**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 | 4MHz |
| Connectivity | UART/USART |
| Peripherals | Brown-out Detect/Reset, POR, PWM, WDT |
| Number of I/O | 16 |
| Program Memory Size | 1.75KB (1K x 14) |
| Program Memory Type | FLASH |
| EEPROM Size | 128 x 8 |
| RAM Size | 224 x 8 |
| Voltage - Supply (Vcc/Vdd) | 2V ~ 5.5V |
| Data Converters | - |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°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/pic16lf627-04i-ss |

## 2.1 Clocking Scheme/Instruction Cycle

The clock input (OSC1/CLKIN/RA7 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 2-2.

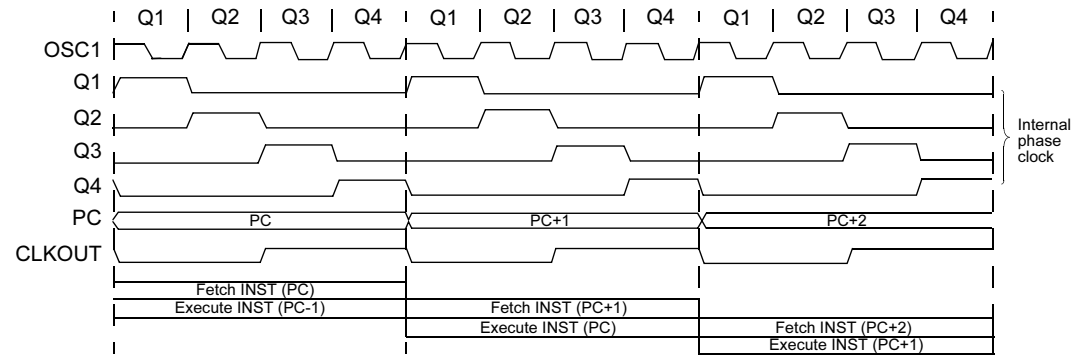
## 2.2 Instruction Flow/Pipelining

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 2-1).

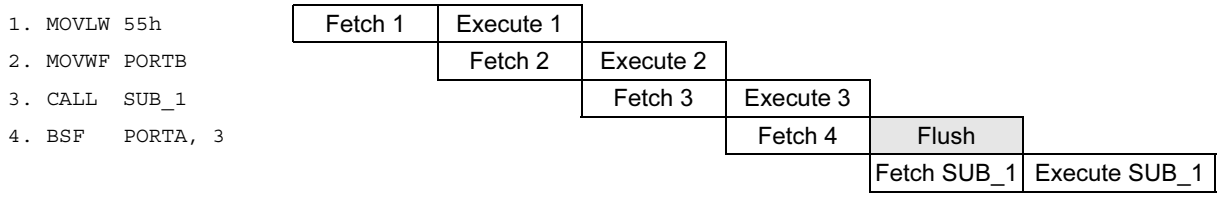
A fetch cycle begins with the program counter (PC) incrementing in Q1.

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 2-2: CLOCK/INSTRUCTION CYCLE**




**EXAMPLE 2-1: INSTRUCTION PIPELINE FLOW**



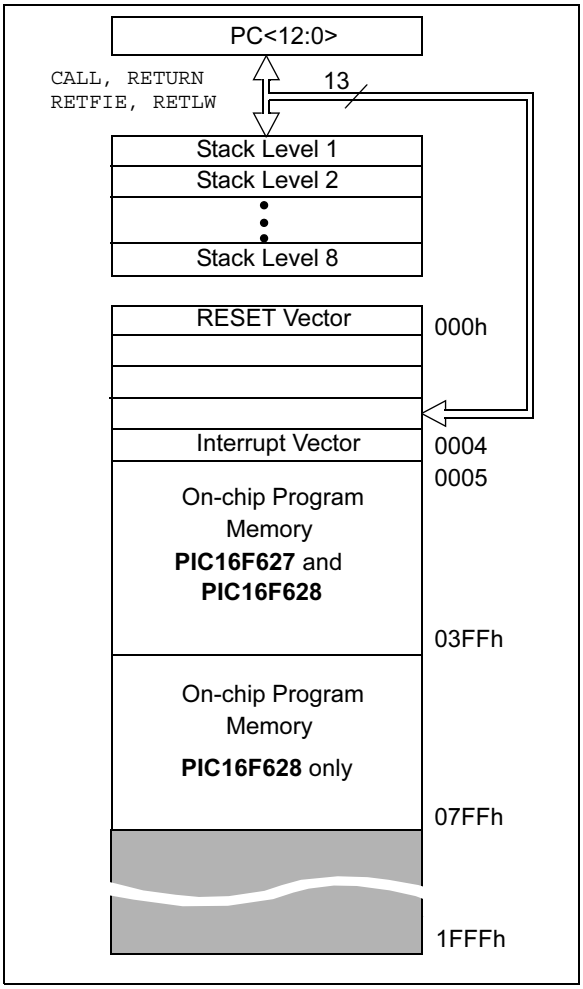

All instructions are single cycle, except for any program branches. These take two cycles since the fetch instruction is "flushed" from the pipeline while the new instruction is being fetched and then executed.

# 3.0 MEMORY ORGANIZATION

## 3.1 Program Memory Organization

The PIC16F62X has a 13-bit program counter capable of addressing an 8K x 14 program memory space. Only the first 1K x 14 (0000h - 03FFh) for the PIC16F627 and 2K x 14 (0000h - 07FFh) for the PIC16F628 are physically implemented. Accessing a location above these boundaries will cause a wrap-around within the first 1K x 14 space (PIC16F627) or 2K x 14 space (PIC16F628). The RESET vector is at 0000h and the interrupt vector is at 0004h (Figure 3-1).

**FIGURE 3-1: PROGRAM MEMORY MAP AND STACK**




## 3.2 Data Memory Organization

The data memory (Figure 3-2) is partitioned into four banks, which contain the general purpose registers and the Special Function Registers (SFR). The SFR's are located in the first 32 locations of each Bank. Register locations 20-7Fh, A0h-FFh, 120h-14Fh, 170h-17Fh and 1F0h-1FFh are general purpose registers implemented as static RAM.

The Table below lists how to access the four banks of registers:

| | RP1 | RP0 |
|---|---|---|
| Bank0 | 0 | 0 |
| Bank1 | 0 | 1 |
| Bank2 | 1 | 0 |
| Bank3 | 1 | 1 |

Addresses F0h-FFh, 170h-17Fh and 1F0h-1FFh are implemented as common RAM and mapped back to addresses 70h-7Fh.

### 3.2.1 GENERAL PURPOSE REGISTER FILE

The register file is organized as 224 x 8 in the PIC16F62X. Each is accessed either directly or indirectly through the File Select Register FSR (See Section 3.4).

### 3.2.2.1 STATUS Register

The STATUS register, shown in Register 3-1, contains the arithmetic status of the ALU, the RESET status and the bank select bits for data memory (SRAM).

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

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

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

**Note 1:** The C and DC bits operate as a $\overline{Borrow}$ and $\overline{Digit\ Borrow}$ out bit, respectively, in subtraction. See the `SUBLW` and `SUBWF` instructions for examples.

**REGISTER 3-1: STATUS REGISTER (ADDRESS: 03h, 83h, 103h, 183h)**

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

bit 6-5 **RP1:RP0**: Register Bank Select bits (used for direct addressing)
`00` = Bank 0 (00h - 7Fh)
`01` = Bank 1 (80h - FFh)
`10` = Bank 2 (100h - 17Fh)
`11` = Bank 3 (180h - 1FFh)

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

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

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

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

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

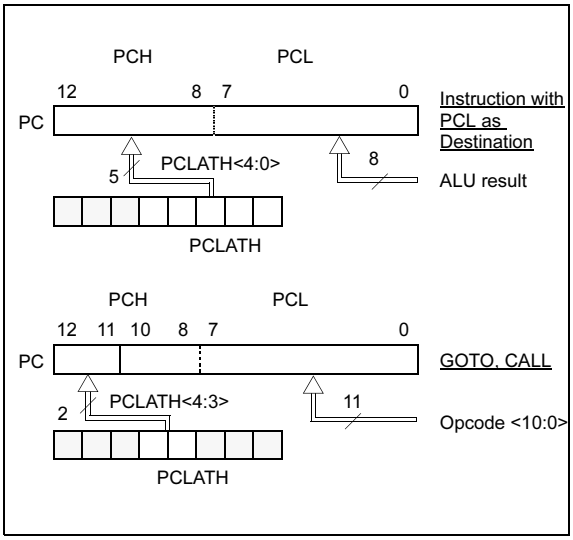
**Note 1:** For $\overline{borrow}$ the polarity is reversed. A subtraction is executed by adding the two's complement of the second operand. For rotate (`RRF`, `RLF`) instructions, this bit is loaded with either the high or low order bit of the source register.

| 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 |

## 3.3 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 3-3 shows the two situations for the loading of the PC. The upper example in the figure shows how the PC is loaded on a write to PCL (PCLATH<4:0> → PCH). The lower example in the figure shows how the PC is loaded during a CALL or GOTO instruction (PCLATH<4:3> → PCH).

**FIGURE 3-3: LOADING OF PC IN DIFFERENT SITUATIONS**




### 3.3.1 COMPUTED GOTO

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

### 3.3.2 STACK

The PIC16F62X family has an 8-level deep x 13-bit wide hardware stack (Figure 3-1 and Figure 3-2). The stack space is not part of either program or data space and the stack pointer is not readable or writable. The PC is PUSHed onto the stack when a CALL instruction is executed or an interrupt causes a branch. The stack is POPed in the event of a RETURN, RETLW or a RETFIE instruction execution. PCLATH is not affected by a PUSH or POP operation.

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

**Note 1:** There are no STATUS bits to indicate stack overflow or stack underflow conditions.

**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.

## 3.4 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 3-4.

A simple program to clear RAM location 20h-2Fh using indirect addressing is shown in Example 3-1.

**EXAMPLE 3-1: 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 5-1: BLOCK DIAGRAM OF RA0/AN0:RA1/AN1 PINS**

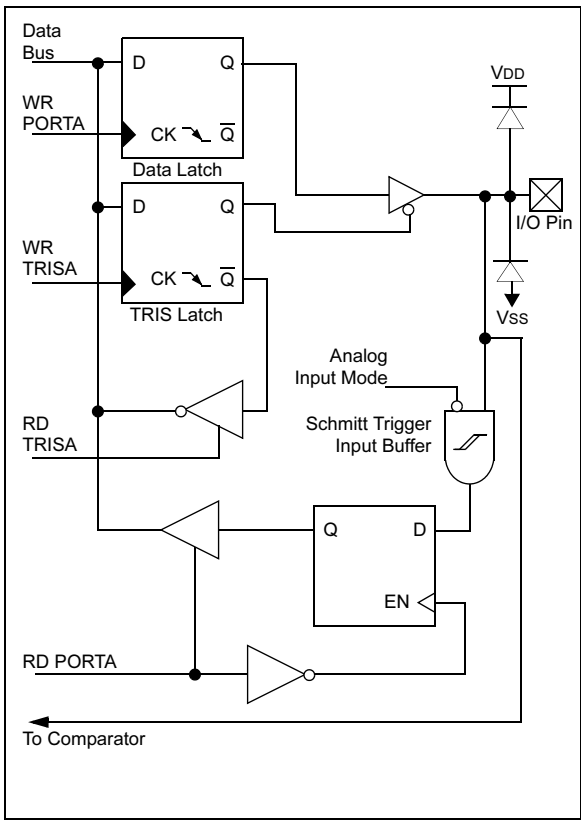



**FIGURE 5-2: BLOCK DIAGRAM OF RA2/VREF PIN**

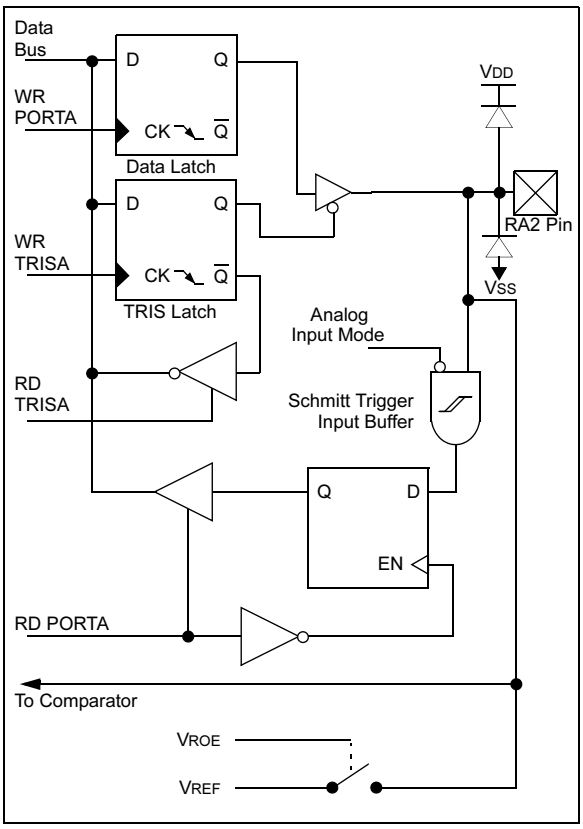
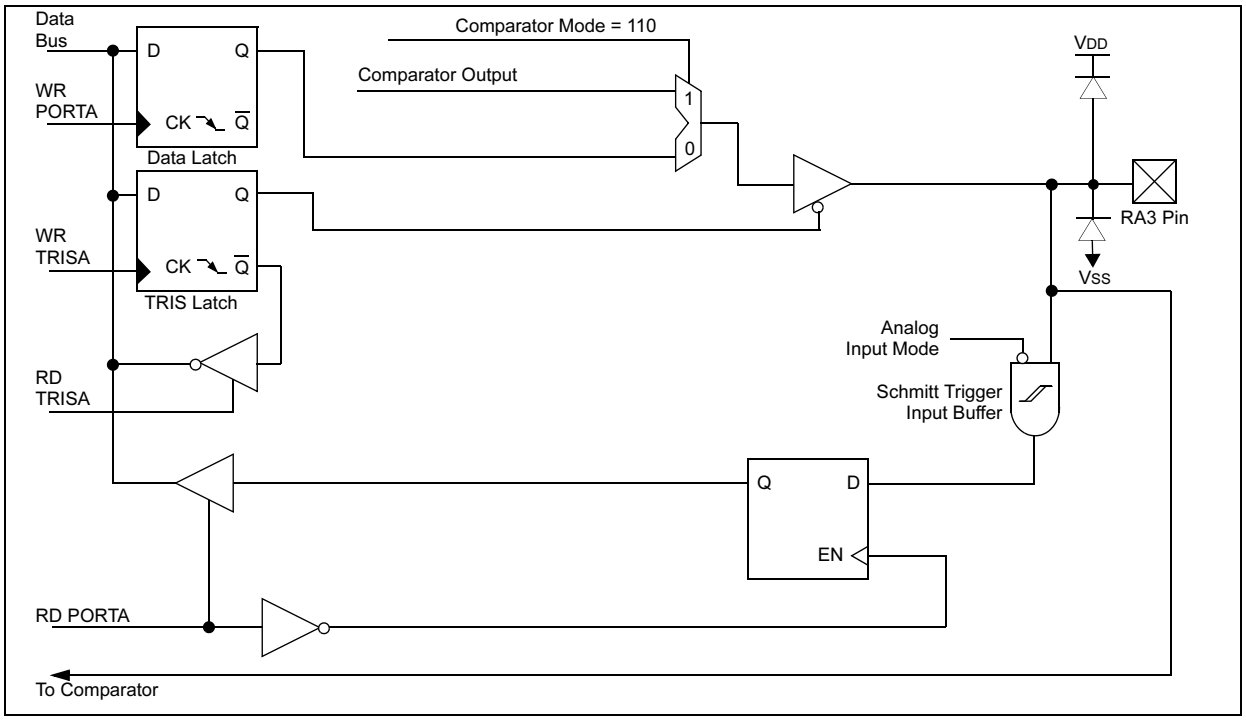
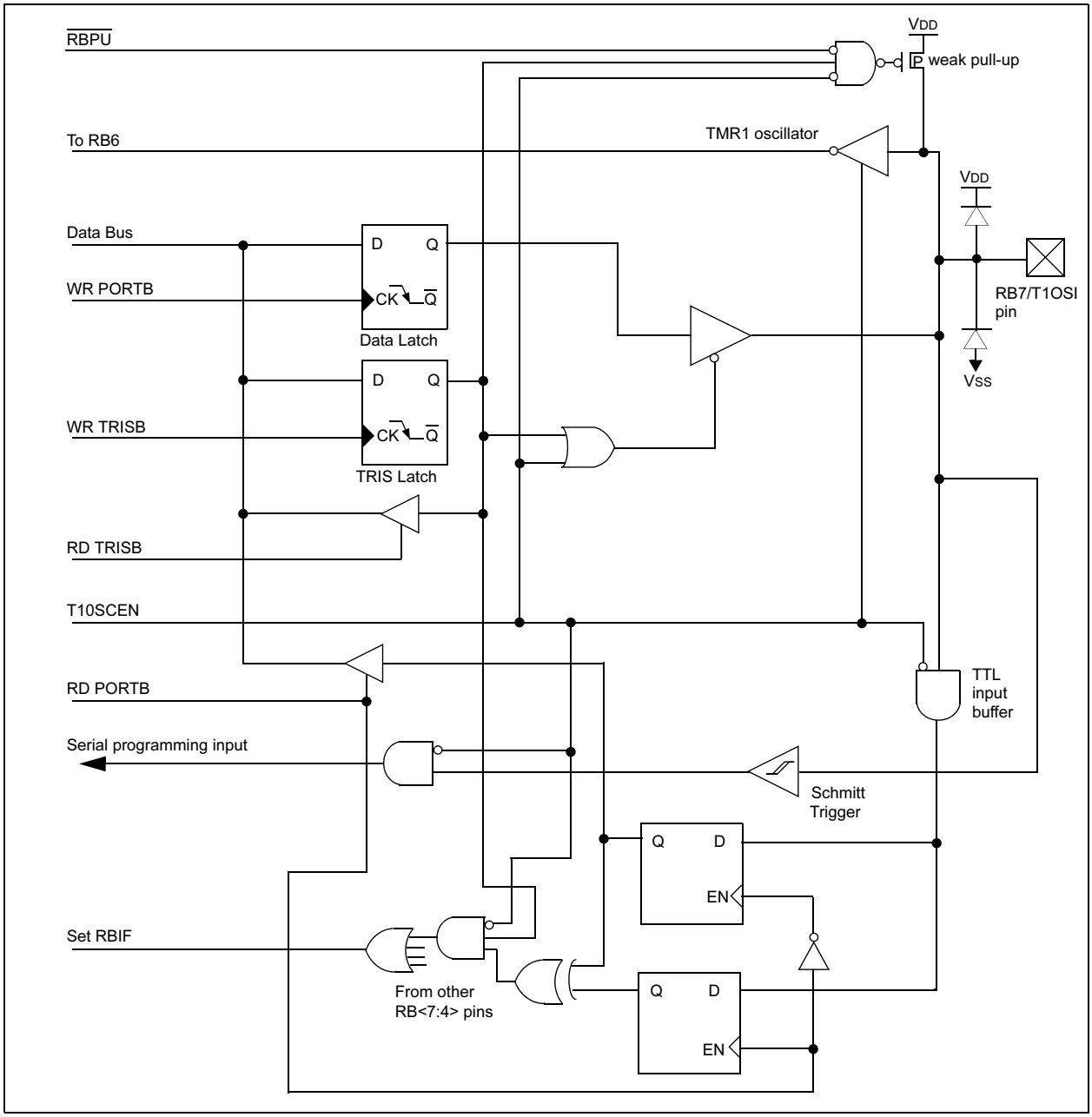



**FIGURE 5-3: BLOCK DIAGRAM OF THE RA3/AN3 PIN**




 © 2003 Microchip Technology Inc.

**FIGURE 5-15: BLOCK DIAGRAM OF THE RB7/T1OSI PIN**




 © 2003 Microchip Technology Inc.

## 9.0 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 10.0) can also be an input to the comparators.

The CMCON register, shown in Register 9-1, controls the comparator input and output multiplexers. A block diagram of the comparator is shown in Figure 9-1.

**REGISTER 9-1: CMCON 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
When C2INV = 0:
1 = C2 $V_{IN}$+ > C2 $V_{IN}$-
0 = C2 $V_{IN}$+ < C2 $V_{IN}$-

When C2INV = 1:
1 = C2 $V_{IN}$+ < C2 $V_{IN}$-
0 = C2 $V_{IN}$+ > C2 $V_{IN}$-

bit 6 **C1OUT**: Comparator 1 Output
When C1INV = 0:
1 = C1 $V_{IN}$+ > C1 $V_{IN}$-
0 = C1 $V_{IN}$+ < C1 $V_{IN}$-

When C1INV = 1:
1 = C1 $V_{IN}$+ < C1 $V_{IN}$-
0 = C1 $V_{IN}$+ > C1 $V_{IN}$-

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

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

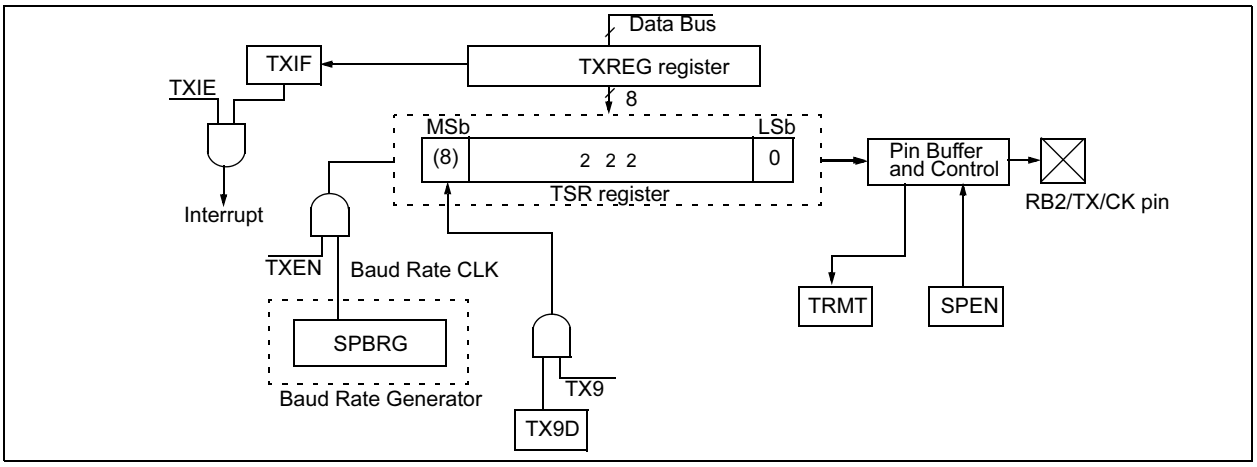
bit 3 **CIS**: Comparator Input Switch
When CM2:CM0: = 001
Then:
1 = C1 $V_{IN}$- connects to RA3
0 = C1 $V_{IN}$- connects to RA0

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

bit 2-0 **CM2:CM0**: Comparator Mode
Figure 9-1 shows the Comparator modes and CM2:CM0 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 |

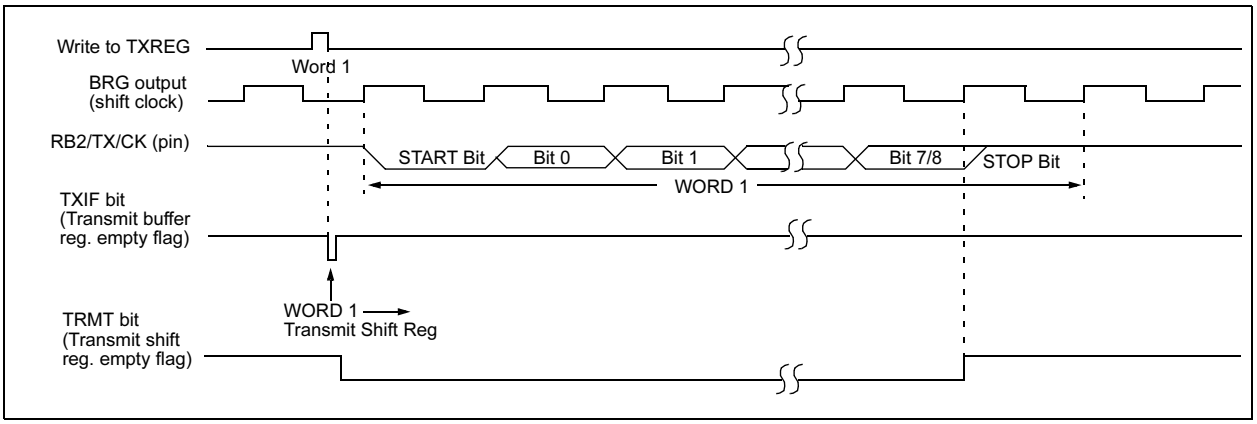
**FIGURE 12-5: USART TRANSMIT BLOCK DIAGRAM**




Steps to follow when setting up an Asynchronous Transmission:

1. Initialize the SPBRG register for the appropriate baud rate. If a high speed baud rate is desired, set bit BRGH. (Section 12.1)
2. Enable the asynchronous serial port by clearing bit SYNC and setting bit SPEN.
3. If interrupts are desired, then set enable bit TXIE.
4. If 9-bit transmission is desired, then set transmit bit TX9.
5. Enable the transmission by setting bit TXEN, which will also set bit TXIF.
6. If 9-bit transmission is selected, the ninth bit should be loaded in bit TX9D.
7. Load data to the TXREG register (starts transmission).

**FIGURE 12-6: ASYNCHRONOUS TRANSMISSION**



**NOTES:**

 © 2003 Microchip Technology Inc.

## 13.3 READING THE EEPROM DATA MEMORY

To read a data memory location, the user must write the address to the EEADR register and then set control bit RD (EECON1<0>). The data is available, in the very next cycle, in the EEDATA register; therefore it can be read in the next instruction. EEDATA will hold this value until another read or until it is written to by the user (during a write operation).

### EXAMPLE 13-1: DATA EEPROM READ

```
BSF     STATUS, RP0  ; Bank 1
MOVLW   CONFIG_ADDR  ;
MOVWF   EEADR        ; Address to read
BSF     EECON1, RD   ; EE Read
MOVF    EEDATA, W    ; W = EEDATA
BCF     STATUS, RP0  ; Bank 0
```

## 13.4 WRITING TO THE EEPROM DATA MEMORY

To write an EEPROM data location, the user must first write the address to the EEADR register and the data to the EEDATA register. Then the user must follow a specific sequence to initiate the write for each byte.

### EXAMPLE 13-2: DATA EEPROM WRITE

Required Sequence

```
BSF     STATUS, RP0    ; Bank 1
BSF     EECON1, WREN   ; Enable write
BCF     INTCON, GIE    ; Disable INTs.
MOVLW   55h            ;
MOVWF   EECON2         ; Write 55h
MOVLW   AAh            ;
MOVWF   EECON2         ; Write AAh
BSF     EECON1,WR      ; Set WR bit
                       ; begin write
BSF     INTCON, GIE    ; Enable INTs.
```

The write will not initiate if the above sequence is not exactly followed (write 55h to EECON2, write AAh to EECON2, then set WR bit) for each byte. We strongly recommend that interrupts be disabled during this code segment. A cycle count is executed during the required sequence. Any number that is not equal to the required cycles to execute the required sequence will cause the data not to be written into the EEPROM.

Additionally, the WREN bit in EECON1 must be set to enable write. This mechanism prevents accidental writes to data EEPROM due to errant (unexpected) code execution (i.e., lost programs). The user should keep the WREN bit clear at all times, except when updating EEPROM. The WREN bit is not cleared by hardware.

After a write sequence has been initiated, clearing the WREN bit will not affect this write cycle. The WR bit will be inhibited from being set unless the WREN bit is set.

At the completion of the write cycle, the WR bit is cleared in hardware and the EE Write Complete Interrupt Flag bit (EEIF) is set. The user can either enable this interrupt or poll this bit. The EEIF bit in the PIR1 registers must be cleared by software.

## 13.5 WRITE VERIFY

Depending on the application, good programming practice may dictate that the value written to the Data EEPROM should be verified (Example 13-3) to the desired value to be written. This should be used in applications where an EEPROM bit will be stressed near the specification limit.

### EXAMPLE 13-3: WRITE VERIFY

```
    BSF   STATUS, RP0  ; Bank 1
    MOVF  EEDATA, W
    BSF   EECON1, RD   ; Read the
                       ; value written
;
; Is the value written (in W reg) and
; read (in EEDATA) the same?
;
    SUBWF EEDATA, W    ;
    BCF   STATUS, RP0  ; Bank0
    BTFSS STATUS, Z    ; Is difference 0?
    GOTO  WRITE_ERR    ; NO, Write error
    :                  ; YES, Good write
    :                  ; Continue program
```

## 13.6 PROTECTION AGAINST SPURIOUS WRITE

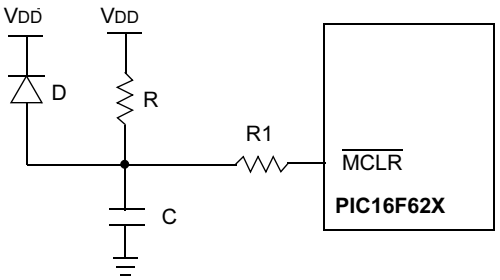
There are conditions when the device may not want to write to the data EEPROM memory. To protect against spurious EEPROM writes, various mechanisms have been built in. On power-up, WREN is cleared. Also, the Power-up Timer (72 ms duration) prevents EEPROM write.

The write initiate sequence, and the WREN bit together help prevent an accidental write during brown-out, power glitch, or software malfunction.

## 13.7 DATA EEPROM OPERATION DURING CODE PROTECT

When the device is code protected, the CPU is able to read and write unscrambled data to the Data EEPROM.

**FIGURE 14-11: EXTERNAL POWER-ON RESET CIRCUIT (FOR SLOW VDD POWER-UP)**

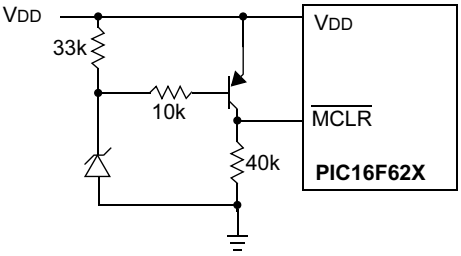



**Note 1:** External Power-on Reset circuit is required only if VDD power-up slope is too slow. The diode D helps discharge the capacitor quickly when VDD powers down.

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

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

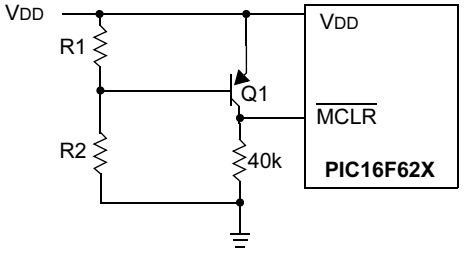
**FIGURE 14-12: EXTERNAL BROWN-OUT PROTECTION CIRCUIT 1**




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

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

**FIGURE 14-13: EXTERNAL BROWN-OUT PROTECTION CIRCUIT 2**




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

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

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

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

 © 2003 Microchip Technology Inc.

### 14.6.1 RB0/INT INTERRUPT

External interrupt on RB0/INT pin is edge triggered: either rising if INTEDG bit (OPTION<6>) is set, or falling, if INTEDG bit is clear. When a valid edge appears on the RB0/INT pin, the INTF bit (INTCON<1>) is set. This interrupt can be disabled by clearing the INTE control bit (INTCON<4>). The INTF bit must be cleared in software in the interrupt service routine before re-enabling this interrupt. The RB0/INT interrupt can wake-up the processor from SLEEP, if the INTE bit was set prior to going into SLEEP. The status of the GIE bit decides whether or not the processor branches to the interrupt vector following wake-up. See Section 14.9 for details on SLEEP, and Figure 14-17 for timing of wake-up from SLEEP through RB0/INT interrupt.

### 14.6.2 TMR0 INTERRUPT

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

### 14.6.3 PORTB INTERRUPT

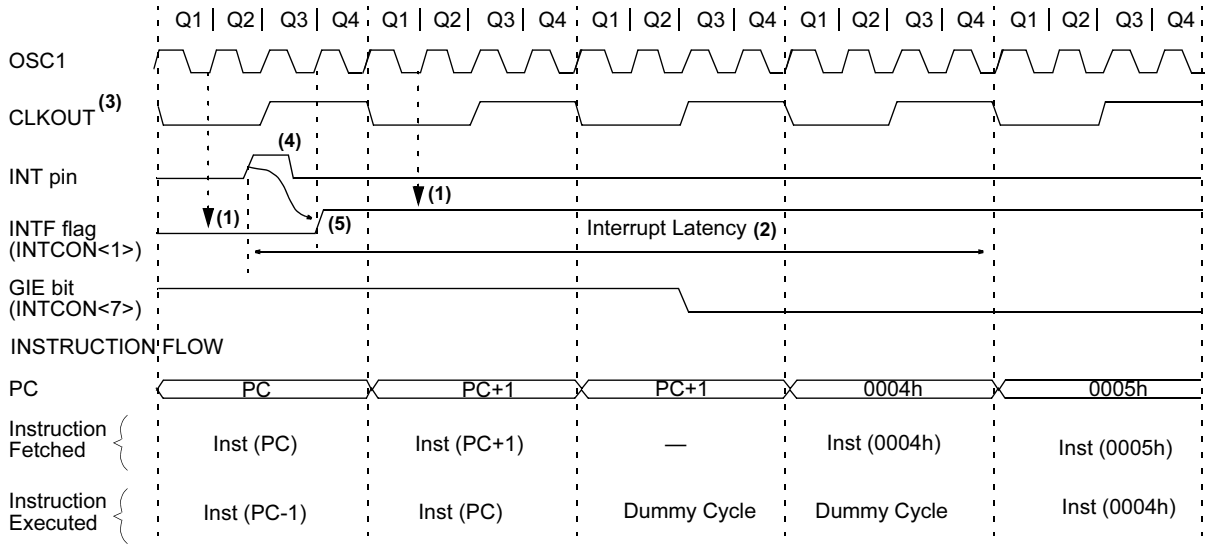
An input change on PORTB <7:4> sets the RBIF (INTCON<0>) bit. The interrupt can be enabled/disabled by setting/clearing the RBIE (INTCON<4>) bit. For operation of PORTB (Section 5.2).

| **Note:** | If a change on the I/O pin should occur when the read operation is being executed (start of the Q2 cycle), then the RBIF interrupt flag may not get set. |
|---|---|

### 14.6.4 COMPARATOR INTERRUPT

See Section 9.6 for complete description of comparator interrupts.

**FIGURE 14-15: INT PIN INTERRUPT TIMING**




**Note 1:** INTF flag is sampled here (every Q1).
**2:** Asynchronous interrupt latency = 3-4 Tcy. Synchronous latency = 3 Tcy, where Tcy = instruction cycle time. Latency is the same whether Inst (PC) is a single cycle or a 2-cycle instruction.
**3:** CLKOUT is available in ER and INTRC Oscillator mode.
**4:** For minimum width of INT pulse, refer to AC specs.
**5:** INTF is enabled to be set anytime during the Q4-Q1 cycles.

 © 2003 Microchip Technology Inc.

**RRF** **Rotate Right f through Carry**

Syntax: [ *label* ] RRF f,d

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

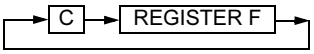
Operation: See description below

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

**SLEEP**

Syntax: [ *label* ] SLEEP

Operands: None

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

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

Encoding:

| | | | |
|---|---|---|---|
| 00 | 0000 | 0110 | 0011 |

Description: The power-down STATUS bit, $\overline{\text{PD}}$ is cleared. Timeout STATUS bit, $\overline{\text{TO}}$ is set. Watchdog Timer and its prescaler are cleared.
The processor is put into SLEEP mode with the oscillator stopped. See Section 14.9 for more details.

Words: 1

Cycles: 1

Example:
```
SLEEP
```

**SUBLW** **Subtract W from Literal**

Syntax: [ *label* ] SUBLW k

Operands: $0 \le k \le 255$

Operation: k - (W) → (W)

Status Affected: C, DC, Z

Encoding:

| | | | |
|---|---|---|---|
| 11 | 110x | kkkk | kkkk |

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

Words: 1

Cycles: 1

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

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

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

 © 2003 Microchip Technology Inc.

.

**Note:** The graphs and tables provided in this section are for design guidance and are not tested.

**FIGURE 18-2: MAXIMUM IDD vs FOSC OVER VDD (HS MODE)**

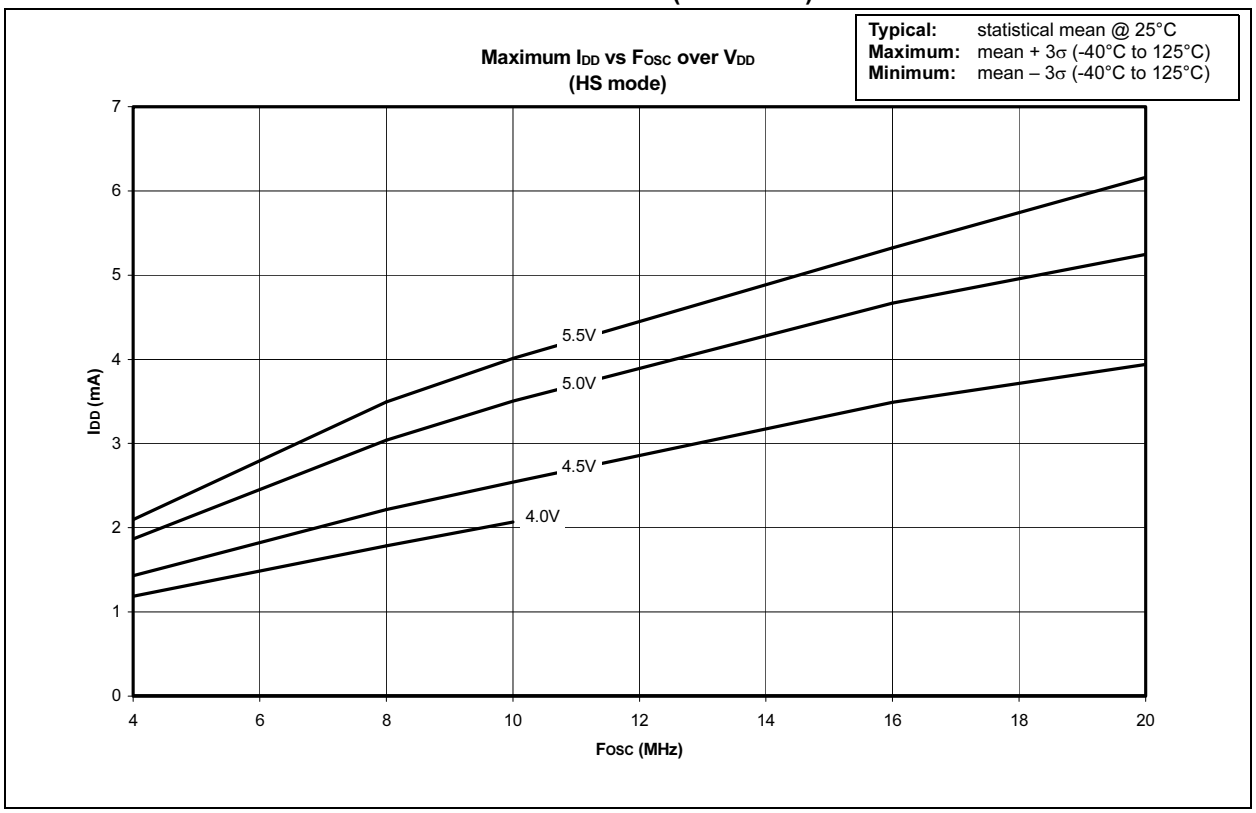
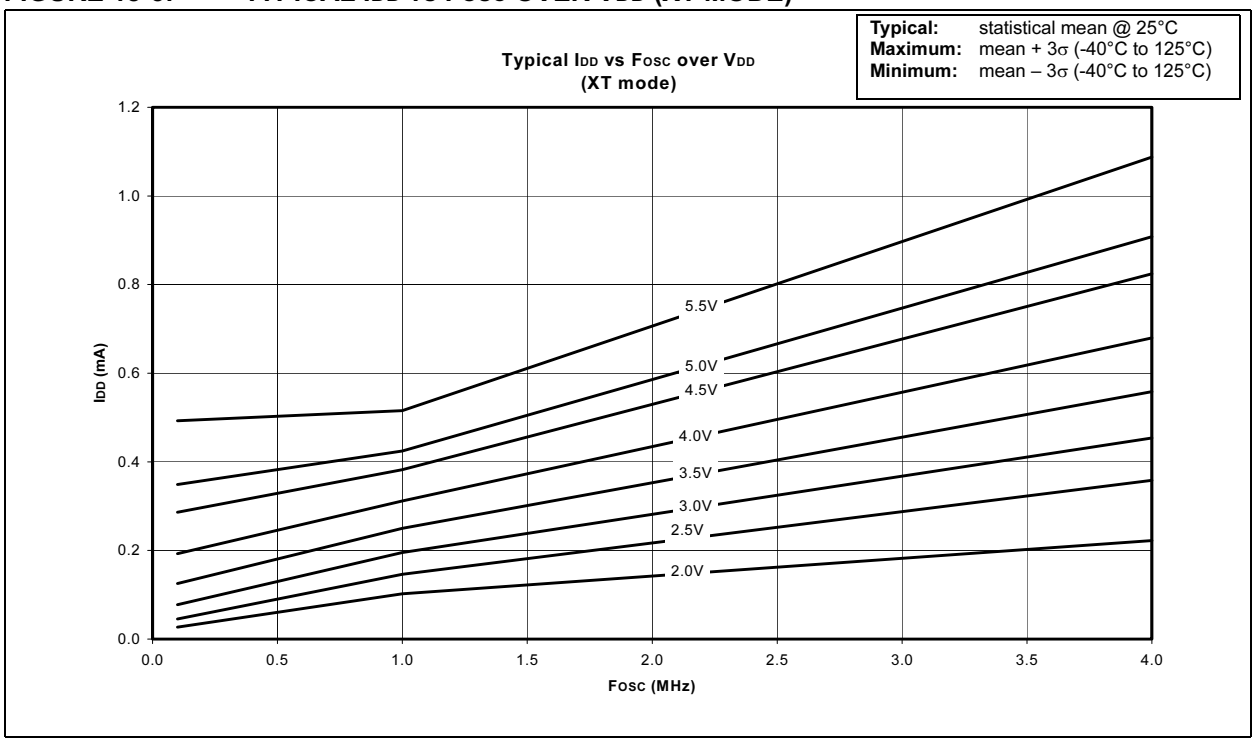



**FIGURE 18-3: TYPICAL IDD vs FOSC OVER VDD (XT MODE)**




 © 2003 Microchip Technology Inc.

**Note:** The graphs and tables provided in this section are for design guidance and are not tested.

**FIGURE 18-24: MAXIMUM IDD VS VDD OVER TEMPERATURE (-40 TO +125°C) INTERNAL 37 kHz OSCILLATOR**

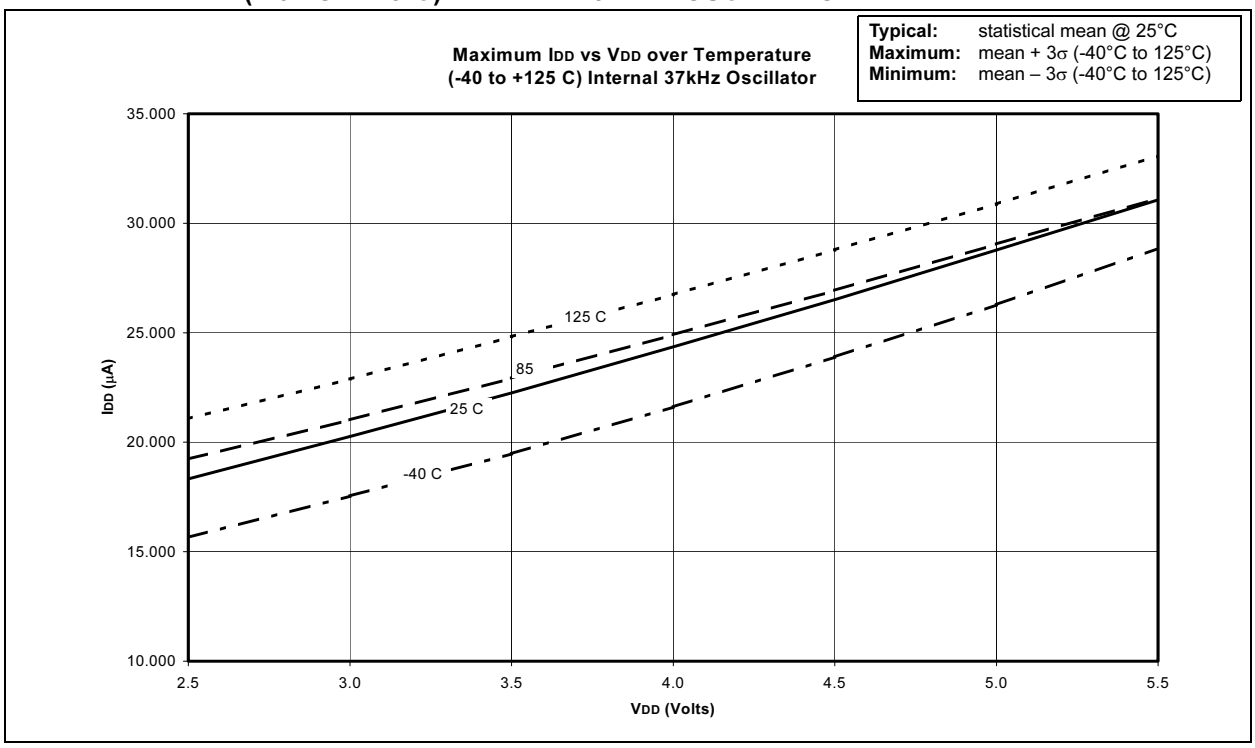
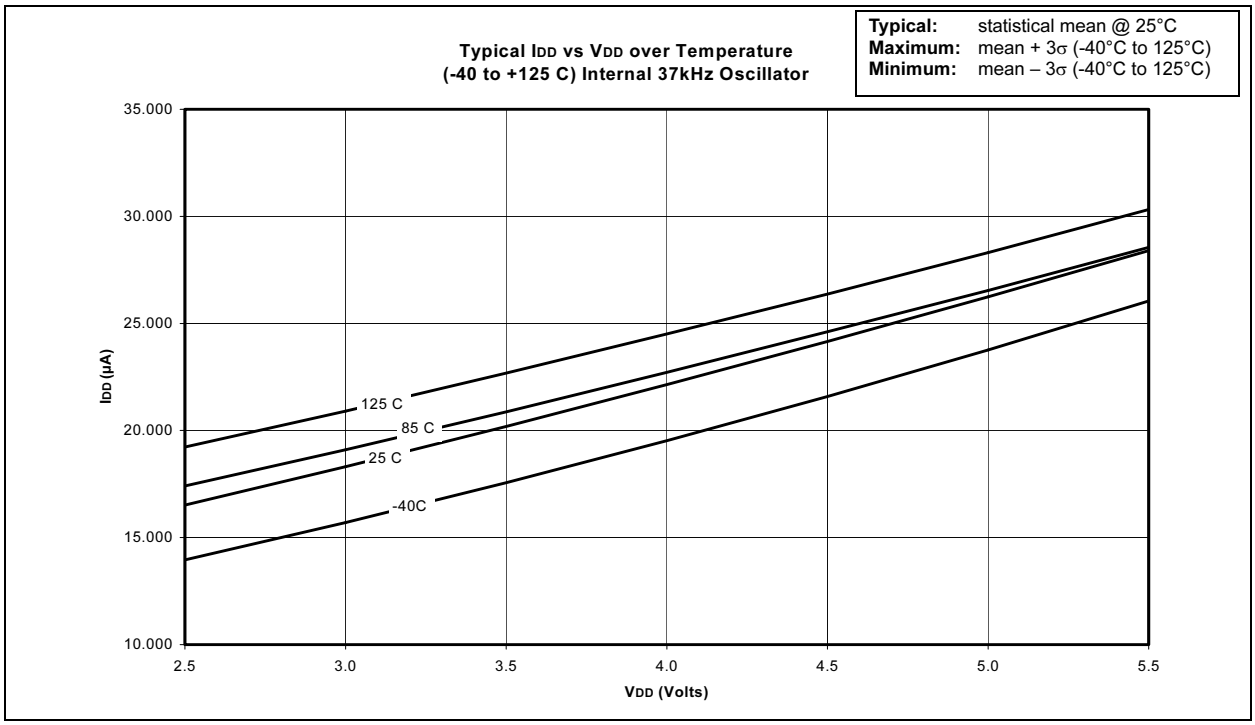
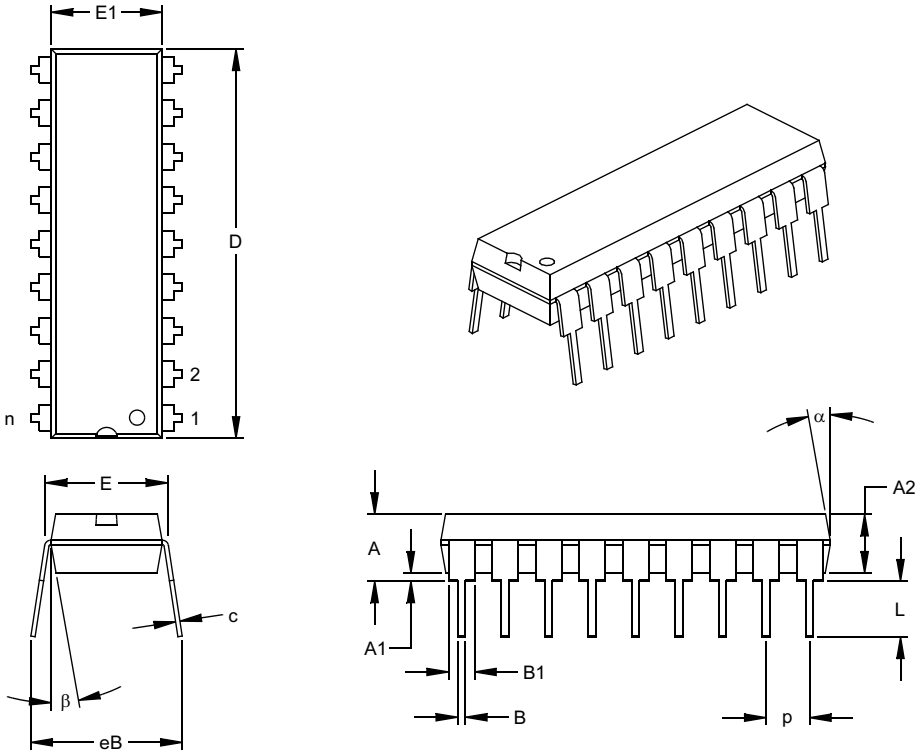



**FIGURE 18-25: TYPICAL IDD VS VDD OVER TEMPERATURE (-40 TO +125°C) INTERNAL 37 kHz OSCILLATOR**



## K04-007 18-Lead Plastic Dual In-line (P) – 300 mil



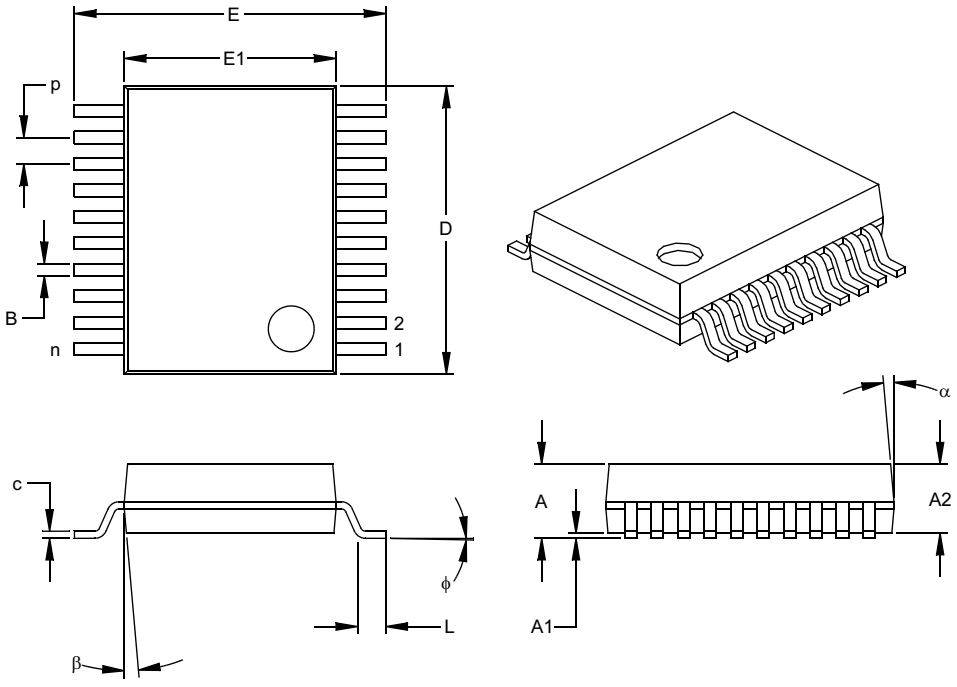

| Units | | INCHES* | | | MILLIMETERS | | |
|---|---|---|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX | MIN | NOM | MAX |
| Number of Pins | n | | 18 | | | 18 | |
| Pitch | p | | .100 | | | 2.54 | |
| Top to Seating Plane | A | .140 | .155 | .170 | 3.56 | 3.94 | 4.32 |
| Molded Package Thickness | A2 | .115 | .130 | .145 | 2.92 | 3.30 | 3.68 |
| Base to Seating Plane | A1 | .015 | | | 0.38 | | |
| Shoulder to Shoulder Width | E | .300 | .313 | .325 | 7.62 | 7.94 | 8.26 |
| Molded Package Width | E1 | .240 | .250 | .260 | 6.10 | 6.35 | 6.60 |
| Overall Length | D | .890 | .898 | .905 | 22.61 | 22.80 | 22.99 |
| Tip to Seating Plane | L | .125 | .130 | .135 | 3.18 | 3.30 | 3.43 |
| Lead Thickness | c | .008 | .012 | .015 | 0.20 | 0.29 | 0.38 |
| Upper Lead Width | B1 | .045 | .058 | .070 | 1.14 | 1.46 | 1.78 |
| Lower Lead Width | B | .014 | .018 | .022 | 0.36 | 0.46 | 0.56 |
| Overall Row Spacing § | eB | .310 | .370 | .430 | 7.87 | 9.40 | 10.92 |
| Mold Draft Angle Top | α | 5 | 10 | 15 | 5 | 10 | 15 |
| Mold Draft Angle Bottom | β | 5 | 10 | 15 | 5 | 10 | 15 |

* 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: MS-001
Drawing No. C04-007

 © 2003 Microchip Technology Inc.

## K04-072 20-Lead Plastic Shrink Small Outline (SS) – 5.30 mm




| 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

 © 2003 Microchip Technology Inc.

## INDEX

 © 2003 Microchip Technology Inc.

## READER RESPONSE

It is our intention to provide you with the best documentation possible to ensure successful use of your Microchip product. If you wish to provide your comments on organization, clarity, subject matter, and ways in which our documentation can better serve you, please FAX your comments to the Technical Publications Manager at (480) 792-4150.

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

To: Technical Publications Manager  Total Pages Sent _____

RE: Reader Response

From: Name _____

Company _____

Address _____

City / State / ZIP / Country _____

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

Application (optional):

Would you like a reply? ___Y ___N

Device: PIC16F62X  Literature Number: DS40300C

Questions:

1. What are the best features of this document?

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

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

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

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

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

7. How would you improve this document?


© 2003 Microchip Technology Inc.