PIC16C8X

8-Bit CMOS EEPROM Microcontrollers

Devices Included in this Data Sheet

- PIC16C83
- PIC16CR83
- PIC16C84
- PIC16C84A
- PIC16CR84
- Extended voltage range devices available (PIC16**LC**8X)

High Performance RISC CPU Features

- Only 35 single word instructions to learn
- All instructions single cycle (400 ns @ 10 MHz) except for program branches which are two-cycle
- Operating speed: DC - 10 MHz clock input
  DC - 400 ns instruction cycle

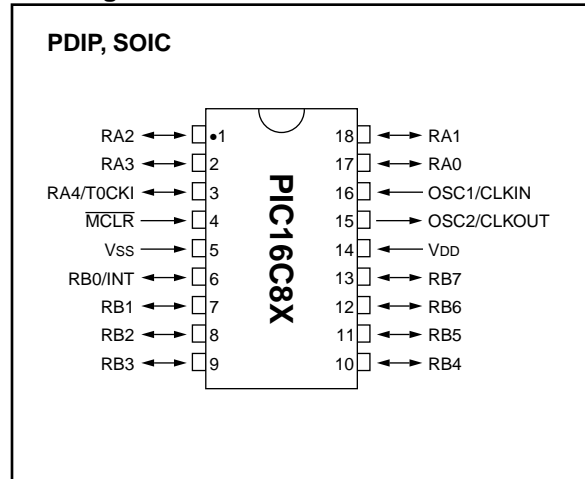| Device	Memory			Freq Max.
	Program	Data		
		RAM	EEPROM	
PIC16C83	512 words	36	64	10 MHz
PIC16CR83	512 words†	36	64	10 MHz
PIC16C84	1 K-words	36	64	10 MHz
PIC16C84A	1 K-words	68	64	10 MHz
PIC16CR84	1 K-words†	68	64	10 MHz

  † ROM Program Memory Devices

- 14-bit wide instructions
- 8-bit wide data path
- 15 special function hardware registers
- Eight-level deep hardware stack
- Direct, indirect and relative addressing modes
- Four interrupt sources:
  - External RB0/INT pin
  - TMR0 timer overflow
  - PORTB<7:4> interrupt on change
  - Data EEPROM write complete
- 1,000,000 data memory EEPROM ERASE/WRITE cycles - Typical
- EEPROM Data Retention > 40 years

Peripheral Features

- 13 I/O pins with individual direction control
- High current sink/source for direct LED drive
  - 25 mA sink max. per pin
  - 20 mA source max. per pin
- TMR0: 8-bit timer/counter with 8-bit programmable prescaler

Pin Diagram

PDIP, SOIC

```
RA2        ← →  [ •1      18 ]  ← →  RA1
RA3        ← →  [  2      17 ]  ← →  RA0
RA4/T0CKI  ← →  [  3      16 ]  ←    OSC1/CLKIN
MCLR       →    [  4   P  15 ]    →  OSC2/CLKOUT
VSS        →    [  5   I  14 ]  ←    VDD
RB0/INT    ← →  [  6   C  13 ]  ← →  RB7
RB1        ← →  [  7   16  12 ]  ← →  RB6
RB2        ← →  [  8   C   11 ]  ← →  RB5
RB3        ← →  [  9   8X  10 ]  ← →  RB4
                      X
```

Special Microcontroller Features

- Power-on Reset (POR)
- Power-up Timer (PWRT)
- 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
- Serial In-System Programming - via two pins (ROM devices support only Data EEPROM programming)

CMOS Technology

- Low-power, high-speed CMOS EEPROM technology
- Fully static design
- Wide operating voltage range:
  - Commercial: 2.0V to 6.0V
  - Industrial: 2.0V to 6.0V
- Low power consumption:
  - < 2 mA typical @ 5V, 4 MHz
  - 15 μA typical @ 2V, 32 kHz
  - < 1 μA typical standby current @ 2V (all devices except PIC16C84)

PIC16C8X

Table of Contents

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, 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

1.0   GENERAL DESCRIPTION

The PIC16C8X is a group in the PIC16CXX family of low-cost, high-performance, CMOS, fully-static, 8-bit microcontrollers. This group contains the following devices:

- PIC16C83
- PIC16CR83
- PIC16C84
- PIC16C84A
- PIC16CR84

All PIC16/17 microcontrollers employ an advanced RISC architecture. PIC16CXX devices have enhanced core features, eight-level deep stack, and multiple internal and external interrupt sources. The separate instruction and data buses of the Harvard architecture allow a 14-bit wide instruction word with a separate 8-bit wide data bus. 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 35 instructions (reduced instruction set) are available. Additionally, a large register set gives some of the architectural innovations used to achieve a very high performance.

PIC16C8X microcontrollers typically achieve a 2:1 code compression and up to a 2:1 speed improvement (at 10 MHz) over other 8-bit microcontrollers in their class.

The PIC16C8X has up to 68 bytes of RAM, 64 bytes of Data EEPROM memory, and 13 I/O pins. A timer/counter is also available.

The PIC16CXX family has special features to 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 LP oscillator minimizes power consumption, XT is a standard crystal, and the HS is for High Speed crystals. The SLEEP (power-down) mode offers power saving. The user can wake the chip from sleep through several external and internal interrupts and resets.

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

The devices with EEPROM program memory allows the same device package to be used for prototyping and production. In-circuit reprogrammability allows the code to be updated without the device being removed from the end application. This is useful in the development of many applications where the device may not be easily accessible, but the prototypes may require code updates. This is also useful for remote applications where the code may need to be updated (such as rate information).

Table 1-1 lists the features of the PIC16C8X, and Appendix F: list the features of all of the Microchip microcontrollers.

A simplified block diagram of the PIC16C8X is shown in Figure 3-1.

The PIC16C8X fits perfectly in applications ranging from high speed automotive and appliance motor control to low-power remote sensors, electronic locks, security devices and smart cards. The EEPROM technology makes customization of application programs (transmitter codes, motor speeds, receiver frequencies, security codes, etc.) extremely fast and convenient. The small footprint packages make this microcontroller series perfect for all applications with space limitations. Low-cost, low-power, high performance, ease of use and I/O flexibility make the PIC16C8X very versatile even in areas where no microcontroller use has been considered before (e.g., timer functions, serial communication, capture and compare, PWM functions and co-processor applications).

The serial in-system programming feature (via two pins) offers flexibility of customizing the product after complete assembly and testing. This feature can be used to serialize a product, store calibration data, or program the device with the current firmware before shipping.

1.1   Family and Upward Compatibility

Those users familiar with the PIC16C5X family of microcontrollers will realize that this is an enhanced version of the PIC16C5X architecture. Please refer to Appendix A for a detailed list of enhancements. Code written for PIC16C5X can be easily ported to the PIC16C8X (Appendix B).

1.2   Development Support

The PIC16CXX family is supported by a full-featured macro assembler, a software simulator, an in-circuit emulator, a low-cost development programmer and a full-featured programmer. A "C" compiler and fuzzy logic support tools are also available.

PIC16C8X

TABLE 1-1: PIC16C8X FAMILIES OF DEVICES

	Maximum Frequency of Operation (MHz)	Program Memory		Data Memory (bytes)	Data EEPROM (bytes)	Timer Module(s)	Interrupt Sources	I/O Pins	Voltage Range (Volts)	Packages
	Clock	**Memory**				**Peripherals**			**Features**	
		EEPROM	ROM							
PIC16C83[1]	10	512	—	36	64	TMR0	4	13	2.0-6.0	18-pin DIP, SOIC
PIC16CR83[1]	10	—	512	36	64	TMR0	4	13	2.0-6.0	18-pin DIP, SOIC
PIC16C84	10	1K	—	36	64	TMR0	4	13	2.0-6.0	18-pin DIP, SOIC
PIC16C84A[1]	10	1K	—	68	64	TMR0	4	13	2.0-6.0	18-pin DIP, SOIC
PIC16CR84[1]	10	—	1K	68	64	TMR0	4	13	2.0-6.0	18-pin DIP, SOIC

Notes: - All PIC16/17 family devices have Power-on Reset, selectable Watchdog Timer, selectable code protect, and high I/O current capability.
- All PIC16CXX family devices use serial programming with clock pin RB6 and data pin RB7.
1: Please contact your local sales office for availability of these devices.

2.0   PIC16C8X 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 this section. When placing orders, please use the "PIC16C8X Product Identification System" at the back of this data sheet to specify the correct part number.

There are four device "types" as indicated in the device number.

1. **C**, as in PIC16**C**84. These devices have EEPROM program memory and operate over the standard voltage range.

2. **LC**, as in PIC16**LC**84. These devices have EEPROM program memory and operate over an extended voltage range.

3. **CR**, as in PIC16**CR**83. These devices have ROM program memory and operate over the standard voltage range.

4. **LCR**, as in PIC16**LCR**84. These devices have ROM program memory and operate over an extended voltage range.

When discussing memory maps and other architectural features, the use of **C** (**CR**) also implies the **LC** (**LCR**) versions.

2.1   Electrically Erasable Devices

These devices are offered in the lower cost plastic package, even though the device can be erased and reprogrammed. This allows the same device to be used for prototype development and pilot programs as well as production.

A further advantage of the electrically erasable version is that they can be erased and reprogrammed in-circuit, or by device programmers, such as Microchip's PICSTART™ or PRO MATE™ programmers.

2.2   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 have all EEPROM locations and configuration options already programmed by the factory. Certain code and prototype verification procedures do apply before production shipments are available.

For information on submitting a QTP code, please contact your Microchip Regional Sales Office.

2.3   Serialized Quick-Turnaround-Production (SQTP^{SM}) Devices

Microchip offers the 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.

For information on submitting a SQTP code, please contact your Microchip Regional Sales Office.

2.4   ROM Devices

Some of Microchip's devices have a corresponding device where the program memory is a ROM. These devices give a cost savings over Microchip's traditional user programmed devices (EPROM, EEPROM).

ROM devices (PIC16CR8X) do not allow serialization information in the program memory space. The user may program this information into the Data EEPROM.

For information on submitting a ROM code, please contact your Microchip Regional Sales Office.

NOTES:

3.0   ARCHITECTURAL OVERVIEW

The high performance of the PIC16CXX family can be attributed to a number of architectural features commonly found in RISC microprocessors. To begin with, the PIC16CXX uses a Harvard architecture. This architecture has the program and data are 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. PIC16C8X opcodes are 14-bits wide, enabling single word instructions. The full 14-bit wide program memory bus fetches a 14-bit instruction in a single cycle. A two-stage pipeline overlaps fetch and execution of instructions (Example 3-1). Consequently, all instructions execute in a single cycle (400 ns @ 10 MHz) except for program branches.

The PIC16C83 and PIC16CR83 address 512 x 14 of program memory, and the PIC16C84, PIC16C84A, and PIC16CR84 address 1K x 14 program memory. All program memory is internal.

The PIC16CXX can directly or indirectly address its register files or data memory. All special function registers including the program counter are mapped in the data memory. 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 PIC16CXX simple yet efficient. In addition, the learning curve is reduced significantly.

PIC16CXX 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. In two-operand instructions, typically one operand is the working register (W register), and the other operand is a file register or an immediate constant. In single operand instructions, the operand is either the W register or a file register.

The W register is an 8-bit working register used for ALU operations. It is not an addressable register.

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 $\overline{borrow}$ and $\overline{digit\ borrow}$ out bit, respectively, in subtraction. See the SUBLW and SUBWF instructions for examples.

A simplified block diagram for the PIC16C8X is shown in Figure 3-1, its corresponding pin description is shown in Table 3-1.
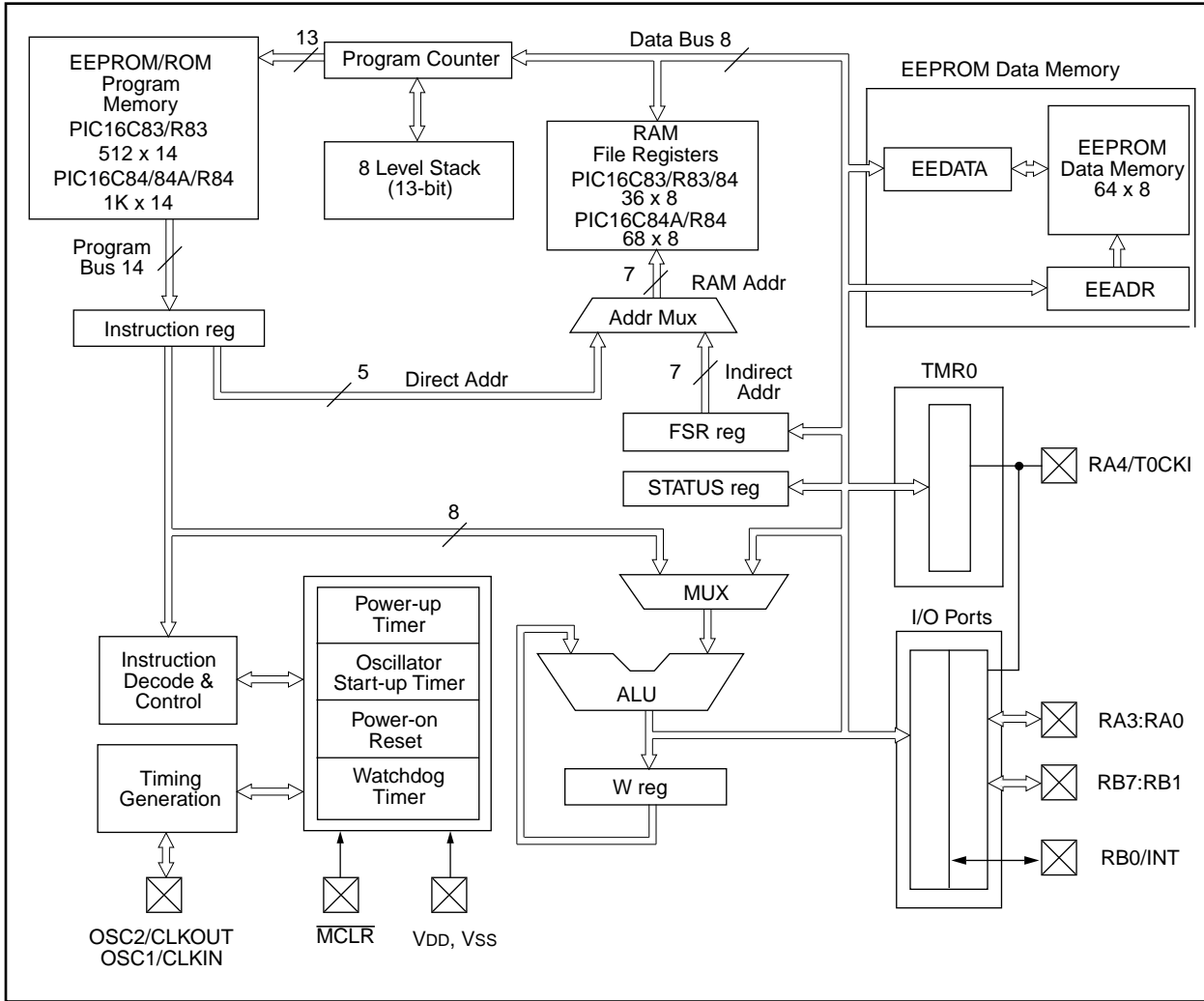
PIC16C8X

FIGURE 3-1: PIC16C8X BLOCK DIAGRAM

© 1995 Microchip Technology Inc.

TABLE 3-1: PIC16C8X PINOUT DESCRIPTION

Pin Name	DIP No.	SOIC No.	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	16	16	I	ST/CMOS [3]	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	15	15	O	—	Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. In RC mode, OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate.
$\overline{\text{MCLR}}$	4	4	I/P	ST	Master clear (reset) input/programming voltage input. This pin is an active low reset to the device.
					PORTA is a bi-directional I/O port.
RA0	17	17	I/O	TTL	
RA1	18	18	I/O	TTL	
RA2	1	1	I/O	TTL	
RA3	2	2	I/O	TTL	
RA4/T0CKI	3	3	I/O	ST	Can also be selected to be the clock input to the TMR0 timer/counter. Output is open drain type.
					PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs.
RB0/INT	6	6	I/O	TTL/ST [1]	RB0/INT can also be selected as an external interrupt pin.
RB1	7	7	I/O	TTL	
RB2	8	8	I/O	TTL	
RB3	9	9	I/O	TTL	
RB4	10	10	I/O	TTL	Interrupt on change pin.
RB5	11	11	I/O	TTL	Interrupt on change pin.
RB6	12	12	I/O	TTL/ST [2]	Interrupt on change pin. Serial programming clock.
RB7	13	13	I/O	TTL/ST [2]	Interrupt on change pin. Serial programming data.
V_{SS}	5	5	P	—	Ground reference for logic and I/O pins.
V_{DD}	14	14	P	—	Positive supply for logic and I/O pins.

Legend: I= input   O = output   I/O = Input/Output   P = power
         — = Not used   TTL = TTL input   ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt (except PIC16C84, which remains TTL).
    2: This buffer is a Schmitt Trigger input when used in serial programming mode.
    3: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

PIC16C8X

3.1 Clocking Scheme/Instruction Cycle

The clock input (from OSC1) is internally divided by four to generate four non-overlapping quadrature clocks namely Q1, Q2, Q3 and Q4. Internally, the program counter (PC) is incremented every Q1, 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 is shown in Figure 3-2.

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 3-1).

A fetch cycle begins with the Program Counter (PC) incrementing in Q1.

In the execution cycle, the fetched instruction is latched into the "Instruction Register" 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 3-2: CLOCK/INSTRUCTION CYCLE
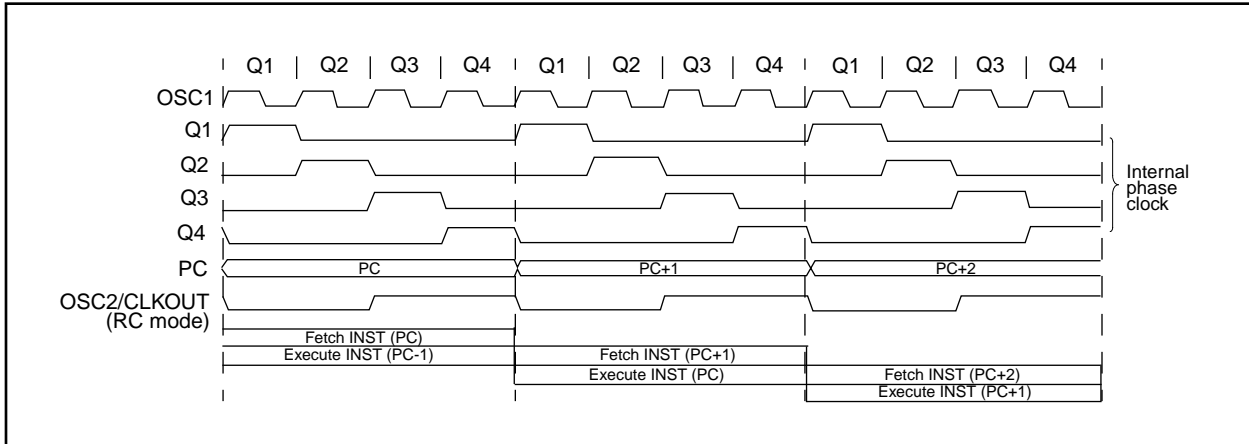
EXAMPLE 3-1: INSTRUCTION PIPELINE FLOW

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.

© 1995 Microchip Technology Inc.
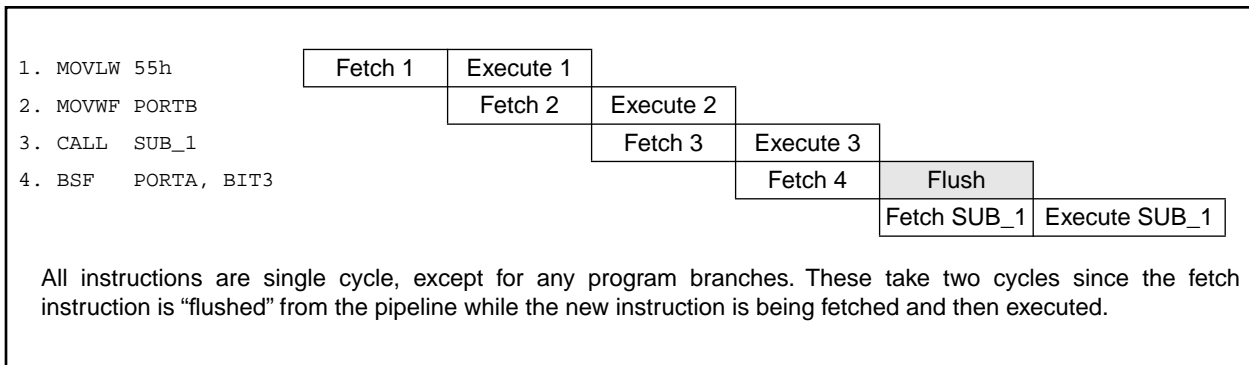
4.0   MEMORY ORGANIZATION

There are two memory blocks in the PIC16C8X. These are the program memory and the 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 the 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.

The data memory area also contains the data EEPROM memory. This memory is not directly mapped into the data memory, but is indirectly mapped. That is an indirect address pointer specifies the address of the data EEPROM memory to read/write. The 64 bytes of data EEPROM memory have the address range 0h-3Fh. More details on the EEPROM memory can be found in Section 7.0.

4.1   Program Memory Organization

The PIC16CXX has a 13-bit program counter capable of addressing an 8K x 14 program memory space. For the PIC16C83 and PIC16CR83 only the first 512 x 14 (0000h-01FFh) are physically implemented, and for the PIC16C84, PIC16C84A, and PIC16CR84 only the first 1K x 14 (0000h-03FFh) are physically implemented. Accessing a location above the physically implemented address will cause a wraparound. For example, for the PIC16C84 locations 20h, 420h, 820h, C20h, 1020h, 1420h, 1820h, and 1C20h will be the same instruction.

The reset vector is at 0000h and the interrupt vector is at 0004h (Figure 4-1).

FIGURE 4-1:   PROGRAM MEMORY MAP AND STACK
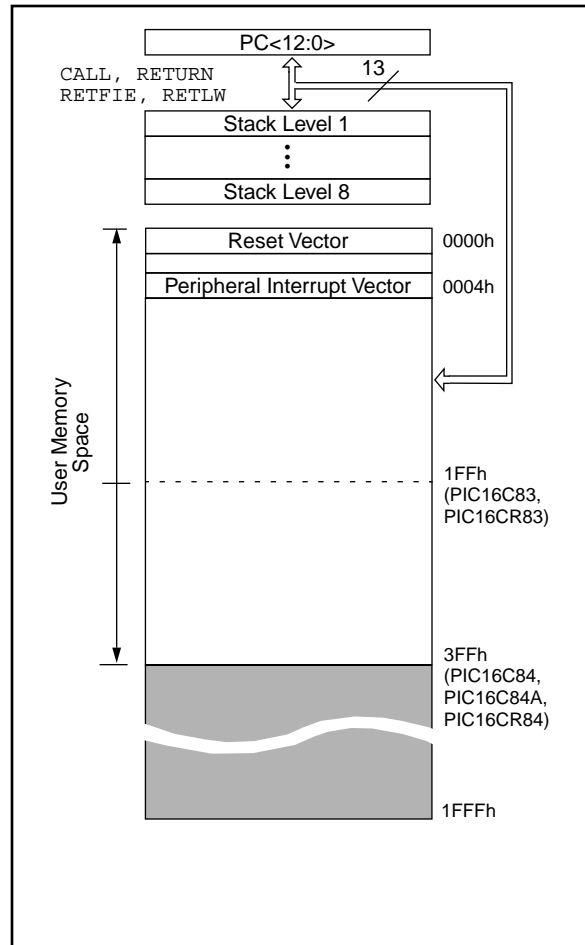
4.2   Data Memory Organization

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

Portions of data memory are banked. This is for both the SFR area and the GPR area. The GPR area is banked to allow greater than 116 bytes of general purpose RAM. The banked areas of the SFR 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 STATUS Register. Figure 4-2 shows the data memory map organization.

Instructions MOVWF and MOVF can move values from the W register to any location in the register file ("F"), and vice-versa.

The entire data memory can be accessed either directly using the absolute address of each register file or indirectly through the File Select Register (FSR) (Section 4.4). Indirect addressing uses the present value of the RP1:RP0 bits for access into the banked areas of data memory.

Data memory is partitioned into two banks which contain the general purpose registers and the special function registers. Bank 0 is selected by clearing the RP0 bit (STATUS<5>). Setting the RP0 bit selects Bank 1. Each Bank extends up to 7Fh (128 bytes). The lower locations of each Bank are reserved for the Special Function Registers. Above the Special Function Registers are General Purpose Registers implemented as static RAM. (Figure 4-2)

4.2.1   GENERAL PURPOSE REGISTER FILE

All devices have some amount of General Purpose Register (GPR) area. Each GPR is 8-bits wide and is accessed either directly, or indirectly through the FSR (Section 4.4).

Architecturally, the GPR area starts at address 0Ch for bank 0 and 8Ch for bank 1. When more than 116 bytes of GPR are present on the device, banking must be performed to access the additional memory space.

PIC16C8X devices have up to 68 bytes of GPR memory, and therefore do not require banking of the GPR memory. Any access to Bank 1 will cause the access to occur in Bank 0. That is, the MSb of the 8-bit direct address will be ignored.

4.2.2   SPECIAL FUNCTION REGISTERS

The Special Function Registers (Figure 4-2 and Table 4-1) are used by the CPU and Peripheral functions to control the device operation. These registers are static RAM.

The special function registers can be classified into two sets, core and peripheral. Those associated with the "core" functions are described in this section. Those related to the operation of the peripheral features are described in the section for that specific feature.
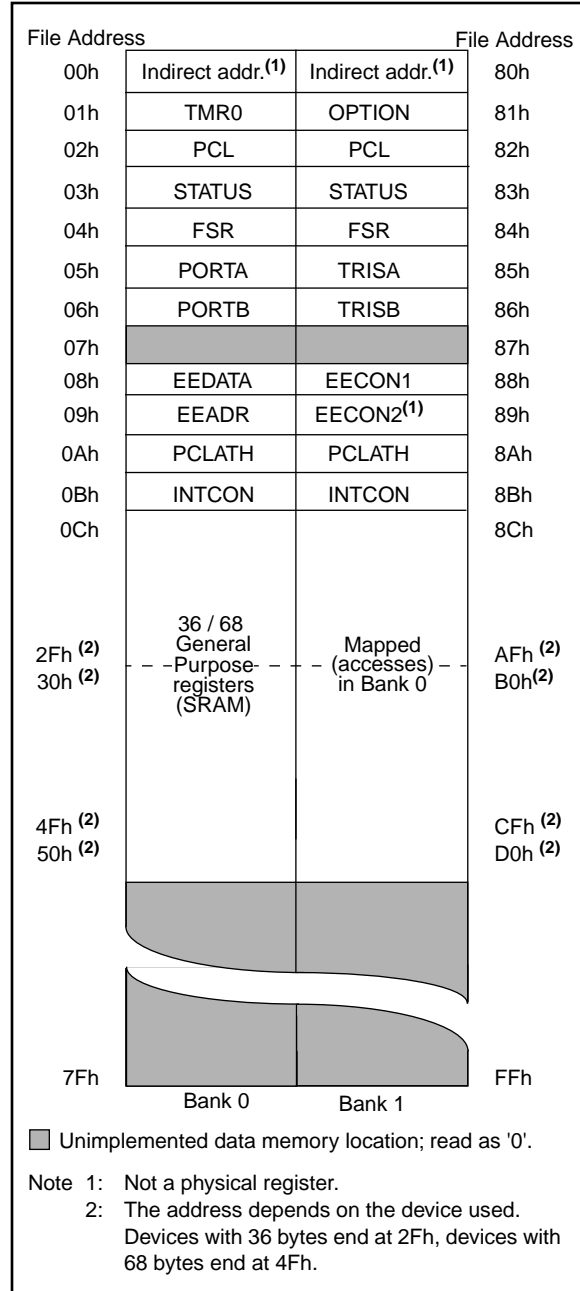
FIGURE 4-2:   REGISTER FILE MAP

TABLE 4-1:   REGISTER FILE SUMMARY

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 (Note3)
Bank 0											
00h	INDF	Uses contents of FSR to address data memory (not a physical register)								---- ----	---- ----
01h	TMR0	8-bit real-time clock/counter								xxxx xxxx	uuuu uuuu
02h	PCL	Low order 8 bits of the Program Counter (PC)								0000 0000	0000 0000
03h	STATUS [2]	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000q quuu
04h	FSR	Indirect data memory address pointer 0								xxxx xxxx	uuuu uuuu
05h	PORTA	—	—	—	RA4/T0CKI	RA3	RA2	RA1	RA0	---x xxxx	---u uuuu
06h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0/INT	xxxx xxxx	uuuu uuuu
07h		Unimplemented location, read as '0'								---- ----	---- ----
08h	EEDATA	EEPROM data register								xxxx xxxx	uuuu uuuu
09h	EEADR	EEPROM address register								xxxx xxxx	uuuu uuuu
0Ah	PCLATH	—	—	—	Write buffer for upper 5 bits of the PC [1]					---0 0000	---0 0000
0Bh	INTCON	GIE	EEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
Bank 1											
80h	INDF	Uses contents of FSR to address data memory (not a physical register)								---- ----	---- ----
81h	OPTION	$\overline{RBPU}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
82h	PCL	Low order 8 bits of Program Counter (PC)								0000 0000	0000 0000
83h	STATUS [2]	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000q quuu
84h	FSR	Indirect data memory address pointer 0								xxxx xxxx	uuuu uuuu
85h	TRISA	—	—	—	PORTA data direction register					---1 1111	---1 1111
86h	TRISB	PORTB data direction register								1111 1111	1111 1111
87h		Unimplemented location, read as '0'								---- ----	---- ----
88h	EECON1	—	—	—	EEIF	WRERR	WREN	WR	RD	---0 x000	---0 q000
89h	EECON2	EEPROM control register 2 (not a physical register)								---- ----	---- ----
0Ah	PCLATH	—	—	—	Write buffer for upper 5 bits of the PC [1]					---0 0000	---0 0000
0Bh	INTCON	GIE	EEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF		

PIC16C8X

4.2.2.1   STATUS REGISTER

The STATUS register contains the arithmetic status of the ALU, the RESET status and the bank select bit for data memory.

As with any register, the STATUS register can be the destination for any instruction. If the STATUS 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 device logic. Furthermore, the $\overline{TO}$ and $\overline{PD}$ bits are not writable. Therefore, the result of an instruction with the STATUS register as destination may be different than intended.

For example, `CLRF STATUS` will clear the upper-three bits and set the Z bit.   This leaves the STATUS register as `000u uluu` (where `u` = unchanged).

Only the `BCF, BSF, SWAPF` and `MOVWF` instructions should be used to alter the STATUS register (Table 9-2) because these instructions do not affect any status bit.

Note 1:	The IRP and RP1 bits (STATUS<7:6>) are not used by the PIC16C8X and should be programmed as cleared. Use of these bits as general purpose R/W bits is NOT recommended, since this may affect upward compatibility with future products.
Note 2:	The C and DC bits operate as a $\overline{\text{borrow}}$ and digit borrow out bit, respectively, in subtraction. See the `SUBLW` and `SUBWF` instructions for examples.
Note 3:	When the STATUS register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. The specified bit(s) will be updated according to device logic.

FIGURE 4-3: STATUS REGISTER (ADDRESS 03h, 83h)

R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C

bit7   bit0

R = Readable bit
W = Writable bit
U = Unimplemented bit,
   read as '0'
- n = Value at POR reset

bit 7: **IRP**: Register Bank Select bit (used for indirect addressing)
0 = Bank 0, 1 (00h - FFh)
1 = Bank 2, 3 (100h - 1FFh)
The IRP bit is not used by the PIC16C8X. IRP should be maintained clear.

bit 6-5: **RP1:RP0**: Register Bank Select bits (used for direct addressing)
00 = Bank 0 (00h - 7Fh)
01 = Bank 1 (80h - FFh)
10 = Bank 2 (100h - 17Fh)
11 = Bank 3 (180h - 1FFh)
Each bank is 128 bytes. Only bit RP0 is used by the PIC16C8X. RP1 should be maintained clear.

bit 4: $\overline{TO}$: Time-out bit
1 = After power-up, CLRWDT instruction, or SLEEP instruction
0 = A WDT time-out occurred

bit 3: $\overline{PD}$: Power-down bit
1 = After power-up or by the CLRWDT instruction
0 = By execution of the SLEEP instruction

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

bit 1: **DC**: Digit carry/$\overline{borrow}$ bit (for ADDWF and ADDLW instructions) (For $\overline{borrow}$ the polarity is reversed)
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

bit 0: **C**: Carry/$\overline{borrow}$ bit (for ADDWF and ADDLW instructions)
1 = A carry-out from the most significant bit of the result occurred
0 = No carry-out from the most significant bit of the result occurred
Note: For $\overline{borrow}$ the polarity is reversed. A subtraction is executed by adding the two's complement of
the second operand. For rotate (RRF, RLF) instructions, this bit is loaded with either the high or low
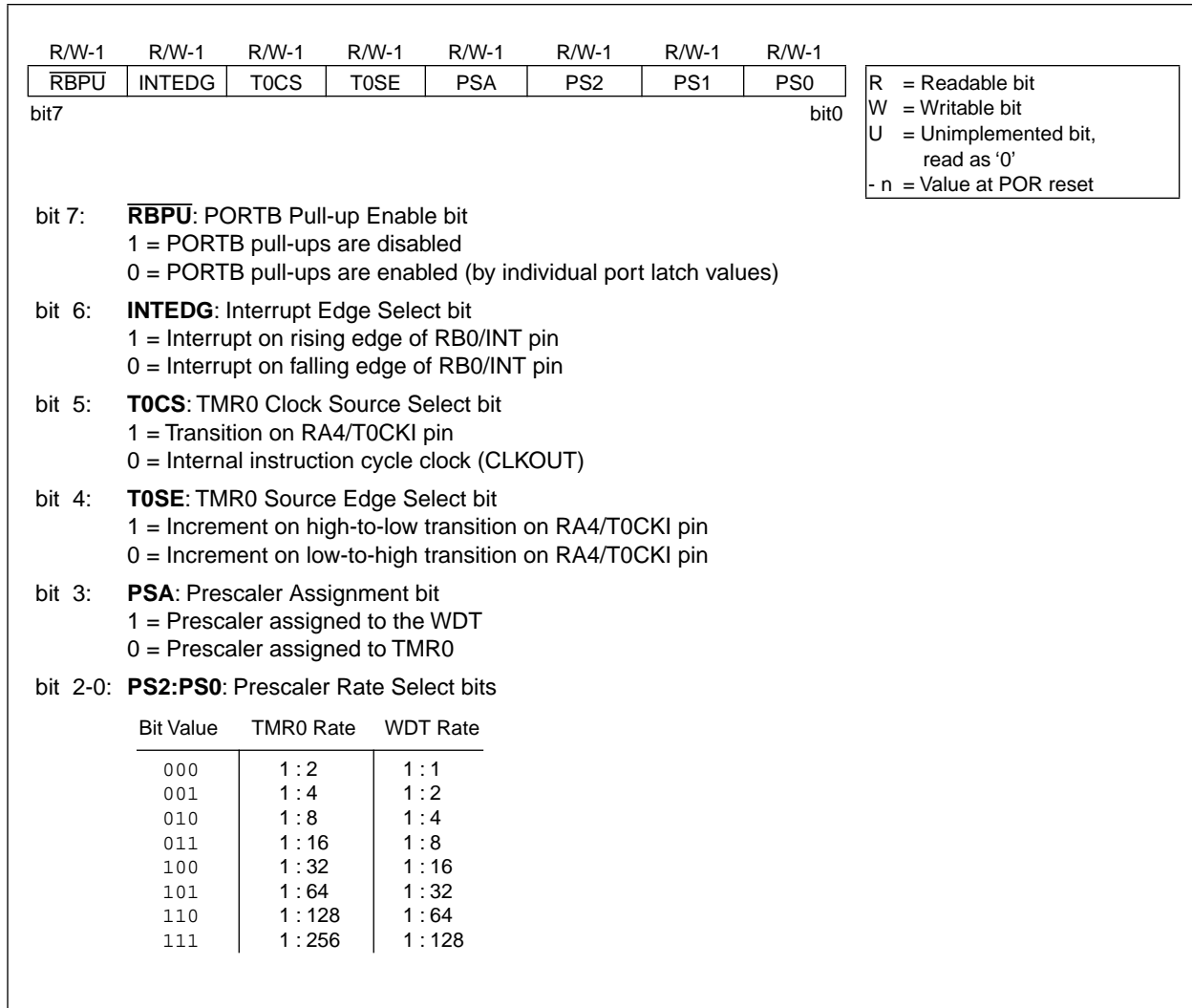order bit of the source register.

PIC16C8X

The OPTION register is a readable and writable register which contains various control bits to configure the TMR0/WDT prescaler, the external INT interrupt, TMR0, and the weak pull-ups on PORTB.

Note:	When the prescaler is assigned to the WDT (PSA = '1'), TMR0 has a 1:1 prescaler assignment.

FIGURE 4-4:   OPTION REGISTER (ADDRESS 81h)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
$\overline{RBPU}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0

bit7   bit0

R = Readable bit
W = Writable bit
U = Unimplemented bit,
       read as '0'
- n = Value at POR reset

bit 7:   **$\overline{RBPU}$**: PORTB Pull-up Enable bit
           1 = PORTB pull-ups are disabled
           0 = PORTB pull-ups are enabled (by individual port latch values)

bit 6:   **INTEDG**: Interrupt Edge Select bit
           1 = Interrupt on rising edge of RB0/INT pin
           0 = Interrupt on falling edge of RB0/INT pin

bit 5:   **T0CS**: TMR0 Clock Source Select bit
           1 = Transition on RA4/T0CKI pin
           0 = Internal instruction cycle clock (CLKOUT)

bit 4:   **T0SE**: TMR0 Source Edge Select bit
           1 = Increment on high-to-low transition on RA4/T0CKI pin
           0 = Increment on low-to-high transition on RA4/T0CKI pin

bit 3:   **PSA**: Prescaler Assignment bit
           1 = Prescaler assigned to the WDT
           0 = Prescaler assigned to TMR0

bit 2-0: **PS2:PS0**: Prescaler Rate Select bits

Bit Value	TMR0 Rate	WDT Rate
000	1 : 2	1 : 1
001	1 : 4	1 : 2
010	1 : 8	1 : 4
011	1 : 16	1 : 8
100	1 : 32	1 : 16
101	1 : 64	1 : 32
110	1 : 128	1 : 64
111	1 : 256	1 : 128

4.2.2.3   INTCON REGISTER

The INTCON register is a readable and writable register which contains the various enable bits for all interrupt sources.

Note:	Interrupt flag bits get set when an interrupt condition occurs regardless of the state of its corresponding enable bit or the global enable bit, GIE (INTCON<7>).

FIGURE 4-5:   INTCON REGISTER (ADDRESS 0Bh, 8Bh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE	EEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF

bit7   bit0

```
R   = Readable bit
W   = Writable bit
U   = Unimplemented bit,
      read as '0'
- n = Value at POR reset
```

bit 7:   **GIE:** Global Interrupt Enable bit
1 = Enables all un-masked interrupts
0 = Disables all interrupts

Note: For the operation of the interrupt structure, please refer to Section 8.5.

bit 6:   **EEIE**: EE Write Complete Interrupt Enable bit
1 = Enables the EE write complete interrupt
0 = Disables the EE write complete interrupt

bit 5:   **T0IE**: TMR0 Overflow Interrupt Enable bit
1 = Enables the TMR0 interrupt
0 = Disables the TMR0 interrupt

bit 4:   **INTE**: RB0/INT Interrupt Enable bit
1 = Enables the RB0/INT interrupt
0 = Disables the RB0/INT interrupt

bit 3:   **RBIE**: RB Port Change Interrupt Enable bit
1 = Enables the RB port change interrupt
0 = Disables the RB port change interrupt

bit 2:   **T0IF**: TMR0 overflow interrupt flag bit
1 = TMR0 has overflowed (must be cleared in software)
0 = TMR0 did not overflow

bit 1:   **INTF**: RB0/INT Interrupt Flag bit
1 = The RB0/INT interrupt occurred
0 = The RB0/INT interrupt did not occur

bit 0:   **RBIF**: RB Port Change Interrupt Flag bit
1 = When at least one of the RB7:RB4 pins changed state (must be cleared in software)
0 = None of the RB7:RB4 pins have changed state

PIC16C8X

4.3   PCL and PCLATH

The Program Counter (PC) is 13-bits wide. The low byte is the PCL register, which is a readable and writable register. The high byte of the PC (PC<12:8>) is not directly readable nor writable and comes from the PCLATH register. The PCLATH (PC latch high) register is a holding register for PC<12:8>. The contents of PCLATH are transferred to the upper byte of the program counter when the PC is loaded with a new value. This occurs during a CALL, GOTO or a write to PCL. The high bits of PC are loaded from PCLATH as shown in Figure 4-6.

FIGURE 4-6:   LOADING OF PC IN DIFFERENT SITUATIONS
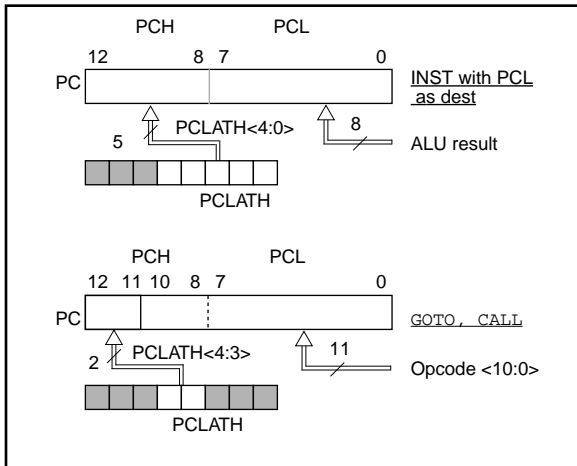
4.3.1   COMPUTED GOTO

A computed GOTO is accomplished by adding an offset to the program counter (ADDWF   PCL). When doing a table read using a computed GOTO method, care should be exercised if the table location crosses a PCL memory boundary (each 256 byte block). Refer to the application note *"Implementing a Table Read"* (AN556).

4.3.2   STACK

The PIC16CXX has an 8 deep x 13-bit wide hardware stack (Figure 4-1). The stack space is not part of either program or data space and the stack pointer is not readable or writable. The entire 13-bit PC is PUSH'ed onto the stack when a CALL instruction is executed or an interrupt is acknowledged. The stack is POP'ed 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. That is, after the stack has been PUSH'ed eight times, the ninth push overwrites the value that was stored from the first push. The tenth push overwrites the second push (and so on).

If the stack is effectively POP'ed nine times, the PC value is the same as the value from the first POP.

Note 1:	There are no status bits to indicate stack overflow or stack underflow conditions.
Note 2:	There are no instructions 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 address

4.3.3   PROGRAM MEMORY PAGING

The PIC16C83 and PIC16CR83 have 512 words of program memory. The PIC16C84, PIC16C84A, and PIC16CR84 have 1K of program memory. The CALL and GOTO instructions have an 11-bit address range. This 11-bit address range allows a branch within a 2K program memory page size. For future PIC16C8X program memory expansion, there must be another two bits to specify the program memory page. These paging bits come from the PCLATH<4:3> bits (Figure 4-6). When doing a CALL or a GOTO instruction, the user must ensure that these page bits (PCLATH<4:3>) are programmed to the desired program memory page. If a CALL instruction (or interrupt) is executed, the entire 13-bit PC is pushed onto the stack. Therefore, manipulation of the PCLATH<4:3> is not required for the return instructions (which POPs the PC from the stack).

Note:	The PIC16C8X ignores the PCLATH<4:3> bits, which are used for program memory pages 1, 2 and 3 (0800h - 1FFFh). The use of PCLATH<4:3> as general purpose R/W bits is not recommended since this may affect upward compatibility with future products.

4.4 Indirect Addressing, INDF and FSR Registers

The INDF register is not a physical register and is used in conjunction with the FSR register to perform indirect addressing.

Indirect addressing is possible by using the INDF register. Any instruction using the INDF register actually accesses data pointed to by the file select register (FSR). Reading INDF itself indirectly (FSR = 0) will produce 00h. Writing to the INDF register indirectly results in a no-operation (although status bits may be affected). An effective 9-bit address is obtained by concatenating the 8-bit FSR register and the IRP bit (STATUS<7>), as shown in Figure 4-7. However, IRP is not used in the PIC16C8X.

A simple program to clear RAM locations 20h-2Fh using indirect addressing is shown in Example 4-1.

EXAMPLE 4-1: INDIRECT ADDRESSING

```
          movlw  0x20  ;initialize pointer
          movwf  FSR   ;  to RAM
NEXT      clrf   INDF  ;clear INDF register
          incf   FSR   ;inc pointer
          btfss  FSR,4 ;all done?
          goto   NEXT  ;NO, clear next
CONTINUE
          :            ;YES, continue
```

FIGURE 4-7: DIRECT/INDIRECT ADDRESSING

Note 1: PIC16C83, PIC16CR83, and PIC16C84 devices.
    2: PIC16C84A and PIC16CR84 devices
    3: For memory map detail see Figure 4-1.
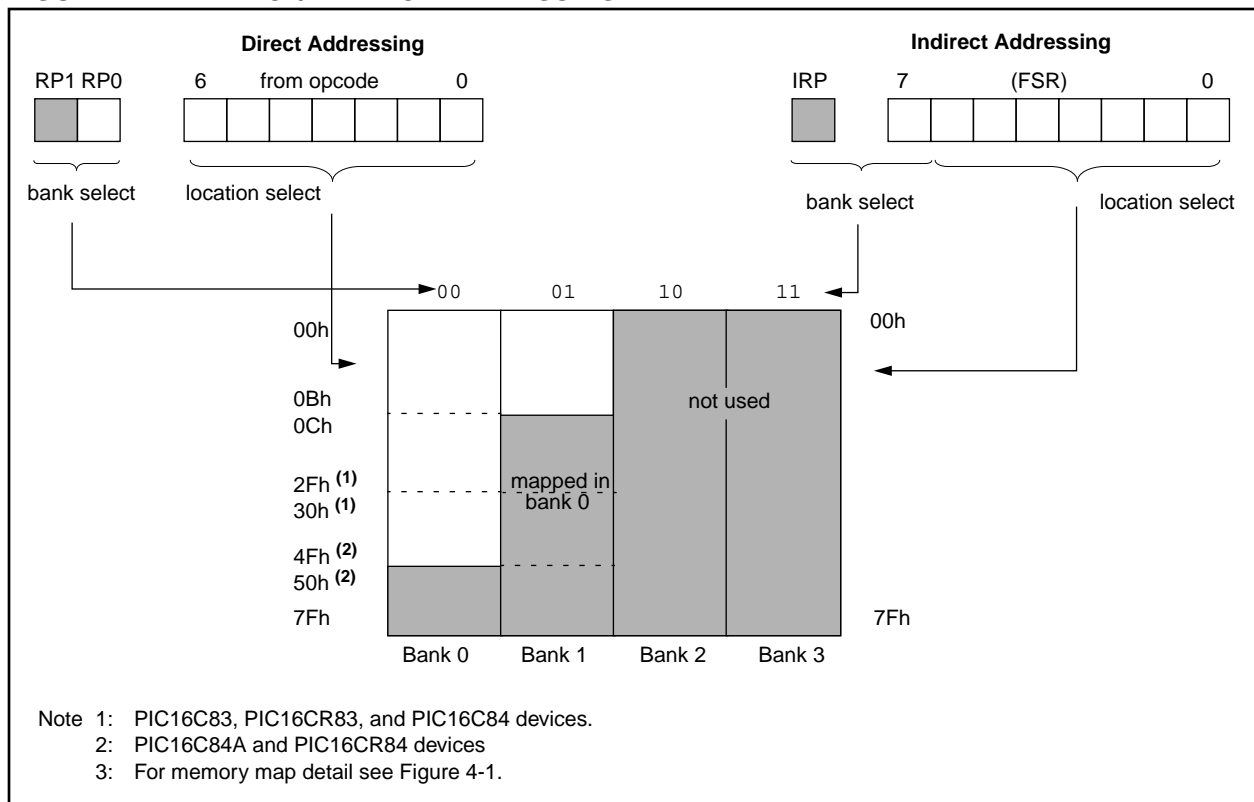
NOTES:

5.0   I/O PORTS

The PIC16C8X family has two ports, PORTA and PORTB. Some port pins are multiplexed with an alternate function for other features on the device.

5.1   PORTA and TRISA Registers

PORTA is a 5-bit wide latch. RA4 is a Schmitt Trigger input and an open drain output. All other RA port pins have TTL input levels and full CMOS output drivers. All pins have data direction bits (TRIS registers) which can configure these pins as output or input.

A '1' on any bit in the TRISA register puts the corresponding output driver in a hi-impedance mode. A '0' on any bit in the TRISA register puts the contents of the output latch on the selected pin(s).

Reading the PORTA register reads the status of the pins whereas writing to it will write to the port latch. All write operations are read-modify-write operations. So a write to a port implies that the port pins are first read, then this value is modified and written to the port data latch.

The RA4 pin is multiplexed with the TMR0 clock input.

FIGURE 5-1:   BLOCK DIAGRAM OF PINS RA3:RA0

Note: I/O pins have protection diodes to VDD and VSS.
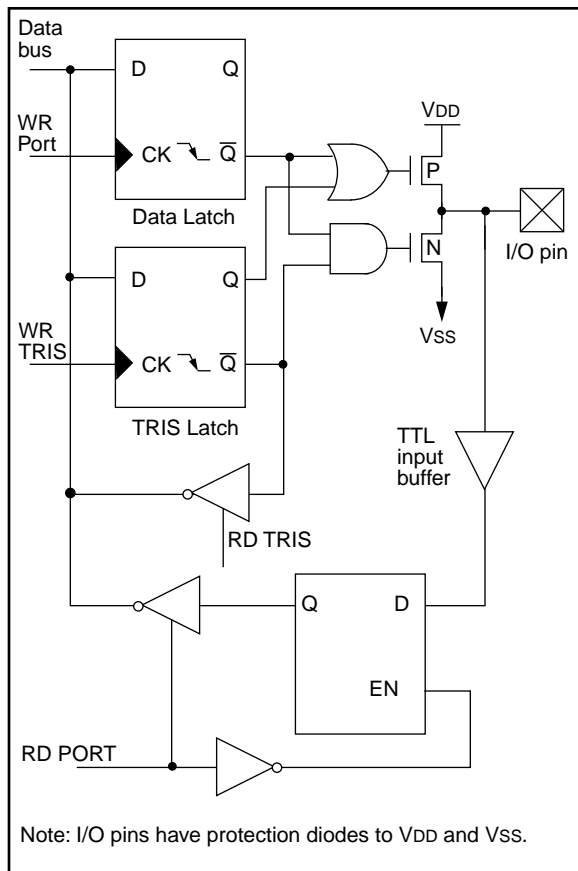
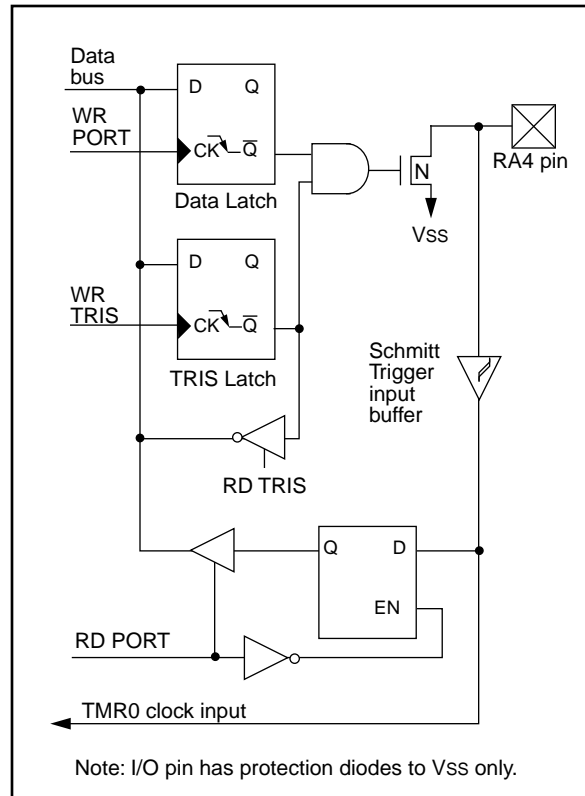**EXAMPLE 5-1:   INITIALIZING PORTA**

```
CLRF    PORTA        ; Initialize PORTA by
                     ; setting output
                     ; data latches
BSF     STATUS, RP0  ; Select Bank 1
MOVLW   0x0F         ; Value used to
                     ; initialize data
                     ; direction
MOVWF   TRISA        ; Set RA<3:0> as inputs
                     ; RA4 as outputs
                     ; TRISA<7:5> are always
                     ; read as '0'.
```

FIGURE 5-2:   BLOCK DIAGRAM OF PIN RA4

Note: I/O pin has protection diodes to VSS only.

Note: For the PIC16C84 Only:
For crystal oscillator configurations operating below 500 kHz, the device may generate a spurious internal Q-clock when PORTA<0> switches state. This does not occur with an external clock in RC mode. To avoid this, the RA0 pin should be kept static, i.e. in input/output mode, pin RA0 should not be toggled.

PIC16C8X

TABLE 5-1: PORTA FUNCTIONS

Name	Bit0	Buffer Type	Function
RA0	bit0	TTL	Input/output
RA1	bit1	TTL	Input/output
RA2	bit2	TTL	Input/output
RA3	bit3	TTL	Input/output
RA4/T0CKI	bit4	ST	Input/output or external clock input for TMR0. Output is open drain type.

Legend: TTL = TTL input, ST = Schmitt Trigger input

TABLE 5-2: SUMMARY OF REGISTERS 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
05h	PORTA	—	—	—	RA4/T0CKI	RA3	RA2	RA1	RA0	---x xxxx	---u uuuu
85h	TRISA	—	—	—	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	---1 1111	---1 1111

Legend: x = unknown, u = unchanged, – = unimplemented read as '0'. Shaded cells are unimplemented, read as '0'
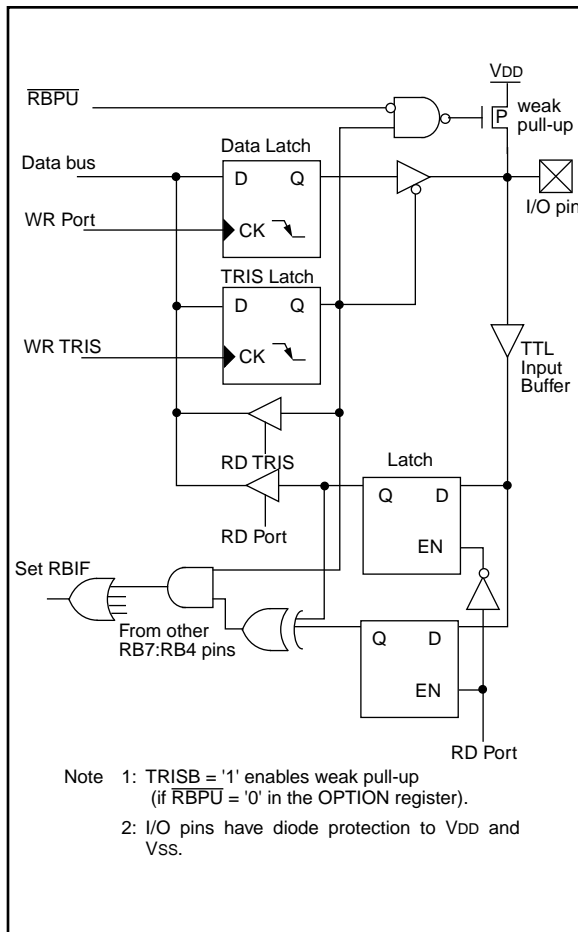
5.2 PORTB and TRISB Registers

PORTB is an 8-bit wide bi-directional port. The corresponding data direction register is TRISB. A '1' on any bit in the TRISB register puts the corresponding output driver in a hi-impedance mode. A '0' on any bit in the TRISB register puts the contents of the output latch on the selected pin(s).

Each of the PORTB pins have a weak internal pull-up. A single control bit can turn on all the pull-ups. This is done by clearing the $\overline{RBPU}$ (OPTION<7>) bit. The weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on a Power-on Reset.

Four of PORTB's pins, RB7:RB4, have an interrupt on change feature. Only pins configured as inputs can cause this interrupt to occur (i.e., any RB7:RB4 pin configured as an output is excluded from the interrupt on change comparison). The pins value in input mode are compared with the old value latched on the last read of PORTB. The "mismatch" outputs of the pins are OR'ed together to generate the RB port change interrupt.

FIGURE 5-3: BLOCK DIAGRAM OF PINS RB7:RB4

Note 1: TRISB = '1' enables weak pull-up
(if $\overline{RBPU}$ = '0' in the OPTION register).

2: I/O pins have diode protection to VDD and VSS.

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

a) Read (or write) PORTB. This will end the mismatch condition.
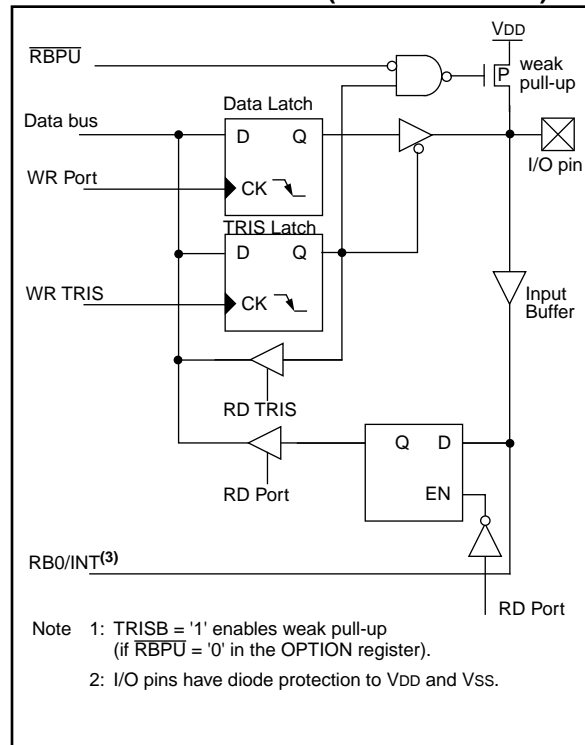
b) Clear flag bit RBIF.

A mismatch condition will continue to set the RBIF bit. Reading 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 these four pins allow easy interface to a key pad and make it possible for wake-up on key-depression (see AN552 in the Embedded Control Handbook).

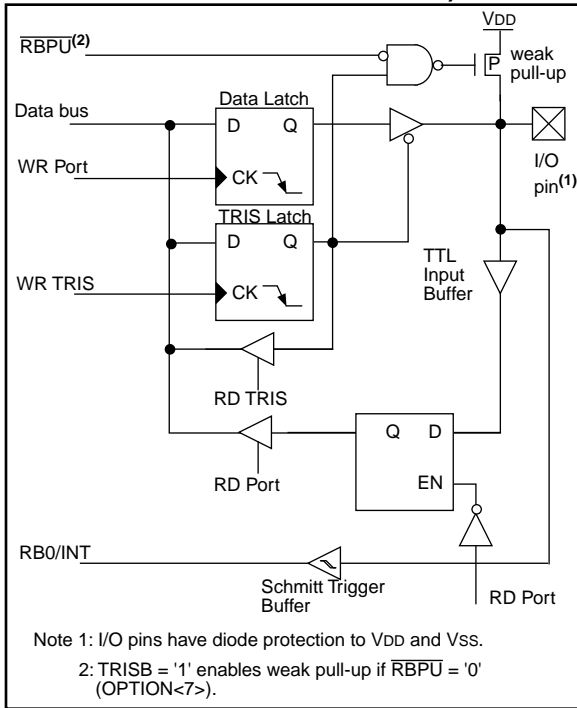| Note 1:	For the PIC16C84 Only; If a change on the I/O pin should occur when a read operation of PORTB is being executed (start of the Q2 cycle), the RBIF interrupt flag bit may not be set.
Note 2:	For all other PIC16C8X devices; For a change on the I/O pin to be recognized, the pulse width must be at least TCY wide.

The interrupt on change feature is recommended for wake-up on key depression operation and operations where PORTB is only used for the interrupt on change feature. Polling of PORTB is not recommended while using the interrupt on change feature.

FIGURE 5-4: BLOCK DIAGRAM OF PINS RB3:RB0 (PIC16C84 ONLY)

Note 1: TRISB = '1' enables weak pull-up
(if $\overline{RBPU}$ = '0' in the OPTION register).

2: I/O pins have diode protection to VDD and VSS.

PIC16C8X

FIGURE 5-5:   BLOCK DIAGRAM OF PINS
               RB3:RB0 (ALL OTHER
               PIC16C8X DEVICES)

Note 1: I/O pins have diode protection to VDD and VSS.
2: TRISB = '1' enables weak pull-up if $\overline{RBPU}$ = '0'
   (OPTION<7>).

EXAMPLE 5-2:   INITIALIZING PORTB

```
CLRF    PORTB        ; Initialize PORTB by
                     ; setting output
                     ; data latches
BSF     STATUS, RP0  ; Select Bank 1
MOVLW   0xCF         ; Value used to
                     ; initialize data
                     ; direction
MOVWF   TRISB        ; Set RB<3:0> as inputs
                     ; RB<5:4> as outputs
                     ; RB<7:6> as inputs
```

TABLE 5-3:   PORTB FUNCTIONS

Name	Bit	Buffer Type	I/O Consistency Function
RB0/INT	bit0	TTL/ST[1]	Input/output pin or external interrupt input. Internal software programmable weak pull-up.
RB1	bit1	TTL	Input/output pin. Internal software programmable weak pull-up.
RB2	bit2	TTL	Input/output pin. Internal software programmable weak pull-up.
RB3	bit3	TTL	Input/output pin. Internal software programmable weak pull-up.
RB4	bit4	TTL	Input/output pin (with interrupt on change). Internal software programmable weak pull-up.
RB5	bit5	TTL	Input/output pin (with interrupt on change). Internal software programmable weak pull-up.
RB6	bit6	TTL/ST[2]	Input/output pin (with interrupt on change). Internal software programmable weak pull-up. Serial programming clock.
RB7	bit7	TTL/ST[2]	Input/output pin (with interrupt on change). Internal software programmable weak pull-up. Serial programming data.

Legend: TTL = TTL input, ST = Schmitt Trigger.
Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt, except for the PIC16C84, which remains TTL.
2: This buffer is a Schmitt Trigger input when used in serial programming mode.

TABLE 5-4:   SUMMARY OF REGISTERS 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
06h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0/INT	xxxx xxxx	uuuu uuuu
86h	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	1111 1111	1111 1111
81h	OPTION	$\overline{RBPU}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111

Legend: x = unknown, u = unchanged. Shaded cells are not used by PORTB.

5.3   I/O Programming Considerations

5.3.1   BI-DIRECTIONAL I/O PORTS

Any instruction which writes, operates internally as a read followed by a write operation. The BCF and BSF instructions, for example, 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 (i.e., 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 rewritten 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 is unknown.

Reading the port register, 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 (i.e., BCF, BSF, etc.) on a port, the value of the port pins is read, the desired operation is done to this value, and this value is then written to the port latch.

A pin actively outputting a Low or High should not be driven from external devices at the same time in order to change the level on this pin ("wired-or", "wired-and"). The resulting high output current may damage the chip.

5.3.2   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 5-6). 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 that the pin voltage stabilizes (load dependent) before the next instruction which causes that file to be read into the CPU is executed. 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.

Example 5-3 shows the effect of two sequential read-modify-write instructions (e.g., BCF, BSF, etc.) on an I/O port.

EXAMPLE 5-3:   READ-MODIFY-WRITE INSTRUCTIONS ON AN I/O PORT

```
;Initial PORT settings: PORTB<7:4> Inputs
;                       PORTB<3:0> Outputs
;PORTB<7:6> have external pull-ups and are
;not connected to other circuitry
;
;                   PORT latch   PORT pins
;                   ----------   ---------
  BCF PORTB, 7    ; 01pp ppp     11pp ppp
  BCF PORTB, 6    ; 10pp ppp     11pp ppp
  BSF STATUS, RP0 ;
  BCF TRISB, 7    ; 10pp ppp     11pp ppp
  BCF TRISB, 6    ; 10pp ppp     10pp ppp
;
;Note that the user may have expected the
;pin values to be 00pp ppp. The 2nd BCF
;caused RB7 to be latched as the pin value
;(high).
```

FIGURE 5-6:   SUCCESSIVE I/O OPERATION
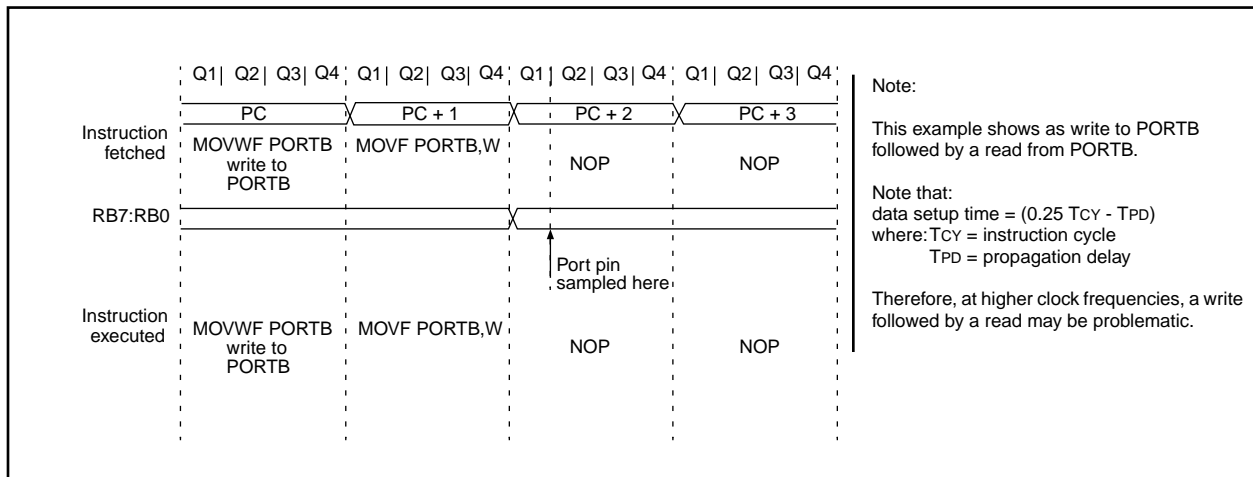
NOTES:

6.0 TIMER0 MODULE AND TMR0 REGISTER

The Timer0 module timer/counter has the following features:

- 8-bit timer/counter
- Readable and writable
- 8-bit software programmable prescaler
- Internal or external clock select
- Interrupt on overflow from FFh to 00h
- Edge select for external clock

Timer mode is selected by clearing the T0CS bit (OPTION<5>). In timer mode, the Timer0 module (Figure 6-1) will increment every instruction cycle (without prescaler). If the TMR0 register is written, the increment is inhibited for the following two cycles (Figure 6-2 and Figure 6-3). The user can work around this by writing an adjusted value to the TMR0 register.

Counter mode is selected by setting the T0CS bit (OPTION<5>). In this mode TMR0 will increment either on every rising or falling edge of pin RA4/T0CKI. The incrementing edge is determined by the T0 source edge select bit, T0SE (OPTION<4>). Clearing bit T0SE selects the rising edge. Restrictions on the external clock input are discussed in detail in Section 6.2.

The prescaler is shared between the Timer0 Module and the Watchdog Timer. The prescaler assignment is controlled, in software, by control bit PSA (OPTION<3>). Clearing bit PSA will assign the prescaler to the Timer0 Module. The prescaler is not readable or writable. When the prescaler (Section 6.3) is assigned to the Timer0 Module, the prescale value (1:2, 1:4, ..., 1:256) is software selectable.

6.1 TMR0 Interrupt

The TMR0 interrupt is generated when the TMR0 register overflows from FFh to 00h. This overflow sets the T0IF bit (INTCON<2>). The interrupt can be masked by clearing enable bit T0IE (INTCON<5>). The T0IF bit must be cleared in software by the Timer0 Module interrupt service routine before re-enabling this interrupt. The TMR0 interrupt (Figure 6-4) cannot wake the processor from SLEEP since the timer is shut off during SLEEP.
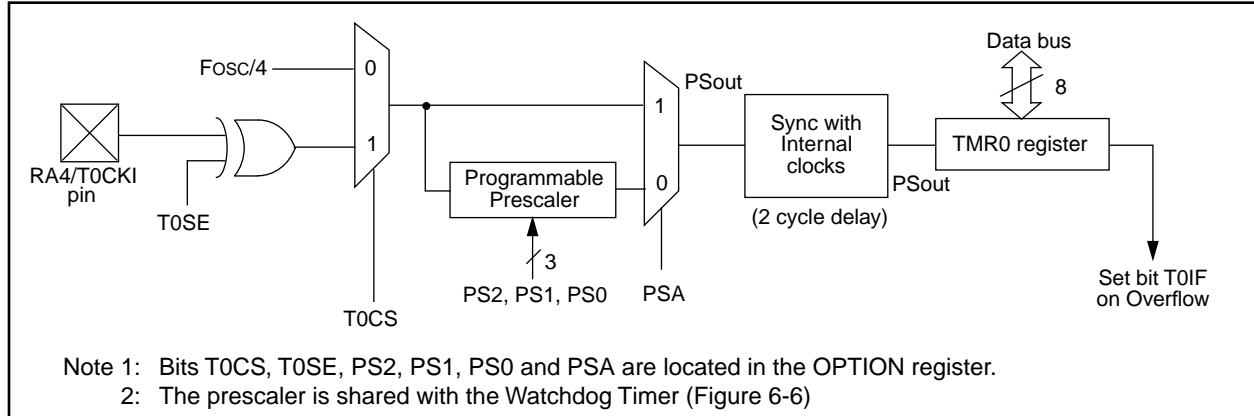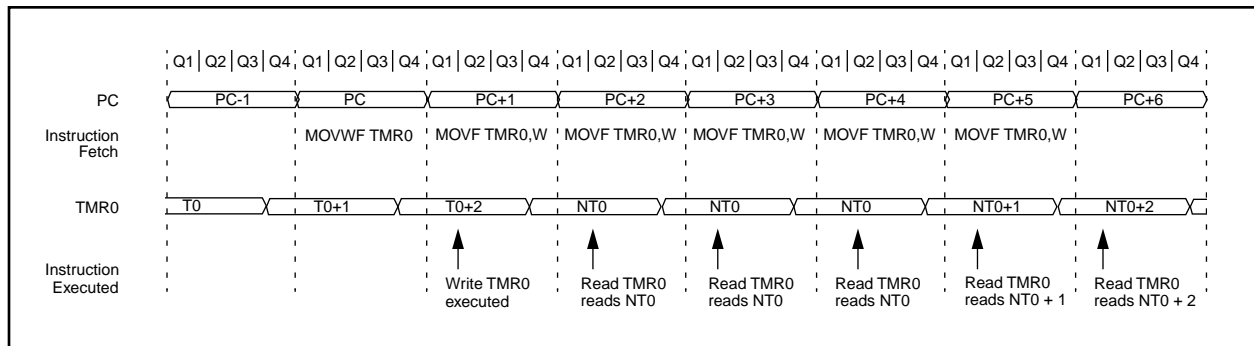
FIGURE 6-1: TMR0 BLOCK DIAGRAM

Note 1: Bits T0CS, T0SE, PS2, PS1, PS0 and PSA are located in the OPTION register.
2: The prescaler is shared with the Watchdog Timer (Figure 6-6)

FIGURE 6-2: TMR0 TIMING: INTERNAL CLOCK/NO PRESCALER

PIC16C8X

FIGURE 6-3: TMR0 TIMING: INTERNAL CLOCK/PRESCALE 1:2

FIGURE 6-4: TMR0 INTERRUPT TIMING

Note 1: T0IF interrupt flag is sampled here (every Q1).
     2: Interrupt latency = 3.25Tcy, where Tcy = instruction cycle time.
     3: CLKOUT is available only in RC oscillator mode.
     4: The timer clock (after the synchronizer circuit) which increments the timer from FFh to 00h immediately sets the T0IF bit.
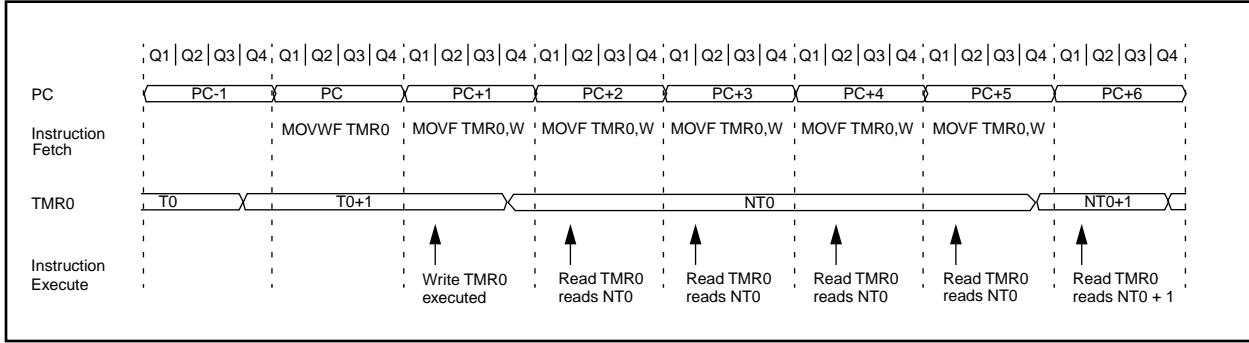   The TMR0 register will roll over 3 Tosc cycles later.
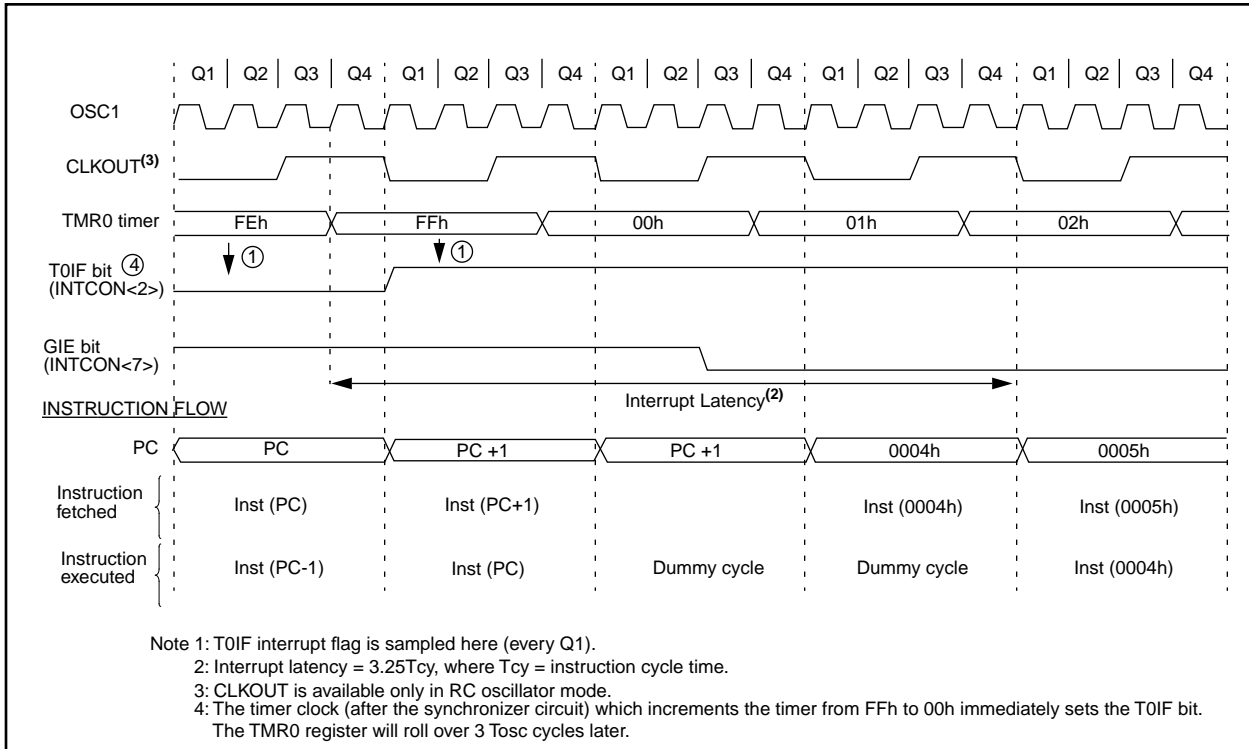
6.2   Using TMR0 with External Clock

When an external clock input is used for TMR0, it must meet certain requirements. The external clock requirement is due to internal phase clock (T$_{OSC}$) synchronization. Also, there is a delay in the actual incrementing of the TMR0 register after synchronization.

6.2.1   EXTERNAL CLOCK SYNCHRONIZATION

When no prescaler is used, the external clock input is the same as the prescaler output. The synchronization of pin RA4/T0CKI with the internal phase clocks is accomplished by sampling the prescaler output on the Q2 and Q4 cycles of the internal phase clocks (Figure 6-5). Therefore, it is necessary for T0CKI to be high for at least 2Tosc (plus a small RC delay) and low for at least 2Tosc (plus a small RC delay). Refer to the electrical specification of the desired device.

When a prescaler is used, the external clock input is divided by an asynchronous ripple counter type prescaler so that the prescaler output is symmetrical. For the external clock to meet the sampling requirement, the ripple counter must be taken into account. Therefore, it is necessary for T0CKI to have a period of at least 4Tosc (plus a small RC delay) divided by the prescaler value. The only requirement on T0CKI high and low time is that they do not violate the minimum pulse width requirement of 10 ns. Refer to parameters 40, 41 and 42 in the AC Electrical Specifications of the desired device.

6.2.2   TMR0 INCREMENT DELAY

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 the Timer0 Module is actually incremented. Figure 6-5 shows the delay from the external clock edge to the timer incrementing.
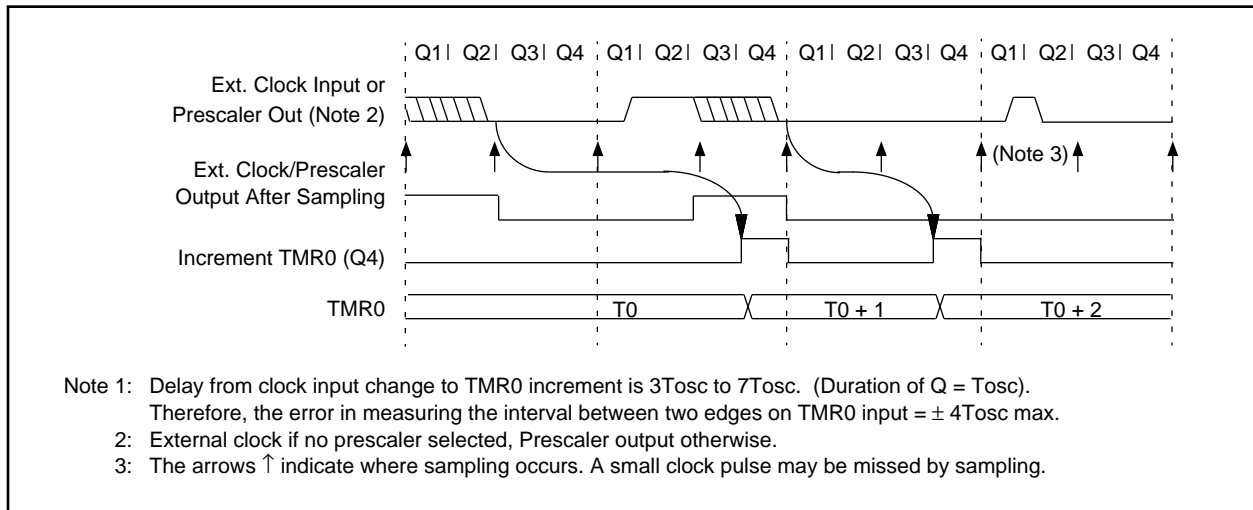
6.3   Prescaler

An 8-bit counter is available as a prescaler for the Timer0 Module, or as a postscaler for the Watchdog Timer (Figure 6-6). For simplicity, this counter is being referred to as "prescaler" throughout this data sheet. Note that there is only one prescaler available which is mutually exclusive between the Timer0 Module and the Watchdog Timer. Thus, a prescaler assignment for the Timer0 Module means that there is no prescaler for the Watchdog Timer, and vice-versa.

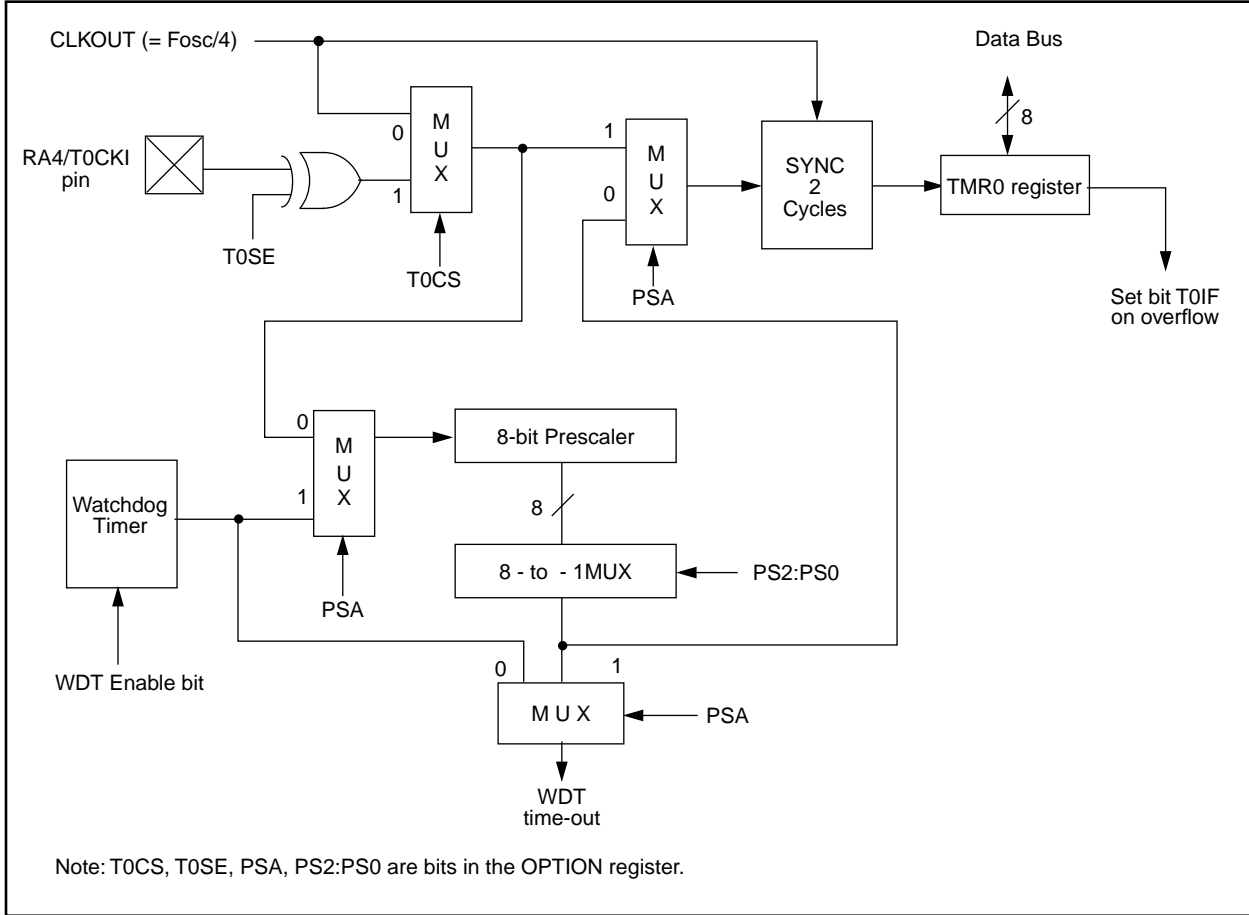The PSA and PS2:PS0 bits (OPTION<3:0>) determine the prescaler assignment and prescale ratio.

When assigned to the Timer0 Module, all instructions writing to the Timer0 Module (e.g., `CLRF 1`, `MOVWF 1`, `BSF  1,x` ....etc.) will clear the prescaler. When assigned to WDT, a `CLRWDT` instruction will clear the prescaler along with the Watchdog Timer. The prescaler is not readable or writable.

FIGURE 6-5:   TIMER0 TIMING WITH EXTERNAL CLOCK

Note 1: Delay from clock input change to TMR0 increment is 3Tosc to 7Tosc. (Duration of Q = Tosc).
        Therefore, the error in measuring the interval between two edges on TMR0 input = ± 4Tosc max.
     2: External clock if no prescaler selected, Prescaler output otherwise.
     3: The arrows ↑ indicate where sampling occurs. A small clock pulse may be missed by sampling.

PIC16C8X

FIGURE 6-6: **BLOCK DIAGRAM OF THE TMR0/WDT PRESCALER**

Note: T0CS, T0SE, PSA, PS2:PS0 are bits in the OPTION register.

PIC16C8X

NOTES:

7.0 DATA EEPROM MEMORY

The EEPROM data memory is readable and writable during normal operation (full V_{DD} range). This memory is not directly mapped in the register file space. Instead it is indirectly addressed through the Special Function Registers. There are four SFRs used to read and write this memory. These registers are:

- EECON1
- EECON2
- EEDATA
- EEADR

EEDATA holds the 8-bit data for read/write, and EEADR holds the address of the EEPROM location being accessed. PIC16C8X devices have 64 bytes of data EEPROM with an address range from 0h to 3Fh.

The EEPROM data memory allows byte read and write. A byte write automatically erases the location and writes the new data (erase before write). The EEPROM data memory is rated for high erase/write cycles. The write time is controlled by an on-chip timer. The write-time will vary with voltage and temperature as well as from chip to chip. Please refer to AC specifications for exact limits.

When the device is code protected, the CPU may continue to read and write the data EEPROM memory. The device programmer can no longer access this memory.

7.1 EEADR

The EEADR register can address up to a maximum of 256 bytes of data EEPROM. Only the first 64 bytes of data EEPROM are implemented and only six of the eight bits in the register (EEADR<5:0>) are required.

The upper two bits are address decoded. This means that these two bits should always be '0' to ensure that the address is in the 64 byte memory space.

FIGURE 7-1: EECON1 REGISTER (ADDRESS 88h)

U	U	U	R/W-0	R/W-x	R/W-0	R/S-0	R/S-x
—	—	—	EEIF	WRERR	WREN	WR	RD

bit7   bit0

R = Readable bit
W = Writable bit
S = Settable bit
U = Unimplemented bit, read as '0'
- n = Value at POR reset

bit 7:5   **Unimplemented:** Read as '0'

bit 4   **EEIF**: EEPROM Write Operation Interrupt Flag bit
1 = The write operation completed (must be cleared in software)
0 = The write operation is not complete or has not been started

bit 3   **WRERR**: EEPROM Error Flag bit
1 = A write operation is prematurely terminated
(any $\overline{MCLR}$ reset or any WDT reset during normal operation)
0 = The write operation completed

bit 2   **WREN**: EEPROM Write Enable bit
1 = Allows write cycles
0 = Inhibits write to the data EEPROM

bit 1   **WR**: Write Control bit
1 = initiates a write cycle. (The bit is cleared by hardware once write is complete. The WR bit can only be set (not cleared) in software.
0 = Write cycle to the data EEPROM is complete

bit 0   **RD**: Read Control bit
1 = Initiates an EEPROM read (read takes one cycle. RD is cleared in hardware. The RD bit can only be set (not cleared) in software).
0 = Does not initiate an EEPROM read

PIC16C8X

7.2   EECON1 and EECON2 Registers

EECON1 is the control register with five low order bits physically implemented. The upper-three bits are non-existent and read as '0's.

Control bits RD and WR initiate read and write, respectively. These bits cannot be cleared, only set, in software. They are cleared in hardware at completion of the read or write operation. The inability to clear the WR bit in software prevents the accidental, premature termination of a write operation.

The WREN bit, when set, will allow a write operation. On power-up, the WREN bit is clear. The WRERR bit is set when a write operation is interrupted by a $\overline{\text{MCLR}}$ reset or a WDT time-out reset during normal operation. In these situations, following reset, the user can check the WRERR bit and rewrite the location. The data and address will be unchanged in the EEDATA and EEADR registers.

Interrupt flag bit EEIF is set when write is complete. It must be cleared in software.

EECON2 is not a physical register. Reading EECON2 will read all '0's. The EECON2 register is used exclusively in the Data EEPROM write sequence.

7.3   Reading the EEPROM Data Memory

To read a data memory location, the user must write the address to the EEADR register and then set control bit RD (EECON1<0>). The data is available, in the very next cycle, in the EEDATA register; therefore it can be read in the next instruction. EEDATA will hold this value until another read or until it is written to by the user (during a write operation).

EXAMPLE 7-1:   DATA EEPROM READ

```
BCF     STATUS, RP0   ; Bank 0
MOVLW   CONFIG_ADDR   ;
MOVWF   EEADR         ; Address to read
BSF     STATUS, RP0   ; Bank 1
BSF     EECON1, RD    ; EE Read
BCF     STATUS, RP0   ; Bank 0
MOVF    EEDATA, W     ; W = EEDATA
```

7.4   Writing to the EEPROM Data Memory

To write an EEPROM data location, the user must first write the address to the EEADR register and the data to the EEDATA register. Then the user must follow a specific sequence to initiate the write for each byte.

EXAMPLE 7-2:   DATA EEPROM WRITE

```
        BSF     STATUS, RP0   ; Bank 1
        BCF     INTCON, GIE   ; Disable INTs.
        MOVLW   55h           ;
        MOVWF   EECON2        ; Write 55h
        MOVLW   AAh           ;
        MOVWF   EECON2        ; Write AAh
        BSF     EECON1,WR     ; Set WR bit
                              ;   begin write
        BSF     INTCON, GIE   ; Enable INTs.
```
Required Sequence (brace spanning MOVLW 55h through BSF EECON1,WR)

The write will not initiate if the above sequence is not exactly followed (write 55h to EECON2, write AAh to EECON2, then set WR bit) for each byte. We strongly recommend that interrupts be disabled during this code segment.

Additionally, the WREN bit in EECON1 must be set to enable write. This mechanism prevents accidental writes to data EEPROM due to errant (unexpected) code execution (i.e., lost programs). The user should keep the WREN bit clear at all times, except when updating EEPROM. The WREN bit is not cleared by hardware

After a write sequence has been initiated, clearing the WREN bit will not affect this write cycle. The WR bit will be inhibited from being set unless the WREN bit is set.

At the completion of the write cycle, the WR bit is cleared in hardware and the EE Write Complete Interrupt Flag bit (EEIF) is set. The user can either enable this interrupt or poll this bit. EEIF must be cleared by software.

> **Note:**   For the PIC16C84 Only;
> The data EEPROM memory E/W cycle time may occasionally exceed the 10 ms specification (typical). To ensure that the write cycle is complete, use the EE interrupt or poll the WR bit (EECON1<1>). Both these events signify the completion of the write cycle.

7.5   Write Verify

Depending on the application, good programming practice may dictate that the value written to the Data EEPROM should be verified (Example 7-3) to the desired value to be written. This should be used in applications where an EEPROM bit will be stressed near the specification limit. The Total Endurance disk will help determine your comfort level.

Generally the EEPROM write failure will be a bit which was written as a '1', but reads back as a '0' (due to leakage off the bit).

EXAMPLE 7-3:   WRITE VERIFY

```
    BCF    STATUS, RP0 ; Bank 0
    :                  ; Any code can go here
    :                  ;
    MOVF   EEDATA, W   ; Must be in Bank 0
    BSF    STATUS, RP0 ; Bank 1
READ
    BSF    EECON1, RD  ; YES, Read the
                       ;   value written
    BCF    STATUS, RP0 ; Bank 0
;
; Is the value written (in W reg) and
;   read (in EEDATA) the same?
;
    SUBWF  EEDATA, W   ;
    BTFSS  STATUS, Z   ; Is difference 0?
    GOTO   WRITE_ERR   ; NO, Write error
    :                  ; YES, Good write
    :                  ; Continue program
```

7.6   Protection Against Spurious Write

There are conditions when the device may not want to write to the data EEPROM memory. To protect against spurious EEPROM writes, various mechanisms have been built in. On power-up, WREN is cleared. Also, the Power-up Timer (72 ms duration) prevents EEPROM write.

The write initiate sequence and the WREN bit together help prevent an accidental write during brown-out, power glitch, or software malfunction.

7.7   Data EEPROM Operation during Code Protect

When the device is code protected, the CPU is able to read and write unscrambled data to the Data EEPROM.

For ROM devices, there are two code protection bits (Section 8.9). One for the ROM program memory and one for the Data EEPROM memory.

7.8   Power Consumption Considerations

Note:	For the PIC16C84 Only; It is recommended that the EEADR<7:6> bits be cleared. When either of these bits is set, the maximum IDD for the device is higher than when both are cleared. The specification is 400 µA. With EEADR<7:6> cleared, the maximum is approximately 150 µA.

TABLE 7-1:   REGISTERS/BITS ASSOCIATED WITH DATA EEPROM

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
08h	EEDATA	EEPROM data register								xxxx xxxx	uuuu uuuu
09h	EEADR	EEPROM address register								xxxx xxxx	uuuu uuuu
88h	EECON1	—	—	—	EEIF	WRERR	WREN	WR	RD	---0 x000	---0 q000
89h	EECON2	EEPROM control register 2								---- ----	---- ----

Legend: x = unknown, u = unchanged, – = unimplemented read as '0', q = value depends upon condition. Shaded cells are not used by Data EEPROM.

NOTES:

8.0 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 PIC16C8X 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 features are:

• OSC selection
• Reset
  - Power-on Reset (POR)
  - Power-up Timer (PWRT)
  - Oscillator Start-up Timer (OST)
• Interrupts
• Watchdog Timer (WDT)
• SLEEP
• Code protection
• ID locations
• In-circuit serial programming

The PIC16C8X has a Watchdog Timer which can be shut off only through configuration 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 72 ms (nominal) on power-up only. This design keeps the device in reset while the power supply stabilizes. With these two timers on-chip, most applications need no external reset circuitry.
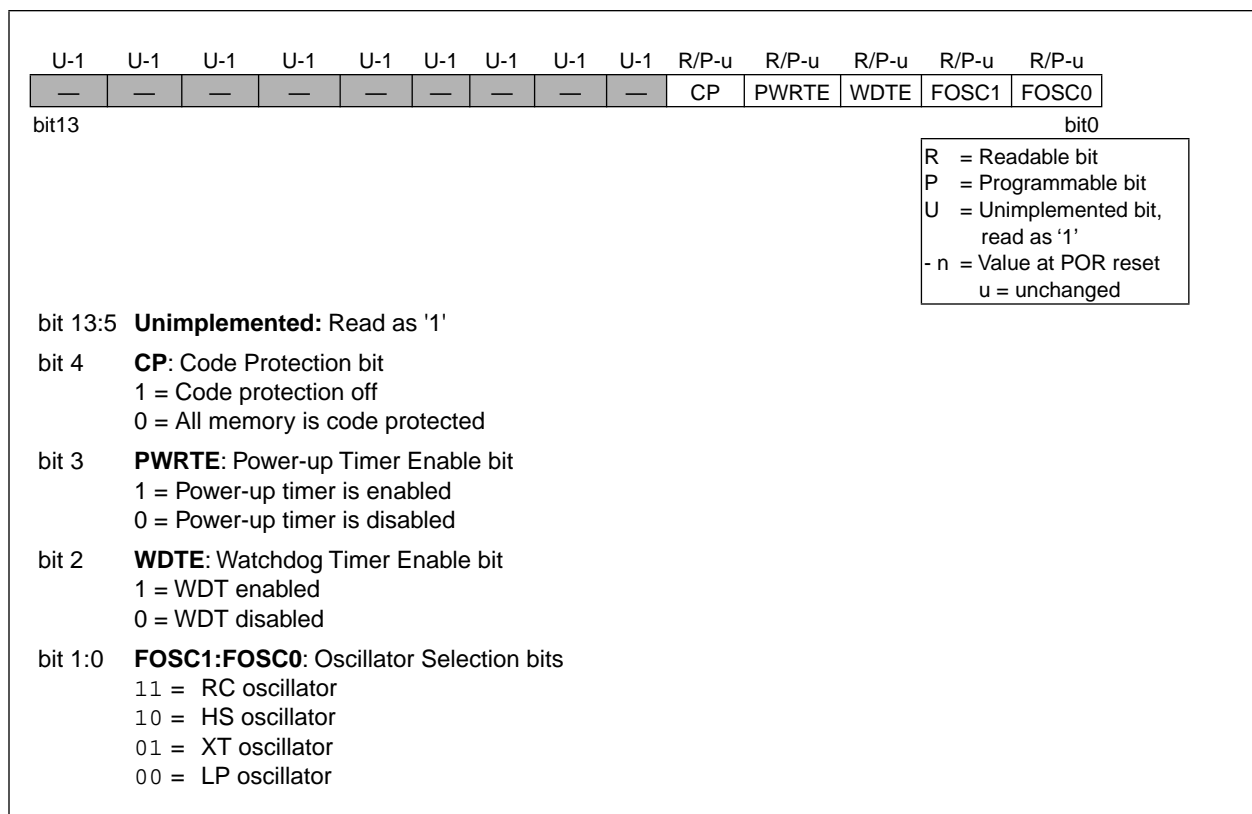
SLEEP mode offers a very low current power-down mode. The user can wake-up from SLEEP through external reset, Watchdog Timer time-out or through an interrupt. Several oscillator options are provided to allow the part to fit the application. The RC oscillator option saves system cost while the LP crystal option saves power. A set of configuration bits are used to select the various options.

8.1 Configuration Bits

The configuration bits can be programmed (read as '0') or left unprogrammed (read as '1') to select various device configurations. These bits are mapped in program memory location 2007h.

Address 2007h is beyond the user program memory space and it belongs to the special test/configuration memory space (2000h - 3FFFh). This space can only be accessed during programming.

FIGURE 8-1: CONFIGURATION WORD - PIC16C84

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u
—	—	—	—	—	—	—	—	—	CP	PWRTE	WDTE	FOSC1	FOSC0

bit13   bit0

```
R   = Readable bit
P   = Programmable bit
U   = Unimplemented bit,
        read as '1'
- n = Value at POR reset
        u = unchanged
```

bit 13:5   **Unimplemented:** Read as '1'

bit 4   **CP**: Code Protection bit
         1 = Code protection off
         0 = All memory is code protected

bit 3   **PWRTE**: Power-up Timer Enable bit
         1 = Power-up timer is enabled
         0 = Power-up timer is disabled

bit 2   **WDTE**: Watchdog Timer Enable bit
         1 = WDT enabled
         0 = WDT disabled

bit 1:0   **FOSC1:FOSC0**: Oscillator Selection bits
         11 = RC oscillator
         10 = HS oscillator
         01 = XT oscillator
         00 = LP oscillator

PIC16C8X

FIGURE 8-2: CONFIGURATION WORD - PIC16CR83 AND PIC16CR84

R-u	R-u	R-u	R-u	R-u	R-u	R/P-u	R-u	R-u	R-u	R-u	R-u	R-u	R-u
CP	CP	CP	CP	CP	CP	DP	CP	CP	CP	$\overline{PWRTE}$	WDTE	FOSC1	FOSC0

bit13   bit0

R = Readable bit
P = Programmable bit
- n = Value at POR reset
u = unchanged

bit 13:8 **CP**: Program Memory Code Protection bit
   1 = Code protection off
   0 = Program memory is code protected

bit 7 **DP**: Data Memory Code Protection bit
   1 = Code protection off
   0 = Data memory is code protected

bit 6:4 **CP**: Program Memory Code Protection bit
   1 = Code protection off
   0 = Program memory is code protected

bit 3 **$\overline{PWRTE}$**: Power-up Timer Enable bit
   1 = Power-up timer is disabled
   0 = Power-up timer is enabled

bit 2 **WDTE**: Watchdog Timer Enable bit
   1 = WDT enabled
   0 = WDT disabled

bit 1:0 **FOSC1:FOSC0**: Oscillator Selection bits
   11 = RC oscillator
   10 = HS oscillator
   01 = XT oscillator
   00 = LP oscillator

FIGURE 8-3: CONFIGURATION WORD - PIC16C83 AND PIC16C84A

R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u
CP	CP	CP	CP	CP	CP	CP	CP	CP	CP	$\overline{PWRTE}$	WDTE	FOSC1	FOSC0

bit13   bit0

R = Readable bit
P = Programmable bit
- n = Value at POR reset
   u = unchanged

bit 13:4 **CP**: Code Protection bit
   1 = Code protection off
   0 = Program memory is code protected

bit 3 **$\overline{PWRTE}$**: Power-up Timer Enable bit
   1 = Power-up timer is disabled
   0 = Power-up timer is enabled

bit 2 **WDTE**: Watchdog Timer Enable bit
   1 = WDT enabled
   0 = WDT disabled

bit 1:0 **FOSC1:FOSC0**: Oscillator Selection bits
   11 = RC oscillator
   10 = HS oscillator
   01 = XT oscillator
   00 = LP oscillator

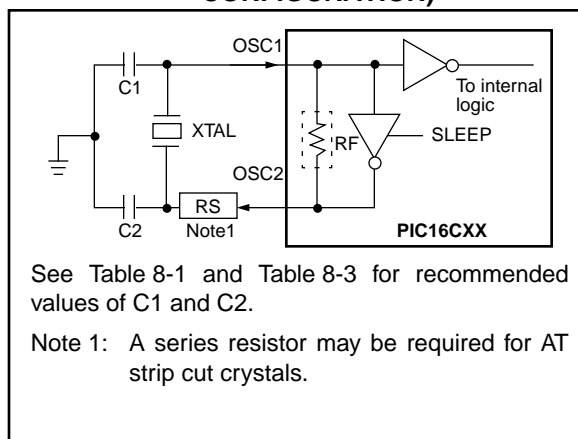## 8.2   Oscillator Configurations

8.2.1   OSCILLATOR TYPES

The PIC16C8X can be operated in four different oscillator modes. The user can program two configuration bits (FOSC1 and FOSC0) to select one of these four modes:

- LP   Low Power Crystal
- XT   Crystal/Resonator
- HS   High Speed Crystal/Resonator
- RC   Resistor/Capacitor

8.2.2   CRYSTAL OSCILLATOR / CERAMIC RESONATORS

In XT, LP or HS modes a crystal or ceramic resonator is connected to the OSC1/CLKIN and OSC2/CLKOUT pins to establish oscillation (Figure 8-4).

FIGURE 8-4:   CRYSTAL/CERAMIC RESONATOR OPERATION (HS, XT OR LP OSC CONFIGURATION)

See Table 8-1 and Table 8-3 for recommended values of C1 and C2.

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

The PIC16C8X 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. When in XT, LP or HS modes, the device can have an external clock source to drive the OSC1/CLKIN pin (Figure 8-5).

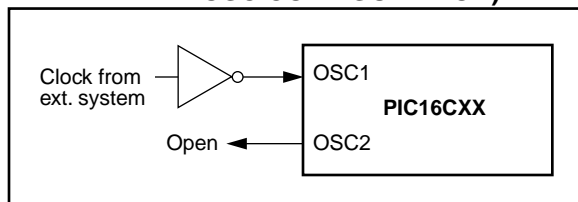### FIGURE 8-5:   EXTERNAL CLOCK INPUT OPERATION (HS, XT OR LP OSC CONFIGURATION)

TABLE 8-1:   PIC16C84 CAPACITOR SELECTION FOR CERAMIC RESONATORS

Ranges Tested:			
Mode	Freq	OSC1/C1	OSC2/C2
XT	455 kHz	47 - 100 pF	47 - 100 pF
	2.0 MHz	15 - 33 pF	15 - 33 pF
	4.0 MHz	15 - 33 pF	15 - 33 pF
HS	8.0 MHz	15 - 33 pF	15 - 33 pF
	10.0 MHz	15 - 33 pF	15 - 33 pF

Note: Recommended values of C1 and C2 are identical to the ranges tested table.

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 the appropriate values of external components.

Resonators Tested:		
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%
10.0 MHz	Murata Erie CSA10.00MTZ	± 0.5%
None of the resonators had built-in capacitors.		

TABLE 8-2:   PIC16C83/R83/84A/R84 CAPACITOR SELECTION FOR CERAMIC RESONATORS

Ranges Tested:			
Mode	Freq	OSC1/C1	OSC2/C2
XT	455 kHz	47 - 100 pF	47 - 100 pF
	2.0 MHz	15 - 33 pF	15 - 33 pF
	4.0 MHz	15 - 33 pF	15 - 33 pF
HS	8.0 MHz	15 - 33 pF	15 - 33 pF
	10.0 MHz	15 - 33 pF	15 - 33 pF

Note: Recommended values of C1 and C2 are identical to the ranges tested table.

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 the appropriate values of external components.

Resonators Tested:		
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%
10.0 MHz	Murata Erie CSA10.00MTZ	± 0.5%
None of the resonators had built-in capacitors.		

PIC16C8X

TABLE 8-3: PIC16C84 CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR

Mode	Freq	OSC1/C1	OSC2/C2
LP	32 kHz 200 kHz	68 - 100 pF 15 - 33 pF	68 - 100 pF 15 - 33 pF
XT	100 kHz 2 MHz 4 MHz	100 - 150 pF 15 - 33 pF 15 - 33 pF	100 - 150 pF 15 - 33 pF 15 - 33 pF
HS	4 MHz 10 MHz	15 - 33 pF 15 - 33 pF	15 - 33 pF 15 - 33 pF

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

For $V_{DD} > 4.5V$, C1 = C2 $\approx$ 30 pF is recommended.

Crystals Tested:

32.768 kHz	Epson C-001R32.768K-A	$\pm$ 20 PPM
100 kHz	Epson C-2 100.00 KC-P	$\pm$ 20 PPM
200 kHz	STD XTL 200.000 KHz	$\pm$ 20 PPM
1.0 MHz	ECS ECS-10-13-2	$\pm$ 50 PPM
2.0 MHz	ECS ECS-20-S-2	$\pm$ 50 PPM
4.0 MHz	ECS ECS-40-S-4	$\pm$ 50 PPM
10.0 MHz	ECS ECS-100-S-4	$\pm$ 50 PPM

TABLE 8-4: PIC16C83/R83/84A/R84 CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR

Mode	Freq	OSC1/C1	OSC2/C2
LP	32 kHz 200 kHz	68 - 100 pF 15 - 33 pF	68 - 100 pF 15 - 33 pF
XT	100 kHz 2 MHz 4 MHz	100 - 150 pF 15 - 33 pF 15 - 33 pF	100 - 150 pF 15 - 33 pF 15 - 33 pF
HS	4 MHz 10 MHz	15 - 33 pF 15 - 33 pF	15 - 33 pF 15 - 33 pF

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

For $V_{DD} > 4.5V$, C1 = C2 $\approx$ 30 pF is recommended.

Crystals Tested:

32.768 kHz	Epson C-001R32.768K-A	$\pm$ 20 PPM
100 kHz	Epson C-2 100.00 KC-P	$\pm$ 20 PPM
200 kHz	STD XTL 200.000 KHz	$\pm$ 20 PPM
1.0 MHz	ECS ECS-10-13-2	$\pm$ 50 PPM
2.0 MHz	ECS ECS-20-S-2	$\pm$ 50 PPM
4.0 MHz	ECS ECS-40-S-4	$\pm$ 50 PPM
10.0 MHz	ECS ECS-100-S-4	$\pm$ 50 PPM

8.2.3   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 are available; one with series resonance, and one with parallel resonance.

Figure 8-6 shows 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 kΩ resistor provides negative feedback for stability. The 10 kΩ potentiometer biases the 74AS04 in the linear region. This could be used for external oscillator designs.

FIGURE 8-6:   EXTERNAL PARALLEL RESONANT CRYSTAL OSCILLATOR CIRCUIT
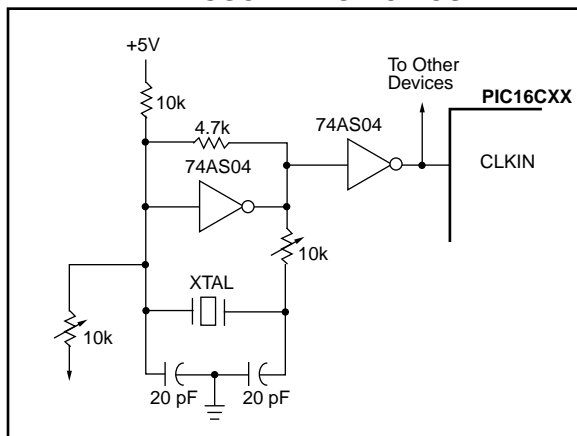
Figure 8-7 shows a series resonant oscillator circuit. This circuit is also designed to use the fundamental frequency of the crystal. The inverter performs a 180-degree phase shift. The 330 kΩ resistors provide the negative feedback to bias the inverters in their linear region.

FIGURE 8-7:   EXTERNAL SERIES RESONANT CRYSTAL OSCILLATOR CIRCUIT
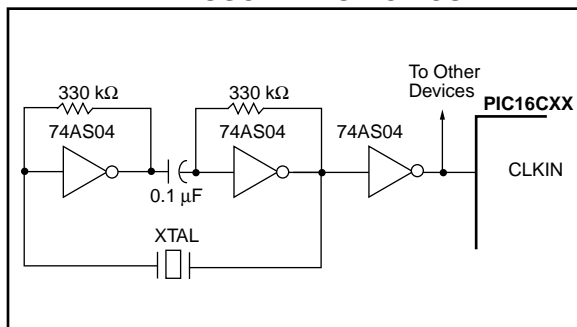
8.2.4   RC OSCILLATOR

For timing insensitive applications the RC device option offers additional cost savings. The RC oscillator frequency is a function of the supply voltage, the resistor (Rext) values, capacitor (Cext) values, and the operating temperature. In addition to this, the oscillator frequency will vary from unit to unit due to normal process parameter variation. Furthermore, the difference in lead frame capacitance between package types also affects the oscillation frequency, especially for low Cext values. The user needs to take into account variation due to tolerance of the external R and C components. Figure 8-8 shows how an R/C combination is connected to the PIC16C8X. For Rext values below 2.2 kΩ, the oscillator operation may become unstable, or stop completely. For very high Rext values (e.g., 1 MΩ), the oscillator becomes sensitive to noise, humidity and leakage. Thus, we recommend keeping Rext between 3 kΩ and 100 kΩ.

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, the oscillation frequency can vary dramatically due to changes in external capacitances, such as PCB trace capacitance or package lead frame capacitance.

See the electrical specification section 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 has a greater affect on RC frequency).

See the electrical specification section for variation of oscillator frequency due to V$_{DD}$ for given Rext/Cext values as well as frequency variation due to operating temperature.

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 8-8:   RC OSCILLATOR MODE
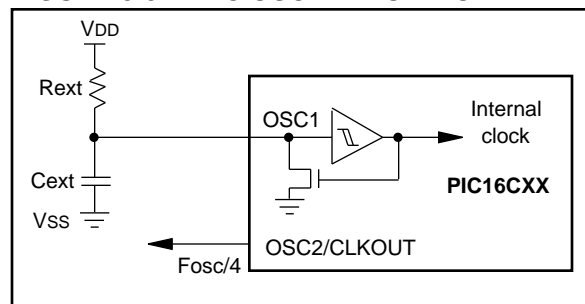
Note:	When the device oscillator is in RC mode, do not drive the OSC1 pin with an external clock or you may damage the device.

PIC16C8X

8.3 Reset

The PIC16C8X differentiates between various kinds of reset:

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

Some registers are not affected in any reset condition; their status is unknown on a POR reset and unchanged in any other reset. Most other registers are reset to a "reset state" on POR, $\overline{MCLR}$ or WDT reset during normal operation and on $\overline{MCLR}$ reset during SLEEP. They are not affected by a WDT reset during SLEEP, since this reset is viewed as the resumption of normal operation. The $\overline{TO}$ and $\overline{PD}$ bits are set or cleared differently in different reset situations (Table 8-6). These bits are used in software to determine the nature of the reset. Table 8-8 gives a full description of reset states for all registers.

Figure 8-9 shows a simplified block diagram of the on-chip reset circuit.

For all devices, except the PIC16C84, the $\overline{MCLR}$ reset path has a noise filter to ignore small pulses. The electrical specifications specifies the pulse width requirements for the $\overline{MCLR}$ pin.

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

### 8.4.1 POWER-ON RESET (POR)

A Power-on Reset pulse is generated on-chip when VDD rise is detected (in the range of 1.2V - 1.7V). To take advantage of the POR, just tie the $\overline{MCLR}$ 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 must be met for this to operate properly. See Electrical Specifications for details.

When the device starts normal operation (exits the reset condition), device operating parameters (voltage, frequency, temperature, ...) must be meet to ensure operation. If these conditions are not met, the device must be held in reset until the operating conditions are met.

For additional information, refer to Application Note AN607, "*Power-up Trouble Shooting.*"

The POR circuit does not produce an internal reset when VDD declines.

FIGURE 8-9: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT

Note 1: This is a separate oscillator from the RC oscillator of the CLKIN pin.

See Table 8-5
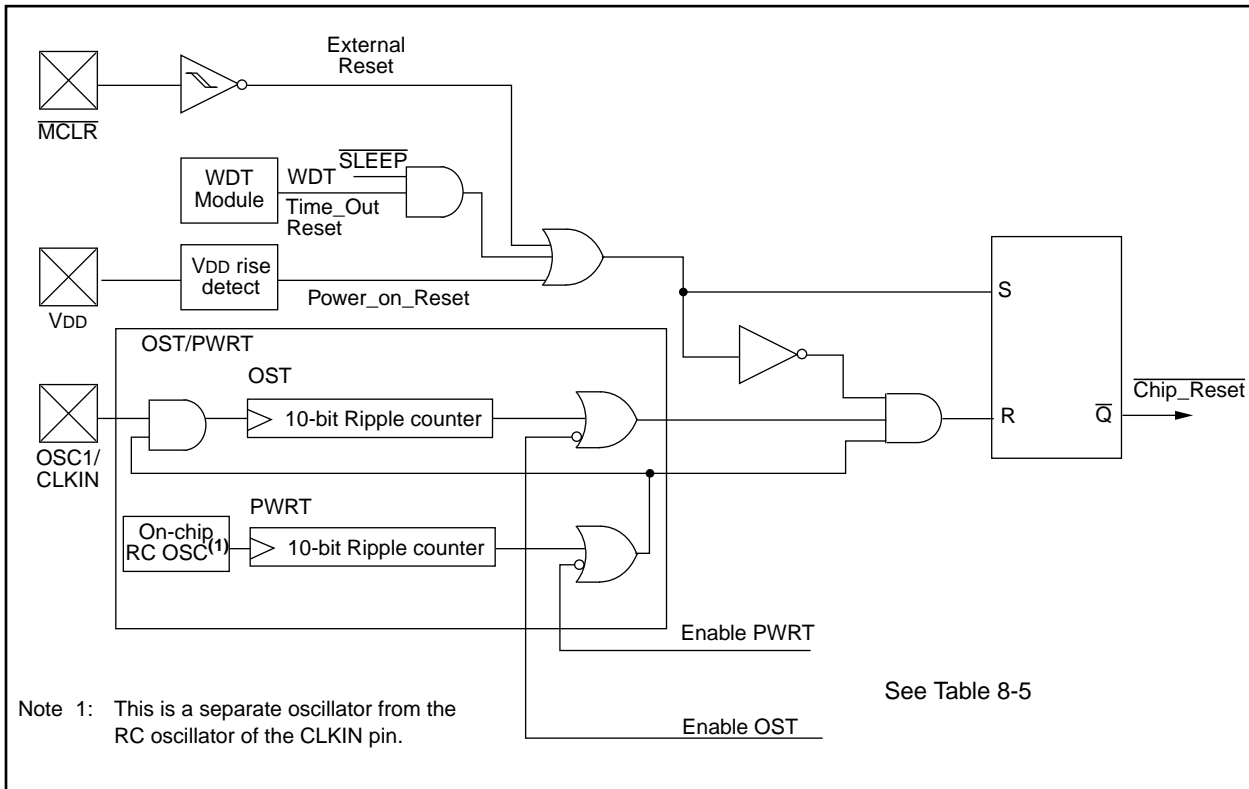
8.4.2 POWER-UP TIMER (PWRT)

The Power-up Timer provides a fixed 72 ms nominal time-out on power-up only, from POR. The Power-up Timer operates on an internal RC oscillator. The chip is kept in reset as long as the PWRT is active. The PWRT delay allows the V$_{DD}$ to rise to an acceptable level. A configuration bit, PWRTE, can enable/disable the PWRT (see Figure 8-1, Figure 8-2, and Figure 8-3 for the operation of the PWRTE bit for a particular device).

The power-up time delay will vary from chip to chip due to V$_{DD}$, temperature, and process variation. See DC parameters for details.

8.4.3 OSCILLATOR START-UP TIMER (OST)

The Oscillator Start-up Timer (OST) provides a 1024 oscillator cycle delay (from OSC1 input) after the PWRT delay ends. This ensures the crystal oscillator or resonator has started and stabilized.

The OST time-out is invoked only for XT, LP and HS modes and only on Power-on Reset or wake-up from SLEEP.

8.4.4 TIME-OUT SEQUENCE

On power-up (Figure 8-10, Figure 8-11, and Figure 8-12) the time-out sequence is as follows: First PWRT time-out is invoked after a POR has expired. Then the OST is activated. The total time-out will vary based on oscillator configuration and PWRTE configuration bit status. For example, in RC mode with the PWRT disabled, there will be no time-out at all.

TABLE 8-5: TIME-OUT IN VARIOUS SITUATIONS

Oscillator Configuration	Power-up		Wake-up from SLEEP
	PWRT Enabled	PWRT Disabled	
XT, HS, LP	72 ms + 1024T$_{OSC}$	1024T$_{OSC}$	1024T$_{OSC}$
RC	72 ms	—	—

Since the time-outs occur from the POR reset pulse, if $\overline{MCLR}$ is kept low long enough, the time-outs will expire. Then bringing $\overline{MCLR}$ high, execution will begin immediately (Figure 8-11). This is useful for testing purposes or to synchronize more than one PIC16CXX device when operating in parallel.

Table 8-6 shows the significance of the $\overline{TO}$ and $\overline{PD}$ bits. Table 8-7 lists the reset conditions for some special registers, while Table 8-8 lists the reset conditions for all the registers.

TABLE 8-6: STATUS BITS AND THEIR SIGNIFICANCE

$\overline{TO}$	$\overline{PD}$	Condition
1	1	Power-on Reset
0	x	Illegal, $\overline{TO}$ is set on POR
x	0	Illegal, $\overline{PD}$ is set on POR
0	1	WDT Reset (during normal operation)
0	0	WDT Wake-up
1	1	$\overline{MCLR}$ Reset during normal operation
1	0	$\overline{MCLR}$ Reset during SLEEP or interrupt wake-up from SLEEP

TABLE 8-7: RESET CONDITION FOR PROGRAM COUNTER AND THE STATUS REGISTER

Condition	Program Counter	STATUS Register
Power-on Reset	000h	0001 1xxx
$\overline{MCLR}$ Reset during normal operation	000h	0001 1uuu
$\overline{MCLR}$ Reset during SLEEP	000h	0001 0uuu
WDT Reset (during normal operation)	000h	0000 1uuu
WDT Wake-up	PC + 1	uuu0 0uuu
Interrupt wake-up from SLEEP	PC + 1 [1]	uuu1 0uuu

Legend: u = unchanged, x = unknown.
Note 1: When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).

PIC16C8X

TABLE 8-8: INITIALIZATION CONDITIONS FOR ALL REGISTERS

Register	Address	Power-on Reset	MCLR Reset during: – normal operation – SLEEP WDT Reset during normal operation	Wake-up from SLEEP: – through interrupt – through WDT time-out
W	—	xxxx xxxx	uuuu uuuu	uuuu uuuu
INDF	00h	---- ----	---- ----	---- ----
TMR0	01h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCL	02h	0000h	0000h	PC + 1[2]
STATUS	03h	0001 1xxx	000q quuu[3]	uuuq quuu[3]
FSR	04h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTA	05h	---x xxxx	---u uuuu	---u uuuu
PORTB	06h	xxxx xxxx	uuuu uuuu	uuuu uuuu
EEDATA	08h	xxxx xxxx	uuuu uuuu	uuuu uuuu
EEADR	09h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCLATH	0Ah	---0 0000	---0 0000	---u uuuu
INTCON	0Bh	0000 000x	0000 000u	uuuu uuuu[1]
INDF	80h	---- ----	---- ----	---- ----
OPTION	81h	1111 1111	1111 1111	uuuu uuuu
PCL	82h	0000h	0000h	PC + 1
STATUS	83h	0001 1xxx	000q quuu[3]	uuuq quuu[3]
FSR	84h	xxxx xxxx	uuuu uuuu	uuuu uuuu
TRISA	85h	---1 1111	---1 1111	---u uuuu
TRISB	86h	1111 1111	1111 1111	uuuu uuuu
EECON1	88h	---0 x000	---0 q000	---0 uuuu
EECON2	89h	---- ----	---- ----	---- ----
PCLATH	8Ah	---0 0000	---0 0000	---u uuuu
INTCON	8Bh	0000 000x	0000 000u	uuuu uuuu[1]

Legend: u = unchanged, x = unknown, – = unimplemented bit read as '0',
        q = value depends on condition.

Note 1: One or more bits in INTCON will be affected (to cause wake-up).
     2: When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).
     3: Table 8-7 lists the reset value for each specific condition.

FIGURE 8-10: TIME-OUT SEQUENCE ON POWER-UP ($\overline{\text{MCLR}}$ NOT TIED TO V$_{DD}$): CASE 1
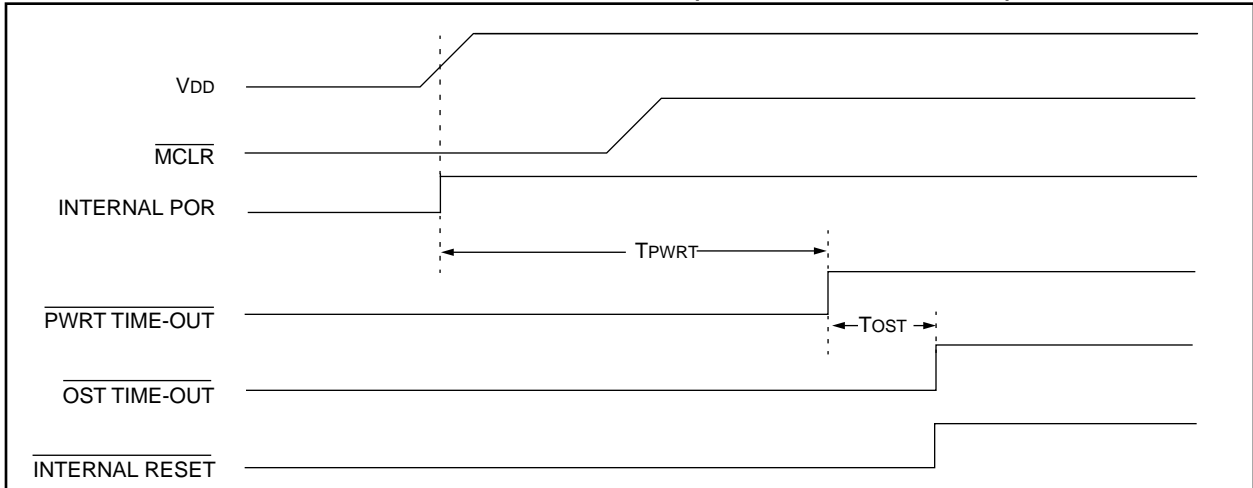

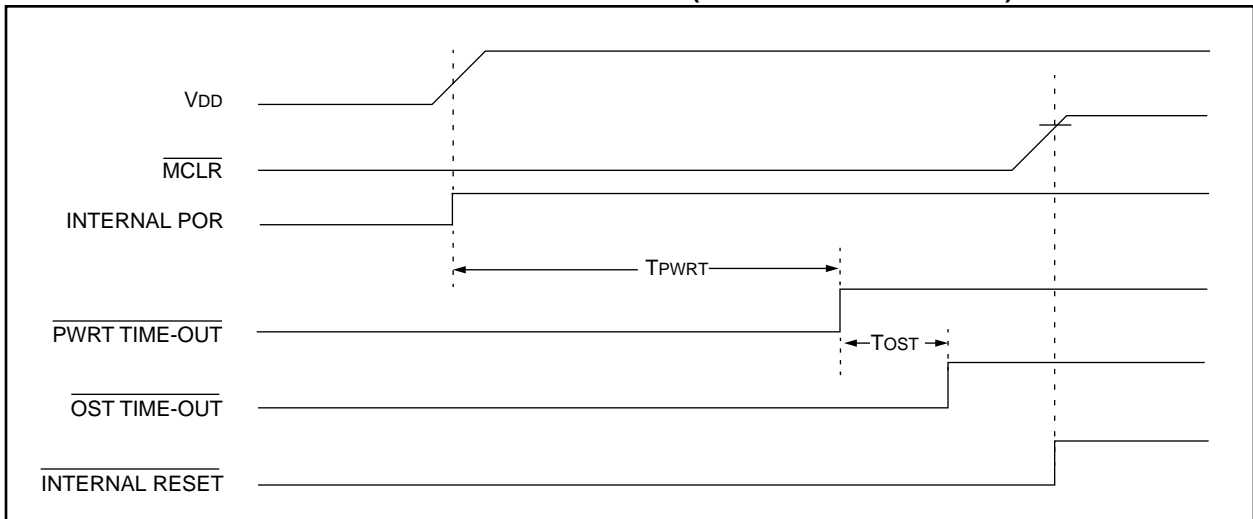FIGURE 8-11: TIME-OUT SEQUENCE ON POWER-UP ($\overline{\text{MCLR}}$ NOT TIED TO V$_{DD}$): CASE 2
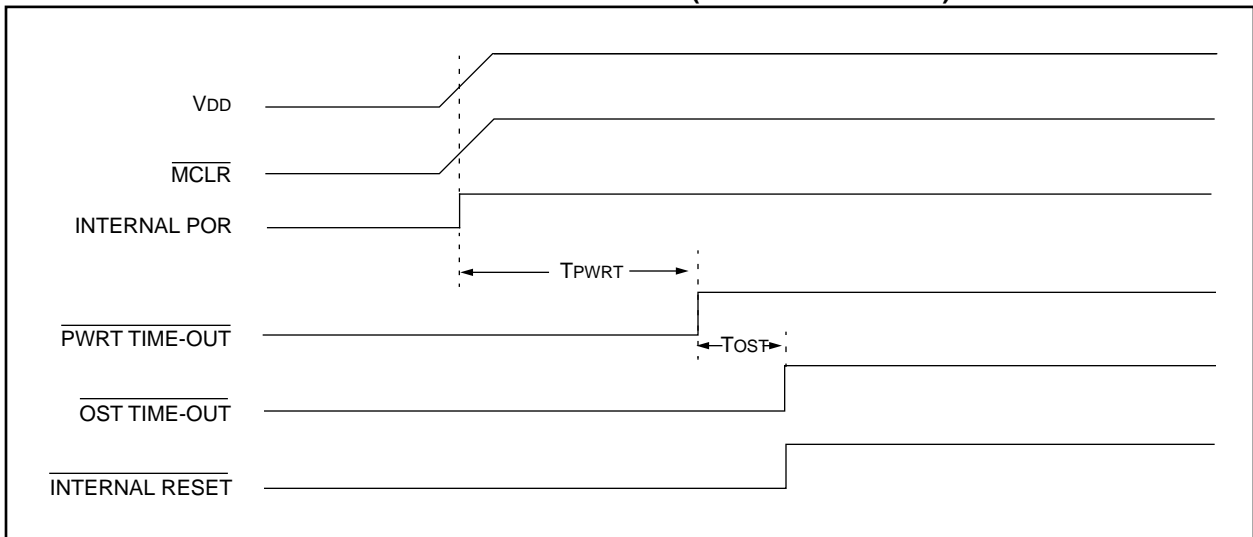

FIGURE 8-12: TIME-OUT SEQUENCE ON POWER-UP ($\overline{\text{MCLR}}$ TIED TO V$_{DD}$)

PIC16C8X
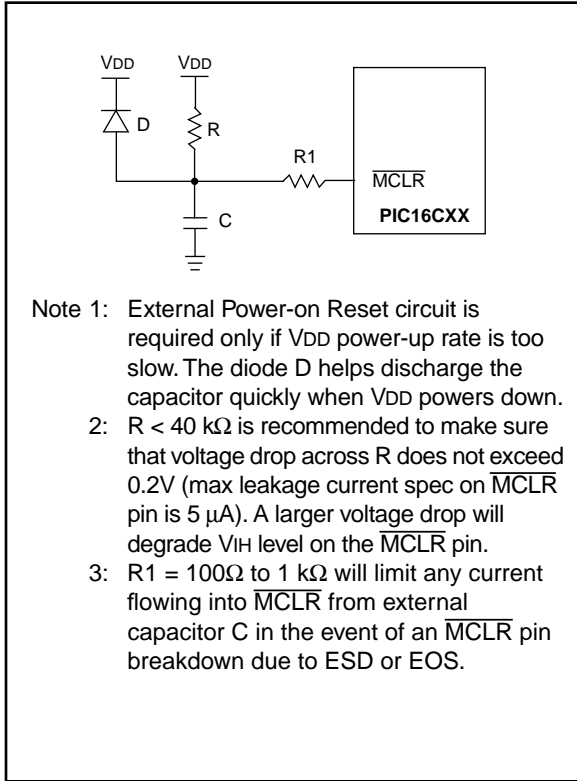
FIGURE 8-13: EXTERNAL POWER-ON RESET CIRCUIT (FOR SLOW V$_{DD}$ POWER-UP)

Note 1: External Power-on Reset circuit is required only if V$_{DD}$ power-up rate is too slow. The diode D helps discharge the capacitor quickly when V$_{DD}$ powers down.

2: R < 40 kΩ is recommended to make sure that voltage drop across R does not exceed 0.2V (max leakage current spec on $\overline{MCLR}$ pin is 5 μA). A larger voltage drop will degrade V$_{IH}$ level on the $\overline{MCLR}$ pin.

3: R1 = 100Ω to 1 kΩ will limit any current flowing into $\overline{MCLR}$ from external capacitor C in the event of an $\overline{MCLR}$ pin breakdown due to ESD or EOS.

FIGURE 8-14: BROWN-OUT PROTECTION CIRCUIT 1

This circuit will activate reset when V$_{DD}$ goes below (Vz + 0.7V) where Vz = Zener voltage.
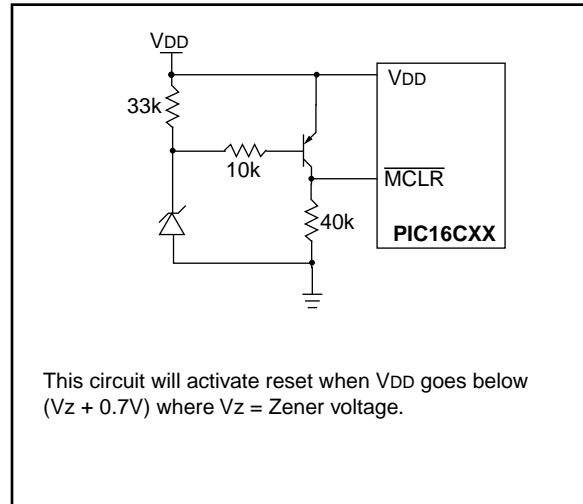
FIGURE 8-15: BROWN-OUT PROTECTION CIRCUIT 2

This brown-out circuit is less expensive, although less accurate. Transistor Q1 turns off when V$_{DD}$ is below a certain level such that:

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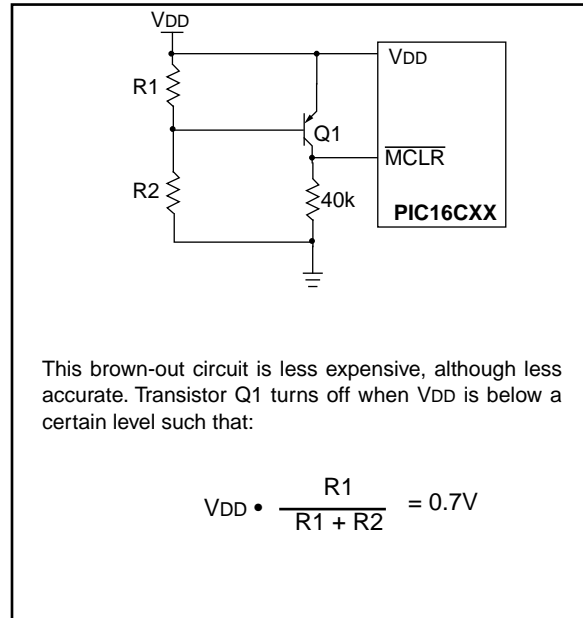
8.5   Interrupts

The PIC16C8X group has 4 sources of interrupt:

• External interrupt RB0/INT pin
• TMR0 overflow interrupt
• PORTB change interrupts (pins RB7:RB4)
• EEPROM write complete interrupt

The interrupt control register (INTCON) records individual interrupt requests in flag bits. It also contains the individual and global interrupt enable bits.

The global interrupt enable bit, GIE (INTCON<7>) enables (if set) all un-masked interrupts or disables (if cleared) all interrupts. Individual interrupts can be disabled through their corresponding enable bits in INTCON register. Bit GIE is cleared on reset.

The "return from interrupt" instruction, RETFIE, exits interrupt routine as well as sets the GIE bit, which re-enable interrupts.

The RB0/INT pin interrupt, the RB port change interrupt and the TMR0 overflow interrupt flags are contained in the INTCON register.

When an interrupt is responded to; the GIE bit is cleared to disable any further interrupt, the return address is pushed onto the stack and the PC is loaded with 0004h. For external interrupt events, such as the RB0/INT pin or PORTB change interrupt, the interrupt latency will be three to four instruction cycles. The exact latency depends when the interrupt event occurs (Figure 8-17). The latency is the same for both one and two cycle instructions. Once in the interrupt service routine the source(s) of the interrupt can be determined by polling the interrupt flag bits. The interrupt flag bit(s) must be cleared in software before re-enabling interrupts to avoid infinite interrupt requests.

Note 1: Individual interrupt flag bits are set regardless of the status of their corresponding mask bit or the GIE bit.

Note 2: For the PIC16C84 Only;
If an interrupt occurs while the Global Interrupt Enable (GIE) bit is being cleared, the GIE bit may unintentionally be re-enabled by the user's Interrupt Service Routine (the RETFIE instruction). The events that would cause this to occur are:

1. An instruction clears the GIE bit while an interrupt is acknowledged

2. The program branches to the Interrupt vector and executes the Interrupt Service Routine.

3. The Interrupt Service Routine completes with the execution of the RETFIE instruction. This causes the GIE bit to be set (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:

1. Ensure that the GIE bit is cleared by the instruction, as shown in the following code:

```
LOOP  BCF    INTCON,GIE  ;Disable All
                         ;  Interrupts
      BTFSC  INTCON,GIE  ;All Interrupts
                         ;  Disabled?
      GOTO   LOOP        ;NO, try again
                         ;  Yes, continue
                         ;  with program
                         ;  flow
```
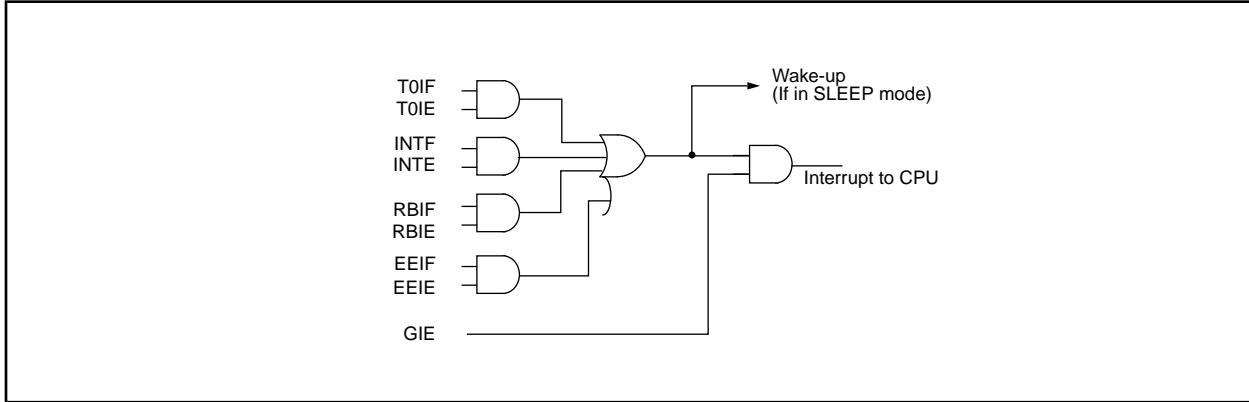

PIC16C8X

FIGURE 8-16: INTERRUPT LOGIC

FIGURE 8-17: INT PIN INTERRUPT TIMING

Note 1: INTF flag is sampled here (every Q1).
     2: Interrupt latency = 3-4Tcy where Tcy = instruction cycle time.
       Latency is the same whether Inst (PC) is a single cycle or a 2-cycle instruction.
     3: CLKOUT is available only in RC oscillator mode.
     4: For minimum width of INT pulse, refer to AC specs.
     5: INTF is enabled to be set anytime during the Q4-Q1 cycles.

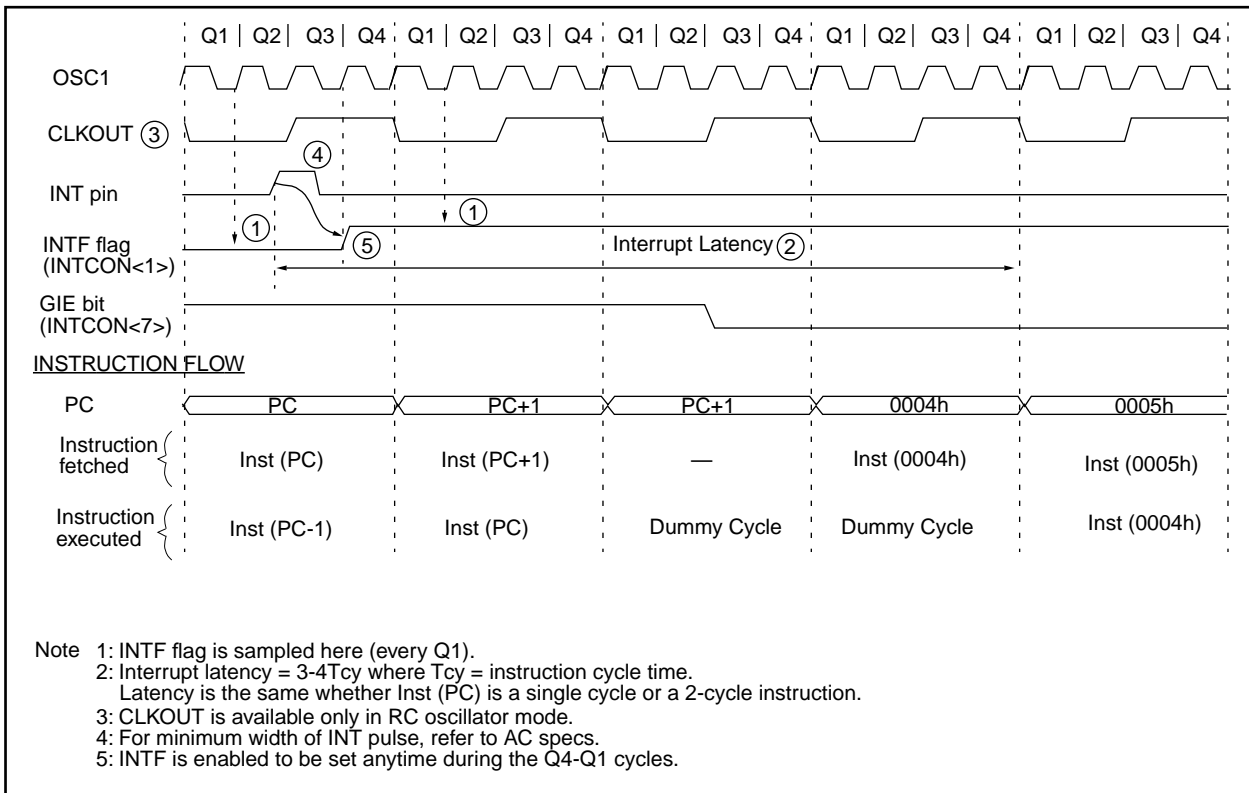### 8.5.1 INT INTERRUPT

External interrupt on RB0/INT pin is edge triggered: either rising if INTEDG bit (OPTION<6>) is set, or falling, if INTEDG bit is clear. When a valid edge appears on the RB0/INT pin, the INTF bit (INTCON<1>) is set. This interrupt can be disabled by clearing control bit INTE (INTCON<4>). Flag bit INTF must be cleared in software via the interrupt service routine before re-enabling this interrupt. The INT interrupt can wake the processor from SLEEP (Section 8.8) only if the INTE bit was set prior to going into SLEEP. The status of the GIE bit decides whether the processor branches to the interrupt vector following wake-up.

8.5.2 TMR0 INTERRUPT

An overflow (FFh → 00h) in TMR0 will set flag bit T0IF (INTCON<2>). The interrupt can be enabled/disabled by setting/clearing enable bit T0IE (INTCON<5>) (Section 6.0).

8.5.3 PORT RB INTERRUPT

An input change on PORTB<7:4> sets flag bit RBIF (INTCON<0>). The interrupt can be enabled/disabled by setting/clearing enable bit RBIE (INTCON<3>) (Section 5.2).

> **Note 1:** For the PIC16C84 Only;
> If a change on an I/O pin should occur when a read operation of PORTB is being executed (start of the Q2 cycle), the RBIF interrupt flag bit may not get set.
>
> **Note 2:** For all other PIC16C8X devices;
> For a change on the I/O pin to be recognized, the pulse width must be at least TCY wide.

8.6 Context Saving During Interrupts

During an interrupt, only the return PC value is saved on the stack. Typically, users wish to save key register values during an interrupt (e.g., W register and STATUS register). This is implemented in software.

Example 8-1 stores and restores the STATUS and W register's values. The User defined registers, W_TEMP and STATUS_TEMP are the temporary storage locations for the W and STATUS registers values.

Example 8-1 does the following:

a) Stores the W register.
b) Stores the STATUS register in STATUS_TEMP.
c) Executes the Interrupt Service Routine code.
d) Restores the STATUS (and bank select bit) register.
e) Restores the W register.

EXAMPLE 8-1: SAVING STATUS AND W REGISTERS IN RAM

```
PUSH    MOVWF   W_TEMP          ; Copy W to TEMP register,
        SWAPF   STATUS, W       ; Swap status to be saved into W
        MOVWF   STATUS_TEMP     ; Save status to STATUS_TEMP register
ISR     :                       :
        :                       ; Interrupt Service Routine
        :                       ;  should configure Bank as required
        :                       ;
POP     SWAPF   STATUS_TEMP, W  ; Swap nibbles in STATUS_TEMP register
                                ; and place result into W
        MOVWF   STATUS          ; Move W into STATUS register
                                ;   (sets bank to original state)
        SWAPF   W_TEMP, F       ; Swap nibbles in W_TEMP and place result in W_TEMP
        SWAPF   W_TEMP, W       ; Swap nibbles in W_TEMP and place result into W
```

PIC16C8X

8.7   Watchdog Timer (WDT)

The Watchdog Timer is a free running on-chip RC oscillator which 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/CLKOUT pins of the device has been stopped, for example, by execution of a SLEEP instruction. During normal operation a WDT time-out generates a device RESET. If the device is in SLEEP mode, a WDT Wake-up causes the device to wake-up and continue with normal operation. The WDT can be permanently disabled by programming configuration bit WDTE as a '0' (Section 8.1).

8.7.1   WDT PERIOD

The WDT has a nominal time-out period of 18 ms, (with no prescaler). 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 prescaler with a division ratio of up to 1:128 can be assigned to the WDT under software control by writing to the OPTION register. Thus, time-out periods up to 2.3 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 and generating a device RESET condition.

The $\overline{\text{TO}}$ bit in the STATUS register will be cleared upon a WDT time-out.

8.7.2   WDT PROGRAMMING CONSIDERATIONS

It should also be taken into account that under worst case conditions (VDD = Min., Temperature = Max., max. WDT prescaler) it may take several seconds before a WDT time-out occurs.
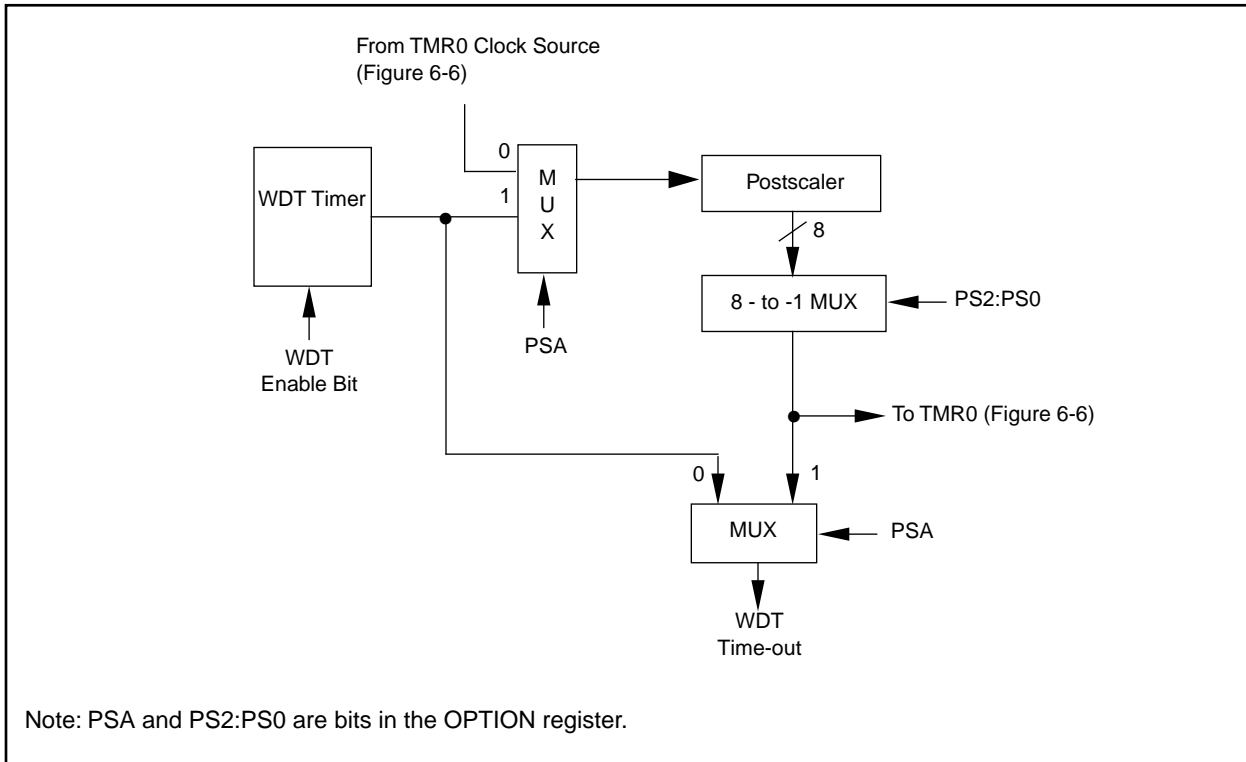
FIGURE 8-18:   WATCHDOG TIMER BLOCK DIAGRAM

Note: PSA and PS2:PS0 are bits in the OPTION register.

TABLE 8-9:   SUMMARY OF REGISTERS 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
2007h	Config. bits	(2)	(2)	(2)	CP	PWRTE[1]	WDTE	FOSC1	FOSC0	(2)	
81h	OPTION	$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111

Legend: x = unknown. Shaded cells are not used by the WDT.

Note 1:   See Figure 8-1, Figure 8-2, and Figure 8-3 for operation of the PWRTE bit.

2:    See Figure 8-1, Figure 8-2, Figure 8-3, and Section 8.9 for operation of the Code and Data protection bits.

8.8   Power-down Mode (SLEEP)

The Power-down mode is entered by executing the SLEEP instruction.

If enabled, the Watchdog Timer will be cleared but keeps running, the $\overline{PD}$ bit (STATUS<3>) is cleared, the $\overline{TO}$ bit (STATUS<4>) is set, and the oscillator driver is turned off. The I/O ports maintain the status they had, before the SLEEP instruction was executed (driving high, low, or hi-impedance).

For the lowest current consumption, in SLEEP mode, place all I/O pins at either at VDD, or VSS, with no external circuitry drawing current from the I/O pin, and disable external clocks. I/O pins that are hi-impedance inputs should be pulled high or low externally to avoid switching currents caused by floating inputs. The T0CKI input should also be at VDD or VSS. The contribution from on-chip pull-ups on PORTB should be considered.

The $\overline{MCLR}$ pin must be at a logic high level (VIHMC).

It should be noted that a RESET generated by a WDT time-out does not drive the $\overline{MCLR}$ pin low.

8.8.1   WAKE-UP FROM SLEEP

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

1. External reset input on $\overline{MCLR}$ pin.
2. WDT Wake-up (if WDT was enabled).
3. Interrupt from RB0/INT pin, RB port change, or data EEPROM write complete.

Peripherals cannot generate interrupts during SLEEP, since no on-chip Q clocks are present.
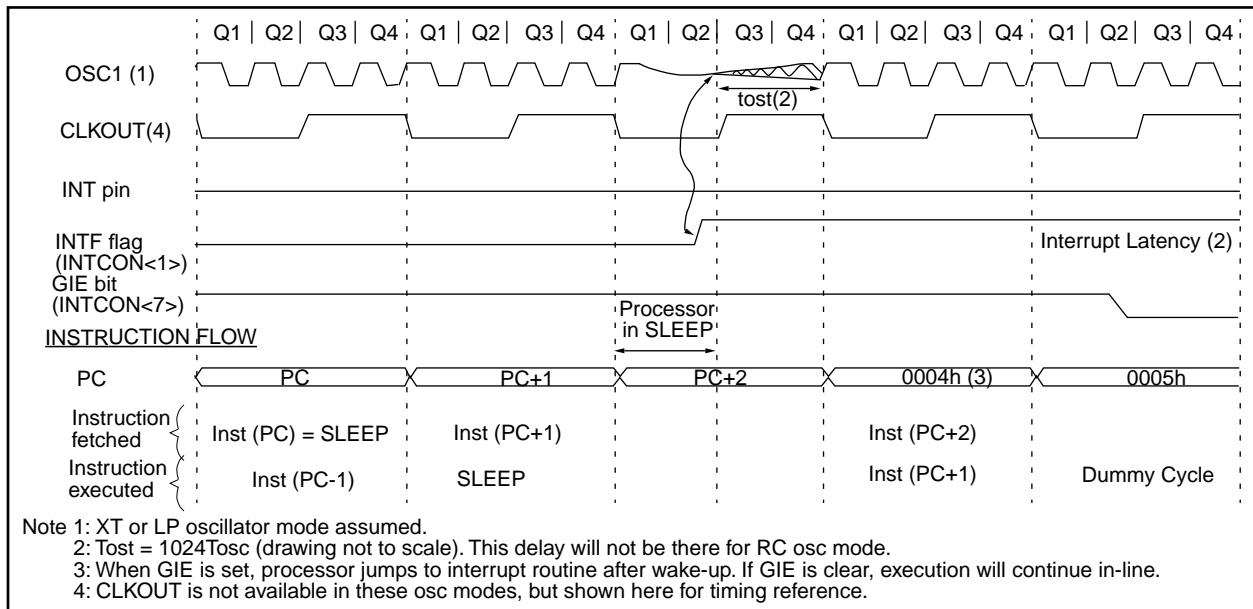
The first event ($\overline{MCLR}$ reset) will cause a device reset. The two latter events are considered a continuation of program execution. The $\overline{TO}$ and $\overline{PD}$ bits can be used to determine the cause of a device reset. The $\overline{PD}$ bit, which is set on power-up, is cleared when SLEEP is invoked. The $\overline{TO}$ bit is cleared if a WDT time-out occurred (and caused wake-up).

While 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 occurs regardless of the state of the GIE bit. If the GIE bit is clear (disabled), the device continues execution at the instruction after the SLEEP instruction. If the GIE bit is set (enabled), the device executes the instruction after the SLEEP instruction and then branches to the interrupt address (0004h). 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 global interrupts are disabled (GIE cleared), but any interrupt source has both its interrupt enable bit and corresponding interrupt flag bits set, the device will immediately wake from sleep. The SLEEP instruction is completely executed.

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

FIGURE 8-19:   WAKE-UP FROM SLEEP THROUGH INTERRUPT

Note 1: XT or LP oscillator mode assumed.
    2: Tost = 1024Tosc (drawing not to scale). This delay will not be there for RC osc mode.
    3: When GIE is set, processor jumps to interrupt routine after wake-up. If GIE is clear, execution will continue in-line.
    4: CLKOUT is not available in these osc modes, but shown here for timing reference.

PIC16C8X

8.9   Code Protection

The code in the program memory and data EEPROM memory can be protected by programming the code protect bit (Figure 8-1, Figure 8-2, and Figure 8-3).

8.9.1   ROM DEVICES

There are two protection configuration bits. One for the program memory, which is specified as part of the ROM code submittal. The second for the EEPROM data memory. When ROM devices complete testing, the EEPROM data memory code protect configuration bit will be programmed to the same state as the program memory code protect configuration bit.

In applications where the device is code protected and the data EEPROM needs to be programmed before being placed in the application, the data EEPROM memory array needs to be erased and then the data memory code protect disabled. This will allow the desired data to be programmed into the device. The sequence to disable the data EEPROM memory protection is shown in the PIC16C84 Programming Specification (Literature number DS30189D) in Section 3.1.1.

After programming the data EEPROM memory array, the data EEPROM memory code protect configuration bit should be programmed as desired.

8.10   ID Locations

Four memory locations (2000h - 2003h) are designated as ID locations to store checksum or other code identification numbers. These locations are not accessible during normal execution but are readable and writable only during program/verify. Only the 4 least significant bits of ID location are usable.

For ROM devices, these values are submitted along with the ROM code.

8.11   In-Circuit Serial Programming

PIC16C8X microcontrollers can be serially programmed while in the end application circuit. This is simply done with two lines for clock and data, and three other lines for power, ground, and the programming voltage. Customers can manufacture boards with unprogrammed devices, and then program the microcontroller just before shipping the product, allowing the most recent firmware or custom firmware to be programmed.

The device is placed into a program/verify mode by holding the RB6 and RB7 pins low, while raising the $\overline{MCLR}$ pin from V_{IL} to V_{IHH} (see programming specification). RB6 becomes the programming clock and RB7 becomes the programming data. Both RB6 and RB7 are Schmitt Trigger inputs in this mode.

After reset, to place the device into programming/verify mode, the program counter (PC) points to location 00h. A 6-bit command is then supplied to the device, 14-bits of program data is then supplied to or from the device, using load or read-type instructions. For complete details of serial programming, please refer to the PIC16CXX Programming Specifications (Literature #DS30189).

FIGURE 8-20:   TYPICAL IN-SYSTEM SERIAL PROGRAMMING CONNECTION
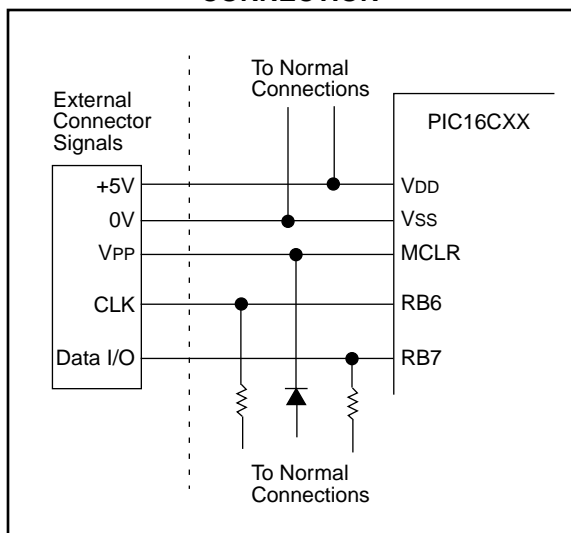
For ROM devices, both the program memory and Data EEPROM memory may be read, but only the Data EEPROM memory may be programmed.

9.0   INSTRUCTION SET SUMMARY

Each PIC16CXX instruction is a 14-bit word divided into an OPCODE which specifies the instruction type and one or more operands which further specify the operation of the instruction. The PIC16CXX instruction set summary in Table 9-2 lists byte-oriented, bit-oriented, and literal and control operations. Table 9-1 shows the opcode field 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' is 0, the result is placed in the W register. If 'd' is 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 address of the file in which the bit is located.

Literal and control operations: 'k' represents an eight or eleven bit constant or literal value.

TABLE 9-1:   OPCODE FIELD DESCRIPTIONS

Field	Description
f	Register file address (0x00 to 0x7F)
W	Working register (accumulator)
b	Bit address within an 8-bit file register
k	Literal field, constant data or label
x	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; d = 0: store result in W, d = 1: store result in file register f. Default is d = 1
label	Label name
TOS	Top of Stack
PC	Program Counter
PCLATH	Program Counter High Latch
GIE	Global Interrupt Enable bit
WDT	Watchdog Timer/Counter
$\overline{TO}$	Time-out bit
$\overline{PD}$	Power-down bit
dest	Destination (Either the W register or the specified register file location)
[ ]	Options
( )	Contents
→	Assigned to
< >	Register bit field
∈	In the set of
italics	User defined term (font is courier)

The instruction set is highly orthogonal and is grouped into three basic categories:

- Byte-oriented
- Bit-oriented
- Literal and control

All instructions are executed within a single instruction cycle, unless a conditional test is true or the program counter is changed as a result of the instruction. The execution takes two instruction cycles with the second cycle executed as a NOP. Each cycle consists of four oscillator periods. Thus, for an oscillator frequency of 4 MHz, the normal instruction execution time is 1 μs. The instruction execution time is 2 μs for program branches.

Table 9-2 lists the instructions recognized by Microchip's assembler (MPASM).

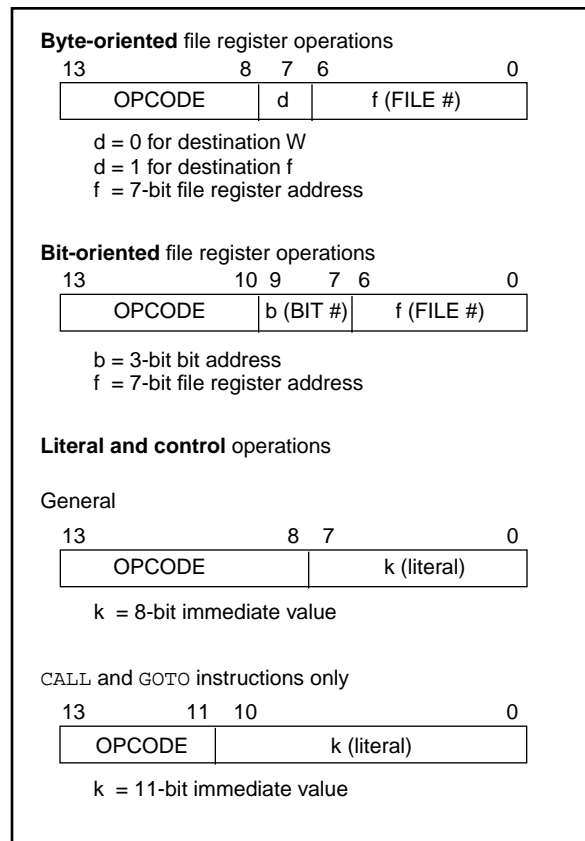Figure 9-1 shows the three general formats of instructions.

Note:	To maintain upward compatibility with future PIC16CXX products, <u>do not use</u> the OPTION and TRIS instructions.

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

0xhh

where h signifies a hexadecimal digit.

FIGURE 9-1:   GENERAL FORMAT FOR INSTRUCTIONS

Byte-oriented file register operations

```
13            8 7  6              0
┌──────────────┬───┬──────────────┐
│   OPCODE     │ d │   f (FILE #)  │
└──────────────┴───┴──────────────┘
```
d = 0 for destination W
d = 1 for destination f
f = 7-bit file register address

Bit-oriented file register operations

```
13         10 9    7 6             0
┌───────────┬───────┬──────────────┐
│  OPCODE   │b (BIT #)│  f (FILE #)  │
└───────────┴───────┴──────────────┘
```
b = 3-bit bit address
f = 7-bit file register address

Literal and control operations

General

```
13            8 7                  0
┌──────────────┬───────────────────┐
│   OPCODE     │    k (literal)     │
└──────────────┴───────────────────┘
```
k = 8-bit immediate value

CALL and GOTO instructions only

```
13      11 10                      0
┌─────────┬─────────────────────────┐
│ OPCODE  │      k (literal)         │
└─────────┴─────────────────────────┘
```
k = 11-bit immediate value

PIC16C8X

TABLE 9-2: INSTRUCTION SET SUMMARY

Mnemonic, Operands		Description	Cycles	14-Bit Opcode MSb ⟶ LSb				Status Affected	Notes
ADDWF	f, d	Add W and f	1	00	0111	dfff	ffff	C,DC,Z	1,2
ANDWF	f, d	AND W with f	1	00	0101	dfff	ffff	Z	1,2
CLRF	f	Clear f	1	00	0001	1fff	ffff	Z	2
CLRW	-	Clear W	1	00	0001	0xxx	xxxx	Z	
COMF	f, d	Complement f	1	00	1001	dfff	ffff	Z	1,2
DECF	f, d	Decrement f	1	00	0011	dfff	ffff	Z	1,2
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	00	1011	dfff	ffff	None	1,2,3
INCF	f, d	Increment f	1	00	1010	dfff	ffff	Z	1,2
INCFSZ	f, d	Increment f, Skip if 0	1(2)	00	1111	dfff	ffff	None	1,2,3
IORWF	f, d	Inclusive OR W with f	1	00	0100	dfff	ffff	Z	1,2
MOVF	f, d	Move f	1	00	1000	dfff	ffff	Z	1,2
MOVWF	f	Move W to f	1	00	0000	1fff	ffff	None	
NOP	-	No Operation	1	00	0000	0xx0	0000	None	
RLF	f, d	Rotate left f through carry	1	00	1101	dfff	ffff	C	1,2
RRF	f, d	Rotate right f through carry	1	00	1100	dfff	ffff	C	1,2
SUBWF	f, d	Subtract W from f	1	00	0010	dfff	ffff	C,DC,Z	1,2
SWAPF	f, d	Swap nibbles in f	1	00	1110	dfff	ffff	None	1,2
XORWF	f, d	Exclusive OR W with f	1	00	0110	dfff	ffff	Z	1,2
BIT-ORIENTED FILE REGISTER OPERATIONS									
BCF	f, b	Bit Clear f	1	01	00bb	bfff	ffff	None	1,2
BSF	f, b	Bit Set f	1	01	01bb	bfff	ffff	None	1,2
BTFSC	f, b	Bit Test f, Skip if Clear	1 (2)	01	10bb	bfff	ffff	None	3
BTFSS	f, b	Bit Test f, Skip if Set	1 (2)	01	11bb	bfff	ffff	None	3
LITERAL AND CONTROL OPERATIONS									
ADDLW	k	Add literal and W	1	11	111x	kkkk	kkkk	C,DC,Z	
ANDLW	k	AND literal with W	1	11	1001	kkkk	kkkk	Z	
CALL	k	Call subroutine	2	10	0kkk	kkkk	kkkk		
CLRWDT	-	Clear Watchdog Timer	1	00	0000	0110	0100	$\overline{TO},\overline{PD}$	
GOTO	k	Go to address	2	10	1kkk	kkkk	kkkk	None	
IORLW	k	Inclusive OR literal with W	1	11	1000	kkkk	kkkk	Z	
MOVLW	k	Move literal to W	1	11	00xx	kkkk	kkkk	None	
RETFIE	-	Return from interrupt	2	00	0000	0000	1001	None	
RETLW	k	Return with literal in W	2	11	01xx	kkkk	kkkk	None	
RETURN	-	Return from subroutine	2	00	0000	0000	1000	None	
SLEEP	-	Go into standby mode	1	00	0000	0110	0011	$\overline{TO},\overline{PD}$	
SUBLW	k	Subtract W from literal	1	11	110x	kkkk	kkkk	C,DC,Z	
XORLW	k	Exclusive OR literal with W	1	11	1010	kkkk	kkkk	Z	

Note 1: When an I/O register is modified as a function of itself (i.e., MOVF PORTB, 1), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.

2: If this instruction is executed on the TMR0 register (and, where applicable, d = 1), the prescaler will be cleared if assigned to the TMR0.

3: If Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.

9.1   Instruction Descriptions

ADDLW	Add Literal and W
Syntax:	[*label*] ADDLW k
Operands:	$0 \le k \le 255$
Operation:	(W) + k $\rightarrow$ (W)
Status Affected:	C, DC, Z

Encoding:

11	111x	kkkk	kkkk

Description: The contents of the W register are added to the eight bit literal 'k' and the result is placed back in the W register.

Words: 1

Cycles: 1

Example   ADDLW   0x15

Before Instruction
   W   =   0x10
After Instruction
   W   =   0x25

ANDLW	AND Literal with W
Syntax:	[*label*] ANDLW k
Operands:	$0 \le k \le 255$
Operation:	(W) .AND. (k) $\rightarrow$ (W)
Status Affected:	Z

Encoding:

11	1001	kkkk	kkkk

Description: The contents of W register is AND'ed with the eight bit literal 'k'. The result is placed back in the W register.

Words: 1

Cycles: 1

Example   ANDLW   0x5F

Before Instruction
   W   =   0xA3
After Instruction
   W   =   0x03

ADDWF	Add W and f
Syntax:	[*label*] ADDWF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(W) + (f) $\rightarrow$ (dest)
Status Affected:	C, DC, Z

Encoding:

00	0111	dfff	ffff

Description: Add the contents of the W register to register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'.

Words: 1

Cycles: 1

Example   ADDWF   FSR, 0

Before Instruction
   W   =   0x17
   FSR =   0xC2
After Instruction
   W   =   0xD9
   FSR =   0xC2

ANDWF	AND W with f
Syntax:	[*label*] ANDWF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(W) .AND. (f) $\rightarrow$ (dest)
Status Affected:	Z

Encoding:

00	0101	dfff	ffff

Description: AND the W register with register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'.

Words: 1

Cycles: 1

Example   ANDWF   FSR, 1

Before Instruction
   W   =   0x17
   FSR =   0xC2
After Instruction
   W   =   0x17
   FSR =   0x02

PIC16C8X

BCF	**Bit Clear f**
Syntax:	[*label*] BCF f,b
Operands:	$0 \leq f \leq 127$ $0 \leq b \leq 7$
Operation:	$0 \rightarrow$ (f<b>)
Status Affected:	None
Encoding:		01	00bb	bfff	ffff	
Description:	Bit 'b' in register 'f' is cleared.
Words:	1
Cycles:	1
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:	$0 \leq f \leq 127$ $0 \leq b \leq 7$
Operation:	$1 \rightarrow$ (f<b>)
Status Affected:	None
Encoding:		01	01bb	bfff	ffff	
Description:	Bit 'b' in register 'f' is set.
Words:	1
Cycles:	1
Example	BSF FLAG_REG, 7

Before Instruction
    FLAG_REG= 0x0A
After Instruction
    FLAG_REG= 0x8A

BTFSC	**Bit Test f, Skip if Clear**
Syntax:	[*label*] BTFSC f,b
Operands:	$0 \leq f \leq 127$ $0 \leq b \leq 7$
Operation:	skip if (f<b>) = 0
Status Affected:	None
Encoding:		01	10bb	bfff	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 2 cycle instruction.
Words:	1
Cycles:	1(2)

Example	HERE BTFSC FLAG,1
	FALSE GOTO PROCESS_CODE
	TRUE •
	              •
	              •

Before Instruction
    PC =   address HERE
After Instruction
    if FLAG<1>=0,
    PC=address   TRUE
    if FLAG<1>=1,
    PC=address   FALSE

BTFSS	Bit Test f, skip if Set

Syntax: [*label*] BTFSS f,b

Operands: 0 ≤ f ≤ 127
0 ≤ b < 7

Operation: skip if (f<b>) = 1

Status Affected: None

Encoding:

01	11bb	bfff	ffff

Description: If bit 'b' in register 'f' is 1 then the next instruction is skipped.
If bit 'b' is 1, then the next instruction fetched during the current instruction execution, is discarded and a NOP is executed instead, making this a 2 cycle instruction.

Words: 1

Cycles: 1(2)

Example

```
HERE    BTFSC  FLAG,1
FALSE   GOTO   PROCESS_CODE
TRUE    •
        •
        •
```

Before Instruction
        PC =   address HERE
After Instruction
        if FLAG<1>=0,
        PC=address   FALSE
        if FLAG<1>=1,
        PC=address   TRUE

CALL	Subroutine Call

Syntax: [*label*] CALL k

Operands: 0 ≤ k ≤ 2047

Operation: (PC)+ 1→ TOS,
k → (PC<10:0>),
(PCLATH<4:3>) → (PC<12:11>)

Status Affected: None

Encoding:

10	0kkk	kkkk	kkkk

Description: Subroutine call. First, return address (PC+1) is pushed onto the stack. The eleven bit immediate address is loaded into PC bits <10:0>. The upper bits of the PC are loaded from PCLATH. CALL is a two cycle instruction.

Words: 1

Cycles: 2

Example

```
HERE    CALL   THERE
```

Before Instruction
        PC =   Address HERE
After Instruction
        PC =   Address THERE
        TOS =   Address HERE

CLRF	Clear f

Syntax: [*label*] CLRF   f

Operands: 0 ≤ f ≤ 127

Operation: 00h → (f)
1 → Z

Status Affected: Z

Encoding:

00	0001	1fff	ffff

Description: The contents of register 'f' are cleared and the Z bit is set.

Words: 1

Cycles: 1

Example

```
CLRF       FLAG_REG
```

Before Instruction
        FLAG_REG =   0x5A
After Instruction
        FLAG_REG =   0x00
        Z   =   1

CLRW	Clear W Register

Syntax: [*label*] CLRW

Operands: None

Operation: 00h → (W)
1 → Z

Status Affected: Z

Encoding:

00	0001	0xxx	xxxx

Description: W register is cleared. Zero bit (Z) is set.

Words: 1

Cycles: 1

Example

```
CLRW
```

Before Instruction
        W =   0x5A
After Instruction
        W =   0x00
        Z =   1

PIC16C8X

CLRWDT	Clear Watchdog Timer
Syntax:	[*label*] CLRWDT
Operands:	None
Operation:	00h → WDT 0 → WDT prescaler, 1 → $\overline{TO}$ 1 → $\overline{PD}$
Status Affected:	$\overline{TO}$, $\overline{PD}$

Encoding:

00	0000	0110	0100

Description:	The CLRWDT instruction resets the watchdog timer. It also resets the prescaler of the WDT. Status bits $\overline{TO}$ and $\overline{PD}$ are set.
Words:	1
Cycles:	1
Example	CLRWDT

Before Instruction
    WDT counter =   ?
After Instruction
    WDT counter =   0x00
    WDT prescale =   0
    $\overline{TO}$ =   1
    $\overline{PD}$ =   1

COMF	Complement f
Syntax:	[*label*] COMF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	$(\overline{f})$ → (dest)
Status Affected:	Z

Encoding:

00	1001	dfff	ffff

Description:	The contents of register 'f' are complemented. If 'd' is 0 the result is stored in W. If 'd' is 1 the result is stored back in register 'f'.
Words:	1
Cycles:	1
Example	COMF REG1,0

Before Instruction
    REG1   =   0x13
After Instruction
    REG1   =   0x13
    W   =   0xEC

DECF	Decrement f
Syntax:	[*label*] DECF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(f) − 1 → (dest)
Status Affected:	Z

Encoding:

00	0011	dfff	ffff

Description:	Decrement register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'.
Words:	1
Cycles:	1
Example	DECF CNT, 1

Before Instruction
    CNT   =   0x01
    Z   =   0
After Instruction
    CNT   =   0x00
    Z   =   1

DECFSZ	Decrement f, Skip if 0
Syntax:	[*label*] DECFSZ f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(f) − 1 → (dest); skip if result = 0
Status Affected:	None

Encoding:

00	1011	dfff	ffff

Description:	The contents of register 'f' are decremented. If 'd' is 0 the result is placed in the W register. 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. A NOP is executed instead making it a two cycle instruction.
Words:	1
Cycles:	1(2)

Example	HERE DECFSZ CNT, 1
	         GOTO LOOP
	CONTINUE •
	         •
	         •

Before Instruction
    PC   =   address HERE
After Instruction
    CNT   =   CNT - 1
    if CNT =   0,
    PC   =   address CONTINUE
    if CNT ≠   0,
    PC   =   address HERE+1

DS30081F-page 58   © 1995 Microchip Technology Inc.

GOTO — Go to address

Syntax:	[*label*] GOTO k
Operands:	0 ≤ k ≤ 2047
Operation:	k → (PC<10:0>) (PCLATH<4:3>) → (PC<12:11>)
Status Affected:	None

Encoding:

10	1kkk	kkkk	kkkk

Description: GOTO is an unconditional branch. The eleven bit immediate value is loaded into PC bits <10:0>. The upper bits of PC are loaded from PCLATH<4:3>. GOTO is a two cycle instruction.

Words:	1
Cycles:	2

Example   GOTO THERE

After Instruction
PC =   Address THERE

INCF — Increment f

Syntax:	[*label*] INCF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f) + 1 → (dest)
Status Affected:	Z

Encoding:

00	1010	dfff	ffff

Description: The contents of register 'f' are incremented. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'.

Words:	1
Cycles:	1

Example   INCF   CNT, 1

Before Instruction
CNT   =   0xFF
Z   =   0
After Instruction
CNT   =   0x00
Z   =   1

INCFSZ — Increment f, Skip if 0

Syntax:	[*label*] INCFSZ f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f) + 1 → (dest), skip if result = 0
Status Affected:	None

Encoding:

00	1111	dfff	ffff

Description: The contents of register 'f' are incremented. If 'd' is 0 the result is placed in the W register. 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. A NOP is executed instead making it a two cycle instruction.

Words:	1
Cycles:	1(2)

Example   HERE   INCFSZ   CNT,
              1
                       GOTO   LOOP
              CONTINUE •
                       •
                       •

Before Instruction
PC   =   addressHERE
After Instruction
CNT =   CNT + 1
if CNT = 0,
PC   =   addressCONTINUE
if CNT≠ 0,
PC   =   addressHERE +1

IORLW — Inclusive OR Literal with W

Syntax:	[*label*] IORLW k
Operands:	0 ≤ k ≤ 255
Operation:	(W) .OR. (k) → (W)
Status Affected:	Z

Encoding:

11	1000	kkkk	kkkk

Description: The contents of the W register are OR'ed to the eight bit literal 'k'. The result is placed in the W register.

Words:	1
Cycles:	1

Example   IORLW   0x35

Before Instruction
W =   0x9A
After Instruction
W =   0xBF

PIC16C8X

IORWF	Inclusive OR W with f
Syntax:	[*label*] IORWF f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	(W) .OR. (f) $\rightarrow$ (W)
Status Affected:	Z

Encoding:

00	0100	dfff	ffff

Description:	Inclusive OR the W register to register 'f'. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'.
Words:	1
Cycles:	1
Example	IORWF RESULT, 0

Before Instruction
  RESULT = 0x13
  W   = 0x91
After Instruction
  RESULT = 0x13
  W   = 0x93

MOVF	Move f
Syntax:	[*label*] MOVF f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	(f) $\rightarrow$ (dest)
Status Affected:	Z

Encoding:

00	1000	dfff	ffff

Description:	The contents of register f is moved to destination d. If d = 0, destination is W register. If d = 1, the destination is file register f itself. d = 1 is useful to test a file register since status flag Z is affected.
Words:	1
Cycles:	1
Example	MOVF FSR, 0

After Instruction
  W = value in FSR register

MOVLW	Move literal to W
Syntax:	[*label*] MOVLW k
Operands:	$0 \leq k \leq 255$
Operation:	k $\rightarrow$ (W)
Status Affected:	None

Encoding:

11	00XX	kkkk	kkkk

Description:	The eight bit literal 'k' is loaded into W register. The don't cares will assemble as 0's.
Words:	1
Cycles:	1
Example	MOVLW 0x5A

After Instruction
  W = 0x5A

MOVWF	Move W to f
Syntax:	[*label*] MOVWF f
Operands:	$0 \leq f \leq 127$
Operation:	(W) $\rightarrow$ (f)
Status Affected:	None

Encoding:

00	0000	1fff	ffff

Description:	Move data from W register to register 'f'.
Words:	1
Cycles:	1
Example	MOVWF OPTION

Before Instruction
  OPTION = 0xFF
  W   = 0x4F
After Instruction
  OPTION = 0x4F
  W   = 0x4F

NOP — No Operation

Syntax:	[*label*] NOP
Operands:	None
Operation:	No operation
Status Affected:	None

Encoding:

00	0000	0xx0	0000

Description:	No operation.
Words:	1
Cycles:	1
Example	NOP

RETFIE — Return from Interrupt

Syntax:	[*label*] RETFIE
Operands:	None
Operation:	TOS → (PC), 1 → GIE
Status Affected:	None

Encoding:

00	0000	0000	1001

Description:	The Stack is popped and Top of Stack (TOS) is loaded into the PC. Interrupts are enabled by setting the Global Interrupt Enable bit. This is a two cycle instruction.
Words:	1
Cycles:	2
Example	RETFIE

After Interrupt
```
        PC  =   TOS
        GIE =   1
```

OPTION — Load Option Register

Syntax:	[*label*] OPTION
Operands:	None
Operation:	(W) → OPTION
Status Affected:	None

Encoding:

00	0000	0110	0010

Description:	The contents of the W register are loaded in the OPTION register. This instruction is supported for code compatibility with PIC16C5X products. Since OPTION is a readable/writable register, the user can directly address it.
Words:	1
Cycles:	1
Example	

> **Note:** **To maintain upward compatibility with future PIC16CXX products, do not use this instruction.**

RETLW — Return Literal to W

Syntax:	[*label*] RETLW k
Operands:	$0 \le k \le 255$
Operation:	k → (W), TOS → (PC)
Status Affected:	None

Encoding:

11	01xx	kkkk	kkkk

Description:	The W register is loaded with the eight bit literal 'k'. The program counter is loaded from the top of the stack (the return address). This is a two cycle instruction.
Words:	1
Cycles:	2

Example
```
        CALL TABLE  ;W contains table
                    ;offset value
        •           ;W now has table value
        •
        •
TABLE   ADDWF PC  ;W = offset
        RETLW k1  ;Begin table
        RETLW k2  ;
        •
        •
        •
        RETLW kn  ; End of table
```

Before Instruction
```
        W   =   0x07
```
After Instruction
```
        W   =   value of k7
```

PIC16C8X

RETURN	Return from Subroutine
Syntax:	[*label*] RETURN
Operands:	None
Operation:	TOS → (PC)
Status Affected:	None
Encoding:	00 0000 0000 1000
Description:	Return from subroutine. The stack is popped and the Top of Stack (TOS) is loaded into the program counter. This is a two cycle instruction.
Words:	1
Cycles:	2
Example	RETURN

After Interrupt
        PC   =   TOS

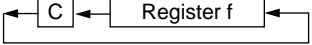| RLF	Rotate Left f through Carry
Syntax:	[*label*] RLF f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	See description below
Status Affected:	C
Encoding:	00 1101 dfff ffff
Description:	The contents of register 'f' are rotated one bit to the left through the Carry Flag. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is stored back in register 'f'.

Words:	1
Cycles:	1
Example	RLF REG1,0

Before Instruction
     REG1   =   1110 0110
     C   =   0
After Instruction
     REG1   =   1110 0110
     W   =   1100 1100
     C   =   1

RRF	Rotate Right f through Carry
Syntax:	[*label*] RRF f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	See description below
Status Affected:	C
Encoding:	00 1100 dfff ffff
Description:	The contents of register 'f' are rotated one bit to the right through the Carry Flag. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'.

Words:	1
Cycles:	1
Example	RRF REG1,0

Before Instruction
     REG1   =   1110 0110
     C   =   0
After Instruction
     REG1   =   1110 0110
     W   =   0111 0011
     C   =   1

SLEEP	Go into Standby Mode
Syntax:	[*label*] SLEEP
Operands:	None
Operation:	00h → WDT, 0 → WDT prescaler 1 → $\overline{TO}$, 0 → $\overline{PD}$
Status Affected:	$\overline{TO}$, $\overline{PD}$
Encoding:	00 0000 0110 0011
Description:	The power down status bit ($\overline{PD}$) is cleared. 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
Example:	SLEEP

SUBLW	**Subtract W from Literal**
Syntax:	[*label*] SUBLW k
Operands:	$0 \le k \le 255$
Operation:	$k - (W) \rightarrow (W)$
Status Affected:	C, DC, Z
Encoding:	`11` `110x` `kkkk` `kkkk`
Description:	The W register is subtracted (2's complement method) from the eight bit literal 'k'. The result is placed in the W register.
Words:	1
Cycles:	1
Example 1:	SUBLW 0x02

Before Instruction

    W   = 1
    C   = ?

After Instruction

    W   = 1
    C   = 1; result is positive

Example 2:   Before Instruction

    W   = 2
    C   = ?

After Instruction

    W   = 0
    C   = 1; result is zero

Example 3:   Before Instruction

    W   = 3
    C   = ?

After Instruction

    W   = FF
    C   = 0; result is negative

SUBWF	**Subtract W from f**
Syntax:	[*label*] SUBWF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	$(f) - (W) \rightarrow (dest)$
Status Affected:	C, DC, Z
Encoding:	`00` `0010` `dfff` `ffff`
Description:	Subtract (2's complement methodize W register from register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'.
Words:	1
Cycles:	1
Example 1:	SUBWF REG1,1

Before Instruction

    REG1 = 3
    W   = 2
    C   = ?

After Instruction

    REG1 = 1
    W   = 2
    C   = 1; result is positive

Example 2:   Before Instruction

    REG1 = 2
    W   = 2
    C   = ?

After Instruction

    REG1 = 0
    W   = 2
    C   = 1; result is zero

Example 3:   Before Instruction

    REG1 = 1
    W   = 2
    C   = ?

After Instruction

    REG1 = FF
    W   = 2
    C   = 0; result is negative

SWAPF	Swap f
Syntax:	[*label* SWAPF f,d]
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	(f<3:0>) $\rightarrow$ (dest<7:4>), (f<7:4>) $\rightarrow$ (dest<3:0>)
Status Affected:	None

Encoding:

00	1110	dfff	ffff

Description:	The upper and lower nibbles of register 'f' are exchanged. If 'd' is 0 the result is placed in W register. If 'd' is 1 the result is placed in register 'f'.
Words:	1
Cycles:	1
Example	SWAP F REG, 0

Before Instruction

            REG1   =   0xA5

After Instruction

            REG1   =   0xA5
            W   =   0x5A

TRIS	Load TRIS Register
Syntax:	[*label*] TRIS f
Operands:	$5 \leq f \leq 7$
Operation:	(W) $\rightarrow$ TRIS register (f)
Status Affected:	None

Encoding:

00	0000	0110	0fff

Description:	The instruction is supported for code compatibility with the PIC16C5X products. Since TRIS registers are readable and writable, the user can directly address them.
Words:	1
Cycles:	1
Example	

Note:	To maintain upward compatibility with future PIC16CXX products, do not use this instruction.

XORLW	Exclusive OR Literal with W
Syntax:	[*label*] XORLW k
Operands:	$0 \leq k \leq 255$
Operation:	(W) .XOR. k $\rightarrow$ (W)
Status Affected:	Z

Encoding:

11	1010	kkkk	kkkk

Description:	The contents of the W register are XOR'ed with the eight bit literal 'k'. The result is placed in the W register.
Words:	1
Cycles:	1
Example:	XORLW 0xAF

Before Instruction

            W   =   0xB5

After Instruction

            W   =   0x1A

XORWF	Exclusive OR W with f
Syntax:	[*label*] XORWF f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	(W) .XOR. (f) $\rightarrow$ (dest)
Status Affected:	Z

Encoding:

00	0110	dfff	ffff

Description:	Exclusive OR the contents of the W register with register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'.
Words:	1
Cycles:	1
Example	XORWF REG 1

Before Instruction

            REG   =   0xAF
            W   =   0xB5

After Instruction

            REG   =   0x1A
            W   =   0xB5

10.0 DEVELOPMENT SUPPORT

10.1 Development Tools

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

• PICMASTER™ Real-Time In-Circuit Emulator
• PRO MATE™ Universal Programmer
• PICSTART™ Low-Cost Prototype Programmer
• PICDEM-1 Low-Cost Demonstration Board
• PICDEM-2 Low-Cost Demonstration Board
• MPASM Assembler
• MPSIM Software Simulator
• C Compiler (MP-C)
• Fuzzy logic development system (*fuzzy*TECH®–MP)

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

The PICMASTER Universal In-Circuit Emulator (Figure 10-1) is intended to provide the product development engineer with a complete microcontroller design tool set for all microcontrollers in the PIC16C5X, PIC16CXX 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 PIC16C5X, PIC16CXX and PIC17CXX 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 better) machine platform and Microsoft® Windows® 3.x environment was chosen to best make these features available to you, the end user.

The PICMASTER Universal Emulator System consists primarily of four major components:

• Host-Interface Card
• Emulator Control Pod
• Target-Specific Emulator Probe
• PC-Host Emulation Control Software

The Windows operating system allows the developer to take full advantage of the many powerful features and functions of the PICMASTER system.
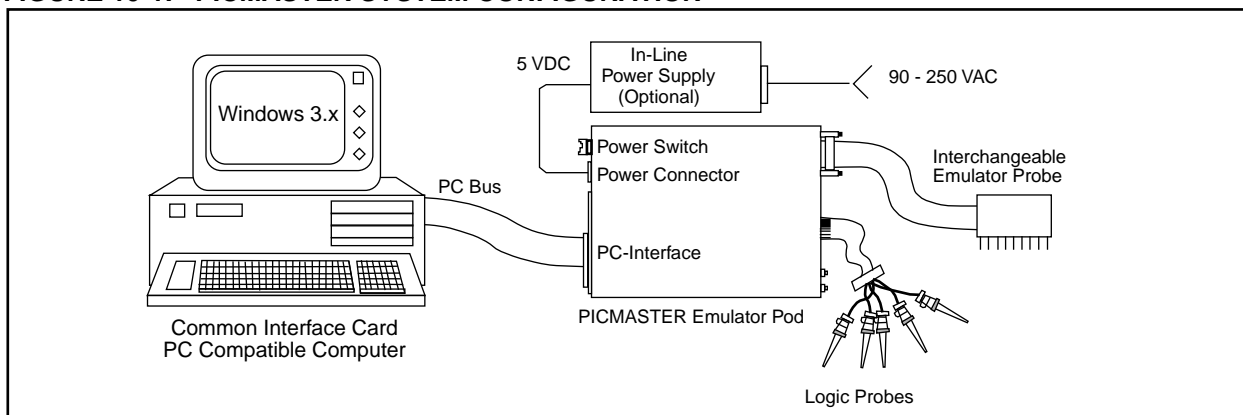
PICMASTER emulation can operate in one window, while a text editor is running in a second window.

PC-Host Emulation Control software takes full advantage of Dynamic Data Exchange (DDE), a feature of Windows. DDE allows data to be dynamically transferred between two or more Windows programs. With this feature, data collected with PICMASTER can be automatically transferred to a spreadsheet or database program for further analysis.

Under Windows, as many as four PICMASTER emulators can be run simultaneously from the same PC making development of multi-microcontroller systems possible (e.g., a system containing a PIC16CXX processor and a PIC17CXX processor).

The PICMASTER probes specifications are shown in Table 10-1.

FIGURE 10-1: PICMASTER SYSTEM CONFIGURATION

PIC16C8X

Devices	PICMASTER PROBE	PROBE	
		Maximum Frequency	Operating Voltage
PIC16C54	PROBE-16D	20 MHz	4.5V - 5.5V
PIC16C54A	PROBE-16D	20 MHz	4.5V - 5.5V
PIC16CR54	PROBE-16D	20 MHz	4.5V - 5.5V
PIC16CR54A	PROBE-16D[1]	20 MHz	4.5V - 5.5V
PIC16CR54B	PROBE-16D[1]	20 MHz	4.5V - 5.5V
PIC16C55	PROBE-16D	20 MHz	4.5V - 5.5V
PIC16CR55	PROBE-16D[1]	20 MHz	4.5V - 5.5V
PIC16C56	PROBE-16D	20 MHz	4.5V - 5.5V
PIC16CR56	PROBE-16D[1]	20 MHz	4.5V - 5.5V
PIC16C57	PROBE-16D	20 MHz	4.5V - 5.5V
PIC16CR57A	PROBE-16D	20 MHz	4.5V - 5.5V
PIC16CR57B	PROBE-16D[1]	20 MHz	4.5V - 5.5V
PIC16C58A	PROBE-16D	20 MHz	4.5V - 5.5V
PIC16CR58A	PROBE-16D	20 MHz	4.5V - 5.5V
PIC16CR58B	PROBE-16D[1]	20 MHz	4.5V - 5.5V
PIC16C61	PROBE-16G	10 MHz	4.5V - 5.5V
PIC16C62	PROBE-16E	10 MHz	4.5V - 5.5V
PIC16C62A	PROBE-16E[1]	10 MHz	4.5V - 5.5V
PIC16CR62	PROBE-16E[1]	10 MHz	4.5V - 5.5V
PIC16C63	PROBE-16F[1]	10 MHz	4.5V - 5.5V
PIC16C64	PROBE-16E	10 MHz	4.5V - 5.5V
PIC16C64A	PROBE-16E[1]	10 MHz	4.5V - 5.5V
PIC16CR64	PROBE-16E[1]	10 MHz	4.5V - 5.5V

TABLE 10-1:   PICMASTER PROBE SPECIFICATION

Devices	PICMASTER PROBE	PROBE	
		Maximum Frequency	Operating Voltage
PIC16C65	PROBE-16F	10 MHz	4.5V - 5.5V
PIC16C65A	PROBE-16F[1]	10 MHz	4.5V - 5.5V
PIC16C620	PROBE-16H	10 MHz	4.5V - 5.5V
PIC16C621	PROBE-16H	10 MHz	4.5V - 5.5V
PIC16C622	PROBE-16H	10 MHz	4.5V - 5.5V
PIC16C70	PROBE-16B[1]	10 MHz	4.5V - 5.5V
PIC16C71	PROBE-16B	10 MHz	4.5V - 5.5V
PIC16C71A	PROBE-16B[1]	10 MHz	4.5V - 5.5V
PIC16C72	PROBE-16F[1]	10 MHz	4.5V - 5.5V
PIC16C73	PROBE-16F	10 MHz	4.5V - 5.5V
PIC16C73A	PROBE-16F[1]	10 MHz	4.5V - 5.5V
PIC16C74	PROBE-16F	10 MHz	4.5V - 5.5V
PIC16C74A	PROBE-16F[1]	10 MHz	4.5V - 5.5V
PIC16C83	PROBE-16C	10 MHz	4.5V - 5.5V
PIC16C84	PROBE-16C	10 MHz	4.5V - 5.5V
PIC17C42	PROBE-17B	20 MHz	4.5V - 5.5V
PIC17C43	PROBE-17B	20 MHz	4.5V - 5.5V
PIC17C44	PROBE-17B	20 MHz	4.5V - 5.5V

Note 1:   This PICMASTER probe can be used to functionally emulate the device listed in the previous column. Contact your Microchip sales office for details.

10.3 PRO MATE: Universal Programmer

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

The PRO MATE has programmable V_{DD} and V_{PP} supplies which allows it to verify programmed memory at V_{DD} min and V_{DD} 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 stand- alone mode the PRO MATE can read, verify or program PIC16C5X, PIC16CXX and PIC17CXX devices. It can also set configuration and code-protect bits in this mode.

In PC-hosted mode, the PRO MATE connects to the PC via one of the COM (RS-232) ports. PC based user-interface software makes using the programmer simple and efficient. The user interface is full-screen and menu-based. Full screen display and editing of data, easy selection of bit configuration and part type, easy selection of V_{DD} min, V_{DD} max and V_{PP} levels, load and store to and from disk files (Intel® hex format) are some of the features of the software. Essential commands such as read, verify, program and blank check can be issued from the screen. Additionally, serial programming support is possible where each part is programmed with a different serial number, sequential or random.

The PRO MATE has a modular "programming socket module". Different socket modules are required for different processor types and/or package types.

PRO MATE supports all PIC16C5X, PIC16CXX and PIC17CXX processors.

10.4 PICSTART Low-Cost Development System

The PICSTART programmer is an easy to use, very low-cost prototype programmer. It connects to the PC via one of the COM (RS-232) ports. A PC-based user interface software makes using the programmer simple and efficient. The user interface is full-screen and menu-based. PICSTART is not recommended for production programming.

10.5 PICDEM-1 Low-Cost PIC16/17 Demonstration Board

The PICDEM-1 is a simple board which demonstrates the capabilities of several of Microchip's microcontrollers. The microcontrollers supported are: PIC16C5X (PIC16C54 to PIC16C58A), PIC16C61, PIC16C62X, PIC16C71, PIC16C8X, PIC17C42, PIC17C43 and PIC17C44. All necessary hardware and software is included to run basic demo programs. The users can program the sample microcontrollers provided with the PICDEM-1 board, on a PRO MATE 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.

10.6 PICDEM-2 Low-Cost PIC16CXX Demonstration Board

The PICDEM-2 is a simple demonstration board that supports the 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 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^2C bus and separate headers for connection to an LCD module and a keypad.

PIC16C8X

10.7   MPLAB Integrated Development Environment Software

The MPLAB 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 (available soon)
• 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 (available soon) 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.

10.8   Assembler (MPASM)

The MPASM Cross Assembler is a PC-hosted symbolic assembler. It supports all microcontroller series including the PIC16C5X, PIC16CXX, 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 allows 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.

• **Data Directives** are those that control the allocation of memory and provide a way to refer to data items symbolically (i.e., by meaningful names).
• **Control Directives** control the MPASM listing display. They allow the specification of titles and sub-titles, page ejects and other listing control. This eases the readability of the printed output file.
• **Conditional Directives** permit sections of conditionally assembled code. This is most useful where additional functionality may wished to be added depending on the product (less functionality for the low end product, then for the high end product). Also this is very helpful in the debugging of a program.
• **Macro Directives** control the execution and data allocation within macro body definitions. This makes very simple the re-use of functions in a program as well as between programs.

10.9   Software Simulator (MPSIM)

The MPSIM 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. MPSIM fully supports symbolic debugging using MP-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.

10.10   C Compiler (MP-C)

The MP-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 PICMASTER Universal Emulator memory display (PICMASTER emulator software versions 1.13 and later).

The MP-C Code Development System is supplied directly by Byte Craft Limited of Waterloo, Ontario, Canada. If you have any questions, please contact your regional Microchip FAE or Microchip technical support personnel at (602) 786-7627.

10.11   Fuzzy Logic Development System (*fuzzy*TECH-MP)

*fuzzy*TECH-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, *fuzzy*TECH-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.

10.12   Development Systems

For convenience, the development tools are packaged into comprehensive systems as listed in Table 10-2.

TABLE 10-2:   DEVELOPMENT SYSTEM PACKAGES

Item	Name	System Description
1.	PICMASTER System	PICMASTER In-Circuit Emulator, PRO MATE Programmer, Assembler, Software Simulator, Samples and your choice of Target Probe.
2.	PICSTART System	PICSTART Low-Cost Prototype Programmer, Assembler, Software Simulator and Samples.
3.	PRO MATE System	PRO MATE Universal Programmer, full featured stand-alone or PC-hosted programmer, Assembler, Simulator

NOTES:

11.0   ELECTRICAL CHARACTERISTICS FOR PIC16C84

Absolute Maximum Ratings †

Ambient temperature under bias...-55°C to +125°C

Storage temperature .... -65°C to +150°C

Voltage on V_{DD} with respect to V_{SS} .... 0 to +7.5V

Voltage on $\overline{MCLR}$ with respect to V_{SS} (Note 2)..0 to +14V

Voltage on all other pins with respect to V_{SS} .... -0.6V to (V_{DD} + 0.6V)

Total power dissipation (Note 1)..800 mW

Maximum current out of V_{SS} pin ....150 mA

Maximum current into V_{DD} pin ....100 mA

Input clamp current, I_{IK} (V_I < 0 or V_I > V_{DD}) ....±20 mA

Output clamp current, I_{OK} ($V0$ < 0 or $V0$ >V_{DD}) ....±20 mA

Maximum output current sunk by any I/O pin...25 mA

Maximum output current sourced by any I/O pin ....20 mA

Maximum current sunk by PORTA ....80 mA

Maximum current sourced by PORTA ....50 mA

Maximum current sunk by PORTB...150 mA

Maximum current sourced by PORTB...100 mA

Note 1: Power dissipation is calculated as follows: Pdis = V_{DD} x {I_{DD} - $\sum I_{OH}$} + $\sum$ {(V_{DD}-V_{OH}) x I_{OH}} + $\sum$(V_{Ol} x I_{OL})

Note 2: Voltage spikes below V_{SS} at the $\overline{MCLR}$ pin, inducing currents greater than 80 mA, may cause latch-up. Thus, a series resistor of 50-100Ω should be used when applying a "low" level to the $\overline{MCLR}$ pin rather than pulling this pin directly to V_{SS}.

† 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.

TABLE 11-1:   CROSS REFERENCE OF DEVICE SPECS FOR OSCILLATOR CONFIGURATIONS AND FREQUENCIES OF OPERATION (COMMERCIAL DEVICES)

OSC	PIC16C84-04	PIC16C84-10	PIC16LC84-04
RC	V_{DD}: 4.0V to 6.0V I_{DD}: 4.5 mA max. at 5.5V I_{PD}: 100 µA max. at 4.0V WDT dis Freq: 4.0 MHz max.	V_{DD}: 4.5V to 5.5V I_{DD}: 1.8 mA typ. at 5.5V I_{PD}: 40.0 µA typ. at 4.5V WDT dis Freq: 4.0 MHz max.	V_{DD}: 2.0V to 6.0V I_{DD}: 4.5 mA typ. at 5.5V I_{PD}: 100 µA typ. at 4V WDT dis Freq: 2.0 MHz max.
XT	V_{DD}: 4.0V to 6.0V I_{DD}: 4.5 mA max. at 5.5V I_{PD}: 100 µA max. at 4.0V WDT dis Freq: 4.0 MHz max.	V_{DD}: 4.5V to 5.5V I_{DD}: 1.8 mA typ. at 5.5V I_{PD}: 40.0 µA typ. at 4.5V WDT dis Freq: 4.0 MHz max.	V_{DD}: 2.0V to 6.0V I_{DD}: 4.5 mA typ. at 5.5V I_{PD}: 100 µA typ. at 4V WDT dis Freq: 2.0 MHz max.
HS	V_{DD}: 4.5V to 5.5V I_{DD}: 4.5 mA typ. at 5.5V I_{PD}: 40.0 µA typ. at 4.5V WDT dis Freq: 4.0 MHz max.	V_{DD}: 4.5V to 5.5V I_{DD}: 10 mA max. at 5.5V typ. I_{PD}: 40.0 µA typ. at 4.5V WDT dis Freq: 10 MHz max.	Do not use in HS mode
LP	V_{DD}: 4.0V to 6.0V I_{DD}: 60 µA typ. at 32 kHz, 2.0V I_{PD}: 26 µA typ. at 2.0V WDT dis Freq: 200 kHz max.	Do not use in LP mode	V_{DD}: 2.0V to 6.0V I_{DD}: 400 µA max. at 32 kHz, 2.0V I_{PD}: 100 µA max. at 4.0V WDT dis Freq: 200 kHz 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.

PIC16C8X

Applicable Devices 83 R83 84 84A R84

11.1   DC CHARACTERISTICS: PIC16C84-04 (Commercial, Industrial)
PIC16C84-10 (Commercial, Industrial)

DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial and 0°C ≤ TA ≤ +70°C for commercial				
Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
D001 D001A	VDD	Supply Voltage	4.0 4.5	— —	6.0 5.5	V V	XT, RC and LP osc configuration HS osc configuration
D002	VDR	RAM Data Retention Voltage (Note 1)	1.5*	—	—	V	Device in SLEEP mode
D003	VPOR	VDD start voltage to ensure Power-on Reset	—	VSS	—	V	See section on Power-on Reset for details
D004	SVDD	VDD rise rate to ensure Power-on Reset	0.05*	—	—	V/ms	See section on Power-on Reset for details
D010 D010A D013	IDD	Supply Current (Note 2)	— — —	1.8 7.3 5.0	4.5 10 10	mA mA mA	RC and XT osc configuration (Note 4) FOSC = 4 MHz, VDD = 5.5V FOSC = 4 MHz, VDD = 5.5V (During EEPROM programming) HS osc configuration (PIC16C84-10) FOSC = 10 MHz, VDD = 5.5V
D020 D021 D021A	IPD	Power-down Current (Note 3)	— — —	40 38 38	100 100 100	μA μA μA	VDD = 4.0V, WDT enabled, -40°C to +85°C VDD = 4.0V, WDT disabled, -0°C to +70°C VDD = 4.0V, WDT disabled, -40°C to +85°C

*   These parameters are characterized but not tested.

†   Data in "Typ" column is at 5.0V, 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 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, T0CKI = VDD,
   $\overline{MCLR}$ = VDD; WDT enabled/disabled as specified.

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

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

11.2   DC CHARACTERISTICS: PIC16LC84-04 (Commercial, Industrial)

DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial and 0°C ≤ TA ≤ +70°C for commercial					
Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
D001	VDD	Supply Voltage	2.0	—	6.0	V	XT, RC, and LP osc configuration
D002	VDR	RAM Data Retention Voltage (Note 1)	1.5 *	—	—	V	Device in SLEEP mode
D003	VPOR	VDD start voltage to ensure Power-on Reset	—	VSS	—	V	See section on Power-on Reset for details
D004	SVDD	VDD rise rate to ensure Power-on Reset	0.05*	—	—	V/ms	See section on Power-on Reset for details
D010 D010A D014	IDD	Supply Current (Note 2)	— — —	1.8 7.3 60	4.5 10 400	mA mA µA	RC and XT osc configuration (Note 4) FOSC = 2 MHz, VDD = 5.5V FOSC = 2 MHz, VDD = 5.5V (During EEPROM programming) LP osc configuration FOSC = 32 kHz, VDD = 2.0V WDT disabled
D020 D021 D021A	IPD	Power-down Current (Note 3)	— — —	26 26 26	100 100 100	µA µA µA	VDD = 2.0V, WDT enabled, -40°C to +85°C VDD = 2.0V, WDT disabled, 0°C to +70°C VDD = 2.0V, WDT disabled, -40°C to +85°C

*   These parameters are characterized but not tested.

†   Data in "Typ" column is at 5.0V, 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, T0CKI = VDD,
$\overline{MCLR}$ = VDD; WDT enabled/disabled as specified.

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

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

PIC16C8X

11.3   DC CHARACTERISTICS: **PIC16C84-04   (Commercial, Industrial)**
PIC16C84-10   (Commercial, Industrial)
PIC16LC84-04 (Commercial, Industrial)

DC CHARACTERISTICS			**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial and 0°C ≤ TA ≤ +70°C for commercial Operating voltage VDD range as described in DC spec Section 11-1 and Section 11-3.				
Parameter No.	**Sym**	**Characteristic**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
	VIL	**Input Low Voltage** I/O ports					
D030		with TTL buffer	Vss	—	0.8	V	4.5 ≤ VDD ≤ 5.5V
D030A			Vss	—	0.16VDD	V	entire range (Note 4)
D031		with Schmitt Trigger buffer	Vss	—	0.2VDD	V	entire range
D032		MCLR, RA4/T0CKI, OSC1 (RC mode)	Vss	—	0.2VDD	V	
D033		OSC1 (XT, HS and LP modes)	Vss	—	0.3VDD	V	Note1
	VIH	**Input High Voltage** I/O ports					
D040		with TTL buffer	0.36VDD	—	VDD	V	4.5 ≤ VDD ≤ 5.5V
D040A			0.48VDD				entire range (Note 4)
D041		with Schmitt Trigger buffer	0.45VDD	—	VDD		entire range
D042		MCLR, RA4/T0CKI, OSC1 (RC mode)	0.85VDD	—	VDD	V	
D043		OSC1 (XT, HS and LP modes)	0.7VDD	—	VDD	V	Note1
D050	VHYS	Hysteresis of Schmitt Trigger inputs	TBD	—	—	V	
D070	IPURB	PORTB weak pull-up current	50*	250*	400*	μA	VDD = 5V, VPIN = VSS
	IIL	**Input Leakage Current** (Notes 2, 3) I/O ports					
D060		I/O ports	—	—	±1	μA	Vss ≤ VPIN ≤ VDD, Pin at hi-impedance
D061		MCLR, RA4/T0CKI	—	—	±5	μA	Vss ≤ VPIN ≤ VDD
D063		OSC1/CLKIN	—	—	±5	μA	Vss ≤ VPIN ≤ VDD, XT, HS and LP osc configuration
	VOL	**Output Low Voltage** I/O ports					
D080		I/O ports	—	—	0.6	V	IOL = 8.5 mA, VDD = 4.5V
D083		OSC2/CLKOUT (RC osc configuration)	—	—	0.6	V	IOL = 1.6 mA, VDD = 4.5V
	VOH	**Output High Voltage**					
D090		I/O ports (Note 3)	VDD - 0.7	—	—	V	IOH = -3.0 mA, VDD = 4.5V
D093		OSC2/CLKOUT (RC osc configuration)	VDD - 0.7	—	—	V	IOH = -1.3 mA, VDD = 4.5V

*   These parameters are characterized but not tested.

†   Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: In RC oscillator configuration, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended that the PIC16C84 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: The user may use better of the two specs.

11.4   DC CHARACTERISTICS: PIC16C84-04   (Commercial, Industrial)
PIC16C84-10   (Commercial, Industrial)
PIC16LC84-04 (Commercial, Industrial)

DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise stated) Operating temperature $-40°C \leq T_A \leq +85°C$ for industrial and $0°C \leq T_A \leq +70°C$ for commercial Operating voltage V_{DD} range as described in DC spec Section 11-1 and Section 11-3.					
Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions	
		Capacitive Loading Specs on Output Pins						
D100	C_{OSC2}	OSC2/CLKOUT pin	—	—	15	pF	In XT, HS and LP modes when external clock is used to drive OSC1.	
D101	C_{IO}	All I/O pins and OSC2 (RC mode)	—	—	50	pF		
		Data EEPROM Memory						
D120	E_D	Endurance	100,000	1,000,000	—	E/W		
D121	V_{DRW}	VDD for read/write	V_{MIN}	—	6.0	V	V_{MIN} = Minimum operating voltage	
D122	T_{DEW}	Erase/Write cycle time	—	10	—	ms	Note1	
		Program EEPROM Memory						
D130	E_P	Endurance	100	—	—	E/W		
D131	V_{PR}	VDD for read	V_{MIN}	—	6.0	V	V_{MIN} = Minimum operating voltage	
D132	V_{PEW}	VDD for erase/write	4.5	—	5.5	V		
D133	T_{PEW}	Erase/Write cycle time	—	10	—	ms	Note1	

†   Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1:   The user should use interrupts or poll the EEIF or WR bits to ensure the write cycle has completed.

PIC16C8X

TABLE 11-2: TIMING PARAMETER SYMBOLOGY

The timing parameter symbols have been created following one of the following formats:

1. TppS2ppS
2. TppS

T			
F	Frequency	T	Time

Lowercase symbols (pp) and their meanings:

pp			
ck	CLKOUT	osc	OSC1
io	I/O port	t0	T0CKI
mc	$\overline{\text{MCLR}}$		

Uppercase symbols and their meanings:

S			
F	Fall	P	Period
H	High	R	Rise
I	Invalid (Hi-impedance)	V	Valid
L	Low	Z	Hi-impedance

FIGURE 11-1: PARAMETER MEASUREMENT INFORMATION

0.7 V$_{DD}$ XTAL
0.8 V$_{DD}$ RC (High)

0.3 V$_{DD}$ XTAL
0.15 V$_{DD}$ RC (Low)

OSC1 Measurement Points

2.0 V$_{DD}$ (High)

0.2 V$_{DD}$ (Low)

I/O Port Measurement Points

All timings are measured between high and low measurement points as indicated in the figure.
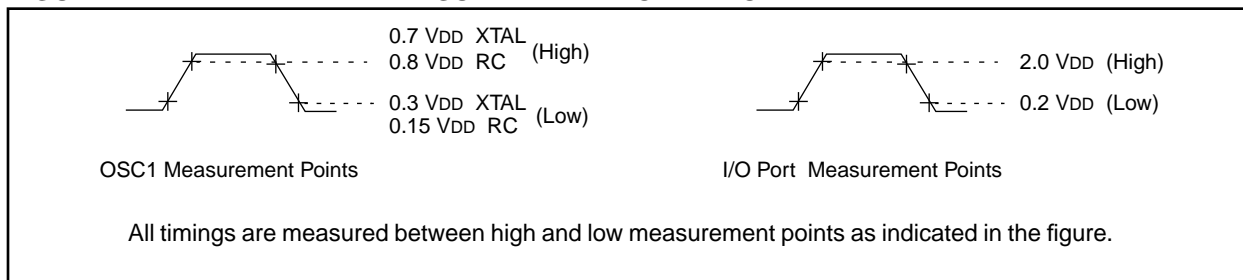
FIGURE 11-2: LOAD CONDITIONS

Load Condition 1

V$_{DD}$/2

R$_L$

Pin

C$_L$

V$_{SS}$

Load Condition 2

Pin

C$_L$

V$_{SS}$

R$_L$ =   464Ω

C$_L$ =   50 pF   for all pins except OSC2.

          15 pF   for OSC2 output.

PIC16C8X
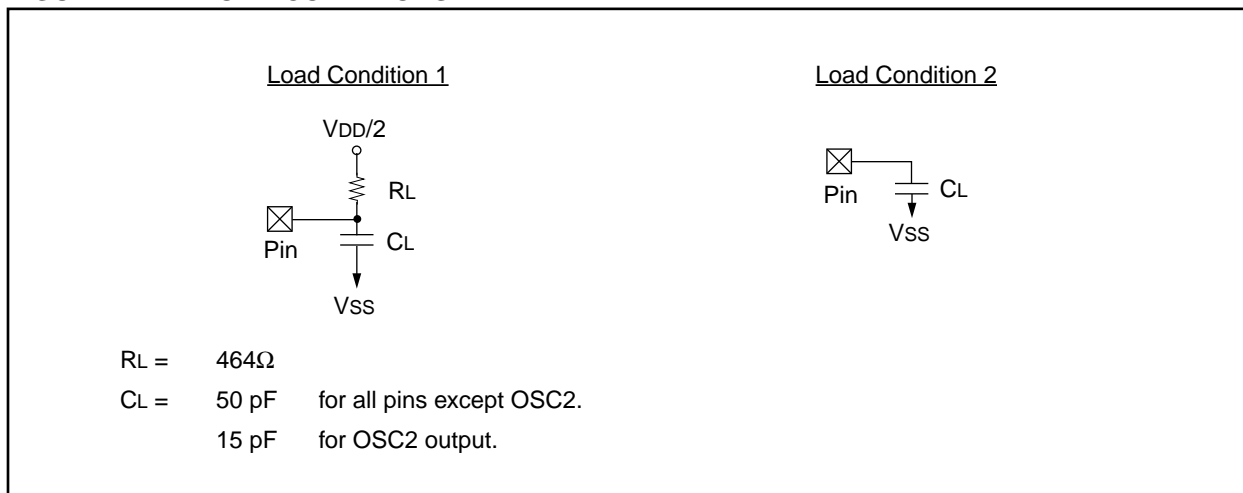
11.5   Timing Diagrams and Specifications

FIGURE 11-3:   EXTERNAL CLOCK TIMING
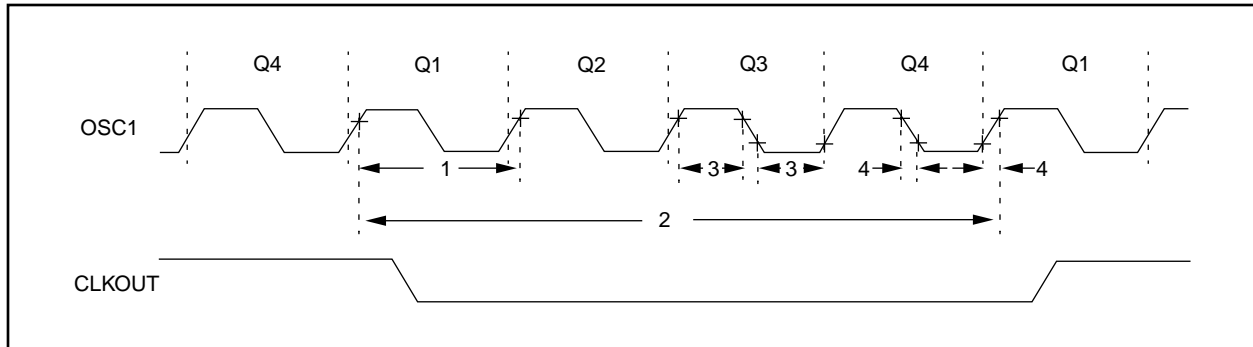
TABLE 11-3:   EXTERNAL CLOCK TIMING REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions	
	Fos	**External CLKIN Frequency** (Note 1)	DC	—	2	MHz	XT, RC osc	PIC16LC84-04
			DC	—	4	MHz	XT, RC osc	PIC16C84-04
			DC	—	10	MHz	HS osc	PIC16C84-10
			DC	—	200	kHz	LP osc	PIC16LC84-04
		Oscillator Frequency (Note 1)	DC	—	2	MHz	RC osc	PIC16LC84-04
			DC	—	4	MHz	RC osc	PIC16C84-04
			0.1	—	2	MHz	XT osc	PIC16LC84-04
			0.1	—	4	MHz	XT osc	PIC16C84-04
			1	—	10	MHz	HS osc	PIC16C84-10
			DC	—	200	kHz	LP osc	PIC16LC84-04
1	TOSC	**External CLKIN Period** (Note 1)	500	—	—	ns	XT, RC osc	PIC16LC84-04
			250	—	—	ns	XT, RC osc	PIC16C84-04
			100	—	—	ns	HS osc	PIC16C84-10
			5	—	—	μs	LP osc	PIC16LC84-04
		Oscillator Period (Note 1)	500	—	—	ns	RC osc	PIC16LC84-04
			250	—	—	ns	RC osc	PIC16C84-04
			500	—	10,000	ns	XT osc	PIC16LC84-04
			250	—	10,000	ns	XT osc	PIC16C84-04
			100	—	1,000	ns	HS osc	PIC16C84-10
			5	—	—	μs	LP osc	PIC16LC84-04
2	TCY	**Instruction Cycle Time (Note 1)**	0.4	4/Fosc	DC	μs		
3	TosL, TosH	Clock in (OSC1) High or Low Time	60 *	—	—	ns	XT osc	PIC16LC84-04
			50 *	—	—	ns	XT osc	PIC16C84-04
			2 *	—	—	μs	LP osc	PIC16LC84-04
			35 *	—	—	ns	HS osc	PIC16C84-10
4	TosR, TosF	Clock in (OSC1) Rise or Fall Time	25 *	—	—	ns	XT osc	PIC16C84-04
			50 *	—	—	ns	LP osc	PIC16LC84-04
			15 *	—	—	ns	HS osc	PIC16C84-10

†   Data in "Typ" column is at 5.0V, 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 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 pin. When an external clock input is used, the "Max." cycle time limit is "DC" (no clock) for all devices.

PIC16C8X

FIGURE 11-4:   CLKOUT AND I/O TIMING

Note: All tests must be done with specified capacitive loads (Figure 11-2) 50 pF on I/O pins and CLKOUT.

TABLE 11-4:   CLKOUT AND I/O TIMING REQUIREMENTS

Parameter No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
10	TosH2ckL	OSC1↑ to CLKOUT↓	PIC16C84	—	15	30 *	ns	Note 1
10A			PIC16LC84	—	15	120 *	ns	Note 1
11	TosH2ckH	OSC1↑ to CLKOUT↑	PIC16C84	—	15	30 *	ns	Note 1
11A			PIC16LC84	—	15	120 *	ns	Note 1
12	TckR	CLKOUT rise time	PIC16C84	—	15	30 *	ns	Note 1
12A			PIC16LC84	—	15	100 *	ns	Note 1
13	TckF	CLKOUT fall time	PIC16C84	—	15	30 *	ns	Note 1
13A			PIC16LC84	—	15	100 *	ns	Note 1
14	TckL2ioV	CLKOUT ↓ to Port out valid		—	—	0.5TCY +20 *	ns	Note 1
15	TioV2ckH	Port in valid before	PIC16C84	0.30TCY + 30 *	—	—	ns	Note 1
		CLKOUT ↑	PIC16LC84	0.30TCY + 80 *	—	—	ns	Note 1
16	TckH2ioI	Port in hold after CLKOUT ↑		0 *	—	—	ns	Note 1
17	TosH2ioV	OSC1↑ (Q1 cycle) to	PIC16C84	—	—	125 *	ns	
		Port out valid	PIC16LC84	—	—	250 *	ns	
18	TosH2ioI	OSC1↑ (Q2 cycle) to Port input invalid (I/O in hold time)		TBD	—	—	ns	
19	TioV2osH	Port input valid to OSC1↑ (I/O in setup time)		TBD	—	—	ns	
20	TioR	Port output rise time	PIC16C84	—	10	25 *	ns	
20A			PIC16LC84	—	10	60 *	ns	
21	TioF	Port output fall time	PIC16C84	—	10	25 *	ns	
21A			PIC16LC84	—	10	60 *	ns	
22	Tinp	INT pin high	PIC16C84	20 *	—	—	ns	
22A		or low time	PIC16LC84	55 *	—	—	ns	
23	Trbp	RB7:RB4 change INT	PIC16C84	20 *	—	—	ns	
23A		high or low time	PIC16LC84	55 *	—	—	ns	

*   These parameters are characterized but not tested.

†   Data in "Typ" column is at 5.0V, 25˚C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1:   Measurements are taken in RC Mode where CLKOUT output is 4 x TOSC.

© 1995 Microchip Technology Inc.

FIGURE 11-5: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING
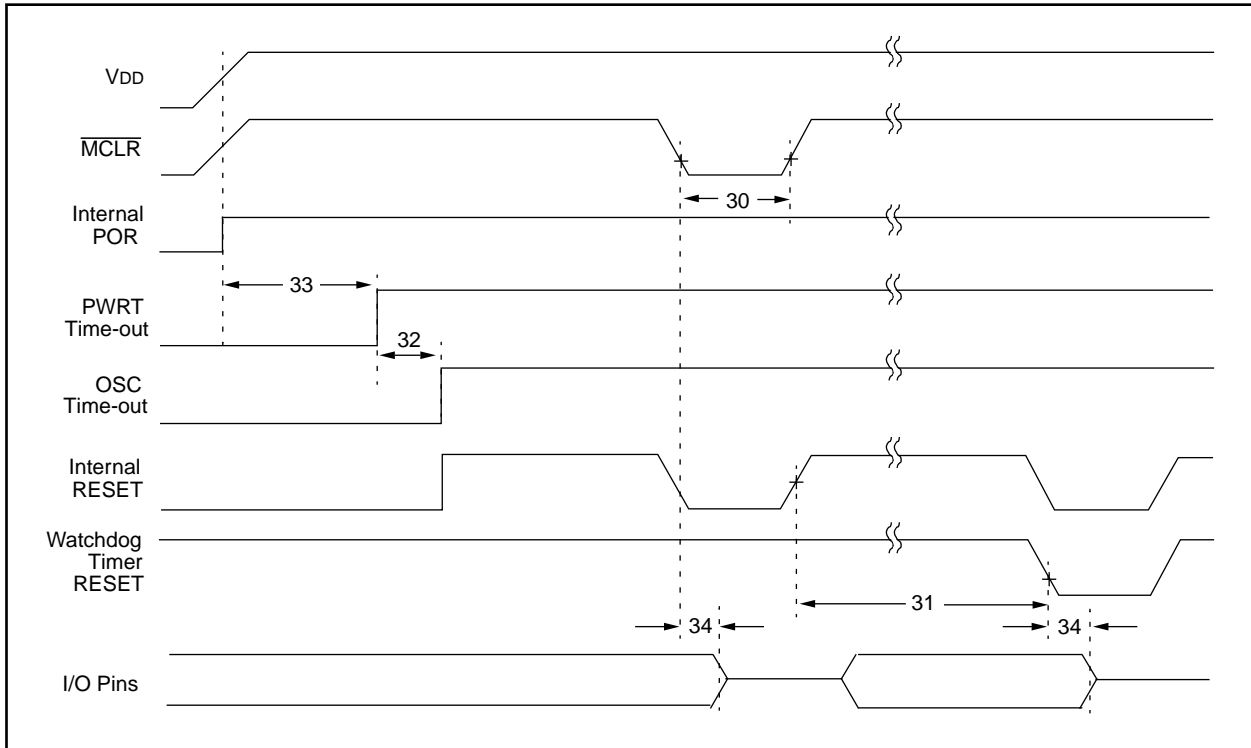
TABLE 11-5: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
30	TmcL	$\overline{MCLR}$ Pulse Width (low)	350 *	—	—	ns	$2.0V \leq V_{DD} \leq 3.0V$
			150 *	—	—	ns	$3.0V \leq V_{DD} \leq 6.0V$
31	Twdt	Watchdog Timer Time-out Period (No Prescaler)	7 *	18	33 *	ms	$V_{DD} = 5V$
32	Tost	Oscillation Start-up Timer Period	—	$1024T_{OSC}$	—	ms	T_{OSC} = OSC1 period
33	Tpwrt	Power-up Timer Period	28 *	72	132 *	ms	$V_{DD} = 5.0V$
34	TIOZ	I/O Hi-impedance from $\overline{MCLR}$ Low or reset	—	—	100 *	ns	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

PIC16C8X

FIGURE 11-6:   TIMER0 CLOCK TIMINGS

TABLE 11-6:   TIMER0 CLOCK REQUIREMENTS

Parameter No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
40	Tt0H	T0CKI High Pulse Width	No Prescaler	$0.5T_{CY} + 20$ *	—	—	ns	
			With Prescaler	50 *	—	—	ns	$2.0V \leq V_{DD} \leq 3.0V$
				30 *	—	—	ns	$3.0V \leq V_{DD} \leq 6.0V$
41	Tt0L	T0CKI Low Pulse Width	No Prescaler	$0.5T_{CY} + 20$ *	—	—	ns	
			With Prescaler	50 *	—	—	ns	$2.0V \leq V_{DD} \leq 3.0V$
				20 *	—	—	ns	$3.0V \leq V_{DD} \leq 6.0V$
42	Tt0P	T0CKI Period		$\dfrac{T_{CY} + 40}{N}$ *	—	—	ns	N = prescale value (2, 4, ..., 256)

*   These parameters are characterized but not tested.

†   Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

12.0   DC & AC CHARACTERISTICS GRAPHS/TABLES FOR PIC16C84

The data 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 is outside specified operating range (e.g., outside specified V_{DD} range). This is for information only and devices are guaranteed 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 + 3σ) and (mean - 3σ) respectively where σ is standard deviation.

FIGURE 12-1:   TYPICAL RC OSCILLATOR FREQUENCY vs. TEMPERATURE
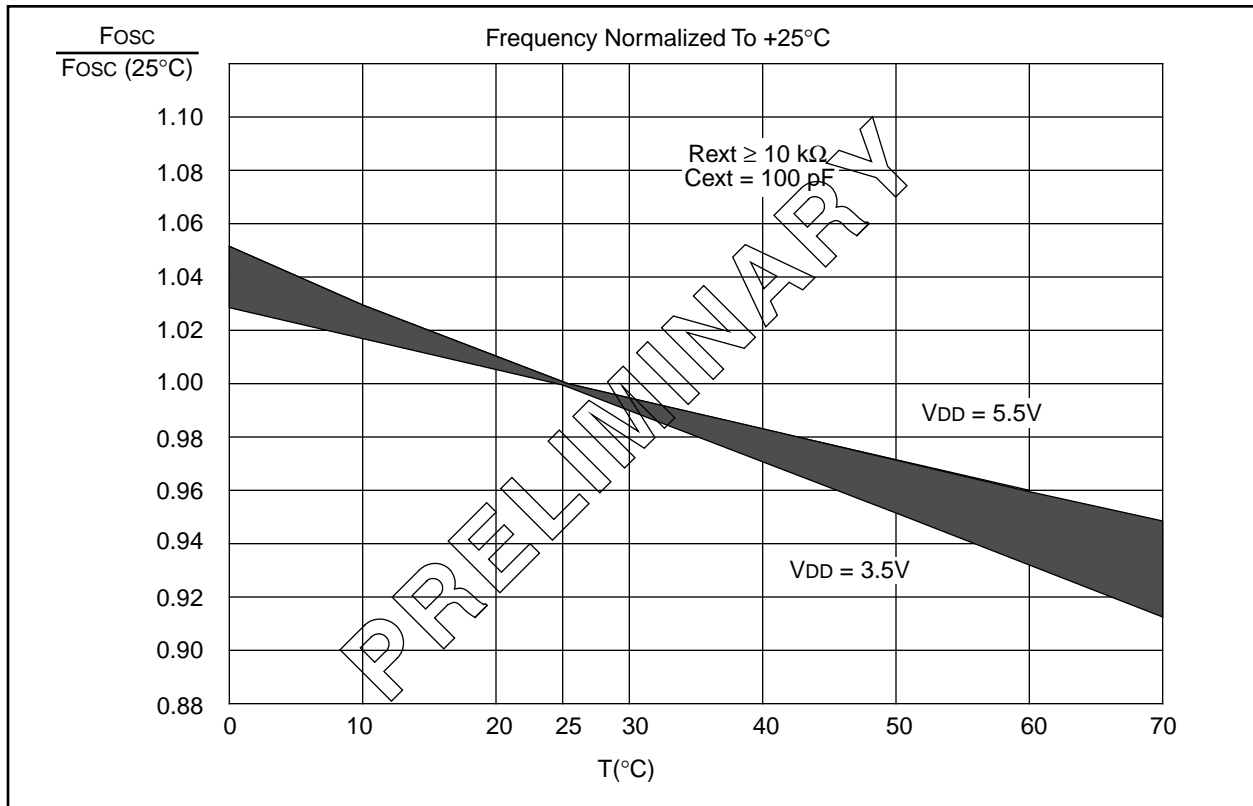
TABLE 12-1:   RC OSCILLATOR FREQUENCIES *

Cext	Rext	Average Fosc @ 5V, 25°C	
20 pF	3.3k	4.68 MHz	± 27%
	5.1k	3.94 MHz	± 25%
	10k	2.34 MHz	± 29%
	100k	250.16 kHz	± 33%
100 pF	3.3k	1.49 MHz	± 25%
	5.1k	1.12 MHz	± 25%
	10k	620.31 kHz	± 30%
	100k	90.25 kHz	± 26%
300 pF	3.3k	524.24 kHz	± 28%
	5.1k	415.52 kHz	± 30%
	10k	270.33 kHz	± 26%
	100k	25.37 kHz	± 25%

*Measured in PDIP Packages. The percentage variation indicated here is part to part variation due to normal process distribution. The variation indicated is ±3 standard deviation from average value.

PIC16C8X

FIGURE 12-2: TYPICAL RC OSCILLATOR FREQUENCY vs. V$_{DD}$
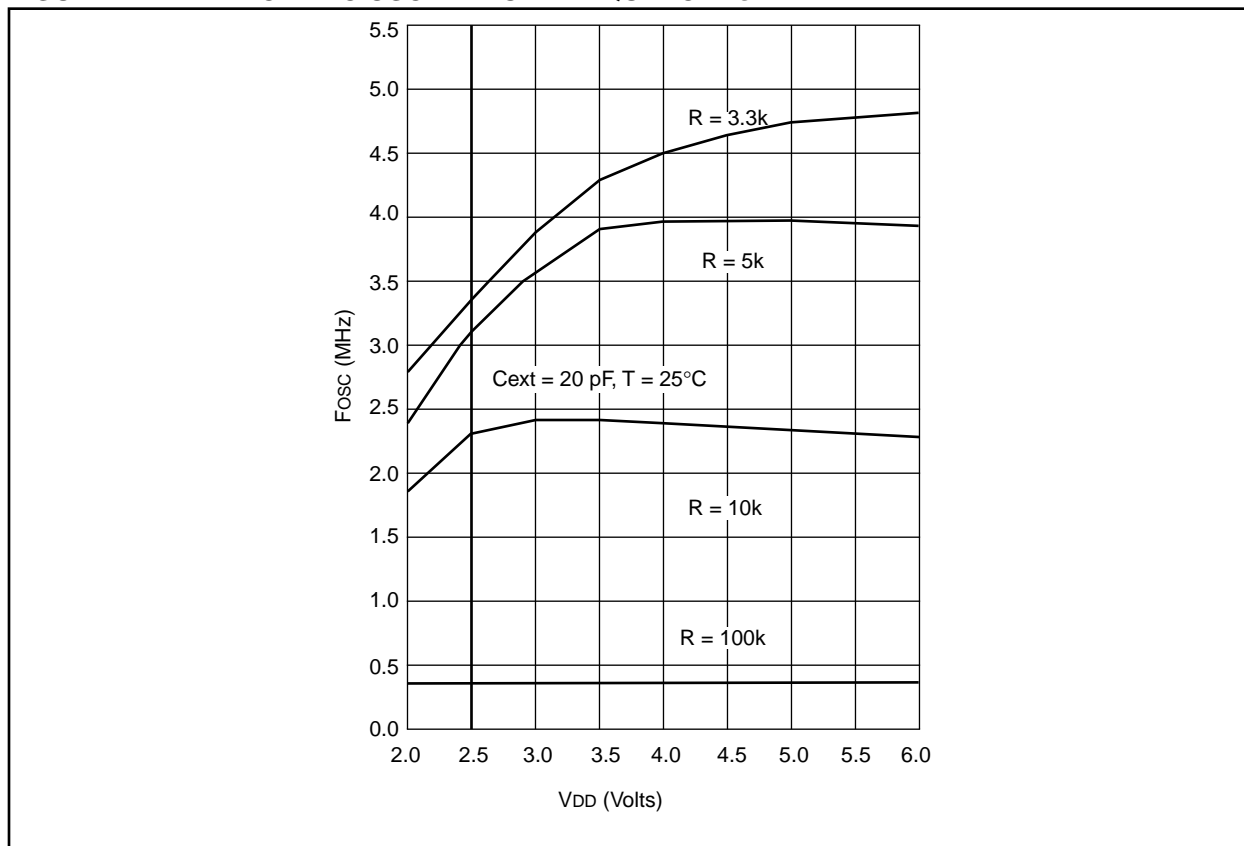
FIGURE 12-3: TYPICAL RC OSCILLATOR FREQUENCY vs. VDD
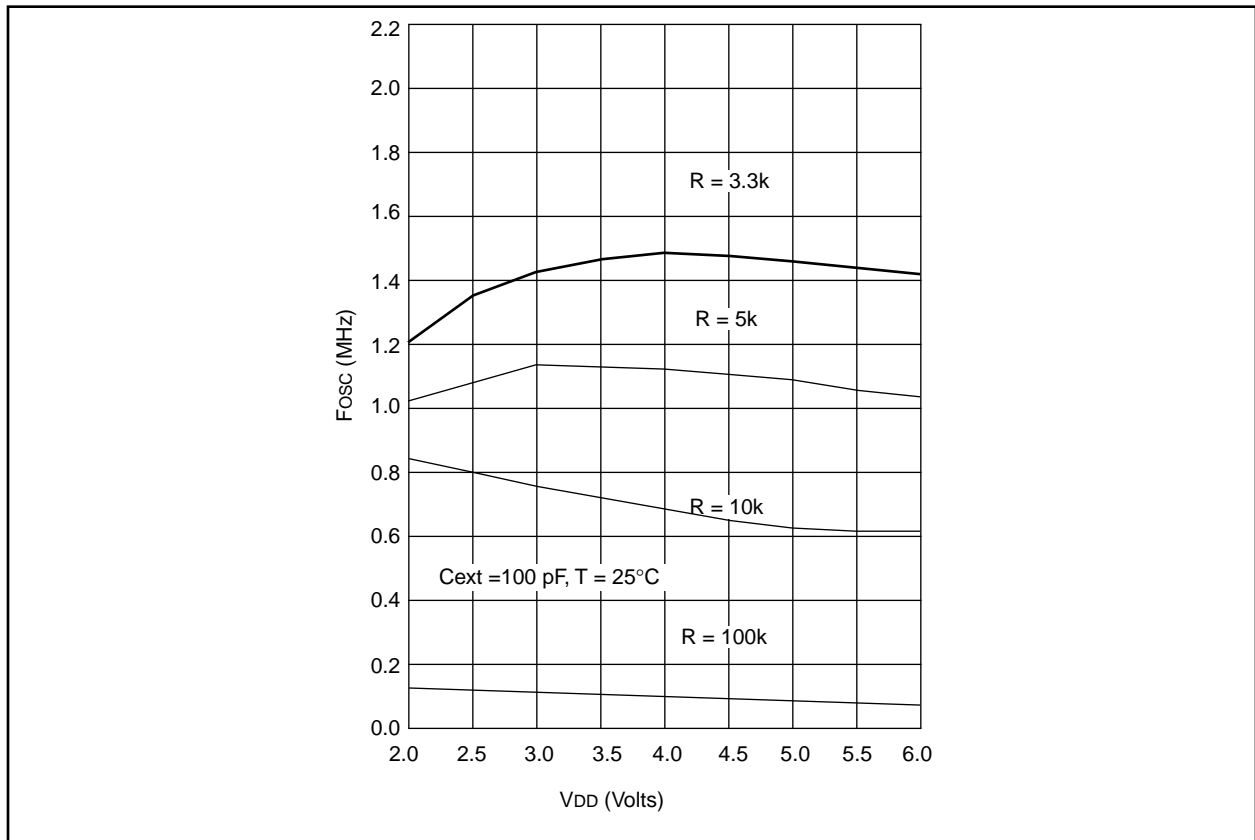
FIGURE 12-4: TYPICAL RC OSCILLATOR FREQUENCY vs. VDD
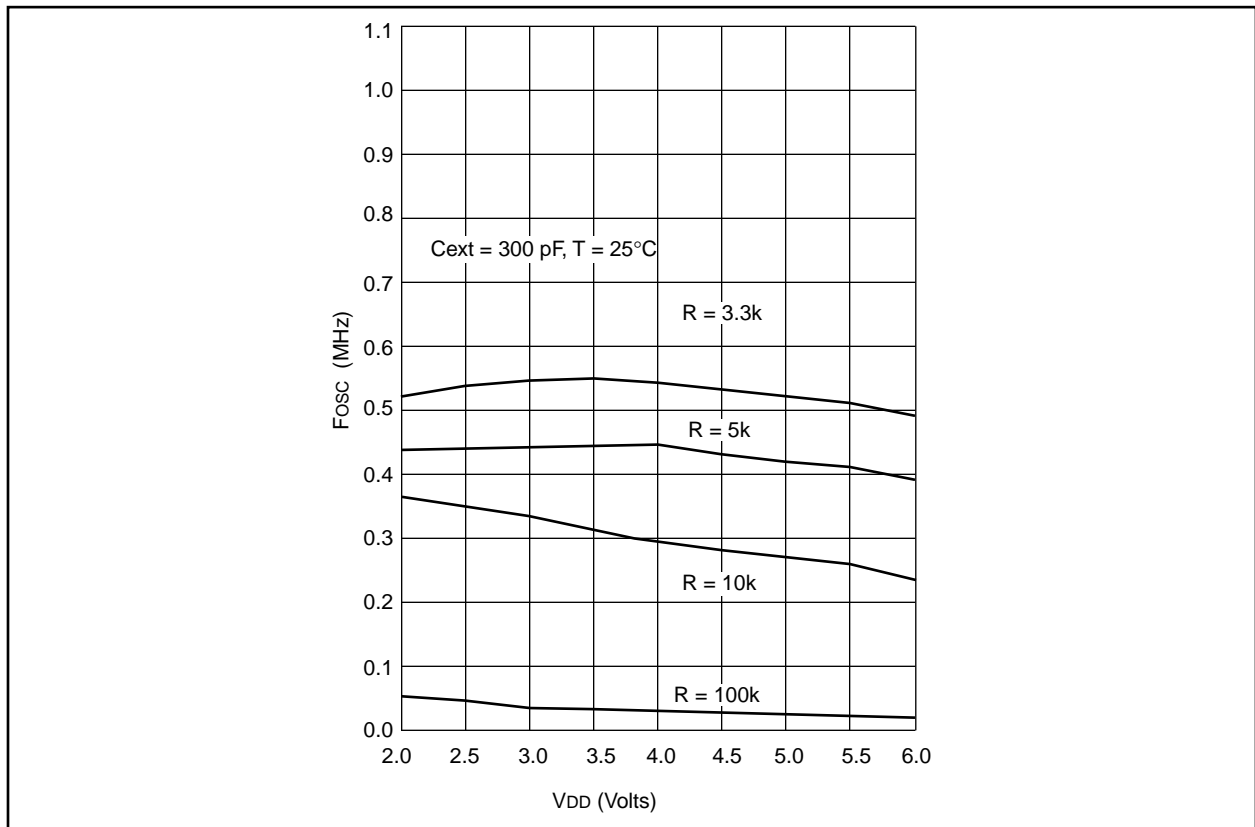
PIC16C8X

FIGURE 12-5: TYPICAL I$_{PD}$ vs. V$_{DD}$ WATCHDOG DISABLED (25˚C)
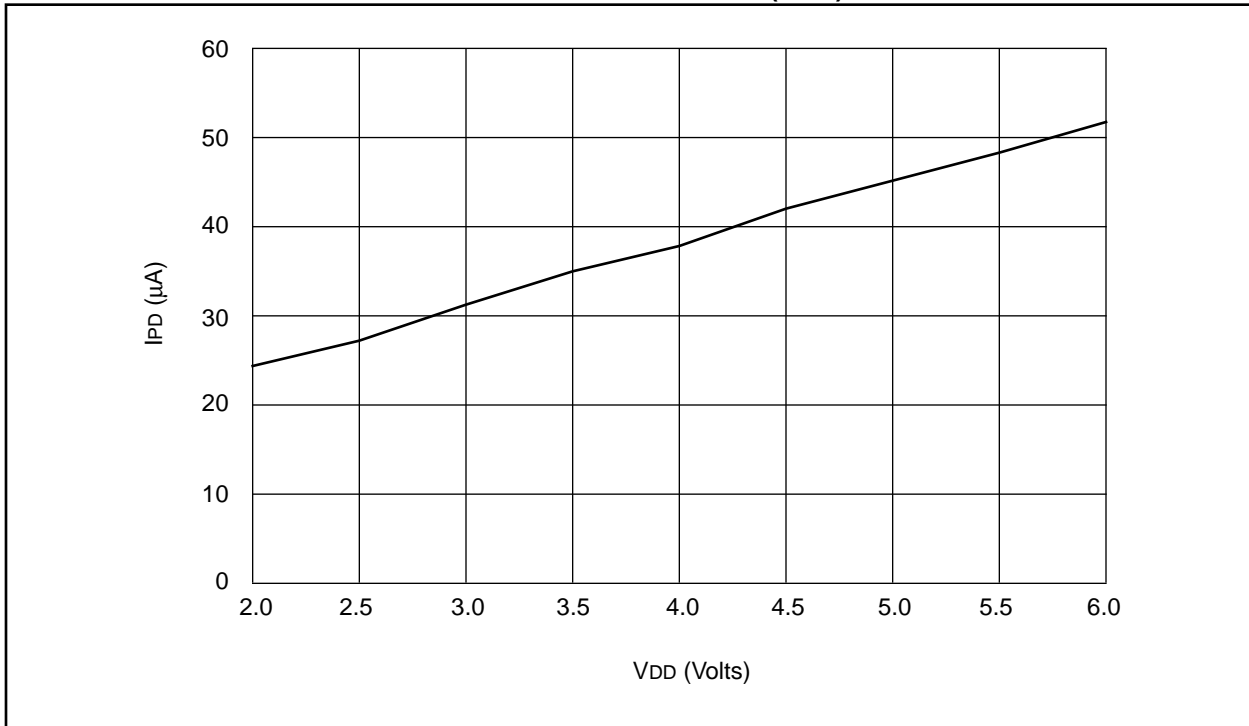
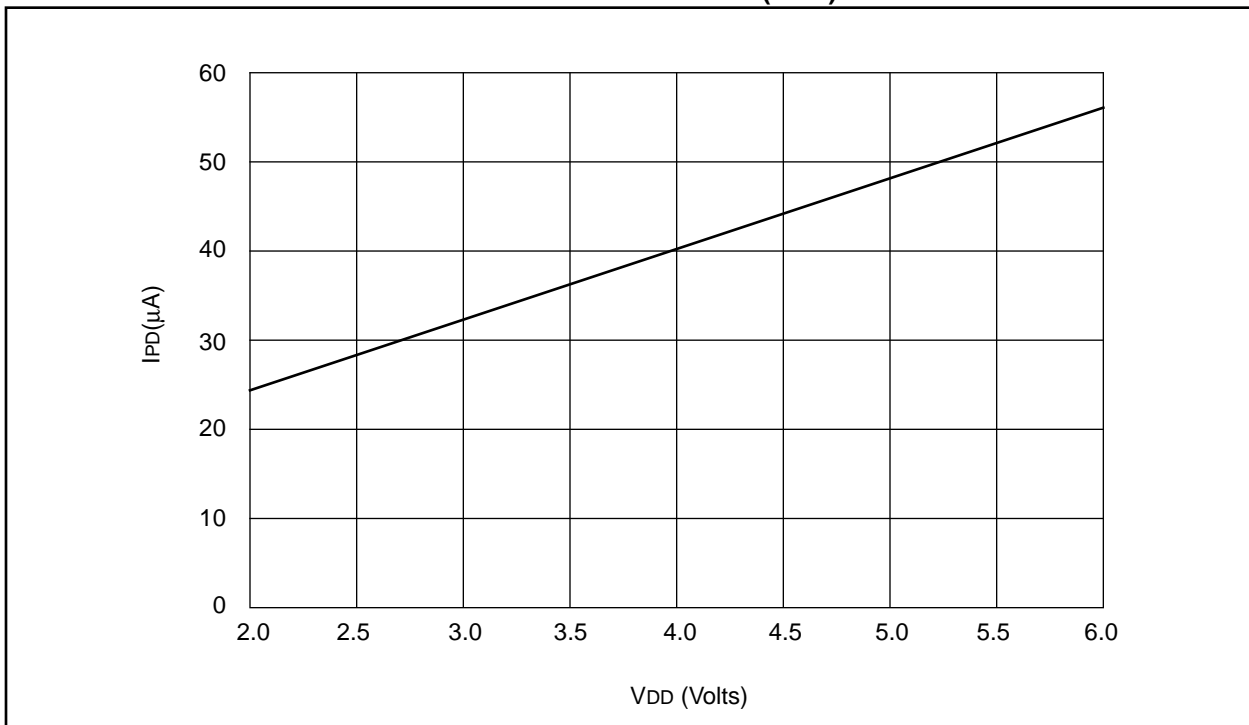**FIGURE 12-6:** TYPICAL I$_{PD}$ vs. V$_{DD}$ WATCHDOG ENABLED (25˚C)

FIGURE 12-7: MAXIMUM I$_{PD}$ vs. V$_{DD}$ WATCHDOG DISABLED
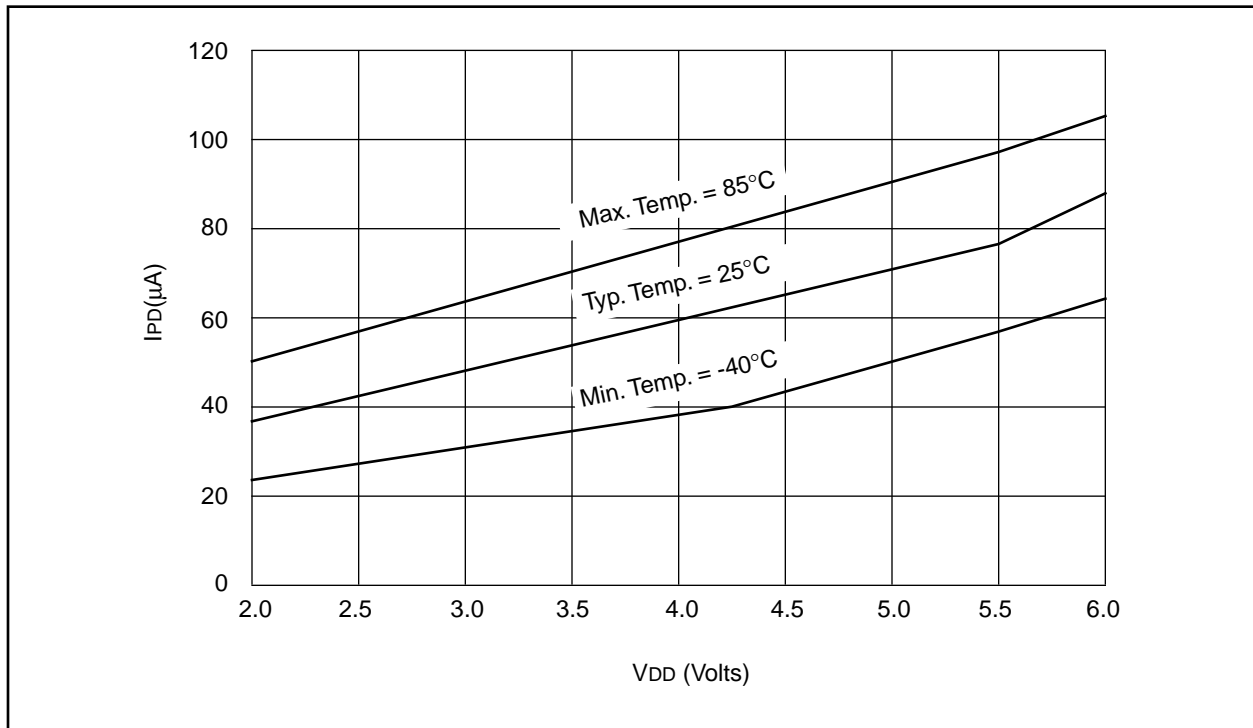

FIGURE 12-8: MAXIMUM I$_{PD}$ vs. V$_{DD}$ WATCHDOG ENABLED*
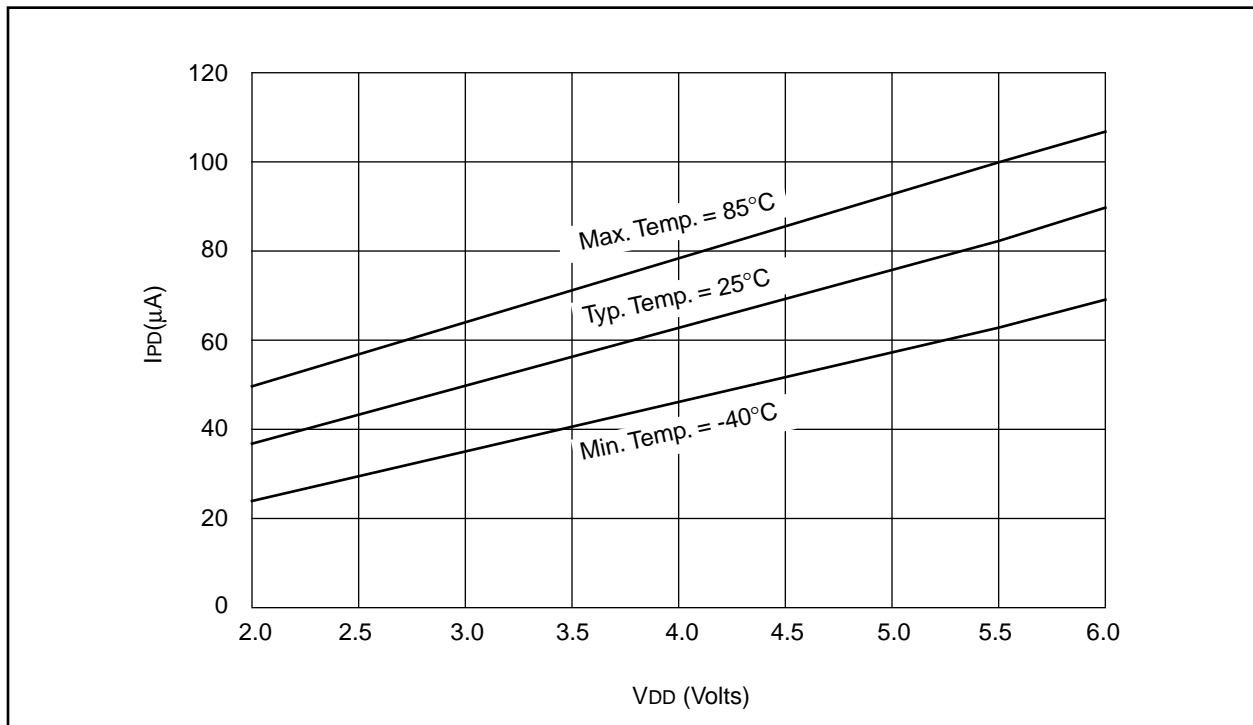
* I$_{PD}$, with Watchdog Timer enabled, has two components: The leakage current which increases with higher temperature and the operating current of the Watchdog Timer logic which increases with lower temperature. At -40°C, the latter dominates explaining the apparently anomalous behavior.

PIC16C8X

FIGURE 12-9: VTH (INPUT THRESHOLD VOLTAGE) OF I/O PINS vs. VDD
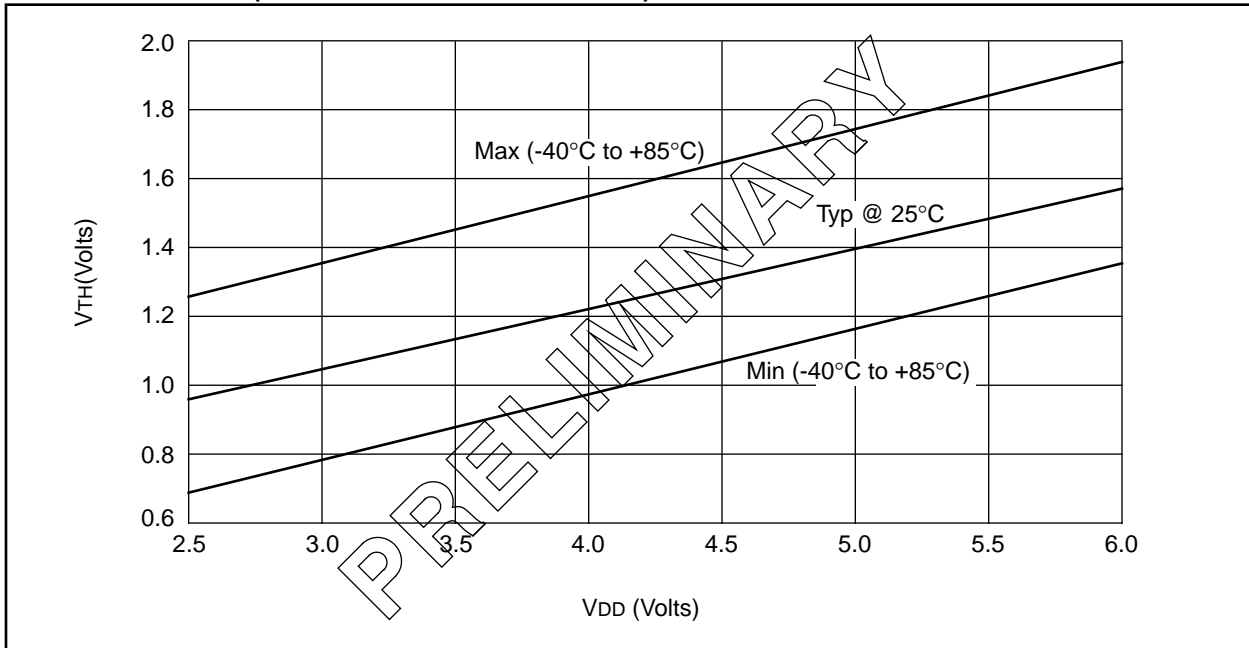
FIGURE 12-10: VTH (INPUT THRESHOLD VOLTAGE) OF OSC1 INPUT (IN XT, HS, AND LP MODES) vs. VDD

© 1995 Microchip Technology Inc.
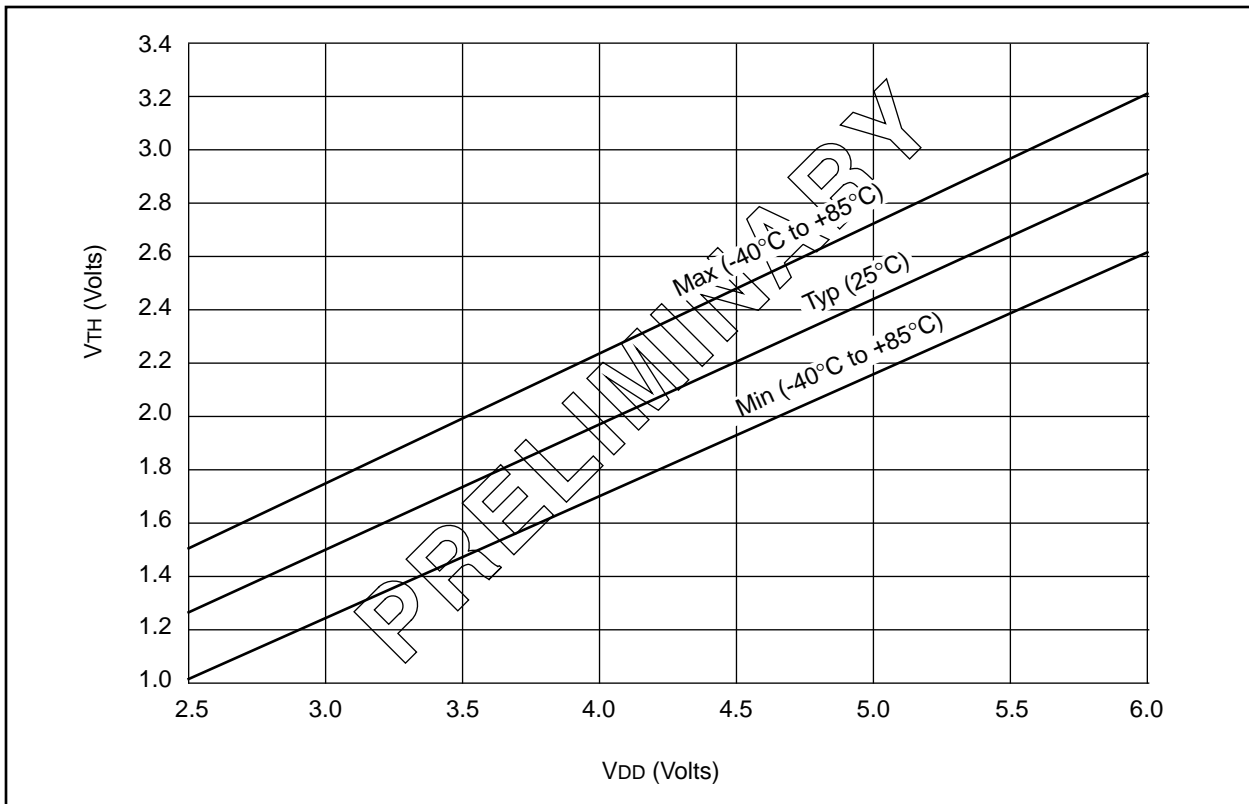
FIGURE 12-11: V$_{IH}$, V$_{IL}$ OF $\overline{MCLR}$, T0CKI and OSC1 (IN RC MODE) vs. V$_{DD}$
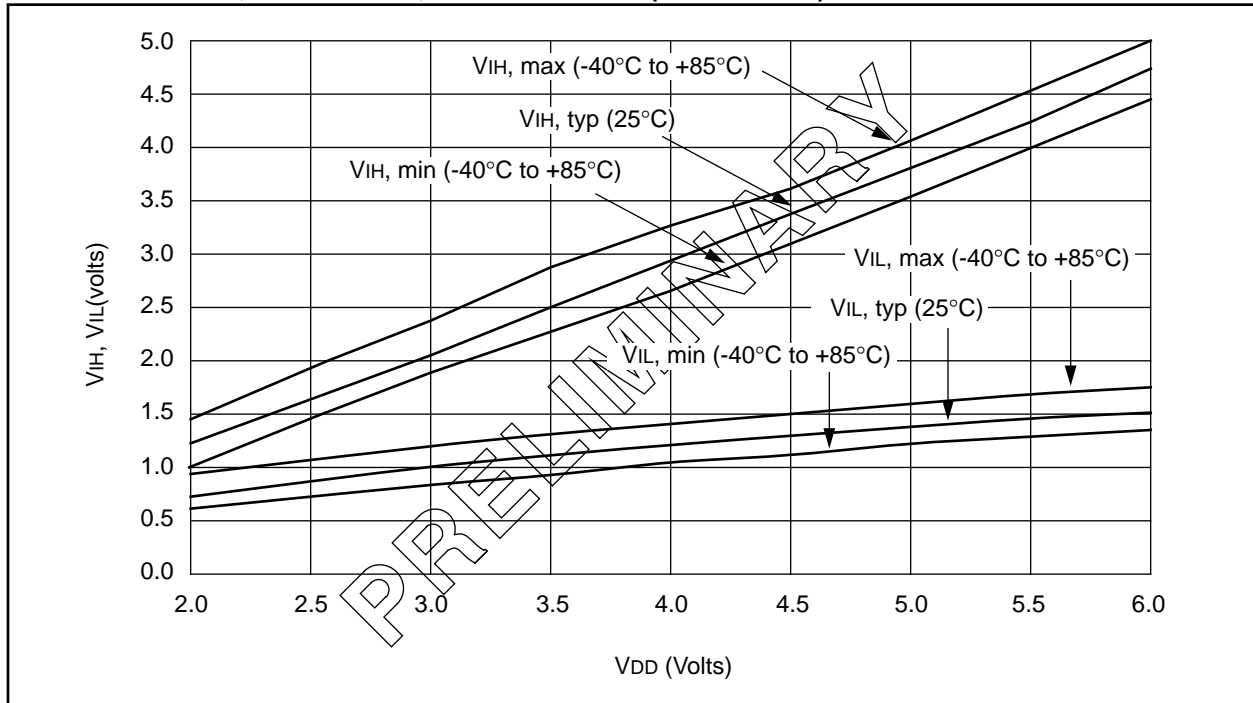
PIC16C8X

FIGURE 12-12: TYPICAL I$_{DD}$ vs. FREQ (EXT CLOCK, 25°C)
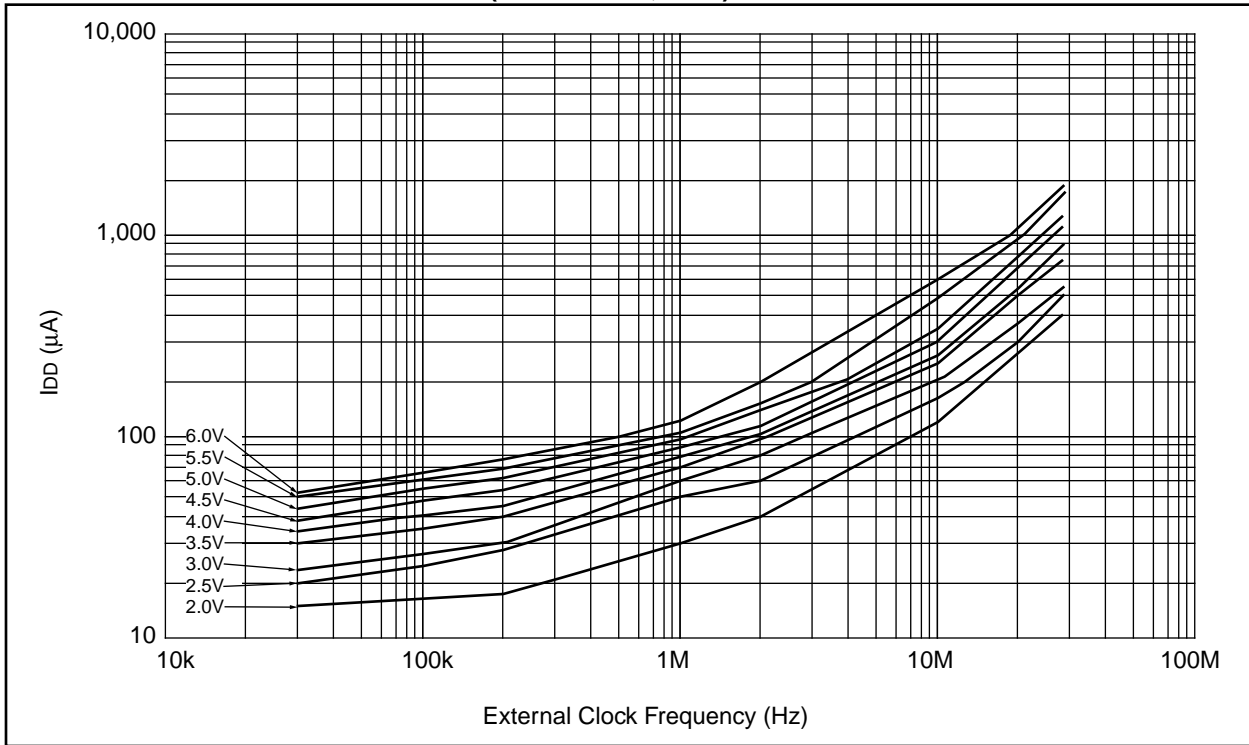
FIGURE 12-13: MAXIMUM I$_{DD}$ vs. FREQ (EXT CLOCK, -40° TO +85°C)

PIC16C8X
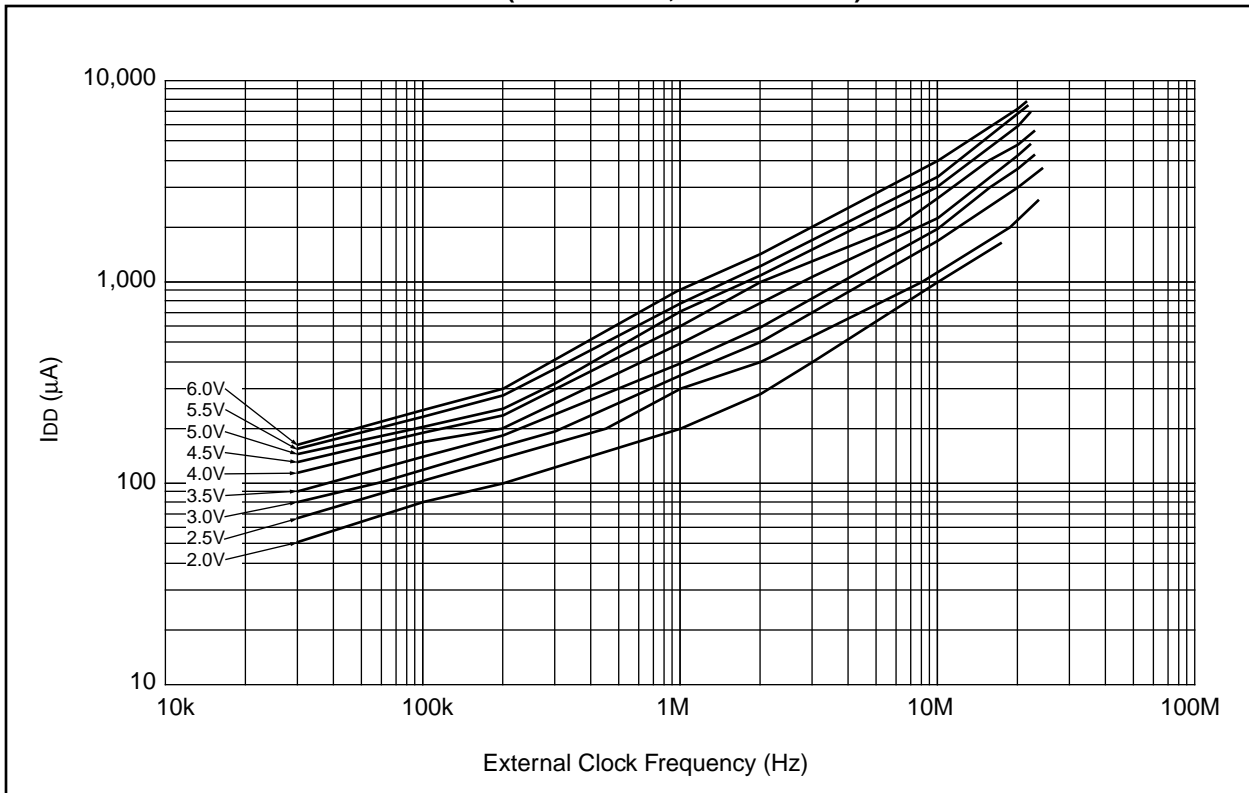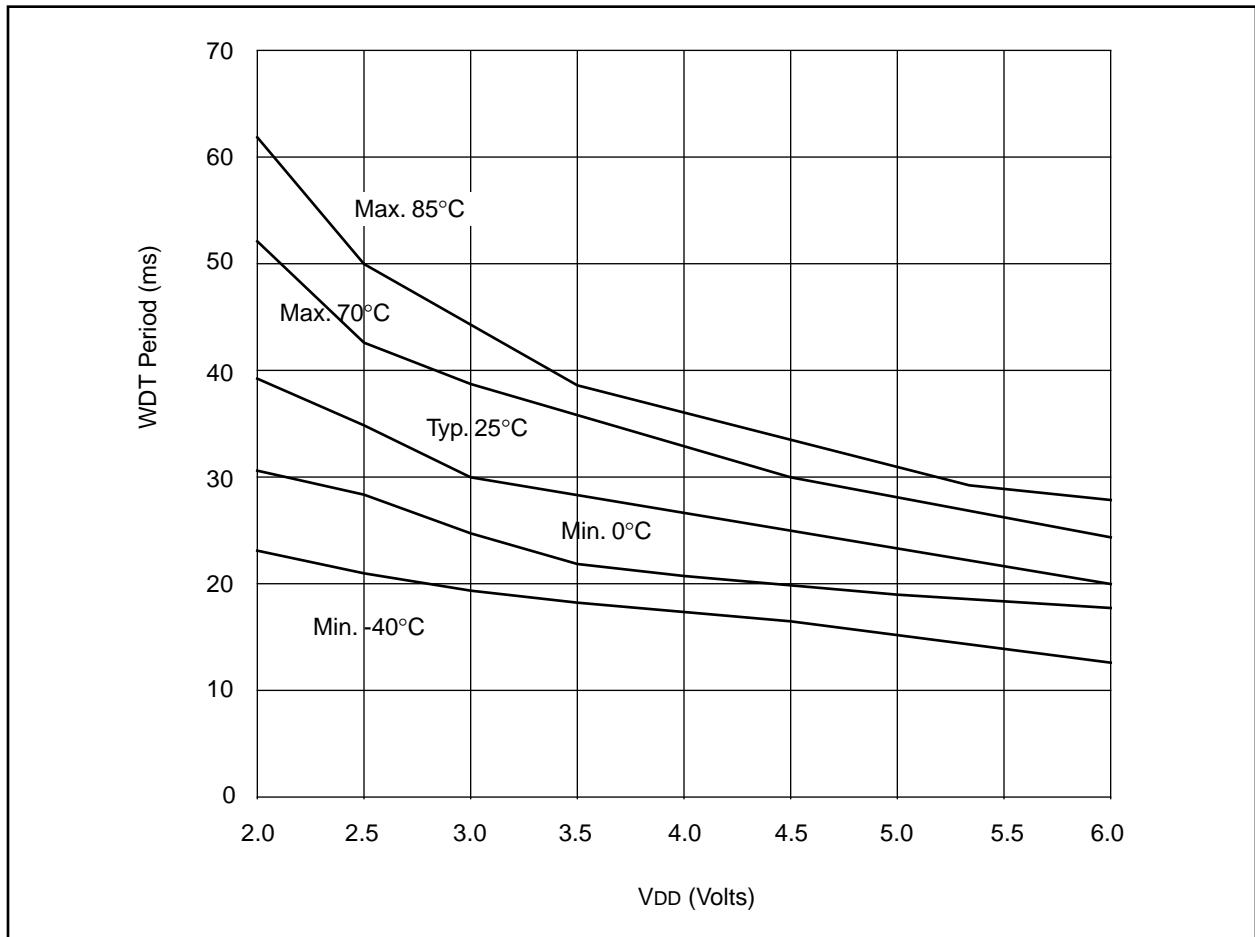
FIGURE 12-14: WDT TIMER TIME-OUT PERIOD vs. VDD

FIGURE 12-15: TRANSCONDUCTANCE (gm) OF HS OSCILLATOR vs. VDD
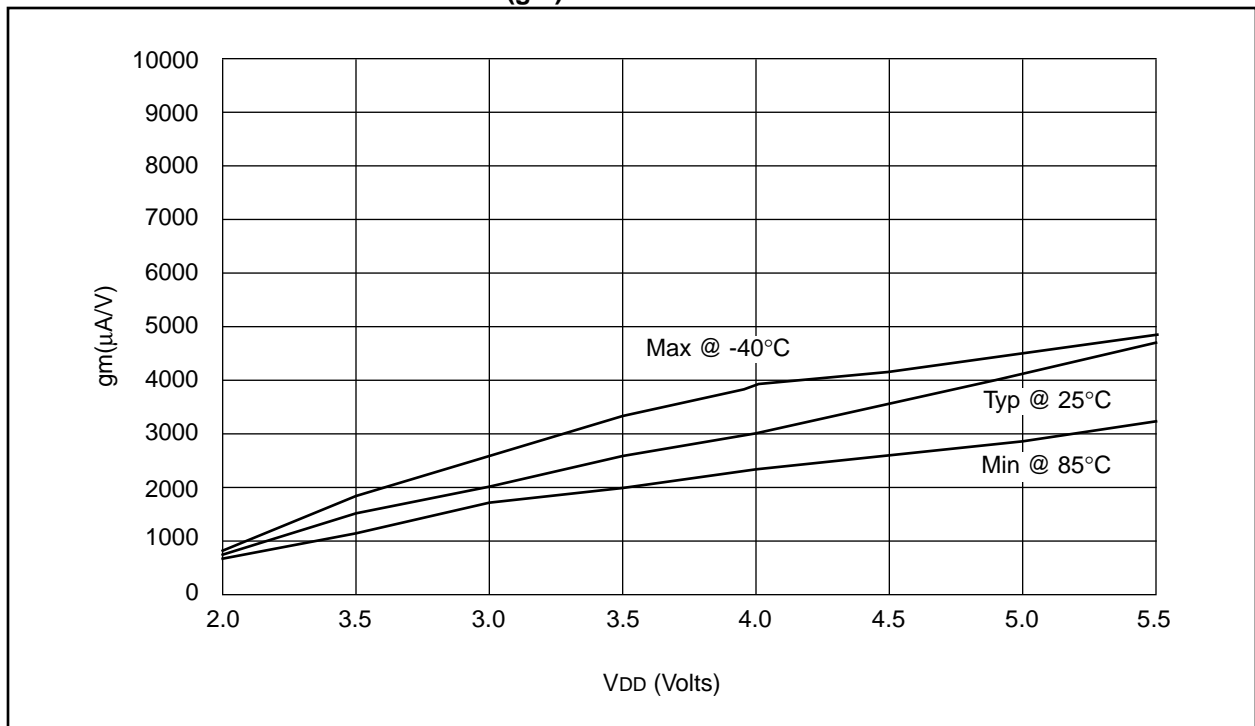
PIC16C8X

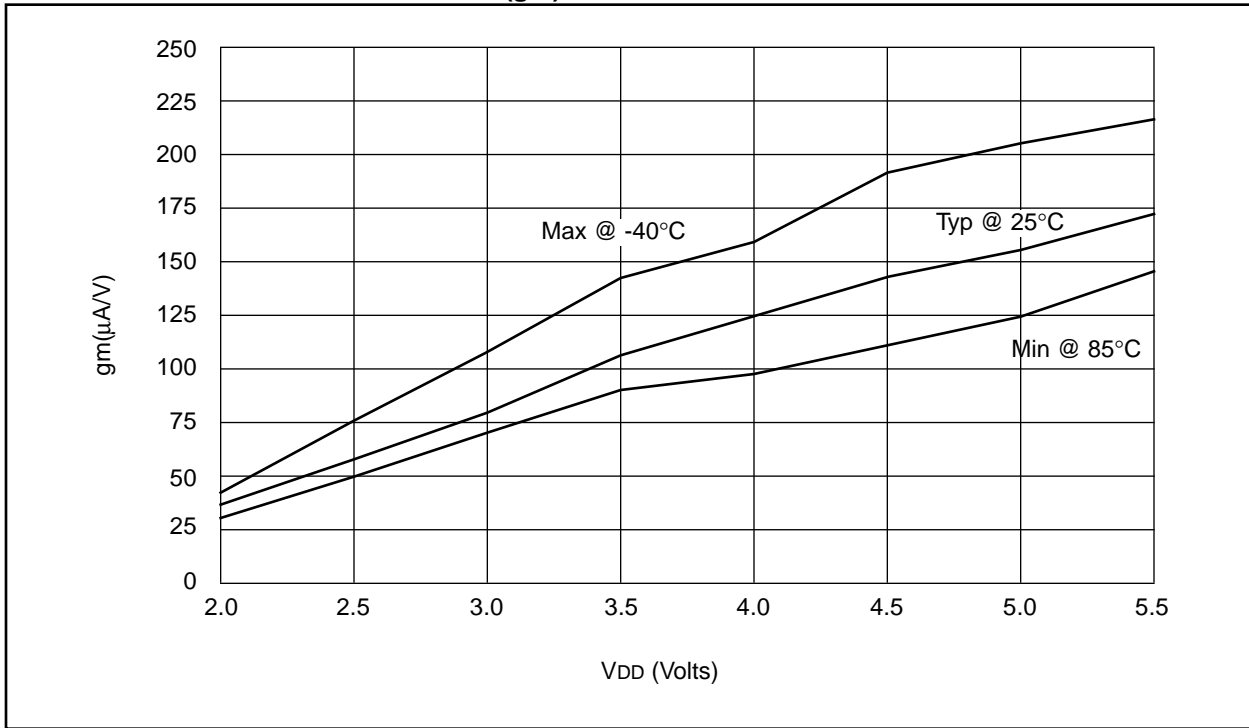**FIGURE 12-16: TRANSCONDUCTANCE (gm) OF LP OSCILLATOR vs. V$_{DD}$**

FIGURE 12-17: TRANSCONDUCTANCE (gm) OF XT OSCILLATOR vs. V$_{DD}$

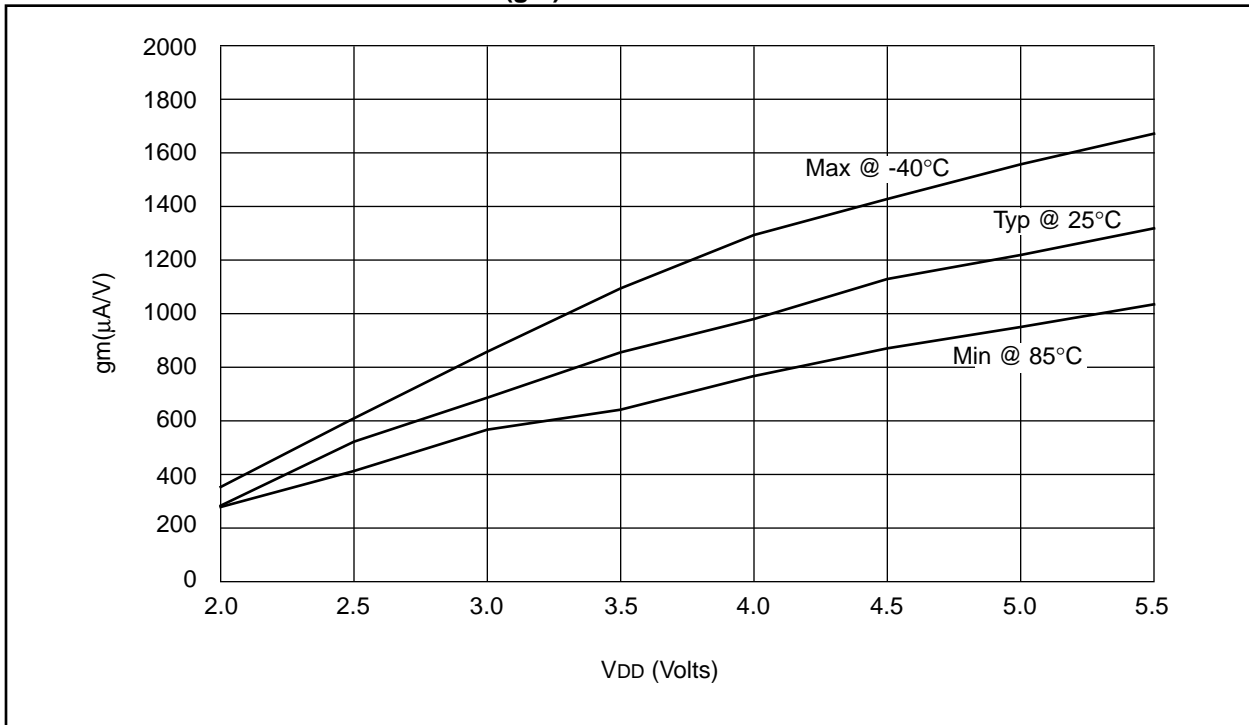**FIGURE 12-18: IOH VS. VOH, VDD = 3V**
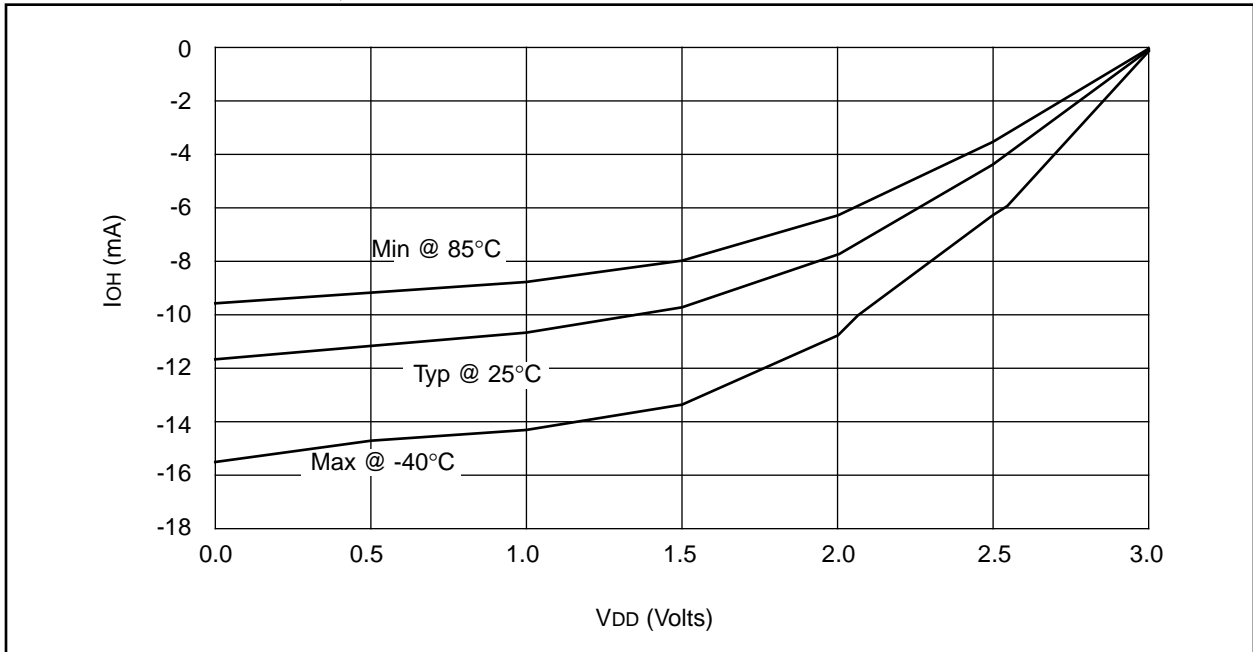
FIGURE 12-19: IOH VS. VOH, VDD = 5V
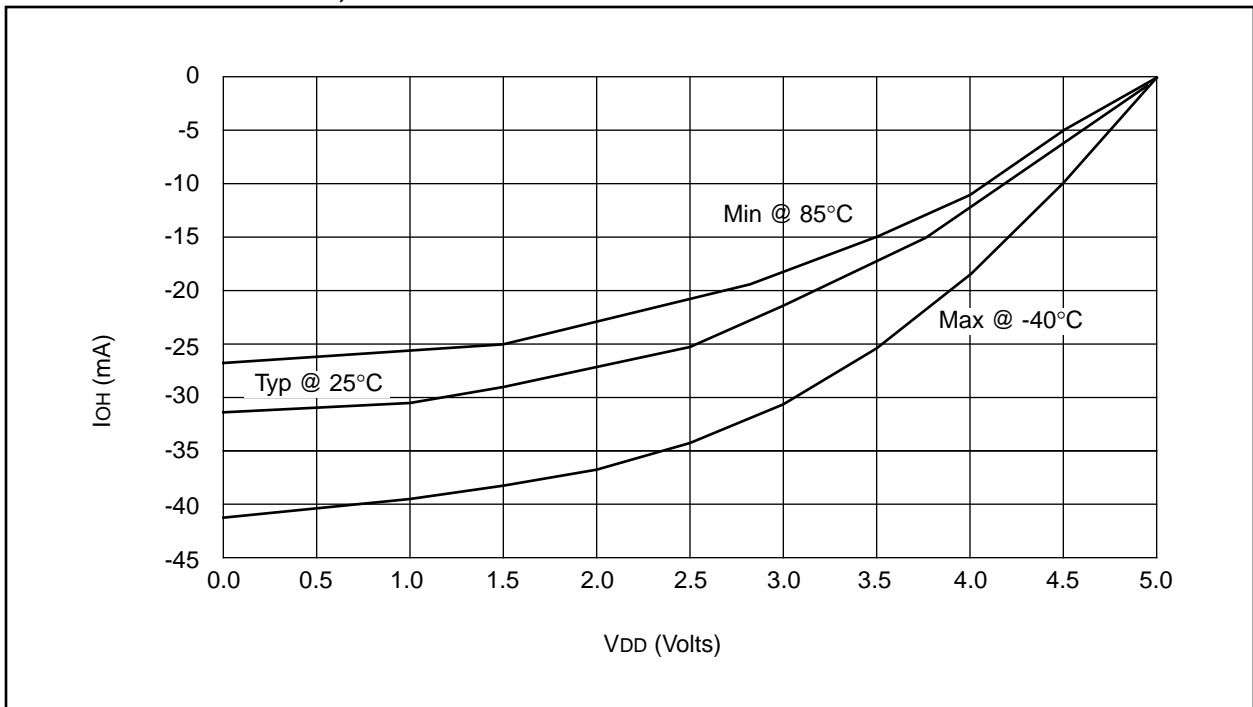
PIC16C8X

FIGURE 12-20: I_OL vs. V_OL, V_DD = 3V
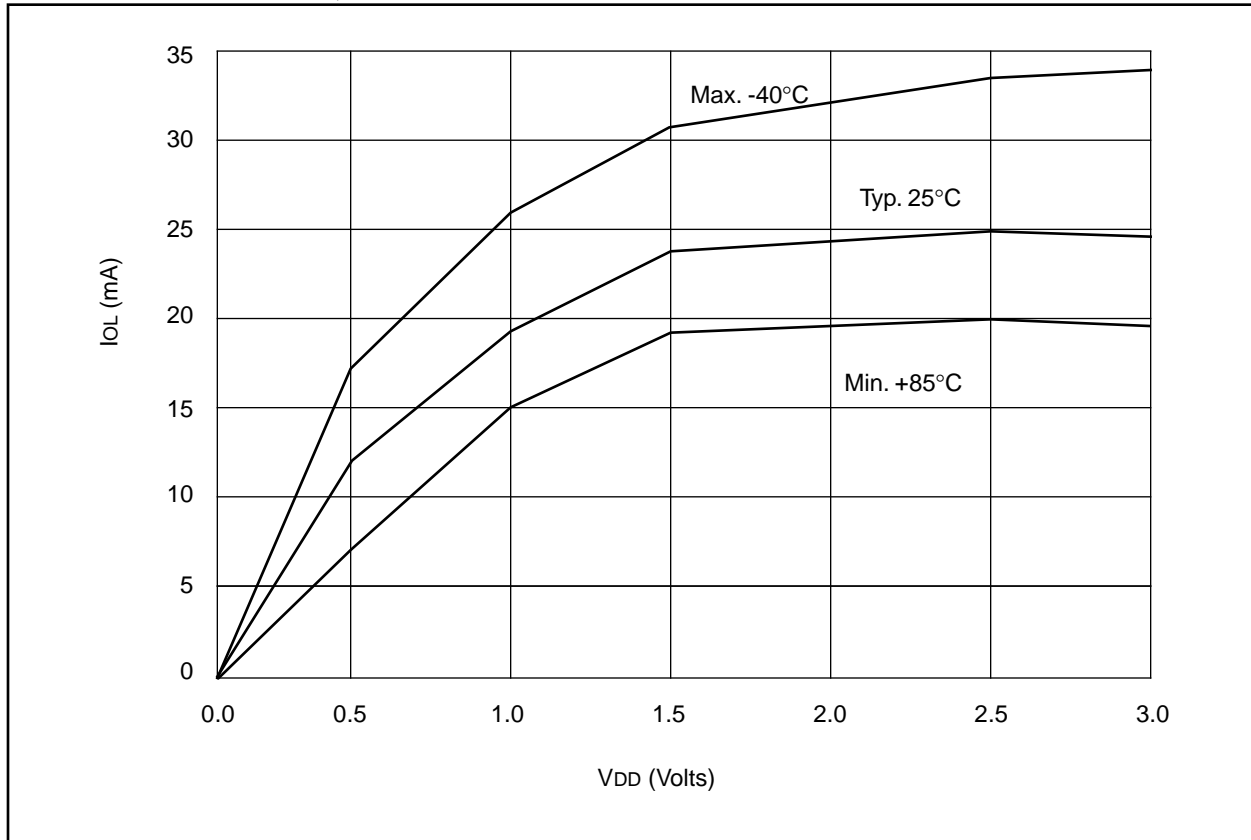
FIGURE 12-21: I_OL vs. V_OL, V_DD = 5V
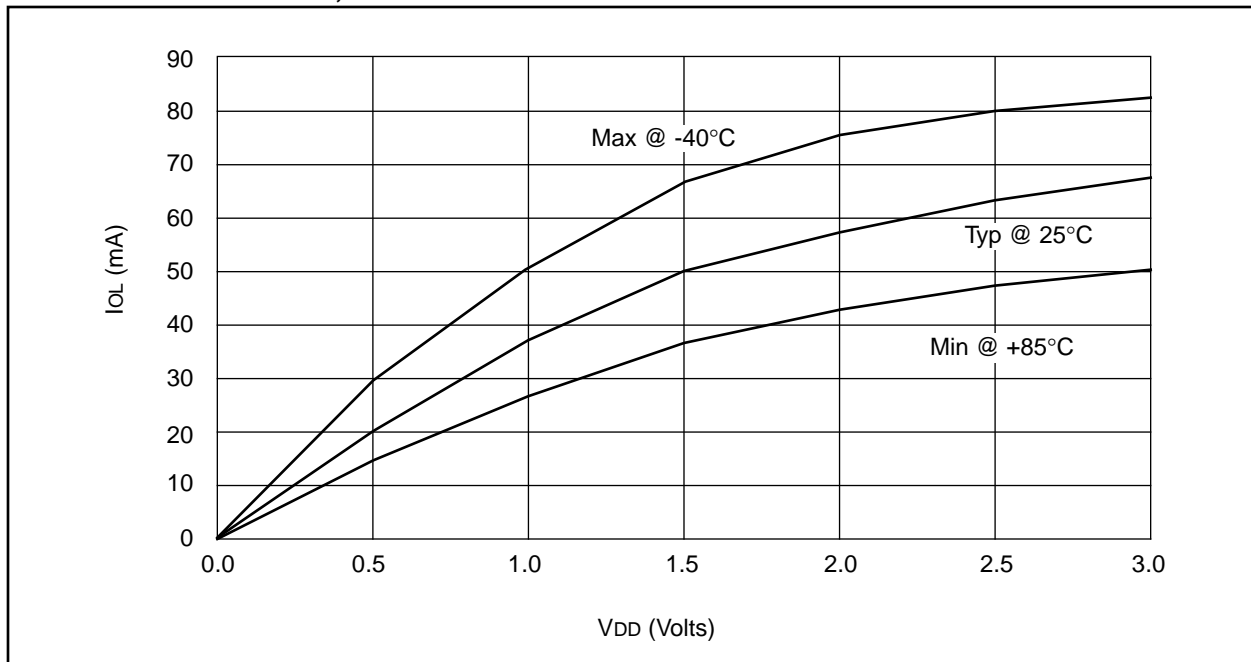
TABLE 12-2: INPUT CAPACITANCE *

Pin Name	Typical Capacitance (pF)	
	18L PDIP	18L SOIC
PORTA	5.0	4.3
PORTB	5.0	4.3
$\overline{\text{MCLR}}$	17.0	17.0
OSC1/CLKIN	4.0	3.5
OSC2/CLKOUT	4.3	3.5
T0CKI	3.2	2.8

* All capacitance values are typical at 25°C. A part to part variation of ±25% (three standard deviations) should be taken into account.

Applicable Devices 83 R83 84 84A R84

NOTES:

13.0   ELECTRICAL CHARACTERISTICS FOR PIC16C83, PIC16CR83, PIC16C84A AND PIC16CR84

Absolute Maximum Ratings †

Ambient temperature under bias..-55°C to +125°C

Storage temperature .... -65°C to +150°C

Voltage on any pin with respect to Vss (except VDD and $\overline{MCLR}$).. -0.6V to (VDD + 0.6V)

Voltage on VDD with respect to Vss .... 0 to +7.5V

Voltage on $\overline{MCLR}$ with respect to Vss (Note 2)..0 to +14V

Total power dissipation (Note 1)..800 mW

Maximum current out of Vss pin ....150 mA

Maximum current into VDD pin ....100 mA

Input clamp current, IIK (VI < 0 or VI > VDD) ....±20 mA

Output clamp current, IOK (V0 < 0 or V0 > VDD) ....±20 mA

Maximum output current sunk by any I/O pin..25 mA

Maximum output current sourced by any I/O pin ....20 mA

Maximum current sunk by PORTA ....80 mA

Maximum current sourced by PORTA..50 mA

Maximum current sunk by PORTB..150 mA

Maximum current sourced by PORTB..100 mA

Note 1: Power dissipation is calculated as follows: Pdis = VDD x {IDD - ∑ IOH} + ∑ {(VDD-VOH) x IOH} + ∑(VOl x IOL)

Note 2: Voltage spikes below Vss at the $\overline{MCLR}$ pin, inducing currents greater than 80 mA, may cause latch-up. Thus, a series resistor of 50-100Ω 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.

PIC16C8X

**TABLE 13-1:   CROSS REFERENCE OF DEVICE SPECS FOR OSCILLATOR CONFIGURATIONS
AND FREQUENCIES OF OPERATION (COMMERCIAL DEVICES)**

OSC	PIC16C84A-04 PIC16CR84-04 PIC16C83-04 PIC16CR83-04	PIC16C84A-10 PIC16CR84-10 PIC16C83-10 PIC16CR83-10	PIC16LC84A-04 PIC16LCR84-94 PIC16LC83-04 PIC16LCR83-04
RC	VDD: 4.0V to 6.0V IDD: 4.5 mA max. at 5.5V IPD: 14 µA max. at 4V WDT dis Freq: 4.0 MHz max.	VDD: 4.5V to 5.5V IDD: 1.8 mA typ. at 5.5V IPD: 1.0 µA typ. at 5.5V WDT dis Freq: 4..0 MHz max.	VDD: 2.0V to 6.0V IDD: 4.5 mA max. at 5.5V IPD: 7.0 µA max. at 2V WDT dis Freq: 2.0 MHz max.
XT	VDD: 4.0V to 6.0V IDD: 4.5 mA max. at 5.5V IPD: 14 µA max. at 4V WDT dis Freq: 4.0 MHz max.	VDD: 4.5V to 5.5V IDD: 1.8 mA typ. at 5.5V IPD: 1.0 µA typ. at 5.5V WDT dis Freq: 4.0 MHz max.	VDD: 2.0V to 6.0V IDD: 4.5 mA max. at 5.5V IPD: 7.0 µA max. at 2V WDT dis Freq: 2.0 MHz max.
HS	VDD: 4.5V to 5.5V IDD: 4.5 mA typ. at 5.5V IPD: 1.0 µA typ. at 4.5V WDT dis Freq: 4.0 MHz max.	VDD: 4.5V to 5.5V IDD: 10 mA max. at 5.5V typ. IPD: 1.0 µA typ. at 4.5V WDT dis Freq: 10 MHz max.	Do not use in HS mode
LP	VDD: 4.0V to 6.0V IDD: 35 µA typ. at 32 kHz, 3.0V IPD: 0.6 µA typ. at 3.0V WDT dis Freq: 200 kHz max.	Do not use in LP mode	VDD: 2.0V to 6.0V IDD: 32 µA max. at 32 kHz, 3.0V IPD: 7 µA max. at 2.0V WDT dis Freq: 200 kHz 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.

Preliminary

13.1 DC CHARACTERISTICS: PIC16C84A, PIC16C83 (Commercial, Industrial)
PIC16CR84, PIC16CR83 (Commercial, Industrial)

DC CHARACTERISTICS			**Standard Operating Conditions (unless otherwise stated)** Operating temperature −40°C ≤ T_A ≤ +85°C for industrial and 0°C ≤ T_A ≤ +70°C for commercial					
Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions	
D001	V_{DD}	Supply Voltage	4.0	—	6.0	V	XT, RC and LP osc configuration	
D001A			4.5	—	5.5	V	HS osc configuration	
D002	V_{DR}	RAM Data Retention Voltage (Note 1)	1.5 *	—	—	V	Device in SLEEP mode	
D003	V_{POR}	V_{DD} start voltage to ensure Power-on Reset	—	V_{SS}	—	V	See section on Power-on Reset for details	
D004	S_{VDD}	V_{DD} rise rate to ensure Power-on Reset	0.05*	—	—	V/ms	See section on Power-on Reset for details	
D010	I_{DD}	Supply Current (Note 2)	—	1.8	4.5	mA	RC and XT osc configuration (Note 4) F_{OSC} = 4.0 MHz, V_{DD} = 5.5V	
D010A			—	7.3	10	mA	F_{OSC} = 4.0 MHz, V_{DD} = 5.5V (During EEPROM programming)	
D013			—	5	10	mA	HS OSC CONFIGURATION (PIC16C84-10) F_{OSC} = 10 MHz, V_{DD} = 5.5V	
D020	I_{PD}	Power-down Current (Note 3)	—	7.0	28	μA	V_{DD} = 4.0V, WDT enabled, −40°C to +85°C	
D021			—	1.0	14	μA	V_{DD} = 4.0V, WDT disabled, −0°C to +70°C	
D021A			—	1.0	16	μA	V_{DD} = 4.0V, WDT disabled, −40°C to +85°C	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: This is the limit to which V_{DD} 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 I_{DD} measurements in active operation mode are:
OSC1=external square wave, from rail to rail; all I/O pins tristated, pulled to V_{DD}, T0CKI = V_{DD}, $\overline{MCLR}$ = V_{DD}; WDT enabled/disabled as specified.

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 V_{DD} and V_{SS}.

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

PIC16C8X

13.2 DC CHARACTERISTICS: PIC16LC84A, PIC16LC83 (Commercial, Industrial)
PIC16LCR84, PIC16LCR83 (Commercial, Industrial)

DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C $\leq$ TA $\leq$ +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	2.0	—	6.0	V	XT, RC, and LP osc configuration
D002	VDR	RAM Data Retention Voltage (Note 1)	1.5 *	—	—	V	Device in SLEEP mode
D003	VPOR	VDD start voltage to ensure Power-on Reset	—	Vss	—	V	See section on Power-on Reset for details
D004	SVDD	VDD rise rate to ensure Power-on Reset	0.05*	—	—	V/ms	See section on Power-on Reset for details
D010 D010A	IDD	Supply Current (Note 2)	— —	1.8 7.3	4.5 10	mA mA	RC and XT osc configuration (Note 4) FOSC = 2.0 MHz, VDD = 5.5V FOSC = 2.0 MHz, VDD = 5.5V (During EEPROM programming)
D014			—	15	32	µA	LP osc configuration FOSC = 32 kHz, VDD = 2.0V, WDT disabled
D020 D021 D021A	IPD	Power-down Current (Note 3)	— — —	3.0 0.4 0.4	16 7.0 9.0	µA µA µA	VDD = 2.0V, WDT enabled, -40°C to +85°C VDD = 2.0V, WDT disabled, 0°C to +70°C VDD = 2.0V, WDT disabled, -40°C to +85°C

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5.0V, 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, T0CKI = VDD, $\overline{MCLR}$ = VDD; WDT enabled/disabled as specified.

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.

Preliminary

13.3   DC CHARACTERISTICS: **PIC16C84A, PIC16C83   (Commercial, Industrial)**
PIC16CR84A, PIC16CR83 (Commercial, Industrial)
PIC16LC84A, PIC16LC83 (Commercial, Industrial)
PIC16LCR84, PIC16LCR83 (Commercial, Industrial)

DC CHARACTERISTICS							**Standard Operating Conditions (unless otherwise stated)** Operating temperature -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 DC spec Section 13-1 and Section 13-3

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
	VIL	**Input Low Voltage** I/O ports					
D030		with TTL buffer	VSS	—	0.8	V	4.5 V $\leq$ VDD $\leq$ 5.5 V
D030A			VSS	—	0.16VDD	V	entire range (Note 4)
D031		with Schmitt Trigger buffer	VSS	—	0.2VDD	V	entire range
D032		$\overline{MCLR}$, RA4/T0CKI	VSS	—	0.2VDD	V	
D033		OSC1 (XT, HS and LP modes)	VSS	—	0.3VDD	V	Note1
D034		OSC1 (RC mode)	VSS	—	0.1VDD	V	
	VIH	**Input High Voltage** I/O ports					
D040		with TTL buffer	2.4	—	VDD	V	4.5 V $\leq$ VDD $\leq$ 5.5V
D040A			0.48VDD	—	VDD	V	entire range (Note 4)
D041		with Schmitt Trigger buffer	0.45VDD	—	VDD	V	entire range
D042		$\overline{MCLR}$, RA4/T0CKI, OSC1 (RC mode)	0.85 VDD	—	VDD	V	
D043		OSC1 (XT, HS and LP modes)	0.7 VDD	—	VDD	V	Note1
D050	VHYS	Hysteresis of Schmitt Trigger inputs	TBD	—	—	V	
D070	IPURB	PORTB weak pull-up current	50*	250*	400*	µA	VDD = 5.0V, VPIN = VSS
	IIL	**Input Leakage Current** (Notes 2, 3)					
D060		I/O ports	—	—	±1	µA	VSS $\leq$ VPIN $\leq$ VDD, Pin at hi-impedance
D061		$\overline{MCLR}$, RA4/T0CKI	—	—	±5	µA	VSS $\leq$ VPIN $\leq$ VDD
D063		OSC1	—	—	±5	µA	VSS $\leq$ VPIN $\leq$ VDD, XT, HS and LP osc configuration
	VOL	**Output Low Voltage** I/O ports					
D080		I/O ports	—	—	0.6	V	IOL = 8.5 mA, VDD = 4.5V
D083		OSC2/CLKOUT	—	—	0.6	V	IOL = 1.6 mA, VDD = 4.5V
	VOH	**Output High Voltage**					
D090		I/O ports (Note 3)	VDD-0.7	—	—	V	IOH = -3.0 mA, VDD = 4.5V
D092		OSC2/CLKOUT	VDD-0.7	—	—	V	IOH = -1.3 mA, VDD = 4.5V

*   These parameters are characterized but not tested.

†   Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: In RC oscillator configuration, the OSC1 pin is a Schmitt Trigger input. Do not drive the PIC16C8X with an external clock while the device is in RC mode, otherwise chip damage may result.

2: The leakage current on the $\overline{MCLR}$ 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: The user may use better of the two specs.

PIC16C8X

13.4 DC CHARACTERISTICS: PIC16C84A, PIC16C83 (Commercial, Industrial)
PIC16CR84A, PIC16CR83 (Commercial, Industrial)
PIC16LC84A, PIC16C83 (Commercial, Industrial)
PIC16LCR84A, PIC16LCR83 (Commercial, Industrial)

DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C $\leq$ T$_A$ $\leq$ +85°C for industrial and 0°C $\leq$ T$_A$ $\leq$ +70°C for commercial Operating voltage V$_{DD}$ range as described in DC spec Section 13-1 and Section 13-3					
Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions	
		Capacitive Loading Specs on Output Pins						
D100	C$_{OSC2}$	OSC2 pin	—	—	15	pF	In XT, HS and LP modes when external clock is used to drive OSC1.	
D101	C$_{IO}$	All I/O pins and OSC2 (RC mode)	—	—	50	pF		
		Data EEPROM Memory						
D120	E$_D$	Endurance	100,000	1,000,000	—	E/W		
D121	V$_{DRW}$	V$_{DD}$ for read/write	V$_{MIN}$	—	6.0	V	V$_{MIN}$ = Minimum operating voltage	
D122	T$_{DEW}$	Erase/Write cycle time	—	—	10	ms		
		Program EEPROM Memory						
D130	E$_P$	Endurance	100	1000	—	E/W		
D131	V$_{PR}$	V$_{DD}$ for read	V$_{MIN}$	—	6.0	V	V$_{MIN}$ = Minimum operating voltage	
D132	V$_{PEW}$	V$_{DD}$ for erase/write	4.5	—	5.5	V		
D133	T$_{PEW}$	Erase/Write cycle time	—	10	—	ms		

† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Preliminary © 1995 Microchip Technology Inc.

TABLE 13-2: TIMING PARAMETER SYMBOLOGY

The timing parameter symbols have been created following one of the following formats:

1. TppS2ppS
2. TppS

T				
F	Frequency		T	Time

Lowercase symbols (pp) and their meanings:

pp				
ck	CLKOUT		osc	OSC1
io	I/O port		t0	T0CKI
mc	$\overline{MCLR}$			

Uppercase symbols and their meanings:

S				
F	Fall		P	Period
H	High		R	Rise
I	Invalid (Hi-impedance)		V	Valid
L	Low		Z	High Impedance

FIGURE 13-1: PARAMETER MEASUREMENT INFORMATION

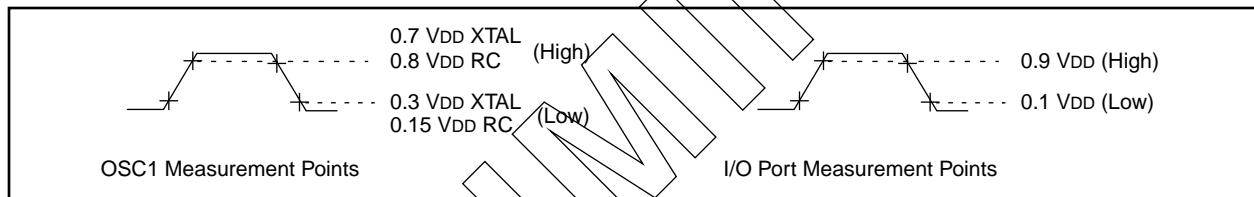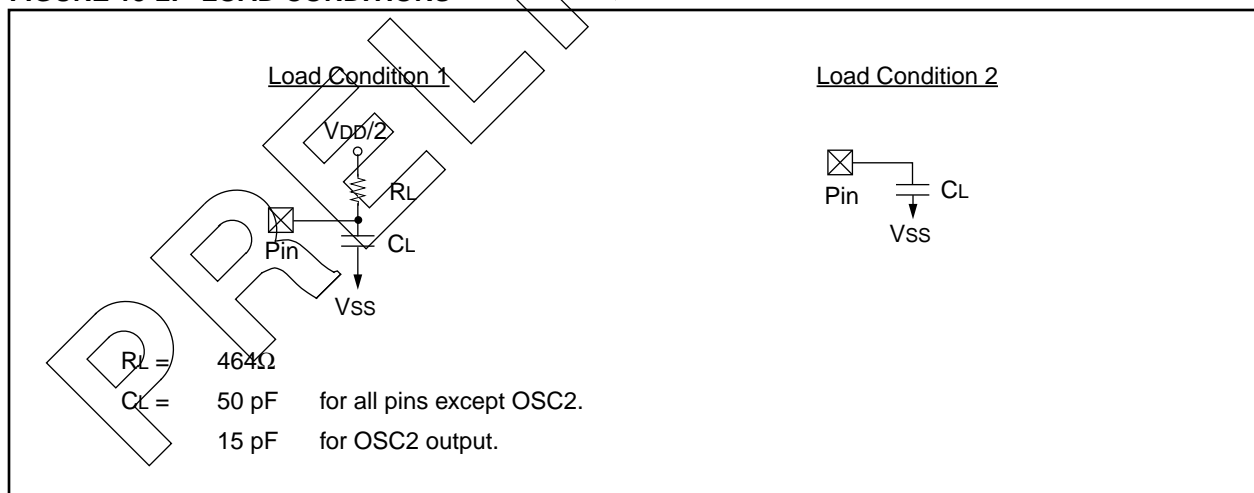All timings are measure between high and low measurement points as indicated in the figures below.

OSC1 Measurement Points   I/O Port Measurement Points

0.7 VDD XTAL / 0.8 VDD RC (High)
0.3 VDD XTAL / 0.15 VDD RC (Low)

0.9 VDD (High)
0.1 VDD (Low)

FIGURE 13-2: LOAD CONDITIONS

Load Condition 1

VDD/2
RL
Pin
CL
Vss

Load Condition 2

Pin
CL
Vss

RL = 464Ω
CL = 50 pF   for all pins except OSC2.
     15 pF   for OSC2 output.

PIC16C8X

13.5 Timing Diagrams and Specifications for PIC16C83, PIC16CR83, PIC16C84A, and PIC16CR84
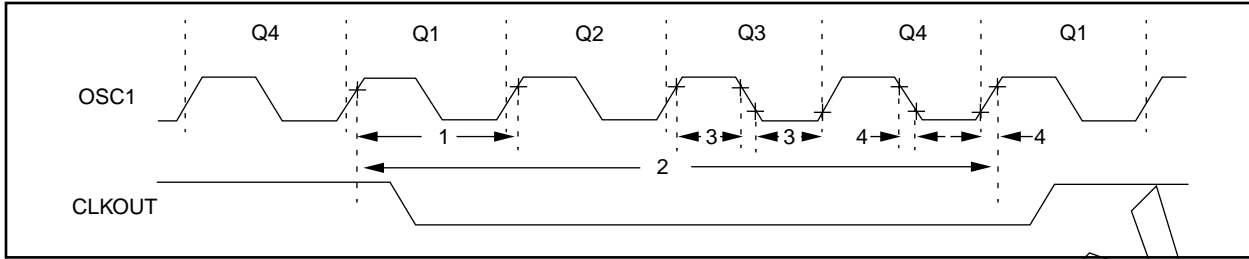
FIGURE 13-3: EXTERNAL CLOCK TIMING

TABLE 13-3: EXTERNAL CLOCK TIMING REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions	
	Fos	**External CLKIN Frequency (Note 1)**	DC	—	2	MHz	XT, RC osc	PIC16LC8X-04 PIC16LCR8X-04
			DC	—	4	MHz	XT, RC osc	PIC16C8X-04 PIC16CR8X-04
			DC	—	10	MHz	HS osc	PIC16C8X-10 PIC16CR8X-10
			DC	—	200	kHz	LP osc	PIC16LC8X-04 PIC16LCR8X-04
		Oscillator Frequency (Note 1)	DC	—	2	MHz	RC osc	PIC16LC8X-04 PIC16LCR8X-04
			DC	—	4	MHz	RC osc	PIC16C8X-04 PIC16CR8X-04
			0.1	—	2	MHz	XT osc	PIC16LC8X-04 PIC16LCR8X-04
			0.1	—	4	MHz	XT osc	PIC16C8X-04 PIC16CR8X-04
			1.0	—	10	MHz	HS osc	PIC16C8X-10 PIC16CR8X-10
			DC	—	200	kHz	LP osc	PIC16LC8X-04 PIC16LCR8X-04

† Data in "Typ" column is at 5.0V, 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 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 pin.
When an external clock input is used, the "Max." cycle time limit is "DC" (no clock) for all devices.

Preliminary

TABLE 13-3: EXTERNAL CLOCK TIMING REQUIREMENTS (CONTINUED)

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions	
1	Tosc	**External CLKIN Period (Note 1)**	500	—	—	ns	XT, RC osc	PIC16LC8X-04 PIC16LCR8X-04
			250	—	—	ns	XT, RC osc	PIC16C8X-04 PIC16CR8X-04
			100	—	—	ns	HS osc	PIC16C8X-10 PIC16CR8X-10
			5.0	—	—	μs	LP osc	PIC16LC8X-04 PIC16LCR8X-04
		Oscillator Period (Note 1)	500	—	—	ns	RC osc	PIC16LC8X-04 PIC16LCR8X-04
			250	—	—	ns	RC osc	PIC16C8X-04 PIC16CR8X-04
			500	—	10,000	ns	XT osc	PIC16LC8X-04 PIC16LCR8X-04
			250	—	10,000	ns	XT osc	PIC16C8X-04 PIC16CR8X-04
			100	—	1,000	ns	HS osc	PIC16C8X-10 PIC16CR8X-10
			5.0	—	—	μs	LP osc	PIC16LC8X-04 PIC16LCR8X-04
2	TCY	**Instruction Cycle Time (Note 1)**	0.4	4/Fosc	DC	μs		
3	TosL, TosH	Clock in (OSC1) High or Low Time	60 *	—	—	ns	XT osc	PIC16LC8X-04 PIC16LCR8X-04
			50 *	—	—	ns	XT osc	PIC16C8X-04 PIC16CR8X-04
			2.0 *	—	—	μs	LP osc	PIC16LC8X-04 PIC16LCR8X-04
			35 *	—	—	ns	HS osc	PIC16C8X-10 PIC16CR8X-10
4	TosR, TosF	Clock in (OSC1) Rise or Fall Time	25 *	—	—	ns	XT osc	PIC16C8X-04 PIC16CR8X-04
			50 *	—	—	ns	LP osc	PIC16LC8X-04 PIC16LCR8X-04
			15 *	—	—	ns	HS osc	PIC16C8X-10 PIC16CR8X-10

† Data in "Typ" column is at 5.0V, 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 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 pin.
When an external clock input is used, the "Max." cycle time limit is "DC" (no clock) for all devices.
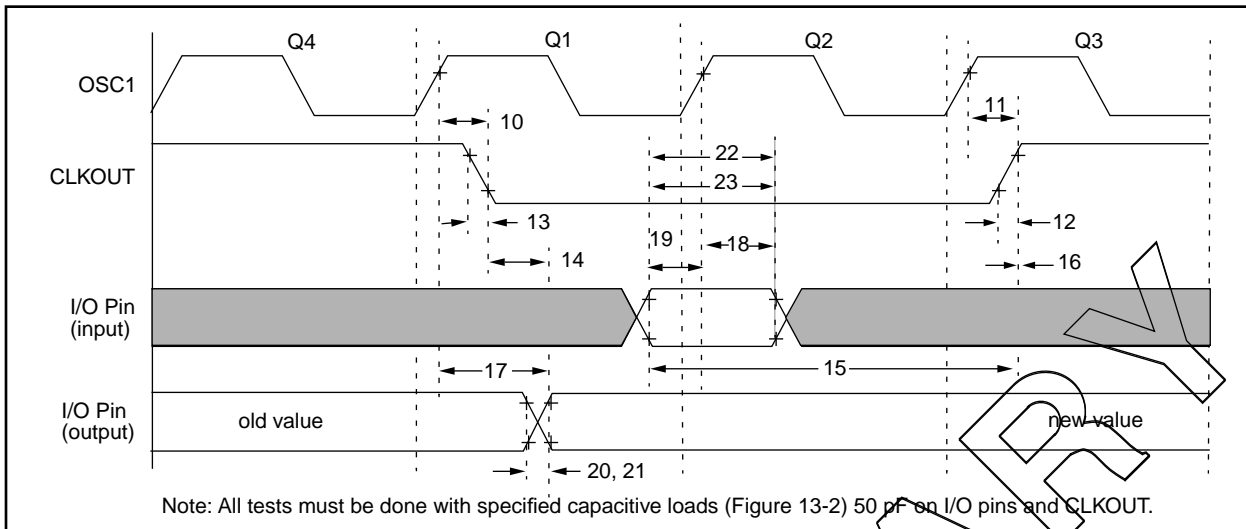
PIC16C8X

FIGURE 13-4: CLKOUT AND I/O TIMING

Note: All tests must be done with specified capacitive loads (Figure 13-2) 50 pf on I/O pins and CLKOUT.

TABLE 13-4: CLKOUT AND I/O TIMING REQUIREMENTS

Parameter No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
10	TosH2ckL	OSC1↑ to CLKOUT↓	PIC16C8X	—	15	30 *	ns	Note 1
10A			PIC16LC8X	—	15	120 *	ns	Note 1
11	TosH2ckH	OSC1↑ to CLKOUT↑	PIC16C8X	—	15	30 *	ns	Note 1
11A			PIC16LC8X	—	15	120 *	ns	Note 1
12	TckR	CLKOUT rise time	PIC16C8X	—	15	30 *	ns	Note 1
12A			PIC16LC8X	—	15	100 *	ns	Note 1
13	TckF	CLKOUT fall time	PIC16C8X	—	15	30 *	ns	Note 1
13A			PIC16LC8X	—	15	100 *	ns	Note 1
14	TckL2ioV	CLKOUT ↓ to Port out valid		—	—	0.5TCY +20 *	ns	Note 1
15	TioV2ckH	Port in valid before	PIC16C8X	0.30TCY + 30 *	—	—	ns	Note 1
		CLKOUT ↑	PIC16LC8X	0.30TCY + 80 *	—	—	ns	Note 1
16	TckH2ioI	Port in hold after CLKOUT ↑		0 *	—	—	ns	Note 1
17	TosH2ioV	OSC1↑ (Q1 cycle) to	PIC16C8X	—	—	125 *	ns	
		Port out valid	PIC16LC8X	—	—	250 *	ns	
18	TosH2ioI	OSC1↑ (Q2 cycle) to Port input invalid (I/O in hold time)		TBD	—	—	ns	
19	TioV2osH	Port input valid to OSC1↑ (I/O in setup time)		TBD	—	—	ns	
20	TioR	Port output rise time	PIC16C8X	—	10	25 *	ns	
20A			PIC16LC8X	—	10	60 *	ns	
21	TioF	Port output fall time	PIC16C8X	—	10	25 *	ns	
21A			PIC16LC8X	—	10	60 *	ns	
22	Tinp	INT pin high	PIC16C8X	20 *	—	—	ns	
22A		or low time	PIC16LC8X	55 *	—	—	ns	
23	Trbp	RB7:RB4 change INT	PIC16C8X	TOSC §	—	—	ns	
23A		high or low time	PIC16LC8X	TOSC §	—	—	ns	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

§ By design

Note 1: Measurements are taken in RC Mode where CLKOUT output is 4 x TOSC.

Preliminary © 1995 Microchip Technology Inc.

PIC16C8X

FIGURE 13-5: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING
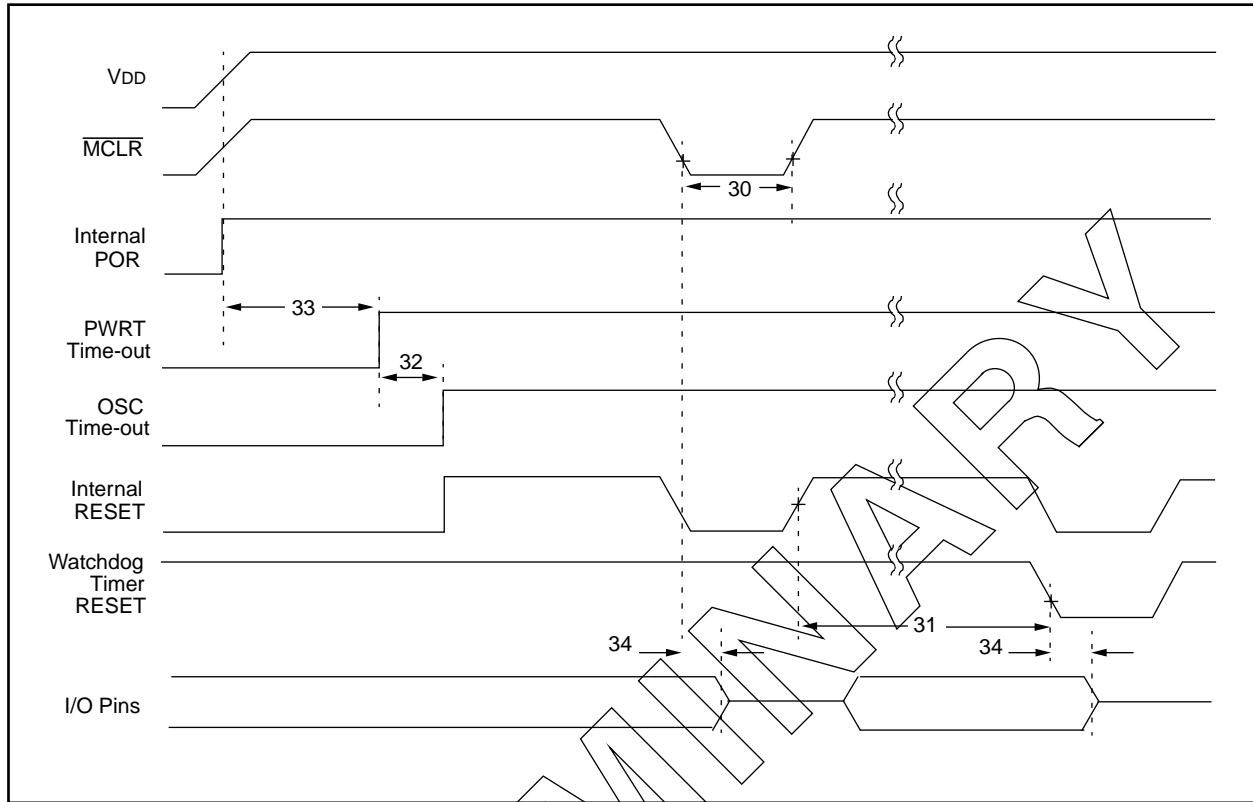

TABLE 13-5: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
30	TmcL	$\overline{MCLR}$ Pulse Width (low)	1000 *	—	—	ns	$2.0V \leq V_{DD} \leq 6.0V$
31	Twdt	Watchdog Timer Time-out Period (No Prescaler)	7 *	18	33 *	ms	$V_{DD} = 5.0V$
32	Tost	Oscillation Start-up Timer Period		1024T_{OSC}		ms	$T_{OSC} = OSC1$ period
33	Tpwrt	Power-up Timer Period	28 *	72	132 *	ms	$V_{DD} = 5.0V$
34	T_{IOZ}	I/O Hi-impedance from $\overline{MCLR}$ Low or reset	—	—	100 *	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.

PIC16C8X

FIGURE 13-6: TIMER0 CLOCK TIMINGS
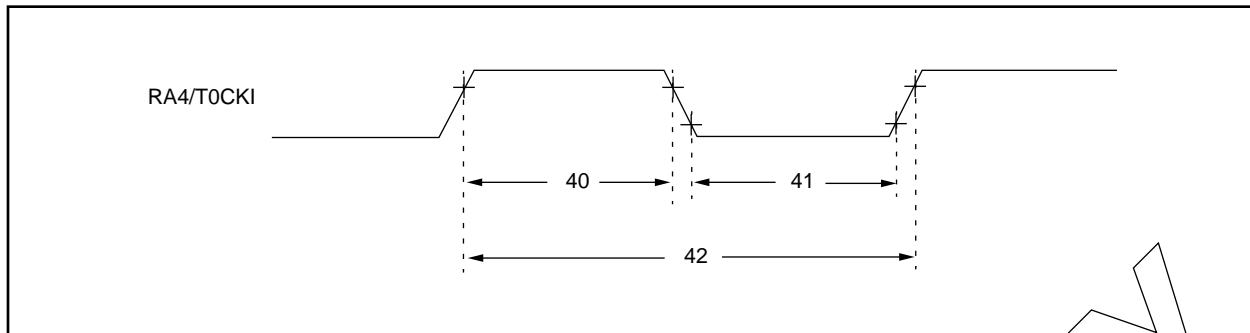
TABLE 13-6: TIMER0 CLOCK REQUIREMENTS

Parameter No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
40	Tt0H	T0CKI High Pulse Width	No Prescaler	0.5T_{CY} + 20 *	—	—	ns	
			With Prescaler	50 *	—	—	ns	2.0V ≤ VDD ≤ 3.0V
				30 *	—	—	ns	3.0V ≤ VDD ≤ 6.0V
41	Tt0L	T0CKI Low Pulse Width	No Prescaler	0.5T_{CY} + 20 *	—	—	ns	
			With Prescaler	50 *	—	—	ns	2.0V ≤ VDD ≤ 3.0V
				20 *	—	—	ns	3.0V ≤ VDD ≤ 6.0V
42	Tt0P	T0CKI Period		$\dfrac{T_{CY} + 40 *}{N}$	—	—	ns	N = prescale value (2, 4, ..., 256)

*   These parameters are characterized but not tested.

†   Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Preliminary

14.0 DC & AC CHARACTERISTICS GRAPHS/TABLES FOR PIC16C83, PIC16CR83, PIC16C84A, AND PIC16CR84

**Device Characteristics Not Available at This Time**

PIC16C8X

TABLE 14-1:   INPUT CAPACITANCE *

Pin Name	Typical Capacitance (pF)	
	18L PDIP	**18L SOIC**
PORTA	5.0	4.3
PORTB	5.0	4.3
$\overline{\text{MCLR}}$	17.0	17.0
OSC1/CLKIN	4.0	3.5
OSC2/CLKOUT	4.3	3.5
T0CKI	3.2	2.8

*   All capacitance values are typical at 25°C. A part to part variation of ±25% (three standard deviations) should be taken into account.

15.0   PACKAGING INFORMATION

15.1   18-Lead Plastic Dual In-line (300 mil)
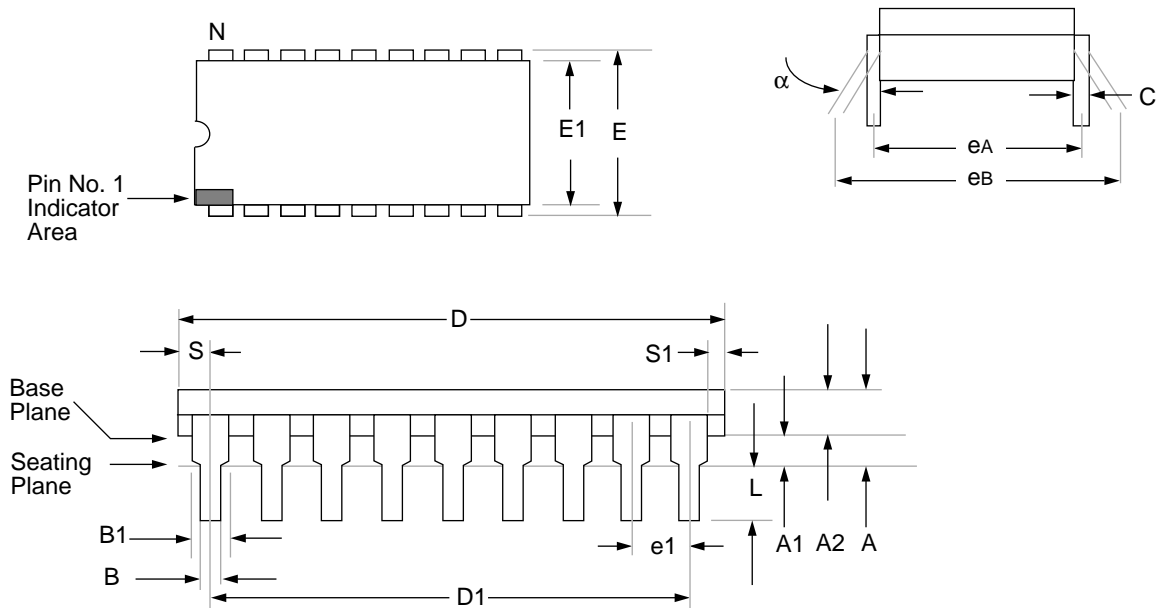
Package Group: Plastic Dual In-Line (PLA)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
α	0°	10°		0°	10°	
A	–	4.064		–	0.160	
A1	0.381	–		0.015	–	
A2	3.048	3.810		0.120	0.150	
B	0.355	0.559		0.014	0.022	
B1	1.524	1.524	**Reference**	0.060	0.060	**Reference**
C	0.203	0.381	**Typical**	0.008	0.015	**Typical**
D	22.479	23.495		0.885	0.925	
D1	20.320	20.320	**Reference**	0.800	0.800	**Reference**
E	7.620	8.255		0.300	0.325	
E1	6.096	7.112		0.240	0.280	
e1	2.489	2.591	**Typical**	0.098	0.102	**Typical**
eA	7.620	7.620	**Reference**	0.300	0.300	**Reference**
eB	7.874	9.906		0.310	0.390	
L	3.048	3.556		0.120	0.140	
N	18	18		18	18	
S	0.889	–		0.035	–	
S1	0.127	–		0.005	–	

PIC16C8X

15.2   18-Lead Plastic Surface Mount (SOIC - Wide, 300 mil Body)
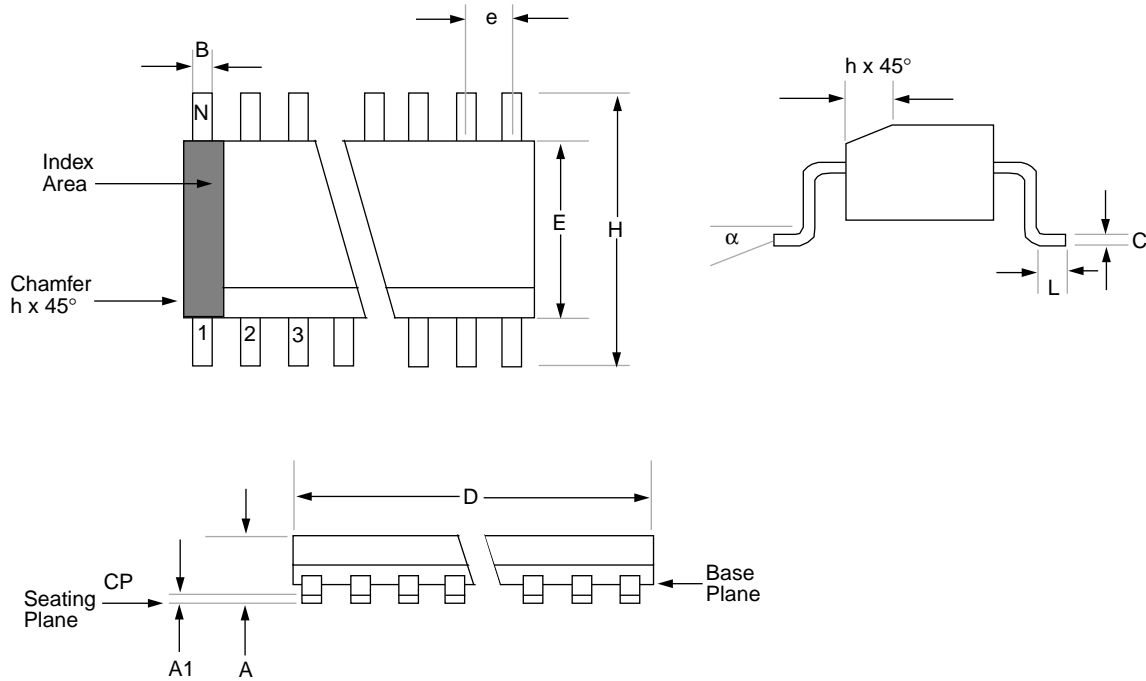
Package Group: Plastic SOIC (SO)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
α	0°	8°		0°	8°	
A	2.362	2.642		0.093	0.104	
A1	0.101	0.300		0.004	0.012	
B	0.355	0.483		0.014	0.019	
C	0.241	0.318		0.009	0.013	
D	11.353	11.735		0.447	0.462	
E	7.416	7.595		0.292	0.299	
e	1.270	1.270	**Reference**	0.050	0.050	**Reference**
H	10.007	10.643		0.394	0.419	
h	0.381	0.762		0.015	0.030	
L	0.406	1.143		0.016	0.045	
N	18	18		18	18	
CP	–	0.102		–	0.004	

15.3   Package Marking Information

18L PDIP

```
MMMMMMMMMMMMXXX
MMMMMMMMXXXXXXXX
      AABB  CDE
```

Example

```
PIC16C84
10I/P
      9305  CBA
```

18L SOIC

```
XXXXXXXX
XXXXXXXX

   AABB  CDE
```

Example

```
PIC16LC84
04I/S0218

   9310  CBA
```
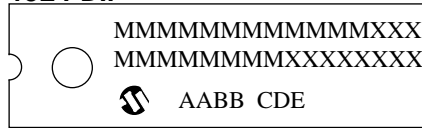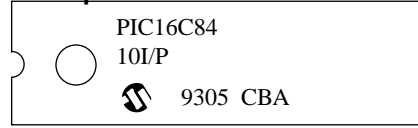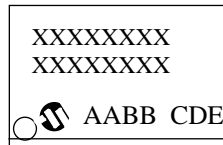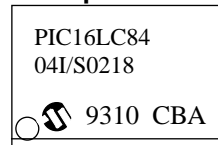
Legend: MM...M	Microchip part number information
XX...X	Customer specific information*
AA	Year code (last two digits of calendar year)
BB	Week code (week of January 1 is week '01')
C	Facility code of the plant at which wafer is manufactured
	C = Chandler, Arizona, U.S.A.,
	S = Tempe, Arizona, U.S.A.
D	Mask revision number
E	Assembly code of the plant or country of origin in which
	part was assembled
Note:	In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line thus limiting the number of available characters for customer specific information.

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

NOTES:

PIC16C8X

APPENDIX A: CHANGES

The following is the list of modifications over the PIC16C5X microcontroller family:

1. Instruction word length is increased to 14 bits. This allows larger page sizes both in program memory (2K now as opposed to 512 before) and the register file (128 bytes now versus 32 bytes before).

2. A PC latch register (PCLATH) is added to handle program memory paging. PA2, PA1 and PA0 bits are removed from the status register and placed in the option register.

3. Data memory paging is redefined slightly. The STATUS register is modified.

4. Four new instructions have been added: RETURN, RETFIE, ADDLW, and SUBLW. Two instructions, TRIS and OPTION, are being phased out although they are kept for compatibility with PIC16C5X.

5. OPTION and TRIS registers are made addressable.

6. Interrupt capability is added. Interrupt vector is at 0004h.

7. Stack size is increased to 8 deep.

8. Reset vector is changed to 0000h.

9. Reset of all registers is revisited. Five different reset (and wake-up) types are recognized. Registers are reset differently.

10. Wake up from SLEEP through interrupt is added.

11. Two separate timers, the Oscillator Start-up Timer (OST) and Power-up Timer (PWRT), are included for more reliable power-up. These timers are invoked selectively to avoid unnecessary delays on power-up and wake-up.

12. PORTB has weak pull-ups and interrupt on change features.

13. T0CKI pin is also a port pin (RA4/T0CKI).

14. FSR is a full 8-bit register.

15. "In system programming" is made possible. The user can program PIC16CXX devices using only five pins: VDD, VSS, VPP, RB6 (clock) and RB7 (data in/out).

APPENDIX B: COMPATIBILITY

To convert code written for PIC16C5X to PIC16C8X, the user should take the following steps:

1. Remove any program memory page select operations (PA2, PA1, PA0 bits) for CALL, GOTO.

2. Revisit any computed jump operations (write to PC or add to PC, etc.) to make sure page bits are set properly under the new scheme.

3. Eliminate any data memory page switching. Redefine data variables for reallocation.

4. Verify all writes to STATUS, OPTION, and FSR registers since these have changed.

5. Change reset vector to 0000h.

PIC16C8X

APPENDIX C: WHAT'S NEW

Here is an overview list of new features:

- Added descriptions and information for the PIC16C84A, PIC16CR84. PIC16C83 and PIC16CR83 devices.

APPENDIX D: WHAT'S CHANGED

The following lists the things that have changed:

1. Section 7.1 no longer says that the upper two bits of EEADR are not address decoded. These two bits are decoded, even when there is no data EEPROM at that location.
2. Changed the format of the Register Definition Figures to be consistent with other new data sheets.
3. Added parameter numbers to DC specs.
4. Added description of MPLAB development tool software.
5. Added new probes in Probe Specification table.
6. Updated device family tables.

APPENDIX E: PIC16C84 CONVERSION CONSIDERATIONS

This appendix discusses some of the issues that you may encounter as you convert your design from a PIC16C84 to any of the newly introduced devices. These new devices are:

• PIC16C83
• PIC16CR83
• PIC16C84A
• PIC16CR84

Some of the issues that may be encountered are:

1. The polarity of the PWRTE configuration bit has been reversed. Ensure that the programmer has this bit correctly set before programming.

2. The PIC16C84A and PIC16CR84 have larger RAM sizes. Ensure that this does not cause an issue with your program.

3. The $\overline{\text{MCLR}}$ pin now has an on-chip filter. The input signal on the MCLR pin will require a longer low pulse to generate an interrupt.

4. Many electrical specifications have been improved. Compare the electrical specifications of the two devices to ensure that this will not cause a compatibility issue.

NOTES: