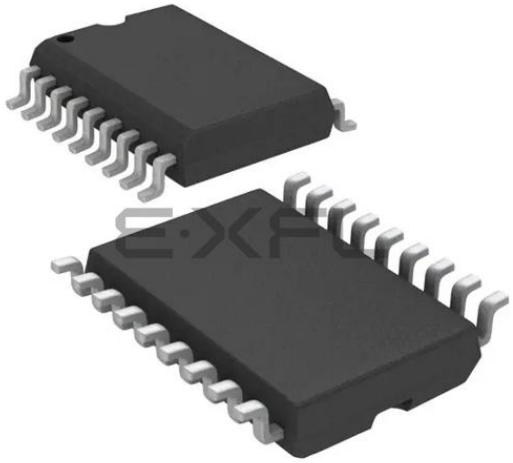


**Welcome to E-XFL.COM**

## 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 | UART/USART |
| Peripherals | Brown-out Detect/Reset, POR, PWM, WDT |
| Number of I/O | 16 |
| Program Memory Size | 7KB (4K x 14) |
| Program Memory Type | FLASH |
| EEPROM Size | 256 x 8 |
| RAM Size | 256 x 8 |
| Voltage - Supply (Vcc/Vdd) | 3V ~ 5.5V |
| Data Converters | - |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 18-SOIC (0.295", 7.50mm Width) |
| Supplier Device Package | 18-SOIC |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic16f648a-i-so |

## 1.0 GENERAL DESCRIPTION

The PIC16F627A/628A/648A are 18-pin Flash-based members of the versatile PIC16F627A/628A/648A family of low-cost, high-performance, CMOS, fully-static, 8-bit microcontrollers.

All PIC® microcontrollers employ an advanced RISC architecture. The PIC16F627A/628A/648A have enhanced core features, an 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, complemented by a large register set.

PIC16F627A/628A/648A microcontrollers typically achieve a 2:1 code compression and a 4:1 speed improvement over other 8-bit microcontrollers in their class.

PIC16F627A/628A/648A devices have integrated features to reduce external components, thus reducing system cost, enhancing system reliability and reducing power consumption.

The PIC16F627A/628A/648A has 8 oscillator configurations. The single-pin RC oscillator provides a low-cost solution. The LP oscillator minimizes power consumption, XT is a standard crystal, and INTOSC is a self-contained precision two-speed internal oscillator. The HS mode is for High-Speed crystals. The EC mode is for an external clock source.

The Sleep (Power-down) mode offers power savings. Users can wake-up the chip from Sleep through several external interrupts, internal interrupts and Resets.

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

Table 1-1 shows the features of the PIC16F627A/628A/648A mid-range microcontroller family.

A simplified block diagram of the PIC16F627A/628A/648A is shown in Figure 3-1.

The PIC16F627A/628A/648A series fits in applications ranging from battery chargers to low power remote sensors. The Flash technology makes customizing application programs (detection levels, pulse generation, timers, etc.) extremely fast and convenient. The small footprint packages makes this microcontroller series ideal for all applications with space limitations. Low cost, low power, high performance, ease of use and I/O flexibility make the PIC16F627A/628A/648A very versatile.

### 1.1 Development Support

The PIC16F627A/628A/648A family is supported by a full-featured macro assembler, a software simulator, an in-circuit emulator, a low cost in-circuit debugger, a low cost development programmer and a full-featured programmer. A Third Party “C” compiler support tool is also available.

**TABLE 1-1: PIC16F627A/628A/648A FAMILY OF DEVICES**

| | | PIC16F627A | PIC16F628A | PIC16F648A | PIC16LF627A | PIC16LF628A | PIC16LF648A |
|---|---|---|---|---|---|---|---|
| Clock | Maximum Frequency of Operation (MHz) | 20 | 20 | 20 | 20 | 20 | 20 |
| Memory | Flash Program Memory (words) | 1024 | 2048 | 4096 | 1024 | 2048 | 4096 |
| | RAM Data Memory (bytes) | 224 | 224 | 256 | 224 | 224 | 256 |
| | EEPROM Data Memory (bytes) | 128 | 128 | 256 | 128 | 128 | 256 |
| Peripherals | Timer module(s) | TMR0, TMR1, TMR2 | TMR0, TMR1, TMR2 | TMR0, TMR1, TMR2 | TMR0, TMR1, TMR2 | TMR0, TMR1, TMR2 | TMR0, TMR1, TMR2 |
| | Comparator(s) | 2 | 2 | 2 | 2 | 2 | 2 |
| | Capture/Compare/ PWM modules | 1 | 1 | 1 | 1 | 1 | 1 |
| | Serial Communications | USART | USART | USART | USART | USART | USART |
| | Internal Voltage Reference | Yes | Yes | Yes | Yes | Yes | Yes |
| Features | Interrupt Sources | 10 | 10 | 10 | 10 | 10 | 10 |
| | I/O Pins | 16 | 16 | 16 | 16 | 16 | 16 |
| | Voltage Range (Volts) | 3.0-5.5 | 3.0-5.5 | 3.0-5.5 | 2.0-5.5 | 2.0-5.5 | 2.0-5.5 |
| | Brown-out Reset | Yes | Yes | Yes | Yes | Yes | Yes |
| | Packages | 18-pin DIP, SOIC, 20-pin SSOP, 28-pin QFN | 18-pin DIP, SOIC, 20-pin SSOP, 28-pin QFN | 18-pin DIP, SOIC, 20-pin SSOP, 28-pin QFN | 18-pin DIP, SOIC, 20-pin SSOP, 28-pin QFN | 18-pin DIP, SOIC, 20-pin SSOP, 28-pin QFN | 18-pin DIP, SOIC, 20-pin SSOP, 28-pin QFN |

All PIC® family devices have Power-on Reset, selectable Watchdog Timer, selectable code-protect and high I/O current capability.
All PIC16F627A/628A/648A family devices use serial programming with clock pin RB6 and data pin RB7.

## 3.0 ARCHITECTURAL OVERVIEW

The high performance of the PIC16F627A/628A/648A family can be attributed to a number of architectural features commonly found in RISC microprocessors. To begin with, the PIC16F627A/628A/648A uses a Harvard architecture in which program and data are accessed from separate memories using separate busses. 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 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.

Table 3-1 lists device memory sizes (Flash, Data and EEPROM).

**TABLE 3-1: DEVICE MEMORY LIST**

| Device | Memory | | |
|---|---|---|---|
| | Flash Program | RAM Data | EEPROM Data |
| PIC16F627A | 1024 x 14 | 224 x 8 | 128 x 8 |
| PIC16F628A | 2048 x 14 | 224 x 8 | 128 x 8 |
| PIC16F648A | 4096 x 14 | 256 x 8 | 256 x 8 |
| PIC16LF627A | 1024 x 14 | 224 x 8 | 128 x 8 |
| PIC16LF628A | 2048 x 14 | 224 x 8 | 128 x 8 |
| PIC16LF648A | 4096 x 14 | 256 x 8 | 256 x 8 |

The PIC16F627A/628A/648A can directly or indirectly address its register files or data memory. All Special Function Registers (SFR), including the program counter, are mapped in the data memory. The PIC16F627A/628A/648A 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' makes programming with the PIC16F627A/628A/648A simple yet efficient. In addition, the learning curve is reduced significantly.

The PIC16F627A/628A/648A 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 Borrow and Digit Borrow out bits, respectively, in subtraction. See the `SUBLW` and `SUBWF` instructions for examples.

A simplified block diagram is shown in Figure 3-1, and a description of the device pins in Table 3-2.

Two types of data memory are provided on the PIC16F627A/628A/648A devices. Nonvolatile EEPROM data memory is provided for long term storage of data, such as calibration values, look-up table data, and any other data which may require periodic updating in the field. These data types are not lost when power is removed. The other data memory provided is regular RAM data memory. Regular RAM data memory is provided for temporary storage of data during normal operation. Data is lost when power is removed.

#### 4.2.2.5 PIR1 Register

This register contains interrupt flag bits.

| | |
|---|---|
| **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. |

**REGISTER 4-5: PIR1 – PERIPHERAL INTERRUPT REGISTER 1 (ADDRESS: 0Ch)**

| R/W-0 | R/W-0 | R-0 | R-0 | U-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| EEIF | CMIF | RCIF | TXIF | — | CCP1IF | TMR2IF | TMR1IF |
| bit 7 | | | | | | | bit 0 |

bit 7 **EEIF:** EEPROM Write Operation Interrupt Flag bit
1 = The write operation completed (must be cleared in software)
0 = The write operation has not completed or has not been started

bit 6 **CMIF**: Comparator Interrupt Flag bit
1 = Comparator output has changed
0 = Comparator output has not changed

bit 5 **RCIF**: USART Receive Interrupt Flag bit
1 = The USART receive buffer is full
0 = The USART receive buffer is empty

bit 4 **TXIF**: USART Transmit Interrupt Flag bit
1 = The USART transmit buffer is empty
0 = The USART transmit buffer is full

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

bit 2 **CCP1IF**: CCP1 Interrupt Flag bit
Capture Mode
1 = A TMR1 register capture occurred (must be cleared in software)
0 = No TMR1 register capture occurred
Compare Mode
1 = A TMR1 register compare match occurred (must be cleared in software)
0 = No TMR1 register compare match occurred
PWM Mode
Unused in this mode

bit 1 **TMR2IF**: TMR2 to PR2 Match Interrupt Flag bit
1 = TMR2 to PR2 match occurred (must be cleared in software)
0 = No TMR2 to PR2 match occurred

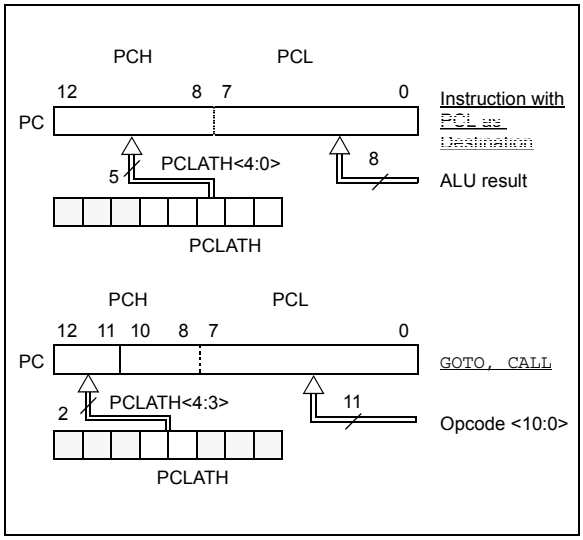
bit 0 **TMR1IF**: TMR1 Overflow Interrupt Flag bit
1 = TMR1 register overflowed (must be cleared in software)
0 = TMR1 register did not overflow

| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| -n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

## 4.3 PCL and PCLATH

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

**FIGURE 4-4: 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 AN556 "*Implementing a Table Read*" (DS00556).

### 4.3.2 STACK

The PIC16F627A/628A/648A family has an 8-level deep x 13-bit wide hardware stack (Figure 4-1). 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.

**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 Indirect Addressing, INDF and FSR Registers

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

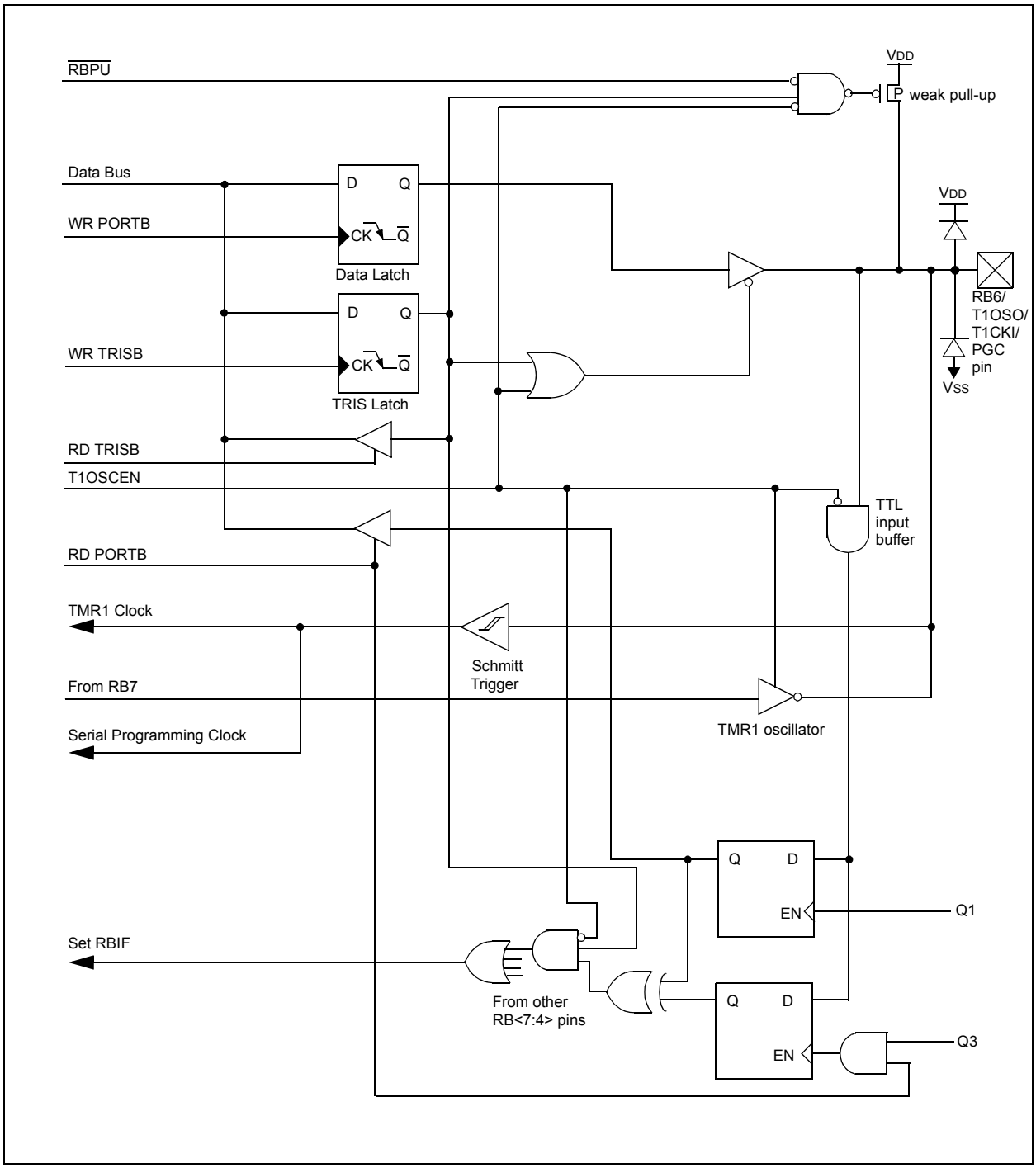
Indirect addressing is possible by using the INDF register. Any instruction using the INDF register actually accesses data pointed to by the File Select Register (FSR). Reading INDF itself indirectly will produce 00h. Writing to the INDF register indirectly results in a no-operation (although Status 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 4-5.

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 clear next
                        ;yes continue
```

**FIGURE 5-14: BLOCK DIAGRAM OF RB6/T1OSO/T1CKI/PGC PIN**



**REGISTER 8-1: T2CON – TIMER2 CONTROL REGISTER (ADDRESS: 12h)**

| U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | TOUTPS3 | TOUTPS2 | TOUTPS1 | TOUTPS0 | TMR2ON | T2CKPS1 | T2CKPS0 |
| bit 7 | | | | | | | bit 0 |

bit 7 **Unimplemented**: Read as '0'

bit 6-3 **TOUTPS<3:0>**: Timer2 Output Postscale Select bits
0000 = 1:1 Postscale Value
0001 = 1:2 Postscale Value
•
•
•
1111 = 1:16 Postscale

bit 2 **TMR2ON**: Timer2 On bit
1 = Timer2 is on
0 = Timer2 is off

bit 1-0 **T2CKPS<1:0>**: Timer2 Clock Prescale Select bits
00 = 1:1 Prescaler Value
01 = 1:4 Prescaler Value
1x = 1:16 Prescaler Value

| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| -n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

**TABLE 8-1: REGISTERS ASSOCIATED WITH TIMER2 AS A TIMER/COUNTER**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0Bh, 8Bh, 10Bh, 18Bh | INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | EEIF | CMIF | RCIF | TXIF | — | CCP1IF | TMR2IF | TMR1IF | 0000 -000 | 0000 -000 |
| 8Ch | PIE1 | EEIE | CMIE | RCIE | TXIE | — | CCP1IE | TMR2IE | TMR1IE | 0000 -000 | 0000 -000 |
| 11h | TMR2 | Timer2 Module's Register | | | | | | | | 0000 0000 | 0000 0000 |
| 12h | T2CON | — | TOUTPS3 | TOUTPS2 | TOUTPS1 | TOUTPS0 | TMR2ON | T2CKPS1 | T2CKPS0 | -000 0000 | -000 0000 |
| 92h | PR2 | Timer2 Period Register | | | | | | | | 1111 1111 | 1111 1111 |

**Legend:** x = unknown, u = unchanged, - = unimplemented read as '0'. Shaded cells are not used by the Timer2 module.

### 9.2.1 CCP PIN CONFIGURATION

The user must configure the RB3/CCP1 pin as an output by clearing the TRISB<3> bit.

| | |
|---|---|
| **Note:** | Clearing the CCP1CON register will force the RB3/CCP1 compare output latch to the default low level. This is not the data latch. |

### 9.2.2 TIMER1 MODE SELECTION

Timer1 must be running in Timer mode or Synchronized Counter mode if the CCP module is using the compare feature. In Asynchronous Counter mode, the compare operation may not work.

### 9.2.3 SOFTWARE INTERRUPT MODE

When generate software interrupt is chosen the CCP1 pin is not affected. Only a CCP interrupt is generated (if enabled).

### 9.2.4 SPECIAL EVENT TRIGGER

In this mode (CCP1M<3:0>=1011), an internal hardware trigger is generated, which may be used to initiate an action. See Register 9-1.

The special event trigger output of the CCP occurs immediately upon a match between the TMR1H, TMR1L register pair and CCPR1H, CCPR1L register pair. The TMR1H, TMR1L register pair is not reset until the next rising edge of the TMR1 clock. This allows the CCPR1 register pair to effectively be a 16-bit programmable period register for Timer1. The special event trigger output also starts an A/D conversion provided that the A/D module is enabled.

| | |
|---|---|
| **Note:** | Removing the match condition by changing the contents of the CCPR1H, CCPR1L register pair between the clock edge that generates the special event trigger and the clock edge that generates the TMR1 Reset will preclude the Reset from occuring. |

**TABLE 9-2: REGISTERS ASSOCIATED WITH CAPTURE, COMPARE, AND TIMER1**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0Bh, 8Bh, 10Bh, 18Bh | INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | EEIF | CMIF | RCIF | TXIF | — | CCP1IF | TMR2IF | TMR1IF | 0000 -000 | 0000 -000 |
| 8Ch | PIE1 | EEIE | CMIE | RCIE | TXIE | — | CCP1IE | TMR2IE | TMR1IE | 0000 -000 | 0000 -000 |
| 86h, 186h | TRISB | PORTB Data Direction Register | | | | | | | | 1111 1111 | 1111 1111 |
| 0Eh | TMR1L | Holding Register for the Least Significant Byte of the 16-bit TMR1 Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 0Fh | TMR1H | Holding Register for the Most Significant Byte of the 16-bit TMR1 Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 10h | T1CON | — | — | T1CKPS1 | T1CKPS0 | T1OSCEN | $\overline{\text{T1SYNC}}$ | TMR1CS | TMR1ON | --00 0000 | --uu uuuu |
| 15h | CCPR1L | Capture/Compare/PWM Register1 (LSB) | | | | | | | | xxxx xxxx | uuuu uuuu |
| 16h | CCPR1H | Capture/Compare/PWM Register1 (MSB) | | | | | | | | xxxx xxxx | uuuu uuuu |
| 17h | CCP1CON | — | — | CCP1X | CCP1Y | CCP1M3 | CCP1M2 | CCP1M1 | CCP1M0 | --00 0000 | --00 0000 |

**Legend:** x = unknown, u = unchanged, - = unimplemented read as '0'. Shaded cells are not used by Capture and Timer1.

**NOTES:**

## 10.0 COMPARATOR MODULE

The comparator module contains two analog comparators. The inputs to the comparators are multiplexed with the RA0 through RA3 pins. The on-chip Voltage Reference (**Section 11.0 "Voltage Reference Module"**) can also be an input to the comparators.

The CMCON register, shown in Register 10-1, controls the comparator input and output multiplexers. A block diagram of the comparator is shown in Figure 10-1.

**REGISTER 10-1: CMCON – COMPARATOR CONFIGURATION REGISTER (ADDRESS: 01Fh)**

| R-0 | R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| C2OUT | C1OUT | C2INV | C1INV | CIS | CM2 | CM1 | CM0 |
| bit 7 | | | | | | | bit 0 |

bit 7 **C2OUT**: Comparator 2 Output bit

When C2INV = 0:
1 = C2 VIN+ > C2 VIN-
0 = C2 VIN+ < C2 VIN-

When C2INV = 1:
1 = C2 VIN+ < C2 VIN-
0 = C2 VIN+ > C2 VIN-

bit 6 **C1OUT**: Comparator 1 Output bit

When C1INV = 0:
1 = C1 VIN+ > C1 VIN-
0 = C1 VIN+ < C1 VIN-

When C1INV = 1:
1 = C1 VIN+ < C1 VIN-
0 = C1 VIN+ > C1 VIN-

bit 5 **C2INV**: Comparator 2 Output Inversion bit
1 = C2 Output inverted
0 = C2 Output not inverted

bit 4 **C1INV**: Comparator 1 Output Inversion bit
1 = C1 Output inverted
0 = C1 Output not inverted

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

When CM<2:0>: = 001
Then:
1 = C1 VIN- connects to RA3
0 = C1 VIN- connects to RA0

When CM<2:0> = 010
Then:
1 = C1 VIN- connects to RA3
C2 VIN- connects to RA2
0 = C1 VIN- connects to RA0
C2 VIN- connects to RA1

bit 2-0 **CM<2:0>**: Comparator Mode bits
Figure 10-1 shows the comparator modes and CM<2:0> bit settings

| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| -n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

### 14.2.3 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 14-2 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° phase shift that a parallel oscillator requires. The 4.7 kΩ resistor provides the negative feedback for stability. The 10 kΩ potentiometers bias the 74AS04 in the linear region. This could be used for external oscillator designs.

**FIGURE 14-2: EXTERNAL PARALLEL RESONANT CRYSTAL OSCILLATOR CIRCUIT**

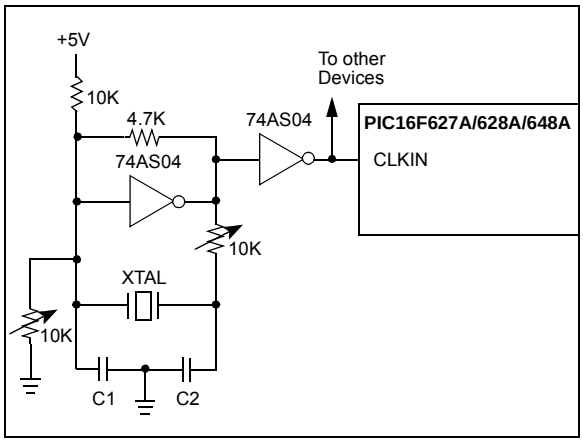
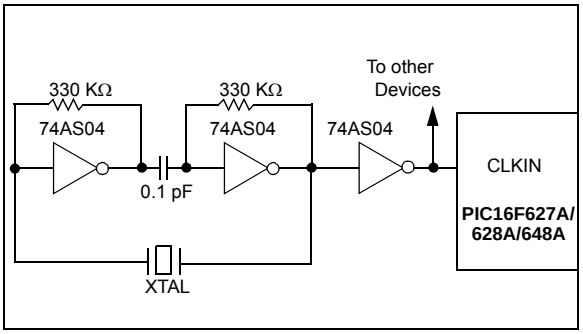



Figure 14-3 shows a series resonant oscillator circuit. This circuit is also designed to use the fundamental frequency of the crystal. The inverter performs a 180° phase shift in a series resonant oscillator circuit. The 330 kΩ resistors provide the negative feedback to bias the inverters in their linear region.

**FIGURE 14-3: EXTERNAL SERIES RESONANT CRYSTAL OSCILLATOR CIRCUIT**




### 14.2.4 PRECISION INTERNAL 4 MHz OSCILLATOR

The internal precision oscillator provides a fixed 4 MHz (nominal) system clock at $V_{DD}$ = 5V and 25°C. See **Section 17.0 "Electrical Specifications"**, for information on variation over voltage and temperature.

### 14.2.5 EXTERNAL CLOCK IN

For applications where a clock is already available elsewhere, users may directly drive the PIC16F627A/628A/648A provided that this external clock source meets the AC/DC timing requirements listed in **Section 17.6 "Timing Diagrams and Specifications"**. Figure 14-4 below shows how an external clock circuit should be configured.

**FIGURE 14-4: EXTERNAL CLOCK INPUT OPERATION (EC, HS, XT OR LP OSC CONFIGURATION)**

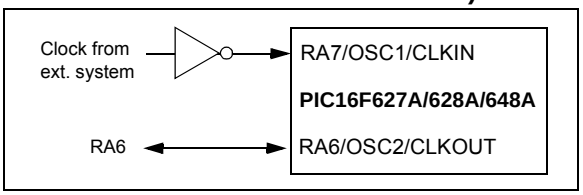


**FIGURE 14-8: TIME OUT SEQUENCE ON POWER-UP (MCLR NOT TIED TO VDD): CASE**

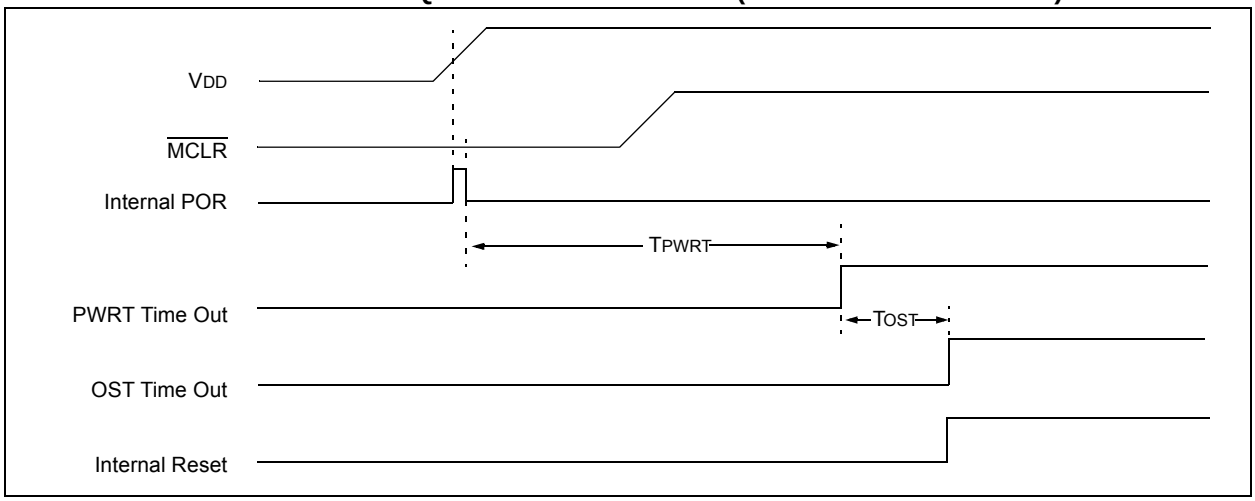



**FIGURE 14-9: TIME OUT SEQUENCE ON POWER-UP (MCLR NOT TIED TO VDD): CASE 2**

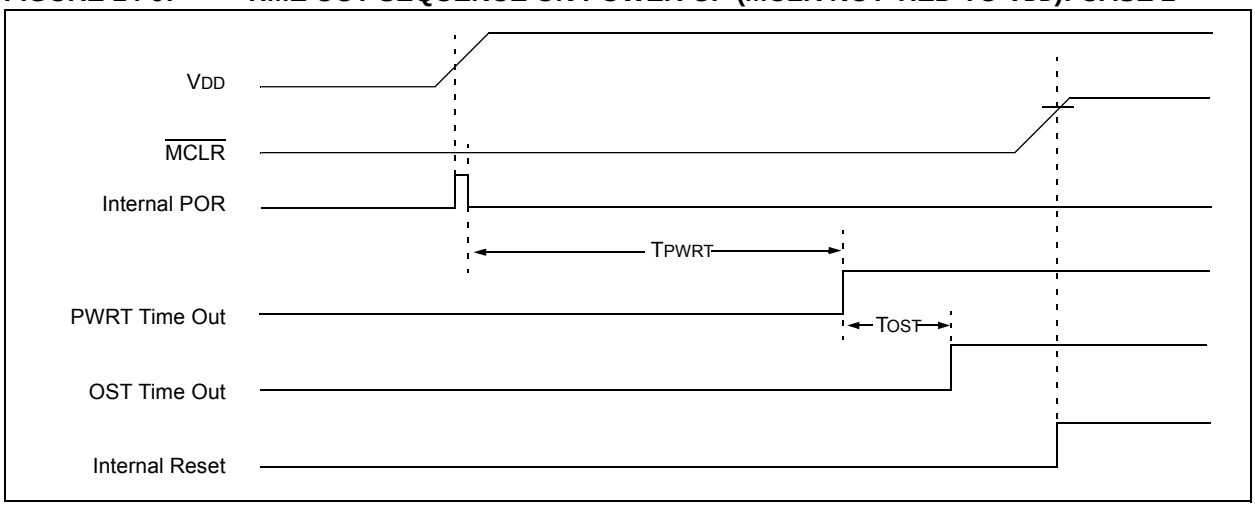



**FIGURE 14-10: TIME OUT SEQUENCE ON POWER-UP (MCLR TIED TO VDD)**

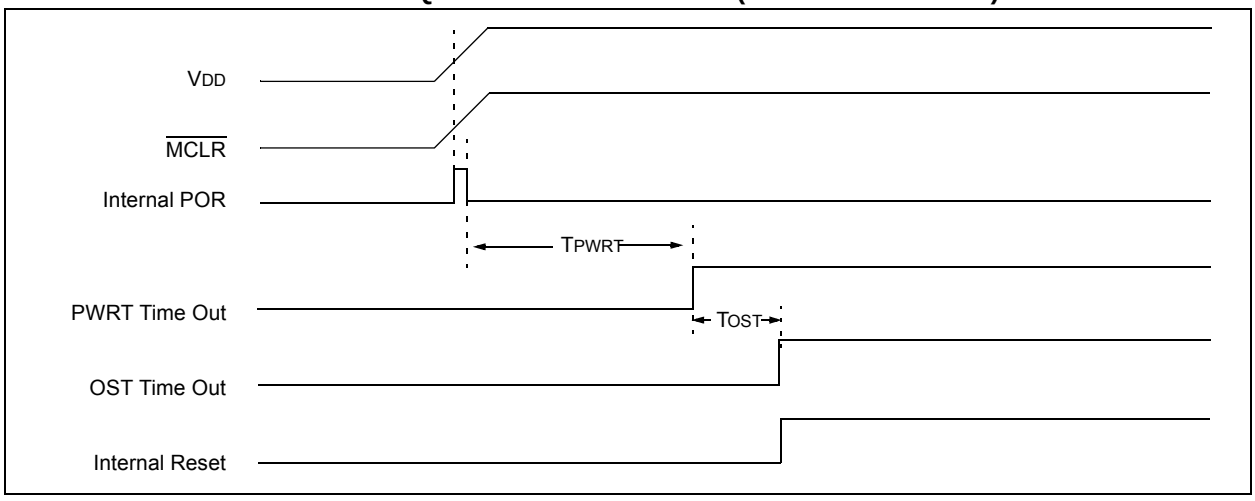
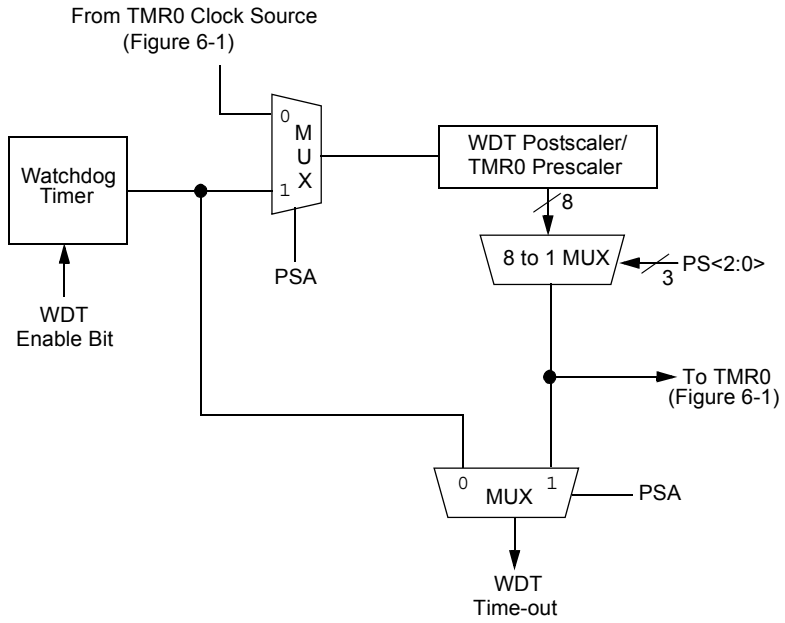


**FIGURE 14-16: WATCHDOG TIMER BLOCK DIAGRAM**




**Note:** T0SE, T0CS, PSA, PS0-PS2 are bits in the OPTION register.

**TABLE 14-9: SUMMARY OF WATCHDOG TIMER REGISTERS**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR Reset | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 2007h | CONFIG | LVP | BOREN | MCLRE | FOSC2 | $\overline{\text{PWRTE}}$ | WDTE | FOSC1 | FOSC0 | uuuu uuuu | uuuu uuuu |
| 81h, 181h | OPTION | $\overline{\text{RBPU}}$ | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 | 1111 1111 | 1111 1111 |

**Legend:** x = unknown, u = unchanged, - = unimplemented read as '0', q = value depends upon condition.

**Note:** Shaded cells are not used by the Watchdog Timer.

## 14.8 Power-Down Mode (Sleep)

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

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

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

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

| **Note:** | It should be noted that a Reset generated by a WDT time-out does not drive $\overline{\text{MCLR}}$ pin low. |
|---|---|

## 14.13 In-Circuit Debugger

Since in-circuit debugging requires the loss of clock, data and $\overline{MCLR}$ pins, MPLAB® ICD 2 development with an 18-pin device is not practical. A special 28-pin PIC16F648A-ICD device is used with MPLAB ICD 2 to provide separate clock, data and $\overline{MCLR}$ pins and frees all normally available pins to the user. Debugging of all three versions of the PIC16F627A/628A/648A is supported by the PIC16F648A-ICD.

This special ICD device is mounted on the top of a header and its signals are routed to the MPLAB ICD 2 connector. On the bottom of the header is an 18-pin socket that plugs into the user's target via an 18-pin stand-off connector.

When the ICD pin on the PIC16F648A-ICD device is held low, the In-Circuit Debugger functionality is enabled. This function allows simple debugging functions when used with MPLAB ICD 2. When the microcontroller has this feature enabled, some of the resources are not available for general use. Table 14-19 shows which features are consumed by the background debugger.

**TABLE 14-19: DEBUGGER RESOURCES**

| I/O pins | ICDCLK, ICDDATA |
|---|---|
| Stack | 1 level |
| Program Memory | Address 0h must be `NOP`<br>300h-3FEh |

The PIC16F648A-ICD device with header is supplied as an assembly. See Microchip Part Number AC162053.

## 16.7 MPLAB SIM Software Simulator

The MPLAB SIM Software Simulator allows code development in a PC-hosted environment by simulating the PIC MCUs and dsPIC® DSCs on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a comprehensive stimulus controller. Registers can be logged to files for further run-time analysis. The trace buffer and logic analyzer display extend the power of the simulator to record and track program execution, actions on I/O, most peripherals and internal registers.

The MPLAB SIM Software Simulator fully supports symbolic debugging using the MPLAB C Compilers, and the MPASM and MPLAB Assemblers. The software simulator offers the flexibility to develop and debug code outside of the hardware laboratory environment, making it an excellent, economical software development tool.

## 16.8 MPLAB REAL ICE In-Circuit Emulator System

MPLAB REAL ICE In-Circuit Emulator System is Microchip's next generation high-speed emulator for Microchip Flash DSC and MCU devices. It debugs and programs PIC® Flash MCUs and dsPIC® Flash DSCs with the easy-to-use, powerful graphical user interface of the MPLAB Integrated Development Environment (IDE), included with each kit.

The emulator is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with either a connector compatible with in-circuit debugger systems (RJ11) or with the new high-speed, noise tolerant, Low-Voltage Differential Signal (LVDS) interconnection (CAT5).

The emulator is field upgradable through future firmware downloads in MPLAB IDE. In upcoming releases of MPLAB IDE, new devices will be supported, and new features will be added. MPLAB REAL ICE offers significant advantages over competitive emulators including low-cost, full-speed emulation, run-time variable watches, trace analysis, complex breakpoints, a ruggedized probe interface and long (up to three meters) interconnection cables.

## 16.9 MPLAB ICD 3 In-Circuit Debugger System

MPLAB ICD 3 In-Circuit Debugger System is Microchip's most cost effective high-speed hardware debugger/programmer for Microchip Flash Digital Signal Controller (DSC) and microcontroller (MCU) devices. It debugs and programs PIC® Flash microcontrollers and dsPIC® DSCs with the powerful, yet easy-to-use graphical user interface of MPLAB Integrated Development Environment (IDE).

The MPLAB ICD 3 In-Circuit Debugger probe is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with a connector compatible with the MPLAB ICD 2 or MPLAB REAL ICE systems (RJ-11). MPLAB ICD 3 supports all MPLAB ICD 2 headers.

## 16.10 PICkit 3 In-Circuit Debugger/ Programmer and PICkit 3 Debug Express

The MPLAB PICkit 3 allows debugging and programming of PIC® and dsPIC® Flash microcontrollers at a most affordable price point using the powerful graphical user interface of the MPLAB Integrated Development Environment (IDE). The MPLAB PICkit 3 is connected to the design engineer's PC using a full speed USB interface and can be connected to the target via an Microchip debug (RJ-11) connector (compatible with MPLAB ICD 3 and MPLAB REAL ICE). The connector uses two device I/O pins and the reset line to implement in-circuit debugging and In-Circuit Serial Programming™.

The PICkit 3 Debug Express include the PICkit 3, demo board and microcontroller, hookup cables and CDROM with user's guide, lessons, tutorial, compiler and MPLAB IDE software.

## 17.2 DC Characteristics: PIC16F627A/628A/648A (Industrial) PIC16LF627A/628A/648A (Industrial)

| DC CHARACTERISTICS | | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature -40°C ≤ Ta ≤ +85°C for industrial | | | | | |
|---|---|---|---|---|---|---|---|
| Param No. | LF and F Device Characteristics | Min† | Typ | Max | Units | Conditions | |
| | | | | | | VDD | Note |
| **Supply Voltage (VDD)** | | | | | | | |
| D001 | LF | 2.0 | — | 5.5 | V | — | |
| | LF/F | 3.0 | — | 5.5 | V | — | |
| **Power-down Base Current (IPD)** | | | | | | | |
| D020 | LF | — | 0.01 | 0.80 | µA | 2.0 | WDT, BOR, Comparators, VREF and T1OSC: disabled |
| | LF/F | — | 0.01 | 0.85 | µA | 3.0 | |
| | | — | 0.02 | 2.7 | µA | 5.0 | |
| **Peripheral Module Current (ΔIMOD)(1)** | | | | | | | |
| D021 | LF | — | 1 | 2.0 | µA | 2.0 | WDT Current |
| | LF/F | — | 2 | 3.4 | µA | 3.0 | |
| | | — | 9 | 17.0 | µA | 5.0 | |
| D022 | LF/F | — | 29 | 52 | µA | 4.5 | BOR Current |
| | | — | 30 | 55 | µA | 5.0 | |
| D023 | LF | — | 15 | 22 | µA | 2.0 | Comparator Current (Both comparators enabled) |
| | LF/F | — | 22 | 37 | µA | 3.0 | |
| | | — | 44 | 68 | µA | 5.0 | |
| D024 | LF | — | 34 | 55 | µA | 2.0 | VREF Current |
| | LF/F | — | 50 | 75 | µA | 3.0 | |
| | | — | 80 | 110 | µA | 5.0 | |
| D025 | LF | — | 1.2 | 2.0 | µA | 2.0 | T1OSC Current |
| | LF/F | — | 1.3 | 2.2 | µA | 3.0 | |
| | | — | 1.8 | 2.9 | µA | 5.0 | |
| **Supply Current (IDD)** | | | | | | | |
| D010 | LF | — | 10 | 15 | µA | 2.0 | FOSC = 32 kHz LP Oscillator Mode |
| | LF/F | — | 15 | 25 | µA | 3.0 | |
| | | — | 28 | 48 | µA | 5.0 | |
| D011 | LF | — | 125 | 190 | µA | 2.0 | FOSC = 1 MHz XT Oscillator Mode |
| | LF/F | — | 175 | 340 | µA | 3.0 | |
| | | — | 320 | 520 | µA | 5.0 | |
| D012 | LF | — | 250 | 350 | µA | 2.0 | FOSC = 4 MHz XT Oscillator Mode |
| | LF/F | — | 450 | 600 | µA | 3.0 | |
| | | — | 710 | 995 | µA | 5.0 | |
| D012A | LF | — | 395 | 465 | µA | 2.0 | FOSC = 4 MHz INTOSC |
| | LF/F | — | 565 | 785 | µA | 3.0 | |
| | | — | 0.895 | 1.3 | mA | 5.0 | |
| D013 | LF/F | — | 2.5 | 2.9 | mA | 4.5 | FOSC = 20 MHz HS Oscillator Mode |
| | | — | 2.75 | 3.3 | mA | 5.0 | |

**Note 1:** 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. Max values should be used when calculating total current consumption.

**TABLE 17-8: TIMER0 AND TIMER1 EXTERNAL 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 | | | Greater of: 20 or $\frac{T_{CY} + 40^*}{N}$ | — | — | ns | N = prescale value (2, 4, ..., 256) |
| 45 | TT1H | T1CKI High Time | Synchronous, No Prescaler | | 0.5TCY + 20* | — | — | ns | |
| | | | Synchronous, with Prescaler | PIC16F62XA | 15* | — | — | ns | |
| | | | | PIC16LF62XA | 25* | — | — | ns | |
| | | | Asynchronous | PIC16F62XA | 30* | — | — | ns | |
| | | | | PIC16LF62XA | 50* | — | — | ns | |
| 46 | TT1L | T1CKI Low Time | Synchronous, No Prescaler | | 0.5TCY + 20* | — | — | ns | |
| | | | Synchronous, with Prescaler | PIC16F62XA | 15* | — | — | ns | |
| | | | | PIC16LF62XA | 25* | — | — | ns | |
| | | | Asynchronous | PIC16F62XA | 30* | — | — | ns | |
| | | | | PIC16LF62XA | 50* | — | — | ns | |
| 47 | TT1P | T1CKI input period | Synchronous | PIC16F62XA | Greater of: 20 or $\frac{T_{CY} + 40^*}{N}$ | — | — | ns | N = prescale value (1, 2, 4, 8) |
| | | | | PIC16LF62XA | Greater of: 20 or $\frac{T_{CY} + 40^*}{N}$ | — | — | — | |
| | | | Asynchronous | PIC16F62XA | 60* | — | — | ns | |
| | | | | PIC16LF62XA | 100* | — | — | ns | |
| | FT1 | Timer1 oscillator input frequency range (oscillator enabled by setting bit T1OSCEN) | | | — | 32.7(1) | — | kHz | |
| 48 | TCKEZTMR1 | Delay from external clock edge to timer increment | | | 2TOSC | — | 7TOSC | — | |

* 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.

**Note 1:** This oscillator is intended to work only with 32.768 kHz watch crystals and their manufactured tolerances. Higher value crystal frequencies may not be compatible with this crystal driver.

**FIGURE 18-6: TYPICAL VREF IPD VS. VDD**

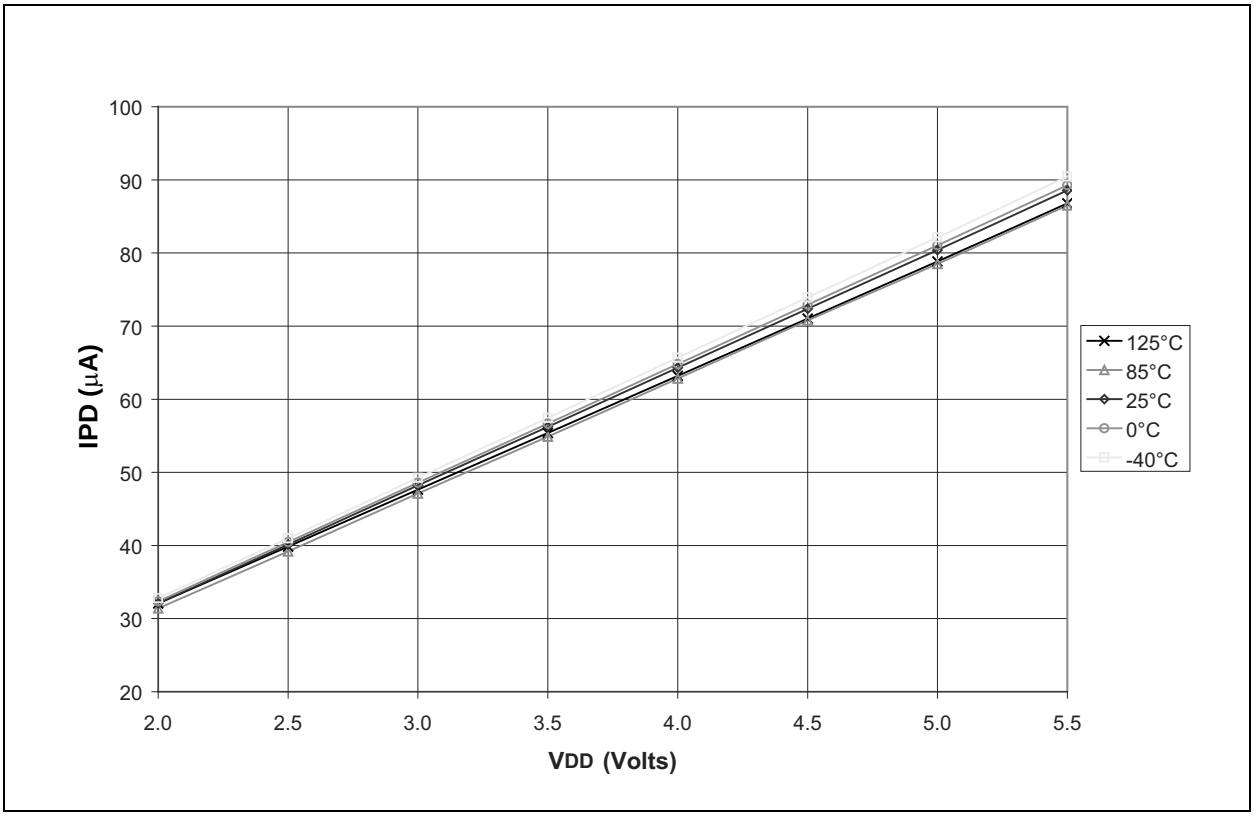



**FIGURE 18-7: TYPICAL WDT IPD VS. VDD**

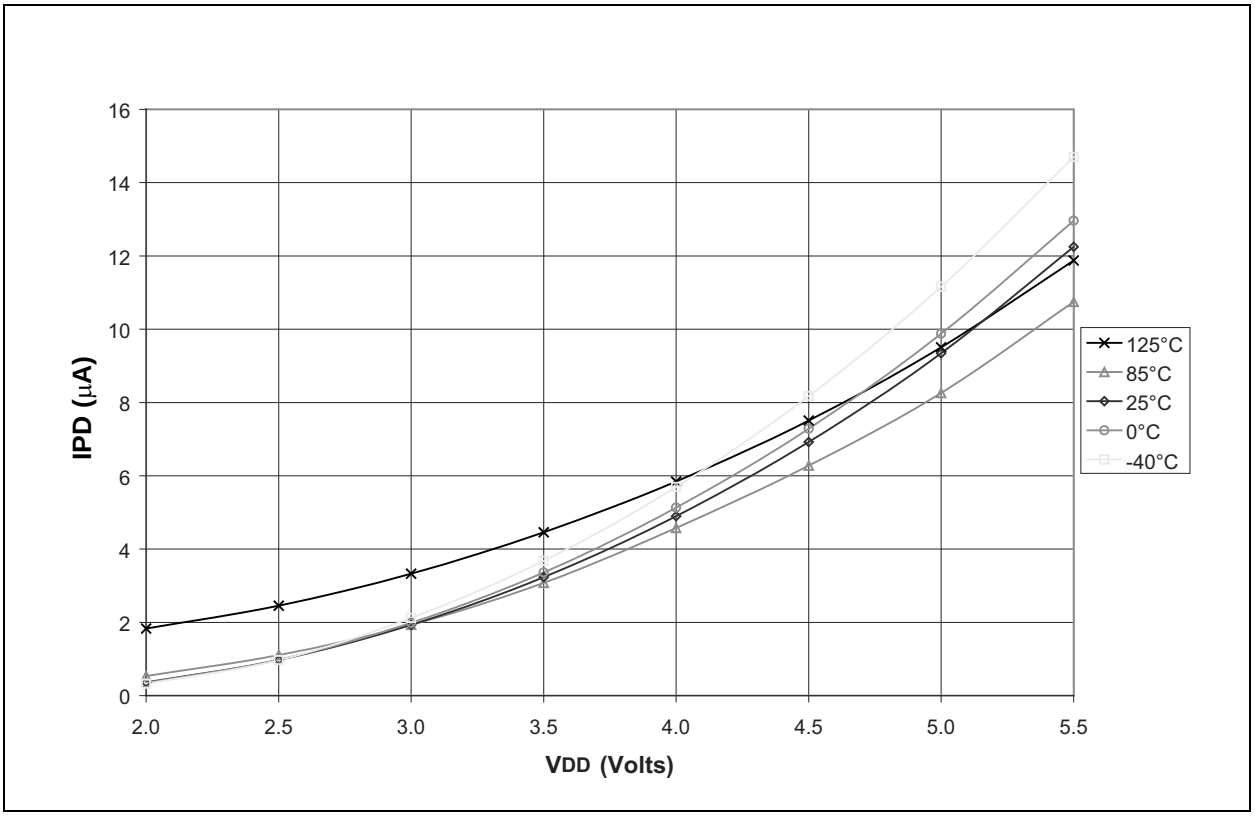
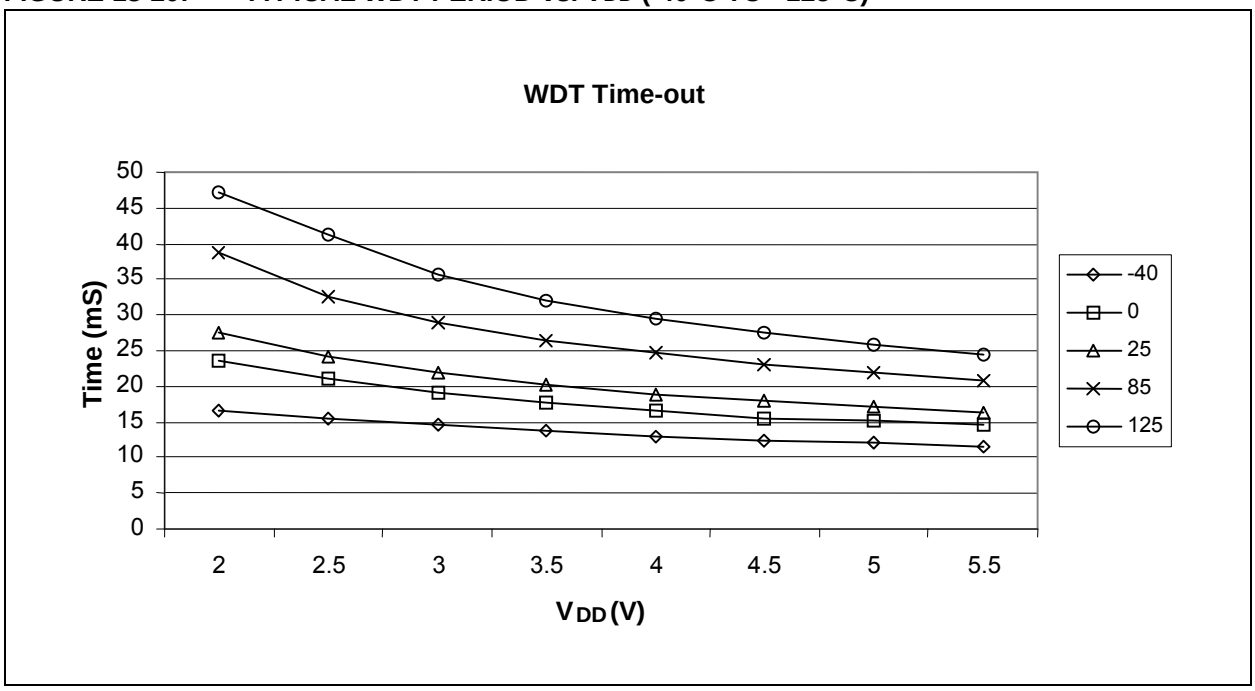
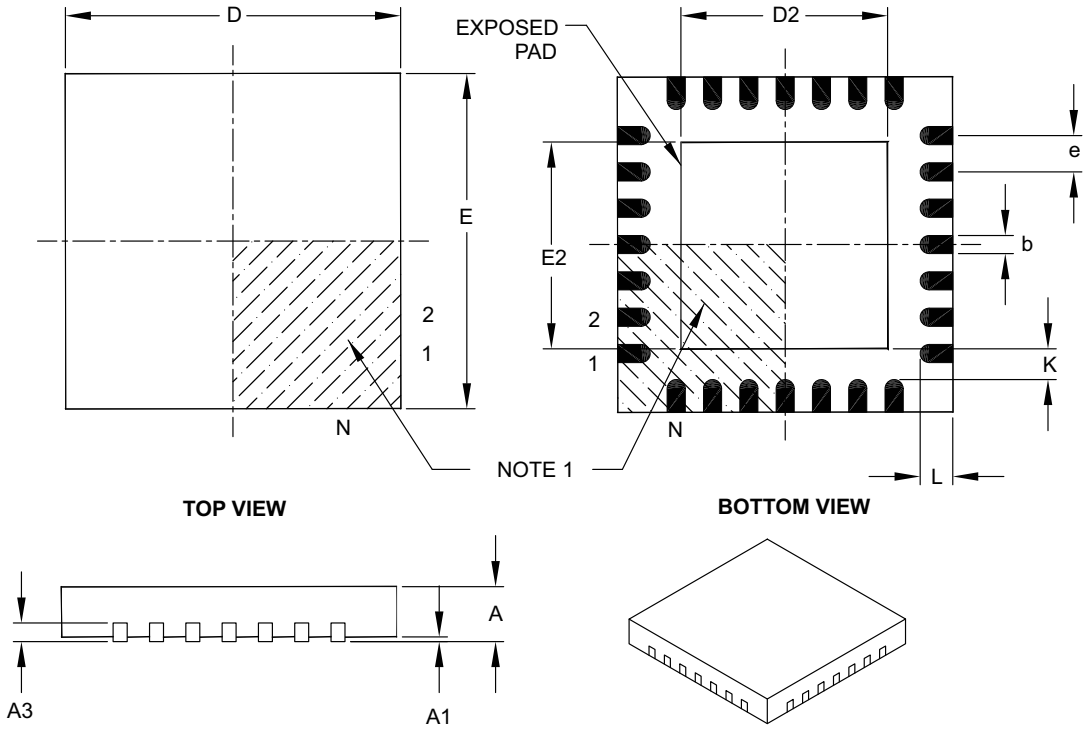


**FIGURE 18-20: TYPICAL WDT PERIOD vs. $V_{DD}$ (-40°C TO +125°C)**



### 28-Lead Plastic Quad Flat, No Lead Package (ML) – 6x6 mm Body [QFN] with 0.55 mm Contact Length

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging




| Units | | MILLIMETERS | | |
|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX |
| Number of Pins | N | | 28 | |
| Pitch | e | | 0.65 BSC | |
| Overall Height | A | 0.80 | 0.90 | 1.00 |
| Standoff | A1 | 0.00 | 0.02 | 0.05 |
| Contact Thickness | A3 | | 0.20 REF | |
| Overall Width | E | | 6.00 BSC | |
| Exposed Pad Width | E2 | 3.65 | 3.70 | 4.20 |
| Overall Length | D | | 6.00 BSC | |
| Exposed Pad Length | D2 | 3.65 | 3.70 | 4.20 |
| Contact Width | b | 0.23 | 0.30 | 0.35 |
| Contact Length | L | 0.50 | 0.55 | 0.70 |
| Contact-to-Exposed Pad | K | 0.20 | – | – |

**Notes:**

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package is saw singulated.
3. Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-105B

## THE MICROCHIP WEB SITE

Microchip provides online support via our WWW site at www.microchip.com. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

## CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at www.microchip.com, click on Customer Change Notification and follow the registration instructions.

## CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support
- Development Systems Information Line

Customers should contact their distributor, representative or field application engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

**Technical support is available through the web site at: http://support.microchip.com**