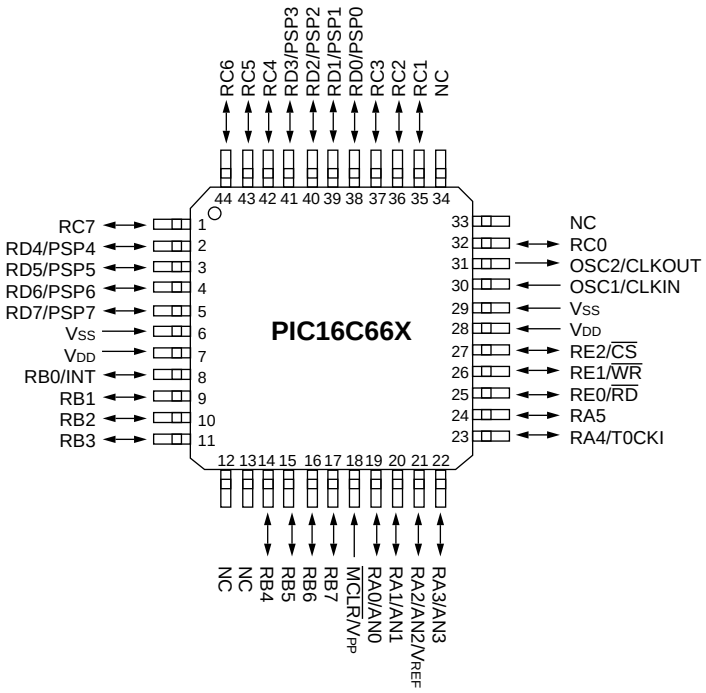
Pin Diagrams (Cont.'d)

TQFP



PLCC

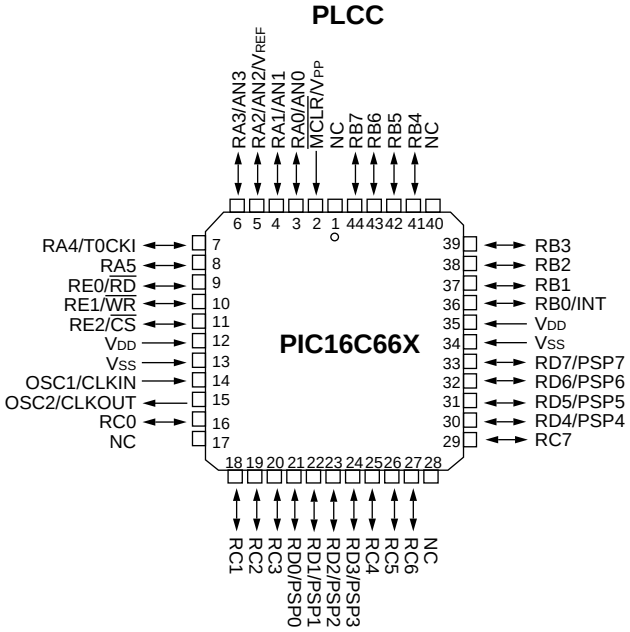


Table of Contents

To Our Valued Customers

We constantly strive to improve the quality of all our products and documentation. We have spent an exceptional amount of time to ensure that these documents are correct. However, we realize that we may have missed a few things. If you find any information that is missing or appears in error, please use the reader response form in the back of this data sheet to inform us. We appreciate your assistance in making this a better document.

NOTES:

1.0 GENERAL DESCRIPTION

PIC16C64X & PIC16C66X devices are 28-pin and 40-pin EPROM-based members of the versatile PIC16CXXX family of low-cost, high-performance, CMOS, fully-static, 8-bit microcontrollers.

All PIC16/17 microcontrollers employ an advanced RISC architecture. The PIC16CXXX family has enhanced core features, eight-level deep stack, and multiple internal and external interrupt sources. The separate instruction and data buses of the Harvard architecture allow a 14-bit wide instruction word with the separate 8-bit wide data. The two-stage instruction pipeline allows all instructions to execute in a single-cycle, except for program branches (which require two cycles). A total of 35 instructions (reduced instruction set) are available. Additionally, a large register set gives some of the architectural innovations used to achieve a very high performance.

PIC16CXXX microcontrollers typically achieve a 2:1 code compression and a 4:1 speed improvement over other 8-bit microcontrollers in its class.

The PIC16C641 has 128 bytes of RAM and the PIC16C642 has 176 bytes of RAM. Both devices have 22 I/O pins, and an 8-bit timer/counter with an 8-bit programmable prescaler. In addition, they have two analog comparators with a programmable on-chip voltage reference module. Program Memory has internal parity error detection circuitry with a Parity Error Reset. The comparator module is ideally suited for applications requiring a low-cost analog interface (e.g., battery chargers, threshold detectors, white goods controllers, etc.).

The PIC16C661 has 128 bytes of RAM and the PIC16C662 has 176 bytes of RAM. Both devices have 33 I/O pins, and an 8-bit timer/counter with an 8-bit programmable prescaler. They also have an 8-bit Parallel Slave Port. In addition, the devices have two analog comparators with a programmable on-chip voltage reference module. Program Memory has internal parity error detection circuitry with a Parity Error Reset. The comparator module is ideally suited for applications requiring a low-cost analog interface (e.g., battery chargers, threshold detectors, white goods controllers, etc.).

PIC16CXXX devices have special features to reduce external components, thus reducing cost, enhancing system reliability and reducing power consumption. There are four oscillator options, of which the single pin RC oscillator provides a low-cost solution, the LP oscillator minimizes power consumption, XT is a standard crystal, and the HS is for High Speed crystals. The SLEEP (power-down) mode offers power saving. The user can wake-up the chip from SLEEP through several external and internal interrupts and resets.

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

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

The PIC16CXXX series fit perfectly in applications ranging from battery chargers to low-power remote sensors. The EPROM technology makes customization of application programs (detection levels, pulse generation, timers, etc.) extremely fast and convenient. The small footprint packages make this microcontroller series perfect for all applications with space limitations. Low-cost, low-power, high-performance, ease of use, and I/O flexibility make the PIC16C64X & PIC16C66X very versatile.

1.1 Family and Upward Compatibility

Those users familiar with the PIC16C5X family of microcontrollers will realize that this is an enhanced version of the PIC16C5X architecture. Please refer to Appendix A for a detailed list of enhancements. Code written for PIC16C5X can be easily ported to the PIC16C64X & PIC16C66X (Appendix B).

1.2 Development Support

PIC16C64X & PIC16C66X devices are supported by the complete line of Microchip Development tools, including:

- MPLAB Integrated Development Environment including MPLAB-Simulator.
- MPASM Universal Assembler and MPLAB-C Universal C compiler.
- PRO MATE II and PICSTART Plus device programmers.
- PICMASTER In-circuit Emulator System
- *fuzzy*TECH-MP Fuzzy Logic Development Tools
- DriveWay Visual Programming Tool

Please refer to Section 11.0 for more details about these and other Microchip development tools.

TABLE 1-1: PIC16C64X & PIC16C66X DEVICE FEATURES

	Clock	Memory			Peripherals				Features			
	Maximum Frequency of Operation (MHz)	EPROM	Data Memory (bytes)	Program Memory	Timer Module(s)	Comparator(s)	Internal Reference Voltage	Parallel Slave Port	Interrupt Sources	I/O Pins	Voltage Range (Volts)	Brown-out Reset	Packages
PIC16C641	20	2K	128	TMR0	2	Yes	-	4	22	3.0-6.0	Yes	28-pin PDIP, SOIC, Windowed CDIP
PIC16C642	20	4K	176	TMR0	2	Yes	-	4	22	3.0-6.0	Yes	28-pin PDIP, SOIC, Windowed CDIP
PIC16C661	20	2K	128	TMR0	2	Yes	Yes	5	33	3.0-6.0	Yes	40-pin PDIP, Windowed CDIP; 44-pin PLCC, TQFP
PIC16C662	20	4K	176	TMR0	2	Yes	Yes	5	33	3.0-6.0	Yes	40-pin PDIP, Windowed CDIP; 44-pin PLCC, TQFP

All PIC16/17 Family devices have Power-on Reset, selectable Watchdog Timer, selectable code protect, and high I/O current capability.
All PIC16CXXX Family devices use serial programming with clock pin RB6 and data pin RB7.

2.0 PIC16C64X & PIC16C66X DEVICE VARIETIES

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

2.1 UV Erasable Devices

The UV erasable version, offered in CERDIP package is optimal for prototype development and pilot programs. This version can be erased and reprogrammed to any of the oscillator modes.

Microchip's PICSTART® Plus and PRO MATE® II programmers both support programming of the PIC16C64X & PIC16C66X.

2.2 One-Time-Programmable (OTP) Devices

The availability of OTP devices is especially useful for customers who need flexibility for frequent code updates and small volume applications. In addition to the program memory, the configuration bits must also be programmed.

2.3 Quick-Turnaround-Production (QTP) Devices

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

2.4 Serialized Quick-Turnaround-Production (SQTP℠) Devices

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

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

NOTES:

3.0 ARCHITECTURAL OVERVIEW

The high performance of the PIC16C64X & PIC16C66X devices can be attributed to a number of architectural features commonly found in RISC microprocessors. To begin with, the PIC16C64X & PIC16C66X use a Harvard architecture in which program and data are accessed from separate memories using separate buses. This improves bandwidth over traditional von Neumann architecture where program and data are fetched from the same memory. Separating program and data memory further allows instructions to be sized differently than an 8-bit wide data word. Instruction opcodes are 14-bits 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 (200 ns @ 20 MHz) except for program branches, which require two cycles.

The PIC16C641 and PIC16C661 both address 2K x 14 on-chip program memory while the PIC16C642 and PIC16C662 address 4K x 14. All program memory is internal.

PIC16C64X & PIC16C66X devices can directly or indirectly address their register files or data memory. All special function registers including the program counter are mapped in the data memory. These devices have an orthogonal (symmetrical) instruction set that makes it possible to carry out any operation on any register using any addressing mode. This symmetrical nature and lack of ‘special optimal situations’ make programming with the PIC16C64X & PIC16C66X simple yet efficient. In addition, the learning curve is reduced significantly.

PIC16C64X & PIC16C66X devices contain an 8-bit ALU and working register. The ALU is a general purpose arithmetic unit. It performs arithmetic and Boolean functions between data in the working register and any register file.

The ALU is 8-bits wide and capable of addition, subtraction, shift, and logical operations. Unless otherwise mentioned, arithmetic operations are two's complement in nature. In two-operand instructions, typically one operand is the working register (W register). The other operand is a file register or an immediate constant. In single operand instructions, the operand is either the W register or a file register.

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

Depending on the instruction executed, the ALU may affect the values of the Carry (C), Digit Carry (DC), and Zero (Z) bits in the STATUS register. The C and DC bits operate as a $\overline{\text{Borrow}}$ and $\overline{\text{Digit}}$ $\overline{\text{Borrow}}$ out bit, respectively, bit in subtraction. See the SUBLW and SUBWF instructions for examples.

FIGURE 3-1: PIC16C641/642 BLOCK DIAGRAM

PIC16C641 has 2K x 14 Program Memory and 128 x 8 RAM

PIC16C642 has 4K x 14 Program Memory and 176 x 8 RAM

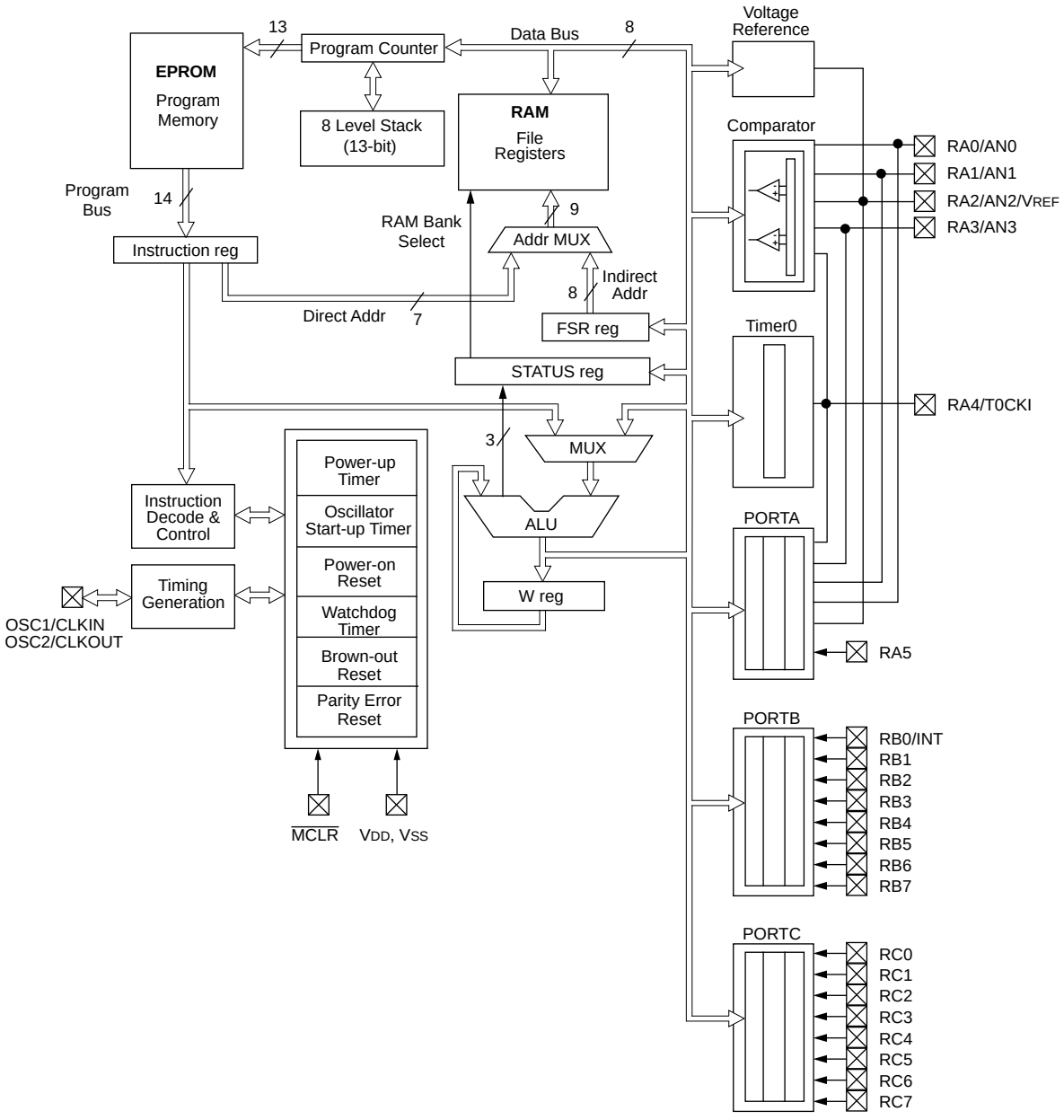


FIGURE 3-2: PIC16C661/662 BLOCK DIAGRAM

PIC16C661 has 2K x 14 Program Memory and 128 x 8 RAM

PIC16C662 has 4K x 14 Program Memory and 176 x 8 RAM

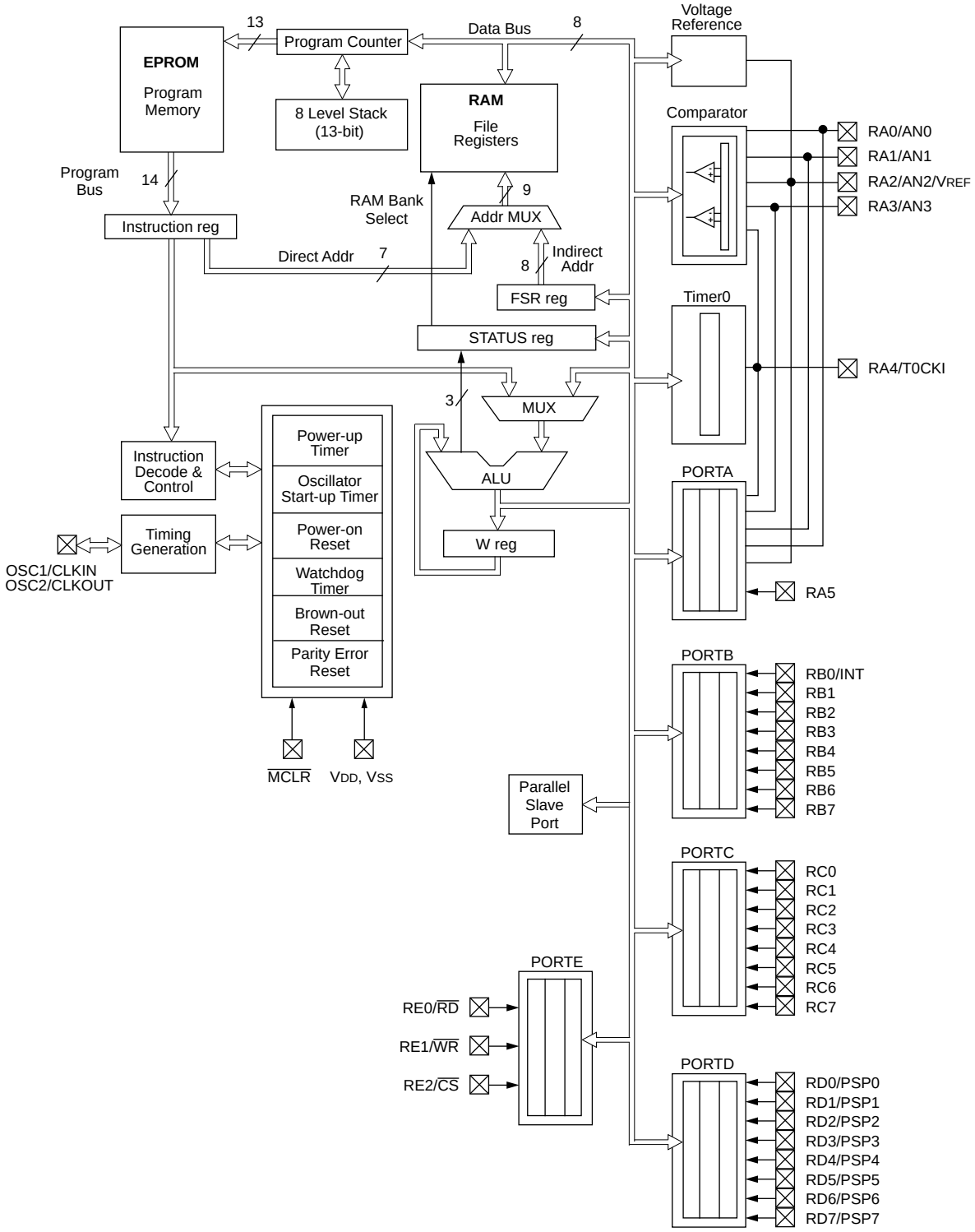


TABLE 3-1: PIC16C641/642 PINOUT DESCRIPTION

Name	Pin #	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	9	I	ST/CMOS	Oscillator crystal input or external clock source input.
OSC2/CLKOUT	10	O	—	Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. In RC mode, OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate.
$\overline{MCLR}$/VPP	1	I/P	ST	Master clear (reset) input or programming voltage input. This pin is an active low reset to the device.
				PORTA is a bi-directional I/O port.
RA0/AN0	2	I/O	ST	Analog comparator input.
RA1/AN1	3	I/O	ST	Analog comparator input.
RA2/AN2/VREF	4	I/O	ST	Analog comparator input or VREF output.
RA3/AN3	5	I/O	ST	Analog comparator input or comparator output.
RA4/T0CKI	6	I/O	ST	Can be selected to be the clock input to the Timer0 timer/counter or a comparator output. Output is open drain type.
RA5	7	I/O	ST	
				PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-ups on all inputs.
RB0/INT	21	I/O	TTL/ST(1)	RB0 can also be selected as an external interrupt pin.
RB1	22	I/O	TTL	
RB2	23	I/O	TTL	
RB3	24	I/O	TTL	
RB4	25	I/O	TTL	Interrupt on change pin.
RB5	26	I/O	TTL	Interrupt on change pin.
RB6	27	I/O	TTL/ST(2)	Interrupt on change pin. Serial programming clock.
RB7	28	I/O	TTL/ST(2)	Interrupt on change pin. Serial programming data.
				PORTC is a bi-directional I/O port.
RC0	11	I/O	ST	
RC1	12	I/O	ST	
RC2	13	I/O	ST	
RC3	14	I/O	ST	
RC4	15	I/O	ST	
RC5	16	I/O	ST	
RC6	17	I/O	ST	
RC7	18	I/O	ST	
VSS	8,19	P	—	Ground reference for logic and I/O pins.
VDD	20	P	—	Positive supply for logic and I/O pins.

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

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.
2: This buffer is a Schmitt Trigger input when used in serial programming mode.

TABLE 3-2: PIC16C661/662 PINOUT DESCRIPTION

Name	DIP Pin #	QFP Pin #	PLCC Pin #	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	13	30	14	I	ST/CMOS	Oscillator crystal input or external clock source input.
OSC2/CLKOUT	14	31	15	O	—	Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. In RC mode, OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate.
$\overline{MCLR}$/VPP	1	18	2	I/P	ST	Master clear (reset) input or programming voltage input. This pin is an active low reset to the device.
						PORTA is a bi-directional I/O port.
RA0/AN0	2	19	3	I/O	ST	Analog comparator input.
RA1/AN1	3	20	4	I/O	ST	Analog comparator input.
RA2/AN2/VREF	4	21	5	I/O	ST	Analog comparator input or VREF output.
RA3/AN3	5	22	6	I/O	ST	Analog comparator input or comparator output.
RA4/T0CKI	6	23	7	I/O	ST	Can be selected to be the clock input to the Timer0 timer/counter or a comparator output. Output is open drain type.
RA5	7	24	8	I/O	ST	
						PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-ups on all inputs.
RB0/INT	33	8	36	I/O	TTL/ST(1)	RB0 can also be selected as an external interrupt pin.
RB1	34	9	37	I/O	TTL	
RB2	35	10	38	I/O	TTL	
RB3	36	11	39	I/O	TTL	
RB4	37	14	41	I/O	TTL	Interrupt on change pin.
RB5	38	15	42	I/O	TTL	Interrupt on change pin.
RB6	39	16	43	I/O	TTL/ST(2)	Interrupt on change pin. Serial programming clock.
RB7	40	17	44	I/O	TTL/ST(2)	Interrupt on change pin. Serial programming data.
						PORTC is a bi-directional I/O port.
RC0	15	32	16	I/O	ST	
RC1	16	35	18	I/O	ST	
RC2	17	36	19	I/O	ST	
RC3	18	37	20	I/O	ST	
RC4	23	42	25	I/O	ST	
RC5	24	43	26	I/O	ST	
RC6	25	44	27	I/O	ST	
RC7	26	1	29	I/O	ST	

Legend: O = output, I = input, TTL = TTL input; I/O = input/output, — = not used; P = power, ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.
2: This buffer is a Schmitt Trigger input when used in serial programming mode.
3: This buffer is a Schmitt Trigger input when configured as a general purpose I/O and a TTL input when used in the Parallel Slave Port Mode (for interfacing to a microprocessor port).

Name	DIP Pin #	QFP Pin #	PLCC Pin #	I/O/P Type	Buffer Type	Description
						PORTD can be a bi-directional I/O port or parallel slave port for interfacing to a microprocessor bus.
RD0/PSP0	19	38	21	I/O	ST/TTL(3)	
RD1/PSP1	20	39	22	I/O	ST/TTL(3)	
RD2/PSP2	21	40	23	I/O	ST/TTL(3)	
RD3/PSP3	22	41	24	I/O	ST/TTL(3)	
RD4/PSP4	27	2	30	I/O	ST/TTL(3)	
RD5/PSP5	28	3	31	I/O	ST/TTL(3)	
RD6/PSP6	29	4	32	I/O	ST/TTL(3)	
RD7/PSP7	30	5	33	I/O	ST/TTL(3)	
						PORTE is a bi-directional I/O port.
RE0/$\overline{RD}$	8	25	9	I/O	ST/TTL(3)	RE0/$\overline{RD}$ read control for parallel slave port.
RE1/$\overline{WR}$	9	26	10	I/O	ST/TTL(3)	RE1/$\overline{WR}$ write control for parallel slave port.
RE2/$\overline{CS}$	10	27	11	I/O	ST/TTL(3)	RE2/$\overline{CS}$ select control for parallel slave port.
VSS	12,31	6,29	13,34	P	—	Ground reference for logic and I/O pins.
VDD	11,32	7,28	12,35	P	—	Positive supply for logic and I/O pins.
NC	—	12,13, 33,34	1,17 28,40	—	—	Not Connected.

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

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.
2: This buffer is a Schmitt Trigger input when used in serial programming mode.
3: This buffer is a Schmitt Trigger input when configured as a general purpose I/O and a TTL input when used in the Parallel Slave Port Mode (for interfacing to a microprocessor port).

3.1 Clocking Scheme/Instruction Cycle

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

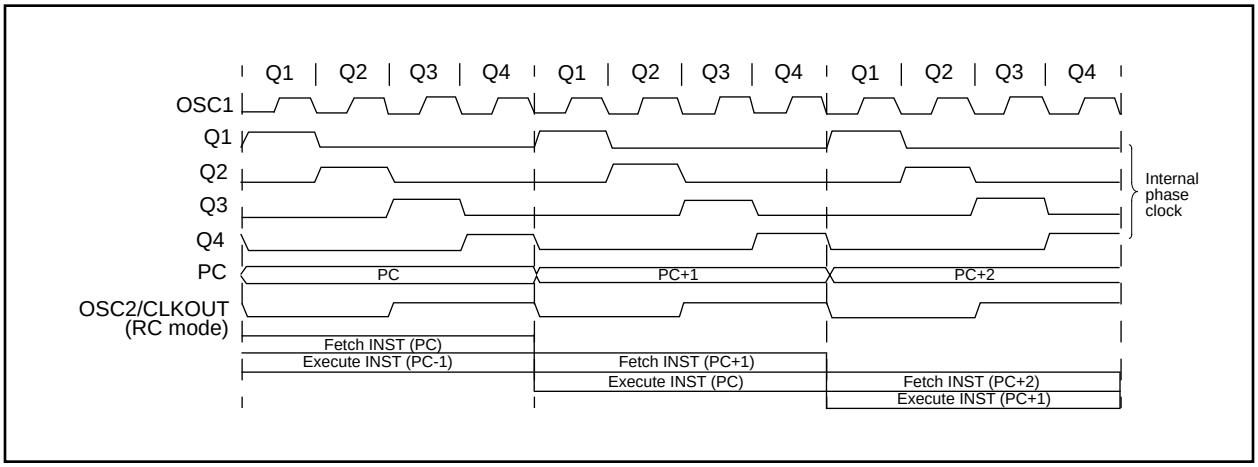
3.2 Instruction Flow/Pipelining

An "Instruction Cycle" consists of four Q cycles (Q1, Q2, Q3, and Q4). The instruction fetch and execute are pipelined such that fetch takes one instruction cycle while decode and execute takes another instruction cycle. However, due to the pipelining, each instruction effectively executes in one cycle. If an instruction causes the program counter to change (e.g., GOTO) then two cycles are required to complete the instruction (Example 3-1).

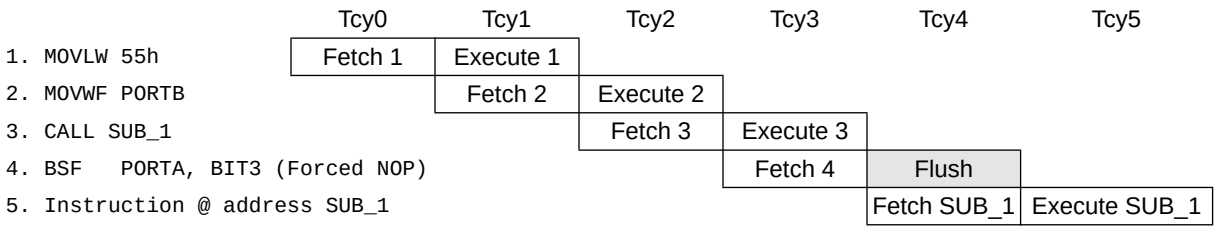
A fetch cycle begins with the program counter (PC) incrementing in Q1.

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

FIGURE 3-3: CLOCK/INSTRUCTION CYCLE



EXAMPLE 3-1: INSTRUCTION PIPELINE FLOW



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

NOTES:

4.0 MEMORY ORGANIZATION

4.1 Program Memory Organization

The PIC16C64X & PIC16C66X have a 13-bit program counter capable of addressing an 8K x 14 program memory space. For the PIC16C641 and PIC16C661 only the first 2K x 14 (0000h - 07FFh) is physically implemented. For the PIC16C642 and PIC16C662 only the first 4K x 14 (0000h - 0FFh) is physically implemented. Accessing a location above the 2K or 4K boundary will cause a wrap-around. The reset vector is at 0000h and the interrupt vector is at 0004h (Figure 4-1 and Figure 4-2). See Section 4.4 for Program Memory paging.

FIGURE 4-1: PIC16C641/661 PROGRAM MEMORY MAP AND STACK

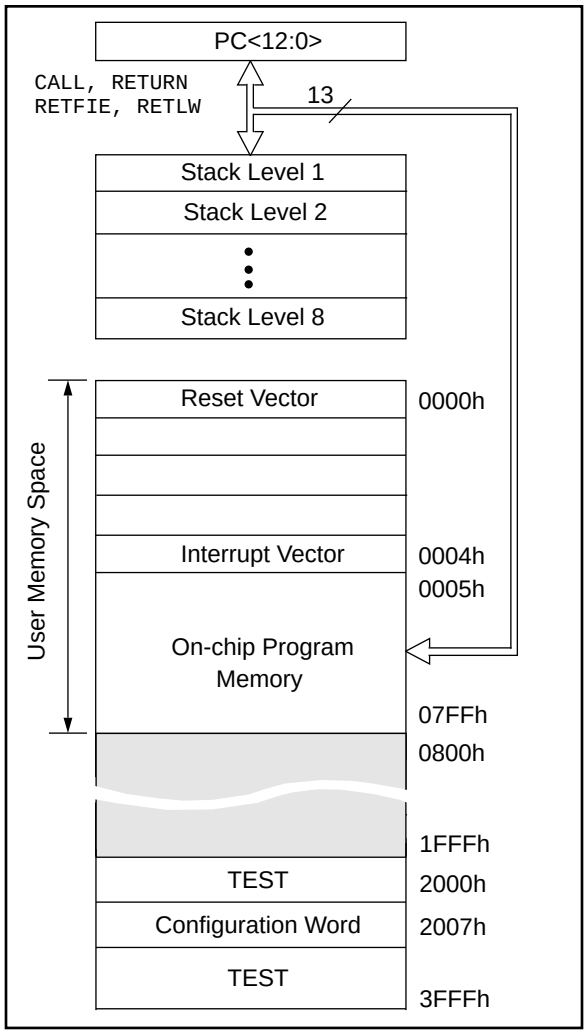
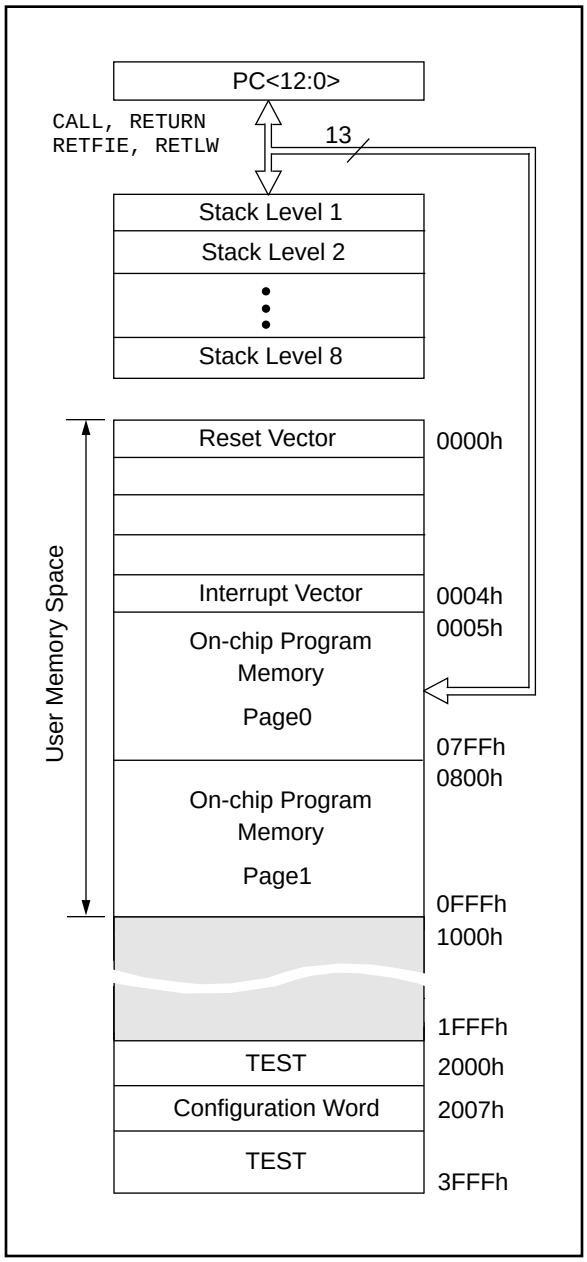


FIGURE 4-2: PIC16C642/662 PROGRAM MEMORY MAP AND STACK



4.2 Data Memory Organization

The data memory (Figure 4-4) is partitioned into two banks which contain the general purpose registers and the special function registers. Bank 0 is selected when bit RP0 (STATUS<5>) is cleared. Bank 1 is selected when the RP0 bit is set. The Special Function Registers are located in the first 32 locations of each Bank. Register locations A0h-EFh (Bank 1) are general purpose registers implemented as static RAM. Some special function registers are mapped in Bank 1.

4.2.1 GENERAL PURPOSE REGISTER FILE

The register file is organized as 176 x 8 for the PIC16C642/662, and 128 x8 for the PIC16C641/661. Each is accessed either directly, or indirectly through the File Select Register FSR (Section 4.5).

FIGURE 4-3: PIC16C641/661 DATA MEMORY MAP

File Address	Bank 0	Bank 1	File Address
00h	INDF(1)	INDF(1)	80h
01h	TMR0	OPTION	81h
02h	PCL	PCL	82h
03h	STATUS	STATUS	83h
04h	FSR	FSR	84h
05h	PORTA	TRISA	85h
06h	PORTB	TRISB	86h
07h	PORTC	TRISC	87h
08h	PORTD(2)	TRISD(2)	88h
09h	PORTE(2)	TRISE(2)	89h
0Ah	PCLATH	PCLATH	8Ah
0Bh	INTCON	INTCON	8Bh
0Ch	PIR1	PIE1	8Ch
0Dh			8Dh
0Eh		PCON	8Eh
0Fh			8Fh
10h			90h
11h			91h
12h			92h
13h			93h
14h			94h
15h			95h
16h			96h
17h			97h
18h			98h
19h			99h
1Ah			9Ah
1Bh			9Bh
1Ch			9Ch
1Dh			9Dh
1Eh			9Eh
1Fh	CMCON	VRCON	9Fh
20h	General Purpose Register	General Purpose Register	A0h
			BFh
			C0h
			EFh
		Mapped in Page 0	F0h
7Fh			FFh

Unimplemented data memory locations, read as '0'.

Note 1: Not a physical register.
2: Not implemented on the PIC16C641.

FIGURE 4-4: PIC16C642/662 DATA MEMORY MAP

File Address	Bank 0	Bank 1	File Address
00h	INDF(1)	INDF(1)	80h
01h	TMR0	OPTION	81h
02h	PCL	PCL	82h
03h	STATUS	STATUS	83h
04h	FSR	FSR	84h
05h	PORTA	TRISA	85h
06h	PORTB	TRISB	86h
07h	PORTC	TRISC	87h
08h	PORTD(2)	TRISD(2)	88h
09h	PORTE(2)	TRISE(2)	89h
0Ah	PCLATH	PCLATH	8Ah
0Bh	INTCON	INTCON	8Bh
0Ch	PIR1	PIE1	8Ch
0Dh			8Dh
0Eh		PCON	8Eh
0Fh			8Fh
10h			90h
11h			91h
12h			92h
13h			93h
14h			94h
15h			95h
16h			96h
17h			97h
18h			98h
19h			99h
1Ah			9Ah
1Bh			9Bh
1Ch			9Ch
1Dh			9Dh
1Eh			9Eh
1Fh	CMCON	VRCON	9Fh
20h	General Purpose Register	General Purpose Register	A0h
			EFh
		Mapped in Bank 0	F0h
7Fh			FFh

Unimplemented data memory locations, read as '0'.

Note 1: Not a physical register.
2: Not implemented on the PIC16C642.

4.2.2 SPECIAL FUNCTION REGISTERS

The special function registers are registers used by the CPU and Peripheral Modules for controlling the desired operation of the device (Table 4-1). These registers are static RAM.

The special function registers can be classified into two sets (core and peripheral). The special function registers associated with the “core” functions are described in this section. Those related to the operation of the peripheral features are described in the section of that peripheral feature.

TABLE 4-1: SPECIAL FUNCTION REGISTERS

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR, PER	Value on all other resets(1)
Bank 0											
00h	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx
01h	TMR0	Timer0 Module's Register								xxxx xxxx	uuuu uuuu
02h	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	0000 0000
03h	STATUS	IRP(2)	RP1(2)	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000q quuu
04h	FSR	Indirect data memory address pointer								xxxx xxxx	uuuu uuuu
05h	PORTA	—	—	PORTA Data Latch when written: PORTA pins when read						--xx 0000	--xu 0000
06h	PORTB	PORTB Data Latch when written: PORTB pins when read								xxxx xxxx	uuuu uuuu
06h	PORTC	PORTC Data Latch when written: PORTC pins when read								xxxx xxxx	uuuu uuuu
06h	PORTD(3)	PORTD Data Latch when written: PORTD pins when read								xxxx xxxx	uuuu uuuu
06h	PORTE(3)	—	—	—	—	—	RE2	RE1	RE0	---- -xxx	---- -uuu
0Ah	PCLATH	—	—	—	Write buffer for upper 5 bits of program counter					---0 0000	---0 0000
0Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF(4)	CMIF	—	—	—	—	—	—	00-- ----	00-- ----
0Dh-1Eh	Unimplemented									–	–
1Fh	CMCON	C2OUT	C1OUT	—	—	CIS	CM2	CM1	CM0	00-- 0000	00-- 0000
Bank 1											
80h	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx
81h	OPTION	$\overline{RBPU}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
82h	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	0000 0000
83h	STATUS	IRP(2)	RP1(2)	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000q quuu
84h	FSR	Indirect data memory address pointer								xxxx xxxx	uuuu uuuu
85h	TRISA	—	—	PORTA Data Direction Register						--11 1111	--11 1111
86h	TRISB	PORTB Data Direction Register								1111 1111	1111 1111
86h	TRISC	PORTC Data Direction Register								1111 1111	1111 1111
86h	TRISD(3)	PORTD Data Direction Register								1111 1111	1111 1111
86h	TRISE(3)	IBF	OBF	IBOV	PSPMODE	—	TRISE2	TRISE1	TRISE0	0000 -111	0000 -111
8Ah	PCLATH	—	—	—	Write buffer for upper 5 bits of program counter					---0 0000	---0 0000
8Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000x
8Ch	PIE1	PSPIE(4)	CMIE	—	—	—	—	—	—	00-- ----	00-- ----
8Dh	Unimplemented									–	–
8Eh	PCON	MPEEN	—	—	—	—	$\overline{PER}$	$\overline{POR}$	$\overline{BOR}$	u--- -qqq	u--- -uuu
8Fh-9Eh	Unimplemented									–	–
9Fh	VRCON	VREN	VROE	VRR	—	VR3	VR2	VR1	VR0	000- 0000	000- 0000

Legend: - = unimplemented locations read as '0', u = unchanged, x = unknown, q = value depends on condition, shaded = unimplemented

Note 1: Other (non power-up) resets include $\overline{MCLR}$ Reset and Watchdog Timer Reset during normal operation.
2: The IRP and RP1 bits are reserved, always maintain these bits clear.
3: The PORTD, PORTE, TRISD, and TRISE registers are not implemented on the PIC16C641/642.
4: Bits PSPIE and PSPIF are reserved on the PIC16C641/642, always maintain these bits clear.

4.2.2.1 STATUS REGISTER

The STATUS register, shown in Figure 4-5, contains the arithmetic status of the ALU, the RESET status, and the bank select bits for data memory.

The STATUS register can be the destination for any instruction, like any other register. If the STATUS register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. These bits are set or cleared according to the device logic. Furthermore, the $\overline{TO}$ and $\overline{PD}$ bits are not writable. Therefore, the result of an instruction with the STATUS register as destination may be different than intended.

For example, CLRF STATUS will clear the upper-three bits and set the Z bit. This leaves the STATUS register as 000uu1uu (where u = unchanged).

It is recommended, therefore, that only BCF, BSF, SWAPF, and MOVWF instructions are used to alter the STATUS register because these instructions do not affect any status bit. For other instructions, not affecting any status bits, see the "Instruction Set Summary."

Note 1: The IRP and RP1 bits (STATUS<7:6>) are reserved on the PIC16C64X & PIC16C66X and should be maintained clear. Use of these bits as general purpose R/W bits is NOT recommended, since this may affect upward compatibility with future products.

Note 2: The C and DC bits operate as a $\overline{Borrow}$ and $\overline{Digit\ Borrow}$ out bit, respectively, in subtraction. See the SUBLW and SUBWF instructions for examples.

FIGURE 4-5: STATUS REGISTER (ADDRESS 03h, 83h)

R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C
bit7							bit0

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

bit 7: **IRP:** Register Bank Select bit (used for indirect addressing)
1 = Bank 2, 3 (100h - 1FFh)
0 = Bank 0, 1 (00h - FFh)
Bit IRP is reserved on the PIC16C64X & PIC16C66X, always maintain this bit clear.

bit 6-5: **RP1:RP0**: Register Bank Select bits (used for direct addressing)
11 = Bank 3 (180h - 1FFh)
10 = Bank 2 (100h - 17Fh)
01 = Bank 1 (80h - FFh)
00 = Bank 0 (00h - 7Fh)
Each bank is 128 bytes. Bit RP1 is reserved on the PIC16C64X & PIC16C66X, always maintain this bit clear.

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

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

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

bit 1: **DC**: Digit carry/$\overline{borrow}$ bit (ADDWF, ADDLW, SUBLW, SUBWF instructions) (for $\overline{borrow}$ the polarity is reversed)
1 = A carry-out from the 4th low order bit of the result occurred
0 = No carry-out from the 4th low order bit of the result

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

4.2.2.2 OPTION REGISTER

The OPTION register is a readable and writable register which contains various control bits to configure the TMR0/WDT prescaler, the external RB0/INT interrupt, TMR0, and the weak pull-ups on PORTB.

Note: To achieve a 1:1 prescaler assignment for TMR0, assign the prescaler to the WDT.

FIGURE 4-6: OPTION REGISTER (ADDRESS 81h)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit7							bit0

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

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

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

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

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

bit 3: **PSA**: Prescaler Assignment bit
1 = Prescaler is assigned to the WDT
0 = Prescaler is assigned to the Timer0 module

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

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

4.2.2.3 INTCON REGISTER

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

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

FIGURE 4-7: INTCON REGISTER (ADDRESS 0Bh, 8Bh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF
bit7							bit0

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

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

bit 6: **PEIE**: Peripheral Interrupt Enable bit
1 = Enables all un-masked peripheral interrupts
0 = Disables all peripheral interrupts

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

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

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

bit 2: **T0IF**: TMR0 Overflow Interrupt Flag bit
1 = TMR0 register overflowed (must be cleared in software)
0 = TMR0 register did not overflow

bit 1: **INTF**: RB0/INT External Interrupt Flag bit
1 = The RB0/INT external interrupt occurred (must be cleared in software)
0 = The RB0/INT external interrupt did not occur

bit 0: **RBIF**: RB Port Change Interrupt Flag bit
1 = When at least one of the RB7:RB4 pins changed state (See Section 5.2 to clear interrupt)
0 = None of the RB7:RB4 pins have changed state

4.2.2.4 PIE1 REGISTER

This register contains the individual enable bits for the comparator and Parallel Slave Port interrupts.

FIGURE 4-8: PIE1 REGISTER (ADDRESS 8Ch)

R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0	U-0
PSPIE(1)	CMIE	—	—	—	—	—	—
bit7							bit0

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

bit 7: **PSPIE(1):** Parallel Slave Port Read/Write Interrupt Enable bit
1 = Enables the PSP read/write interrupt
0 = Disables the PSP read/write interrupt

bit 6: **CMIE**: Comparator Interrupt Enable bit
1 = Enables the Comparator interrupt
0 = Disables the Comparator interrupt

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

Note 1: Bit PSPIE is reserved on the PIC16C641/642, always maintain this bit clear.

4.2.2.5 PIR1 REGISTER

This register contains the individual flag bits for the comparator and Parallel Slave Port interrupts.

Note:	Interrupt flag bits get set when an interrupt condition occurs regardless of the state of its corresponding enable bit or the global enable bit, GIE (INTCON<7>). User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

FIGURE 4-9: PIR1 REGISTER (ADDRESS 0Ch)

R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0	U-0
PSPIF(1)	CMIF	—	—	—	—	—	—
bit7							bit0

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

bit 7: **PSPIF(1):** Parallel Slave Port Interrupt Flag bit
1 = A read or write operation has taken place (must be cleared in software)
0 = No read or write operation has taken place

bit 6: **CMIF**: Comparator Interrupt Flag bit
1 = Comparator input has changed (must be cleared in software)
0 = Comparator input has not changed

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

Note 1: Bit PSPIF is reserved on the PIC16C641/642, always maintain this bit clear.

4.2.2.6 PCON REGISTER

The PCON register contains flag bits to differentiate between a Power-on Reset (POR), an external $\overline{MCLR}$ reset, WDT reset, Brown-out Reset ($\overline{BOR}$), and Parity Error Reset ($\overline{PER}$). The PCON register also contains a status bit, MPEEN, which reflects the value of the MPEEN bit in Configuration Word. See Table 9-4 for status of these bits on various resets.

Note: $\overline{BOR}$ is unknown on Power-on Reset. It must then be set by the user and checked on subsequent resets to see if $\overline{BOR}$ is cleared, indicating a brown-out has occurred. The $\overline{BOR}$ status bit is a “don't care” and is not necessarily predictable if the brown-out circuit is disabled (by programming the BODEN bit in the Configuration word).

FIGURE 4-10: PCON REGISTER (ADDRESS 8Eh)

R-U	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-u
MPEEN	—	—	—	—	$\overline{PER}$	$\overline{POR}$	$\overline{BOR}$
bit7							bit0

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

bit 7: **MPEEN**: Memory Parity Error Circuitry Status bit
Reflects the value of Configuration Word bit, MPEEN

bit 6-3: **Unimplemented**: Read as '0'

bit 2: **$\overline{PER}$**: Memory Parity Error Reset Status bit
1 = No error occurred
0 = Program memory fetch parity error occurred
(must be set in software after a Parity Error Reset occurs)

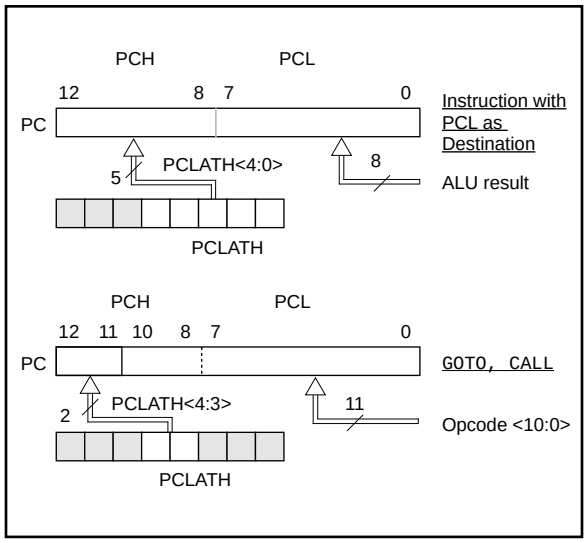
bit 1: **$\overline{POR}$**: Power-on Reset Status bit
1 = No Power-on Reset occurred
0 = A Power-on Reset occurred (must be set in software after a Power-on Reset occurs)

bit 0: **$\overline{BOR}$**: Brown-out Reset Status bit
1 = No Brown-out Reset occurred
0 = A Brown-out Reset occurred (must be set in software after a Brown-out Reset occurs)

4.3 PCL and PCLATH

The program counter (PC) is 13-bits wide. The low byte comes from the PCL register, which is readable and writable. The high byte (PC<12:8>) is not directly readable or writable and comes from PCLATH. On any reset, the PC is cleared. Figure 4-11 shows the two situations for the loading of the PC. The upper example in the figure shows how the PC is loaded on a write to PCL (PCLATH<4:0> → PCH). The lower example in the figure shows how the PC is loaded during a CALL or GOTO instruction (PCLATH<4:3> → PCH).

FIGURE 4-11: LOADING OF PC IN DIFFERENT SITUATIONS



4.3.1 COMPUTED GOTO

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

4.3.2 STACK

PIC16C64X & PIC16C66X devices have an 8 level deep x 13-bit wide hardware stack (Figure 4-2). 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 onto the stack when a CALL instruction is executed or an interrupt causes a branch. The stack is POPed in the event of a RETURN, RETLW or a RETFIE instruction execution. PCLATH is not affected by a PUSH or POP operation.

The stack operates as a circular buffer. This means that after the stack has been PUSHed eight times, the ninth push overwrites the value that was stored from the first push. The tenth push overwrites the second push (and so on).

Note 1:	There are no status bits to indicate stack overflow or stack underflow conditions.
Note 2:	There are no instructions mnemonics called PUSH or POP. These are actions that occur from the execution of the CALL, RETURN, RETLW, and RETFIE instructions, or the vectoring to an interrupt address.

4.4 Program Memory Paging

PIC16C642 and PIC16C662 devices have 4K of program memory, but the CALL and GOTO instructions only have an 11-bit address range. This 11-bit address range allows a branch within a 2K program memory page size. To allow CALL and GOTO instructions to address the entire 4K program memory address range, there must be another bit to specify the program memory page. This paging bit comes from the PCLATH<3> bit (Figure 4-11). When doing a CALL or GOTO instruction, the user must ensure that this page select bit (PCLATH<3>) is programmed so that the desired program memory page is addressed. If a return from a CALL instruction (or interrupt) is executed, the entire 13-bit PC is pushed onto the stack. Therefore, manipulation of the PCLATH<3> bit is not required for the return instructions (which POPs the address from the stack).

Note:	The PIC16C64X & PIC16C66X ignore the PCLATH<4> bit, which is used for program memory pages 2 and 3 (1000h - 1FFFh). The use of PCLATH<4> as a general purpose read/write bit is not recommended since this may affect upward compatibility with future products.

4.5 Indirect Addressing, INDF, and FSR Registers

The INDF register is not a physical register. Addressing the INDF register will cause indirect addressing.

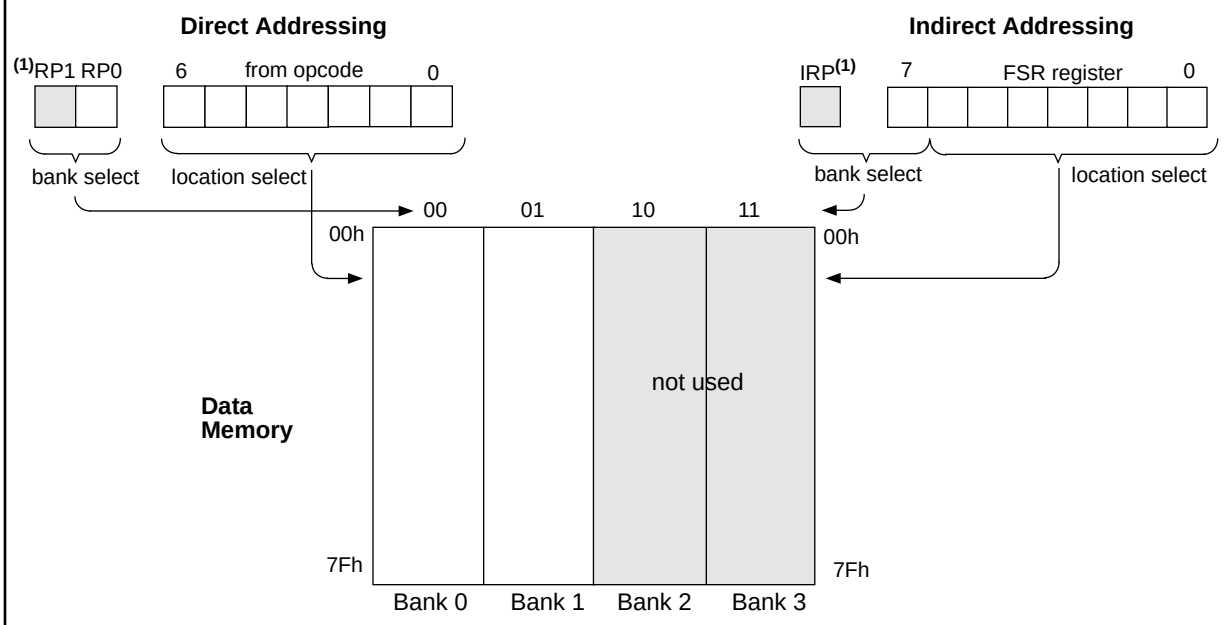
Indirect addressing is possible by using the INDF register. Any instruction using the INDF register actually accesses data pointed to by the file select register (FSR). Reading INDF itself indirectly will produce 00h. Writing to the INDF register indirectly results in a no-operation (although status bits may be affected). An effective 9-bit address is obtained by concatenating the 8-bit FSR register and the IRP bit (STATUS<7>), as shown in Figure 4-12. However, bit IRP is not used in the PIC16C64X & PIC16C66X.

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

EXAMPLE 4-1: INDIRECT ADDRESSING

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

FIGURE 4-12: DIRECT/INDIRECT ADDRESSING



For memory map detail see Figure 4-3 and Figure 4-4.

Note 1: Bits RP1 and IRP are reserved, always maintain these bits clear.

5.0 I/O PORTS

The PIC16C641 and PIC16C642 have three ports, PORTA, PORTB, and PORTC. PIC16C661 and PIC16C662 devices have five ports, PORTA through PORTE. Some pins for these I/O ports are multiplexed with alternate functions for the peripheral features on the device. In general, when a peripheral is enabled, that pin may not be used as a general purpose I/O pin.

5.1 PORTA and TRISA Registers

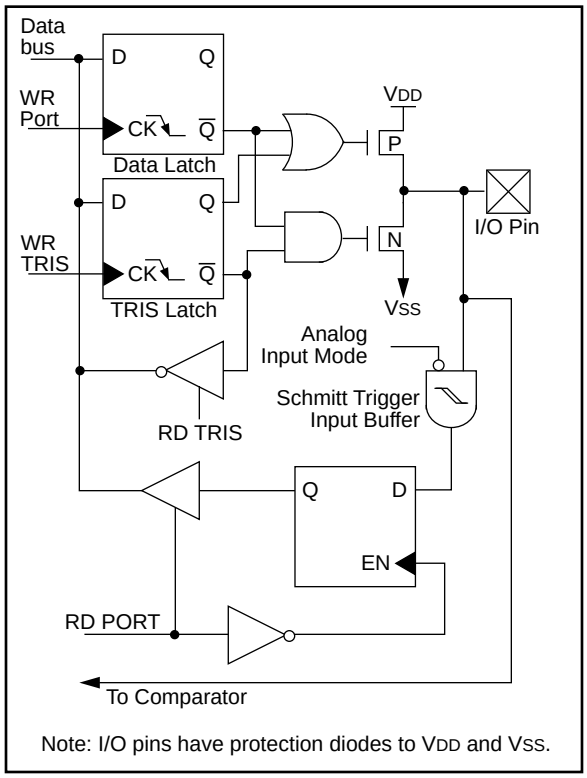
PORTA is a 6-bit wide latch. RA4 is a Schmitt Trigger input and an open drain output. Pin RA4 is multiplexed with the T0CKI clock input. All other RA port pins have Schmitt Trigger input levels and full CMOS output drivers. All pins have data direction bits (TRIS registers) which can configure these pins as input or output.

Setting a bit in the TRISA register puts the corresponding output driver in a hi-impedance mode. Clearing a bit in the TRISA register puts the contents of the output latch on the selected pin.

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

The PORTA pins are multiplexed with comparator and voltage reference functions. The operation of these pins are selected by control bits in the CMCON (comparator control register) register and the VRCON (voltage reference control) register. When selected as comparator inputs, these pins will read as '0's.

FIGURE 5-1: BLOCK DIAGRAM OF RA1:RA0 PINS



Note: I/O pins have protection diodes to VDD and VSS.

Note:	On reset, the TRISA register is set to all inputs. The digital inputs are disabled and the comparator inputs are forced to ground to reduce excess current consumption.

TRISA controls the direction of the RA pins, even when they are being used as comparator inputs. The user must make sure to keep the pins configured as inputs when using them as comparator inputs.

The RA2 pin will also function as the output for the voltage reference. When in this mode, the VREF pin is a very hi-impedance output. The user must set the TRISA<2> bit and use hi-impedance loads.

In one of the comparator modes defined by the CMCON register, pins RA3 and RA4 become outputs of the comparators. The TRISA<4:3> bits must be cleared to enable outputs to use this function.

EXAMPLE 5-1: INITIALIZING PORTA

```
CLRF    PORTA        ;Initialize PORTA by
                     ;clearing output latches
MOVLW   0x07         ;Turn comparators off,
MOVWF   CMCON        ;enable pins for I/O
BSF     STATUS, RP0  ;Select bank1
MOVLW   0x1F         ;Value to initialize
                     ;data direction
MOVWF   TRISA        ;Set RA<4:0> as inputs
                     ;TRISA<7:5> are clear
```

FIGURE 5-2: BLOCK DIAGRAM OF RA2 PIN

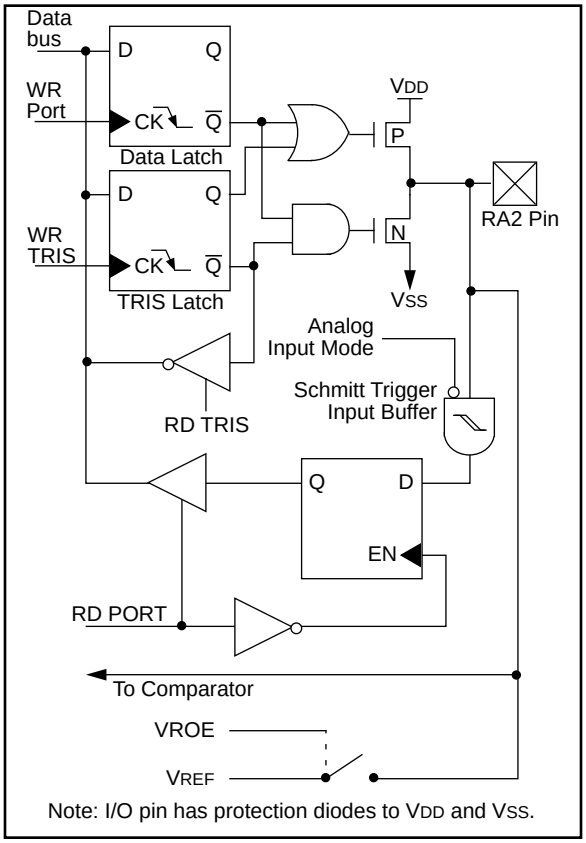


FIGURE 5-3: BLOCK DIAGRAM OF RA3 PIN

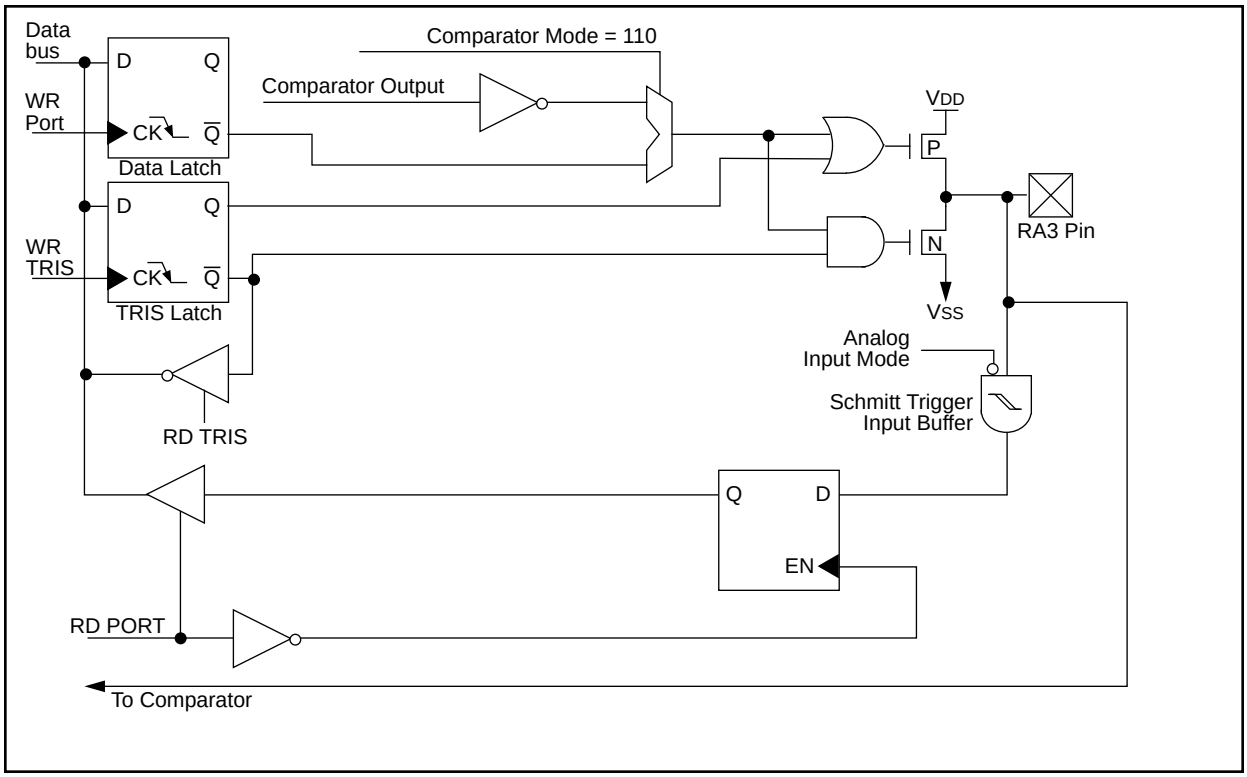


FIGURE 5-4: BLOCK DIAGRAM OF RA4 PIN

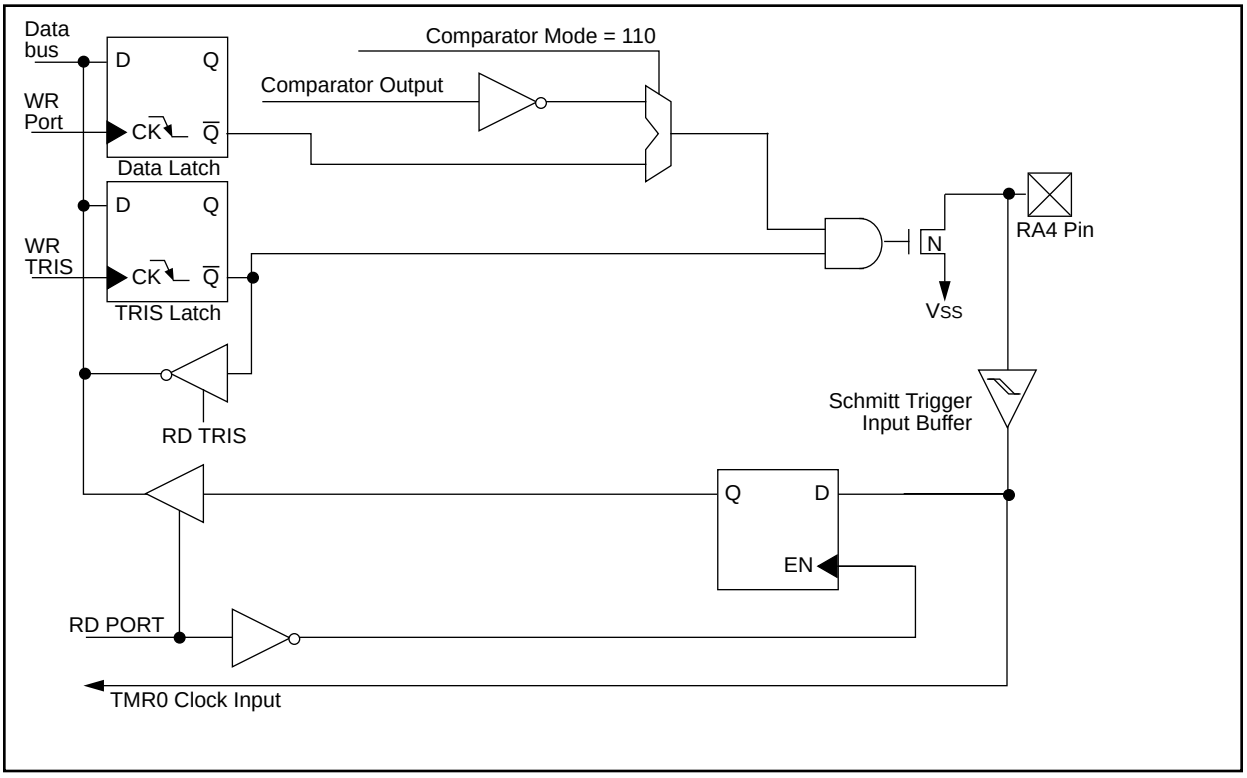


TABLE 5-1: PORTA FUNCTIONS

Name	Bit #	Buffer Type	Function
RA0/AN0	bit0	ST	Input/output or comparator input.
RA1/AN1	bit1	ST	Input/output or comparator input.
RA2/AN2/VREF	bit2	ST	Input/output or comparator input or VREF output.
RA3/AN3	bit3	ST	Input/output or comparator input/output.
RA4/T0CKI	bit4	ST	Input/output or external clock input for TMR0 or comparator output. Output is open drain type.
RA5	bit5	ST	Input/output.

Legend: ST = Schmitt Trigger input

TABLE 5-2: SUMMARY OF REGISTERS ASSOCIATED WITH PORTA

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other resets
05h	PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	--xx 0000	--uu 0000
85h	TRISA	—	—	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	--11 1111	--11 1111
1Fh	CMCON	C2OUT	C1OUT	—	—	CIS	CM2	CM1	CM0	00-- 0000	00-- 0000
9Fh	VRCON	VREN	VROE	VRR	—	VR3	VR2	VR1	VR0	000- 0000	000- 0000

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

5.2 PORTB and TRISB Registers

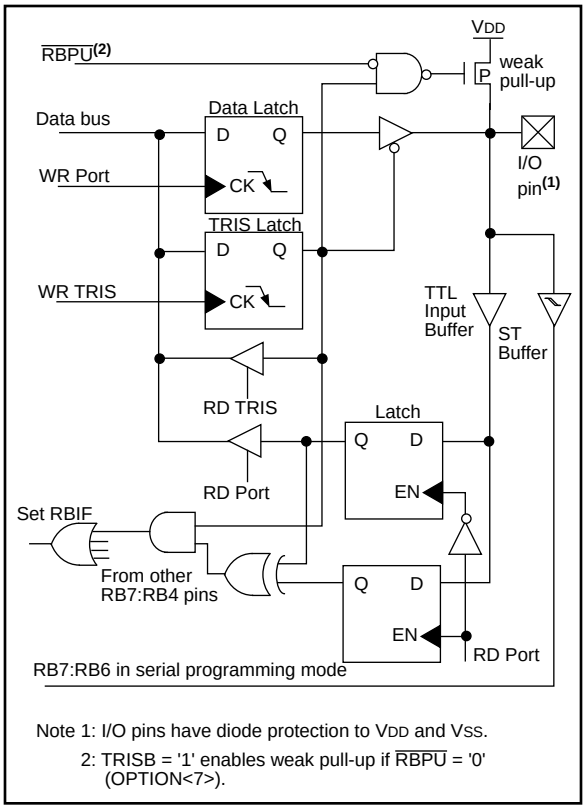
PORTB is an 8-bit wide bi-directional port. The corresponding data direction register is TRISB. Setting a bit in the TRISB register puts the corresponding output driver in a hi-impedance mode. Clearing a bit in the TRISB register puts the contents of the output latch on the selected pin(s).

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

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

Four of PORTB's pins, RB7:RB4, have an interrupt on change feature. Only pins configured as inputs can cause this interrupt to occur (i.e., any RB7:RB4 pin configured as an output is excluded from the interrupt on change comparison). The input pins (of RB7:RB4) are compared with the old value latched on the last read of PORTB. The "mismatch" outputs of RB7:RB4 are OR'ed together to generate the RBIF interrupt (flag latched in (INTCON<0>)).

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



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

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

a) Any read or write of PORTB. This will end the mismatch condition.

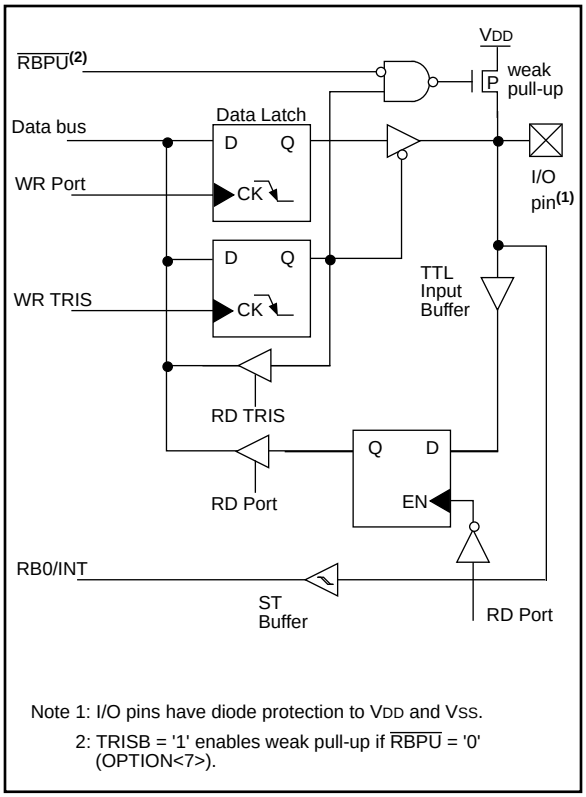
b) Clear flag bit RBIF.

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

This interrupt on mismatch feature, together with software configurable pull-ups on these four pins allow easy interface to a keypad and make it possible for wake-up on key-depression. (See AN552 in the Microchip *Embedded Control Handbook*.)

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

FIGURE 5-6: BLOCK DIAGRAM OF RB3:RB0 PINS



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

EXAMPLE 5-2: INITIALIZING PORTB

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

TABLE 5-3: PORTB FUNCTIONS

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

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

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.

2: This buffer is a Schmitt Trigger input when used in serial programming mode.

TABLE 5-4: SUMMARY OF REGISTERS ASSOCIATED WITH PORTB

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other resets
06h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	uuuu uuuu
86h	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	1111 1111	1111 1111
81h	OPTION	$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111

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

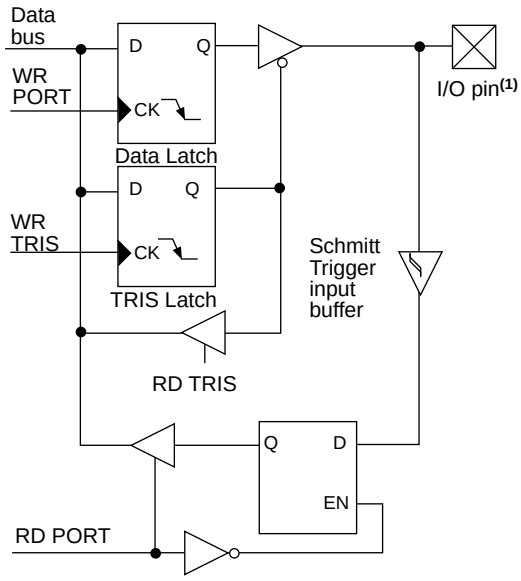
5.3 PORTC and TRISC Registers

PORTC is an 8-bit bi-directional port. Each pin is individually configurable as an input or output through the TRISC register. PORTC pins have Schmitt Trigger input buffers.

EXAMPLE 5-3: INITIALIZING PORTC

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

FIGURE 5-7: PORTC BLOCK DIAGRAM (IN I/O PORT MODE)



Note 1: I/O pins have protection diodes to VDD and VSS.

TABLE 5-5: PORTC FUNCTIONS

Name	Bit#	Buffer Type	Function
RC0	bit0	ST	Input/output
RC1	bit1	ST	Input/output
RC2	bit2	ST	Input/output
RC3	bit3	ST	Input/output
RC4	bit4	ST	Input/output
RC5	bit5	ST	Input/output
RC6	bit6	ST	Input/output
RC7	bit7	ST	Input/output

Legend: ST = Schmitt Trigger input

TABLE 5-6: SUMMARY OF REGISTERS ASSOCIATED WITH PORTC

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other resets
07h	PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxx xxxx	uuuu uuuu
87h	TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	1111 1111	1111 1111

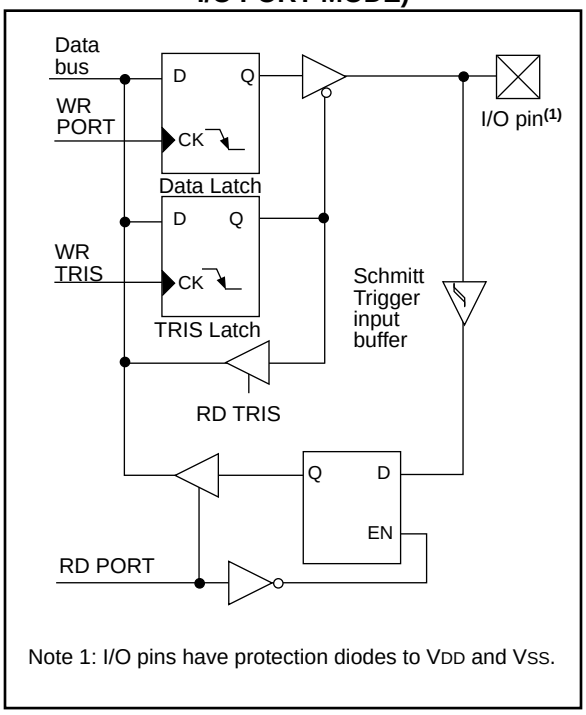
Legend: x = unknown, u = unchanged.

5.4 PORTD and TRISD Registers (PIC16C661 and PIC16C662 only)

PORTD is an 8-bit port with Schmitt Trigger input buffers. Each pin is individually configurable as an input or output.

PORTD can be configured as an 8-bit wide microprocessor port (parallel slave port) by setting control bit PSPMODE (TRISE<4>). In this mode, the input buffers are TTL.

FIGURE 5-8: PORTD BLOCK DIAGRAM (IN I/O PORT MODE)



Note 1: I/O pins have protection diodes to VDD and VSS.

TABLE 5-7: PORTD FUNCTIONS

Name	Bit#	Buffer Type	Function
RD0/PSP0	bit0	ST/TTL(1)	Input/output port pin or parallel slave port bit0
RD1/PSP1	bit1	ST/TTL(1)	Input/output port pin or parallel slave port bit1
RD2/PSP2	bit2	ST/TTL(1)	Input/output port pin or parallel slave port bit2
RD3/PSP3	bit3	ST/TTL(1)	Input/output port pin or parallel slave port bit3
RD4/PSP4	bit4	ST/TTL(1)	Input/output port pin or parallel slave port bit4
RD5/PSP5	bit5	ST/TTL(1)	Input/output port pin or parallel slave port bit5
RD6/PSP6	bit6	ST/TTL(1)	Input/output port pin or parallel slave port bit6
RD7/PSP7	bit7	ST/TTL(1)	Input/output port pin or parallel slave port bit7

Legend: ST = Schmitt Trigger input, TTL = TTL input
Note 1: Input buffers are Schmitt Triggers when in I/O mode and TTL buffers when in Parallel Slave Port Mode.

TABLE 5-8: SUMMARY OF REGISTERS ASSOCIATED WITH PORTD

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other resets
08h	PORTD	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	xxxx xxxx	uuuu uuuu
88h	TRISD	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	1111 1111	1111 1111
89h	TRISE	IBF	OBF	IBOV	PSPMODE	—	TRISE2	TRISE1	TRISE0	0000 -111	0000 -111

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

5.5 PORTE and TRISE Register (PIC16C661 and PIC16C662 only)

PORTE has three pins RE0/$\overline{RD}$, RE1/$\overline{WR}$, and RE2/$\overline{CS}$, which are individually configurable as inputs or outputs. These pins have Schmitt Trigger input buffers.

I/O PORTE becomes control inputs for the microprocessor port when bit PSPMODE (TRISE<4>) is set. In this mode, the user must make sure that the TRISE<2:0> bits are set (pins are configured as digital inputs). In this mode the input buffers are TTL.

Figure 5-9 shows the TRISE register, which also controls the parallel slave port operation.

FIGURE 5-9: TRISE REGISTER (ADDRESS 89h)

R-0	R-0	R/W-0	R/W-0	U-0	R/W-1	R/W-1	R/W-1
IBF	OBF	IBOV	PSPMODE	—	TRISE2	TRISE1	TRISE0
bit7							bit0

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

bit 7: **IBF:** Input Buffer Full Status bit
1 = A word has been received and waiting to be read by the CPU
0 = No word has been received

bit 6: **OBF**: Output Buffer Full Status bit
1 = The output buffer still holds a previously written word
0 = The output buffer has been read

bit 5: **IBOV**: Input Buffer Overflow Detect bit (in microprocessor mode)
1 = A write occurred when a previously input word has not been read (must be cleared in software)
0 = No overflow occurred

bit 4: **PSPMODE**: Parallel Slave Port Mode Select bit
1 = Parallel slave port mode
0 = General purpose I/O mode

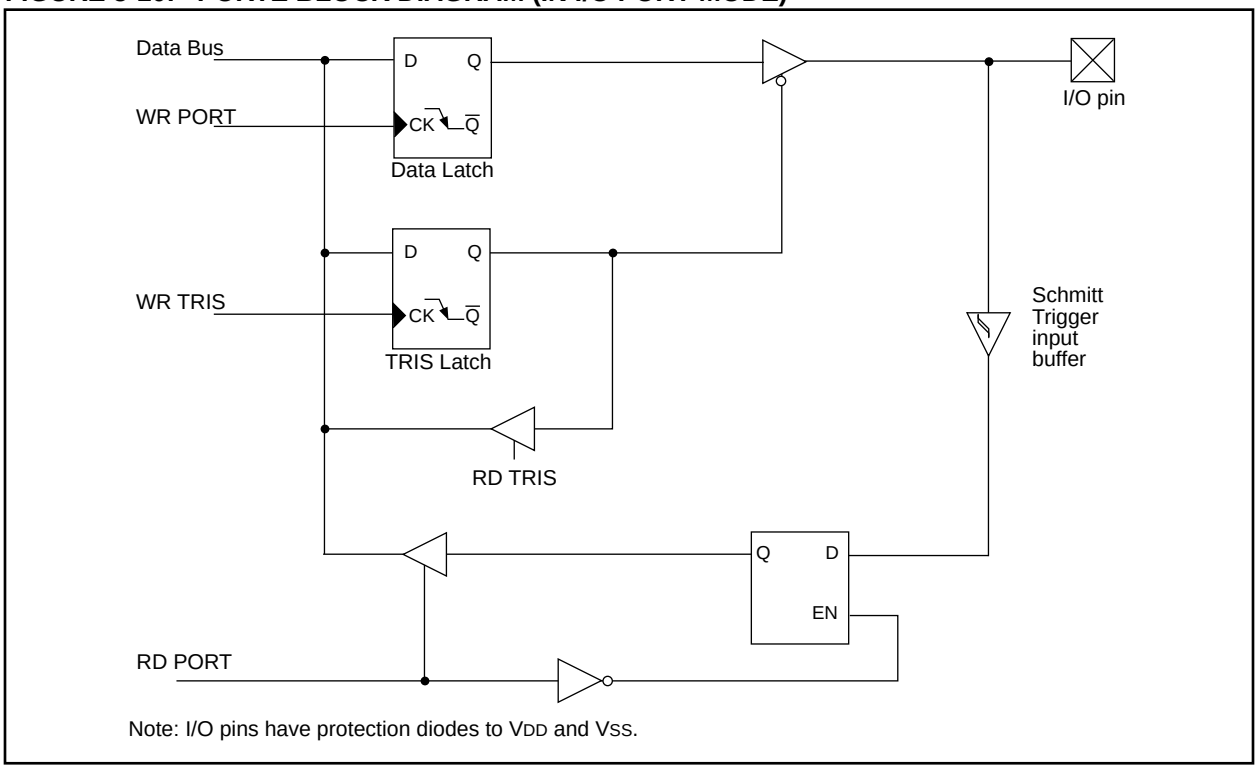
bit 3: **Unimplemented**: Read as '0'

bit 2: **TRISE2**: Direction control bit for pin RE2/$\overline{CS}$
1 = Input
0 = Output

bit 1: **TRISE1**: Direction control bit for pin RE1/$\overline{WR}$
1 = Input
0 = Output

bit 0: **TRISE0**: Direction control bit for pin RE0/$\overline{RD}$
1 = Input
0 = Output

FIGURE 5-10: PORTE BLOCK DIAGRAM (IN I/O PORT MODE)



Note: I/O pins have protection diodes to VDD and VSS.

TABLE 5-9: PORTE FUNCTIONS

Name	Bit#	Buffer Type	Function
RE0/$\overline{RD}$	bit0	ST/TTL(1)	Input/output port pin or read control input in parallel slave port mode: $\overline{RD}$ 1 = Not a read operation 0 = Read operation. Reads PORTD register (if chip selected)
RE1/$\overline{WR}$	bit1	ST/TTL(1)	Input/output port pin or write control input in parallel slave port mode: $\overline{WR}$ 1 = Not a write operation 0 = Write operation. Writes PORTD register (if chip selected)
RE2/$\overline{CS}$	bit2	ST/TTL(1)	Input/output port pin or chip select control input in parallel slave port mode: $\overline{CS}$ 1 = Device is not selected 0 = Device is selected

Legend: ST = Schmitt Trigger input, TTL = TTL input
Note 1: Input buffers are Schmitt Triggers when in I/O mode and TTL buffers when in Parallel Slave Port Mode.

TABLE 5-10: SUMMARY OF REGISTERS ASSOCIATED WITH PORTE

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other resets
09h	PORTE	—	—	—	—	—	RE2	RE1	RE0	---- -xxx	---- -uuu
89h	TRISE	IBF	OBF	IBOV	PSPMODE	—	TRISE2	TRISE1	TRISE0	0000 -111	0000 -111

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

5.6 I/O Programming Considerations

5.6.1 BI-DIRECTIONAL I/O PORTS

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

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

Example 5-4 shows the effect of two sequential read-modify-write instructions on an I/O port.

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

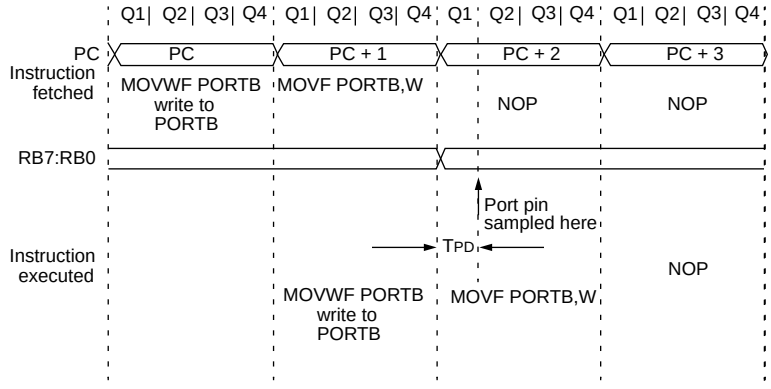
EXAMPLE 5-4: READ-MODIFY-WRITE INSTRUCTIONS ON AN I/O PORT

```
;Initial PORT settings: PORTB<7:4> Inputs
;                       PORTB<3:0> Outputs
;PORTB<7:6> have external pull-ups and are
;not connected to other circuitry
;
;                     PORT latch  PORT pins
;                     ----------  ---------
  BCF PORTB, 7     ; 01pp pppp    11pp pppp
  BCF PORTB, 6     ; 10pp pppp    11pp pppp
  BCF STATUS, RP1 ;
  BSF STATUS, RP0 ;
  BCF TRISB, 7     ; 10pp pppp    11pp pppp
  BCF TRISB, 6     ; 10pp pppp    10pp pppp
;
;Note that the user may have expected the
;pin values to be 00pp ppp. The 2nd BCF
;caused RB7 to be latched as the pin value
;(high).
```

5.6.2 SUCCESSIVE OPERATIONS ON I/O PORTS

The actual write to an I/O port happens at the end of an instruction cycle, whereas for reading, the data must be valid at the beginning of the instruction cycle (Figure 5-11). 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 an NOP or another instruction not accessing this I/O port.

FIGURE 5-11: SUCCESSIVE I/O OPERATION



Note:

This example shows a write to PORTB followed by a read from PORTB.

Note that:

data setup time = (0.25TCY - TPD)

where TCY = instruction cycle
TPD = propagation delay

Therefore, at higher clock frequencies, a write followed by a read may be problematic.

5.7 Parallel Slave Port (PIC16C661 and PIC16C662 only)

PORTD operates as an 8-bit wide parallel slave port, or as a microprocessor port when control bit PSPMODE (TRISE<4>) is set. In slave mode it is asynchronously readable and writable by the external world through $\overline{RD}$ control input pin (RE0/$\overline{RD}$) and $\overline{WR}$ control input pin (RE1/$\overline{WR}$).

It can directly interface to an 8-bit microprocessor data bus. The external microprocessor can read or write the PORTD latch as an 8-bit latch. Setting PSPMODE enables port pin RE0/$\overline{RD}$ to be the $\overline{RD}$ input, RE1/$\overline{WR}$ to be the $\overline{WR}$ input and RE2/$\overline{CS}$ to be the $\overline{CS}$ (chip select) input. For this functionality, the corresponding data direction bits of the TRISE register (TRISE<2:0>) must be configured as inputs (set).

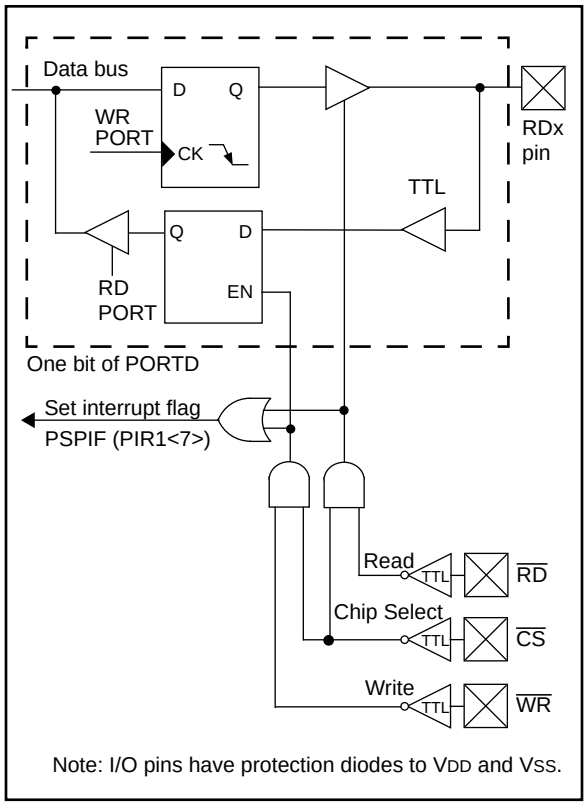
There are actually two 8-bit latches, one for data-out (from the PIC16/17) and one for data input. The user writes 8-bit data to PORTD data latch and reads data from the port pin latch (note that they have the same address). In this mode, the TRISD register is ignored since the microprocessor is controlling the direction of data flow.

Input Buffer Full Status Flag bit IBF (TRISE<7>) is set if a received word is waiting to be read by the CPU. Once the PORTD input latch is read, bit IBF is cleared. IBF is a read only status bit. Output Buffer Full Status Flag bit OBF (TRISE<6>) is set if a word written to PORTD latch is waiting to be read by the external bus. Once the PORTD output latch is read by the microprocessor, bit OBF is cleared. Input Buffer Overflow Status flag bit IBOV (TRISE<5>) is set if a second write to the microprocessor port is attempted when the previous word has not been read by the CPU (the first word is retained in the buffer).

When not in Parallel Slave Port mode, bits IBF and OBF are held clear. However, if flag bit IBOV was previously set, it must be cleared in software.

An interrupt is generated and latched into flag bit PSPIF (PIR1<7>) when a read or a write operation is completed. Flag bit PSPIF must be cleared by user software. The interrupt can be disabled by clearing the interrupt enable bit PSPIE (PIE1<7>).

FIGURE 5-12: PORTD AND PORTE AS A PARALLEL SLAVE PORT



Note: I/O pins have protection diodes to VDD and VSS.

TABLE 5-11: REGISTERS ASSOCIATED WITH PARALLEL SLAVE PORT

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other resets
08h	PORTD	PSP7	PSP6	PSP5	PSP4	PSP3	PSP2	PSP1	PSP0	xxxx xxxx	uuuu uuuu
09h	PORTE	—	—	—	—	—	RE2	RE1	RE0	---- -xxx	---- -uuu
89h	TRISE	IBF	OBF	IBOV	PSPMODE	—	TRISE2	TRISE1	TRISE0	0000 -111	0000 -111
0Ch	PIR1	PSPIF(1)	CMIF	—	—	—	—	—	—	00-- ----	00-- ----
8Ch	PIE1	PSPIE(1)	CMIE	—	—	—	—	—	—	00-- ----	00-- ----

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

Note 1: These bits are reserved on the PIC16C641/642, always maintain these bits clear.

NOTES:

6.0 TIMER0 MODULE

The Timer0 module has the following features:

- 8-bit timer/counter register, TMR0
  - Read and write capability
  - Interrupt on overflow from FFh to 00h
- 8-bit software programmable prescaler
- Internal or external clock select
  - Edge select for external clock

Figure 6-1 is a simplified block diagram of the Timer0 module.

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

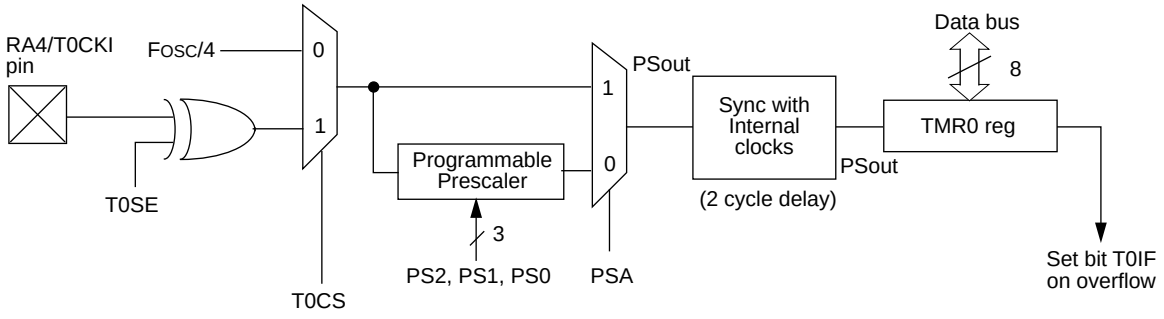
Counter mode is selected by setting bit T0CS. In this mode, Timer0 will increment either on every rising or falling edge of pin RA4/T0CKI. The incrementing edge is determined by the source edge select bit T0SE (OPTION<4>). Clearing bit T0SE selects the rising edge. Restrictions on the external clock input are discussed in detail in Section 6.2.

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

6.1 Timer0 Interrupt

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

FIGURE 6-1: TIMER0 BLOCK DIAGRAM



Note 1: Bits, T0CS, T0SE, PSA, and PS2, PS1, PS0 are (OPTION<5:0).
2: The prescaler is shared with Watchdog Timer (refer to Figure 6-6 for detailed diagram).

FIGURE 6-2: TIMER0 TIMING: INTERNAL CLOCK/NO PRESCALER

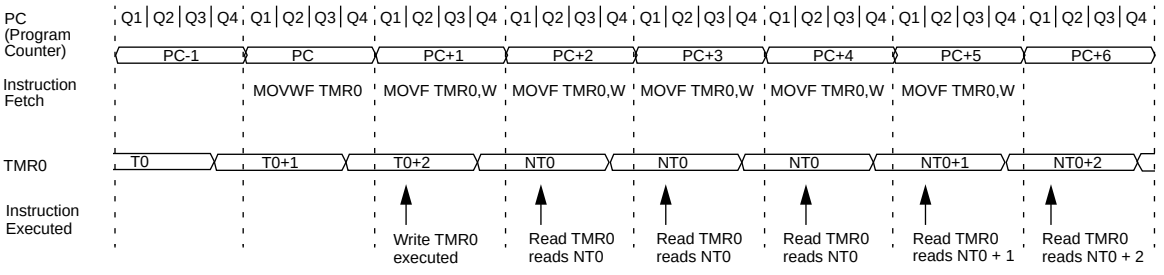


FIGURE 6-3: TIMER0 TIMING: INTERNAL CLOCK/PRESCALE 1:2

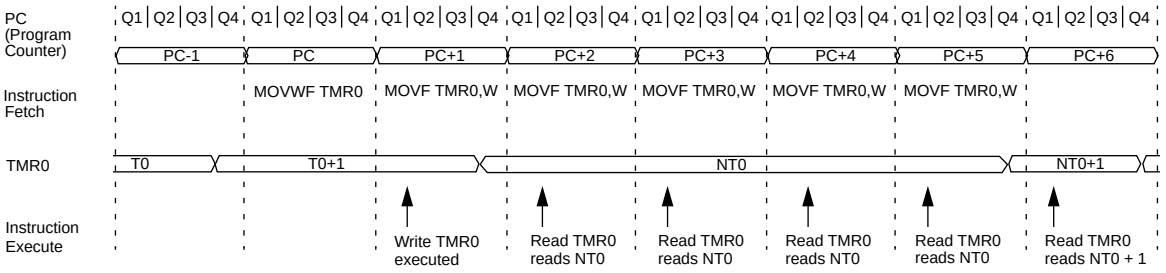
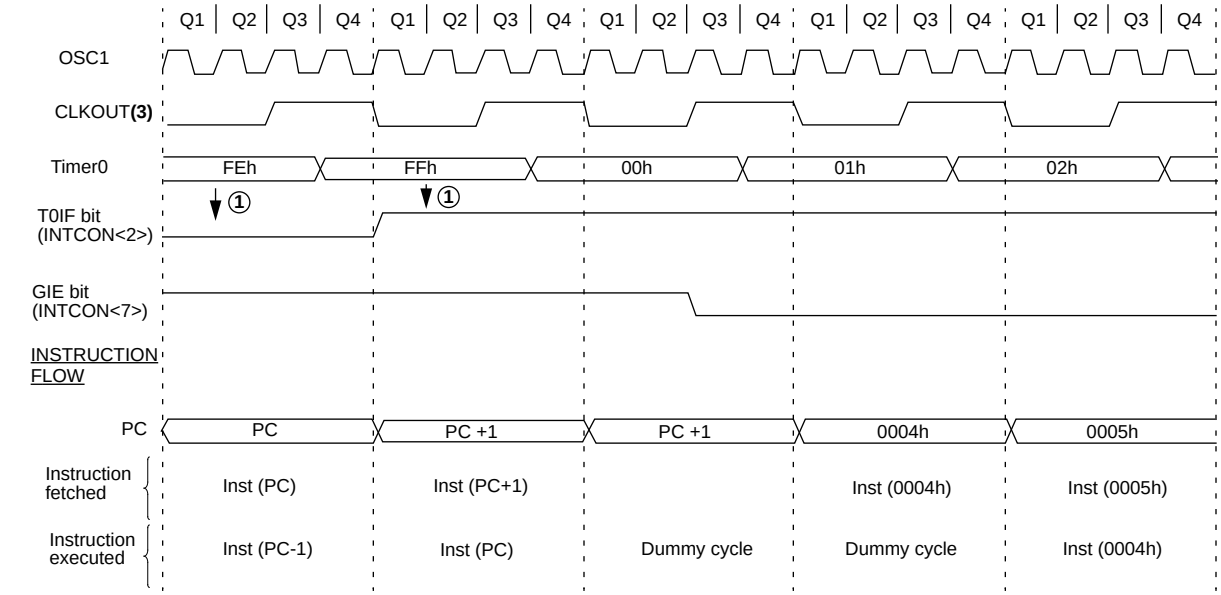


FIGURE 6-4: TIMER0 INTERRUPT TIMING



Note 1: Interrupt flag bit T0IF is sampled here (every Q1).
2: Interrupt latency = 4Tcy where Tcy = instruction cycle time.
3: CLKOUT is available only in RC oscillator mode.

6.2 Using Timer0 with External Clock

When an external clock input is used for Timer0, it must meet certain requirements. The requirements ensure the external clock can be synchronized with the internal phase clock (TOSC). Also, there is a delay in the actual incrementing of Timer0 after synchronization.

6.2.1 EXTERNAL CLOCK SYNCHRONIZATION

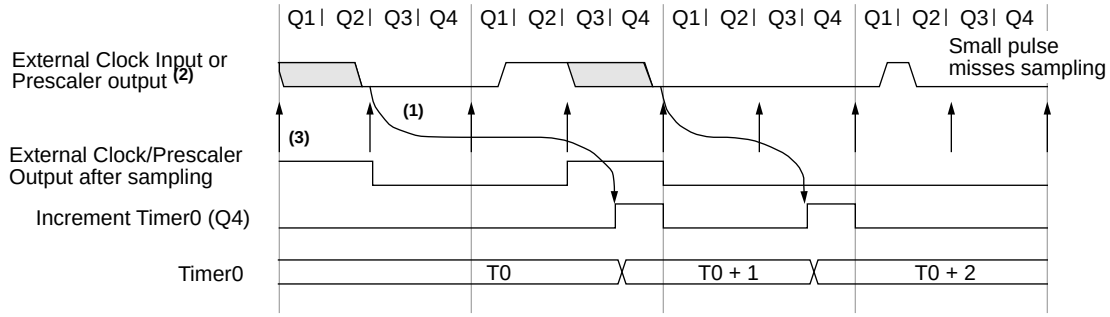
When no prescaler is used, the external clock input is the same as the prescaler output. The synchronization of T0CKI with the internal phase clocks is accomplished by sampling the prescaler output on the Q2 and Q4 cycles of the internal phase clocks (Figure 6-5). Therefore, it is necessary for T0CKI to be high for at least 2Tosc (and a small RC delay of 20 ns) and low for at least 2Tosc (and a small RC delay of 20 ns). Refer to the electrical specification of the desired device.

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

6.2.2 TIMER0 INCREMENT DELAY

Since the prescaler output is synchronized with the internal clocks, there is a small delay from the time the external clock edge occurs to the time the Timer0 module is actually incremented. Figure 6-5 shows the delay from the external clock edge to the timer incrementing.

FIGURE 6-5: TIMER0 TIMING WITH EXTERNAL CLOCK



Note 1: Delay from clock input change to Timer0 increment is 3Tosc to 7Tosc. (Duration of Q = Tosc). Therefore, the error in measuring the interval between two edges on Timer0 input = ±4Tosc max.
2: External clock if no prescaler selected, prescaler output otherwise.
3: The arrows indicate the points in time where sampling occurs.

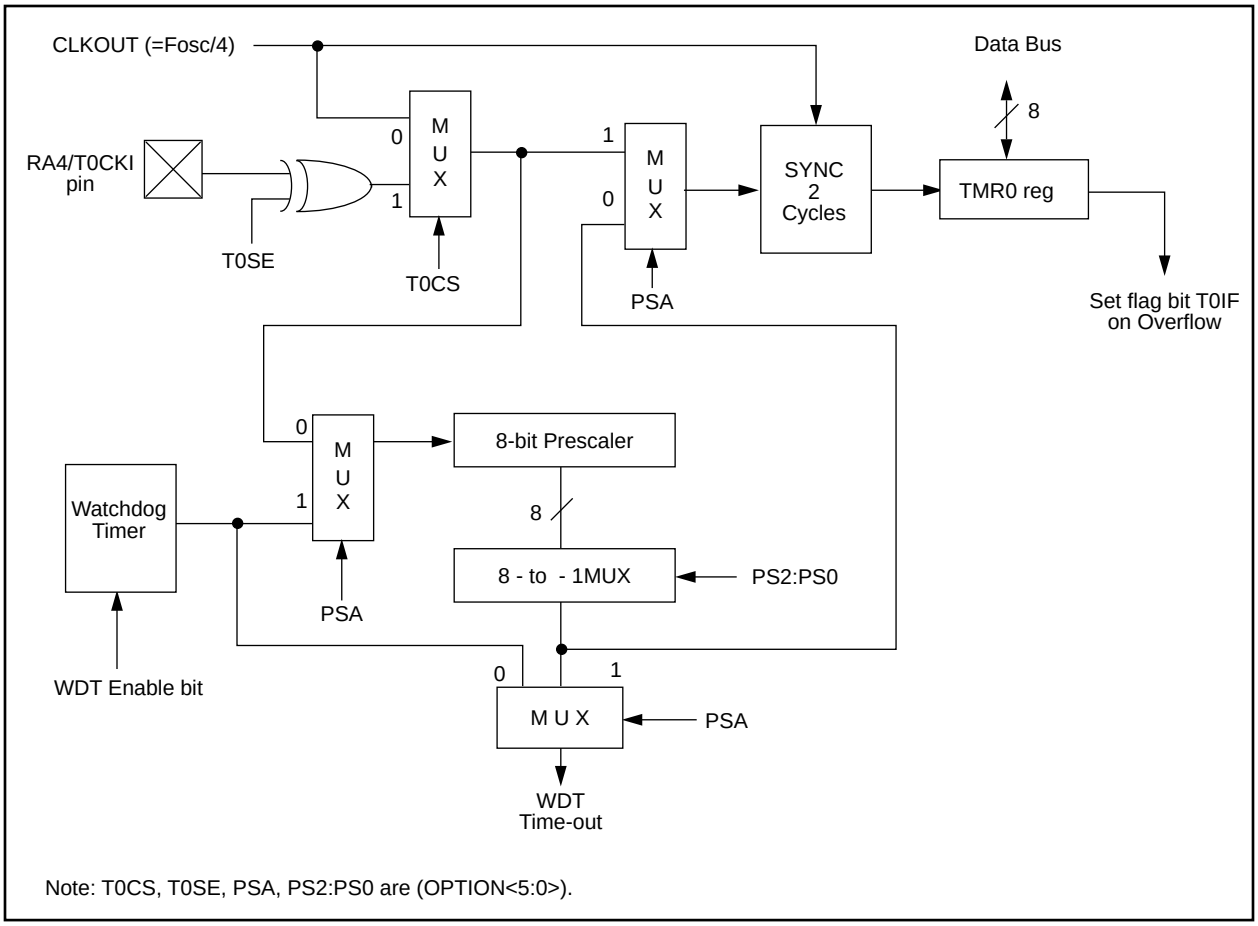
6.3 Prescaler

An 8-bit counter is available as a prescaler for the Timer0 module or as a postscaler for the Watchdog Timer (WDT), respectively (Figure 6-6). For simplicity, this counter is being referred to as “prescaler” throughout this data sheet. Note that the prescaler may be used by either the Timer0 module or the Watchdog Timer, but not both. Thus, a prescaler assignment for the Timer0 module means that there is no prescaler for the Watchdog Timer, and vice-versa.

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

When assigned to the Timer0 module, all instructions writing to the TMR0 register (e.g., `CLRF 1, MOVWF 1, BSF 1,x`) will clear the prescaler count. When assigned to Watchdog Timer, a `CLRWDT` instruction will clear the prescaler count along with the Watchdog Timer. The prescaler is not readable or writable.

FIGURE 6-6: BLOCK DIAGRAM OF THE TIMER0/WDT PRESCALER



Note: T0CS, T0SE, PSA, PS2:PS0 are (OPTION<5:0>).

6.3.1 SWITCHING PRESCALER ASSIGNMENT

The prescaler assignment is fully under software control, i.e., it can be changed “on the fly” during program execution.

Note:	To avoid an unintended device RESET, the following instruction sequence (shown in Example 6-1) must be executed when changing the prescaler assignment from Timer0 to the WDT. This precaution must be followed even if the WDT is disabled.

EXAMPLE 6-1: CHANGING PRESCALER (TIMER0→WDT)

```
BCF     STATUS, RP0  ;Bank 0
CLRF    TMR0         ;Clear TMR0 & Prescaler
BSF     STATUS, RP0  ;Bank 1
CLRWDT               ;Clears WDT
MOVLW   b'xxxx1xxx'  ;Select new prescale
MOVWF   OPTION_REG   ;value & WDT
BCF     STATUS, RP0  ;Bank 0
```

To change prescaler from the WDT to the Timer0 module, use the sequence shown in Example 6-2.

EXAMPLE 6-2: CHANGING PRESCALER (WDT→TIMER0)

```
CLRWDT               ;Clear WDT and
                     ;prescaler
BSF     STATUS, RP0  ;Bank 1
MOVLW   b'xxxx0xxx'  ;Select TMR0, new
                     ;prescale value and
MOVWF   OPTION_REG   ;clock source
BCF     STATUS, RP0  ;Bank 0
```

TABLE 6-1: REGISTERS ASSOCIATED WITH TIMER0

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other resets
01h	TMR0	Timer0 module’s register								xxxx xxxx	uuuu uuuu
0Bh/8Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
81h	OPTION	$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
85h	TRISA	—	—	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	--11 1111	--11 1111

Legend: x = unknown, u = unchanged, - = unimplemented locations read as '0'. Shaded cells are not used by Timer0.

NOTES:

7.0 COMPARATOR MODULE

The comparator module contains two analog comparators. The inputs to the comparators are multiplexed with pins RA0 through RA4. The on-chip Voltage Reference (Section 8.0) can also be an input to the comparators.

The CMCON register, shown in Figure 7-1, controls the comparator input and output multiplexers. A block diagram of the comparator is shown in Figure 7-2.

FIGURE 7-1: CMCON REGISTER (ADDRESS 1Fh)

R-0	R-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
C2OUT	C1OUT	—	—	CIS	CM2	CM1	CM0
bit7							bit0

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

bit 7: **C2OUT**: Comparator 2 output
1 = C2 V_{IN}+ > C2 V_{IN}–
0 = C2 V_{IN}+ < C2 V_{IN}–

bit 6: **C1OUT**: Comparator 1 output
1 = C1 V_{IN}+ > C1 V_{IN}–
0 = C1 V_{IN}+ < C1 V_{IN}–

bit 5-4: **Unimplemented:** Read as '0'

bit 3: **CIS**: Comparator Input Switch

When CM2:CM0: = 001:
Then:
1 = C1 V_{IN}– connects to RA3
0 = C1 V_{IN}– connects to RA0

When CM2:CM0 = 010:
Then:
1 = C1 V_{IN}– connects to RA3
C2 V_{IN}– connects to RA2
0 = C1 V_{IN}– connects to RA0
C2 V_{IN}– connects to RA1

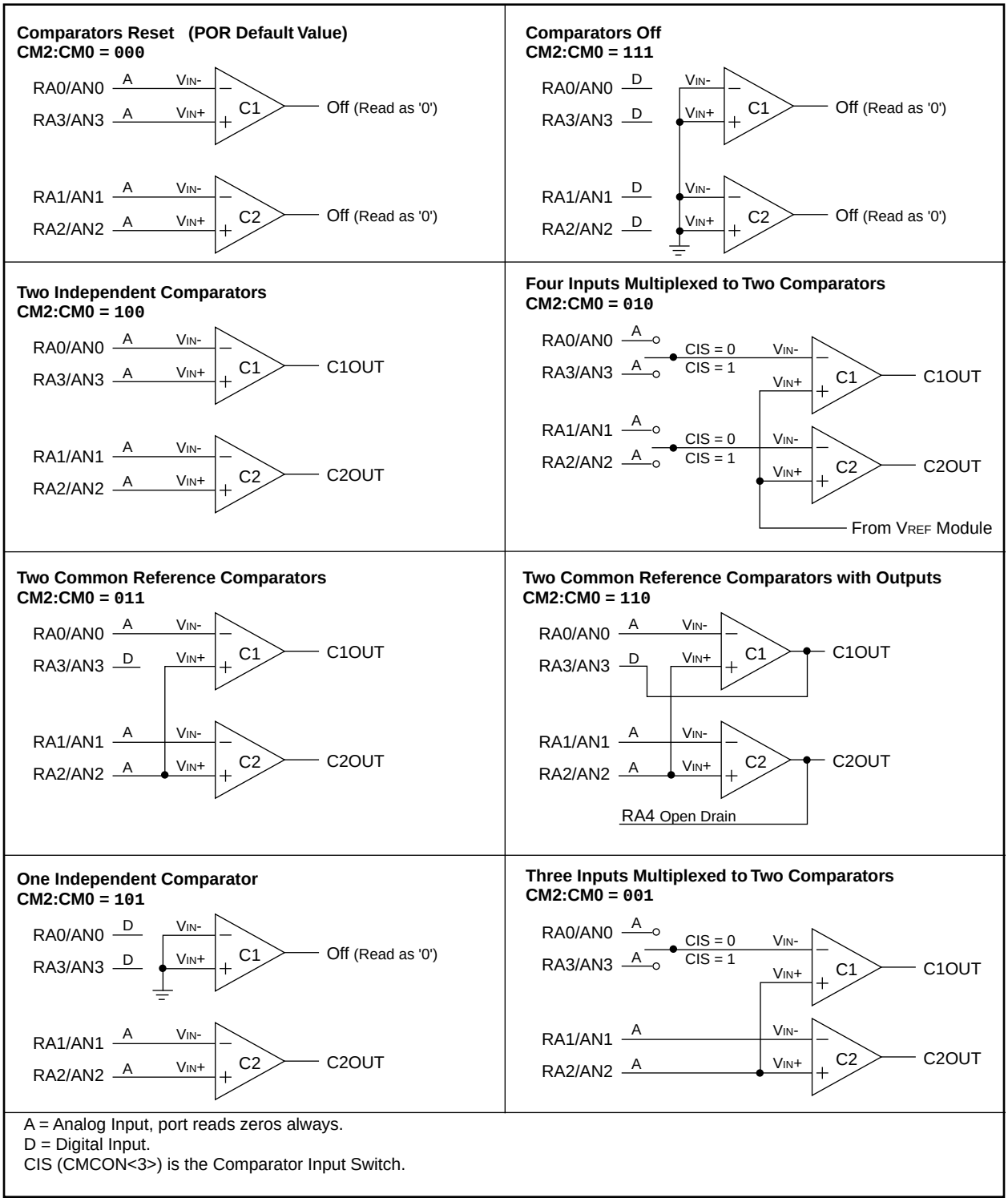
bit 2-0: **CM2:CM0**: Comparator mode
Figure 7-2 shows the comparator modes and CM2:CM0 bit settings.

7.1 Comparator Configuration

There are eight modes of operation for the comparators. The CMCON register is used to select the mode. Figure 7-2 shows the eight possible modes. The TRISA register controls the data direction of the comparator pins for each mode. If the comparator mode is changed, the comparator output level may not be valid for the specified mode change delay shown in Table 12-2.

Note:	Comparator interrupts should be disabled during a comparator mode change otherwise a false interrupt may occur.

FIGURE 7-2: COMPARATOR I/O OPERATING MODES



A = Analog Input, port reads zeros always.
D = Digital Input.
CIS (CMCON<3>) is the Comparator Input Switch.

The code example in Example 7-1 depicts the steps required to configure the comparator module. RA3 and RA4 are configured as digital outputs. RA0 and RA1 are configured as the V- inputs and RA2 as the V+ input to both comparators.

EXAMPLE 7-1: INITIALIZING THE COMPARATOR MODULE

```
FLAG_REG  EQU 0x20
CLRF      FLAG_REG     ;Init Flag Register
CLRF      PORTA        ;Init PORTA
ANDLW     0xC0         ;Mask Comp bits
IORWF     FLAG_REG,F   ;Bits to Flag_Reg
MOVLW     0x03         ;Init Comp Mode
MOVWF     CMCON        ;CM2:CM0 = 011
BSF       STATUS,RP0   ;Select Bank 1
MOVLW     0x07         ;Init Data direction
MOVWF     TRISA        ;RA<2:0> to inputs
                       ;RA<4:3> to outputs
                       ;TRISA<7:5> read '0'
BCF       STATUS,RP0   ;Select Bank 0
CALL      DELAY_10µs   ;10 µs delay
MOVF      CMCON,F      ;Read CMCON to end
                       ;change condition
BCF       PIR1,CMIF    ;Clear Pending Ints
BSF       STATUS,RP0   ;Select Bank 1
BSF       PIE1,CMIE    ;Enable Comp Ints
BCF       STATUS,RP0   ;Select Bank 0
BSF       INTCON,PEIE  ;Enable Periph Ints
BSF       INTCON,GIE   ;Global Int enable
```

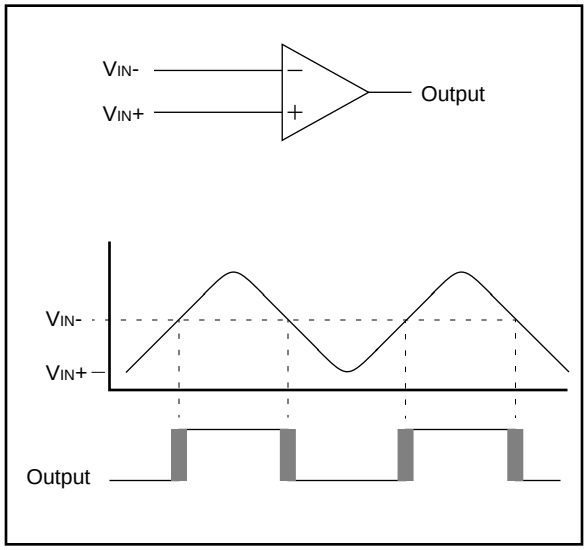
7.2 Comparator Operation

A single comparator is shown in Figure 7-3 along with the relationship between the analog input levels and the digital output. When the analog input at $V_{IN}+$ is less than the analog input $V_{IN}-$, the output of the comparator is a digital low level. When the analog input at $V_{IN}+$ is greater than the analog input $V_{IN}-$, the output of the comparator is a digital high level. The shaded areas of the output of the comparator in Figure 7-3 represents the uncertainty due to input offsets and response time.

7.3 Comparator Reference

An external or internal reference signal may be used depending on the comparator operating mode. The analog signal that is present at $V_{IN}-$ is compared to the signal at $V_{IN}+$, and the digital output of the comparator is adjusted accordingly (Figure 7-3).

FIGURE 7-3: SINGLE COMPARATOR



7.3.1 EXTERNAL REFERENCE SIGNAL

When external voltage references are used, the comparator module can be configured to have the comparators operate from the same or different reference sources. However, threshold detector applications may require the same reference. The reference signal must be between V_{SS} and V_{DD}, and can be applied to either pin of the comparator(s).

7.3.2 INTERNAL REFERENCE SIGNAL

The comparator module also allows the selection of an internally generated voltage reference for the comparators. Section 8.0, contains a detailed description of the Voltage Reference Module that provides this signal. The internal reference signal is used when the comparators are in mode CM2:CM0 = 010 (Figure 7-2). In this mode, the internal voltage reference is applied to the $V_{IN}+$ pin of both comparators.

7.4 Comparator Response Time

Response time is the minimum time, after selecting a new reference voltage or input source, before the comparator output is guaranteed to have a valid level. If the internal reference is changed, the maximum delay of the internal voltage reference must be considered when using the comparator outputs. Otherwise, the maximum delay of the comparators should be used (Table 12-2 and Table 12-3).

7.5 Comparator Outputs

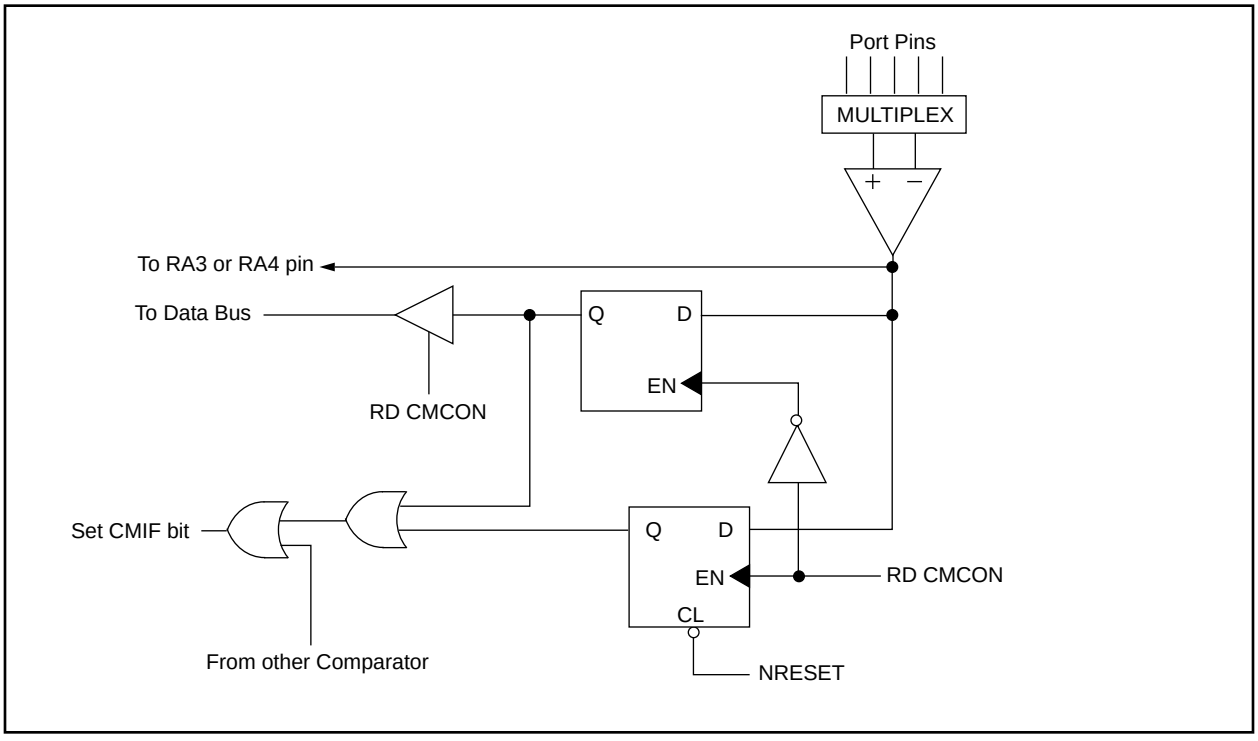
The comparator outputs are read through the CMCON register. These bits are read only. The comparator outputs may also be directly output to the RA3 and RA4 I/O pins. When CM2:CM0 = 110, multiplexors in the output path of the RA3 and RA4 pins will switch and the output of each pin will be the unsynchronized output of the comparator. The uncertainty of each of the comparators is related to the input offset voltage and the response time given in the specifications. Figure 7-4 shows the comparator output block diagram.

The TRISA bits will still function as an output enable/ disable for the RA3 and RA4 pins while in this mode.

Note 1: When reading the PORTA register, all pins configured as analog inputs will read as a '0'. Pins configured as digital inputs will convert an analog input according to the Schmitt Trigger input specification.

Note 2: Analog levels on any pin that is defined as a digital input may cause the input buffer to consume more current than is specified.

FIGURE 7-4: COMPARATOR OUTPUT BLOCK DIAGRAM



7.6 Comparator Interrupts

The comparator interrupt flag is set whenever there is a change in the output value of either comparator. User software will need to maintain information about the status of the output bits, as read from CMCON<7:6>, to determine the actual change that has occurred. The CMIF bit (PIR1<6>), is the comparator interrupt flag and must be cleared in user software.

To enable the Comparator interrupt the following bits must be set:

- CMIE (PIE1<6>)
- PEIE (INTCON<6>)
- GIE (INTCON<7>)

The user, in the interrupt service routine, can clear the interrupt in the following manner:

a) Any read or write of CMCON. This will end the mismatch condition.

b) Clear flag bit CMIF.

A mismatch condition will continue to set flag bit CMIF. Reading CMCON will end the mismatch condition, and allow flag bit CMIF to be cleared.

7.7 Comparator Operation During SLEEP

When a comparator is active and the device is placed in SLEEP mode, the comparator remains active and the interrupt is functional if enabled. This interrupt will wake up the device from SLEEP mode when enabled. While the comparator is powered up, higher sleep currents than shown in the power-down current specification will occur. Each comparator that is operational will consume additional current as shown in the comparator specifications. To minimize power consumption while in SLEEP mode, turn off the comparators, CM2:CM0 = 111, before entering sleep. If the device wakes up from sleep, the contents of the CMCON register are not affected.

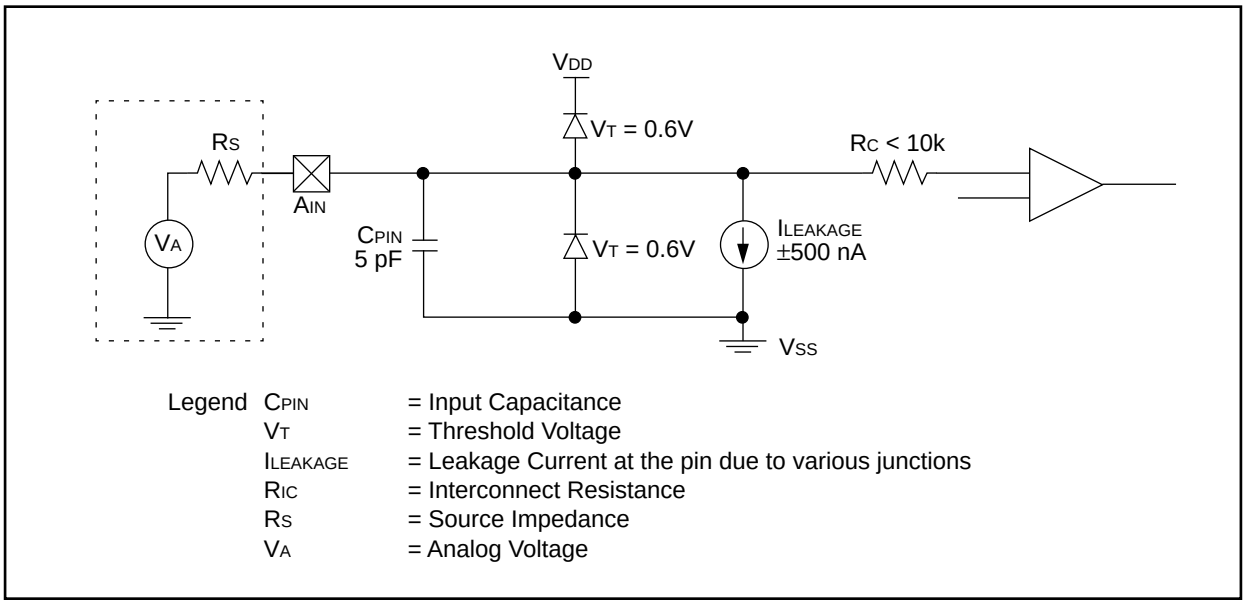
7.8 Effects of a RESET

A device reset forces the CMCON register to its reset state. This forces the comparator module to be in the comparator reset mode, CM2:CM0 = 000. This ensures that all potential inputs are analog inputs. Device current is minimized when analog inputs are present at reset time. The comparators will be powered down during the reset interval.

7.9 Analog Input Connection Considerations

A simplified circuit for an analog input is shown in Figure 7-5. Since the analog pins are connected to a digital output, they have reverse biased diodes to V_{DD} and V_{SS}. The analog input therefore, must be between V_{SS} and V_{DD}. If the input voltage deviates from this range by more than 0.6V in either direction, one of the diodes is forward biased and a latch-up may occur. A maximum source impedance of 10 kΩ is recommended for the analog sources. Any external component connected to an analog input pin, such as a capacitor or a Zener diode, should have very little leakage current.

FIGURE 7-5: ANALOG INPUT MODEL



Legend
- C_{PIN} = Input Capacitance
- V_T = Threshold Voltage
- $I_{LEAKAGE}$ = Leakage Current at the pin due to various junctions
- R_{IC} = Interconnect Resistance
- R_S = Source Impedance
- V_A = Analog Voltage

TABLE 7-1: REGISTERS ASSOCIATED WITH THE COMPARATOR MODULE

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other resets
1Fh	CMCON	C2OUT	C1OUT	—	—	CIS	CM2	CM1	CM0	00-- 0000	00-- 0000
9Fh	VRCON	VREN	VROE	VRR	—	VR3	VR2	VR1	VR0	000- 0000	000- 0000
0Bh/8Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF(1)	CMIF	—	—	—	—	—	—	00-- ----	00-- ----
8Ch	PIE1	PSPIE(1)	CMIE	—	—	—	—	—	—	00-- ----	00-- ----
85h	TRISA	—	—	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	--11 1111	--11 1111

Note 1: These bits are reserved on the PIC16C641/642, always maintain these bits clear.

8.0 VOLTAGE REFERENCE MODULE

The Voltage Reference is a 16-tap resistor ladder network that provides a selectable voltage reference. The resistor ladder is segmented to provide two ranges of VREF values and has a power-down function to conserve power when the reference module is not being used.

The VRCON register, shown in Figure 8-1, controls the operation of the Voltage Reference Module. The block diagram is given in Figure 8-2.

FIGURE 8-1: VRCON REGISTER (ADDRESS 9Fh)

R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
VREN	VROE	VRR	—	VR3	VR2	VR1	VR0
bit7							bit0

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

bit 7: **VREN:** VREF Enable
1 = VREF circuit powered up
0 = VREF circuit powered down, no IDD drain

bit 6: **VROE:** VREF Output Enable
1 = VREF is output on RA2 pin
0 = VREF is disconnected from RA2 pin

bit 5: **VRR:** VREF Range selection
1 = Low Range
0 = High Range

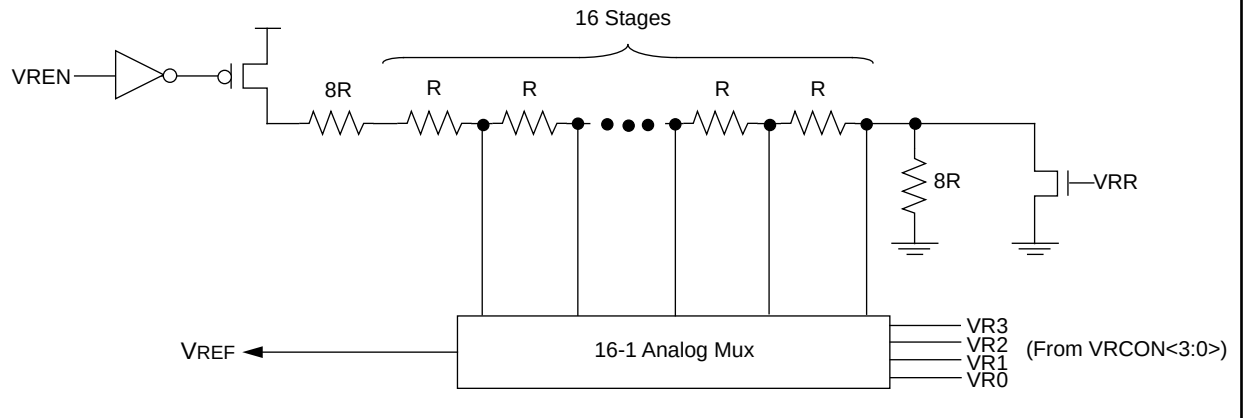
bit 4: **Unimplemented:** Read as '0'

bit 3-0: **VR3:VR0**: VREF value selection 0 ≤ VR3:VR0 ≤ 15

When: VRR = 1
Then: VREF = (VR3:VR0/ 24) • VDD

When: VRR = 0
Then: VREF = 1/4 • VDD + (VR3:VR0/ 32) • VDD

FIGURE 8-2: VOLTAGE REFERENCE BLOCK DIAGRAM



Note: R is defined in Table 12-3.

8.1 Configuring the Voltage Reference

The Voltage Reference Module can output 16 distinct voltage levels for each range.

The equations used to calculate the output of the Voltage Reference are as follows:

If VRR = 1
Then $V_{REF} = (VR3:VR0/24) \cdot V_{DD}$

If VRR = 0
Then $V_{REF} = (V_{DD} \cdot 1/4) + (VR3:VR0/32) \cdot V_{DD}$

The settling time of the Voltage Reference must be considered when changing the VREF output (Table 12-2). Example 8-1 shows an example of how to configure the Voltage Reference for an output voltage of 1.25V with VDD = 5.0V.

EXAMPLE 8-1: VOLTAGE REFERENCE CONFIGURATION

```
MOVLW   0x02          ; 4 inputs muxed
MOVWF   CMCON         ; to 2 comparators
BSF     STATUS,RP0    ; Select Bank 1
MOVLW   0x07          ; RA3:RA0 to outputs
MOVWF   TRISA         ;
MOVLW   0xA6          ; enable Vref low
MOVWF   VRCON         ; range, VR3:VR0 = 6
BCF     STATUS,RP0    ; Select Bank 0
CALL    DELAY_10µs      ; 10 µs delay
```

8.2 Voltage Reference Accuracy/Error

The full range of VSS to VDD cannot be realized due to the construction of the module. The transistors on the top and bottom of the resistor ladder network (Figure 8-2) keep VREF from approaching VSS or VDD. The Voltage Reference is VDD derived and therefore, the VREF output changes with fluctuations in VDD. The absolute accuracy of the Voltage Reference can be found in Table 12-3.

8.3 Operation During Sleep

When the device wakes up from sleep through an interrupt or a Watchdog Timer time-out, the contents of the VRCON register are not affected. To minimize current consumption in SLEEP mode, the Voltage Reference Module should be disabled.

8.4 Effects of a Reset

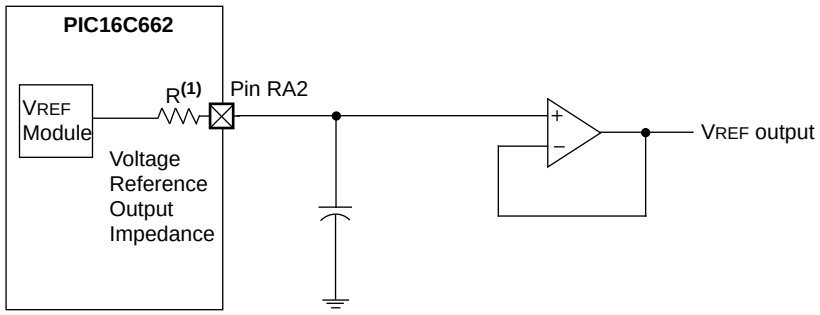
A device reset disables the Voltage Reference by clearing bit VREN (VRCON<7>). This reset also disconnects the reference from the RA2 pin by clearing bit VROE (VRCON<6>) and selects the high voltage range by clearing bit VRR (VRCON<5>). The VREF value select bits, VRCON<3:0>, are also cleared.

8.5 Connection Considerations

The Voltage Reference Module operates independently of the comparator module. The output of the reference generator may be connected to the RA2 pin if the TRISA<2> bit is set and bit VROE is set. Enabling the Voltage Reference output onto the RA2 pin with an input signal present will increase current consumption. Connecting RA2 as a digital output with VREF enabled will also increase current consumption.

The RA2 pin can be used as a simple D/A output with limited drive capability. Due to the limited drive capability, a buffer must be used in conjunction with the Voltage Reference output for external connections to VREF. Figure 8-3 shows an example buffering technique.

FIGURE 8-3: VOLTAGE REFERENCE OUTPUT BUFFER EXAMPLE



Note 1: R is dependent upon the Voltage Reference Configuration VRCON<3:0> and VRCON<5>.

TABLE 8-1: REGISTERS ASSOCIATED WITH VOLTAGE REFERENCE

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value On POR, BOR	Value on all other resets
9Fh	VRCON	VREN	VROE	VRR	—	VR3	VR2	VR1	VR0	000- 0000	000- 0000
1Fh	CMCON	C2OUT	C1OUT	—	—	CIS	CM2	CM1	CM0	00-- 0000	00-- 0000
85h	TRISA	—	—	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	--11 1111	--11 1111

9.0 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 PIC16C64X & PIC16C66X families have a host of such features intended to maximize system reliability, minimize cost through elimination of external components, provide power saving operating modes and offer code protection.

These are:

1. Oscillator selection
2. Resets
   Power-on Reset (POR)
   Power-up Timer (PWRT)
   Oscillator Start-up Timer (OST)
   Brown-out Reset (BOR)
   Parity Error Reset (PER)
3. Interrupts
4. Watchdog Timer (WDT)
5. SLEEP
6. Code protection
7. ID Locations
8. In-circuit serial programming

The PIC16C64X & PIC16C66X has a Watchdog Timer which is enabled by a configuration bit (WDTE). 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. Circuitry has been provided for checking program memory parity with a reset when an error is indicated. There is also circuitry to reset the device if a brown-out occurs which provides at least a 72 ms reset. With these three functions on-chip, most applications need no external reset circuitry.

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 wake-up or through an interrupt. Several oscillator options are also made available to allow the part to fit the application. The RC oscillator option saves system cost while the LP crystal option saves power. A set of configuration bits are used to select various options.

9.1 Configuration Bits

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

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), which can be accessed only during programming.

FIGURE 9-1: CONFIGURATION WORD

CP1	CP0	CP1	CP0	CP1	CP0	MPEEN	BODEN	CP1	CP0	$\overline{\text{PWRTE}}$	WDTE	FOSC1	FOSC0
bit13													bit0

CONFIG REGISTER: Address 2007h

bit 13-8 5-4: **CP1:CP0:** Code protection bits(2)
- 11 = Code protection off
- 10 = Upper half of program memory code protected
- 01 = Upper 3/4th of program memory code protected
- 00 = All memory is code protected

bit 7: **MPEEN**: Memory Parity Error Enable
- 1 = Memory Parity Checking is enabled
- 0 = Memory Parity Checking is disabled

bit 6: **BODEN**: Brown-out Reset Enable bit (1)
- 1 = BOR enabled
- 0 = BOR disabled

bit 3: **$\overline{\text{PWRTE}}$**: Power-up Timer Enable bit (1)
- 1 = PWRT disabled
- 0 = PWRT enabled

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

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

Note 1: Enabling Brown-out Reset automatically enables the Power-up Timer (PWRT) regardless of the value of bit $\overline{\text{PWRTE}}$. Ensure the Power-up Timer is enabled anytime Brown-out Reset is enabled.

2: All of the CP1:CP0 pairs have to be given the same value to enable the code protection scheme listed.

9.2 Oscillator Configurations

9.2.1 OSCILLATOR TYPES

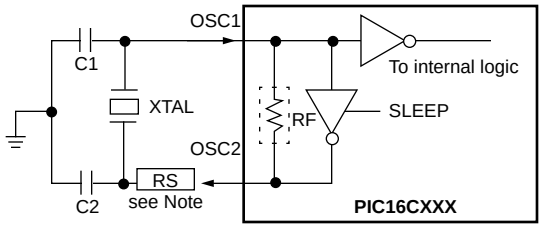
The PIC16CXXX can be operated in four different oscillator modes. The user can program two configuration bits (FOSC1 and FOSC0) to select one of these four modes:

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

9.2.2 CRYSTAL OSCILLATOR / CERAMIC RESONATORS

In XT, LP or HS modes a crystal or ceramic resonator is connected to the OSC1 and OSC2 pins to establish oscillation (Figure 9-2). The PIC16CXXX oscillator design requires the use of a parallel cut crystal. Use of a series cut crystal may give a frequency out of the crystal manufacturers specifications. When in XT, LP or HS modes, the device can have an external clock source to drive the OSC1 pin (Figure 9-3).

FIGURE 9-2: CRYSTAL OPERATION (OR CERAMIC RESONATOR) (HS, XT OR LP OSC CONFIGURATION)



See Table 9-1 or Table 9-2 for recommended values of C1 and C2.

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

FIGURE 9-3: EXTERNAL CLOCK INPUT OPERATION (HS, XT OR LP OSC CONFIGURATION)

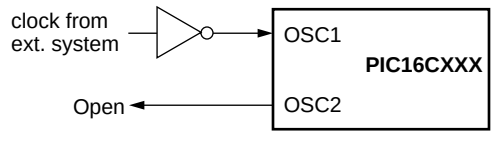


TABLE 9-1: CAPACITOR SELECTION FOR CERAMIC RESONATORS (PRELIMINARY)

Ranges tested:		
Mode	**Freq**	**OSC1**
XT	455 kHz 2.0 MHz 4.0 MHz	22 - 100 pF 15 - 68 pF 15 - 68 pF
HS	8.0 MHz 16.0 MHz	10 - 68 pF 10 - 22 pF

Note: Recommended values of C1 and C2 are identical to the ranges tested table.
Higher capacitance increases the stability of the oscillator but also increases the start-up time. These values are for design guidance only. Since each resonator has its own characteristics, the user should consult the resonator manufacturer for appropriate values of external components.

Resonators used:		
455 kHz	Panasonic EFO-A455K04B	±0.3%
2.0 MHz	Murata Erie CSA2.00MG	±0.5%
4.0 MHz	Murata Erie CSA4.00MG	±0.5%
8.0 MHz	Murata Erie CSA8.00MT	±0.5%
16.0 MHz	Murata Erie CSA16.00MX	±0.5%

All resonators used did not have built-in capacitors.

TABLE 9-2: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR (PRELIMINARY)

Mode	Freq	OSC1	OSC2
LP	32 kHz 200 kHz	68 - 100 pF 15 - 30 pF	68 - 100 pF 15 - 30 pF
XT	100 kHz 2 MHz 4 MHz	68 - 150 pF 15 - 30 pF 15 - 30 pF	150 - 200 pF 15 - 30 pF 15 - 30 pF
HS	8 MHz 10 MHz 20 MHz	15 - 30 pF 15 - 30 pF 15 - 30 pF	15 - 30 pF 15 - 30 pF 15 - 30 pF

Higher capacitance increases the stability of the 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.

Crystals used:		
32.768 kHz	Epson C-001R32.768K-A	± 20 PPM
100 kHz	Epson C-2 100.00 KC-P	± 20 PPM
200 kHz	STD XTL 200.000 kHz	± 20 PPM
2.0 MHz	ECS ECS-20-S-2	± 50 PPM
4.0 MHz	ECS ECS-40-S-4	± 50 PPM
10.0 MHz	ECS ECS-100-S-4	± 50 PPM
20.0 MHz	ECS ECS-200-S-4	± 50 PPM

9.2.3 EXTERNAL CRYSTAL OSCILLATOR CIRCUIT

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

Figure 9-4 shows implementation of a parallel resonant oscillator circuit. The circuit is designed to use the fundamental frequency of the crystal. The 74AS04 inverter performs the 180-degree phase shift that a parallel oscillator requires. The 4.7 kΩ resistor provides the negative feedback for stability. The 10 kΩ potentiometer biases the 74AS04 in the linear region. This could be used for external oscillator designs.

FIGURE 9-4: EXTERNAL PARALLEL RESONANT CRYSTAL OSCILLATOR CIRCUIT

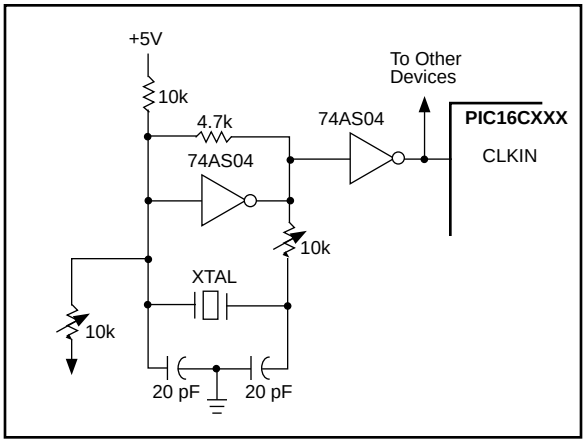
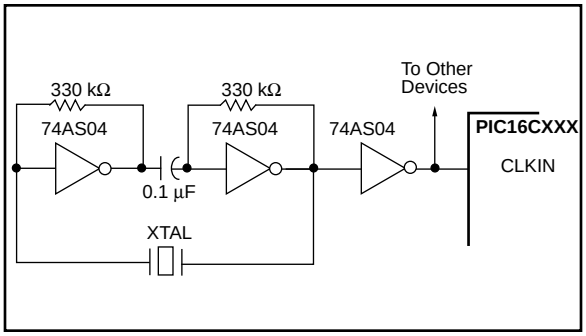


Figure 9-5 shows a series resonant oscillator circuit. This circuit is also designed to use the fundamental frequency of the crystal. The inverter performs a 180-degree phase shift in a series resonant oscillator circuit. The 330 kΩ resistors provide the negative feedback to bias the inverters in their linear region.

FIGURE 9-5: EXTERNAL SERIES RESONANT CRYSTAL OSCILLATOR CIRCUIT



9.2.4 RC OSCILLATOR

For timing insensitive applications the "RC" device option offers additional cost savings. The RC oscillator frequency is a function of the supply voltage, the resistor (Rext) and capacitor (Cext) values, and the operating temperature. In addition to this, the oscillator frequency will vary from unit to unit due to normal process parameter variation. Furthermore, the difference in lead frame capacitance between package types will also affect the oscillation frequency, especially for low Cext values. The user also needs to take into account variation due to tolerance of external R and C components used. Figure 9-6 shows how the R/C combination is connected to the PIC16CXXX. For Rext values below 2.2 kΩ, the oscillator operation may become unstable, or stop completely. For very high Rext values (e.g. 1 MΩ), the oscillator becomes sensitive to noise, humidity and leakage. Thus, we recommend to keep Rext between 3 kΩ and 100 kΩ.

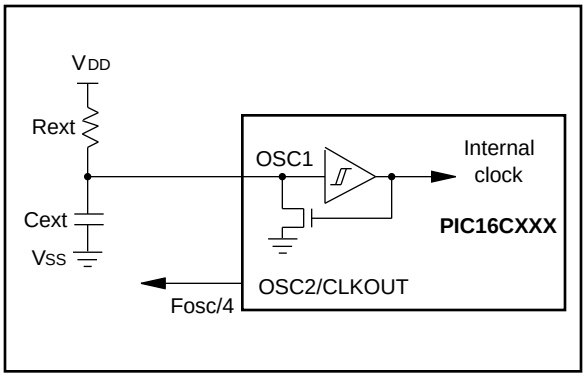
Although the oscillator will operate with no external capacitor (Cext = 0 pF), we recommend using values above 20 pF for noise and stability reasons. With no or small external capacitance, the oscillation frequency can vary dramatically due to changes in external capacitances, such as PCB trace capacitance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

See characterization data for desired device 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 characterization data for desired device for variation of oscillator frequency due to VDD for given Rext/Cext values as well as frequency variation due to operating temperature for given R, C, and VDD values.

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

FIGURE 9-6: RC OSCILLATOR MODE



9.3 Reset

The PIC16CXXX differentiates between various kinds of reset:

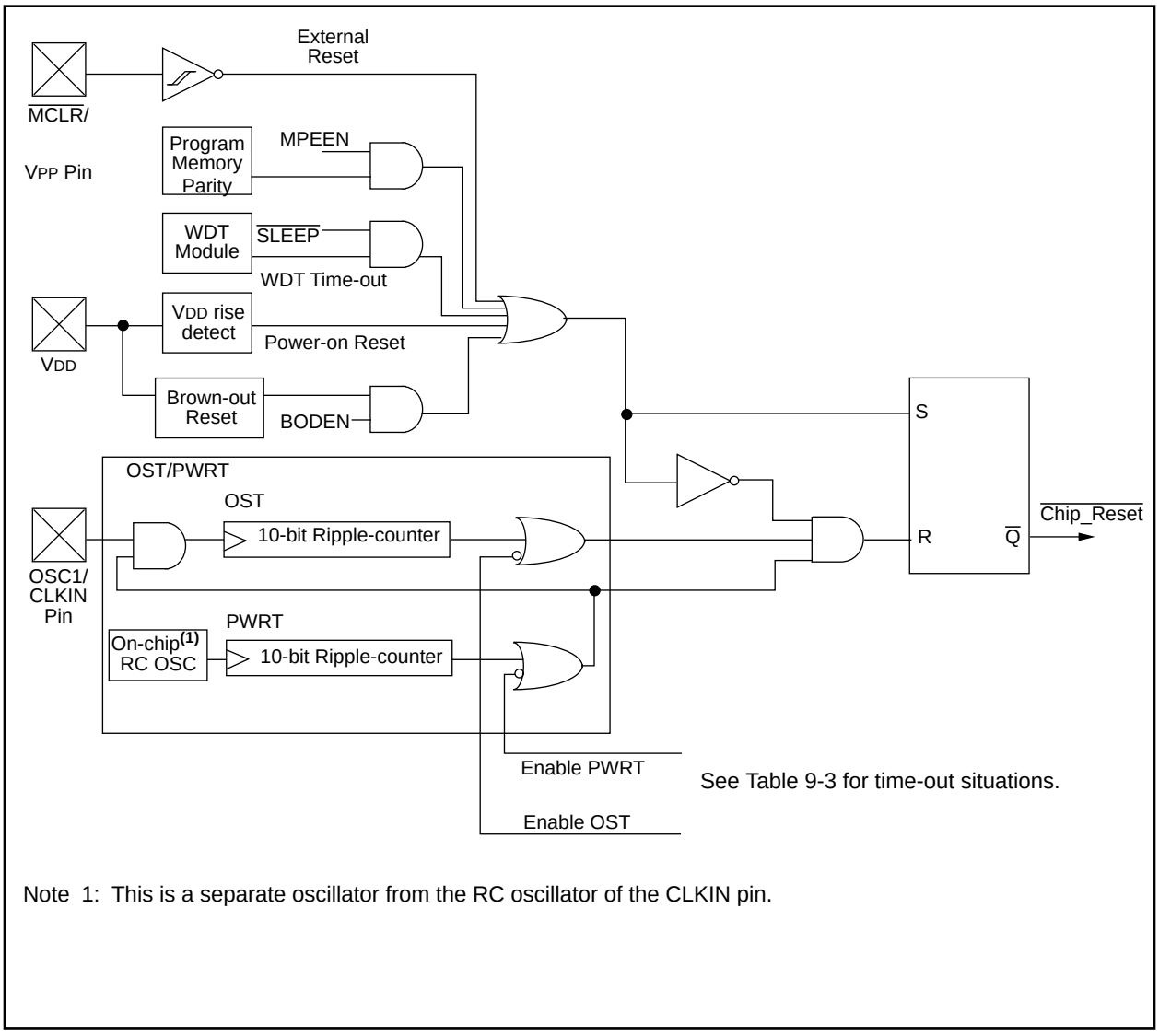
a) Power-on reset (POR)
b) $\overline{MCLR}$ reset during normal operation
c) $\overline{MCLR}$ reset during SLEEP
d) WDT reset (normal operation)
e) Brown-out Reset (BOR)
f) Parity Error Reset (PER)

Some registers are not affected in any reset condition; their status is unknown on POR and unchanged in any other reset. Most other registers are reset to a "reset state" on Power-on reset, $\overline{MCLR}$, WDT reset, Brown-out Reset, Parity Error Reset, and on $\overline{MCLR}$ reset during SLEEP. They are not affected by a WDT wake-up, since this is viewed as the resumption of normal operation. $\overline{TO}$ and $\overline{PD}$ bits are set or cleared differently in different reset situations as indicated in Table 9-4. These bits are used in software to determine the nature of the reset. See Table 9-6 for a full description of reset states of all registers.

A simplified block diagram of the on-chip reset circuit is shown in Figure 9-7.

The $\overline{MCLR}$ reset path has a noise filter to detect and ignore small pulses. See Table 12-6 for pulse width specification.

FIGURE 9-7: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT



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

9.4 Power-on Reset (POR), Power-up Timer (PWRT), Oscillator Start-up Timer (OST), Brown-out Reset (BOR), and Parity Error Reset (PER)

9.4.1 POWER-ON RESET (POR)

A Power-on Reset pulse is generated on-chip when VDD rise is detected (in the range of 1.6V to 1.8V). To take advantage of the POR, just tie the $\overline{MCLR}$ pin directly (or through a resistor) to VDD. This will eliminate external RC components usually needed to create a Power-on Reset. A maximum rise time for VDD is required. See Electrical Specifications for details.

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

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

9.4.2 POWER-UP TIMER (PWRT)

The Power-up Timer provides a fixed 72 ms (nominal) delay on power-up only, from POR or BOR. 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 VDD to rise to an acceptable level. A configuration bit, $\overline{PWRTE}$ can disable (if set) or enable (if cleared or programmed) the Power-up Timer. The Power-up Timer should always be enabled when Brown-out Reset is enabled.

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

9.4.3 OSCILLATOR START-UP TIMER (OST)

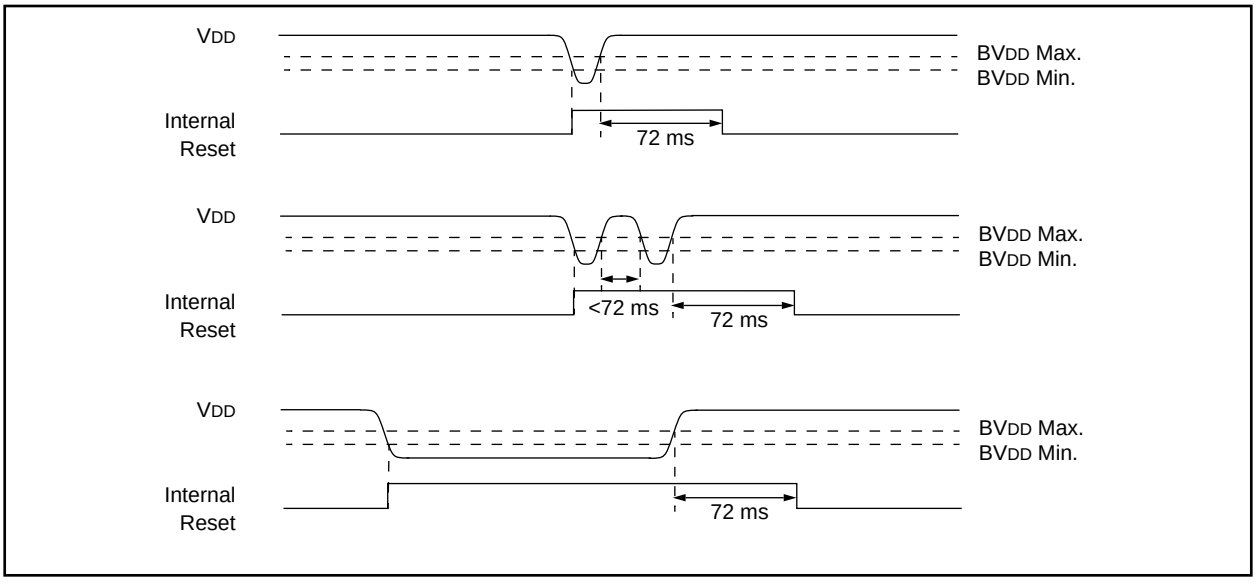
The Oscillator Start-Up Timer (OST) provides a 1024 oscillator cycle (from OSC1 input) delay after the PWRT delay is over. This ensures that the crystal oscillator or resonator has started and stabilized.

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

9.4.4 BROWN-OUT RESET (BOR)

PIC16C64X & PIC16C66X devices have on-chip Brown-out Reset circuitry. A configuration bit, BODEN, can disable (if clear/programmed) or enable (if set) the Brown-out Reset circuitry. If VDD falls below 4.0V (Parameter D005 in ES section) for greater than parameter 35 in Table 12-6, the brown-out situation will reset the chip. A reset is not guaranteed to occur if VDD falls below 4.0V for less than parameter 35. The chip will remain in Brown-out Reset until VDD rises above BVDD. The Power-up Timer will now be invoked and will keep the chip in reset an additional 72 ms. If VDD drops below BVDD while the Power-up Timer is running, the chip will go back into a Brown-out Reset and the Power-up Timer will be initialized. Once VDD rises above BVDD, the Power-up Timer will execute a 72 ms time delay. The Power-up Timer should always be enabled when Brown-out Reset is enabled. Figure 9-8 shows typical Brown-out situations.

FIGURE 9-8: BROWN-OUT SITUATIONS



9.4.5 PARITY ERROR RESET (PER)

PIC16C64X & PIC16C66X devices have on-chip parity bits that can be used to verify the contents of program memory. Parity bits may be useful in applications in order to increase overall reliability of a system.

There are two parity bits for each word of Program Memory. The parity bits are computed on alternating bits of the program word. One computation is performed using even parity, the other using odd parity. As a program executes, the parity is verified. The even parity bit is XOR'd with the even bits in the program memory word. The odd parity bit is negated and XOR'd with the odd bits in the program memory word. When an error is detected, a reset is generated and the $\overline{PER}$ flag bit in the PCON register is set. This indication can allow software to act on a failure. However, there is no indication of the program memory location of the failure of the Program Memory. This flag can only be cleared in software or by a POR.

The parity array is user selectable during programming. Bit7 of the configuration word located at address 2007h can be programmed (read as '0') to disable parity checking. If left unprogrammed (read as '1'), parity checking is enabled.

9.4.6 TIME-OUT SEQUENCE

On power-up, the time-out sequence is as follows: First PWRT time-out is invoked after POR has expired. Then the OST is activated. The total time-out will vary based on oscillator configuration and $\overline{PWRTE}$ bit status. For example, in RC mode with the $\overline{PWRTE}$ bit set (PWRT disabled), there will be no time-out at all. Figure 9-9, Figure 9-10 and Figure 9-11 depict time-out sequences.

Since the time-outs occur from the POR pulse, if $\overline{MCLR}$ is kept low long enough, the time-outs will expire. Then bringing $\overline{MCLR}$ high will begin execution immediately (Figure 9-10). This is useful for testing purposes or to synchronize more than one device operating in parallel.

Table 9-5 shows the reset conditions for some special registers, while Table 9-6 shows the reset conditions for all the registers.

9.4.7 POWER CONTROL/STATUS REGISTER (PCON)

The power control/status register, PCON (address 8Eh) has four bits. See Figure 4-10 for register.

Bit0 is $\overline{BOR}$ (Brown-out Reset). $\overline{BOR}$ is unknown on a Power-on-reset. It must initially be set by the user and checked on subsequent resets to see if $\overline{BOR}$ = '0' indicating that a Brown-out Reset has occurred. The $\overline{BOR}$ status bit is a "don't care" bit and is not necessarily predictable if the brown-out circuit is disabled (by clearing the BODEN bit in the Configuration word).

Bit1 is $\overline{POR}$ (Power-on Reset). It is cleared on a Power-on Reset and is unaffected otherwise. The user set this bit following a Power-on Reset. On subsequent resets if $\overline{POR}$ is '0', it will indicate that a Power-on Reset must have occurred.

Bit2 is $\overline{PER}$ (Parity Error Reset). It is cleared on a Parity Error Reset and must be set by user software. It will also be set on a Power-on Reset.

Bit7 is MPEEN (Memory Parity Error Enable). This bit reflects the status of the MPEEN bit in configuration word. It is unaffected by any reset or interrupt.

TABLE 9-3: TIME-OUT IN VARIOUS SITUATIONS

Oscillator Configuration	Power-up		Brown-out Reset	Wake-up from SLEEP
	$\overline{PWRTE}$ = 0	$\overline{PWRTE}$ = 1		
XT, HS, LP	72 ms + 1024 TOSC	1024 TOSC	72 ms + 1024 TOSC	1024 TOSC
RC	72 ms	—	72 ms	—

TABLE 9-4: STATUS BITS AND THEIR SIGNIFICANCE

$\overline{PER}$	$\overline{POR}$	$\overline{BOR}$	$\overline{TO}$	$\overline{PD}$	
1	0	x	1	1	Power-on Reset
x	0	x	0	x	Illegal, $\overline{TO}$ is set on POR
x	0	x	x	0	Illegal, $\overline{PD}$ is set on POR
1	1	0	1	1	Brown-out Reset
1	1	1	0	1	WDT Reset
1	1	1	0	0	WDT Wake-up
1	1	1	u	u	$\overline{MCLR}$ reset during normal operation
1	1	1	1	0	$\overline{MCLR}$ reset during SLEEP
0	1	1	1	1	Parity Error Reset
0	0	x	x	x	Illegal, $\overline{PER}$ is set on POR
0	x	0	x	x	Illegal, $\overline{PER}$ is set on BOR

TABLE 9-5: INITIALIZATION CONDITION FOR SPECIAL REGISTERS

Condition	Program Counter	STATUS Register	PCON Register
Power-on Reset	000h	0001 1xxx	u--- -10x
$\overline{MCLR}$ reset during normal operation	000h	000u uuuu	u--- -uuu
$\overline{MCLR}$ reset during SLEEP	000h	0001 0uuu	u--- -uuu
WDT reset	000h	0000 1uuu	u--- -uuu
WDT Wake-up	PC + 1	uuu0 0uuu	u--- -uuu
Brown-out Reset	000h	0001 1uuu	u--- -uu0
Parity Error Reset	000h	0001 1uuu	1--- -0uu
Interrupt Wake-up from SLEEP	PC + 1(1)	uuu1 0uuu	u--- -uuu

Legend: u = unchanged, x = unknown, - = unimplemented bit, reads as '0'.
Note 1: When the wake-up is due to an interrupt and global enable bit, GIE is set, the PC is loaded with the interrupt vector (0004h) after execution of PC+1.

TABLE 9-6: INITIALIZATION CONDITION FOR REGISTERS

Register	Address	Power-on Reset Brown-out Reset Parity Error Reset	MCLR Reset during: - normal operation - SLEEP or WDT Reset	Wake up from SLEEP through: - interrupt - WDT time-out
W	-	xxxx xxxx	uuuu uuuu	uuuu uuuu
INDF	00h	-	-	-
TMR0	01h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCL	02h	0000 0000	0000 0000	PC + 1(2)
STATUS	03h	0001 1xxx	000q quuu(3)	uuuq quuu(3)
FSR	04h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTA	05h	--xx 0000	--xu 0000	--uu uuuu
PORTB	06h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTC	07h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTD(4)	08h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTE(4)	09h	---- -xxx	---- -uuu	---- -uuu
CMCON	1Fh	00-- 0000	00-- 0000	uu-- uuuu
PCLATH	0Ah	---0 0000	---0 0000	---u uuuu
INTCON	0Bh	0000 000x	0000 000u	uuuu uuuu(1)
PIR1	0Ch	00-- ----	00-- ----	uu-- ----(1)
OPTION	81h	1111 1111	1111 1111	uuuu uuuu
TRISA	85h	--11 1111	--11 1111	--uu uuuu
TRISB	86h	1111 1111	1111 1111	uuuu uuuu
TRISC	87h	1111 1111	1111 1111	uuuu uuuu
TRISD(4)	88h	1111 1111	1111 1111	uuuu uuuu
TRISE(4)	89h	0000 -111	0000 -111	uuuu -uuu
PIE1	8Ch	00-- ----	00-- ----	uu-- ----
PCON	8Eh	u--- -qqq	u--- -uuu	u--- -uuu
VRCON	9Fh	000- 0000	000- 0000	uuu- uuuu

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

Note 1: One or more bits in INTCON and/or PIR1 will be affected (to cause wake-up).

2: When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).

3: See Table 9-5 for reset value for specific condition.

4: These registers are associated with the Parallel Slave Port and are not implemented on the PIC16C641/642.

FIGURE 9-9: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ NOT TIED TO V_{DD}): CASE 1

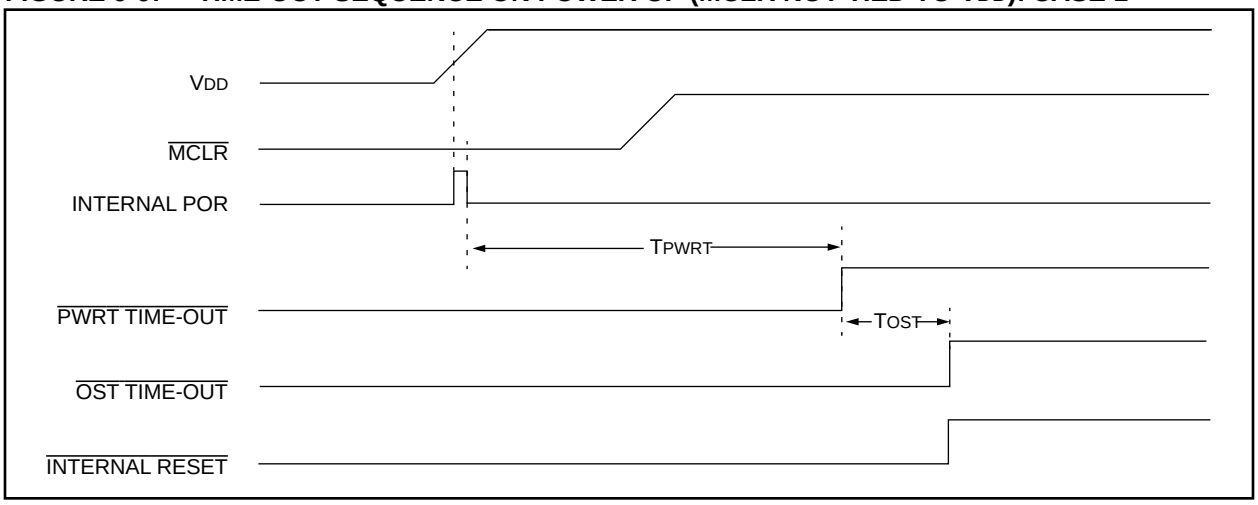


FIGURE 9-10: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ NOT TIED TO V_{DD}): CASE 2

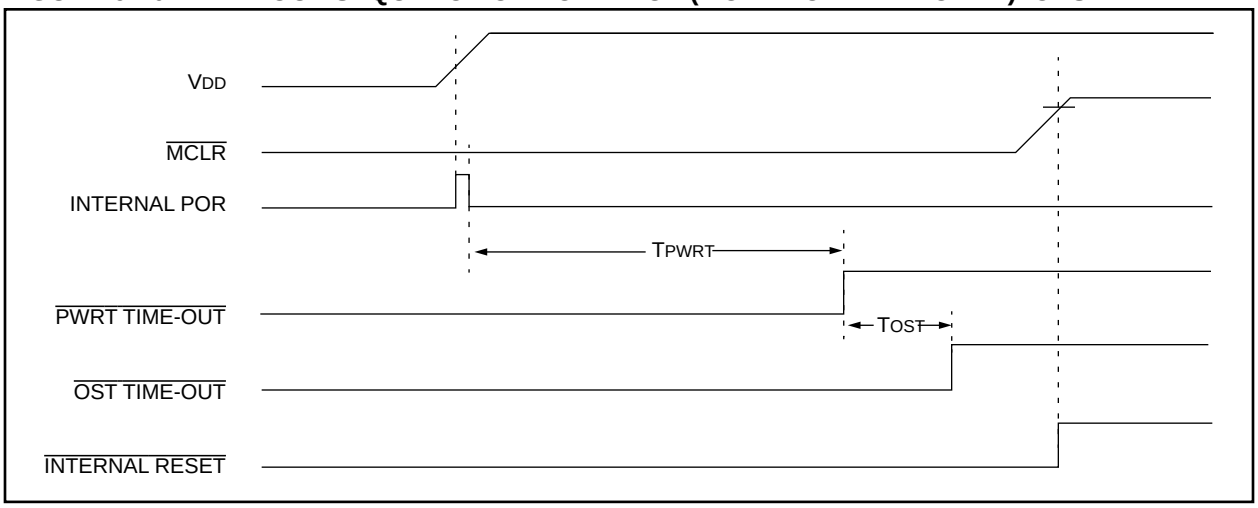


FIGURE 9-11: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ TIED TO V_{DD})

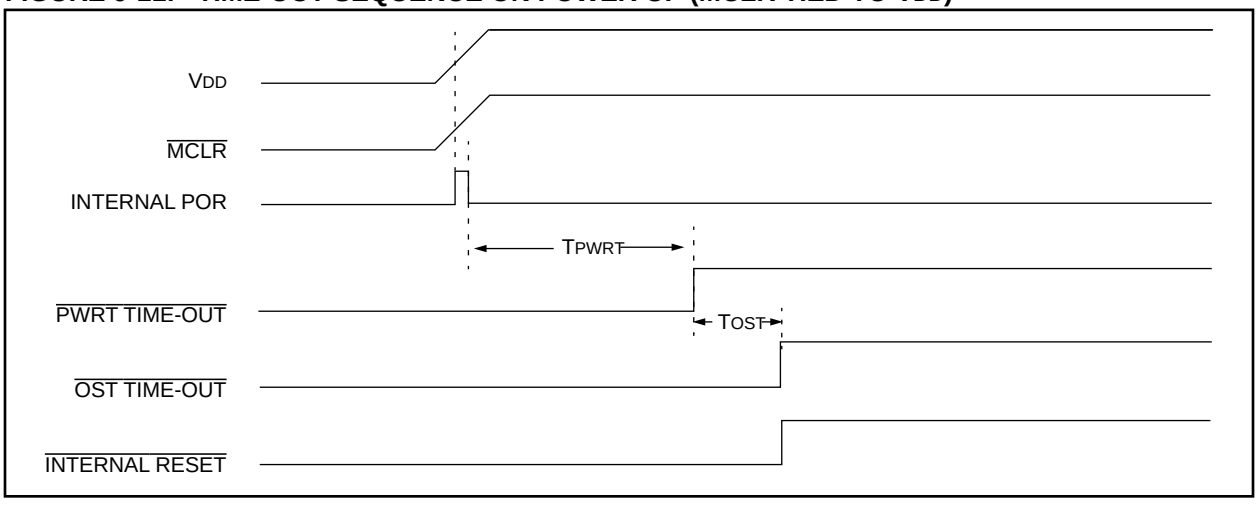
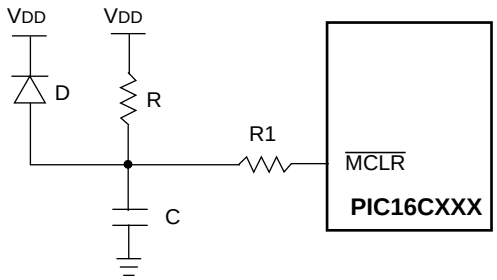


FIGURE 9-12: EXTERNAL POWER-ON RESET CIRCUIT (FOR SLOW VDD POWER-UP)

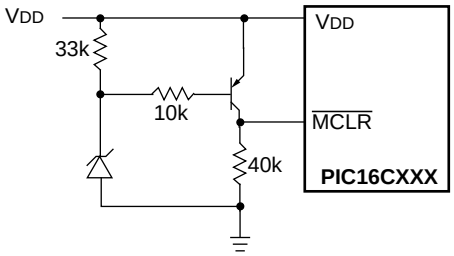


Note 1: External power-on reset circuit is required only if VDD power-up slope is too slow. The diode D helps discharge the capacitor quickly when VDD powers down.

2: R < 40 kΩ is recommended to make sure that voltage drop across R does not violate the device's electrical specification.

3: R1 = 100Ω to 1 kΩ will limit any current flowing into MCLR from external capacitor C in the event of MCLR/VPP pin breakdown due to Electrostatic Discharge (ESD) or Electrical Overstress (EOS).

FIGURE 9-13: EXTERNAL BROWN-OUT PROTECTION CIRCUIT 1

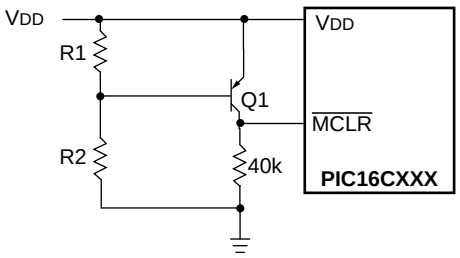


Note 1: This circuit will activate reset when VDD goes below (Vz + 0.7V) where Vz = Zener voltage.

2: Internal Brown-out Reset circuitry should be disabled when using this circuit.

3: Resistors should be adjusted for the characteristics of the transistor.

FIGURE 9-14: EXTERNAL BROWN-OUT PROTECTION CIRCUIT 2



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

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

2: Internal Brown-out Reset circuitry should be disabled when using this circuit.

3: Resistors should be adjusted for the characteristics of the transistor.

9.5 Interrupts

The PIC16C641 and PIC16C642 have four sources of interrupt, while the PIC16C661 and PIC16C662 have five sources:

- External interrupt RB0/INT
- TMR0 overflow interrupt
- PORTB change interrupts (pins RB7:RB4)
- Comparator interrupt
- Parallel Slave Port interrupt (PIC16C661/662)

The interrupt control register, (INTCON), records individual core interrupt requests in flag bits. It also has various individual enable bits and the global interrupt enable bit.

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

The "return from interrupt" instruction, RETFIE, exits the interrupt routine as well as sets the GIE bit, which allows any pending interrupt to execute.

Those interrupts associated with the "core" have their flag and enable bits in the INTCON register. The core interrupts are: RB0/INT pin interrupt, the RB port change interrupt, and the TMR0 overflow interrupt. The INTCON register also contains the Peripheral Interrupt Enable bit, PEIE. Bit PEIE will enable/mask the peripheral interrupts (CM and PSP) from vectoring when bit PEIE is set/cleared.

Flag bits PSPIF and CMIF are contained in special function register PIR1. The corresponding interrupt enable bits (PSPIE and CMIE) are contained in special function register PIE1.

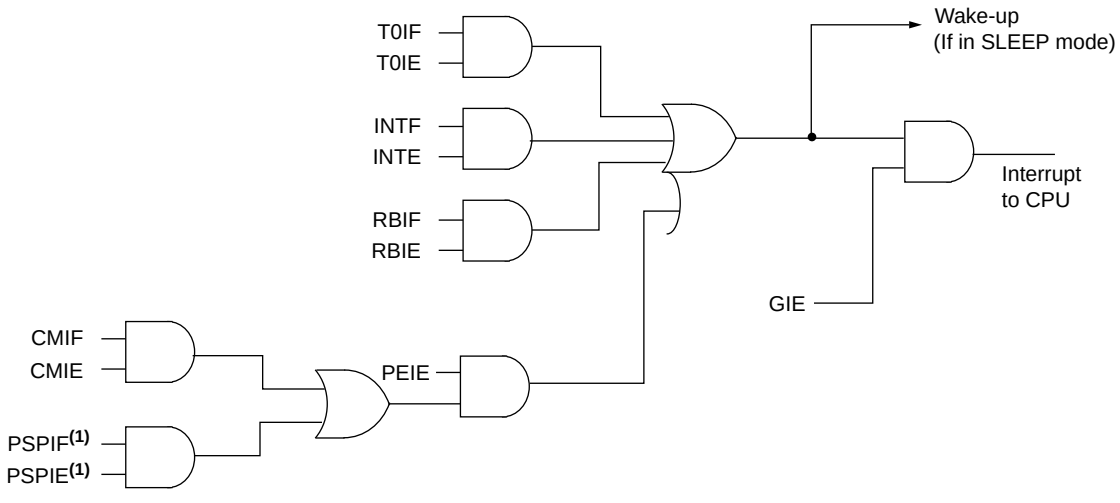
When an interrupt is responded to, the GIE is cleared to disable any further interrupt, the return address is pushed into the stack and the PC is loaded with 0004h. Once in the interrupt service routine the source(s) of the interrupt can be determined by polling the interrupt flag bits. The interrupt flag bit(s) must be cleared in software before re-enabling interrupts to avoid recursive interrupts.

For external interrupt events, such as the RB0/INT or Port RB change interrupt, the interrupt latency will be three or four instruction cycles. The exact latency depends when the interrupt event occurs (Figure 9-16). 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 multiple interrupt requests. Individual interrupt flag bits are set regardless of the status of their corresponding mask bit or the GIE bit.

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

Note 2: When an instruction that clears the GIE bit is executed, any interrupts that were pending for execution in the next cycle are ignored. The CPU will execute a NOP in the cycle immediately following the instruction which clears the GIE bit. The interrupts which were ignored are still pending to be serviced when the GIE bit is set again.

FIGURE 9-15: INTERRUPT LOGIC



Note 1: The Parallel Slave Port is implemented on the PIC16C661 and PIC16C662 only.

9.5.1 RB0/INT INTERRUPT

The external interrupt on the RB0/INT pin is edge triggered: either rising if bit INTEDG (OPTION<6>) is set, or falling, if bit INTEDG is clear. When a valid edge appears on the RB0/INT pin, flag bit INTF (INTCON<1>) is set. This interrupt can be enabled/disabled by setting/clearing enable bit INTE (INTCON<4>). The INTF bit must be cleared in software in the interrupt service routine before re-enabling this interrupt. The RB0/INT interrupt can wake-up the processor from SLEEP, if bit INTE was set 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 9.8 for details on SLEEP and Figure 9-19 for timing of wake-up from SLEEP through RB0/INT interrupt.

9.5.2 TMR0 INTERRUPT

An overflow (FFh $\rightarrow$ 00h) in the TMR0 register will set the T0IF (INTCON<2>) bit. The interrupt can be enabled/disabled by setting/clearing T0IE (INTCON<5>) bit. For operation of the Timer0 module, see Section 6.0.

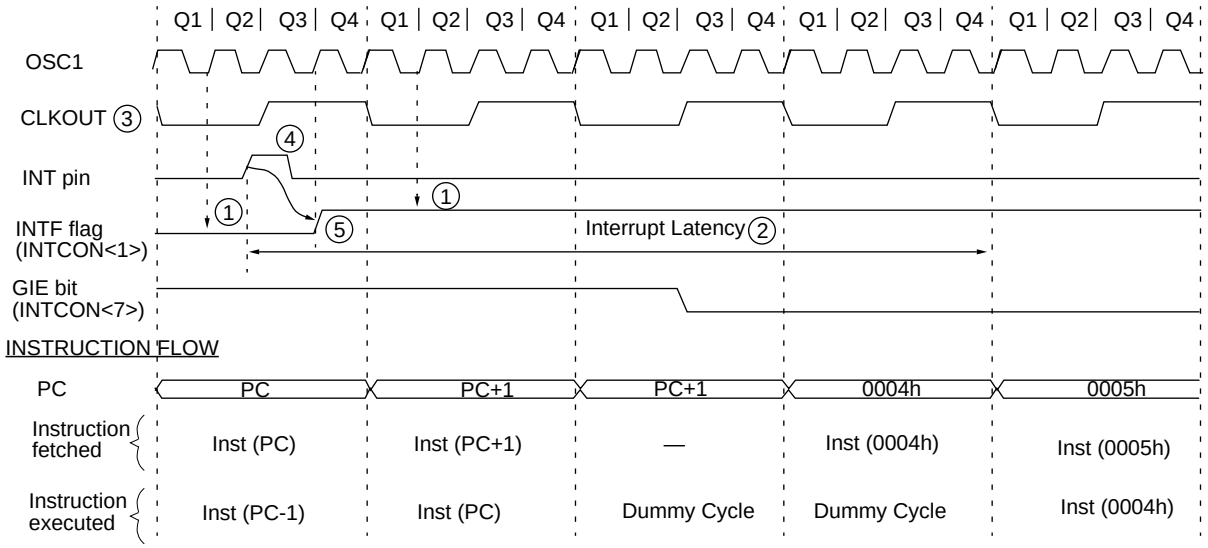
9.5.3 PORTB INTERRUPT

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

9.5.4 COMPARATOR INTERRUPT

See Section 7.6 for complete description of the comparator interrupt.

FIGURE 9-16: RB0/INT PIN INTERRUPT TIMING



Note 1: INTF flag is sampled here (every Q1).
2: Interrupt latency = 3-4 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 of INT pulse, refer to AC specs.
5: INTF is enabled to be set anytime during the Q4-Q1 cycles.

9.6 Context Saving During Interrupts

During an interrupt, only the return PC value is saved on the stack. Typically, users may wish to save key registers during an interrupt e.g. W register and STATUS register. This will have to be implemented in software.

Example 9-1 stores and restores the STATUS and W registers. The user register, W_TEMP, must be defined in both banks and must be defined at the same offset from the bank base address (i.e., W_TEMP is defined at 0x70 - 0x7F in Bank 0). The user register, STATUS_TEMP, must be defined in Bank 0.

Example 9-1:

- Stores the W register regardless of current bank
- Stores the STATUS register in Bank 0
- Executes the ISR code
- Restores the STATUS (and bank select bit register)
- Restores the W register

EXAMPLE 9-1: SAVING THE STATUS AND W REGISTERS IN RAM

```
MOVWF   W_TEMP          ; Copy W to a Temporary Register regardless of current bank
SWAPF   STATUS,W        ; Swap STATUS nibbles and place into W register
BCF     STATUS,RP0      ; Change to Bank 0 regardless of current bank
MOVWF   STATUS_TEMP     ; Save STATUS to a Temporary register in Bank 0
    :
    : (Interrupt Service Routine)
    :
SWAPF   STATUS_TEMP,W   ; Swap original STATUS register value into W (restores original bank)
MOVWF   STATUS          ; Restore STATUS register from W register
SWAPF   W_TEMP,F        ; Swap W_Temp nibbles and return value to W_Temp
SWAPF   W_TEMP,W        ; Swap W_Temp to W to restore original W value without affecting STATUS
```


9.7 Watchdog Timer (WDT)

The Watchdog Timer (WDT) is a free running on-chip RC oscillator which does not require any external components. The block diagram is shown in Figure 9-17. This RC oscillator is separate from the RC oscillator of the OSC1/CLKIN pin. This means that the WDT will run, even if the clock on the OSC1 and OSC2 pins has been stopped, for example, by execution of a SLEEP instruction. During normal operation, a WDT time-out generates a device RESET. If the device is in SLEEP mode, a WDT time-out causes the device to wake-up and continue with normal operation, this is known as a WDT wake-up. The WDT can be permanently disabled by clearing configuration bit WDTE (Section 9.1).

9.7.1 WDT PERIOD

The WDT has a nominal time-out period of 18 ms, (with no prescaler). The time-out period varies with temperature, VDD and process variations from part to part (see DC specs). If longer time-outs 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 of up to 2.3 seconds can be realized.

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

The $\overline{TO}$ bit in the STATUS register will be cleared upon a Watchdog Timer time-out (WDT Reset and WDT wake-up).

9.7.2 WDT PROGRAMMING CONSIDERATIONS

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 time-out occurs.

Note:	When the prescaler is assigned to the WDT, always execute a CLRWDT instruction before changing the prescale value, otherwise a WDT reset may occur.

FIGURE 9-17: WATCHDOG TIMER BLOCK DIAGRAM

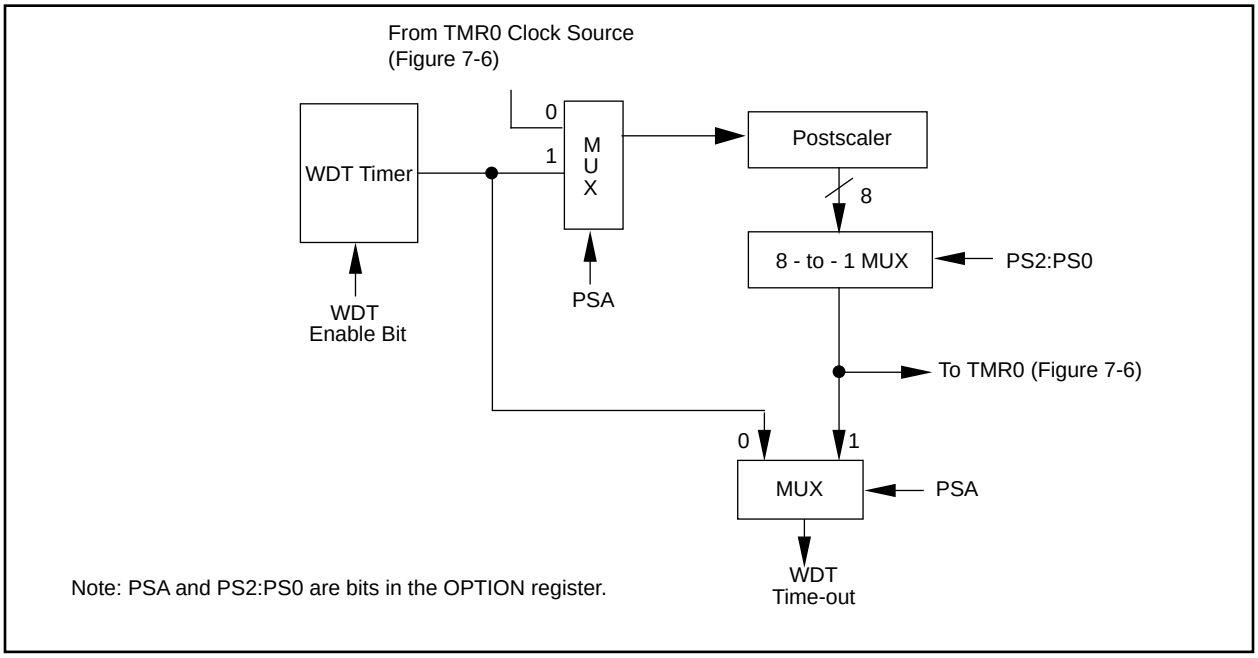


FIGURE 9-18: SUMMARY OF WATCHDOG TIMER REGISTERS

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
2007h	Config. bits	MPEEN	BODEN(1)	CP1	CP0	$\overline{PWRTE}$(1)	WDTE	FOSC1	FOSC0
81h	OPTION	$\overline{RBPU}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0

Legend: Shaded cells are not used by the Watchdog Timer.
Note 1: See Figure 9-1 for details of the operation of these bits.

9.8 Power-Down Mode (SLEEP)

Power-down mode is entered by executing a SLEEP instruction.

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

For lowest current consumption in this mode, all I/O pins should be either at V_{DD}, or V_{SS}, with no external circuitry drawing current from the I/O pin and the comparators and V_{REF} module should be disabled. I/O pins that are hi-impedance inputs should be pulled high or low externally to avoid switching currents caused by floating inputs. The T0CKI input should also be at V_{DD} or V_{SS} for lowest current consumption. The contribution from on chip pull-ups on PORTB should be considered.

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

9.8.1 WAKE-UP FROM SLEEP

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

1. Any device reset
2. Watchdog Timer Wake-up (if WDT was enabled)
3. Interrupt from RB0/INT pin, RB Port change, or the Comparator.

The first event will reset the device upon wake-up. However the latter two events will wake the device and then resume program execution. The $\overline{TO}$ and $\overline{PD}$ bits in the STATUS register can be used to determine the cause of device reset. The $\overline{PD}$ bit, which is set on power-up is cleared when SLEEP is invoked. The $\overline{TO}$ bit is cleared if WDT wake-up occurred.

When the SLEEP instruction is being executed, the next instruction (PC + 1) is pre-fetched. For the device to wake-up through an interrupt event, the corresponding interrupt enable bit must be set (enabled). Wake-up is regardless of the state of the GIE bit. If the GIE bit is clear (disabled), the device continues execution at the instruction after the SLEEP instruction. If the GIE bit is set (enabled), the device executes the instruction after the SLEEP instruction and then branches to the interrupt address (0004h). In cases where the execution of the instruction following SLEEP is not desirable, the user should have an NOP after the SLEEP instruction.

9.8.2 WAKE-UP USING INTERRUPTS

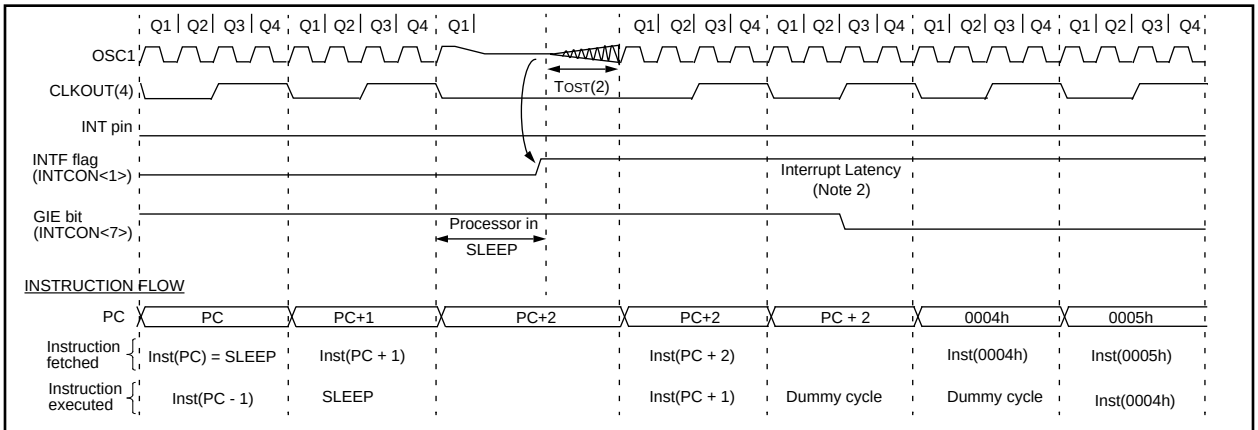
When global interrupts are disabled (GIE cleared) and any interrupt source has both its interrupt enable bit and interrupt flag set, one of the following events will occur:

- If the interrupt occurs before the execution of a SLEEP instruction, the SLEEP instruction will complete as an NOP. Therefore, the WDT and WDT postscaler will not be cleared, the $\overline{TO}$ bit will not be set and $\overline{PD}$ bit will not be cleared.
- If the interrupt occurs during or after the execution of a SLEEP instruction, the device will immediately wake-up from sleep. The SLEEP instruction will be completely executed before the wake-up. Therefore, the WDT and WDT postscaler will be cleared, the $\overline{TO}$ bit will be set and the $\overline{PD}$ bit will be cleared.

Even if the flag bits were checked before executing a SLEEP instruction, it may be possible for flag bits to become set before the SLEEP instruction completes. To determine whether a SLEEP instruction executed, test the $\overline{PD}$ bit. If the $\overline{PD}$ bit is set, the SLEEP instruction was executed as an NOP.

To ensure that the WDT is clear, a CLRWDT instruction should be executed before a SLEEP instruction.

FIGURE 9-19: WAKE-UP FROM SLEEP THROUGH INTERRUPT



Note 1: XT, HS or LP oscillator mode assumed.
2: TOST = 1024TOSC (drawing not to scale) This delay will not be there for RC osc mode.
3: GIE = '1' assumed. In this case after wake- up, the processor jumps to the 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.

9.9 Code Protection

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

Note:	Microchip does not recommend code protecting windowed devices.

9.10 ID Locations

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

9.11 In-Circuit Serial Programming

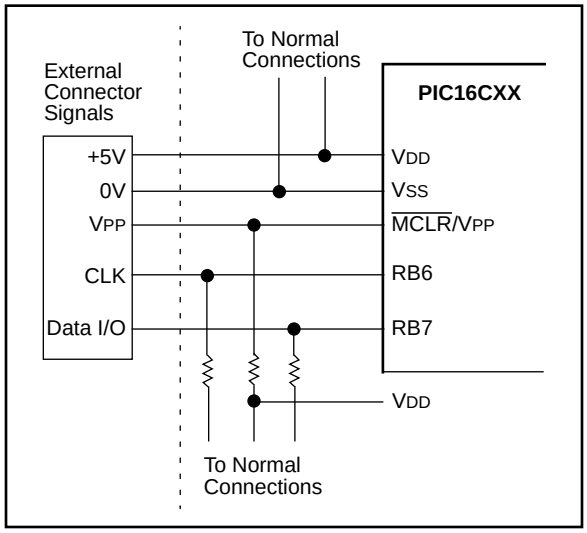
The PIC16CXX microcontrollers can be serially programmed while in the end application circuit. This is simply done with two lines for clock and data, and three other lines for power, ground, and the programming voltage. This allows customers to manufacture boards with unprogrammed devices, and then program the microcontroller just before shipping the product. This also allows the most recent firmware or a custom firmware to be programmed.

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

After reset, to place the device into programming/verify mode, the program counter (PC) is at location 00h. A 6-bit command is then supplied to the device. Depending on the command, 14-bits of program data are then supplied to or from the device, depending if the command was a load or a read. For complete details of serial programming, please refer to the PIC16C6X/7X Programming Specifications (Literature #DS30228).

A typical in-circuit serial programming connection is shown in Figure 9-20.

FIGURE 9-20: TYPICAL IN-CIRCUIT SERIAL PROGRAMMING CONNECTION



NOTES:

10.0 INSTRUCTION SET SUMMARY

Each PIC16CXX instruction is a 14-bit word divided into an OPCODE which specifies the instruction type and one or more operands which further specify the operation of the instruction. The PIC16CXX instruction set summary in Table 10-2 lists **byte-oriented**, **bit-oriented**, and **literal and control** operations. Table 10-1 shows the opcode field descriptions.

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

The destination designator specifies where the result of the operation is to be placed. If 'd' is zero, the result is placed in the W register. If 'd' is one, the result is placed in the file register specified in the instruction.

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

For **literal and control** operations, 'k' represents an eight or eleven bit constant or literal value.

TABLE 10-1: OPCODE FIELD DESCRIPTIONS

Field	Description
f	Register file address (0x00 to 0x7F)
W	Working register (accumulator)
b	Bit address within an 8-bit file register
k	Literal field, constant data or label
x	Don't care location (= 0 or 1) The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all Microchip software tools.
d	Destination select; d = 0: store result in W, d = 1: store result in file register f. Default is d = 1
label	Label name
TOS	Top of Stack
PC	Program Counter
PCLATH	Program Counter High Latch
GIE	Global Interrupt Enable bit
WDT	Watchdog Timer/Counter
$\overline{TO}$	Time-out bit
$\overline{PD}$	Power-down bit
dest	Destination either the W register or the specified register file location
[ ]	Options
( )	Contents
→	Assigned to
< >	Register bit field
∈	In the set of
italics	User defined term (font is courier)

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

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

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

Table 10-2 lists the instructions recognized by the MPASM assembler.

Figure 10-1 shows the three general formats that the instructions can have.

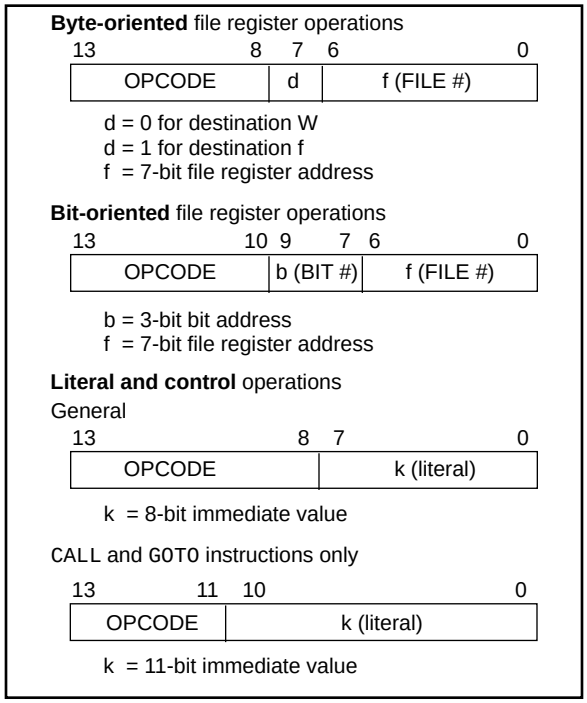
Note: To maintain upward compatibility with future PIC16CXX products, do not use the OPTION and TRIS instructions.

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

0xhh

where h signifies a hexadecimal digit.

FIGURE 10-1: GENERAL FORMAT FOR INSTRUCTIONS



10.1 Special Function Registers as Source/Destination

The PIC16C64X & PIC16C66X's orthogonal instruction set allows read and write of all file registers, including special function registers. There are some special situations the user should be aware of:

10.1.1 STATUS AS DESTINATION

If an instruction writes to STATUS, the Z, C, and DC bits may be set or cleared as a result of the instruction and overwrite the original data bits written. For example, executing `CLRF STATUS` will clear register STATUS, and then set the Z bit leaving `0000 0100`b in the register.

10.1.2 PCL AS SOURCE OR DESTINATION

Read, write or read-modify-write on PCL may have the following results:

Read PC:	PCL → dest
Write PCL:	PCLATH → PCH; 8-bit destination value → PCL
Read-Modify-Write:	PCL→ ALU operand PCLATH → PCH; 8-bit result → PCL

Where PCH = program counter high byte (not an addressable register), PCLATH = Program counter high holding latch, dest = destination, WREG or f.

10.1.3 BIT MANIPULATION

All bit manipulation instructions are done by first reading the entire register, operating on the selected bit and writing the result back (read-modify-write). The user should keep this in mind when operating on special function registers, such as ports.

TABLE 10-2: INSTRUCTION SET

Mnemonic, Operands		Description	Cycles	14-Bit Opcode MSb			LSb	Status Affected	Notes
BYTE-ORIENTED FILE REGISTER OPERATIONS									
ADDWF	**f, d**	Add W and f	1	00	0111	dfff	ffff	C,DC,Z	1,2
ANDWF	**f, d**	AND W with f	1	00	0101	dfff	ffff	Z	1,2
CLRF	**f**	Clear f	1	00	0001	lfff	ffff	Z	2
CLRW	**-**	Clear W	1	00	0001	0000	0011	Z	
COMF	**f, d**	Complement f	1	00	1001	dfff	ffff	Z	1,2
DECF	**f, d**	Decrement f	1	00	0011	dfff	ffff	Z	1,2
DECFSZ	**f, d**	Decrement f, Skip if 0	1(2)	00	1011	dfff	ffff		1,2,3
INCF	**f, d**	Increment f	1	00	1010	dfff	ffff	Z	1,2
INCFSZ	**f, d**	Increment f, Skip if 0	1(2)	00	1111	dfff	ffff		1,2,3
IORWF	**f, d**	Inclusive OR W with f	1	00	0100	dfff	ffff	Z	1,2
MOVF	**f, d**	Move f	1	00	1000	dfff	ffff	Z	1,2
MOVWF	**f**	Move W to f	1	00	0000	lfff	ffff		
NOP	**-**	No Operation	1	00	0000	0xx0	0000		
RLF	**f, d**	Rotate Left f through Carry	1	00	1101	dfff	ffff	C	1,2
RRF	**f, d**	Rotate Right f through Carry	1	00	1100	dfff	ffff	C	1,2
SUBWF	**f, d**	Subtract W from f	1	00	0010	dfff	ffff	C,DC,Z	1,2
SWAPF	**f, d**	Swap nibbles in f	1	00	1110	dfff	ffff		1,2
XORWF	**f, d**	Exclusive OR W with f	1	00	0110	dfff	ffff	Z	1,2
BIT-ORIENTED FILE REGISTER OPERATIONS									
BCF	**f, b**	Bit Clear f	1	01	00bb	bfff	ffff		1,2
BSF	**f, b**	Bit Set f	1	01	01bb	bfff	ffff		1,2
BTFSC	**f, b**	Bit Test f, Skip if Clear	1 (2)	01	10bb	bfff	ffff		3
BTFSS	**f, b**	Bit Test f, Skip if Set	1 (2)	01	11bb	bfff	ffff		3
LITERAL AND CONTROL OPERATIONS									
ADDLW	**k**	Add literal and W	1	11	111x	kkkk	kkkk	C,DC,Z	
ANDLW	**k**	AND literal with W	1	11	1001	kkkk	kkkk	Z	
CALL	**k**	Call subroutine	2	10	0kkk	kkkk	kkkk		
CLRWDT	**-**	Clear Watchdog Timer	1	00	0000	0110	0100	$\overline{TO}$,$\overline{PD}$	
GOTO	**k**	Go to address	2	10	1kkk	kkkk	kkkk		
IORLW	**k**	Inclusive OR literal with W	1	11	1000	kkkk	kkkk	Z	
MOVLW	**k**	Move literal to W	1	11	00xx	kkkk	kkkk		
RETFIE	**-**	Return from interrupt	2	00	0000	0000	1001		
RETLW	**k**	Return with literal in W	2	11	01xx	kkkk	kkkk		
RETURN	**-**	Return from Subroutine	2	00	0000	0000	1000		
SLEEP	**-**	Go into standby mode	1	00	0000	0110	0011	$\overline{TO}$,$\overline{PD}$	
SUBLW	**k**	Subtract W from literal	1	11	110x	kkkk	kkkk	C,DC,Z	
XORLW	**k**	Exclusive OR literal with W	1	11	1010	kkkk	kkkk	Z	

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

2: If this instruction is executed on the TMR0 register (and, where applicable, d = 1), the prescaler will be cleared if assigned to the Timer0 Module.

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

10.2 Instruction Descriptions

ADDLW — **Add Literal and W**

Syntax: [*label*] ADDLW k

Operands: $0 \le k \le 255$

Operation: (W) + k → (W)

Status Affected: C, DC, Z

Encoding:

11	111x	kkkk	kkkk

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

Words: 1

Cycles: 1

Example
```
ADDLW   0x15
```
Before Instruction
W = 0x10
After Instruction
W = 0x25

ANDLW — **And Literal with W**

Syntax: [*label*] ANDLW k

Operands: $0 \le k \le 255$

Operation: (W) .AND. (k) → (W)

Status Affected: Z

Encoding:

11	1001	kkkk	kkkk

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

Words: 1

Cycles: 1

Example
```
ANDLW   0x5F
```
Before Instruction
W = 0xA3
After Instruction
W = 0x03

ADDWF — **Add W and f**

Syntax: [*label*] ADDWF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (W) + (f) → (dest)

Status Affected: C, DC, Z

Encoding:

00	0111	dfff	ffff

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

Words: 1

Cycles: 1

Example
```
ADDWF   FSR, 0
```
Before Instruction
W = 0x17
FSR = 0xC2
After Instruction
W = 0xD9
FSR = 0xC2

ANDWF — **AND W with f**

Syntax: [*label*] ANDWF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (W) .AND. (f) → (dest)

Status Affected: Z

Encoding:

00	0101	dfff	ffff

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

Words: 1

Cycles: 1

Example
```
ANDWF   FSR, 1
```
Before Instruction
W = 0x17
FSR = 0xC2
After Instruction
W = 0x17
FSR = 0x02

BCF **Bit Clear f**

Syntax: [*label*] BCF f,b

Operands: 0 ≤ f ≤ 127
0 ≤ b ≤ 7

Operation: 0 → (f<b>)

Status Affected: None

Encoding:

01	00bb	bfff	ffff

Description: Bit 'b' in register 'f' is cleared.

Words: 1

Cycles: 1

Example
```
BCF     FLAG_REG, 7
```
Before Instruction
FLAG_REG = 0xC7
After Instruction
FLAG_REG = 0x47

BSF **Bit Set f**

Syntax: [*label*] BSF f,b

Operands: 0 ≤ f ≤ 127
0 ≤ b ≤ 7

Operation: 1 → (f<b>)

Status Affected: None

Encoding:

01	01bb	bfff	ffff

Description: Bit 'b' in register 'f' is set.

Words: 1

Cycles: 1

Example
```
BSF     FLAG_REG,   7
```
Before Instruction
FLAG_REG = 0x0A
After Instruction
FLAG_REG = 0x8A

BTFSC **Bit Test, Skip if Clear**

Syntax: [*label*] BTFSC f,b

Operands: 0 ≤ f ≤ 127
0 ≤ b ≤ 7

Operation: skip if (f<b>) = 0

Status Affected: None

Encoding:

01	10bb	bfff	ffff

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

Words: 1

Cycles: 1(2)

Example
```
HERE    BTFSC   FLAG,1
FALSE   GOTO    PROCESS_CODE
TRUE    •
        •
        •
```
Before Instruction
PC = address HERE
After Instruction
if FLAG<1> = 0,
PC = address TRUE
if FLAG<1>=1,
PC = address FALSE

BTFSS **Bit Test f, Skip if Set**

Syntax: [*label*] BTFSS f,b

Operands: 0 ≤ f ≤ 127
0 ≤ b < 7

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

Status Affected: None

Encoding:

01	11bb	bfff	ffff

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

Words: 1

Cycles: 1(2)

Example

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

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

CALL **Call Subroutine**

Syntax: [*label*] CALL k

Operands: 0 ≤ k ≤ 2047

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

Status Affected: None

Encoding:

10	0kkk	kkkk	kkkk

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

Words: 1

Cycles: 2

Example

```
HERE    CALL    THERE
```

Before Instruction
PC = Address HERE
After Instruction
PC = Address THERE
TOS = Address HERE+1

CLRF **Clear f**

Syntax: [*label*] CLRF f

Operands: 0 ≤ f ≤ 127

Operation: 00h → (f)
1 → Z

Status Affected: Z

Encoding:

00	0001	1fff	ffff

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

Words: 1

Cycles: 1

Example

```
CLRF    FLAG_REG
```

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

CLRW **Clear W**

Syntax: [*label*] CLRW

Operands: None

Operation: 00h → (W)
1 → Z

Status Affected: Z

Encoding:

00	0001	0000	0011

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

Words: 1

Cycles: 1

Example

```
CLRW
```

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

CLRWDT **Clear Watchdog Timer**

Syntax: [*label*] CLRWDT

Operands: None

Operation: 00h → WDT
0 → WDT prescaler,
1 → $\overline{TO}$
1 → $\overline{PD}$

Status Affected: $\overline{TO}$, $\overline{PD}$

Encoding:

00	0000	0110	0100

Description: CLRWDT instruction resets the Watchdog Timer. It also resets the prescaler of the WDT. Status bits $\overline{TO}$ and $\overline{PD}$ are set.

Words: 1

Cycles: 1

Example
```
CLRWDT
```
Before Instruction
WDT counter = ?
After Instruction
WDT counter = 0x00
WDT prescaler = 0
$\overline{TO}$ = 1
$\overline{PD}$ = 1

COMF **Complement f**

Syntax: [*label*] COMF f,d

Operands: 0 ≤ f ≤ 127
d ∈ [0,1]

Operation: ($\bar{f}$) → (dest)

Status Affected: Z

Encoding:

00	1001	dfff	ffff

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

Words: 1

Cycles: 1

Example
```
COMF    REG1,0
```
Before Instruction
REG1 = 0x13
After Instruction
REG1 = 0x13
W = 0xEC

DECF **Decrement f**

Syntax: [*label*] DECF f,d

Operands: 0 ≤ f ≤ 127
d ∈ [0,1]

Operation: (f) - 1 → (dest)

Status Affected: Z

Encoding:

00	0011	dfff	ffff

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

Words: 1

Cycles: 1

Example
```
DECF    CNT, 1
```
Before Instruction
CNT = 0x01
Z = 0
After Instruction
CNT = 0x00
Z = 1

DECFSZ **Decrement f, Skip if 0**

Syntax: [*label*] DECFSZ f,d

Operands: 0 ≤ f ≤ 127
d ∈ [0,1]

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

Status Affected: None

Encoding:

00	1011	dfff	ffff

Description: The contents of register 'f' are decremented. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'.
If the result is 0, the next instruction, which is already fetched, is discarded. A NOP is executed instead making it a two cycle instruction.

Words: 1

Cycles: 1(2)

Example
```
HERE      DECFSZ   CNT, 1
          GOTO     LOOP
CONTINUE  •
          •
          •
```
Before Instruction
PC = address HERE
After Instruction
CNT = CNT - 1
if CNT = 0,
PC = address CONTINUE
if CNT ≠ 0,
PC = address HERE+1

GOTO **Unconditional Branch**

Syntax: [*label*] GOTO k

Operands: 0 ≤ k ≤ 2047

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

Status Affected: None

Encoding:

10	1kkk	kkkk	kkkk

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

Words: 1

Cycles: 2

Example
```
GOTO THERE
```
After Instruction
PC = Address THERE

INCF **Increment f**

Syntax: [*label*] INCF f,d

Operands: 0 ≤ f ≤ 127
d ∈ [0,1]

Operation: (f) + 1 → (dest)

Status Affected: Z

Encoding:

00	1010	dfff	ffff

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

Words: 1

Cycles: 1

Example
```
INCF CNT, 1
```
Before Instruction
CNT = 0xFF
Z = 0
After Instruction
CNT = 0x00
Z = 1

INCFSZ **Increment f, Skip if 0**

Syntax: [*label*] INCFSZ f,d

Operands: 0 ≤ f ≤ 127
d ∈ [0,1]

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

Status Affected: None

Encoding:

00	1111	dfff	ffff

Description: The contents of register 'f' are incremented. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'.
If the result is 0, the next instruction, which is already fetched, is discarded. A NOP is executed instead making it a two cycle instruction.

Words: 1

Cycles: 1(2)

Example
```
HERE     INCFSZ   CNT, 1
         GOTO     LOOP
CONTINUE •
         •
         •
```
Before Instruction
PC = address HERE
After Instruction
CNT = CNT + 1
if CNT= 0,
PC = address CONTINUE
if CNT≠ 0,
PC = address HERE +1

IORLW **Inclusive OR Literal with W**

Syntax: [*label*] IORLW k

Operands: 0 ≤ k ≤ 255

Operation: (W) .OR. k → (W)

Status Affected: Z

Encoding:

11	1000	kkkk	kkkk

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

Words: 1

Cycles: 1

Example
```
IORLW 0x35
```
Before Instruction
W = 0x9A
After Instruction
W = 0xBF
Z = 1

IORWF **Inclusive OR W with f**

Syntax: [*label*] IORWF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (W) .OR. (f) → (dest)

Status Affected: Z

Encoding:

00	0100	dfff	ffff

Description: Inclusive OR the W register with register 'f'. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'.

Words: 1

Cycles: 1

Example
```
IORWF          RESULT, 0
```
Before Instruction
RESULT = 0x13
W = 0x91
After Instruction
RESULT = 0x13
W = 0x93
Z = 1

MOVF **Move f**

Syntax: [*label*] MOVF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (f) → (dest)

Status Affected: Z

Encoding:

00	1000	dfff	ffff

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

Words: 1

Cycles: 1

Example
```
MOVF     FSR, 0
```
After Instruction
W = value in FSR register
Z = 1

MOVLW **Move Literal to W**

Syntax: [*label*] MOVLW k

Operands: $0 \le k \le 255$

Operation: k → (W)

Status Affected: None

Encoding:

11	00xx	kkkk	kkkk

Description: The eight bit literal 'k' is loaded into W register. The don't cares will assemble as 0's.

Words: 1

Cycles: 1

Example
```
MOVLW    0x5A
```
After Instruction
W = 0x5A

MOVWF **Move W to f**

Syntax: [*label*] MOVWF f

Operands: $0 \le f \le 127$

Operation: (W) → (f)

Status Affected: None

Encoding:

00	0000	1fff	ffff

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

Words: 1

Cycles: 1

Example
```
MOVWF       OPTION
```
Before Instruction
OPTION = 0xFF
W = 0x4F
After Instruction
OPTION = 0x4F
W = 0x4F

NOP — **No Operation**

Syntax: [*label*] NOP

Operands: None

Operation: No operation

Status Affected: None

Encoding:

00	0000	0xx0	0000

Description: No operation.

Words: 1

Cycles: 1

Example

```
NOP
```

RETFIE — **Return from Interrupt**

Syntax: [*label*] RETFIE

Operands: None

Operation: TOS → PC,
1 → GIE

Status Affected: None

Encoding:

00	0000	0000	1001

Description: Return from Interrupt. Stack is POPed and Top of Stack (TOS) is loaded in the PC. Interrupts are enabled by setting Global Interrupt Enable bit, GIE (INTCON<7>). This is a two cycle instruction.

Words: 1

Cycles: 2

Example

```
RETFIE
```

After Interrupt
PC = TOS
GIE = 1

OPTION — **Load Option Register**

Syntax: [*label*] OPTION

Operands: None

Operation: (W) → OPTION

Status Affected: None

Encoding:

00	0000	0110	0010

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

Words: 1

Cycles: 1

Example

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

RETLW — **Return with Literal in W**

Syntax: [*label*] RETLW k

Operands: 0 ≤ k ≤ 255

Operation: k → (W);
TOS → PC

Status Affected: None

Encoding:

11	01xx	kkkk	kkkk

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

Words: 1

Cycles: 2

Example

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

Before Instruction
W = 0x07
After Instruction
W = value of k8

RETURN **Return from Subroutine**

Syntax: [*label*] RETURN

Operands: None

Operation: TOS → PC

Status Affected: None

Encoding:

00	0000	0000	1000

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

Words: 1

Cycles: 2

Example

```
RETURN
```

After Interrupt

```
PC  =  TOS
```

RRF **Rotate Right f through Carry**

Syntax: [*label*] RRF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

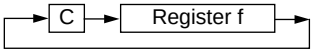
Operation: See description below

Status Affected: C

Encoding:

00	1100	dfff	ffff

Description: The contents of register 'f' are rotated one bit to the right through the Carry Flag. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'.



Words: 1

Cycles: 1

Example

```
RRF          REG1,0
```

Before Instruction

```
REG1   =   1110 0110
C      =   0
```

After Instruction

```
REG1   =   1110 0110
W      =   0111 0011
C      =   0
```

RLF **Rotate Left f through Carry**

Syntax: [*label*] RLF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

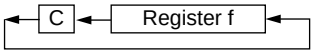
Operation: See description below

Status Affected: C

Encoding:

00	1101	dfff	ffff

Description: The contents of register 'f' are rotated one bit to the left through the Carry Flag. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is stored back in register 'f'.



Words: 1

Cycles: 1

Example

```
RLF          REG1,0
```

Before Instruction

```
REG1   =   1110 0110
C      =   0
```

After Instruction

```
REG1   =   1110 0110
W      =   1100 1100
C      =   1
```

SLEEP

Syntax: [*label*] SLEEP

Operands: None

Operation: 00h → WDT,
0 → WDT prescaler,
1 → $\overline{TO}$,
0 → $\overline{PD}$

Status Affected: $\overline{TO}$, $\overline{PD}$

Encoding:

00	0000	0110	0011

Description: The power-down status bit, $\overline{PD}$ is cleared. Time-out status bit, $\overline{TO}$ is set. Watchdog Timer and its prescaler are cleared.
The processor is put into SLEEP mode with the oscillator stopped. See Power-Down Mode (SLEEP) for more details.

Words: 1

Cycles: 1

Example: SLEEP

SUBLW **Subtract W from Literal**

Syntax: [*label*] SUBLW k

Operands: $0 \le k \le 255$

Operation: k - (W) → (W)

Status Affected: C, DC, Z

Encoding:

11	110x	kkkk	kkkk

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

Words: 1

Cycles: 1

Example 1:
```
SUBLW   0x02
```
Before Instruction
```
W = 1
C = ?
```
After Instruction
```
W = 1
C = 1; result is positive
```

Example 2: Before Instruction
```
W = 2
C = ?
```
After Instruction
```
W = 0
C = 1;  result is zero
```

Example 3: Before Instruction
```
W = 3
C = ?
```
After Instruction
```
W = 0xFF
C = 0; result is negative
```

SUBWF **Subtract W from f**

Syntax: [*label*] SUBWF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (f) - (W) → (dest)

Status Affected: C, DC, Z

Encoding:

00	0010	dfff	ffff

Description: Subtract (2's complement method) W register from register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'.

Words: 1

Cycles: 1

Example 1:
```
SUBWF   REG1,1
```
Before Instruction
```
REG1 = 3
W    = 2
C    = ?
```
After Instruction
```
REG1 = 1
W    = 2
C    = 1; result is positive
```

Example 2: Before Instruction
```
REG1 = 2
W    = 2
C    = ?
```
After Instruction
```
REG1 = 0
W    = 2
C    = 1; result is zero
```

Example 3: Before Instruction
```
REG1 = 1
W    = 2
C    = ?
```
After Instruction
```
REG1 = 0xFF
W    = 2
C    = 0; result is negative
```

SWAPF **Swap Nibbles in f**

Syntax: [*label*] SWAPF f,d

Operands: 0 ≤ f ≤ 127
d ∈ [0,1]

Operation: (f<3:0>) → (dest<7:4>),
(f<7:4>) → (dest<3:0>)

Status Affected: None

Encoding:

00	1110	dfff	ffff

Description: The upper and lower nibbles of register 'f' are exchanged. If 'd' is 0 the result is placed in W register. If 'd' is 1 the result is placed in register 'f'.

Words: 1

Cycles: 1

Example

```
SWAPF  REG,   0
```

Before Instruction

REG1 = 0xA5

After Instruction

REG1 = 0xA5
W = 0x5A

TRIS **Load TRIS Register**

Syntax: [*label*] TRIS f

Operands: 5 ≤ f ≤ 7

Operation: (W) → TRIS register f;

Status Affected: None

Encoding:

00	0000	0110	0fff

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

Words: 1

Cycles: 1

Example

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

XORLW **Exclusive OR Literal with W**

Syntax: [*label*] XORLW k

Operands: 0 ≤ k ≤ 255

Operation: (W) .XOR. k → (W)

Status Affected: Z

Encoding:

11	1010	kkkk	kkkk

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

Words: 1

Cycles: 1

Example

```
XORLW  0xAF
```

Before Instruction

W = 0xB5

After Instruction

W = 0x1A

XORWF **Exclusive OR W with f**

Syntax: [*label*] XORWF f,d

Operands: 0 ≤ f ≤ 127
d ∈ [0,1]

Operation: (W) .XOR. (f) → (dest)

Status Affected: Z

Encoding:

00	0110	dfff	ffff

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

Words: 1

Cycles: 1

Example

```
XORWF  REG  1
```

Before Instruction

REG = 0xAF
W = 0xB5

After Instruction

REG = 0x1A
W = 0xB5

NOTES:

11.0 DEVELOPMENT SUPPORT

11.1 Development Tools

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

- PICMASTER/PICMASTER CE Real-Time In-Circuit Emulator
- ICEPIC Low-Cost PIC16C5X and PIC16CXX In-Circuit Emulator
- PRO MATE® II Universal Programmer
- PICSTART® Plus Entry-Level Prototype Programmer
- PICDEM-1 Low-Cost Demonstration Board
- PICDEM-2 Low-Cost Demonstration Board
- PICDEM-3 Low-Cost Demonstration Board
- MPASM Assembler
- MPLAB-SIM Software Simulator
- MPLAB-C (C Compiler)
- Fuzzy logic development system (fuzzyTECH®–MP)

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

The PICMASTER Universal In-Circuit Emulator is intended to provide the product development engineer with a complete microcontroller design tool set for all microcontrollers in the PIC12C5XX, PIC14000, PIC16C5X, PIC16CXX and PIC17CXX families. PICMASTER is supplied with the MPLAB™ Integrated Development Environment (IDE), which allows editing, “make” and download, and source debugging from a single environment.

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

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

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

11.3 ICEPIC: Low-cost PIC16CXX In-Circuit Emulator

ICEPIC is a low-cost in-circuit emulator solution for the Microchip PIC16C5X and PIC16CXX families of 8-bit OTP microcontrollers.

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

11.4 PRO MATE II: Universal Programmer

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

The PRO MATE II has programmable V_{DD} and V_{PP} supplies which allows it to verify programmed memory at V_{DD} min and V_{DD} max for maximum reliability. It has an LCD display for displaying error messages, keys to enter commands and a modular detachable socket assembly to support various package types. In stand-alone mode the PRO MATE II can read, verify or program PIC16C5X, PIC16CXX, PIC17CXX and PIC14000 devices. It can also set configuration and code-protect bits in this mode.

11.5 PICSTART Plus Entry Level Development System

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

PICSTART Plus supports all PIC12C5XX, PIC14000, PIC16C5X, PIC16CXX and PIC17CXX devices with up to 40 pins. Larger pin count devices such as the PIC16C923 and PIC16C924 may be supported with an adapter socket.

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

The PICDEM-1 is a simple board which demonstrates the capabilities of several of Microchip's microcontrollers. The microcontrollers supported are: PIC16C5X (PIC16C54 to PIC16C58A), PIC16C61, PIC16C62X, PIC16C71, PIC16C8X, PIC17C42, PIC17C43 and PIC17C44. All necessary hardware and software is included to run basic demo programs. The users can program the sample microcontrollers provided with the PICDEM-1 board, on a PRO MATE II or PICSTART-16B programmer, and easily test firmware. The user can also connect the PICDEM-1 board to the PICMASTER emulator and download the firmware to the emulator for testing. Additional prototype area is available for the user to build some additional hardware and connect it to the microcontroller socket(s). Some of the features include an RS-232 interface, a potentiometer for simulated analog input, push-button switches and eight LEDs connected to PORTB.

11.7 PICDEM-2 Low-Cost PIC16CXX Demonstration Board

The PICDEM-2 is a simple demonstration board that supports the PIC16C62, PIC16C64, PIC16C65, PIC16C73 and PIC16C74 microcontrollers. All the necessary hardware and software is included to run the basic demonstration programs. The user can program the sample microcontrollers provided with the PICDEM-2 board, on a PRO MATE II programmer or PICSTART-16C, and easily test firmware. The PICMASTER emulator may also be used with the PICDEM-2 board to test firmware. Additional prototype area has been provided to the user for adding additional hardware and connecting it to the microcontroller socket(s). Some of the features include a RS-232 interface, push-button switches, a potentiometer for simulated analog input, a Serial EEPROM to demonstrate usage of the I^2C bus and separate headers for connection to an LCD module and a keypad.

11.8 PICDEM-3 Low-Cost PIC16CXX Demonstration Board

The PICDEM-3 is a simple demonstration board that supports the PIC16C923 and PIC16C924 in the PLCC package. It will also support future 44-pin PLCC microcontrollers with a LCD Module. All the necessary hardware and software is included to run the basic demonstration programs. The user can program the sample microcontrollers provided with the PICDEM-3 board, on a PRO MATE II programmer or PICSTART Plus with an adapter socket, and easily test firmware. The PICMASTER emulator may also be used with the PICDEM-3 board to test firmware. Additional prototype area has been provided to the user for adding hardware and connecting it to the microcontroller socket(s). Some of the features include an RS-232 interface, push-button switches, a potentiometer for simulated analog input, a thermistor and separate headers for connection to an external LCD module and a keypad. Also provided on the PICDEM-3 board is an LCD panel, with 4 commons and 12 segments, that is capable of displaying time, temperature and day of the week. The PICDEM-3 provides an additional RS-232 interface and Windows 3.1 software for showing the demultiplexed LCD signals on a PC. A simple serial interface allows the user to construct a hardware demultiplexer for the LCD signals. PICDEM-3 will be available in the 3rd quarter of 1996.

11.9 MPLAB Integrated Development Environment Software

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

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

MPLAB allows you to:

- Edit your source files (either assembly or 'C')
- One touch assemble (or compile) and download to PIC16/17 tools (automatically updates all project information)
- Debug using:
  - source files
  - absolute listing file
- Transfer data dynamically via DDE (soon to be replaced by OLE)
- Run up to four emulators on the same PC

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

11.10 Assembler (MPASM)

The MPASM Universal Macro Assembler is a PC-hosted symbolic assembler. It supports all microcontroller series including the PIC12C5XX, PIC14000, PIC16C5X, PIC16CXX, and PIC17CXX families.

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

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

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

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

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

11.11 Software Simulator (MPLAB-SIM)

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

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

11.12 C Compiler (MPLAB-C)

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

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

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

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

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

11.14 MP-DriveWay™ – Application Code Generator

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

11.15 SEEVAL® Evaluation and Programming System

The SEEVAL SEEPROM Designer's Kit supports all Microchip 2-wire and 3-wire Serial EEPROMs. The kit includes everything necessary to read, write, erase or program special features of any Microchip SEEPROM product including Smart Serials™ and secure serials. The Total Endurance™ Disk is included to aid in trade-off analysis and reliability calculations. The total kit can significantly reduce time-to-market and result in an optimized system.

11.16 TrueGauge® Intelligent Battery Management

The TrueGauge development tool supports system development with the MTA11200B TrueGauge Intelligent Battery Management IC. System design verification can be accomplished before hardware prototypes are built. User interface is graphically-oriented and measured data can be saved in a file for exporting to Microsoft Excel.

11.17 KEELOQ® Evaluation and Programming Tools

KEELOQ evaluation and programming tools support Microchips HCS Secure Data Products. The HCS evaluation kit includes an LCD display to show changing codes, a decoder to decode transmissions, and a programming interface to program test transmitters.

TABLE 11-1: DEVELOPMENT TOOLS FROM MICROCHIP

Product	** MPLAB™ Integrated Development Environment	MPLAB™ C Compiler	MP-DriveWay Applications Code Generator	fuzzyTECH®-MP Explorer/Edition Fuzzy Logic Dev. Tool	*** PICMASTER®/ PICMASTER-CE In-Circuit Emulator	ICEPIC Low-Cost In-Circuit Emulator	****PRO MATE™ II Universal Microchip Programmer	PICSTART® Lite Ultra Low-Cost Dev. Kit	PICSTART® Plus Low-Cost Universal Dev. Kit
PIC12C508, 509	SW007002	SW006005	—	—	EM167015/ EM167101	—	DV007003	—	DV003001
PIC14000	SW007002	SW006005	—	—	EM147001/ EM147101	—	DV007003	—	DV003001
PIC16C52, 54, 54A, 55, 56, 57, 58A	SW007002	SW006005	SW006006	DV005001/ DV005002	EM167015/ EM167101	EM167201	DV007003	DV162003	DV003001
PIC16C554, 556, 558	SW007002	SW006005	—	DV005001/ DV005002	EM167033/ EM167113	—-	DV007003	—	DV003001
PIC16C61	SW007002	SW006005	SW006006	DV005001/ DV005002	EM167021/ N/A	EM167205	DV007003	DV162003	DV003001
PIC16C62, 62A, 64, 64A	SW007002	SW006005	SW006006	DV005001/ DV005002	EM167025/ EM167103	EM167203	DV007003	DV162002	DV003001
PIC16C620, 621, 622	SW007002	SW006005	SW006006	DV005001/ DV005002	EM167023/ EM167109	EM167202	DV007003	DV162003	DV003001
PIC16C63, 65, 65A, 73, 73A, 74, 74A	SW007002	SW006005	SW006006	DV005001/ DV005002	EM167025/ EM167103	EM167204	DV007003	DV162002	DV003001
PIC16C641, 642, 661, 662*	SW007002	SW006005	—	—	EM167035/ EM167105	—-	DV007003	DV162002	DV003001
PIC16C71	SW007002	SW006005	SW006006	DV005001/ DV005002	EM167027/ EM167105	EM167205	DV007003	DV162003	DV003001
PIC16C710, 711	SW007002	SW006005	SW006006	DV005001/ DV005002	EM167027/ EM167105	—	DV007003	DV162003	DV003001
PIC16C72	SW007002	SW006005	SW006006	—	EM167025/ EM167103	—	DV007003	DV162002	DV003001
PIC16F83	SW007002	SW006005	SW006006	DV005001/ DV005002	EM167029/ EM167107	—	DV007003	DV162003	DV003001
PIC16C84	SW007002	SW006005	SW006006	DV005001/ DV005002	EM167029/ EM167107	EM167206	DV007003	DV162003	DV003001
PIC16F84	SW007002	SW006005	SW006006	DV005001/ DV005002	EM167029/ EM167107	—	DV007003	DV162003	DV003001
PIC16C923, 924*	SW007002	SW006005	SW006006	DV005001/ DV005002	EM167031/ EM167111	—	DV007003	—	DV003001
PIC17C42, 42A, 43, 44	SW007002	SW006005	SW006006	DV005001/ DV005002	EM177007/ EM177107	—	DV007003	—	DV003001

*Contact Microchip Technology for availability date
**MPLAB Integrated Development Environment includes MPLAB-SIM Simulator and MPASM Assembler
***All PICMASTER and PICMASTER-CE ordering part numbers above include PRO MATE II programmer
****PRO MATE socket modules are ordered separately. See development systems ordering guide for specific ordering part numbers

Product	TRUEGAUGE® Development Kit	SEEVAL® Designers Kit	Hopping Code Security Programmer Kit	Hopping Code Security Eval/Demo Kit
All 2 wire and 3 wire Serial EEPROM's	N/A	DV243001	N/A	N/A
MTA11200B	DV114001	N/A	N/A	N/A
HCS200, 300, 301 *	N/A	N/A	PG306001	DM303001

12.0 ELECTRICAL SPECIFICATIONS

Absolute Maximum Ratings †

Ambient Temperature under bias ..... –40° to +125°C

Storage Temperature ..... –65° to +150°C

Voltage on any pin with respect to V_{SS} (except V_{DD} and $\overline{MCLR}$) .....–0.3V to V_{DD} + 0.3V

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

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

Total power Dissipation (Note 1) 1.0W

Maximum Current out of V_{SS} pin .....300 mA

Maximum Current into V_{DD} pin .....250 mA

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

Output Clamp Current, I_{OK} (V_o <0 or V_o>V_{DD}) .....±20 mA

Maximum Output Current sunk by any I/O pin .....25 mA

Maximum Output Current sourced by any I/O pin.....25 mA

Maximum current sunk by PORTA, PORTB, and PORTE (combined) (Note 2).....200 mA

Maximum current sourced by PORTA, PORTB, and PORTE (combined) (Note 2) .....200 mA

Maximum current sunk by PORTC and PORTD (combined) (Note 2).....200 mA

Maximum current sourced by PORTC and PORTD (combined) (Note 2).....200 mA

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

Note 2: PORTD and PORTE are not implemented on the PIC16C641 and PIC16C642.

† **NOTICE**: Stresses above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

TABLE 12-1: CROSS REFERENCE OF DEVICE SPECS FOR OSCILLATOR CONFIGURATIONS AND FREQUENCIES OF OPERATION (COMMERCIAL DEVICES)

OSC	PIC16C641-04 PIC16C642-04 PIC16C661-04 PIC16C662-04	PIC16C641-10 PIC16C642-10 PIC16C661-10 PIC16C662-10	PIC16C641-20 PIC16C642-20 PIC16C661-20 PIC16C662-20	PIC16LC641-04 PIC16LC642-04 PIC16LC661-04 PIC16LC662-04	JW Devices
RC	V_{DD}: 4.0V to 6.0V I_{DD}: 5 mA max. @ 5.5V I_{PD}: 21 µA max. @ 4.0V Freq: 4.0 MHz max.	V_{DD}: 4.5V to 5.5V I_{DD}: 2.7 mA typ. @ 5.5V I_{PD}: 1.5 µA typ. @ 4.0V Freq: 4.0 MHz max.	V_{DD}: 4.5V to 5.5V I_{DD}: 2.7 mA typ. @ 5.5V I_{PD}: 1.5 µA typ. @ 4.0V Freq: 4.0 MHz max.	V_{DD}: 3.0V to 6.0V I_{DD}: 2.0 mA typ. @ 3.0V I_{PD}: 0.9 µA typ. @ 3.0V Freq: 4.0 MHz max.	V_{DD}: 4.0V to 6.0V I_{DD}: 5 mA max. @ 5.5V I_{PD}: 21 µA max. @ 4.0V Freq: 4.0 MHz Max.
XT	V_{DD}: 4.0V to 6.0V I_{DD}: 5 mA max. @ 5.5V I_{PD}: 21 µA max. @ 4.0V Freq: 4.0 MHz max.	V_{DD}: 4.5V to 5.5V I_{DD}: 2.7 mA typ. @ 5.5V I_{PD}: 1.5 µA typ. @ 4.0V Freq: 4.0 MHz max.	V_{DD}: 4.5V to 5.5V I_{DD}: 2.7 mA typ. @ 5.5V I_{PD}: 1.5 µA typ. @ 4.0V Freq: 4.0 MHz max.	V_{DD}: 3.0V to 6.0V I_{DD}: 2.0 mA typ. @ 3.0V I_{PD}: 0.9 µA typ. @ 3.0V Freq: 4.0 MHz max.	V_{DD}: 4.0V to 6.0V I_{DD}: 5 mA max. @ 5.5V I_{PD}: 21 µA max. @ 4.0V Freq: 4.0 MHz max.
HS	V_{DD}: 4.5V to 5.5V I_{DD}: 13.5 mA typ. @ 5.5V I_{PD}: 1.5 µA typ. @ 4.5V Freq: 4.0 MHz max.	V_{DD}: 4.5V to 5.5V I_{DD}: 30 mA max. @ 5.5V I_{PD}: 1.5 µA typ. @ 4.5V Freq: 10 MHz max.	V_{DD}: 4.5V to 5.5V I_{DD}: 30 mA max. @ 5.5V I_{PD}: 1.5 µA typ. @ 4.5V Freq: 20 MHz max.	Do not use in HS mode	V_{DD}: 4.5V to 5.5V I_{DD}: 30 mA max. @ 5.5V I_{PD}: 1.5 µA typ. @ 4.5V Freq: 10 MHz max.
LP	V_{DD}: 4.0V to 6.0V I_{DD}: 52.5 µA typ. @ 32 kHz, 4.0V I_{PD}: 0.9 µA typ. @ 4.0V Freq: 200 kHz max.	Do not use in LP mode	Do not use in LP mode	V_{DD}: 3.0V to 6.0V I_{DD}: 48 µA max. @ 32 kHz, 3.0V I_{PD}: 5.0 µA max. @ 3.0V Freq: 200 kHz max.	V_{DD}: 3.0V to 6.0V I_{DD}: 48 µA max. @ 32 kHz, 3.0V I_{PD}: 5.0 µA max. @ 3.0V Freq: 200 kHz max.

The shaded sections indicate oscillator selections which are tested for functionality, but not for MIN/MAX specifications. It is recommended that the user select the device type that ensures the specifications required.

12.1 DC Characteristics: PIC16C641/642/661/662-04 (Commercial, Industrial, Automotive) PIC16C641/642/661/662-10 (Commercial, Industrial, Automotive) PIC16C641/642/661/662-20 (Commercial, Industrial, Automotive)

Standard Operating Conditions (unless otherwise stated)
Operating temperature –40°C ≤ TA ≤ +85°C for industrial,
0°C ≤ TA ≤ +70°C commercial, and
–40°C ≤ TA ≤ +125°C automotive

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
D001	VDD	Supply Voltage	4.0	–	6.0	V	XT, RC and LP osc configuration
D001A			4.5	–	5.5	V	HS osc configuration
D002*	VDR	RAM Data Retention Voltage(1)	1.5	–	–	V	Device in SLEEP mode
D003	VPOR	VDD start voltage to ensure internal Power-on Reset signal	–	VSS	–	V	See section on Power-on Reset for details
D004*	SVDD	VDD rise rate to ensure internal Power-on Reset signal	0.05	–	–	V/ms	See section on Power-on Reset for details
D005	VBOR	Brown-out Reset Voltage	3.7	4.0	4.3	V	BODEN configuration bit is clear
			3.7	4.0	4.4	V	Automotive
D010	IDD	Supply Current(2)	–	2.7	5	mA	XT and RC osc configuration FOSC = 4 MHz, VDD = 5.5V, WDT disabled (4)
D010A			–	35	70	µA	LP osc configuration, PIC16C64X & PIC16C66X-04 only FOSC = 32 kHz, VDD = 4.0V, WDT disabled
D013			–	13.5	30	mA	HS osc configuration FOSC = 20 MHz, VDD = 5.5V, WDT disabled
		Module Differential Current (5)					
D015	ΔIBOR	Brown-out Reset Current	–	350	425	µA	BODEN bit is clear, VDD = 5.0V
D016	ΔICOMP	Comparator Current for each Comparator	–	–	100	µA	VDD = 4.0V
D017	ΔIVREF	VREF Current	–	–	300	µA	VDD = 4.0V
D021	ΔIWDT	WDT Current	–	6.0	20	µA	VDD = 4.0V
			–	–	25	µA	Automotive
D021	IPD	Power-down Current (3)	–	1.5	21	µA	VDD = 4.0V, WDT disabled
			–	2.5	24	µA	Automotive

* These parameters are characterized but not tested.

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

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

2: The supply current is mainly a function of the operating voltage and frequency. Other factors such as I/O pin loading and switching rate, oscillator type, internal code execution pattern, and temperature also have an impact on the current consumption.
The test conditions for all IDD measurements in active operation mode are:
OSC1 = external square wave, from rail to rail; all I/O pins tri-stated™, pulled to VDD,
$\overline{MCLR}$ = VDD; WDT enabled/disabled as specified.

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

4: For RC osc configuration, current through Rext is not included. The current through the resistor can be estimated by the formula Ir = VDD/2Rext (mA) with Rext in kΩ.

5: The Δ current is the additional current consumed when this peripheral is enabled. This current should be added to the base IDD or IPD measurement.

12.2 DC Characteristics: PIC16LC641/642/661/662-04 (Commercial, Industrial)

Standard Operating Conditions (unless otherwise stated) Operating temperature –40°C ≤ TA ≤ +85°C for industrial and 0°C ≤ TA ≤ +70°C commercial							
Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
D001	VDD	Supply Voltage	3.0	–	6.0	V	XT, RC, and LP osc configuration
D002*	VDR	RAM Data Retention Voltage (1)	1.5	–	–	V	Device in SLEEP mode
D003	VPOR	VDD start voltage to ensure internal Power-on Reset signal	–	VSS	–	V	See section on Power-on Reset for details
D004*	SVDD	VDD rise rate to ensure internal Power-on Reset signal	0.05	–	–	V/ms	See section on Power-on Reset for details
D005	VBOR	Brown-out Reset Voltage	3.7	4.0	4.3	V	BODEN configuration bit is clear
D010	IDD	Supply Current (2)	–	2.0	3.8	mA	XT and RC osc configuration FOSC = 4.0 MHz, VDD = 3.0V, WDT disabled (4)
D010A			–	22.5	48	µA	LP osc configuration FOSC = 32 kHz, VDD = 3.0V, WDT disabled
		Module Differential Current (5)					
D015	ΔIBOR	Brown-out Reset Current	–	350	425	µA	BODEN bit is clear, VDD = 5.0V
D016	ΔICOMP	Comparator Current for each Comparator	–	–	100	µA	VDD = 3.0V
D017	ΔIVREF	VREF Current	–	–	300	µA	VDD = 3.0V
D021	ΔIWDT	WDT Current	–	6.0	20	µA	VDD = 3.0V
D021	IPD	Power-down Current (3)	–	0.9	5	µA	VDD = 3.0V, WDT disabled

* These parameters are characterized but not tested.

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

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

2: The supply current is mainly a function of the operating voltage and frequency. Other factors such as I/O pin loading and switching rate, oscillator type, internal code execution pattern, and temperature also have an impact on the current consumption.
The test conditions for all IDD measurements in active operation mode are:
OSC1=external square wave, from rail to rail; all I/O pins tristated, pulled to VDD,
MCLR = VDD; WDT enabled/disabled as specified.

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

4: For RC osc configuration, current through Rext is not included. The current through the resistor can be estimated by the formula Ir = VDD/2Rext (mA) with Rext in kΩ.

5: The Δ current is the additional current consumed when this peripheral is enabled. This current should be added to the base IDD or IPD measurement.

12.3 DC Characteristics: PIC16C641/661 (Commercial, Industrial, Automotive) PIC16C642/662 (Commercial, Industrial, Automotive) PIC16LC641/661 (Commercial, Industrial) PIC16LC642/662 (Commercial, Industrial)

Standard Operating Conditions (unless otherwise stated)
Operating temperature –40°C ≤ TA ≤ +85°C for industrial,
0°C ≤ TA ≤ +70°C commercial, and
–40°C ≤ TA ≤ +125°C automotive
Operating voltage VDD range as described in DC spec Section 12.1 and 12.2

Param No.	Sym	Characteristic	Min	Typ †	Max	Unit	Conditions
	VIL	**Input Low Voltage**					
		I/O ports					
D030		with TTL buffer	VSS	-	0.15VDD	V	For entire VDD range
			VSS	-	0.8V	V	4.5V ≤ VDD ≤ 5.5V
D031		with Schmitt Trigger input	VSS	-	0.2VDD	V	
D032		MCLR, RA4/T0CKI,OSC1 (in RC mode)	Vss	-	0.2VDD	V	(1)
D033		OSC1 (XT and HS modes)	Vss	-	0.3VDD	V	
		OSC1 (LP modes)	Vss	-	0.6VDD-1.0	V	
	VIH	**Input High Voltage**					
		I/O ports					
D040		with TTL buffer	2.0	-	VDD	V	
D041		with Schmitt Trigger input	0.25VDD to 0.8V	-	VDD	V	
D042		MCLR RA4/T0CKI	0.8VDD	-	VDD	V	
D043		OSC1 (XT, HS, LP modes)	0.7VDD	-	VDD	V	
D043A		OSC1 (RC mode)	0.9VDD	-	-	V	(1)
D070	IPURB	**PORTB weak pull-up current**	50	200	400	µA	VDD = 5.0V, VPIN = VSS
	IIL	**Input Leakage Current(2,3)**					
		I/O ports (Except PORTA)	-	-	±1.0	µA	VSS ≤ VPIN ≤ VDD, pin at hi-impedance
D060		PORTA	-	-	±0.5	µA	Vss ≤ VPIN ≤ VDD, pin at hi-impedance
D061		RA4/T0CKI	-	-	±1.0	µA	Vss ≤ VPIN ≤ VDD
D063		OSC1, MCLR	-	-	±5.0	µA	Vss ≤ VPIN ≤ VDD, XT, HS and LP osc configuration
	VOL	**Output Low Voltage**					
D080		I/O ports	-	-	0.6	V	IOL = 8.5 mA, VDD = 4.5V, -40° to +85°C
			-	-	0.6	V	IOL = 7.0 mA, VDD = 4.5V, +125°C
D083		OSC2/CLKOUT	-	-	0.6	V	IOL = 1.6 mA, VDD = 4.5V, -40° to +85°C
		(RC only)	-	-	0.6	V	IOL = 1.2 mA, VDD = 4.5V, +125°C

* These parameters are characterized but not tested.

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

Note 1: In RC oscillator configuration, the OSC1 pin is a Schmitt Trigger input. It is not recommended that the PIC16C64X & PIC16C66X be driven with external clock in RC mode.

2: The leakage current on the MCLR pin is strongly dependent on applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

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

Standard Operating Conditions (unless otherwise stated)
Operating temperature –40°C ≤ TA ≤ +85°C for industrial,
0°C ≤ TA ≤ +70°C commercial, and
–40°C ≤ TA ≤ +125°C automotive
Operating voltage VDD range as described in DC spec Section 12.1 and 12.2

Param No.	Sym	Characteristic	Min	Typ †	Max	Unit	Conditions
	VOH	Output High Voltage (3)					
D090		I/O ports (Except RA4)	VDD-0.7	-	-	V	IOH = -3.0 mA, VDD = 4.5V, -40° to +85°C
			VDD-0.7	-	-	V	IOH = -2.5 mA, VDD = 4.5V, +125°C
D092		OSC2/CLKOUT	VDD-0.7	-	-	V	IOH = -1.3 mA, VDD=4.5V, -40° to +85°C
		(RC only)	VDD-0.7	-	-	V	IOH = -1.0 mA, VDD = 4.5V, +125°C
		Capacitive Loading Specs on Output Pins					
D100	COSC2	OSC2 pin	-	-	15	pF	In XT, HS and LP modes when external clock used to drive OSC1.
D101	CIO	All I/O pins/OSC2 (in RC mode)	-	-	50	pF	

* These parameters are characterized but not tested.

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

Note 1: In RC oscillator configuration, the OSC1 pin is a Schmitt Trigger input. It is not recommended that the PIC16C64X & PIC16C66X be driven with external clock in RC mode.

2: The leakage current on the $\overline{MCLR}$ pin is strongly dependent on applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

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

TABLE 12-2: COMPARATOR SPECIFICATIONS

Operating Conditions: 3.0V < VDD < 6.0V, -40˚C < TA < +125˚C, unless otherwise stated. Current consumption is specified in Table 12-1.

Characteristics	Sym	Min	Typ	Max	Units	Comments
Input offset voltage		-	± 5.0	± 10	mV	
Input common mode voltage*		0	-	VDD - 1.5	V	
CMRR*		35	-	-	db	
Response Time(1)*		-	150	400 600	ns ns	PIC16C64X/66X PIC16LC64X/66X
Comparator Mode Change to Output Valid*		-	-	10	µs	

* These parameters are characterized but not tested.

Note 1: Response time measured with one comparator input at (VDD - 1.5)/2 while the other input transitions from VSS to VDD.

TABLE 12-3: VOLTAGE REFERENCE SPECIFICATIONS

Operating Conditions: 3.0V < VDD < 6.0V, -40°C < TA < +125°C, unless otherwise stated. Current consumption is specified in Table 12-1.

Characteristics	Sym	Min	Typ	Max	Units	Comments
Resolution		VDD/24	-	VDD/32	LSb	
Absolute Accuracy		- -	- -	1/4 1/2	LSb LSb	Low Range (VRR = 1) High Range (VRR = 0)
Unit Resistor Value (R)*		-	2k	-	Ω	Figure 8-2
Settling Time(1)*		-	-	10	µs	

* These parameters are characterized but not tested.

Note 1: Settling time measured while VRR = 1 and VR<3:0> transitions from 0000 to 1111.

12.4 Timing Parameter Symbology

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

1. TppS2ppS
2. TppS

T	
F Frequency	T Time

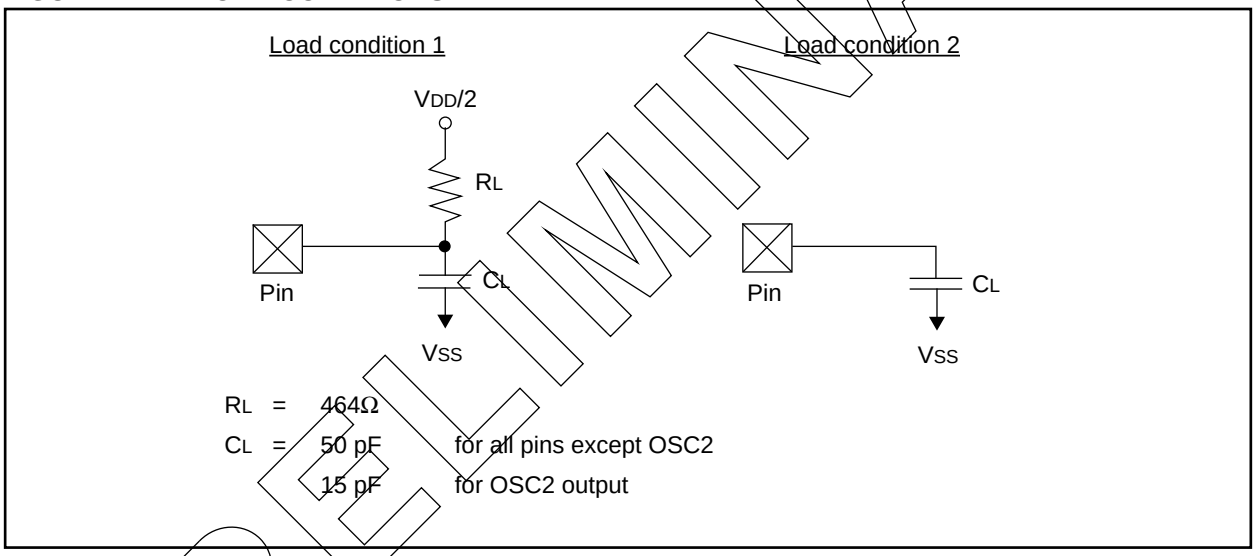
Lowercase subscripts (pp) and their meanings:

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

Uppercase letters and their meanings:

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

FIGURE 12-1: LOAD CONDITIONS



12.5 Timing Diagrams and Specifications

FIGURE 12-2: EXTERNAL CLOCK TIMING

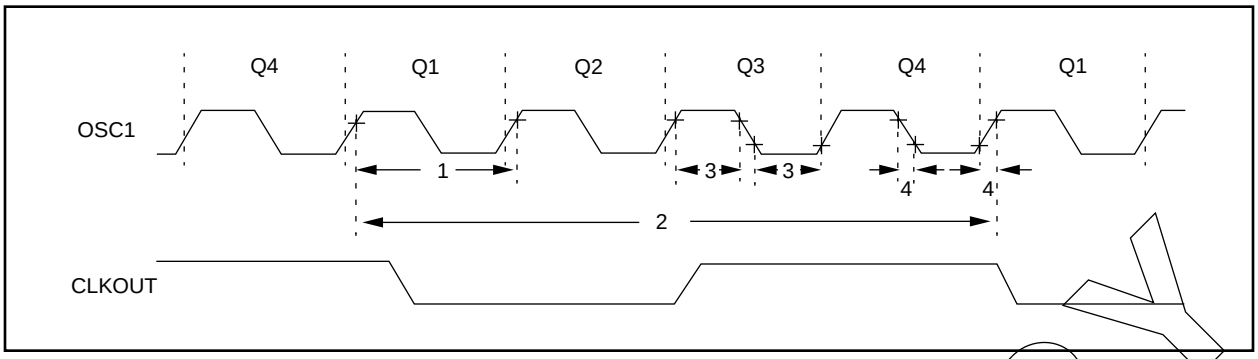


TABLE 12-4: EXTERNAL CLOCK TIMING REQUIREMENTS

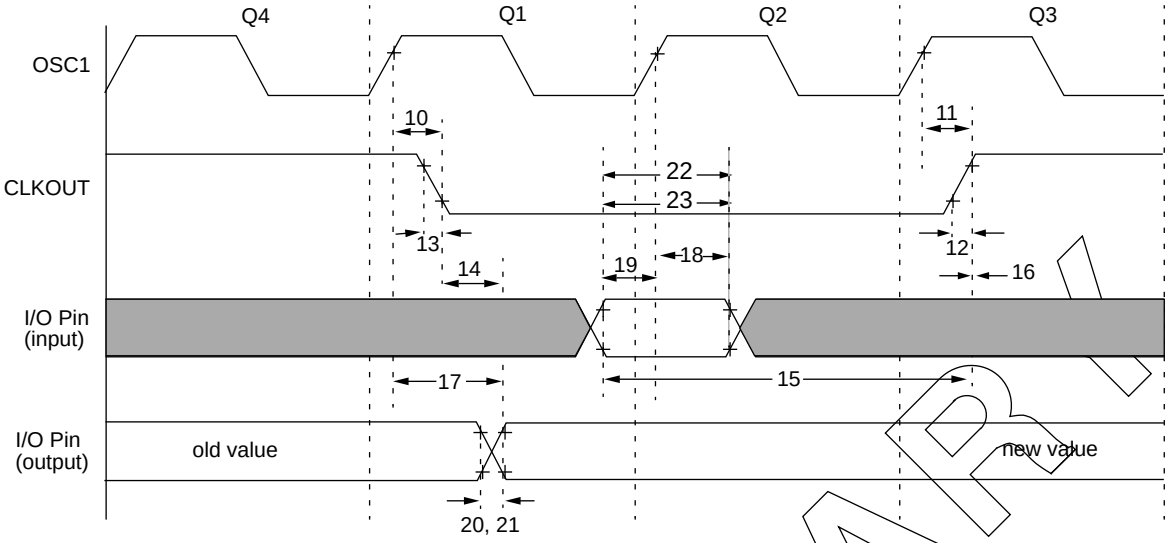
Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
	Fosc	**External CLKIN Frequency(1)**	DC	—	4	MHz	XT and RC osc mode, VDD = 5.0V
			DC	—	20	MHz	HS osc mode
			DC	—	200	kHz	LP osc mode
		Oscillator Frequency (1)	DC	—	4	MHz	RC osc mode, VDD = 5.0V
			0.1	—	4	MHz	XT osc mode
			4	—	20	MHz	HS osc mode
			5	—	200	kHz	LP osc mode
1	Tosc	**External CLKIN Period(1)**	250	—	—	ns	XT and RC osc mode
			50	—	—	ns	HS osc mode
			5	—	—	µs	LP osc mode
		Oscillator Period(1)	250	—	—	ns	RC osc mode
			250	—	10,000	ns	XT osc mode
			50	—	250	ns	HS osc mode
			5	—	—	µs	LP osc mode
2	TCY	**Instruction Cycle Time(1)**	200	—	DC	ns	TCY = FOSC/4
3*	TosL, TosH	**External Clock in (OSC1) High or Low Time**	100	—	—	ns	XT osc mode
			2.5	—	—	µs	LP osc mode
			15	—	—	ns	HS osc mode
4*	TosR, TosF	**External Clock in (OSC1) Rise or Fall Time**	—	—	25	ns	XT osc mode
			—	—	50	ns	LP osc mode
			—	—	15	ns	HS osc mode

* These parameters are characterized but not tested.

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

Note 1: 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.

FIGURE 12-3: CLKOUT AND I/O TIMING



Note: See Figure 12-1 for load conditions.

TABLE 12-5: CLKOUT AND I/O TIMING REQUIREMENTS

Parameter No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
10*	TosH2ckL	OSC1↑ to CLKOUT↓		—	75	200	ns	Note 1
11*	TosH2ckH	OSC1↑ to CLKOUT↑		—	75	200	ns	Note 1
12*	TckR	CLKOUT rise time		—	35	100	ns	Note 1
13*	TckF	CLKOUT fall time		—	35	100	ns	Note 1
14*	TckL2ioV	CLKOUT ↓ to Port out valid		—	—	0.5TCY + 20	ns	Note 1
15*	TioV2ckH	Port in valid before CLKOUT ↑		TOSC + 200	—	—	ns	Note 1
16*	TckH2ioI	Port in hold after CLKOUT ↑		0	—	—	ns	Note 1
17*	TosH2ioV	OSC1↑ (Q1 cycle) to Port out valid		—	50	150	ns	
18*	TosH2ioI	OSC1↑ (Q2 cycle) to Port input invalid (I/O in hold time)	PIC16C64X/66X	100	—	—	ns	
			PIC16LC64X/66X	200	—	—	ns	
19*	TioV2osH	Port input valid to OSC1↑ (I/O in setup time)		0	—	—	ns	
20*	TioR	Port output rise time	PIC16C64X/66X	—	10	40	ns	
			PIC16LC64X/66X	—	—	80	ns	
21*	TioF	Port output fall time	PIC16C64X/66X	—	10	40	ns	
			PIC16LC64X/66X	—	—	80	ns	
22††*	Tinp	INT pin high or low time		TCY	—	—	ns	
23††*	Trbp	RB7:RB4 change INT high or low time		TCY	—	—	ns	

* These parameters are characterized but not tested.

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

†† These parameters are asynchronous events not related to any internal clock edges.

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

FIGURE 12-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, AND POWER-UP TIMER TIMING

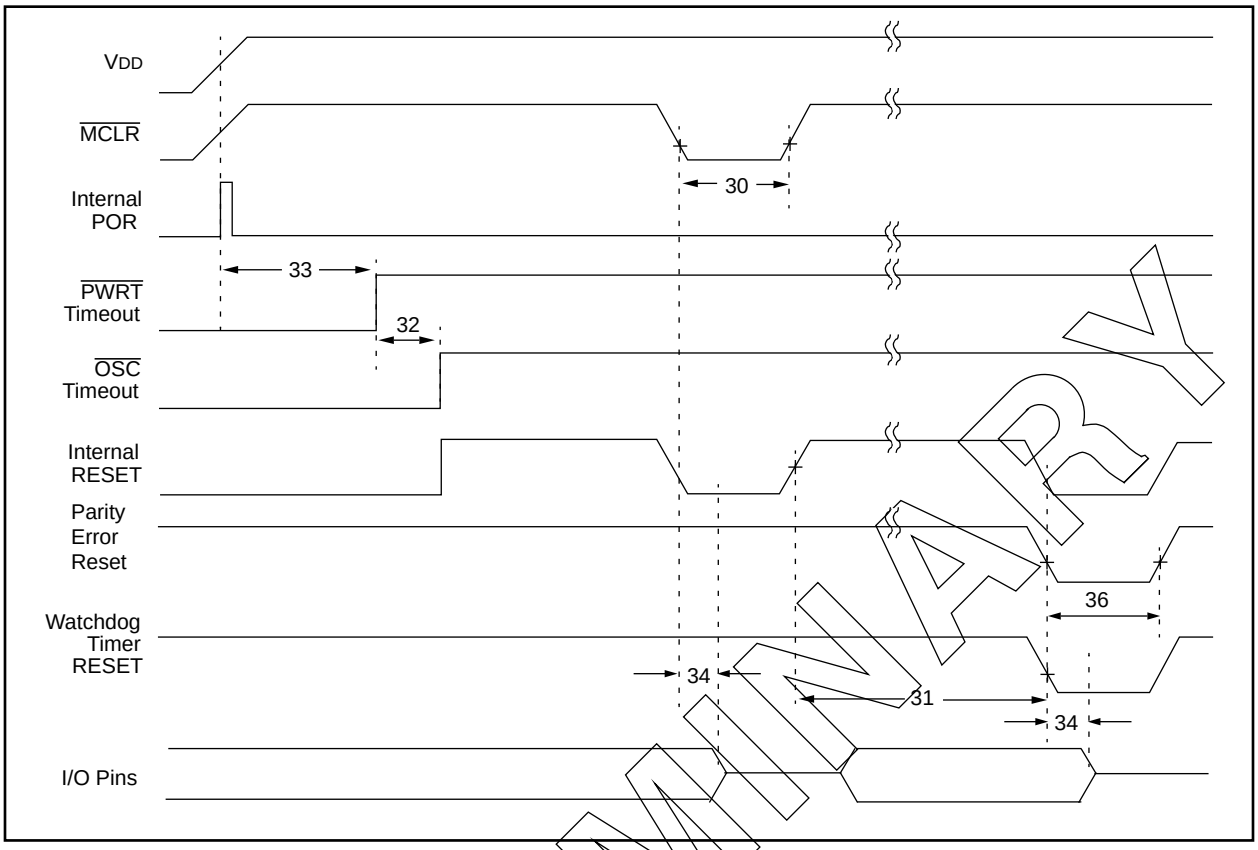


FIGURE 12-5: BROWN-OUT RESET TIMING

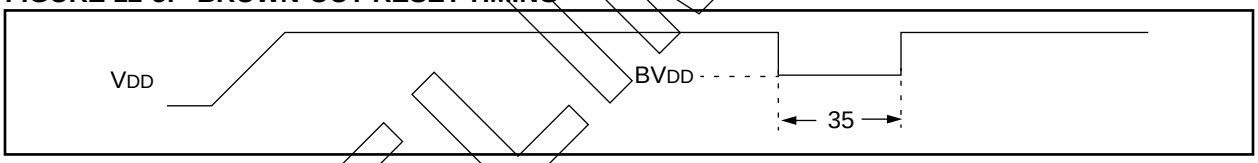


TABLE 12-6: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, POWER-UP TIMER, AND BROWN-OUT RESET REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
30	TmcL	MCLR Pulse Width (low)	2	—	—	µs	VDD = 5V, -40˚C to +125˚C
31*	Twdt	Watchdog Timer Time-out Period (No Prescaler)	7	18	33	ms	VDD = 5V, -40˚C to +125˚C
32	Tost	Oscillation Start-up Timer Period	—	1024TOSC	—	—	TOSC = OSC1 period
33*	Tpwrt	Power up Timer Period	28	72	132	ms	VDD = 5V, -40˚C to +125˚C
34	TIOZ	I/O Hi-impedance from MCLR Low or Watchdog Timer Reset	—	—	2.1	µs	
35	TBOR	Brown-out Reset pulse width	100	—	—	µs	VDD ≤ BVDD (D005)
36	TPER	Parity Error Reset	—	TBD	—	µs	

* These parameters are characterized but not tested.

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

FIGURE 12-6: TIMER0 CLOCK TIMING

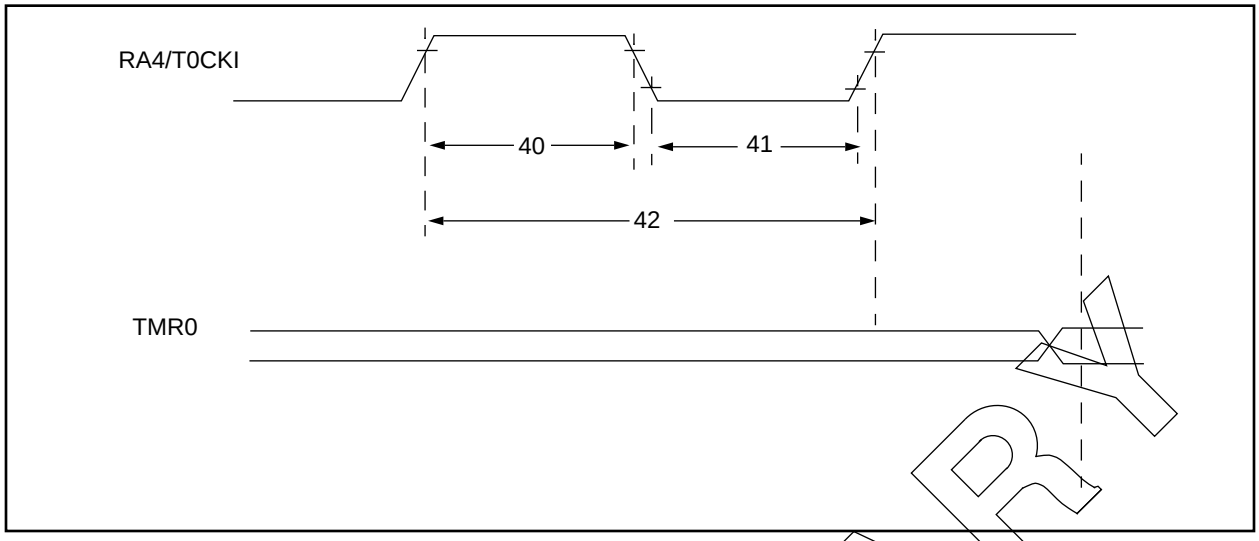


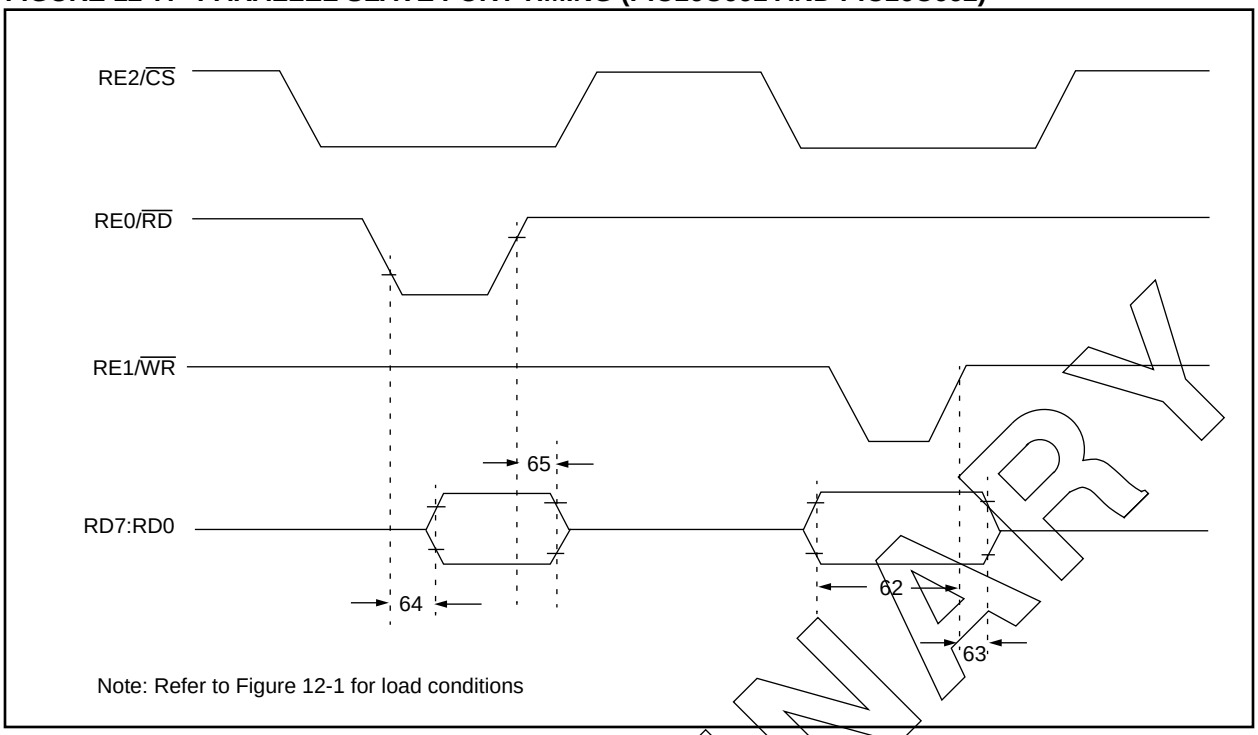
TABLE 12-7: TIMER0 CLOCK REQUIREMENTS

Param No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
40*	Tt0H	T0CKI High Pulse Width	No Prescaler	0.5TCY + 20	—	—	ns	
			With Prescaler	10	—	—	ns	
41*	Tt0L	T0CKI Low Pulse Width	No Prescaler	0.5TCY + 20	—	—	ns	
			With Prescaler	10	—	—	ns	
42*	Tt0P	T0CKI Period		$\frac{T_{CY} + 40}{N}$	—	—	ns	N = prescale value (1, 2, 4, …, 256)

* These parameters are characterized but not tested.

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

FIGURE 12-7: PARALLEL SLAVE PORT TIMING (PIC16C661 AND PIC16C662)



Note: Refer to Figure 12-1 for load conditions

TABLE 12-8: PARALLEL SLAVE PORT REQUIREMENTS (PIC16C661 AND PIC16C662)

Parameter No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
62	TdtV2wrH	Data in valid before $\overline{WR}\uparrow$ or $\overline{CS}\uparrow$ (setup time)		20	—	—	ns	
63*	TwrH2dtI	$\overline{WR}\uparrow$ or $\overline{CS}\uparrow$ to data–in invalid (hold time)	PIC16C66X	20	—	—	ns	
			PIC16LC66X	35	—	—	ns	
64	TrdL2dtV	$\overline{RD}\downarrow$ and $\overline{CS}\downarrow$ to data–out valid		—	—	80	ns	
65	TrdH2dtI	$\overline{RD}\uparrow$ or $\overline{CS}\downarrow$ to data–out invalid		10	—	30	ns	

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

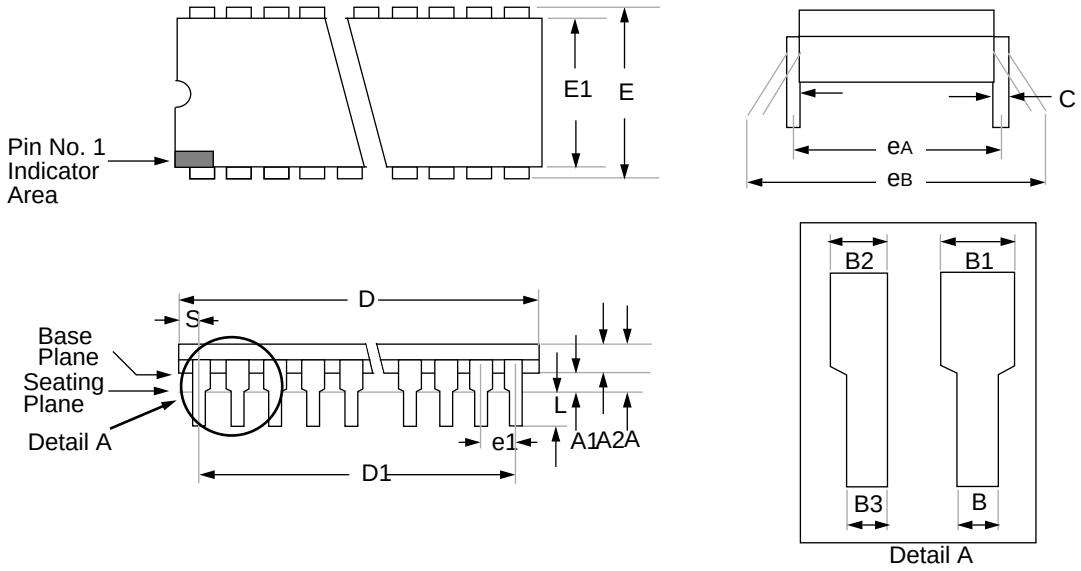
13.0 DEVICE CHARACTERIZATION INFORMATION

NOT AVAILABLE AT THIS TIME.

NOTES:

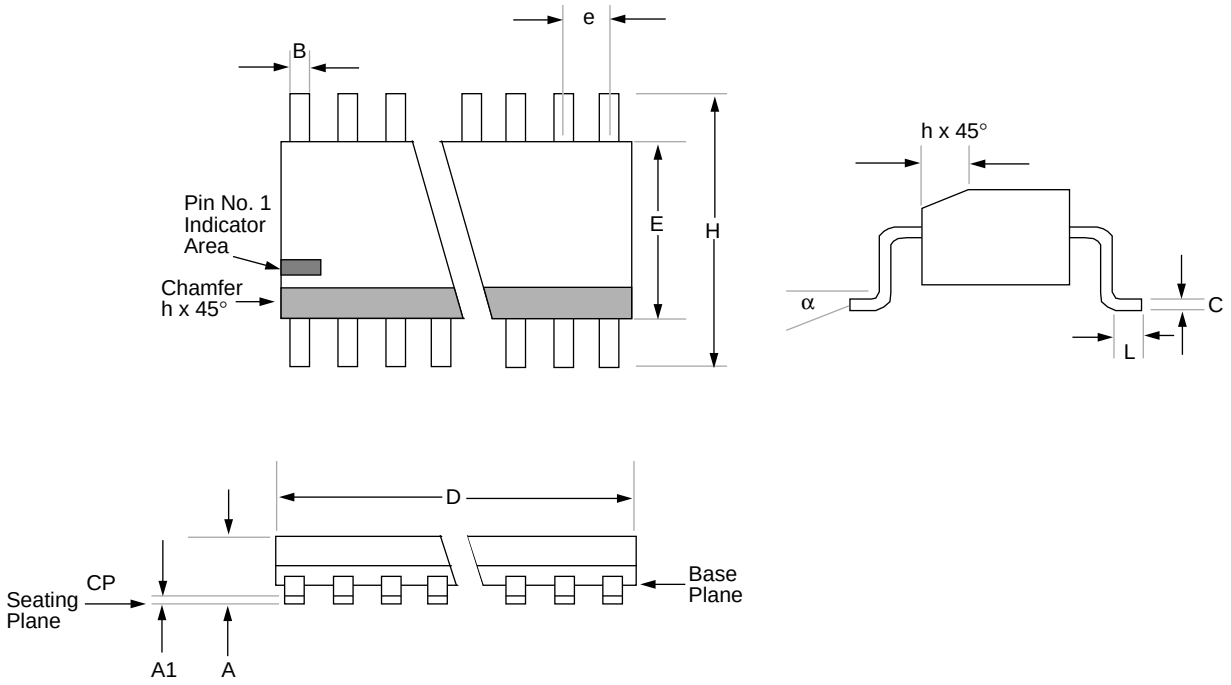
14.0 PACKAGING INFORMATION

Package Type: 28-Lead Skinny Plastic Dual In-Line (SP) - 300 mil



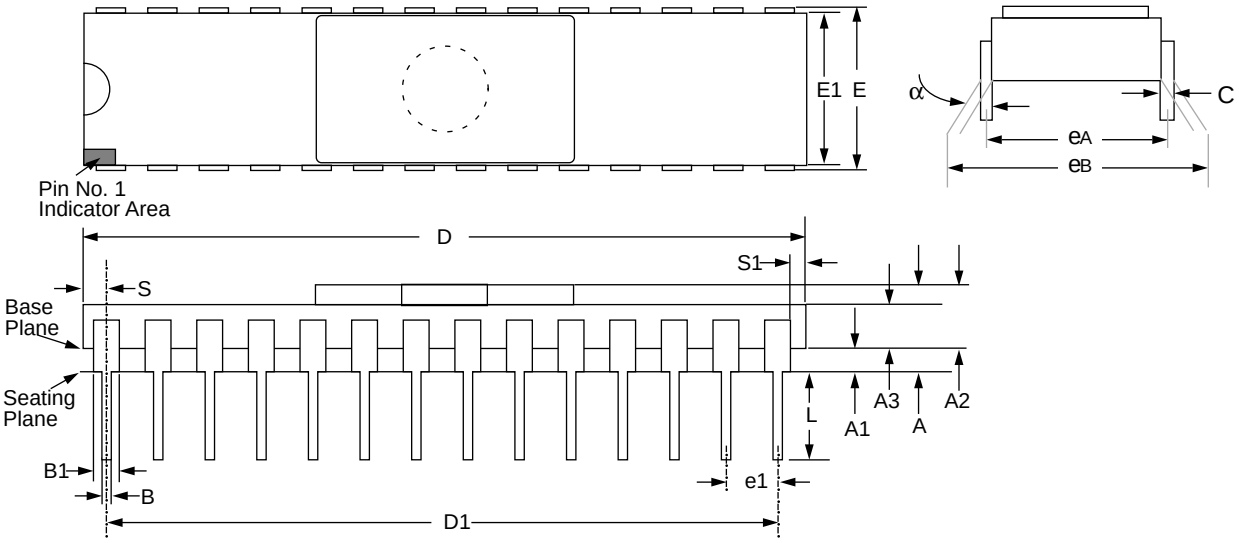
Package Group: Plastic Dual In-Line (PLA)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
A	3.632	4.572		0.143	0.180	
A1	0.381	—		0.015	—	
A2	3.175	3.556		0.125	0.140	
B	0.406	0.559		0.016	0.022	
B1	1.016	1.651	**Typical**	0.040	0.065	**Typical**
B2	0.762	1.016	**4 places**	0.030	0.040	**4 places**
B3	0.203	0.508	**4 places**	0.008	0.020	**4 places**
C	0.203	0.331	**Typical**	0.008	0.013	**Typical**
D	34.163	35.179		1.385	1.395	
D1	33.020	33.020	**BSC**	1.300	1.300	**BSC**
E	7.874	8.382		0.310	0.330	
E1	7.112	7.493		0.280	0.295	
e1	2.540	2.540	**Typical**	0.100	0.100	**Typical**
eA	7.874	7.874	**BSC**	0.310	0.310	**BSC**
eB	8.128	9.906		0.320	0.390	
L	3.175	3.683		0.125	0.145	
S	0.584	1.220		0.023	0.048	

Package Type: 28-Lead Plastic Small Outline (SO) - Wide, 300 mil Body



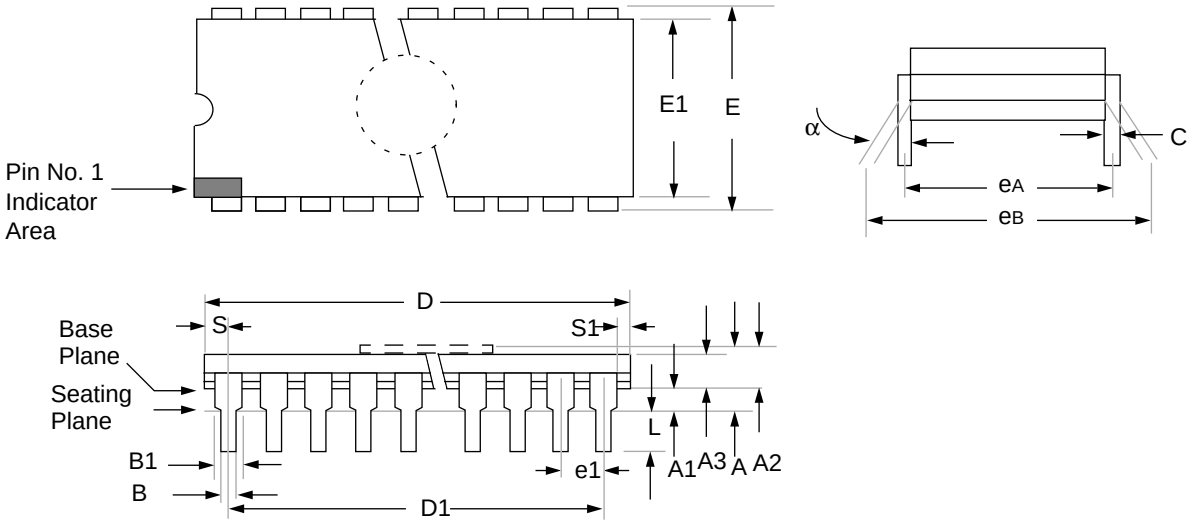
Package Group: Plastic SOIC (SO)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
α	0°	8°		0°	8°	
A	2.362	2.642		0.093	0.104	
A1	0.101	0.300		0.004	0.012	
B	0.355	0.483		0.014	0.019	
C	0.241	0.318		0.009	0.013	
D	17.703	18.085		0.697	0.712	
E	7.416	7.595		0.292	0.299	
e	1.270	1.270	**BSC**	0.050	0.050	**BSC**
H	10.007	10.643		0.394	0.419	
h	0.381	0.762		0.015	0.030	
L	0.406	1.143		0.016	0.045	
CP	—	0.102		—	0.004	

Package Type: 28-Lead Ceramic Side Brazed Dual In-Line with Window (JW) (300 mil)



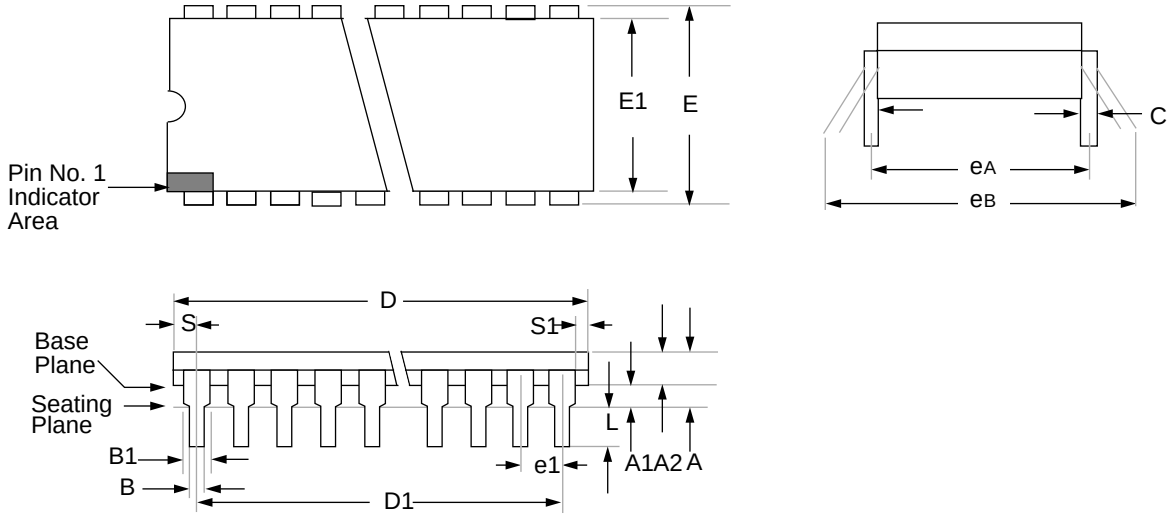
Package Group: Ceramic Side Brazed Dual In-Line (CER)						
Symbol	Millimeters			Inches		
	Min	Max	Notes	Min	Max	Notes
α	0°	10°		0°	10°	
A	3.937	5.030		0.155	0.198	
A1	1.016	1.524		0.040	0.060	
A2	2.921	3.506		0.115	0.138	
A3	1.930	2.388		0.076	0.094	
B	0.406	0.508		0.016	0.020	
B1	1.219	1.321	**Typical**	0.048	0.052	
C	0.228	0.305	**Typical**	0.009	0.012	
D	35.204	35.916		1.386	1.414	
D1	32.893	33.147	**BSC**	1.295	1.305	
E	7.620	8.128		0.300	0.320	
E1	7.366	7.620		0.290	0.300	
e1	2.413	2.667	**Typical**	0.095	0.105	
eA	7.366	7.874	**BSC**	0.290	0.310	
eB	7.594	8.179		0.299	0.322	
L	3.302	4.064		0.130	0.160	
S	1.143	1.397		0.045	0.055	
S1	0.533	0.737		0.021	0.029	

Package Type: 40-Lead Ceramic Dual In-Line with Window (JW) - (600 mil)



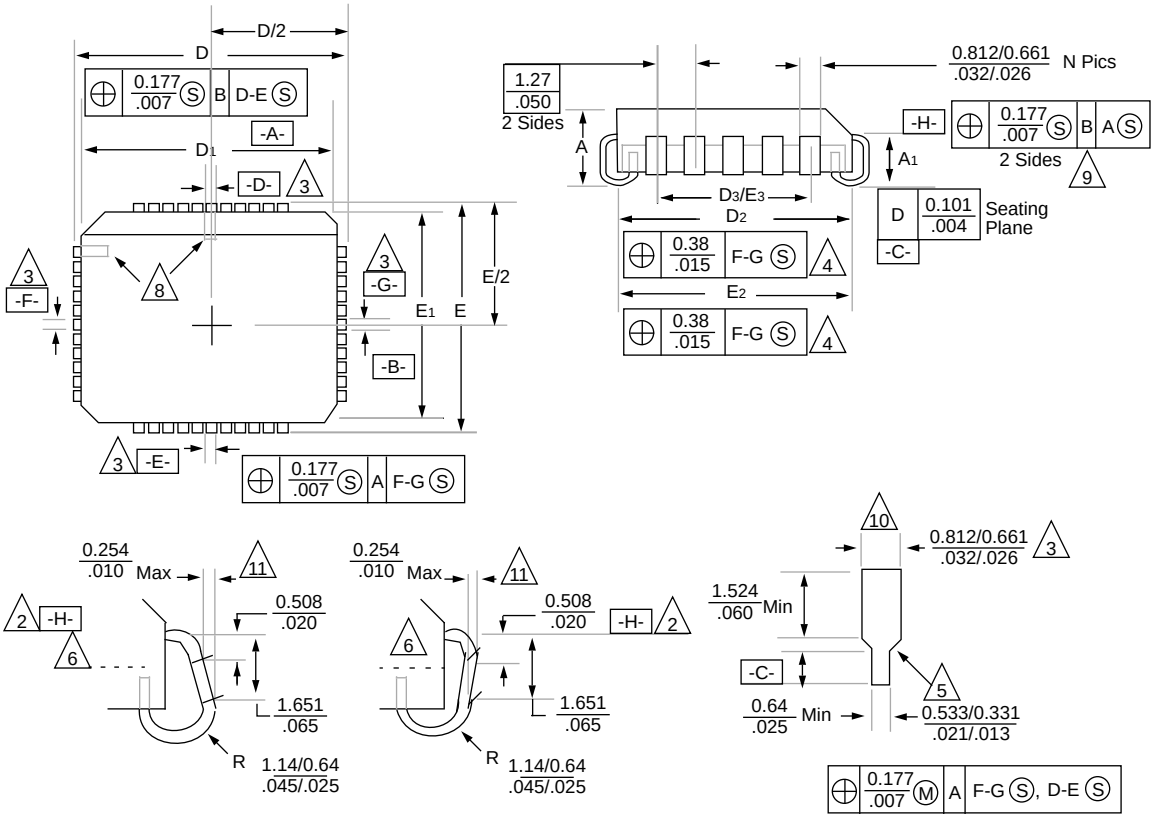
Package Group: Ceramic CERDIP Dual In-Line (CDP)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
α	0°	10°		0°	10°	
A	4.318	5.715		0.170	0.225	
A1	0.381	1.778		0.015	0.070	
A2	3.810	4.699		0.150	0.185	
A3	3.810	4.445		0.150	0.175	
B	0.355	0.585		0.014	0.023	
B1	1.270	1.651	**Typical**	0.050	0.065	**Typical**
C	0.203	0.381	**Typical**	0.008	0.015	**Typical**
D	51.435	52.705		2.025	2.075	
D1	48.260	48.260	**BSC**	1.900	1.900	**BSC**
E	15.240	15.875		0.600	0.625	
E1	12.954	15.240		0.510	0.600	
e1	2.540	2.540	**BSC**	0.100	0.100	**BSC**
eA	14.986	16.002	**Typical**	0.590	0.630	**Typical**
eB	15.240	18.034		0.600	0.710	
L	3.175	3.810		0.125	0.150	
S	1.016	2.286		0.040	0.090	
S1	0.381	1.778		0.015	0.070	

Package Type: 40-Lead Plastic Dual In-Line (P) - 600 mil



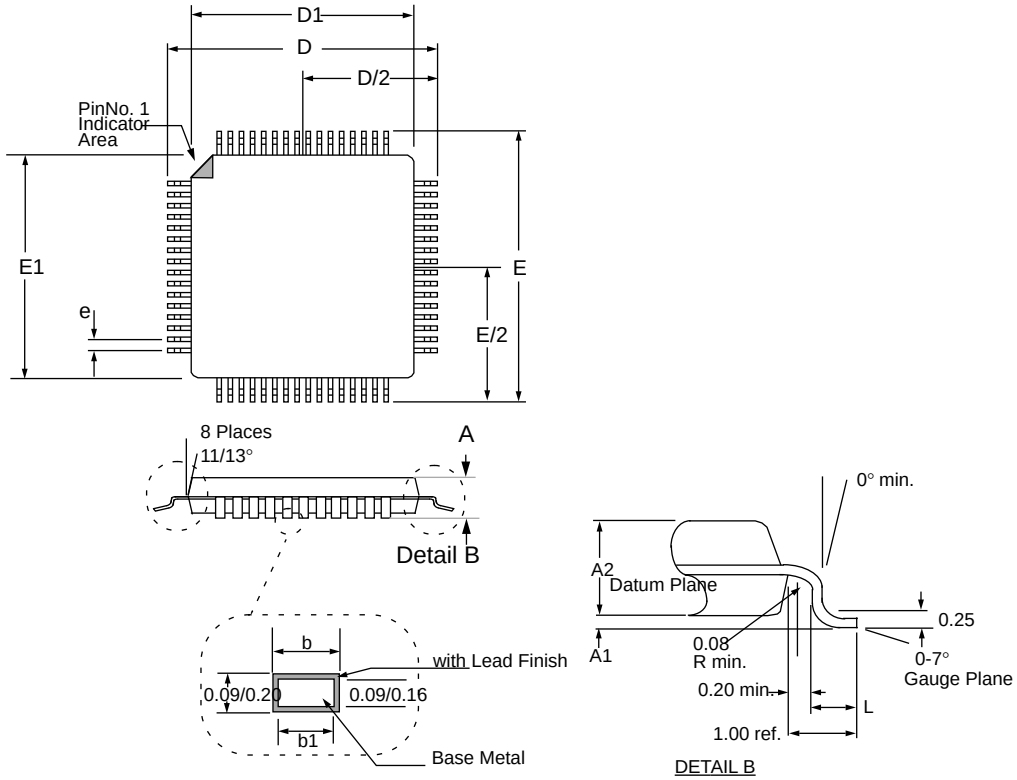
Package Group: Plastic Dual In-Line (PLA)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
A	—	5.080		—	0.200	
A1	0.381	—		0.015	—	
A2	3.175	4.064		0.125	0.160	
B	0.355	0.559		0.014	0.022	
B1	1.270	1.778	**Typical**	0.050	0.070	**Typical**
C	0.203	0.381	**Typical**	0.008	0.015	**Typical**
D	51.181	52.197		2.015	2.055	
D1	48.260	48.260	**BSC**	1.900	1.900	**BSC**
E	15.240	15.875		0.600	0.625	
E1	13.462	13.970		0.530	0.550	
e1	2.489	2.591	**Typical**	0.098	0.102	**Typical**
eA	15.240	15.240	**BSC**	0.600	0.600	**BSC**
eB	15.748	17.272		0.620	0.680	
L	2.921	3.683		0.115	0.145	
S	1.270	—		0.050	—	
S1	0.508	—		0.020	—	

Package Type: 44-Lead Plastic Leaded Chip Carrier (L) - Square



Package Group: Plastic Leaded Chip Carrier (PLCC)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
A	4.191	4.572		0.165	0.180	
A1	2.413	2.921		0.095	0.115	
D	17.399	17.653		0.685	0.695	
D1	16.510	16.663		0.650	0.656	
D2	15.494	16.002		0.610	0.630	
D3	12.700	12.700	**BSC**	0.500	0.500	**BSC**
E	17.399	17.653		0.685	0.695	
E1	16.510	16.663		0.650	0.656	
E2	15.494	16.002		0.610	0.630	
E3	12.700	12.700	**BSC**	0.500	0.500	**BSC**
CP	—	0.102		—	0.004	
LT	0.203	0.381		0.008	0.015	

Package Type: 44-Lead Thin Plastic Quad Flatpack (PT/TQ) - 10x10x1 mm Body 1.0/0.10 mm Lead Form

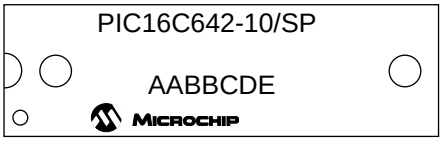


Package Group: Plastic TQFP						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
α	0°	7°		0°	7°	
A	—	1.200		—	0.047	
A1	0.050	0.150		0.002	0.006	
A2	0.950	1.050		0.037	0.041	
b	0.300	0.450		0.012	0.018	
b1	0.300	0.400		0.012	0.016	
D	12.0	12.0	**BSC**	0.472	0.0472	**BSC**
D1	10.0	10.0	**BSC**	0.394	0.394	**BSC**
E	12.0	12.0	**BSC**	0.472	0.472	**BSC**
E1	10.0	10.0	**BSC**	0.394	0.394	**BSC**
e	0.8	0.8	**BSC**	0.031	0.031	**BSC**
L	0.450	0.750		0.018	0.030	

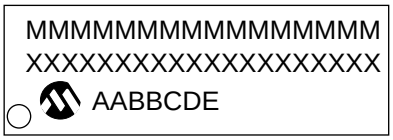
14.1 Package Marking Information

28-Lead PDIP (Skinny DIP)

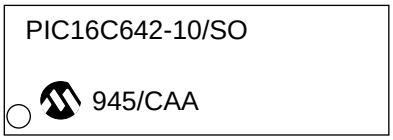
Example



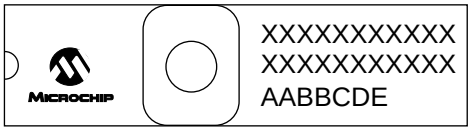
28-Lead SOIC



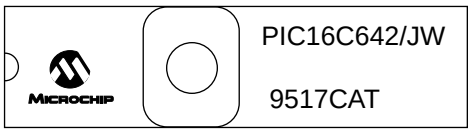
Example



28-Lead Side Brazed Skinny Windowed



Example



Legend:	MM...MMicrochip 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

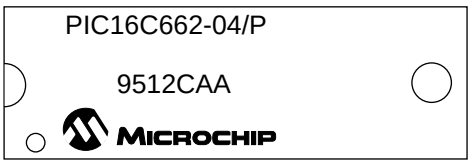
Note:In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line thus limiting the number of available characters for customer specific information.

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

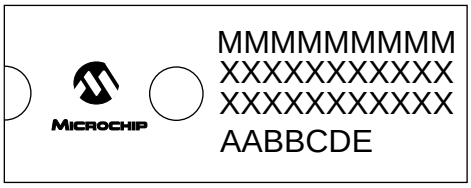
14.2 Package Marking Information

40-Lead PDIP

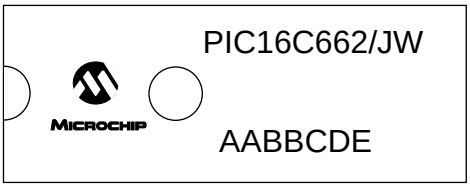
Example



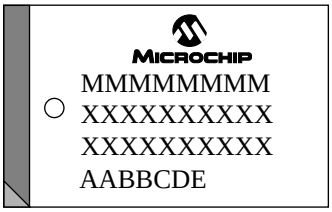
40-Lead CERDIP Windowed



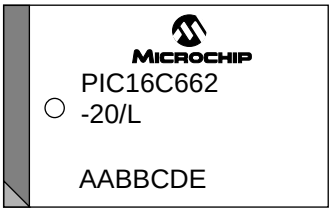
Example



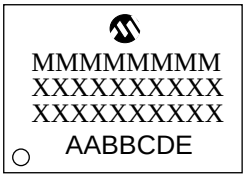
44-Lead PLCC



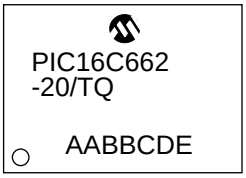
Example



44-Lead TQFP



Example



Legend:	MM...MMicrochip 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 cannot be marked on one line, it will be carried over to the next line thus limiting the number of available characters for customer specific information.

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

NOTES:

APPENDIX A: ENHANCEMENTS

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

1. Instruction word length is increased to 14 bits. This allows larger page sizes both in program memory (4K now as opposed to 512 before) and register file (up to 176 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 slightly redefined. STATUS register is modified.
4. Four new instructions have been added: `RETURN`, `RETFIE`, `ADDLW`, and `SUBLW`.
   Two instructions `TRIS` and `OPTION` are being phased out although they are kept for compatibility with PIC16C5X.
5. `OPTION` and `TRIS` registers are made addressable.
6. Interrupt capability is added. Interrupt vector is at 0004h.
7. Stack size is increased to 8 deep.
8. Reset vector is changed to 0000h.
9. Reset of all registers is revisited. Six 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 can be invoked selectively to avoid unnecessary delays on power-up and wake-up.
12. PORTB has weak pull-ups and interrupt on change feature.
13. Timer0 clock input, T0CKI pin is also a port pin (RA4/T0CKI) and has a TRIS bit.
14. FSR is made a full 8-bit register.
15. “In-circuit programming” is made possible. The user can program PIC16CXX devices using only five pins: V_{DD}, V_{SS}, V_{PP}, RB6 (clock) and RB7 (data in/out).
16. PCON status register is added with a Power-on Reset status bit ($\overline{POR}$), a Brown-out Reset status bit ($\overline{BOR}$), a Parity Error Reset ($\overline{PER}$), and a Memory Parity Enable (MPEEN) bit.
17. Code protection scheme is enhanced such that portions of the program memory can be protected, while the remainder is unprotected.
18. PORTA inputs are now Schmitt Trigger inputs.
19. Brown-out Reset circuitry has been added.

APPENDIX B: COMPATIBILITY

To convert code written for PIC16C5X to PIC16CXX, 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.

APPENDIX C: WHAT'S NEW

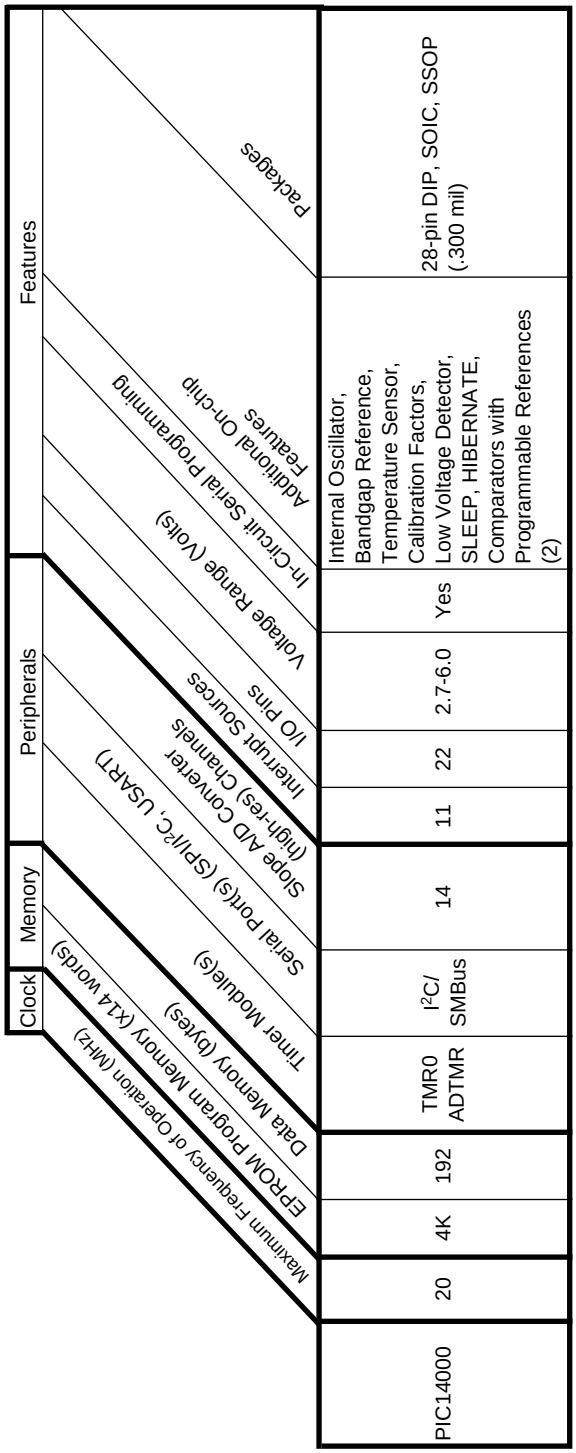
New Data Sheet

APPENDIX D: WHAT'S CHANGED

New Data Sheet

APPENDIX E: PIC16/17 MICROCONTROLLERS

E.1 PIC14000 Devices



E.2 PIC16C5X Family of Devices

	Clock	Memory				Peripherals		Features	
	Maximum Frequency of Operation (MHz)	Program Memory (x12 words)	EPROM	ROM	RAM Data Memory (bytes)	Timer Module(s)	I/O Pins	Voltage Range (Volts)	Number of Instructions	Packages
PIC16C52	4	384	—	25	TMR0	12	2.5-6.25	33	18-pin DIP, SOIC
PIC16C54	20	512	—	25	TMR0	12	2.5-6.25	33	18-pin DIP, SOIC; 20-pin SSOP
PIC16C54A	20	512	—	25	TMR0	12	2.0-6.25	33	18-pin DIP, SOIC; 20-pin SSOP
PIC16CR54A	20	—	512	25	TMR0	12	2.0-6.25	33	18-pin DIP, SOIC; 20-pin SSOP
PIC16C55	20	512	—	24	TMR0	20	2.5-6.25	33	28-pin DIP, SOIC, SSOP
PIC16C56	20	1K	—	25	TMR0	12	2.5-6.25	33	18-pin DIP, SOIC; 20-pin SSOP
PIC16C57	20	2K	—	72	TMR0	20	2.5-6.25	33	28-pin DIP, SOIC, SSOP
PIC16CR57B	20	—	2K	72	TMR0	20	2.5-6.25	33	28-pin DIP, SOIC, SSOP
PIC16C58A	20	2K	—	73	TMR0	12	2.0-6.25	33	18-pin DIP, SOIC; 20-pin SSOP
PIC16CR58A	20	—	2K	73	TMR0	12	2.5-6.25	33	18-pin DIP, SOIC; 20-pin SSOP

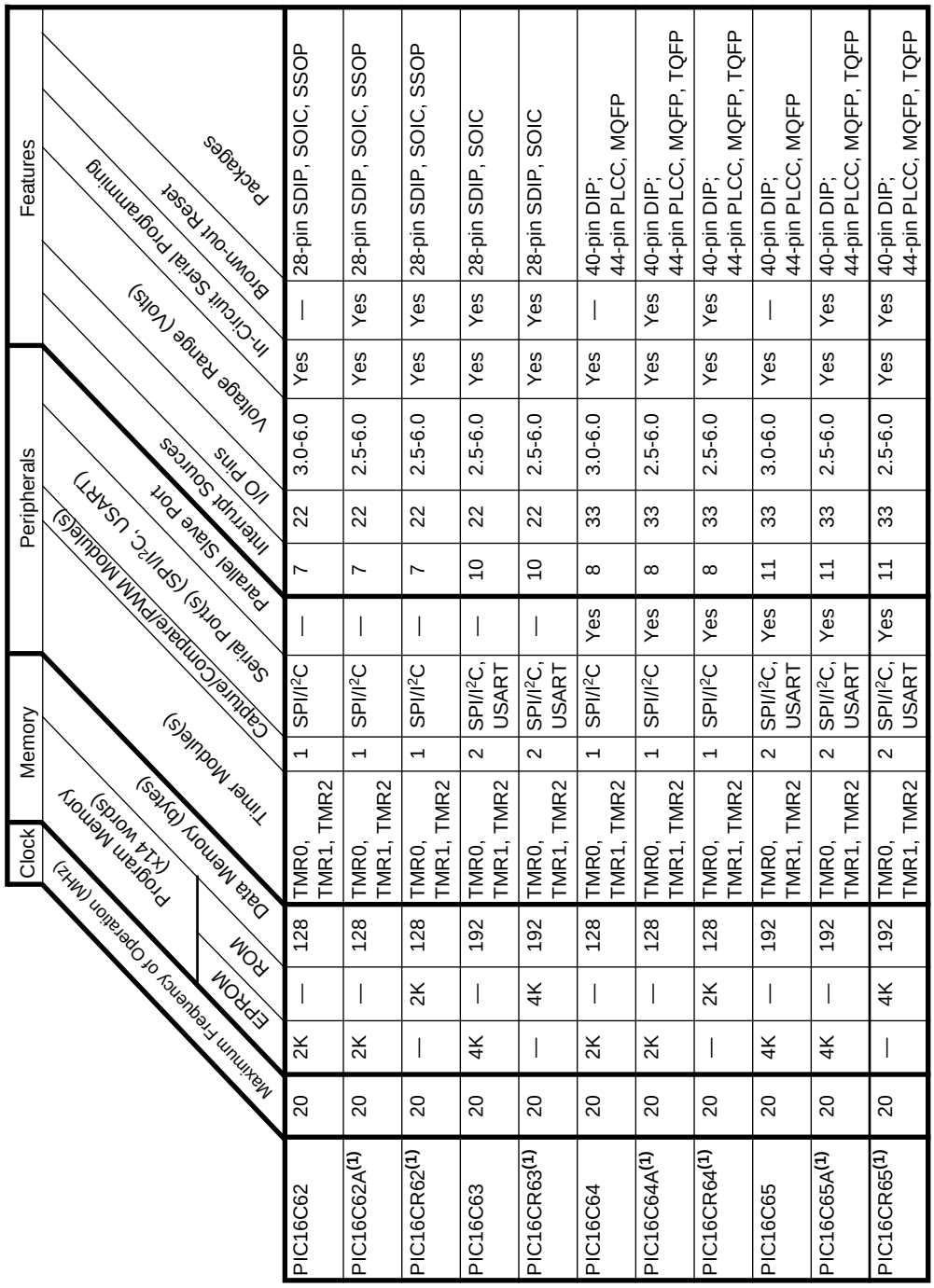
All PIC16/17 Family devices have Power-On Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability.

E.3 PIC16CXXX Family of Devices

	Clock	Memory			Peripherals			Features	
	Maximum Frequency of Operation (MHz)	EPROM	Data Memory (bytes)	Timer Module(s)	Comparator(s)	Interrupt Sources	I/O Pins	Voltage Range (Volts)	Brown-out Reset	Packages
PIC16C554	20	512	80	TMR0	—	3	13	2.5-6.0	—	18-pin DIP, SOIC; 20-pin SSOP
PIC16C556	20	1K	80	TMR0	—	3	13	2.5-6.0	—	18-pin DIP, SOIC; 20-pin SSOP
PIC16C558	20	2K	128	TMR0	—	3	13	2.5-6.0	—	18-pin DIP, SOIC; 20-pin SSOP
PIC16C620	20	512	80	TMR0	2	Yes	4	13	2.5-6.0	Yes	18-pin DIP, SOIC; 20-pin SSOP
PIC16C621	20	1K	80	TMR0	2	Yes	4	13	2.5-6.0	Yes	18-pin DIP, SOIC; 20-pin SSOP
PIC16C622	20	2K	128	TMR0	2	Yes	4	13	2.5-6.0	Yes	18-pin DIP, SOIC; 20-pin SSOP
PIC16C641	20	2K	128	TMR0	2	Yes	4	22	3.0-6.0	Yes	28-pin PDIP, SOIC Windowed CDIP
PIC16C642	20	4K	176	TMR0	2	Yes	4	22	3.0-6.0	Yes	28-pin PDIP, SOIC Windowed CDIP
PIC16C661	20	2K	128	TMR0	2	Yes	5	33	3.0-6.0	Yes	40-pin PDIP, Windowed CDIP; 44-pin PLCC, MQFP
PIC16C662	20	4K	176	TMR0	2	Yes	5	33	3.0-6.0	Yes	40-pin PDIP, Windowed CDIP; 44-pin PLCC, MQFP

All PIC16/17 Family devices have Power-on Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability.
All PIC16C6XXX Family devices use serial programming with clock pin RB6 and data pin RB7.

E.4 PIC16C6X Family of Devices



	Clock	Memory				Peripherals						Features			
	Maximum Frequency of Operation (MHz)	EPROM	ROM	Data Memory (bytes)	Timer Module(s)	Capture/Compare/PWM Module(s)	Serial Port(s) (SPI/I²C, USART)	Parallel Slave Port	Interrupt Sources	I/O Pins	Voltage Range (Volts)	In-Circuit Serial Programming	Brown-out Reset	Packages
PIC16C62	20	2K	—	128	TMR0, TMR1, TMR2	1	SPI/I²C	—	7	22	3.0-6.0	Yes	—	28-pin SDIP, SOIC, SSOP
PIC16C62A(1)	20	2K	—	128	TMR0, TMR1, TMR2	1	SPI/I²C	—	7	22	2.5-6.0	Yes	Yes	28-pin SDIP, SOIC, SSOP
PIC16CR62(1)	20	—	2K	128	TMR0, TMR1, TMR2	1	SPI/I²C	—	7	22	2.5-6.0	Yes	Yes	28-pin SDIP, SOIC, SSOP
PIC16C63	20	4K	—	192	TMR0, TMR1, TMR2	2	SPI/I²C, USART	—	10	22	2.5-6.0	Yes	Yes	28-pin SDIP, SOIC
PIC16CR63(1)	20	—	4K	192	TMR0, TMR1, TMR2	2	SPI/I²C, USART	—	10	22	2.5-6.0	Yes	Yes	28-pin SDIP, SOIC
PIC16C64	20	2K	—	128	TMR0, TMR1, TMR2	1	SPI/I²C	Yes	8	33	3.0-6.0	Yes	—	40-pin DIP; 44-pin PLCC, MQFP
PIC16C64A(1)	20	2K	—	128	TMR0, TMR1, TMR2	1	SPI/I²C	Yes	8	33	2.5-6.0	Yes	Yes	40-pin DIP; 44-pin PLCC, MQFP, TQFP
PIC16CR64(1)	20	—	2K	128	TMR0, TMR1, TMR2	1	SPI/I²C	Yes	8	33	2.5-6.0	Yes	Yes	40-pin DIP; 44-pin PLCC, MQFP, TQFP
PIC16C65	20	4K	—	192	TMR0, TMR1, TMR2	2	SPI/I²C, USART	Yes	11	33	3.0-6.0	Yes	—	40-pin DIP; 44-pin PLCC, MQFP
PIC16C65A(1)	20	4K	—	192	TMR0, TMR1, TMR2	2	SPI/I²C, USART	Yes	11	33	2.5-6.0	Yes	Yes	40-pin DIP; 44-pin PLCC, MQFP, TQFP
PIC16CR65(1)	20	—	4K	192	TMR0, TMR1, TMR2	2	SPI/I²C, USART	Yes	11	33	2.5-6.0	Yes	Yes	40-pin DIP; 44-pin PLCC, MQFP, TQFP

All PIC16/17 family devices have Power-on Reset, selectable Watchdog Timer, selectable code protect, and high I/O current capability.
All PIC16C6X family devices use serial programming with clock pin RB6 and data pin RB7.

Note 1: Please contact your local sales office for availability of these devices.

E.5 PIC16C7X Family of Devices

	Clock	Memory			Peripherals							Features			
	Maximum Frequency of Operation (MHz)	EPROM Program Memory (x14 words)	Data Memory (bytes)	Timer Module(s)	Capture/Compare/PWM Module(s)	Serial Port(s) (SPI/I²C, USART)	Parallel Slave Port	A/D Converter (8-bit) Channels	Interrupt Sources	I/O Pins	Voltage Range (Volts)	In-Circuit Serial Programming	Brown-out Reset	Packages
PIC16C710	20	512	36	TMR0	—	—	—	4	4	13	2.5-6.0	Yes	Yes	18-pin DIP, SOIC; 20-pin SSOP
PIC16C71	20	1K	36	TMR0	—	—	—	4	4	13	2.5-6.0	Yes	—	18-pin DIP, SOIC
PIC16C711	20	1K	68	TMR0	—	—	—	4	4	13	2.5-6.0	Yes	Yes	18-pin DIP, SOIC; 20-pin SSOP
PIC16C72	20	2K	128	TMR0, TMR1, TMR2	1	SPI/I²C	—	5	8	22	2.5-6.0	Yes	Yes	28-pin SDIP, SOIC, SSOP
PIC16C73	20	4K	192	TMR0, TMR1, TMR2	2	SPI/I²C, USART	—	5	11	22	2.5-6.0	Yes	—	28-pin SDIP, SOIC
PIC16C73A	20	4K	192	TMR0, TMR1, TMR2	2	SPI/I²C, USART	—	5	11	22	2.5-6.0	Yes	Yes	28-pin SDIP, SOIC
PIC16C74	20	4K	192	TMR0, TMR1, TMR2	2	SPI/I²C, USART	Yes	8	12	33	2.5-6.0	Yes	—	40-pin DIP; 44-pin PLCC, MQFP
PIC16C74A	20	4K	192	TMR0, TMR1, TMR2	2	SPI/I²C, USART	Yes	8	12	33	2.5-6.0	Yes	Yes	40-pin DIP; 44-pin PLCC, MQFP, TQFP

All PIC16/17 Family devices have Power-on Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability.
All PIC16C7X Family devices use serial programming with clock pin RB6 and data pin RB7.

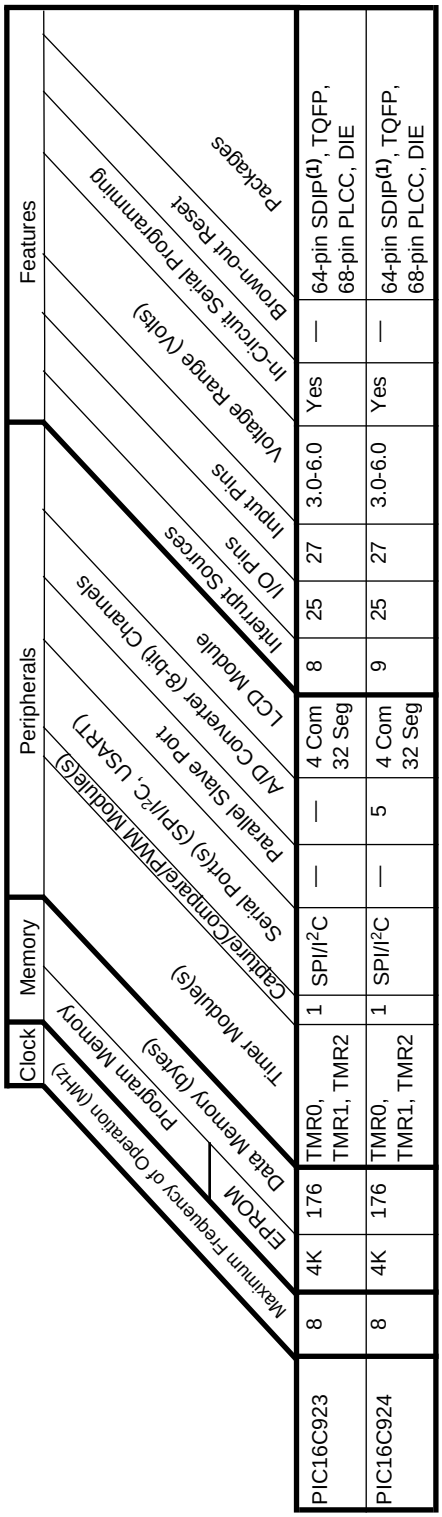
E.6 PIC16C8X Family of Devices

	Clock	Memory					Peripherals		Features	
	Maximum Frequency of Operation (MHz)	Program Memory	Flash	EEPROM	ROM	Data Memory (bytes)	Data EEPROM (bytes)	Timer Module(s)	Interrupt Sources	I/O Pins	Voltage Range (Volts)	Packages
PIC16C84	10	—	1K	—	36	64	TMR0	4	13	2.0-6.0	18-pin DIP, SOIC
PIC16F84(1)	10	1K	—	—	68	64	TMR0	4	13	2.0-6.0	18-pin DIP, SOIC
PIC16CR84(1)	10	—	—	1K	68	64	TMR0	4	13	2.0-6.0	18-pin DIP, SOIC
PIC16F83(1)	10	512	—	—	36	64	TMR0	4	13	2.0-6.0	18-pin DIP, SOIC
PIC16CR83(1)	10	—	—	512	36	64	TMR0	4	13	2.0-6.0	18-pin DIP, SOIC

All PIC16/17 family devices have Power-on Reset, selectable Watchdog Timer, selectable code protect, and high I/O current capability.
All PIC16C8X family devices use serial programming with clock pin RB6 and data pin RB7.

Note 1: Please contact your local sales office for availability of these devices.

E.7 PIC16C9XX Family Of Devices



All PIC16/17 Family devices have Power-on Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability.
All PIC16CXX Family devices use serial programming with clock pin RB6 and data pin RB7.

Note 1: Please contact your local Microchip representative for availability of this package.

E.8 PIC17CXX Family of Devices

	Clock	Memory				Peripherals				Features			
	Maximum Frequency of Operation (MHz)	EPROM	ROM	Program Memory (Words)	RAM Data Memory (bytes)	Timer Module(s)	Captures PWMs	Serial Port(s) (USART)	Hardware Multiply	External Interrupts	I/O Pins	Voltage Range (Volts)	Number of Instructions	Packages
PIC17C42	25	2K	—	232		TMR0,TMR1, TMR2,TMR3	2 2	Yes	—	Yes	33	4.5-5.5	55	40-pin DIP; 44-pin PLCC, MQFP
PIC17C42A	25	2K	—	232		TMR0,TMR1, TMR2,TMR3	2 2	Yes	Yes	Yes	33	2.5-6.0	58	40-pin DIP; 44-pin PLCC, TQFP, MQFP
PIC17CR42	25	—	2K	232		TMR0,TMR1, TMR2,TMR3	2 2	Yes	Yes	Yes	33	2.5-6.0	58	40-pin DIP; 44-pin PLCC, TQFP, MQFP
PIC17C43	25	4K	—	454		TMR0,TMR1, TMR2,TMR3	2 2	Yes	Yes	Yes	33	2.5-6.0	58	40-pin DIP; 44-pin PLCC, TQFP, MQFP
PIC17CR43	25	—	4K	454		TMR0,TMR1, TMR2,TMR3	2 2	Yes	Yes	Yes	33	2.5-6.0	58	40-pin DIP; 44-pin PLCC, TQFP, MQFP
PIC17C44	25	8K		454		TMR0,TMR1, TMR2,TMR3	2 2	Yes	Yes	Yes	33	2.5-6.0	58	40-pin DIP; 44-pin PLCC, TQFP, MQFP

All PIC16/17 Family devices have Power-on Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability.

PIN COMPATIBILITY

Devices that have the same package type and VDD, VSS and $\overline{MCLR}$ pin locations are said to be pin compatible. This allows these different devices to operate in the same socket. Compatible devices may only requires minor software modification to allow proper operation in the application socket (ex., PIC16C56 and PIC16C61 devices). Not all devices in the same package size are pin compatible; for example, the PIC16C62 is compatible with the PIC16C63, but not the PIC16C55.

Pin compatibility does not mean that the devices offer the same features. As an example, the PIC16C54 is pin compatible with the PIC16C71, but does not have an A/D converter, weak pull-ups on PORTB, or interrupts.

TABLE E-1: PIN COMPATIBLE DEVICES

Pin Compatible Devices	Package
PIC12C508, PIC12C509	8-pin
PIC16C54, PIC16C54A, PIC16CR54A, PIC16C56, PIC16C58A, PIC16CR58A, PIC16C61, PIC16C554, PIC16C556, PIC16C558 PIC16C620, PIC16C621, PIC16C622, PIC16C710, PIC16C71, PIC16C711, PIC16F83, PIC16CR83, PIC16C84, PIC16F84A, PIC16CR84	18-pin 20-pin
PIC16C55, PIC16C57, PIC16CR57B	28-pin
PIC16C62, PIC16CR62, PIC16C62A, PIC16C63, PIC16C72, PIC16C73, PIC16C73A	28-pin
PIC16C64, PIC16CR64, PIC16C64A, PIC16C65, PIC16C65A, PIC16C74, PIC16C74A	40-pin
PIC17C42, PIC17CR42, PIC17C42A, PIC17C43, PIC17CR43, PIC17C44	40-pin
PIC16C923, PIC16C924	64/68-pin

NOTES:

INDEX

LIST OF EXAMPLES

LIST OF FIGURES

LIST OF TABLES

ON-LINE SUPPORT

Microchip provides two methods of on-line support. These are the Microchip BBS and the Microchip World Wide Web (WWW) site.

Use Microchip's Bulletin Board Service (BBS) to get current information and help about Microchip products. Microchip provides the BBS communication channel for you to use in extending your technical staff with microcontroller and memory experts.

To provide you with the most responsive service possible, the Microchip systems team monitors the BBS, posts the latest component data and software tool updates, provides technical help and embedded systems insights, and discusses how Microchip products provide project solutions.

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

Connecting to the Microchip Internet Web Site

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

www.microchip.com

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

ftp.mchip.com/biz/mchip

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

- Latest Microchip Press Releases
- Technical Support Section with Frequently Asked Questions
- Design Tips
- Device Errata
- Job Postings
- Microchip Consultant Program Member Listing
- Links to other useful web sites related to Microchip Products

Connecting to the Microchip BBS

Connect worldwide to the Microchip BBS using either the Internet or the CompuServe® communications network.

Internet:

You can telnet or ftp to the Microchip BBS at the address:

mchipbbs.microchip.com

CompuServe Communications Network:

When using the BBS via the Compuserve Network, in most cases, a local call is your only expense. The Microchip BBS connection does not use CompuServe membership services, therefore you do not need CompuServe membership to join Microchip's BBS. There is no charge for connecting to the Microchip BBS.

The procedure to connect will vary slightly from country to country. Please check with your local CompuServe agent for details if you have a problem. CompuServe service allow multiple users various baud rates depending on the local point of access.

The following connect procedure applies in most locations.

1. Set your modem to 8-bit, No parity, and One stop (8N1). This is not the normal CompuServe setting which is 7E1.
2. Dial your local CompuServe access number.
3. Depress the <Enter> key and a garbage string will appear because CompuServe is expecting a 7E1 setting.
4. Type +, depress the <Enter> key and “Host Name:” will appear.
5. Type MCHIPBBS, depress the <Enter> key and you will be connected to the Microchip BBS.

In the United States, to find the CompuServe phone number closest to you, set your modem to 7E1 and dial (800) 848-4480 for 300-2400 baud or (800) 331-7166 for 9600-14400 baud connection. After the system responds with “Host Name:”, type NETWORK, depress the <Enter> key and follow CompuServe's directions.

For voice information (or calling from overseas), you may call (614) 723-1550 for your local CompuServe number.

Microchip regularly uses the Microchip BBS to distribute technical information, application notes, source code, errata sheets, bug reports, and interim patches for Microchip systems software products. For each SIG, a moderator monitors, scans, and approves or disapproves files submitted to the SIG. No executable files are accepted from the user community in general to limit the spread of computer viruses.

Systems Information and Upgrade Hot Line

The Systems Information and Upgrade Line provides system users a listing of the latest versions of all of Microchip's development systems software products. Plus, this line provides information on how customers can receive any currently available upgrade kits.The Hot Line Numbers are:

1-800-755-2345 for U.S. and most of Canada, and

1-602-786-7302 for the rest of the world.

READER RESPONSE

It is our intention to provide you with the best documentation possible to ensure successful use of your Microchip product. If you wish to provide your comments on organization, clarity, subject matter, and ways in which our documentation can better serve you, please FAX your comments to the Technical Publications Manager at (602) 786-7578.

Please list the following information, and use this outline to provide us with your comments about this Data Sheet.

To: Technical Publications Manager Total Pages Sent _____

RE: Reader Response

From: Name _____

Company _____

Address _____

City / State / ZIP / Country _____

Telephone: (_____) _____ - _____ FAX: (_____) _____ - _____

Application (optional):

Would you like a reply? ___Y ___N

Device: **PIC16C64X & PIC16C66X** Literature Number: **DS30559A**

Questions:

1. What are the best features of this document?

2. How does this document meet your hardware and software development needs?

3. Do you find the organization of this data sheet easy to follow? If not, why?

4. What additions to the data sheet do you think would enhance the structure and subject?

5. What deletions from the data sheet could be made without affecting the overall usefulness?

6. Is there any incorrect or misleading information (what and where)?

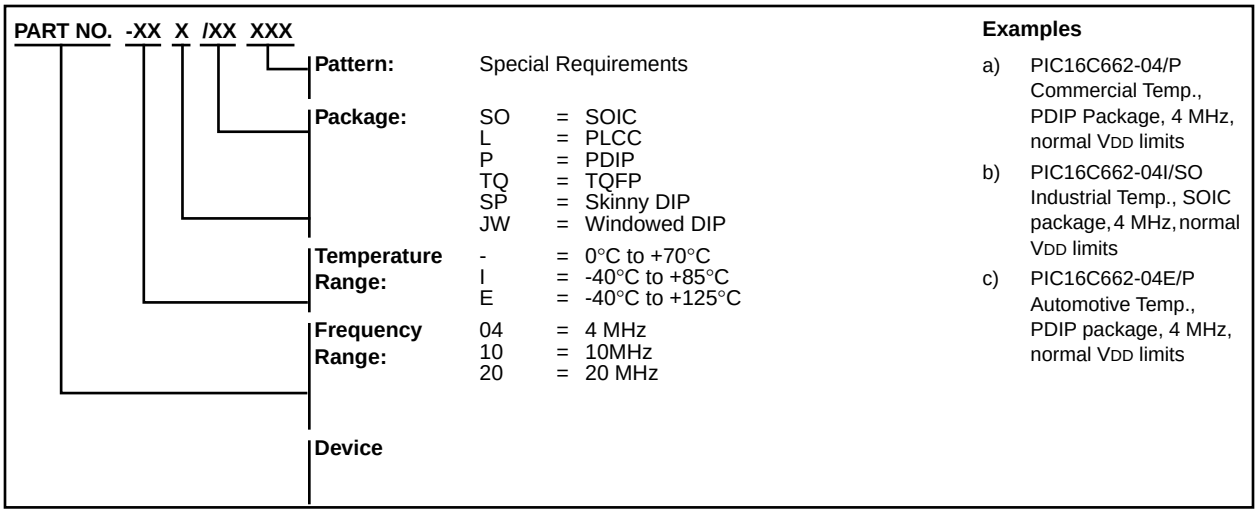
7. How would you improve this document?

8. How would you improve our software, systems, and silicon products?

NOTES:

NOTES:

PIC16C64X & PIC16C66X PRODUCT IDENTIFICATION SYSTEM



Please contact your local sales office for exact ordering procedures.

JW devices are UV erasable and can be programmed to any device configuration. JW devices meet the electrical requirements of each oscillator type (including LC devices).

Sales and Support

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

1. Your local Microchip sales office (see below)
2. The Microchip Corporate Literature Center U.S. FAX: (602) 786-7277
3. The Microchip's Bulletin Board, via your local CompuServe number (CompuServe membership NOT required).

Please specify which device, revision of silicon and Data Sheet (include Literature #) you are using.
For latest version information and upgrade kits for Microchip Development Tools, please call 1-800-755-2345 or 1-602-786-7302.

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

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

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

Trademarks

DNV Certification, Inc. USA
ANSI-RAB QMS ACCREDITED
DNV
DNV MSC The Netherlands Accredited by the RvA
ISO 9001 / QS-9000 REGISTERED FIRM

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

WORLDWIDE SALES AND SERVICE

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200 Fax: 480-792-7277
Technical Support: 480-792-7627
Web Address: http://www.microchip.com

Rocky Mountain
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7966 Fax: 480-792-7456

Atlanta
500 Sugar Mill Road, Suite 200B
Atlanta, GA 30350
Tel: 770-640-0034 Fax: 770-640-0307

Boston
2 Lan Drive, Suite 120
Westford, MA 01886
Tel: 978-692-3848 Fax: 978-692-3821

Chicago
333 Pierce Road, Suite 180
Itasca, IL 60143
Tel: 630-285-0071 Fax: 630-285-0075

Dallas
4570 Westgrove Drive, Suite 160
Addison, TX 75001
Tel: 972-818-7423 Fax: 972-818-2924

Detroit
Tri-Atria Office Building
32255 Northwestern Highway, Suite 190
Farmington Hills, MI 48334
Tel: 248-538-2250 Fax: 248-538-2260

Kokomo
2767 S. Albright Road
Kokomo, Indiana 46902
Tel: 765-864-8360 Fax: 765-864-8387

Los Angeles
18201 Von Karman, Suite 1090
Irvine, CA 92612
Tel: 949-263-1888 Fax: 949-263-1338

New York
150 Motor Parkway, Suite 202
Hauppauge, NY 11788
Tel: 631-273-5305 Fax: 631-273-5335

San Jose
Microchip Technology Inc.
2107 North First Street, Suite 590
San Jose, CA 95131
Tel: 408-436-7950 Fax: 408-436-7955

Toronto
6285 Northam Drive, Suite 108
Mississauga, Ontario L4V 1X5, Canada
Tel: 905-673-0699 Fax: 905-673-6509

ASIA/PACIFIC

Australia
Microchip Technology Australia Pty Ltd
Suite 22, 41 Rawson Street
Epping 2121, NSW
Australia
Tel: 61-2-9868-6733 Fax: 61-2-9868-6755

China - Beijing
Microchip Technology Consulting (Shanghai) Co., Ltd., Beijing Liaison Office
Unit 915
Bei Hai Wan Tai Bldg.
No. 6 Chaoyangmen Beidajie
Beijing, 100027, No. China
Tel: 86-10-85282100 Fax: 86-10-85282104

China - Chengdu
Microchip Technology Consulting (Shanghai) Co., Ltd., Chengdu Liaison Office
Rm. 2401, 24th Floor,
Ming Xing Financial Tower
No. 88 TIDU Street
Chengdu 610016, China
Tel: 86-28-6766200 Fax: 86-28-6766599

China - Fuzhou
Microchip Technology Consulting (Shanghai) Co., Ltd., Fuzhou Liaison Office
Unit 28F, World Trade Plaza
No. 71 Wusi Road
Fuzhou 350001, China
Tel: 86-591-7503506 Fax: 86-591-7503521

China - Shanghai
Microchip Technology Consulting (Shanghai) Co., Ltd.
Room 701, Bldg. B
Far East International Plaza
No. 317 Xian Xia Road
Shanghai, 200051
Tel: 86-21-6275-5700 Fax: 86-21-6275-5060

China - Shenzhen
Microchip Technology Consulting (Shanghai) Co., Ltd., Shenzhen Liaison Office
Rm. 1315, 13/F, Shenzhen Kerry Centre,
Renminnan Lu
Shenzhen 518001, China
Tel: 86-755-2350361 Fax: 86-755-2366086

Hong Kong
Microchip Technology Hongkong Ltd.
Unit 901-6, Tower 2, Metroplaza
223 Hing Fong Road
Kwai Fong, N.T., Hong Kong
Tel: 852-2401-1200 Fax: 852-2401-3431

India
Microchip Technology Inc.
India Liaison Office
Divyasree Chambers
1 Floor, Wing A (A3/A4)
No. 11, O'Shaugnessey Road
Bangalore, 560 025, India
Tel: 91-80-2290061 Fax: 91-80-2290062

Japan
Microchip Technology Japan K.K.
Benex S-1 6F
3-18-20, Shinyokohama
Kohoku-Ku, Yokohama-shi
Kanagawa, 222-0033, Japan
Tel: 81-45-471- 6166 Fax: 81-45-471-6122

Korea
Microchip Technology Korea
168-1, Youngbo Bldg. 3 Floor
Samsung-Dong, Kangnam-Ku
Seoul, Korea 135-882
Tel: 82-2-554-7200 Fax: 82-2-558-5934

Singapore
Microchip Technology Singapore Pte Ltd.
200 Middle Road
#07-02 Prime Centre
Singapore, 188980
Tel: 65-334-8870 Fax: 65-334-8850

Taiwan
Microchip Technology Taiwan
11F-3, No. 207
Tung Hua North Road
Taipei, 105, Taiwan
Tel: 886-2-2717-7175 Fax: 886-2-2545-0139

EUROPE

Denmark
Microchip Technology Nordic ApS
Regus Business Centre
Lautrup hoj 1-3
Ballerup DK-2750 Denmark
Tel: 45 4420 9895 Fax: 45 4420 9910

France
Microchip Technology SARL
Parc d'Activite du Moulin de Massy
43 Rue du Saule Trapu
Batiment A - ler Etage
91300 Massy, France
Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79

Germany
Microchip Technology GmbH
Gustav-Heinemann Ring 125
D-81739 Munich, Germany
Tel: 49-89-627-144 0 Fax: 49-89-627-144-44

Italy
Microchip Technology SRL
Centro Direzionale Colleoni
Palazzo Taurus 1 V. Le Colleoni 1
20041 Agrate Brianza
Milan, Italy
Tel: 39-039-65791-1 Fax: 39-039-6899883

United Kingdom
Arizona Microchip Technology Ltd.
505 Eskdale Road
Winnersh Triangle
Wokingham
Berkshire, England RG41 5TU
Tel: 44 118 921 5869 Fax: 44-118 921-5820

01/18/02