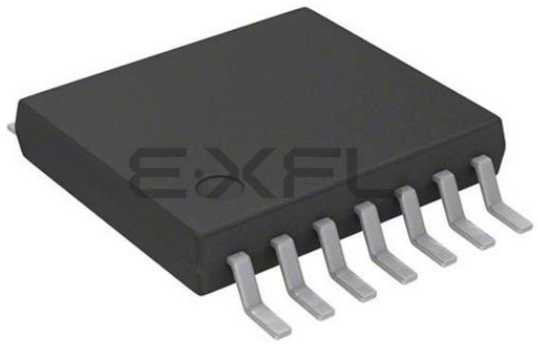


**Welcome to E-XFL.COM**

## 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, PWM, WDT |
| Number of I/O | 11 |
| Program Memory Size | 3.5KB (2K x 14) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 128 x 8 |
| Voltage - Supply (Vcc/Vdd) | 2V ~ 5V |
| Data Converters | A/D 8x10b; D/A 1x9b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 14-TSSOP (0.173", 4.40mm Width) |
| Supplier Device Package | 14-TSSOP |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic16hv753t-i-st |

### 2.3.3 INTCON REGISTER

The INTCON register is a readable and writable register, which contains the various enable and flag bits for TMR0 register overflow, IOCIE change and external RA2/INT pin interrupts.

**Note:** Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the Global Enable bit, GIE of the INTCON register. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

**REGISTER 2-3: INTCON: INTERRUPT CONTROL REGISTER**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| GIE | PEIE | T0IE | INTE | IOCIE | T0IF | INTF | IOCIF |
| bit 7 | | | | | | | bit 0 |

| **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 |

bit 7 **GIE:** Global Interrupt Enable bit
1 = Enables all unmasked interrupts
0 = Disables all interrupts

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

bit 5 **T0IE:** Timer0 Overflow Interrupt Enable bit
1 = Enables the Timer0 interrupt
0 = Disables the Timer0 interrupt

bit 4 **INTE:** RA2/INT External Interrupt Enable bit
1 = Enables the RA2/INT external interrupt
0 = Disables the RA2/INT external interrupt

bit 3 **IOCIE:** Interrupt-on-Change Interrupt Enable bit(1)
1 = Enables the IOC change interrupt
0 = Disables the IOC change interrupt

bit 2 **T0IF:** Timer0 Overflow Interrupt Flag bit(2)
1 = Timer0 register has overflowed (must be cleared in software)
0 = Timer0 register did not overflow

bit 1 **INTF:** RA2/INT External Interrupt Flag bit
1 = The RA2/INT external interrupt occurred (must be cleared in software)
0 = The RA2/INT external interrupt did not occur

bit 0 **IOCIF:** Interrupt-on-Change Interrupt Flag bit
1 = An IOC pin has changed state and generated an interrupt
0 = No pin interrupts have been generated

**Note 1:** IOC register must also be enabled.
**2:** T0IF bit is set when TMR0 rolls over. TMR0 is unchanged on Reset and should be initialized before clearing T0IF bit.

## 3.5 Writing the Flash Program Memory

A word of the Flash program memory may only be written to if the word is in an unprotected segment of memory.

Flash program memory must be written in four-word blocks. See Figure 3-2 and Figure 3-3 for more details. A block consists of four words with sequential addresses, with a lower boundary defined by an address, where PMADRL<1:0> = 00. All block writes to program memory are done as 16-word erase by four-word write operations. The write operation is edge-aligned and cannot occur across boundaries.

To write program data, it must first be loaded into the buffer registers (see Figure 3-2). This is accomplished by first writing the destination address to PMADRL and PMADRH and then writing the data to PMDATL and PMDATH. After the address and data have been set up, then the following sequence of events must be executed:

1. Write 55h, then AAh, to PMCON2 (Flash programming sequence).
2. Set the WR control bit of the PMCON1 register.

All four buffer register locations should be written to with correct data. If less than four words are being written to in the block of four words, then a read from the program memory location(s) not being written to must be performed. This takes the data from the program location(s) not being written and loads it into the PMDATL and PMDATH registers. Then the sequence of events to transfer data to the buffer registers must be executed.

To transfer data from the buffer registers to the program memory, the PMADRL and PMADRH must point to the last location in the four-word block (PMADRL<1:0> = 11). Then the following sequence of events must be executed:

1. Write 55h, then AAh, to PMCON2 (Flash programming sequence).
2. Set control bit WR of the PMCON1 register to begin the write operation.

The user must follow the same specific sequence to initiate the write for each word in the program block, writing each program word in sequence (000, 001, 010, 011). When the write is performed on the last word (PMADRL<1:0> = 11), a block of sixteen words is automatically erased and the content of the four-word buffer registers are written into the program memory.

After the "BSF PMCON1,WR" instruction, the processor requires two cycles to set up the erase/write operation. The user must place two NOP instructions after the WR bit is set. Since data is being written to buffer registers, the writing of the first three words of the block appears to occur immediately. The processor will halt internal operations for the typical 4 ms, only during the cycle in which the erase takes place (i.e., the last word of the sixteen-word block erase). This is not Sleep mode as the clocks and peripherals will continue to run. After the four-word write cycle, the processor will resume operation with the third instruction after the PMCON1 write instruction. The above sequence must be repeated for the higher 12 words.

## 3.6 Protection Against Spurious Write

There are conditions when the device should not write to the program memory. To protect against spurious writes, various mechanisms have been built in. On power-up, WREN is cleared. Also, the Power-up Timer (64 ms duration) prevents program memory writes.

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

## 3.7 Operation During Code-Protect

When the device is code-protected, the CPU is able to read and write unscrambled data to the program memory.

## 3.8 Operation During Write Protect

When the program memory is write-protected, the CPU can read and execute from the program memory. The portions of program memory that are write-protected can be modified by the CPU using the PMCON registers, but the protected program memory cannot be modified using ICSP mode.

**TABLE 3-1: SUMMARY OF REGISTERS ASSOCIATED WITH FLASH PROGRAM MEMORY**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| PMCON1 | — | — | — | — | — | WREN | WR | RD | 27 |
| PMCON2 | Program Memory Control Register 2 | | | | | | | | 27 |
| PMADRL | PMADRL<7:0> | | | | | | | | 26 |
| PMADRH | — | — | — | — | — | — | PMADRH<1:0> | | 26 |
| PMDATL | PMDATL<7:0> | | | | | | | | 26 |
| PMDATH | — | — | PMDATH<5:0> | | | | | | 26 |
| INTCON | GIE | PEIE | T0IE | INTE | IOCIE | T0IF | INTF | IOCIF | 17 |

**Legend:** — = unimplemented location, read as '0'. Shaded cells are not used by Flash program memory module.
* Page provides register information.

**TABLE 3-2: SUMMARY OF CONFIGURATION WORD WITH FLASH PROGRAM MEMORY**

| Name | Bits | Bit -/7 | Bit -/6 | Bit 13/5 | Bit 12/4 | Bit 11/3 | Bit 10/2 | Bit 9/1 | Bit 8/0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|---|
| CONFIG(1) | 13:8 | — | — | $\overline{\text{DEBUG}}$ | $\overline{\text{CLKOUTEN}}$ | WRT<1:0> | | BOREN<1:0> | | 150 |
| | 7:0 | — | $\overline{\text{CP}}$ | MCLRE | $\overline{\text{PWRTE}}$ | WDTE | — | — | FOSC0 | |

**Legend:** — = unimplemented location, read as '1'. Shaded cells are not used by Flash program memory.
**Note 1:** See Configuration Word register (Register 19-1) for operation of all register bits.

## 7.3 Timer1 Prescaler

Timer1 has four prescaler options allowing one, two, four or eight divisions of the clock input. The T1CKPS bits of the T1CON register control the prescale counter. The prescale counter is not directly readable or writable; however, the prescaler counter is cleared upon a write to TMR1H or TMR1L.

## 7.4 Timer1 Operation in Asynchronous Counter Mode

If control bit $\overline{\text{T1SYNC}}$ of the T1CON register is set, the external clock input is not synchronized. The timer increments asynchronously to the internal phase clocks. If external clock source is selected then the timer will continue to run during Sleep and can generate an interrupt on overflow, which will wake-up the processor. However, special precautions in software are needed to read/write the timer (see **Section 7.4.1 "Reading and Writing Timer1 in Asynchronous Counter Mode"**).

| | |
|---|---|
| **Note:** | When switching from synchronous to asynchronous operation, it is possible to skip an increment. When switching from asynchronous to synchronous operation, it is possible to produce an additional increment. |

### 7.4.1 READING AND WRITING TIMER1 IN ASYNCHRONOUS COUNTER MODE

Reading TMR1H or TMR1L while the timer is running from an external asynchronous clock will ensure a valid read (taken care of in hardware). However, the user should keep in mind that reading the 16-bit timer in two 8-bit values itself, poses certain problems, since the timer may overflow between the reads.

For writes, it is recommended that the user simply stop the timer and write the desired values. A write contention may occur by writing to the timer registers, while the register is incrementing. This may produce an unpredictable value in the TMR1H:TMR1L register pair.

## 7.5 Timer1 Gate

Timer1 can be configured to count freely or the count can be enabled and disabled using Timer1 gate circuitry. This is also referred to as Timer1 gate count enable.

Timer1 gate can also be driven by multiple selectable sources.

### 7.5.1 TIMER1 GATE COUNT ENABLE

The Timer1 gate is enabled by setting the TMR1GE bit of the T1GCON register. The polarity of the Timer1 gate is configured using the T1GPOL bit of the T1GCON register.

When Timer1 Gate (T1G) input is active, Timer1 will increment on the rising edge of the Timer1 clock source. When Timer1 gate input is inactive, no incrementing will occur and Timer1 will hold the current count. See Figure 7-3 for timing details.

**TABLE 7-3: TIMER1 GATE ENABLE SELECTIONS**

| T1CLK | T1GPOL | T1G | Timer1 Operation |
|---|---|---|---|
| ↑ | 0 | 0 | Counts |
| ↑ | 0 | 1 | Holds Count |
| ↑ | 1 | 0 | Holds Count |
| ↑ | 1 | 1 | Counts |

### 7.5.2 TIMER1 GATE SOURCE SELECTION

The Timer1 gate source can be selected from one of four different sources. Source selection is controlled by the T1GSS bits of the T1GCON register. The polarity for each available source is also selectable. Polarity selection is controlled by the T1GPOL bit of the T1GCON register.

**TABLE 7-4: TIMER1 GATE SOURCES**

| T1GSS | Timer1 Gate Source |
|---|---|
| 11 | SYNCC2OUT |
| 10 | SYNCC1OUT |
| 01 | Overflow of Timer0<br>(TMR0 increments from FFh to 00h) |
| 00 | Timer1 Gate Pin |

## 8.0 TIMER2 MODULE

The Timer2 module is an 8-bit timer with the following features:

- 8-bit timer register (TMR2)
- 8-bit period register (PR2)
- Interrupt on TMR2 match with PR2
- Software programmable prescaler (1:1, 1:4, 1:16, 1:64)
- Software programmable postscaler (1:1 to 1:16)

See Figure 8-1 for a block diagram of Timer2.

### 8.1 Timer2 Operation

The clock input to the Timer2 module is the system instruction clock (FOSC/4). The clock is fed into the Timer2 prescaler, which has prescale options of 1:1, 1:4 or 1:16. The output of the prescaler is then used to increment the TMR2 register.

The values of TMR2 and PR2 are constantly compared to determine when they match. TMR2 will increment from 00h until it matches the value in PR2. When a match occurs, two things happen:

- TMR2 is reset to 00h on the next increment cycle.
- The Timer2 postscaler is incremented

The match output of the Timer2/PR2 comparator is then fed into the Timer2 postscaler. The postscaler has postscale options of 1:1 to 1:16 inclusive. The output of the Timer2 postscaler is used to set the TMR2IF interrupt flag bit in the PIR1 register.

The TMR2 and PR2 registers are both fully readable and writable. On any Reset, the TMR2 register is set to 00h and the PR2 register is set to FFh.

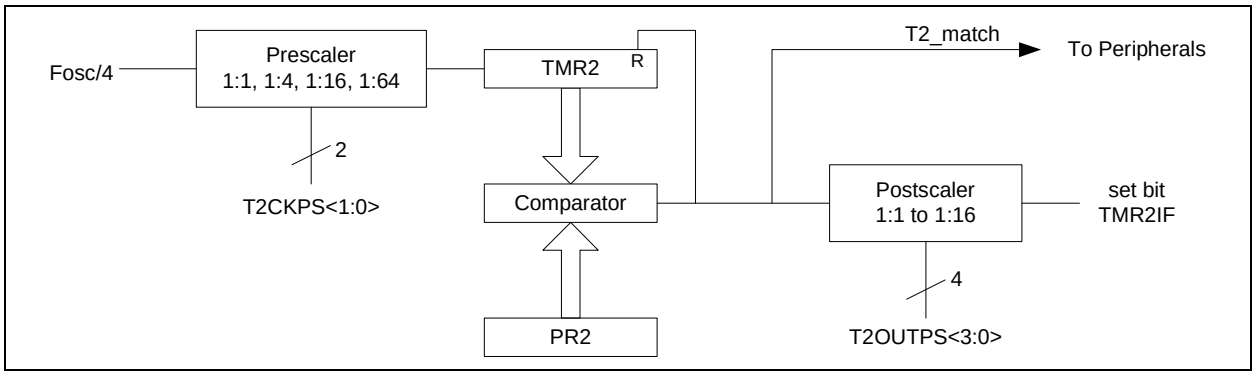
Timer2 is turned on by setting the TMR2ON bit in the T2CON register to a '1'. Timer2 is turned off by clearing the TMR2ON bit to a '0'.

The Timer2 prescaler is controlled by the T2CKPS bits in the T2CON register. The Timer2 postscaler is controlled by the T2OUTPS bits in the T2CON register. The prescaler and postscaler counters are cleared when:

- A write to TMR2 occurs.
- A write to T2CON occurs.
- Any device Reset occurs (Power-on Reset, $\overline{\text{MCLR}}$ Reset, Watchdog Timer Reset, or Brown-out Reset).

| **Note:** | TMR2 is not cleared when T2CON is written. |
|---|---|

**FIGURE 8-1: TIMER2 BLOCK DIAGRAM**



## 11.4 Output Control

Upon disabling, or immediately after enabling the COG module, the complementary drive is configured with COGxOUT0 drive inactive and COGxOUT1 drive active.

### 11.4.1 OUTPUT ENABLES

Each COG output pin has an individual output enable control. Output enables are selected with the GxOE0 and GxOE1 bits of the COGxCON0 register (Register 11-1). When an output enable control is cleared, the module asserts no control over the pin. When an output enable is set, the override value or PWM waveform is applied to the pin per the port priority selection.

The device pin output enable control bits are independent of the GxEN bit of the COGxCON0 register, which enables the COG. When GxEN is cleared, and shutdown is not active, the Reset state PWM levels are present on the COG output pins. The PWM levels are affected by the polarity controls. If shutdown is active when GxEN is cleared, the shutdown override levels will be present on the COG output pins. Note that setting the GxASE bit while the GxEN bit is cleared will activate shutdown which can only be cleared by either a rising event while the GxEN bit is set, or a device Reset.

### 11.4.2 POLARITY CONTROL

The polarity of each COG output can be selected independently. When the output polarity bit is set, the corresponding output is active-low. Clearing the output polarity bit configures the corresponding output as active-high. However, polarity does not affect the shutdown override levels.

Output polarity is selected with the GxPOL0 and GxPOL1 bits of the COGxCON0 register (Register 11-1).

## 11.5 Dead-Band Control

The dead-band control provides for non-overlapping PWM output signals to prevent shoot-through current in the external power switches.

The COG contains two dead-band timers. One dead-band timer is used for rising event dead-band control. The other is used for falling event dead-band control. Timer modes are selectable as either:

- Asynchronous delay chain
- Synchronous counter

The dead-band Timer mode is selected for the COGxOUT0 and COGxOUT1 dead-band times with the respective GxRDBTS and GxFDBTS bits of the COGxCON1 register (Register 11-2).

### 11.5.1 ASYNCHRONOUS DELAY CHAIN DEAD-BAND DELAY

Asynchronous dead-band delay is determined by the time it takes the input to propagate through a series of delay elements. Each delay element is a nominal five nanoseconds.

Set the COGxDBR register (Register 11-9) value to the desired number of delay elements in the COGxOUT0 dead band. Set the COGxDBF register (Register 11-10) value to the desired number of delay elements in the COGxOUT1 dead band. When the value is zero, dead-band delay is disabled.

### 11.5.2 SYNCHRONOUS COUNTER DEAD-BAND DELAY

Synchronous counter dead band is timed by counting COG_clock periods from zero up to the value in the dead-band count register. Use Equation 11-1 to calculate dead-band times.

Set the COGxDBR count register value to obtain the desired dead-band time of the COGxOUT0 output. Set the COGxDBF count register value to obtain the desired dead-band time of the COGxOUT1 output. When the value is zero, dead-band delay is disabled.

### 11.5.3 SYNCHRONOUS COUNTER DEAD-BAND TIME UNCERTAINTY

When the rising and falling events that trigger the dead-band counters come from asynchronous inputs, it creates uncertainty in the synchronous counter dead-band time. The maximum uncertainty is equal to one COG_clock period. Refer to Equation 11-1 for more detail.

When event input sources are asynchronous with no phase delay, use the asynchronous delay chain dead-band mode to avoid the dead-band time uncertainty.

## 11.13 Register Definitions: COG Control

**REGISTER 11-1: COGxCON0: COG CONTROL REGISTER 0**

| R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | U-0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| GxEN | GxOE1 | GxOE0 | GxPOL1 | GxPOL0 | GxLD | — | GxMD |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| ‘1’ = Bit is set | ‘0’ = Bit is cleared | q = Value depends on condition |

bit 7 **GxEN:** COGx Enable bit
1 = Module is enabled
0 = Module is disabled

bit 6 **GxOE1:** COGxOUT1 Output Enable bit
1 = COGxOUT1 is available on associated I/O pin
0 = COGxOUT1 is not available on associated I/O pin

bit 5 **GxOE0:** COGxOUT0 Output Enable bit
1 = COGxOUT0 is available on associated I/O pin
0 = COGxOUT0 is not available on associated I/O pin

bit 4 **GxPOL1:** COGxOUT1 Output Polarity bit
1 = Output is inverted polarity
0 = Output is normal polarity

bit 3 **GxPOL0:** COGxOUT0 Output Polarity bit
1 = Output is inverted polarity
0 = Output is normal polarity

bit 2 **GxLD:** COGx Load Buffers bit
1 = Phase, blanking, and dead-band buffers to be loaded with register values on next input events
0 = Register to buffer transfer is complete

bit 1 **Unimplemented:** Read as ‘0’

bit 0 **GxMD:** COGx Mode bit
1 = COG outputs operate in Push-Pull mode
0 = COG outputs operate in Synchronous mode

**REGISTER 11-6: COGxFSIM: COG FALLING EVENT SOURCE INPUT MODE REGISTER**

| U-0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| — | GxFMHLT2 | GxFMHLT1 | GxFMT2M | GxFMFLT | GxFMCCP1 | GxFMC2 | GxFMC1 |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | q = Value depends on condition |

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

bit 6 **GxFMHLT2:** COGx Falling Event Input Source 6 Mode bit(1)
GxFIHLT2 = 1:
1 = HLTimer2 low-to-high transition will cause a falling event after falling event phase delay
0 = HLTimer2 high level will cause an immediate falling event
GxFIHLT2 = 0:
HLTimer2 has no effect on falling event

bit 5 **GxFMHLT1:** COGx Falling Event Input Source 5 Mode bit(1)
GxFIHLT1 = 1:
1 = HLTimer1 low-to-high transition will cause a falling event after falling event phase delay
0 = HLTimer1 high level will cause an immediate falling event
GxFIHLT1 = 0:
HLTimer1 has no effect on falling event

bit 4 **GxFMT2M:** COGx Falling Event Input Source 4 Mode bit(1)
GxFIT2M = 1:
1 = Timer2 match with PR2 low-to-high transition will cause a falling event after falling event phase delay
0 = Timer2 match with PR2 high level will cause an immediate falling event
GxFIT2M = 0:
Timer2 match with PR2 has no effect on falling event

bit 3 **GxFMFLT:** COGx Falling Event Input Source 3 Mode bit
GxFIFLT = 1:
1 = COGxFLT pin low-to-high transition will cause a falling event after falling event phase delay
0 = COGxFLT pin high level will cause an immediate falling event
GxFIFLT = 0:
COGxFLT pin has no effect on falling event

bit 2 **GxFMCCP1:** COGx Falling Event Input Source 2 Mode bit
GxFICCP1 = 1:
1 = CCP1 low-to-high transition will cause a falling event after falling event phase delay
0 = CCP1 high level will cause an immediate falling event
GxFICCP1 = 0:
CCP1 has no effect on falling event

bit 1 **GxFMC2:** COGx Falling Event Input Source 1 Mode bit
GxFIC2 = 1:
1 = Comparator 2 low-to-high transition will cause a falling event after falling event phase delay
0 = Comparator 2 high level will cause an immediate falling event
GxFIC2 = 0:
Comparator 2 has no effect on falling event

bit 0 **GxFMC1:** COGx Falling Event Input Source 0 Mode bit
GxFIC1 = 1:
1 = Comparator 1 low-to-high transition will cause a falling event after falling event phase delay
0 = Comparator 1 high level will cause an immediate falling event
GxFIC1 = 0:
Comparator 1 has no effect on falling event

**Note 1:** These sources are pulses and therefore the only benefit of Edge mode over Level mode is that they can be delayed by falling event phase delay.

**REGISTER 11-11: COGxBKR: COG RISING EVENT BLANKING COUNT REGISTER**

| U-0 | U-0 | U-0 | U-0 | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u |
|---|---|---|---|---|---|---|---|
| — | — | — | — | GxBKR<3:0> | | | |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| ‘1’ = Bit is set | ‘0’ = Bit is cleared | q = Value depends on condition |

bit 7-4 **Unimplemented:** Read as ‘0’

bit 3-0 **GxBKR<3:0>:** Rising Event Blanking Count Value bits

= Number of COGx clock periods to inhibit falling event inputs

**REGISTER 11-12: COGxBKF: COG FALLING EVENT BLANKING COUNT REGISTER**

| U-0 | U-0 | U-0 | U-0 | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u |
|---|---|---|---|---|---|---|---|
| — | — | — | — | GxBKF<3:0> | | | |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| ‘1’ = Bit is set | ‘0’ = Bit is cleared | q = Value depends on condition |

bit 7-4 **Unimplemented:** Read as ‘0’

bit 3-0 **GxBKF<3:0>:** Falling Event Blanking Count Value bits

= Number of COGx clock periods to inhibit rising event inputs

## 15.2 Comparator Control

Each comparator has two control registers: CMxCON0 and CMxCON1.

The CMxCON0 registers (see Register 15-1) contain Control and Status bits for the following:

- Enable
- Output selection
- Output pin enable
- Output polarity
- Speed/Power selection
- Hysteresis enable
- Output synchronization

The CMxCON1 registers (see Register 15-2) contain Control bits for the following:

- Interrupt edge polarity (rising and/or falling)
- Positive input channel selection
- Negative input channel selection

### 15.2.1 COMPARATOR ENABLE

Setting the CxON bit of the CMxCON0 register enables the comparator for operation. Clearing the CxON bit disables the comparator resulting in minimum current consumption.

### 15.2.2 COMPARATOR OUTPUT SELECTION

The output of the comparator can be monitored by reading either the CxOUT bit of the CMxCON0 register or the MCOUTx bit of the CMOUT register. In order to make the output available for an external connection, the following conditions must be true:

- CxOE bit of the CMxCON0 register must be set
- Corresponding TRIS bit must be cleared
- CxON bit of the CMxCON0 register must be set

| | |
|---|---|
| **Note 1:** | The CxOE bit of the CMxCON0 register overrides the PORT data latch. Setting the CxON bit of the CMxCON0 register has no impact on the port override. |
| **2:** | The internal output of the comparator is latched with each instruction cycle. Unless otherwise specified, external outputs are not latched. |

### 15.2.3 COMPARATOR OUTPUT POLARITY

Inverting the output of the comparator is functionally equivalent to swapping the comparator inputs. The polarity of the comparator output can be inverted by setting the CxPOL bit of the CMxCON0 register. Clearing the CxPOL bit results in a non-inverted output.

Table 15-1 shows the output state versus input conditions, including polarity control.

**TABLE 15-1: COMPARATOR OUTPUT STATE VS. INPUT CONDITIONS**

| Input Condition | CxPOL | CxOUT |
|---|---|---|
| CxVN > CxVP | 0 | 0 |
| CxVN < CxVP | 0 | 1 |
| CxVN > CxVP | 1 | 1 |
| CxVN < CxVP | 1 | 0 |

### 15.2.4 COMPARATOR SPEED/POWER SELECTION

The trade-off between speed or power can be optimized during program execution with the CxSP control bit. The default state for this bit is '1' which selects the normal speed mode. Device power consumption can be optimized at the cost of slower comparator propagation delay by clearing the CxSP bit to '0'.

## 15.3 Comparator Hysteresis

A selectable amount of separation voltage can be added to the input pins of each comparator to provide a hysteresis function to the overall operation. Hysteresis is enabled by setting the CxHYS bit of the CMxCON0 register.

See **Section 22.0 "Electrical Specifications"** for more information.

## 15.4 Timer1 Gate Operation

The output resulting from a comparator operation can be used as a source for gate control of Timer1. See **Section 7.5 "Timer1 Gate"** for more information. This feature is useful for timing the duration or interval of an analog event.

It is recommended that the comparator output be synchronized to Timer1. This ensures that Timer1 does not increment while a change in the comparator is occurring.

## 16.4 Register Definitions: OPA Control

**REGISTER 16-1: OPAxCON0: OP AMP CONTROL REGISTER**

| R/W-0/0 | U-0 | U-0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| OPAxEN | — | — | OPAxUGM | OPAxNCH<1:0> | | OPAxPCH<1:0> | |
| bit 7 | | | | | | | bit 0 |

| **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 |

bit 7 **OPAxEN:** OPAx Enable bit
1 = OPAx is enabled
0 = OPAx is disabled

bit 6-5 **Unimplemented:** Read as '0'

bit 4 **OPAxUGM:** OPAx Unity Gain Mode Enable bit
1 = OPAx is in Unity gain mode
0 = OPAx is not in Unity gain mode - operates as a three-terminal op amp

bit 3-2 **OPAxNCH<1:0>:** OPAx Negative Input Source Selection bit
11 = OPAx- connects to FVR_buffer1
10 = OPAx- connects to DAC1_output
0x = OPAx- connects to OPAxIN- pin

bit 1-0 **OPAxPCH<1:0>:** OPAx Positive Input Source Selection bit
11 = OPAx+ connects to FVR_buffer1
10 = OPAx+ connects to DAC1_output
01 = OPAx+ connects to SLOPE_output
00 = OPAx+ connects to OPAxIN+ pin

**TABLE 16-1: REGISTERS ASSOCIATED WITH THE OPA MODULE**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| OPA1CON0 | OPA1EN | — | — | OPA1UGM | OPA1NCH<1:0> | | OPA1PCH<1:0> | | 134 |
| TRISC | — | — | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 | 49 |
| ANSELC | — | — | — | — | ANSC3 | ANSC2 | ANSC1 | ANSC0 | 50 |

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

## 18.0 INSTRUCTION SET SUMMARY

The PIC16F753/HV753 instruction set is highly orthogonal and is comprised of three basic categories:

- **Byte-oriented** operations
- **Bit-oriented** operations
- **Literal and control** operations

Each PIC16 instruction is a 14-bit word divided into an **opcode,** which specifies the instruction type and one or more **operands,** which further specify the operation of the instruction. The formats for each of the categories is presented in Figure 18-1, while the various opcode fields are summarized in Table 18-1.

Table 18-2 lists the instructions recognized by the MPASM™ assembler.

For **byte-oriented** instructions, 'f' represents a file register designator and 'd' represents a destination designator. The file register designator specifies which file register is to be used by the instruction.

The destination designator specifies where the result of the operation is to be placed. If 'd' is zero, the result is placed in the W register. If 'd' is one, the result is placed in the file register specified in the instruction.

For **bit-oriented** instructions, 'b' represents a bit field designator, which selects the bit affected by the operation, while 'f' represents the address of the file in which the bit is located.

For **literal and control** operations, 'k' represents an 8-bit or 11-bit constant, or literal value.

One instruction cycle consists of four oscillator periods; for an oscillator frequency of 4 MHz, this gives a normal instruction execution time of 1 μs. All instructions are executed within a single instruction cycle, unless a conditional test is true, or the program counter is changed as a result of an instruction. When this occurs, the execution takes two instruction cycles, with the second cycle executed as a NOP.

All instruction examples use the format '0xhh' to represent a hexadecimal number, where 'h' signifies a hexadecimal digit.

### 18.1 Read-Modify-Write Operations

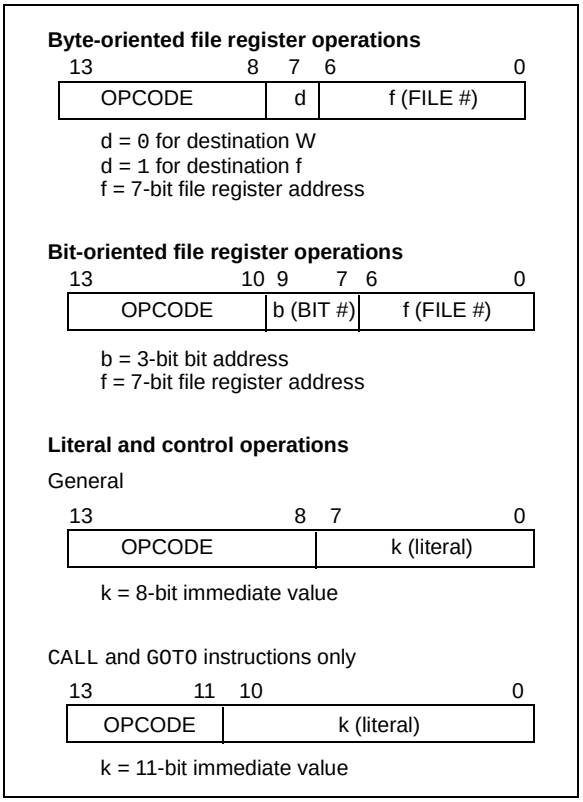
Any instruction that specifies a file register as part of the instruction performs a Read-Modify-Write (RMW) operation. The register is read, the data is modified, and the result is stored according to either the instruction or the destination designator 'd'. A read operation is performed on a register even if the instruction writes to that register.

For example, a CLRF PORTA instruction will read PORTA, clear all the data bits, then write the result back to PORTA. This example would have the unintended consequence of clearing the condition that set the IOCIF flag.

**TABLE 18-1: OPCODE FIELD DESCRIPTIONS**

| Field | Description |
|---|---|
| f | Register file address (0x00 to 0x7F) |
| W | Working register (accumulator) |
| b | Bit address within an 8-bit file register |
| k | Literal field, constant data or label |
| x | Don't care location (= 0 or 1).<br>The assembler will generate code with x = 0.<br>It is the recommended form of use for compatibility with all Microchip software tools. |
| d | Destination select; d = 0: store result in W,<br>d = 1: store result in file register f.<br>Default is d = 1. |
| PC | Program Counter |
| $\overline{TO}$ | Time-out bit |
| C | Carry bit |
| DC | Digit carry bit |
| Z | Zero bit |
| $\overline{PD}$ | Power-down bit |

**FIGURE 18-1: GENERAL FORMAT FOR INSTRUCTIONS**



**REGISTER 19-1: CONFIGURATION WORD**

| R/P-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 |
|---|---|---|---|---|---|
| $\overline{\text{DEBUG}}$ | $\overline{\text{CLKOUTEN}}$ | WRT<1:0> | | BOREN<1:0> | |
| bit 13 | | | | | bit 8 |

| U-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 | U-1 | U-1 | R/P-1 |
|---|---|---|---|---|---|---|---|
| — | $\overline{\text{CP}}$ | MCLRE | $\overline{\text{PWRTE}}$ | WDTE | — | — | FOSC0 |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | P = Programmable bit | U = Unimplemented bit, read as ‘1’ |
| ‘0’ = Bit is cleared | ‘1’ = Bit is set | -n = Value when blank or after Bulk Erase |

bit 13 **$\overline{\text{DEBUG}}$:** Debug Mode Enable bit(2)
1 = Background debugger is disabled
0 = Background debugger is enabled

bit 12 **$\overline{\text{CLKOUTEN}}$:** Clock Out Enable bit
1 = Clock out function disabled. CLKOUT pin acts as I/O pin
0 = General purpose I/O disabled. CLKOUT pin acts as CLKOUT

bit 11-10 **WRT<1:0>:** Flash Program Memory Self-Write Enable bit
11 = Write protection off
10 = 000h to FFh write-protected, 100h to 3FFh may be modified by PMCON1 control
01 = 000h to 1FFh write-protected, 200h to 3FFh may be modified by PMCON1 control
00 = 000h to 3FFh write-protected, entire program is write-protected

bit 8-9 **BOREN<1:0>:** Brown-out Reset Enable bits
11 = BOR enabled
10 = BOR enabled during operation and disabled in Sleep
0x = BOR disabled

bit 7 **Unimplemented:** Read as ‘1’

bit 6 **$\overline{\text{CP}}$:** Code Protection bit
1 = Program memory code protection is disabled
0 = Program memory code protection is enabled

bit 5 **MCLRE:** $\overline{\text{MCLR}}$/$V_{PP}$ Pin Function Select bit
1 = $\overline{\text{MCLR}}$ pin is $\overline{\text{MCLR}}$ function and weak internal pull-up is enabled
0 = $\overline{\text{MCLR}}$ pin is input function, $\overline{\text{MCLR}}$ function is internally disabled

bit 4 **$\overline{\text{PWRTE}}$:** Power-up Timer Enable bit(1)
1 = PWRT disabled
0 = PWRT enabled

bit 3 **WDTE:** Watchdog Timer Enable bit
1 = WDT enabled
0 = WDT disabled

bit 2-1 **Unimplemented:** Read as ‘1’

bit 0 **FOSC:** Oscillator Selection bits
1 = EC oscillator selected: CLKIN on RA5/CLKIN
0 = Internal oscillator: I/O function on RA5/CLKIN

**Note 1:** Enabling Brown-out Reset does not automatically enable Power-up Timer.
**2:** The Configuration bit is managed automatically by the device development tools. The user should not attempt to manually write this bit location. However, the user should ensure that this location has been programmed to a ‘1’ and the device checksum is correct for proper operation of production software.

**FIGURE 19-5: TIME-OUT SEQUENCE ON POWER-UP (DELAYED $\overline{MCLR}$): CASE 2**

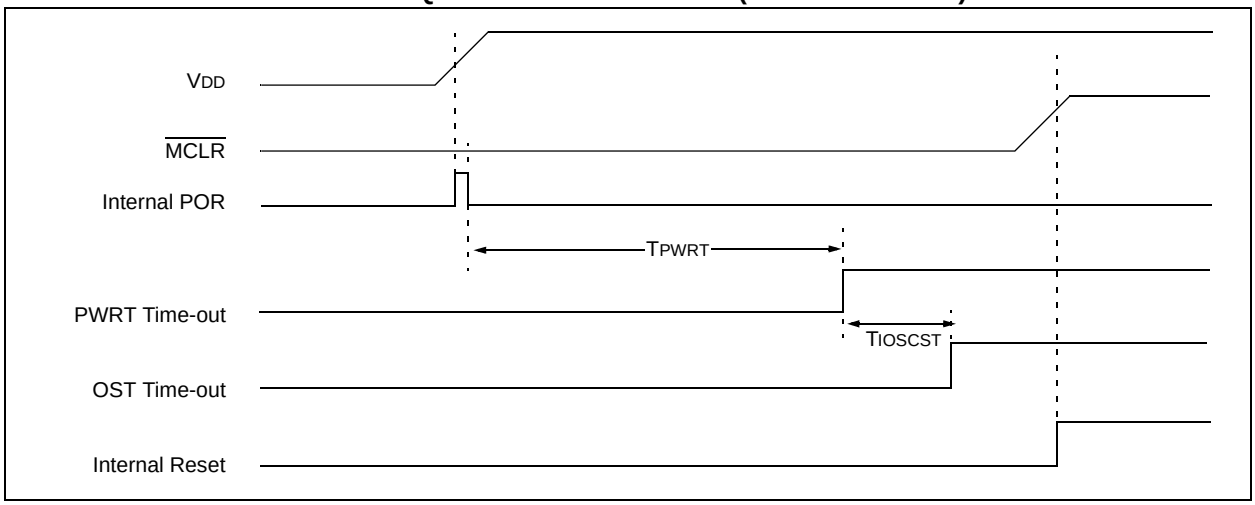



**FIGURE 19-6: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ WITH VDD)**

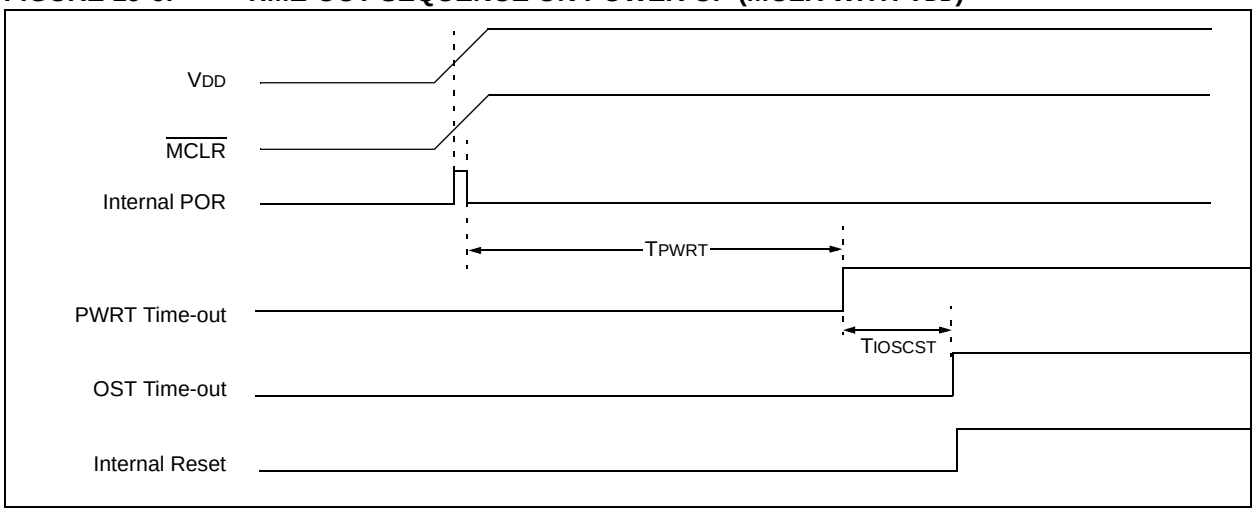


**TABLE 22-4: I/O PORTS**

| DC CHARACTERISTICS | | | Standard Operating Conditions (unless otherwise stated) | | | | |
|---|---|---|---|---|---|---|---|
| Param No. | Sym. | Characteristic | Min. | Typ† | Max. | Units | Conditions |
| | VIL | **Input Low Voltage** | | | | | |
| | | I/O PORT: | | | | | |
| D030 | | with TTL buffer | — | — | 0.8 | V | 4.5V ≤ VDD ≤ 5.5V |
| D030A | | | — | — | 0.15 VDD | V | 2.0V ≤ VDD ≤ 4.5V |
| D031 | | with Schmitt Trigger buffer | — | — | 0.2 VDD | V | 2.0V ≤ VDD ≤ 5.5V |
| | VIH | **Input High Voltage** | | | | | |
| | | I/O PORT: | | | | | |
| D040 | | with TTL buffer | 2.0 | — | — | V | 4.5V ≤ VDD ≤ 5.5V |
| D040A | | | 0.25 VDD + 0.8 | — | — | V | 2.0V ≤ VDD ≤ 4.5V |
| D041 | | with Schmitt Trigger buffer | 0.8 VDD | — | — | V | 2.0V ≤ VDD ≤ 5.5V |
| D042 | | $\overline{MCLR}$ | 0.8 VDD | — | — | V | |
| | IIL | **Input Leakage Current(1)** | | | | | |
| D060 | | I/O ports | — | ± 0.1 | ± 1 | µA | VSS ≤ VPIN ≤ VDD, Pin at high-impedance, 85°C |
| D061 | | RA3/$\overline{MCLR}$(2) | — | ± 0.7 | ± 5 | µA | VSS ≤ VPIN ≤ VDD, Pin at high-impedance, 85°C |
| D063 | | | — | ± 0.1 | ± 5 | µA | EC Configuration |
| | IPUR | **Weak Pull-up Current(3)** | | | | | |
| D070* | | | 50 | 250 | 400 | µA | VDD = 5.0V, VPIN = VSS |
| | VOL | **Output Low Voltage** | | | | | |
| D080 | | I/O Ports (excluding RC4, RC5) | — | — | 0.6 | V | IOL = 7 mA, VDD = 4.5V -40°C ≤ TA ≤ +125°C IOL = 8.5 mA, VDD = 4.5V -40°C ≤ TA ≤ +85°C |
| | | I/O Ports RC4 and RC5 | — | — | 0.6 | V | IOL = 14 mA, VDD = 4.5V -40°C ≤ TA ≤ +125°C IOL = 17 mA, VDD = 4.5V -40°C ≤ TA ≤ +85°C |
| | VOH | **Output High Voltage** | | | | | |
| D090 | | I/O Ports (excluding RC4, RC5) | VDD-0.7 | — | — | V | IOH = -2.5 mA, VDD = 4.5V -40°C ≤ TA ≤ +125°C IOH = -3 mA, VDD = 4.5V -40°C ≤ TA ≤ +85°C |
| | | I/O Ports RC4 and RC5 | VDD-0.7 | — | — | V | IOH = -5 mA, VDD = 4.5V -40°C ≤ TA ≤ +125°C IOH = -6 mA, VDD = 4.5V -40°C ≤ TA ≤ +85°C |

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** Negative current is defined as current sourced by the pin.

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

**3:** This specification applies to all weak pull-up pins, including the weak pull-up found on RA3/$\overline{MCLR}$. When RA3/$\overline{MCLR}$ is configured as $\overline{MCLR}$ Reset pin, the weak pull-up is always enabled.

**FIGURE 23-7: IDD TYPICAL, EC OSCILLATOR, HIGH-POWER MODE, PIC16F753 ONLY**

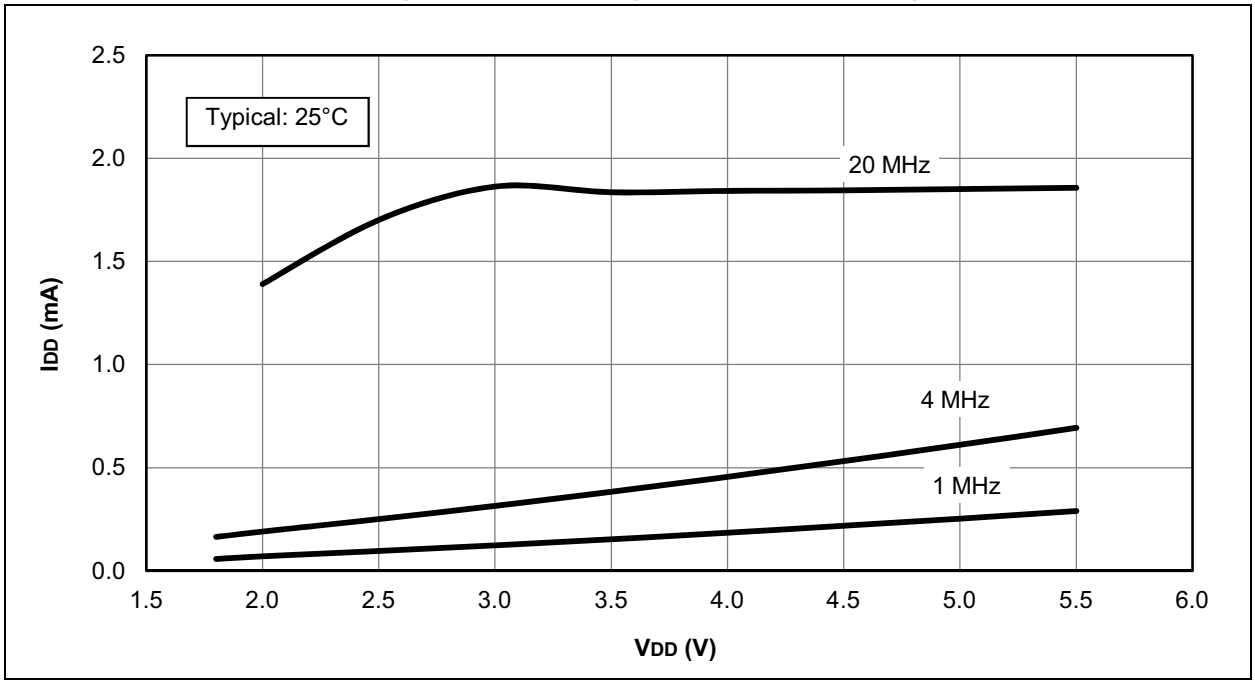



**FIGURE 23-8: IDD MAXIMUM, EC OSCILLATOR, HIGH-POWER MODE, PIC16F753 ONLY**

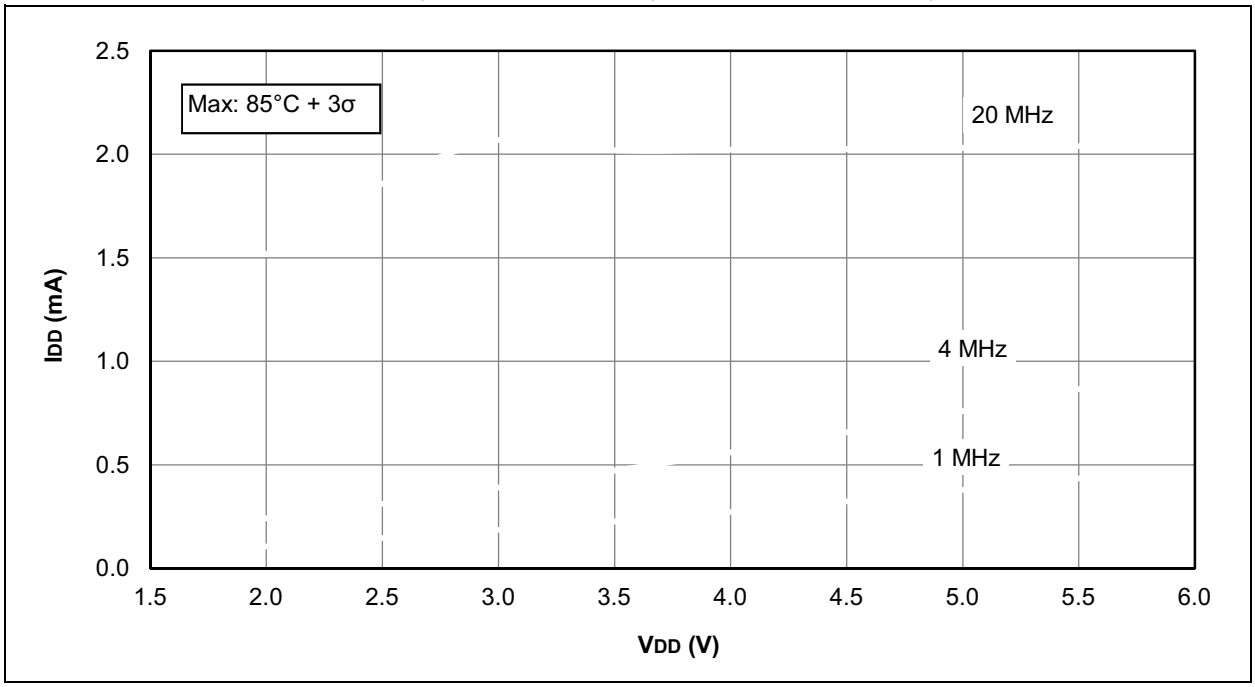


**FIGURE 23-20: IPD, BROWN-OUT RESET (BOR), PIC16HV753 ONLY**

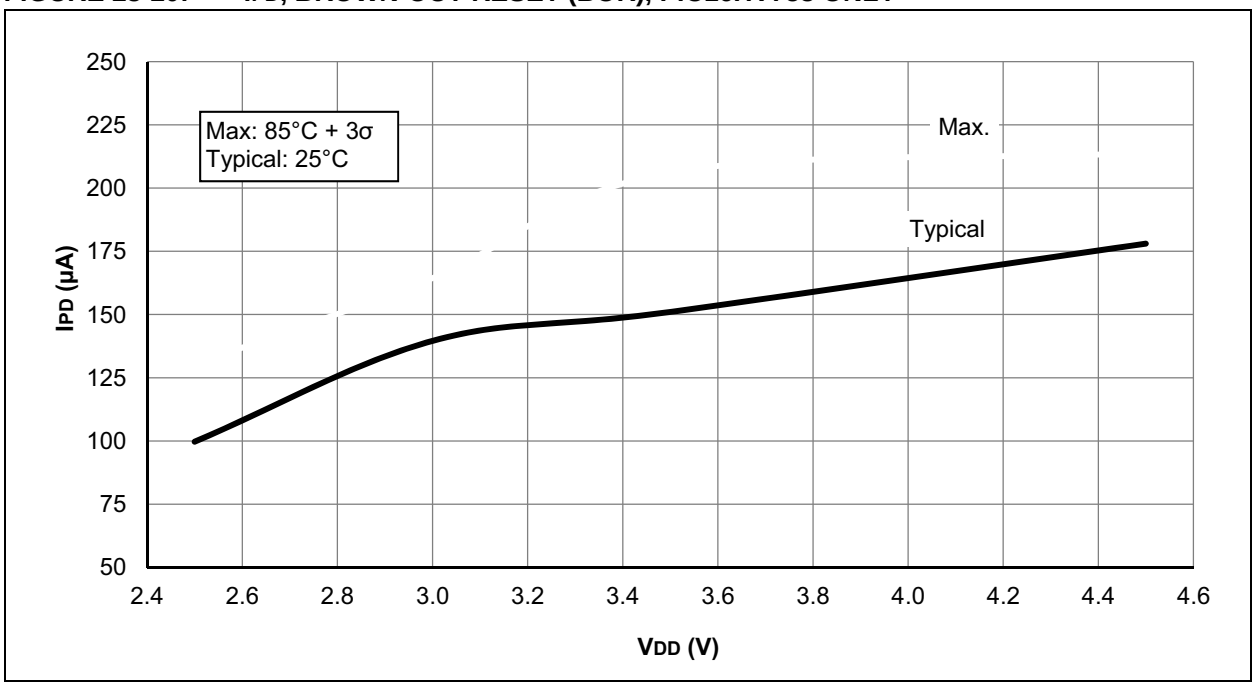
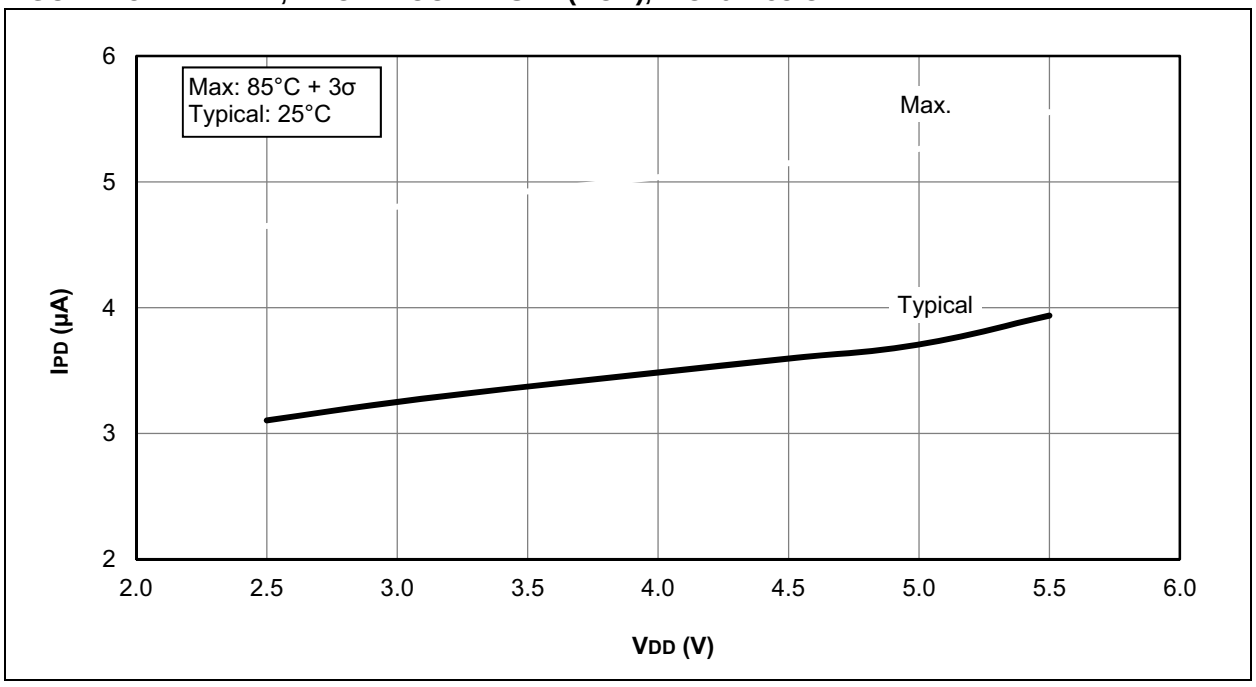
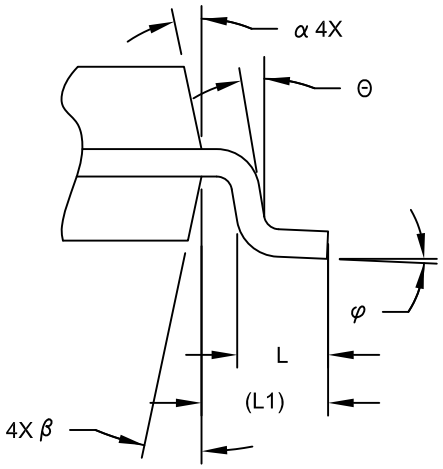


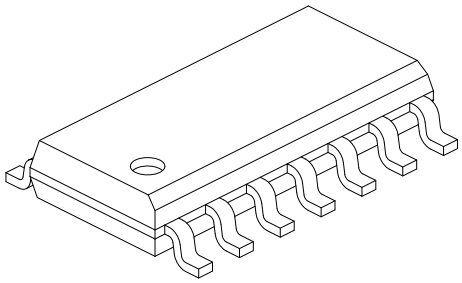

**FIGURE 23-21: IPD, BROWN-OUT RESET (BOR), PIC16F753 ONLY**



### 14-Lead Plastic Small Outline (SL) - Narrow, 3.90 mm Body [SOIC]

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



**VIEW C**

| Units | | MILLIMETERS | | |
|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX |
| Number of Pins | N | | 14 | |
| Pitch | e | | 1.27 BSC | |
| Overall Height | A | - | - | 1.75 |
| Molded Package Thickness | A2 | 1.25 | - | - |
| Standoff § | A1 | 0.10 | - | 0.25 |
| Overall Width | E | | 6.00 BSC | |
| Molded Package Width | E1 | | 3.90 BSC | |
| Overall Length | D | | 8.65 BSC | |
| Chamfer (Optional) | h | 0.25 | - | 0.50 |
| Foot Length | L | 0.40 | - | 1.27 |
| Footprint | L1 | | 1.04 REF | |
| Lead Angle | Θ | 0° | - | - |
| Foot Angle | φ | 0° | - | 8° |
| Lead Thickness | c | 0.10 | - | 0.25 |
| Lead Width | b | 0.31 | - | 0.51 |
| Mold Draft Angle Top | α | 5° | - | 15° |
| Mold Draft Angle Bottom | β | 5° | - | 15° |

**Notes:**

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic
3. Dimension D does not include mold flash, protrusions or gate burrs, which shall not exceed 0.15 mm per end. Dimension E1 does not include interlead flash or protrusion, which shall not exceed 0.25 mm per side.
4. Dimensioning and tolerancing per ASME Y14.5M

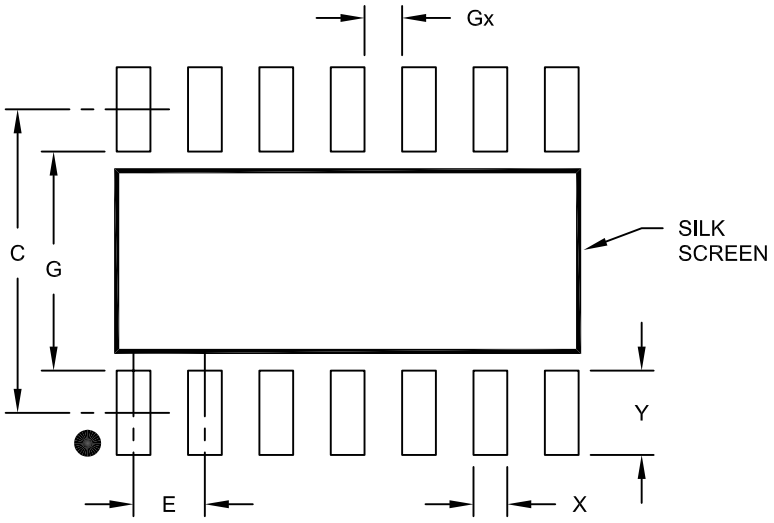
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

   REF: Reference Dimension, usually without tolerance, for information purposes only.
5. Datums A & B to be determined at Datum H.

Microchip Technology Drawing No. C04-065C Sheet 2 of 2

14-Lead Plastic Small Outline (SL) - Narrow, 3.90 mm Body [SOIC]

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




RECOMMENDED LAND PATTERN

| Units | | MILLIMETERS | | |
|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX |
| Contact Pitch | E | | 1.27 BSC | |
| Contact Pad Spacing | C | | 5.40 | |
| Contact Pad Width | X | | | 0.60 |
| Contact Pad Length | Y | | | 1.50 |
| Distance Between Pads | Gx | 0.67 | | |
| Distance Between Pads | G | 3.90 | | |

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2065A

## THE MICROCHIP WEBSITE

Microchip provides online support via our website at www.microchip.com. This website is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the website 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 website at www.microchip.com. Under "Support", 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

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 website at: http://www.microchip.com/support**