

Welcome to E-XFL.COM

What is "[Embedded - Microcontrollers](#)"?

"[Embedded - Microcontrollers](#)" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "[Embedded - Microcontrollers](#)"

Details

Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	4MHz
Connectivity	-
Peripherals	POR, WDT
Number of I/O	13
Program Memory Size	1.75KB (1K x 14)
Program Memory Type	FLASH
EEPROM Size	64 x 8
RAM Size	68 x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 6V
Data Converters	-
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	18-DIP (0.300", 7.62mm)
Supplier Device Package	18-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lf84-04i-p

PIC16F8X

Table of Contents

1.0	General Description	3
2.0	PIC16F8X Device Varieties	5
3.0	Architectural Overview	7
4.0	Memory Organization	11
5.0	I/O Ports	21
6.0	Timer0 Module and TMR0 Register	27
7.0	Data EEPROM Memory	33
8.0	Special Features of the CPU	37
9.0	Instruction Set Summary	53
10.0	Development Support	69
11.0	Electrical Characteristics for PIC16F83 and PIC16F84	73
12.0	Electrical Characteristics for PIC16CR83 and PIC16CR84	85
13.0	DC & AC Characteristics Graphs/Tables	97
14.0	Packaging Information	109
Appendix A:	Feature Improvements - From PIC16C5X To PIC16F8X	113
Appendix B:	Code Compatibility - from PIC16C5X to PIC16F8X	113
Appendix C:	What's New In This Data Sheet	114
Appendix D:	What's Changed In This Data Sheet	114
Appendix E:	Conversion Considerations - PIC16C84 to PIC16F83/F84 And PIC16CR83/CR84	115
Index	117
On-Line Support	119
Reader Response	120
PIC16F8X Product Identification System	121
Sales and Support	121

To Our Valued Customers

We constantly strive to improve the quality of all our products and documentation. We have spent a great deal 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.

PIC16F8X

NOTES:

PIC16F8X

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.

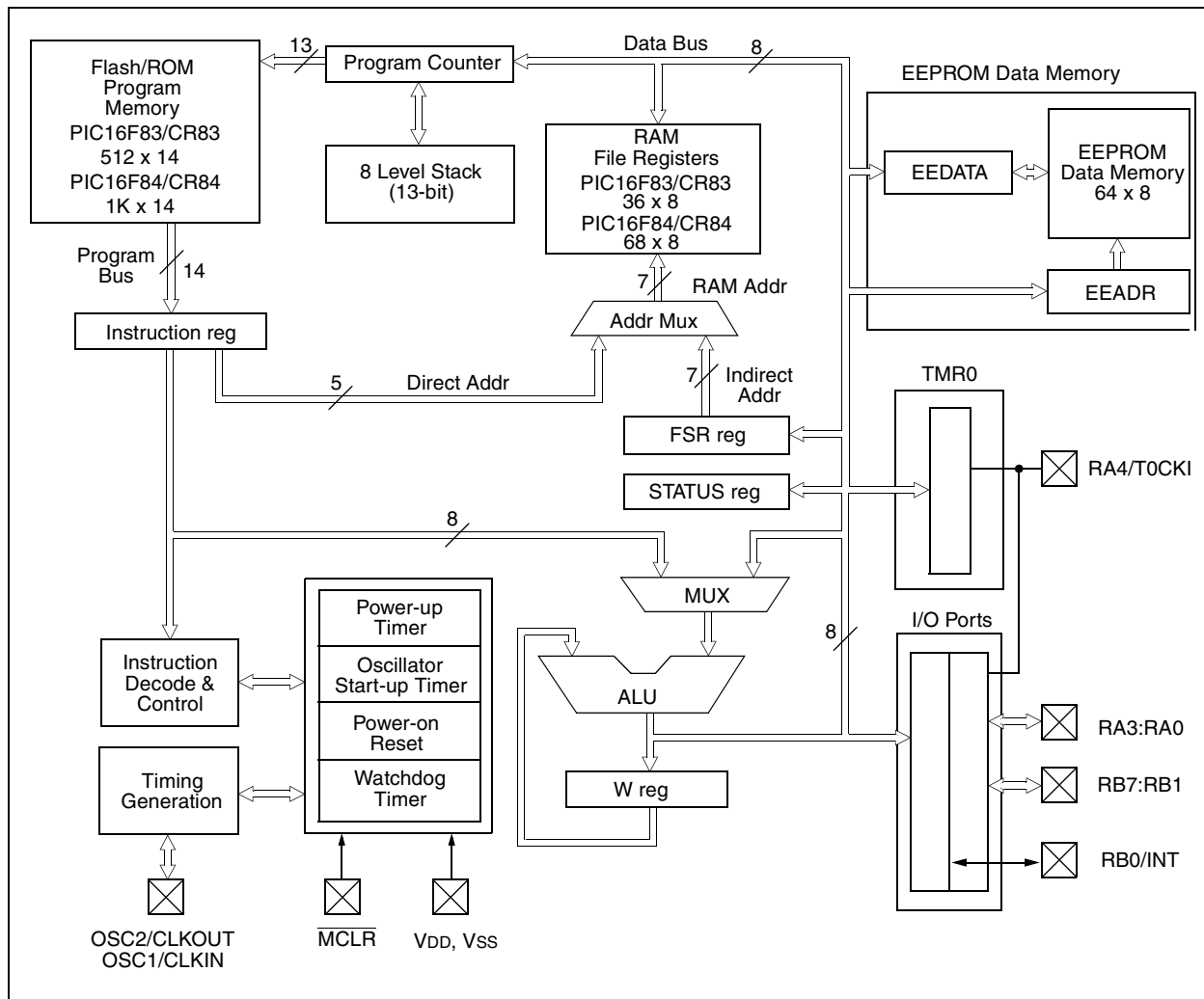
The ALU is 8-bits wide and capable of addition, subtraction, shift and logical operations. Unless otherwise mentioned, arithmetic operations are two's complement in nature. In two-operand instructions, typically one operand is the working register (W register), and the other operand is a file register 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 borrow and digit borrow out bit, respectively, in subtraction. See the `SUBLW` and `SUBWF` instructions for examples.

A simplified block diagram for the PIC16F8X is shown in Figure 3-1, its corresponding pin description is shown in Table 3-1.

FIGURE 3-1: PIC16F8X BLOCK DIAGRAM



4.0 MEMORY ORGANIZATION

There are two memory blocks in the PIC16F8X. 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 PIC16FXX has a 13-bit program counter capable of addressing an 8K x 14 program memory space. For the PIC16F83 and PIC16CR83, the first 512 x 14 (0000h-01FFh) are physically implemented (Figure 4-1). For the PIC16F84 and PIC16CR84, the first 1K x 14 (0000h-03FFh) are physically implemented (Figure 4-2). Accessing a location above the physically implemented address will cause a wrap-around. For example, for the PIC16F84 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: PROGRAM MEMORY MAP AND STACK - PIC16F83/CR83

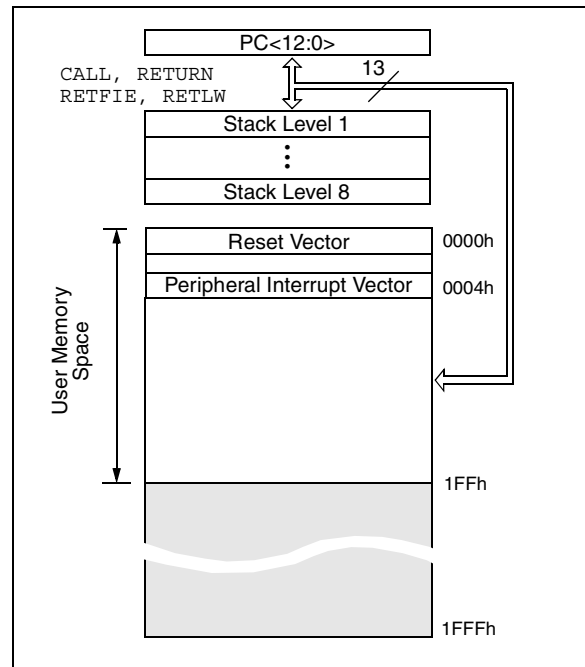
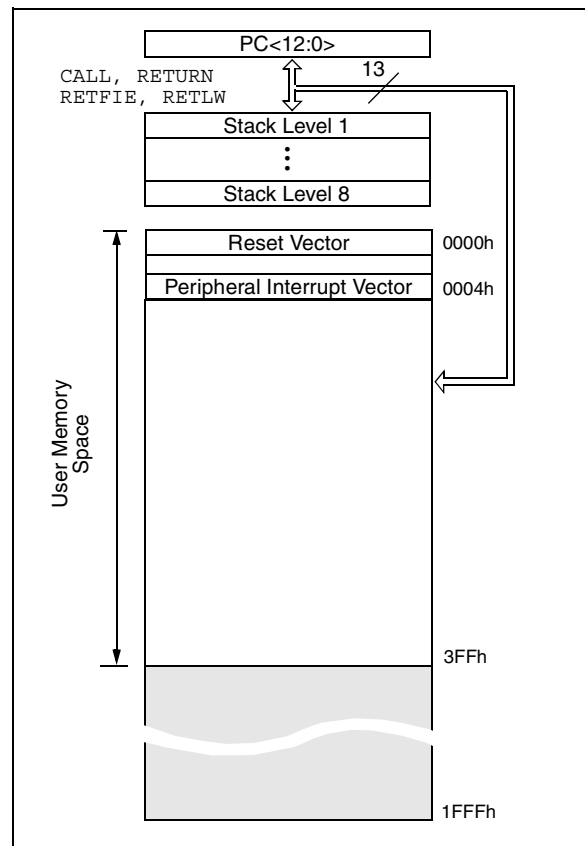


FIGURE 4-2: PROGRAM MEMORY MAP AND STACK - PIC16F84/CR84



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 5-5). 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.

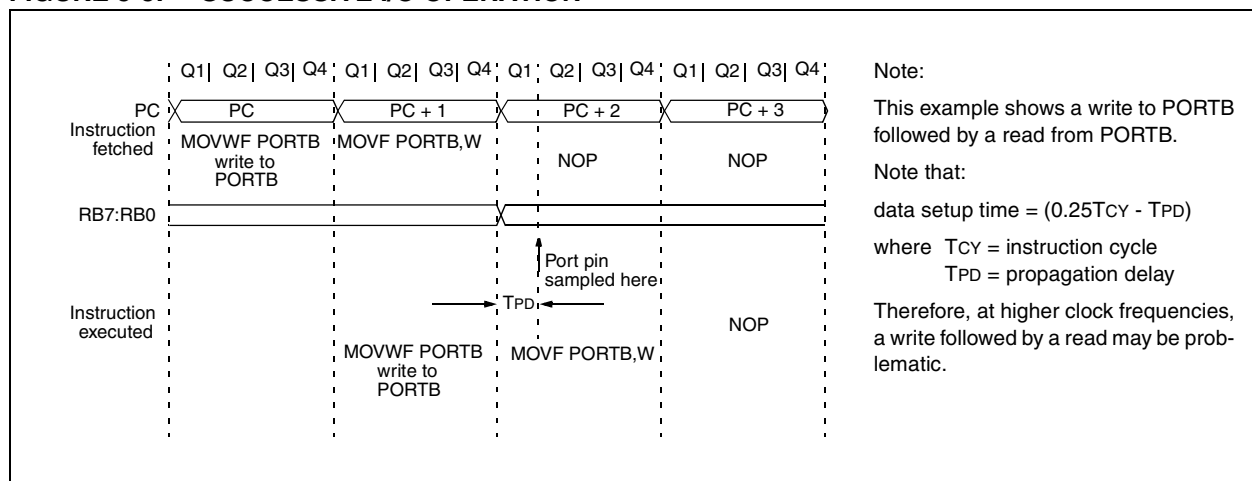
Example 5-1 shows the effect of two sequential read-modify-write instructions (e.g., `BCF`, `BSF`, etc.) on an I/O port.

EXAMPLE 5-1: 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 5-5: SUCCESSIVE I/O OPERATION



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 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 7-1: DATA EEPROM WRITE

```
BSF     STATUS, RP0    ; Bank 1
BCF     INTCON, GIE    ; Disable INTs.
BSF     EECON1, WREN   ; Enable Write
MOVLW  55h             ;
```

Required Sequence	MOVWF	EECON2	; Write 55h
	MOVLW	AAh	;
	MOVWF	EECON2	; Write AAh
	BSF	EECON1,WR	; Set WR bit
			; begin write
	BSF	INTCON, GIE	; Enable INTs.

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.

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 PIC16F8X 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 PIC16F8X 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.

8.4 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 $\overline{\text{MCLR}}$ 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 met 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.

8.5 Power-up Timer (PWRT)

The Power-up Timer (PWRT) provides a fixed 72 ms nominal time-out (T_{PWRT}) from POR (Figure 8-10, Figure 8-11, Figure 8-12 and Figure 8-13). 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 VDD to rise to an acceptable level (Possible exception shown in Figure 8-13).

A configuration bit, PWRTE, can enable/disable the PWRT. See either Figure 8-1 or Figure 8-2 for the operation of the PWRTE bit for a particular device.

The power-up time delay T_{PWRT} will vary from chip to chip due to VDD, temperature, and process variation. See DC parameters for details.

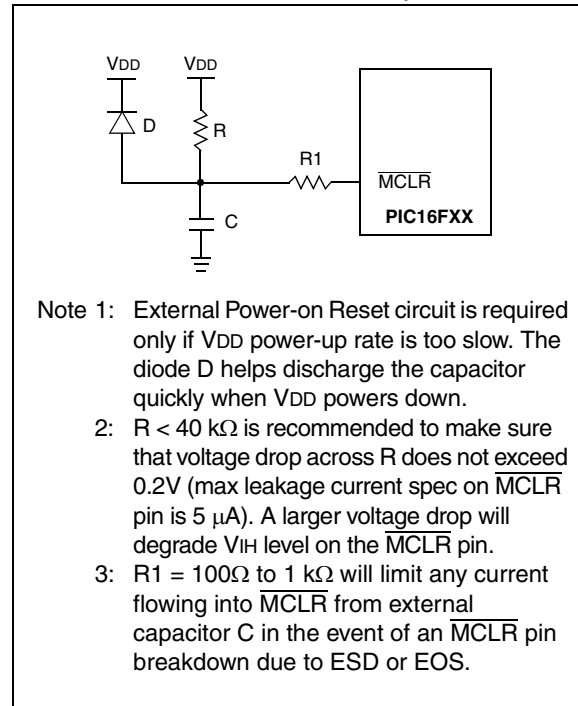
8.6 Oscillator Start-up Timer (OST)

The Oscillator Start-up Timer (OST) provides a 1024 oscillator cycle delay (from OSC1 input) after the PWRT delay ends (Figure 8-10, Figure 8-11, Figure 8-12 and Figure 8-13). This ensures the crystal oscillator or resonator has started and stabilized.

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

When VDD rises very slowly, it is possible that the T_{PWRT} time-out and T_{OST} time-out will expire before VDD has reached its final value. In this case (Figure 8-13), an external power-on reset circuit may be necessary (Figure 8-9).

FIGURE 8-9: EXTERNAL POWER-ON RESET CIRCUIT (FOR SLOW VDD POWER-UP)



PIC16F8X

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

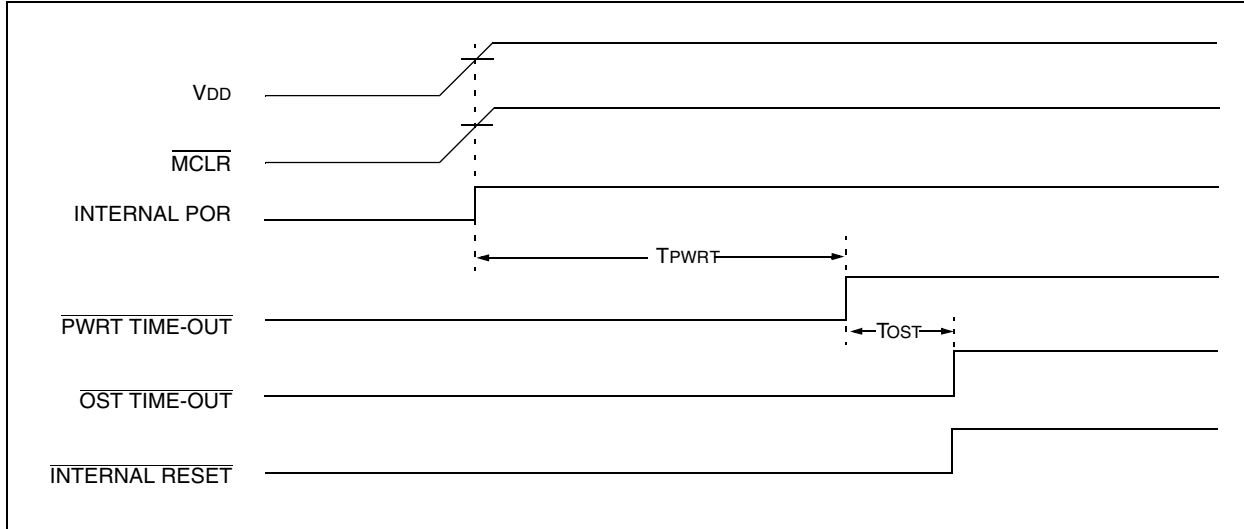
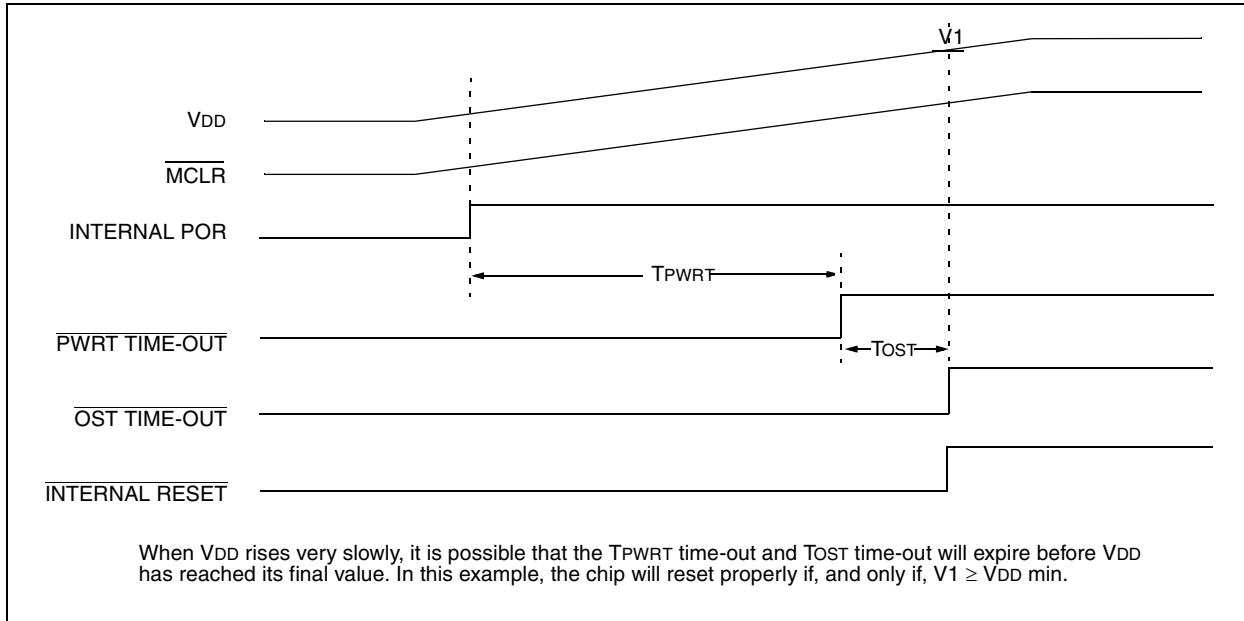


FIGURE 8-13: TIME-OUT SEQUENCE ON POWER-UP ($\overline{\text{MCLR}}$ TIED TO V_{DD}): SLOW V_{DD} RISE TIME



8.11 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).

8.11.1 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_REG 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{TO} bit in the STATUS register will be cleared upon a WDT time-out.

8.11.2 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



TABLE 8-7 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)	(2)	PWRTE ⁽¹⁾	WDTE	FOSC1	FOSC0	(2)	
81h	OPTION_REG	RBPUP	INTEDG	TOCS	TOSE	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 and Figure 8-2 for operation of the PWRTE bit.

2: See Figure 8-1, Figure 8-2 and Section 8.13 for operation of the Code and Data protection bits.

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.

For **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 zero, the result is placed in the W register. If 'd' is one, the result is placed in the file register specified in the instruction.

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

For **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
TO	Time-out bit
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
<i>italics</i>	User defined term (font is courier)

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

- **Byte-oriented** operations
- **Bit-oriented** operations
- **Literal and control** operations

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

Table 9-2 lists the instructions recognized by the MPASM assembler.

Figure 9-1 shows the general formats that the instructions can have.

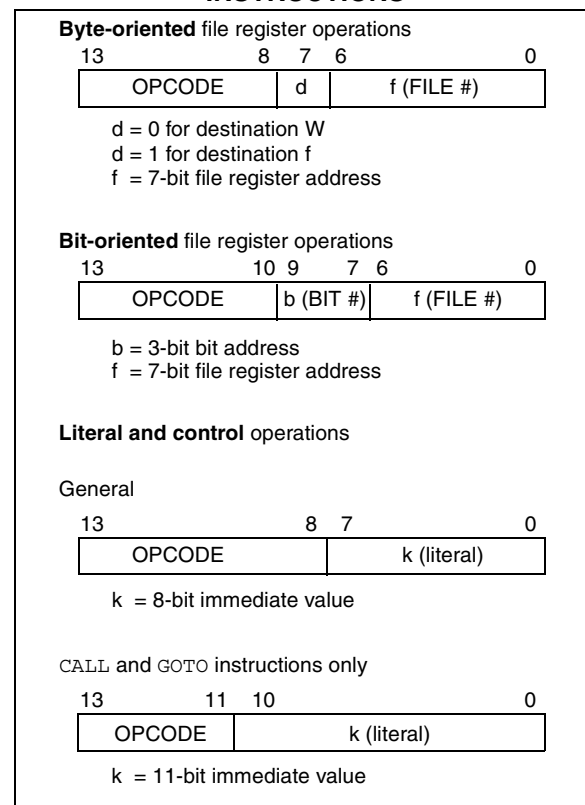
Note: To maintain upward compatibility with future PIC16CXX products, do not use the `OPTION` and `TRIS` instructions.

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



PIC16F8X

BCF	Bit Clear f								
Syntax:	[label] BCF f,b								
Operands:	0 ≤ f ≤ 127 0 ≤ b ≤ 7								
Operation:	0 → (f)								
Status Affected:	None								
Encoding:	<table border="1"> <tr> <td>01</td> <td>00bb</td> <td>bfff</td> <td>ffff</td> </tr> </table>	01	00bb	bfff	ffff				
01	00bb	bfff	ffff						
Description:	Bit 'b' in register 'f' is cleared.								
Words:	1								
Cycles:	1								
Q Cycle Activity:	<table border="1"> <thead> <tr> <th>Q1</th> <th>Q2</th> <th>Q3</th> <th>Q4</th> </tr> </thead> <tbody> <tr> <td>Decode</td> <td>Read register 'f'</td> <td>Process data</td> <td>Write register 'f'</td> </tr> </tbody> </table>	Q1	Q2	Q3	Q4	Decode	Read register 'f'	Process data	Write register 'f'
Q1	Q2	Q3	Q4						
Decode	Read register 'f'	Process data	Write register 'f'						

Example

```

BCF    FLAG_REG, 7

Before Instruction
      FLAG_REG = 0xC7
After Instruction
      FLAG_REG = 0x47
    
```

BSF	Bit Set f								
Syntax:	[label] BSF f,b								
Operands:	0 ≤ f ≤ 127 0 ≤ b ≤ 7								
Operation:	1 → (f)								
Status Affected:	None								
Encoding:	<table border="1"> <tr> <td>01</td> <td>01bb</td> <td>bfff</td> <td>ffff</td> </tr> </table>	01	01bb	bfff	ffff				
01	01bb	bfff	ffff						
Description:	Bit 'b' in register 'f' is set.								
Words:	1								
Cycles:	1								
Q Cycle Activity:	<table border="1"> <thead> <tr> <th>Q1</th> <th>Q2</th> <th>Q3</th> <th>Q4</th> </tr> </thead> <tbody> <tr> <td>Decode</td> <td>Read register 'f'</td> <td>Process data</td> <td>Write register 'f'</td> </tr> </tbody> </table>	Q1	Q2	Q3	Q4	Decode	Read register 'f'	Process data	Write register 'f'
Q1	Q2	Q3	Q4						
Decode	Read register 'f'	Process data	Write register 'f'						

Example

```

BSF    FLAG_REG, 7

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

BTFSC	Bit Test, Skip if Clear								
Syntax:	[label] BTFSC f,b								
Operands:	0 ≤ f ≤ 127 0 ≤ b ≤ 7								
Operation:	skip if (f) = 0								
Status Affected:	None								
Encoding:	<table border="1"> <tr> <td>01</td> <td>10bb</td> <td>bfff</td> <td>ffff</td> </tr> </table>	01	10bb	bfff	ffff				
01	10bb	bfff	ffff						
Description:	If bit 'b' in register 'f' is '1' then the next instruction is executed. If bit 'b', in register 'f', is '0' then the next instruction is discarded, and a NOP is executed instead, making this a 2TCY instruction.								
Words:	1								
Cycles:	1(2)								
Q Cycle Activity:	<table border="1"> <thead> <tr> <th>Q1</th> <th>Q2</th> <th>Q3</th> <th>Q4</th> </tr> </thead> <tbody> <tr> <td>Decode</td> <td>Read register 'f'</td> <td>Process data</td> <td>No-Operation</td> </tr> </tbody> </table>	Q1	Q2	Q3	Q4	Decode	Read register 'f'	Process data	No-Operation
Q1	Q2	Q3	Q4						
Decode	Read register 'f'	Process data	No-Operation						

If Skip:

(2nd Cycle)

Q1	Q2	Q3	Q4
No-Operation	No-Operation	No-Operation	No-Operation

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

PIC16F8X

10.6 PICDEM-1 Low-Cost PIC MCU Demonstration Board

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

10.7 PICDEM-2 Low-Cost PIC16CXX Demonstration Board

The PICDEM-2 is a simple demonstration board that supports the PIC16C62, PIC16C64, PIC16C65, PIC16C73 and PIC16C74 microcontrollers. All the necessary hardware and software is included to run the basic demonstration programs. The user can program the sample microcontrollers provided with the PICDEM-2 board, on a PRO MATE II programmer or PICSTART-Plus, and easily test firmware. The PICMASTER emulator may also be used with the PICDEM-2 board to test firmware. Additional prototype area has been provided to the user for adding additional hardware and connecting it to the microcontroller socket(s). Some of the features include a RS-232 interface, push-button switches, a potentiometer for simulated analog input, a Serial EEPROM to demonstrate usage of the I²C bus and separate headers for connection to an LCD module and a keypad.

10.8 PICDEM-3 Low-Cost PIC16CXXX Demonstration Board

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

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

10.9 MPLAB™ Integrated Development Environment Software

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

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

MPLAB allows you to:

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

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

10.10 Assembler (MPASM)

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

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

MPASM allows full symbolic debugging from PICMASTER, Microchip's Universal Emulator System.

10.4 DC CHARACTERISTICS: PIC16F84, PIC16F83 (Commercial, Industrial) PIC16LF84, PIC16F83 (Commercial, Industrial)

DC Characteristics All Pins Except Power Supply Pins		Standard Operating Conditions (unless otherwise stated) Operating temperature $0^{\circ}\text{C} \leq T_A \leq +70^{\circ}\text{C}$ (commercial) $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (industrial) Operating voltage V_{DD} range as described in DC spec Section 10.1 and Section 10.2.					
Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
Capacitive Loading Specs on Output Pins							
D100	Cosc2	OSC2 pin	—	—	15	pF	In XT, HS and LP modes when external clock is used to drive OSC1.
D101	Cio	All I/O pins and OSC2 (RC mode)	—	—	50	pF	
Data EEPROM Memory							
D120	Ed	Endurance	1M	10M	—	EW	25°C at 5V V _{MIN} = Minimum operating voltage
D121	VDRW	V _{DD} for read/write	V _{MIN}	—	6.0	V	
D122	TDEW	Erase/Write cycle time	—	10	20*	ms	
Program Flash Memory							
D130	EP	Endurance	100	1000	—	EW	V _{MIN} = Minimum operating voltage
D131	VPR	V _{DD} for read	V _{MIN}	—	6.0	V	
D132	VPEW	V _{DD} for erase/write	4.5	—	5.5	V	
D133	TPEW	Erase/Write cycle time	—	10	—	ms	

* 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

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

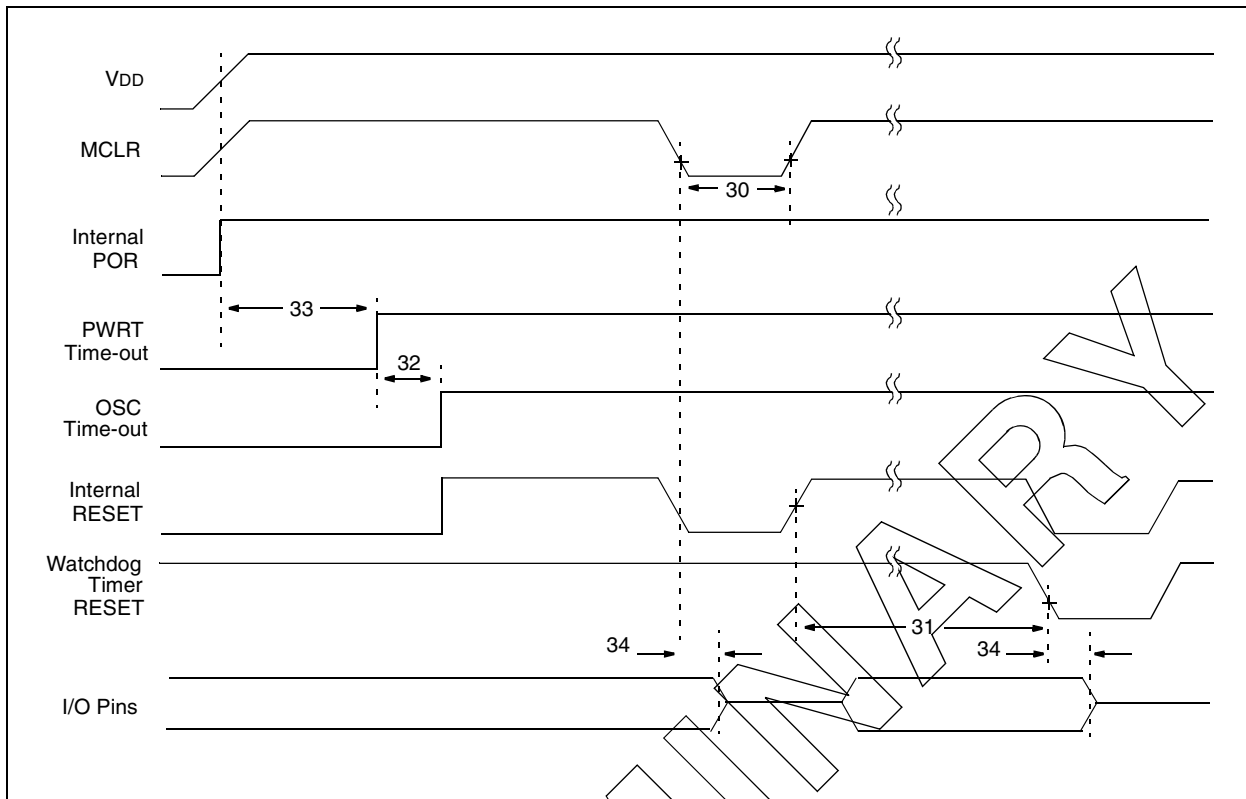


TABLE 11-5 RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER REQUIREMENTS

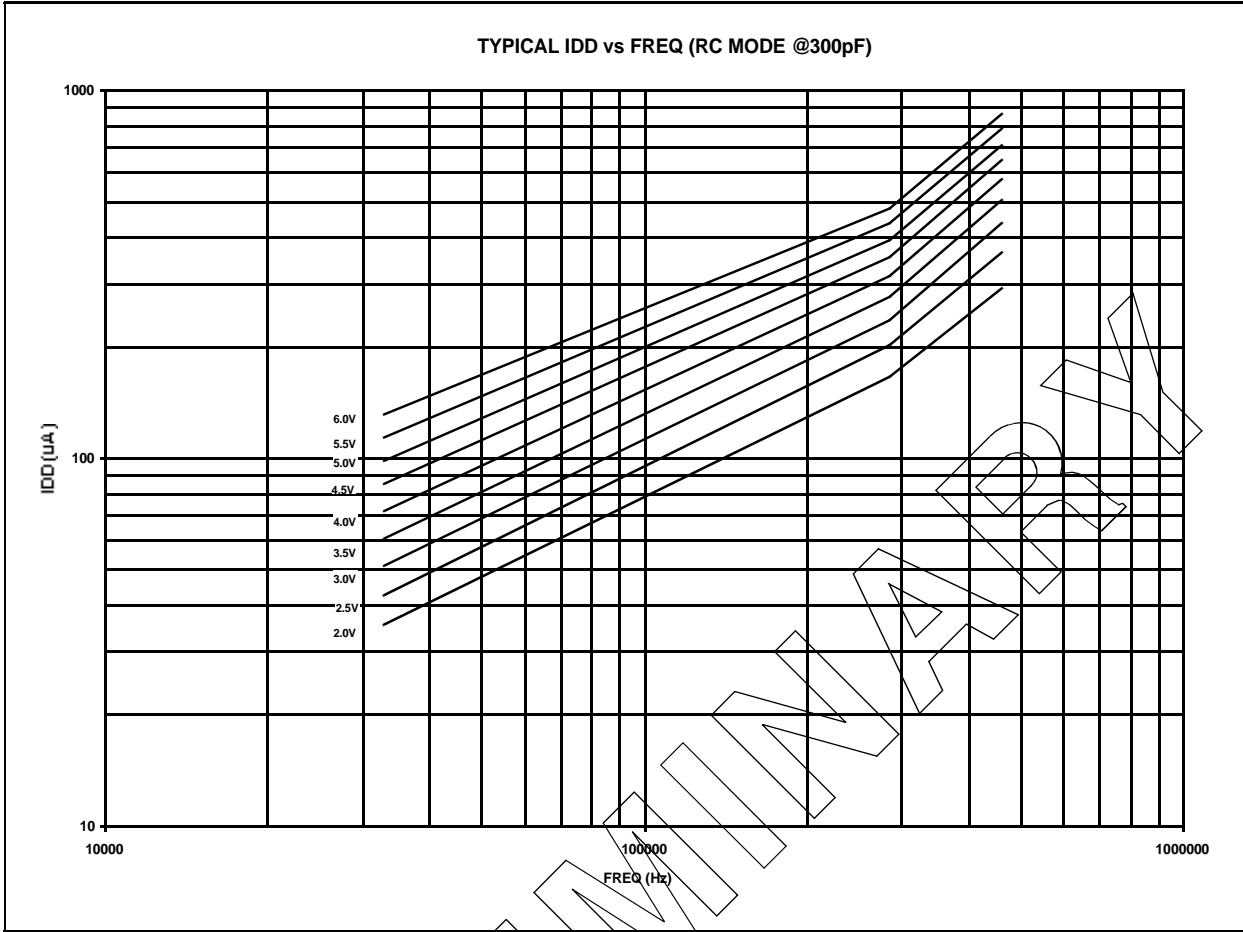
Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
30	Tmcl	MCLR Pulse Width (low)	1000 *	—	—	ns	$2.0V \leq V_{DD} \leq 6.0V$
31	Twdt	Watchdog Timer Time-out Period (No Prescaler)	7 *	18	33 *	ms	$V_{DD} = 5.0V$
32	Tost	Oscillation Start-up Timer Period	—	$1024T_{osc}$	—	ms	$T_{osc} = OSC1$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

PIC16F8X

FIGURE 12-12: TYPICAL I_{DD} vs. FREQUENCY (RC MODE @300PF, 25°C)



PIC16F8X

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

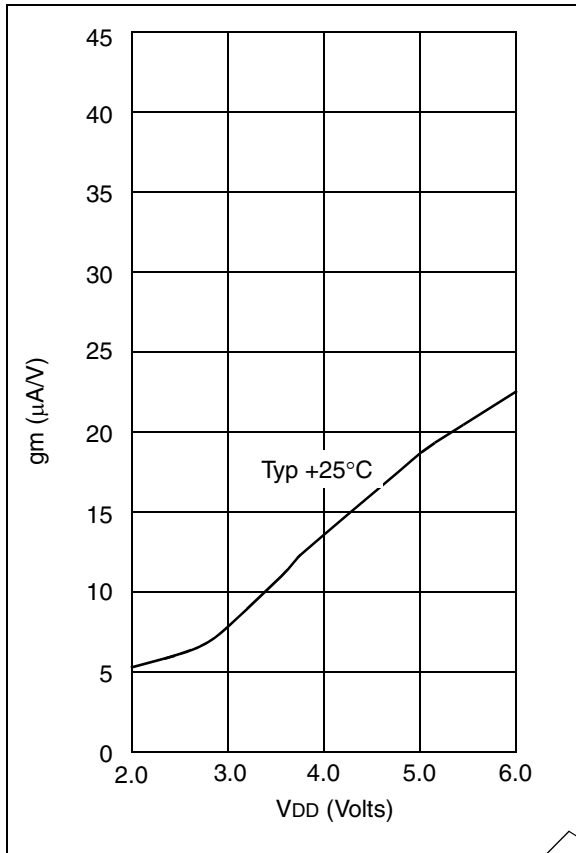
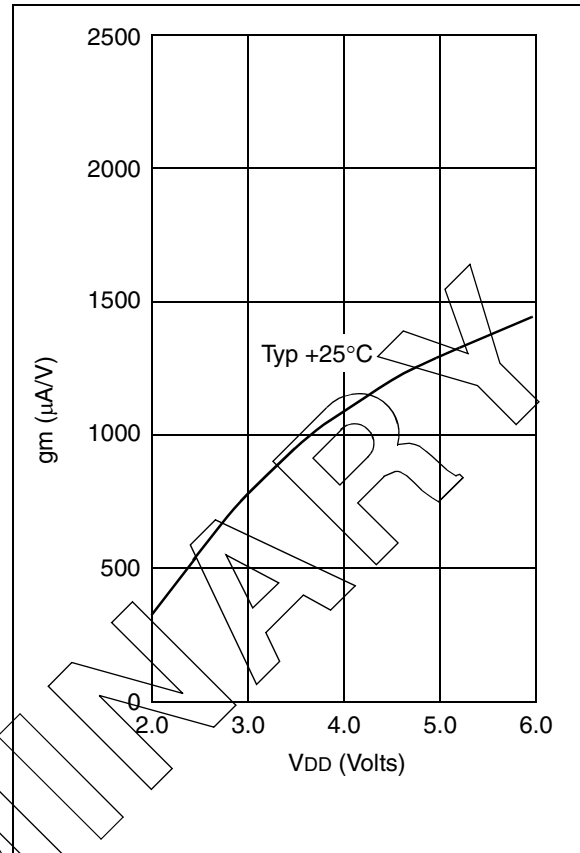


FIGURE 12-16: TRANSCONDUCTANCE (gm) OF XT OSCILLATOR vs. VDD



PRELIMINARY

PIC16F8X

FIGURE 12-21: TYPICAL DATA MEMORY ERASE/WRITE CYCLE TIME VS. VDD

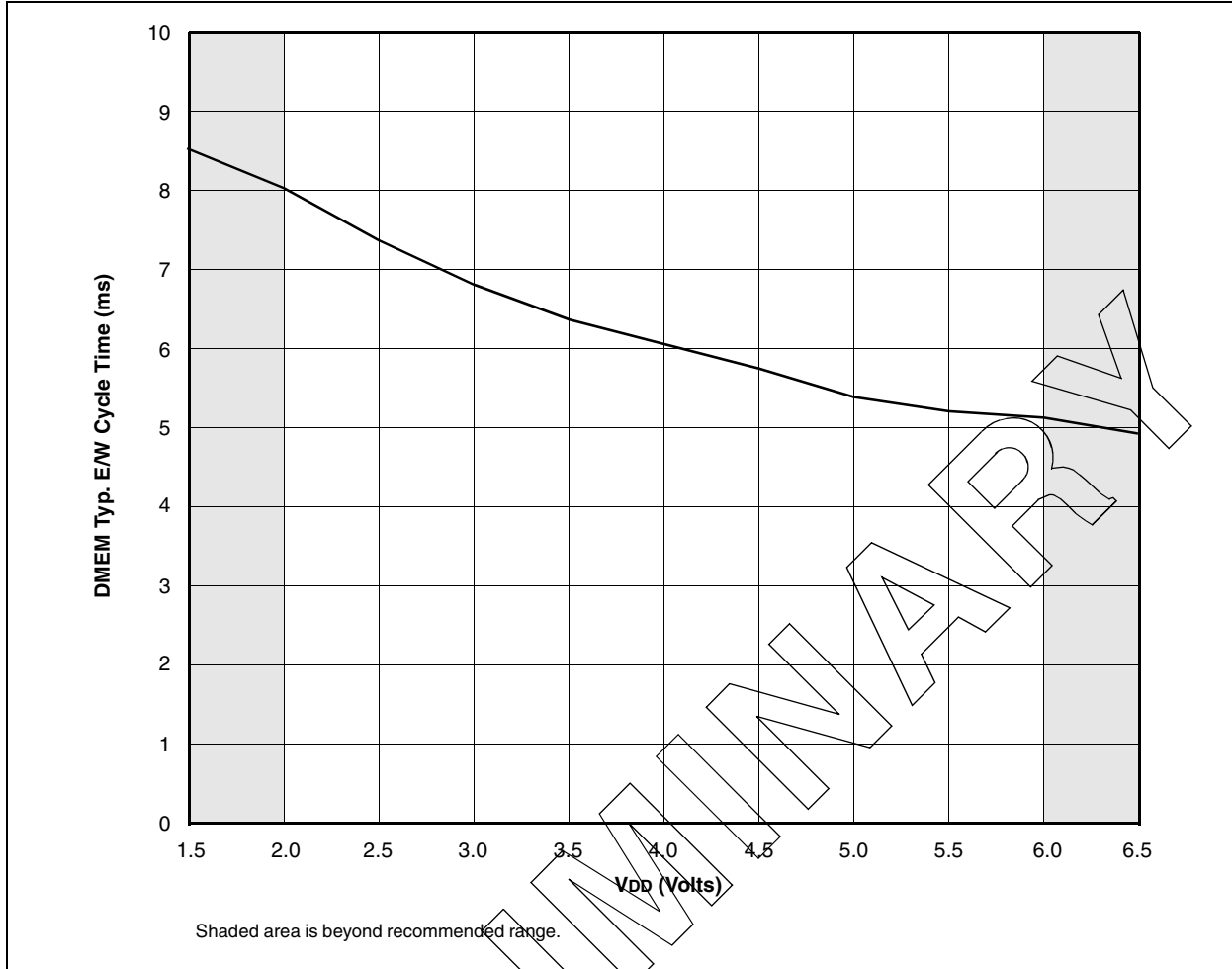


TABLE 12-2 INPUT CAPACITANCE*

Pin Name	Typical Capacitance (pF)	
	18L PDIP	18L SOIC
PORTA	5.0	4.3
PORTB	5.0	4.3
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 $\pm 25\%$ (three standard deviations) should be taken into account.

PIC16F8X

P

Paging, Program Memory	18
PCL	18, 42
PCLATH	18, 42
\overline{PD}	15, 41, 46
PICDEM-1 Low-Cost PIC MCU Demo Board	70
PICDEM-2 Low-Cost PIC16CXX Demo Board	70
PICDEM-3 Low-Cost PIC16CXXX Demo Board	70
PICMASTER® In-Circuit Emulator	69
PICSTART® Plus Entry Level Development System	69
Pinout Descriptions	9
POR	43
Oscillator Start-up Timer (OST)	37, 43
Power-on Reset (POR)	37, 42, 43
Power-up Timer (PWRT)	37, 43
Time-out Sequence	46
Time-out Sequence on Power-up	44
\overline{TO}	15, 41, 46
Port RB Interrupt	48
PORTA	9, 21, 42
PORTB	9, 23, 42
Power-down Mode (SLEEP)	51
Prescaler	29
PRO MATE® II Universal Programmer	69
Product Identification System	121

R

RBIF bit	23, 48
RC Oscillator	46
Read-Modify-Write	25
Register File	12
Reset	37, 41
Reset on Brown-Out	46

S

Saving W Register and STATUS in RAM	49
SEEVAL® Evaluation and Programming System	71
SLEEP	37, 41, 51
Software Simulator (MPLAB-SIM)	71
Special Features of the CPU	37
Special Function Registers	12
Stack	18
Overflows	18
Underflows	18
STATUS	7, 15, 42

T

time-out	42
Timer0	
Switching Prescaler Assignment	31
T0IF	48
Timer0 Module	27
TMR0 Interrupt	48
TMR0 with External Clock	29
Timing Diagrams	
Time-out Sequence	44
Timing Diagrams and Specifications	80, 92
TRISA	21
TRISB	23, 42

W

W	42
Wake-up from SLEEP	42, 51
Watchdog Timer (WDT)	37, 41, 42, 50
WDT	42
Period	50

Programming Considerations	50
Time-out	42

X

XT	46
----------	----

Z

Zero bit	7
----------------	---

ON-LINE SUPPORT

Microchip provides on-line support on the Microchip World Wide Web (WWW) site.

The web site is used by Microchip as a means to make files and information easily available to customers. To view the site, the user must have access to the Internet and a web browser, such as Netscape or Microsoft Explorer. Files are also available for FTP download from our FTP site.

Connecting to the Microchip Internet Web Site

The Microchip web site is available by using your favorite Internet browser to attach to:

www.microchip.com

The file transfer site is available by using an FTP service to connect to:

<ftp://ftp.futureone.com/pub/microchip>

The web site and file transfer site provide a variety of services. Users may download files for the latest Development Tools, Data Sheets, Application Notes, User's Guides, Articles and Sample Programs. A variety of Microchip specific business information is also available, including listings of Microchip sales offices, distributors and factory representatives. Other data available for consideration is:

- Latest Microchip Press Releases
- Technical Support Section with Frequently Asked Questions
- Design Tips
- Device Errata
- Job Postings
- Microchip Consultant Program Member Listing
- Links to other useful web sites related to Microchip Products
- Conferences for products, Development Systems, technical information and more
- Listing of seminars and events