instruction is "flushed" from the pipeline while the new instruction is being fetched and then executed.

#### RESET 4.0

The PIC17CXX differentiates between various kinds of

- Power-on Reset (POR)
- 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 MCLR or WDT Reset and on 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 TO and 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/OE and RE2/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)

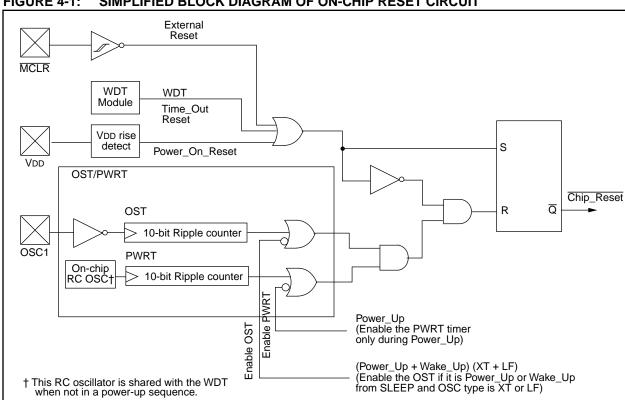
#### 4.1.1 POWER-ON RESET (POR)

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

The power-up time delay will vary from chip to chip and to VDD and temperature. See DC parameters for details.



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

### 4.1.3 OSCILLATOR START-UP TIMER (OST)

The Oscillator Start-up Timer (OST) provides a 1024 oscillator cycle (1024Tosc) delay after  $\overline{\text{MCLR}}$  is detected high or a wake-up from SLEEP event occurs.

The OST time-out is invoked only for XT and LF oscillator modes on a Power-on Reset or a Wake-up from SLEEP.

The OST counts the oscillator pulses on the OSC1/CLKIN pin. The counter only starts incrementing after the amplitude of the signal reaches the oscillator input thresholds. This delay allows the crystal oscillator or resonator to stabilize before the device exits reset. The length of time-out is a function of the crystal/resonator frequency.

#### 4.1.4 TIME-OUT SEQUENCE

On power-up the time-out sequence is as follows: First the internal POR signal goes high when the POR trip point is reached. If MCLR is high, then both the OST and PWRT timers start. In general the PWRT time-out is longer, except with low frequency crystals/resonators. The total time-out also varies based on oscillator configuration. Table 4-1 shows the times that are associated with the oscillator configuration. Figure 4-2 and Figure 4-3 display these time-out sequences.

If the device voltage is not within electrical specification at the end of a time-out, the  $\overline{\text{MCLR}}/\text{VPP}$  pin must be held low until the voltage is within the device specification. The use of an external RC delay is sufficient for many of these applications.

TABLE 4-1: TIME-OUT IN VARIOUS SITUATIONS

Oscillator Configuration	Power-up	Wake up from SLEEP	MCLR Reset
XT, LF	Greater of: 96 ms or 1024Tosc	1024Tosc	_
EC, RC	Greater of: 96 ms or 1024Tosc	_	_

The time-out sequence begins from the first rising edge of  $\overline{\text{MCLR}}$ .

Table 4-3 shows the reset conditions for some special registers, while Table 4-4 shows the initialization conditions for all the registers. The shaded registers (in Table 4-4) are for all devices except the PIC17C42. In the PIC17C42, the PRODH and PRODL registers are general purpose RAM.

TABLE 4-2: STATUS BITS AND THEIR SIGNIFICANCE

TO	PD	Event
1	1	Power-on Reset, MCLR Reset during normal operation, or CLRWDT instruction executed
1	0	MCLR Reset during SLEEP or interrupt wake-up from SLEEP
0	1	WDT Reset during normal operation
0	0	WDT Reset during SLEEP

In Figure 4-2, Figure 4-3 and Figure 4-4, TPWRT > TOST, as would be the case in higher frequency crystals. For lower frequency crystals, (i.e., 32 kHz) TOST would be greater.

TABLE 4-3: RESET CONDITION FOR THE PROGRAM COUNTER AND THE CPUSTA REGISTER

Event		PCH:PCL	CPUSTA	OST Active		
Power-on Reset	r-on Reset		ver-on Reset		11 11	Yes
MCLR Reset during normal ope	ration	0000h	11 11	No		
MCLR Reset during SLEEP		0000h	11 10	Yes (2)		
WDT Reset during normal opera	ation	0000h	11 01	No		
WDT Reset during SLEEP (3)		0000h	11 00	Yes (2)		
Interrupt wake-up from SLEEP	GLINTD is set	PC + 1	11 10	Yes (2)		
	GLINTD is clear	PC + 1 <sup>(1)</sup>	10 10	Yes (2)		

Legend: u = unchanged, x = unknown, - = unimplemented read as '0'.

- Note 1: On wake-up, this instruction is executed. The instruction at the appropriate interrupt vector is fetched and then executed.
  - 2: The OST is only active when the Oscillator is configured for XT or LF modes.
  - 3: The Program Counter = 0, that is the device branches to the reset vector. This is different from the mid-range devices.

FIGURE 4-2: TIME-OUT SEQUENCE ON POWER-UP (MCLR TIED TO VDD)

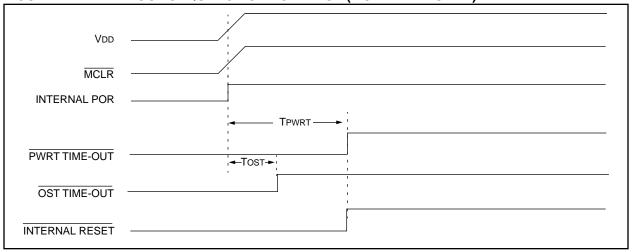


FIGURE 4-3: TIME-OUT SEQUENCE ON POWER-UP (MCLR NOT TIED TO VDD)

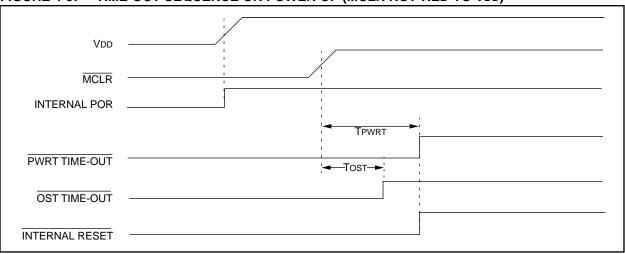
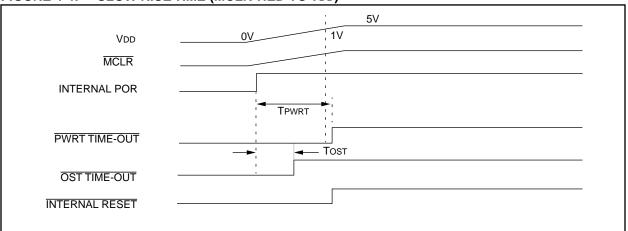
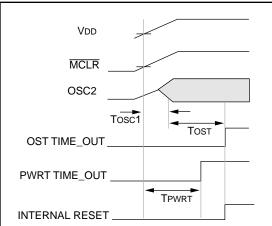


FIGURE 4-4: SLOW RISE TIME (MCLR TIED TO VDD)



#### FIGURE 4-5: OSCILLATOR START-UPTIME



This figure shows in greater detail the timings involved with the oscillator start-up timer. In this example the low frequency crystal start-up time is larger than power-up time (TPWRT).

Tosc1 = time for the crystal oscillator to react to an oscillation level detectable by the Oscillator Start-up Timer (ost).

Tost = 1024Tosc.

# FIGURE 4-6: USING ON-CHIP POR

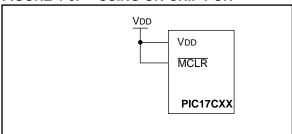


FIGURE 4-7: BROWN-OUT PROTECTION CIRCUIT 1

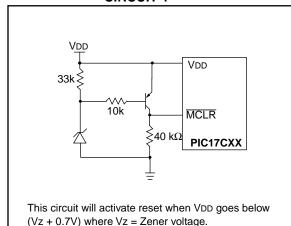
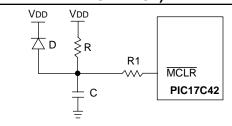
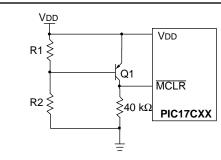


FIGURE 4-8: PIC17C42 EXTERNAL
POWER-ON RESET CIRCUIT
(FOR SLOW VDD
POWER-UP)



- Note 1: An external Power-on Reset circuit is required only if VDD power-up time is too slow. The diode D helps discharge the capacitor quickly when VDD powers down.
  - 2:  $R < 40 \text{ k}\Omega$  is recommended to ensure that the voltage drop across R does not exceed 0.2V (max. leakage current spec. on the  $\overline{MCLR}/VPP$  pin is 5  $\mu$ A). A larger voltage drop will degrade VIH level on the  $\overline{MCLR}/VPP$  pin.
  - 3:  $R1 = 100\Omega$  to 1 k $\Omega$  will limit any current flowing into  $\overline{MCLR}$  from external capacitor C in the event of  $\overline{MCLR}$ /VPP pin breakdown due to Electrostatic Discharge (ESD) or (Electrical Overstress) EOS.

FIGURE 4-9: BROWN-OUT PROTECTION CIRCUIT 2



This brown-out circuit is less expensive, albeit less accurate. Transistor Q1 turns off when VDD is below a certain level such that:

$$V_{DD} \bullet \frac{R1}{R1 + R2} = 0.7V$$

TABLE 4-4: INITIALIZATION CONDITIONS FOR SPECIAL FUNCTION REGISTERS

Register	Address	Power-on R	Reset		LR Reset DT Reset		from SLEEP h interrupt
Unbanked							
INDF0	00h	0000 000	00	0000	0000	0000	0000
FSR0	01h	XXXX XXX	xx	uuuu	uuuu	uuuu	uuuu
PCL	02h	0000h		000	)0h	PC +	1(2)
PCLATH	03h	0000 000	00	0000	0000	uuuu	uuuu
ALUSTA	04h	1111 xx	xx	1111	uuuu	1111	uuuu
T0STA	05h	0000 000	0 –	0000	000-	0000	000-
CPUSTA <sup>(3)</sup>	06h	11 11-		11	dd	uu	dd
INTSTA	07h	0000 000	00	0000	0000	uuuu	uuuu <mark>(1)</mark>
INDF1	08h	0000 000	00	0000	0000	uuuu	uuuu
FSR1	09h	xxxx xxx	xx	uuuu	uuuu	uuuu	uuuu
WREG	0Ah	xxxx xxx	xx	uuuu	uuuu	uuuu	uuuu
TMR0L	0Bh	xxxx xxx	xx	uuuu	uuuu	uuuu	uuuu
TMR0H	0Ch	XXXX XXX	xx	uuuu	uuuu	uuuu	uuuu
TBLPTRL (4)	0Dh	xxxx xxx	xx	uuuu	uuuu	uuuu	uuuu
TBLPTRH (4)	0Eh	xxxx xxx	xx	uuuu	uuuu	uuuu	uuuu
TBLPTRL (5)	0Dh	0000 000	00	0000	0000	uuuu	uuuu
TBLPTRH (5)	0Eh	0000 000	00	0000	0000	uuuu	uuuu
BSR	0Fh	0000 000	00	0000	0000	uuuu	uuuu
Bank 0							
PORTA	10h	0-xx xx	xx	0-uu	uuuu	uuuu	uuuu
DDRB	11h	1111 11:	11	1111	1111	uuuu	uuuu
PORTB	12h	xxxx xxx	xx	uuuu	uuuu	uuuu	uuuu
RCSTA	13h	0000 -00	0x	0000	-00u	uuuu	-uuu
RCREG	14h	xxxx xxx	xx	uuuu	uuuu	uuuu	uuuu
TXSTA	15h	0000:	1x	0000	1u	uuuu	uu
TXREG	16h	XXXX XXX	xx	uuuu	uuuu	uuuu	uuuu
SPBRG	17h	XXXX XXX	xx	uuuu	uuuu	uuuu	uuuu
Bank 1							
DDRC	10h	1111 11:	11	1111	1111	uuuu	uuuu
PORTC	11h	XXXX XXX	xx	uuuu	uuuu	uuuu	uuuu
DDRD	12h	1111 11:	11	1111	1111	uuuu	uuuu
PORTD	13h	xxxx xxx	xx	uuuu	uuuu	uuuu	uuuu
DDRE	14h	1:	11		-111		-uuu
PORTE	15h	x:	xx		-uuu		-uuu
PIR	16h	0000 003	10	0000	0010	uuuu	uuuu(1)
PIE	17h	0000 000	00	0000	0000	uuuu	uuuu

Legend: u = unchanged, x = unknown, - = unimplemented read as '0', <math>q = value depends on condition.

Note 1: One or more bits in INTSTA, PIR will be affected (to cause wake-up).

- 3: See Table 4-3 for reset value of specific condition.
- 4: Only applies to the PIC17C42.
- 5: Does not apply to the PIC17C42.

<sup>2:</sup> When the wake-up is due to an interrupt and the GLINTD bit is cleared, the PC is loaded with the interrupt vector.

TABLE 4-4: INITIALIZATION CONDITIONS FOR SPECIAL FUNCTION REGISTERS (Cont.'d)

Register	Address	Power-on Reset	MCLR Reset WDT Reset	Wake-up from SLEEP through interrupt
Bank 2	<u> </u>		·	<u> </u>
TMR1	10h	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR2	11h	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR3L	12h	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR3H	13h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PR1	14h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PR2	15h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PR3/CA1L	16h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PR3/CA1H	17h	xxxx xxxx	uuuu uuuu	uuuu uuuu
Bank 3				
PW1DCL	10h	xx	uu	uu
PW2DCL	11h	xx	uu	uu
PW1DCH	12h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PW2DCH	13h	xxxx xxxx	uuuu uuuu	uuuu uuuu
CA2L	14h	xxxx xxxx	uuuu uuuu	uuuu uuuu
CA2H	15h	xxxx xxxx	uuuu uuuu	uuuu uuuu
TCON1	16h	0000 0000	0000 0000	uuuu uuuu
TCON2	17h	0000 0000	0000 0000	uuuu uuuu
Unbanked				
PRODL (5)	18h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PRODH (5)	19h	xxxx xxxx	uuuu uuuu	uuuu uuuu

Legend: u = unchanged, x = unknown, - = unimplemented read as '0', <math>q = value depends on condition.

- Note 1: One or more bits in INTSTA, PIR will be affected (to cause wake-up).
  - 2: When the wake-up is due to an interrupt and the GLINTD bit is cleared, the PC is loaded with the interrupt vector.
  - 3: See Table 4-3 for reset value of specific condition.
  - 4: Only applies to the PIC17C42.
  - 5: Does not apply to the PIC17C42.

## 5.0 INTERRUPTS

The PIC17C4X devices have 11 sources of interrupt:

- External interrupt from the RA0/INT pin
- · Change on RB7:RB0 pins
- TMR0 Overflow
- TMR1 Overflow
- TMR2 Overflow
- TMR3 Overflow
- · USART Transmit buffer empty
- · USART Receive buffer full
- Capture1
- Capture2
- T0CKI edge occurred

There are four registers used in the control and status of interrupts. These are:

- CPUSTA
- INTSTA
- PIE
- PIR

The CPUSTA register contains the GLINTD bit. This is the Global Interrupt Disable bit. When this bit is set, all interrupts are disabled. This bit is part of the controller core functionality and is described in the Memory Organization section.

When an interrupt is responded to, the GLINTD bit is automatically set to disable any further interrupt, the return address is pushed onto the stack and the PC is loaded with the interrupt vector address. There are four interrupt vectors. Each vector address is for a specific interrupt source (except the peripheral interrupts which have the same vector address). These sources are:

- External interrupt from the RA0/INT pin
- TMR0 Overflow
- T0CKI edge occurred
- · Any peripheral interrupt

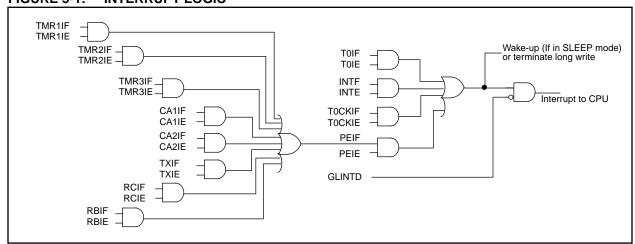
When program execution vectors to one of these interrupt vector addresses (except for the peripheral interrupt address), the interrupt flag bit is automatically cleared. Vectoring to the peripheral interrupt vector address does not automatically clear the source of the interrupt. In the peripheral interrupt service routine, the source(s) of the interrupt can be determined by testing the interrupt flag bits. The interrupt flag bit(s) must be cleared in software before re-enabling interrupts to avoid infinite interrupt requests.

All of the individual interrupt flag bits will be set regardless of the status of their corresponding mask bit or the GLINTD bit.

For external interrupt events, there will be an interrupt latency. For two cycle instructions, the latency could be one instruction cycle longer.

The "return from interrupt" instruction, RETFIE, can be used to mark the end of the interrupt service routine. When this instruction is executed, the stack is "POPed", and the GLINTD bit is cleared (to re-enable interrupts).

# FIGURE 5-1: INTERRUPT LOGIC



# 5.1 <u>Interrupt Status Register (INTSTA)</u>

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: TOIF, INTF, TOCKIF, 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	TOCKIF	TOIF	INTF	PEIE	T0CKIE	TOIE	INTE	R = Readable bit	
bit7	1001111				1001112	. 0.12	bit0	W = Writable bit	
DIL							DILU	- n = Value at POR reset	
bit 7:	bit 7: <b>PEIF</b> : 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:									
bit 5:	<b>T0IF</b> : 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:	TOCKIE: External Interrupt on T0CKI Pin Enable bit  1 = Enable software specified edge interrupt on the RA1/T0CKI pin  0 = Disable interrupt on the RA1/T0CKI pin								
bit 1:	T0IE: TMR0 Overflow Interrupt Enable bit  1 = Enable TMR0 overflow interrupt  0 = Disable TMR0 overflow interrupt								
bit 0:	INTE: External Interrupt on RA0/INT Pin Enable bit  1 = Enable software specified edge interrupt on the RA0/INT pin  0 = Disable software specified edge interrupt on the RA0/INT pin								

# 5.2 <u>Peripheral Interrupt Enable Register</u> (PIE)

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

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

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

# 5.3 <u>Peripheral Interrupt Request Register</u> (PIR)

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

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

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

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

Note:

# 5.4 <u>Interrupt Operation</u>

Global Interrupt Disable bit, GLINTD (CPUSTA<4>), enables all unmasked interrupts (if clear) or disables all interrupts (if set). Individual interrupts can be disabled through their corresponding enable bits in the INTSTA register. Peripheral interrupts need either the global peripheral enable PEIE bit disabled, or the specific peripheral enable bit disabled. Disabling the peripherals via the global peripheral enable bit, disables all peripheral interrupts. GLINTD is set on reset (interrupts disabled).

The RETFIE instruction allows returning from interrupt and re-enable interrupts at the same time.

When an interrupt is responded to, the GLINTD bit is automatically set to disable any further interrupt, the return address is pushed onto the stack and the PC is loaded with interrupt vector. There are four interrupt vectors to reduce interrupt latency.

The peripheral interrupt vector has multiple interrupt sources. Once in the peripheral interrupt service routine, the source(s) of the interrupt can be determined by polling the interrupt flag bits. The peripheral interrupt flag bit(s) must be cleared in software before reenabling interrupts to avoid continuous interrupts.

The PIC17C4X devices have four interrupt vectors. These vectors and their hardware priority are shown in Table 5-1. If two enabled interrupts occur "at the same time", the interrupt of the highest priority will be serviced first. This means that the vector address of that interrupt will be loaded into the program counter (PC).

TABLE 5-1: INTERRUPT VECTORS/ PRIORITIES

Address	Vector	Priority
0008h	External Interrupt on RA0/ INT pin (INTF)	1 (Highest)
0010h	TMR0 overflow interrupt (T0IF)	2
0018h	External Interrupt on T0CKI (T0CKIF)	3
0020h	Peripherals (PEIF)	4 (Lowest)

- Note 1: Individual interrupt flag bits are set regardless of the status of their corresponding mask bit or the GLINTD bit.
- Note 2: When disabling any of the INTSTA enable bits, the GLINTD bit should be set (disabled).
- Note 3: For the PIC17C42 only:

If an interrupt occurs while the Global Interrupt Disable (GLINTD) bit is being set, the GLINTD bit may unintentionally be reenabled by the user's Interrupt Service Routine (the RETFIE instruction). The events that would cause this to occur are:

- An interrupt occurs simultaneously with an instruction that sets the GLINTD bit.
- The program branches to the Interrupt vector and executes the Interrupt Service Routine.
- The Interrupt Service Routine completes with the execution of the RET-FIE instruction. This causes the GLINTD bit to be cleared (enables interrupts), and the program returns to the instruction after the one which was meant to disable interrupts.

The method to ensure that interrupts are globally disabled is:

 Ensure that the GLINTD bit was set by the instruction, as shown in the following code:

```
LOOP BSF CPUSTA, GLINTD; Disable Global; Interrupt
BTFSS CPUSTA, GLINTD; Global Interrupt; Disabled?
GOTO LOOP; NO, try again; YES, continue; with program; low
```

#### 5.5 **RA0/INT Interrupt**

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

#### 5.6 **TMR0 Interrupt**

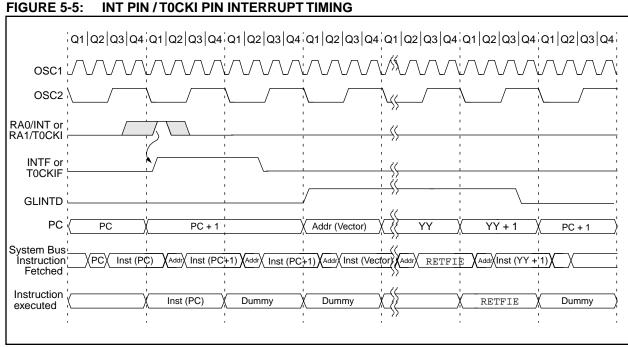
An overflow (FFFFh  $\rightarrow$  0000h) in TMR0 will set the TOIF (INTSTA<5>) bit. The interrupt can be enabled/ disabled by setting/clearing the T0IE control bit (INTSTA<1>). For operation of the Timer0 module, see Section 11.0.

#### 5.7 **T0CKI Interrupt**

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

#### 5.8 Peripheral Interrupt

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



# 5.9 Context Saving During Interrupts

During an interrupt, only the returned PC value is saved on the stack. Typically, users may wish to save key registers during an interrupt; e.g. WREG, ALUSTA and the BSR registers. This requires implementation in software.

Example 5-1 shows the saving and restoring of information for an interrupt service routine. The PUSH and POP routines could either be in each interrupt service routine or could be subroutines that were called. Depending on the application, other registers may also need to be saved, such as PCLATH.

### **EXAMPLE 5-1: SAVING STATUS AND WREG IN RAM**

```
; The addresses that are used to store the CPUSTA and WREG values
; must be in the data memory address range of 18h - 1Fh. Up to
; 8 locations can be saved and restored using
; the MOVFP instruction. This instruction neither affects the status
; bits, nor corrupts the WREG register.
;
       MOVFP WREG, TEMP_W
                            ; Save WREG
PUSH
       MOVFP
              ALUSTA, TEMP_STATUS ; Save ALUSTA
       MOVFP
               BSR, TEMP_BSR
                              ; Save BSR
ISR
                                   ; This is the interrupt service routine
       MOVFP
              TEMP_W, WREG
POP
                                  ; Restore WREG
       MOVFP
              TEMP_STATUS, ALUSTA ; Restore ALUSTA
       MOVFP
              TEMP_BSR, BSR ; Restore BSR
       RETFIE
                                   ; Return from Interrupts enabled
```

NOTES:

# 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

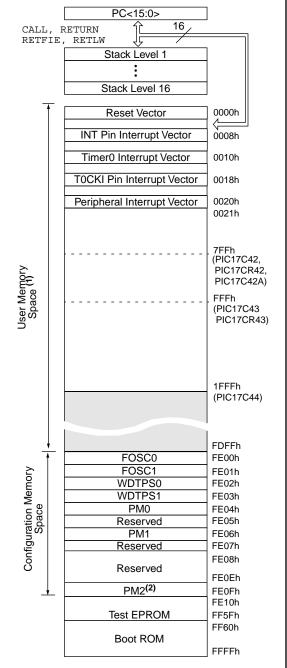
The microcontroller and protected microcontroller modes only allow internal execution. Any access beyond the program memory reads unknown data. The protected microcontroller mode also enables the code protection feature.

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



- Note 1: User memory space may be internal, external, or both. The memory configuration depends on the processor mode.
  - 2: This location is reserved on the PIC17C42.

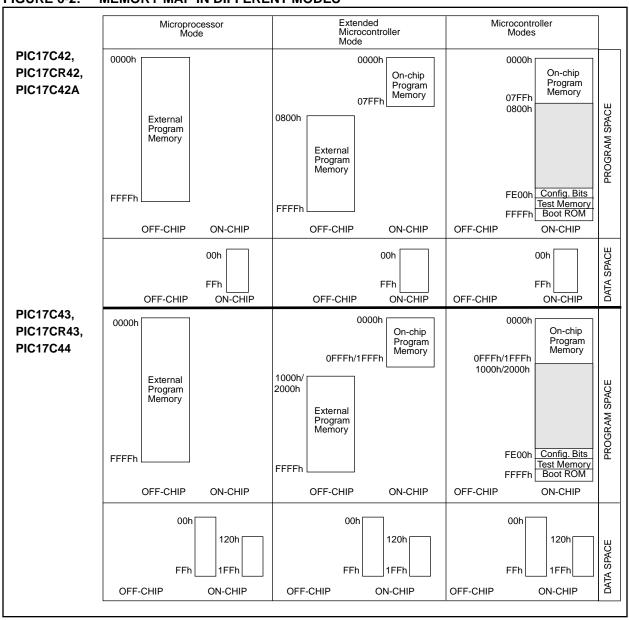
TABLE 6-1: MODE MEMORY ACCESS

Operating Mode	Internal Program Memory	Configuration Bits, Test Memory, Boot ROM		
Microprocessor	No Access	No Access		
Microcontroller	Access	Access		
Extended Microcontroller	Access	No Access		
Protected Microcontroller	Access	Access		

The PIC17C4X can operate in modes where the program memory is off-chip. They are the microprocessor and extended microcontroller modes. The microprocessor mode is the default for an unprogrammed device.

Regardless of the processor mode, data memory is always on-chip.

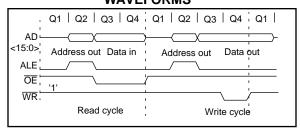
FIGURE 6-2: MEMORY MAP IN DIFFERENT MODES



#### 6.1.2 EXTERNAL MEMORY INTERFACE

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

FIGURE 6-3: EXTERNAL PROGRAM MEMORY ACCESS WAVEFORMS



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

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

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

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

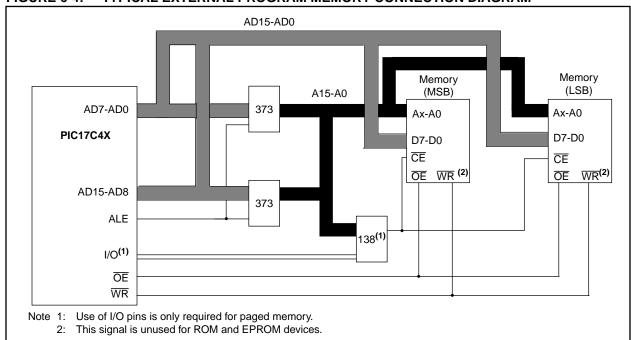
TABLE 6-2: EPROM MEMORY ACCESS TIME ORDERING SUFFIX

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

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

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

FIGURE 6-4: TYPICAL EXTERNAL PROGRAM MEMORY CONNECTION DIAGRAM



# 6.2 <u>Data Memory Organization</u>

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

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

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

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

### 6.2.1 GENERAL PURPOSE REGISTER (GPR)

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

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

### 6.2.2 SPECIAL FUNCTION REGISTERS (SFR)

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

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

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

FIGURE 6-5: PIC17C42 REGISTER FILE MAP

Addr	Unbanked			
00h	INDF0			
01h	FSR0			
02h	PCL			
03h	PCLATH			
04h	ALUSTA			
05h	T0STA			
06h	CPUSTA			
07h	INTSTA			
08h	INDF1			
09h	FSR1			
0Ah	WREG			
0Bh	TMR0L			
0Ch	TMR0H			
0Dh	TBLPTRL			
0Eh	TBLPTRH			
0Fh	BSR			
	Bank 0	Bank 1 <sup>(1)</sup>	Bank 2 <sup>(1)</sup>	Bank 3 <sup>(1)</sup>
10h	PORTA	DDRC	TMR1	PW1DCL
11h	DDRB	PORTC	TMR2	PW2DCL
12h	PORTB	DDRD	TMR3L	PW1DCH
13h	RCSTA	PORTD	TMR3H	PW2DCH
14h	RCREG	DDRE	PR1	CA2L
15h	TXSTA	PORTE	PR2	CA2H
16h	TXREG	PIR	PR3L/CA1L	TCON1
17h	SPBRG	PIE	PR3H/CA1H	TCON2
18h				
1Eh				
	General			
2011	Purpose			
	IVAIVI			
18h 1Fh 20h				

Note 1: SFR file locations 10h - 17h are banked. All other SFRs ignore the Bank Select Register (BSR) bits.

FIGURE 6-6: PIC17CR42/42A/43/R43/44
REGISTER FILE MAP

Addr	Unbanked			
00h	INDF0			
01h	FSR0			
02h	PCL			
03h	PCLATH			
04h	ALUSTA			
05h	TOSTA			
06h	CPUSTA			
07h	INTSTA			
08h	INDF1			
09h	FSR1			
0Ah	WREG			
0Bh	TMR0L			
0Ch	TMR0H			
0Dh	TBLPTRL			
0Eh	TBLPTRH			
0Fh	BSR			
	Bank 0	Bank 1 <sup>(1)</sup>	Bank 2 <sup>(1)</sup>	Bank 3 <sup>(1)</sup>
10h	PORTA	DDRC	TMR1	PW1DCL
11h	DDRB	PORTC	TMR2	PW2DCL
12h	PORTB	DDRD	TMR3L	PW1DCH
13h	RCSTA	PORTD	TMR3H	PW2DCH
14h	RCREG	DDRE	PR1	CA2L
15h	TXSTA	PORTE	PR2	CA2H
16h	TXREG	PIR	PR3L/CA1L	TCON1
17h	SPBRG	PIE	PR3H/CA1H	TCON2
18h	PRODL			
19h	PRODH			
1Ah				
4				
1Fh			1	
1Fh 20h	General Purpose RAM <b>(2)</b>	General Purpose RAM <sup>(2)</sup>		

- Note 1: SFR file locations 10h 17h are banked. All other SFRs ignore the Bank Select Register (BSR) bits.
  - 2: General Purpose Registers (GPR) locations 20h FFh and 120h 1FFh are banked. All other GPRs ignore the Bank Select Register (BSR) bits.

TABLE 6-3: SPECIAL FUNCTION REGISTERS

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)
Unbanke	ed										
00h	INDF0	Uses conte	ents of FS	R0 to addres	s data mem	ory (not a p	hysical regis	ster)			
01h	FSR0	Indirect da	ta memor	y address po	inter 0					xxxx xxxx	uuuu uuuu
02h	PCL	Low order	8-bits of P	C						0000 0000	0000 0000
03h <sup>(1)</sup>	PCLATH	Holding re	gister for u	upper 8-bits o	of PC					0000 0000	uuuu uuuu
04h	ALUSTA	FS3	FS2	FS1	FS0	OV	Z	DC	С	1111 xxxx	1111 uuuu
05h	TOSTA	INTEDG	T0SE	T0CS	PS3	PS2	PS1	PS0	_	0000 000-	0000 000-
06h <sup>(2)</sup>	CPUSTA	_	_	STKAV	GLINTD	TO	PD	_	_	11 11	11 qq
07h	INTSTA	PEIF	T0CKIF	TOIF	INTF	PEIE	T0CKIE	TOIE	INTE	0000 0000	0000 0000
08h	INDF1	Uses conte	ents of FS	R1 to addres	s data mem	ory (not a p	hysical regis	ster)			
09h	FSR1	Indirect da	ta memor	y address po	inter 1					xxxx xxxx	uuuu uuuu
0Ah	WREG	Working re	egister							xxxx xxxx	uuuu uuuu
0Bh	TMR0L	TMR0 regi	ister; low b	yte						xxxx xxxx	uuuu uuuu
0Ch	TMR0H	TMR0 regi	ister; high	byte						xxxx xxxx	uuuu uuuu
0Dh	TBLPTRL	Low byte of	of program	memory tab	le pointer					(4)	(4)
0Eh	TBLPTRH	High byte	of program	n memory tab	ole pointer					(4)	(4)
0Fh	BSR	Bank selec	ct register							0000 0000	0000 0000
Bank 0											
10h	PORTA	RBPU	_	RA5	RA4	RA3	RA2	RA1/T0CKI	RA0/INT	0-xx xxxx	0-uu uuuu
11h	DDRB	Data direc	tion registe	er for PORTE	3					1111 1111	1111 1111
12h	PORTB	PORTB da	ata latch							xxxx xxxx	uuuu uuuu
13h	RCSTA	SPEN	RX9	SREN	CREN	_	FERR	OERR	RX9D	0000 -00x	0000 -00u
14h	RCREG	Serial port	receive re	egister						xxxx xxxx	uuuu uuuu
15h	TXSTA	CSRC	TX9	TXEN	SYNC	_	_	TRMT	TX9D	00001x	00001u
16h	TXREG	Serial port	transmit r	egister						xxxx xxxx	uuuu uuuu
17h	SPBRG	Baud rate	generator	register						xxxx xxxx	uuuu uuuu
Bank 1											
10h	DDRC	Data direc	tion registe	er for PORTO						1111 1111	1111 1111
11h	PORTC	RC7/ AD7	RC6/ AD6	RC5/ AD5	RC4/ AD4	RC3/ AD3	RC2/ AD2	RC1/ AD1	RC0/ AD0	xxxx xxxx	uuuu uuuu
12h	DDRD	Data direc	tion registe	er for PORTE	)					1111 1111	1111 1111
13h	PORTD	RD7/ AD15	RD6/ AD14	RD5/ AD13	RD4/ AD12	RD3/ AD11	RD2/ AD10	RD1/ AD9	RD0/ AD8	xxxx xxxx	uuuu uuuu
14h	DDRE	Data direc	tion registe	er for PORTE						111	111
15h	PORTE	_	_	_	_	_	RE2/WR	RE1/OE			

#### **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 reg	ister; high l	oyte						xxxx xxxx	uuuu uuuu
14h	PR1	Timer1 pe	riod registe	er						xxxx xxxx	uuuu uuuu
15h	PR2	Timer2 period register								xxxx xxxx	uuuu uuuu
16h	PR3L/CA1L	Timer3 pe	Timer3 period register, low byte/capture1 register; low byte							xxxx xxxx	uuuu uuuu
17h	PR3H/CA1H	Timer3 pe	riod registe	er, high byte/	capture1 rec	gister; high l	oyte			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	CA10VF	PWM2ON	PWM10N	CA1/PR3	TMR3ON	TMR2ON	TMR10N	0000 0000	0000 0000
Unbanke	ed										
18h <sup>(5)</sup>	PRODL	Low Byte	of 16-bit Pr	oduct (8 x 8	Hardware M	fultiply)				xxxx xxxx	uuuu uuuu
19h <sup>(5)</sup>	PRODH	High Byte	of 16-bit P	roduct (8 x 8	B Hardware N	Multiply)				xxxx xxxx	uuuu uuuu

x = unknown, u = unchanged, - = unimplemented read as '0', q - value depends on condition. Shaded cells are unimplemented, read as '0'. The upper byte of the program counter is not directly accessible. PCLATH is a holding register for PC<15:8> whose contents are updated Legend: Note 1: from or transferred to the upper byte of the program counter.

The TO and PD status bits in CPUSTA are not affected by a MCLR reset.

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

The following values are for both TBLPTRL and TBLPTRH:

<sup>3:</sup> 4:

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) 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 -									
bit7 bit0 W = Writable bit -n = Value at POR reset	<u>R/W - 1</u>	R/W - 1	R/W - 1	R/W - 1	R/W - x	R/W - x	R/W - x	R/W - x	
-n = Value at POR reset	FS3	FS2	FS1	FS0	OV	Z	DC	С	
	bit7							bit0	
	bit 7-6:	FS3:FS2:	FSR1 Mo	de Select	bits				
bit 7-6: FS3:FS2: FSR1 Mode Select bits		00 = Post	auto-dec	rement ES	R1 value				

- 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.

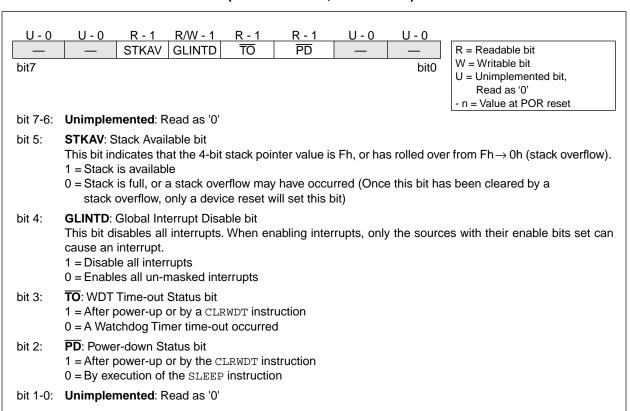
0 = No carry-out from the most significant bit of the result

Note: For borrow the polarity is reversed.

#### 6.2.2.2 CPU STATUS REGISTER (CPUSTA)

The CPUSTA register contains the status and control bits for the CPU. This register is used to globally enable/disable interrupts. If only a specific interrupt is desired to be enabled/disabled, please refer to the INTerrupt STAtus (INTSTA) register and the Peripheral Interrupt Enable (PIE) register. This register also indicates if the stack is available and contains the Power-down ( $\overline{\text{PD}}$ ) and Time-out ( $\overline{\text{TO}}$ ) bits. The  $\overline{\text{TO}}$ ,  $\overline{\text{PD}}$ , and STKAV bits are not writable. These bits are set and cleared according to device logic. Therefore, the result of an instruction with the CPUSTA register as destination may be different than intended.

#### FIGURE 6-8: CPUSTA REGISTER (ADDRESS: 06h, UNBANKED)



# 6.2.2.3 TMR0 STATUS/CONTROL REGISTER (T0STA)

This register contains various control bits. Bit7 (INTEDG) is used to control the edge upon which a signal on the RA0/INT pin will set the RB0/INT interrupt flag. The other bits configure the Timer0 prescaler and clock source. (Figure 11-1).

#### FIGURE 6-9: TOSTA REGISTER (ADDRESS: 05h, UNBANKED)

 R/W - 0
 P/W - 0
 <t

R = Readable bit
W = Writable bit
U = Unimplemented,
reads as '0'

-n = Value at POR reset

bit 7: INTEDG: RA0/INT Pin Interrupt Edge Select bit

This bit selects the edge upon which the interrupt is detected.

1 = Rising edge of RA0/INT pin generates interrupt

0 = Falling edge of RA0/INT pin generates interrupt

bit 6: T0SE: Timer0 Clock Input Edge Select bit

This bit selects the edge upon which TMR0 will increment.

When TOCS = 0

 ${\tt 1 = Rising\ edge\ of\ RA1/T0CKI\ pin\ increments\ TMR0\ and/or\ generates\ a\ T0CKIF\ interrupt}$ 

0 = Falling edge of RA1/T0CKI pin increments TMR0 and/or generates a T0CKIF interrupt

 $\underline{\text{When T0CS}} = 1$ 

Don't care

bit 5: TOCS: Timer0 Clock Source Select bit

This bit selects the clock source for Timer0.

1 = Internal instruction clock cycle (Tcy)

0 = TOCKI pin

bit 4-1: PS3:PS0: Timer0 Prescale Selection bits

These bits select the prescale value for Timer0.

PS3:PS0	Prescale Value
0000	1:1 1:2
0001 0010	1:2 1:4
0011	1:8
0100	1:16
0101	1:32
0110	1:64
0111	1:128
1xxx	1:256

bit 0: Unimplemented: Read as '0'

### 6.3 Stack Operation

The PIC17C4X devices have a 16 x 16-bit wide hardware stack (Figure 6-1). The stack is not part of either the program or data memory space, and the stack pointer is neither readable nor writable. The PC is "PUSHed" onto the stack when a CALL instruction is executed or an interrupt is acknowledged. The stack is "POPed" in the event of a RETURN, RETLW, or a RETFIE instruction execution. PCLATH is not affected by a "PUSH" or a "POP" operation.

The stack operates as a circular buffer, with the stack pointer initialized to '0' after all resets. There is a stack available bit (STKAV) to allow software to ensure that the stack has not overflowed. The STKAV bit is set after a device reset. When the stack pointer equals Fh, STKAV is cleared. When the stack pointer rolls over from Fh to 0h, the STKAV bit will be held clear until a device reset.

- Note 1: There is not a status bit for stack underflow. The STKAV bit can be used to detect the underflow which results in the stack pointer being at the top of stack.
- Note 2: There are no instruction mnemonics called PUSH or POP. These are actions that occur from the execution of the CALL, RETURN, RETLW, and RETFIE instructions, or the vectoring to an interrupt vector.
- Note 3: After a reset, if a "POP" operation occurs before a "PUSH" operation, the STKAV bit will be cleared. This will appear as if the stack is full (underflow has occurred). If a "PUSH" operation occurs next (before another "POP"), the STKAV bit will be locked clear. Only a device reset will cause this bit to set.

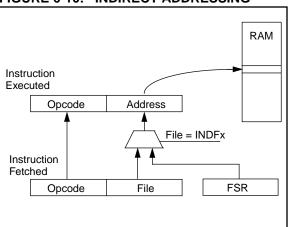
After the device is "PUSHed" sixteen times (without a "POP"), the seventeenth push overwrites the value from the first push. The eighteenth push overwrites the second push (and so on).

### 6.4 Indirect Addressing

Indirect addressing is a mode of addressing data memory where the data memory address in the instruction is not fixed. That is, the register that is to be read or written can be modified by the program. This can be useful for data tables in the data memory. Figure 6-10 shows the operation of indirect addressing. This shows the moving of the value to the data memory address specified by the value of the FSR register.

Example 6-1 shows the use of indirect addressing to clear RAM in a minimum number of instructions. A similar concept could be used to move a defined number of bytes (block) of data to the USART transmit register (TXREG). The starting address of the block of data to be transmitted could easily be modified by the program.

FIGURE 6-10: INDIRECT ADDRESSING



#### 6.4.1 INDIRECT ADDRESSING REGISTERS

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

- · INDF0 and FSR0
- · INDF1 and FSR1

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

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

#### 6.4.2 INDIRECT ADDRESSING OPERATION

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

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

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

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

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

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

If the source or destination of the indirect address is in banked memory, the location accessed will be determined by the value in the BSR. A simple program to clear RAM from 20h - FFh is shown in Example 6-1.

#### **EXAMPLE 6-1: INDIRECT ADDRESSING**

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

# 6.5 <u>Table Pointer (TBLPTRL and TBLPTRH)</u>

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

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

## 6.6 <u>Table Latch (TBLATH, TBLATL)</u>

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

### 6.7 Program Counter Module

The Program Counter (PC) is a 16-bit register. PCL, the low byte of the PC, is mapped in the data memory. PCL is readable and writable just as is any other register. PCH is the high byte of the PC and is not directly addressable. Since PCH is not mapped in data or program memory, an 8-bit register PCLATH (PC high latch) is used as a holding latch for the high byte of the PC. PCLATH is mapped into data memory. The user can read or write PCH through PCLATH.

The 16-bit wide PC is incremented after each instruction fetch during Q1 unless:

- Modified by GOTO, CALL, LCALL, RETURN, RETLW, or RETFIE instruction
- · Modified by an interrupt response
- Due to destination write to PCL by an instruction

"Skips" are equivalent to a forced NOP cycle at the skipped address.

Figure 6-11 and Figure 6-12 show the operation of the program counter for various situations.

FIGURE 6-11: PROGRAM COUNTER OPERATION

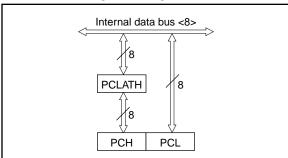
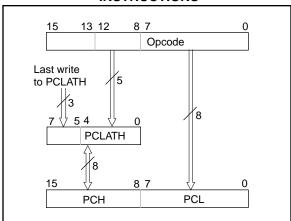


FIGURE 6-12: PROGRAM COUNTER USING THE CALL AND GOTO INSTRUCTIONS



Using Figure 6-11, the operations of the PC and PCLATH for different instructions are as follows:

#### a) LCALL instructions:

An 8-bit destination address is provided in the instruction (opcode). PCLATH is unchanged.

PCLATH → PCH

Opcode<7:0> → PCL

b) Read instructions on PCL:

Any instruction that reads PCL.

 $\text{PCL} \to \text{data bus} \to \text{ALU}$  or destination

PCH → PCLATH

c) Write instructions on PCL:

Any instruction that writes to PCL.

8-bit data  $\rightarrow$  data bus  $\rightarrow$  PCL

PCLATH → PCH

d) Read-Modify-Write instructions on PCL:

Any instruction that does a read-write-modify operation on PCL, such as ADDWF PCL.

Read:  $PCL \rightarrow data bus \rightarrow ALU$ 

Write: 8-bit result  $\rightarrow$  data bus  $\rightarrow$  PCL

PCLATH → PCH

e) RETURN instruction:

 $PCH \rightarrow PCLATH$ 

Stack<MRU> → PC<15:0>

Using Figure 6-12, the operation of the PC and PCLATH for GOTO and CALL instructions is a follows:

#### CALL, GOTO instructions:

A 13-bit destination address is provided in the instruction (opcode).

Opcode<12:0> → PC <12:0>

PC<15:13> → PCLATH<7:5>

Opcode<12:8>  $\rightarrow$  PCLATH <4:0>

The read-modify-write only affects the PCL with the result. PCH is loaded with the value in the PCLATH. For example, ADDWF PCL will result in a jump within the current page. If PC = 03F0h, WREG = 30h and PCLATH = 03h before instruction, PC = 0320h after the instruction. To accomplish a true 16-bit computed jump, the user needs to compute the 16-bit destination address, write the high byte to PCLATH and then write the low value to PCL.

The following PC related operations do not change PCLATH:

- a) LCALL, RETLW, and RETFIE instructions.
- b) Interrupt vector is forced onto the PC.
- Read-modify-write instructions on PCL (e.g. BSF PCL).

### 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.

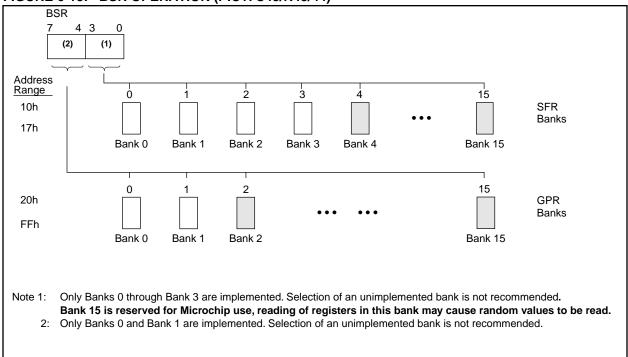
All the Special Function Registers (SFRs) are mapped into the data memory space. In order to accommodate the large number of registers, a banking scheme has been used. A segment of the SFRs, from address 10h to address 17h, is banked. The lower nibble of the bank select register (BSR) selects the currently 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 MOVLR 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.

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)



Note:

# 7.0 TABLE READS AND TABLE WRITES

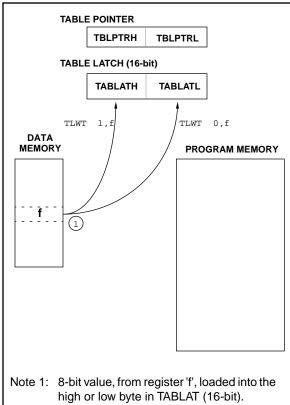
The PIC17C4X has four instructions that allow the processor to move data from the data memory space to the program memory space, and vice versa. Since the program memory space is 16-bits wide and the data memory space is 8-bits wide, two operations are required to move 16-bit values to/from the data memory.

The TLWT t,f and TABLWT t,i,f instructions are used to write data from the data memory space to the program memory space. The TLRD t,f and TABLRD t,i,f instructions are used to write data from the program memory space to the data memory space.

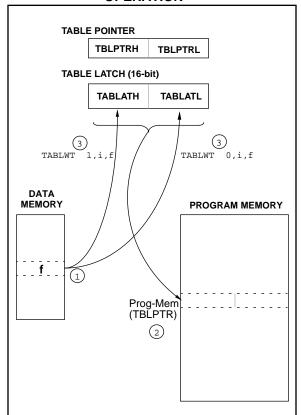
The program memory can be internal or external. For the program memory access to be external, the device needs to be operating in extended microcontroller or microprocessor mode.

Figure 7-1 through Figure 7-4 show the operation of these four instructions.

FIGURE 7-1: TLWT INSTRUCTION OPERATION



# FIGURE 7-2: TABLWT INSTRUCTION OPERATION



- Note 1: 8-bit value, from register 'f', loaded into the high or low byte in TABLAT (16-bit).
  - 2: 16-bit TABLAT value written to address Program Memory (TBLPTR).
  - 3: If "i" = 1, then TBLPTR = TBLPTR + 1, If "i" = 0, then TBLPTR is unchanged.

#### FIGURE 7-3: **TLRD INSTRUCTION OPERATION**

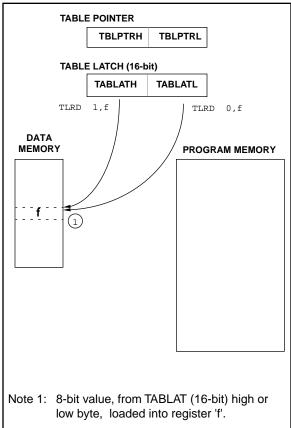
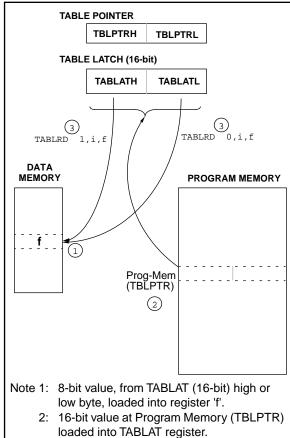


FIGURE 7-4: **TABLRD INSTRUCTION OPERATION** 



- loaded into TABLAT register.
- 3: If "i" = 1, then TBLPTR = TBLPTR + 1, If "i" = 0, then TBLPTR is unchanged.

### 7.1 <u>Table Writes to Internal Memory</u>

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:

- Disable all interrupt sources, except the source to terminate EPROM program write.
- 2. Raise MCLR/VPP pin to the programming voltage.
- 3. Clear the WDT.
- 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 TOCKI, RAO/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).

### 7.2 <u>Table Writes to External Memory</u>

Table writes to external memory are always two-cycle instructions. The second cycle writes the data to the external memory location. The sequence of events for an external memory write are the same for an internal write.

Note:

If an interrupt is pending or occurs during the TABLWT, the two cycle table write completes. The RAO/INT, TMR0, or TOCKI interrupt flag is automatically cleared or the pending peripheral interrupt is acknowledged.

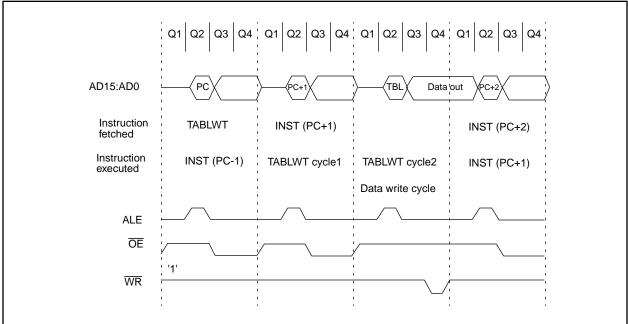
#### 7.2.2 TABLE WRITE CODE

The "i" operand of the TABLWT instruction can specify that the value in the 16-bit TBLPTR register is automatically incremented for the next write. In Example 7-1, the TBLPTR register is not automatically incremented.

#### **EXAMPLE 7-1: TABLE WRITE**

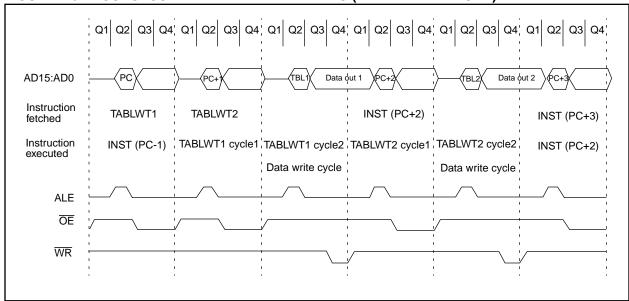
```
CLRWDT
                    ; Clear WDT
MOVLW HIGH (TBL_ADDR) ; Load the Table
MOVWF
      TBLPTRH ;
                       address
MOVLW LOW (TBL_ADDR) ;
MOVWF TBLPTRL
MOVLW HIGH (DATA) ; Load HI byte
TLWT 1, WREG
                   ; in TABLATCH
MOVLW LOW (DATA)
                   ; Load LO byte
                    ; in TABLATCH
TABLWT 0,0,WREG
                       and write to
                       program memory
                       (Ext. SRAM)
```





Note: If external write GLINTD = '1', Enable bit = '1', '1' → Flag bit, Do table write. The highest pending interrupt is cleared.





### 7.3 Table Reads

The table read allows the program memory to be read. This allows constant data to be stored in the program memory space, and retrieved into data memory when needed. Example 7-2 reads the 16-bit value at program memory address TBLPTR. After the dummy byte has been read from the TABLATH, the TABLATH is loaded with the 16-bit data from program memory address TBLPTR + 1. The first read loads the data into the latch, and can be considered a dummy read (unknown data loaded into 'f'). INDFO should be configured for either auto-increment or auto-decrement.

#### **EXAMPLE 7-2: TABLE READ**

HIGH (TBL\_ADDR) ; Load the Table MOVLW MOVWF TBLPTRH address LOW (TBL\_ADDR) MOVLW MOVWF TBLPTRL TABLRD 0,0,DUMMY ; Dummy read, ; Updates TABLATCH TLRD 1, INDF0 ; Read HI byte of TABLATCH TABLRD 0,1,INDF0 ; Read LO byte of TABLATCH and Update TABLATCH

FIGURE 7-7: TABLRD TIMING

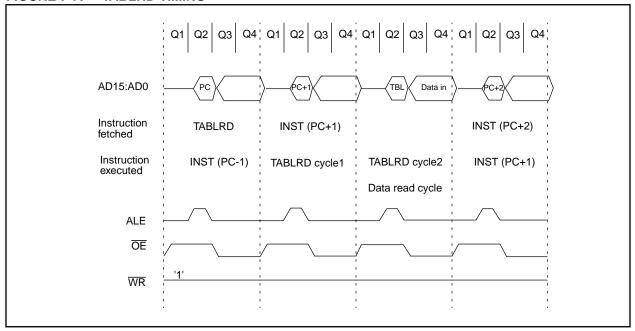
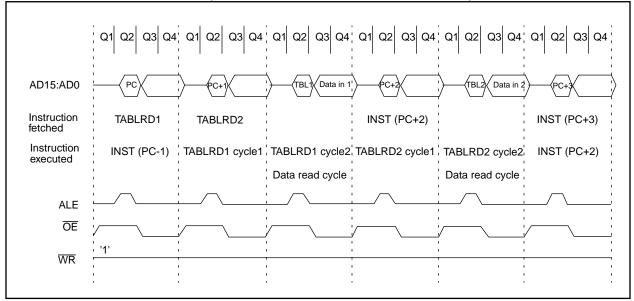


FIGURE 7-8: TABLRD TIMING (CONSECUTIVE TABLRD INSTRUCTIONS)



### 8.0 HARDWARE MULTIPLIER

All PIC17C4X devices except the PIC17C42, have an 8 x 8 hardware multiplier included in the ALU of the device. By making the multiply a hardware operation, it completes in a single instruction cycle. This is an unsigned multiply that gives a 16-bit result. The result is stored into the 16-bit PRODuct register (PRODH:PRODL). The multiplier does not affect any flags in the ALUSTA register.

Making the 8 x 8 multiplier execute in a single cycle gives the following advantages:

- · Higher computational throughput
- Reduces code size requirements for multiply algorithms

The performance increase allows the device to be used in applications previously reserved for Digital Signal Processors.

Table 8-1 shows a performance comparison between the PIC17C42 and all other PIC17CXX devices, which have the single cycle hardware multiply.

Example 8-1 shows the sequence to do an 8 x 8 unsigned multiply. Only one instruction is required when one argument of the multiply is already loaded in the WREG register.

Example 8-2 shows the sequence to do an  $8 \times 8$  signed multiply. To account for the sign bits of the arguments, each argument's most significant bit (MSb) is tested and the appropriate subtractions are done.

#### **EXAMPLE 8-1:** 8 x 8 MULTIPLY ROUTINE

```
MOVFP ARG1, WREG
MULWF ARG2 ; ARG1 * ARG2 ->
; PRODH:PRODL
```

# EXAMPLE 8-2: 8 x 8 SIGNED MULTIPLY ROUTINE

```
ARG1, WREG
MOVFP
        ARG2
MULWF
                   ; ARG1 * ARG2 ->
                   ; PRODH:PRODL
        ARG2, SB
                   ; Test Sign Bit
BTFSC
        PRODH, F
SUBWF
                   ; PRODH = PRODH
                            - ARG1
MOVFP
        ARG2, WREG
BTFSC
        ARG1, SB ; Test Sign Bit
SUBWF
        PRODH, F
                   ; PRODH = PRODH
                            - ARG2
```

TABLE 8-1: PERFORMANCE COMPARISON

Davidina	Davisa	Program Memory	Cycles (May)	Time		
Routine	Device	(Words)	Cycles (Max)	@ 25 MHz	@ 33 MHz	
8 x 8 unsigned	PIC17C42	13	69	11.04 μs	N/A	
	All other PIC17CXX devices	1	1	160 ns	121 ns	
8 x 8 signed	PIC17C42	_	_	_	N/A	
	All other PIC17CXX devices	6	6	960 ns	727 ns	
16 x 16 unsigned	PIC17C42	21	242	38.72 μs	N/A	
	All other PIC17CXX devices	24	24	3.84 µs	2.91 μs	
16 x 16 signed	PIC17C42	52	254	40.64 μs	N/A	
	All other PIC17CXX devices	36	36	5.76 μs	4.36 μs	

# PIC17C4X

Example 8-3 shows the sequence to do a 16  $\times$  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

RES3:RES0 = ARG1H:ARG1L \* ARG2H:ARG2L

=  $(ARG1H * ARG2H * 2^{16}) + (ARG1H * ARG2L * 2^{8}) +$ 

(ARG1L \* ARG2H \* 2<sup>8</sup>) +

(ARG1L \* ARG2L)

#### **EXAMPLE 8-3: 16 x 16 MULTIPLY ROUTINE**

```
ARG1L, WREG
MOVFP
        ARG2L ; ARG1L * ARG2L ->
{\tt MULWF}
                  ; PRODH:PRODL
MOVPF
        PRODH, RES1 ;
MOVPF
        PRODL, RESO ;
MOVFP
        ARG1H, WREG
        ARG2H ; ARG1H * ARG2H ->
MULWF
                     PRODH: PRODL
        PRODH, RES3 ;
MOVPF
        PRODL, RES2 ;
MOVPF
MOVFP
        ARG1L, WREG
        ARG2H ; ARG1L * ARG2H ->
MULWF
                ; PRODH:PRODL
MOVFP
        PRODL, WREG;
ADDWF
       RES1, F ; Add cross
        PRODH, WREG; products
MOVFP
ADDWFC
        RES2, F
CLRF
        WREG, F
                  ;
       RES3, F
ADDWFC
        ARG1H, WREG;
MOVFP
MULWF
        ARG2L ; ARG1H * ARG2L ->
                 ; PRODH:PRODL
MOVFP
        PRODL, WREG;
        RES1, F ; Add cross
ADDWF
MOVFP
        PRODH, WREG; products
ADDWFC
        RES2, F ;
        WREG, F
CLRF
                  ;
ADDWFC
        RES3, F
```

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

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

#### RES3:RES0

```
= ARG1H:ARG1L * ARG2H:ARG2L

= (ARG1H * ARG2H * 2<sup>16</sup>) + (ARG1H * ARG2L * 2<sup>8</sup>) + (ARG1L * ARG2H * 2<sup>8</sup>) + (ARG1L * ARG2L) + (-1 * ARG2H<7> * ARG1H:ARG1L * 2<sup>16</sup>) + (-1 * ARG1H<7> * ARG2H:ARG2L * 2<sup>16</sup>)
```

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

```
MOVFP
          ARG1L, WREG
  MULWF
          ARG2L
                    ; ARG1L * ARG2L ->
                     ; PRODH:PRODL
  MOVPF
          PRODH, RES1 ;
  MOVPF
          PRODL, RESO ;
  MOVFP
          ARG1H, WREG
  MULWF
          ARG2H ; ARG1H * ARG2H ->
                     ; PRODH:PRODL
  MOVPF
          PRODH, RES3 ;
  MOVPF
          PRODL, RES2 ;
  MOVED
          ARG1L, WREG
          ARG2H ; ARG1L * ARG2H ->
  MULWF
                    ; PRODH:PRODL
  MOVFP
          PRODL, WREG;
          RES1, F ; Add cross
  ADDWF
          PRODH, WREG; products
  MOVFP
  ADDWFC
          RES2, F
                  ;
          WREG, F
  CLRF
                     ;
  ADDWFC
          RES3, F
  MOVFP
          ARG1H, WREG;
          ARG2L ; ARG1H * ARG2L ->
  MULWF
                    ; PRODH:PRODL
  MOVED
          PRODL, WREG;
          RES1, F ; Add cross
  ADDWF
          PRODH, WREG; products
  MOVFP
          RES2, F ;
  ADDWFC
  CLRF
          WREG, F
                     ;
                    ;
  ADDWFC
          RES3, F
          ARG2H, 7
                   ; ARG2H:ARG2L neg?
  BTFSS
  GOTO
          SIGN_ARG1 ; no, check ARG1
  MOVFP
          ARG1L, WREG;
          RES2 ;
  SUBWE
  MOVFP
          ARG1H, WREG;
  SUBWFB
          RES3
SIGN_ARG1
          ARG1H, 7
                    ; ARG1H:ARG1L neg?
  BTFSS
  GOTO
          CONT_CODE ; no, done
  MOVFP
          ARG2L, WREG;
  SUBWF
          RES2 ;
  MOVED
          ARG2H, WREG;
  SUBWFB
          RES3
CONT_CODE
```

# PIC17C4X

NOTES:

#### 9.0 **I/O PORTS**

The PIC17C4X devices have five I/O ports, PORTA through PORTE. PORTB through PORTE have a corresponding Data Direction Register (DDR), which is used to configure the port pins as inputs or outputs. These five ports are made up of 33 I/O pins. Some of these ports pins are multiplexed with alternate functions.

PORTC, PORTD, and PORTE are multiplexed with the system bus. These pins are 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, these pins are general purpose I/O.

PORTA and PORTB are multiplexed with the peripheral features of the device. These peripheral features are:

- · Timer modules
- Capture module
- · PWM module
- USART/SCI module
- · External Interrupt pin

When some of these peripheral modules are turned on, the port pin will automatically configure to the alternate function. The modules that do this are:

- · PWM module
- USART/SCI module

When a pin is automatically configured as an output by a peripheral module, the pins data direction (DDR) bit is unknown. After disabling the peripheral module, the user should re-initialize the DDR bit to the desired configuration.

The other peripheral modules (which require an input) must have their data direction bit configured appropriately.

Note: A pin that is a peripheral input, can be configured as an output (DDRx<y> is cleared).

The peripheral events will be determined by the action output on the port pin.

### 9.1 PORTA Register

PORTA is a 6-bit wide latch. PORTA does not have a corresponding Data Direction Register (DDR).

Reading PORTA reads the status of the pins.

The RA1 pin is multiplexed with TMR0 clock input, and RA4 and RA5 are multiplexed with the USART functions. The control of RA4 and RA5 as outputs is automatically configured by the USART module.

#### 9.1.1 USING RA2, RA3 AS OUTPUTS

The RA2 and RA3 pins are open drain outputs. To use the RA2 or the RA3 pin(s) as output(s), simply write to the PORTA register the desired value. A '0' will cause the pin to drive low, while a '1' will cause the pin to float (hi-impedance). An external pull-up resistor should be used to pull the pin high. Writes to PORTA will not affect the other pins.

**Note:** When using the RA2 or RA3 pin(s) as output(s), read-modify-write instructions (such as BCF, BSF, BTG) on PORTA are not rec-

ommended.

Such operations read the port pins, do the desired operation, and then write this value to the data latch. This may inadvertently cause the RA2 or RA3 pins to switch from input to output (or vice-versa).

It is recommended to use a shadow register for PORTA. Do the bit operations on this shadow register and then move it to PORTA.

FIGURE 9-1: RA0 AND RA1 BLOCK DIAGRAM

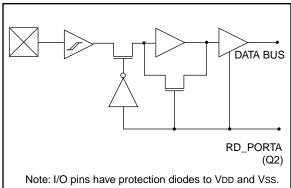


FIGURE 9-2: RA2 AND RA3 BLOCK DIAGRAM

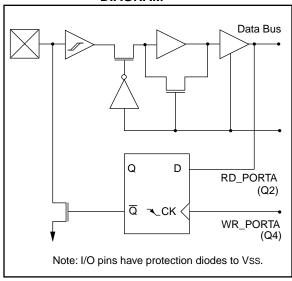


FIGURE 9-3: RA4 AND RA5 BLOCK DIAGRAM

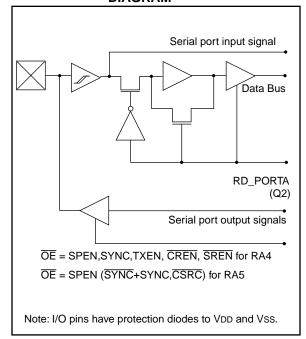


TABLE 9-1: PORTA FUNCTIONS

Name	Bit0	Buffer Type	Function
RA0/INT	bit0	ST	Input or external interrupt input.
RA1/T0CKI	bit1	ST	Input or clock input to the TMR0 timer/counter, and/or an external interrupt input.
RA2	bit2	ST	Input/Output. Output is open drain type.
RA3	bit3	ST	Input/Output. Output is open drain type.
RA4/RX/DT	bit4	ST	Input or USART Asynchronous Receive or USART Synchronous Data.
RA5/TX/CK	bit5	ST	Input or USART Asynchronous Transmit or USART Synchronous Clock.
RBPU	bit7	_	Control bit for PORTB weak pull-ups.

Legend: ST = Schmitt Trigger input.

TABLE 9-2: REGISTERS/BITS ASSOCIATED WITH PORTA

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)
10h, Bank 0	PORTA	RBPU	_	RA5	RA4	RA3	RA2	RA1/T0CKI	RA0/INT	0-xx xxxx	0-uu uuuu
05h, Unbanked	T0STA	INTEDG	T0SE	T0CS	PS3	PS2	PS1	PS0	_	0000 000-	0000 000-
13h, Bank 0	RCSTA	SPEN	RC9	SREN	CREN	_	FERR	OERR	RC9D	0000 -00x	0000 -00u
15h, Bank 0	TXSTA	CSRC	TX9	TXEN	SYNC	_	_	TRMT	TX9D	00001x	00001u

Legend: x = unknown, u = unchanged,  $- = unimplemented reads as '0'. Shaded cells are not used by PORTA. Note 1: Other (non power-up) resets include: external reset through <math>\overline{MCLR}$  and the Watchdog Timer Reset.

### 9.2 PORTB and DDRB Registers

PORTB is an 8-bit wide bi-directional port. The corresponding data direction register is DDRB. A '1' in DDRB configures the corresponding port pin as an input. A '0' in the DDRB register configures the corresponding port pin as an output. Reading PORTB reads the status of the pins, whereas writing to it will write to the port latch.

Each of the PORTB pins has a weak internal pull-up. A single control bit can turn on all the pull-ups. This is done by clearing the RBPU (PORTA<7>) bit. The weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are enabled on any reset.

PORTB also has an interrupt on change feature. Only pins configured as inputs can cause this interrupt to occur (i.e. any RB7:RB0 pin configured as an output is excluded from the interrupt on change comparison). The input pins (of RB7:RB0) are compared with the value in the PORTB data latch. The "mismatch" outputs of RB7:RB0 are OR'ed together to generate the PORTB Interrupt Flag RBIF (PIR<7>).

This interrupt can wake the device from SLEEP. The user, in the interrupt service routine, can clear the interrupt by:

- Read-Write PORTB (such as; MOVPF PORTB, PORTB). This will end mismatch condition.
- b) Then, clear the RBIF bit.

A mismatch condition will continue to set the RBIF bit. Reading then writing PORTB will end the mismatch condition, and allow the RBIF bit to be cleared.

This interrupt on mismatch feature, together with software configurable pull-ups on this port, allows easy interface to a key pad and make it possible for wake-up on key-depression. For an example, refer to AN552 in the *Embedded Control Handbook*.

The interrupt on change feature is recommended for wake-up on operations where PORTB is only used for the interrupt on change feature and key depression operation.

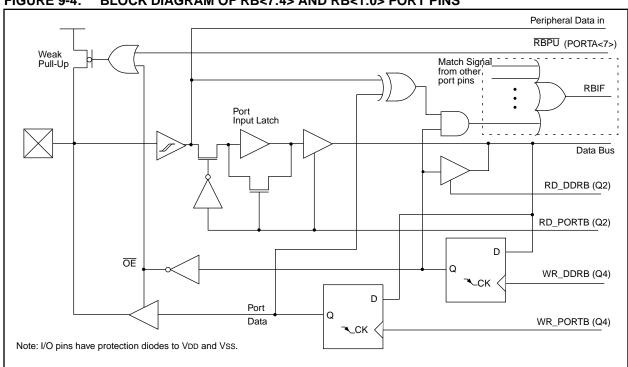


FIGURE 9-4: BLOCK DIAGRAM OF RB<7:4> AND RB<1:0> PORT PINS

Peripheral Data in RBPU (PORTA<7>) Weak Pull-Up Match Signal\_ from other, port pins **RBIF** Port Input Latch Data Bus RD\_DDRB (Q2) RD\_PORTB (Q2) D  $\overline{\mathsf{OE}}$ WR\_DDRB (Q4) **~**CK D Port Q Data WR\_PORTB (Q4) PWM\_output PWM\_select Note: I/O pins have protection diodes to VDD and Vss.

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

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 pullup and interrupt on change features.
RB1/CAP2	bit1	ST	Input/Output or the RB1/CAP2 input pin. Software programmable weak pullup 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 d	ata latch							xxxx xxxx	uuuu uuuu
11h, Bank 0	DDRB	Data dired	ction registe	er for PORTE	3					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	TO	PD	_	_	11 11	11 qq
07h, Unbanked	INTSTA	PEIF	T0CKIF	TOIF	INTF	PEIE	T0CKIE	TOIE	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	CA10VF	PWM2ON	PWM10N	CA1/PR3	TMR3ON	TMR2ON	TMR10N	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 MCLR and the Watchdog Timer Reset.

### 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 gen-

eral purpose I/O.

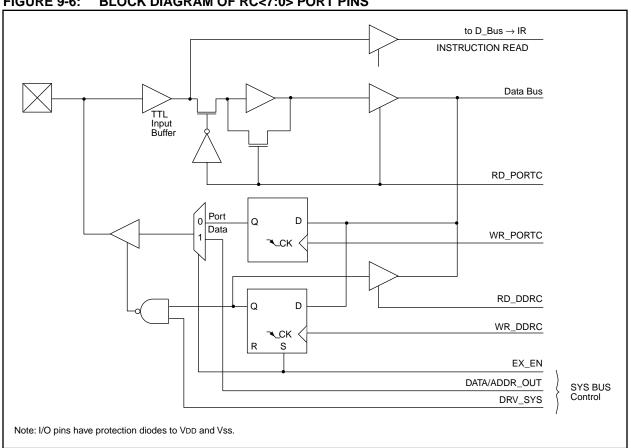
Example 9-2 shows the instruction sequence to initialize PORTC. The Bank Select Register (BSR) must be selected to Bank 1 for the port to be initialized.

### **EXAMPLE 9-2: INITIALIZING PORTC**

MOVLB	1	;	Select Bank 1
CLRF	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



**TABLE 9-5: PORTC FUNCTIONS** 

Name	Bit	Buffer Type	Function
RC0/AD0	bit0	TTL	Input/Output or system bus address/data pin.
RC1/AD1	bit1	TTL	Input/Output or system bus address/data pin.
RC2/AD2	bit2	TTL	Input/Output or system bus address/data pin.
RC3/AD3	bit3	TTL	Input/Output or system bus address/data pin.
RC4/AD4	bit4	TTL	Input/Output or system bus address/data pin.
RC5/AD5	bit5	TTL	Input/Output or system bus address/data pin.
RC6/AD6	bit6	TTL	Input/Output or system bus address/data pin.
RC7/AD7	bit7	TTL	Input/Output or system bus address/data pin.

Legend: TTL = TTL input.

## TABLE 9-6: REGISTERS/BITS ASSOCIATED WITH PORTC

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)
11h, Bank 1	PORTC	RC7/ AD7	RC6/ AD6	RC5/ AD5	RC4/ AD4	RC3/ AD3	RC2/ AD2	RC1/ AD1	RC0/ AD0	xxxx xxxx	uuuu uuuu
10h, Bank 1	DDRC	Data dired	ata direction register for PORTC							1111 1111	1111 1111

Legend: x = unknown, u = unchanged.

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

#### 9.4 **PORTD and DDRD Registers**

PORTD is an 8-bit bi-directional port. The corresponding data direction register is DDRD. A '1' in DDRD configures the corresponding port pin as an input. A '0' in the DDRC register configures the corresponding port pin as an output. Reading PORTD reads the status of the pins, whereas writing to it will write to the port latch. PORTD is multiplexed with the system bus. When operating as the system bus, PORTD is the high order byte of the address/data bus (AD15:AD8). The timing for the system bus is shown in the Electrical Characteristics section.

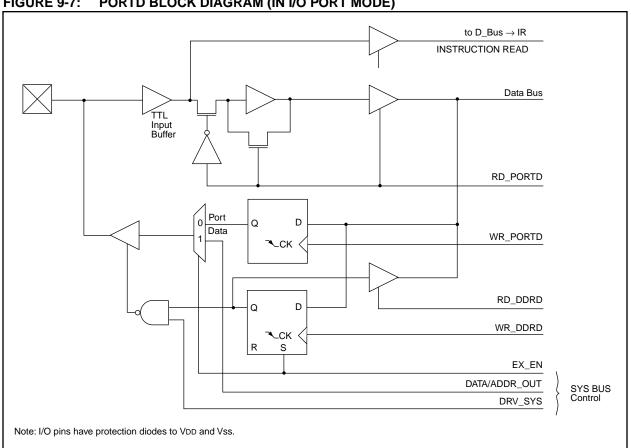
Note: This port is configured as the system bus when the device's configuration bits are selected to Microprocessor or Extended Microcontroller modes. In the two other microcontroller modes, this port is a general purpose I/O.

Example 9-3 shows the instruction sequence to initialize PORTD. The Bank Select Register (BSR) must be selected to Bank 1 for the port to be initialized.

### **EXAMPLE 9-3: INITIALIZING PORTD**

MOVLB	1	;	Select Bank 1
CLRF	PORTD	;	Initialize PORTD data
		;	latches before setting
		;	the data direction
		;	register
MOVLW	0xCF	;	Value used to initialize
		;	data direction
MOVWF	DDRD	;	Set RD<3:0> as inputs
		;	RD<5:4> as outputs
		;	RD<7:6> as inputs

FIGURE 9-7: PORTD BLOCK DIAGRAM (IN I/O PORT MODE)



**TABLE 9-7: PORTD FUNCTIONS** 

Name	Bit	Buffer Type	Function
RD0/AD8	bit0	TTL	Input/Output or system bus address/data pin.
RD1/AD9	bit1	TTL	Input/Output or system bus address/data pin.
RD2/AD10	bit2	TTL	Input/Output or system bus address/data pin.
RD3/AD11	bit3	TTL	Input/Output or system bus address/data pin.
RD4/AD12	bit4	TTL	Input/Output or system bus address/data pin.
RD5/AD13	bit5	TTL	Input/Output or system bus address/data pin.
RD6/AD14	bit6	TTL	Input/Output or system bus address/data pin.
RD7/AD15	bit7	TTL	Input/Output or system bus address/data pin.

Legend: TTL = TTL input.

## TABLE 9-8: REGISTERS/BITS ASSOCIATED WITH PORTD

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)
13h, Bank 1	PORTD	RD7/ AD15	RD6/ AD14	RD5/ AD13	RD4/ AD12	RD3/ AD11	RD2/ AD10	RD1/ AD9	RD0/ AD8	xxxx xxxx	uuuu uuuu
12h, Bank 1	DDRD	Data dired	Data direction register for PORTD							1111 1111	1111 1111

Legend: x = unknown, u = unchanged.

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

#### 9.4.1 PORTE AND DDRE REGISTER

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

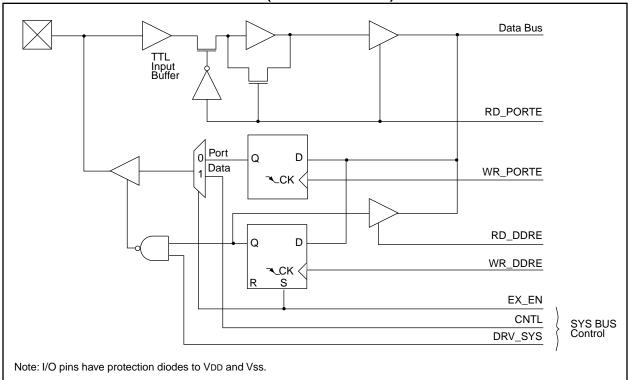
**Note:** This port is configured as the system bus when the device's configuration bits are selected to Microprocessor or Extended Microcontroller modes. In the two other microcontroller modes, this port is a general purpose I/O.

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

#### **EXAMPLE 9-4: INITIALIZING PORTE**

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

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



**TABLE 9-9: PORTE FUNCTIONS** 

Name	Bit	Buffer Type	Function
RE0/ALE	bit0	TTL	Input/Output or system bus Address Latch Enable (ALE) control pin.
RE1/OE	bit1	TTL	Input/Output or system bus Output Enable (OE) control pin.
RE2/WR	bit2	TTL	Input/Output or system bus Write (WR) control pin.

Legend: TTL = TTL input.

## TABLE 9-10: REGISTERS/BITS ASSOCIATED WITH PORTE

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)
15h, Bank 1	PORTE	_	_	_	_	_	RE2/WR	RE1/OE	RE0/ALE	xxx	uuu
14h, Bank 1	DDRE	Data dired	Data direction register for PORTE							111	111

Legend: x = unknown, u = unchanged, - = unimplemented read as '0'. Shaded cells are not used by PORTE.

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

### 9.5 <u>I/O Programming Considerations</u>

#### 9.5.1 BI-DIRECTIONAL I/O PORTS

Any instruction which writes, operates internally as a read followed by a write operation. For example, the BCF and BSF instructions read the register into the CPU, execute the bit operation, and write the result back to the register. Caution must be used when these instructions are applied to a port with both inputs and outputs defined. For example, a BSF operation on bit5 of PORTB will cause all eight bits of PORTB to be read into the CPU. Then the BSF operation takes place on bit5 and PORTB is written to the output latches. If another bit of PORTB is used as a bi-directional I/O pin (e.g. bit0) and it is defined as an input at this time, the input signal present on the pin itself would be read into the CPU and re-written to the data latch of this particular pin, overwriting the previous content. As long as the pin stays in the input mode, no problem occurs. However, if bit0 is switched into output mode later on, the content of the data latch may now be unknown.

Reading a port reads the values of the port pins. Writing to the port register writes the value to the port latch. When using read-modify-write instructions (BCF , BSF, BTG, etc.) on a port, the value of the port pins is read, the desired operation is performed with this value, and the value is then written to the port latch.

Example 9-5 shows the effect of two sequential read-modify-write instructions on an I/O port

# EXAMPLE 9-5: READ MODIFY WRITE INSTRUCTIONS ON AN I/O PORT

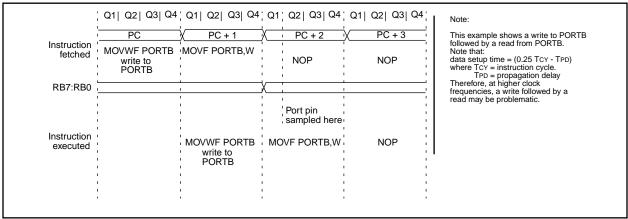
```
; Initial PORT settings: PORTB<7:4> Inputs
                         PORTB<3:0> Outputs
; PORTB<7:6> have pull-ups and are
; not connected to other circuitry
                        PORT latch PORT pins
;
                        _____
                                    _____
;
         PORTB, 7
   BCF
                        01pp pppp
                                    11pp pppp
         PORTB, 6
                        10pp pppp
                                    11pp pppp
   BCF
         DDRB. 7
                        10pp pppp
                                    11pp pppp
   BCF
         DDRB, 6
                        10pp pppp
                                    10pp pppp
; Note that the user may have expected the
 pin values to be 00pp pppp. The 2nd BCF
; caused RB7 to be latched as the pin value
; (High).
```

Note: A pin actively outputting a Low or High should not be driven from external devices in order to change the level on this pin (i.e. "wired-or", "wired-and"). The resulting high output currents may damage the device.

#### 9.5.2 SUCCESSIVE OPERATIONS ON I/O PORTS

The actual write to an I/O port happens at the end of an instruction cycle, whereas for reading, the data must be valid at the beginning of the instruction cycle (Figure 9-9). Therefore, care must be exercised if a write followed by a read operation is carried out on the same I/O port. The sequence of instructions should be such to allow the pin voltage to stabilize (load dependent) before executing the instruction that reads the values on that I/O port. Otherwise, the previous state of that pin may be read into the CPU rather than the "new" state. When in doubt, it is better to separate these instructions with a NOP or another instruction not accessing this I/O port.





# 10.0 OVERVIEW OF TIMER RESOURCES

The PIC17C4X has four timer modules. Each module can generate an interrupt to indicate that an event has occurred. These timers are called:

- Timer0 16-bit timer with programmable 8-bit prescaler
- Timer1 8-bit timer
- Timer2 8-bit timer
- Timer3 16-bit timer

For enhanced time-base functionality, two input Captures and two Pulse Width Modulation (PWM) outputs are possible. The PWMs use the TMR1 and TMR2 resources and the input Captures use the TMR3 resource.

#### 10.1 <u>Timer0 Overview</u>

The Timer0 module is a simple 16-bit overflow counter. The clock source can be either the internal system clock (Fosc/4) or an external clock.

The Timer0 module also has a programmable prescaler option. The PS3:PS0 bits (T0STA<4:1>) determine the prescaler value. TMR0 can increment at the following rates: 1:1, 1:2, 1:4, 1:8, 1:16, 1:32, 1:64, 1:128, 1:256.

When Timer0's clock source is an external clock, the Timer0 module can be selected to increment on either the rising or falling edge.

Synchronization of the external clock occurs after the prescaler. When the prescaler is used, the external clock frequency may be higher then the device's frequency. The maximum frequency is 50 MHz, given the high and low time requirements of the clock.

### 10.2 <u>Timer1 Overview</u>

The TImer0 module is an 8-bit timer/counter with an 8-bit period register (PR1). When the TMR1 value rolls over from the period match value to 0h, the TMR1IF flag is set, and an interrupt will be generated when enabled. In counter mode, the clock comes from the RB4/TCLK12 pin, which can also be selected to be the clock for the Timer2 module.

TMR1 can be concatenated to TMR2 to form a 16-bit timer. The TMR1 register is the LSB and TMR2 is the MSB. When in the 16-bit timer mode, there is a corresponding 16-bit period register (PR2:PR1). When the TMR2:TMR1 value rolls over from the period match value to 0h, the TMR1IF flag is set, and an interrupt will be generated when enabled.

### 10.3 <u>Timer2 Overview</u>

The TMR2 module is an 8-bit timer/counter with an 8-bit period register (PR2). When the TMR2 value rolls over from the period match value to 0h, the TMR2IF flag is set, and an interrupt will be generated when enabled. In counter mode, the clock comes from the RB4/TCLK12 pin, which can also be selected to be the clock for the TMR1 module.

TMR1 can be concatenated to TMR2 to form a 16-bit timer. The TMR2 register is the MSB and TMR1 is the LSB. When in the 16-bit timer mode, there is a corresponding 16-bit period register (PR2:PR1). When the TMR2:TMR1 value rolls over from the period match value to 0h, the TMR1IF flag is set, and an interrupt will be generated when enabled.

### 10.4 <u>Timer3 Overview</u>

The TImer3 module is a 16-bit timer/counter with a 16-bit period register. When the TMR3H:TMR3L value rolls over to 0h, the TMR3IF bit is set and an interrupt will be generated when enabled. In counter mode, the clock comes from the RB5/TCLK3 pin.

When operating in the dual capture mode, the period registers become the second 16-bit capture register.

#### 10.5 Role of the Timer/Counters

The timer modules are general purpose, but have dedicated resources associated with them. Tlmer1 and Timer2 are the time-bases for the two Pulse Width Modulation (PWM) outputs, while Timer3 is the time-base for the two input captures.

# PIC17C4X

NOTES:

## 11.0 TIMER0

The Timer0 module consists of a 16-bit timer/counter, TMR0. The high byte is TMR0H and the low byte is TMR0L. A software programmable 8-bit prescaler makes an effective 24-bit overflow timer. The clock source is also software programmable as either the internal instruction clock or the RA1/T0CKI pin. The control bits for this module are in register T0STA (Figure 11-1).

FIGURE 11-1: TOSTA REGISTER (ADDRESS: 05h, UNBANKED)

R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	U - 0	
INTEDO		TOCS	PS3	PS2	PS1	PS0	_	R = Readable bit W = Writable bit
bit7							bit0	U = Unimplemented, Read as '0' -n = Value at POR reset
bit 7:	INTEDG: R This bit seld 1 = Rising 6 0 = Falling 6	ects the ed edge of RA	ge upon w 0/INT pin g	hich the in generates i	terrupt is d nterrupt	etected		
bit 6:		ects the ed S = 0 edge of RA edge of RA	ge upon w	hich TMR0 n incremei	nts TMR0 a	and/or gene		CKIF interrupt CKIF interrupt
bit 5:	T0CS: Timer0 Clock Source Select bit This bit selects the clock source for TMR0. 1 = Internal instruction clock cycle (TcY) 0 = T0CKI pin							
bit 4-1:	PS3:PS0: 7 These bits				R0.			
	PS3:PS0	Pre	scale Valu	Э				
	0000 0001 0010 0011 0100 0101 0110 0111 1xxx		1:1 1:2 1:4 1:8 1:16 1:32 1:64 1:128 1:256					
bit 0:	Unimplem	<b>ented</b> : Rea	nd as '0'					

© 1996 Microchip Technology Inc.

### 11.1 <u>Timer0 Operation</u>

When the TOCS (TOSTA<5>) bit is set, TMR0 increments on the internal clock. When TOCS is clear, TMR0 increments on the external clock (RA1/TOCKI pin). The external clock edge can be configured in software. When the TOSE (TOSTA<6>) bit is set, the timer will increment on the rising edge of the RA1/TOCKI pin. When TOSE is clear, the timer will increment on the falling edge of the RA1/TOCKI 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 (TOIF) is set. The TMR0 interrupt can be masked by clearing the corresponding TMR0 Interrupt Enable bit (TOIE). The TMR0 Interrupt Flag bit (TOIF) is automatically cleared when vectoring to the TMR0 interrupt vector.

### 11.2 <u>Using Timer0 with External Clock</u>

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 ±4Tosc (±121 ns @ 33 MHz).

FIGURE 11-2: TIMERO MODULE BLOCK DIAGRAM

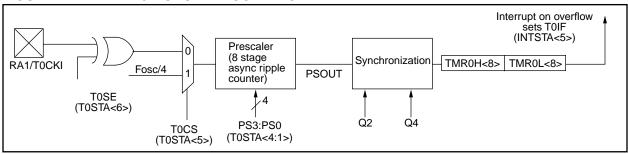
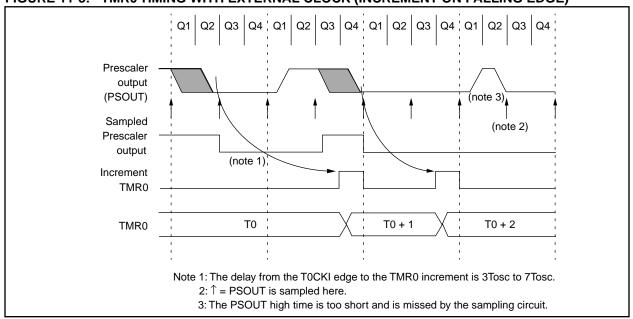


FIGURE 11-3: TMR0 TIMING WITH EXTERNAL CLOCK (INCREMENT ON FALLING EDGE)



#### 11.3 Read/Write Consideration for TMR0

Although TMR0 is a 16-bit timer/counter, only 8-bits at a time can be read or written during a single instruction cycle. Care must be taken during any read or write.

#### 11.3.1 READING 16-BIT VALUE

The problem in reading the entire 16-bit value is that after reading the low (or high) byte, its value may change from FFh to 00h.

Example 11-1 shows a 16-bit read. To ensure a proper read, interrupts must be disabled during this routine.

#### **EXAMPLE 11-1: 16-BIT READ**

MOVPF	TMROL,	TMPLO	;read low tmr0
MOVPF	TMROH,	TMPHI	;read high tmr0
MOVFP	TMPLO,	WREG	;tmplo -> wreg
CPFSLT	TMR0L		<pre>;tmr01 &lt; wreg?</pre>
RETURN			;no then return
MOVPF	TMROL,	TMPLO	;read low tmr0
MOVPF	TMROH,	TMPHI	;read high tmr0
RETURN			;return

#### 11.3.2 WRITING A 16-BIT VALUE TO TMR0

Since writing to either TMR0L or TMR0H will effectively inhibit increment of that half of the TMR0 in the next cycle (following write), but not inhibit increment of the other half, the user must write to TMR0L first and TMR0H next in two consecutive instructions, as shown in Example 11-2. The interrupt must be disabled. Any write to either TMR0L or TMR0H clears the prescaler.

#### **EXAMPLE 11-2: 16-BIT WRITE**

```
BSF CPUSTA, GLINTD ; Disable interrupt
MOVFP RAM_L, TMR0L ;
MOVFP RAM_H, TMR0H ;
BCF CPUSTA, GLINTD ; Done, enable interrupt
```

#### 11.4 Prescaler Assignments

Timer0 has an 8-bit prescaler. The prescaler assignment is fully under software control; i.e., it can be changed "on the fly" during program execution. When changing the prescaler assignment, clearing the prescaler is recommended before changing assignment. The value of the prescaler is "unknown," and assigning a value that is less then the present value makes it difficult to take this unknown time into account.

#### FIGURE 11-4: TMR0 TIMING: WRITE HIGH OR LOW BYTE

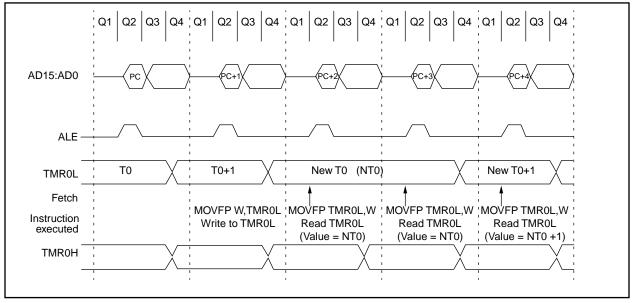


FIGURE 11-5: TMR0 READ/WRITE IN TIMER MODE

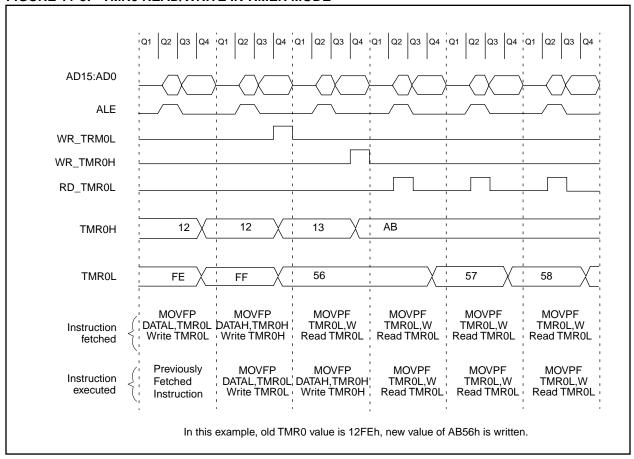


TABLE 11-1: REGISTERS/BITS ASSOCIATED WITH TIMERO

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	TOSTA	INTEDG	T0SE	T0CS	PS3	PS2	PS1	PS0	-	0000 000-	0000 000-
06h, Unbanked	CPUSTA	_	_	STKAV	GLINTD	TO	PD	_	-	11 11	11 qq
07h, Unbanked	INTSTA	PEIF	T0CKIF	TOIF	INTF	PEIE	T0CKIE	TOIE	INTE	0000 0000	0000 0000
0Bh, Unbanked	TMR0L	TMR0 reg	TMR0 register; low byte								uuuu uuuu
0Ch, Unbanked	TMR0H	TMR0 reg	TMR0 register; high byte								uuuu uuuu

Legend: x = unknown, u = unchanged, - = unimplemented read as a '0', g - 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.

# 12.0 TIMER1, TIMER2, TIMER3, PWMS AND CAPTURES

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

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

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

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

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

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

R/W - 0	R/W-0 R/W-0 R/W-0 R/W-0 R/W-0 R/W-0	
CA2ED <sup>2</sup>	CA2ED0 CA1ED1 CA1ED0 T16 TMR3CS TMR2CS TMR1CS	R = Readable bit
bit7	bit0	W = Writable bit -n = Value at POR reset
bit 7-6:	CA2ED1:CA2ED0: Capture2 Mode Select bits  00 = Capture on every falling edge  01 = Capture on every rising edge  10 = Capture on every 4th rising edge  11 = Capture on every 16th rising edge	
bit 5-4:	CA1ED1:CA1ED0: Capture1 Mode Select bits  00 = Capture on every falling edge  01 = Capture on every rising edge  10 = Capture on every 4th rising edge  11 = Capture on every 16th rising edge	
bit 3:	T16: Timer1:Timer2 Mode Select bit  1 = Timer1 and Timer2 form a 16-bit timer  0 = Timer1 and Timer2 are two 8-bit timers	
bit 2:	<b>TMR3CS</b> : Timer3 Clock Source Select bit 1 = TMR3 increments off the falling edge of the RB5/TCLK3 pin 0 = TMR3 increments off the internal clock	
bit 1:	TMR2CS: Timer2 Clock Source Select bit  1 = TMR2 increments off the falling edge of the RB4/TCLK12 pin  0 = TMR2 increments off the internal clock	
bit 0:	<b>TMR1CS</b> : Timer1 Clock Source Select bit 1 = TMR1 increments off the falling edge of the RB4/TCLK12 pin 0 = TMR1 increments off the internal clock	

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

R - 0 R - 0 R/W - 0 R/

bit7 bit0

R = Readable bit W = Writable bit

-n = Value at POR reset

bit 7: CA2OVF: Capture2 Overflow Status bit

This bit indicates that the capture value had not been read from the capture register pair (CA2H:CA2L) before the next capture event occurred. The capture register retains the oldest unread capture value (last capture before overflow). Subsequent capture events will not update the capture register with the Timer3 value until the capture register has been read (both bytes).

- 1 = Overflow occurred on Capture2 register
- 0 = No overflow occurred on Capture2 register
- bit 6: CA10VF: Capture1 Overflow Status bit

This bit indicates that the capture value had not been read from the capture register pair (PR3H/CA2H:PR3L/CA2L) before the next capture event occurred. The capture register retains the oldest unread capture value (last capture before overflow). Subsequent capture events will not update the capture register with the TMR3 value until the capture register has been read (both bytes).

- 1 = Overflow occurred on Capture1 register
- 0 = No overflow occurred on Capture1 register
- bit 5: **PWM2ON**: PWM2 On bit
  - 1 = PWM2 is enabled (The RB3/PWM2 pin ignores the state of the DDRB<3> bit)
  - 0 = PWM2 is disabled (The RB3/PWM2 pin uses the state of the DDRB<3> bit for data direction)
- bit 4: **PWM1ON**: PWM1 On bit
  - 1 = PWM1 is enabled (The RB2/PWM1 pin ignores the state of the DDRB<2> bit)
  - 0 = PWM1 is disabled (The RB2/PWM1 pin uses the state of the DDRB<2> bit for data direction)
- bit 3: CA1/PR3: CA1/PR3 Register Mode Select bit
  - 1 = Enables Capture1 (PR3H/CA1H:PR3L/CA1L is the Capture1 register. Timer3 runs without a period register)
  - 0 = Enables the Period register (PR3H/CA1H:PR3L/CA1L is the Period register for Timer3)
- bit 2: TMR3ON: Timer3 On bit
  - 1 = Starts Timer3
  - 0 = Stops Timer3
- bit 1: TMR2ON: Timer2 On bit

This bit controls the incrementing of the Timer2 register. When Timer2:Timer1 form the 16-bit timer (T16 is set), TMR2ON must be set. This allows the MSB of the timer to increment.

- 1 = Starts Timer2 (Must be enabled if the T16 bit (TCON1<3>) is set)
- 0 = Stops Timer2
- bit 0: TMR10N: Timer1 On bit

When T16 is set (in 16-bit Timer Mode)

- 1 = Starts 16-bit Timer2:Timer1
- 0 = Stops 16-bit Timer2:Timer1

#### When T16 is clear (in 8-bit Timer Mode)

- 1 = Starts 8-bit Timer1
- 0 = Stops 8-bit Timer1

#### 12.1 Timer1 and Timer2

#### 12.1.1 TIMER1, TIMER2 IN 8-BIT MODE

Both Timer1 and Timer2 will operate in 8-bit mode when the T16 bit is clear. These two timers can be independently configured to increment from the internal instruction cycle clock or from an external clock source on the RB4/TCLK12 pin. The timer clock source is configured by the TMRxCS bit (x = 1 for Timer1 or = 2 for Timer2). When TMRxCS is clear, the clock source is internal and increments once every instruction cycle (Fosc/4). When TMRxCS is set, the clock source is the RB4/TCLK12 pin, and the timer will increment on every falling edge of the RB4/TCLK12 pin.

The timer increments from 00h until it equals the Period register (PRx). It then resets to 00h at the next increment cycle. The timer interrupt flag is set when the timer is reset. TMR1 and TMR2 have individual interrupt flag bits. The TMR1 interrupt flag bit is latched into TMR1IF, and the TMR2 interrupt flag bit is latched into TMR2IF.

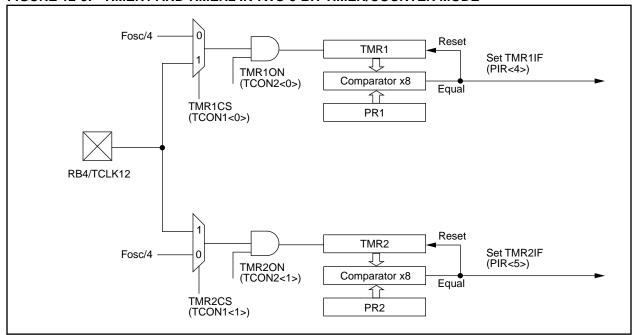
Each timer also has a corresponding interrupt enable bit (TMRxIE). The timer interrupt can be enabled by setting this bit and disabled by clearing this bit. For peripheral interrupts to be enabled, the Peripheral Interrupt Enable bit must be enabled (PEIE is set) and global interrupts must be enabled (GLINTD is cleared).

The timers can be turned on and off under software control. When the Timerx On control bit (TMRxON) is set, the timer increments from the clock source. When TMRxON is cleared, the timer is turned off and cannot cause the timer interrupt flag to be set.

### 12.1.1.1 EXTERNAL CLOCK INPUT FOR TIMER1 OR TIMER2

When TMRxCS is set, the clock source is the RB4/TCLK12 pin, and the timer will increment on every falling edge on the RB4/TCLK12 pin. The TCLK12 input is synchronized with internal phase clocks. This causes a delay from the time a falling edge appears on TCLK12 to the time TMR1 or TMR2 is actually incremented. For the external clock input timing requirements, see the Electrical Specification section.

FIGURE 12-3: TIMER1 AND TIMER2 IN TWO 8-BIT TIMER/COUNTER MODE



#### 12.1.2 TIMER1 & TIMER2 IN 16-BIT MODE

To select 16-bit mode, the T16 bit must be set. In this mode TMR1 and TMR2 are concatenated to form a 16-bit timer (TMR2:TMR1). The 16-bit timer increments until it matches the 16-bit period register (PR2:PR1). On the following timer clock, the timer value is reset to 0h, and the TMR1IF bit is set.

When selecting the clock source for the 16-bit timer, the TMR1CS bit controls the entire 16-bit timer and TMR2CS is a "don't care." When TMR1CS is clear, the timer increments once every instruction cycle (Fosc/4). When TMR1CS is set, the timer increments on every falling edge of the RB4/TCLK12 pin. For the 16-bit timer to increment, both TMR1ON and TMR2ON bits must be set (Table 12-1).

### 12.1.2.1 EXTERNAL CLOCK INPUT FOR TMR1:TMR2

When TMR1CS is set, the 16-bit TMR2:TMR1 increments on the falling edge of clock input TCLK12. The input on the RB4/TCLK12 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 RB4/TCLK12 to the time TMR2:TMR1 is actually incremented. For the external clock input timing requirements, see the Electrical Specification section.

TABLE 12-1: TURNING ON 16-BIT TIMER

TMR2ON	TMR10N	Result
1	1	16-bit timer (TMR2:TMR1) ON
0	1	Only TMR1 increments
х	0	16-bit timer OFF

FIGURE 12-4: TMR1 AND TMR2 IN 16-BIT TIMER/COUNTER MODE

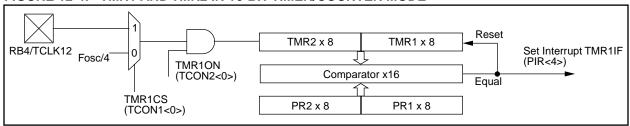


TABLE 12-2: SUMMARY OF TIMER1 AND TIMER2 REGISTERS

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 3	TCON1	CA2ED1	CA2ED0	CA1ED1	CA1ED0	T16	TMR3CS	TMR2CS	TMR1CS	0000 0000	0000 0000
17h, Bank 3	TCON2	CA2OVF	CA10VF	PWM2ON	PWM10N	CA1/PR3	TMR3ON	TMR2ON	TMR10N	0000 0000	0000 0000
10h, Bank 2	TMR1	Timer1 re	gister							xxxx xxxx	uuuu uuuu
11h, Bank 2	TMR2	Timer2 re	gister							xxxx xxxx	uuuu uuuu
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
07h, Unbanked	INTSTA	PEIF	T0CKIF	TOIF	INTF	PEIE	T0CKIE	TOIE	INTE	0000 0000	0000 0000
06h, Unbanked	CPUSTA	_	_	STKAV	GLINTD	TO	PD	_	_	11 11	11 qq
14h, Bank 2	PR1	Timer1 pe	riod registe	r						xxxx xxxx	uuuu uuuu
15h, Bank 2	PR2	Timer2 pe	riod registe	r						xxxx xxxx	uuuu uuuu
10h, Bank 3	PW1DCL	DC1	DC0	_	_	_	_	_	_	xx	uu
11h, Bank 3	PW2DCL	DC1	DC0	TM2PW2	_	_	_	_	_	xx0	uu0
12h, Bank 3	PW1DCH	DC9	DC8	DC7	DC6	DC5	DC4	DC3	DC2	xxxx xxxx	uuuu uuuu
13h, Bank 3	PW2DCH	DC9	DC8	DC7	DC6	DC5	DC4	DC3	DC2	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 Timer1 or Timer2.

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

### 12.1.3 USING PULSE WIDTH MODULATION (PWM) OUTPUTS WITH TMR1 AND TMR2

Two high speed pulse width modulation (PWM) outputs are provided. The PWM1 output uses Timer1 as its time-base, while PWM2 may be software configured to use either Timer1 or Timer2 as the time-base. The PWM outputs are on the RB2/PWM1 and RB3/PWM2 pins.

Each PWM output has a maximum resolution of 10-bits. At 10-bit resolution, the PWM output frequency is 24.4 kHz (@ 25 MHz clock) and at 8-bit resolution the PWM output frequency is 97.7 kHz. The duty cycle of the output can vary from 0% to 100%.

Figure 12-5 shows a simplified block diagram of the PWM module. The duty cycle register is double buffered for glitch free operation. Figure 12-6 shows how a glitch could occur if the duty cycle registers were not double buffered.

The user needs to set the PWM1ON bit (TCON2<4>) to enable the PWM1 output. When the PWM1ON bit is set, the RB2/PWM1 pin is configured as PWM1 output and forced as an output irrespective of the data direction bit (DDRB<2>). When the PWM1ON bit is clear,

#### 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:

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:

$$= \frac{\log\left(\frac{FOSC}{FPWM}\right)}{\log(2)}$$
 bits

The PWMx duty cycle is as follows:

PWMx Duty Cycle = (DCx) x Tosc

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 then 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 (kHz)									
Frequency	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).

### 12.1.3.3.1 MAX RESOLUTION/FREQUENCY FOR EXTERNAL CLOCK INPUT

The use of an external clock for the PWM time-base (Timer1 or Timer2) limits the PWM output to a maximum resolution of 8-bits. The PWxDCL<7:6> bits must be kept cleared. Use of any other value will distort the PWM output. All resolutions are supported when internal clock mode is selected. The maximum attainable frequency is also lower. This is a result of the timing requirements of an external clock input for a timer (see the Electrical Specification section). The maximum PWM frequency, when the timers clock source is the RB4/TCLK12 pin, is shown in Table 12-3 (standard resolution mode).

#### 12.2 Timer3

Timer3 is a 16-bit timer consisting of the TMR3H and TMR3L registers. TMR3H is the high byte of the timer and TMR3L is the low byte. This timer has an associated 16-bit period register (PR3H/CA1H:PR3L/CA1L). This period register can be software configured to be a second 16-bit capture register.

When the TMR3CS bit (TCON1<2>) is clear, the timer increments every instruction cycle (Fosc/4). When TMR3CS is set, the timer increments on every falling edge of the RB5/TCLK3 pin. In either mode, the TMR3ON bit must be set for the timer to increment. When TMR3ON is clear, the timer will not increment or set the TMR3IF bit.

Timer3 has two modes of operation, depending on the CA1/PR3 bit (TCON2<3>). These modes are:

- One capture and one period register mode
- · Dual capture register mode

The PIC17C4X has up to two 16-bit capture registers that capture the 16-bit value of TMR3 when events are detected on capture pins. There are two capture pins (RB0/CAP1 and RB1/CAP2), one for each capture register. The capture pins are multiplexed with PORTB pins. An event can be:

- · a rising edge
- · a falling edge
- every 4th rising edge
- · every 16th rising edge

Each 16-bit capture register has an interrupt flag associated with it. The flag is set when a capture is made. The capture module is truly part of the Timer3 block. Figure 12-7 and Figure 12-8 show the block diagrams for the two modes of operation.

TABLE 12-4: REGISTERS/BITS ASSOCIATED WITH PWM

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 3	TCON1	CA2ED1	CA2ED0	CA1ED1	CA1ED0	T16	TMR3CS	TMR2CS	TMR1CS	0000 0000	0000 0000
17h, Bank 3	TCON2	CA2OVF	CA10VF	PWM2ON	PWM1ON	CA1/PR3	TMR3ON	TMR2ON	TMR10N	0000 0000	0000 0000
10h, Bank 2	TMR1	Timer1 reg	ister							xxxx xxxx	uuuu uuuu
11h, Bank 2	TMR2	Timer2 reg	ister							xxxx xxxx	uuuu uuuu
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
07h, Unbanked	INTSTA	PEIF	T0CKIF	T0IF	INTF	PEIE	T0CKIE	TOIE	INTE	0000 0000	0000 0000
06h, Unbanked	CPUSTA	_	_	STKAV	GLINTD	TO	PD	_	_	11 11	11 qq
10h, Bank 3	PW1DCL	DC1	DC0	_	_	_	_	_	_	xx	uu
11h, Bank 3	PW2DCL	DC1	DC0	TM2PW2	_	_	_	-	_	xx0	uu0
12h, Bank 3	PW1DCH	DC9	DC8	DC7	DC6	DC5	DC4	DC3	DC2	xxxx xxxx	uuuu uuuu
13h, Bank 3	PW2DCH	DC9	DC8	DC7	DC6	DC5	DC4	DC3	DC2	xxxx xxxx	uuuu uuuu

Legend: x = unknown, u = unchanged, - = unimplemented read as '0', <math>q = value depends on conditions, shaded cells are not used by PWM.

### 12.2.1 ONE CAPTURE AND ONE PERIOD REGISTER MODE

In this mode registers PR3H/CA1H and PR3L/CA1L constitute a 16-bit period register. A block diagram is shown in Figure 12-7. The timer increments until it equals the period register and then resets to 0000h. TMR3 Interrupt Flag bit (TMR3IF) is set at this point. This interrupt can be disabled by clearing the TMR3 Interrupt Enable bit (TMR3IE). TMR3IF must be cleared in software.

This mode is selected if control bit CA1/PR3 is clear. In this mode, the Capture1 register, consisting of high byte (PR3H/CA1H) and low byte (PR3L/CA1L), is configured as the period control register for TMR3. Capture1 is disabled in this mode, and the corresponding Interrupt bit CA1IF is never set. TMR3 increments until it equals the value in the period register and then resets to 0000h.

Capture2 is active in this mode. The CA2ED1 and CA2ED0 bits determine the event on which capture will occur. The possible events are:

- · Capture on every falling edge
- · Capture on every rising edge
- · Capture every 4th rising edge
- · Capture every 16th rising edge

When a capture takes place, an interrupt flag is latched into the CA2IF bit. This interrupt can be enabled by setting the corresponding mask bit CA2IE. The Peripheral Interrupt Enable bit (PEIE) must be set and the Global Interrupt Disable bit (GLINTD) must be cleared for the interrupt to be acknowledged. The CA2IF interrupt flag bit must be cleared in software.

When the capture prescale select is changed, the prescaler is not reset and an event may be generated. Therefore, the first capture after such a change will be ambiguous. However, it sets the time-base for the next capture. The prescaler is reset upon chip reset.

Capture pin RB1/CAP2 is a multiplexed pin. When used as a port pin, Capture2 is not disabled. However, the user can simply disable the Capture2 interrupt by clearing CA2IE. If RB1/CAP2 is used as an output pin, the user can activate a capture by writing to the port pin. This may be useful during development phase to emulate a capture interrupt.

The input on capture pin RB1/CAP2 is synchronized internally to internal phase clocks. This imposes certain restrictions on the input waveform (see the Electrical Specification section for timing).

The Capture2 overflow status flag bit is double buffered. The master bit is set if one captured word is already residing in the Capture2 register and another "event" has occurred on the RB1/CA2 pin. The new event will not transfer the Timer3 value to the capture register, protecting the previous unread capture value. When the user reads both the high and the low bytes (in any order) of the Capture2 register, the master overflow bit is transferred to the slave overflow bit (CA2OVF) and then the master bit is reset. The user can then read TCON2 to determine the value of CA2OVF.

The recommended sequence to read capture registers and capture overflow flag bits is shown in Example 12-1.

### EXAMPLE 12-1: SEQUENCE TO READ CAPTURE REGISTERS

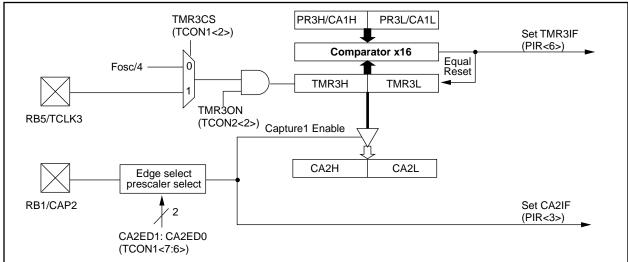
MOVLB 3 ;Select Bank 3

MOVPF CA2L,LO\_BYTE ;Read Capture2 low
;byte, store in LO\_BYTE

MOVPF CA2H,HI\_BYTE ;Read Capture2 high
;byte, store in HI\_BYTE

MOVPF TCON2,STAT\_VAL ;Read TCON2 into file
;STAT\_VAL

FIGURE 12-7: TIMER3 WITH ONE CAPTURE AND ONE PERIOD REGISTER BLOCK DIAGRAM



#### 12.2.2 DUAL CAPTURE REGISTER MODE

This mode is selected by setting CA1/PR3. A block diagram is shown in Figure 12-8. In this mode, TMR3 runs without a period register and increments from 0000h to FFFFh and rolls over to 0000h. The TMR3 interrupt Flag (TMR3IF) is set on this roll over. The TMR3IF bit must be cleared in software.

Registers PR3H/CA1H and PR3L/CA1L make a 16-bit capture register (Capture1). It captures events on pin RB0/CAP1. Capture mode is configured by the CA1ED1 and CA1ED0 bits. Capture1 Interrupt Flag bit (CA1IF) is set on the capture event. The corresponding interrupt mask bit is CA1IE. The Capture1 Overflow Status bit is CA1OVF.

The Capture2 overflow status flag bit is double buffered. The master bit is set if one captured word is already residing in the Capture2 register and another "event" has occurred on the RB1/CA2 pin. The new event will not transfer the TMR3 value to the capture register which protects the previous unread capture value. When the user reads both the high and the low bytes (in any order) of the Capture2 register, the master overflow bit is transferred to the slave overflow bit (CA2OVF) and then the master bit is reset. The user can then read TCON2 to determine the value of CA2OVF.

The operation of the Capture1 feature is identical to Capture2 (as described in Section 12.2.1).

FIGURE 12-8: TIMER3 WITH TWO CAPTURE REGISTERS BLOCK DIAGRAM

TABLE 12-5: REGISTERS ASSOCIATED WITH CAPTURE

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 3	TCON1	CA2ED1	CA2ED0	CA1ED1	CA1ED0	T16	TMR3CS	TMR2CS	TMR1CS	0000 0000	0000 0000
17h, Bank 3	TCON2	CA2OVF	CA10VF	PWM2ON	PWM1ON	CA1/PR3	TMR3ON		TMR10N	0000 0000	0000 0000
12h, Bank 2	TMR3L	TMR3 reg	ister; low by	yte						xxxx xxxx	uuuu uuuu
13h, Bank 2	TMR3H	TMR3 reg	ister; high b	oyte						xxxx xxxx	uuuu uuuu
16h, Bank 1	PIR	RBIF	TMR3IF	TMR2IF	TMR1IF	CA2IF	CA1IF	TXIF		0000 0010	0000 0010
17h, Bank 1	PIE		TMR3IE	TMR2IE	TMR1IE	CA2IE	CA1IE	TXIE	RCIE	0000 0000	0000 0000
07h, Unbanked	INTSTA	PEIF	T0CKIF			PEIE					

#### 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.

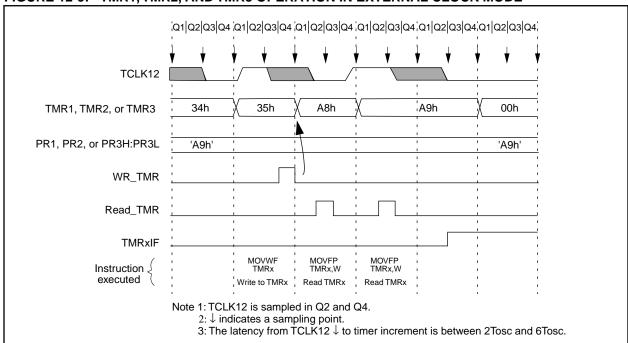
#### **EXAMPLE 12-2: WRITING TO TMR3**

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**

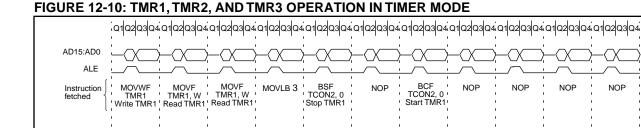
MOVPF	TMR3L,	TMPLO	read low tmr0
MOVPF	TMR3H,	TMPHI	;read high tmr0
MOVFP	TMPLO,	WREG	;tmplo -> wreg
CPFSLT	TMR3L,	WREG	<pre>;tmr01 &lt; wreg?</pre>
RETURN			;no then return
MOVPF	TMR3L,	TMPLO	;read low tmr0
MOVPF	TMR3H,	TMPHI	<pre>;read high tmr0</pre>
RETURN			;return

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



08h

00h



05h

06h

07h

04h

TMR1 reads 04h

TABLE 12-6: SUMMARY OF TMR1, TMR2, AND TMR3 REGISTERS

TMR1 reads 03h

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 3	TCON1	CA2ED1	CA2ED0	CA1ED1	CA1ED0	T16	TMR3CS	TMR2CS	TMR1CS	0000 0000	0000 0000
17h, Bank 3	TCON2	CA2OVF	CA10VF	PWM2ON	PWM10N	CA1/PR3	TMR3ON	TMR2ON	TMR10N	0000 0000	0000 0000
10h, Bank 2	TMR1	Timer1 re	gister							xxxx xxxx	uuuu uuuu
11h, Bank 2	TMR2	Timer2 re	gister							xxxx xxxx	uuuu uuuu
12h, Bank 2	TMR3L	TMR3 reg	ister; low by	rte .						xxxx xxxx	uuuu uuuu
13h, Bank 2	TMR3H	TMR3 reg	ister; high b	yte						xxxx xxxx	uuuu uuuu
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
07h, Unbanked	INTSTA	PEIF	T0CKIF	TOIF	INTF	PEIE	T0CKIE	TOIE	INTE	0000 0000	0000 0000
06h, Unbanked	CPUSTA	_	_	STKAV	GLINTD	TO	PD	_	_	11 11	11 qq
14h, Bank 2	PR1	Timer1 pe	riod registe	r						xxxx xxxx	uuuu uuuu
15h, Bank 2	PR2	Timer2 pe	riod registe	r						xxxx xxxx	uuuu uuuu
16h, Bank 2	PR3L/CA1L	Timer3 pe	riod/capture	e1 register; l	ow byte					xxxx xxxx	uuuu uuuu
17h, Bank 2	PR3H/CA1H	Timer3 pe	riod/capture	e1 register; l	high byte					xxxx xxxx	uuuu uuuu
10h, Bank 3	PW1DCL	DC1	DC0	_	_	_	_	_	_	xx	uu
11h, Bank 3	PW2DCL	DC1	DC0	TM2PW2	_	_	_	_	_	xx0	uu0
12h, Bank 3	PW1DCH	DC9	DC8	DC7	DC6	DC5	DC4	DC3	DC2	xxxx xxxx	uuuu uuuu
13h, Bank 3	PW2DCH	DC9	DC8	DC7	DC6	DC5	DC4	DC3	DC2	xxxx xxxx	uuuu uuuu
14h, Bank 3	CA2L	Capture2	low byte	xxxx xxxx	uuuu uuuu						
15h, Bank 3	CA2H	Capture2	high byte							xxxx xxxx	uuuu uuuu

Legend: x = unknown, u = unchanged, - = unimplemented read as '0', <math>q - value depends on condition, shaded cells are not used by TMR1, TMR2 or TMR3.

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

TMR1

PR1

TMR10N

WR\_TMR1

WR\_TCON2

TMR1IF

RD\_TMR1

04h

05h

03h

## PIC17C4X

NOTES:

# 13.0 UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER (USART) MODULE

The USART module is a serial I/O module. The USART can be configured as a full duplex asynchronous system that can communicate with peripheral devices such as CRT terminals and personal computers, or it can be configured as a half duplex synchronous system that can communicate with peripheral devices such as A/D or D/A integrated circuits, Serial EEPROMs etc. The USART can be configured in the following modes:

- Asynchronous (full duplex)
- Synchronous Master (half duplex)
- Synchronous Slave (half duplex)

The SPEN (RCSTA<7>) bit has to be set in order to configure RA4 and RA5 as the Serial Communication Interface.

The USART module will control the direction of the RA4/RX/DT and RA5/TX/CK pins, depending on the states of the USART configuration bits in the RCSTA and TXSTA registers. The bits that control I/O direction are:

- SPEN
- TXEN
- SREN
- CREN
- CSRC

The Transmit Status And Control Register is shown in Figure 13-1, while the Receive Status And Control Register is shown in Figure 13-2.

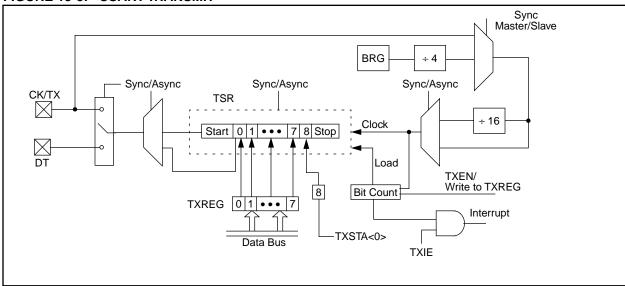
#### FIGURE 13-1: TXSTA REGISTER (ADDRESS: 15h, BANK 0)

<u>R/W - 0</u> CSRC	R/W - 0 TX9	R/W - 0 TXEN	R/W - 0 SYNC	<u>U - 0</u>	U - 0 —	R - 1 TRMT	R/W - x TX9D	R = Readable bit
oit7	11.15						bit0	W = Writable bit -n = Value at POR reset (x = unknown)
oit 7:	CSRC: C Synchron 1 = Maste 0 = Slave Asynchron Don't care	ous mode r Mode (C mode (Clo nous mod	lock gene	rated inte	rnally from I	BRG)		
oit 6:	<b>TX9</b> : 9-bit 1 = Select 0 = Select	s 9-bit tra	nsmission					
oit 5:	TXEN: Tra 1 = Transr 0 = Transr SREN/CR	nit enable nit disable	d d	in SYNC	mode			
oit 4:	SYNC: US (Synchror 1 = Synch 0 = Async	nous/Asyn ronous m	chronous) ode					
oit 3-2:	Unimplen	nented: R	ead as '0'					
oit 1:	<b>TRMT</b> : Tra 1 = TSR e 0 = TSR fi	mpty	ft Register	(TSR) E	mpty bit			
oit 0:	TYQD: Oth	hit of tran	emit data	(can he i	sed to calci	ulated the	parity in so	ftware)

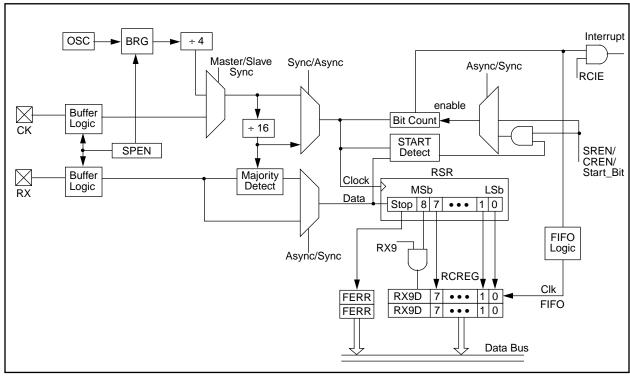
### FIGURE 13-2: RCSTA REGISTER (ADDRESS: 13h, BANK 0)

R/W - (	R/W - 0         R/W - 0         U - 0         R - 0         R - 0         R - x           RX9         SREN         CREN         —         FERR         OERR         RX9D         R = Readable bit
bit7	bit 0 W = Writable bit -n = Value at POR reset (x = unknown)
bit 7:	SPEN: Serial Port Enable bit  1 = Configures RA5/RX/DT and RA4/TX/CK pins as serial port pins  0 = Serial port disabled
bit 6:	RX9: 9-bit Receive Enable bit  1 = Selects 9-bit reception  0 = Selects 8-bit reception
bit 5:	SREN: Single Receive Enable bit This bit enables the reception of a single byte. After receiving the byte, this bit is automatically cleared.  Synchronous mode:  1 = Enable reception 0 = Disable reception Note: This bit is ignored in synchronous slave reception.  Asynchronous mode: Don't care
bit 4:	CREN: Continuous Receive Enable bit This bit enables the continuous reception of serial data.  Asynchronous mode:  1 = Enable reception 0 = Disables reception  Synchronous mode: 1 = Enables continuous reception until CREN is cleared (CREN overrides SREN) 0 = Disables continuous reception
bit 3:	Unimplemented: Read as '0'
bit 2:	FERR: Framing Error bit  1 = Framing error (Updated by reading RCREG)  0 = No framing error
bit 1:	OERR: Overrun Error bit  1 = Overrun (Cleared by clearing CREN)  0 = No overrun error
bit 0:	<b>RX9D</b> : 9th bit of receive data (can be the software calculated parity bit)

FIGURE 13-3: USART TRANSMIT



#### FIGURE 13-4: USART RECEIVE



#### 13.1 <u>USART Baud Rate Generator (BRG)</u>

The BRG supports both the Asynchronous and Synchronous modes of the USART. It is a dedicated 8-bit baud rate generator. The SPBRG register controls the period of a free running 8-bit timer. Table 13-1 shows the formula for computation of the baud rate for different USART modes. These only apply when the USART is in synchronous master mode (internal clock) and asynchronous mode.

Given the desired baud rate and Fosc, the nearest integer value between 0 and 255 can be calculated using the formula below. The error in baud rate can then be determined.

TABLE 13-1: BAUD RATE FORMULA

SYNC	Mode	Baud Rate
0	Asynchronous	Fosc/(64(X+1))
1	Synchronous	Fosc/(4(X+1))

X = value in SPBRG (0 to 255)

Example 13-1 shows the calculation of the baud rate error for the following conditions:

FOSC = 16 MHz Desired Baud Rate = 9600 SYNC = 0

### EXAMPLE 13-1: CALCULATING BAUD RATE ERROR

Desired Baud rate=Fosc / (64 (X + 1))

9600 = 16000000 / (64 (X + 1))

X = 25.042 = 25

Calculated Baud Rate=16000000 / (64 (25 + 1))

= 9615

Error = (Calculated Baud Rate - Desired Baud Rate)

Desired Baud Rate

= (9615 - 9600) / 9600

= 0.16%

Writing a new value to the SPBRG, causes the BRG timer to be reset (or cleared), this ensures that the BRG does not wait for a timer overflow before outputting the new baud rate.

#### TABLE 13-2: REGISTERS ASSOCIATED WITH BAUD RATE GENERATOR

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)
13h, Bank 0	RCSTA	SPEN	RX9	SREN	CREN	_	FERR	OERR	RX9D	0000 -00x	0000 -00u
15h, Bank 0	TXSTA	CSRC	TX9	TXEN	SYNC	_	_	TRMT	TX9D	00001x	00001u
17h, Bank 0	SPBRG	Baud rate	Baud rate generator register								uuuu uuuu

Legend: x = unknown, u = unchanged, - = unimplemented read as a '0', shaded cells are not used by the Baud Rate Generator.

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

TABLE 13-3: BAUD RATES FOR SYNCHRONOUS MODE

BAUD RATE	Fosc = 3	3 MHz	SPBRG value	Fosc = 2	5 MHz	SPBRG value	FOSC = 20	0 MHz	SPBRG value	FOSC = 10	6 MHz	SPBRG value
(K)	KBAUD	%ERROR	(decimal)	KBAUD	%ERROR	(decimal)	KBAUD	%ERROR	(decimal)	KBAUD	%ERROR	(decimal)
0.3	NA	_	_	NA	_	_	NA	_	_	NA	_	_
1.2	NA	_	_	NA	_	_	NA	_	_	NA	_	_
2.4	NA	_	_	NA	_	_	NA	_	_	NA	_	-
9.6	NA	_	_	NA	_	_	NA	_	_	NA	_	-
19.2	NA	_	_	NA	_	_	19.53	+1.73	255	19.23	+0.16	207
76.8	77.10	+0.39	106	77.16	+0.47	80	76.92	+0.16	64	76.92	+0.16	51
96	95.93	-0.07	85	96.15	+0.16	64	96.15	+0.16	51	95.24	-0.79	41
300	294.64	-1.79	27	297.62	-0.79	20	294.1	-1.96	16	307.69	+2.56	12
500	485.29	-2.94	16	480.77	-3.85	12	500	0	9	500	0	7
HIGH	8250	_	0	6250	_	0	5000	_	0	4000	_	0
LOW	32.22	_	255	24.41	_	255	19.53	_	255	15.625	_	255

BAUD	FOSC = 10 MI	Hz	SPBRG	FOSC = 7.159	) MHz	SPBRG	FOSC = 5.068	3 MHz	SPBRG
RATE (K)	KBAUD	%ERROR	value (decimal)	KBAUD	%ERROR	value (decimal)	KBAUD	%ERROR	value (decimal)
0.3	NA	_	_	NA	_	_	NA	_	_
1.2	NA	_	_	NA	_	_	NA	_	_
2.4	NA	_	_	NA	_	_	NA	_	_
9.6	9.766	+1.73	255	9.622	+0.23	185	9.6	0	131
19.2	19.23	+0.16	129	19.24	+0.23	92	19.2	0	65
76.8	75.76	-1.36	32	77.82	+1.32	22	79.2	+3.13	15
96	96.15	+0.16	25	94.20	-1.88	18	97.48	+1.54	12
300	312.5	+4.17	7	298.3	-0.57	5	316.8	+5.60	3
500	500	0	4	NA	_	_	NA	_	_
HIGH	2500	_	0	1789.8	_	0	1267	_	0
LOW	9.766	_	255	6.991	_	255	4.950	_	255

BAUD	Fosc = 3.579	MHz	SPBRG	FOSC = 1 MHz		SPBRG	FOSC = 32.76	8 kHz	SPBRG
RATE (K)	KBAUD	%ERROR	value (decimal)	KBAUD	%ERROR	value (decimal)	KBAUD	%ERROR	value (decimal)
0.3	NA	_	_	NA	_	_	0.303	+1.14	26
1.2	NA	_	_	1.202	+0.16	207	1.170	-2.48	6
2.4	NA	_	_	2.404	+0.16	103	NA	_	_
9.6	9.622	+0.23	92	9.615	+0.16	25	NA	_	_
19.2	19.04	-0.83	46	19.24	+0.16	12	NA	_	_
76.8	74.57	-2.90	11	83.34	+8.51	2	NA	_	_
96	99.43	_3.57	8	NA	_	_	NA	_	_
300	298.3	-0.57	2	NA	_	_	NA	_	_
500	NA	_	_	NA	_	_	NA	_	_
HIGH	894.9	_	0	250	_	0	8.192	_	0
LOW	3.496	_	255	0.976	_	255	0.032	_	255

TABLE 13-4: BAUD RATES FOR ASYNCHRONOUS MODE

BAUD RATE	Fosc = 3	3 MHz	SPBRG value	FOSC = 2	5 MHz	SPBRG value	FOSC = 2	0 MHz	SPBRG value	FOSC = 1	6 MHz	SPBRG value
(K)	KBAUD	%ERROR	(decimal)									
0.3	NA	_	_									
1.2	NA	_	_	NA	_	_	1.221	+1.73	255	1.202	+0.16	207
2.4	2.398	-0.07	214	2.396	0.14	162	2.404	+0.16	129	2.404	+0.16	103
9.6	9.548	-0.54	53	9.53	-0.76	40	9.469	-1.36	32	9.615	+0.16	25
19.2	19.09	-0.54	26	19.53	+1.73	19	19.53	+1.73	15	19.23	+0.16	12
76.8	73.66	-4.09	6	78.13	+1.73	4	78.13	+1.73	3	83.33	+8.51	2
96	103.12	+7.42	4	97.65	+1.73	3	104.2	+8.51	2	NA	_	-
300	257.81	-14.06	1	390.63	+30.21	0	312.5	+4.17	0	NA	_	-
500	515.62	+3.13	0	NA	_	_	NA	_	_	NA	_	-
HIGH	515.62	_	0	-	_	0	312.5	_	0	250	_	0
LOW	2.014	_	255	1.53	_	255	1.221	_	255	0.977	_	255

BAUD	Fosc = 10 MH	łz	SPBRG	FOSC = 7.159	) MHz	SPBRG	FOSC = 5.068	3 MHz	SPBRG
RATE (K)	KBAUD	%ERROR	value (decimal)	KBAUD	%ERROR	value (decimal)	KBAUD	%ERROR	value (decimal)
0.3	NA	_	_	NA	_	_	0.31	+3.13	255
1.2	1.202	+0.16	129	1.203	_0.23	92	1.2	0	65
2.4	2.404	+0.16	64	2.380	-0.83	46	2.4	0	32
9.6	9.766	+1.73	15	9.322	-2.90	11	9.9	-3.13	7
19.2	19.53	+1.73	7	18.64	-2.90	5	19.8	+3.13	3
76.8	78.13	+1.73	1	NA	_	_	79.2	+3.13	0
96	NA	_	_	NA	_	_	NA	_	_
300	NA	_	_	NA	_	_	NA	_	_
500	NA	_	_	NA	_	_	NA	_	_
HIGH	156.3	_	0	111.9	_	0	79.2	_	0
LOW	0.610	_	255	0.437	_	255	0.309	_	2 <b>55</b>

BAUD RATE	Fosc = 3.579	MHz	SPBRG value	FOSC = 1 MHz	Z	SPBRG value	FOSC = 32.76	88 kHz	SPBRG value
(K)	KBAUD	%ERROR	(decimal)	KBAUD	%ERROR	(decimal)	KBAUD	%ERROR	(decimal)
0.3	0.301	+0.23	185	0.300	+0.16	51	0.256	-14.67	1
1.2	1.190	-0.83	46	1.202	+0.16	12	NA	_	_
2.4	2.432	+1.32	22	2.232	-6.99	6	NA	_	_
9.6	9.322	-2.90	5	NA	_	_	NA	_	_
19.2	18.64	-2.90	2	NA	_	_	NA	_	_
76.8	NA	_	_	NA	_	_	NA	_	_
96	NA	_	_	NA	_	_	NA	_	_
300	NA	_	_	NA	_	_	NA	_	_
500	NA	_	_	NA	_	_	NA	_	_
HIGH	55.93	_	0	15.63	_	0	0.512	_	0
LOW	0.218	_	255	0.061	_	255	0.002	_	255

#### 13.2 <u>USART Asynchronous Mode</u>

In this mode, the USART uses standard nonreturn-to-zero (NRZ) format (one start bit, eight or nine data bits, and one stop bit). The most common data format is 8-bits. An on-chip dedicated 8-bit baud rate generator can be used to derive standard baud rate frequencies from the oscillator. The USART's transmitter and receiver are functionally independent but use the same data format and baud rate. The baud rate generator produces a clock x64 of the bit shift rate. Parity is not supported by the hardware, but can be implemented in software (and stored as the ninth data bit). Asynchronous mode is stopped during SLEEP.

The asynchronous mode is selected by clearing the SYNC bit (TXSTA<4>).

The USART Asynchronous module consists of the following important elements:

- · Baud Rate Generator
- Sampling Circuit
- · Asynchronous Transmitter
- · Asynchronous Receiver

#### 13.2.1 USART ASYNCHRONOUS TRANSMITTER

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

**Note:** The TSR is not mapped in data memory, so it is not available to the user.

Transmission enabled setting by TXEN (TXSTA<5>) bit. The actual transmission will not occur until TXREG has been loaded with data and the baud rate generator (BRG) has produced a shift clock (Figure 13-5). The transmission can also be started by first loading TXREG and then setting TXEN. Normally when transmission is first started, the TSR is empty, so a transfer to TXREG will result in an immediate transfer to TSR resulting in an empty TXREG. A back-to-back transfer is thus possible (Figure 13-6). Clearing TXEN during a transmission will cause the transmission to be aborted. This will reset the transmitter and the RA5/TX/CK pin will revert to hi-impedance.

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

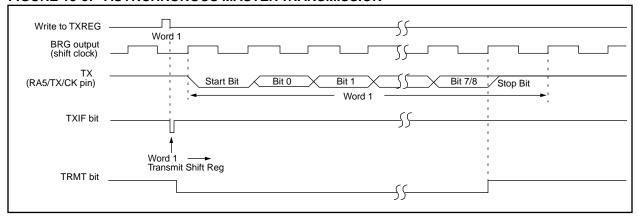
Steps to follow when setting up an Asynchronous Transmission:

- Initialize the SPBRG register for the appropriate baud rate.
- Enable the asynchronous serial port by clearing the SYNC bit and setting the SPEN bit.
- 3. If interrupts are desired, then set the TXIE bit.
- If 9-bit transmission is desired, then set the TX9 bit.
- Load data to the TXREG register.
- 6. If 9-bit transmission is selected, the ninth bit should be loaded in TX9D.
- 7. Enable the transmission by setting TXEN (starts transmission).

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

#### FIGURE 13-5: ASYNCHRONOUS MASTER TRANSMISSION



### FIGURE 13-6: ASYNCHRONOUS MASTER TRANSMISSION (BACK TO BACK)

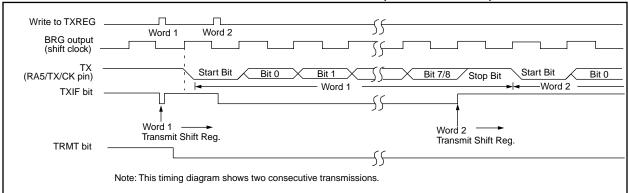


TABLE 13-5: REGISTERS ASSOCIATED WITH ASYNCHRONOUS TRANSMISSION

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
16h, Bank 0	TXREG	Serial port	transmit re	egister						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	00001x	00001u
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 transmission.

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

#### 13.2.2 USART ASYNCHRONOUS RECEIVER

The receiver block diagram is shown in Figure 13-4. The data comes in the RA4/RX/DT pin and drives the data recovery block. The data recovery block is actually a high speed shifter operating at 16 times the baud rate, whereas the main receive serial shifter operates at the bit rate or at Fosc.

Once asynchronous mode is selected, reception is enabled by setting bit CREN (RCSTA<4>).

The heart of the receiver is the receive (serial) shift register (RSR). After sampling the stop bit, the received data in the RSR is transferred to the RCREG (if it is empty). If the transfer is complete, the interrupt bit RCIF (PIR<0>) is set. The actual interrupt can be enabled/disabled by setting/clearing the RCIE (PIE<0>) bit. RCIF is a read only bit which is cleared by the hardware. It is cleared when RCREG has been read and is empty. RCREG is a double buffered register; (i.e. it is a two deep FIFO). It is possible for two bytes of data to be received and transferred to the RCREG FIFO and a third byte begin shifting to the RSR. On detection of the stop bit of the third byte, if the RCREG is still full, then the overrun error bit, OERR (RCSTA<1>) will be set. The word in the RSR will be lost. RCREG can be read twice to retrieve the two bytes in the FIFO. The OERR bit has to be cleared in software which is done by resetting the receive logic (CREN is set). If the OERR bit is set, transfers from the RSR to RCREG are inhibited, so it is essential to clear the OERR bit if it is set. The framing error bit FERR (RCSTA<2>) is set if a stop bit is not detected.

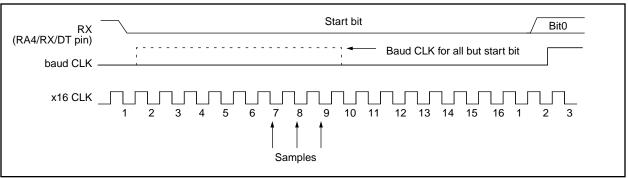
Note: The FERR and the 9th receive bit are buffered the same way as the receive data. Reading the RCREG register will allow the RX9D and FERR bits to be loaded with values for the next received Received data; therefore, it is essential for the user to read the RCSTA register before reading RCREG in order not to lose the old FERR and RX9D information.

#### 13.2.3 SAMPLING

The data on the RA4/RX/DT pin is sampled three times by a majority detect circuit to determine if a high or a low level is present at the RA4/RX/DT pin. The sampling is done on the seventh, eighth and ninth falling edges of a x16 clock (Figure 11-3).

The x16 clock is a free running clock, and the three sample points occur at a frequency of every 16 falling edges.





### PIC17C4X

Steps to follow when setting up an Asynchronous Reception:

- Initialize the SPBRG register for the appropriate baud rate.
- Enable the asynchronous serial port by clearing the SYNC bit and setting the SPEN bit.
- 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.
- The RCIF bit will be set when reception completes and an interrupt will be generated if the RCIE bit was set.

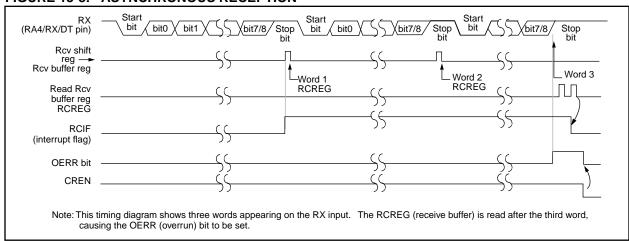
- Read RCSTA to get the ninth bit (if enabled) and FERR bit to determine if any error occurred during reception.
- 8. Read RCREG for the 8-bit received data.

Note:

If an overrun error occurred, clear the error by clearing the OERR bit.

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	00001x	00001u
17h, Bank 0	SPBRG	Baud rate	Baud rate generator register								uuuu uuuu

 $\label{eq:local_equation} \textbf{Legend:} \quad \textbf{x} = \textbf{unknown}, \ \textbf{u} = \textbf{unchanged}, \ \textbf{-} = \textbf{unimplemented read as a} \ \underline{\textbf{0'}}, \ \underline{\textbf{shaded cells are not used for asynchronous reception}.$ 

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

#### 13.3 <u>USART Synchronous Master Mode</u>

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

### 13.3.1 USART SYNCHRONOUS MASTER TRANSMISSION

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

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

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

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

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

Steps to follow when setting up a Synchronous Master Transmission:

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

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

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

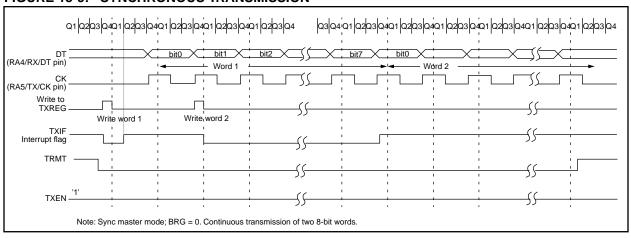
TABLE 13-7: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER TRANSMISSION

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
16h, Bank 0	TXREG	TX7	TX6	TX5	TX4	TX3	TX2	TX1	TX0	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	00001x	00001u
17h, Bank 0	SPBRG	Baud rate	Baud rate generator register								uuuu uuuu

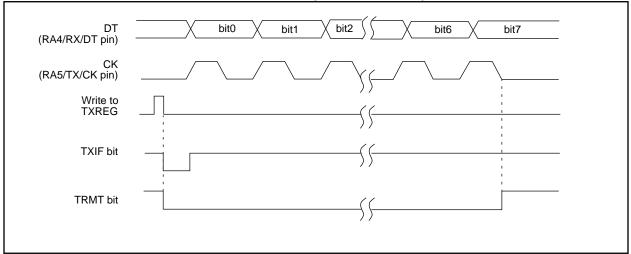
Legend: x = unknown, u = unchanged, - = unimplemented read as a '0', shaded cells are not used for synchronous master transmission.

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

#### FIGURE 13-9: SYNCHRONOUS TRANSMISSION







### 13.3.2 USART SYNCHRONOUS MASTER RECEPTION

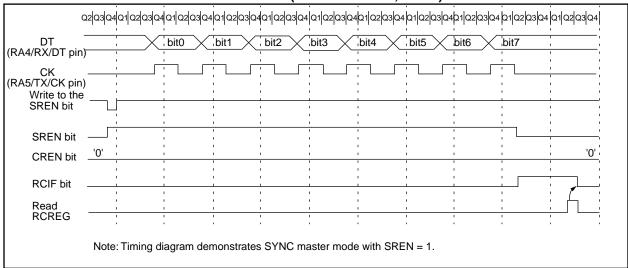
Once synchronous mode is selected, reception is enabled by setting either the SREN (RCSTA<5>) bit or the CREN (RCSTA<4>) bit. Data is sampled on the RA4/RX/DT pin on the falling edge of the clock. If SREN is set, then only a single word is received. If CREN is set, the reception is continuous until CREN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

Steps to follow when setting up a Synchronous Master Reception:

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

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

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



#### 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	00001x	00001u
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 MCLR and Watchdog Timer Reset.

#### 13.4 <u>USART Synchronous Slave Mode</u>

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:

- 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.
- If 9-bit transmission is desired, then set the TX9 bit.
- 5. Start transmission by loading data to TXREG.
- 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 then 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:

- 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.
- The RCIF bit will be set when reception is complete and an interrupt will be generated if the RCIE bit was set.
- Read RCSTA to get the ninth bit (if enabled) and determine if any error occurred during reception.
- 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.

TABLE 13-9: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE TRANSMISSION

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
16h, Bank 0	TXREG	TX7	TX6	TX5	TX4	TX3	TX2	TX1	TX0	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	00001x	00001u
17h, Bank 0	SPBRG	Baud rate	Baud rate generator register								uuuu uuuu

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

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

#### TABLE 13-10: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE 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, Bank1	PIR	RBIF	TMR3IF	TMR2IF	TMR1IF	CA2IF	CA1IF	TXIF	RCIF	0000 0010	0000 0010
13h, Bank0	RCSTA	SPEN	RX9	SREN	CREN		FERR	OERR	RX9D	0000 -00x	0000 -00u
14h, Bank0	RCREG	RX7	RX6	RX5	RX4	RX3	RX2	RX1	RX0	xxxx xxxx	uuuu uuuu
17h, Bank1	PIE	RBIE	TMR3IE	TMR2IE	TMR1IE	CA2IE	CA1IE	TXIE	RCIE	0000 0000	0000 0000
15h, Bank 0	TXSTA	CSRC	TX9	TXEN	SYNC	_		TRMT	TX9D	00001x	00001u
17h, Bank0	SPBRG	Baud rate	Baud rate generator register								uuuu uuuu

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

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

# 14.0 SPECIAL FEATURES OF THE CPU

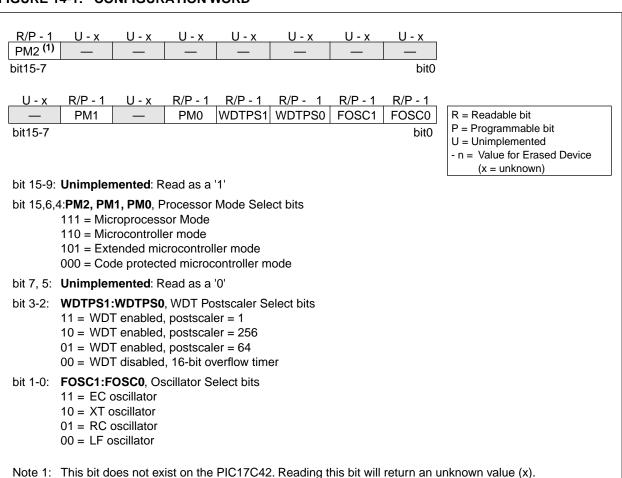
What sets a microcontroller apart from other processors are special circuits to deal with the needs of real time applications. The PIC17CXX family has a host of such features intended to maximize system reliability, minimize cost through elimination of external components, provide power saving operating modes and offer code protection. These are:

- · OSC selection
- Reset
  - Power-on Reset (POR)
  - Power-up Timer (PWRT)
  - Oscillator Start-up Timer (OST)
- Interrupts
- Watchdog Timer (WDT)
- SLEEP
- · Code protection

The PIC17CXX has a Watchdog Timer which can be shut off only through EPROM bits. It runs off its own RC oscillator for added reliability. There are two timers that offer necessary delays on power-up. One is the Oscillator Start-up Timer (OST), intended to keep the chip in RESET until the crystal oscillator is stable. The other is the Power-up Timer (PWRT), which provides a fixed delay of 96 ms (nominal) on power-up only, designed to keep the part in RESET while the power supply stabilizes. With these two timers on-chip, most applications need no external reset circuitry.

The SLEEP mode is designed to offer a very low current power-down mode. The user can wake from SLEEP through external reset, Watchdog Timer Reset or through an interrupt. Several oscillator options are also made available to allow the part to fit the application. The RC oscillator option saves system cost while the LF crystal option saves power. Configuration bits are used to select various options. This configuration word has the format shown in Figure 14-1.

#### FIGURE 14-1: CONFIGURATION WORD



#### 14.1 Configuration Bits

The PIC17CXX has up to seven configuration locations (Table 14-1). These locations can be programmed (read as '0') or left unprogrammed (read as '1') to select various device configurations. Any write to a configuration location, regardless of the data, will program that configuration bit. A TABLWT instruction is required to write to program memory locations. The configuration bits can be read by using the TABLRD instructions. Reading any configuration location between FE00h and FE07h will read the low byte of the configuration word (Figure 14-1) into the TABLATL register. The TABLATH register will be FFh. Reading a configuration location between FE08h and FE0Fh will read the high byte of the configuration word into the TABLATL register. The TABLATH register will be FFh.

Addresses FE00h thorough FE0Fh are only in the program memory space for microcontroller and code protected microcontroller modes. A device programmer will be able to read the configuration word in any processor mode. See programming specifications for more detail.

TABLE 14-1: CONFIGURATION LOCATIONS

Bit	Address
FOSC0	FE00h
FOSC1	FE01h
WDTPS0	FE02h
WDTPS1	FE03h
PM0	FE04h
PM1	FE06h
PM2 <sup>(1)</sup>	FE0Fh <sup>(1)</sup>

Note 1: This location does not exist on the PIC17C42.

Note: When programming the desired configuration locations, they must be programmed in ascending order. Starting with address FE00h.

#### 14.2 <u>Oscillator Configurations</u>

#### 14.2.1 OSCILLATOR TYPES

The PIC17CXX can be operated in four different oscillator modes. The user can program two configuration bits (FOSC1:FOSC0) to select one of these four modes:

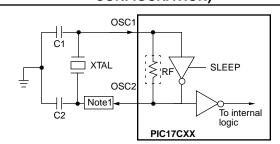
LF: Low Power Crystal
 XT: Crystal/Resonator
 EC: External Clock Input
 RC: Resistor/Capacitor

### 14.2.2 CRYSTAL OSCILLATOR / CERAMIC RESONATORS

In XT or LF modes, a crystal or ceramic resonator is connected to the OSC1/CLKIN and OSC2/CLKOUT pins to establish oscillation (Figure 14-2). The PIC17CXX Oscillator design requires the use of a parallel cut crystal. Use of a series cut crystal may give a frequency out of the crystal manufacturers specifications.

For frequencies above 20 MHz, it is common for the crystal to be an overtone mode crystal. Use of overtone mode crystals require a tank circuit to attenuate the gain at the fundamental frequency. Figure 14-3 shows an example of this.

FIGURE 14-2: CRYSTAL OR CERAMIC
RESONATOR OPERATION
(XT OR LF OSC
CONFIGURATION)



See Table 14-2 and Table 14-3 for recommended values of C1 and C2.

Note 1: A series resistor may be required for AT strip cut crystals.

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

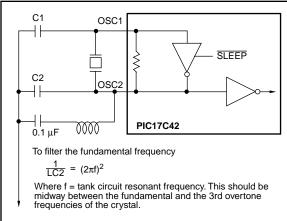


TABLE 14-2: CAPACITOR SELECTION FOR CERAMIC RESONATORS

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

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 <sup>(1)</sup>	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 <sup>(2)</sup>	15-47 pF	15-47 pF
	16 MHz	TBD	TBD
	25 MHz	15-47 pF	15-47 pF
	32 MHz <sup>(3)</sup>	0 (3)	<sub>0</sub> (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  $\approx$  30 pF is recommended.
  - 2: Rs of  $330\Omega$  is required for a capacitor combination of 15/15 pF.
  - 3: Only the capacitance of the board was present.

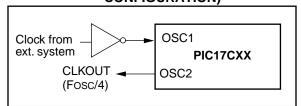
#### **Crystals Used:**

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	± 50 PPM
	ECS-80-18-1	
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/CLK-OUT pin is the CLKOUT output (4 Tosc).

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



### 14.2.4 EXTERNAL CRYSTAL OSCILLATOR CIRCUIT

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

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

FIGURE 14-5: EXTERNAL PARALLEL RESONANT CRYSTAL OSCILLATOR CIRCUIT

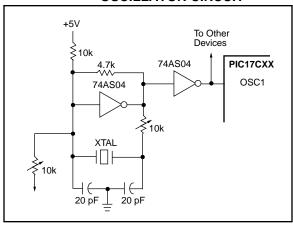
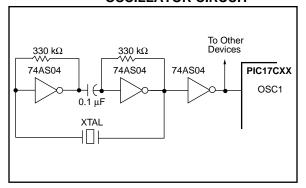


Figure 14-6 shows a series resonant oscillator circuit. This circuit is also designed to use the fundamental frequency of the crystal. The inverter performs a 180-degree phase shift in a series resonant oscillator circuit. The 330 k $\Omega$  resistors provide the negative feedback to bias the inverters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

FIGURE 14-6: EXTERNAL SERIES
RESONANT CRYSTAL
OSCILLATOR CIRCUIT



#### 14.2.5 RC OSCILLATOR

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

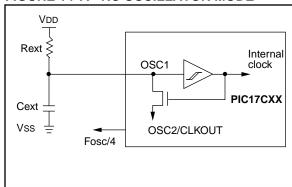
Although the oscillator will operate with no external capacitor (Cext = 0 pF), we recommend using values above 20 pF for noise and stability reasons. With little or no external capacitance, oscillation frequency can vary dramatically due to changes in external capacitances, such as PCB trace capacitance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

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

See Section 18.0 for variation of oscillator frequency due to VDD for given Rext/Cext values as well as frequency variation due to operating temperature for given R, C, and VDD values.

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

FIGURE 14-7: RC OSCILLATOR MODE



#### 14.3 Watchdog Timer (WDT)

The Watchdog Timer's function is to recover from software malfunction. The WDT uses an internal free running on-chip RC oscillator for its clock source. This does not require any external components. This RC oscillator is separate from the RC oscillator of the OSC1/CLKIN pin. That means that the WDT will run, even if the clock on the OSC1/CLKIN and OSC2/CLK-OUT pins of the device has been stopped, for example, by execution of a SLEEP instruction. During normal operation and SLEEP mode, a WDT time-out generates a device RESET. The WDT can be permanently disabled by programming the configuration bits WDTPS1:WDTPS0 as '00' (Section 14.1).

Under normal operation, the WDT must be cleared on a regular interval. This time is less the minimum WDT overflow time. Not clearing the WDT in this time frame will cause the WDT to overflow and reset the device.

#### 14.3.1 WDT PERIOD

The WDT has a nominal time-out period of 12 ms, (with postscaler = 1). The time-out periods vary with temperature, VDD and process variations from part to part (see DC specs). If longer time-out periods are desired, a postscaler with a division ratio of up to 1:256 can be assigned to the WDT. Thus, typical time-out periods up to 3.0 seconds can be realized.

The CLRWDT and SLEEP instructions clear the WDT and the postscaler (if assigned to the WDT) and prevent it from timing out thus generating a device RESET condition.

The TO bit in the CPUSTA register will be cleared upon a WDT time-out.

#### 14.3.2 CLEARING THE WDT AND POSTSCALER

The WDT and postscaler are cleared when:

- · The device is in the reset state
- A SLEEP instruction is executed
- A CLRWDT instruction is executed
- · Wake-up from SLEEP by an interrupt

The WDT counter/postscaler will start counting on the first edge after the device exits the reset state.

#### 14.3.3 WDT PROGRAMMING CONSIDERATIONS

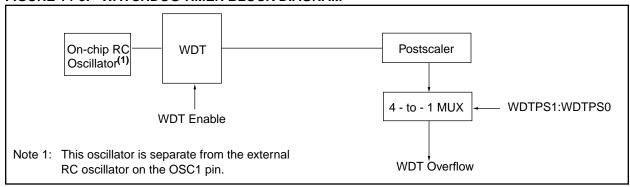
It should also be taken in account that under worst case conditions (VDD = Min., Temperature = Max., max. WDT postscaler) it may take several seconds before a WDT time-out occurs.

The WDT and postscaler is the Power-up Timer during the Power-on Reset sequence.

#### 14.3.4 WDT AS NORMAL TIMER

When the WDT is selected as a normal timer, the clock source is the device clock. Neither the WDT nor the postscaler are directly readable or writable. The overflow time is 65536 Tosc cycles. On overflow, the  $\overline{\text{TO}}$  bit is cleared (device is not reset). The CLRWDT instruction can be used to set the  $\overline{\text{TO}}$  bit. This allows the WDT to be a simple overflow timer. When in sleep, the WDT does not increment.

#### FIGURE 14-8: WATCHDOG TIMER BLOCK DIAGRAM



#### TABLE 14-4: REGISTERS/BITS ASSOCIATED WITH THE WATCHDOG TIMER

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)
_	Config	_	PM1	_	PM0	WDTPS1	WDTPS0	FOSC1	FOSC0	(Note 2)	(Note 2)
06h, Unbanked	CPUSTA	I	_	STKAV	GLINTD	TO	PD	1	1	11 11	11 qq

Legend: -= unimplemented read as '0', q - value depends on condition, shaded cells are not used by the WDT.

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

2: This value will be as the device was programmed, or if unprogrammed, will read as all '1's.

#### 14.4 Power-down Mode (SLEEP)

The Power-down mode is entered by executing a SLEEP instruction. This clears the Watchdog Timer and postscaler (if enabled). The PD bit is cleared and the TO bit is set (in the CPUSTA register). In SLEEP mode, the oscillator driver is turned off. The I/O ports maintain their status (driving high, low, or hi-impedance).

The MCLR/VPP pin must be at a logic high level (VIHMC). A WDT time-out RESET does not drive the MCLR/VPP pin low.

#### 14.4.1 WAKE-UP FROM SLEEP

The device can wake up from SLEEP through one of the following events:

- · A POR reset
- External reset input on MCLR/VPP pin
- WDT Reset (if WDT was enabled)
- Interrupt from RA0/INT pin, RB port change, T0CKI interrupt, or some Peripheral Interrupts

The following peripheral interrupts can wake-up from SLEEP:

- Capture1 interrupt
- · Capture2 interrupt
- USART synchronous slave transmit interrupt
- USART synchronous slave receive interrupt

Other peripherals can not generate interrupts since during SLEEP, no on-chip Q clocks are present.

Any reset event will cause a device reset. Any interrupt event is considered a continuation of program execution. The  $\overline{\text{TO}}$  and  $\overline{\text{PD}}$  bits in the CPUSTA register can be used to determine the cause of device reset. The PD bit, which is set on power-up, is cleared when SLEEP is invoked. The TO bit is cleared if WDT time-out occurred (and caused wake-up).

When the SLEEP instruction is being executed, the next instruction (PC + 1) is pre-fetched. For the device to wake-up through an interrupt event, the corresponding interrupt enable bit must be set (enabled). Wake-up is regardless of the state of the GLINTD bit. If the GLINTD bit is set (disabled), the device continues execution at the instruction after the SLEEP instruction. If the GLINTD bit is clear (enabled), the device executes the instruction after the SLEEP instruction and then branches to the interrupt vector address. In cases where the execution of the instruction following SLEEP is not desirable, the user should have a NOP after the SLEEP instruction.

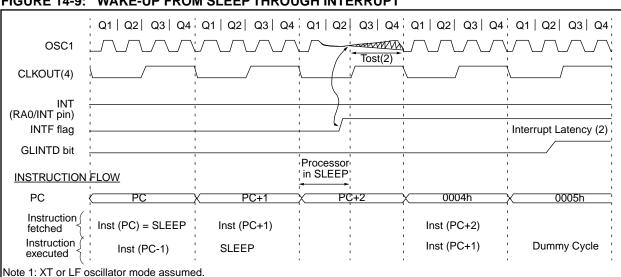
Note: If the global interrupts are disabled (GLINTD is set), but any interrupt source has both its interrupt enable bit and the corresponding interrupt flag bits set, the device will immediately wake-up from sleep. The TO bit is set, and the PD bit is cleared.

The WDT is cleared when the device wake from SLEEP, regardless of the source of wake-up.

#### WAKE-UP DELAY 14.4.1.1

When the oscillator type is configured in XT or LF mode, the Oscillator Start-up Timer (OST) is activated on wake-up. The OST will keep the device in reset for 1024Tosc. This needs to be taken into account when considering the interrupt response time when coming out of SLEEP.

#### FIGURE 14-9: WAKE-UP FROM SLEEP THROUGH INTERRUPT



- 2: Tost = 1024Tosc (drawing not to scale). This delay will not be there for RC osc mode.

  3: When GLINTD = 0 processor jumps to interrupt routine after wake-up. If GLINTD = 1, execution will continue in line.
- 4: CLKOUT is not available in these osc modes, but shown here for timing reference.

#### 14.4.2 MINIMIZING CURRENT CONSUMPTION

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

#### 14.5 Code Protection

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

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

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

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

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

#### 15.0 INSTRUCTION SET SUMMARY

The PIC17CXX instruction set consists of 58 instructions. Each instruction is a 16-bit word divided into an OPCODE and one or more operands. The opcode specifies the instruction type, while the operand(s) further specify the operation of the instruction. The PIC17CXX instruction set can be grouped into three types:

- · byte-oriented
- · bit-oriented
- · literal and control operations.

These formats are shown in Figure 15-1.

Table 15-1 shows the field descriptions for the opcodes. These descriptions are useful for understanding the opcodes in Table 15-2 and in each specific instruction descriptions.

**byte-oriented instructions**, 'f' represents a file register designator and 'd' represents a destination designator. The file register designator specifies which file register is to be used by the instruction.

The destination designator specifies where the result of the operation is to be placed. If 'd' = '0', the result is placed in the WREG register. If 'd' = '1', the result is placed in the file register specified by the instruction.

**bit-oriented instructions**, 'b' represents a bit field designator which selects the number of the bit affected by the operation, while 'f' represents the number of the file in which the bit is located.

**literal and control operations**, 'k' represents an 8- or 11-bit constant or literal value.

The instruction set is highly orthogonal and is grouped into:

- · byte-oriented operations
- · bit-oriented operations
- · literal and control operations

All instructions are executed within one single instruction cycle, unless:

- · a conditional test is true
- the program counter is changed as a result of an instruction
- a table read or a table write instruction is executed (in this case, the execution takes two instruction cycles with the second cycle executed as a NOP)

One instruction cycle consists of four oscillator periods. Thus, for an oscillator frequency of 25 MHz, the normal instruction execution time is 160 ns. If a conditional test is true or the program counter is changed as a result of an instruction, the instruction execution time is 320 ns.

TABLE 15-1: OPCODE FIELD DESCRIPTIONS

	DESCRIPTIONS
Field	Description
f	Register file address (00h to FFh)
р	Peripheral register file address (00h to 1Fh)
i	Table pointer control i = '0' (do not change) i = '1' (increment after instruction execution)
t	Table byte select t = '0' (perform operation on lower byte) t = '1' (perform operation on upper byte literal field, constant data)
WREG	Working register (accumulator)
b	Bit address within an 8-bit file register
k	Literal field, constant data or label
х	Don't care location (= '0' or '1') The assembler will generate code with $x$ = '0'. It is the recommended form of use for compatibility with all Microchip software tools.
d	Destination select 0 = store result in WREG 1 = store result in file register f Default is d = '1'
u	Unused, encoded as '0'
s	Destination select 0 = store result in file register f and in the WREG 1 = store result in file register f Default is s = '1'
label	Label name
C,DC, Z,OV	ALU status bits Carry, Digit Carry, Zero, Overflow
GLINTD	Global Interrupt Disable bit (CPUSTA<4>)
TBLPTR	Table Pointer (16-bit)
TBLAT	Table Latch (16-bit) consists of high byte (TBLATH) and low byte (TBLATL)
TBLATL	Table Latch low byte
TBLATH	Table Latch high byte
TOS	Top of Stack
PC	Program Counter
BSR	Bank Select Register
WDT	Watchdog Timer Counter
TO	Time-out bit
PD	Power-down bit
dest	Destination either the WREG register or the specified register file location
[]	Options
()	Contents
$\rightarrow$	Assigned to
<>	Register bit field
€	In the set of
italics	User defined term (font is courier)

### PIC17C4X

Table 15-2 lists the instructions recognized by the MPASM assembler.

**Note 1:** Any unused opcode is Reserved. Use of any reserved opcode may cause unexpected operation.

**Note 2:** The shaded instructions are not available in the PIC17C42

All instruction examples use the following format to represent a hexadecimal number:

0xhh

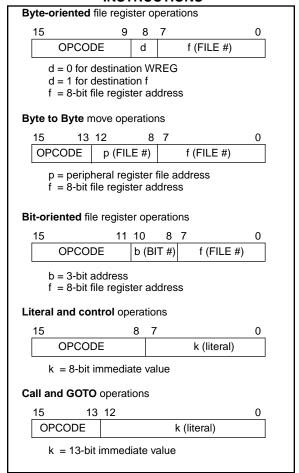
where h signifies a hexadecimal digit.

To represent a binary number:

0000 0100b

where b signifies a binary string.

### FIGURE 15-1: GENERAL FORMAT FOR INSTRUCTIONS



#### 15.1 <u>Special Function Registers as</u> Source/Destination

The PIC17C4X's orthogonal instruction set allows read and write of all file registers, including special function registers. There are some special situations the user should be aware of:

#### 15.1.1 ALUSTA AS DESTINATION

If an instruction writes to ALUSTA, the Z, C, DC and OV bits may be set or cleared as a result of the instruction and overwrite the original data bits written. For example, executing

### 15.2 Q Cycle Activity

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

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

Q1: Instruction Decode Cycle or forced NOP

Q2: Instruction Read Cycle or NOP

Q3: Instruction Execute

Q4: Instruction Write Cycle or NOP

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

### FIGURE 15-2: Q CYCLE ACTIVITY

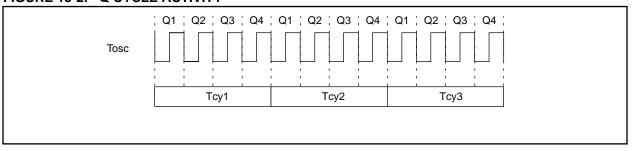


TABLE 15-2: PIC17CXX INSTRUCTION SET

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

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

- Note 1: 2's Complement method.
  - 2: Unsigned arithmetic.
  - 3: If s = 11', only the file is affected: If s = '0', both the WREG register and the file are affected; If only the Working register (WREG) is required to be affected, then f = WREG must be specified.
  - 4: During an LCALL, the contents of PCLATH are loaded into the MSB of the PC and kkkk kkkk is loaded into the LSB of the PC (PCL)
  - 5: Multiple cycle instruction for EPROM programming when table pointer selects internal EPROM. The instruction is terminated by an interrupt event. When writing to external program memory, it is a two-cycle instruction
  - 6: Two-cycle instruction when condition is true, else single cycle instruction.
  - 7: Two-cycle instruction except for TABLRD to PCL (program counter low byte) in which case it takes 3 cycles.
  - 8: A "skip" means that instruction fetched during execution of current instruction is not executed, instead an NOP is executed.
  - 9: These instructions are not available on the PIC17C42.

TABLE 15-2: PIC17CXX INSTRUCTION SET (Cont.'d)

Mnemonic,		Description		16-bit Opcode				Status	Notes
Operands				MSb	MSb		LSb	Affected	
TABLWT	t,i,f	Table Write	2	1010	11ti	ffff	ffff	None	5
TLRD	t,f	Table Latch Read	1	1010	00tx	ffff	ffff	None	
TLWT	t,f	Table Latch Write	1	1010	01tx	ffff	ffff	None	
TSTFSZ	f	Test f, skip if 0	1 (2)	0011	0011	ffff	ffff	None	6,8
XORWF	f,d	Exclusive OR WREG with f	1	0000	110d	ffff	ffff	Z	
BIT-ORIENT	ED FIL	E REGISTER OPERATIONS							
BCF	f,b	Bit Clear f	1	1000	1bbb	ffff	ffff	None	
BSF	f,b	Bit Set f	1	1000	0bbb	ffff	ffff	None	
BTFSC	f,b	Bit test, skip if clear	1 (2)	1001	1bbb	ffff	ffff	None	6,8
BTFSS	f,b	Bit test, skip if set	1 (2)	1001	0bbb	ffff	ffff	None	6,8
BTG	f,b	Bit Toggle f	1	0011	1bbb	ffff	ffff	None	
LITERAL A	ND CO	NTROL OPERATIONS		•					
ADDLW	k	ADD literal to WREG	1	1011	0001	kkkk	kkkk	OV,C,DC,Z	
ANDLW	k	AND literal with WREG	1	1011	0101	kkkk	kkkk	Z	
CALL	k	Subroutine Call	2	111k	kkkk	kkkk	kkkk	None	7
CLRWDT	_	Clear Watchdog Timer	1	0000	0000	0000	0100	TO,PD	
GOTO	k	Unconditional Branch	2	110k	kkkk	kkkk	kkkk	None	7
IORLW	k	Inclusive OR literal with WREG	1	1011	0011	kkkk	kkkk	Z	
LCALL	k	Long Call	2	1011	0111	kkkk	kkkk	None	4,7
MOVLB	k	Move literal to low nibble in BSR	1	1011	1000	uuuu	kkkk	None	
MOVLR	k	Move literal to high nibble in BSR	1	1011	101x	kkkk	uuuu	None	9
MOVLW	k	Move literal to WREG	1	1011	0000	kkkk	kkkk	None	
MULLW	k	Multiply literal with WREG	1	1011	1100	kkkk	kkkk	None	9
RETFIE	_	Return from interrupt (and enable interrupts)	2	0000	0000	0000	0101	GLINTD	7
RETLW	k	Return literal to WREG	2	1011	0110	kkkk	kkkk	None	7
RETURN	_	Return from subroutine	2	0000	0000	0000	0010	None	7
SLEEP	_	Enter SLEEP Mode	1	0000	0000	0000	0011	TO, PD	
SUBLW	k	Subtract WREG from literal	1	1011	0010	kkkk	kkkk	OV,C,DC,Z	
XORLW	k	Exclusive OR literal with WREG	1	1011	0100	kkkk	kkkk	Z	

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

- Note 1: 2's Complement method.
  - 2: Unsigned arithmetic.
  - 3: If s = '1', only the file is affected: If s = '0', both the WREG register and the file are affected; If only the Working register (WREG) is required to be affected, then f = WREG must be specified.
  - 4: During an LCALL, the contents of PCLATH are loaded into the MSB of the PC and kkkk kkkk is loaded into the LSB of the PC (PCL)
  - 5: Multiple cycle instruction for EPROM programming when table pointer selects internal EPROM. The instruction is terminated by an interrupt event. When writing to external program memory, it is a two-cycle instruction.
  - 6: Two-cycle instruction when condition is true, else single cycle instruction.
  - 7: Two-cycle instruction except for TABLRD to PCL (program counter low byte) in which case it takes 3 cycles.
  - 8: A "skip" means that instruction fetched during execution of current instruction is not executed, instead an NOP is executed.
  - 9: These instructions are not available on the PIC17C42.

**ADDLW ADD Literal to WREG** Syntax: [label] ADDLW Operands:  $0 \le k \le 255$ Operation:  $(WREG) + k \rightarrow (WREG)$ Status Affected: OV, C, DC, Z Encoding: 1011 0001 kkkk kkkk Description: The contents of WREG are added to the 8-bit literal 'k' and 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: ADDLW 0x15

> Before Instruction WREG = 0x10After Instruction WREG = 0x25

**ADDWF** ADD WREG to f [ label ] ADDWF Syntax: f,d Operands:  $0 \le f \le 255$  $d \in [0,1]$ Operation: (WREG) + (f)  $\rightarrow$  (dest) Status Affected: OV, C, DC, Z 0000 111d ffff Encoding: ffff Description: Add WREG to register 'f'. If 'd' is 0 the result is stored in WREG. If 'd' is 1 the result is stored back in register 'f'. Words: 1 1 Cycles: Q Cycle Activity: Q1 Q2 Q3 Q4

Decode Execute Write to Read destination register 'f'

Example: ADDWF REG, 0

Before Instruction

WREG 0x17 REG 0xC2

After Instruction

WREG = 0xD9 REG 0xC2

**ADDWFC** ADD WREG and Carry bit to f Syntax: [ label ] ADDWFC Operands:  $0 \le f \le 255$  $d \in [0,1]$ Operation: (WREG) + (f) +  $C \rightarrow$  (dest) Status Affected: OV, C, DC, Z 000d Encoding: 0001 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 = 0x02WREG = 0x4D

After Instruction

Carry bit = 0REG = 0x02WREG = 0x50 ANDLW And Literal with WREG

Syntax: [ label ] ANDLW k

Operands:  $0 \le k \le 255$ 

Operation: (WREG) .AND. (k)  $\rightarrow$  (WREG)

Status Affected: Z

Encoding: 1011 0101 kkkk kkkk

Description: The contents of WREG are AND'ed 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 | Execute | Write to | WREG

Example: ANDLW 0x5F

Before Instruction

WREG = 0xA3

After Instruction

WREG = 0x03

ANDWF	AND WREG with f				
Syntax:	[ label ] ANDWF f,d				
Operands:	$0 \le f \le 255$ $d \in [0,1]$				
Operation:	(WREG) .AND. (f) $\rightarrow$ (dest)				
Status Affected:	Z				
Encoding:	0000 101d ffff	ffff			
Description:	The contents of WREG are AND'ed with register 'f'. If 'd' is 0 the result is stored in WREG. If 'd' is 1 the result is stored back in register 'f'.				

Words: 1 Cycles: 1

Q Cycle Activity:

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

Example: ANDWF REG, 1

Before Instruction

 $\begin{array}{rcl} \mathsf{WREG} &=& \mathsf{0x17} \\ \mathsf{REG} &=& \mathsf{0xC2} \end{array}$ 

After Instruction

WREG = 0x17 REG = 0x02

BCF	•	Bit Clear	f				
Synt	ax:	[label] BCF f,b					
Ope	rands:	$0 \le f \le 255$ $0 \le b \le 7$	$0 \le f \le 255$ $0 \le b \le 7$				
Ope	ration:	$0 \rightarrow (f < b >$	)				
State	us Affected:	None	None				
Enco	oding:	1000	1bbb	1bbb ff		ffff	
Des	cription:	Bit 'b' in reg	Bit 'b' in register 'f' is cleared.				
Wor	ds:	1					
Cycl	es:	1	1				
Q C	ycle Activity:						
	Q1	Q2	Q	3		Q4	
	Decode	Read register 'f'	Exec	ute		Write aister 'f'	

Example: BCF FLAG\_REG, 7

Before Instruction FLAG\_REG = 0xC7

After Instruction FLAG\_REG = 0x47

**BSF** Bit Set f Syntax: [label] BSF f,b Operands:  $0 \le f \le 255$  $0 \le b \le 7$ Operation:  $1 \rightarrow (f < b >)$ Status Affected: None Encoding: 1000 0bbb ffff ffff Description: Bit 'b' in register 'f' is set.

Words: 1

Q Cycle Activity:

Cycles:

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

Example: BSF FLAG\_REG, 7

1

Before Instruction FLAG\_REG= 0x0A

After Instruction

FLAG\_REG= 0x8A

BTFSC	Bit Test,	skip if Cle	ear		
Syntax:	[ label ] BTFSC f,b				
Operands:	$0 \le f \le 255$ $0 \le b \le 7$				
Operation:	skip if $(f < b >) = 0$				
Status Affected:	None				
Encoding:	1001	1bbb	ffff	ffff	
Description:	If bit 'b' in register 'f' is 0 then the next instruction is skipped.  If bit 'b' is 0 then the next instruction fetched during the current instruction execution is discarded, and a NOP is executed instead, making this a two-cycle				
	instruction	•			

Words: 1 Cycles: 1(2)

Q Cycle Activity:

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

If skip:

Q1	Q2	Q3	Q4
Forced NOP	NOP	Execute	NOP

Example: HERE BTFSC FLAG,1 FALSE :

TRUE :

Before Instruction

PC = address (HERE)

After Instruction

If FLAG<1> = 0;

PC = address (TRUE)

If FLAG<1> = 1;

PC = address (FALSE)

BTF	SS	Bit Test,	Bit Test, skip if Set				
Synt	ax:	[label] E	BTFSS f,b	)			
Ope	rands:	$0 \le f \le 12^{n}$ $0 \le b < 7$	7				
Ope	ration:	skip if (f<	o>) = 1				
State	us Affected:	None					
Enco	oding:	1001	0bbb	fff	f	ffff	
Des	cription:	If bit 'b' in r instruction If bit 'b' is 1 fetched dureution, is deuted instevents.	is skipped., then the ring the cui iscarded a ad, making	next in rent in	struc nstruc	ction ction exe- is exe-	
Wor	ds:	1	1				
Cycl	es:	1(2)					
Q C	ycle Activity:						
	Q1	Q2	Q3			Q4	
	Decode	Read register 'f'	Execu	ıte	1	NOP	
If sk	ip:		•				
	Q1	Q2	Q3			Q4	
	Forced NOP	NOP	Execu	ıte	1	NOP	
	mple:	FALSE TRUE	BTFSS :	FLAG,	1		
	Before Instruction PC = address (HERE)						

0;

address (FALSE)

address (TRUE)

BTG	Bit Toggle	e f			
Syntax:	[label] B	TG f,b			
Operands:	$0 \le f \le 255$ $0 \le b < 7$	$0 \le f \le 255$ $0 \le b < 7$			
Operation:	$(\overline{f < b >}) \rightarrow ($	(f <b>)</b>			
Status Affected:	Status Affected: None				
Encoding:	0011	1bbb	ffff	ffff	
Description:	ion: Bit 'b' in data memory location 'f' is inverted.				
Words:	1				
Cycles:	1				
Q Cycle Activity:					
Q1	Q2	Q3		Q4	
Decode	Read register 'f'	Executi	•	/rite ster 'f'	
Example:	BTG I	PORTC,	4		
Before Instruction:  PORTC = 0111 0101 [0x75]					
After Instruct		)101 <b>[0x6</b> 5	i]		

After Instruction
If FLAG<1> =

PC

If FLAG<1> PC CALLSubroutine CallSyntax:[label] CALL kOperands: $0 \le k \le 4095$ Operation: $PC+1 \rightarrow TOS, k \rightarrow PC<12:0>, k<12:8> \rightarrow PCLATH<4:0>; PC<15:13> \rightarrow PCLATH<7:5>$ 

Status Affected: None

Encoding: 111k kkkk kkkk kkkk

Description: Subroutine call within 8K page. First, return address (PC+1) is pushed onto the stack. The 13-bit value is loaded into PC bits<12:0>. Then the upper-eight bits of the PC are copied into PCLATH.

Call is a two-cycle instruction.

See LCALL for calls outside 8K memory appears

space.

Words: 1 Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'<7:0>	Execute	NOP
Forced NOP	NOP	Execute	NOP

Example: HERE CALL THERE

Before Instruction

PC = Address(HERE)

After Instruction

PC = Address(THERE)
TOS = Address(HERE + 1)

Syntax:	[label] CLRF f,s
Operands:	$0 \le f \le 255$
Operation:	$00h \rightarrow f, s \in [0,1]$

Clear f

00h → dest

Status Affected: None

Encoding: 0010 100s fffff ffff

Description: Clears the contents of the specified register(s).

s = 0: Data memory location 'f' and

WREG are cleared.

s = 1: Data memory location 'f' is

cleared.

Words: 1
Cycles: 1

Q Cycle Activity:

**CLRF** 

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Execute	Write register 'f' and other specified register

Example: CLRF FLAG\_REG

Before Instruction

 $FLAG_REG = 0x5A$ 

After Instruction

 $FLAG_REG = 0x00$ 

**CLRWDT Clear Watchdog Timer** Syntax: [label] CLRWDT Operands: None Operation:  $00h \to WDT$  $0 \rightarrow WDT$  postscaler,  $1 \rightarrow \overline{TO}$  $1 \rightarrow \overline{PD}$ TO, PD Status Affected: Encoding: 0000 0000 0000 0100 CLRWDT instruction resets the watchdog Description: timer. It also resets the prescaler of the WDT. Status bits  $\overline{\text{TO}}$  and  $\overline{\text{PD}}$  are set. Words: 1 Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register ALUSTA	Execute	NOP

Example: CLRWDT

Before Instruction

WDT counter ?

After Instruction

WDT counter 0x00 WDT Postscaler 0 TO 1  $\overline{\mathsf{PD}}$ 

CON	/IF	Complem	ent f				
Synt	ax:	[label] (	[ label ] COMF f,d				
Ope	rands:	$0 \le f \le 255$ $d \in [0,1]$	$0 \le f \le 255$ $d \in [0,1]$				
Ope	ration:	$(\overline{f}) \rightarrow (dest)$					
Statu	us Affected: Z						
Enco	oding:	0001	001d	ffff	ffff		
Desc	cription:	The content mented. If 'or WREG. If 'or back in reg	d' is 0 the d' is 1 the	result is	stored in		
Word	ds:	1					
Cycles:		1					
Q Cy	cle Activity:						
	Q1	Q2	Q3		Q4		
	Decode	Read	Execu	te	Write		

register 'f' register 'f'

Example: COMF REG1,0

Before Instruction

REG1 0x13

After Instruction

REG1 0x13 WREG 0xEC

Compare f with WREG, **CPFSEQ CPFSGT** skip if f = WREG [label] CPFSEQ f Syntax: Syntax: Operands: Operands:  $0 \le f \le 255$ (f) - (WREG).Operation: Operation: skip if (f) = (WREG)(unsigned comparison) Status Affected: None Encoding: Encoding: 0011 0001 ffff ffff Description: Description: Compares the contents of data memory location 'f' to the contents of WREG by performing an unsigned subtraction. If 'f' = WREG then the fetched instruction is discarded and an NOP is executed instead making this a two-cycle instruction. Words: 1 Words: Cycles: 1 (2) Cycles: Q Cycle Activity: Q1 Q3 Q4 Q2 Decode Read Execute NOP register 'f' If skip:

Q1 Q2 Q3 Q4 Forced NOP NOP NOP Execute

Example: HERE CPFSEQ REG

NEQUAL : EQUAL

Before Instruction

PC Address HERE **WREG** ? REG ?

After Instruction

If REG WREG:

PC Address (EQUAL) =

If REG WREG; #

> PC Address (NEQUAL)

Compare f with WREG, skip if f > WREG

[label] CPFSGT f

 $0 \le f \le 255$ (f) - (WREG),skip if (f) > (WREG)

(unsigned comparison)

Status Affected: None

0011 0010 ffff ffff

Compares the contents of data memory

location 'f' to the contents of the WREG by performing an unsigned subtraction. If the contents of 'f' > the contents of WREG then the fetched instruction is discarded and an NOP is executed instead making this a two-cycle instruc-

1 1 (2)

Q Cycle Activity:

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

If skip:

Q4 Q1 Q2 Q3 Forced NOP NOP NOP Execute

Example: HERE CPFSGT REG

NGREATER GREATER

Before Instruction

PC Address (HERE)

**WREG** 

After Instruction

If REG WREG;

> PC. Address (GREATER)

If REG ≤ WREG:

> PC Address (NGREATER)

CPFSLT		f with WREG	i,	DAW	Decimal Adjust WREG Register
	skip if f <			Syntax:	[label] DAW f,s
Syntax:	[label] (	CPFSLT f		Operands:	$0 \le f \le 255$
Operands:	$0 \le f \le 25$	5			s ∈ [0,1]
Operation:	(f) – (WRE) skip if (f) < (unsigned)			Operation:	If [WREG<3:0> >9] .OR. [DC = 1] then WREG<3:0> + 6 $\rightarrow$ f<3:0>, s<3:0>; else
Status Affected:	None	. ,			WREG<3:0> $\rightarrow$ f<3:0>, s<3:0>;
Encoding:	0011	0000 fff	f ffff		If [WREG<7:4> >9] .OR. [C = 1] then
Description:	location 'f' to performing	the contents of to the contents of an unsigned su ents of 'f' < the c	of WREG by ubtraction.	Status Affecte	WREG<7:4> + 6 $\rightarrow$ f<7:4>, s<7:4> else WREG<7:4> $\rightarrow$ f<7:4>, s<7:4>
		en the fetched in		Encoding:	0010 111s ffff ffff
	instead mation.	and an NOP is e		Description:	DAW adjusts the eight bit value in WREG resulting from the earlier addition of two variables (each in packed
Words:	1				BCD format) and produces a correct packed BCD result.
Cycles:	1 (2)				s = 0: Result is placed in Data
Q Cycle Activity:					memory location 'f' and WREG.
Q1	Q2	Q3	Q4		s = 1: Result is placed in Data
Decode	Read register 'f'	Execute	NOP		memory location 'f'.
If skip:				Words:	1
Q1	Q2	Q3	Q4	Cycles:	1
Forced NOP	NOP	Execute	NOP	Q Cycle Activ	-
Example:  Before Instru	NLESS LESS	CPFSLT REG : :		Q1 Decode	register 'f' register 'f' and other
PC		ddress (HERE)			specified register
W	= ?			Example1:	DAW REG1, 0
After Instruct If REG PC If REG PC	< W = Ac ≥ W	REG; ddress (LESS) REG; ddress (NLESS)	)	Before In WRE REG C DC	struction G = 0xA5
				After Inst WRE REG C DC <u>Example 2</u> : Before In WRE	G = 0x05 1 = 0x05 = 1 = 0 struction G = 0xCE
				REG C DC	1 = ?? = 0 = 0

After Instruction WREG =

REG1

DC

0x24

0x24

0

DECF	Decreme	ent f			
Syntax:	[ label ]	DECF f,	d		
Operands:	$0 \le f \le 255$ $d \in [0,1]$				
Operation:	$(f)-1 \rightarrow (dest)$				
Status Affected:	OV, C, DC, Z				
Encoding:	0000 011d ffff ffff				
Description:	Decrement register 'f'. If 'd' is 0 the result is stored in WREG. If 'd' is 1 the result is stored back in register 'f'.				
Words:	1				
Cycles:	1				
Q Cycle Activity:					

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

CNT, 1

Example: DECF

Before Instruction  $\begin{array}{rcl} CNT & = & 0x01 \\ Z & = & 0 \end{array}$ After Instruction

 $\begin{array}{rcl} \mathsf{CNT} & = & \mathsf{0x00} \\ \mathsf{Z} & = & \mathsf{1} \end{array}$ 

DECFSZ	Decreme	nt f, ski	p if 0			
Syntax:	[label] [	[label] DECFSZ f,d				
Operands:	$0 \le f \le 258$ $d \in [0,1]$	$0 \le f \le 255$ $d \in [0,1]$				
Operation:		(f) $-1 \rightarrow$ (dest); skip if result = 0				
Status Affected:	Affected: None					
Encoding:	0001	011d	ffff	ffff		
Description:	The contents of register 'f' are decremented. If 'd' is 0 the result is placed in WREG. If 'd' is 1 the result is placed back in register 'f'.  If the result is 0, the next instruction, which is already fetched, is discarded, and an NOP is executed instead making it a two-cycle instruction.					
Words:	1					
Cycles:	1(2)					
Q Cycle Activity:						
Q1	Q2	Q	3	Q4		
Decode	Read register 'f'	Exec		Write to estination		

Before Instruction

Example:

PC = Address (HERE)

CONTINUE

HERE

After Instruction

CNT = CNT - 1If CNT = 0;

PC = Address (CONTINUE)

If CNT  $\neq$  0; PC = Ad

PC = Address (HERE+1)

CNT, 1

LOOP

DECFSZ

GOTO

DCFSNZ	Decreme	nt f, ski	p if not (	)	
Syntax:	[label] D	CFSNZ	f,d		
Operands:	$0 \le f \le 255$ $d \in [0,1]$				
Operation:	(f) $-1 \rightarrow (dest)$ ; skip if not 0				
Status Affected:	None				
Encoding:	0010	011d	ffff	ffff	
Description:	The contents of register 'f' are decremented. If 'd' is 0 the result is placed in WREG. If 'd' is 1 the result is placed back in register 'f'.				
	If the result which is all and an NC ing it a two	ready feto P is exec	ched, is di cuted inste	scarded,	
Words:	1				
Cycles:	1(2)				

vvoius.	•
Cycles:	1(
Q Cycle Activity:	

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

If skip:

Q1	Q2	Q3	Q4
Forced NOP	NOP	Execute	NOP

Example: HERE DCFSNZ TEMP, 1 ZERO NZERO

Before Instruction

TEMP\_VALUE

After Instruction

TEMP\_VALUE TEMP\_VALUE - 1,

If TEMP\_VALUE

PC Address (ZERO)

If TEMP\_VALUE

PC Address (NZERO)

GOTO	Unconditional Branch				
Syntax:	[ label ] GOTO k				
Operands:	$0 \le k \le 8191$				
Operation:	$k \rightarrow PC<12:0>;$ $k<12:8> \rightarrow PCLATH<4:0>,$ $PC<15:13> \rightarrow PCLATH<7:5>$				
Status Affected:	None				
Encoding:	110k	kkkk	kkkk	kkkk	
Description:	GOTO allows an unconditional branch anywhere within an 8K page boundary. The thirteen bit immediate value is loaded into PC bits <12:0>. Then the upper eight bits of PC are loaded into PCLATH. GOTO is always a two-cycle instruction.				
Words:	1				

Cycl	es:	2		
Q C	cle Activity:			
	Q1	Q2	Q3	Q4
	Decode	Read literal	Execute	NOP

Execute

NOP

Example: GOTO THERE

After Instruction

Forced NOP

PC = Address (THERE)

'k'<7:0>

NOP

INCF	Increment f				
Syntax:	[ label ] INCF f,d				
Operands:	$0 \le f \le 255$ $d \in [0,1]$				
Operation:	$(f) + 1 \rightarrow (dest)$				
Status Affected:	OV, C, DC, Z				
Encoding:	0001	010d	ffff	ffff	
Description:	The contents of register 'f' are incremented. If 'd' is 0 the result is placed in WREG. If 'd' is 1 the result is placed back in register 'f'.				
Words:	1				
Cycles:	1				
Q Cycle Activity:					
<b>-</b> .			_		

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

Example: INCF CNT, 1

Before Instruction

CNT = 0xFF Z = 0 C = ?

After Instruction

 $\begin{array}{rcl} {\sf CNT} & = & {\sf 0x00} \\ {\sf Z} & = & 1 \\ {\sf C} & = & 1 \end{array}$ 

INC	FSZ	Incremen	t f, skip if (	)	
Synt	ax:	[ label ]	INCFSZ f,	d	
Ope	rands:	$0 \le f \le 255$ $d \in [0,1]$	5		
Ope	Operation: (f) + 1 $\rightarrow$ (dest) skip if result = 0				
Stati	us Affected:	None			
Enco	oding:	0001	111d f	fff ffff	
Des	cription:	mented. If 'd' is 0 the result is placed in WREG. If 'd' is 1 the result is placed back in register 'f'.  If the result is 0, the next instruction,			
which is already fetched, is discarded, and an NOP is executed instead makin it a two-cycle instruction.			instead making		
Word	ds:	1			
Cycl	es:	1(2)			
Q Cy	ycle Activity:				
	Q1	Q2	Q3	Q4	
	Decode	Read register 'f'	Execute	Write to destination	
If ski	ip:				
	Q1	Q2	Q3	Q4	
	Forced NOP	NOP	Execute	NOP	
<u>Exar</u>	<u>mple</u> :	NZERO	INCFSZ ( :	CNT, 1	
	Before Instruction PC = Address (HERE)				
	After Instruct CNT If CNT PC If CNT PC	= CNT + = 0; = Address ≠ 0;	1 S(ZERO) S(NZERO)		

INFSNZ	Incremen	nt f, skip if	not 0	
Syntax:	[label] I	[label] INFSNZ f,d		
Operands:	$0 \le f \le 255$ $d \in [0,1]$	$0 \le f \le 255$ $d \in [0,1]$		
Operation:	$(f) + 1 \rightarrow 0$	(f) + 1 $\rightarrow$ (dest), skip if not 0		
Status Affected:	None	None		
Encoding:	0010	010d f	fff	ffff
Description:	The contents of register 'f' are incremented. If 'd' is 0 the result is placed in WREG. If 'd' is 1 the result is placed back in register 'f'.  If the result is not 0, the next instruction, which is already fetched, is discarded, and an NOP is executed instead making it a two-cycle instruction.			
Words:	1			
Cycles:	1(2)			
Q Cycle Activity:				
Q1	Q2	Q3		Q4
Decode	Read register 'f'	Execute		Vrite to stination
If skip:				
Q1	Q2	Q3		Q4
Forced NOP	NOP	Execute		NOP
Example:	HERE ZERO NZERO	INFSNZ R	EG, 1	

Before Instruction REG REG

After Instruction

REG + 1 REG

If REG =

PC = Address (ZERO)

If REG =

PC = Address (NZERO) **IORLW Inclusive OR Literal with WREG** 

Syntax: [label] IORLW k

Operands:  $0 \le k \le 255$ 

Operation: (WREG) .OR. (k)  $\rightarrow$  (WREG)

Status Affected: Ζ

Encoding: 1011 0011 kkkk kkkk

Description: The contents of WREG are OR'ed with

the eight bit literal 'k'. The result is

placed in WREG.

Words: Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read	Execute	Write to
	literal 'k'		WREG

0x35 Example: IORLW

Before Instruction

WREG = 0x9A

After Instruction

WREG = 0xBF **IORWF** Inclusive OR WREG with f [label] IORWF f,d Syntax: Operands:  $0 \le f \le 255$  $d \in [0,1]$ (WREG) .OR. (f)  $\rightarrow$  (dest) Operation: Status Affected: Ζ 0000 Encoding: ffff ffff 100d Description: Inclusive OR WREG with register 'f'. If 'd' is 0 the result is placed in WREG. If 'd' is 1 the result is placed back in register 'f'. Words: 1 Cycles: 1 Q Cycle Activity:

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

Example: IORWF RESULT, 0

Before Instruction

 $\begin{array}{lll} \mathsf{RESULT} &=& \mathsf{0x}13 \\ \mathsf{WREG} &=& \mathsf{0x}91 \end{array}$ 

After Instruction

RESULT = 0x13 WREG = 0x93

LCALL Long Call

Syntax: [label] LCALL k Operands:  $0 \le k \le 255$ 

Operation:  $PC + 1 \rightarrow TOS$ ;

 $k \rightarrow PCL, (PCLATH) \rightarrow PCH$ 

Status Affected: None

Encoding: 1011 0111 kkkk kkkk

Description: LCALL allows an unconditional subroutine call to anywhere within the 64k pro-

gram memory space.

First, the return address (PC + 1) is pushed onto the stack. A 16-bit destination address is then loaded into the program counter. The lower 8-bits of the destination address is embedded in the instruction. The upper 8-bits of PC is loaded from PC high holding latch,

PCLATH.

Words: 1 Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read	Execute	Write
	literal 'k'		register PCL
Forced NOP	NOP	Execute	NOP

Example: MOVLW HIGH(SUBROUTINE)

MOVPF WREG, PCLATH LCALL LOW(SUBROUTINE)

Before Instruction

SUBROUTINE = 16-bit Address

PC = ?

After Instruction

PC = Address (SUBROUTINE)

**MOVFP** Move f to p [label] MOVFP f,p Syntax: Operands:  $0 \le f \le 255$  $0 \le p \le 31$ Operation:  $(f) \rightarrow (p)$ Status Affected: None ffff ffff Encoding: 011p pppp Move data from data memory location 'f' Description: to data memory location 'p'. Location 'f' can be anywhere in the 256 word data space (00h to FFh) while 'p' can be 00h to 1Fh. Either 'p' or 'f' can be WREG (a useful special situation). MOVFP is particularly useful for transferring a data memory location to a peripheral register (such as the transmit buffer or an I/O port). Both 'f' and 'p' can be indirectly addressed.

Words: 1 Cycles: 1

Q Cycle Activity:

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

Example: MOVFP REG1, REG2

Before Instruction

REG1 = 0x33, REG2 = 0x11

After Instruction

REG1 = 0x33, REG2 = 0x33 MOVLB Move Literal to low nibble in BSR

Syntax: [label] MOVLB k

Operands:  $0 \le k \le 15$ 

Operation:  $k \rightarrow (BSR<3:0>)$ 

Status Affected: None

Encoding: 1011 1000 uuuu kkkk

Description: The four bit literal 'k' is loaded in the Bank Select Register (BSR). Only the low 4-bits of the Bank Select Register are affected. The upper 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	Execute	Write literal
	literal 'u:k'		'k' to
			BSR<3:0>

Example: MOVLB 0x5

Before Instruction

BSR register = 0x22

After Instruction

BSR register = 0x25

**Note:** For the PIC17C42, only the low four bits of the BSR register are physically implemented. The upper nibble is read as '0'.

Move Literal to high nibble in **MOVLR BSR** Syntax: [label] MOVLR k Operands:  $0 \le k \le 15$ Operation:  $k \rightarrow (BSR < 7:4>)$ Status Affected: None Encoding: 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: Q Cycle Activity:

	Q1	Q2	Q3	Q4
	Decode	Read literal 'k:u'	Execute	Write literal 'k' to BSR<7:4>
Exa	mple:	MOVLR 5		

Before Instruction

BSR register = 0x22

After Instruction

BSR register = 0x52

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

MO\	/LW	Move Literal to WREG				
Synt	ax:	[ label ]	MOVLW	/ k		
Ope	rands:	$0 \leq k \leq 255$				
Ope	ration:	$k \rightarrow (WREG)$				
State	us Affected:	None				
Enco	oding:	1011	0000	kkk	k	kkkk
Description:		The eight I WREG.	oit literal '	k' is lo	adeo	d into
Wor	ds:	1				
Cycl	es:	1				
Q Cycle Activity:						
	Q1	Q2	Q3	3		Q4
	Decode	Read	Exec	ute		Vrite to
		literal 'k'			V	VREG

0x5A

Example: MOVLW

After Instruction

WREG = 0x5A

MOVPF	Move p to f				
Syntax:	[label] MOVPF p,f				
Operands:	$0 \le f \le 255$ $0 \le p \le 31$				
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:					

	register 'p'		
Example:	MOVPF	REG1,	REG2

Q2

Read

Q3

Execute

Q4

Write

register 'f'

Before Instruction

Q1

Decode

REG1 0x11 REG2 0x33

After Instruction

REG1 0x11 REG2 0x11

BAOV/JA/F	NA \A/I	2501-6		
MOVWF	Move Wi	REG to t		
Syntax:	[ label ]	MOVWI	= f	
Operands:	$0 \le f \le 25$	55		
Operation:	(WREG)	$\rightarrow$ (f)		
Status Affected:	None			
Encoding:	0000	0001	ffff	ffff
Description:	Move data Location 'f word data	f' can be a	•	
Words:	1			
Cycles:	1			
Q Cycle Activity:				

Q1	Q2

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

Example: MOVWF REG

Before Instruction

WREG = 0x4F REG 0xFF

After Instruction

WREG = 0x4F REG 0x4F

MULLW	Multiply Literal with WREG			
Syntax:	[ label ] MULLW k			
Operands:	$0 \le k \le 255$			
Operation:	$(k\;x\;WREG)\toPRODH:PRODL$			
Status Affected:	None			
Encoding:	1011 1100 kkkk kkkk			
Description:	An unsigned multiplication is carried out between the contents of WREG and the 8-bit literal 'k'. The 16-bit result is placed in PRODH:PRODL register pair. PRODH contains the high byte.  WREG is unchanged.  None of the status flags are affected.  Note that neither overflow nor carry is possible in this operation. A zero result is possible but not detected.			
Words:	1			
Cycles:	1			
Q Cycle Activity:				

Q1	Q2	Q3	Q4
Decode	Read	Execute	Write
	literal 'k'		registers
			PRODH:
			PRODL

Example: MULLW 0xC4

Before Instruction

WREG = 0xE2PRODH = ? PRODL = ?

After Instruction

WREG = 0xC4 PRODH = 0xAD PRODL = 0x08

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

MULWF Multiply WREG with f

Syntax: [ label ] MULWF f

Operands:  $0 \le f \le 255$ 

Operation:  $(WREG x f) \rightarrow PRODH:PRODL$ 

Status Affected: None

Encoding: 0011 0100 ffff ffff

Description: An unsigned multiplication is carried out between the contents of WREG

and the register file location 'f'. The 16-bit result is stored in the PRODH:PRODL register pair. PRODH contains the high byte. Both WREG and 'f' are unchanged. None of the status flags are affected. Note that neither overflow nor carry is possible in this operation. A zero result is possible but not detected.

Words: 1
Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read	Execute	Write
	register 'f'		registers
			PRODH:
			PRODL

Example: MULWF REG

Before Instruction

WREG = 0xC4
REG = 0xB5
PRODH = ?
PRODL = ?

After Instruction

WREG = 0xC4
REG = 0xB5
PRODH = 0x8A
PRODL = 0x94

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

NEGW	Negate W				
Syntax:	[label] N	NEGW	f,s		
Operands:	$0 \le F \le 255$ $s \in [0,1]$				
Operation:	$\frac{\overline{WREG} + 1 \to (f);}{\overline{WREG} + 1 \to s}$				
Status Affected:	OV, C, DC, Z				
Encoding:	0010	110s	ffff	ffff	
Description:	WREG is negated using two's complement. If 's' is 0 the result is placed in WREG and data memory location 'f'. If 's' is 1 the result is placed only in data memory location 'f'.				
Words:	1				
Cycles:	1				

Q1	Q2	Q3	Q4
Decode	Read	Execute	Write
	register 'f'		register 'f'
			and other
			specified
			register

Example: NEGW REG, 0

Before Instruction

Q Cycle Activity:

WREG = 0011 1010 [0x3A], REG = 1010 1011 [0xAB]

After Instruction

WREG = 1100 0111 [0xC6] REG = 1100 0111 [0xC6]

NOF	•	No Oper	No Operation				
Synt	ax:	[ label ]	NOP				
Ope	rands:	None					
Ope	ration:	No opera	ition				
Stat	us Affected:	None					
Encoding:		0000	0000	000	0	0000	
Des	cription:	No operat	No operation.				
Wor	ds:	1					
Cycles:		1					
Q Cycle Activity:							
	Q1	Q2	Q	3		Q4	
	Decode	NOP	Exec	ute		NOP	

Example:

None.

RETFIE Return from Interrupt

Syntax: [label] RETFIE

Operands: None

Operation:  $TOS \rightarrow (PC)$ ;

 $0 \to \text{GLINTD};$ 

PCLATH is unchanged.

Status Affected: GLINTD

Encoding: 0000 0000 0000 0101

Description: Return from Interrupt. Stack is POP'ed and Top of Stack (TOS) is loaded in the PC. Interrupts are enabled by clearing the GLINTD bit. GLINTD is the global

interrupt disable bit (CPUSTA<4>).

Words: 1 Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register T0STA	Execute	NOP
Forced NOP	NOP	Execute	NOP

Example: RETFIE

After Interrupt

PC = TOSGLINTD = 0 RETLW Return Literal to WREG

Syntax: [label] RETLW k

Operands:  $0 \le k \le 255$ 

Operation:  $k \rightarrow (WREG); TOS \rightarrow (PC);$ 

PCLATH is unchanged

Status Affected: None

Encoding: 1011 0110 kkkk kkkk

Description: WREG is loaded with the eight bit literal 'k'. The program counter is loaded from

the top of the stack (the return address).
The high address latch (PCLATH)

remains unchanged.

Words: 1 Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read	Execute	Write to
	literal 'k'		WREG
Forced NOP	NOP	Execute	NOP

Example: CALL TABLE ; WREG contains table

; offset value ; WREG now has ; table value

: TABLE

ADDWF PC ; WREG = offset RETLW k0 ; Begin table RETLW k1 ;

: :

RETLW kn ; End of table

Before Instruction

WREG = 0x07

After Instruction

WREG = value of k7

RETURN	Return from Subroutine			
Syntax:	[ label ]	RETUR	N	
Operands:	None			
Operation:	$TOS \rightarrow PC;$			
Status Affected:	None			
Encoding:	0000	0000	0000	0010
Description:	Return from subroutine. The stack is popped and the top of the stack (TOS) is loaded into the program counter.			
Words:	1			

Words: 1
Cycles: 2

Q	Cycle	Activity:	
		_	

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

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

Example: RETURN

After Interrupt PC = TOS

RLCF	Rotate Left f through Carry			
Syntax:	[ label ]	RLCF f,	d	
Operands:	$0 \le f \le 25$ $d \in [0,1]$	5		
Operation:	$f$ < $n$ > $\rightarrow$ $d$ $f$ < $7$ > $\rightarrow$ $C$ $C$ $\rightarrow$ $d$ < $0$ :	;		
Status Affected:	С			
Encoding:	0001	101d	ffff	ffff
Description:	one bit to t Flag. If 'd'	nts of regis the left thro is 0 the res d' is 1 the r gister 'f'. regis	ugh the ( ult is plac result is s	Carry ced in
Words:	1			
Cycles:	1			
Q Cycle Activity:				
Q1	Q2	Q3		Q4
Decode	Read register 'f'	Execute		rite to tination
Example:	RLCF	REG	, 0	

Before Instruction

REG = 1110 0110 C = 0

After Instruction

REG = 1110 0110 WREG = 1100 1100 C = 1

RLNCF	Rotate L	.eft f (no	carr	y)
Syntax:	[ label ]	RLNCF	f,d	
Operands:	$0 \le f \le 25$ $d \in [0,1]$	55		
Operation:	$f$ < $n$ > $\rightarrow 0$ $f$ < $7$ > $\rightarrow 0$	,		
Status Affected:	None			
Encoding:	0010	001d	ffi	ff ffff
Description:	one bit to	the left. If WREG. If ck in regis	'd' is ( 'd' is 1	are rotated the result is the result is
Words:	1			
Cycles:	1			
Q Cycle Activity				
Q1	Q2	Q3		Q4
Decode	Read register 'f'	Execu	te	Write to destination
Example:	RLNCF	REC	3, 1	
Before Instr C REG After Instruc C	= 0 = 1110	1011		

REG = 1101 0111

RRC	F	Ro	tate R	ight f th	rougl	h Ca	arry
Synt	ax:	[ <i>la</i>	ibel]	RRCF	f,d		
Oper	rands:	0 ≤	≤ f ≤ 25	55			
		d∈	[0,1]				
Oper	ration:			l <n-1>;</n-1>			
			ightarrow  o ( $ ightarrow$ d<7	,			
Stati	us Affected:	C	→ u<1				
	oding:	Ť	0001	100d	fff	£	ffff
	cription:			nts of reg			
		Fla WF	g. If 'd' REG. If	the right t is 0 the re 'd' is 1 the gister 'f'.	sult is	plac	ced in
			C	reg	jister f		] <b>-</b>
Word	ds:	1					
Cycle	es:	1					
Q Cy	cle Activity:						
	Q1		Q2	Q:	3		Q4
	Decode		ead	Exec	ute	-	Vrite to
l		reg	ster 'f'			aes	stination
<u>Exar</u>	<u>nple</u> :	RR	CF		REG1,	, 0	
	Before Instru	ıction					
	REG1 C	=	1110 0	0110			
	After Instruc	tion					
	REG1 WREG	=	1110 0111	0110 0011			
	C	=	0	0011			

RRNCF	Rotate Right f (no carry)	SETF	Set f
Syntax:	[ label ] RRNCF f,d	Syntax:	[ label ] SETF f,s
Operands:	$0 \le f \le 255$ $d \in [0,1]$	Operands:	$0 \le f \le 255$ s ∈ [0,1]
Operation:	$  f \rightarrow d;                                    $	Operation:	
Status Affected:	None	Status Affected:	None
Encoding:	0010 000d ffff ffff	Encoding:	0010 101s ffff ffff
Description:	The contents of register 'f' are rotated one bit to the right. If 'd' is 0 the result is placed in WREG. If 'd' is 1 the result is placed back in register 'f'.	Description:	If 's' is 0, both the data memory location 'f' and WREG are set to FFh. If 's' is 1 only the data memory location 'f' is set to FFh.
	register f	Words:	1
\\/anda.	4	Cycles:	1
Words:	1	Q Cycle Activity:	
Cycles:	1	Q1	Q2 Q3 Q4
Q Cycle Activity: Q1	Q2 Q3 Q4	Decode	Read Execute Write register 'f'
Decode	Read Execute Write to register 'f' destination		register 'f' register 'f' and other specified register
Example 1:	RRNCF REG, 1		
Before Instru WREG REG	ction = ? = 1101 0111	Example1: Before Instru REG	= 0xDA
After Instruct WREG	ion = 0	WREG After Instruc	
REG	= 1110 1011	REG WREG <u>Example2</u> :	= 0xFF = 0xFF SETF REG, 1
Example 2:	RRNCF REG, 0	Before Instru	,
Before Instru WREG REG	ction = ? = 1101 0111	REG WREG	= 0xDA = 0x05
After Instruct WREG REG	ion = 1110 1011 = 1101 0111	After Instruct REG WREG	tion = 0xFF = 0x05

SLEEP Enter SLEEP mode

Syntax: [label] SLEEP

Operands: None

Operation:  $00h \rightarrow WDT$ ;

 $0 \rightarrow WDT$  postscaler;

 $\begin{array}{l} 1 \rightarrow \overline{TO}; \\ 0 \rightarrow \overline{PD} \end{array}$ 

Status Affected: TO, PD

Encoding: 0000 0000 0000 0011

Description: The power down status bit  $(\overline{PD})$  is cleared. The time-out status bit  $(\overline{TO})$  is

set. Watchdog Timer and its prescaler

are cleared.

The processor is put into SLEEP

mode with the oscillator stopped.

Words: 1
Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register PCLATH	Execute	NOP

Example: SLEEP

Before Instruction

 $\overline{\text{TO}} = ?$   $\overline{\text{PD}} = ?$ 

After Instruction

 $\overline{\text{TO}} = 1 †$  $\overline{\text{PD}} = 0$ 

† If WDT causes wake-up, this bit is cleared

SUBLW Subtract WREG from Literal

Syntax: [label] SUBLW k

Operands:  $0 \le k \le 255$ 

Operation:  $k - (WREG) \rightarrow (WREG)$ 

Status Affected: OV, C, DC, Z

Encoding: 1011 0010 kkkk kkkk

Description: WREG is subtracted from the eight bit literal 'k'. The result is placed in

WREG.

Words: 1
Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read	Execute	Write to
	literal 'k'		WRFG

Example 1: SUBLW 0x02

Before Instruction

WREG = 1 C = ?

After Instruction

WREG = 1

C = 1 ; result is positive

Z = 0

Example 2:

Before Instruction

WREG = 2 C = ?

After Instruction

WREG = 0

C = 1 ; result is zero

Z = 1

Example 3:

Before Instruction

WREG = 3

C = ?

After Instruction

WREG = FF ; (2's complement) C = 0 ; result is negative

Z = 1

SUBWF		WREG from	n f	SUE	WFB	Subtrac Borrow	n f with		
Syntax:		SUBWF f,d		Synt	ax.		[ label ] SUBWFB f,d		
Operands:	$0 \le f \le 25$ $d \in [0,1]$	5		•	rands:	0 ≤ f ≤ 2		-	
Operation:	(f) - (W)	→ (deet)		Оро	arido.	d ∈ [0,1]			
Status Affected:				Ope	ration:	(f) – (W)	$-\overline{C} \rightarrow (dest)$		
	OV, C, D			•	us Affected:	OV, C, E			
Encoding:	0000	010d ff:			oding:	0000	001d ff:	ff ffff	
Description: Words:	compleme result is st	VREG from regent method). If tored in WREG tored back in re	'd' is 0 the i. If 'd' is 1 the		cription:	Subtract (borrow) ment met stored in	WREG and the from register 'f' hod). If 'd' is 0 twREG. If 'd' is ck in register 'f'	carry flag (2's comple- he result is 1 the result is	
Cycles:	1			\^/~ =	da.		ck in register i	•	
Q Cycle Activity:				Wor		1			
Q1	Q2	Q3	Q4	Cycl		1			
Decode	Read	Execute	Write to	Q C	cle Activity:			•	
	register 'f'		destination		Q1	Q2	Q3	Q4	
Example 1:	SUBWF	REG1, 1			Decode	Read register 'f'	Execute	Write to destination	
Before Instru REG1	uction = 3			<u>Exar</u>	nple 1:	SUBWFB	REG1, 1		
WREG	= 2			Before Instruction					
C After Instruc					REG1 WREG C	= 0x19 = 0x0D = 1	(0001 100 (0000 110	,	
REG1 WREG	= 1 = 2				After Instruc	•			
C		result is positiv	/e		REG1	= 0x0C	(0000 101	1)	
Z	= 0				WREG	= 0x0D	(0000 110	,	
Example 2:					C Z	= 1 = 0	; result is po	sitive	
Before Instru	uction			Evo	_		DEG1 0		
REG1	= 2				Example2: SUBWFB REG1,0				
WREG C	= 2 = ?				Before Instru REG1	uction = 0x1B			
After Instruc	·				WREG	= 0x1B $=$ 0x1A	(0001 101	•	
REG1	= 0				С	= 0			
WREG	= 2				After Instruc				
C Z	= 1 ; = 1	result is zero			REG1 WREG		(0001 101	1)	
Example 3:	- '				C Z	= 0x00 = 1 = 1	; result is ze	ro	
Before Instru	uction			Evo	nple3:		DEG1 1		
REG1	= 1					SUBWFB	REGI,I		
WREG C	= 2 = ?				Before Instru REG1	uction = 0x03	(0000 001	1)	
After Instruc	•				WREG	= 0x05 = 0x0E	(0000 001		
REG1	= FF				С	= 1	,	,	
WREG	= 2				After Instruc	tion			
C Z		result is negat	ive		REG1	= 0xF5		0) [2's comp]	
۷	= 0				WREG C	= 0x0E = 0	(0000 110 ; result is ne		
					Z	= 0	,	<b>5</b> ** <b>-</b>	

SWAPF	Swap f					
Syntax:	[label] S	SWAPF f,	d			
Operands:	$0 \le f \le 255$ $d \in [0,1]$	5				
Operation: $f<3:0> \rightarrow dest<7:4>;$ $f<7:4> \rightarrow dest<3:0>$						
Status Affected: None						
Encoding:	0001	110d	ffff	ffff		
Description:	'f' are excha placed in W	The upper and lower nibbles of register 'f' are exchanged. If 'd' is 0 the result is placed in WREG. If 'd' is 1 the result is placed in register 'f'.				
Words:	1					
Cycles:	1					
Q Cycle Activity:						
Q1	Q2	Q3		Q4		
Decode	Read register 'f'	Execute	1 -	Vrite to stination		
Example:	SWAPF R	REG, 0				
Before Instru REG	Before Instruction REG = 0x53					
After Instruction  REG = 0x35						

TAB	LRD	Table Rea	ıd				
Synt	ax:	[ label ]	TABLRD	t,i,f			
Ope	rands:	$0 \le f \le 255$ $i \in [0,1]$ $t \in [0,1]$	5				
Ope	ration:	If t = 1, TBLATH $\rightarrow$ f; If t = 0, TBLATL $\rightarrow$ f; Prog Mem (TBLPTR) $\rightarrow$ TBLAT; If i = 1, TBLPTR + 1 $\rightarrow$ TBLPTR					
Stati	us Affected:	None					
Enco	oding:	1010	10ti	ffff	ffff		
Desc	cription:	is mov	<ol> <li>A byte of the table latch (TBLAT) is moved to register file 'f'.</li> <li>If t = 0: the high byte is moved;</li> <li>If t = 1: the low byte is moved</li> </ol>				
		If t = 1: the low byte is moved  2. Then the contents of the program memory location pointed to by the 16-bit Table Pointer (TBLPTR) is loaded into the 16-bit Table Latch (TBLAT).					
		-	: TBLPTF : TBLPTF increme	R is not	mented;		
Word	ds:	1					
Cycl	es:	2 (3 cycle	2 (3 cycle if f = PCL)				
Q C	ycle Activity:						
	Q1	Q2	Q3		Q4		
	Decode	Read register TBLATH or TBLATL	Execu	.	Write gister 'f'		

TABLRD	Table Re	ead			
Example1:	TABLRD	1, 1,	REG ;		
Before Instru	ction				
REG		=	0x53		
TBLATH		=	0xAA		
TBLATL		=	0x55		
TBLPTR		=	074 1000		
MEMORY	(TBLPTR)	=	0x1234		
After Instruct	ion (table v	vrite co	mpletion)		
REG		=	0xAA		
TBLATH		=	0x12		
TBLATL		=	0x34		
TBLPTR		=	0xA357		
MEMORY	(TBLPTR)	=	0x5678		
Example2:	TABLRD	0, 0,	REG ;		
Before Instru	ction				
REG		=	0x53		
TBLATH		=	0xAA		
TBLATL		=	0x55		
TBLPTR		=	0xA356		
MEMORY	(TBLPTR)	=	0x1234		
After Instruction (table write completion)					
REG		=	0x55		
TBLATH		=	0x12		
TBLATL		=	0x34		
TBLPTR		=			
MEMORY	(TBLPTR)	=	0x1234		

TABLWT	Table Write				
Syntax:	[ label ] TABLWT t,i,f				
Operands:	$0 \le f \le 255$ $i \in [0,1]$ $t \in [0,1]$				
Operation:	$\begin{split} &\text{If } t = 0, \\ &f \to \text{TBLATL}; \\ &\text{If } t = 1, \\ &f \to \text{TBLATH}; \\ &\text{TBLAT} \to \text{Prog Mem (TBLPTR)}; \\ &\text{If } i = 1, \\ &\text{TBLPTR} + 1 \to \text{TBLPTR} \end{split}$				
Status Affected:	None				
Encoding:	1010 11ti ffff ffff				
Description:	Load value in 'f' into 16-bit table latch (TBLAT)     If t = 0: load into low byte;     If t = 1: load into high byte				
	<ol> <li>The contents of TBLAT is written to the program memory location pointed to by TBLPTR If TBLPTR points to external program memory location, then the instruction takes two-cycle If TBLPTR points to an internal EPROM location, then the instruction is terminated when an interrupt is received.</li> </ol>				

ote: The MCLR/VPP pin must be at the programming voltage for successful programming of internal

memory.

If  $\overline{\text{MCLR}}/\text{VPP} = \text{VDD}$ 

the programming sequence of internal memory will be executed, but will not be successful (although the internal memory location may be disturbed)

The TBLPTR can be automatically incremented
 If i = 0; TBLPTR is not

incremented

If i = 1; TBLPTR is incremented

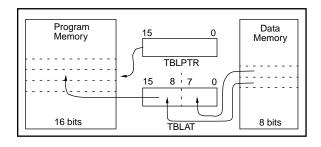
Words: 1

Cycles: 2 (many if write is to on-chip EPROM program memory)

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read	Execute	Write
	register 'f'		register
			TBLATH or
			TBLATL

TABLWT	Table Write			
Example1:	TABLWT 0,	1,	REG	
Before Instructi	ion			
REG	=	:	0x53	
TBLATH	=	:	0xAA	
TBLATL	=	:	0x55	
TBLPTR	=	:	0xA356	
MEMORY(1	TBLPTR) =	:	0xFFFF	
After Instruction (table write completion)				
REG		:	0x53	
TBLATH	=	:	0x53	
TBLATL	=	:	0x55	
TBLPTR	=	:	0xA357	
MEMORY(1	$\Gamma BLPTR - 1) =$	:	0x5355	
Example 2:	TABLWT 1,	0,	REG	
Before Instruct	ion			
REG	=	:	0x53	
TBLATH	=	:	0xAA	
TBLATL	=	:	0x55	
TBLPTR	=	:	0xA356	
MEMORY(1	TBLPTR) =	:	0xFFFF	
After Instruction	n (table write	cor	mpletion)	
REG	=	:	0x53	
TBLATH	=	:	0xAA	



0x53

0xA356

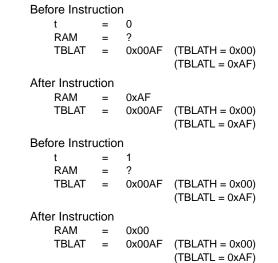
0xAA53

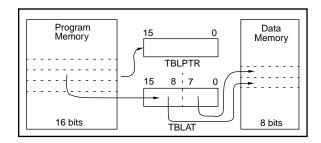
**TBLATL** 

**TBLPTR** 

MEMORY(TBLPTR)

TLR	TLRD Table Latch Read						
Synt	ax:	[ label ]	TLRD t,f				
Ope	rands:	$0 \le f \le 25$ $t \in [0,1]$	$0 \le f \le 255$ $t \in [0,1]$				
Ope	ration:	If $t = 1$ ,	$\Gamma L \to f;$ $\Gamma H \to f$				
State	us Affected:	None					
Enco	oding:	1010	00tx	ffff	ffff		
Des	cription:	(TBLAT) in is unaffect If t = 1; hig If t = 0; low This instru with TABLE	Read data from 16-bit table latch (TBLAT) into file register 'f'. Table Latch is unaffected.  If t = 1; high byte is read  If t = 0; low byte is read  This instruction is used in conjunction with TABLRD to transfer data from program memory to data memory.				
Wor	ds:	1	1				
Cycl	es:	1					
Q C	ycle Activity:						
	Q1	Q2	Q	3	Q4		
	Decode	Read register TBLATH or TBLATL	Exec		Write egister 'f'		
<u>Exar</u>	mple:	TLRD	t, RAM				





TLWT	Table Latch Write					
Syntax:	[label] TLWT	,f				
Operands:	$0 \le f \le 255$ $t \in [0,1]$					
Operation:	If $t = 0$ , $f \rightarrow TBLATL$ ; If $t = 1$ , $f \rightarrow TBLATH$					
Status Affected:	None					
Encoding:	1010 01tx	ffff	ffff			
Description:	Data from file register 'f' is written into the 16-bit table latch (TBLAT).					
	If t = 1; high byte is	written				
	If $t = 0$ ; low byte is	written				
	This instruction is used in conjunction with TABLWT to transfer data from data memory to program memory.					
Words:	1					
Cycles:	1					
Q Cycle Activity:						

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Execute	Write register TBLATH or TBLATL

Example: TLWT t, RAM

Before Instruction

t = 0RAM = 0xB7

TBLAT = 0x0000 (TBLATH = 0x00) (TBLATL = 0x00)

After Instruction

RAM = 0xB7

TBLAT = 0x00B7 (TBLATH = 0x00) (TBLATL = 0xB7)

Before Instruction

t = 1RAM = 0xB7

TBLAT = 0x0000 (TBLATH = 0x00)

(TBLATL = 0x00)

After Instruction

RAM = 0xB7

TBLAT = 0xB700 (TBLATH = 0xB7)

(TBLATL = 0x00)

**TSTFSZ** Test f, skip if 0 [label] TSTFSZ f Syntax: Operands:  $0 \leq f \leq 255$ Operation: skip if f = 0Status Affected: None Encoding: 0011 0011 ffff ffff If 'f' = 0, the next instruction, fetched Description: during the current instruction execution, is discarded and an NOP is executed making this a two-cycle instruction. Words: 1 Cycles: 1 (2)

Q Cycle Activity:

 Q1
 Q2
 Q3
 Q4

 Decode
 Read register 'f'
 Execute
 NOP

If skip:

 Q1
 Q2
 Q3
 Q4

 Forced NOP
 NOP
 Execute
 NOP

Example: HERE TSTFSZ CNT

NZERO :

Before Instruction

PC = Address(HERE)

After Instruction

If CNT = 0x00,

PC = Address (ZERO)

If CNT  $\neq$  0x00,

PC = Address (NZERO)

**Exclusive OR Literal with XORLW WREG** Syntax: [label] XORLW k Operands:  $0 \le k \le 255$ Operation: (WREG) .XOR.  $k \rightarrow$  (WREG) Status Affected: Ζ 1011 Encoding: 0100 kkkk kkkk Description: The contents of WREG are XOR'ed with the 8-bit literal 'k'. The result is placed in WREG. Words: 1 Cycles: 1 Q Cycle Activity: Q1 Q2 Q3 Q4 Write to Decode Read Execute

Example: XORLW 0xAF

literal 'k'

**WREG** 

Before Instruction

WREG = 0xB5

After Instruction

WREG = 0x1A

XORWF Exclusive OR WR				REG v	with	ı f	
Synt	ax:	[label]	KORWF	f,d			
Ope	rands:	$0 \le f \le 258$ $d \in [0,1]$	5				
Ope	Operation: (WREG) .XOR. (f) $\rightarrow$ (dest)				)		
State	us Affected:	Z					
Enco	oding:	0000	110d	fff	f	ffff	
Des	cription:	Exclusive C with registe stored in W stored back	er 'f'. If 'd' 'REG. If '	is 0 th d' is 1	e re the	sult is	
Wor	ds:	1	1				
Cycl	es:	1	1				
Q C	ycle Activity:						
	Q1	Q2	Q3	3		Q4	
	Decode	Read register 'f'	Exec	ute	-	Vrite to stination	

Example: XORWF REG, 1

Before Instruction

REG = 0xAF WREG = 0xB5

After Instruction

REG = 0x1A WREG = 0xB5

NOTES:

### 16.0 DEVELOPMENT SUPPORT

### 16.1 <u>Development Tools</u>

The PIC16/17 microcontrollers are supported with a full range of hardware and software development tools:

- PICMASTER/PICMASTER CE Real-Time In-Circuit Emulator
- ICEPIC Low-Cost PIC16C5X and PIC16CXXX In-Circuit Emulator
- PRO MATE® II Universal Programmer
- PICSTART® Plus Entry-Level Prototype Programmer
- PICDEM-1 Low-Cost Demonstration Board
- PICDEM-2 Low-Cost Demonstration Board
- PICDEM-3 Low-Cost Demonstration Board
- MPASM Assembler
- MPLAB-SIM Software Simulator
- MPLAB-C (C Compiler)
- Fuzzy logic development system (fuzzyTECH®-MP)

# 16.2 PICMASTER: High Performance Universal In-Circuit Emulator with MPLAB IDE

The PICMASTER Universal In-Circuit Emulator is intended to provide the product development engineer with a complete microcontroller design tool set for all microcontrollers in the PIC12C5XX, PIC14000, PIC16C5X, PIC16CXXX and PIC17CXX families. PICMASTER is supplied with the MPLAB™ Integrated Development Environment (IDE), which allows editing, "make" and download, and source debugging from a single environment.

Interchangeable target probes allow the system to be easily reconfigured for emulation of different processors. The universal architecture of the PICMASTER allows expansion to support all new Microchip microcontrollers.

The PICMASTER Emulator System has been designed as a real-time emulation system with advanced features that are generally found on more expensive development tools. The PC compatible 386 (and higher) machine platform and Microsoft Windows® 3.x environment were chosen to best make these features available to you, the end user.

A CE compliant version of PICMASTER is available for European Union (EU) countries.

#### 16.3 <u>ICEPIC: Low-cost PIC16CXXX</u> In-Circuit Emulator

ICEPIC is a low-cost in-circuit emulator solution for the Microchip PIC16C5X and PIC16CXXX families of 8-bit OTP microcontrollers.

ICEPIC is designed to operate on PC-compatible machines ranging from 286-AT® through Pentium<sup>TM</sup> based machines under Windows 3.x environment. ICEPIC features real time, non-intrusive emulation.

#### 16.4 PRO MATE II: Universal Programmer

The PRO MATE II Universal Programmer is a full-featured programmer capable of operating in stand-alone mode as well as PC-hosted mode.

The PRO MATE II has programmable VDD and VPP supplies which allows it to verify programmed memory at VDD min and VDD max for maximum reliability. It has an LCD display for displaying error messages, keys to enter commands and a modular detachable socket assembly to support various package types. In standalone mode the PRO MATE II can read, verify or program PIC16C5X, PIC16CXXX, PIC17CXX and PIC14000 devices. It can also set configuration and code-protect bits in this mode.

### 16.5 <u>PICSTART Plus Entry Level</u> <u>Development System</u>

The PICSTART programmer is an easy-to-use, low-cost prototype programmer. It connects to the PC via one of the COM (RS-232) ports. MPLAB Integrated Development Environment software makes using the programmer simple and efficient. PICSTART Plus is not recommended for production programming.

PICSTART Plus supports all PIC12C5XX, PIC14000, PIC16C5X, PIC16CXXX and PIC17CXX devices with up to 40 pins. Larger pin count devices such as the PIC16C923 and PIC16C924 may be supported with an adapter socket.

#### 16.6 <u>PICDEM-1 Low-Cost PIC16/17</u> 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 <u>PICDEM-2 Low-Cost PIC16CXX</u> 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.

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

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

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

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

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

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

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

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

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

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

# 16.13 <u>Fuzzy Logic Development System</u> (<u>fuzzyTECH-MP</u>)

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

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

#### 16.14 <u>MP-DriveWay™ – Application Code</u> Generator

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

#### 16.15 <u>SEEVAL® Evaluation and</u> <u>Programming System</u>

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

# 16.16 <u>TrueGauge<sup>®</sup> Intelligent Battery</u> Management

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

#### 16.17 <u>Keeloq® Evaluation and</u> <u>Programming Tools</u>

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

TABLE 16-1: DEVELOPMENT TOOLS FROM MICROCHIP

Product	** MPLAB™ Integrated Development Environment	MPLAB™ C Compiler	MP-DriveWay Applications Code Generator	fuzzyTECH®-MP Explorer/Edition Fuzzy Logic Dev. Tool	*** PICMASTER®/ 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	I	I	EM167015/ EM167101	I	DV007003	I	DV003001
PIC14000	SW007002	SW006005	I	I	EM147001/ EM147101	1	DV007003	I	DV003001
PIC16C52, 54, 54A, 55, 56, 57, 58A	SW007002	SW006005	SW006006	DV005001/ DV005002	EM167015/ EM167101	EM167201	DV007003	DV162003	DV003001
PIC16C554, 556, 558	SW007002	SW006005	I	DV005001/ DV005002	EM167033/ EM167113		DV007003	I	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		I	EM167035/ EM167105	İ	DV007003	DV162002	DV003001
PIC16C71	SW007002	SW006005	SW006006	DV005001/ DV005002	EM167027/ EM167105	EM167205	DV007003	DV162003	DV003001
PIC16C710, 711	SW007002	SW006005	SW006006	DV005001/ DV005002	EM167027/ EM167105	1	DV007003	DV162003	DV003001
PIC16C72	SW007002	SW006005	SW006006	I	EM167025/ EM167103	1	DV007003	DV162002	DV003001
PIC16F83	SW007002	SW006005	SW006006	DV005001/ DV005002	EM167029/ EM167107	1	DV007003	DV162003	DV003001
PIC16C84	SW007002	SW006005	SW006006	DV005001/ DV005002	EM167029/ EM167107	EM167206	DV007003	DV162003	DV003001
PIC16F84	SW007002	SW006005	SW006006	DV005001/ DV005002	EM167029/ EM167107	1	DV007003	DV162003	DV003001
PIC16C923, 924*	SW007002	SW006005	SW006006	DV005001/ DV005002	EM167031/ EM167111		DV007003	1	DV003001
PIC17C42, 42A, 43, 44	SW007002	SW006005	SW006006	DV005001/ DV005002	EM177007/ EM177107	1	DV007003	I	DV003001
*Contact Microchip Technology for availability date **MPLAB Integrated Development Environment includes MPLAB-SIM Simulator and MPASM Assembler	inology for availa velopment Enviro	ability date	s MPLAB-SIM Si	mulator and	***AII PICMASTER and PICMA PRO MATE II programmer ****PRO MATE socket modules: ordering guide for specific o	and PICMAST rogrammer t modules are or specific ord	II PICMASTER and PICMASTER-CE ordering pa PRO MATE II programmer RO MATE socket modules are ordered separately ordering guide for specific ordering part numbers	***All PICMASTER and PICMASTER-CE ordering part numbers above include PRO MATE II programmer PRO MATE II programmer PRO MATE socket modules are ordered separately. See development systems ordering guide for specific ordering part numbers	lude stems
Product	TRUEGAUG	TRUEGAUGE® Development Kit		SEEVAL® Designers Kit	Hopping Code Security Programmer Kit	Security Prog		Hopping Code Security Eval/Demo Kit	ity 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	001

#### PIC17C42 ELECTRICAL CHARACTERISTICS

#### **Absolute Maximum Ratings †**

Ambient temperature under bias	55 to +125°C
Storage temperature	65°C to +150°C
Voltage on VDD with respect to Vss	0 to +7.5V
Voltage on MCLR with respect to Vss (Note 2)	0.6V to +14V
Voltage on RA2 and RA3 with respect to Vss	0.6V to +12V
Voltage on all other pins with respect to Vss	0.6V to VDD + 0.6V
Total power dissipation (Note 1)	1.0W
Maximum current out of Vss pin(s) - Total	250 mA
Maximum current into VDD pin(s) - Total	200 mA
nput clamp current, lik (Vi < 0 or Vi > VDD)	±20 mA
Output clamp current, loк (Vo < 0 or Vo > VDD)	±20 mA
Maximum output current sunk by any I/O pin (except RA2 and RA3)	35 mA
Maximum output current sunk by RA2 or RA3 pins	60 mA
Maximum output current sourced by any I/O pin	20 mA
Maximum current sunk by PORTA and PORTB (combined)	150 mA
Maximum current sourced by PORTA and PORTB (combined)	100 mA
Maximum current sunk by PORTC, PORTD and PORTE (combined)	150 mA
Maximum current sourced by PORTC, PORTD and PORTE (combined)	100 mA
<b>Note 1.</b> Power dissination is calculated as follows: Pdis = $VDD \times \{IDD - \sum IOH\} + \sum \{(VDD)^{-1}\}$	

- Note 2: Voltage spikes below Vss at the MCLR pin, inducing currents greater than 80 mA, may cause latch-up. Thus, a series resistor of 50-100 $\Omega$  should be used when applying a "low" level to the  $\overline{MCLR}$  pin rather than pulling this pin directly to Vss.

† NOTICE: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

© 1996 Microchip Technology Inc.

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	VDD: 4.5V to 5.5V
	IDD: 6 mA max.	IDD: 6 mA max.
	IPD: 5 μA max. at 5.5V (WDT disabled)	IPD: 5 μA max. at 5.5V (WDT disabled)
	Freq: 4 MHz max.	Freq: 4 MHz max.
XT	VDD: 4.5V to 5.5V	VDD: 4.5V to 5.5V
	IDD: 24 mA max.	IDD: 38 mA max.
	IPD: 5 μA max. at 5.5V (WDT disabled)	IPD: 5 μA max. at 5.5V (WDT disabled)
	Freq: 16 MHz max.	Freq: 25 MHz max.
EC	VDD: 4.5V to 5.5V	VDD: 4.5V to 5.5V
	IDD: 24 mA max.	IDD: 38 mA max.
	IPD: 5 μA max. at 5.5V (WDT disabled)	IPD: 5 μA max. at 5.5V (WDT disabled)
	Freq: 16 MHz max.	Freq: 25 MHz max.
LF	VDD: 4.5V to 5.5V	VDD: 4.5V to 5.5V
	IDD: 150 μA max. at 32 kHz (WDT enabled)	IDD: 150 μA max. at 32 kHz (WDT enabled)
	IPD: 5 μA max. at 5.5V (WDT disabled)	IPD: 5 μA max. at 5.5V (WDT disabled)
	Freq: 2 MHz max.	Freq: 2 MHz max.

#### 17.1 DC CHARACTERISTICS:

PIC17C42-16 (Commercial, Industrial) PIC17C42-25 (Commercial, Industrial)

			Standard	d Opera	ating C	ondition	ns (unless otherwise stated)
DC CHARA	CTERIS	STICS	Operating	g tempe	erature		
DO CHARA	O I LIVIC	71100				-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	4.5	_	5.5	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	IDD	Supply Current	_	3	6	mA	Fosc = 4 MHz (Note 4)
D011		(Note 2)	_	6	12 *	mA	Fosc = 8 MHz
D012			_	11	24 *	mA	Fosc = 16 MHz
D013			_	19	38	mA	Fosc = 25 MHz
D014			_	95	150	μΑ	Fosc = 32 kHz
							WDT enabled (EC osc configuration)
D020	IPD	Power-down Current	_	10	40	μΑ	VDD = 5.5V, WDT enabled
D021		(Note 3)	_	< 1	5	μΑ	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 need to be considered.

For the RC oscillator, the current through the external pull-up resistor (R) can be estimated as: VDD / (2 • R). For capacitive loads, The current can be estimated (for an individual I/O pin) as (CL • VDD) • f

CL = Total capacitive load on the I/O pin; f = average frequency on 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, 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 IR = VDD/2Rext (mA) with Rext in kOhm.

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

Standard Operating Conditions (unless otherwise stated)

Operating temperature

DC CHARACTERISTICS

 $-40^{\circ}$ C  $\leq$  TA  $\leq$  +85 $^{\circ}$ C for industrial and  $0^{\circ}$ C  $\leq$  TA  $\leq$  +70 $^{\circ}$ C for commercial

Operating voltage VDD range as described in Section 17.1

Parameter			1	- 11.3			
No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
		Input Low Voltage					
	VIL	I/O ports					
D030		with TTL buffer	Vss	_	0.8	V	
D031		with Schmitt Trigger buffer	Vss	_	0.2VDD	V	
D032		MCLR, OSC1 (in EC and RC mode)	Vss	_	0.2VDD	V	Note1
D033		OSC1 (in XT, and LF mode)	_	0.5VDD	_	V	
		Input High Voltage					
	VIH	I/O ports		_			
D040		with TTL buffer	2.0	_	VDD	V	
D041		with Schmitt Trigger buffer	0.8VDD	_	Vdd	V	
D042		MCLR	0.8VDD	_	Vdd	V	Note1
D043		OSC1 (XT, and LF mode)	_	0.5VDD	_	V	
D050	VHYS	Hysteresis of Schmitt Trigger inputs	0.15VDD*	_	_	V	
		Input Leakage Current (Notes 2, 3)					
D060	IIL	I/O ports (except RA2, RA3)	_	_	±1	μΑ	Vss ≤ VPIN ≤ VDD, I/O Pin at hi-impedance PORTB weak pull-ups dis- abled
D061		MCLR	_	_	±2	μΑ	VPIN = Vss or VPIN = VDD
D062		RA2, RA3			±2	μΑ	Vss ≤ VRA2, VRA3 ≤ 12V
D063		OSC1, TEST	_	_	±1	μΑ	Vss ≤ VPIN ≤ VDD
D064		MCLR	_	_	10	μΑ	VMCLR = VPP = 12V (when not programming)
D070	IPURB	PORTB weak pull-up current	60	200	400	μΑ	VPIN = VSS, 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.

## Standard Operating Conditions (unless otherwise stated) Operating temperature

DC CHARACTERISTICS

-40°C  $\leq$  TA  $\leq$  +85°C for industrial and 0°C  $\leq$  TA  $\leq$  +70°C for commercial

Operating voltage VDD range as described in Section 17.1

Parameter							
No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
		Output Low Voltage					
D080	Vol	I/O ports (except RA2 and RA3)	_	_	0.1VDD	V	IOL = 4 mA
D081		with TTL buffer	_	_	0.4	V	IOL = 6 mA, VDD = 4.5V Note 6
D082		RA2 and RA3	_	_	3.0	V	IOL = 60.0 mA, VDD = 5.5V
D083		OSC2/CLKOUT	_	_	0.4	V	IOL = 2 mA, VDD = 4.5V
		(RC and EC osc modes)					
		Output High Voltage (Note 3)					
D090	Vон	I/O ports (except RA2 and RA3)	0.9Vdd	_	_	V	IOH = -2 mA
D091		with TTL buffer	2.4	_	_	V	IOH = -6.0 mA, VDD = 4.5V Note 6
D092		RA2 and RA3	_	_	12	V	Pulled-up to externally applied voltage
D093		OSC2/CLKOUT	2.4	_	_	V	IOH = -5 mA, VDD = 4.5V
		(RC and EC osc modes)					
		Capacitive Loading Specs on Output Pins					
D100	Cosc2	OSC2 pin	_	_	25 ††	pF	In EC or RC osc modes when OSC2 pin is outputting CLKOUT. External clock is used to drive OSC1.
D101	Сю	All I/O pins and OSC2 (in RC mode)	_	_	50 ††	pF	
D102	CAD	System Interface Bus (PORTC, PORTD and PORTE)	_	_	100 ††	pF	In Microprocessor or Extended Microcontroller mode

- 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.

DC CHADA	CTEDI		Standard C Operating to		•	ns (ur	nless otherwise stated)
DC CHARA	CIERI	51105			-40°C	; ≤ <b>T</b> A	≤ +40°C
			Operating v	oltage Vi	DD range a	s desc	ribed in Section 17.1
Parameter							
No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
		Internal Program Memory Programming Specs (Note 4)					
D110	VPP	Voltage on MCLR/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 MCLR/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/exter- nal 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 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 is not recommended.
  - 6: For TTL buffers, the better of the two specifications may be used.

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

### 17.3 <u>Timing Parameter Symbology</u>

The timing parameter symbols have been created using one of the following formats:

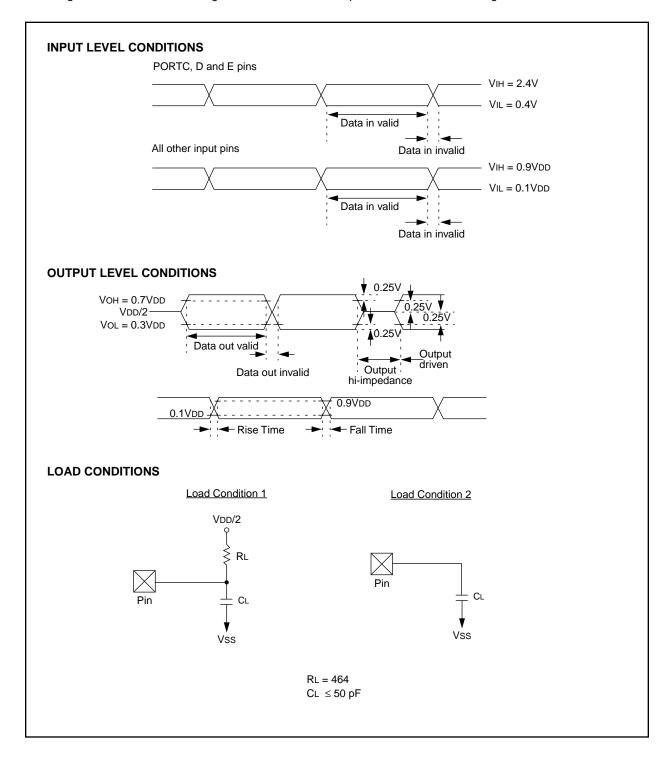
1. TppS2ppS

2. TppS

2. TppS				
Т				
F	Frequency	Т	Time	
Lowerd	case symbols (pp) and their meanings:			
рр				
ad	Address/Data	ost	Oscillator Start-up Timer	
al	ALE	pwrt	Power-up Timer	
СС	Capture1 and Capture2	rb	PORTB	
ck	CLKOUT or clock	rd	RD	
dt	Data in	rw	$\overline{RD}$ or $\overline{WR}$	
in	INT pin	tO	TOCKI	
io	I/O port	t123	TCLK12 and TCLK3	
mc	MCLR	wdt	Watchdog Timer	
oe	ŌĒ	wr	WR	
os	OSC1			
Upper	case symbols and their meanings:			
S				
D	Driven	L	Low	
E	Edge	P	Period	
F	Fall	R	Rise	
Н	High	V	Valid	
1	Invalid (Hi-impedance)	Z	Hi-impedance	

#### FIGURE 17-1: PARAMETER MEASUREMENT INFORMATION

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



#### 17.4 Timing Diagrams and Specifications

#### FIGURE 17-2: EXTERNAL CLOCK TIMING

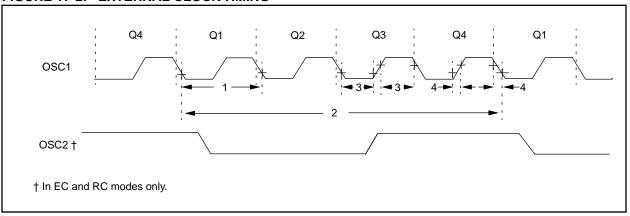


TABLE 17-2: EXTERNAL CLOCK TIMING REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Тур†	Max	Units	Conditions
	Fosc	External CLKIN Frequency	DC	_	16	MHz	EC osc mode - PIC17C42-16
		(Note 1)	DC	_	25	MHz	- PIC17C42-25
		Oscillator Frequency	DC	_	4	MHz	RC osc mode
		(Note 1)	1	_	16	MHz	XT osc mode - PIC17C42-16
			1	_	25	MHz	- PIC17C42-25
			DC	_	2	MHz	LF osc mode
1	Tosc	External CLKIN Period	62.5	_	_	ns	EC osc mode - PIC17C42-16
		(Note 1)	40	_	_	ns	- PIC17C42-25
		Oscillator Period	250	_	_	ns	RC osc mode
		(Note 1)	62.5	_	1,000	ns	XT osc mode - PIC17C42-16
			40	_	1,000	ns	- PIC17C42-25
			500	_	_	ns	LF osc mode
2	Tcy	Instruction Cycle Time (Note 1)	160	4/Fosc	DC	ns	
3	TosL,	Clock in (OSC1) High or Low Time	10 ‡	_	_	ns	EC oscillator
	TosH						
4	TosR,	Clock in (OSC1) Rise or Fall Time	_	_	5‡	ns	EC oscillator
	TosF						

<sup>†</sup> 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: Instruction cycle period (TcY) equals four times the input oscillator time-base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at "min." values with an external clock applied to the OSC1 pin.

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

<sup>†</sup> These parameters are for design guidance only and are not tested, nor characterized.

FIGURE 17-3: CLKOUT AND I/O TIMING

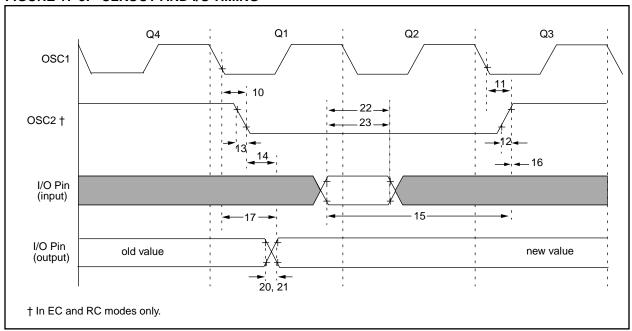


TABLE 17-3: CLKOUT AND I/O TIMING REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Тур†	Max	Units	Conditions
10	TosH2ckL	OSC1↑ to CLKOUT↓	_	15 ‡	30 ‡	ns	Note 1
11	TosH2ckH	OSC1↑ to CLKOUT↑	_	15 ‡	30 ‡	ns	Note 1
12	TckR	CLKOUT rise time	_	5‡	15 ‡	ns	Note 1
13	TckF	CLKOUT fall time	_	5‡	15 ‡	ns	Note 1
14	TckH2ioV	CLKOUT <sup>↑</sup> to Port out valid	_	_	0.5Tcy + 20‡	ns	Note 1
15	TioV2ckH	Port in valid before CLKOUT↑	0.25Tcy + 25 ‡	_	_	ns	Note 1
16	TckH2ioI	Port in hold after CLKOUT↑	0 ‡	_	_	ns	Note 1
17	TosH2ioV	OSC1↑ (Q1 cycle) to Port out valid	_	_	100 ‡	ns	
20	TioR	Port output rise time	_	10 ‡	35 ‡	ns	
21	TioF	Port output fall time	_	10 ‡	35 ‡	ns	
22	TinHL	INT pin high or low time	25 *	_	_	ns	
23	TrbHL	RB7:RB0 change INT high or low time	25 *	_	_	ns	

<sup>\*</sup> These parameters are characterized but not tested.

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

<sup>†</sup> These parameters are for design guidance only and are not tested, nor characterized.

Note 1: Measurements are taken in EC Mode where OSC2 output = 4 x Tosc = Tcy.

FIGURE 17-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING

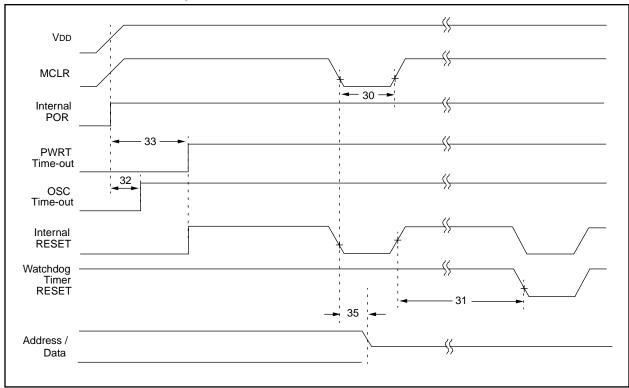


TABLE 17-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Тур†	Max	Units	Conditions
30	TmcL	MCLR Pulse Width (low)	100 *	_	_	ns	
31	Twdt	Watchdog Timer Time-out Period (Prescale = 1)	5 *	12	25 *	ms	
32	Tost	Oscillation Start-up Timer Period		1024 Tosc §		ms	Tosc = OSC1 period
33	Tpwrt	Power-up Timer Period	40 *	96	200 *	ms	
35	TmcL2adl	MCLR to System Interface bus (AD15:AD0) invalid	_	_	100 *	ns	

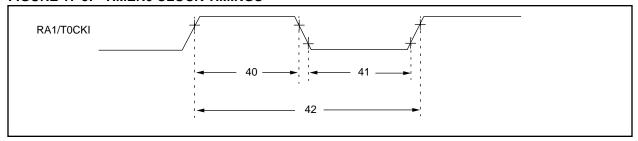
<sup>\*</sup> These parameters are characterized but not tested.

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

<sup>‡</sup> These parameters are for design guidance only and are not tested, nor characterized.

<sup>§</sup> This specification ensured by design.

#### FIGURE 17-5: TIMERO CLOCK TIMINGS



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

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

- \* These parameters are characterized but not tested.
- † Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.
- § This specification ensured by design.

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

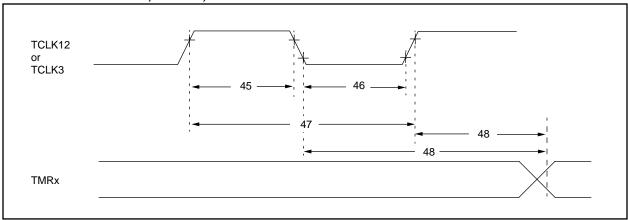
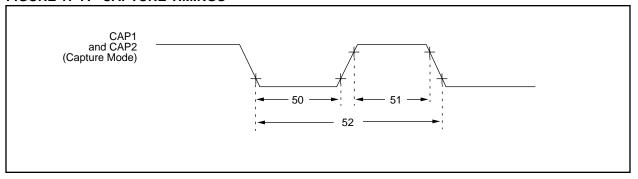


TABLE 17-6: TIMER1, TIMER2, AND TIMER3 CLOCK REQUIREMENTS

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

- These parameters are characterized but not tested.
- † Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.
- § This specification ensured by design.

#### FIGURE 17-7: CAPTURE TIMINGS

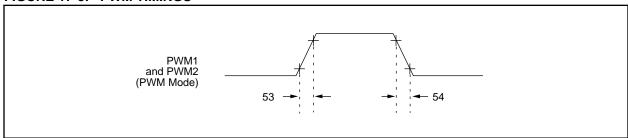


#### **TABLE 17-7: CAPTURE REQUIREMENTS**

Parameter							
No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
50	TccL	Capture1 and Capture2 input low time	10 *	_	_	ns	
51	TccH	Capture1 and Capture2 input high time	10 *	_	_	ns	
52	TccP	Capture1 and Capture2 input period	2 Tcy § N		_	ns	N = prescale value (4 or 16)

- \* These parameters are characterized but not tested.
- † Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.
- § This specification ensured by design.

#### FIGURE 17-8: PWM TIMINGS

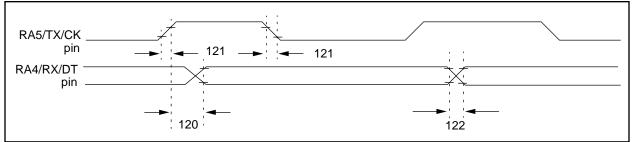


#### **TABLE 17-8: PWM REQUIREMENTS**

Parameter No.	Sym	Characteristic	Min	Тур†	Max	Units	Conditions
53	TccR	PWM1 and PWM2 output rise time	_	10 *	35 *§	ns	
54	TccF	PWM1 and PWM2 output fall time	_	10 *	35 *§	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.

#### FIGURE 17-9: USART MODULE: SYNCHRONOUS TRANSMISSION (MASTER/SLAVE) TIMING



#### TABLE 17-9: SERIAL PORT SYNCHRONOUS TRANSMISSION REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Тур†	Max	Units	Conditions
120	TckH2dtV	SYNC XMIT (MASTER & SLAVE) Clock high to data out valid	_	_	65	ns	
121	TckRF	Clock out rise time and fall time (Master Mode)	_	10	35	ns	
122	TdtRF	Data out rise time and fall time	_	10	35	ns	

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

#### FIGURE 17-10: USART MODULE: SYNCHRONOUS RECEIVE (MASTER/SLAVE) TIMING

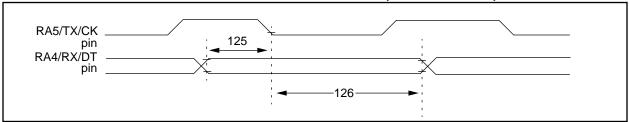


TABLE 17-10: SERIAL PORT SYNCHRONOUS RECEIVE REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Тур†	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	

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

FIGURE 17-11: MEMORY INTERFACE WRITE TIMING

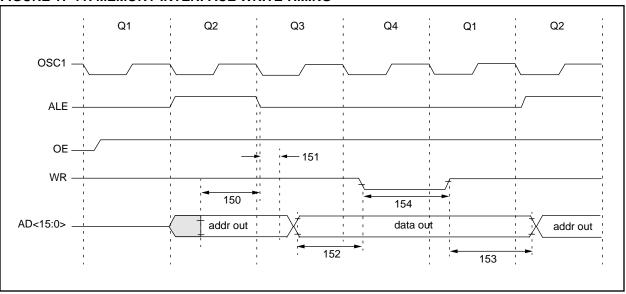


TABLE 17-11: MEMORY INTERFACE WRITE REQUIREMENTS

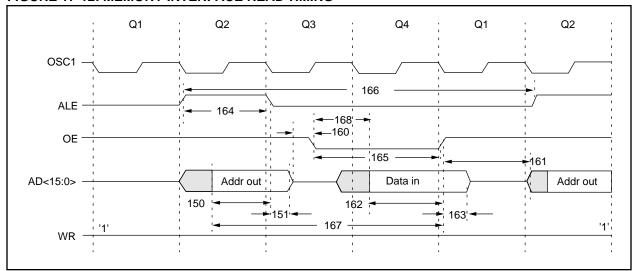
Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
150	TadV2alL	AD<15:0> (address) valid to ALE↓ (address setup time)	0.25Tcy - 30	_	_	ns	
151	TalL2adl	ALE↓ to address out invalid (address hold time)	0	_	_	ns	
152	TadV2wrL	Data out valid to WR↓ (data setup time)	0.25Tcy - 40	_	_	ns	
153	TwrH2adl	WR↑ to data out invalid (data hold time)	_	0.25Tcy §	_	ns	
154	TwrL	WR pulse width	_	0.25Tcy §	_	ns	

<sup>\*</sup> These parameters are characterized but not tested.

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

<sup>§</sup> This specification is guaranteed by design.

#### FIGURE 17-12: MEMORY INTERFACE READ TIMING



**TABLE 17-12: MEMORY INTERFACE READ REQUIREMENTS** 

Parameter No.	Sym	Characteristic	Min	Тур†	Max	Units	Conditions
150	TadV2alL	AD<15:0> (address) valid to ALE↓ (address setup time)	0.25Tcy - 30	_	_	ns	
151	TalL2adl	ALE↓ to address out invalid (address hold time)	5*	_	_	ns	
160	TadZ2oeL	AD<15:0> high impedance to OE↓	0*	_	_	ns	
161	ToeH2adD	OE↑ to AD<15:0> driven	0.25Tcy - 15	_	_	ns	
162	TadV2oeH	Data in valid before <del>OE</del> ↑ (data setup time)	35	_	_	ns	
163	ToeH2adI	OE↑to data in invalid (data hold time)	0	_	_	ns	
164	TalH	ALE pulse width	_	0.25Tcy §	_	ns	
165	ToeL	OE pulse width	0.5Tcy - 35 §	_	_	ns	
166	TalH2alH	ALE↑ to ALE↑ (cycle time)	_	Tcy §	_	ns	
167	Tacc	Address access time	_	_	0.75 Tcy-40	ns	
168	Toe	Output enable access time (OE low to Data Valid)	_	_	0.5 Tcy - 60	ns	

<sup>\*</sup> These parameters are characterized but not tested.

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

<sup>§</sup> This specification guaranteed by design.

#### 18.0 PIC17C42 DC AND AC CHARACTERISTICS

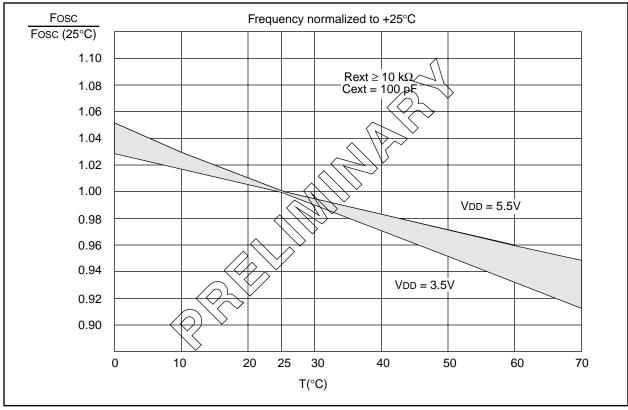
The graphs and tables provided in this section are for design guidance and are not tested or guaranteed. In some graphs or tables the data presented are outside specified operating range (e.g. outside specified VDD range). This is for information only and devices are ensured to operate properly only 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 (mean +  $3\sigma$ ) and (mean -  $3\sigma$ ) respectively where  $\sigma$  is standard deviation.

TABLE 18-1: PIN CAPACITANCE PER PACKAGE TYPE

Din Name		Typical Capacitance (pF)								
Pin Name	40-pin DIP	44-pin PLCC	44-pin MQFP	44-pin TQFP						
All pins, except MCLR, VDD, and VSS	10	10	10	10						
MCLR pin	20	20	20	20						

FIGURE 18-1: TYPICAL RC OSCILLATOR FREQUENCY vs. TEMPERATURE



© 1996 Microchip Technology Inc.

FIGURE 18-2: TYPICAL RC OSCILLATOR FREQUENCY vs. VDD

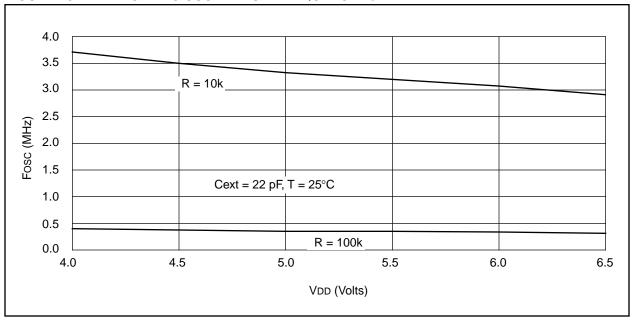


FIGURE 18-3: TYPICAL RC OSCILLATOR FREQUENCY vs. VDD

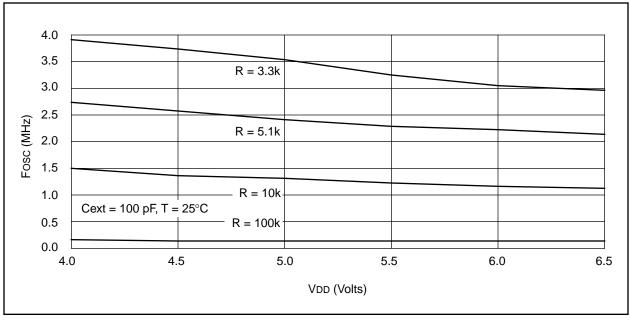


FIGURE 18-4: TYPICAL RC OSCILLATOR FREQUENCY vs. VDD

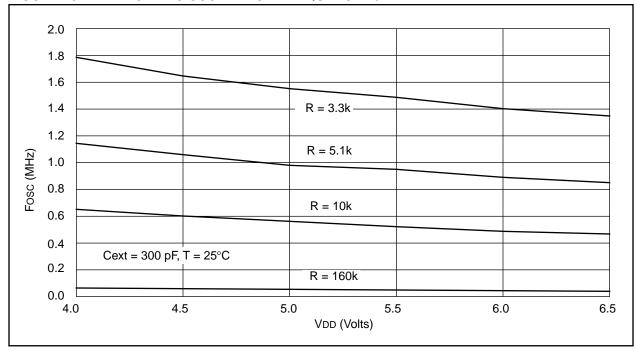


TABLE 18-2: RC OSCILLATOR FREQUENCIES

Cext	Rext	Average Fosc @ 5V, 25°C		
22 pF	10k	3.33 MHz	± 12%	
	100k	353 kHz	± 13%	
100 pF	3.3k	3.54 MHz	± 10%	
	5.1k	2.43 MHz	± 14%	
	10k	1.30 MHz	± 17%	
	100k	129 kHz	± 10%	
300 pF	3.3k	1.54 MHz	± 14%	
	5.1k	980 kHz	± 12%	
	10k	564 kHz	± 16%	
	160k	35 kHz	± 18%	

Applicable Devices 42 R42 42A 43 R43 44

FIGURE 18-5: TRANSCONDUCTANCE (gm) OF LF OSCILLATOR vs. VDD

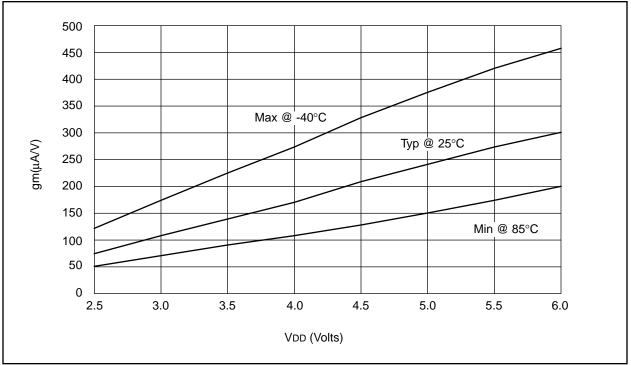


FIGURE 18-6: TRANSCONDUCTANCE (gm) OF XT OSCILLATOR vs. VDD

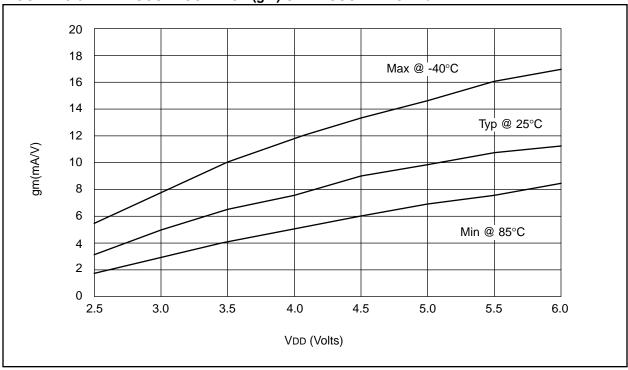


FIGURE 18-7: TYPICAL IDD vs. FREQUENCY (EXTERNAL CLOCK 25°C)

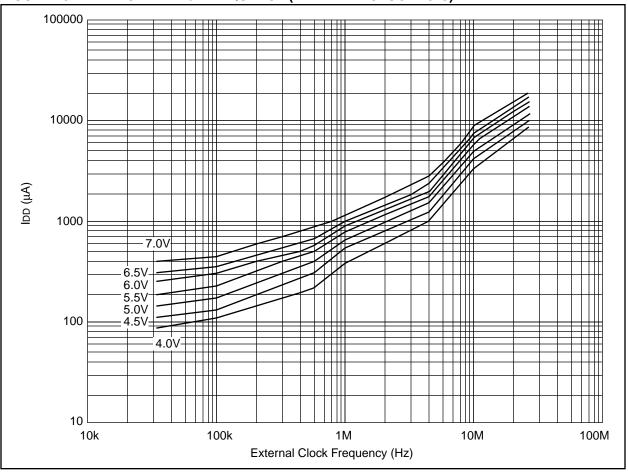


FIGURE 18-8: MAXIMUM IDD vs. FREQUENCY (EXTERNAL CLOCK 125°C TO -40°C)

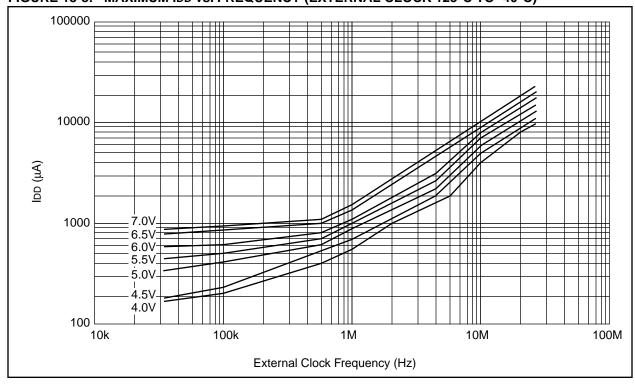


FIGURE 18-9: TYPICAL IPD vs. VDD WATCHDOG DISABLED 25°C

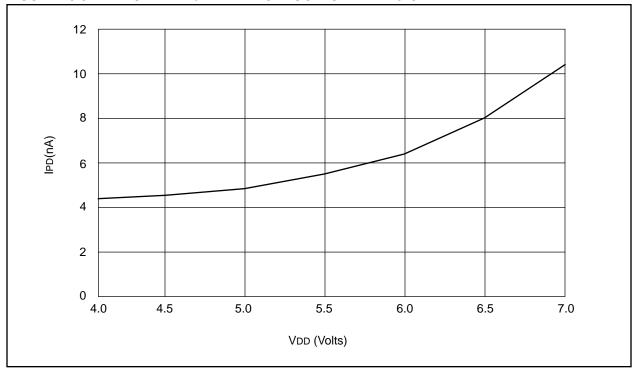


FIGURE 18-10: MAXIMUM IPD vs. VDD WATCHDOG DISABLED

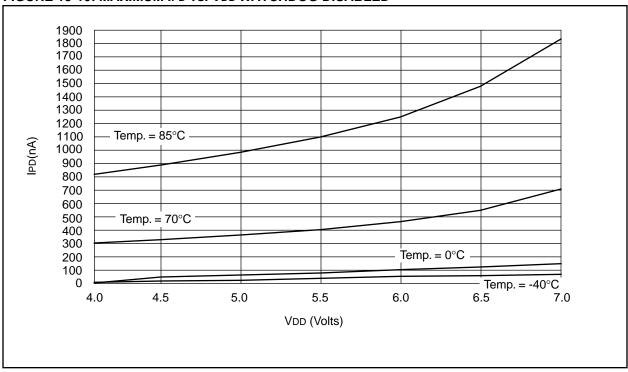


FIGURE 18-11: TYPICAL IPD vs. VDD WATCHDOG ENABLED 25°C

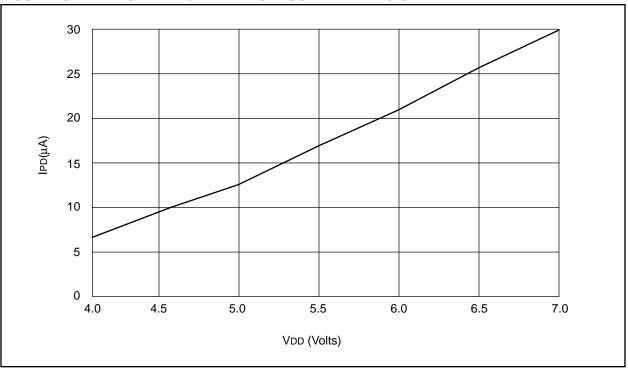


FIGURE 18-12: MAXIMUM IPD vs. VDD WATCHDOG ENABLED

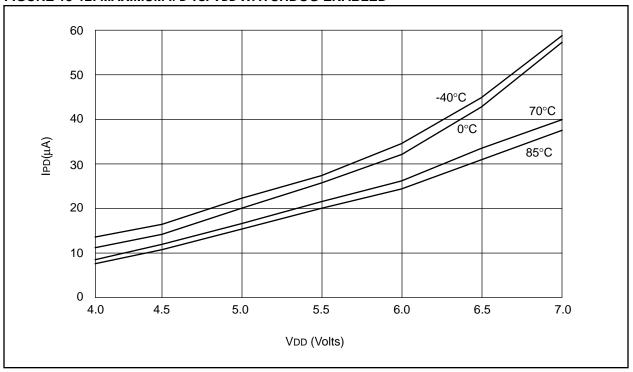


FIGURE 18-13: WDT TIMER TIME-OUT PERIOD vs. VDD

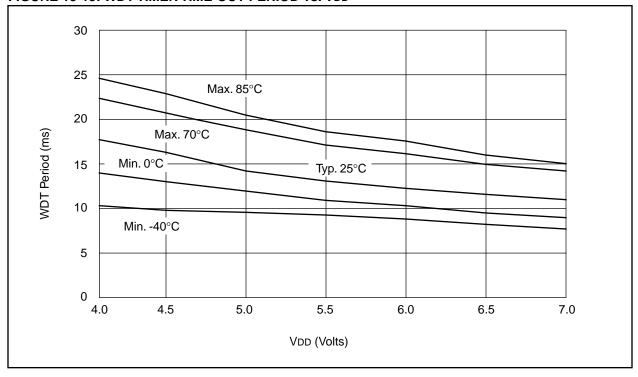


FIGURE 18-14: IOH vs. VOH, VDD = 3V

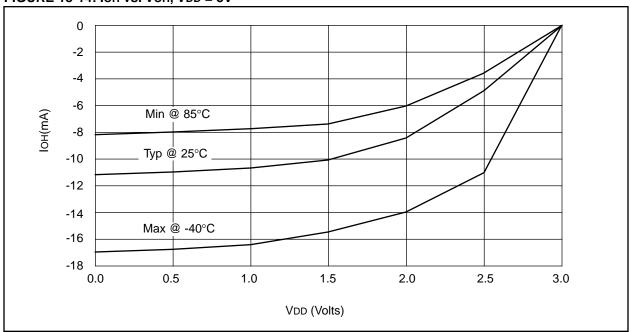


FIGURE 18-15: IOH vs. VOH, VDD = 5V

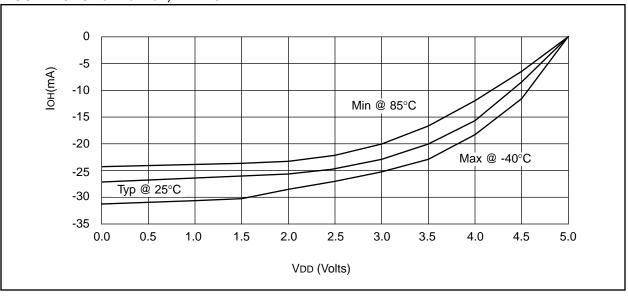
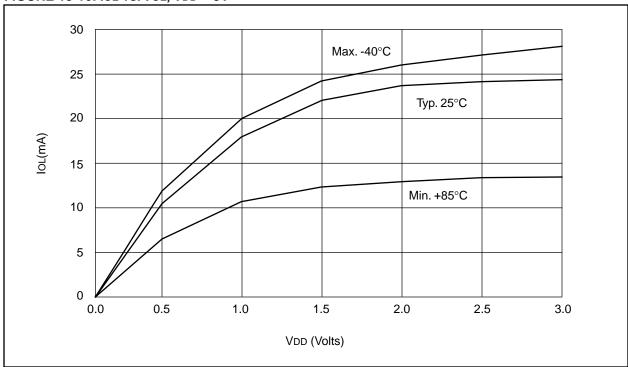


FIGURE 18-16: IOL vs. VOL, VDD = 3V



**Applicable Devices** 42 R42 42A 43 R43 44

FIGURE 18-17: IoL vs. Vol, VDD = 5V

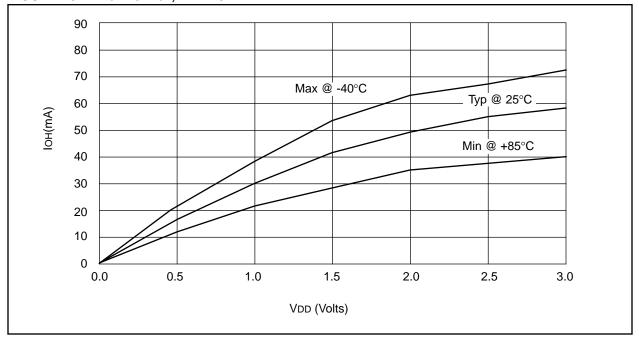


FIGURE 18-18: VTH (INPUT THRESHOLD VOLTAGE) OF I/O PINS (TTL) VS. VDD

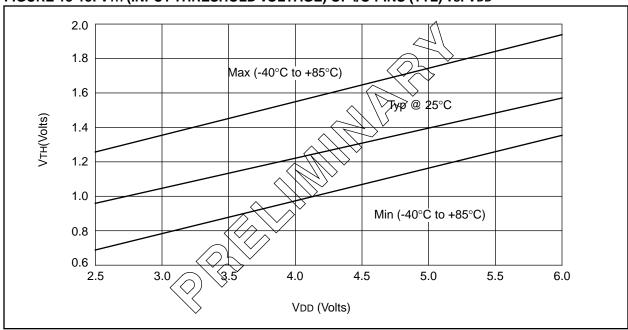


FIGURE 18-19: VTH, VIL of I/O PINS (SCHMITT TRIGGER) vs. VDD

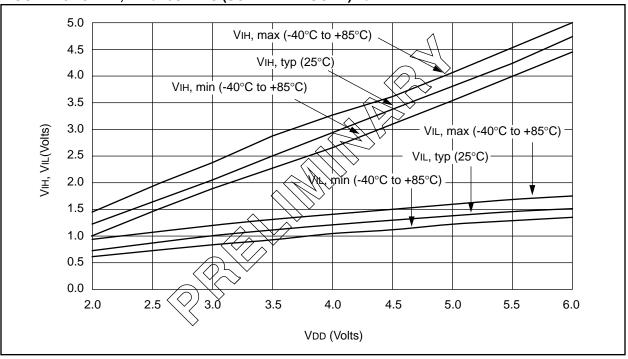
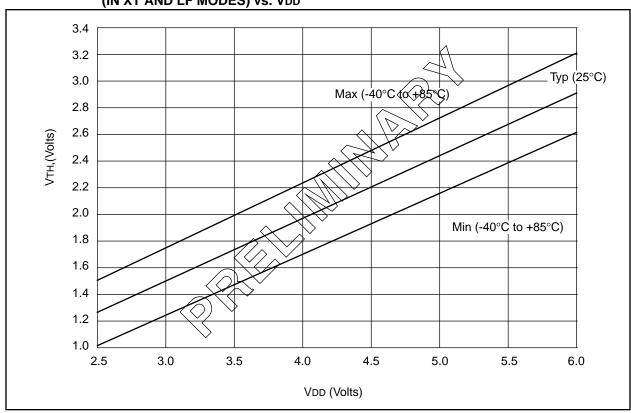


FIGURE 18-20: VTH (INPUT THRESHOLD VOLTAGE) OF OSC1 INPUT (IN XT AND LF MODES) vs. VDD



NOTES:

#### PIC17CR42/42A/43/R43/44 ELECTRICAL CHARACTERISTICS

#### **Absolute Maximum Ratings †**

Ambient temperature under bias	55 to +125°C
Storage temperature	65°C to +150°C
/oltage on VDD with respect to Vss	0 to +7.5\
/oltage on MCLR with respect to Vss (Note 2)	0.6V to +14V
/oltage on RA2 and RA3 with respect to Vss	0.6V to +14V
/oltage on all other pins with respect to Vss	0.6V to VDD + 0.6V
Total power dissipation (Note 1)	1.0W
Maximum current out of Vss pin(s) - total	250 mA
Maximum current into VDD pin(s) - total	200 mA
nput clamp current, liκ (Vi < 0 or Vi > VDD)	±20 mA
Dutput clamp current, loκ (Vo < 0 or Vo > VDD)	±20 mA
Maximum output current sunk by any I/O pin (except RA2 and RA3)	35 mA
Maximum output current sunk by RA2 or RA3 pins	60 mA
Maximum output current sourced by any I/O pin	20 mA
Maximum current sunk by PORTA and PORTB (combined)	150 mA
Maximum current sourced by PORTA and PORTB (combined)	100 m/
Maximum current sunk by PORTC, PORTD and PORTE (combined)	150 m <i>A</i>
Maximum current sourced by PORTC, PORTD and PORTE (combined)	100 m/
<b>Note 1:</b> Power dissipation is calculated as follows: Pdis = $VDD \times \{IDD - \sum IOH\} + \sum \{(VDD - VDD)\}$	$\sqrt{OH} \times IOH + \Sigma(VOL \times IOL)$

Note 2: Voltage spikes below Vss at the MCLR pin, inducing currents greater than 80 mA, may cause latch-up. Thus, a series resistor of  $50-100\Omega$  should be used when applying a "low" level to the  $\overline{MCLR}$  pin rather than pulling this pin directly to Vss.

† NOTICE: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

© 1996 Microchip Technology Inc.

TABLE 19-1: CROSS REFERENCE OF DEVICE SPECS FOR OSCILLATOR CONFIGURATIONS AND FREQUENCIES OF OPERATION (COMMERCIAL DEVICES)

osc	PIC17LCR42-08 PIC17LC42A-08 PIC17LC43-08 PIC17LCR43-08 PIC17LC44-08	PIC17CR42-16 PIC17C42A-16 PIC17C43-16 PIC17CR43-16 PIC17C44-16	PIC17CR42-25 PIC17C42A-25 PIC17C43-25 PIC17CR43-25 PIC17C44-25	PIC17CR42-33 PIC17C42A-33 PIC17C43-33 PIC17CR43-33 PIC17C44-33	JW Devices (Ceramic Windowed Devices)
RC	VDD: 2.5V to 6.0V	VDD: 4.5V to 6.0V	VDD: 4.5V to 6.0V	VDD: 4.5V to 6.0V	VDD: 4.5V to 6.0V
	IDD: 6 mA max.	IDD: 6 mA max.	IDD: 6 mA max.	IDD: 6 mA max.	IDD: 6 mA max.
	IPD: 5 μA max. at 5.5V	IPD: 5 μA max. at 5.5V	IPD: 5 μA max. at 5.5V	IPD: 5 μA max. at 5.5V	IPD: 5 µA max. at 5.5V
	WDT disabled	WDT disabled	WDT disabled	WDT disabled	WDT disabled
	Freq: 4 MHz max.	Freq: 4 MHz max.	Freq: 4 MHz max.	Freq: 4 MHz max.	Freq: 4 MHz max.
X	VDD: 2.5V to 6.0V	VDD: 4.5V to 6.0V	VDD: 4.5V to 6.0V	VDD: 4.5V to 6.0V	VDD: 4.5V to 6.0V
	IDD: 12 mA max.	IDD: 24 mA max.	IDD: 38 mA max.	IDD: 38 mA max.	IDD: 38 mA max.
	IPD: 5 μA max. at 5.5V	IPD: 5 µA max. at 5.5V	IPD: 5 μA max. at 5.5V	IPD: 5 μA max. at 5.5V	IPD: 5 µA max. at 5.5V
	WDT disabled	WDT disabled	WDT disabled	WDT disabled	WDT disabled
	Freq: 8 MHz max.	Freq: 16 MHz max.	Freq: 25 MHz max.	Freq: 33 MHz max.	Freq: 33 MHz max.
EC	EC Vpp: 2.5V to 6.0V	VDD: 4.5V to 6.0V	VDD: 4.5V to 6.0V	VDD: 4.5V to 6.0V	VDD: 4.5V to 6.0V
	Ipp: 12 mA max.	IDD: 24 mA max.	IDD: 38 mA max.	IDD: 38 mA max.	IDD: 38 mA max.
	Ipp: 5 μA max. at 5.5V	IPD: 5 µA max. at 5.5V	IPD: 5 μA max. at 5.5V	IPD: 5 μA max. at 5.5V	IPD: 5 µA max. at 5.5V
	WDT disabled	WDT disabled	WDT disabled	WDT disabled	WDT disabled
	Freq: 8 MHz max.	Freq: 16 MHz Max	Freq: 25 MHz max.	Freq: 33 MHz max.	Freq: 33 MHz max.
F	LF         VDD:         2.5V to 6.0V         VDD:         4.5V to 6.0V         VDD:         4.5V to 6.0V         VDD:         4.5V to 6.0V           IDD:         150 μA max. at 32 kHz         IDD:         95 μA typ. at 32 kHz         IDD:         95 μA typ. at 32 kHz         IDD:         95 μA typ. at 32 kHz           IPD:         5 μA max. at 5.5V         IPD:         < 1 μA typ. at 5.5V	VDD: 4.5V to 6.0V IDD: 95 μA typ. at 32 kHz IPD: < 1 μA typ. at 5.5V WDT disabled Freq: 2 MHz max.	VDD: 4.5V to 6.0V IDD: 95 μA typ. at 32 kHz IPD: <1 μA typ. at 5.5V WDT disabled Freq: 2 MHz max.		VDD: 2.5V to 6.0V IDD: 150 μA max. at 32 kHz IPD: 5 μA max. at 5.5V WDT disabled Freq: 2 MHz max.

The shaded sections indicate oscillator selections which are tested for functionality, but not for MIN/MAX specifications. It is recommended that the user select the device type that ensures the specifications required.

#### 19.1 DC CHARACTERISTICS:

PIC17CR42/42A/43/R43/44-16 (Commercial, Industrial) PIC17CR42/42A/43/R43/44-25 (Commercial, Industrial) PIC17CR42/42A/43/R43/44-33 (Commercial, Industrial)

DC CHADAC	FEDICT	00	Standard Operating	-	_		s (unless otherwise stated)
DC CHARAC	I EKISTI	LS .				-40°C	≤ TA ≤ +85°C for industrial and
						0°C	$\leq$ TA $\leq$ +70°C for commercial
Parameter							
No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
D001	VDD	Supply Voltage	4.5	_	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 *	I	_	mV/ms	See section on Power-on Reset for details
D010	IDD	Supply Current	_	3	6	mA	Fosc = 4 MHz (Note 4)
D011		(Note 2)	_	6	12 *	mA	Fosc = 8 MHz
D012			_	11	24 *	mA	Fosc = 16 MHz
D013			_	19	38	mA	Fosc = 25 MHz
D015			_	25	50	mA	Fosc = 33 MHz
D014			_	95	150	μΑ	Fosc = 32 kHz,
							WDT enabled (EC osc configuration)
D020	IPD	Power-down	_	10	40	μΑ	VDD = 5.5V, WDT enabled
D021		Current (Note 3)	-	< 1	5	μΑ	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, TOCKI = 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: VDD / (2 • R). For capacitive loads, the current can be estimated (for an individual I/O pin) as (CL • VDD) • f

CL = 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 and VSS.
- 4: For RC osc configuration, current through Rext is not included. The current through the resistor can be estimated by the formula IR = VDD/2Rext (mA) with Rext in kOhm.

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

DC CHARA	<b>Standard</b> Operating	•	_		s (unless otherwise stated)  ≤ TA ≤ +85°C for industrial and ≤ TA ≤ +70°C for commercial		
Parameter No.	Sym	Characteristic	Min	Тур†	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: VDD / (2 • R). For capacitive loads, the current can be estimated (for an individual I/O pin) as (CL • VDD) • f

CL = 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 IR = VDD/2Rext (mA) with Rext in kOhm.

#### 19.3 DC CHARACTERISTICS:

**DC CHARACTERISTICS** 

PIC17CR42/42A/43/R43/44-16 (Commercial, Industrial) PIC17CR42/42A/43/R43/44-25 (Commercial, Industrial) PIC17CR42/42A/43/R43/44-33 (Commercial, Industrial) PIC17LCR42/42A/43/R43/44-08 (Commercial, Industrial)

Standard Operating Conditions (unless otherwise stated)

Operating temperature

 $-40^{\circ}$ C  $\leq$  TA  $\leq$  +85 $^{\circ}$ C for industrial and 0°C  $\leq$  TA  $\leq$  +70°C for commercial

Operating voltage VDD range as described in Section 19.1

Parameter					3 3		ribed in Occilon 15.1
No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
		Input Low Voltage					
	VIL	I/O ports					
D030		with TTL buffer	Vss	_	0.8		$4.5V \le VDD \le 5.5V$
			Vss	_	0.2Vdd		2.5V ≤ VDD ≤ 4.5V
D031		with Schmitt Trigger buffer	Vss	_	0.2Vdd	V	
D032		MCLR, OSC1 (in EC and RC mode)	Vss	_	0.2VDD	V	Note1
D033		OSC1 (in XT, and LF mode)	_	0.5VDD	_	V	
		Input High Voltage					
	VIH	I/O ports					
D040		with TTL buffer	2.0	_	VDD		$4.5V \le VDD \le 5.5V$
			1 + 0.2VDD	_	VDD		$2.5V \le VDD \le 4.5V$
D041		with Schmitt Trigger buffer	0.8VDD	_	VDD	V	
D042		MCLR	0.8VDD	_	VDD	V	Note1
D043		OSC1 (XT, and LF mode)	_	0.5VDD	_	V	
D050	VHYS	Hysteresis of	0.15VDD *	-	-	V	
		Schmitt Trigger inputs					
		Input Leakage Current (Notes 2, 3)					
D060	lıL	I/O ports (except RA2, RA3)	_	_	±1	μΑ	Vss ≤ VPIN ≤ VDD, I/O Pin at hi-impedance PORTB weak pull-ups disabled
D061		MCLR	_	_	±2	μΑ	VPIN = Vss or VPIN = VDD
D062		RA2, RA3			±2	μΑ	Vss ≤ VRA2, VRA3 ≤ 12V
D063		OSC1, TEST (EC, RC modes)	_	_	±1	μΑ	$Vss \le VPIN \le VDD$
D063B		OSC1, TEST (XT, LF modes)	_	_	VPIN	μΑ	RF $\geq$ 1 M $\Omega$ , see Figure 14.2
D064		MCLR	_	_	10	μА	VMCLR = VPP = 12V (when not programming)
D070		PORTB weak pull-up current	60	200	400	μА	VPIN = VSS, $\overline{RBPU} = 0$ 4.5V $\leq$ VDD $\leq$ 6.0V

- 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 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 is not recommended.
  - 6: For TTL buffers, the better of the two specifications may be used.

### Standard Operating Conditions (unless otherwise stated) Operating temperature

#### DC CHARACTERISTICS

 $-40^{\circ}\text{C} \le \text{TA} \le +85^{\circ}\text{C}$  for industrial and  $0^{\circ}\text{C} \le \text{TA} \le +70^{\circ}\text{C}$  for commercial

Operating voltage VDD range as described in Section 19.1

Parameter							
No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
		Output Low Voltage					
D080	VOL	I/O ports (except RA2 and RA3)					IOL = VDD/1.250 mA
			_	_	0.1Vdd	V	4.5V ≤ VDD ≤ 6.0V
			_	_	0.1VDD *	V	VDD = 2.5V
D081		with TTL buffer	-	_	0.4	V	IOL = 6 mA, VDD = 4.5V Note 6
D082		RA2 and RA3	_	_	3.0	V	IOL = 60.0  mA, VDD = 6.0V
D083		OSC2/CLKOUT	_	_	0.4	V	IOL = 1 mA, VDD = 4.5V
D084		(RC and EC osc modes)	-	_	0.1VDD *	V	IOL = VDD/5 mA (PIC17LC43/LC44 only)
		Output High Voltage (Note 3)					
D090	Vон	I/O ports (except RA2 and RA3)					IOH = -VDD/2.500 mA
			0.9Vdd	_	_	V	4.5V ≤ VDD ≤ 6.0V
			0.9Vdd *	_	_	V	VDD = 2.5V
D091		with TTL buffer	2.4	_	_	V	IOH = -6.0 mA, VDD=4.5V Note 6
D092		RA2 and RA3	-	_	12	V	Pulled-up to externally applied voltage
D093		OSC2/CLKOUT	2.4	_	_	V	IOH = -5 mA, VDD = 4.5V
D094		(RC and EC osc modes)	0.9VDD *	_	_	V	IOH = -VDD/5 mA (PIC17LC43/LC44 only)
		Capacitive Loading Specs on Output Pins					
D100	Cosc <sub>2</sub>	OSC2/CLKOUT pin	-	_	25	pF	In EC or RC osc modes when OSC2 pin is outputting CLKOUT. external clock is used to drive OSC1.
D101	Сю	All I/O pins and OSC2 (in RC mode)	-	_	50	pF	
D102	CAD	System Interface Bus (PORTC, PORTD and PORTE)	-	_	50	pF	In Microprocessor or Extended Microcontroller mode

- \* 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 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 is not recommended.
  - 6: For TTL buffers, the better of the two specifications may be used.

# Standard Operating Conditions (unless otherwise stated) Operating temperature

DC CHARACTERISTICS

 $-40^{\circ}\text{C} \leq \text{TA} \leq +40^{\circ}\text{C}$ 

Operating voltage VDD range as described in Section 19.1

Parameter							
No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
		Internal Program Memory Programming Specs (Note 4)					
D110	VPP	Voltage on MCLR/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 MCLR/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.
  - 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 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.

### Applicable Devices | 42 | R42 | 42A | 43 | R43 | 44

### 19.4 <u>Timing Parameter Symbology</u>

The timing parameter symbols have been created following one of the following formats:

1. TppS2ppS	3. Tcc:st	(I <sup>2</sup> C specifications only)
2.TppS	4.Ts	(I <sup>2</sup> C specifications only)

T				
F	Frequency	T	Time	

Lowercase symbols (pp) and their meanings:

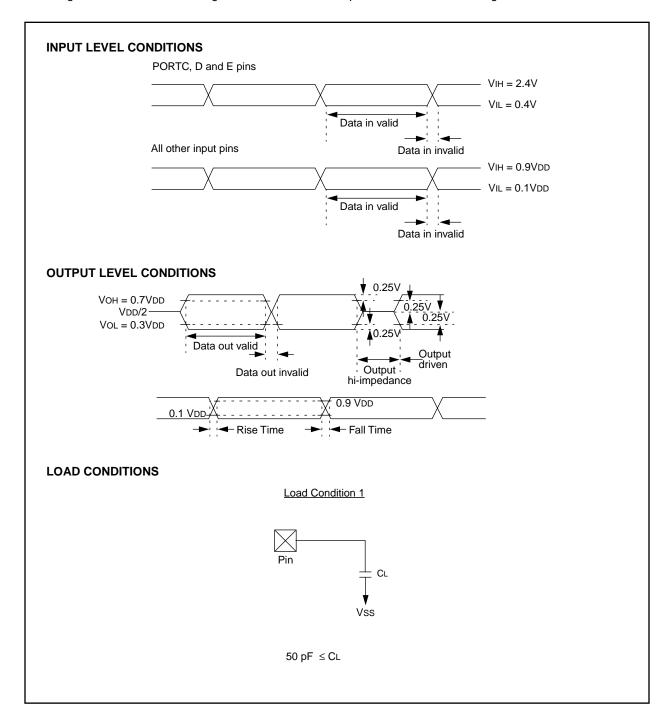
рр			
ad	Address/Data	ost	Oscillator Start-Up Timer
al	ALE	pwrt	Power-Up Timer
СС	Capture1 and Capture2	rb	PORTB
ck	CLKOUT or clock	rd	RD
dt	Data in	rw	$\overline{RD}$ or $\overline{WR}$
in	INT pin	t0	T0CKI
io	I/O port	t123	TCLK12 and TCLK3
mc	MCLR	wdt	Watchdog Timer
oe	ŌĒ	wr	WR
os	OSC1		

Uppercase symbols and their meanings:

S			
D	Driven	L	Low
E	Edge	P	Period
F	Fall	R	Rise
Н	High	V	Valid
1	Invalid (Hi-impedance)	Z	Hi-impedance

### FIGURE 19-1: PARAMETER MEASUREMENT INFORMATION

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



### 19.5 <u>Timing Diagrams and Specifications</u>

#### FIGURE 19-2: EXTERNAL CLOCK TIMING

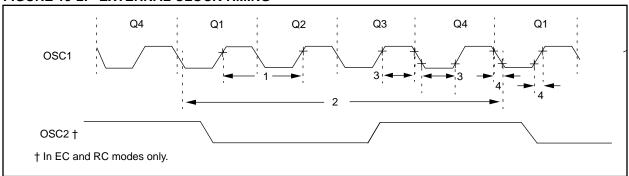


TABLE 19-2: EXTERNAL CLOCK TIMING REQUIREMENTS

Param							
No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
	Fosc	External CLKIN Frequency	DC		8	MHz	EC osc mode - 08 devices (8 MHz devices)
		(Note 1)	DC	_	16	MHz	- 16 devices (16 MHz devices)
		, ,	DC	_	25	MHz	- 25 devices (25 MHz devices)
			DC	_	33	MHz	- 33 devices (33 MHz devices)
		Oscillator Frequency	DC	_	4	MHz	RC osc mode
		(Note 1)	1	_	8	MHz	XT osc mode - 08 devices (8 MHz devices)
			1	_	16	MHz	- 16 devices (16 MHz devices)
			1	_	25	MHz	- 25 devices (25 MHz devices)
			1	_	33	MHz	- 33 devices (33 MHz devices)
			DC	-	2	MHz	LF osc mode
1	Tosc	External CLKIN Period	125		_	ns	EC osc mode - 08 devices (8 MHz devices)
		(Note 1)	62.5	_	_	ns	- 16 devices (16 MHz devices)
			40	_	_	ns	- 25 devices (25 MHz devices)
			30.3	1	_	ns	- 33 devices (33 MHz devices)
		Oscillator Period	250		_	ns	RC osc mode
		(Note 1)	125	_	1,000	ns	XT osc mode - 08 devices (8 MHz devices)
			62.5	_	1,000	ns	- 16 devices (16 MHz devices)
			40	_	1,000	ns	- 25 devices (25 MHz devices)
			30.3	_	1,000	ns	- 33 devices (33 MHz devices)
			500		_	ns	LF osc mode
2	Tcy	Instruction Cycle Time (Note 1)	121.2	4/Fosc	DC	ns	
3	TosL,	Clock in (OSC1)	10 ‡	_	_	ns	EC oscillator
	TosH	high or low time	'				
4	TosR,	Clock in (OSC1)	_	_	5‡	ns	EC oscillator
	TosF	rise or fall time					

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

<sup>‡</sup> These parameters are for design guidance only and are not tested, nor characterized.

Note 1: Instruction cycle period (TCY) equals four times the input oscillator time base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at "min." values with an external clock applied to the OSC1/CLKIN pin. When an external clock input is used, the "max." cycle time limit is "DC" (no clock) for all devices.

FIGURE 19-3: CLKOUT AND I/O TIMING

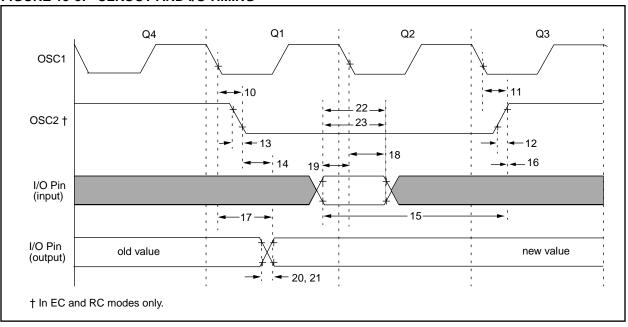


TABLE 19-3: CLKOUT AND I/O TIMING REQUIREMENTS

Parameter No.	Sym	Characteristic		Min	Тур†	Max	Units	Conditions
10	TosH2ckL	OSC1↓ to CLKOUT	·\	_	15 ‡	30 ‡	ns	Note 1
11	TosH2ckH	OSC1↓ to CLKOUT	<b>↑</b>	_	15 ‡	30 ‡	ns	Note 1
12	TckR	CLKOUT rise time		_	5‡	15 ‡	ns	Note 1
13	TckF	CLKOUT fall time		_	5‡	5 ‡ 15 ‡		Note 1
14	TckH2ioV	CLKOUT ↑ to Port out valid	PIC17CR42/42A/43/ R43/44	_	_	0.5Tcy + 20 ‡	ns	Note 1
			PIC17LCR42/42A/43/ R43/44	_	_	0.5Tcy + 50 ‡	ns	Note 1
15	TioV2ckH Port in valid before CLKOUT↑ PIC17CR42/42A/43/ R43/44		0.25Tcy + 25 ‡	_	_	ns	Note 1	
			PIC17LCR42/42A/43/ R43/44	0.25Tcy + 50 ‡	_	_	ns	Note 1
16	TckH2iol	Port in hold after Cl	KOUT <sup>↑</sup>	0 ‡			ns	Note 1
17	TosH2ioV	OSC1↓ (Q1 cycle) t	o Port out valid	_	_	<b>—</b> 100 ‡		
18	TosH2iol	OSC1↓ (Q2 cycle) t (I/O in hold time)	o Port input invalid	0 ‡	_	_	ns	
19	TioV2osH	Port input valid to OSC1↓ (I/O in setup time)		30 ‡	_	_	ns	
20	TioR	Port output rise time		_	10 ‡	35 ‡	ns	
21	TioF	Port output fall time		_	10 ‡	35 ‡	ns	
22	TinHL	INT pin high or low	25 *	_	_	ns		
23	TrbHL	RB7:RB0 change IN	NT high or low time	25 *		_	ns	

<sup>\*</sup> These parameters are characterized but not tested.

<sup>†</sup> 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: Measurements are taken in EC Mode where CLKOUT output is 4 x Tosc.

FIGURE 19-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, AND POWER-UP TIMER TIMING

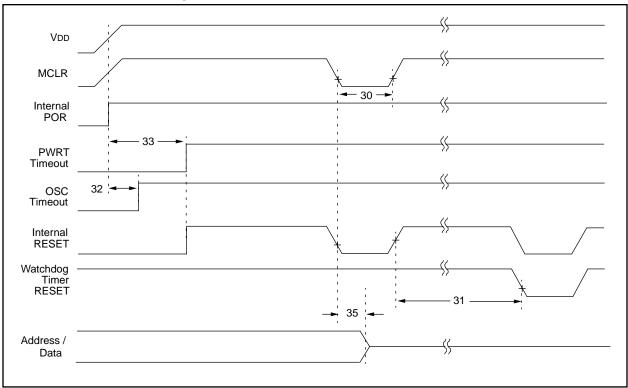


TABLE 19-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER REQUIREMENTS

Parameter No.	Sym	Characteristic		Min	Тур†	Max	Units	Conditions
30	TmcL	MCLR Pulse Width (low)			_	_	ns	VDD = 5V
31	Twdt	Watchdog Timer Time-out Period (Prescale = 1)			12	25 *	ms	VDD = 5V
32	Tost	Oscillation Start-up Time	r Period	_	1024Tosc§	_	ms	Tosc = OSC1 period
33	Tpwrt	Power-up Timer Period		40 *	96	200 *	ms	VDD = 5V
35	TmcL2adI	MCLR to System Inter- face bus (AD15:AD0>) PIC17CR42/42A/ 43/R43/44		_	_	100 *	ns	
		invalid	PIC17LCR42/ 42A/43/R43/44	_	_	120 *	ns	

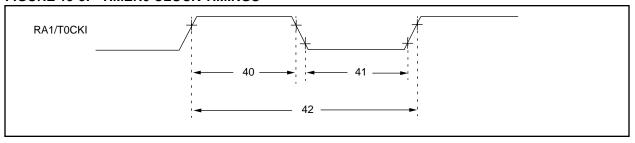
<sup>\*</sup> These parameters are characterized but not tested.

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

<sup>‡</sup> These parameters are for design guidance only and are not tested, nor characterized.

<sup>§</sup> This specification ensured by design.

### FIGURE 19-5: TIMERO CLOCK TIMINGS



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

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

- These parameters are characterized but not tested.
- † Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.
- § This specification ensured by design.

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

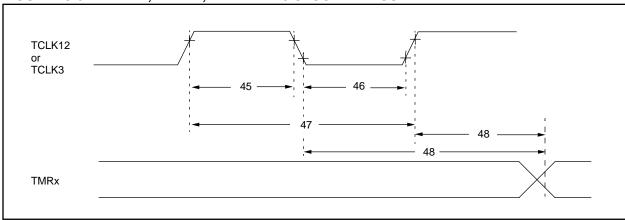
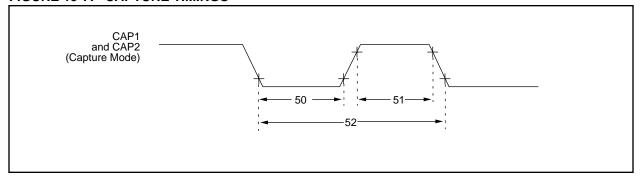


TABLE 19-6: TIMER1, TIMER2, AND TIMER3 CLOCK REQUIREMENTS

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

- \* These parameters are characterized but not tested.
- † Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.
- § This specification ensured by design.

### FIGURE 19-7: CAPTURE TIMINGS

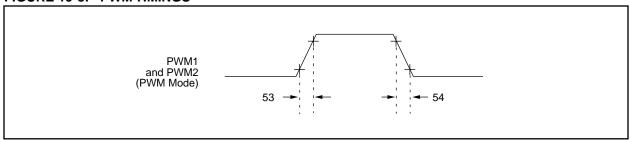


### **TABLE 19-7: CAPTURE REQUIREMENTS**

Parameter No.	Sym	Characteristic	Min	Тур†	Max	Units	Conditions
50	TccL	Capture1 and Capture2 input low time	10 *	_	_	ns	
51	TccH	Capture1 and Capture2 input high time	10 *	_	_	ns	
52	TccP	Capture1 and Capture2 input period	2TCY § N	_	_	ns	N = prescale value (4 or 16)

- These parameters are characterized but not tested.
- † Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.
- § This specification ensured by design.

#### FIGURE 19-8: PWM TIMINGS



### TABLE 19-8: PWM REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Тур†	Max	Units	Conditions
53	TccR	PWM1 and PWM2 output rise time	_	10 *	35 *§	ns	
54	TccF	PWM1 and PWM2 output fall time	_	10 *	35 *§	ns	

These parameters are characterized but not tested.

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

<sup>§</sup> This specification ensured by design.

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

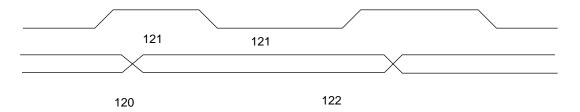


TABLE 19-9: SYNCHRONOUS TRANSMISSION REQUIREMENTS

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

<sup>†</sup> 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	Тур†	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	

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

FIGURE 19-11: MEMORY INTERFACE WRITETIMING (NOT SUPPORTED IN PIC17LC4X DEVICES)

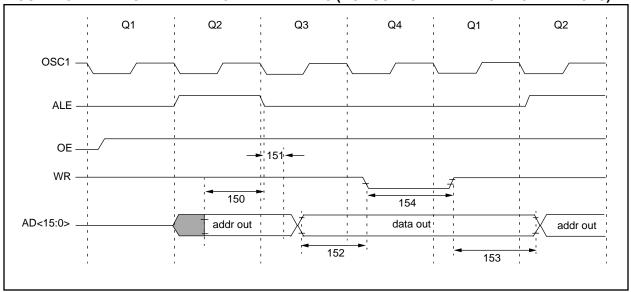


TABLE 19-11: MEMORY INTERFACE WRITE REQUIREMENTS (NOT SUPPORTED IN PIC17LC4X DEVICES)

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
150	TadV2alL	AD<15:0> (address) valid to ALE↓ (address setup time)	0.25Tcy - 10	_	1	ns	
151	TalL2adI	ALE↓ to address out invalid (address hold time)	0	_	_	ns	
152	TadV2wrL	Data out valid to WR↓ (data setup time)	0.25Tcy - 40	_	_	ns	
153	TwrH2adl	WR↑ to data out invalid (data hold time)	_	0.25Tcy §	_	ns	
154	TwrL	WR pulse width	_	0.25Tcy §	_	ns	

These parameters are characterized but not tested.

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

<sup>§</sup> This specification ensured by design.

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

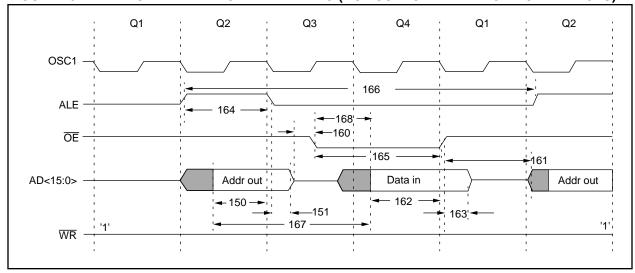


TABLE 19-12: MEMORY INTERFACE READ REQUIREMENTS (NOT SUPPORTED IN PIC17LC4X DEVICES)

Parameter No.	Sym	Characteristic	Min	Тур†	Max	Units	Conditions
150	TadV2alL	AD15:AD0 (address) valid to ALE↓ (address setup time)	0.25Tcy - 10	_	_	ns	
151	TalL2adl	ALE↓ to address out invalid (address hold time)	5*	_	_	ns	
160	TadZ2oeL	AD15:AD0 hi-impedance to <del>OE</del> ↓	0*	_	_	ns	
161	ToeH2adD	OE↑ to AD15:AD0 driven	0.25Tcy - 15	_	_	ns	
162	TadV2oeH	Data in valid before <del>OE</del> ↑ (data setup time)	35	_	_	ns	
163	ToeH2adl	OE↑to data in invalid (data hold time)	0	_	_	ns	
164	TalH	ALE pulse width	_	0.25Tcy §		ns	
165	ToeL	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 (OE low to Data Valid)	_	_	0.5Tcy - 45	ns	

<sup>\*</sup> These parameters are characterized but not tested.

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

<sup>§</sup> This specification ensured by design.

**NOTES:** 

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

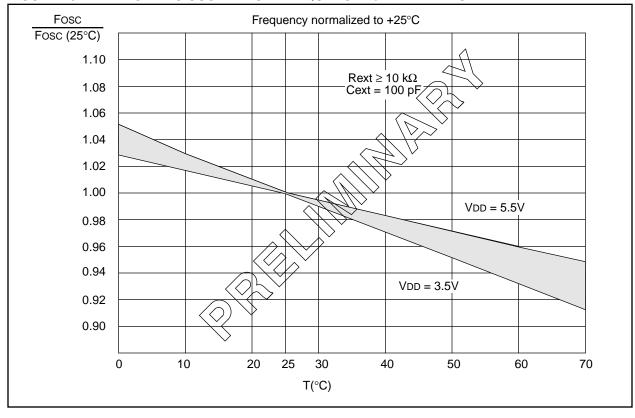
The graphs and tables provided in this section are for design guidance and are not tested nor guaranteed. In some graphs or tables the data presented is outside specified operating range (e.g. outside specified VDD range). This is for information only and devices are ensured to operate properly only 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 (mean +  $3\sigma$ ) and (mean -  $3\sigma$ ) respectively where  $\sigma$  is standard deviation.

TABLE 20-1: PIN CAPACITANCE PER PACKAGE TYPE

Din Name	Typical Capacitance (pF)						
Pin Name	40-pin DIP	44-pin PLCC	44-pin MQFP	44-pin TQFP			
All pins, except MCLR, VDD, and VSS	10	10	10	10			
MCLR pin	20	20	20	20			

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



© 1996 Microchip Technology Inc.

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

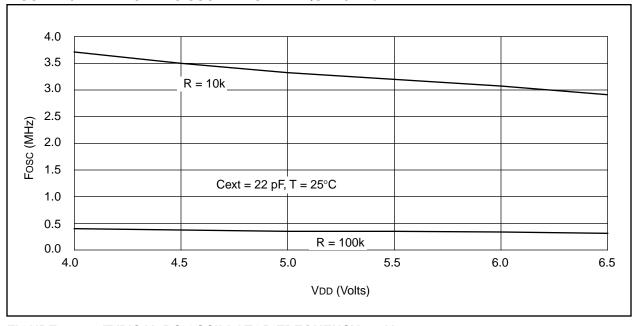


FIGURE 20-3: TYPICAL RC OSCILLATOR FREQUENCY vs. VDD

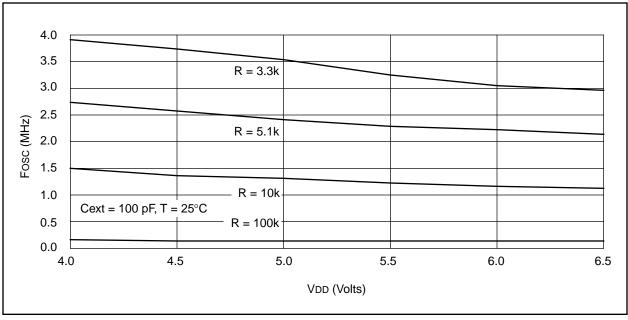


FIGURE 20-4: TYPICAL RC OSCILLATOR FREQUENCY vs. VDD

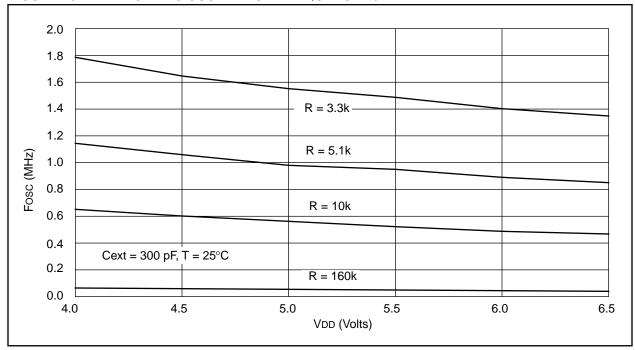


TABLE 20-2: RC OSCILLATOR FREQUENCIES

Cext	Rext		rage 5V, 25°C
22 pF	10k	3.33 MHz	± 12%
	100k	353 kHz	± 13%
100 pF	3.3k	3.54 MHz	± 10%
	5.1k	2.43 MHz	± 14%
	10k	1.30 MHz	± 17%
	100k	129 kHz	± 10%
300 pF	3.3k	1.54 MHz	± 14%
	5.1k	980 kHz	± 12%
	10k	564 kHz	± 16%
	160k	35 kHz	± 18%

FIGURE 20-5: TRANSCONDUCTANCE (gm) OF LF OSCILLATOR vs. VDD

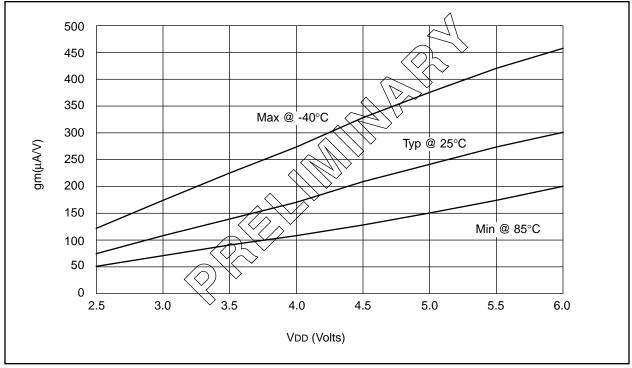


FIGURE 20-6: TRANSCONDUCTANCE (gm) OF XT OSCILLATOR vs. VDD

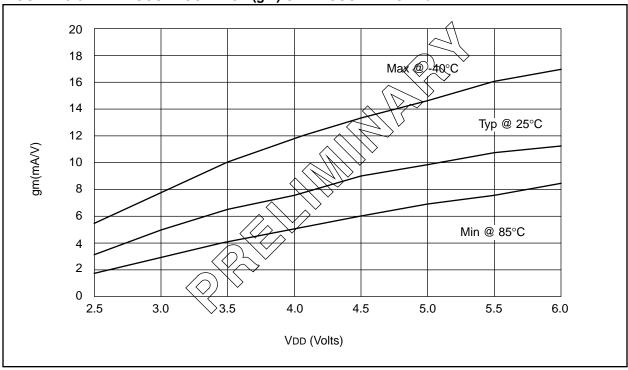


FIGURE 20-7: TYPICAL IDD vs. FREQUENCY (EXTERNAL CLOCK 25°C)

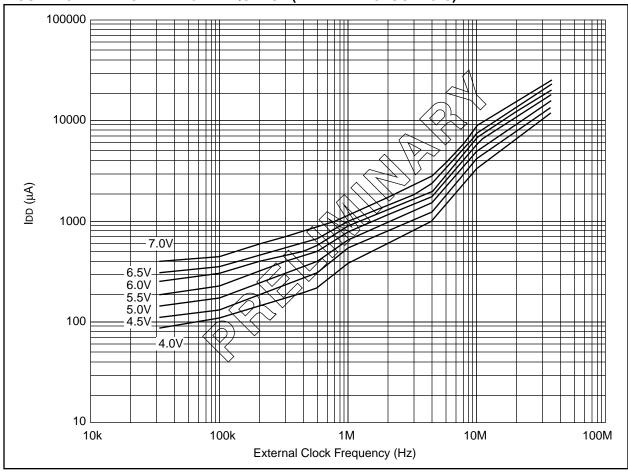


FIGURE 20-8: MAXIMUM IDD vs. FREQUENCY (EXTERNAL CLOCK 125°C TO -40°C)

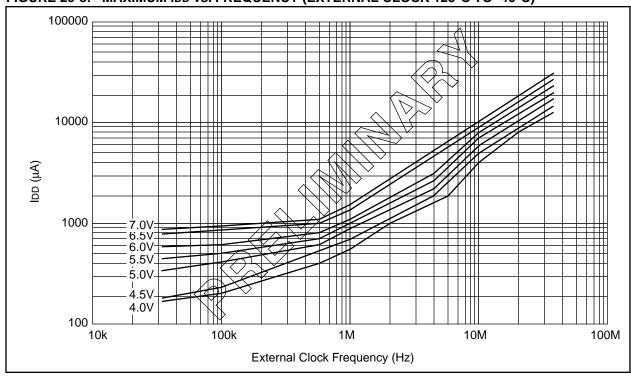


FIGURE 20-9: TYPICAL IPD vs. VDD WATCHDOG DISABLED 25°C

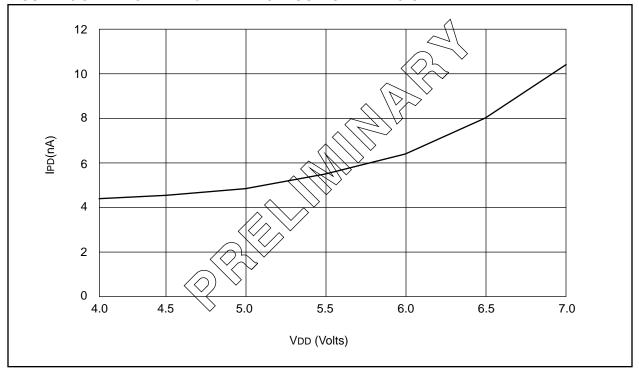


FIGURE 20-10: MAXIMUM IPD vs. VDD WATCHDOG DISABLED

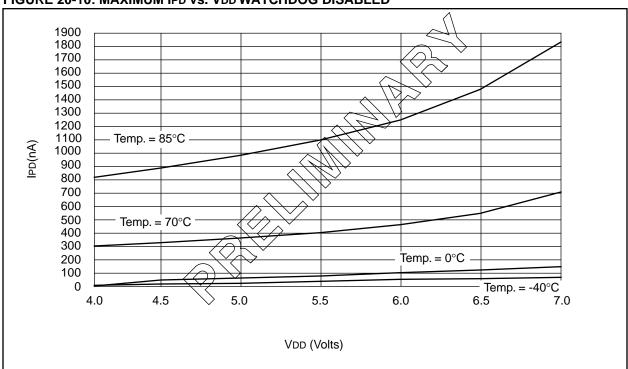


FIGURE 20-11: TYPICAL IPD vs. VDD WATCHDOG ENABLED 25°C

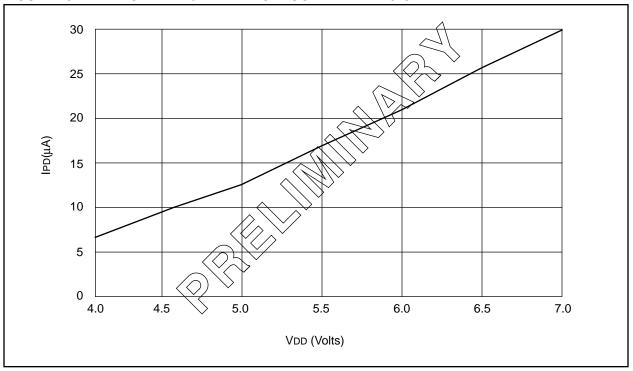
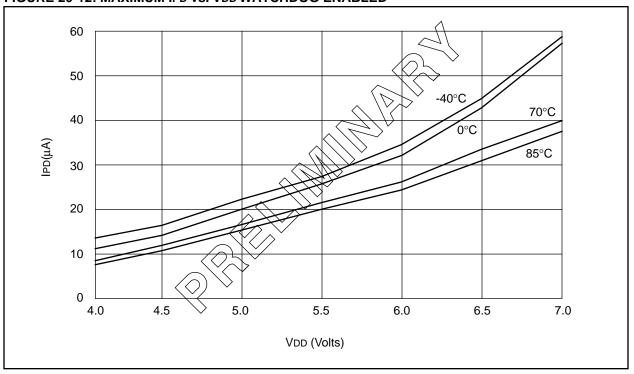


FIGURE 20-12: MAXIMUM IPD vs. VDD WATCHDOG ENABLED



**Applicable Devices** 42 R42 42A 43 R43 44

FIGURE 20-13: WDT TIMER TIME-OUT PERIOD vs. VDD

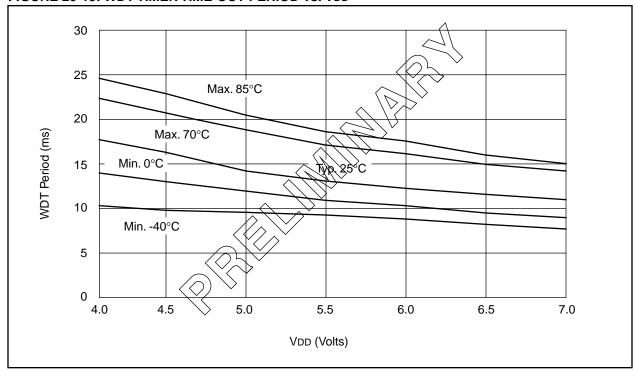


FIGURE 20-14: IOH vs. VOH, VDD = 3V

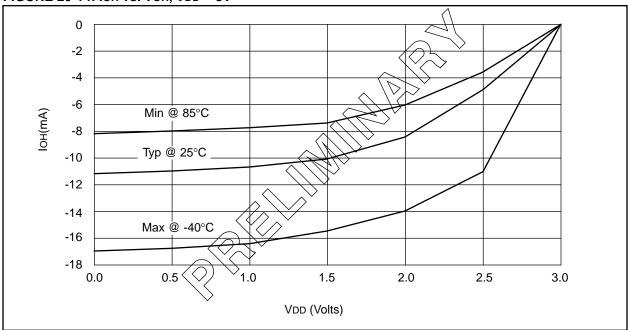
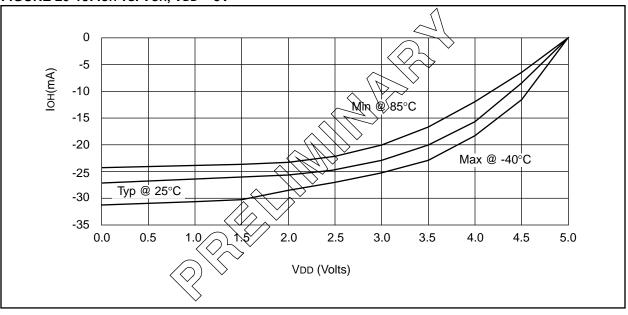
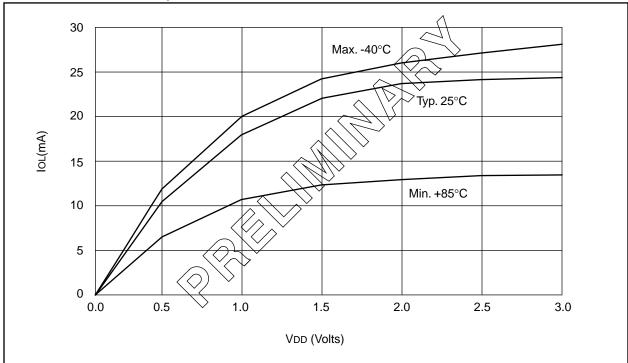


FIGURE 20-15: IOH vs. VOH, VDD = 5V

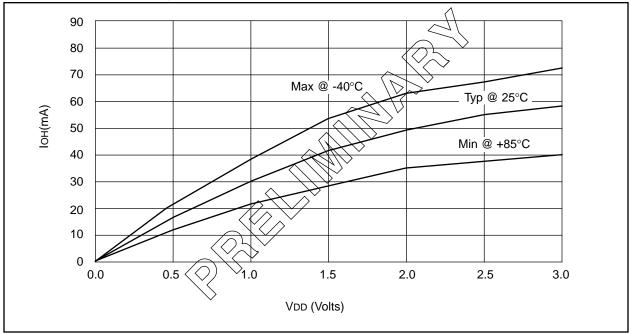






**Applicable Devices** 42 R42 42A 43 R43 44

FIGURE 20-17: IoL vs. Vol, VDD = 5V





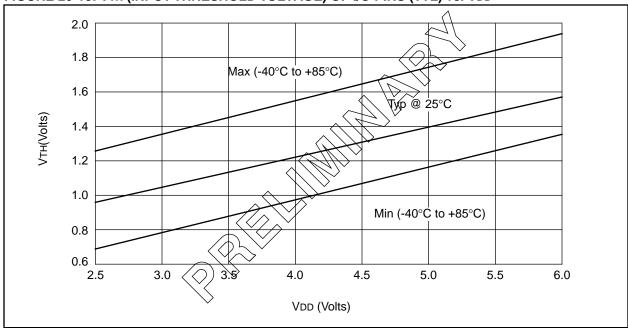


FIGURE 20-19: VTH, VIL of I/O PINS (SCHMITT TRIGGER) VS. VDD

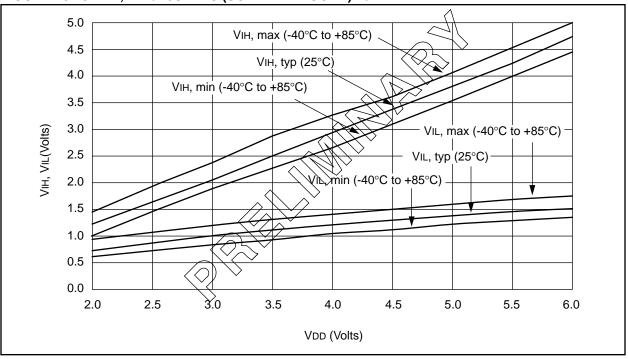
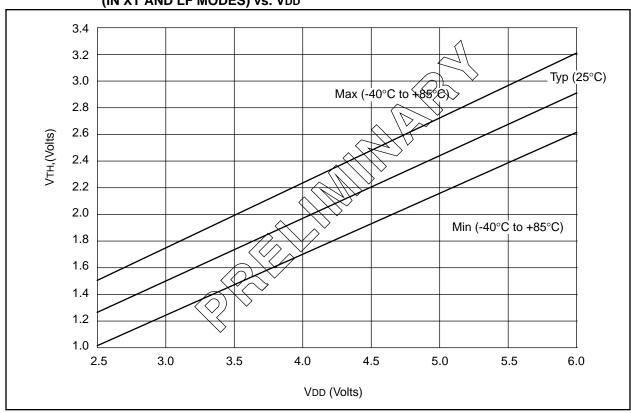


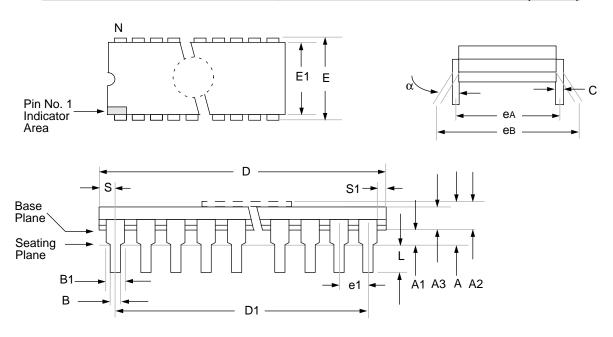
FIGURE 20-20: VTH (INPUT THRESHOLD VOLTAGE) OF OSC1 INPUT (IN XT AND LF MODES) vs. VDD



**NOTES:** 

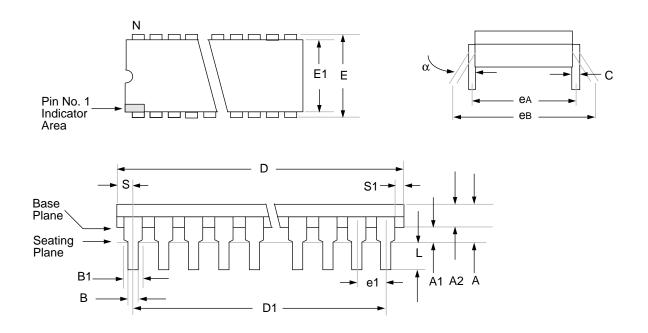
### 21.0 PACKAGING INFORMATION

### 21.1 40-Lead Ceramic CERDIP Dual In-line, and CERDIP Dual In-line with Window (600 mil)

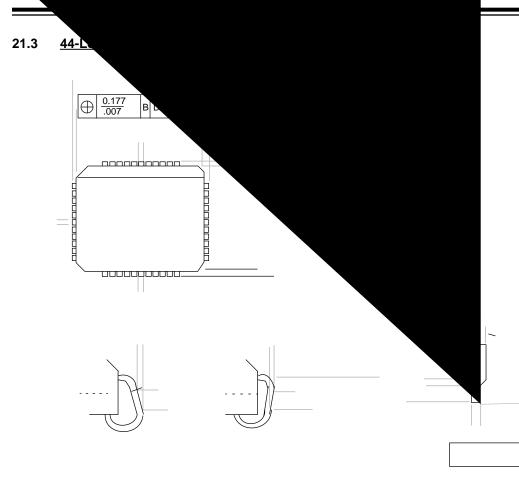


	Package Group: Ceramic CERDIP Dual In-Line (CDP)									
		Millimeters			Inches					
Symbol	Min	Max	Notes	Min	Max	Notes				
α	0°	10°		0°	10°					
Α	4.318	5.715		0.170	0.225					
A1	0.381	1.778		0.015	0.070					
A2	3.810	4.699		0.150	0.185					
A3	3.810	4.445		0.150	0.175					
В	0.355	0.585		0.014	0.023					
B1	1.270	1.651	Typical	0.050	0.065	Typical				
С	0.203	0.381	Typical	0.008	0.015	Typical				
D	51.435	52.705		2.025	2.075					
D1	48.260	48.260	Reference	1.900	1.900	Reference				
Е	15.240	15.875		0.600	0.625					
E1	12.954	15.240		0.510	0.600					
e1	2.540	2.540	Reference	0.100	0.100	Reference				
eA	14.986	16.002	Typical	0.590	0.630	Typical				
eB	15.240	18.034		0.600	0.710					
L	3.175	3.810		0.125	0.150					
N	40	40		40	40					
S	1.016	2.286		0.040	0.090					
S1	0.381	1.778		0.015	0.070					

### 21.2 40-Lead Plastic Dual In-line (600 mil)

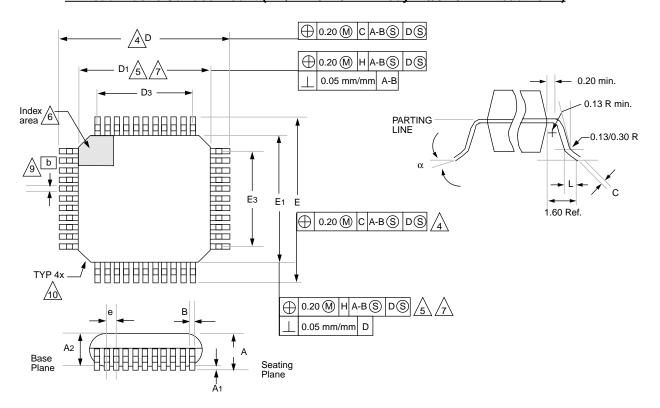


	Package Group: Plastic Dual In-Line (PLA)									
		Millimeters		Inches						
Symbol	Min	Max	Notes	Min	Max	Notes				
α	0°	10°		0°	10°					
Α	_	5.080		_	0.200					
A1	0.381	_		0.015	_					
A2	3.175	4.064		0.125	0.160					
В	0.355	0.559		0.014	0.022					
B1	1.270	1.778	Typical	0.050	0.070	Typical				
С	0.203	0.381	Typical	0.008	0.015	Typical				
D	51.181	52.197		2.015	2.055					
D1	48.260	48.260	Reference	1.900	1.900	Reference				
Е	15.240	15.875		0.600	0.625					
E1	13.462	13.970		0.530	0.550					
e1	2.489	2.591	Typical	0.098	0.102	Typical				
eA	15.240	15.240	Reference	0.600	0.600	Reference				
eB	15.240	17.272		0.600	0.680					
L	2.921	3.683		0.115	0.145					
N	40	40		40	40					
S	1.270	_		0.050	_					
S1	0.508	_		0.020	_					



	Package Group: Plastic Leaded Chip Carrier (PLCC)									
		Millimeters			Inches					
Symbol	Min	Max	Notes	Min	Max	Notes				
Α	4.191	4.572		0.165	0.180					
A1	2.413	2.921		0.095	0.115					
D	17.399	17.653		0.685	0.695					
D1	16.510	16.663		0.650	0.656					
D2	15.494	16.002		0.610	0.630					
D3	12.700	12.700	Reference	0.500	0.500	Reference				
Е	17.399	17.653		0.685	0.695					
E1	16.510	16.663		0.650	0.656					
E2	15.494	16.002		0.610	0.630					
E3	12.700	12.700	Reference	0.500	0.500	Reference				
N	44	44		44	44					
CP	_	0.102		_	0.004					
LT	0.203	0.381		0.008	0.015					

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



		Packag	ge Group: Plastic	c MQFP		
		Millimeters		Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
α	0°	7°		0°	7°	
Α	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
С	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
е	0.800	0.800		0.031	0.032	
L	0.730	1.030		0.028	0.041	
N	44	44		44	44	
СР	0.102	_		0.004	_	

#### 1.0ø (0.039ø) Ref. 8888888888 11°/13°(4x) Pin#1 Pin#1 $\blacksquare$ 2 == 2 == 0° Min < H <del>-</del>EI-I E1 Е ш 11°/13°(4x) ΙПП **Detail B** -3.0ø (0.118ø) Ref. R 1 0.08 Min Option 1 (TOP side) R 0.08/0.20 Option 2 (TOP side) Gage Plane Lead Finish Base Metal 0.20 Min С - c1 **Detail A Detail B** 1.00 Ref 1.00 Ref. b1 **Detail B Detail A**

### 21.5 44-Lead Plastic Surface Mount (TQFP 10x10 mm Body 1.0/0.10 mm Lead Form)

	Package Group: Plastic TQFP									
		Millimeters			Inches					
Symbol	Min	Max	Notes	Min	Max	Notes				
Α	1.00	1.20		0.039	0.047					
A1	0.05	0.15		0.002	0.006					
A2	0.95	1.05		0.037	0.041					
D	11.75	12.25		0.463	0.482					
D1	9.90	10.10		0.390	0.398					
Е	11.75	12.25		0.463	0.482					
E1	9.90	10.10		0.390	0.398					
L	0.45	0.75		0.018	0.030					
е	0.80	BSC		0.031	BSC					
b	0.30	0.45		0.012	0.018					
b1	0.30	0.40		0.012	0.016					
С	0.09	0.20		0.004	0.008					
c1	0.09	0.16		0.004	0.006					
N	44	44		44	44					
Θ	0°	7°		0°	<b>7</b> °					

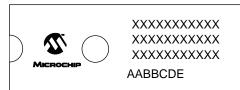
- Note 1: Dimensions D1 and E1 do not include mold protrusion. Allowable mold protrusion is 0.25m/m (0.010") per side. D1 and E1 dimensions including mold mismatch.
  - 2: Dimension "b" does not include Dambar protrusion, allowable Dambar protrusion shall be 0.08m/m (0.003")max.
  - 3: This outline conforms to JEDEC MS-026.

### 21.6 Package Marking Information

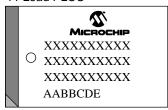
40-Lead PDIP/CERDIP



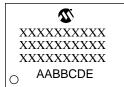
40 Lead CERDIP Windowed



44-Lead PLCC



44-Lead MQFP



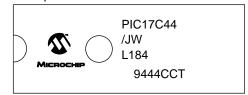
44-Lead TQFP



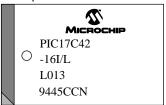
Example



Example



Example



Example



Example



Microchip part number information Legend: MM...M XX...X Customer specific information\* AA Year code (last 2 digits of calendar year) BB Week code (week of January 1 is week '01') С 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 Ε Assembly code of the plant or country of origin in which part was assembled

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.

#### **APPENDIX A: MODIFICATIONS**

The following is the list of modifications over the PIC16CXX microcontroller family:

- Instruction word length is increased to 16-bit.
   This allows larger page sizes both in program memory (8 Kwords verses 2 Kwords) and register file (256 bytes versus 128 bytes).
- Four modes of operation: microcontroller, protected microcontroller, extended microcontroller, and microprocessor.
- 22 new instructions.
   The MOVF, TRIS and OPTION instructions have been removed.
- 4 new instructions for transferring data between data memory and program memory. This can be used to "self program" the EPROM program memory.
- Single cycle data memory to data memory transfers possible (MOVPF and MOVFP instructions).
   These instructions do not affect the Working register (WREG).
- 6. W register (WREG) is now directly addressable.
- A PC high latch register (PCLATH) is extended to 8-bits. The PCLATCH register is now both readable and writable.
- 8. Data memory paging is redefined slightly.
- DDR registers replaces function of TRIS registers.
- Multiple Interrupt vectors added. This can decrease the latency for servicing the interrupt.
- 11. Stack size is increased to 16 deep.
- 12. BSR register for data memory paging.
- Wake up from SLEEP operates slightly differently.
- 14. The Oscillator Start-Up Timer (OST) and Power-Up Timer (PWRT) operate in parallel and not in series.
- PORTB interrupt on change feature works on all eight port pins.
- 16. TMR0 is 16-bit plus 8-bit prescaler.
- Second indirect addressing register added (FSR1 and FSR2). Configuration bits can select the FSR registers to auto-increment, auto-decrement, remain unchanged after an indirect address.
- 18. Hardware multiplier added (8 x 8  $\rightarrow$  16-bit) (PIC17C43 and PIC17C44 only).
- 19. Peripheral modules operate slightly differently.
- 20. Oscillator modes slightly redefined.
- Control/Status bits and registers have been placed in different registers and the control bit for globally enabling interrupts has inverse polarity.
- 22. Addition of a test mode pin.
- In-circuit serial programming is not implemented.

#### APPENDIX B: COMPATIBILITY

To convert code written for PIC16CXX to PIC17CXX, the user should take the following steps:

- Remove any TRIS and OPTION instructions, and implement the equivalent code.
- Separate the interrupt service routine into its four vectors.
- 3. Replace:

```
MOVF REG1, W with:
MOVFP REG1, WREG
```

4. Replace:

```
MOVF REG1, W
MOVWF REG2
with:
MOVPF REG1, REG2; Addr(REG1)<20h
or
MOVFP REG1, REG2; Addr(REG2)<20h
```

Note:		If REG1 and REG2 are both at addresses							
	greater t	greater then 20h, two instructions are							
	required.								
	MOVFP	REG1,	WREG	;					
	MOVPF	WREG,	REG2	;					

- 5. Ensure that all bit names and register names are updated to new data memory map location.
- 6. Verify data memory banking.
- 7. Verify mode of operation for indirect addressing.
- 8. Verify peripheral routines for compatibility.
- Weak pull-ups are enabled on reset.

To convert code from the PIC17C42 to all the other PIC17C4X devices, the user should take the following steps.

- If the hardware multiply is to be used, ensure that any variables at address 18h and 19h are moved to another address.
- Ensure that the upper nibble of the BSR was not written with a non-zero value. This may cause unexpected operation since the RAM bank is no longer 0.
- The disabling of global interrupts has been enhanced so there is no additional testing of the GLINTD bit after a BSF CPUSTA, GLINTD instruction.

### **APPENDIX C: WHAT'S NEW**

The structure of the document has been made consistent with other data sheets. This ensures that important topics are covered across all PIC16/17 families. Here is an overview of new features.

Added the following devices:

PIC17CR42

PIC17C42A

PIC17CR43

A 33 MHz option is now available.

#### **APPENDIX D: WHAT'S CHANGED**

To make software more portable across the different PIC16/17 families, the name of several registers and control bits have been changed. This allows control bits that have the same function, to have the same name (regardless of processor family). Care must still be taken, since they may not be at the same special function register address. The following shows the register and bit names that have been changed:

Old Name	New Name
TX8/9	TX9
RC8/9	RX9
RCD8	RX9D
TXD8	TX9D

Instruction DECFSNZ corrected to DCFSNZ

Instruction INCFSNZ corrected to INFSNZ

Enhanced discussion on PWM to include equation for determining bits of PWM resolution.

Section 13.2.2 and 13.3.2 have had the description of updating the FERR and RX9 bits enhanced.

The location of configuration bit PM2 was changed (Figure 6-1 and Figure 14-1).

Enhanced description of the operation of the INTSTA register.

Added note to discussion of interrupt operation.

Tightened electrical spec D110.

Corrected steps for setting up USART Asynchronous Reception.

### **APPENDIX E: PIC16/17 MICROCONTROLLERS**

### E.1 PIC14000 Devices

Features Argelian Argument Arg	28-pin DIP, SOIC, SSOP (.300 mil)
erals (State of State	Interpretation of the second o
1 Siles is	Yes
oherals of the same of the sam	2.7-6.0
To State State	22
Periphe Periphe Solid So	11
	41
Solvon Art Solvon	I <sup>2</sup> C/ SMBus
Clock Memory of Oroginal World	TMR0 ADTMR 8
To Tale Tale of	192
To Linking to the state of the	7 <del>4</del>
Tage 1	20
	PIC14000

#### **PIC16C5X Family of Devices E.2**

				O	Clock	Memory	Perip	Peripherals	Features
	*SON	TOSI JUNION	10 10 Taller 10 10 10 10 10 10 10 10 10 10 10 10 10	(3-1/W) 10-16-16-01 AN EVEC WASA WORLD WOR	(Sellow resed was	(9)	\$440A	(SHON) SCILES SCENON	SHON SE SEGENTS
PIC16C52	4	384	ı	25	TMR0	12	2.5-6.25	33	18-pin DIP, SOIC
PIC16C54	20	512	ı	25	TMR0	12	2.5-6.25	33	18-pin DIP, SOIC; 20-pin SSOP
PIC16C54A	20	512	ı	25	TMR0	12	2.0-6.25	33	18-pin DIP, SOIC; 20-pin SSOP
PIC16CR54A	20	I	512	25	TMR0	12	2.0-6.25	33	18-pin DIP, SOIC; 20-pin SSOP
PIC16C55	20	512	ı	24	TMR0	20	2.5-6.25	33	28-pin DIP, SOIC, SSOP
PIC16C56	20	7,	I	25	TMR0	12	2.5-6.25	33	18-pin DIP, SOIC; 20-pin SSOP
PIC16C57	20	2K	ı	72	TMR0	20	2.5-6.25	33	28-pin DIP, SOIC, SSOP
PIC16CR57B	20	I	2K	72	TMR0	20	2.5-6.25	33	28-pin DIP, SOIC, SSOP
PIC16C58A	20	2K	I	73	TMR0	12	2.0-6.25	33	18-pin DIP, SOIC; 20-pin SSOP
PIC16CR58A	20	1	2K	73	TMR0	12	2.5-6.25	33	18-pin DIP, SOIC; 20-pin SSOP
All PIC16/17		devices	have F	Power-On	ו Reset, selectab	le Watch	ndog Timer, s	selectab	Family devices have Power-On Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability.

#### **PIC16CXXX Family of Devices** E.3

Memory Peripherals Features	88 E1107 82	Sold Miles	Confidence of the County of th	Contigue and the tries of the country of the tries of the country of the tries of the country of the country of the tries of the country of the count	Comparatories (200100 Control of the tribute (200100 Control o	Onto a series of the source of the series of the source of the series of	Contidado (Septembrie) (Septemb
	SOLOS SUR SUR BULLON	3 13	3 13	3 13	Yes 4 13	7	Yes 4 13
(3)	T. C. SOL W. FOLL	TMR0	TMR0 -	TMR0	TMR0	TMR0	
	OHON LEGION GEO WOODS HUNDER	512 80	1K 80	2K 128	512 80	1K 80	_
	TO STORY	20	20	20	20	20	
		PIC16C554	PIC16C556	PIC16C558	PIC16C620	PIC16C621	

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

All PIC16C6XXX Family devices use serial programming with clock pin RB6 and data pin RB7.

### E.4 PIC16C6X Family of Devices

					Clock	Memory	)IIV		"	Peripherals	erals			Features
		\	Tolog	John Bold	TO LORD ON SIND LORD OF START	1 / 1	September States September	*Add offer	TO THE STATE OF TH	100 TO TO TO	\ \\ 3\		SHON	10010
	197	THE WAR	A TOUR	10	N POLITY	Mon V	GO SINGE	SHO K	Tolele	To ton Land		They ou	is inoto the selfor	Selestor A Thousand
PIC16C62	20	2K	1	128	TMR2	_	SPI/I²C	I	2	22	3.0-6.0	Yes		28-pin SDIP, SOIC, SSOP
PIC16C62A <sup>(1)</sup>	20	2K		128	TMR0, TMR1, TMR2	-	SPI/I²C	1	2	22	2.5-6.0	Yes	Yes	28-pin SDIP, SOIC, SSOP
PIC16CR62 <sup>(1)</sup>	20	I	2K	128	TMR0, TMR1, TMR2	1	SPI/I²C	I	2	22	2.5-6.0	Yes	Yes	28-pin SDIP, SOIC, SSOP
PIC16C63	20	4K	1	192	TMR0, TMR1, TMR2	2	SPI/I²C, USART	I	10	22	2.5-6.0	Yes	Yes	28-pin SDIP, SOIC
PIC16CR63 <sup>(1)</sup>	20	I	4K	192	TMR0, TMR1, TMR2	7	SPI/I²C, USART	ı	10	22	2.5-6.0	Yes	Yes	28-pin SDIP, SOIC
PIC16C64	20	2K	-	128	TMR0, TMR1, TMR2	1	SPI/I²C	Yes	8	33	3.0-6.0	Yes		40-pin DIP; 44-pin PLCC, MQFP
PIC16C64A <sup>(1)</sup>	20	2K	1	128	TMR0, TMR1, TMR2	~	SPI/I2C	Yes	8	33	2.5-6.0	Yes	Yes	40-pin DIP; 44-pin PLCC, MQFP, TQFP
PIC16CR64 <sup>(1)</sup>	20		2K	128	TMR0, TMR1, TMR2	-	SPI/I²C	Yes	8	33	2.5-6.0	Yes	Yes	40-pin DIP; 44-pin PLCC, MQFP, TQFP
PIC16C65	20	4 <sub>K</sub>	1	192	TMR0, TMR1, TMR2	2	SPI/I²C, USART	Yes	11	33	3.0-6.0	Yes		40-pin DIP; 44-pin PLCC, MQFP
PIC16C65A <sup>(1)</sup>	20	<del>4</del>	1	192	TMR0, TMR1, TMR2	2	SPI/I²C, USART	Yes	11	33	2.5-6.0	Yes	Yes	40-pin DIP; 44-pin PLCC, MQFP, TQFP
PIC16CR65 <sup>(1)</sup>	20	I	4K	192	TMR0, TMR1, TMR2	2	SPI/I²C, USART	Yes	11	33	2.5-6.0	Yes	Yes	40-pin DIP; 44-pin PLCC, MQFP, TQFP
0.010 = 4	١,	1	1-1-1	7 1 1 1 1			i		i					

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

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

Please contact your local sales office for availability of these devices. <del>...</del> Note

### **PIC16C7X Family of Devices** E.5

				Clock		Memory			Peri	Peripherals	<u>s</u>	T		Features
				Stand of the stand	(B)			State S	EX OF		Seureno		(a)	Committee
	No.	L'ELALLA .	Tolland Wood !!	Senton suni suningen	**************************************	State NOO ON MAN STATE OF STAT	STOP TO	18 8 8 V	HOD STUDY	So to tople	Con Solet Serios of the Mos of the Color of	Soft Services	ON SHOULD HE	Soleto Hoole of Soleto State of Soleto Solet
PIC16C710	20	512	36	TMR0				4	4	13	3.0-6.0	Yes	Yes	18-pin DIP, SOIC; 20-pin SSOP
PIC16C71	50	누	36	TMR0	Ι	I	ı	4	4	13	3.0-6.0	Yes	I	18-pin DIP, SOIC
PIC16C711	70	<del></del>	89	TMR0		I	1	4	4	13	3.0-6.0	Yes	Yes	18-pin DIP, SOIC; 20-pin SSOP
PIC16C72	20	¥	128	TMR0, TMR1, TMR2	_	SPI/I <sup>2</sup> C	ı	2	ω	22	2.5-6.0	Yes	Yes	28-pin SDIP, SOIC, SSOP
PIC16C73	20	<del>4</del>	192	TMR0, TMR1, TMR2	7	SPI/I²C, USART	I	2	7	22	3.0-6.0	Yes	1	28-pin SDIP, SOIC
PIC16C73A <sup>(1)</sup>	20	<del>4</del>	192	TMR0, TMR1, TMR2	7	SPI/I²C, USART	I	2	7	22	2.5-6.0	Yes	Yes	28-pin SDIP, SOIC
PIC16C74	20	<del>4</del>	192	TMR0, TMR1, TMR2	7	SPI/I <sup>2</sup> C, USART	Yes	8	12	33	3.0-6.0	Yes	I	40-pin DIP; 44-pin PLCC, MQFP
PIC16C74A <sup>(1)</sup>	20	<del>4</del>	192	TMR0, TMR1, TMR2	7	SPI/I <sup>2</sup> C, USART	Yes	8	12	33	2.5-6.0	Yes	Yes	Yes 40-pin DIP; 44-pin PLCC, MQFP, TQFP
All P	C16/1	7 Fami	ilv devi	ices have Power-c	Ę	Reset sel	actable	∘ Watc	ի ժող	rimer	selectable	Coder	rotect	All PIC16/17 Family devices have Dower-on Reset selectable Watchdor Timer selectable code protect and bigh I/O current

All PIC16/17 Family devices have Power-on Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability.
All PIC16C7X Family devices use serial programming with clock pin RB6 and data pin RB7.
Please contact your local sales office for availability of these devices.

÷ Note

#### **E.6 PIC16C8X Family of Devices**

					င္ပ	Clock	Me	Memory		Peripherals	erals Features
				13	CALINO LOGI	\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \	Tou				
			1	ELBOO TO	\ <b>\</b>	VUEL	Sax Whele	(6)			(SIL)
		1	TOUGH		<b>%</b>		SONO TO	(S)8/1		SOJINE	N of I
	•	YUN		100	\	WOW.	BOW TO THOM THOM	00.	10/2	Suic day	SOCO, LA SOCO
	No.	III.	(SP)	Sil.	**************************************	10 S	Pull Se		(8) (8)	367 YO	TORO .
PIC16C84	10	I	<del>1</del>	-	36	64	TMR0	4	13	2.0-6.0	18-pin DIP, SOIC
PIC16F84 <sup>(1)</sup>	10	눚	ı	I	68	64	TMR0	4	13	2.0-6.0	18-pin DIP, SOIC
PIC16CR84 <sup>(1)</sup>	10	ı	1	누 논	68	64	TMR0	4	13	2.0-6.0	18-pin DIP, SOIC
PIC16F83 <sup>(1)</sup>	10	512	I	I	36	64	TMR0	4	13	2.0-6.0	18-pin DIP, SOIC
PIC16CR83 <sup>(1)</sup>	10		1	512	36	64	TMR0	4	13	2.0-6.0	2.0-6.0   18-pin DIP, SOIC
										l l	

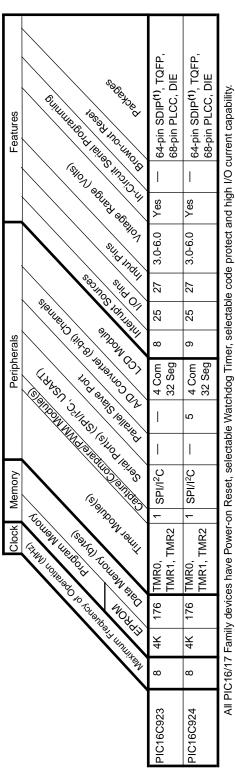
All PIC16/17 family devices have Power-on Reset, selectable Watchdog Timer, selectable code protect, and high I/O current capability.

All PIC16C8X family devices use serial programming with clock pin RB6 and data pin RB7.

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

Note 1:

# E.7 PIC16C9XX Family Of Devices



All PIC16CXX Family devices use serial programming with clock pin RB6 and data pin RB7. Please contact your local Microchip representative for availability of this package. Note

# E.8 PIC17CXX Family of Devices

					Clock	Memory	lory		Pe	Peripherals	sls				Features	
	The state of the s	iy unusa	Tolonbe, Woday	Tollie and tolling the state of	SOUDON GOLON LOUIS NOON TON TON TON TON TON TON TON TON TO			(14 kS D) (S RO DE HOLE) (S SE LINGE) (S SE LINGE)	Josho d'in	Adhin's stendient	Adillo legge	Statife in the field of the state of the sta	Secures sources of status of secures sources s	Solled Soll	Stollouds of teaching of the stolloud of the s	
PIC17C42	25	2K	I	232	TMR0,TMR1, TMR2,TMR3	2	2	Yes	I	Yes	11	33	4.5-5.5	22	40-pin DIP; 44-pin PLCC, MQFP	
PIC17C42A	25	X	I	232	TMR0,TMR1, TMR2,TMR3	7	7	Yes	Yes	Yes	7	33	2.5-6.0	58	40-pin DIP; 44-pin PLCC, TQFP, MQFP	
PIC17CR42	25	I	X	232	TMR0,TMR1, TMR2,TMR3	7	7	Yes	Yes	Yes	7	33	2.5-6.0	58	40-pin DIP; 44-pin PLCC, TQFP, MQFP	
PIC17C43	25	¥	I	454	TMR0,TMR1, TMR2,TMR3	2	2	Yes	Yes	Yes	11	33	2.5-6.0	28	40-pin DIP; 44-pin PLCC, TQFP, MQFP	
PIC17CR43	25	I	<del>\$</del>	454	TMR0,TMR1, TMR2,TMR3	7	7	Yes	Yes	Yes	7	33	2.5-6.0	58	40-pin DIP; 44-pin PLCC, TQFP, MQFP	
PIC17C44	25	8		454	TMR0,TMR1, TMR2,TMR3	2	2	Yes	Yes	Yes	11	33	2.5-6.0	58	40-pin DIP; 44-pin PLCC, TQFP, MQFP	
All F	IC16/1	7 Fan	nily de	vices ha	ave Power-on Re	eset	sel,	ectable <sup>1</sup>	Watcho	log Tin	ier, se	lectab	le code pr	otect a	All PIC16/17 Family devices have Power-on Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability.	١.

DS30412C-page 220

# PIN COMPATIBILITY

Devices that have the same package type and VDD, Vss and MCLR pin locations are said to be pin compatible. This allows these different devices to operate in the same socket. Compatible devices may only requires minor software modification to allow proper operation in the application socket (ex., PIC16C56 and PIC16C61 devices). Not all devices in the same package size are pin compatible; for example, the PIC16C62 is compatible with the PIC16C63, but not the PIC16C55.

Pin compatibility does not mean that the devices offer the same features. As an example, the PIC16C54 is pin compatible with the PIC16C71, but does not have an A/D converter, weak pull-ups on PORTB, or interrupts.

TABLE E-1: PIN COMPATIBLE DEVICES

Pin Compatible Devices	Package
PIC12C508, PIC12C509	8-pin
PIC16C54, PIC16C54A, PIC16CR54A, PIC16C56, PIC16C58A, PIC16CR58A, PIC16C61, PIC16C554, PIC16C556, PIC16C558 PIC16C620, PIC16C621, PIC16C622, PIC16C710, PIC16C71, PIC16C711, PIC16F83, PIC16CR83, PIC16C84, PIC16F84A, PIC16CR84	18-pin 20-pin
PIC16C55, PIC16C57, PIC16CR57B	28-pin
PIC16C62, PIC16CR62, PIC16C62A, PIC16C63, PIC16C72, PIC16C73, PIC16C73A	28-pin
PIC16C64, PIC16CR64, PIC16C64A, PIC16C65, PIC16C65A, PIC16C74, PIC16C74A	40-pin
PIC17C42, PIC17CR42, PIC17C42A, PIC17C43, PIC17CR43, PIC17C44	40-pin
PIC16C923, PIC16C924	64/68-pin

**NOTES:** 

# 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.

 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

- 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 TO bit for changing state from set to clear. Example 1 shows putting the PIC17C42 to sleep.

# **EXAMPLE F-1: PIC17C42 TO 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

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

# **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.

NOTES:

#### **INDEX** CA1IE ......23 CA1IF ......24 CA1OVF ......72 Α CA2ED0 ......71 CA2ED1 ......71 ADDLW ......112 CA2H ......20, 35 ADDWF ......112 CA2IE ......23, 78 ADDWFC ......113 CA2IF ......24, 78 ALU ......9 ALU STATUS Register (ALUSTA) ......36 CA2OVF ......72 ALUSTA ......34, 36, 108 Calculating Baud Rate Error .....86 ALUSTA Register .......36 CALL ......39, 117 ANDLW ......113 Capacitor Selection ANDWF ......114 Ceramic Resonators ......101 Application Notes Crystal Oscillator ......101 AN552 ......55 Capture ......71, 78 Assembler .......144 Capture Sequence to Read Example ......78 Asynchronous Master Transmission ......90 Capture1 Asynchronous Transmitter ......89 Mode ......71 Overflow ......72 Capture2 В Mode 71 Overflow .......72 Bank Select Register (BSR) ......42 Carry (C) ......9 Banking .......42 Ceramic Resonators ......100 Baud Rate Formula ......86 Circular Buffer ......39 Baud Rate Generator (BRG) ......86 Clearing the Prescaler ......103 **Baud Rates** Clock/Instruction Cycle (Figure) ......14 Asynchronous Mode ......88 Clocking Scheme/Instruction Cycle (Section) ......14 Synchronous Mode ......87 CLRF ......117 BCF ......114 CLRWDT ......118 Bit Manipulation ......108 Code Protection .......99, 106 **Block Diagrams** COMF ......118 On-chip Reset Circuit ......15 Configuration PIC17C42 ......10 Bits ......100 PORTD ......60 Locations ......100 Oscillator ......100 Word ......99 RA0 and RA1 .....53 CPFSEQ ......119 RA2 and RA3 ......54 CPFSGT ......119 RA4 and RA5 ......54 CPFSLT ......120 RB3:RB2 Port Pins ......56 CPU STATUS Register (CPUSTA) ......37 RB7:RB4 and RB1:RB0 Port Pins ......55 CPUSTA ......34, 37, 105 RC7:RC0 Port Pins ......58 CREN ......84 Timer3 with One Capture and One Period Register .. 78 Crystal Operation, Overtone Crystals ......101 TMR1 and TMR2 in 16-bit Timer/Counter Mode ....... 74 Crystal or Ceramic Resonator Operation ......100 TMR1 and TMR2 in Two 8-bit Timer/Counter Mode .. 73 TMR3 with Two Capture Registers ......79 CSRC ......83 WDT ......104 BORROW .......9 BRG ......86 D BSF ......115 Data Memory BSR ......34, 42 GPR ......29, 32 BSR Operation ......42 Indirect Addressing ......39 BTFSC ......115 Organization ......32 BTFSS ......116 SFR ......29, 32 BTG .......116 Transfer to Program Memory ......43 DAW ......120 DC ......9, 36 C DDRB ......19, 34, 55 DDRC ......19, 34, 58 DDRD ......19, 34, 60 C Compiler (MP-C) .......145 DDRE ......19, 34, 62 DECF ......121 CA1ED0 ......71 DECFSNZ ......122 CA1ED1 ......71 DECFSZ ......121

Delay From External Clock Edge68	FOSC199
Development Support143	FS0
Development Tools143	FS1
Device Drawings	FS2
44-Lead Plastic Surface Mount (MQFP	FS3
10x10 mm Body 1.6/0.15 mm Lead Form)209	FSR0
DIGIT BORROW9	FSR1
Digit Carry (DC)9	Fuzzy Logic Dev. System (fuzzyTECH®-MP) 143, 145
Duty Cycle75	
	G
E	
Floatrical Characteristics	General Format for Instructions
Electrical Characteristics	General Purpose RAM
PIC17C42	General Purpose RAM Bank
Absolute Maximum Ratings147	General Purpose Register (GPR)
Capture Timing	GLINTD
CLKOUT and I/O Timing156	GOTO
DC Characteristics	GPR (General Purpose Register)
External Clock Timing	Graphs
Memory Interface Read Timing162	IOH vs. VOH, VDD = 3V
Memory Interface Write Timing161	IOH vs. VOH, VDD = 5V
PWM Timing159	IOL vs. VOL, VDD = 3V
RESET, Watchdog Timer, Oscillator Start-up	IOL vs. VOL, VDD = 5V
Timer and Power-up Timer157	Maximum IDD vs. Frequency
Timer0 Clock Timings158	(External Clock 125°C to -40°C) 167, 197
Timer1, Timer2 and Timer3 Clock Timing 158	Maximum IPD vs. VDD Watchdog Disabled 168, 198
USART Module, Synchronous Receive 160	Maximum IPD vs. VDD Watchdog Enabled 169, 199
USART Module, Synchronous Transmission 160	RC Oscillator Frequency vs.
PIC17C43/44	VDD (Cext = 100 pF)
Absolute Maximum Ratings175	RC Oscillator Frequency vs.
Capture Timing188	VDD (Cext = 22 pF)
CLKOUT and I/O Timing185	RC Oscillator Frequency vs.
DC Characteristics177	VDD (Cext = 300 pF)
External Clock Timing184	Transconductance of LF Oscillator vs.VDD 166, 196
Memory Interface Read Timing191	Transconductance of XT Oscillator vs. VDD 166, 196
Memory Interface Write Timing190	Typical IDD vs. Frequency
Parameter Measurement Information183	(External Clock 25°C) 167, 197
RESET, Watchdog Timer, Oscillator Start-up	Typical IPD vs. VDD Watchdog Disabled 25°C . 168, 198
Timer and Power-up Timer Timing186	Typical IPD vs. VDD Watchdog Enabled 25°C 169, 199
Timer0 Clock Timing187	Typical RC Oscillator vs. Temperature 163, 193
Timer1, Timer2 and Timer3 Clock Timing 187	VTH (Input Threshold Voltage) of I/O Pins vs.
Timing Parameter Symbology182	VDD 172, 202
USART Module Synchronous Receive	VTH (Input Threshold Voltage) of OSC1 Input
Timing189	(In XT, HS, and LP Modes) vs. VDD 173, 203
USART Module Synchronous Transmission	VTH, VIL of $\overline{MCLR}$ , T0CKI and OSC1
Timing189	(In RC Mode) vs. VDD 173, 203
EPROM Memory Access Time Order Suffix31	WDT Timer Time-Out Period vs. VDD 170, 200
Extended Microcontroller29	
Extended Microcontroller Mode31	Н
External Memory Interface31	п
External Program Memory Waveforms31	Hardware Multiplier49
F	
•	I
Family of Devices6	I/O Ports
PIC14000213	Bi-directional64
PIC16C5X214	I/O Ports
PIC16CXXX215	
PIC16C6X216	Programming Considerations
PIC16C7X217	Read-Modify-Write Instructions
PIC16C8X218	INCF
PIC16C9XX219	INCF
PIC17CXX220	INCFSINZ
FERR84, 91	INDF0
FOSC099	INDF1

Indirect Addressing		TSTFSZ	140
Indirect Addressing	39	XORLW	141
Operation		XORWF	
Registers		Instruction Set Summary	
Initialization Conditions For Special Function Registers		INT Pin	
Initializing PORTB		INTE	
Initializing PORTC		INTEDG	
Initializing PORTD		Interrupt on Change Feature	
Initializing PORTE	62	Interrupt Status Register (INTSTA)	22
Instruction Flow/Pipelining		Interrupts	
Instruction Set	110	Context Saving	27
ADDLW	112	Flag bits	
ADDWF	112	TMR1IE	21
ADDWFC		TMR1IF	
ANDLW		TMR2IE	
ANDWF		TMR2IF	
BCF		TMR3IE	
BSF	_	TMR3IF	
BTFSC		Interrupts	
BTFSS	116	Logic	
BTG	116	Operation	25
CALL	117	Peripheral Interrupt Enable	23
CLRF	117	Peripheral Interrupt Request	24
CLRWDT	118	PWM	
COMF		Status Register	
CPFSEQ		Table Write Interaction	
CPFSGT	_	Timing	
	_	<u> </u>	20
CPFSLT	_	Vectors	
DAW		Peripheral Interrupt	
DECF	121	RA0/INT Interrupt	
DECFSNZ	122	T0CKI Interrupt	26
DECFSZ	121	TMR0 Interrupt	26
GOTO	122	Vectors/Priorities	25
INCF		Wake-up from SLEEP	
INCFSNZ		INTF	
INCFSZ		INTSTA	
	_		
IORLW		INTSTA Register	
IORWF		IORLW	
LCALL		IORWF	125
MOVFP	126		
MOVLB	126	1	
MOVLR	127	L	
MOVLW	127		
MOVPF		LCALL	
MOVWF		Long Writes	45
MULLW	_		
		24	
MULWF		M	
NEGW			
NOP	130	Memory	
RETFIE	131	External Interface	31
RETLW	131	External Memory Waveforms	31
RETURN	132	Memory Map (Different Modes)	
RLCF	132	Mode Memory Access	
RLNCF			
RRCF		Organization	
		Program Memory	
RRNCF		Program Memory Map	
SETF	_	Microcontroller	29
SLEEP		Microprocessor	29
SUBLW	135	Minimizing Current Consumption	106
SUBWF	136	MOVFP	
SUBWFB	136	MOVLB	
SWAPF		MOVLR	
TABLRD13			
TABLWT13		MOVLW	
		MOVPF	
TLRD		MOVWF	
TLWT	140	MPASM Assembler	143, 144

MP-C C Compiler145	PORTD	19, 34, 60
MPSIM Software Simulator143, 145	PORTE	19, 34, 62
MULLW129	Power-down Mode	105
Multiply Examples	Power-on Reset (POR)	15, 99
16 x 16 Routine50	Power-up Timer (PWRT)	15, 99
16 x 16 Signed Routine51	PR1	20, 35
8 x 8 Routine49	PR2	20, 35
8 x 8 Signed Routine49	PR3/CA1H	20
MULWF129	PR3/CA1L	20
	PR3H/CA1H	35
A.I.	PR3L/CA1L	35
N	Prescaler Assignments	69
	PRO MATE® Universal Programmer	
NEGW130	PRODH	
NOP130	PRODL	
	Program Counter (PC)	41
0	Program Memory	
O .	External Access Waveforms	31
OERR84	External Connection Diagram	
	Map	
Opcode Field Descriptions	Modes	20
OSC Selection	Extended Microcontroller	20
Oscillator	Microcontroller	
Configuration	Microprocessor	
Crystal	Protected Microcontroller	20
External Clock101		
External Crystal Circuit102	Operation	
External Parallel Resonant Crystal Circuit 102	Organization	29
External Series Resonant Crystal Circuit102	Transfers from Data Memory	
RC102	Protected Microcontroller	
RC Frequencies165, 195	PS0	
Oscillator Start-up Time (Figure)18	PS1	
Oscillator Start-up Timer (OST)15, 99	PS2	,
OST15, 99	PS3	38, 67
OV9, 36	PUSH	27, 39
Overflow (OV)9	PW1DCH	20, 35
	PW1DCL	20, 35
<b>D</b>	PW2DCH	20, 35
P	PW2DCL	20, 35
	PWM	71, 75
Package Marking Information210	Duty Cycle	
Packaging Information205	External Clock Source	
Parameter Measurement Information154	Frequency vs. Resolution	
PC (Program Counter)41	Interrupts	
PCH41	Max Resolution/Frequency for External	
PCL34, 41, 108	Clock Input	77
PCLATH34, 41	Output	
PD	Periods	
PEIE	PWM1	
PEIF22	PWM1ON	
Peripheral Bank42	PWM2	
Peripheral Interrupt Enable	PWM2ON	
Peripheral Interrupt Request (PIR)24	PWRT	•
PICDEM-1 Low-Cost PIC16/17 Demo Board143, 144	FVVK1	15, 99
PICDEM-2 Low-Cost PIC16CXX Demo Board 143, 144		
PICDEM-3 Low-Cost PIC16C9XXX Demo Board144	R	
PICMASTER® RT In-Circuit Emulator		
PICSTART® Low-Cost Development System143	RA1/T0CKI pin	67
	RBIE	
PIE	RBIF	
Pin Compatible Devices	RBPU	
PIR	RC Oscillator	
PM0		
PM199, 106	RC Oscillator Frequencies	•
POP27, 39	RCIE	
POR15, 99	RCIF	
PORTA	RCREG 19, 34	
PORTB	RCSTA 19	
PORTC19, 34, 58	Reading 16-bit Value	69

Receive Status and Control Register83	SWAPF	137
Register File Map33	SYNC	83
Registers	Synchronous Master Mode	93
ALUSTA27, 36	Synchronous Master Reception	95
BRG86	Synchronous Master Transmission	93
BSR27	Synchronous Slave Mode	97
CPUSTA37	•	
File Map	<b>-</b>	
FSR040	Т	
FSR140		
INDF040	T0CKI Pin	
INDF140	TOCKIE	
INTSTA	T0CKIF	
PIE	TOCS	38, 67
PIR	TOIE	22
RCSTA	TOIF	22
Special Function Table34	T0SE	38, 67
TOSTA	T0STA	34, 38
TCON1	T16	71
TCON2	Table Latch	40
	Table Pointer	40
TMR181	Table Read	
TMR281	Example	48
TMR381	Section	
TXSTA83	Table Reads Section	
WREG27		
Reset	TABLRD Operation	
Section15	Timing	
Status Bits and Their Significance	TLRD	
Time-Out in Various Situations	TLRD Operation	44
Time-Out Sequence	Table Write	
RETFIE	Code	46
	Interaction	45
RETLW	Section	
RETURN	TABLWT Operation	
RLCF	Terminating Long Writes	
RLNCF	Timing	
RRCF	TLWT Operation	
RRNCF	To External Memory	
RX Pin Sampling Scheme91		
RX984	To Internal Memory	
RX9D84	TABLRD	
	TABLWT	, ,
	TBLATH	
S	TBLATL	40
	TBLPTRH	34, 40
Sampling91	TBLPTRL	34, 40
Saving STATUS and WREG in RAM27	TCLK12	71
SETF	TCLK3	71
SFR	TCON1	20. 35
SFR (Special Function Registers)	TCON2	•
SFR As Source/Destination	Terminating Long Writes	·
Signed Math 9	Time-Out Sequence	
<u> </u>	Timer Resources	
SLEEP		
Software Simulator (MPSIM)145	Timer0	b/
SPBRG	Timer1	_
Special Features of the CPU	16-bit Mode	
Special Function Registers29, 32, 34, 108	Clock Source Select	
SPEN	On bit	72
SREN84	Section	71, 73
Stack	Timer2	
Operation	16-bit Mode	74
Pointer	Clock Source Select	
Stack	On bit	
STKAL 39	Section	
STKAV	Timer3	1, 1,
SUBLW	Clock Source Select	7/
SUBWF	On bit	
SUBWFB 136	Section	71. 77

Timing Diagrams		Using with PWM	75
Asynchronous Master Transmission	90	TMR1CS	
Asynchronous Reception		TMR1IE	23
Back to Back Asynchronous Master Transmission		TMR1IF	
Interrupt (INT, TMR0 Pins)		TMR1ON	
PIC17C42 Capture1		TMR2	,
PIC17C42 CLKOUT and I/O1		8-bit Mode	
PIC17C42 Memory Interface Read1		External Clock Input	
PIC17C42 Memory Interface Write		In Timer Mode	
PIC17C42 PWM Timing1	59	Timing in External Clock Mode	
PIC17C42 RESET, Watchdog Timer, Oscillator		Two 8-bit Timer/Counter Mode	
Start-up Timer and Power-up Timer1		Using with PWM	
PIC17C42 Timer0 Clock1		TMR2CS	
PIC17C42 Timer1, Timer2 and Timer3 Clock1	58	TMR2IE	
PIC17C42 USART Module, Synchronous		TMR2IF	24
Receive1	60	TMR2ON	72
PIC17C42 USART Module, Synchronous		TMR3	
Transmission1		Dual Capture1 Register Mode	
PIC17C43/44 Capture Timing1		Example, Reading From	
PIC17C43/44 CLKOUT and I/O1	85	Example, Writing To	
PIC17C43/44 External Clock	84	External Clock Input	80
PIC17C43/44 Memory Interface Read	91	In Timer Mode	
PIC17C43/44 Memory Interface Write	90	One Capture and One Period Register	: Mode 78
PIC17C43/44 PWM Timing1	88	Overview	65
PIC17C43/44 RESET, Watchdog Timer, Oscillator		Reading/Writing	80
Start-up Timer and Power-up Timer1	86	Timing in External Clock Mode	80
PIC17C43/44 Timer0 Clock1	87	TMR3CS	71, 77
PIC17C43/44 Timer1, Timer2 and Timer3 Clock 1	87	TMR3H	20, 35
PIC17C43/44 USART Module Synchronous		TMR3IE	23
Receive1	89	TMR3IF	24, 77
PIC17C43/44 USART Module Synchronous		TMR3L	20, 35
Transmission	89	TMR3ON	72, 77
Synchronous Reception		TO	
Synchronous Transmission		Transmit Status and Control Register	
Table Read		TRMT	
Table Write		TSTFSZ	
TMR068,		Turning on 16-bit Timer	
TMR0 Read/Write in Timer Mode		TX9	
TMR1, TMR2, and TMR3 in External Clock Mode		TX9d	
TMR1, TMR2, and TMR3 in Timer Mode		TXEN	
Wake-Up from SLEEP		TXIE	
Fiming Diagrams and Specifications1		TXIF	
Fiming Parameter Symbology1		TXREG19,	
TLRD		TXSTA	
TLWT		17.017	10, 04, 02, 00, 00
TMR0	40		
16-bit Read	69	U	
16-bit Write			
Clock Timing		Upward Compatibility	5
Module		USART	
		Asynchronous Master Transmission	90
Operation		Asynchronous Mode	
Overview		Asynchronous Receive	
Prescaler Assignments		Asynchronous Transmitter	89
Read/Write Considerations		Baud Rate Generator	
Read/Write in Timer Mode		Synchronous Master Mode	
Timing		Synchronous Master Reception	
TMR0 STATUS/Control Register (T0STA)		Synchronous Master Transmission	
TMR0H		Synchronous Slave Mode	
TMR0L		Synchronous Slave Transmit	
TMR120,		Synomical Glave Handlin	
8-bit Mode			
External Clock Input		W	
Overview			
Timer Mode		Wake-up from SLEEP	105
Timing in External Clock Mode		Wake-up from SLEEP Through Interrupt	
Two 8-bit Timer/Counter Mode	73	Watchdog Timer	

WDT	00 402
WDT	
Clearing the WDT	
Normal Timer	103
Period	
Programming Considerations	103
WDTPS0	
WDTPS1	
WREG	
X	
XORLW	141
XORWF	
XORVVI	141
Z	
_	
Z	0.36
	•
Zero (Z)	9

### Example 3-1: Signed Math ......9 Example 3-2: Instruction Pipeline Flow ......14 Example 5-1: Saving STATUS and WREG in RAM ......27 Example 6-1: Indirect Addressing......40 Example 7-1: Table Write ......46 Example 7-2: Table Read......48 Example 8-1: 8 x 8 Multiply Routine......49 Example 8-2: 8 x 8 Signed Multiply Routine......49 Example 8-3: 16 x 16 Multiply Routine ......50 Example 8-4: 16 x 16 Signed Multiply Routine.....51 Example 9-1: Initializing PORTB ......57 Example 9-2: Initializing PORTC......58 Example 9-3: Initializing PORTD......60 Example 9-4: Initializing PORTE ......62 Example 9-5: Read Modify Write Instructions on an I/O Port......64 Example 11-1: 16-Bit Read ......69 Example 11-2: 16-Bit Write......69 Example 12-1: Sequence to Read Capture Registers.......78 Example 12-2: Writing to TMR3 ......80 Example 12-3: Reading from TMR3.....80 Example 13-1: Calculating Baud Rate Error.....86 Example F-1: PIC17C42 to Sleep......223 LIST OF FIGURES Figure 3-1: PIC17C42 Block Diagram .....10 PIC17CR42/42A/43/R43/44 Block Figure 3-2: Diagram.....11 Figure 3-3: Clock/Instruction Cycle.....14 Figure 4-1: Simplified Block Diagram of On-chip Reset Circuit......15 Figure 4-2: Time-Out Sequence on Power-Up (MCLR Tied to VDD) ......17 Figure 4-3: Time-Out Sequence on Power-Up (MCLR NOT Tied to VDD).....17 Figure 4-4: Slow Rise Time (MCLR Tied to VDD) ......17 Figure 4-5: Oscillator Start-Up Time ......18 Figure 4-6: Using On-Chip POR ......18 Figure 4-7: Brown-out Protection Circuit 1......18 Figure 4-8: PIC17C42 External Power-On Reset Circuit (For Slow VDD Power-Up)......18 Figure 4-9: Brown-out Protection Circuit 2......18 Figure 5-1: Interrupt Logic ......21 INTSTA Register (Address: 07h, Figure 5-2: Unbanked)......22 Figure 5-3: PIE Register (Address: 17h, Bank 1) ......23 Figure 5-4: PIR Register (Address: 16h, Bank 1) ......24 INT Pin / T0CKI Pin Interrupt Timing......26 Figure 5-5: Figure 6-1: Program Memory Map and Stack.....29 Figure 6-2: Memory Map in Different Modes ......30 Figure 6-3: External Program Memory Access Waveforms ......31 Typical External Program Memory Figure 6-4: Connection Diagram.....31

**LIST OF EXAMPLES** 

PIC17C42 Register File Map......33

File Map......33

Unbanked)......36

Unbanked)......37

Unbanked)......38

Indirect Addressing......39

Program Counter Operation ......41

PIC17CR42/42A/43/R43/44 Register

ALUSTA Register (Address: 04h,

CPUSTA Register (Address: 06h,

T0STA Register (Address: 05h,

Figure 6-5:

Figure 6-6:

Figure 6-7:

Figure 6-8:

Figure 6-9:

Figure 6-10:

Figure 6-11:

Figure 6-12:	Program Counter using The CALL and	Figure 14-3:	Crystal Operation, Overtone Crystals
Ciaura 6 12.	GOTO Instructions	Figure 14 4	(XT OSC Configuration)
Figure 6-13:	BSR Operation (PIC17C43/R43/44)	Figure 14-4:	External Clock Input Operation
Figure 7-1:	TLWT Instruction Operation	Ciauro 14 E	(EC OSC Configuration)
Figure 7-2:	TABLWT Instruction Operation	Figure 14-5:	External Parallel Resonant Crystal
Figure 7-3:	TLRD Instruction Operation	F: 44 C	Oscillator Circuit
Figure 7-4:	TABLED Instruction Operation	Figure 14-6:	External Series Resonant Crystal
Figure 7-5:	TABLWT Write Timing	F: 44.7	Oscillator Circuit
	(External Memory)46	Figure 14-7:	RC Oscillator Mode
Figure 7-6:	Consecutive TABLWT Write Timing	Figure 14-8:	Watchdog Timer Block Diagram 104
	(External Memory)47	Figure 14-9:	Wake-up From Sleep Through Interrupt 105
Figure 7-7:	TABLRD Timing48	Figure 15-1:	General Format for Instructions 108
Figure 7-8:	TABLRD Timing (Consecutive TABLRD	Figure 15-2:	Q Cycle Activity 109
	Instructions)48	Figure 17-1:	Parameter Measurement Information 154
Figure 9-1:	RA0 and RA1 Block Diagram53	Figure 17-2:	External Clock Timing 155
Figure 9-2:	RA2 and RA3 Block Diagram54	Figure 17-3:	CLKOUT and I/O Timing 156
Figure 9-3:	RA4 and RA5 Block Diagram54	Figure 17-4:	Reset, Watchdog Timer,
Figure 9-4:	Block Diagram of RB<7:4> and RB<1:0>	ŭ	Oscillator Start-Up Timer and
· ·	Port Pins55		Power-Up Timer Timing 157
Figure 9-5:	Block Diagram of RB3 and RB2 Port Pins56	Figure 17-5:	Timer0 Clock Timings 158
Figure 9-6:	Block Diagram of RC<7:0> Port Pins58	Figure 17-6:	Timer1, Timer2, And Timer3 Clock
Figure 9-7:	PORTD Block Diagram	94.0 0.	Timings
riguic o 7.	(in I/O Port Mode)60	Figure 17-7:	Capture Timings
Figure 0.9:		Figure 17-8:	PWM Timings
Figure 9-8:	PORTE Block Diagram (in I/O Bort Mode) 62	Figure 17-9:	USART Module: Synchronous
Figure 0.0.	(in I/O Port Mode)	rigule 17-9.	· · · · · · · · · · · · · · · · · · ·
Figure 9-9:	Successive I/O Operation	F' 47 40	Transmission (Master/Slave) Timing 160
Figure 11-1:	TOSTA Register (Address: 05h,	Figure 17-10:	USART Module: Synchronous Receive
=	Unbanked)		(Master/Slave) Timing
Figure 11-2:	Timer0 Module Block Diagram68	Figure 17-11:	Memory Interface Write Timing 161
Figure 11-3:	TMR0 Timing with External Clock	Figure 17-12:	Memory Interface Read Timing 162
	(Increment on Falling Edge)68	Figure 18-1:	Typical RC Oscillator Frequency
Figure 11-4:	TMR0 Timing: Write High or Low Byte 69		vs. Temperature 163
Figure 11-5:	TMR0 Read/Write in Timer Mode70	Figure 18-2:	Typical RC Oscillator Frequency
Figure 12-1:	TCON1 Register (Address: 16h, Bank 3)71		vs. VDD
Figure 12-2:	TCON2 Register (Address: 17h, Bank 3)72	Figure 18-3:	Typical RC Oscillator Frequency
Figure 12-3:	Timer1 and Timer2 in Two 8-bit	•	vs. VDD
· ·	Timer/Counter Mode73	Figure 18-4:	Typical RC Oscillator Frequency
Figure 12-4:	TMR1 and TMR2 in 16-bit Timer/Counter	· ·	vs. VDD
Ü	Mode74	Figure 18-5:	Transconductance (gm) of LF Oscillator
Figure 12-5:	Simplified PWM Block Diagram75		vs. VDD
Figure 12-6:	PWM Output75	Figure 18-6:	Transconductance (gm) of XT Oscillator
Figure 12-7:	Timer3 with One Capture and One	ga. 0 . 0 0.	vs. VDD
riguic 12 7.	Period Register Block Diagram78	Figure 18-7:	Typical IDD vs. Frequency (External
Figure 12-8:	Timer3 with Two Capture Registers	riguic 10-7.	Clock 25°C)
rigule 12-0.	Block Diagram79	Figure 18-8:	Maximum IDD vs. Frequency (External
Figure 12 0:		rigule 10-0.	
Figure 12-9:	TMR1, TMR2, and TMR3 Operation in	F: 40 0:	Clock 125°C to -40°C)
Fig. 40.40	External Clock Mode80	Figure 18-9:	Typical IPD vs. VDD Watchdog
Figure 12-10:	<u> </u>	E: 10.10	Disabled 25°C
=	Timer Mode81	Figure 18-10:	Maximum IPD vs. VDD Watchdog
Figure 13-1:	TXSTA Register (Address: 15h, Bank 0) 83		Disabled
Figure 13-2:	RCSTA Register (Address: 13h, Bank 0) 84	Figure 18-11:	Typical IPD vs. VDD Watchdog
Figure 13-3:	USART Transmit85		Enabled 25°C 169
Figure 13-4:	USART Receive85	Figure 18-12:	Maximum IPD vs. VDD Watchdog
Figure 13-5:	Asynchronous Master Transmission90		Enabled 169
Figure 13-6:	Asynchronous Master Transmission	Figure 18-13:	WDT Timer Time-Out Period vs. VDD 170
	(Back to Back)90	Figure 18-14:	IOH vs. VOH, VDD = 3V
Figure 13-7:	RX Pin Sampling Scheme91	Figure 18-15:	IOH vs. VOH, VDD = 5V
Figure 13-8:	Asynchronous Reception92	Figure 18-16:	IOL vs. VOL, VDD = 3V
Figure 13-9:	Synchronous Transmission94		IOL vs. VOL, VDD = 5V
Figure 13-10:	Synchronous Transmission		VTH (Input Threshold Voltage) of
-	(Through TXEN)94	-	I/O Pins (TTL) vs. VDD 172
Figure 13-11:	Synchronous Reception (Master Mode,	Figure 18-19:	VTH, VIL of I/O Pins (Schmitt Trigger) vs.
3	SREN)95	3.12.12.101	VDD
Figure 14-1:	Configuration Word99	Figure 18-20:	VTH (Input Threshold Voltage) of OSC1
Figure 14-2:	Crystal or Ceramic Resonator Operation	5	Input (In XT and LF Modes) vs. VDD 173
9 1 7 2.	(XT or LF OSC Configuration)100	Figure 19-1:	Parameter Measurement Information 183
	(AT 51 E1 500 Configuration) 100	1 1gul 6 13-1.	i didinotoi wododioment iniomiation 100

Figure 19-2:	External Clock Timing		Table 6-2:	EPROM Memory Access Time	21
Figure 19-3:	CLKOUT and I/O Timing	00	Table 6-3:	Ordering Suffix	
Figure 19-4:	Reset, Watchdog Timer,		Table 6-3.	Special Function RegistersInterrupt - Table Write Interaction	
	Oscillator Start-Up Timer, and	06			
Figure 10 F:	Power-Up Timer Timing		Table 8-1: Table 9-1:	Performance Comparison	
Figure 19-5:	Timer0 Clock Timings	87		PORTA Functions	
Figure 19-6:	Timer1, Timer2, and Timer3 Clock	0.7	Table 9-2:	Registers/Bits Associated with PORTA	
E: 10 7	Timings		Table 9-3:	PORTB Functions	
Figure 19-7:	Capture Timings		Table 9-4:	Registers/Bits Associated with PORTB	
Figure 19-8:	PWM Timings	88	Table 9-5:	PORTC Functions	
Figure 19-9:	USART Module: Synchronous		Table 9-6:	Registers/Bits Associated with PORTC	
	Transmission (Master/Slave) Timing 18	89	Table 9-7:	PORTD Functions	
Figure 19-10:	USART Module: Synchronous		Table 9-8:	Registers/Bits Associated with PORTD	
	Receive (Master/Slave) Timing 18	89	Table 9-9:	PORTE Functions	
Figure 19-11:	Memory Interface Write Timing		Table 9-10:	Registers/Bits Associated with PORTE	
	(Not Supported in PIC17LC4X Devices) 19	90	Table 11-1:	Registers/Bits Associated with Timer0	
Figure 19-12:	Memory Interface Read Timing		Table 12-1:	Turning On 16-bit Timer	.74
Figure 20-1:	(Not Supported in PIC17LC4X Devices) 19 Typical RC Oscillator Frequency vs.	91	Table 12-2:	Summary of Timer1 and Timer2 Registers	.74
-	Temperature 19	93	Table 12-3:	PWM Frequency vs. Resolution at	
Figure 20-2:	Typical RC Oscillator Frequency			25 MHz	.76
o .	vs. VDD	94	Table 12-4:	Registers/Bits Associated with PWM	
Figure 20-3:	Typical RC Oscillator Frequency		Table 12-5:	Registers Associated with Capture	
	vs. VDD	94	Table 12-6:	Summary of TMR1, TMR2, and TMR3	
Figure 20-4:	Typical RC Oscillator Frequency	•		Registers	.81
ga. o _o	vs. VDD	95	Table 13-1:	Baud Rate Formula	
Figure 20-5:	Transconductance (gm) of LF Oscillator	00	Table 13-2:	Registers Associated with Baud Rate	-00
1 iguio 20 0.	vs. VDD	96	14010 10 2.	Generator	86
Figure 20-6:	Transconductance (gm) of XT Oscillator	00	Table 13-3:	Baud Rates for Synchronous Mode	
riguic 20-0.	vs. VDD	96	Table 13-4:	Baud Rates for Asynchronous Mode	
Figure 20-7:	Typical IDD vs. Frequency (External	50	Table 13-5:	Registers Associated with Asynchronous	.00
riguic 20-7.	Clock 25°C)	07	Table 10-0.	Transmission	۵r
Figure 20 9:	•	91	Toble 12 6:		.90
Figure 20-8:	Maximum IDD vs. Frequency (External	07	Table 13-6:	Registers Associated with Asynchronous	00
Figure 20 0:	Clock 125°C to -40°C)	91	Table 13-7:	Reception	.52
Figure 20-9:	Typical IPD vs. VDD Watchdog Disabled 25°C19	00	Table 13-7.	Registers Associated with Synchronous Master Transmission	0.4
Figure 20 10:		90	Table 12 9:		.94
rigule 20-10.	Maximum IPD vs. VDD Watchdog	00	Table 13-8:	Registers Associated with Synchronous	00
F: 20 44.	Disabled	90	Table 40.0	Master Reception	.90
Figure 20-11:	Typical IPD vs. VDD Watchdog	00	Table 13-9:	Registers Associated with Synchronous	00
F: 00 40	Enabled 25°C	99	T-1-1- 40 40	Slave Transmission	.98
Figure 20-12:	Maximum IPD vs. VDD Watchdog		Table 13-10:	Registers Associated with Synchronous	
F: 00.40	Enabled		T 11 444	Slave Reception	
-	WDT Timer Time-Out Period vs. VDD 20		Table 14-1:	Configuration Locations1	UU
	IOH vs. VOH, VDD = 3V		Table 14-2:	Capacitor Selection for Ceramic	
	IOH vs. VOH, VDD = 5V			Resonators1	01
	IOL vs. VOL, VDD = 3V		Table 14-3:	Capacitor Selection for Crystal	
-	IOL vs. VOL, VDD = 5V	02		OscillatoR1	.01
Figure 20-18:	Vтн (Input Threshold Voltage) of		Table 14-4:	Registers/Bits Associated with the	
	I/O Pins (TTL) vs. VDD20	02		Watchdog Timer1	
Figure 20-19:	VTH, VIL of I/O Pins (Schmitt Trigger)		Table 15-1:	Opcode Field Descriptions1	
	vs. Vdd	03	Table 15-2:	PIC17CXX Instruction Set1	10
Figure 20-20:	VTH (Input Threshold Voltage) of OSC1		Table 16-1:	development tools from microchip1	46
	Input (In XT and LF Modes) vs. VDD 20	03	Table 17-1:	Cross Reference of Device Specs for	
				Oscillator Configurations and Frequencies	
LIST OF TAE	BLES			of Operation (Commercial Devices)1	48
Table 1-1:	PIC17CXX Family of Devices	. 6	Table 17-2:	External Clock Timing Requirements1	55
Table 3-1:	Pinout Descriptions		Table 17-3:	CLKOUT and I/O Timing Requirements1	
Table 4-1:	Time-Out in Various Situations		Table 17-4:	Reset, Watchdog Timer,	
Table 4-2:	STATUS Bits and Their Significance			Oscillator Start-Up Timer and	
Table 4-3:	Reset Condition for the Program Counter			Power-Up Timer Requirements1	57
	and the CPUSTA Register	16	Table 17-5:	Timer0 Clock Requirements1	
Table 4-4:	Initialization Conditions For Special		Table 17-6:	Timer1, Timer2, and Timer3 Clock	-
	Function Registers	19	-	Requirements1	58
Table 5-1:	Interrupt Vectors/Priorities		Table 17-7:	Capture Requirements1	
Table 6-1:	Mode Memory Access		Table 17-8:	PWM Requirements1	
-	, -		-		_

Table 17-9:	Serial Port Synchronous Transmission
	Requirements160
Table 17-10:	Serial Port Synchronous Receive
	Requirements160
Table 17-11:	Memory Interface Write Requirements 161
Table 17-12:	Memory Interface Read Requirements 162
Table 18-1:	Pin Capacitance per Package Type 163
Table 18-2:	RC Oscillator Frequencies165
Table 19-1:	Cross Reference of Device Specs for
	Oscillator Configurations and Frequencies
	of Operation (Commercial Devices)176
Table 19-2:	External Clock Timing Requirements 184
Table 19-3:	CLKOUT and I/O Timing Requirements 185
Table 19-4:	Reset, Watchdog Timer,
	Oscillator Start-Up Timer and
	Power-Up Timer Requirements186
Table 19-5:	Timer0 Clock Requirements 187
Table 19-6:	Timer1, Timer2, and Timer3 Clock
	Requirements187
Table 19-7:	Capture Requirements188
Table 19-8:	PWM Requirements188
Table 19-9:	Synchronous Transmission
	Requirements189
Table 19-10:	Synchronous Receive Requirements 189
Table 19-11:	Memory Interface Write Requirements
	(Not Supported in PIC17LC4X Devices) 190
Table 19-12:	Memory Interface read Requirements
	(Not Supported in PIC17LC4X Devices)191
Table 20-1:	Pin Capacitance per Package Type 193
Table 20-2:	RC Oscillator Frequencies195
Table E-1:	Pin Compatible Devices221
LIST OF EQ	UATIONS
Equation 8-1:	16 x 16 Unsigned Multiplication
Faustian C O	Algorithm
Equation 8-2:	16 x 16 Signed Multiplication
	Algorithm51

## **ON-LINE SUPPORT**

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

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

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

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

# Connecting to the Microchip Internet Web Site

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

### www.microchip.com

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

## ftp.mchip.com/biz/mchip

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

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

# **Connecting to the Microchip BBS**

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

## **Internet:**

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

# mchipbbs.microchip.com

## **CompuServe Communications Network:**

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

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

The following connect procedure applies in most locations.

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

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

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

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

# Systems Information and Upgrade Hot Line

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

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

960513

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

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

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

# **READER RESPONSE**

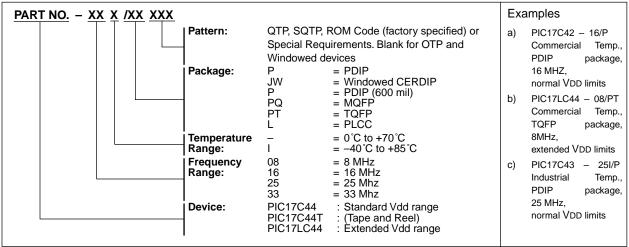
It is our intention to provide you with the best documentation possible to ensure successful use of your Microchip product. If you wish to provide your comments on organization, clarity, subject matter, and ways in which our documentation can better serve you, please FAX your comments to the Technical Publications Manager at (602) 786-7578.

Please list the following information, and use this outline to provide us with your comments about this Data Sheet.

To:	Technical Publications Manager	Total Pages Sent	
RE:	Reader Response		
Fron	m: Name		
		FAV. (	
App	olication (optional):		
	uld you like a reply?YN		
		re Number: <b>DS30412C</b>	
	estions:		
1.	What are the best features of this document?		
2.	How does this document meet your hardware and software development needs?		
3.	. Do you find the organization of this data sheet easy to follow? If not, why?		
4.	What additions to the data sheet do you t	hink would enhance the structure and subject?	
٦.	What additions to the data sheet do you to	mink would enhance the structure and subject:	
5.	What deletions from the data sheet could	be made without affecting the overall usefulness?	
6.	Is there any incorrect or misleading inform	nation (what and where)?	
7.	How would you improve this document?		
٠.	Tiow would you improve this document:		
8.	How would you improve our software, sys	tems, and silicon products?	

# **PIC17C4X Product Identification System**

To order or to obtain information, e.g., on pricing or delivery, please use the listed part numbers, and refer to the factory or the listed sales offices.



Sales and Support

Products supported by a preliminary Data Sheet may possibly have an errata sheet describing minor operational differences and recommended workarounds. To determine if an errata sheet exists for a particular device, please contact one of the following:

- 1. Your local Microchip sales office (see below)
- 2. The Microchip Corporate Literature Center U.S. FAX: (602) 786-7277
- 3.The Microchip's Bulletin Board, via your local CompuServe number (CompuServe membership NOT required).

Please specify which device, revision of silicon and Data Sheet (include Literature #) you are using.

For latest version information and upgrade kits for Microchip Development Tools, please call 1-800-755-2345 or 1-602-786-7302.

NOTES:

NOTES:



# WORLDWIDE SALES AND SERVICE

### **AMERICAS**

## **Corporate Office**

Microchip Technology Inc. 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-786-7200 Fax: 480-786-7277 Technical Support: 480-786-7627 Web Address: http://www.microchip.com

### Atlanta

Microchip Technology Inc. 500 Sugar Mill Road, Suite 200B Atlanta, GA 30350 Tel: 770-640-0034 Fax: 770-640-0307

### **Boston**

Microchip Technology Inc. 5 Mount Royal Avenue Marlborough, MA 01752 Tel: 508-480-9990 Fax: 508-480-8575

## Chicago

Microchip Technology Inc. 333 Pierce Road, Suite 180 Itasca, IL 60143

Tel: 630-285-0071 Fax: 630-285-0075

### **Dallas**

Microchip Technology Inc. 4570 Westgrove Drive, Suite 160 Addison, TX 75248 Tel: 972-818-7423 Fax: 972-818-2924

### Dayton

Microchip Technology Inc. Two Prestige Place, Suite 150 Miamisburg, OH 45342

Tel: 937-291-1654 Fax: 937-291-9175

## Detroit

Microchip Technology Inc. Tri-Atria Office Building 32255 Northwestern Highway, Suite 190 Farmington Hills, MI 48334 Tel: 248-538-2250 Fax: 248-538-2260

# Los Angeles

Microchip Technology Inc. 18201 Von Karman, Suite 1090 Irvine, CA 92612 Tel: 949-263-1888 Fax: 949-263-1338

# **New York**

Microchip Technology Inc. 150 Motor Parkway, Suite 202 Hauppauge, NY 11788 Tel: 631-273-5305 Fax: 631-273-5335

### San Jose

Microchip Technology Inc. 2107 North First Street, Suite 590 San Jose, CA 95131 Tel: 408-436-7950 Fax: 408-436-7955

## **AMERICAS** (continued)

### **Toronto**

Microchip Technology Inc. 5925 Airport Road, Suite 200 Mississauga, Ontario L4V 1W1, Canada Tel: 905-405-6279 Fax: 905-405-6253

### ASIA/PACIFIC

### **Hong Kong**

Microchip Asia Pacific Unit 2101, Tower 2 Metroplaza 223 Hing Fong Road Kwai Fong, N.T., Hong Kong Tel: 852-2-401-1200 Fax: 852-2-401-3431

## Beijing

Microchip Technology, Beijing Unit 915, 6 Chaoyangmen Bei Dajie Dong Erhuan Road, Dongcheng District New China Hong Kong Manhattan Building Beijing 100027 PRC Tel: 86-10-85282100 Fax: 86-10-85282104

### India

Microchip Technology Inc. India Liaison Office No. 6, Legacy, Convent Road Bangalore 560 025, India Tel: 91-80-229-0061 Fax: 91-80-229-0062

### Japan

Microchip Technology Intl. Inc. Benex S-1 6F 3-18-20, Shinyokohama Kohoku-Ku, Yokohama-shi Kanagawa 222-0033 Japan Tel: 81-45-471- 6166 Fax: 81-45-471-6122

### Korea

Microchip Technology Korea 168-1, Youngbo Bldg. 3 Floor Samsung-Dong, Kangnam-Ku Seoul, Korea Tel: 82-2-554-7200 Fax: 82-2-558-5934

### Shanghai

Microchip Technology RM 406 Shanghai Golden Bridge Bldg. 2077 Yan'an Road West, Hong Qiao District Shanghai, PRC 200335 Tel: 86-21-6275-5700 Fax: 86 21-6275-5060

## ASIA/PACIFIC (continued)

## Singapore

Microchip Technology Singapore Pte Ltd. 200 Middle Road #07-02 Prime Centre Singapore 188980

Tel: 65-334-8870 Fax: 65-334-8850

## Taiwan, R.O.C

Microchip Technology Taiwan 10F-1C 207 Tung Hua North Road Taipei, Taiwan, ROC Tel: 886-2-2717-7175 Fax: 886-2-2545-0139

## **EUROPE**

## **United Kingdom** Arizona Microchip Technology Ltd.

505 Eskdale Road Winnersh Triangle Wokingham Berkshire, England RG41 5TU Tel: 44 118 921 5858 Fax: 44-118 921-5835

### Denmark

Microchip Technology Denmark ApS Regus Business Centre Lautrup hoj 1-3 Ballerup DK-2750 Denmark Tel: 45 4420 9895 Fax: 45 4420 9910

### France

Arizona Microchip Technology SARL Parc d'Activite du Moulin de Massy 43 Rue du Saule Trapu Batiment A - Ier Etage 91300 Massy, France Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79

### Germany

Arizona Microchip Technology GmbH Gustav-Heinemann-Ring 125 D-81739 München, Germany Tel: 49-89-627-144 0 Fax: 49-89-627-144-44

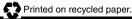
# Italy

Arizona Microchip Technology SRL Centro Direzionale Colleoni Palazzo Taurus 1 V. Le Colleoni 1 20041 Agrate Brianza Milan, Italy Tel: 39-039-65791-1 Fax: 39-039-6899883



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.

All rights reserved. © 1999 Microchip Technology Incorporated. Printed in the USA. 11/99



Information contained in this publication regarding device applications and the like is intended for suggestion only and may be superseded by updates. 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 infiningement of patents or other intellectual property rights arising from such use or otherwise. Use of Microchips products ac 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. The Microchip logo and name are registered trademarks of Microchip Technology Inc. in the U.S.A. and other countries. All rights reserved. All other trademarks mentioned herein are the property of their respective companies.