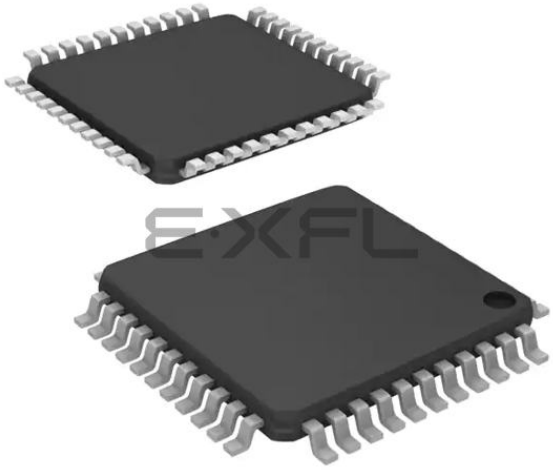


Welcome to E-XFL.COM

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	Obsolete
Core Processor	PIC
Core Size	8-Bit
Speed	4MHz
Connectivity	-
Peripherals	Brown-out Detect/Reset, LED, POR, WDT
Number of I/O	33
Program Memory Size	7KB (4K x 14)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	176 x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 6V
Data Converters	-
Oscillator Type	External
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	44-TQFP
Supplier Device Package	44-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lc662t-04-pt

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.

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.

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.

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

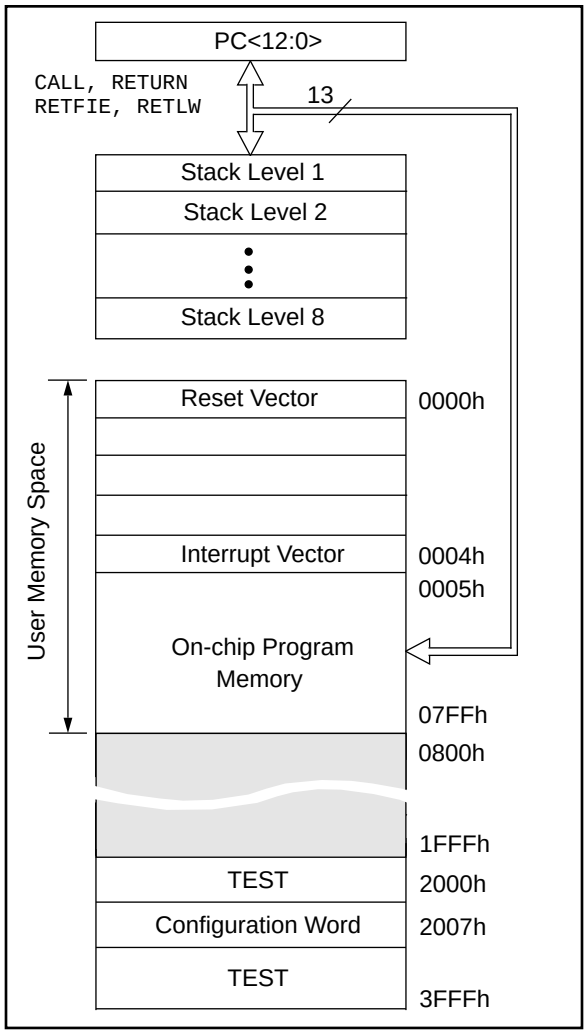
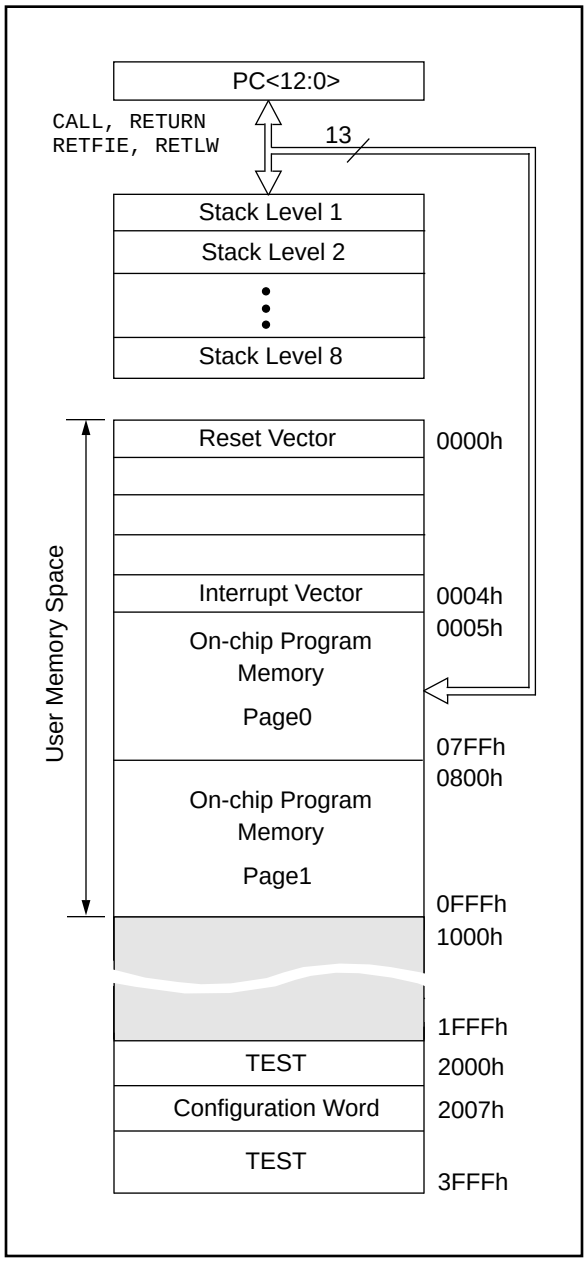


FIGURE 4-2: PIC16C642/662 PROGRAM MEMORY MAP AND STACK



4.2.2.2 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

4.2.2.6 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 (BOR), and Parity Error Reset (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)

4.5 Indirect Addressing, INDF, and FSR Registers

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

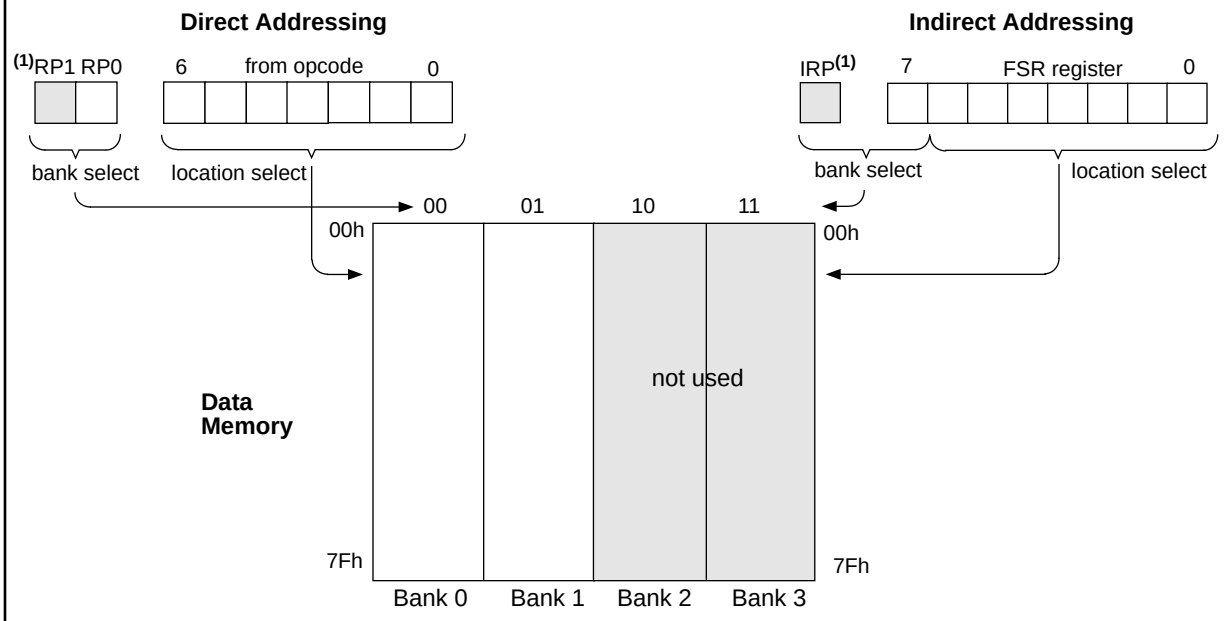
Indirect addressing is possible by using the INDF register. Any instruction using the INDF register actually accesses data pointed to by the file select register (FSR). Reading INDF itself indirectly will produce 00h. Writing to the INDF register indirectly results in a no-operation (although status 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.

5.2 PORTB and TRISB Registers

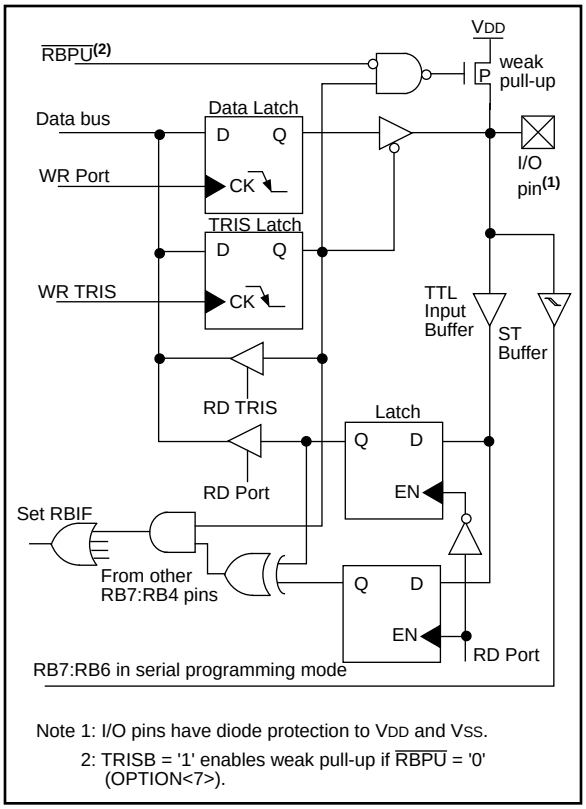
PORTB is an 8-bit wide bi-directional port. The corresponding data direction register is TRISB. Setting a bit in the TRISB register puts the corresponding output driver in a hi-impedance mode. Clearing a bit in the TRISB register puts the contents of the output latch on the selected pin(s).

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:

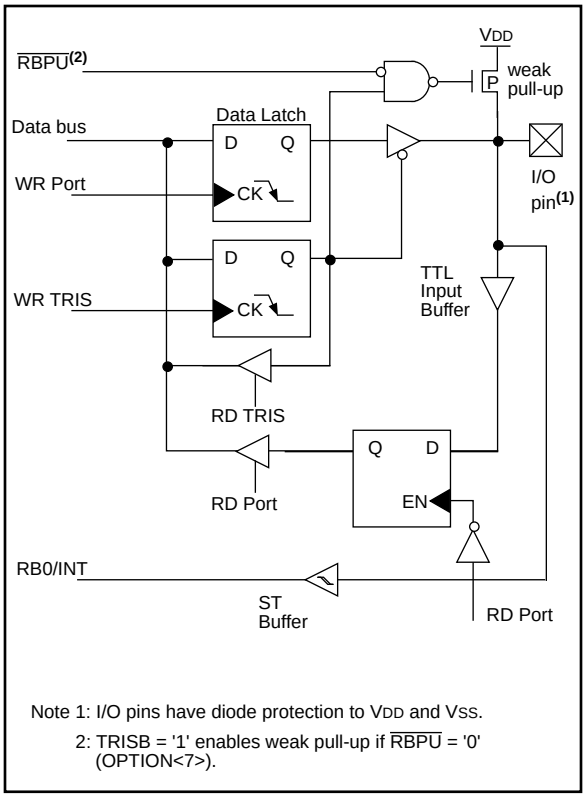
a) Any read or write of PORTB. This will end the mismatch condition.
b) Clear flag bit RBIF.

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>).

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

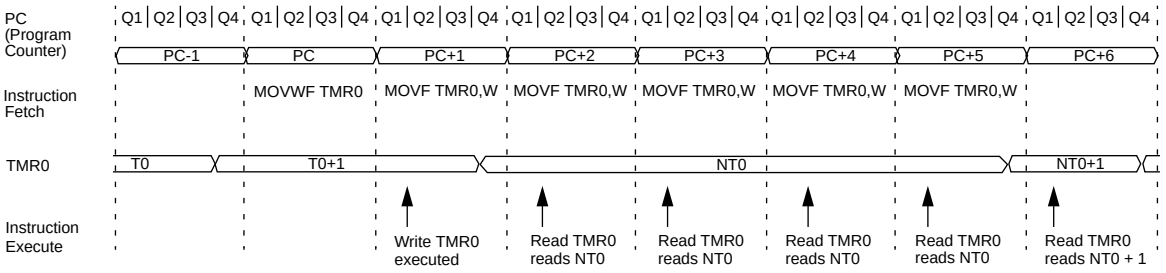
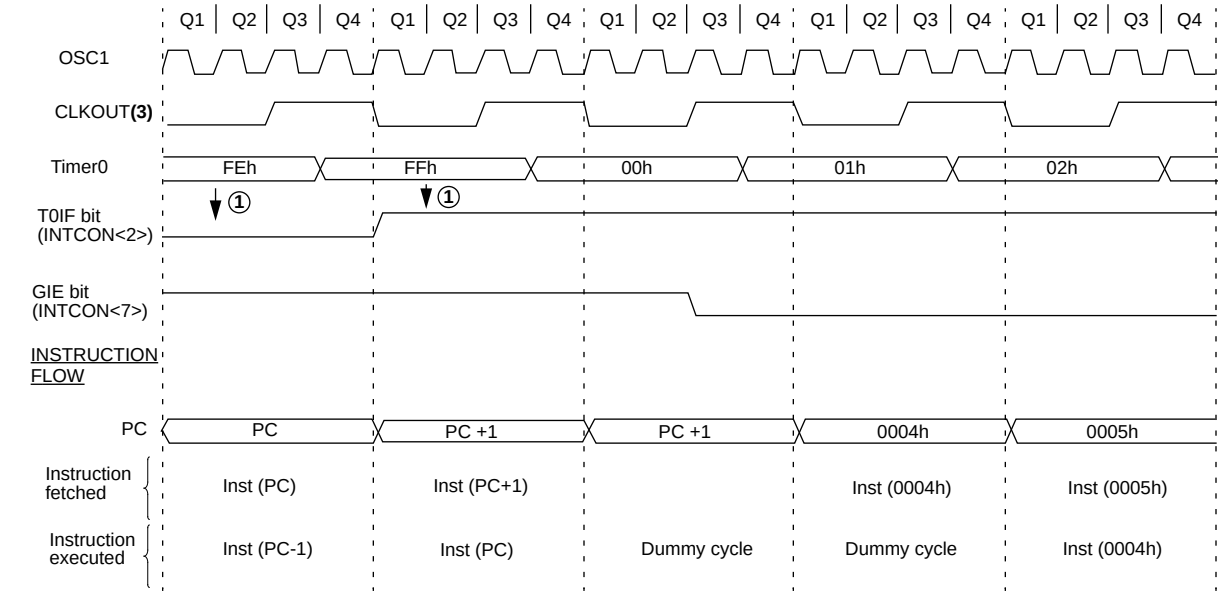


FIGURE 6-4: TIMER0 INTERRUPT TIMING



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.

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.

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.

9.6 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
```


NOTES:

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

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

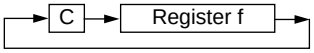
Operation: See description below

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

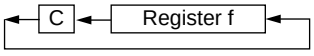
Operation: See description below

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

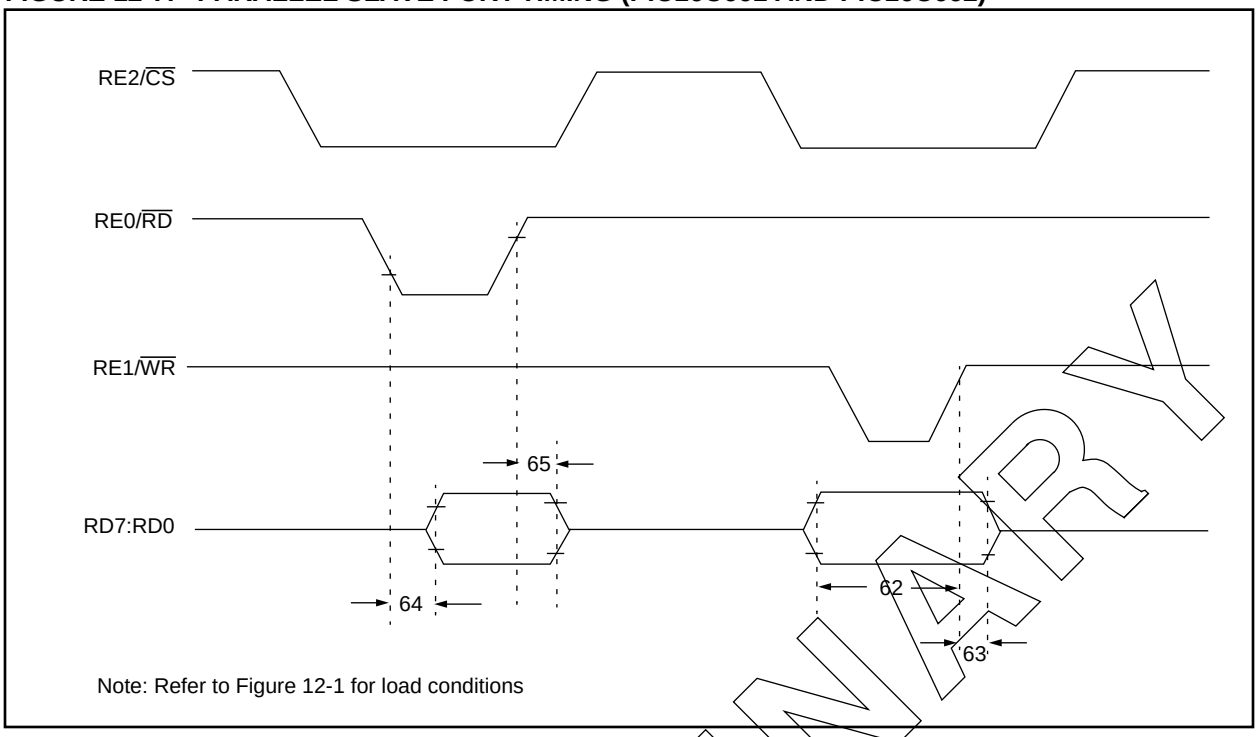
Description: The power-down status bit, $\overline{PD}$ is cleared. Time-out status bit, $\overline{TO}$ is set. Watchdog Timer and its prescaler are cleared.
The processor is put into SLEEP mode with the oscillator stopped. See Power-Down Mode (SLEEP) for more details.

Words: 1

Cycles: 1

Example: SLEEP

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



Note: Refer to Figure 12-1 for load conditions

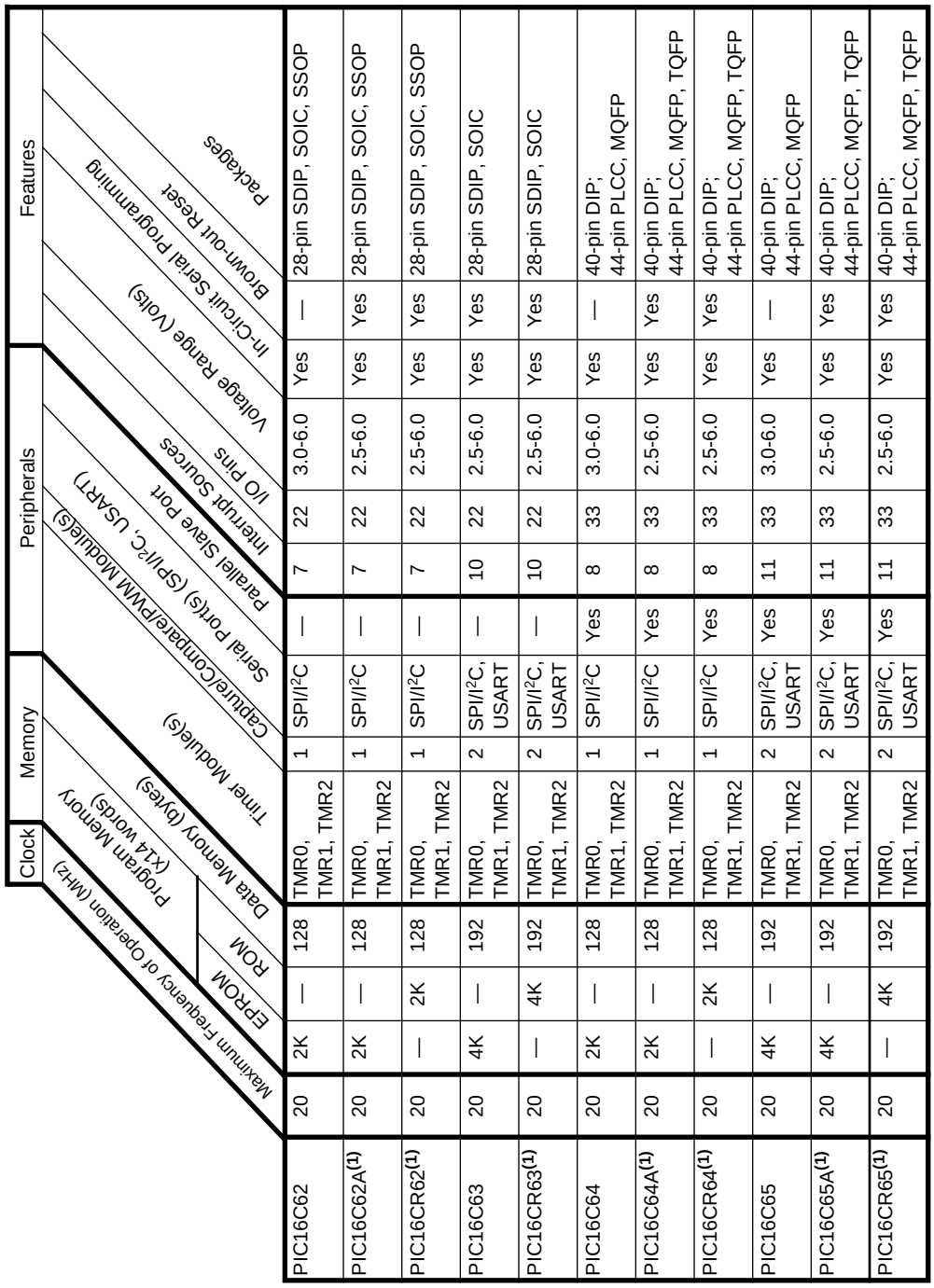
TABLE 12-8: PARALLEL SLAVE PORT REQUIREMENTS (PIC16C661 AND PIC16C662)

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.

NOTES:

E.4 PIC16C6X Family of Devices

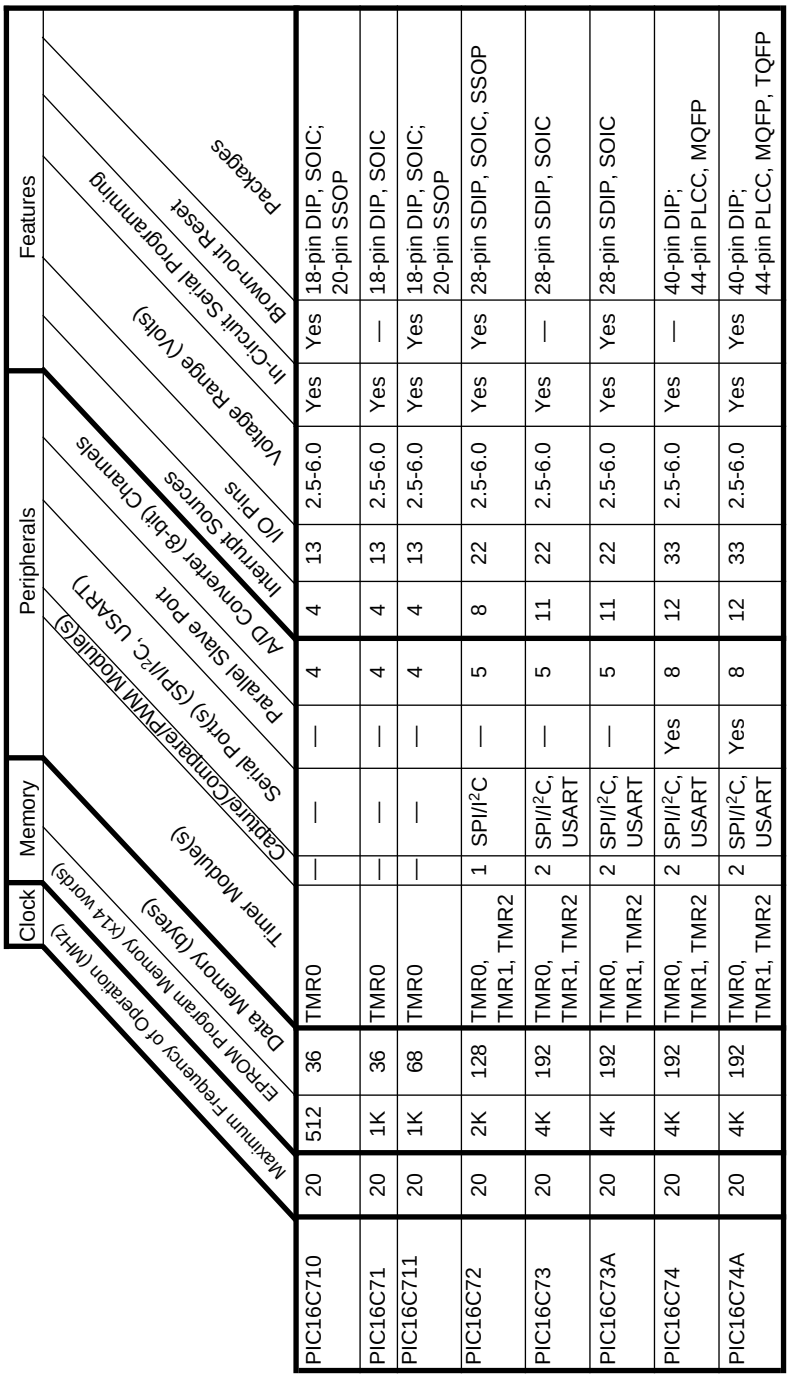


	Clock	Memory				Peripherals					Features			
	Maximum Frequency of Operation (MHz)	EPROM	ROM	Data Memory (bytes)	Timer Module(s)	Capture/Compare/PWM Module(s)	Serial Port(s) (SPI/I²C, USART)	Parallel Slave Port	Interrupt Sources	I/O Pins	Voltage Range (Volts)	In-Circuit Serial Programming	Brown-out Reset	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.

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.