Microchip

PIC16C84

8-Bit CMOS EEPROM Microcontroller

FEATURES

High performance RISC-like CPU

- Only 35 single word instructions to learn
- All instructions single cycle (400 ns) except for program branches which are two-cycle
- Operating speed: DC - 10 MHz clock input
  DC - 400 ns instruction cycle
- 14-bit wide instructions
- 8-bit wide data path
- 1024 x 14 on-chip EEPROM program memory
- 36 x 8 general purpose registers (SRAM)
- 15 special function hardware registers
- 64 x 8 EEPROM data memory
- 8 levels deep hardware stack
- Direct, indirect and relative addressing modes
- Four interrupt sources:
  - External INT pin
  - RTCC timer overflow
  - PortB<7:4> interrupt on change
  - Data EEPROM write complete

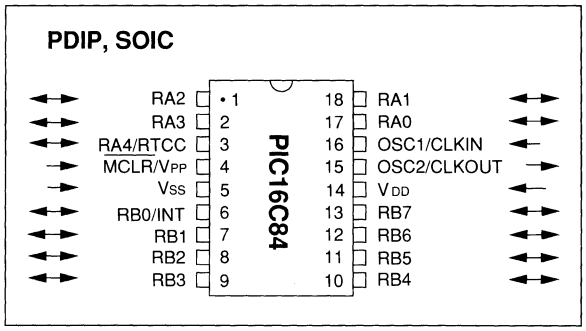
Peripheral features

- 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
- 8-bit real time clock/counter (RTCC) with 8-bit programmable prescaler

Special microcontroller features

- Power-on reset
- Power-up timer
- Oscillator start-up timer
- Watchdog timer (WDT) with its own on-chip RC oscillator for reliable operation
- Security EEPROM fuse for code-protection
- Power saving SLEEP mode
- User selectable oscillator options:
  - RC oscillator: RC
  - Crystal/resonator: XT
  - High-speed crystal/resonator: HS
  - Power-saving low-frequency crystal: LP
- Serial, In-System Programming (ISP) of EEPROM program and data memory using only two pins

FIGURE A - PIN CONFIGURATION



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
  - Automotive: 2.0V to 6.0V
- Low power consumption
  - < 2 mA @ 5V, 4 MHz
  - 15 µA typical @ 2V, 32 KHz
  - < 1 µA typical standby current @ 2V

INTRODUCTION

The PIC16C84 is a high-performance, low-cost, CMOS, fully-static 8-bit microcontroller with 1K x 14 EEPROM program memory and 64 bytes of EEPROM data memory. It is the second member of an enhanced family of PIC16CXX microcontrollers (customers familiar with the PIC16C5X products may refer to Appendix A for a list of enhancements).

Its high performance is due to instructions that are all single word (14-bit wide), which execute in single cycle (400 ns at 10 MHz clock) except for program-branches which take two cycles (800 ns).

The PIC16C84 has four interrupt sources and an eight level hardware stack.

The peripherals include an 8-bit timer/counter with an 8-bit prescaler (effectively a 16 bit timer) and 13 bi-directional I/O pins. The high current drive (25 mA max. sink, 20 mA max source) of the I/O pins help reduce external drivers and therefore, system cost.

The PIC16C84 product is supported by an assembler, an in-circuit emulator and a production quality programmer. These tools are supported on IBM PC and compatible machines.

FIGURE B - PIC16C84 BLOCK DIAGRAM

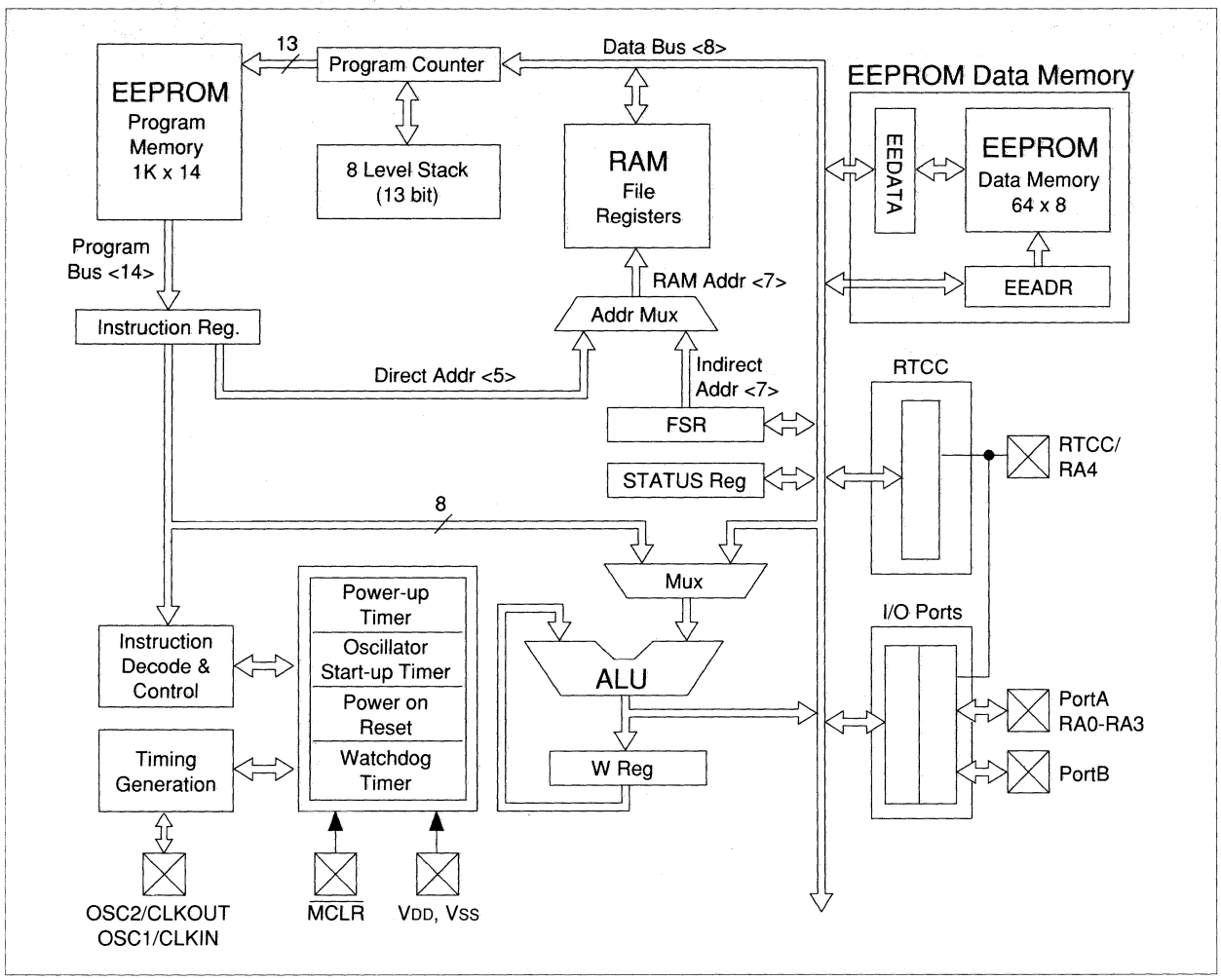


Table of Contents

Table of Figures

Table of Tables

1.0 GENERAL DESCRIPTION

The PIC16C84 is a low-cost, high-performance, CMOS, fully static, EEPROM-based 8-bit microcontroller. The EEPROM program memory is intended to be used for code development as well as One-Time-Programmable memory for full production. The program memory can not be updated during code execution. However, a special "in-system-programming" capability using only two pins to serially input and output data allows users to update program code of the PIC16C84 embedded in a system. The EEPROM data memory (64 bytes) is readable and writable during normal execution at full VDD range (2.0V - 6.0V).

The PIC16C84 employs an advanced RISC-like architecture. A reduced set of 35 instructions, single word instructions (14-bit wide), single cycle instructions except for 2-cycle program branches, instruction pipelining, large register set and separate instruction and data memory (Harvard architecture) schemes are some of the architectural innovation used to achieve very high performance. The PIC16C84 typically achieves a 2:1 code compression and a 4:1 speed improvement over other 8 bit microcontrollers in its class.

The PIC16C84 is equipped with special features to reduce external components and thus reduce cost, enhance system reliability and reduce power consumption. There are four oscillator options, of which the single pin RC oscillator provides a low cost solution and the LP oscillator minimizes power consumption. The SLEEP (power down) mode offers power saving. The user can wake up the chip from SLEEP through external interrupts and reset.

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

1.1 Compatibility with PIC16C5X

Those users familiar with the PIC16C5X family of microcontrollers will realize that this is an improved version of the PIC16C5X architecture. Please refer to Appendix A for a detailed list of modifications. Code written for PIC16C5X can be easily ported to the PIC16C84 (see Appendix B).

1.2 Applications

The PIC16C84 fits perfectly in applications ranging from high speed automotive and appliance control to low-power remote sensors, electronic locks and security devices. The PIC16C84 is also ideal for smart cards and RF tags. The EEPROM technology makes customization of application programs (transmitter codes, motor speeds, receiver frequencies, etc.) extremely fast and convenient. The small footprint packages for through hole or surface mounting make this microcontroller series perfect for all applications with space limitations.

Low cost, low power, high performance, ease of use, and I/O flexibility makes the PIC16C84 very versatile even in areas where no microcontroller use has been considered before (e.g. timer functions, replacement of "glue" logic in larger systems, co-processor applications).

Additionally, the In System Programmability of the PIC16C84 (using only two pins for data transfer) offers flexibility to customize a product after complete assembly and test.

This feature can be used to serialize a product, store calibration data available only after final test or to upgrade the firmware on finished goods.

2.0 PIC16C84 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 and tables in this section. When placing orders, please use the "PIC16C84 Product Identification System" on the back page of this data sheet to specify the correct part number.

2.1 Quick-Turnaround-Production (QTP) Devices

Microchip offers a QTP Programming Service for factory production orders. This service is made available for users who chose not to program a medium to high quantity of units and whose code patterns have stabilized. All EEPROM program memory locations and fuse options already programmed by the factory. Certain code and prototype verification procedures do apply before production shipments are available. Please contact your Microchip Technology sales office for more details.

2.2 Serialized Quick-Turnaround-Production (SQTP) Devices

Microchip offers the unique programming service where few locations in each device is programmed with a different serial number. The serial number may be random, pseudo-random or sequential.

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

3.0 ARCHITECTURAL OVERVIEW

The high performance of the PIC16C84 can be attributed to a number of architectural features commonly found in RISC microprocessors. To begin with, the PIC16C84 uses a Harvard architecture, in which, program and data are accessed from separate memories. This improves bandwidth over traditional Von-Neuman architecture where program and data are fetched from the same memory. Separating program and data memory further allows instructions to be sized differently than 8-bit wide data word. In PIC16C84, op-codes are 14-bit wide making it possible to have all single word instructions. A 14 bit wide program memory access bus fetches a 14 bit instruction in a single cycle. A two-stage pipeline overlaps fetch and execution of instructions. Consequently, all instructions (35) execute in a single cycle except for program branches.

The PIC16C84 address 1K x 14 program memory space, all on-chip. Program execution is internal only (microcontroller mode).

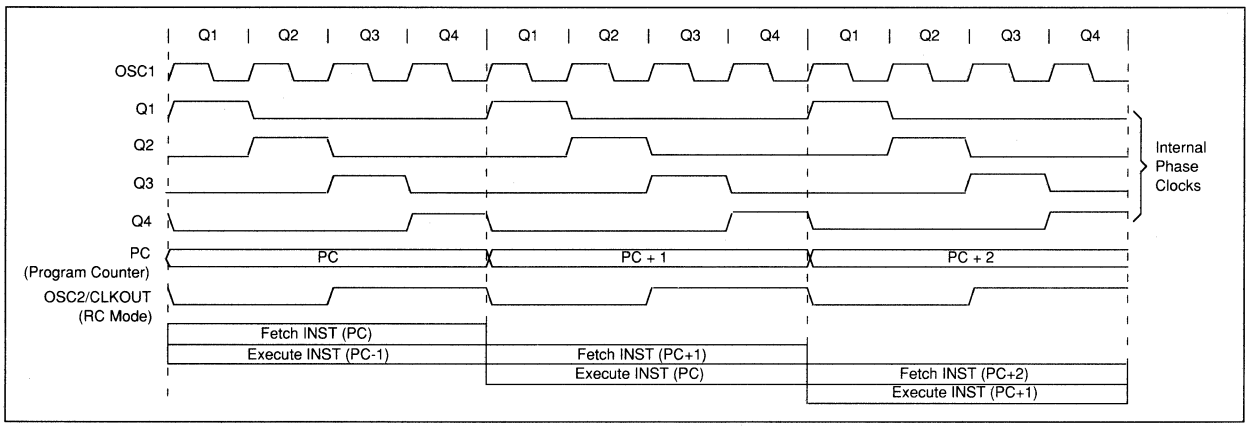
The PIC16C84 can directly or indirectly address its 48 register files or data memory. All special function registers including the program counter are mapped in the data memory. The instruction set is fairly orthogonal (symmetrical) which makes it possible to carry out any operation on any register using any addressing mode. This symmetrical nature and lack of 'special optimal situations' make programming with the PIC16C84 simple yet efficient. In addition, the learning curve is reduced significantly.

3.1 - PIC16C84 PINOUT DESCRIPTION

Pin name	Pin Type	Pin function	
		Normal operation	Serial In-System Programming (ISP) Mode
VDD	P	Power	Power
VSS	P	Ground	Ground
OSC1/CLKIN	I	Clock input/oscillator connection	-
OSC2/CLKOUT	I/O	Oscillator connection/CLKOUT output. It is CLKOUT in RC oscillator mode and oscillator connection in all other modes.	-
$\overline{MCLR}$/VPP	I/P	Master clear (external reset) input. Active low.	Master clear. Apply high voltage (VPP) to enter programming mode.
RA4/RTCC	I	Open-drain output/input pin. It is also the clock input to RTCC timer/counter: Schmitt trigger input buffer	-
RA0	I/O	Bidirectional I/O pin. TTL input levels	-
RA1	I/O	Bidirectional I/O pin. TTL input levels	-
RA2	I/O	Bidirectional I/O pin. TTL input levels	-
RA3	I/O	Bidirectional I/O pin. TTL input levels	-
RB0/INT	I/O	Bidirectional I/O pin/External interrupt input. TTL input levels	-
RB1	I/O	Bidirectional I/O pin. TTL input levels	-
RB2	I/O	Bidirectional I/O pin. TTL input levels	-
RB3	I/O	Bidirectional I/O pin. TTL input levels	-
RB4	I/O	Bidirectional I/O pin. TTL input levels	-
RB5	I/O	Bidirectional I/O pin. TTL input levels	-
RB6	I/O	Bidirectional I/O pin. TTL input levels	Clock input
RB7	I/O	Bidirectional I/O pin. TTL input levels	Data input/output

Legend: I = input, O = output, I/O = input/output, P = power. - = Not used.

FIGURE 3.2.1 - CLOCK/INSTRUCTION CYCLE



3.2 Clocking Scheme/Instruction Cycle

The clock input (from pin OSC1) is internally divided by four to generate four non overlapping quadrature clocks namely Q1, Q2, Q3 and Q4. Internally, the PC is incremented every Q1, an instruction is fetched from the program memory and latched into instruction register in Q4. It is decoded and executed during the following Q1 through Q4. The clocks and instruction execution flow is shown in Figure 3.2.1.

3.3 Instruction Flow/Pipelining

An "Instruction Cycle" consists of Q1, Q2, Q3 and Q4 cycles. The Instruction fetch and execute cycles are pipelined such that fetch takes one instruction cycle while the 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.

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

The fetched instruction is latched into the "Instruction Register (IR)" which is decoded and executed during Q2, Q3 and Q4. Data memory is read during Q2 (operand read) and written during Q4 (destination write).

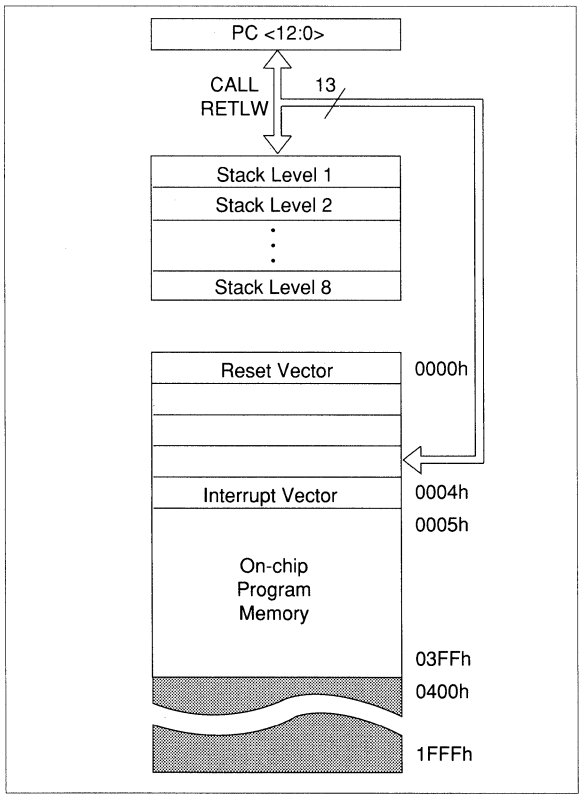
3.4 Program Memory Organization

The PIC16C84 has a 13-bit wide program counter (figure 3.4.1) capable of addressing an 8K x 14 program memory space. Only the first 1K x 14 (0000h - 03FFh) are physically implemented. Accessing a location above 3FFh will cause a wrap-around within the first 1K x 14 space. The reset vector is at 0000h and the interrupt vector is at 0004h.

The EEPROM program memory of the PIC16C84 is rated for limited Erase/write cycles. To program the program memory, the part must be put into a special mode by raising $\overline{MCLR}$ pin to high voltage (see section 11.5 for programming specification). Also, V_{DD} must be 4.5V to 5.5V during programming. The PIC16C84 is not suitable for applications where program memory is updated in the user application frequently.

The program memory can be programmed serially using two data/clock pins (see section 11) which makes in-system programming (ISP) possible. This allows the user to customize the system during final testing or upgrade a system in the field.

FIGURE 3.4.1 - PROGRAM MEMORY MAP AND STACK



3.5 Program Counter Module

The program counter (PC) is 13-bit wide. The low byte, PCL is a readable and writable register (02h). The high byte of the PC, PCH is not directly readable or writable. The high byte of the PC can be written through the PCLATH register (0Ah). When the PC is loaded with a new value during a CALL, GOTO or a write to PCL, the high bits of PC are loaded from PCLATH as shown in figure 3.5.1.

3.6 Stack

The PIC16C84 has an 8 deep x 13 bit wide hardware stack. The stack space is not part of either program or data space and the stack pointer is not readable or writable. The PC is pushed in the stack when a CALL instruction is executed or an interrupt is acknowledged. The stack is popped in the event of a RETURN, RETLW or RETFIE instruction execution. PCLATH (0Ah) is not affected by a PUSH or a POP operation.

3.7 Register File Organization

The register file is organized as 128 x 8. It is accessed either directly or indirectly through the file select register FSR. It is also referred to as the data memory. There are several register file page select bits in the STATUS register allowing up to four pages. However, data memory extends only up to 2Fh. The first 12 locations are used to map special function registers. Locations 0Ch - 2Fh are general purpose registers implemented as static RAM. Some special function registers are mapped in page 1. When in page 1, accessing locations 8Ch - AFh will access the RAM in page 0 (Figure 3.7.1).

3.7.1 REGISTER FILE ADDRESSING MODES

The register file can be addressed directly or indirectly. In both modes, up to 512 register locations can be addressed.

Direct addressing mode: An effective 9-bit direct address is obtained by concantenating 7 bits of direct address from the opcode and two bits (RP1, RP0) from the status register as shown in figure 3.7.1.1.

Indirect addressing mode: Indirect addressing is possible by using file address 00h. Any instruction using f0 as file register actually accesses data pointed to by the file select register, FSR (address 04h). Reading f0 itself indirectly will produce 00h. Writing to f0 indirectly results in a no-operation (although status bits may be affected). An effective 9-bit address is obtained by concantenating the 8-bit FSR register and the IRP bit from the status register as shown in figure 3.7.1.1.

Please note that some special function registers are mapped in page 1. It will be necessary to set RP0 bit to address then. Both RP1 and IRP bits are essentially not used.

For convenience, the general purpose registers are mapped both in page 0 and page 1.

FIGURE 3.5.1 - LOADING OF PC IN DIFFERENT SITUATIONS

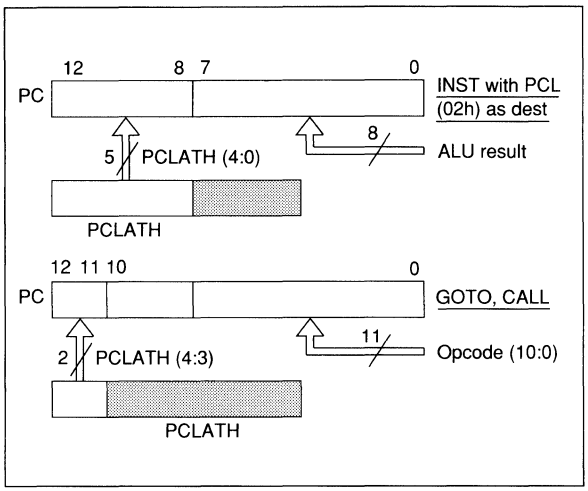


FIGURE 3.7.1 - REGISTER FILE MAP

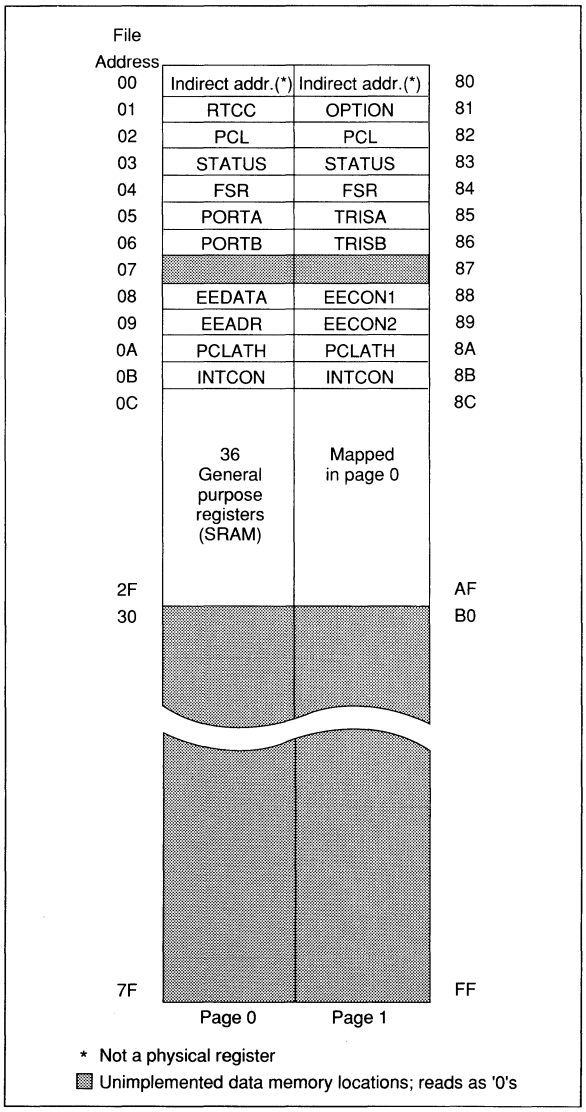
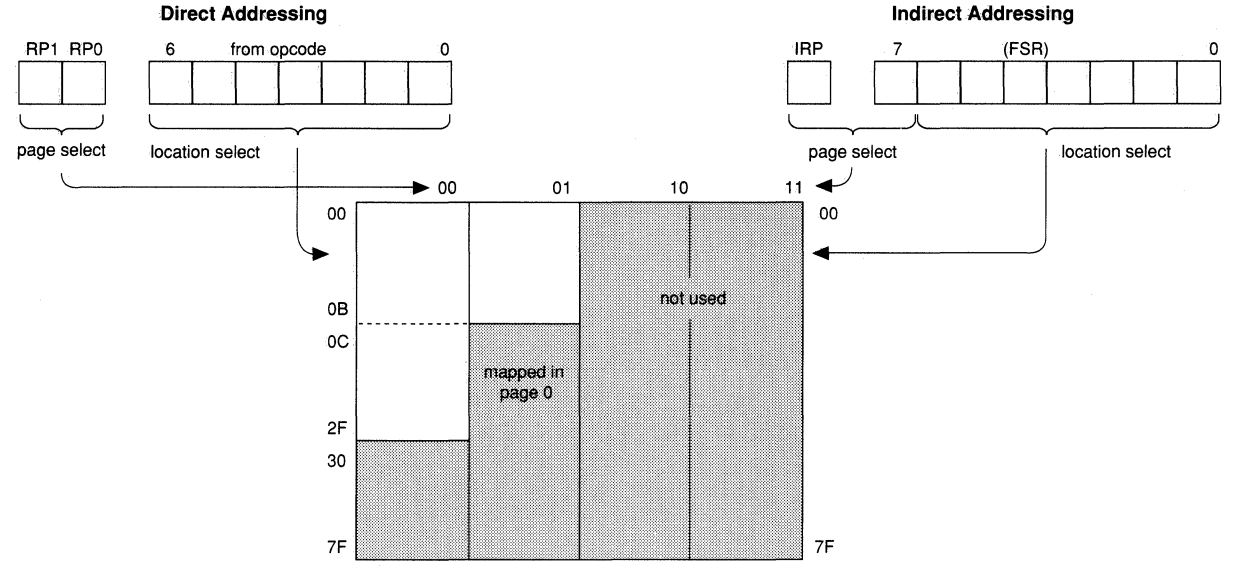


FIGURE 3.7.1.1 - DIRECT/INDIRECT ADDRESSING



Note: For memory map detail, see figure 3.7.1

FIGURE 3.7.2 - REGISTER FILE SUMMARY (PIC16C84)

Filename		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit1	Bit 0	Value on power on reset
Page 0:										
00	IND0	Uses contents of FSR to address data memory (not a physical register)								00000000
01	RTCC	8 Bit Real Time clock counter								XXXXXXXX
02	PCL	Low order 8 bits of PC								00000000
03	STATUS	IRP	RP1	RP0	TO	PD	Z	DC	C	00011XXX
04	FSR	Indirect data memory, address pointer 0								XXXXXXXX
05	PORTA	-	-	-	RA4/RT	RA3/AIN3/V_{REF}	RA2/AIN2	RA1/AIN1	RA0/AIN0	
06	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0/INT	
07		Not implemented								
08	EEDATA	EEPROM Data Register								XXXXXXXX
09	EEADR	EEPROM Address Register								XXXXXXXX
0A	PCLATH	Holding register for high byte of PC (Note 1)								---00000
0B	INTCON	GIE	ADIE	RTIE	INTE	RBIE	RTIF	INTF	RBIF	0000000X
Page 1:										
80	IND0	MAPPED IN PAGE 0								
81	OPTION	$\overline{RBPU}$	INTEDG	RTS	RTE	PSA	PS2	PS1	PS0	11111111
82	PCL	Mapped in page 0								
83	STATUS	Mapped in page 0								
84	FSR	Mapped in page 0								
85	TRISA	PORTA (f05) data direction register								---11111
86	TRISB	PORTB (f06) data direction register								11111111
87		Not implemented								
88	EECON1	-	-	-	EEIF	WRERR	WREN	WR	RD	0000X000
89	EECON2	Not a physical register								
8A	PCLATH	Mapped in page 0								
8B	INTCON	Mapped in page 0								

Notes: 1: The upper byte of the program counter is not directly accessible. f0A is a holding register for PC<15:8> whose contents are updated from or transfered to the upper byte of the program counter.

x = unknown
u = unchanged

3.8 Indirect Addressing Register (f00)

It is not a physical register. Addressing f0 will cause an indirect addressing. See 3.7.1.1 for details.

3.8.1 RTCC (f01)

8-bit real time clock counter. See section 6.4 for details.

3.8.2 PCL (f02)

Low order 8-bits of the PC. See section 3.5 for details.

3.9 STATUS Register (f03)

This register contains the arithmetic status of the ALU, the RESET status, and the page preselect bits for data memory.

The status register (f03) can be the destination for any instruction like any other register. However, the status bits are set following the write operation (Q4). Furthermore, TO and PD bits are not writable. Therefore, the result of an instruction with status register as destination may be different than intended. For example, CLRF f3 will clear all bits except for TO and PD and then set the Z bit and leave status register as 000UU100 (where U = unchanged).

It is recommended, therefore, that only BCF, BSF and MOVWF instructions are used to alter the status registers because these instructions do not affect any status bit.

For other instructions, affecting any status bits, see section "Instruction Set Summary" (Section 4.0).

3.9.1 CARRY/BORROW AND DIGIT CARRY/ BORROW BITS

The carry bit (C) is a carry out in addition operations (ADDWF, ADDLW) and a borrow out in subtract operations (SUBWF, SUBLW). The following examples explain operation of carry/borrow bit:

```
;SUBLW Example #1
;
MOVLW   0x01    ;wreg=1
SUBLW   0x02    ;wreg= 2-wreg = 2-1=1
                ;Carry=1: result is positive
;
;SUBLW Example #2
;
MOVLW   0x02    ;wreg=2
SUBLW   0x01    ;wreg=1-wreg=1-2=FFh
                ;Carry=0: Result is negative
;
;SUBWF Example #1
;
clrf    0x20    ;f(20h)=0
movlw   1       ;wreg=1
subwf   0x20    ;f(20h)=f(20h)-wreg=0-1=FFh
                ;Carry=0:Result is negative
;
;SUBWF Example #2
movlw   0xFF    ;
movwf   0x20    ;f(20h)=FFh
clrw            ;wreg=0
subwf   0x20    ;f(20h)=f(20h)-wreg=FFh-0=FFh
                ;Carry=1: Result is positive
;
```

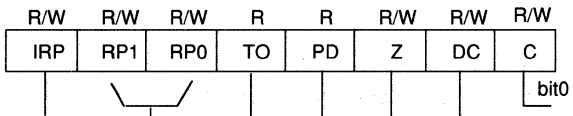
The digit carry operates in the same way as the carry bit, i.e.: it is a borrow in subtract operations.

3.9.2 TIME OUT AND POWER DOWN STATUS BITS (TO, PD)

The "TO" and "PD" bits in the status register f03 can be tested to determine if a RESET condition has been caused by a watchdog timer time-out, a power-up condition, or a wake-up from SLEEP by the watchdog timer or MCLR pin.

These status bits are only affected by events listed in Table 3.9.2.1.

FIGURE 3.9.1 - STATUS REGISTER



ADDRESS: 03h
RESET CONDITION: 000??XXX
TO, PD are set or reset as shown in Table 3.9.2.1

CARRY/$\overline{\text{BORROW}}$ BIT:
For ADDWF, SUBWF , ADDLW and SUBLW instructions, this bit is set if there is a carry out from the most significant bit of the resultant. Note that 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.

DIGIT CARRY/$\overline{\text{BORROW}}$ BIT:
For ADDWF, SUBWF , ADDLW and SUBLW instructions, this bit is set if there is a carry out from the 4th low order bit of the resultant.

ZERO BIT:
Set if the result of an arithmetic or logic operation is zero.
Reset otherwise. Additionally, MOVF instruction will affect the Z bit.

POWER DOWN BIT:
Set to "1" during power up or by a CLRWDT command. This bit is cleared to "0" by a SLEEP instruction.

TIME-OUT BIT:
Set to "1" during power up and by the CLRWDT and SLEEP command. This bit is cleared to "0" by a watchdog timer time out.

REGISTER PAGE SELECT BITS FOR DIRECT ADDRESSING:

RP1,0:
00 : page 0 (00h - 7Fh)
01 : page 1 (80h - FFh)
10 : page 2 (100h - 17Fh)
11 : page 3 (180h - 1FFh)

Each page is 128 bytes.
Only RP0 is useful in PIC16C84. Bit RP1 can be used as a general purpose read/write bit. However, this may affect upward compatibility with future products.

REGISTER PAGE SELECT BITS FOR INDIRECT ADDRESSING:
IRP0: 0 : page 0,1 (00h - FFh)
1 : page 2,3 (100h - 1FFh)
This bit is effectively not used in the PIC16C84.
It may be used as a general purpose read/write bit. However, this may affect upward compatibility with future product.

TABLE 3.9.2.1 - EVENTS AFFECTING PD/ TO STATUS BITS

Event	TO	PD	Remarks
Power-up	1	1	
WDT Timeout	0	U	No effect on PD
SLEEP instruction	1	0	
CLRWDT instruction	1	1	

U: unchanged
Note: A WDT timeout will occur regardless of the status of the TO bit. A SLEEP instruction will be executed, regardless of the status of the PD bit. Table 3.9.2.2 reflects the status of PD and TO after the corresponding event.

TABLE 3.9.2.2 - PD/TO STATUS AFTER RESET

TO	PD	RESET was caused by
0	0	WDT wake-up from SLEEP
0	1	WDT time-out (not during SLEEP)
U	0	$\overline{\text{MCLR}}$ wake-up from SLEEP
1	1	Power-up
U	U	$\overline{\text{MCLR}}$ reset during normal operation

U: unchanged
Note: The PD and TO bit maintain their status until an event of Table 3.9.2.1 occurs. A low-pulse on the $\overline{\text{MCLR}}$ input does not change the PD and TO status bits.

3.10 Arithmetic and Logic Unit (ALU)

The ALU is 8 bit wide and capable of addition, subtraction, shift and logical operations. Unless otherwise mentioned, arithmetic operations are 2's complement in nature. In two-operand instructions, typically one operand is the working register (W register) or the accumulator. 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.

3.11 W Register

The W register is an 8-bit working register (or accumulator) used for ALU operations. It is not in the data memory.

3.12 Interrupts

The PIC16C84 has four sources of interrupt: external interrupt from RB0/INT pin, RTCC timer/counter overflow interrupt, end of data EEPROM write, and interrupt on change on RB<7:4> pins. All the interrupts vector to a single interrupt vector at 0004h. The interrupt control register INTCON records individual interrupt requests in flag bits. It also has individual and global mask bits. The only exception is the data EEPROM write completion interrupt flag (EEIF) which resides in EECON1 register.

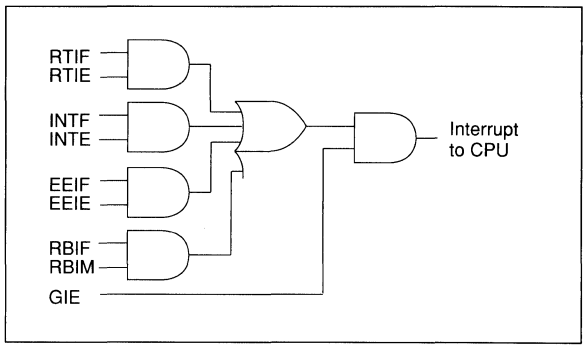
A global interrupt enable bit, GIE (INTCON<7>) enables (if = 1) all un-masked interrupts or disables (if = 0) all interrupts. Individual interrupts can be disabled through their corresponding mask bit in INTCON register. GIE is cleared on reset.

When an interrupt is responded to, the GIE is cleared to disable any further interrupt, the return address is pushed into the stack and the PC is loaded with 0004h. The interrupt latency is approximately five cycles for external events such as INT pin or PortB interrupts (figure 3.12.1.1). It is one cycle less for internal events such as RTCC timer overflow (figure 6.4.4). The latency is the same for one or 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 recursive interrupts. Individual interrupt flag bits are set regardless of the status of their corresponding mask bit or the GIE bit.

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

FIGURE 3.12.1 - INTERRUPT LOGIC



3.12.1 INT INTERRUPT

External interrupt on RB1/INT pin is edge triggered: either rising (if INTEDG = 1, OPTION<6>) or falling (if INTEDG = 0). When a valid edge appears on INT pin, INTF bit is set (INTCON <1>). This interrupt can be disabled by setting INTE control bit INTCON<4> to '0'. The INTF bit (INTCON<1>) must be cleared in software in the interrupt service routine before re-enabling this interrupt. The INT interrupt can wake up the processor from SLEEP if INTE bit was set to '1' prior to going into SLEEP. The status of the GIE bit decides whether or not the processor branches to the interrupt vector following wake-up. See section 5.5 for details on SLEEP and figure 5.5.1 for timing of wake-up from SLEEP through INT interrupt.

3.12.2 RTCC INTERRUPT

An overflow (FFh → 00h) in the RTCC will set the RTIF (INTCON<2>) bit. The interrupt can be enabled/disabled by setting/resetting RTIE (INTCON<5>) bit. See section 6.4 for details.

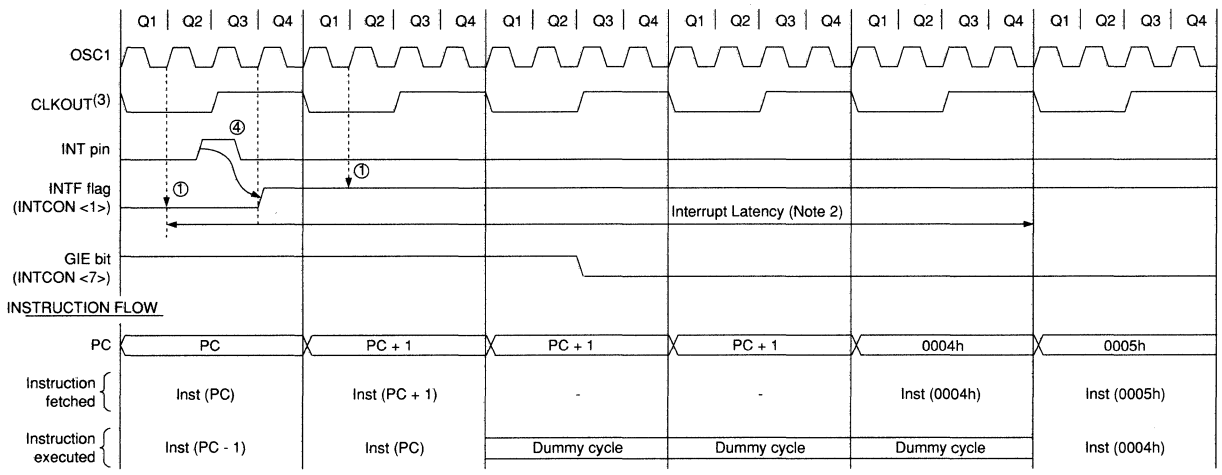
3.12.3 PORT RB INTERRUPT

An input change on port B <7:4> will set the RBIF (INTCON<0>) bit. The interrupt can be enabled/disabled by setting/resetting RBIE (INTCON<4>) bit.

3.12.4 EEPROM WRITE INTERRUPT

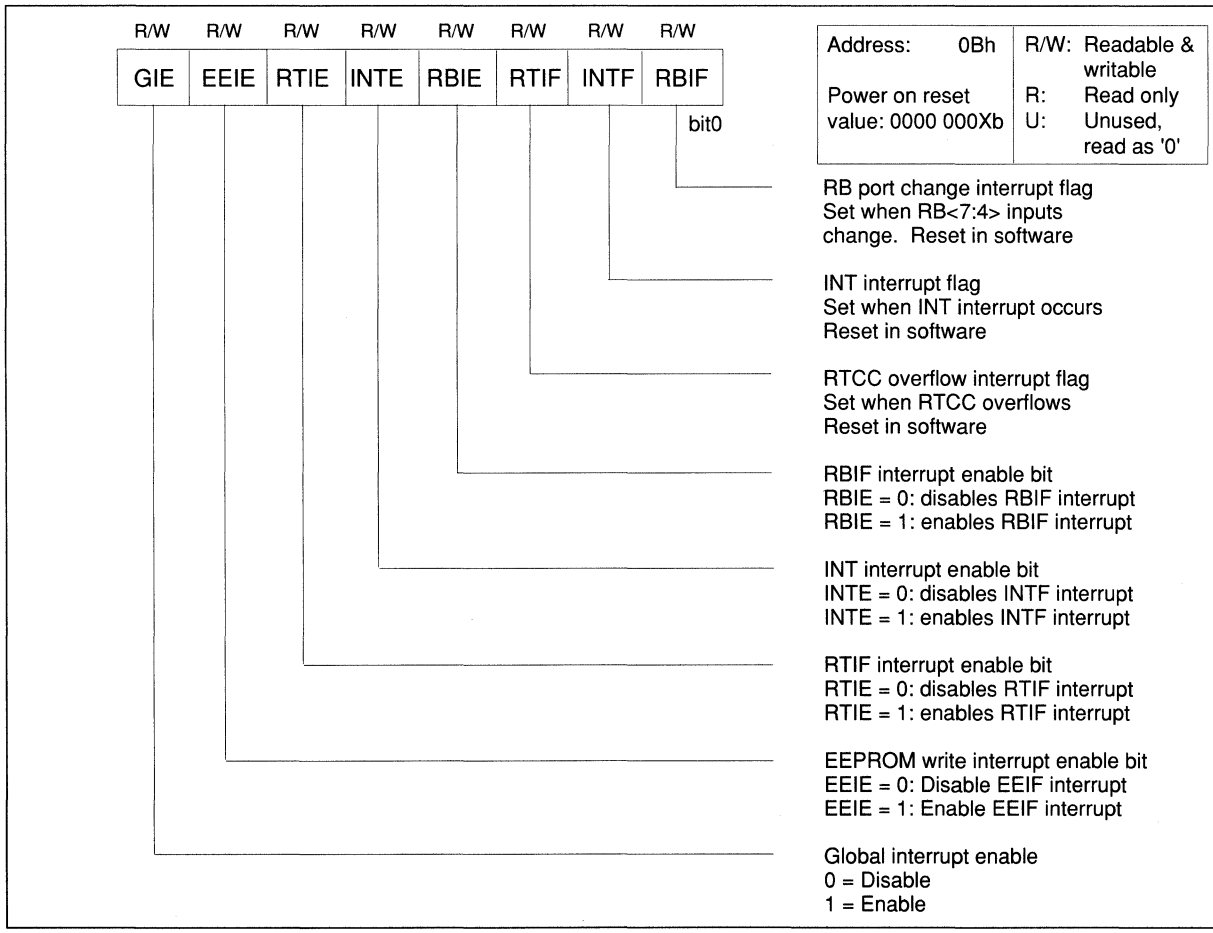
The EEPROM write interrupt flag, EEIF (EECON1<4>) when a data EEPROM write is complete. The interrupt can be masked by setting the EEIE bit (INTCON<6>) to '0'. See section 6.1 for details on EEPROM write interrupt.

FIGURE 3.12.1.1 - INT PIN INTERRUPT TIMING



Notes:
1. INTF flag is sampled here (every Q1)
2. Interrupt latency = 4.75 Tcy 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 spec of INT pulse, refer to AC specs.

FIGURE 3.12.2 - INTCON REGISTER



4.0 INSTRUCTION SET SUMMARY

Each PIC 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 PIC instruction set summary in Table 4.1 lists byte-oriented, bit-oriented, and literal and control operations.

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 utilized 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.

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. One instruction cycle consists of four oscillator periods. Thus, for an oscillator frequency of 4 MHz, the normal instruction execution time is 1 µsec. If a conditional test is true or the program counter is changed as a result of an instruction, the instruction execution time is 2 µsec.

Notes to Table 4.1

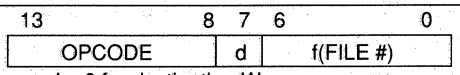
Note 1: TRIS and OPTION instructions are included in the instruction set for upward compatibility with the PIC16C5X products. Their use in new code is not recommended. Since TRIS and OPTION are made addressible registers, the user may simply write to them. These instructions may not be supported in future PIC16CXX products.

Note 2: When an I/O register is modified as a function of itself (e.g. MOVF 6,1), 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 output and is driven low by an external device, the data will be written back with a '0'.

Note 3: If this instruction is executed on file register f1 (and, where applicable, d=1), the prescaler will be cleared if assigned to the RTCC.

4.1 INSTRUCTION SET

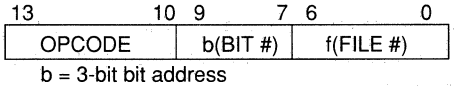
BYTE-ORIENTED FILE REGISTER OPERATIONS



d = 0 for destination W
d = 1 for destination f
f = 7-bit file register address

Instruction-Binary	(Hex)	Name	Mnemonic, Operands		Operation	Status affected	Notes
00 0111 dfff ffff	07ff	Add W and f	**ADDWF**	**f, d**	W + f → d	C, DC, Z	2,3
00 0101 dfff ffff	05ff	AND W and f	**ANDWF**	**f, d**	W & f → d	Z	2,3
00 0001 1fff ffff	018f	Clear f	**CLRF**	**f**	0 → f	Z	3
00 0001 0XXX XXXX	0100	Clear W	**CLRW**	**-**	0 → W	Z	
00 1001 dfff ffff	09ff	Complement f	**COMF**	**f, d**	$\bar{f}$ → d	Z	2,3
00 0011 dfff ffff	03ff	Decrement f	**DECF**	**f, d**	f -1 → d	Z	2,3
00 1011 dfff ffff	0Bff	Decrement f,Skip if Zero	**DECFSZ**	**f, d**	f - 1 → d, skip if zero	None	2,3
00 1010 dfff ffff	0Aff	Increment f	**INCF**	**f, d**	f + 1 → d	Z	2,3
00 1111 dfff ffff	0Fff	Increment f,Skip if zero	**INCFSZ**	**f, d**	f + 1 → d, skip if zero	None	2,3
00 0100 dfff ffff	04ff	Inclusive OR W and f	**IORWF**	**f, d**	W v f → d	Z	2,3
00 1000 dfff ffff	08ff	Move f	**MOVF**	**f, d**	f → d	Z	2,3
00 0000 1fff ffff	008f	Move W to f	**MOVWF**	**f**	W → f	None	3
00 0000 0XX0 0000	0000	No Operation	**NOP**	**-**	-	None	
00 1101 dfff ffff	0Dff	Rotate left f	**RLF**	**f, d**	f<n>→d<n+1>,C→d<0>,f<7> →C	C	2,3
00 1100 dfff ffff	0Cff	Rotate right f	**RRF**	**f, d**	f<n>→d<n-1>, C→d<7>, f<0>→C	C	2,3
00 0010 dfff ffff	02ff	Subtract W from f	**SUBWF**	**f, d**	f - W → d [f + $\bar{W}$ + 1 → d]	C, DC, Z	2,3
00 1110 dfff ffff	0Eff	Swap halves f	**SWAPF**	**f, d**	f<0-3> ↔ f<4-7> → d	None	2,3
00 0110 dfff ffff	06ff	Exclusive OR W and f	**XORWF**	**f, d**	W ⊕ f → d	Z	2,3

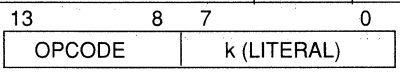
BIT-ORIENTED FILE REGISTER OPERATIONS



b = 3-bit bit address
f = 7-bit file register address

01 00bb bfff ffff	1bff	Bit Clear f	**BCF**	**f, b**	0 → f(b)	None	2,3
01 01bb bfff ffff	1bff	Bit Set f	**BSF**	**f, b**	1 → f(b)	None	2,3
01 10bb bfff ffff	1bff	Bit Test f,Skip if Clear	**BTFSC**	**f, b**	Test bit (b) in file (f): Skip if clear	None	
01 11bb bfff ffff	1bff	Bit Test f, Skip if Set	**BTFSS**	**f, b**	Test bit (b) in file (f): Skip if set	None	

LITERAL AND CONTROL OPERATIONS



k = 8-bit immediate value.

11 111X kkkk kkkk	3Ekk	Add literal to W	**ADDLW**	**k**	k + W → W	C,DC,Z	
11 1001 kkkk kkkk	39kk	AND Literal and W	**ANDLW**	**k**	k & W→ W	Z	
10 0kkk kkkk kkkk	2kkk	Call subroutine	**CALL**	**k**	PC + 1 → TOS, k → PC <10:0>, PCLATH <4:3> → PC <12:11>;	None	
00 0000 0110 0100	0064	Clear Watchdog timer	**CLRWDT**	**-**	0→WDT(and prescaler,if assigned)	TO, PD	
10 1kkk kkkk kkkk	2kkk	Go To address	**GOTO**	**k**	k → PC <10:0>, PCLATH <4:3> → PC <12:11>;	None	
11 1000 kkkk kkkk	38kk	Incl. OR Literal and W	**IORLW**	**k**	k v W → W	Z	
11 00XX kkkk kkkk	30kk	Move Literal to W	**MOVLW**	**k**	k → W	None	
00 0000 0000 1001	0009	Return from interrupt	**RETFIE**	**-**	TOS → PC, '1' → GIE	None	
11 01XX kkkk kkkk	34kk	Return, place literal in W	**RETLW**	**k**	k → W, TOS→ PC	None	
00 0000 0000 1000	0008	Return from subroutine	**RETURN**	**-**	TOS → PC	None	
00 0000 0110 0011	0063	Go into standby mode	**SLEEP**	**-**	0 → WDT, stop oscillator	TO, PD	
11 110X kkkk kkkk	3Ckk	Subtract W from literal	**SUBLW**	**k**	k - W → W	C,DC,Z	
11 1010 kkkk kkkk	3Akk	Excl. OR Literal and W	**XORLW**	**k**	k ⊕ W → W	Z	

00 0000 0110 0010	0062	Load OPTION register	**OPTION**	**-**	W → OPTION register	None	1
00 0000 0110 0fff	006f	Tristate port f	**TRIS**	**f**	W→ I/O control register f	None	1

X = 0 or 1. The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all software tools.
Notes: See previous page

4.2 INSTRUCTION DESCRIPTION

ADDLW Add Literal to W

Syntax: ADDLW k

Encoding:	11	111X	kkkk	kkkk

Words: 1

Cycles: 1

Operation: (W + k) → W

Status bits: C, DC, Z

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

ADDWF ADD W to f

Syntax: ADDWF f,d

Encoding:	00	0111	dfff	ffff

Words: 1

Cycles: 1

Operation: (W + f) → d

Status bits: C, DC, Z

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".

ANDLW AND Literal and W

Syntax: ANDLW k

Encoding:	11	1001	kkkk	kkkk

Words: 1

Cycles: 1

Operation: (W .AND. k) → W

Status bits: Z

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

ANDWF AND W with f

Syntax: ANDWF f,d

Encoding:	00	0101	dfff	ffff

Words: 1

Cycles: 1

Operation: (W .AND. f) → d

Status bits: Z

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".

BCF Bit Clear f

Syntax: BCF f,b

Encoding:	01	00bb	bfff	ffff

Words: 1

Cycles: 1

Operation: 0 → f(b)

Status bits: None

Description: Bit "b" in register "f" is reset to 0.

BSF Bit Set f

Syntax: BSF f,b

Encoding:	01	01bb	bfff	ffff

Words: 1

Cycles: 1

Operation: 1 → f(b)

Status bits: None

Description: Bit "b" in register "f" is set to 1.

BTFSC Bit Test, skip if Clear

Syntax: BTFSC f,b

Encoding:	01	10bb	bfff	ffff

Words: 1

Cycles: 1(2)

Operation: skip if f(b) = 0

Status bits: None

Description: If bit "b" in register "f" is "0" then the next instruction is skipped.

If bit "b" in register "f" is "0", the next instruction, fetched during the current instruction execution, is discarded and a NOP is executed instead making this a 2 cycle instruction.

BTFSS Bit Test, skip if Set

Syntax: BTFSS f,b

Encoding:	01	11bb	bfff	ffff

Words: 1

Cycles: 1 (2)

Operation: skip if f(b) = 1

Status bits: None

Description: If bit "b" in register "f" is "1" then the next instruction is skipped.

If bit "b" in register "f" is "0", the next instruction, fetched during the current instruction execution, is discarded and a NOP is executed instead making this a 2 cycle instruction.

CALL Subroutine Call

Syntax: CALL k

Encoding:	10	0kkk	kkkk	kkkk

Words: 1

Cycles: 2

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

Status bits: None

Description: Subroutine call. First, return address (PC + 1) is pushed into the stack. The eleven bit value is loaded into PC bits <10:0>. The upper bits of the PC are loaded from PCLATH (f03). CALL is a two cycle instruction.

CLRF Clear f

Syntax: CLRF f

Encoding:	00	0001	1fff	ffff

Words: 1

Cycles: 1

Operation: 00h → f

Status bits: Z

Description: The contents of register "f" are set to 0.

CLRW Clear W Register

Syntax: CLRW

Encoding:	00	0001	0XXX	XXXX

Words: 1

Cycles: 1

Operation: 00h →W

Status bits: Z

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

CLRWDT Clear Watchdog Timer

Syntax: CLRWDT

Encoding:	00	0000	0110	0100

Words: 1

Cycles: 1

Operation: 00h →WDT, 0 → WDT prescaler,

Status bits: 1 → TO, 1 → PD

Description: CLRWDT instruction resets the watchdog timer.It also resets the prescaler of the WDT. Status bits TO and PD are set.

COMF Complement f

Syntax: COMF f,d

Encoding:	00	1001	dfff	ffff

Words: 1

Cycles: 1

Operation: $\bar{f}$ → d

Status bits: Z

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".

DECF Decrement f

Syntax: DECF f,d

Encoding:	00	0011	dfff	ffff

Words: 1

Cycles: 1

Operation: (f-1) → d

Status bits: Z

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".

DECFSZ Decrement f, skip if 0

Syntax: DECFSZ f,d

Encoding:	00	1011	dfff	ffff

Words: 1

Cycles: 1 (2)

Operation: (f - 1) → d; skip if result = 0

Status bits: None

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.

GOTO Unconditional Branch

Syntax: GOTO k

Encoding:	10	1kkk	kkkk	kkkk

Words: 1

Cycles: 2

Operation: k → PC<10:0>, PCLATH<4:3> → PC<12:11>

Status bits: None

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.

INCF Increment f

Syntax: INCF f,d

Encoding:	00	1010	dfff	ffff

Words: 1

Cycles: 1

Operation: (f + 1) → d

Status bits: Z

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".

INCFSZ Increment f, skip if 0

Syntax: INCFSZ f,d

Encoding:	00	1111	dfff	ffff

Words: 1

Cycles: 1 (2)

Operation: (f + 1) → d, skip if result = 0

Status bits: None

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.

IORLW Inclusive OR Literal with W

Syntax: IORLW k

Encoding:	11	1000	kkkk	kkkk

Words: 1

Cycles: 1

Operation: (W .OR. k) → W

Status bits: Z

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

IORWF Inclusive OR W with f

Syntax: IORWF f,d

Encoding:	00	0100	dfff	ffff

Words: 1

Cycles: 1

Operation: (W .OR. f) → d

Status bits: Z

Description: Inclusive OR 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".

MOVLW Move Literal to W

Syntax: MOVLW k

Encoding:	11	00XX	kkkk	kkkk

Words: 1

Cycles: 1

Operation: k → W

Status bits: None

Description: The eight bit literal "k" is loaded into W register.

MOVF Move f

Syntax: MOVF f,d

Encoding:	00	1000	dfff	ffff

Words: 1

Cycles: 1

Operation: f → d

Status bits: Z

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. It is useful, however, to test a file register since status flag Z is affected.

MOVWF Move W to f

Syntax: MOVWF f

Encoding:	00	0000	1fff	ffff

Words: 1

Cycles: 1

Operation: W → f

Status bits: None

Description: Move data from W register to register "f".

NOP No Operation

Syntax: NOP

Encoding:

00	0000	0XX0	0000

Words: 1

Cycles: 1

Operation: No operation

Status bits: None

Description: No operation

OPTION Load Option Register

Syntax: OPTION

Encoding:

00	0000	0110	0010

Words: 1

Cycles: 1

Operation: W → OPTION;

Status bits: None

Description: The contents of the W register is loaded in the OPTION register. Refer to Fig. 6.5.1 for OPTION register settings.

This instruction is supported for code compatibility with PIC16C5X products. Since OPTION is a readable/writable register, the user can directly address it.

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

RETFIE Return from Interrupt

Syntax: RETFIE

Encoding:

00	0000	0000	1001

Words: 1

Cycles: 2

Operation: TOS → PC, 1 → GIE;

Status bits: GLINTD

Description: Return from Interrupt. Stack is popped and Top of the Stack (TOS) is loaded in PC. Interrupts are enabled by setting the GIE bit. GIE is the global interrupt enable bit (bit 7, register INTCON). This is a two cycle instruction.

RETLW Return Literal to W

Syntax: RETLW k

Encoding:

11	01XX	kkkk	kkkk

Words: 1

Cycles: 2

Operation: k → W; TOS → PC;

Status bits: None

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.

RETURN Return from Subroutine

Syntax: RETURN

Encoding:

00	0000	0000	1000

Words: 1

Cycles: 2

Operation: TOS → PC;

Description: Return from subroutine. The stack is popped and the top of the stack (TOS) is loaded into the program counter. This is a two cycle instruction.

RLF Rotate Left f through Carry

Syntax: RLF f,d

Encoding:

00	1101	dfff	ffff

Words: 1

Cycles: 1

Operation: f<n> → d<n+1>, f<7> → C, C → d<0>;

Status bits: C

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".

RRF Rotate Right f through Carry

Syntax: RRF f,d

Encoding:

00	1100	dfff	ffff

Words: 1

Cycles: 1

Operation: f<n> → d<n-1>, f<0> → C, C → d<7>;

Status bits: C

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".

SLEEP

Syntax: SLEEP

Encoding:

00	0000	0110	0011

Words: 1

Cycles: 1

Operation: 0 → PD, 1 → TO;
00h → WDT, 0 → WDT prescaler;

Status bits: TO, PD

Description: The power down status bit (PD) is cleared. Time-out status bit (TO) is set. Watchdog Timer and its prescaler are cleared.

The processor is put into SLEEP mode with the oscillator stopped. See section on SLEEP mode for more details.

SUBLW Subtract W from Literal

Syntax: SUBLW k

Encoding:	11	110X	kkkk	kkkk

Words: 1

Cycles: 1

Operation: (k - W) → W

Status bits: C, DC, Z

Description: The W register is subtracted (2's complement method) from the eight bit literal "k". The result is placed in the W register.

```
Example
;SUBLW Example #1
;
MOVLW   0x01    ;wreg=1
SUBLW   0x02    ;wreg= 2-wreg = 2-1=1
                ;Carry=1: result is positive
;
;SUBLW Example #2
;
MOVLW   0x02    ;wreg=2
SUBLW   0x01    ;wreg=1-wreg=1-2=FFh
                ;Carry=0: Result is negative
```

SUBWF Subtract W from f

Syntax: SUBWF f,d

Encoding:	00	0010	dfff	ffff

Words: 1

Cycles: 1

Operation: (f-W) → d

Status bits: C, DC, Z

Description: Subtract (2's complement method) the 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".

```
Example:
;SUBWF Example #1
;
clrf    0x20    ;f(20h)=0
movlw   1       ;wreg=1
subwf   0x20    ;f(20h)=f(20h)-wreg=0-1=FFh
                ;Carry=0:Result is negative
;
;SUBWF Example #2
movlw   0xFF    ;
movwf   0x20    ;f(20h)=FFh
clrw            ;wreg=0
subwf   0x20    ;f(20h)=f(20h)-wreg=FFh-0=FFh
                ;Carry=1: Result is positive
;
```

SWAPF Swap f

Syntax: SWAPF f,d

Encoding:	00	1110	dfff	ffff

Words: 1

Cycles: 1

Operation: f<0:3> → d<4:7>, f<4:7> → d<0:3>;

Status bits: None

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".

TRIS Load TRIS Register

(Use of this instruction is not recommended for new (PIC16CXX) designs).

Syntax: TRIS f

Encoding:	00	0000	0110	0fff

Words: 1

Cycles: 1

Operation: W → TRIS register f;

Status bits: None

Description: The I/O Control Register (or data direction register of the I/O port) is loaded with the contents of the W register.

The TRIS instruction configures an I/O port to either output or input (high-impedance). The valid values for "f" are 5 & 6 for PIC16C84.

This instruction is supported for code compatibility with the PIC16C5X products. Since TRIS registers are readable and writable, the user can directly address them.

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

A '1' in the TRIS register configures the corresponding port pin as an input. A '0' in the TRIS register configures the corresponding port pin as an output.

An example to show I/O Port B (F6) configured such that the 4 pins corresponding to the LSBs of Port B are inputs (hi-impedance) and the other 4 pins are outputs, is shown below.

```
Example:
MOVLW   0x0F
TRIS    6
```

XORLW Exclusive OR literal with W

Syntax: XORLW k

Encoding:

11	1010	kkkk	kkkk

Words: 1

Cycles: 1

Operation: (W .XOR. k) → W

Status bits: Z

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

XORWF Exclusive OR W with f

Syntax: XORWF f,d

Encoding:

00	0110	dfff	ffff

Words: 1

Cycles: 1

Operation: (W .XOR. f) → d

Status bits: Z

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".

5.0 SPECIAL FEATURES OF THE CPU

What sets apart a microcontroller from other processors are special circuits to deal with the needs of real time applications. The PIC16C84 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.

The on-chip watchdog timer can only be shut off through an EEPROM fuse. 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, designed to keep the part in reset while the power supply stabilizes. With these two timers on chip, most applications need no external reset circuitry.

The SLEEP mode is designed to offer a very low current power-down mode. The user can wake up from SLEEP through external reset, watchdog timer time-out or interrupt. Several oscillator options are also made available to allow the part to fit the application. The RC oscillator option saves system cost while the LP crystal option saves power. A set of EEPROM configuration bits (fuses) are used to select various options (section 5.6).

5.1 RESET

The PIC16C84 differentiates between various kinds of reset:

a) Power on reset (POR)

b) $\overline{\text{MCLR}}$ Reset during normal operation

c) $\overline{\text{MCLR}}$ reset during SLEEP

d) WDT time-out reset during normal operation

e) WDT time-out reset during SLEEP

Some registers are not reset; they are unknown on POR and unchanged in any other reset. Most other registers are reset to "reset state" on power-on reset (POR), on $\overline{\text{MCLR}}$ or WDT reset during normal operation and on $\overline{\text{MCLR}}$ reset during SLEEP. They are not affected by a WDT reset during SLEEP, since this reset is viewed as the resumption of normal operation. There are a few exceptions to this. The PC is always reset to all 0's (0000h). Finally, TO and PD bits are set or cleared differently in different reset situations as indicated in section 3.9.1. These bits are used in software to determine the nature of reset. See Table 5.1.1 for a full description of reset states of all registers.

5.2 Power-on-reset (POR), Power-up-timer (PWRT) and Oscillator Start-up timer (OST)

Power-on-reset (POR): A power-on-reset pulse is generated on-chip when V_{DD} rise is detected (in the range of 1.2V to 1.8V). To take advantage of the POR, just tie MCLR pin directly (or through a resistor) to V_{DD}. This will eliminate external RC components usually needed to create power-on-reset.

FIGURE 5.0.1 - SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT

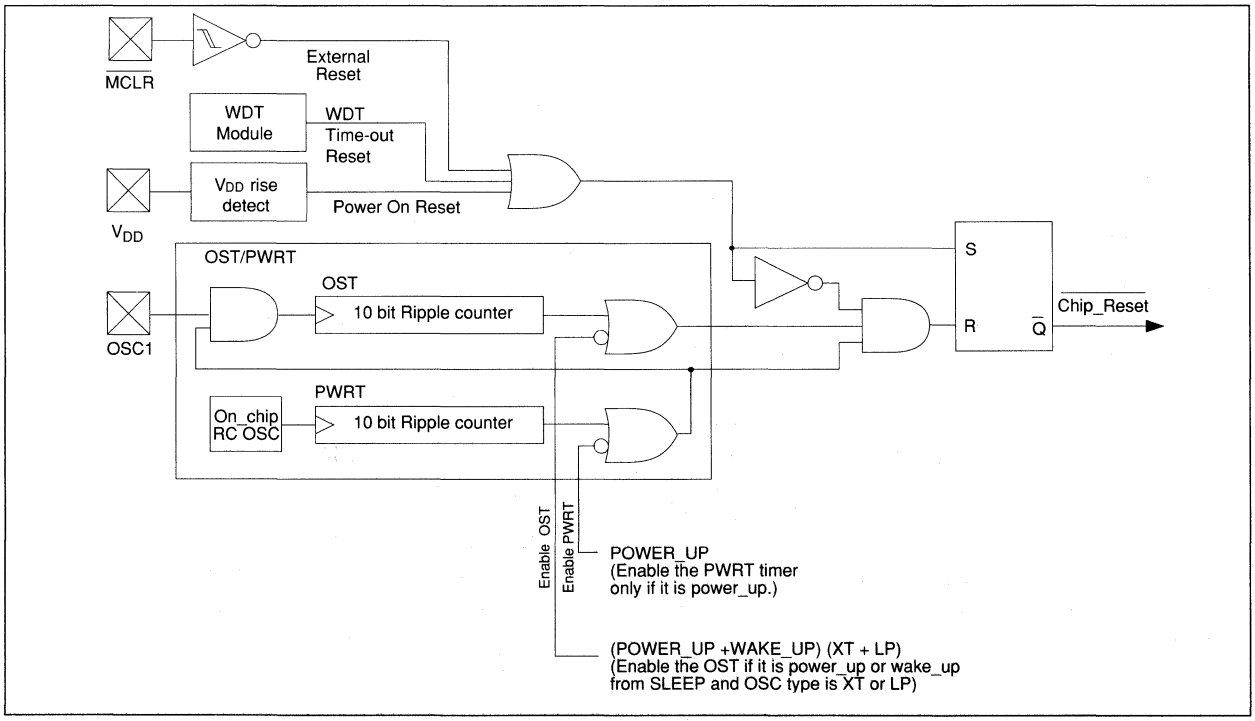


TABLE 5.1.1 - RESET CONDITIONS FOR REGISTERS

Register	Address	Power-on reset (POR)	WDT time-out reset during normal operation	WDT time-out reset during SLEEP	$\overline{MCLR}$ reset during normal operation	$\overline{MCLR}$ reset during SLEEP	Wake-up through interrupt
W	-	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
INDIR	00h	-	-	-	-	-	-
RTCC	01h	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
PC	02h	0000h	0000h	PC + 1	0000h	0000h	PC + 1
STATUS	03h	0001 1xxx	0000 1uuu	uuu0 0uuu	000u uuuu	0001 0uuu	uuu1 0uuu
FSR	04h	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
PORT A	05h	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
PORT B	06h	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
TRIS A	85h	---1 1111	---1 1111	---u uuuu	---1 1111	---1 1111	---u uuuu
TRIS B	86h	1111 1111	1111 1111	uuuu uuuu	1111 1111	1111 1111	uuuu uuuu
OPTION	81h	1111 1111	1111 1111	uuuu uuuu	1111 1111	1111 1111	uuuu uuuu
EEDATA	08h	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
EEADR	09h	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
EECON1	88h	---0 0000	---0 ?000†	---u uuuu	---0 ?000†	---0 ?000†	---u uuuu
EECON2	89h	-	-	-	-	-	-
PCLATH	0Ah	---0 0000	---0 0000	---u uuuu	---0 0000	---0 0000	---u uuuu
INTCON	0Bh	0000 000x	0000 000u	uuuu uuuu	0000 000u	0000 0000	uuuu uuuu*

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

* In the event of wake-up through interrupt, one or more of the interrupt flags will be set. Other bits in INTCON will remain unchanged.

† WRERR (bit3) will be set if reset occurred during EEPROM write.

The POR circuit does not produce internal reset when VDD declines (or goes through a brown-out).

Power-up Timer (PWRT): The power-up timer provides a fixed 72ms 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 PWRT is active. The PWRT delay allows the VDD to rise to an acceptable level. A configuration fuse, PWRTE can enable (if = 1) or disable (if = 0 or programmed) the power-up timer (section 5.6).

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

Oscillator Start-up Timer (OST): The oscillator start-up timer (OST) provides 1024 oscillator cycle (from OSC1 input) delay after the PWRT delay is over. This guarantees that 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.

Time-out Sequence: On power up the time-out sequence is as follows: First PWRT time-out is invoked after POR has expired. Then TOST is activated. The total time-out will vary based on oscillator configuration and PWRTE fuse status. For example, in RC mode with PWRTE set to '0' (PWRT disabled), there will be no time-out at all. Figures 5.2.1, 5.2.2 and 5.2.3 depict time-out sequences. Table 5.2.1 shows time outs on power-up versus wake-up from SLEEP.

Since the time-outs occur from POR pulse, if $\overline{MCLR}$ is kept low long enough, the time-outs will expire. Then bringing $\overline{MCLR}$ high will begin execution immediately. This is useful for testing purposes or to synchronize more than one PIC operating in conjunction.

TABLE 5.2.1 - TIME-OUT IN VARIOUS SITUATIONS

Oscillator Configuration	Power up		Wake up from SLEEP
	PWRTE = 1	PWRTE = 0	
XT, HS, LP	72 ms + 1024 tosc	1024 tosc	1024 tosc
RC	72 ms	-	-

FIGURE 5.2.1 - TIME-OUT SEQUENCE ON POWER-UP (MCLR NOT TIED TO VDD): Case 1

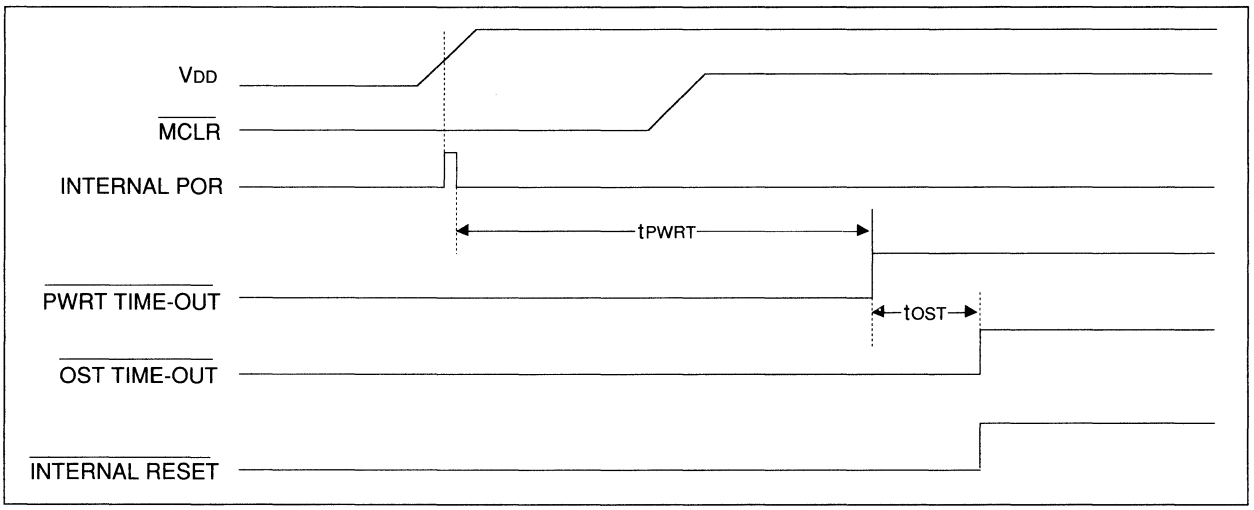


FIGURE 5.2.2 - TIME-OUT SEQUENCE ON POWER-UP (MCLR NOT TIED TO VDD): Case 2

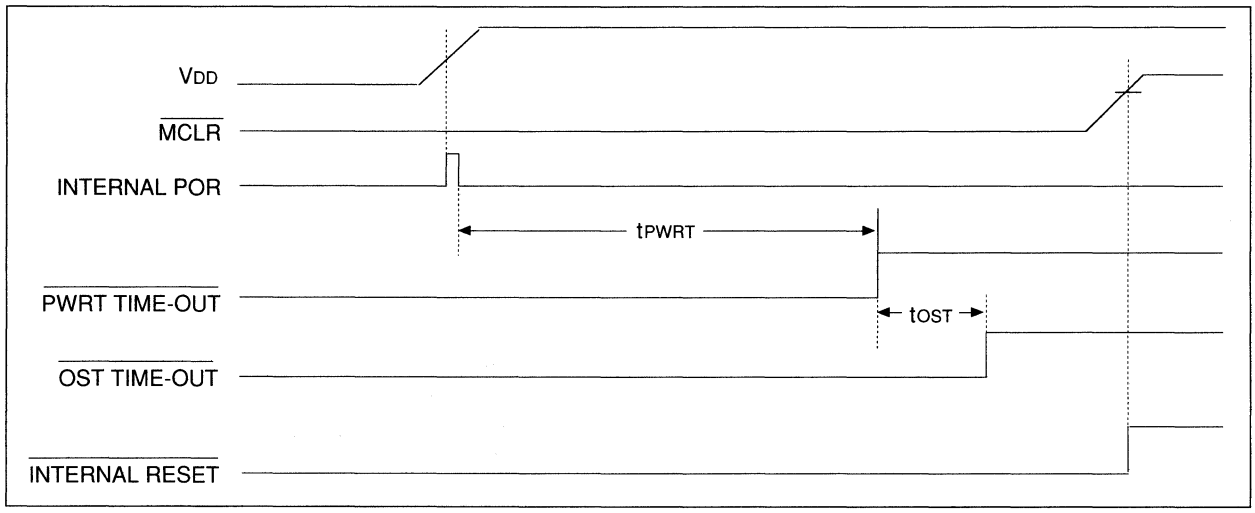


FIGURE 5.2.3 - TIME-OUT SEQUENCE ON POWER-UP (MCLR TIED TO VDD)

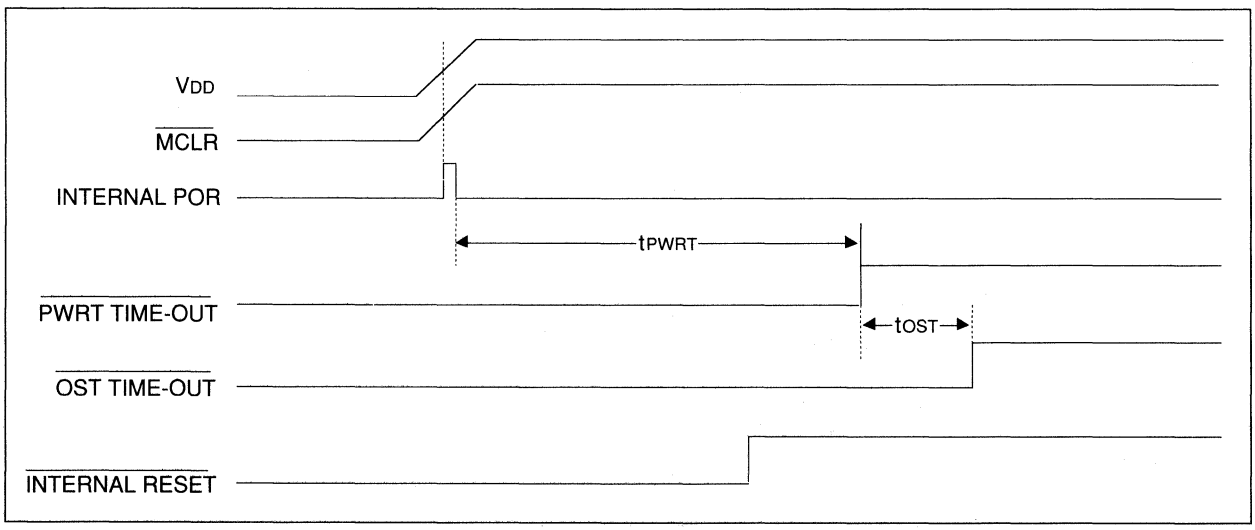
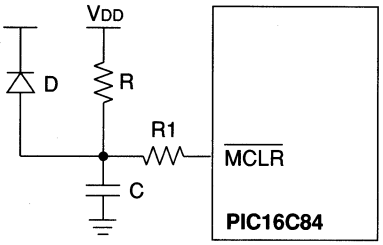
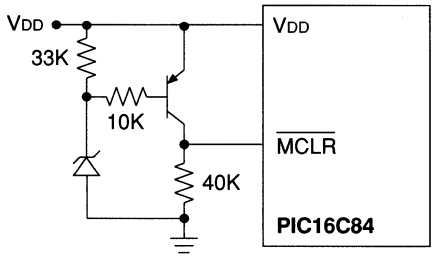


FIGURE 5.2.4 - EXTERNAL POWER ON RESET CIRCUIT



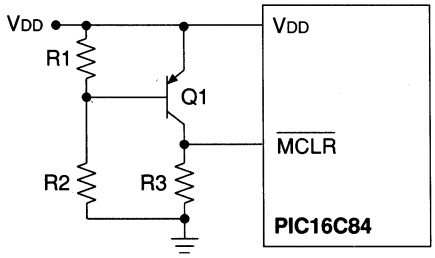
Notes:
1. External power on reset circuit is required only if VDD power-up slope is too slow. The diode D helps discharge the capacitor quickly when VDD powers down.
2. R < 40KΩ is recommended to make sure that voltage drop across R does not exceed 0.2V (max leakage current spec on MCLR pin is 5μA). A larger voltage drop will degrade VIH level on MCLR pin.
3. R1 = 100Ω to 1KΩ will limit any current flowing into MCLR from external capacitor C in the event of MCLR pin breakdown due to ESD or EOS.

FIGURE 5.2.5 - BROWN OUT PROTECTION CIRCUIT 1



Notes:
1. This circuit will activate reset when VDD goes below (Vz + 0.7V) where Vz = Zener voltage.

FIGURE 5.2.6 - BROWN OUT PROTECTION CIRCUIT 2



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

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

5.3 Watchdog Timer (WDT)

The watchdog timer is realized as a free running on-chip RC oscillator which does not require any external components. That means that the WDT will run, even if the clock on the OSC1 and OSC2 pins of the device has been stopped, for example, by execution of a SLEEP instruction. A WDT timeout generates a device RESET condition. The WDT can be permanently disabled by programming the configuration fuse WDTE as a '0' (section 5.6).

5.3.1 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 prescaler, if assigned to the WDT, and prevent it from timing out and generating a device RESET condition.

The status bit "TO" in file register f3 will be cleared upon a watchdog timer timeout.

5.3.2 WDT PROGRAMMING CONSIDERATIONS

In a noisy application environment the OPTION register can get corrupted. The OPTION register should be updated at regular intervals.

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

5.4 Oscillator Configurations

5.4.1 OSCILLATOR TYPES

The PIC16C84 can be operated in 4 different oscillator options. The user can program two configuration fuses (FOSC1 and FOSC0) to select one of these four modes.

5.4.2 CRYSTAL OSCILLATOR

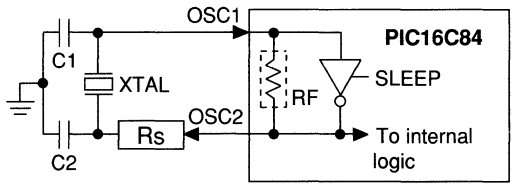
In XT, HS, or LP modes a crystal or ceramic resonator is connected to the OSC1 and OSC2 pins to establish oscillation (Figure 5.4.1).

TABLE 5.4.1 - CAPACITOR SELECTION FOR CERAMIC RESONATORS

Oscillator Type	Resonator Frequency	Capacitor Range C1 = C2
XT	455 KHz	150 - 330 pF
	2.0 MHz	20 - 330 pF
	4.0 MHz	20 - 330 pF
HS	10.0 MHz	20 - 200 pF

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

FIGURE 5.4.1 - CRYSTAL OPERATION (OR CERAMIC RESONATOR) (HS, XT OR LP OSC CONFIGURATION)



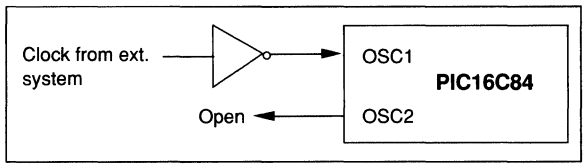
Rs may be necessary in HS mode or in XT mode for AT strip-cut crystals. See Tables 5.4.1 and 5.4.2 for recommended values of C1, C2 and Rs.

TABLE 5.4.2 - CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR

Osc Type	Freq	C1	C2
LP	32 KHz	30 pF	30-50 pF
	100 KHz	15 pF	15 pF
	200 KHz	0 - 15 pF	0 - 15 pF
XT	100 KHz	15 - 30 pF	200 - 300 pF
	200 KHz	15 - 30 pF	100 - 200 pF
	455 KHz	15 - 30 pF	15 - 100 pF
	1 MHz	15 - 30 pF	15 - 30 pF
	2 MHz	15 pF	15 pF
	4 MHz	15 pF	15 pF
HS	4 MHz	15 pF	15 pF
	10 MHz	15 pF	15 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.

FIGURE 5.4.2 - EXTERNAL CLOCK INPUT OPERATION (HS, XT, or LP OSC CONFIGURATION)

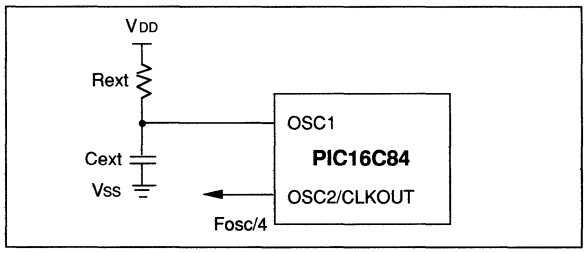


5.4.3 RC OSCILLATOR

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) and capacitor (Cext) values, and the operation 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 will also affect the oscillation frequency, especially for low Cext values. The user also needs to take into account variation to due tolerance of external R and C components used. Figure 5.4.3 shows how the R/C combination is connected to the PIC16C84. For Rext values below 2.2 kOhm, the oscillator operation may become unstable, or stop completely. For very high Rext values (e.g. 1 MOhm), the oscillator becomes sensitive to noise, humidity and leakage. Thus, we recommend to keep Rext between 3 kOhm and 100 kOhm.

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 no or small external capacitance, the oscillation frequency can vary dramatically due to changes in external capacitances, such as PCB trace capacitance or package lead frame capacitance.

FIGURE 5.4.3 - RC OSCILLATOR (RC TYPE ONLY)



See section 9.0 for RC frequency variation from part to part due to normal process variation. The variation is larger for larger R (since leakage current variation will affect RC frequency more for large R) and for smaller C (since variation of input capacitance will affect RC frequency more).

See characteristics in section 9.0 for variation of oscillator frequency due to VDD for given Rext/Cext values as well as frequency variation due to operating temperature for given R, C, and VDD values.

In RC mode, 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 3.2.1 for timing).

5.5 Power Down Mode (Sleep)

The power down mode is entered by executing a SLEEP instruction.

If enabled, the watchdog timer will be cleared but keeps running, the bit "PD" in the status register (f03) is cleared, the "TO" bit is set, and the oscillator driver is turned off. The I/O ports maintain the status they had, before the SLEEP command was executed (driving high, low, or hi-impedance).

For lowest curent consumption in this mode, all I/O pins should be either at VDD, or VSS, with no external circuitry drawing current from the I/O pin. I/O pins that are in the High-Z mode should be pulled high or low externally to avoid switching currents caused by floating inputs. The RTCC input should also be at VDD or VSS for lowest current consumption. The contribution from on chip pull-ups on PortB should be considered.

The 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 MCLR pin low.

5.5.1 WAKE-UP FROM SLEEP

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

a. External reset input on $\overline{\text{MCLR}}$ pin

b. Watchdog timer timeout reset (if WDT was enabled)

c. Interrupt from INT pin, RB port change or data EEPROM write completion.

The first event will cause a device reset. The two latter events are considered a continuation of program execution. The TO and PD bits in the STATUS register can be used to determine the cause of device reset. PD bit, which is set on power-up is cleared when SLEEP is invoked. TO bit is cleared if WDT time-out occurred (and caused wake-up).

For the device to wake up through an interrupt, the corresponding interrupt mask bit must be enabled. On wake-up, the device will continue to execute code in-line if global interrupt was disabled (GIE = 0) or branch to interrupt service routine if GIE was enabled (see figure 5.5.5.1).

5.6 Configuration Fuses

The PIC16C84 has five configuration fuses which are EEPROM bits. These fuses can be programmed (reads '0') or left unprogrammed (reads '1') to select various device configurations. These bits are mapped in program memory location 2007h.

The user will note that address 2007h is beyond the user program memory space. In fact, it belongs to the special test/configuration memory space (2000h - 3FFFh). However, through a special mode, this location can be accessed during programming.

See description of fuses in figure 5.6.1.

5.7 ID Locations

The PIC16C84 has four ID locations (2000h - 2003h) mapped in the test program memory for storing code revision number, manufacturing information or other useful information. As with the configuration word, these locations are readable and writable through a programmer. They are not accessible during normal code execution.

If the chip is code protected, it is recommended that the user uses only the lower seven bits of the ID locations and program the higher seven bits as '0'. This way the ID locations will be readable even after code protection.

5.8 Code Protection

The code in the program memory can be protected by blowing the code protect fuse (CP).

When code protected, the contents of the program memory cannot be read out in a way that the program code can be reconstructed. In addition, all memory locations are protected against programming.

All EPROM data memory locations can not be programmed nor can they be read out.

Once code protected, the CP fuse can be erased only through a chip erase. A chip erase will erase EEPROM program and data memory before erasing the code-protect fuse. Refer to PIC16C84 programming specification for details.

5.8.1 VERIFYING A CODE-PROTECTED PIC

When code protected verifying any program memory location will read a scrambled output which looks like "0000000xxxxxxx" (binary) where X is 1 or 0. To verify a device after code protection, follow this procedure:

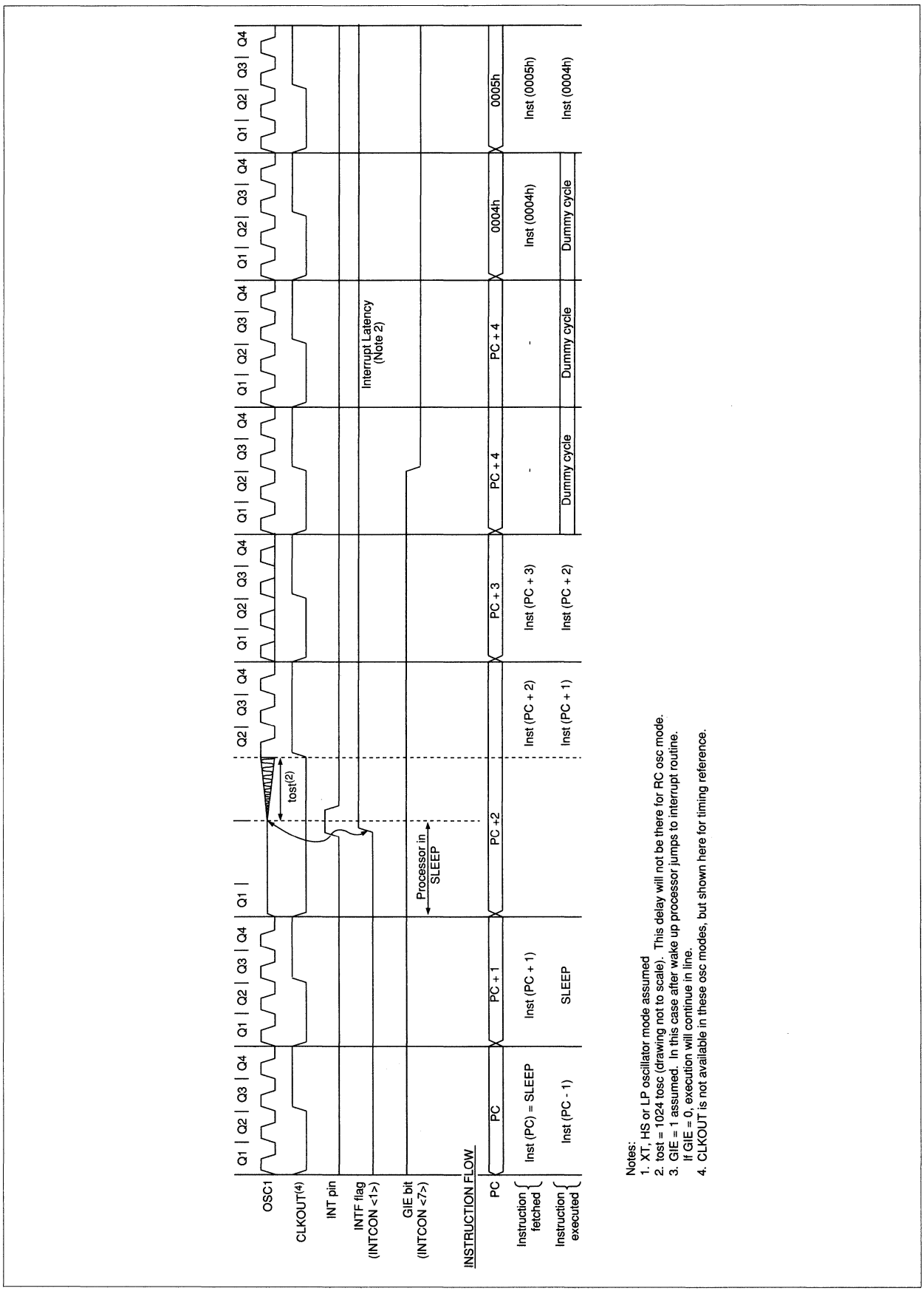
a. First, program and verify a good device without code protecting it.

b. Next, blow its code protection fuse and then load its contents in a file.

c. Verify any code-protected PIC against this file.

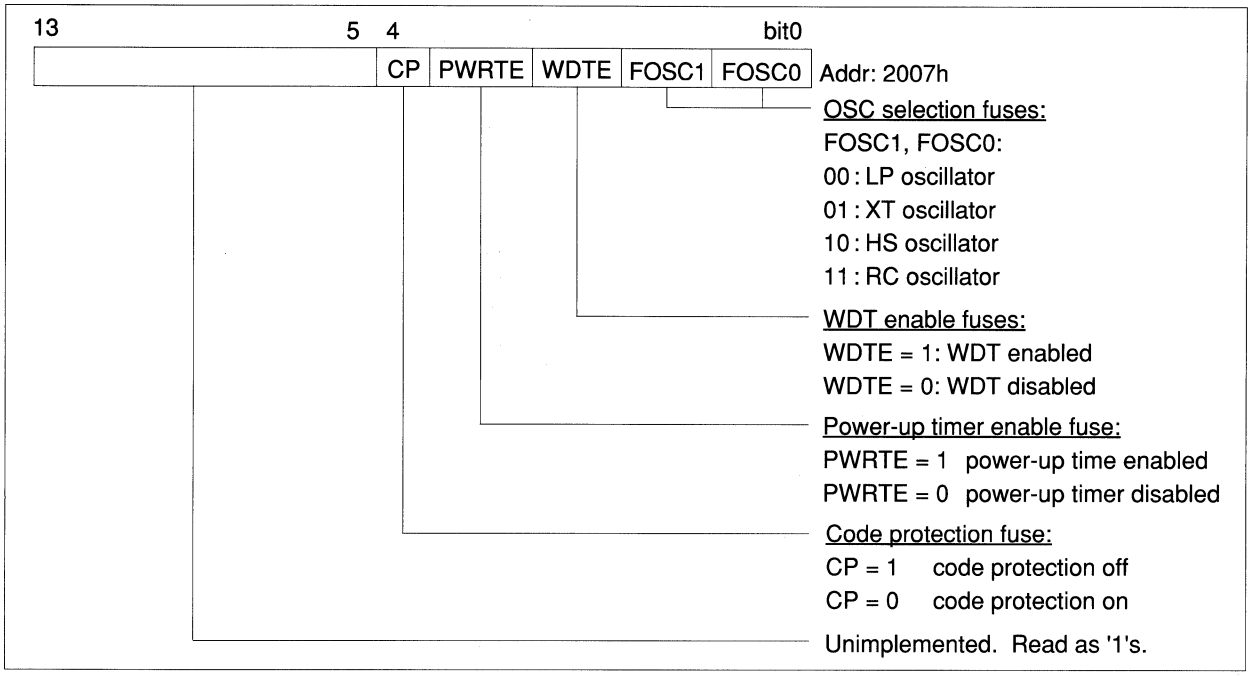
EEPROM data memory can not be verified after code protection. The user can embed code for self testing the data memory in the program.

FIGURE 5.5.1.1 - WAKE UP FROM SLEEP THROUGH INTERRUPT



Notes:
1. XT, HS or LP oscillator mode assumed
2. tost = 1024 tosc (drawing not to scale). This delay will not be there for RC osc mode.
3. GIE = 1 assumed. In this case after wake up processor jumps to interrupt routine.
   If GIE = 0, execution will continue in line.
4. CLKOUT is not available in these osc modes, but shown here for timing reference.

FIGURE 5.6.1 - CONFIGURATION WORD



6.0 OVERVIEW OF PERIPHERALS

The PIC16C84 has 13 I/O pins organized as two I/O ports, PortA (5 bit) and PortB (8-bit). There is one general purpose timer/counter, RTCC which 8-bit wide with 8-bit programmable prescaler. It is separate from the watchdog timer. The PIC16C84 also has a 64 x 8 EEPROM data memory accessible through an 8-bit data register and address register.

6.1 PortA

PortA is a 5 bit wide port with pins RA<4:0>. Port pins RA<3:0> are bidirectional whereas RA4 has a open-collector output. PortA is file register 05h. Its corresponding direction control register TRISA is mapped in page 1 of register file at address 85h. TRISA is a five-bit wide register with bits <4:0> physically implemented. Refer to figures 6.1.1 and 6.1.2 for block diagram of PortA pins.

FIGURE 6.1.1 - BLOCK DIAGRAM OF RA0 - RA3 PINS

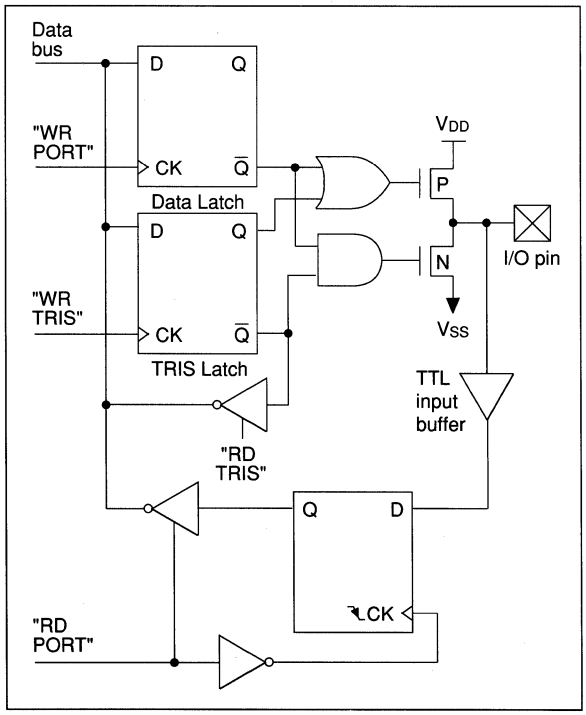


TABLE 6.1.1 - PORTA FUNCTIONS

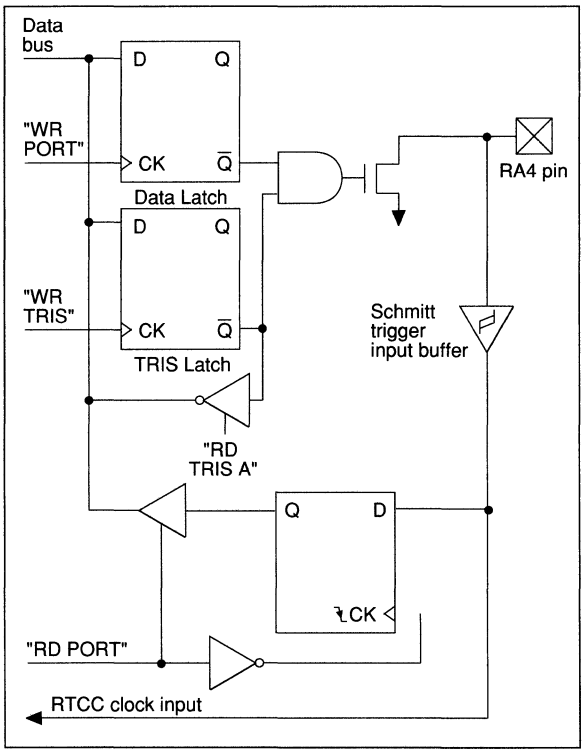
Port Pin	Bit	Pin Function	Alternate Function
RA0	bit0	Input/output port. TTL input levels	-
RA1	bit1	Input/output port. TTL input levels	-
RA2	bit2	Input/output port. TTL input levels	-
RA3	bit3	Input/output port. TTL input levels	-
RA4/RT	bit4	Input/output port. Output is open collector type. Input is Schmitt trigger type.	External clock input for RTCC timer/counter

TABLE 6.1.2 - SUMMARY OF PORTA REGISTERS

Register Name	Function	Address	Power-on Reset Value
PORTA	PortA pins when read PortA latch when written	05h	---X XXXX
TRISA	PortA data direction register	85h	---1 1111

Notes: 1: x = unknown, - = unimplemented, reads as a '0'.
2: For reset values of registers in other reset situations refer to table 5.1.1.

FIGURE 6.1.2 - BLOCK DIAGRAM OF RA4 PIN



6.2 PortB

PortB is an 8-bit wide bidirectional port (file register address 06h). The corresponding data direction register is TRISB (address 86h). A '1' in TRISB sets the corresponding port pin as an input. Reading PortB register reads the status of the pins whereas writing to it will write to the port latch. See figures 6.2.1 and 6.2.2 for block diagrams of the PortB pins.

Each of the PortB pins has a weak internal pull-up (~100 µA typical). The weak pull-up is automatically turned off if the port pin is configured as an output. Furthermore, a single control bit RBPU OPTION<7> can turn off (RBPU = 1) all the pull-ups. The pull-ups are disabled on power on reset.

Port B has an interrupt on change feature on four of its pins, RB<7:4>. When configured as input, the inputs on these pins are sampled and latched every Q1. The new input is compared with the old latched value in every instruction cycle. An active high output is generated on mismatch between the pin and the latch. The "mismatch" outputs of RB4, RB5, RB6 and RB7 are OR'ed together to generate the RBIF interrupt (latched in INTCON<0>. Any pin configured as output is excluded from the comparison. This interrupt can wake the chip up from SLEEP. The user, in interrupt service routine can clear the interrupt in one of two ways:

a) Disable the interrupt by clearing RBIE INTCON<3> bit.

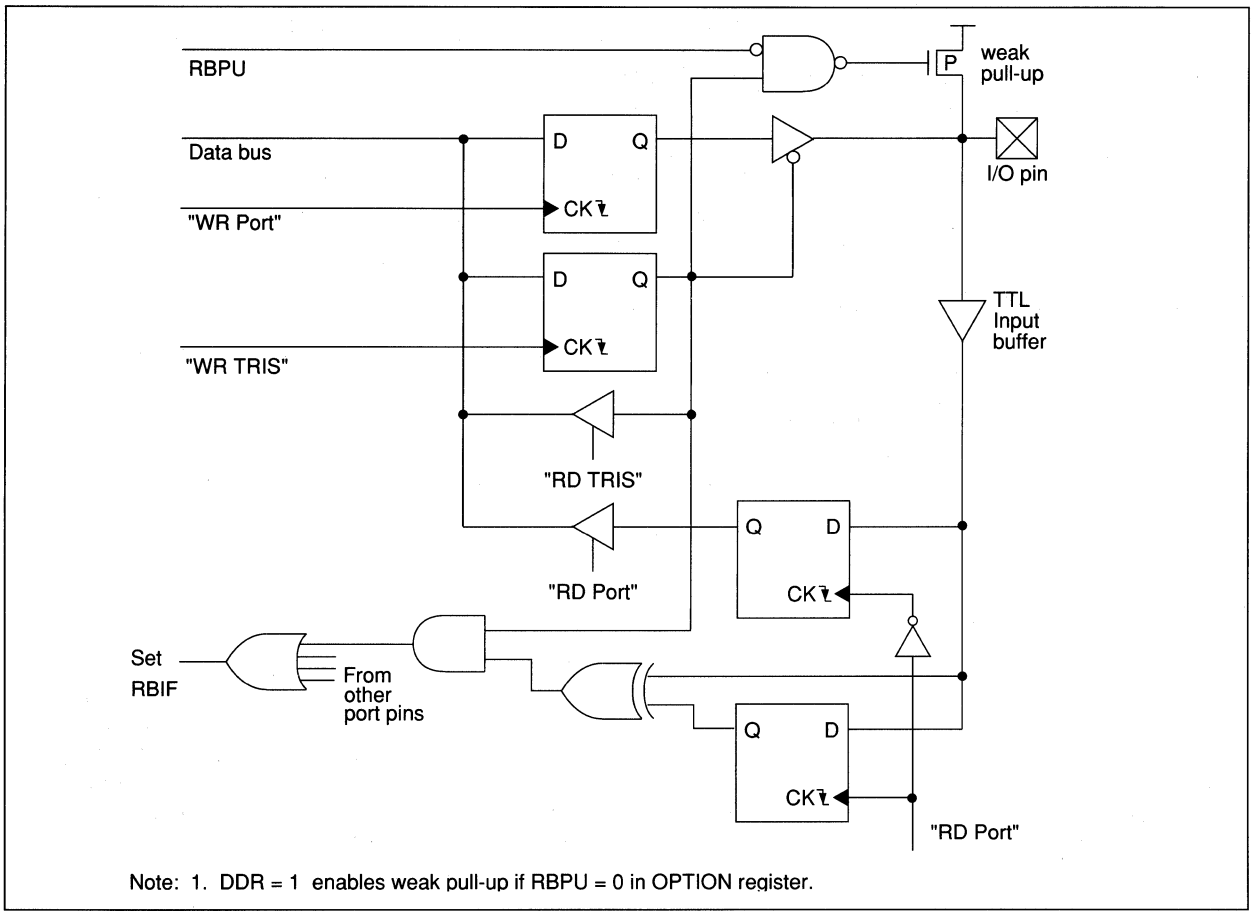
b) Read Port B. This will end mismatch condition.

c) Next, clear RBIF bit.

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.

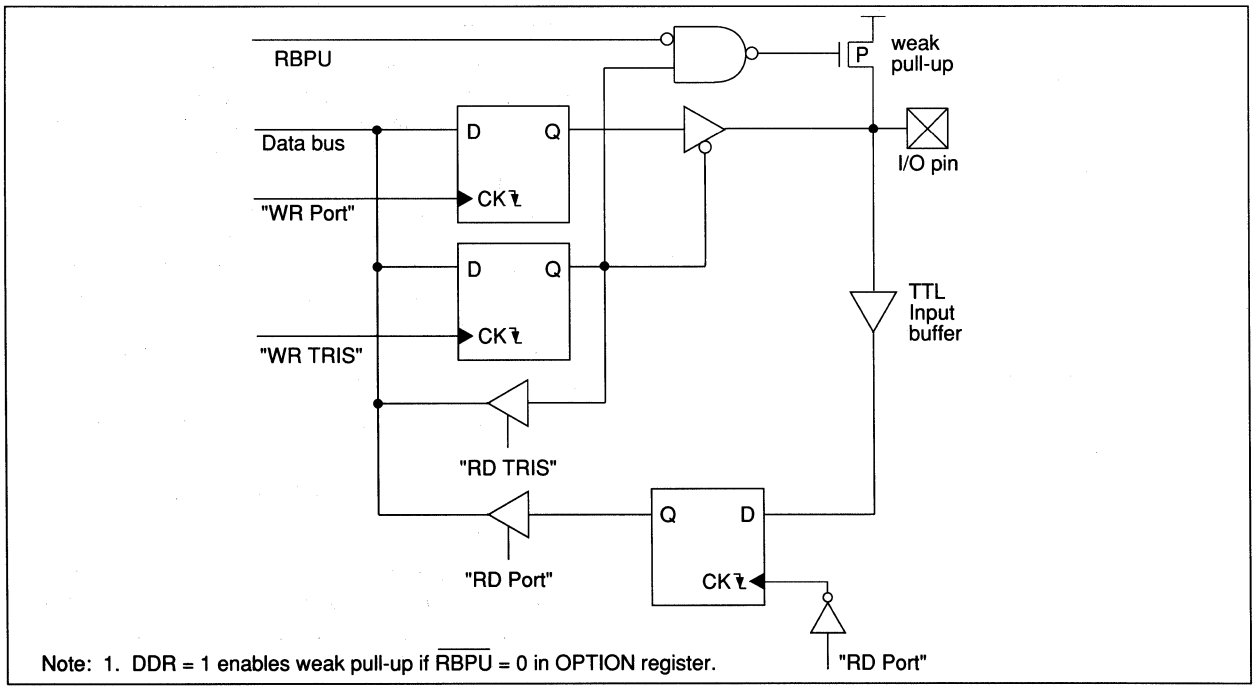
Finally, port pin RB0 is multiplexed with external interrupt input INT.

FIGURE 6.2.1 - BLOCK DIAGRAM OF PORT PINS RB<7:4>



Note: 1. DDR = 1 enables weak pull-up if RBPU = 0 in OPTION register.

FIGURE 6.2.2 - BLOCK DIAGRAM OF PORT PINS RB<3:0>



Note: 1. DDR = 1 enables weak pull-up if $\overline{\text{RBPU}}$ = 0 in OPTION register.

TABLE 6.2.1 - PORTB FUNCTIONS

Port Pin	Bit	Pin Function	Alternate Function
RB0/INT	bit0	Input/output port pin. TTL input levels and internal software programmable weak pull-up	External interrupt input
RB1	bit1	Input/output port pin. TTL input levels and internal software programmable weak pull-up	-
RB2	bit2	Input/output port pin. TTL input levels and internal software programmable weak pull-up	-
RB3	bit3	Input/output port pin. TTL input levels and internal software programmable weak pull-up	-
RB4	bit4	Input/output port pin. TTL input levels and internal software programmable weak pull-up	Interrupt on port change
RB5	bit5	Input/output port pin. TTL input levels and internal software programmable weak pull-up	Interrupt on port change
RB6	bit6	Input/output port pin. TTL input levels and internal software programmable weak pull-up	Interrupt on port change
RB7	bit7	Input/output port pin. TTL input levels and internal software programmable weak pull-up	Interrupt on port change

TABLE 6.2.2 - SUMMARY OF PORTB REGISTERS

Register Name	Function	Address	Power-on Reset Value
PORTB	PortB pins when read PortB latch when written	06h	XXXX XXXX
TRISB	PortB data direction register	86h	1111 1111
OPTION	Weak pull-up on/off control ($\overline{\text{RBPU}}$ bit)	88h	1111 1111

6.3 I/O Programming Considerations

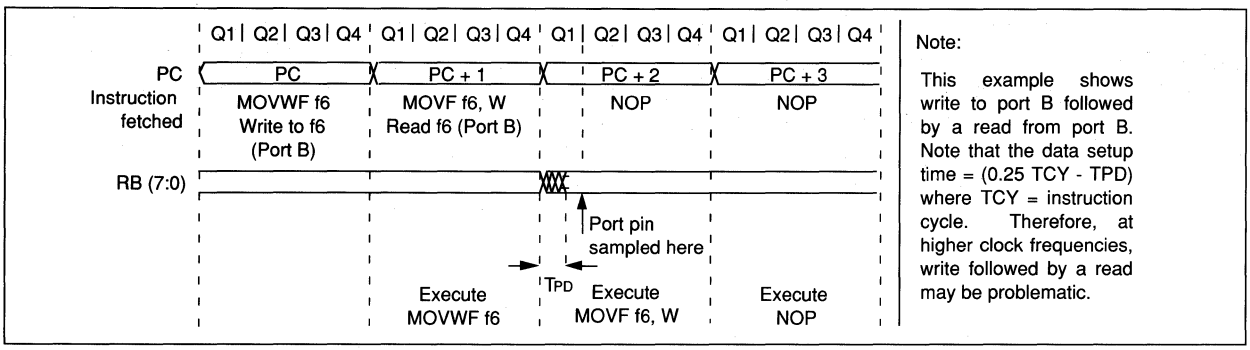
6.3.1 BIDIRECTIONAL I/O PORTS

Some instructions operate internally as read followed by write operations. The BCF and BSF instructions, for example, read the entire port into the CPU, execute the bit operation, and re-output the result. Caution must be used when these instructions are applied to a port where one or more pins are used as input/outputs. For example, a BSF operation on bit 5 of f6 (Port B) will cause all eight bits of f6 to be read into the CPU. Then the BSF operation takes place on bit 5 and f6 is re-output to the output latches. If another bit of f6 is used as a bidirectional I/O pin (say bit 0) and it is defined as an input at this time, the input signal present on the pin itself would be read into the CPU and re-written to the data latch of this particular pin, overwriting the previous content. As long as the pin stays in the input mode, no problem occurs. However, if bit 0 is switched into output mode later on, the content of the data latch may now be unknown.

A pin actively outputting a "0" or "1" 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 currents may damage the chip.

For "wired-or" outputs (assuming negative logic), it is recommended to use external pull-up resistors on the corresponding pins. The pin should be left in high-impedance mode, unless a "0" has to be output. Thus, external devices can drive this pin "0" as well. "Wired-and" outputs can be realized in the same way, but with external pull-down resistors and only actively driving the "1" level from the PIC. The resistor values are user selectable, but should not force output currents above the specified limits (see DC Characteristics).

FIGURE 6.3.1 - SUCCESSIVE I/O OPERATION



6.3.2 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 (see figure 6.3.1). Therefore, care must be exercised if a write followed by a read operation is carried out on the same I/O port. The sequence of instructions should be such to allow the pin voltage to stabilize (load dependent) before 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 an other instruction not accessing this I/O port.

6.3.3 OPERATION IN NOISY ENVIRONMENT

In noisy application environments, such as keyboards which are exposed to ESD (Electro Static Discharge), register contents can get corrupted due to noise spikes. The on-chip watchdog timer will take care of all situations involving program sequence "lock-ups." However, if an I/O control register gets corrupted, the program sequence may still be executed properly although an input pin may have switched unintentionally to an output. In this case, the program would always read the same value on this pin. This may result, for example, in a keyboard "lock-up" situation without leading to a watchdog timer timeout. Thus, it is recommended to redefine all I/O pins in regular time intervals (inputs as well as outputs). The optimal strategy is to update the I/O control register every time before reading input data or writing output data .

6.4 Real Time Clock/Counter (RTCC)

The RTCC timer/counter has the following features:

- 8 bit timer/counter
- Readable and writable (file address 01h)
- 8 bit software programmable prescaler
- Internal or external clock select
- Interrupt on overflow

Figure 6.4.1 is a simplified block diagram of the RTCC module.

Timer mode is selected by setting RTS bit to '0' (OPTION register). In timer mode, the RTCC will increment every instruction cycle (without prescaler). If RTCC (f01) is written, increment is inhibited for the following two cycles (see figures 6.4.2 and 6.4.3). The user can work around this by writing an adjusted value to the RTCC.

Counter mode is selected by setting RTS bit to '1' (OPTION register). In this mode RTCC will increment either on every rising or falling edge of pin RA4/RTCC. This is determined by control bit RTE (OPTION register). RTE = 0 selects rising edge. Restrictions on external clock input is discussed in detail in section 6.4.2.

The prescaler is shared between the RTCC and the watchdog timer. The prescaler assignment is controlled in software by control bit, PSA (OPTION register). PSA = 0 will assign the prescaler to RTCC. The prescaler is not readable or writable. When the prescaler is assigned to the RTCC, prescale value of 1:2, 1:4, ..., 1:256 are selectable. Section 6.4.3 details the operation of the prescaler.

RTCC interrupt is generated when the RTCC timer/counter overflows from FFh to 00h. It sets the RTIF bit INTCON<2>. The interrupt can be masked by setting RTIE bit INTCON<5> to '0'. RTIF bit must be cleared in software in the RTCC interrupt service routine before re-enabling this interrupt. See figure 6.4.4 for RTCC interrupt timing. The RTCC interrupt can not wake the processor from SLEEP since the timer is shut off during SLEEP.

6.4.2 USING RTCC WITH EXTERNAL CLOCK

When external clock input is used for RTCC, it is synchronized with the internal phase clocks. Therefore, the external clock input must meet certain requirements.

Also there is some delay from the occurance of the external clock edge to the actual incrementing of RTCC. Referring to Figure 6.4.1.1, the synchronization is done after the prescaler. The output of the prescaler is sampled twice in every instruction cycle to detect rising or falling edges. Therefore, it is necessary for PSOUT to be high for at least 2 tosc and low for at least 2 tosc where tosc = oscillator time period.

When no prescaler is used, PSOUT (Prescaler output, see Figure 6.4.1) is the same as RTCC clock input and therefore the requirements are:

TRTH = RTCC high time ≥ 2tosc + 20 ns

TRTL = RTCC low time ≥ 2tosc + 20 ns

When prescaler is used, the RTCC input is divided by the asynchronous ripple counter-type prescaler and so the prescaler output is symmetrical.

Then: PSOUT high time = PSOUT low time = $\frac{N \cdot T_{RT}}{2}$

where TRT = RTCC input period and N = prescale value (2, 4, ..., 256). The requirement is, therefore $\frac{N \cdot T_{RT}}{2}$ ≥ 2 tosc + 20 ns, or $T_{RT} \geq \frac{4 \text{ tosc} + 40 \text{ ns}}{N}$.

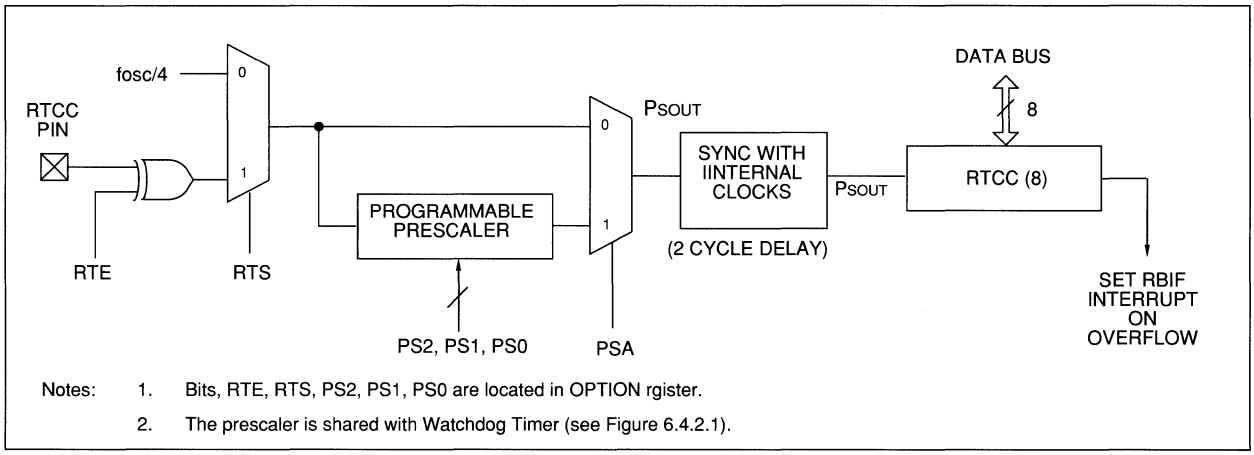
The user will notice that no requirement on RTCC high time or low time is specified. However, if the high time or low time on RTCC is too small then the pulse may not be detected, hence a minimum high or low time of 10 ns is required. In summary, the RTCC input requirements are:

TRT = RTCC period ≥ (4 tosc + 40 ns)/N

TRTH = RTCC high time ≥ 10 ns

TRTL = RTCC low time ≥ 10 ns

FIGURE 6.4.1 - RTCC BLOCK DIAGRAM (SIMPLIFIED)



Notes: 1. Bits, RTE, RTS, PS2, PS1, PS0 are located in OPTION rgister.

2. The prescaler is shared with Watchdog Timer (see Figure 6.4.2.1).

FIGURE 6.4.2 - RTCC TIMING: INT CLOCK/NO PRESCALE

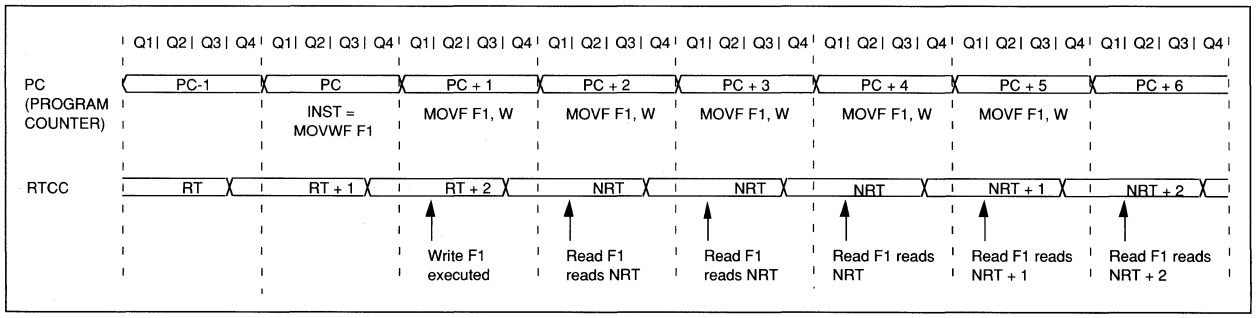


FIGURE 6.4.3 - RTCC TIMING: INT CLOCK/PRESCALE 1:2

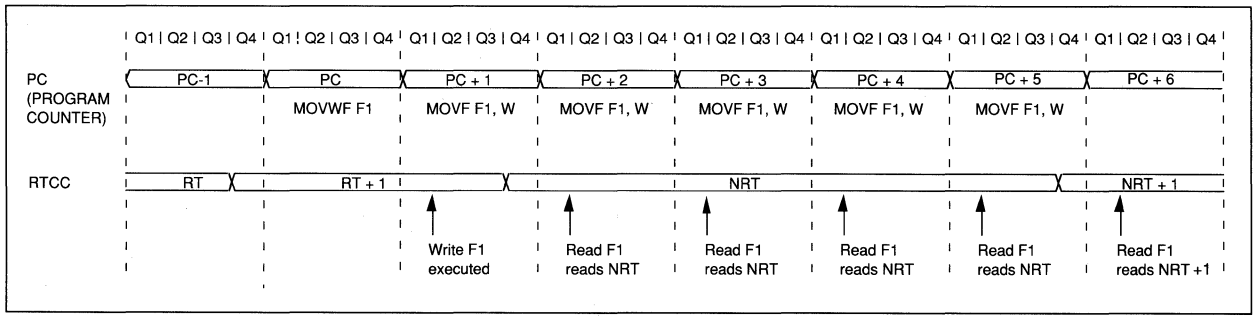
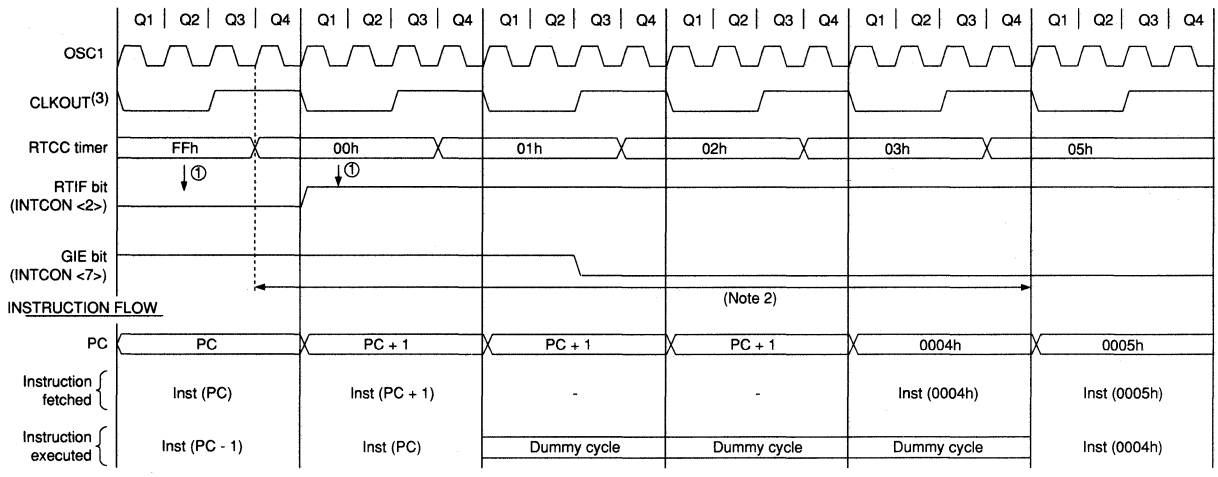


FIGURE 6.4.4 - RTCC INTERRUPT TIMING



Notes:
1. RTIF interrupt flag is sampled here (every Q1)
2. Interrupt latency = 4 Tcy where Tcy = instruction cycle time.
3. CLKOUT is available only in RC oscillator mode.

Delay from external clock edge: Since the prescaler output is synchronized with the internal clocks, there is a small delay from the time the external clock edge occurs to the time the RTCC is actually incremented. Referring to Figure 6.4.1.1, the reader can see that this delay is between 3 tosc and 7 tosc. Thus, for example, measuring the interval between two edges (e.g. period) will be accurate within ±4 tosc (±400 ns @ 10 MHz).

6.4.3 PRESCALER

An 8-bit counter is available as a prescaler for the RTCC, or as a post-scaler for the watchdog timer, respectively (Figure 6.4.3.1). 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 exclusively shared between the RTCC and the watchdog timer. Thus, a prescaler assignment for the RTCC means that there is no prescaler for the watchdog timer, and vice versa.

The PSA and PS0-PS2 bits in the OPTION register determine the prescaler assignment and pre-scale ratio. When assigned to the RTCC, all instructions writing to the RTCC (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.

```
1. MOVLW  B'xx0x0xxx'; Select internal clock and select new
2. OPTION             ;prescaler value. If new prescale value
                      ; is = '000' or '001', then select any
                      ; other prescale value temporarily.
3. CLRF 1             ; Clear RTCC and prescaler.
4. MOVLW  B'xxxx1xxx'; Select WDT, do not change prescale
                      ; value.
5. OPTION             ;
6. CLRWDT             ; Clears WDT and prescaler.
7. MOVLW  B'xxxx1xxx'; Select new prescale value.
8. OPTION                       ;
```

Step 1 and 2 are only required if an external RTCC source is used. Steps 7 and 8 are necessary only if the desired prescale value is '000' or '001'.

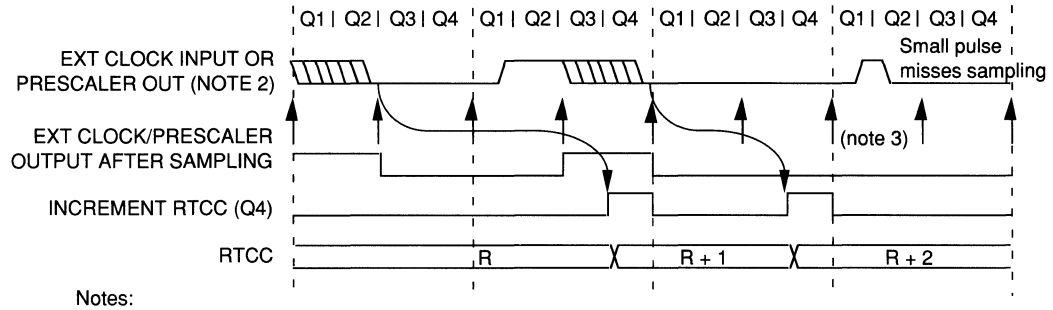
CHANGING PRESCALER FROM WDT TO RTCC

To change prescaler from WDT to RTCC use the following sequence:

```
1. CLRWDT             ; Clear WDT and prescaler
2. MOVLW B'xxxx0xxx'  ; Select RTCC, new prescale value
                      ; and clock source
3. OPTION             ;
```

The above precautions must be taken even if WDT is disabled.

FIGURE 6.4.2.1 - RTCC TIMING WITH EXTERNAL CLOCK



Notes:

1. Delay from clock input change to RTCC increment is 3 tosc to 7 tosc. (Duration of Q = tosc). Therefore, the error in measuring the interval between two edges on RTCC input = ± 4 tosc max.
2. External clock if no prescaler selected, Prescaler output otherwise.
3. The arrows indicate the points in time where sampling occurs.

TABLE 6.4 - SUMMARY OF RTCC REGISTERS

Register Name	Function	Address	Power-on Reset Value
RTCC	Timer/counter register	01h	XXXX XXXX
OPTION	Configuration and prescaler assignment bits for RTCC	81h	1111 1111
INTCON	RTCC overflow interrupt flag and mask bits	0Bh	0000 000X

FIGURE 6.4.3.1 - BLOCK DIAGRAM OF THE RTCC/WDT PRESCALER

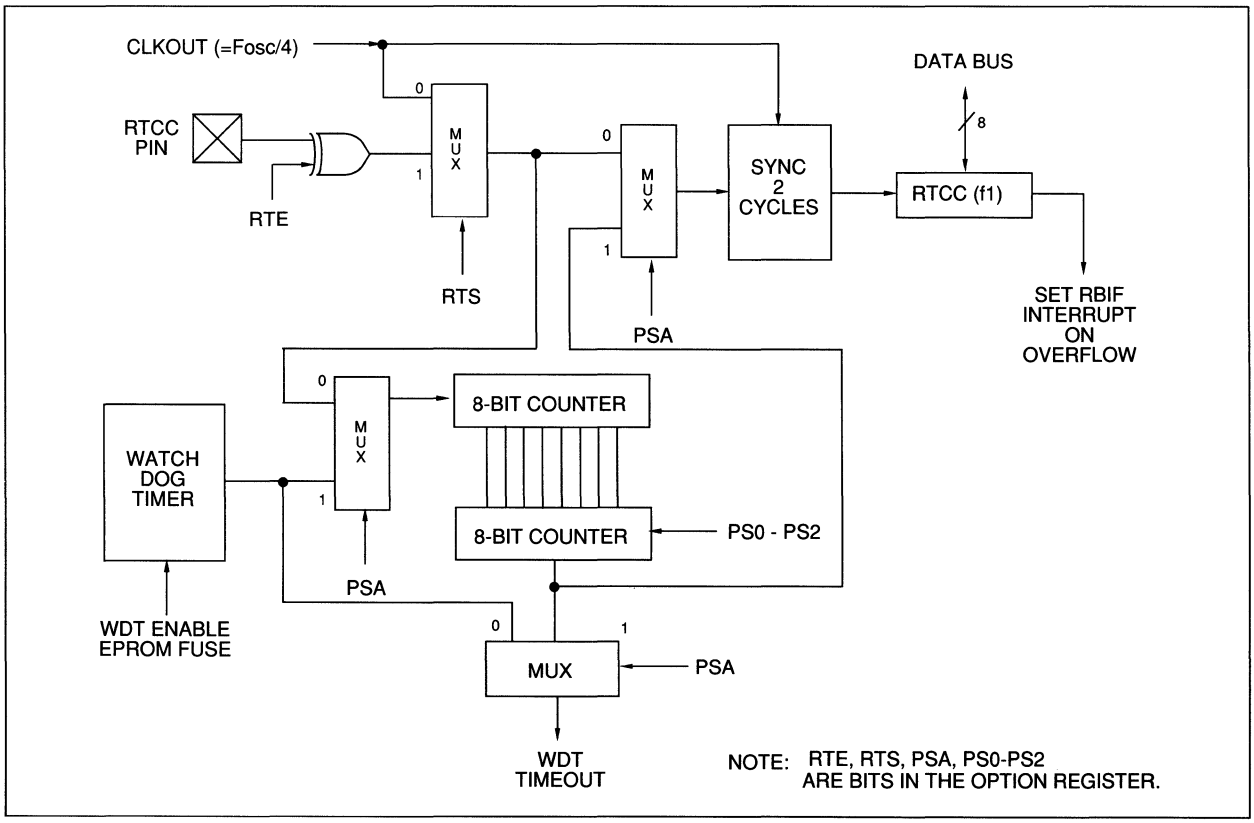
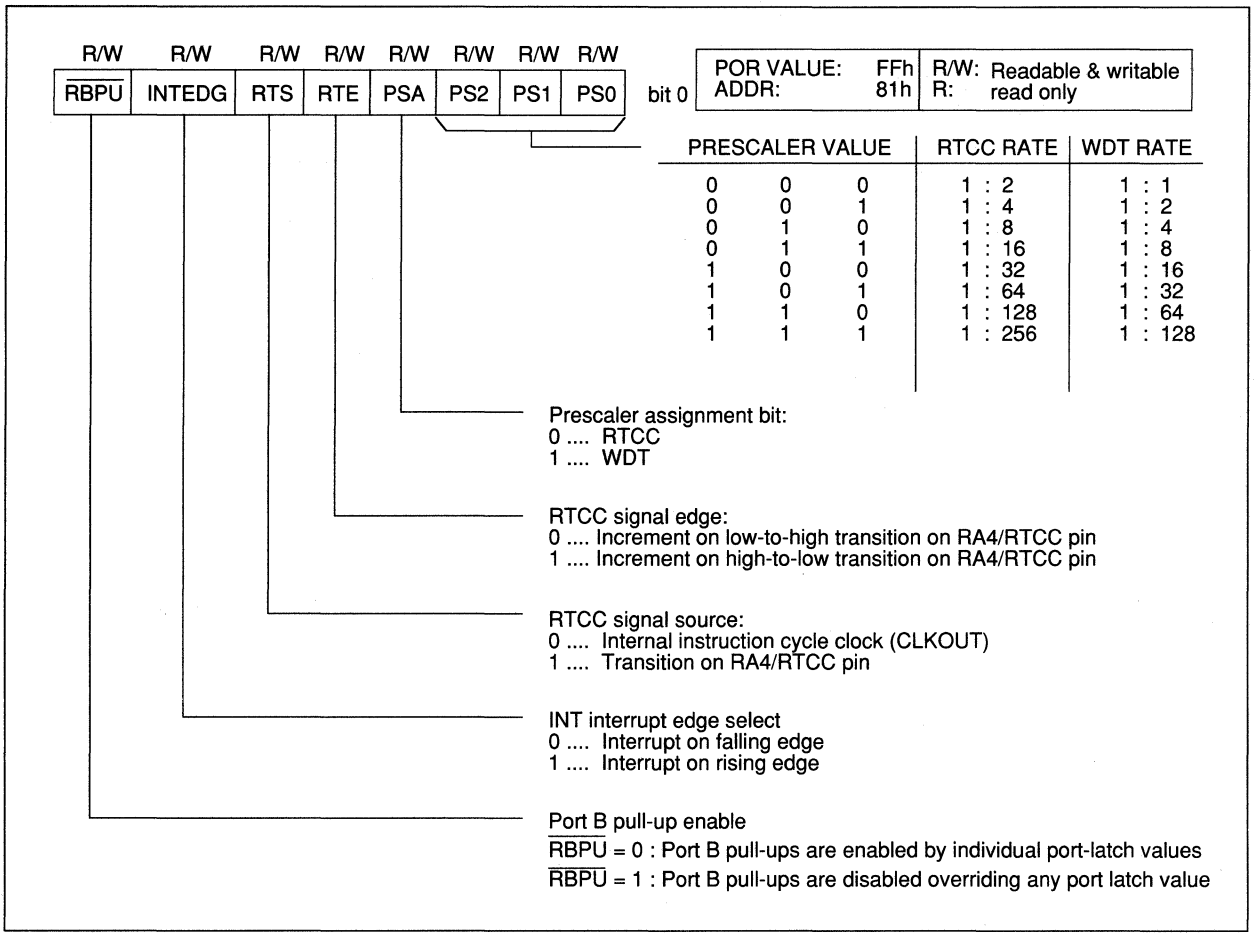


FIGURE 6.5.1 - OPTION REGISTER



6.5 OPTION Register

The OPTION register (address 81h) is a readable and writable register which contains various control bits to configure the prescaler, the external INT interrupt edge select, the RTCC and the weak pull-ups on PortB.

6.6 EEPROM Data Memory

The PIC16C84 has 64x8 EEPROM data memory which is readable and writable during normal operation (full VDD range). This memory is not directly mapped in the register file space. Instead it is accessed through two registers: EEDATA<08h> which holds the 8-bit data for read/write, EEADR<09h> which holds the address of the EEPROM location being accessed. Additionally, there are two control registers: EECON1<88h> and EECON2<89h>.

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 nominally 10 ms, and is controlled by an on-chip timer. The actual write-time will vary with voltage and temperature as well as from chip to chip. Please refer to AC specifications for exact limits.

6.6.1 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).

6.6.2 WRITING TO THE EEPROM DATA MEMORY

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

```
movlw   55h
movwf   EECON2
movlw   AAh
movwf   EECON2
bsf     EECON1,WR   ;set WR bit
                    ;begin write
```

Write will not initiate if this sequence (write 55h to EECON2, write AAh to EECON2, then set WR bit) is not followed with exact timing. The user must disable interrupts during this code segment.

Additionally WREN bit in EECON1 must be set to '1' to enable write. This mechanism is to prevent accidental writes to data EEPROM due to errant (unexpected) code execution i.e. lost programs. The user is recommended to keep WREN off at all times except when updating EEPROM. Furthermore, the code segments that enables WREN and initiates write should be kept at separate locations to prevent accidental execution of both of them in the event of a software malfunction.

At the end of the write, the WR bit is cleared in hardware and the EE write complete interrupt flag is set (bit EEIF). The user can either enable this interrupt or poll this bit. EEIF must be cleared in software.

TABLE 6.6.2 - SUMMARY OF EEPROM REGISTERS

Register Name	Function	Address	Power-on-Reset Value
EEDATA	EEPROM data register	08h	XXXX XXXX
EEADR	EEPROM address register	09h	XXXX XXXX
EECON1	EEPROM control register1	88h	0000 X000
EECON2	EEPROM control register2	89h	-

6.6.3 EECON1 AND EECON2 REGISTERS

EECON1 (address 88h) is the control register with 5 low order bits physically implemented. The upper 3-bits are non existent and read as '0's.

Control bits RD and WR initiate read and write, respectively. These bits can only be set in software. They are reset in hardware at completion of read or write operation. Inability to clear WR bit in software prevents accidental termination of a write operation prematurely.

WREN bit, when set will allow a write operation. On power-up WREN = 0. WRERR bit is set when a write operation is interrupted by $\overline{MCLR}$ reset or a WDT time-out reset during normal operation. In these situations, following reset the user can check for WRERR bit and rewrite the location. The data and address will be unchanged in EEDATA and EEADR registers.

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

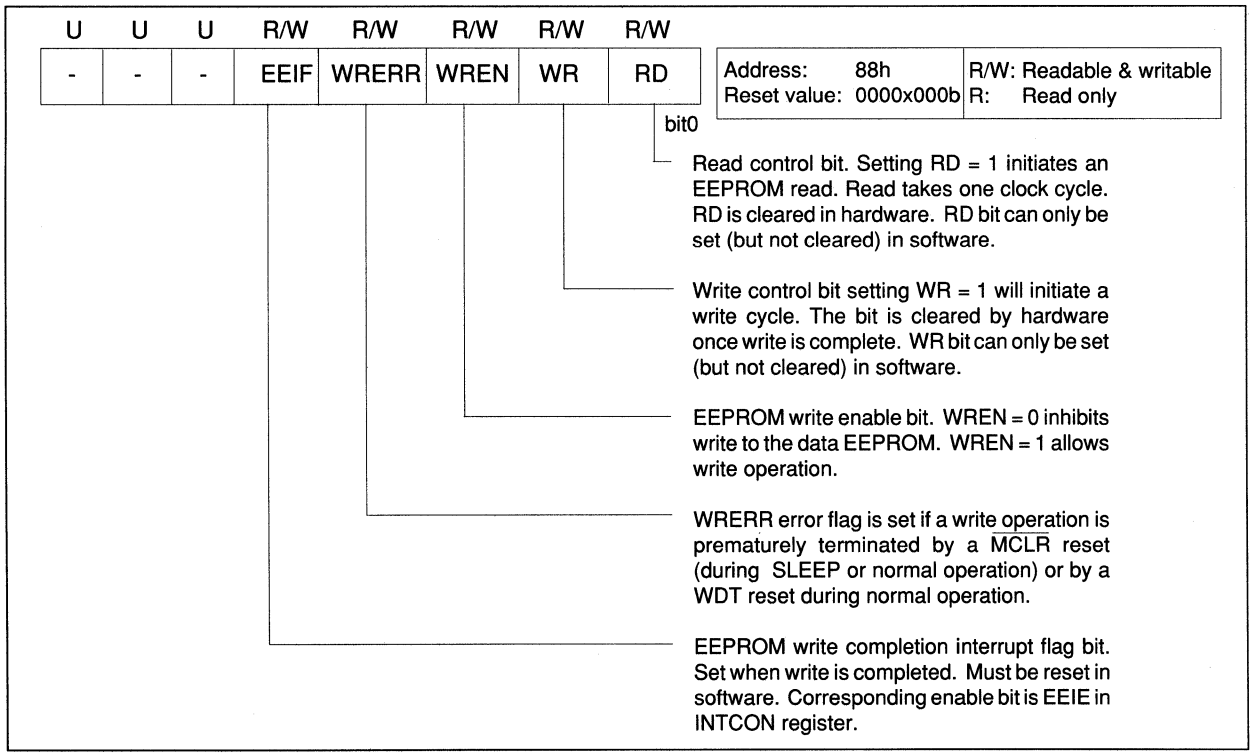
EECON2 is not a physical register. Reading EECON2 will read '0's.

6.6.4 PROTECTION AGAINST SPURIOUS WRITE

Various mechanisms are built in to prevent spurious EEPROM write. On power-up WREN is set to '0'. 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 or power glitch or software malfunction.

FIGURE 6.6.3.1 - EECON1 REGISTER



7.0 ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings*

Ambient temperature under bias .....-55 to+ 125°C
Storage Temperature ..... - 65°C to +150°C
Voltage on any pin with respect to VSS
(except VDD and MCLR) .....-0.6V to VDD +0.6V
Voltage on VDD with respect to VSS .....0 to +7.5 V
Voltage on MCLR with respect to VSS
(Note 2) .....0 to +14 V
Total power Dissipation (Note 1) .....800 mW
Max. Current out of VSS pin .....150 mA
Max. Current into VDD pin .....100 mA
Max. Current into an input pin .....±500 µA
Max. Output Current sunk by any I/O pin .....25 mA
Max. Output Current sourced by any I/O pin .....20 mA
Max. Output Current sunk by I/O port A .....80 mA
Max. Output Current sunk by I/O port B .....150 mA
Max. Output Current sourced by I/O port A .....50 mA
Max. Output Current sourced by I/O port B .....100 mA

*Notice: Stresses above those listed under "Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device or compliance to AC and DC parametric specifications 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.

Notes: 1. Total power dissipation should not exceed 800 mW for the package. Power dissipation is calculated as follows:
$Pdis = V_{DD} \times \{I_{DD} - \sum I_{oh}\} + \sum\{(V_{DD}-V_{oh}) \times I_{oh}\} + \sum(V_{ol} \times I_{ol})$

2. Voltage spikes below VSS at the MCLR pin, inducing currents greater than 80mA, may cause latch-up. Thus, a series resistor of 50-100Ω should be used when applying a "low' level to the MCLR pin rather than pulling this pin directly to VSS.

7.1 DC CHARACTERISTICS: PIC16C84-04 (COMMERCIAL, INDUSTRIAL, AUTOMOTIVE) PIC16C84-10 (COMMERCIAL, INDUSTRIAL, AUTOMOTIVE)

DC CHARACTERISTICS

Standard Operating Conditions (unless otherwise stated)
Operating temperature -40°C ≤ TA ≤ + 125°C for automotive,
-40°C ≤ TA ≤ + 85°C for industrial and
0°C ≤ TA ≤ +70°C for commercial
Operating voltage VDD = 4.0V to 6.0V

Characteristic	Sym	Min	Typ (Note 1)	Max	Units	Conditions
Supply Voltage	VDD	4.0 4.5		6.0 5.5	V V	XT, RC and LP osc configuration HS osc configuration
RAM Data Retention Voltage (Note 2)	VDR		1.5		V	Device in SLEEP mode
VDD start voltage to guarantee power on reset	VPOR		VSS		V	See section 5.2 for details on power on reset
VDD rise rate to guarantee power on reset	SVDD	0.05*			V/ms	See section 5.2 for details on power on reset
Supply Current (Note 3)	IDD		1.8 35 5	4.5 70 10	mA µA mA	FOSC = 4 MHz, VDD = 5.5V (Note 5) FOSC = 32 KHz, VDD = 4.0V, WDT disabled, LP osc configuration FOSC = 10 MHz, VDD = 5.5V, HS osc configuration (PIC16C84-10 only)
Power Down Current (Note 4)	IPD		7 1.0 1.0 1.0	28 14 16 TBD	µA 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 VDD = 4.0V, WDT disabled, -40°C to +125°C

* These parameters are characterized but not tested.

Note 1: Data in the column labeled "Typical" is based on characterization results at 25°C. This data is for design guidance only and is not tested for, or guaranteed by Microchip Technology.

Note 2: This is the limit to which VDD can be lowered in SLEEP mode without losing RAM data.

Note 3: 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, RT = VDD, $\overline{MCLR}$ = VDD; WDT enabled/disabled as specified.

Note 4: 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-impedence state and tied to VDD and VSS.

Note 5: For RC osc configuration, current through Rext (external pull-up resistor) is not included. The current through the resistor can be estimated by the formula Ir = VDD/2Rext (mA) with Rext in kOhm.

7.2 DC CHARACTERISTICS: PIC16LC84-04 (COMMERCIAL, INDUSTRIAL, AUTOMOTIVE)

DC CHARACTERISTICS						Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ + 125°C for automotive, -40°C ≤ TA ≤ + 85°C for industrial and 0°C ≤ TA ≤ +70°C for commercial Operating voltage VDD = 2.0V to 6.0V
Characteristic	**Sym**	**Min**	**Typ (Note 1)**	**Max**	**Units**	**Conditions**
Supply Voltage	VDD	2.0 4.5		6.0 5.5	V V	XT, RC and LP osc configuration HS osc configuration
RAM Data Retention Voltage (Note 2)	VDR		1.5		V	Device in SLEEP mode
VDD start voltage to guarantee power on reset	VPOR		VSS		V	See section 5.2 for details on power on reset
VDD rise rate to guarantee power on reset	SVDD	0.05*			V/ms	See section 5.2 for details on power on reset
Supply Current (Note 3)	IDD		1.8 15	4.5 32	mA µA	FOSC = 4 MHz, VDD = 5.5V (Note 5) FOSC = 32 KHz, VDD = 2.0V, WDT disabled, LP osc configuration
Power Down Current (Note 4)	IPD		3 0.4 0.4 0.4	16 7 9 TBD	µA 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 VDD = 2.0V, WDT disabled, -40°C to +125°C

* These parameters are characterized but not tested.

Note 1: Data in the column labeled "Typical" is based on characterization results at 25°C. This data is for design guidance only and is not tested for, or guaranteed by Microchip Technology.

Note 2: This is the limit to which VDD can be lowered in SLEEP mode without losing RAM data.

Note 3: 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, RT = VDD, $\overline{MCLR}$ = VDD; WDT enabled/disabled as specified.

Note 4: 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-impedence state and tied to VDD and VSS.

Note 5: For RC osc configuration, current through Rext (external pull-up resistor) is not included. The current through the resistor can be estimated by the formula Ir = VDD/2Rext (mA) with Rext in kOhm.

Notes to Table 7.3

Note 1: Data in the column labeled "Typical" is based on characterization results at 25 ° C. This data is for design guidance only and is not tested for, or guaranteed by Microchip Technology.

Note 2 : In RC oscillator configuration, the OSC1 pin is a Schmitt trigger input. It is not recommended that the PIC16C84 be driven with external clock in RC mode.

Note 3 : 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.

Note 4 : Negative current is defined as coming out of the pin.

Note 5: The user may use better of the two specifications.

7.3 DC CHARACTERISTICS: PIC16C84-04 (COMMERCIAL, INDUSTRIAL, AUTOMOTIVE) PIC16C84-10 (COMMERCIAL, INDUSTRIAL, AUTOMOTIVE) PIC16LC84-04 (COMMERCIAL, INDUSTRIAL, AUTOMOTIVE)

DC CHARACTERISTICS

Standard Operating Conditions (unless otherwise stated)
Operating temperature -40°C ≤ TA ≤ +125°C for automotive,
-40 ≤ TA ≤ +85°C for industrial
and 0°C ≤ TA ≤ +70°C for commercial
Operating voltage VDD range as described in DC spec table 7.1

Characteristic	Sym	Min	Typ (Note 1)	Max	Units	Conditions
Input Low Voltage						
I/O ports	VIL	VSS		0.2 VDD	V	
MCLR, RTCC, OSC1 (in RC configuration)		VSS		0.2 VDD	V	Note 2
OSC1 (in XT, HS and LP configuration)		VSS		0.3 VDD	V	
Input High Voltage						
I/O ports	VIH	2.0		VDD	V	VDD ≤ 5.5V (note 5)
		0.7 VDD				For entire VDD range (note 5)
MCLR, RTCC, OSC1 (in RC configuration)		0.85 VDD		VDD	V	Note 2
OSC1 (XT, HS and LP configuration)		0.7 VDD		VDD	V	
Input Leakage Current (Notes 3, 4)						
I/O ports RA, RB	IIL			±1	µA	VSS ≤ VPIN ≤ VDD, Pin at hi-impedance
MCLR, RTCC				±5	µA	VSS ≤ VPIN ≤ VDD
OSC1				±5	µA	VSS ≤ VPIN ≤ VDD , XT, HS and LP osc configuration
PortB Weak Pull-up Current	IPU	50	100	150	µA	VPIN = VSS, VDD = 5.0V
Output Low Voltage						
I/O Ports	VOL			0.6	V	IOL = 8.5 mA, VDD = 4.5V, -40°C to +85°C
				0.6	V	IOL = 6.0 mA, VDD = 4.5V, -40°C to +125°C
OSC2/CLKOUT				0.6	V	IOL = 1.6 mA, VDD = 4.5V, -40°C to +85°C
(RC osc configuration)				0.6	V	IOL = 1.2 mA, VDD = 4.5V, -40°C to +125°C
Output High Voltage						
I/O Ports (Note 4)	VOH	VDD-0.7			V	IOH = -3.0 mA, VDD = 4.5V, -40°C to +85°C
		VDD-0.7			V	IOH = -2.5 mA, VDD = 4.5V, -40°C to +125°C
OSC2/CLKOUT		VDD-0.7			V	IOH = -1.3 mA, VDD = 4.5V, -40°C to +85°C
(RC osc configuration)		VDD-0.7			V	IOH = -1.0 mA, VDD = 4.5V, -40°C to +125°C
Data EEPROM						
Endurance	Ed	100,000	1,000,000		E/W cycles	
VDD for read/write	Vdrw		Full VDD range			
Erase/write cycle time	tdew			10	ms	
Program EEPROM Memory						See section 11 for more details on programming
Endurance	Ep	100	1000		E/W cycles	
Erase/write cycle time	tpew			10	ms	
VDD for read	Vpr		Full VDD range			
VDD for erase/write	Vpw	4.5		5.5	V	

Notes on previous page.

7.4 AC CHARACTERISTICS: PIC16C84-04 (COMMERCIAL, INDUSTRIAL, AUTOMOTIVE) PIC16C84-10 (COMMERCIAL, INDUSTRIAL, AUTOMOTIVE) PIC16LC84-04 (COMMERCIAL, INDUSTRIAL, AUTOMOTIVE)

AC CHARACTERISTICS

Standard Operating Conditions (unless otherwise stated)
Operating temperature -40°C ≤ TA ≤ +125°C for automotive,
-40 ≤ TA ≤ +85°C for industrial
and 0°C ≤ TA ≤ +70°C for commercial
Operating voltage VDD range as described in DC spec table 7.1

Characteristic	Sym	Min	Typ (Note 1)	Max	Units	Conditions
External CLOCKIN Frequency (Note 2)	FOSC	DC		4	MHz	XT, RC osc mode. VDD ≥ 3V
		DC		2	MHz	XT, RC osc mode. 2V ≥ VDD ≥ 3V
		DC		4	MHz	HS osc mode (PIC16C84-04, PIC16LC84-04)
		DC		10	MHz	HS osc mode (PIC16C84-10)
		DC		200	KHz	LP osc mode
Oscillator Frequency (Note 2)	FOSC	DC		4	MHz	RC osc mode
		0.1		4	MHz	XT osc mode
		1		4	MHz	HS osc mode (PIC16C84-04 PIC16LC84-04)
		1		10	MHz	HS osc mode (PIC16C84-10)
		DC		200	KHz	LP osc mode
Instruction Cycle Time (Note 2)	TCY	1.0	4/FOSC	DC	µs	
External Clock in Timing (Note 4)						
Clock in (OSC1) High or Low Time						
XT oscillator type	TCKHLXT	50			ns	
LP oscillator type	TCKHLLP	2			µs	
HS oscillator type	TCKHLHS	20			ns	
Clock in (OSC1) Rise or Fall Time						
XT oscillator type	TCKRFXT	25			ns	
LP oscillator type	TCKRFLP	50			ns	
HS oscillator type	TCKRFHS	25			ns	
OSC1 high to CLKOUT low	TOSH2CKL				ns	Note 6
OSC1 high to CLKOUT high	TOSH2CKH				ns	Note 6
CLKOUT output rise time	TCKR				ns	Note 6
CLKOUT output fall time	TCKF				ns	Note 6
RESET Timing						
MCLR Pulse Width (low)	TMCL	100			ns	
RTCC Input Timing, No Prescaler						
RTCC High Pulse Width	TRTH	0.5 TCY+ 20*			ns	Note 3
RTCC Low Pulse Width	TRTL	0.5 TCY+ 20*			ns	Note 3
RTCC Input Timing, With Prescaler						
RTCC High Pulse Width	TRTH	10			ns	Note 3
RTCC Low Pulse Width	TRTL	10			ns	Note 3
RTCC Period	TRTP	$\frac{T_{CY} + 40}{N}$ *			ns	Note 3. Where N = prescale value (2,4, ..., 256)
Watchdog Timer Timeout Period (No Prescaler)	TWDT	7*	18*	33*	ms	VDD = 5V, -40°C to +125°C
OST/PWRT Timings						
Oscillation Start-up Timer Period	TOST		1024 tosc		ms	tosc = OSC1 period
Power up timer period	TPWRT	28*	72*	132*	ms	VDD = 5V, -40°C to +125°C
I/O Timing						
I/O Pin Input Valid Before CLKOUT↑	TioV2ckH	0.25 TCY+ 30			ns	Note 6
I/O Pin Input Hold After CLKOUT↑	TckH2ioI	0			ns	Note 6
I/O Pin Output Valid After CLKOUT↓	TioV2ckL			40	ns	Note 6

* These parameters are characterized, but not tested.

(Cont. on next page)

7.4 AC CHARACTERISTICS: PIC16C84-04 (COMMERCIAL, INDUSTRIAL, AUTOMOTIVE) PIC16C84-10 (COMMERCIAL, INDUSTRIAL, AUTOMOTIVE) PIC16LC84-04 (COMMERCIAL, INDUSTRIAL, AUTOMOTIVE)

AC CHARACTERISTICS	Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +125°C for automotive, -40 ≤ TA ≤ +85°C for industrial and 0°C ≤ TA ≤ +70°C for commercial Operating voltage VDD range as described in DC spec table 7.1

Characteristic	Sym	Min	Typ (Note 1)	Max	Units	Conditions
I/O Timing (cont.)						
I/O Pin Input Valid Before OSC↑ (I/O Setup Time)	TioV2osH				ns	
OSC1↑ to I/O pin input invalid (I/O hold time)	TosH2ioL				ns	
OSC1↑ to I/O pin output valid	TosH2ioV				ns	
I/O pin output rise time	TioR				ns	
I/O pin output fall time	TioF				ns	
Interrupt Timing						
INT pin high or low time	TinP	20			ns	
RB <7:4> input change time for interrupt to be recognized	TrbP	20			ns	
Capacitive Loading Specs on Output Pins						
OSC2 pin	COSC2			15	pF	

* These parameters are characterized, but not tested.

NOTES TO TABLE 7.4

Note 1: Data in the column labeled "Typical" is based on characterization results at 25°C. This data is for design guidance only and is not tested for, or guaranteed by Microchip Technology.

Note 2: 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.

Note 3: For a detailed explanation of RTCC input clock requirements see section 6.4.1.

Note 4: Clock-in high-time is the duration for which clock input is at VIHOSC or higher. Clock-in low-time is the duration for which clock input is at VIHOSC or lower.

Note 5: All AC parameters are tested or characterized with these capacitive loads.

Note 6: CLKOUT is available only in RC oscillator mode.

7.5 Electrical Structure of Pins

FIGURE 7.5.1 - ELECTRICAL STRUCTURE OF I/O PINS (RA, RB)

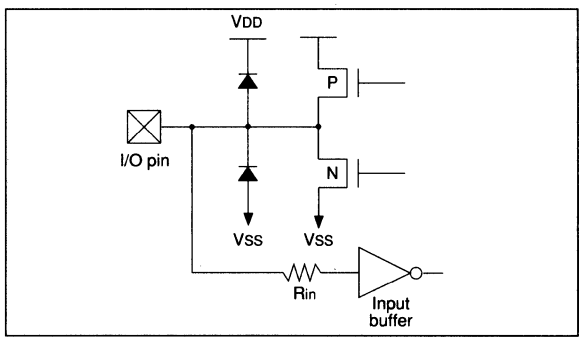
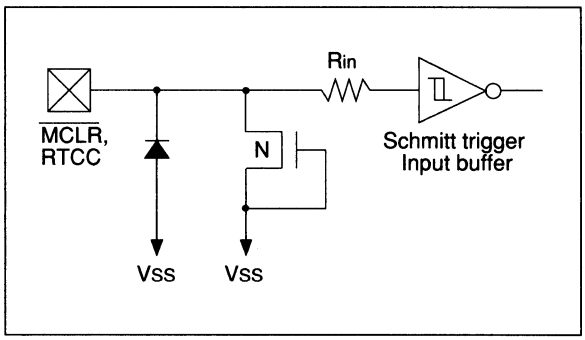


FIGURE 7.5.2 - ELECTRICAL STRUCTURE OF MCLR AND RTCC PINS



Notes to figures 7.5.1 and 7.5.2: The diodes and the grounded gate (or output driver) NMOS device are carefully designed to protect against ESD (Electrostatic discharge) and EOS (Electrical overstress). Rin is a small resistance to further protect the input buffer from ESD.

8.0 TIMING DIAGRAMS

FIGURE 8.0.1 - RTCC TIMING

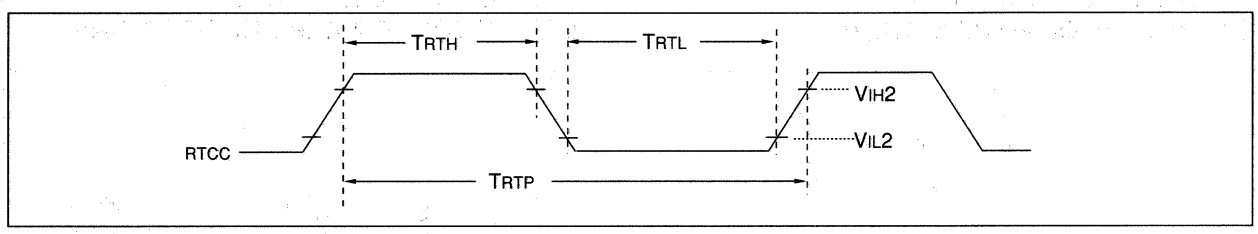
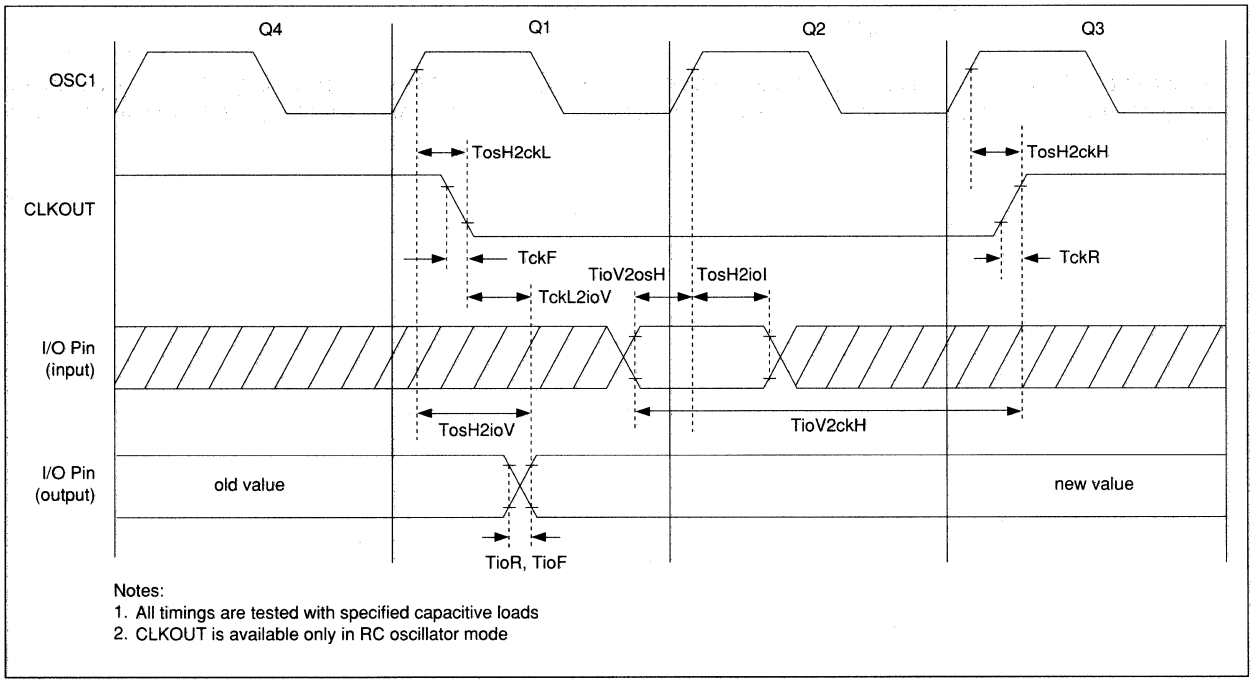


FIGURE 8.0.2 - INPUT/OUTPUT TIMING FOR I/O PORTS



Notes:
1. All timings are tested with specified capacitive loads
2. CLKOUT is available only in RC oscillator mode

9.0 DC & AC CHARACTERISTICS GRAPHS/TABLES:

The graphs and tables provided in this section are for design guidance and are not tested or guaranteed. **In some graphs or tables the data presented are outside specified operating range (e.g. outside specified VDD range). This is for information only and devices are 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 9.0.1 - TYPICAL RC OSCILLATOR FREQUENCY vs. TEMPERATURE

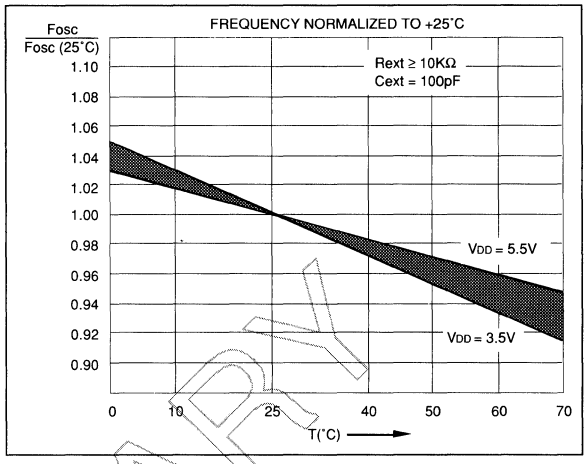


FIGURE 9.0.2 - TYPICAL RC OSCILLATOR FREQUENCY vs VDD

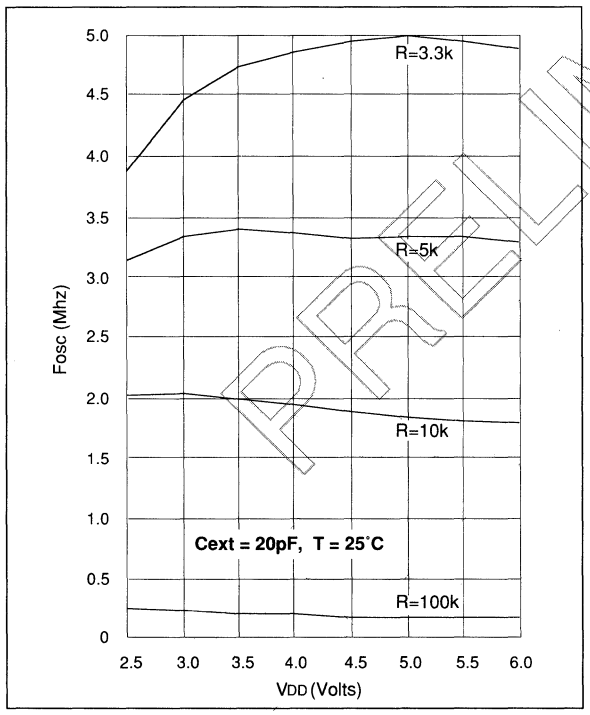


FIGURE 9.0.3 - TYPICAL RC OSCILLATOR FREQUENCY vs VDD

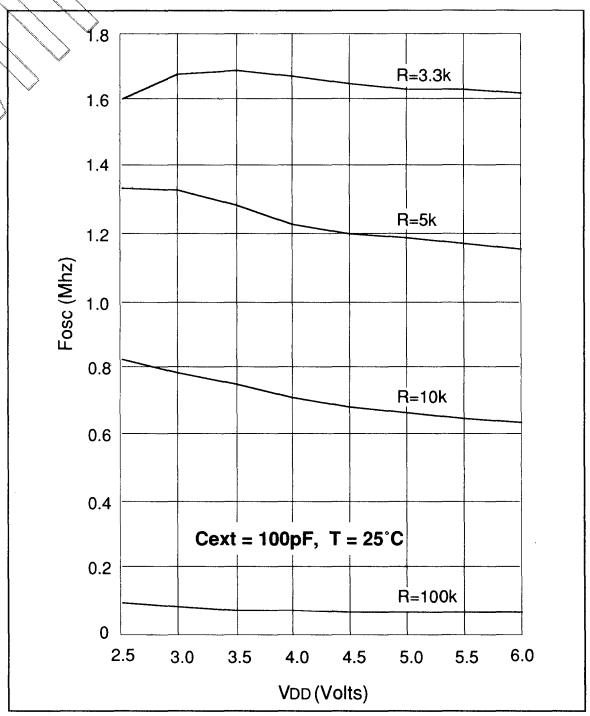


FIGURE 9.0.4 - TYPICAL RC OSCILLATOR FREQUENCY vs VDD

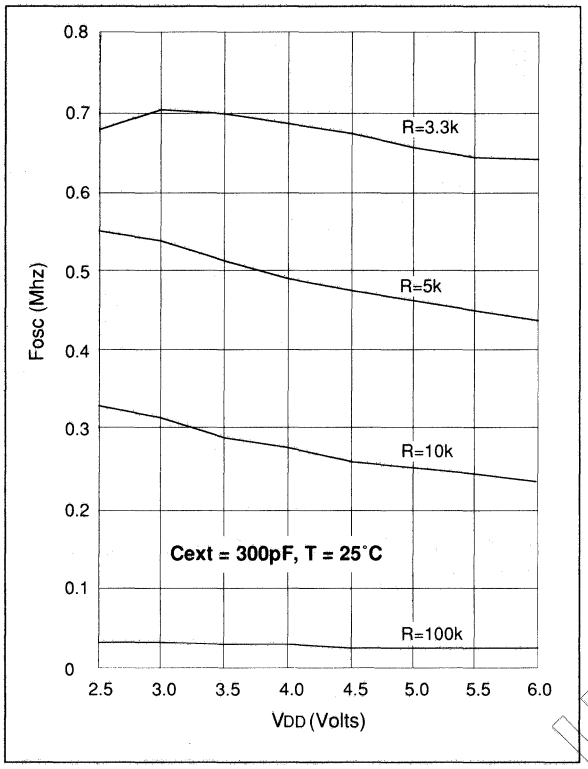


TABLE 9.0.1 - RC OSCILLATOR FREQUENCIES

Cext	Rext	Average Fosc @ 5V, 25°C	
20pf	3.3k	4.71 MHz	± 28%
	5k	3.31 MHz	± 25%
	10k	1.91 MHz	± 24%
	100k	207.76 KHz	± 39%
100pf	3.3k	1.65 MHz	± 18%
	5k	1.23 MHz	± 21%
	10k	711.54 KHz	± 18%
	100k	75.62 KHz	± 28%
300pf	3.3k	672.78 KHz	± 14%
	5k	489.49 KHz	± 13%
	10k	275.73 KHz	± 13%
	100k	28.12 KHz	± 23%

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.

FIGURE 9.0.5 - TYPICAL Ipd vs VDD WATCHDOG DISABLED 25°C

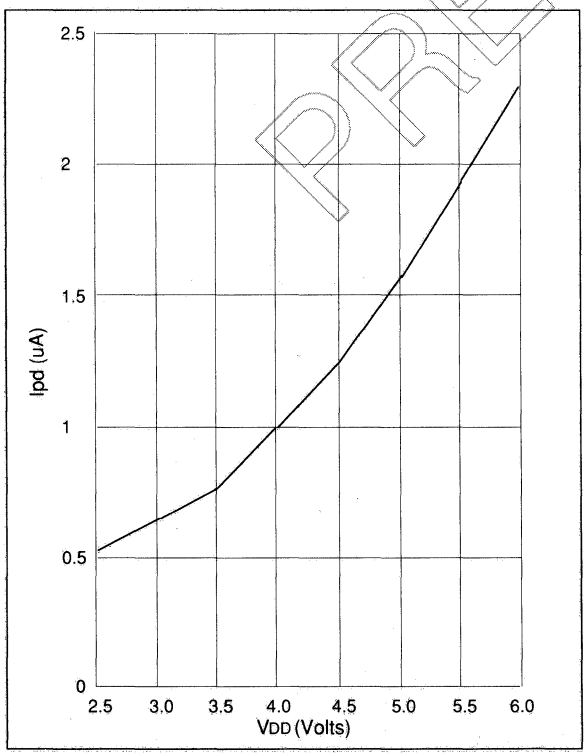


FIGURE 9.0.6 - TYPICAL Ipd vs VDD WATCHDOG ENABLED 25°C

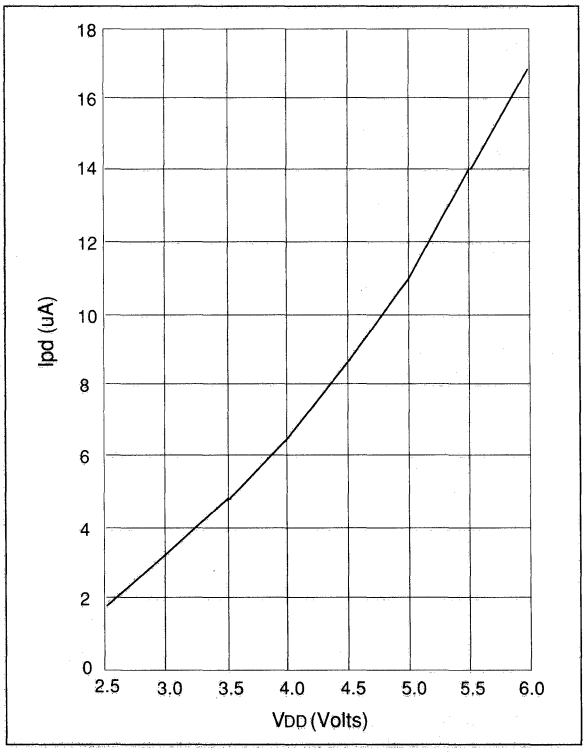


FIGURE 9.0.7 - MAXIMUM Ipd vs VDD WATCHDOG DISABLED

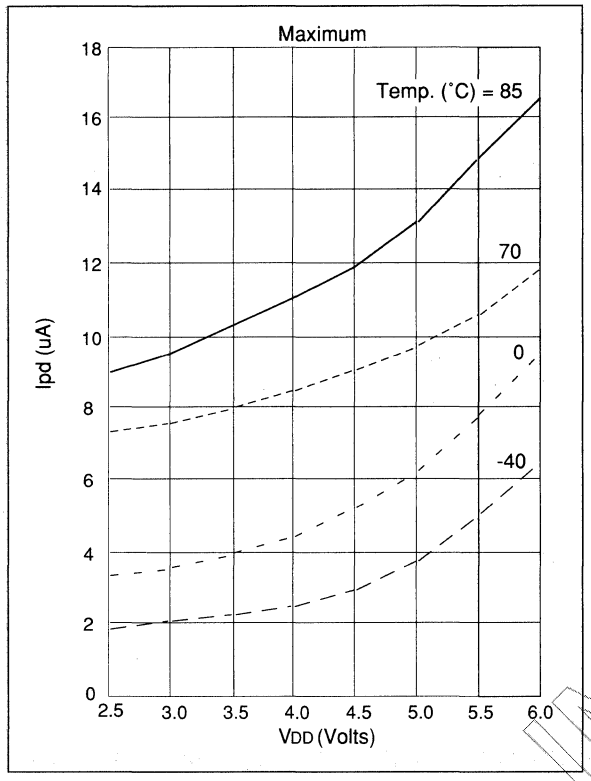
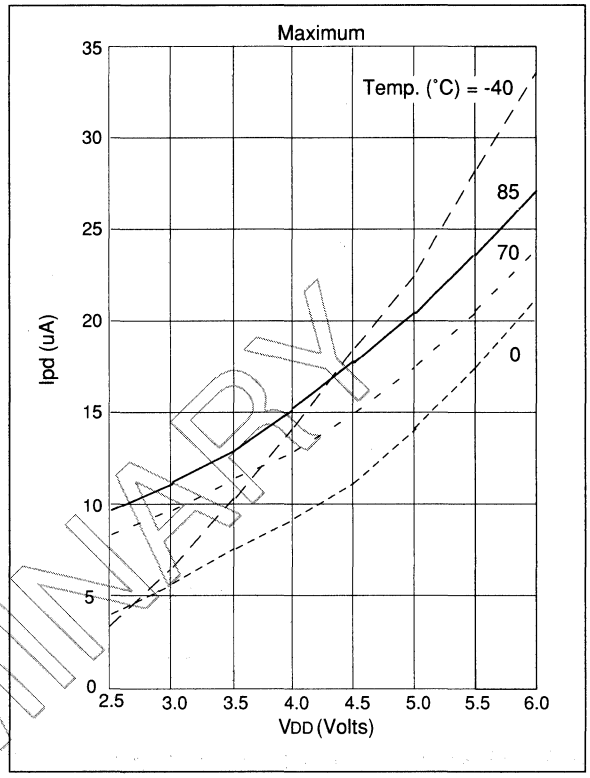


FIGURE 9.0.8 - MAXIMUM Ipd vs VDD WATCHDOG ENABLED*



* IPD, 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.

FIGURE 9.0.9 - VTH (INPUT THRESHOLD VOLTAGE) OF I/O PINS vs VDD

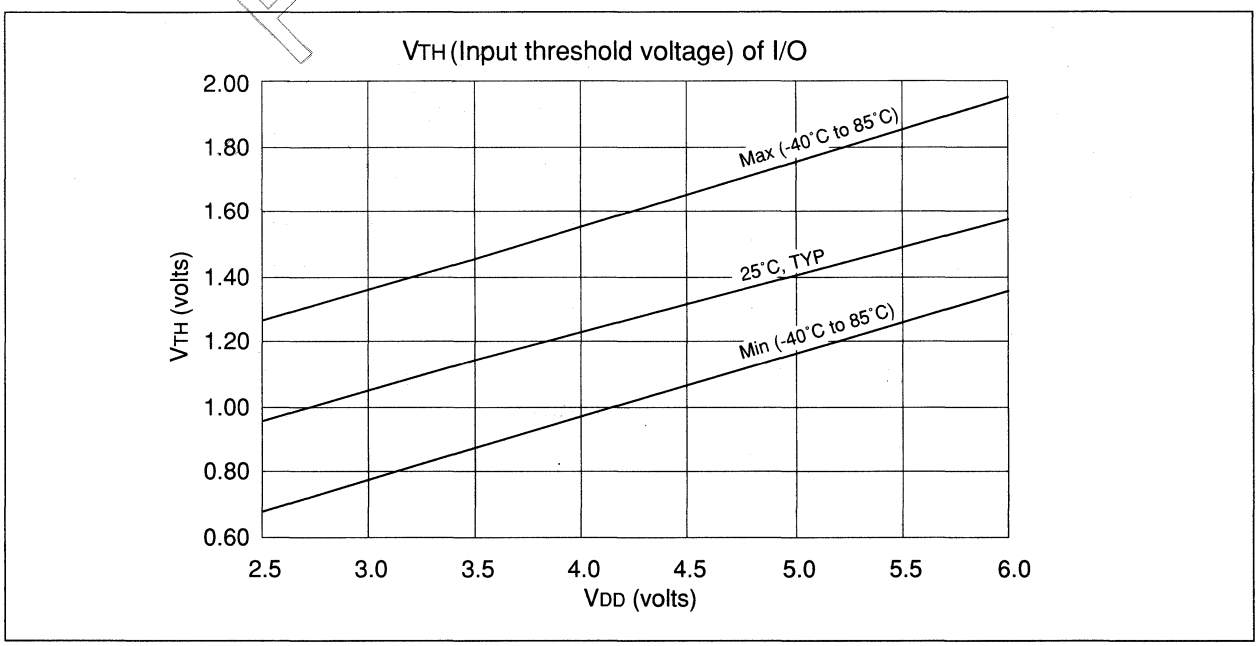
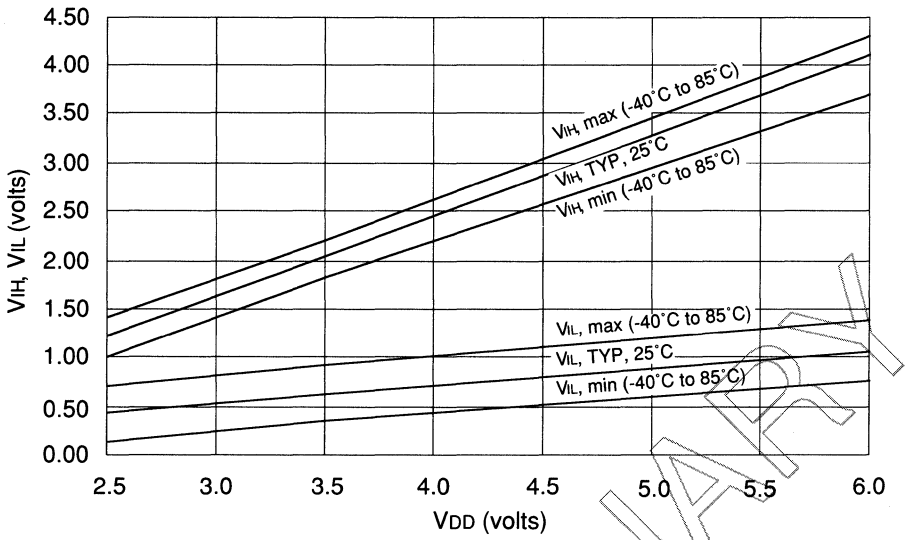


FIGURE 9.0.10 - V_{IH}, V_{IL} OF $\overline{MCLR}$, RTCC AND OSC1 (IN RC MODE) vs V_{DD}



Note: These input pins have Schmitt trigger input buffer.

FIGURE 9.0.11 - V_{TH} (INPUT THRESHOLD VOLTAGE) OF OSC1 INPUT (IN XT, HS, AND LP MODES) vs V_{DD}

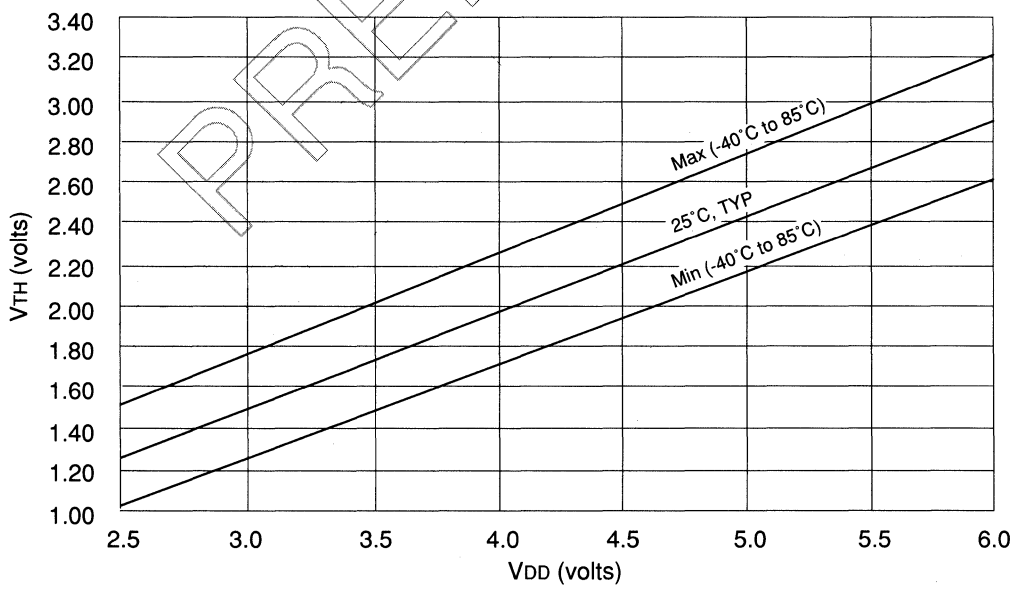


FIGURE 9.0.12 - TYPICAL IDD vs FREQ (EXT CLOCK, 25°C)

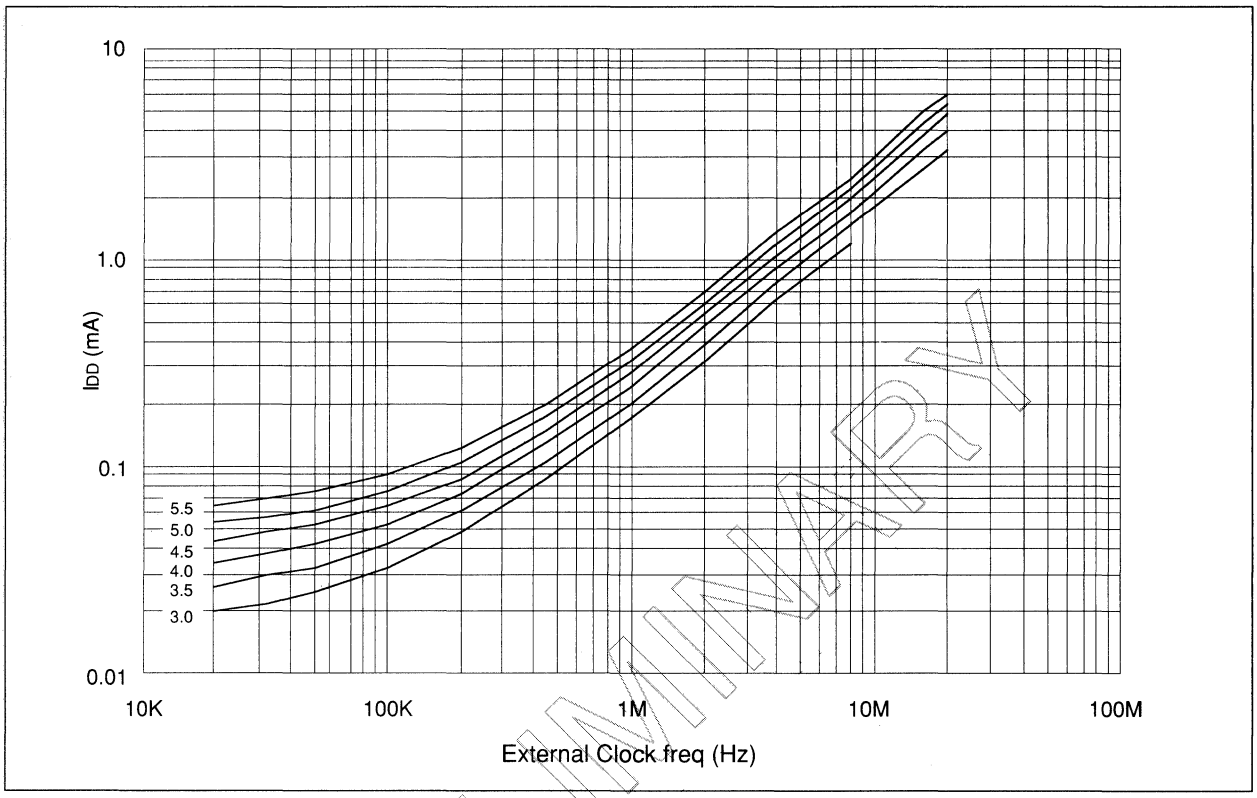


FIGURE 9.0.13 - MAXIMUM IDD vs FREQ (EXT CLOCK, -40° to +85°C)

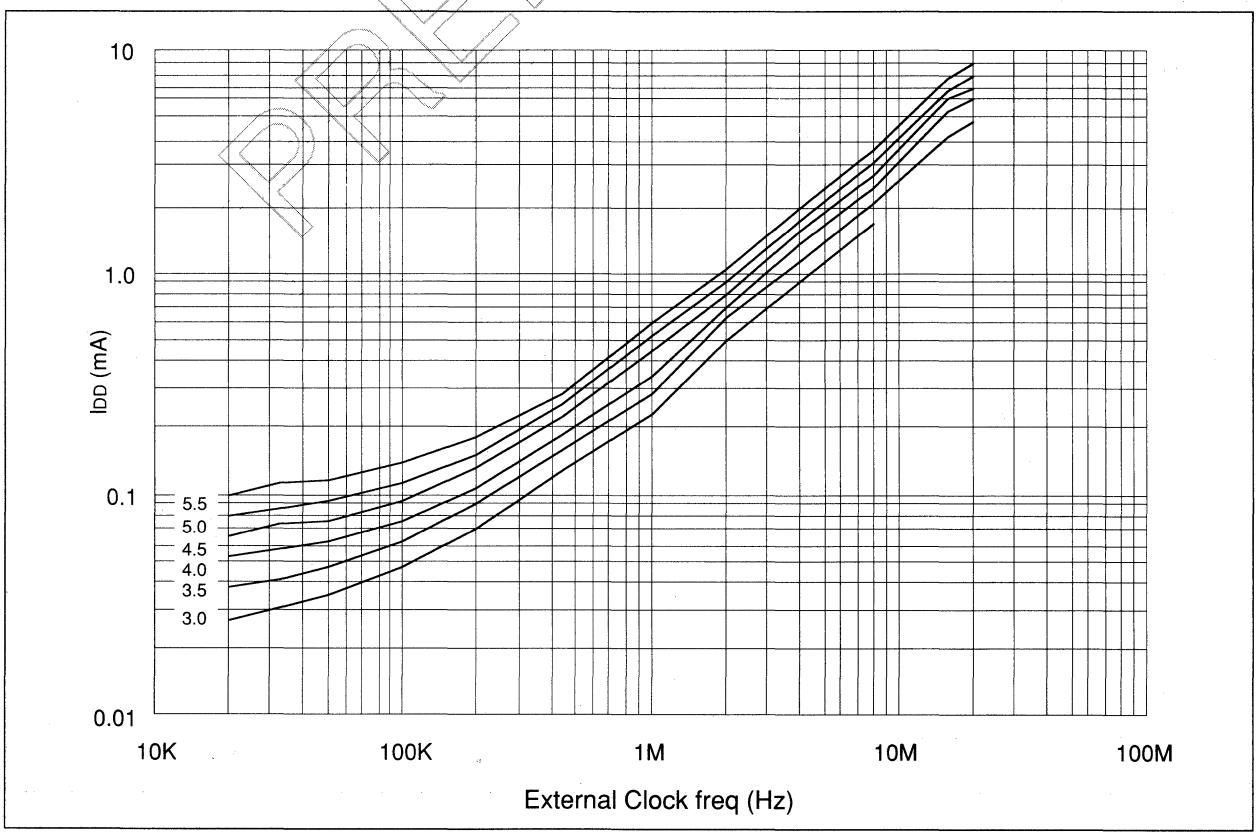


FIGURE 9.0.14 - WDT Timer Time-out Period vs VDD

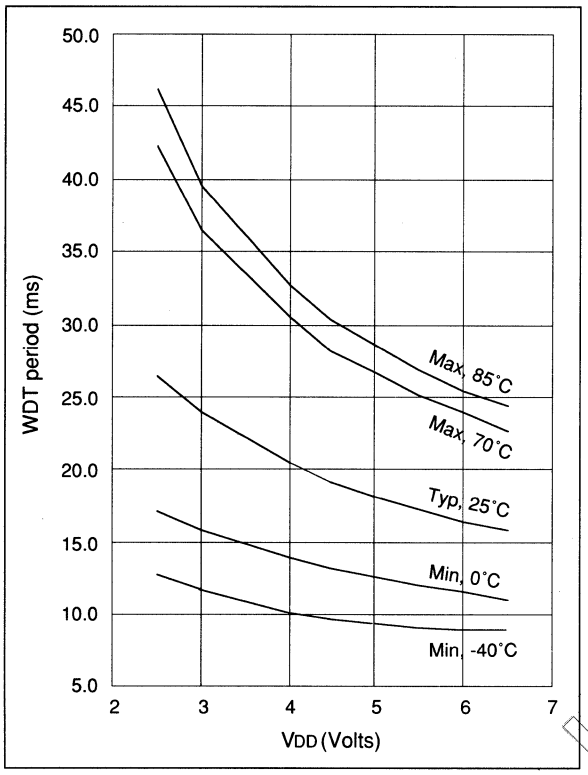


FIGURE 9.0.15 - Transconductance (gm) of HS Oscillator vs VDD

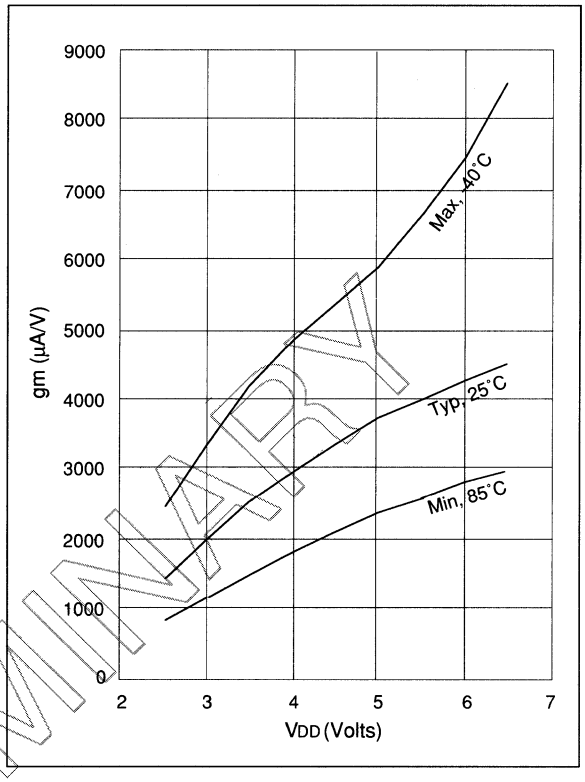


FIGURE 9.0.16 - Transconductance (gm) of LP Oscillator vs VDD

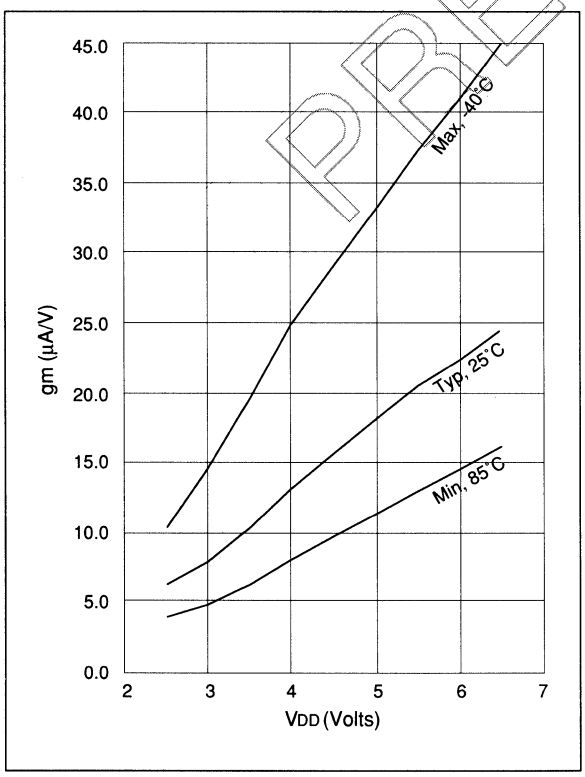


FIGURE 9.0.17 - Transconductance (gm) of XT Oscillator vs VDD

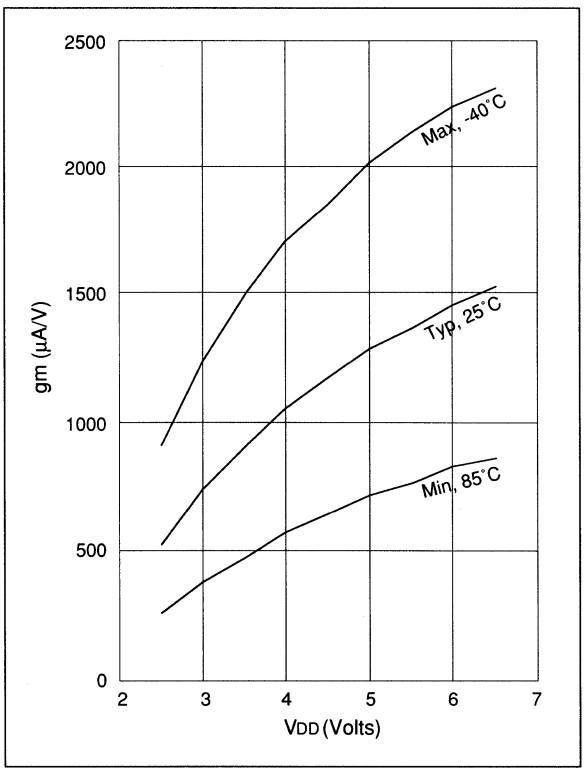


FIGURE 9.0.18 - IOH vs VOH, VDD = 3V

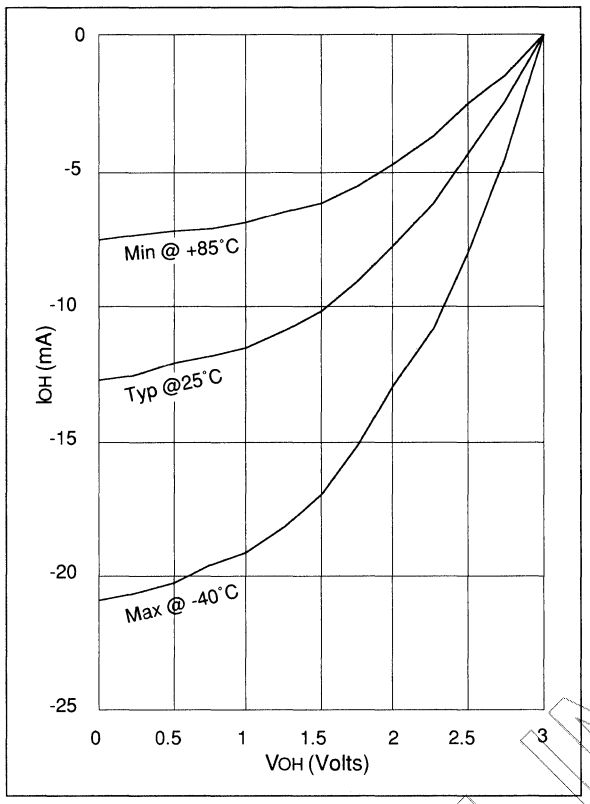


FIGURE 9.0.19 - IOH vs VOH, VDD = 5V

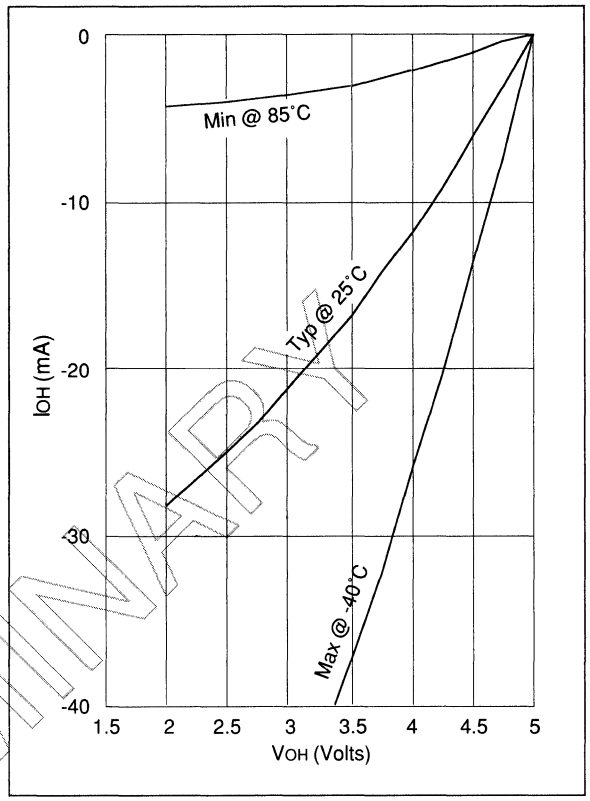


FIGURE 9.0.20 - IOL vs VOL, VDD = 3V

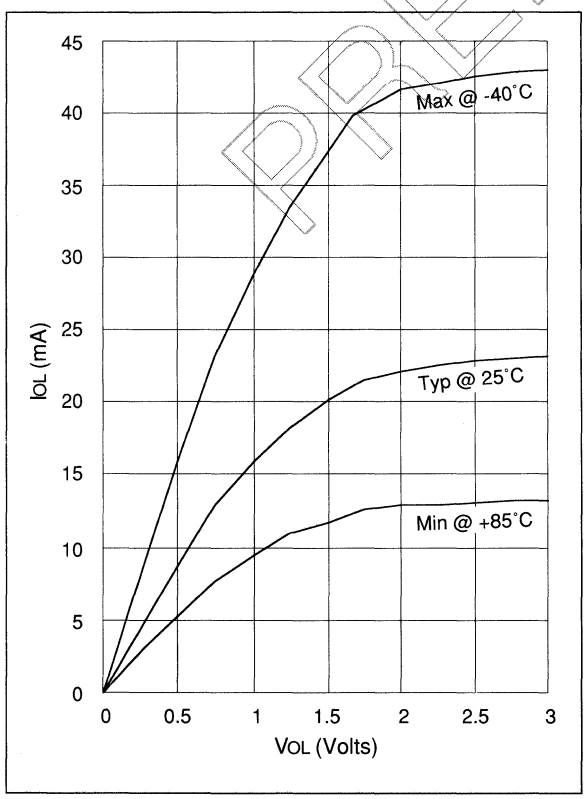


FIGURE 9.0.21 - IOL vs VOL, VDD = 5V

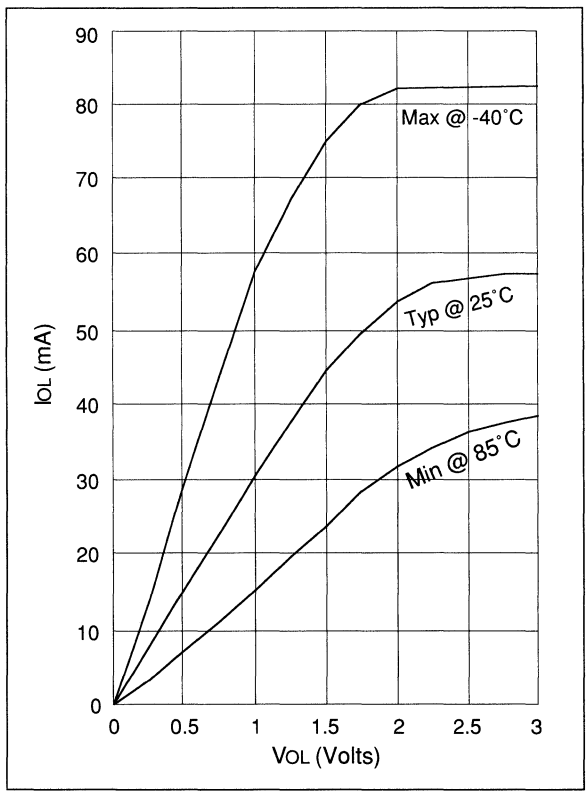


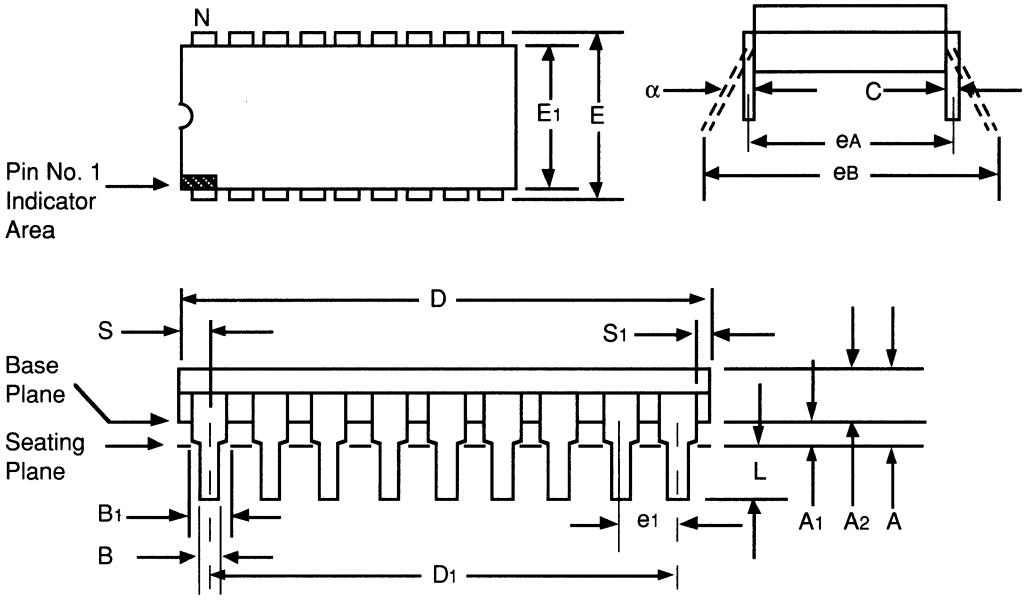
TABLE 9.0.2 - INPUT CAPACITANCE *

Pin Name	Typical Capacitance (pF)	
	18L PDIP	18L SOIC
RA port	5.0	4.3
RB port	5.0	4.3
MCLR	17.0	17.0
OSC1	4.0	3.5
OSC2/CLKOUT	4.3	3.5
RTCC	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.

10.0 PACKAGING DIAGRAMS AND DIMENSIONS

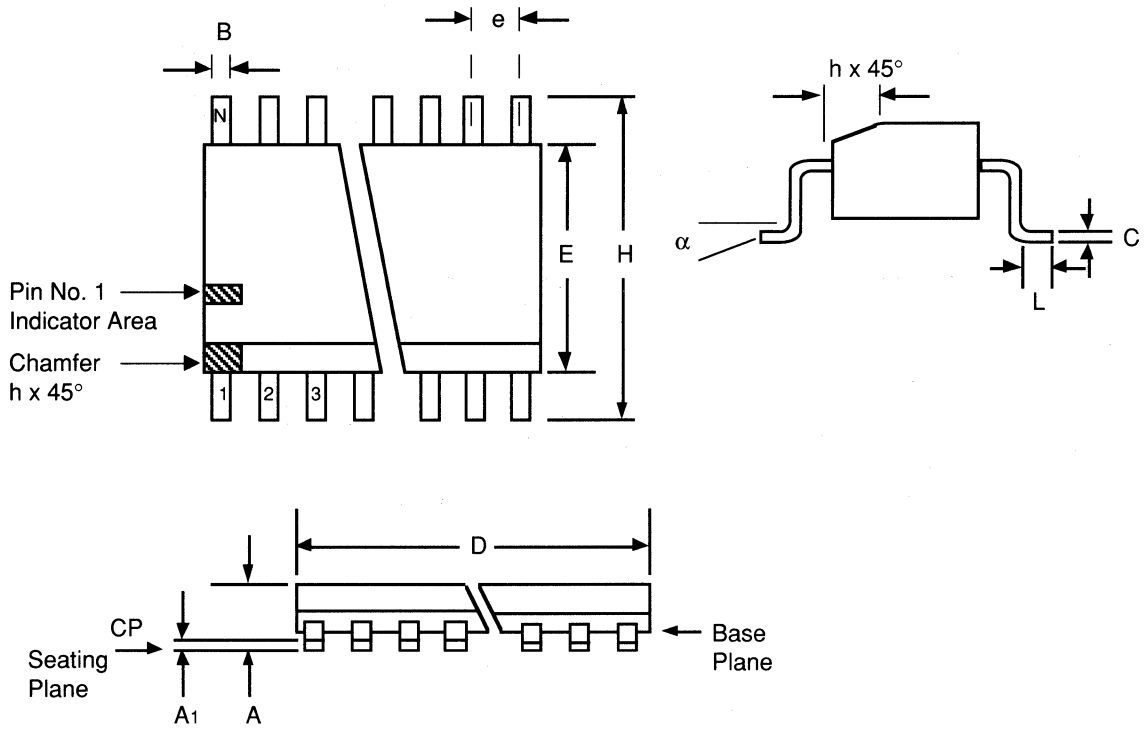
10.1 18-LEAD PLASTIC DUAL IN-LINE (.300 mil)



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	
A_1	0.381	–		0.015	–	
A_2	3.048	3.810		0.120	0.150	
B	0.356	0.559		0.014	0.022	
B_1	1.524	1.524	**Typical**	0.060	0.060	**Typical**
C	0.203	0.381	**Typical**	0.008	0.015	**Typical**
D	22.479	23.495		0.885	0.925	
D_1	20.320	20.32	**Reference**	0.800	0.800	**Reference**
E	7.620	8.255		0.300	0.325	
E_1	6.096	7.112		0.240	0.280	
e_1	2.489	2.591	**Typical**	0.098	0.102	**Typical**
e_A	7.620	7.620	**Reference**	0.300	0.300	**Reference**
e_B	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	–	
S_1	0.127	–		0.005	–	

PACKAGING DIAGRAMS AND DIMENSIONS (CONT.)

10.2 18-LEAD PLASTIC SURFACE MOUNT (SOIC - WIDE, 300 mil BODY)

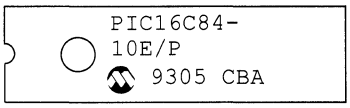


Package Group: Plastic SOIC (SO)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
α	0°	8°		0°	8°	
A	2.3622	2.6416		0.093	0.104	
A_1	0.1016	0.2997		0.004	0.0118	
B	0.3556	0.4826		0.014	0.019	
C	0.2413	0.3175		0.0095	0.0125	
D	11.3538	11.7348		0.447	0.462	
E	7.4168	7.5946		0.292	0.299	
e	1.270	1.270	**Typical**	0.050	0.050	**Typical**
H	10.0076	10.6426		0.394	0.419	
h	0.381	0.762		0.015	0.030	
L	0.4064	1.143		0.016	0.045	
N	18	18		18	18	
CP	–	0.1016		–	0.004	

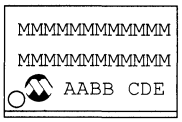
10.3 PACKAGE MARKING INFORMATION

18L PDIP

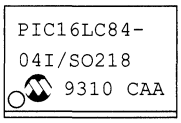
Example



18L SOIC



Example



Legend:	MM...M	Microchip part number information
	XX...X	Customer specific information*
	AA	Year code (last 2 digits of calendar year)
	BB	Week code (week of January 1 is week '01')
	C	Facility code of the plant at which wafer is manufactured. C = Chandler, Arizona, U.S.A.
	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 can not 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.

11.0 PROGRAMMING THE PIC16C84

The PIC16C84 is programmed using one of two methods, serial or parallel. The serial mode will allow the PIC16C84 to be programmed while in the users system using only five pins: VDD, VSS, MCLR/VPP, RB6 and RB7. This allows for increased design flexability. The parallel mode will provide faster programming as the data is loaded into the PIC16C84 with a greater throughput. Either mode may be selected at the start of the programming process. The parallel mode is intended for programmers. In either mode, both program and data memory can be programmed. You can get complete programming information in the PIC16C84 programming specification (DS30189).

12.0 DEVELOPMENT SUPPORT

12.1 PICMASTER™: High Performance Universal In-Circuit Emulator System

The PICMASTER Universal In-Circuit Emulator System is intended to provide the product development engineer with a complete microcontroller design tool set for all microcontrollers in the PIC16CXX and PIC17CXX families. This system currently supports the PIC16CR54, PIC16C54, PIC16C55, PIC16C56, PIC16C57, PIC17C42, PIC16C71 and PIC16C84.

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 PIC16CXX and PIC17CXX microcontrollers.

The Emulator System is designed to operate on low-cost PC compatible machines ranging from 80286-AT class ISA-bus systems through the new 80486 EISA-bus machines. The development software runs in the Microsoft Windows® 3.1 environment, allowing the operator access to a wide range of supporting software and accessories.

Provided with the PICMASTER System is a high performance real-time In-Circuit Emulator, a programmer unit and a macro assembler program.

Coupled with the user's choice of text editor, the system is ready for development of products containing any of Microchip's microcontroller products.

A "Quick Start" PIC Product Sample Pak containing user programmable parts is included for additional convenience.

Microchip provides additional customer support to developers through an Electronic Bulletin Board System (EBBS). Customers have access to the latest updates in software as well as application source code examples. Consult your local sales representative for information on accessing the BBS system.

12.1.1 HOST SYSTEM REQUIREMENTS

The PICMASTER has been designed as a real-time emulation system with advanced features generally found on more expensive development tools. The AT platform and Windows 3.X environment was chosen to best make these features available to you the end user. To properly take advantages of these features, PICMASTER requires installation on a system having the following minimum configuration:

- PC AT compatible machine: 80286, 386SX, 386DX, or 80486 with ISA or EISA Bus.
- EGA, VGA, 8514/A, Hercules graphic card (EGA or higher recommended).
- MSDOS / PCDOS version 3.1 or greater.
- Microsoft Windows® version 3.0 or greater operating in either standard or 386 enhanced mode).
- 1 Mbyte RAM (2 Mbytes recommended).
- One 5.25" floppy disk drive.
- Approximately 10 Mbytes of hard disk (1 Mbyte required for PICMASTER, remainder for Windows 3.X system).
- One 8-bit PC AT (ISA) I/O expansion slot (half size)
- Microsoft® mouse or compatible (highly recommended).

12.1.2 EMULATOR SYSTEM COMPONENTS

The PICMASTER Emulator Universal System consists primarily of 4 major components:

- **Host-Interface Card:** The PC Host Interface Card connects the emulator system to an IBM PC compatible system. This high-speed parallel interface requires a single half-size standard AT / ISA slot in the host system. A 37-conductor cable connects the interface card to the external Emulator Control Pod.
- **Emulator Control Pod:** The Emulator Control Pod contains all emulation and control logic common to all microcontroller devices. Emulation memory, trace memory, event and cycle timers, and trace/breakpoint logic are contained here. The Pod controls and interfaces to an interchangeable target-specific emulator probe via a 14" precision ribbon cable.
- **Target-specific Emulator Probe:** A probe specific to microcontroller family to be emulated is installed on the ribbon cable coming from the control pod. This probe configures the universal system for emulation of a specific microcontroller.
- **PC Host Emulation Control Software:** Host software necessary to control and provide a working user interface is the last major component of the system. The emulation software runs in the Windows 3.X environment, and provides the user with full display, alter, and control of the system under emulation. The Control Software is also universal to all microcontroller families.

The Windows 3.X System is a multitasking operating system which will 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. Dynamic Data Exchange (DDE), a feature of Windows 3.X, will be available in this and future versions of the software. 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 3.X, two or more PICMASTER emulators can run simultaneously on the same PC making development of multi-microcontroller systems possible (e.g., a system containing a PIC16Cxx processor and a PIC17Cxx processor).

FIGURE 12.1.1 - PICMASTER

FIGURE 12.1.2 - PICMASTER SYSTEM CONFIGURATION

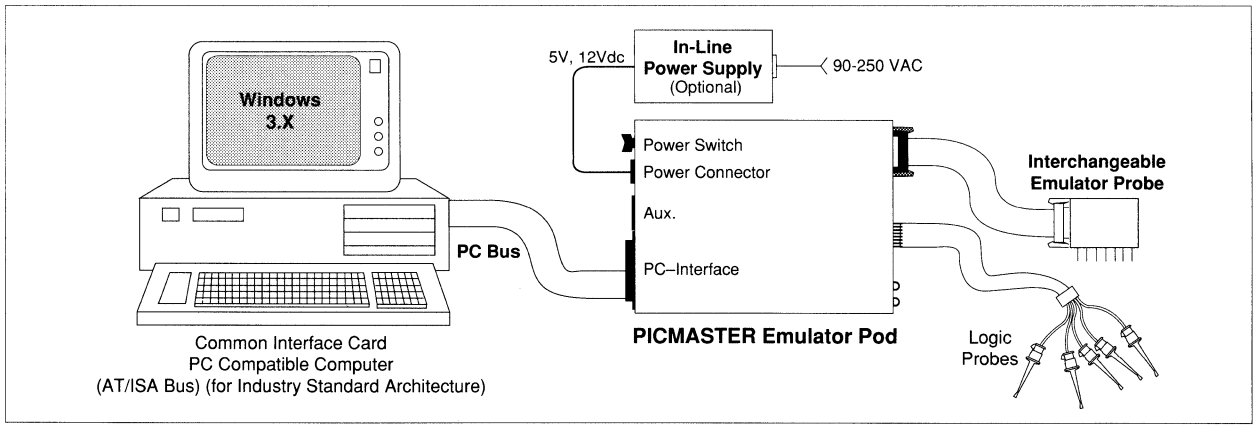
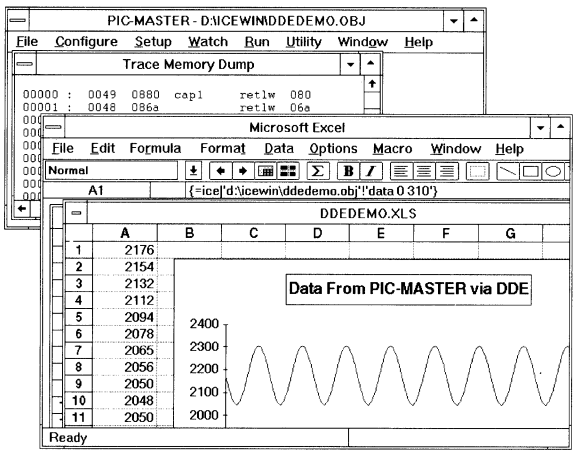


FIGURE 12.1.3 - PICMASTER TYPICAL SCREEN



12.2 PICALC Cross-Assembler

The PIC Cross Assembler PICALC is a PC hosted software development tool supporting the PIC16C5X and PIC16CXX series microcontrollers. PICALC offers a full featured **Macro and Conditional assembly** capability. PICALC supports "include" file facility for modular code development. It generates various object code formats including several Hex formats to support Microchip's proprietary development tools as well as third party tools. Also supports Hex (default), Decimal and Octal Source and listing formats. An assembler users manual is available for detailed support.

12.3 PRO MASTER™

The PRO MASTER programmer is a production quality programmer capable of operating in stand alone mode as well as PC-hosted mode.

The PRO MASTER has programmable VDD and VPP supplies which allows it to verify the PIC at VDD min and VDD max for maximum reliability . It has an LCD display for displaying error messages, keys to enter commands and a modular detachable socket assembly to support various package types. In stand alone mode the PRO MASTER can read, verify or program a part. It can also set fuse configuration and code-protect in this mode. It's EEPROM memory holds data and parametric information even when powered down. It is ideal for low to moderate volume production.

In PC-hosted mode, the PRO MASTER connects to the PC via one of the COM (RS232) ports. A 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 fuse configuration and part type, easy selection of VDD min, VDD max and VPP 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, 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 MASTER has a modular "programming socket module". Different socket modules are required for different processor types and/or package types. It is planned that the PRO MASTER will support all current and future PIC16CXX and PIC17CXX processors. Currently socket modules are available for the PIC16C54, PIC16C55, PIC16C56, PIC16C57, PIC17C42, PIC16C71 and the PIC16C84.

APPENDIX A

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

1. Instruction word length is increased to 14 bit. This allows larger page sizes both in program memory (2K now as opposed to 512 before) and register file (128 bytes now versus 32 bytes before).
2. A PC high latch register (PCLATH) is added to handle program memory paging. PA2, PA1, PA0 bits are removed from status register.
3. Data memory paging is redefined slightly. 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 addressible.
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 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 feature.
13. RTCC pin is also a port pin (RA4) now.
14. Location 07h (PortC) is unimplemented and not a general purpose register.
15. FSR is made a full eight bit register.
16. "In system programming" is made possible. The user can program the PIC16C84 using only five pins: V_{DD}, V_{SS}, $\overline{MCLR}/V_{PP}$, RB6 (clock) and RB7 (data in/out).

APPENDIX B

To convert code written for PIC16C5X to PIC16C84, 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 to reallocate them.
4. Verify all writes to STATUS, OPTION, and FSR registers since these have changed.
5. Change reset vector to 0000h.
6. Note that location 07h is an unimplemented data memory location.

Index

Products supported by a preliminary Data Sheet may possibly have an errata sheet describing minor operational differences and recommended workarounds. To determine if an errata sheet exists for a particular device, please contact one of the following:

1. Your local Microchip sales office (see last page of Data Sheet for listing)
2. The Microchip Corporate Literature Center U.S. FAX: (602) 786-7277
3. The Microchip's PIC Bulletin Board, via your local Compuserve number (Compuserve membership NOT required). See Microchip's Bulletin Board User's document (literature # DS30128B).

Please specify which PIC device, revision of silicon and Data Sheet (include Literature #) you are using.

SALES AND SUPPORT

To order or to obtain information, e.g., on pricing or delivery, please use the listed part numbers, and refer to the factory or the listed sales offices.

PART NO. - XX X /XX XXX

Pattern:	3-Digit Pattern Code for QTP (blank otherwise)	
Package:	P = PDIP	
	SO = SOIC (Gull Wing, 300 mil body)	
Temperature Range:	- = 0° C to +70°C (T for tape/reel)	
	I = -40° C to +85°C (S for tape/reel)	
	E = -40°C to +125°C	
Frequency Range:	04 = 4 MHz	
	10 = 10 MHz	
Device:	PIC16C84 : VDD range 4.0V to 6.0V	
	PIC16LC84 : VDD range 3.0V to 6.0V	

Examples:
a) PIC16C84 - 04/P 301 = Commercial temp., PDIP package, 4 MHz, normal VDD limits, QTP pattern # 301
b) PIC16LC84 - 04I/SO = Industrial temp., SOIC package, 4 MHz, extended VDD limits
c) PIC16C84 - 10E/P = Automotive temp., PDIP package, 10 MHz, normal VDD limits

Microchip Technology Incorporated **Printed in USA © 9301**

JAPAN

Microchip Technology International Inc.
Shinyokohama Gotoh Bldg. 8F, 3-22-4
Shinyokohama, Kohoku-Ku, Yokohama-Shi
Kanagawa 222 Japan
Tel: 45/471-6166 Fax: 45/471-6122

EUROPE

United Kingdom

Arizona Microchip Technology LTD.
Unit 3, Meadow Bank, Furlong Road
Bourne End, Bucks SL8 5AJ
Tel: 62-885-1077 Fax: 62-885-0178

Germany

Arizona Microchip Technology GMBH
Alte Landstrasse 12-14
D-8012 Ottobrunn, Germany
Tel: 089 609 6072 Fax: 089 609 1997

France

Arizona Microchip Technology SARL
2, Rue Du Buisson aux Fraises
F-91300 Massy, France
Tel: 1 69 30 90 90 Fax: 1 69 30 90 79

UNITED STATES

Corporate Office

Microchip Technology Inc.
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 602 786-7200 Fax: 602 899-9210

North-East Region

Microchip Technology Inc.
Five The Mountain Road, Suite 120
Framingham, MA 01701
Tel: 508 820-3334 Fax: 508 820-4326

Mid-Atlantic Region

Microchip Technology Inc.
150 Motor Parkway, Suite 416
Hauppauge, NY 11788
Tel: 516 273-5305 Fax: 516 273-5335

Microchip®

South-East Region

Microchip Technology Inc.
1521 Johnson Ferry Road NE, Suite 170
Marietta, GA 30062
Tel: 404 509-8800 Fax: 404 509-8600

North-Central Region

Microchip Technology Inc.
665 Tollgate Road, Unit C
Elgin, IL 60123-9312
Tel: 708 741-0171 Fax: 708 741-0638

South-Central Region

Microchip Technology Inc.
17480 N Dallas Parkway, Suite 114
Dallas, TX 75287
Tel: 214 733-0391 Fax: 214 250-4631

North-West Region

Microchip Technology Inc.
2107 N First Street, Suite 410
San Jose, CA 95131
Tel: 408 436-7950 Fax: 408 436-7955

South-West Region

Microchip Technology Inc.
1411 W 190th Street, Suite 230
Gardena, CA 90248-4307
Tel: 310 323-1888 Fax: 310 323-1424

Preliminary