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, POR, WDT |
| Number of I/O | 13 |
| Program Memory Size | 896B (512 x 14) |
| Program Memory Type | OTP |
| EEPROM Size | - |
| RAM Size | 96 x 8 |
| Voltage - Supply (Vcc/Vdd) | 2.5V ~ 5.5V |
| Data Converters | - |
| Oscillator Type | External |
| Operating Temperature | 0°C ~ 70°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 20-SSOP (0.209", 5.30mm Width) |
| Supplier Device Package | 20-SSOP |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic16c620a-20-ss |

## 3.1 Clocking Scheme/Instruction Cycle

The clock input (OSC1/CLKIN pin) 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 3-2.

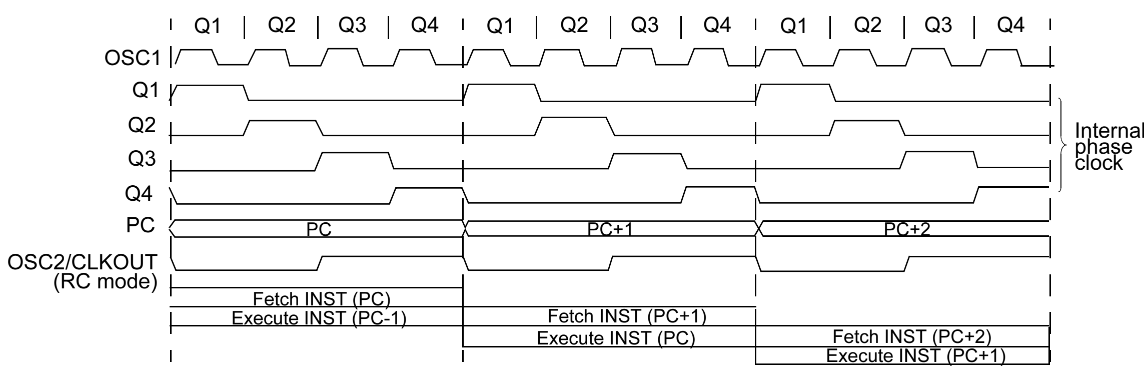
## 3.2 Instruction Flow/Pipelining

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 3-2: CLOCK/INSTRUCTION CYCLE**




**EXAMPLE 3-1: INSTRUCTION PIPELINE FLOW**

| | | | | | | |
|---|---|---|---|---|---|---|
| 1. `MOVLW 55h` | Fetch 1 | Execute 1 | | | | |
| 2. `MOVWF PORTB` | | Fetch 2 | Execute 2 | | | |
| 3. `CALL SUB_1` | | | Fetch 3 | Execute 3 | | |
| 4. `BSF PORTA, BIT3` | | | | Fetch 4 | Flush | |
| | | | | | Fetch SUB_1 | Execute SUB_1 |

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

#### 4.2.2.1 STATUS Register

The STATUS register, shown in Register 4-1, contains the arithmetic status of the ALU, the RESET status and the bank select bits for data memory.

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

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

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

> **Note 1:** The IRP and RP1 bits (STATUS<7:6>) are not used by the PIC16C62X and should be programmed as ’0'. Use of these bits as general purpose R/W bits is NOT recommended, since this may affect upward compatibility with future products.
>
> **2:** The C and DC bits operate as a $\overline{\text{Borrow}}$ and Digit Borrow out bit, respectively, in subtraction. See the `SUBLW` and `SUBWF` instructions for examples.

**REGISTER 4-1: STATUS REGISTER (ADDRESS 03H OR 83H)**

| Reserved | Reserved | R/W-0 | R-1 | R-1 | R/W-x | R/W-x | R/W-x |
|---|---|---|---|---|---|---|---|
| IRP | RP1 | RP0 | $\overline{\text{TO}}$ | $\overline{\text{PD}}$ | Z | DC | C |
| bit 7 | | | | | | | bit 0 |

bit 7 **IRP:** Register Bank Select bit (used for indirect addressing)
`1` = Bank 2, 3 (100h - 1FFh)
`0` = Bank 0, 1 (00h - FFh)
The IRP bit is reserved on the PIC16C62X; always maintain this bit clear.

bit 6-5 **RP<1:0>**: Register Bank Select bits (used for direct addressing)
`01` = Bank 1 (80h - FFh)
`00` = Bank 0 (00h - 7Fh)
Each bank is 128 bytes. The RP1 bit is reserved on the PIC16C62X; always maintain this bit clear.

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

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

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

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

bit 0 **C**: Carry/$\overline{\text{borrow}}$ bit (`ADDWF`, `ADDLW`, `SUBLW`, `SUBWF` instructions)
`1` = A carry-out from the Most Significant bit of the result occurred
`0` = No carry-out from the Most Significant bit of the result occurred

> **Note:** For $\overline{\text{borrow}}$ the polarity is reversed. A subtraction is executed by adding the two’s complement of the second operand. For rotate (`RRF`, `RLF`) instructions, this bit is loaded with either the high or low order bit of the source register.

| 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 |

### 4.2.2.3 INTCON Register

The INTCON register is a readable and writable register, which contains the various enable and flag bits for all interrupt sources except the comparator module. See Section 4.2.2.4 and Section 4.2.2.5 for a description of the comparator enable and 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>). |
|---|---|

**REGISTER 4-3: INTCON REGISTER (ADDRESS 0BH OR 8BH)**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-x |
|---|---|---|---|---|---|---|---|
| GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF |
| bit 7 | | | | | | | bit 0 |

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

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

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

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

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

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

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

bit 0 **RBIF**: RB Port Change Interrupt Flag bit
1 = When at least one of the RB<7:4> pins changed state (must be cleared in software)
0 = None of the RB<7:4> pins have changed state

| 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 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, 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 | Value on All Other RESETS |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 06h | PORTB | RB7 | RB6 | RB5 | RB4 | RB3 | RB2 | RB1 | RB0 | xxxx xxxx | uuuu uuuu |
| 86h | TRISB | TRISB7 | TRISB6 | TRISB5 | TRISB4 | TRISB3 | TRISB2 | TRISB1 | TRISB0 | 1111 1111 | 1111 1111 |
| 81h | OPTION | $\overline{\text{RBPU}}$ | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 | 1111 1111 | 1111 1111 |

Legend: u = unchanged, x = unknown

**Note 1:** Shaded bits are not used by PORTB.

## 6.0 TIMER0 MODULE

The Timer0 module timer/counter has the following features:

- 8-bit timer/counter
- Readable and writable
- 8-bit software programmable prescaler
- Internal or external clock select
- Interrupt on overflow from FFh to 00h
- Edge select for external clock

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

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

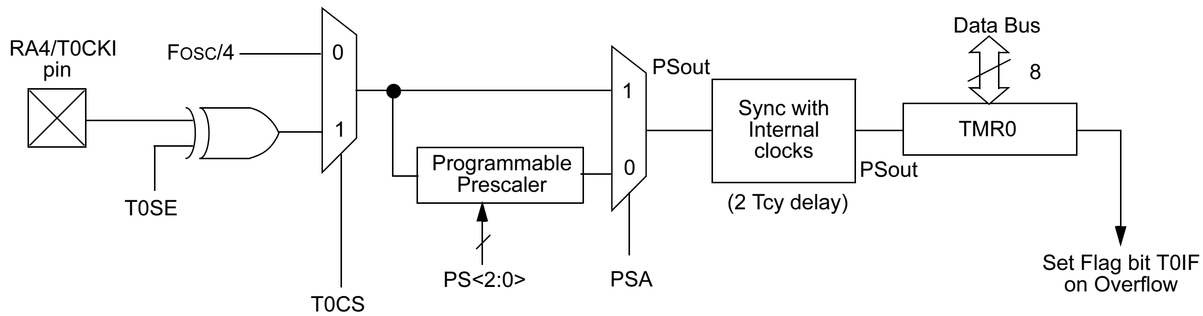
Counter mode is selected by setting the T0CS bit. In this mode, Timer0 will increment either on every rising or falling edge of pin RA4/T0CKI. The incrementing edge is determined by the source edge (T0SE) control bit (OPTION<4>). Clearing the T0SE bit selects the rising edge. Restrictions on the external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

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

### 6.1 TIMER0 Interrupt

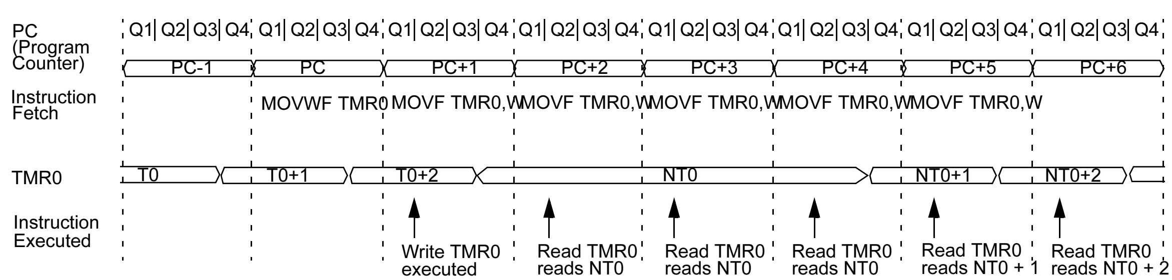
Timer0 interrupt is generated when the TMR0 register timer/counter overflows from FFh to 00h. This overflow sets the T0IF bit. The interrupt can be masked by clearing the T0IE bit (INTCON<5>). The T0IF bit (INTCON<2>) must be cleared in software by the Timer0 module interrupt service routine before re-enabling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

**FIGURE 6-1: TIMER0 BLOCK DIAGRAM**




**Note 1:** Bits T0SE, T0CS, PS2, PS1, PS0 and PSA are located in the OPTION register.
**2:** The prescaler is shared with Watchdog Timer (Figure 6-6).

**FIGURE 6-2: TIMER0 (TMR0) TIMING: INTERNAL CLOCK/NO PRESCALER**



### 9.2.3 EXTERNAL CRYSTAL OSCILLATOR CIRCUIT

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

Figure 9-3 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 9-3: EXTERNAL PARALLEL RESONANT CRYSTAL OSCILLATOR CIRCUIT**

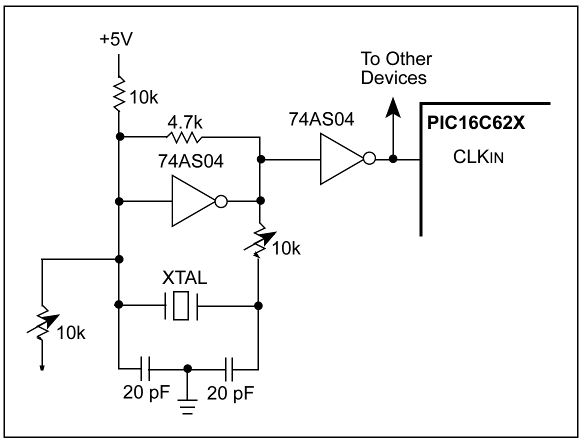
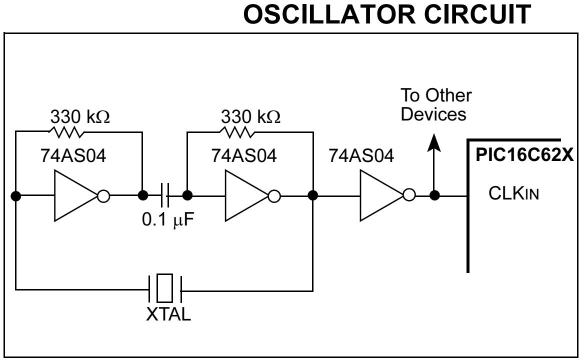



Figure 9-4 shows a series resonant oscillator circuit. This circuit is also designed to use the fundamental frequency of the crystal. The inverter performs a 180° phase shift in a series resonant oscillator circuit. The 330 kΩ resistors provide the negative feedback to bias the inverters in their linear region.

**FIGURE 9-4: EXTERNAL SERIES RESONANT CRYSTAL OSCILLATOR CIRCUIT**




### 9.2.4 RC OSCILLATOR

For timing insensitive applications the "RC" device option offers additional cost savings. The RC oscillator frequency is a function of the supply voltage, the resistor (REXT) and capacitor (CEXT) values, and the operating temperature. In addition to this, the oscillator frequency will vary from unit to unit due to normal process parameter variation. Furthermore, the difference in lead frame capacitance between package types will also affect the oscillation frequency, especially for low CEXT values. The user also needs to take into account variation due to tolerance of external R and C components used. Figure 9-5 shows how the R/C combination is connected to the PIC16C62X. For REXT values below 2.2 kΩ, the oscillator operation may become unstable or stop completely. For very high REXT values (e.g., 1 MΩ), the oscillator becomes sensitive to noise, humidity and leakage. Thus, we recommend to keep REXT between 3 kΩ and 100 kΩ.

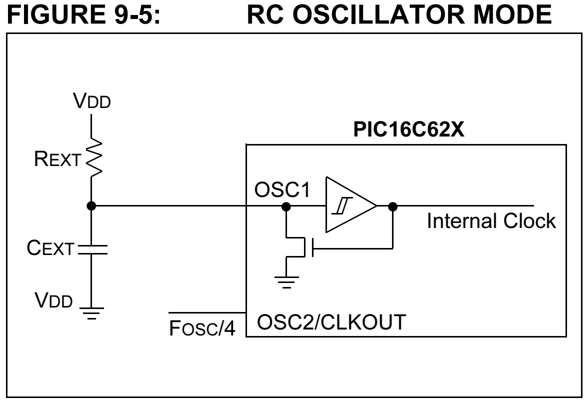
Although the oscillator will operate with no external capacitor (CEXT = 0 pF), we recommend using values above 20 pF for noise and stability reasons. With no or small external capacitance, the oscillation frequency can vary dramatically due to changes in external capacitances, such as PCB trace capacitance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

See Section 13.0 for RC frequency variation from part to part due to normal process variation. The variation is larger for larger R (since leakage current variation will affect RC frequency more for large R) and for smaller C (since variation of input capacitance will affect RC frequency more).

See Section 13.0 for variation of oscillator frequency due to VDD for given REXT/CEXT values, as well as frequency variation due to operating temperature for given R, C and VDD values.

The oscillator frequency, divided by 4, is available on the OSC2/CLKOUT pin, and can be used for test purposes or to synchronize other logic (Figure 3-2 for waveform).

**FIGURE 9-5: RC OSCILLATOR MODE**



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

### 9.4.1 POWER-ON RESET (POR)

The on-chip POR circuit holds the chip in RESET until VDD has reached a high enough level for proper operation. To take advantage of the POR, just tie the $\overline{MCLR}$ pin through a resistor to VDD. This will eliminate external RC components usually needed to create Power-on Reset. A maximum rise time for VDD is required. See Electrical Specifications for details.

The POR circuit does not produce an internal RESET when VDD declines.

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

For additional information, refer to Application Note AN607, “Power-up Trouble Shooting”.

### 9.4.2 POWER-UP TIMER (PWRT)

The Power-up Timer provides a fixed 72 ms (nominal) time-out on power-up only, from POR or Brown-out Reset. The Power-up Timer operates on an internal RC oscillator. The chip is kept in RESET as long as PWRT is active. The PWRT delay allows the VDD to rise to an acceptable level. A configuration bit, $\overline{PWRTE}$ can disable (if set) or enable (if cleared or programmed) the Power-up Timer. The Power-up Timer should always be enabled when Brown-out Reset is enabled.

The Power-up Time delay will vary from chip-to-chip and due to VDD, temperature and process variation. See DC parameters for details.

### 9.4.3 OSCILLATOR START-UP TIMER (OST)

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

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

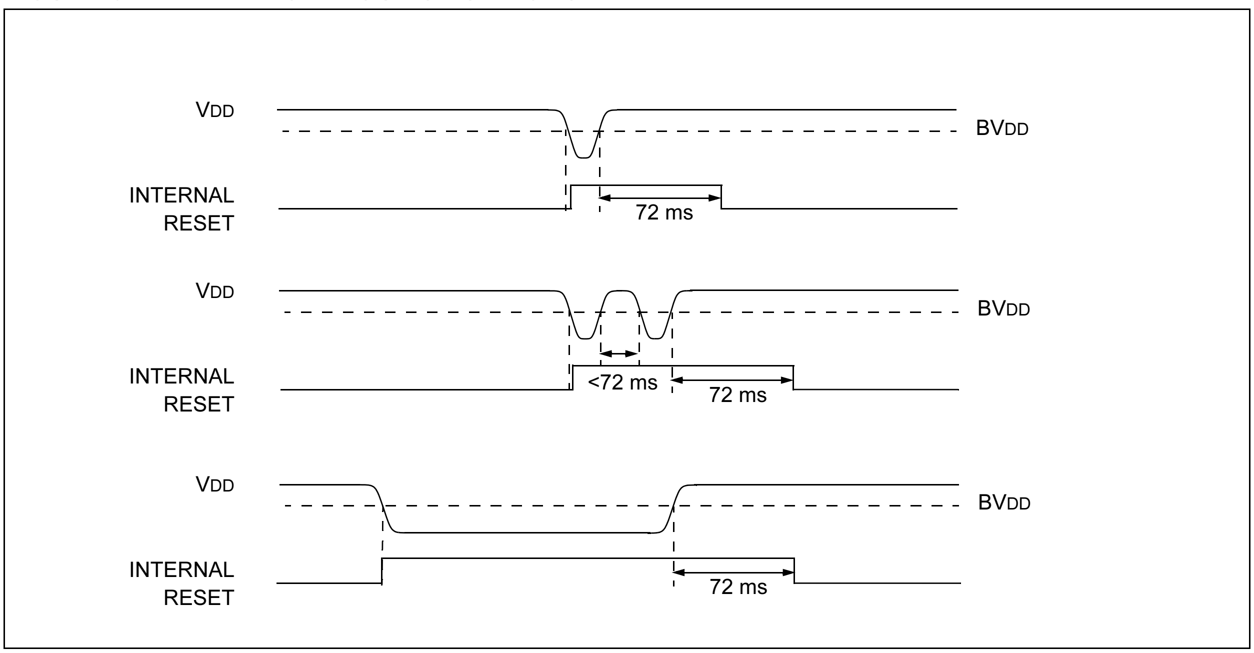
### 9.4.4 BROWN-OUT RESET (BOR)

The PIC16C62X members have on-chip Brown-out Reset circuitry. A configuration bit, BODEN, can disable (if clear/programmed) or enable (if set) the Brown-out Reset circuitry. If VDD falls below 4.0V refer to VBOR parameter D005 (VBOR) for greater than parameter (TBOR) in Table 12-5. The brown-out situation will RESET the chip. A RESET won’t occur if VDD falls below 4.0V for less than parameter (TBOR).

On any RESET (Power-on, Brown-out, Watchdog, etc.) the chip will remain in RESET until VDD rises above BVDD. The Power-up Timer will now be invoked and will keep the chip in RESET an additional 72 ms.

If VDD drops below BVDD while the Power-up Timer is running, the chip will go back into a Brown-out Reset and the Power-up Timer will be re-initialized. Once VDD rises above BVDD, the Power-Up Timer will execute a 72 ms RESET. The Power-up Timer should always be enabled when Brown-out Reset is enabled. Figure 9-7 shows typical Brown-out situations.

**FIGURE 9-7: BROWN-OUT SITUATIONS**



### 9.4.5 TIME-OUT SEQUENCE

On power-up the time-out sequence is as follows: First PWRT time-out is invoked after POR has expired. Then OST is activated. The total time-out will vary based on oscillator configuration and $\overline{\text{PWRTE}}$ bit status. For example, in RC mode with $\overline{\text{PWRTE}}$ bit erased (PWRT disabled), there will be no time-out at all. Figure 9-8, Figure 9-9 and Figure 9-10 depict time-out sequences.

Since the time-outs occur from the POR pulse, if $\overline{\text{MCLR}}$ is kept low long enough, the time-outs will expire. Then bringing $\overline{\text{MCLR}}$ high will begin execution immediately (see Figure 9-9). This is useful for testing purposes or to synchronize more than one PIC16C62X device operating in parallel.

Table 9-4 shows the RESET conditions for some special registers, while Table 9-5 shows the RESET conditions for all the registers.

### 9.4.6 POWER CONTROL (PCON)/ STATUS REGISTER

The power control/STATUS register, PCON (address 8Eh), has two bits.

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

Bit1 is $\overline{\text{POR}}$ (Power-on Reset). It is a '0' on Power-on Reset and unaffected otherwise. The user must write a '1' to this bit following a Power-on Reset. On a subsequent RESET, if $\overline{\text{POR}}$ is '0', it will indicate that a Power-on Reset must have occurred ($V_{DD}$ may have gone too low).

**TABLE 9-1: TIME-OUT IN VARIOUS SITUATIONS**

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

**TABLE 9-2: STATUS/PCON BITS AND THEIR SIGNIFICANCE**

| POR | BOR | TO | PD | |
|---|---|---|---|---|
| 0 | X | 1 | 1 | Power-on Reset |
| 0 | X | 0 | X | Illegal, $\overline{\text{TO}}$ is set on $\overline{\text{POR}}$ |
| 0 | X | X | 0 | Illegal, $\overline{\text{PD}}$ is set on $\overline{\text{POR}}$ |
| 1 | 0 | X | X | Brown-out Reset |
| 1 | 1 | 0 | u | WDT Reset |
| 1 | 1 | 0 | 0 | WDT Wake-up |
| 1 | 1 | u | u | $\overline{\text{MCLR}}$ Reset during normal operation |
| 1 | 1 | 1 | 0 | $\overline{\text{MCLR}}$ Reset during SLEEP |

Legend: u = unchanged, x = unknown

**TABLE 9-3: SUMMARY OF REGISTERS ASSOCIATED WITH BROWN-OUT**

| 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(1) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 83h | STATUS | | | | $\overline{\text{TO}}$ | $\overline{\text{PD}}$ | | | | 0001 1xxx | 000q quuu |
| 8Eh | PCON | — | — | — | — | — | — | $\overline{\text{POR}}$ | $\overline{\text{BOR}}$ | ---- --0x | ---- --uq |

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

**Note 1:** Other (non Power-up) Resets include $\overline{\text{MCLR}}$ Reset, Brown-out Reset and Watchdog Timer Reset during normal operation.

**TABLE 9-4: INITIALIZATION CONDITION FOR SPECIAL REGISTERS**

| Condition | Program Counter | STATUS Register | PCON Register |
|---|---|---|---|
| Power-on Reset | 000h | 0001 1xxx | ---- --0x |
| $\overline{MCLR}$ Reset during normal operation | 000h | 000u uuuu | ---- --uu |
| $\overline{MCLR}$ Reset during SLEEP | 000h | 0001 0uuu | ---- --uu |
| WDT Reset | 000h | 0000 uuuu | ---- --uu |
| WDT Wake-up | PC + 1 | uuu0 0uuu | ---- --uu |
| Brown-out Reset | 000h | 000x xuuu | ---- --u0 |
| Interrupt Wake-up from SLEEP | PC + 1(1) | uuu1 0uuu | ---- --uu |

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

**TABLE 9-5: INITIALIZATION CONDITION FOR REGISTERS**

| Register | Address | Power-on Reset | • $\overline{MCLR}$ Reset during normal operation<br>• $\overline{MCLR}$ Reset during SLEEP<br>• WDT Reset<br>• Brown-out Reset (1) | • Wake-up from SLEEP through interrupt<br>• Wake-up from SLEEP through WDT time-out |
|---|---|---|---|---|
| W | — | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| INDF | 00h | — | — | — |
| TMR0 | 01h | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| PCL | 02h | 0000 0000 | 0000 0000 | PC + 1(3) |
| STATUS | 03h | 0001 1xxx | 000q quuu(4) | uuuq quuu(4) |
| FSR | 04h | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| PORTA | 05h | ---x xxxx | ---u uuuu | ---u uuuu |
| PORTB | 06h | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| CMCON | 1Fh | 00-- 0000 | 00-- 0000 | uu-- uuuu |
| PCLATH | 0Ah | ---0 0000 | ---0 0000 | ---u uuuu |
| INTCON | 0Bh | 0000 000x | 0000 000u | uuuu uqqq(2) |
| PIR1 | 0Ch | -0-- ---- | -0-- ---- | -q-- ----(2,5) |
| OPTION | 81h | 1111 1111 | 1111 1111 | uuuu uuuu |
| TRISA | 85h | ---1 1111 | ---1 1111 | ---u uuuu |
| TRISB | 86h | 1111 1111 | 1111 1111 | uuuu uuuu |
| PIE1 | 8Ch | -0-- ---- | -0-- ---- | -u-- ---- |
| PCON | 8Eh | ---- --0x | ---- --uq(1,6) | ---- --uu |
| VRCON | 9Fh | 000- 0000 | 000- 0000 | uuu- uuuu |

Legend: u = unchanged, x = unknown, - = unimplemented bit, reads as '0',q = value depends on condition.
**Note 1:** If $V_{DD}$ goes too low, Power-on Reset will be activated and registers will be affected differently.
**2:** One or more bits in INTCON, PIR1 and/or PIR2 will be affected (to cause wake-up).
**3:** When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).
**4:** See Table 9-4 for RESET value for specific condition.
**5:** If wake-up was due to comparator input changing, then bit 6 = 1. All other interrupts generating a wake-up will cause bit 6 = u.
**6:** If RESET was due to brown-out, then bit 0 = 0. All other RESETS will cause bit 0 = u.

### BCF — Bit Clear f

| | |
|---|---|
| Syntax: | [ *label* ] BCF f,b |
| Operands: | $0 \le f \le 127$<br>$0 \le b \le 7$ |
| Operation: | $0 \rightarrow$ (f<b>) |
| Status Affected: | None |
| Encoding: | `01` `00bb` `bfff` `ffff` |
| Description: | Bit 'b' in register 'f' is cleared. |
| Words: | 1 |
| Cycles: | 1 |
| Example | `BCF FLAG_REG, 7`<br>Before Instruction<br>FLAG_REG = 0xC7<br>After Instruction<br>FLAG_REG = 0x47 |

### BSF — Bit Set f

| | |
|---|---|
| Syntax: | [ *label* ] BSF f,b |
| Operands: | $0 \le f \le 127$<br>$0 \le b \le 7$ |
| Operation: | $1 \rightarrow$ (f<b>) |
| Status Affected: | None |
| Encoding: | `01` `01bb` `bfff` `ffff` |
| Description: | Bit 'b' in register 'f' is set. |
| Words: | 1 |
| Cycles: | 1 |
| Example | `BSF FLAG_REG, 7`<br>Before Instruction<br>FLAG_REG = 0x0A<br>After Instruction<br>FLAG_REG = 0x8A |

### BTFSC — Bit Test, Skip if Clear

| | |
|---|---|
| Syntax: | [ *label* ] BTFSC f,b |
| Operands: | $0 \le f \le 127$<br>$0 \le b \le 7$ |
| Operation: | skip if (f<b>) = 0 |
| Status Affected: | None |
| Encoding: | `01` `10bb` `bfff` `ffff` |
| Description: | If bit 'b' in register 'f' is '0', then the next instruction is skipped.<br>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 two-cycle instruction. |
| Words: | 1 |
| Cycles: | 1(2) |
| Example | `HERE BTFSC` FLAG,1<br>`FALSE GOTO` PROCESS_CODE<br>`TRUE` •<br>•<br>•<br>Before Instruction<br>PC = address `HERE`<br>After Instruction<br>if FLAG<1> = 0,<br>PC = address `TRUE`<br>if FLAG<1>=1,<br>PC = address `FALSE` |

### INCFSZ — Increment f, Skip if 0

| | |
|---|---|
| Syntax: | [ *label* ] INCFSZ f,d |
| Operands: | $0 \le f \le 127$<br>$d \in [0,1]$ |
| Operation: | (f) + 1 → (dest), skip if result = 0 |
| Status Affected: | None |
| Encoding: | `00` `1111` `dfff` `ffff` |
| Description: | The contents of register 'f' are incremented. If 'd' is 0 the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'.<br>If the result is 0, the next instruction, which is already fetched, is discarded. A `NOP` is executed instead making it a two-cycle instruction. |
| Words: | 1 |
| Cycles: | 1(2) |

Example

```
HERE      INCFSZ     CNT,  1
          GOTO       LOOP
CONTINUE  •
          •
          •
```

Before Instruction
PC = address `HERE`
After Instruction
CNT = CNT + 1
if CNT= 0,
PC = address `CONTINUE`
if CNT≠ 0,
PC = address `HERE +1`

### IORLW — Inclusive OR Literal with W

| | |
|---|---|
| Syntax: | [ *label* ] IORLW k |
| Operands: | $0 \le k \le 255$ |
| Operation: | (W) .OR. k → (W) |
| Status Affected: | Z |
| Encoding: | `11` `1000` `kkkk` `kkkk` |
| Description: | The contents of the W register is OR'ed with the eight bit literal 'k'. The result is placed in the W register. |
| Words: | 1 |
| Cycles: | 1 |

Example `IORLW 0x35`

Before Instruction
W = 0x9A
After Instruction
W = 0xBF
Z = 1

### IORWF — Inclusive OR W with f

| | |
|---|---|
| Syntax: | [ *label* ] IORWF f,d |
| Operands: | $0 \le f \le 127$<br>$d \in [0,1]$ |
| Operation: | (W) .OR. (f) → (dest) |
| Status Affected: | $\overline{Z}$ |
| Encoding: | `00` `0100` `dfff` `ffff` |
| Description: | Inclusive OR the W register with register 'f'. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'. |
| Words: | 1 |
| Cycles: | 1 |

Example `IORWF` RESULT, 0

Before Instruction
RESULT = 0x13
W = 0x91
After Instruction
RESULT = 0x13
W = 0x93
Z = 1

### MOVLW — Move Literal to W

| | |
|---|---|
| Syntax: | [ *label* ] MOVLW k |
| Operands: | $0 \le k \le 255$ |
| Operation: | k → (W) |
| Status Affected: | None |
| Encoding: | `11` `00xx` `kkkk` `kkkk` |
| Description: | The eight bit literal 'k' is loaded into W register. The don't cares will assemble as 0's. |
| Words: | 1 |
| Cycles: | 1 |

Example `MOVLW 0x5A`

After Instruction
W = 0x5A

## SUBLW — Subtract W from Literal

| | |
|---|---|
| Syntax: | [ *label* ] SUBLW k |
| Operands: | $0 \le k \le 255$ |
| Operation: | k - (W) → (W) |
| Status Affected: | C, DC, Z |
| Encoding: | `11` `110x` `kkkk` `kkkk` |
| Description: | The W register is subtracted (2's complement method) from the eight bit literal 'k'. The result is placed in the W register. |
| Words: | 1 |
| Cycles: | 1 |

Example 1: `SUBLW 0x02`

Before Instruction
W = 1
C = ?

After Instruction
W = 1
C = 1; result is positive

Example 2: Before Instruction
W = 2
C = ?

After Instruction
W = 0
C = 1; result is zero

Example 3: Before Instruction
W = 3
C = ?

After Instruction
W = 0xFF
C = 0; result is negative

## SUBWF — Subtract W from f

| | |
|---|---|
| Syntax: | [ *label* ] SUBWF f,d |
| Operands: | $0 \le f \le 127$<br>$d \in [0,1]$ |
| Operation: | (f) - (W) → (dest) |
| Status Affected: | C, DC, Z |
| Encoding: | `00` `0010` `dfff` `ffff` |
| Description: | Subtract (2's complement method) W register from register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'. |
| Words: | 1 |
| Cycles: | 1 |

Example 1: `SUBWF REG1,1`

Before Instruction
REG1= 3
W = 2
C = ?

After Instruction
REG1= 1
W = 2
C = 1; result is positive

Example 2: Before Instruction
REG1= 2
W = 2
C = ?

After Instruction
REG1= 0
W = 2
C = 1; result is zero

Example 3: Before Instruction
REG1= 1
W = 2
C = ?

After Instruction
REG1= 0xFF
W = 2
C = 0; result is negative

**NOTES:**

## 12.1 DC Characteristics: PIC16C62X-04 (Commercial, Industrial, Extended) PIC16C62X-20 (Commercial, Industrial, Extended) PIC16LC62X-04 (Commercial, Industrial, Extended)

| PIC16C62X | | | **Standard Operating Conditions (unless otherwise stated)**<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial and<br>0°C ≤ TA ≤ +70°C for commercial and<br>-40°C ≤ TA ≤ +125°C for extended | | | | |
|---|---|---|---|---|---|---|---|
| **PIC16LC62X** | | | **Standard Operating Conditions (unless otherwise stated)**<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial and<br>0°C ≤ TA ≤ +70°C for commercial and<br>-40°C ≤ TA ≤ +125°C for extended<br>Operating voltage VDD range is the PIC16C62X range. | | | | |
| **Param. No.** | **Sym** | **Characteristic** | **Min** | **Typ†** | **Max** | **Units** | **Conditions** |
| D001 | VDD | Supply Voltage | 3.0 | — | 6.0 | V | See Figures 12-1, 12-2, 12-3, 12-4, and 12-5 |
| D001 | VDD | Supply Voltage | 2.5 | — | 6.0 | V | See Figures 12-1, 12-2, 12-3, 12-4, and 12-5 |
| D002 | VDR | RAM Data Retention Voltage(1) | — | 1.5* | — | V | Device in SLEEP mode |
| D002 | VDR | RAM Data Retention Voltage(1) | — | 1.5* | — | V | Device in SLEEP mode |
| D003 | VPOR | VDD start voltage to ensure Power-on Reset | — | Vss | — | V | See section on Power-on Reset for details |
| D003 | VPOR | VDD start voltage to ensure Power-on Reset | — | Vss | — | V | See section on Power-on Reset for details |
| D004 | SVDD | VDD rise rate to ensure Power-on Reset | 0.05* | — | — | V/ms | See section on Power-on Reset for details |
| D004 | SVDD | VDD rise rate to ensure Power-on Reset | 0.05* | — | — | V/ms | See section on Power-on Reset for details |
| D005 | VBOR | Brown-out Detect Voltage | 3.7 | 4.0 | 4.3 | V | BOREN configuration bit is cleared |
| D005 | VBOR | Brown-out Detect Voltage | 3.7 | 4.0 | 4.3 | V | BOREN configuration bit is cleared |
| D010 | IDD | Supply Current(2) | —<br>—<br>— | 1.8<br>35<br>9.0 | 3.3<br>70<br>20 | mA<br>μA<br>mA | FOSC = 4 MHz, VDD = 5.5V, WDT disabled, XT mode, (**Note 4**)*<br>FOSC = 32 kHz, VDD = 4.0V, WDT disabled, LP mode<br>FOSC = 20 MHz, VDD = 5.5V, WDT disabled, HS mode |
| D010 | IDD | Supply Current(2) | —<br>— | 1.4<br>26 | 2.5<br>53 | mA<br>μA | FOSC = 2.0 MHz, VDD = 3.0V, WDT disabled, XT mode, (**Note 4**)<br>FOSC = 32 kHz, VDD = 3.0V, WDT disabled, LP mode |
| D020 | IPD | Power-down Current(3) | — | 1.0 | 2.5<br>15 | μA<br>μA | VDD=4.0V, WDT disabled<br>(125°C) |
| D020 | IPD | Power-down Current(3) | — | 0.7 | 2 | μA | VDD=3.0V, WDT disabled |

* 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 is the limit to which VDD can be lowered 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: Ir = VDD/2REXT (mA) with REXT in kΩ.

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

## 12.2 DC Characteristics: PIC16C62XA-04 (Commercial, Industrial, Extended) PIC16C62XA-20 (Commercial, Industrial, Extended) PIC16LC62XA-04 (Commercial, Industrial, Extended)

| PIC16C62XA | | | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial and<br>0°C ≤ TA ≤ +70°C for commercial and<br>-40°C ≤ TA ≤ +125°C for extended | | | | |
|---|---|---|---|---|---|---|---|
| **PIC16LC62XA** | | | **Standard Operating Conditions (unless otherwise stated)**<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial and<br>0°C ≤ TA ≤ +70°C for commercial and<br>-40°C ≤ TA ≤ +125°C for extended | | | | |
| **Param. No.** | **Sym** | **Characteristic** | **Min** | **Typ†** | **Max** | **Units** | **Conditions** |
| D001 | VDD | Supply Voltage | 3.0 | — | 5.5 | V | See Figures 12-1, 12-2, 12-3, 12-4, and 12-5 |
| D001 | VDD | Supply Voltage | 2.5 | — | 5.5 | V | See Figures 12-1, 12-2, 12-3, 12-4, and 12-5 |
| D002 | VDR | RAM Data Retention Voltage(1) | — | 1.5* | — | V | Device in SLEEP mode |
| D002 | VDR | RAM Data Retention Voltage(1) | — | 1.5* | — | V | Device in SLEEP mode |
| D003 | VPOR | VDD start voltage to ensure Power-on Reset | — | VSS | — | V | See section on Power-on Reset for details |
| D003 | VPOR | VDD start voltage to ensure Power-on Reset | — | VSS | — | V | See section on Power-on Reset for details |
| D004 | SVDD | VDD rise rate to ensure Power-on Reset | 0.05* | — | — | V/ms | See section on Power-on Reset for details |
| D004 | SVDD | VDD rise rate to ensure Power-on Reset | 0.05* | — | — | V/ms | See section on Power-on Reset for details |
| D005 | VBOR | Brown-out Detect Voltage | 3.7 | 4.0 | 4.35 | V | BOREN configuration bit is cleared |
| D005 | VBOR | Brown-out Detect Voltage | 3.7 | 4.0 | 4.35 | V | BOREN configuration bit is cleared |

* 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 is the limit to which VDD can be lowered without losing RAM data.

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

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

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

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

**6:** Commercial temperature range only.

## 12.4 DC Characteristics: PIC16C62X/C62XA/CR62XA (Commercial, Industrial, Extended) PIC16LC62X/LC62XA/LCR62XA (Commercial, Industrial, Extended)

| PIC16C62X/C62XA/CR62XA | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial and<br>0°C ≤ TA ≤ +70°C for commercial and<br>-40°C ≤ TA ≤ +125°C for extended |
|---|---|
| PIC16LC62X/LC62XA/LCR62XA | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial and<br>0°C ≤ TA ≤ +70°C for commercial and<br>-40°C ≤ TA ≤ +125°C for extended |

| Param. No. | Sym | Characteristic | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| | VIL | **Input Low Voltage** | | | | | |
| | | I/O ports | | | | | |
| D030 | | with TTL buffer | Vss | — | 0.8V<br>0.15 VDD | V | VDD = 4.5V to 5.5V<br>otherwise |
| D031 | | with Schmitt Trigger input | Vss | — | 0.2 VDD | V | |
| D032 | | $\overline{\text{MCLR}}$, RA4/T0CKI,OSC1 (in RC mode) | Vss | — | 0.2 VDD | V | **(Note 1)** |
| D033 | | OSC1 (in XT and HS) | Vss | — | 0.3 VDD | V | |
| | | OSC1 (in LP) | Vss | — | 0.6 VDD-1.0 | V | |
| | VIL | **Input Low Voltage** | | | | | |
| | | I/O ports | | | | | |
| D030 | | with TTL buffer | Vss | — | 0.8V<br>0.15 VDD | V | VDD = 4.5V to 5.5V<br>otherwise |
| D031 | | with Schmitt Trigger input | Vss | — | 0.2 VDD | V | |
| D032 | | $\overline{\text{MCLR}}$, RA4/T0CKI,OSC1 (in RC mode) | Vss | — | 0.2 VDD | V | **(Note 1)** |
| D033 | | OSC1 (in XT and HS) | Vss | — | 0.3 VDD | V | |
| | | OSC1 (in LP) | Vss | — | 0.6 VDD-1.0 | V | |
| | VIH | Input High Voltage | | | | | |
| | | I/O ports | | | | | |
| D040 | | with TTL buffer | 2.0V<br>0.25 VDD + 0.8V | — | VDD<br>VDD | V | VDD = 4.5V to 5.5V<br>otherwise |
| D041 | | with Schmitt Trigger input | 0.8 VDD | — | VDD | | |
| D042 | | $\overline{\text{MCLR}}$ RA4/T0CKI | 0.8 VDD | — | VDD | V | |
| D043<br>D043A | | OSC1 (XT, HS and LP)<br>OSC1 (in RC mode) | 0.7 VDD<br>0.9 VDD | — | VDD | V | **(Note 1)** |

* 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:** In RC oscillator configuration, the OSC1 pin is a Schmitt Trigger input. It is not recommended that the PIC16C62X(A) be driven with external clock in RC mode.

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

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

## 12.4 DC Characteristics: PIC16C62X/C62XA/CR62XA (Commercial, Industrial, Extended) PIC16LC62X/LC62XA/LCR62XA (Commercial, Industrial, Extended) (CONT.)

| PIC16C62X/C62XA/CR62XA | | | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial and<br>0°C ≤ TA ≤ +70°C for commercial and<br>-40°C ≤ TA ≤ +125°C for extended | | | | |
|---|---|---|---|---|---|---|---|
| **PIC16LC62X/LC62XA/LCR62XA** | | | **Standard Operating Conditions (unless otherwise stated)**<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial and<br>0°C ≤ TA ≤ +70°C for commercial and<br>-40°C ≤ TA ≤ +125°C for extended | | | | |
| **Param. No.** | **Sym** | **Characteristic** | **Min** | **Typ†** | **Max** | **Units** | **Conditions** |
| | $V_{OL}$ | Output Low Voltage | | | | | |
| D080 | | I/O ports | — | — | 0.6 | V | $I_{OL}$ = 8.5 mA, $V_{DD}$ = 4.5V, -40° to +85°C |
| | | | — | — | 0.6 | V | $I_{OL}$ = 7.0 mA, $V_{DD}$ = 4.5V, +125°C |
| D083 | | OSC2/CLKOUT (RC only) | — | — | 0.6 | V | $I_{OL}$ = 1.6 mA, $V_{DD}$ = 4.5V, -40° to +85°C |
| | | | — | — | 0.6 | V | $I_{OL}$ = 1.2 mA, $V_{DD}$ = 4.5V, +125°C |
| | $V_{OH}$ | Output High Voltage(3) | | | | | |
| D090 | | I/O ports (Except RA4) | $V_{DD}$-0.7 | — | — | V | $I_{OH}$ = -3.0 mA, $V_{DD}$ = 4.5V, -40° to +85°C |
| | | | $V_{DD}$-0.7 | — | — | V | $I_{OH}$ = -2.5 mA, $V_{DD}$ = 4.5V, +125°C |
| D092 | | OSC2/CLKOUT (RC only) | $V_{DD}$-0.7 | — | — | V | $I_{OH}$ = -1.3 mA, $V_{DD}$ = 4.5V, -40° to +85°C |
| | | | $V_{DD}$-0.7 | — | — | V | $I_{OH}$ = -1.0 mA, $V_{DD}$ = 4.5V, +125°C |
| | $V_{OH}$ | Output High Voltage(3) | | | | | |
| D090 | | I/O ports (Except RA4) | $V_{DD}$-0.7 | — | — | V | $I_{OH}$ = -3.0 mA, $V_{DD}$ = 4.5V, -40° to +85°C |
| | | | $V_{DD}$-0.7 | — | — | V | $I_{OH}$ = -2.5 mA, $V_{DD}$ = 4.5V, +125°C |
| D092 | | OSC2/CLKOUT (RC only) | $V_{DD}$-0.7 | — | — | V | $I_{OH}$ = -1.3 mA, $V_{DD}$ = 4.5V, -40° to +85°C |
| | | | $V_{DD}$-0.7 | — | — | V | $I_{OH}$ = -1.0 mA, $V_{DD}$ = 4.5V, +125°C |
| *D150 | $V_{OD}$ | Open-Drain High Voltage | | | 10*<br>8.5* | V | RA4 pin PIC16C62X, PIC16LC62X<br>RA4 pin PIC16C62XA, PIC16LC62XA, PIC16CR62XA, PIC16LCR62XA |
| *D150 | $V_{OD}$ | Open-Drain High Voltage | | | 10*<br>8.5* | V | RA4 pin PIC16C62X, PIC16LC62X<br>RA4 pin PIC16C62XA, PIC16LC62XA, PIC16CR62XA, PIC16LCR62XA |
| | | Capacitive Loading Specs on Output Pins | | | | | |
| D100 | COSC2 | OSC2 pin | | | 15 | pF | In XT, HS and LP modes when external clock used to drive OSC1. |
| D101 | $C_{IO}$ | All I/O pins/OSC2 (in RC mode) | | | 50 | pF | |
| | | Capacitive Loading Specs on Output Pins | | | | | |
| D100 | COSC2 | OSC2 pin | | | 15 | pF | In XT, HS and LP modes when external clock used to drive OSC1. |
| D101 | $C_{IO}$ | All I/O pins/OSC2 (in RC mode) | | | 50 | pF | |

* 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:** In RC oscillator configuration, the OSC1 pin is a Schmitt Trigger input. It is not recommended that the PIC16C62X(A) be driven with external clock in RC mode.

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

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

**FIGURE 13-5: IOH VS. VOH, VDD = 3.0V)**

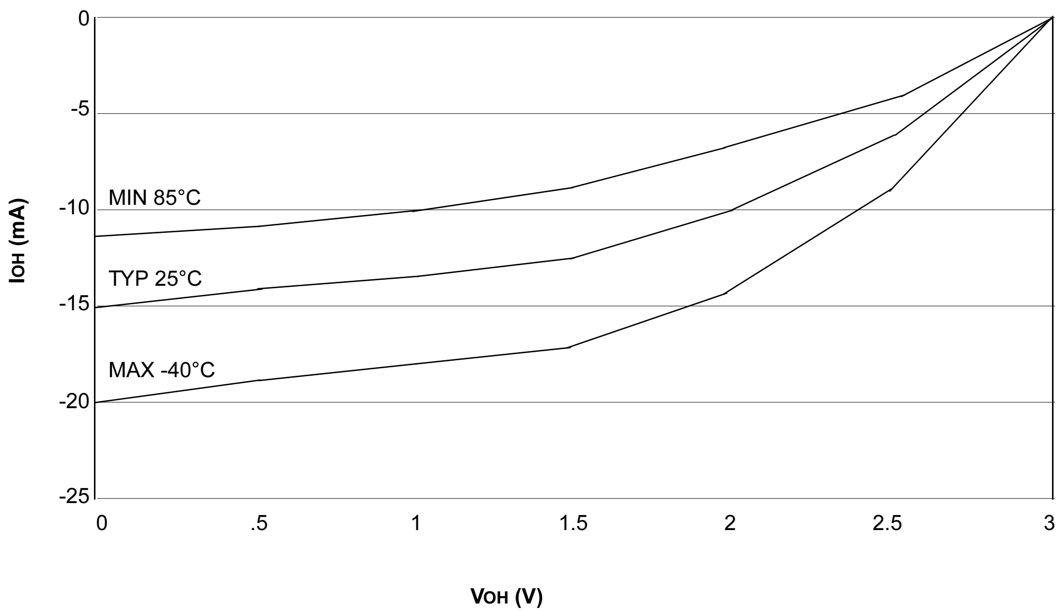
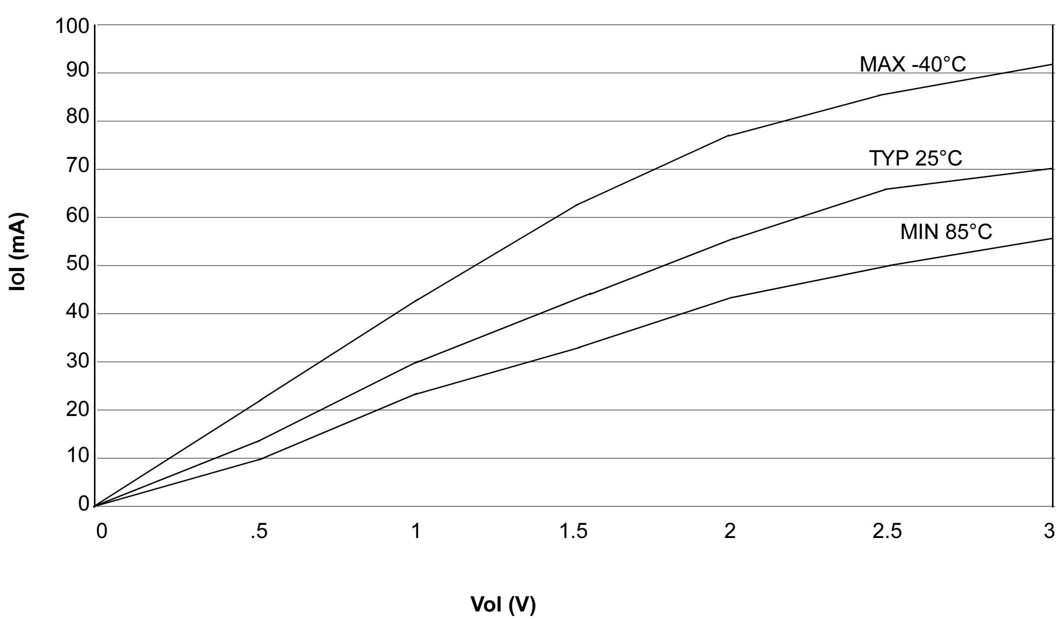
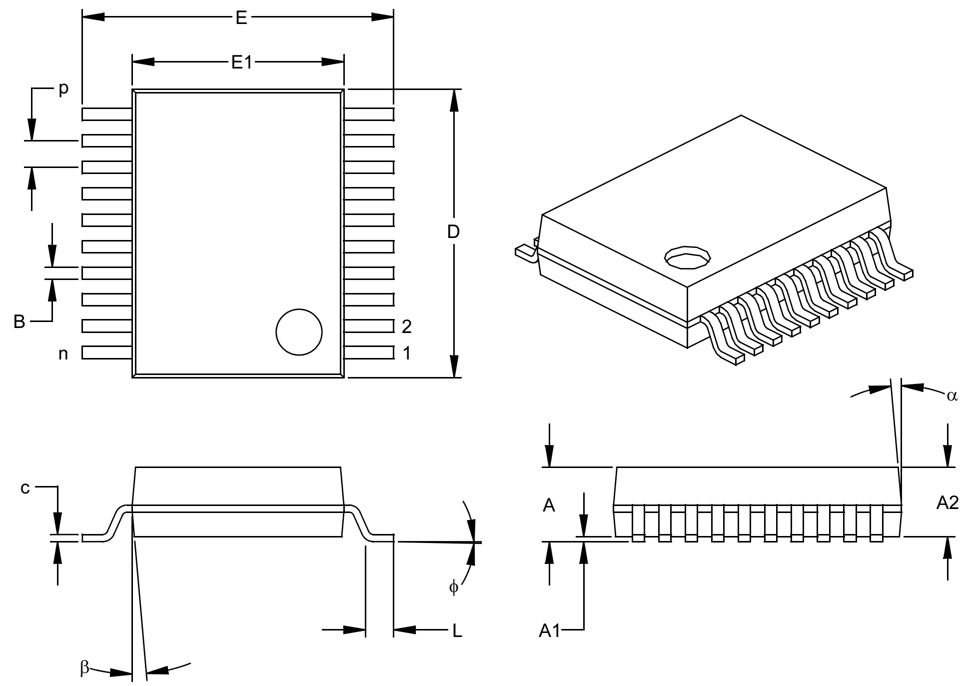



**FIGURE 13-6: IOl VS. VOL, VDD = 5.5V)**



## 20-Lead Plastic Shrink Small Outline (SS) – 209 mil, 5.30 mm (SSOP)




| Units | | INCHES* | | | MILLIMETERS | | |
|---|---|---|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX | MIN | NOM | MAX |
| Number of Pins | n | | 20 | | | 20 | |
| Pitch | p | | .026 | | | 0.65 | |
| Overall Height | A | .068 | .073 | .078 | 1.73 | 1.85 | 1.98 |
| Molded Package Thickness | A2 | .064 | .068 | .072 | 1.63 | 1.73 | 1.83 |
| Standoff § | A1 | .002 | .006 | .010 | 0.05 | 0.15 | 0.25 |
| Overall Width | E | .299 | .309 | .322 | 7.59 | 7.85 | 8.18 |
| Molded Package Width | E1 | .201 | .207 | .212 | 5.11 | 5.25 | 5.38 |
| Overall Length | D | .278 | .284 | .289 | 7.06 | 7.20 | 7.34 |
| Foot Length | L | .022 | .030 | .037 | 0.56 | 0.75 | 0.94 |
| Lead Thickness | c | .004 | .007 | .010 | 0.10 | 0.18 | 0.25 |
| Foot Angle | φ | 0 | 4 | 8 | 0.00 | 101.60 | 203.20 |
| Lead Width | B | .010 | .013 | .015 | 0.25 | 0.32 | 0.38 |
| Mold Draft Angle Top | α | 0 | 5 | 10 | 0 | 5 | 10 |
| Mold Draft Angle Bottom | β | 0 | 5 | 10 | 0 | 5 | 10 |

* Controlling Parameter
§ Significant Characteristic

Notes:
Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010” (0.254mm) per side.
JEDEC Equivalent: MO-150
Drawing No. C04-072

## PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

| PART NO. | -XX | X | /XX | XXX |
|---|---|---|---|---|
| Device | Frequency Range | Temperature Range | Package | Pattern |

| | |
|---|---|
| Device | PIC16C62X: VDD range 3.0V to 6.0V<br>PIC16C62XT: VDD range 3.0V to 6.0V (Tape and Reel)<br>PIC16C62XA: VDD range 3.0V to 5.5V<br>PIC16C62XAT: VDD range 3.0V to 5.5V (Tape and Reel)<br>PIC16LC62X: VDD range 2.5V to 6.0V<br>PIC16LC62XT: VDD range 2.5V to 6.0V (Tape and Reel)<br>PIC16LC62XA: VDD range 2.5V to 5.5V<br>PIC16LC62XAT: VDD range 2.5V to 5.5V (Tape and Reel)<br>PIC16CR620A: VDD range 2.5V to 5.5V<br>PIC16CR620AT: VDD range 2.5V to 5.5V (Tape and Reel)<br>PIC16LCR620A: VDD range 2.0V to 5.5V<br>PIC16LCR620AT: VDD range 2.0V to 5.5V (Tape and Reel) |
| Frequency Range | 04 200 kHz (LP osc)<br>04 4 MHz (XT and RC osc)<br>20 20 MHz (HS osc) |
| Temperature Range | - = 0°C to +70°C<br>I = -40°C to +85°C<br>E = -40°C to +125°C |
| Package | P = PDIP<br>SO = SOIC (Gull Wing, 300 mil body)<br>SS = SSOP (209 mil)<br>JW* = Windowed CERDIP |
| Pattern | 3-Digit Pattern Code for QTP (blank otherwise) |

**Examples:**

a) PIC16C621A - 04/P 301 = Commercial temp., PDIP package, 4 MHz, normal VDD limits, QTP pattern #301.

b) PIC16LC622- 04I/SO = Industrial temp., SOIC package, 200 kHz, extended VDD limits.

* JW Devices are UV erasable and can be programmed to any device configuration. JW Devices meet the electrical requirement of each oscillator type.

## Sales and Support

**Data Sheets**

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

1. Your local Microchip sales office
2. The Microchip Corporate Literature Center U.S. FAX: (480) 792-7277
3. The Microchip Worldwide Site (www.microchip.com)

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

**New Customer Notification System**

Register on our web site (www.microchip.com/cn) to receive the most current information on our products.