



Welcome to E-XFL.COM

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

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

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

Details

Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	20MHz
Connectivity	-
Peripherals	Brown-out Detect/Reset, LED, POR, WDT
Number of I/O	22
Program Memory Size	7KB (4K x 14)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	176 x 8
Voltage - Supply (Vcc/Vdd)	4V ~ 6V
Data Converters	-
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	28-DIP (0.300", 7.62mm)
Supplier Device Package	28-SPDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16c642-20i-sp

PIC16C64X & PIC16C66X

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
- fuzzyTECH-MP Fuzzy Logic Development Tools
- DriveWay Visual Programming Tool

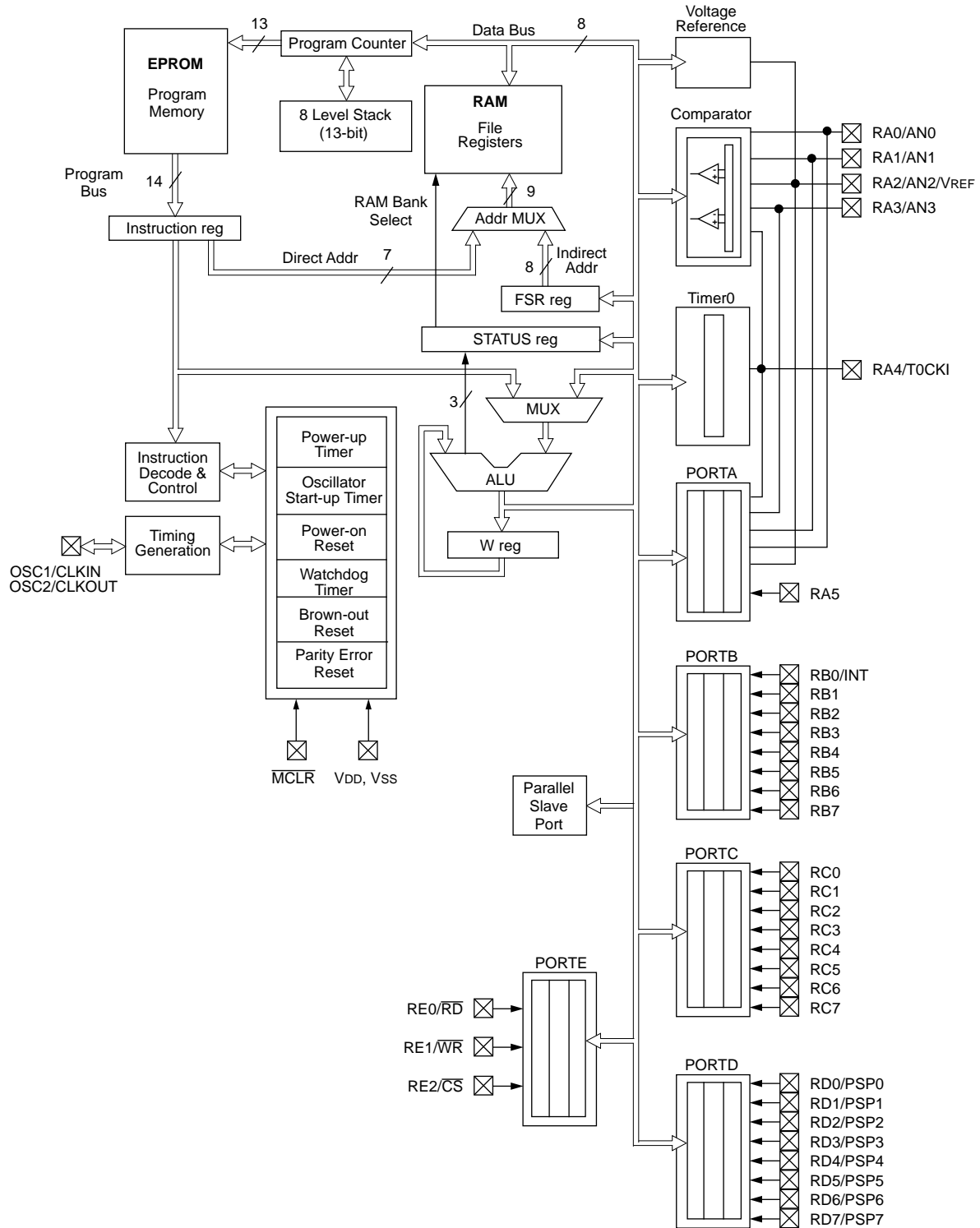
Please refer to Section 11.0 for more details about these and other Microchip development tools.

PIC16C64X & PIC16C66X

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



PIC16C64X & PIC16C66X

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.
MCLR/VPP	1	I/P	ST	Master clear (reset) input or programming voltage input. This pin is an active low reset to the device.
RA0/AN0	2	I/O	ST	PORTA is a bi-directional I/O port. Analog comparator input. Analog comparator input. Analog comparator input or VREF output. Analog comparator input or comparator output. Can be selected to be the clock input to the Timer0 timer/counter or a comparator output. Output is open drain type.
RA1/AN1	3	I/O	ST	
RA2/AN2/VREF	4	I/O	ST	
RA3/AN3	5	I/O	ST	
RA4/T0CKI	6	I/O	ST	
RA5	7	I/O	ST	
RB0/INT	21	I/O	TTL/ST ⁽¹⁾	PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-ups on all inputs. RB0 can also be selected as an external interrupt pin. Interrupt on change pin. Interrupt on change pin. Interrupt on change pin. Serial programming clock. Interrupt on change pin. Serial programming data.
RB1	22	I/O	TTL	
RB2	23	I/O	TTL	
RB3	24	I/O	TTL	
RB4	25	I/O	TTL	
RB5	26	I/O	TTL	
RB6	27	I/O	TTL/ST ⁽²⁾	
RB7	28	I/O	TTL/ST ⁽²⁾	
RC0	11	I/O	ST	PORTC is a bi-directional I/O port.
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.

PIC16C64X & PIC16C66X

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.

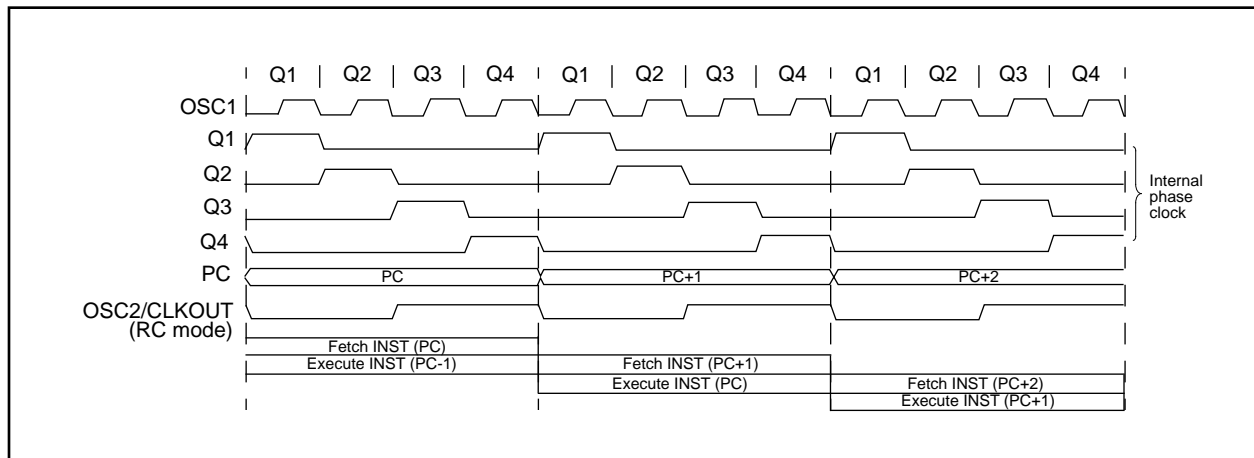
3.2 Instruction Flow/Pipelining

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).

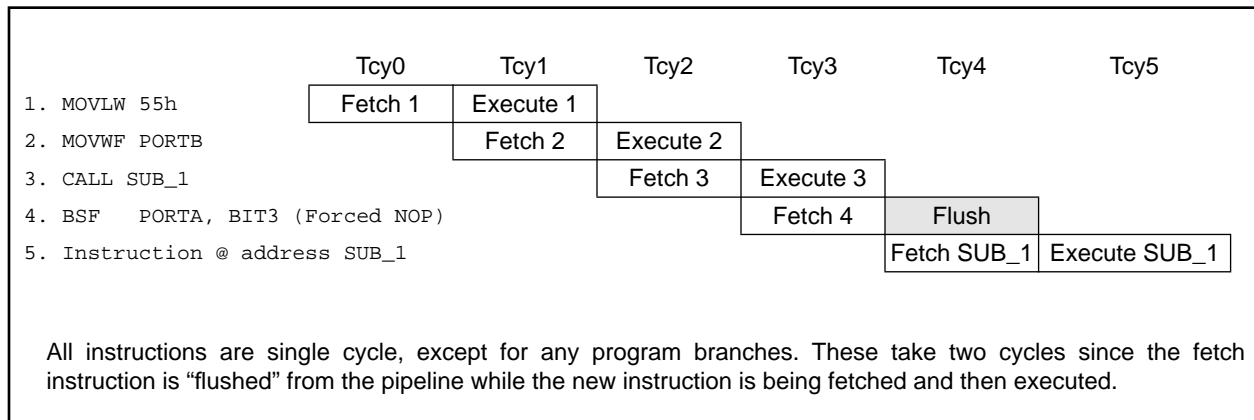
A fetch cycle begins with the program counter (PC) incrementing in Q1.

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



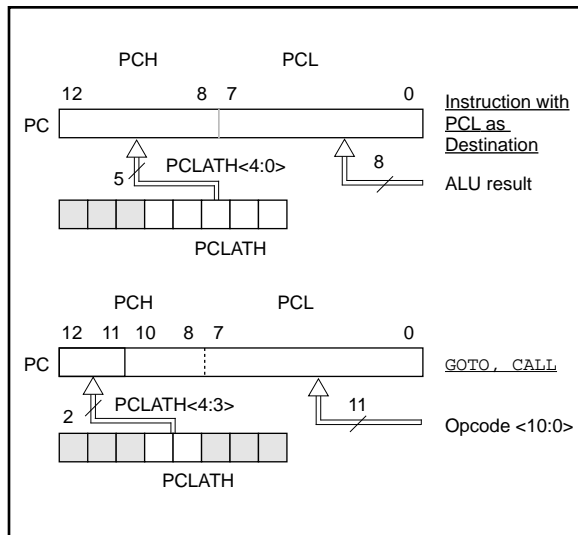
EXAMPLE 3-1: INSTRUCTION PIPELINE FLOW



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.

PIC16C64X & PIC16C66X

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	RBP _U	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111

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

PIC16C64X & PIC16C66X

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)

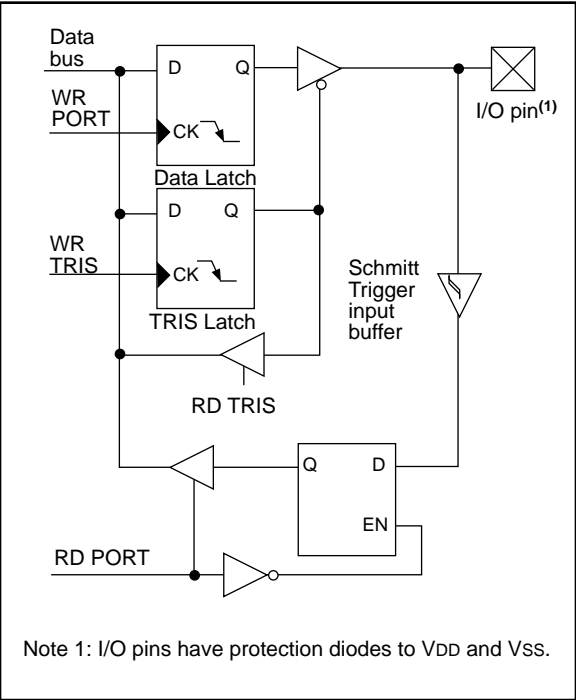


TABLE 5-5: PORTC FUNCTIONS

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.

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

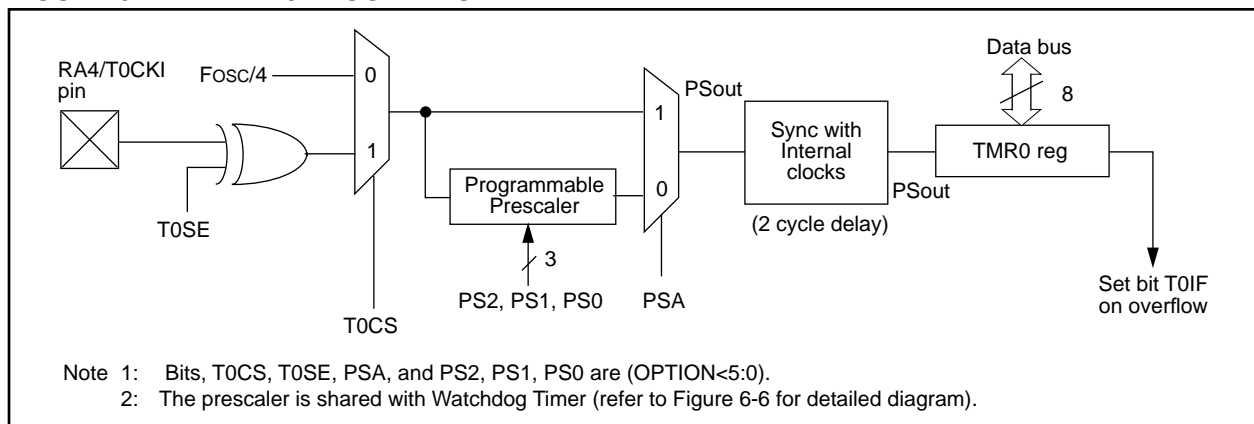
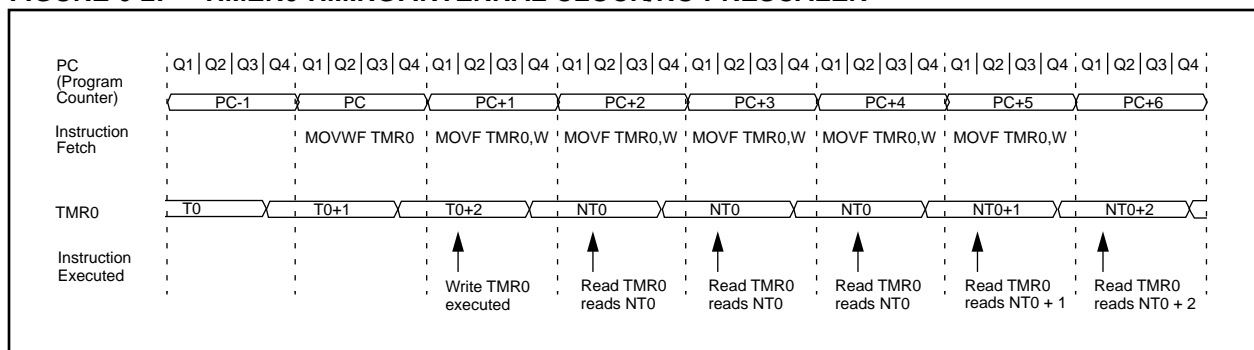


FIGURE 6-2: TIMER0 TIMING: INTERNAL CLOCK/NO PRESCALER



PIC16C64X & PIC16C66X

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	PWRT ^{TE}	WDTE	FOSC1	FOSC0	CONFIG REGISTER:	Address
bit13														2007h	
bit 13-8														bit0	
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:															
PWRT^{TE}: 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 PWRT ^{TE} . Ensure the Power-up Timer is enabled anytime Brown-out Reset is enabled.															
Note 2: All of the CP1:CP0 pairs have to be given the same value to enable the code protection scheme listed.															

PIC16C64X & PIC16C66X

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 → 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.

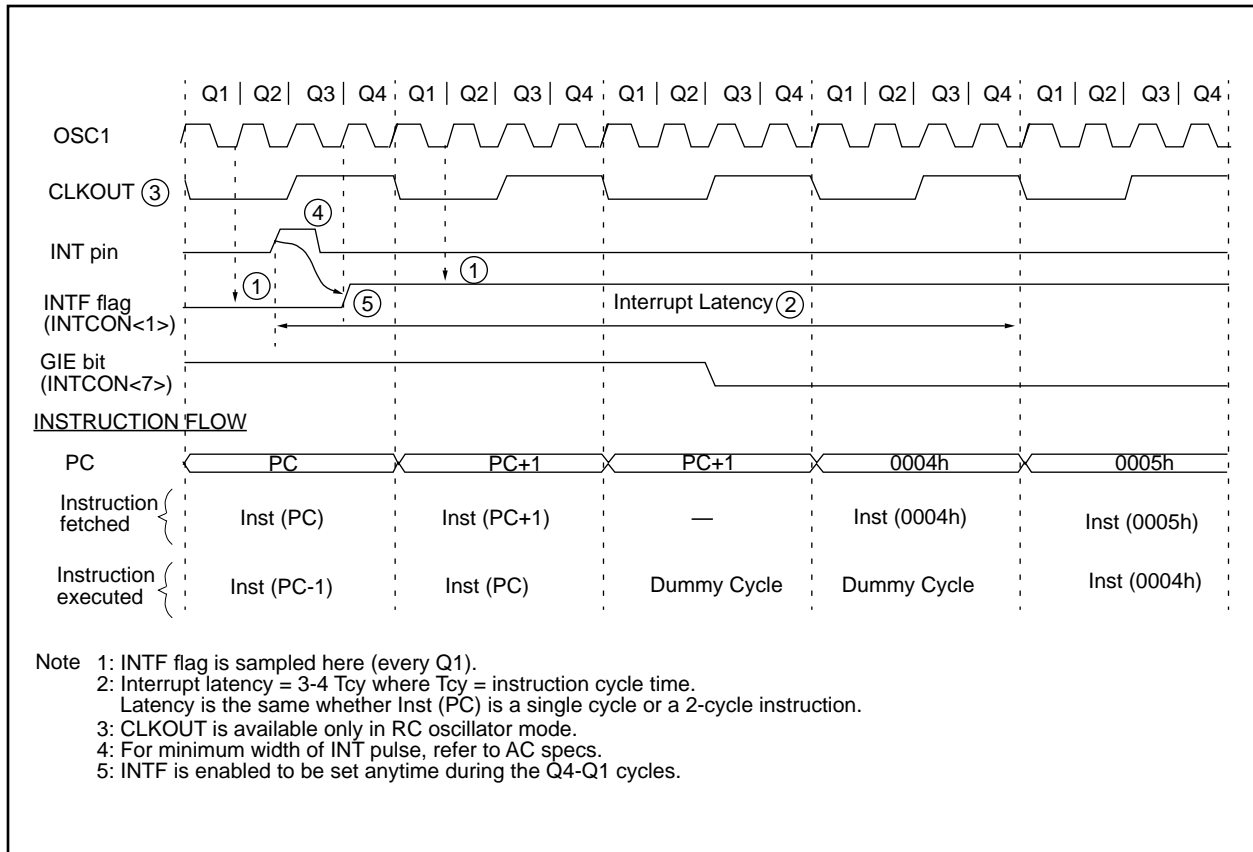
9.5.3 PORTB INTERRUPT

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



PIC16C64X & PIC16C66X

BCF		Bit Clear f							
Syntax:	[<i>label</i>] BCF f,b								
Operands:	$0 \leq f \leq 127$ $0 \leq b \leq 7$								
Operation:	$0 \rightarrow (f)$								
Status Affected:	None								
Encoding:	<table border="1"><tr><td>01</td><td>00bb</td><td>bfff</td><td>ffff</td></tr></table>				01	00bb	bfff	ffff	
01	00bb	bfff	ffff						
Description:	Bit 'b' in register 'f' is cleared.								
Words:	1								
Cycles:	1								
Example	BCF FLAG_REG, 7								
	Before Instruction								
	FLAG_REG = 0xC7								
	After Instruction								
	FLAG_REG = 0x47								

BSF		Bit Set f			
Syntax:	[<i>label</i>] BSF f,b				
Operands:	$0 \leq f \leq 127$ $0 \leq b \leq 7$				
Operation:	$1 \rightarrow (f)$				
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:	[<i>label</i>] BTFSC f,b																			
Operands:	$0 \leq f \leq 127$ $0 \leq b \leq 7$																			
Operation:	skip if (f) = 0																			
Status Affected:	None																			
Encoding:	<table border="1"><tr><td>01</td><td>10bb</td><td>bfff</td><td>ffff</td></tr></table>					01	10bb	bfff	ffff											
01	10bb	bfff	ffff																	
Description:	<p>If bit 'b' in register 'f' is '0' then the next instruction is skipped.</p> <p>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.</p>																			
Words:	1																			
Cycles:	1(2)																			
Example	<table><tr><td>HERE</td><td>BTFSC</td><td>FLAG, 1</td></tr><tr><td>FALSE</td><td>GOTO</td><td>PROCESS_CODE</td></tr><tr><td>TRUE</td><td>.</td><td></td></tr><tr><td></td><td>.</td><td></td></tr><tr><td></td><td>.</td><td></td></tr></table> <p>Before Instruction</p> <p>PC = address HERE</p> <p>After Instruction</p> <p>if FLAG<1> = 0,</p> <p>PC = address TRUE</p> <p>if FLAG<1>=1,</p> <p>PC = address FALSE</p>					HERE	BTFSC	FLAG, 1	FALSE	GOTO	PROCESS_CODE	TRUE	.			.			.	
HERE	BTFSC	FLAG, 1																		
FALSE	GOTO	PROCESS_CODE																		
TRUE	.																			
	.																			
	.																			

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 VDD and VPP supplies which allows it to verify programmed memory at VDD min and VDD max for maximum reliability. It has an LCD display for displaying error messages, keys to enter commands and a modular detachable socket assembly to support various package types. In 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.

PIC16C64X & PIC16C66X

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 (fuzzyTECH-MP)

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

Both versions include Microchip's *fuzzyLAB*™ 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.

PIC16C64X & PIC16C66X

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 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial, $0^{\circ}\text{C} \leq T_A \leq +70^{\circ}\text{C}$ commercial, and $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ automotive							
Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
D001 D001A	VDD	Supply Voltage	4.0 4.5	— —	6.0 5.5	V V	XT, RC and LP osc configuration HS osc configuration
D002*	VDR	RAM Data Retention Voltage ⁽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 3.7	4.0 4.0	4.3 4.4	V V	BODEN configuration bit is clear Automotive
D010 D010A D013	IDD	Supply Current ⁽²⁾	— —	2.7 35	5 70	mA μA	XT and RC osc configuration FOSC = 4 MHz, VDD = 5.5V, WDT disabled ⁽⁴⁾ LP osc configuration, PIC16C64X & PIC16C66X-04 only FOSC = 32 kHz, VDD = 4.0V, WDT disabled HS osc configuration FOSC = 20 MHz, VDD = 5.5V, WDT disabled
D015 D016	ΔI_{BOR} ΔI_{COMP}	Module Differential Current ⁽⁵⁾ Brown-out Reset Current Comparator Current for each Comparator	— —	350 —	425 100	μA μA	BODEN bit is clear, VDD = 5.0V VDD = 4.0V
D017 D021	ΔI_{VREF} ΔI_{WDT}	VREF Current WDT Current	— —	— 6.0	300 20	μA μA	VDD = 4.0V VDD = 4.0V Automotive
D021	IPD	Power-down Current ⁽³⁾	— —	1.5 2.5	21 24	μA μA	VDD = 4.0V, WDT disabled 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,

MCLR = VDD; WDT enabled/disabled as specified.

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

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

PIC16C64X & PIC16C66X

12.5 Timing Diagrams and Specifications

FIGURE 12-2: EXTERNAL CLOCK TIMING

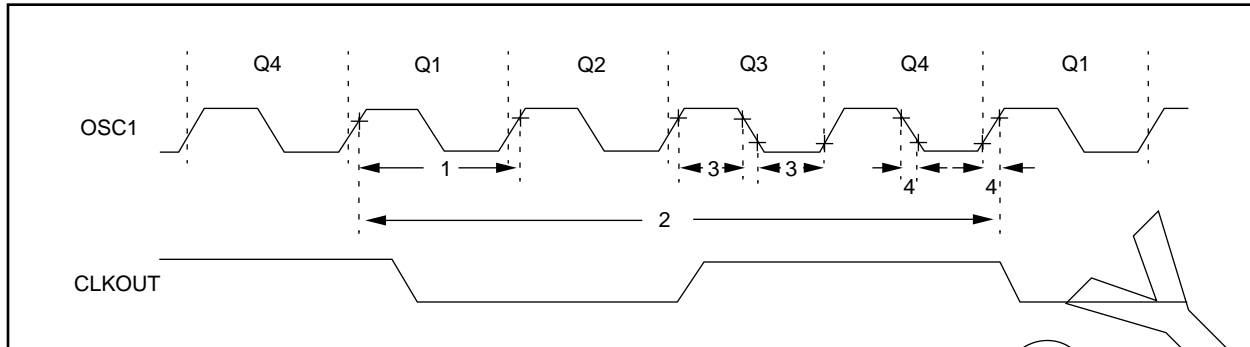


TABLE 12-4: EXTERNAL CLOCK TIMING REQUIREMENTS

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.

PIC16C64X & PIC16C66X

NOTES:

PIC16C64X & PIC16C66X

E.5 PIC16C7X Family of Devices

	Clock			Memory			Peripherals				Features		
	Maximum Frequency of Operation (MHz)	EPROM Program Memory (K x 4 words)	Data Memory (bytes)	Timer Module(s)	Capable/Compare/PWM Module(s)	Serial Ports (SPI/I ² C, 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
PIC16C710	20	512	36	TMR0	—	—	—	4	4	13	2.5-6.0	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	18-pin DIP, SOIC; 20-pin SSOP
PIC16C72	20	2K	128	TMR0, TMR1, TMR2	1 SPI/I ² C	—	—	5	8	22	2.5-6.0	Yes	28-pin SDIP, SOIC, SSOP
PIC16C73	20	4K	192	TMR0, TMR1, TMR2	2 SPI/I ² C, USART	—	—	5	11	22	2.5-6.0	Yes	28-pin SDIP, SOIC
PIC16C73A	20	4K	192	TMR0, TMR1, TMR2	2 SPI/I ² C, USART	—	—	5	11	22	2.5-6.0	Yes	28-pin SDIP, SOIC
PIC16C74	20	4K	192	TMR0, TMR1, TMR2	2 SPI/I ² C, USART	Yes	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/I ² C, USART	Yes	Yes	8	12	33	2.5-6.0	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.

PIC16C64X & PIC16C66X

E.7 PIC16C9XX Family Of Devices

	Clock		Memory		Peripherals								Features			
	8	4K	176	TMRO, TMR1, TMR2	1	SPI/I ² C	—	—	4 Com 32 Seg	8	25	27	3.0-6.0	Yes	—	64-pin SDIP(1), TQFP, 68-pin PLCC, DIE
PIC16C923	8	4K	176	TMRO, TMR1, TMR2	1	SPI/I ² C	—	—	4 Com 32 Seg	8	25	27	3.0-6.0	Yes	—	64-pin SDIP(1), TQFP, 68-pin PLCC, DIE
PIC16C924	8	4K	176	TMRO, TMR1, TMR2	1	SPI/I ² C	—	5	4 Com 32 Seg	9	25	27	3.0-6.0	Yes	—	64-pin SDIP(1), TQFP, 68-pin PLCC, DIE

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.

PIC16C64X & PIC16C66X

PORTE Register	36
Ports	
Parallel Slave Port	39
PORTA	29
PORTB	32
PORTC	34
PORTD	14
PORTE	14
Power Control/Status Register (PCON)	61
Power-down Mode (SLEEP)	70
Power-on Reset (POR)	60
Power-up Timer (PWRT)	60
Prescaler	44
PRO MATE® Universal Programmer	87
Program Memory Organization	17
PSPMODE bit	35, 36

Q

Quick-Turnaround-Production (QTP) Devices	7
---	---

R

RA2 pin	30
RC Oscillator	58
Reset	59
RETFIE Instruction	82
RETLW Instruction	82
RETURN Instruction	83
RLF Instruction	83
RRF Instruction	83

S

Serialized Quick-Turnaround-Production (SQTP)	
Devices	7
SFR	74
SFR As Source/Destination	74
SLEEP Instruction	83
Software Simulator (MPLAB-SIM)	89
Special Features of the CPU	55
Special Function Registers	19, 74
Stack	27
STATUS Register	21
SUBLW Instruction	84
SUBWF Instruction	84
SWAPF Instruction	85
Switching Prescalers	45

T

Timer Modules	
Timer0	
Block Diagram	41
Counter Mode	41
External Clock	43
Interrupt	41
Prescaler	44
Section	41
Timer Mode	41
Timing Diagram	41
TMR0 register	41
Timing Diagrams and Specifications	98
TMR0 Interrupt	67
TRIS Instruction	85

TRISA	29
TRISB	32
TRISC Register	34
TRISD Register	35
TRISE Register	36

V

Voltage Reference Module	53
VRCON Register	53

W

Watchdog Timer (WDT)	69
----------------------------	----

X

XORLW Instruction	85
XORWF Instruction	85

LIST OF EXAMPLES

Example 3-1: Instruction Pipeline Flow	15
Example 4-1: Indirect Addressing	28
Example 5-1: Initializing PORTA	29
Example 5-2: Initializing PORTC	34
Example 5-3: Read-Modify-Write Instructions on an I/O Port	38
Example 6-1: Changing Prescaler (Timer0→WDT)	45
Example 6-2: Changing Prescaler (WDT→Timer0)	45
Example 7-1: Initializing Comparator Module	49
Example 8-1: Voltage Reference Configuration	54
Example 9-1: Saving the STATUS and W Registers in RAM	68

LIST OF FIGURES

Figure 3-1: PIC16C641/642 Block Diagram	10
Figure 3-2: PIC16C661/662 Block Diagram	11
Figure 3-3: Clock/Instruction Cycle	15
Figure 4-1: PIC16C641/661 Program Memory Map and Stack	17
Figure 4-2: PIC16C642/662 Program Memory Map and Stack	17
Figure 4-3: PIC16C641/661 Data Memory Map	18
Figure 4-4: PIC16C642/662 Data Memory Map	19
Figure 4-5: STATUS Register (Address 03h, 83h)	21
Figure 4-6: OPTION Register (address 81h)	22
Figure 4-7: INTCON Register (address 0Bh, 8Bh)	23
Figure 4-8: PIE1 Register (address 8Ch)	24
Figure 4-9: PIR1 Register (address 0Ch)	25
Figure 4-10: PCON Register (Address 8Eh)	26
Figure 4-11: Loading Of PC In Different Situations	27
Figure 4-12: Direct/indirect Addressing	28
Figure 5-1: Block Diagram of RA1:RA0 Pins	29
Figure 5-2: Block Diagram of RA2 Pin	30
Figure 5-3: Block Diagram of RA3 Pin	30
Figure 5-4: Block Diagram of RA4 Pin	31
Figure 5-5: Block Diagram of RB7:RB4 Pins	32
Figure 5-6: Block Diagram of RB3:RB0 Pins	32
Figure 5-7: PORTC Block Diagram (in I/O port Mode)	34
Figure 5-8: PORTD Block Diagram (in I/O Port Mode)	35
Figure 5-9: TRISE Register (Address 89h)	36
Figure 5-10: PORTE Block Diagram (in I/O Port Mode)	37
Figure 5-11: Successive I/O Operation	38
Figure 5-12: PORTD and PORTE as a Parallel Slave Port	39
Figure 6-1: Timer0 Block Diagram	41
Figure 6-2: Timer0 Timing: Internal Clock/No Prescaler	41
Figure 6-3: Timer0 Timing: Internal Clock/Prescale 1:2... ..	42

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.

Information contained in this publication regarding device applications and the like is intended through suggestion only and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. No representation or warranty is given and no liability is assumed by Microchip Technology Incorporated with respect to the accuracy or use of such information, or infringement of patents or other intellectual property rights arising from such use or otherwise. Use of Microchip's products as critical components in life support systems is not authorized except with express written approval by Microchip. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights.

Trademarks


The Microchip name and logo, the Microchip logo, FilterLab, KEELOQ, microID, MPLAB, PIC, PICmicro, PICMASTER, PICSTART, PRO MATE, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

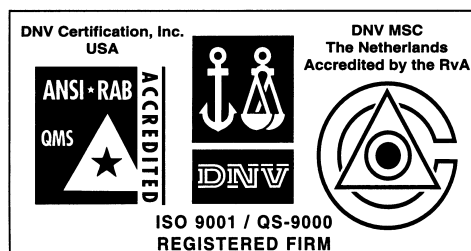
dsPIC, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, microPort, Migratable Memory, MPASM, MPLIB, MPLINK, MPSIM, MXDEV, PICC, PICDEM, PICDEM.net, rPIC, Select Mode and Total Endurance are trademarks of Microchip Technology Incorporated in the U.S.A.

Serialized Quick Turn Programming (SQTP) is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.

© 2002, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

 Printed on recycled paper.



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.