



Welcome to [E-XFL.COM](https://www.e-xfl.com)

### What is "[Embedded - Microcontrollers](#)"?

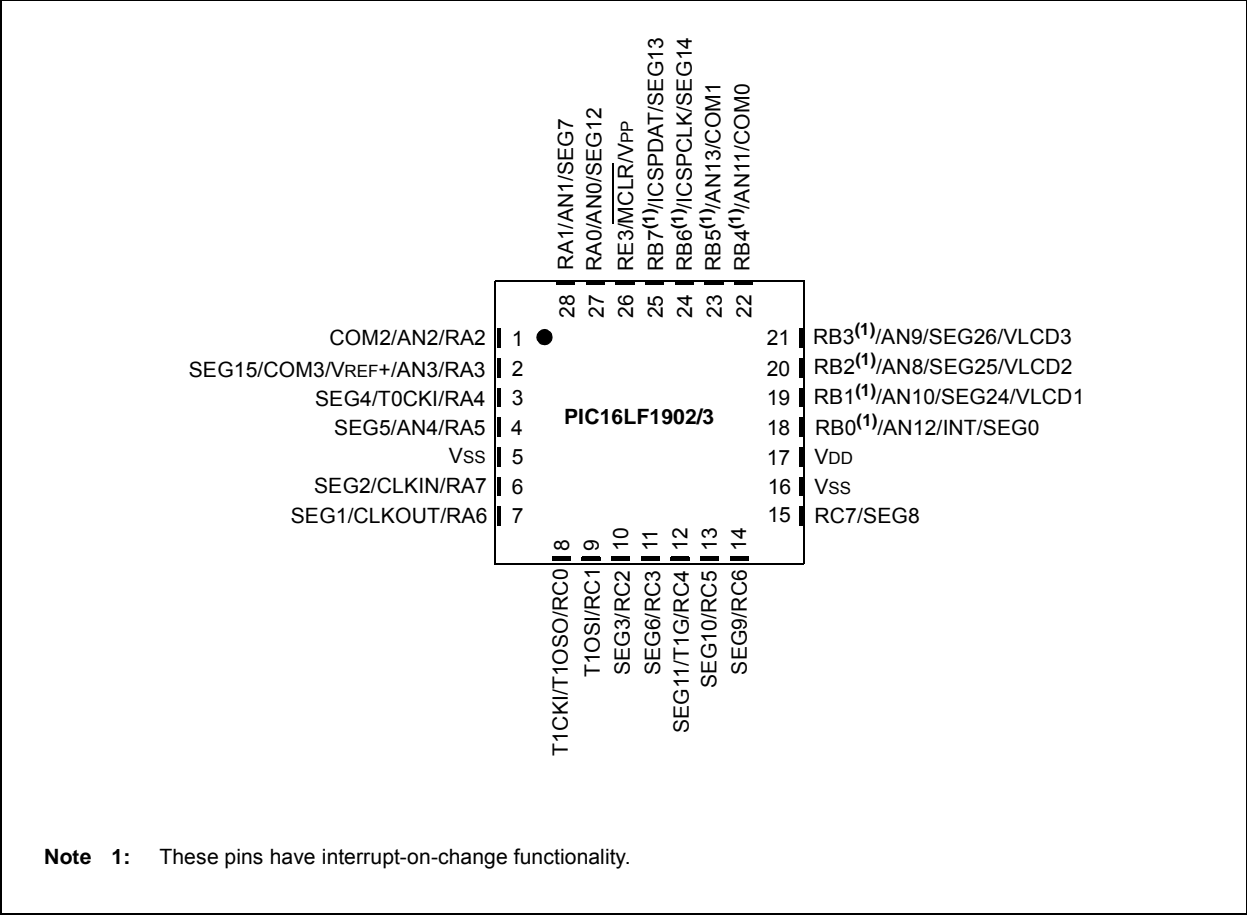
"[Embedded - Microcontrollers](#)" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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, LCD, POR, PWM, WDT
Number of I/O	25
Program Memory Size	3.5KB (2K x 14)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 11x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Through Hole
Package / Case	28-DIP (0.300", 7.62mm)
Supplier Device Package	28-SPDIP
Purchase URL	<a href="https://www.e-xfl.com/product-detail/microchip-technology/pic16lf1902-e-sp">https://www.e-xfl.com/product-detail/microchip-technology/pic16lf1902-e-sp</a>

FIGURE 2: 28-PIN UQFN



**TABLE 3-5: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)**

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
Bank 2											
10Ch	LATA	PORTA Data Latch								xxxx xxxx	uuuu uuuu
10Dh	LATB	PORTB Data Latch								xxxx xxxx	uuuu uuuu
10Eh	LATC	PORTC Data Latch								xxxx xxxx	uuuu uuuu
10Fh to 115h	—	Unimplemented								—	—
116h	BORCON	SBOREN	BORFS	—	—	—	—	—	BORRDY	10-- ---q	uu-- ---u
117h	FVRCON	FVREN	FVRRDY	TSEN	TSRNG	—	—	ADFVR1	ADFVR0	0q00 --00	0q00 --00
118h to 11Fh	—	Unimplemented								—	—
Bank 3											
18Ch	ANSELA	—	—	ANSA5	—	ANSA3	ANSA2	ANSA1	ANSA0	--1- 1111	--11 1111
18Dh	ANSELB	—	—	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0	--11 1111	--11 1111
18Eh	—	Unimplemented								—	—
18Fh	—	Unimplemented								—	—
190h	—	Unimplemented								—	—
191h	PMADRL	Program Memory Address Register Low Byte								0000 0000	0000 0000
192h	PMADRH	— <sup>(2)</sup>	Program Memory Address Register High Byte							1000 0000	1000 0000
193h	PMDATL	Program Memory Read Data Register Low Byte								xxxx xxxx	uuuu uuuu
194h	PMDATH	—	—	Program Memory Read Data Register High Byte						--xx xxxx	--uu uuuu
195h	PMCON1	— <sup>(2)</sup>	CFGFS	LWLO	FREE	WRERR	WREN	WR	RD	1000 x000	1000 q000
196h	PMCON2	Program Memory Control Register 2								0000 0000	0000 0000
197h to 19Fh	—	Unimplemented								—	—
Bank 4											
20Ch	—	Unimplemented								—	—
20Dh	WPUB	WPUB7	WPUB6	WPUB5	WPUB4	WPUB3	WPUB2	WPUB1	WPUB0	1111 1111	1111 1111
20Eh	—	Unimplemented								—	—
20Fh	—	Unimplemented								—	—
210h	WPUE	—	—	—	—	WPUE3	—	—	—	---- 1---	---- 1---
211h to 21Fh	—	Unimplemented								—	—
Bank 5											
28Ch — 29Fh	—	Unimplemented								—	—
Bank 6											
30Ch — 31Fh	—	Unimplemented								—	—

**Legend:** x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', r = reserved.  
Shaded locations are unimplemented, read as '0'.

**Note 1:** These registers can be addressed from any bank.  
**Note 2:** Unimplemented, read as '1'.

# PIC16LF1902/3

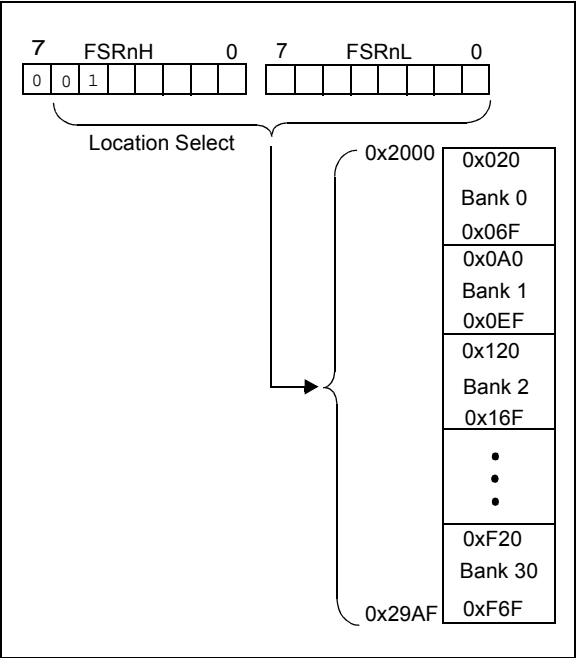
## 3.5.2 LINEAR DATA MEMORY

The linear data memory is the region from FSR address 0x2000 to FSR address 0x29AF. This region is a virtual region that points back to the 80-byte blocks of GPR memory in all the banks.

Unimplemented memory reads as 0x00. Use of the linear data memory region allows buffers to be larger than 80 bytes because incrementing the FSR beyond one bank will go directly to the GPR memory of the next bank.

The 16 bytes of common memory are not included in the linear data memory region.

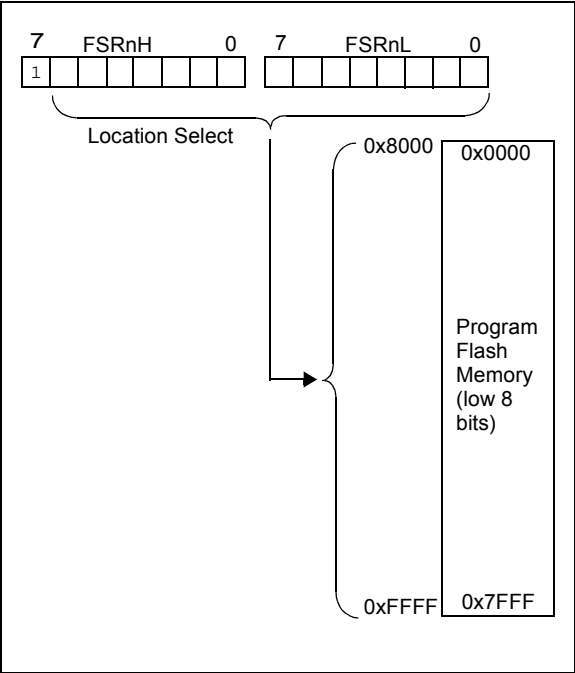
FIGURE 3-11: LINEAR DATA MEMORY MAP



## 3.5.3 PROGRAM FLASH MEMORY

To make constant data access easier, the entire program Flash memory is mapped to the upper half of the FSR address space. When the MSB of FSRnH is set, the lower 15 bits are the address in program memory which will be accessed through INDF. Only the lower eight bits of each memory location is accessible via INDF. Writing to the program Flash memory cannot be accomplished via the FSR/INDF interface. All instructions that access program Flash memory via the FSR/INDF interface will require one additional instruction cycle to complete.

FIGURE 3-12: PROGRAM FLASH MEMORY MAP



# PIC16LF1902/3

## 5.12 Power Control (PCON) Register

The PCON register bits are shown in Register 5-2.

The Power Control (PCON) register contains flag bits to differentiate between a:

- Power-on Reset ( $\overline{\text{POR}}$ )
- Brown-out Reset ( $\overline{\text{BOR}}$ )
- Reset Instruction Reset ( $\overline{\text{RI}}$ )
- MCLR Reset ( $\overline{\text{RMCLR}}$ )
- Watchdog Timer Reset ( $\overline{\text{RWDT}}$ )
- Stack Underflow Reset (STKUNF)
- Stack Overflow Reset (STKOVF)

### REGISTER 5-2: PCON: POWER CONTROL REGISTER

R/W/HS-0/q	R/W/HS-0/q	U-0	R/W/HC-1/q	R/W/HC-1/q	R/W/HC-1/q	R/W/HC-q/u	R/W/HC-q/u
STKOVF	STKUNF	—	$\overline{\text{RWDT}}$	$\overline{\text{RMCLR}}$	$\overline{\text{RI}}$	$\overline{\text{POR}}$	$\overline{\text{BOR}}$
bit 7							bit 0

#### Legend:

HC = Bit is cleared by hardware

HS = Bit is set by hardware

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

u = Bit is unchanged

x = Bit is unknown

-m/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	<b>STKOVF:</b> Stack Overflow Flag bit 1 = A Stack Overflow occurred 0 = A Stack Overflow has not occurred or set to '0' by firmware
bit 6	<b>STKUNF:</b> Stack Underflow Flag bit 1 = A Stack Underflow occurred 0 = A Stack Underflow has not occurred or set to '0' by firmware
bit 5	<b>Unimplemented:</b> Read as '0'
bit 4	<b><math>\overline{\text{RWDT}}</math>:</b> Watchdog Timer Reset Flag bit 1 = A Watchdog Timer Reset has not occurred or set to '1' by firmware 0 = A Watchdog Timer Reset has occurred (set to '0' in hardware when a Watchdog Timer Reset)
bit 3	<b><math>\overline{\text{RMCLR}}</math>:</b> MCLR Reset Flag bit 1 = A $\overline{\text{MCLR}}$ Reset has not occurred or set to '1' by firmware 0 = A MCLR Reset has occurred (set to '0' in hardware when a $\overline{\text{MCLR}}$ Reset occurs)
bit 2	<b><math>\overline{\text{RI}}</math>:</b> RESET Instruction Flag bit 1 = A RESET instruction has not been executed or set to '1' by firmware 0 = A RESET instruction has been executed (set to '0' in hardware upon executing a RESET instruction)
bit 1	<b><math>\overline{\text{POR}}</math>:</b> Power-on Reset Status bit 1 = No Power-on Reset occurred 0 = A Power-on Reset occurred (must be set in software after a Power-on Reset occurs)
bit 0	<b><math>\overline{\text{BOR}}</math>:</b> Brown-out Reset Status bit 1 = No Brown-out Reset occurred 0 = A Brown-out Reset occurred (must be set in software after a Power-on Reset or Brown-out Reset occurs)

## 7.6.4 PIR1 REGISTER

The PIR1 register contains the interrupt flag bits, as shown in Register 7-4.

**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 7-4: PIR1: PERIPHERAL INTERRUPT REQUEST REGISTER 1**

R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0	U-0	R/W-0/0
TMR1GIF	ADIF	—	—	—	—	—	TMR1IF
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

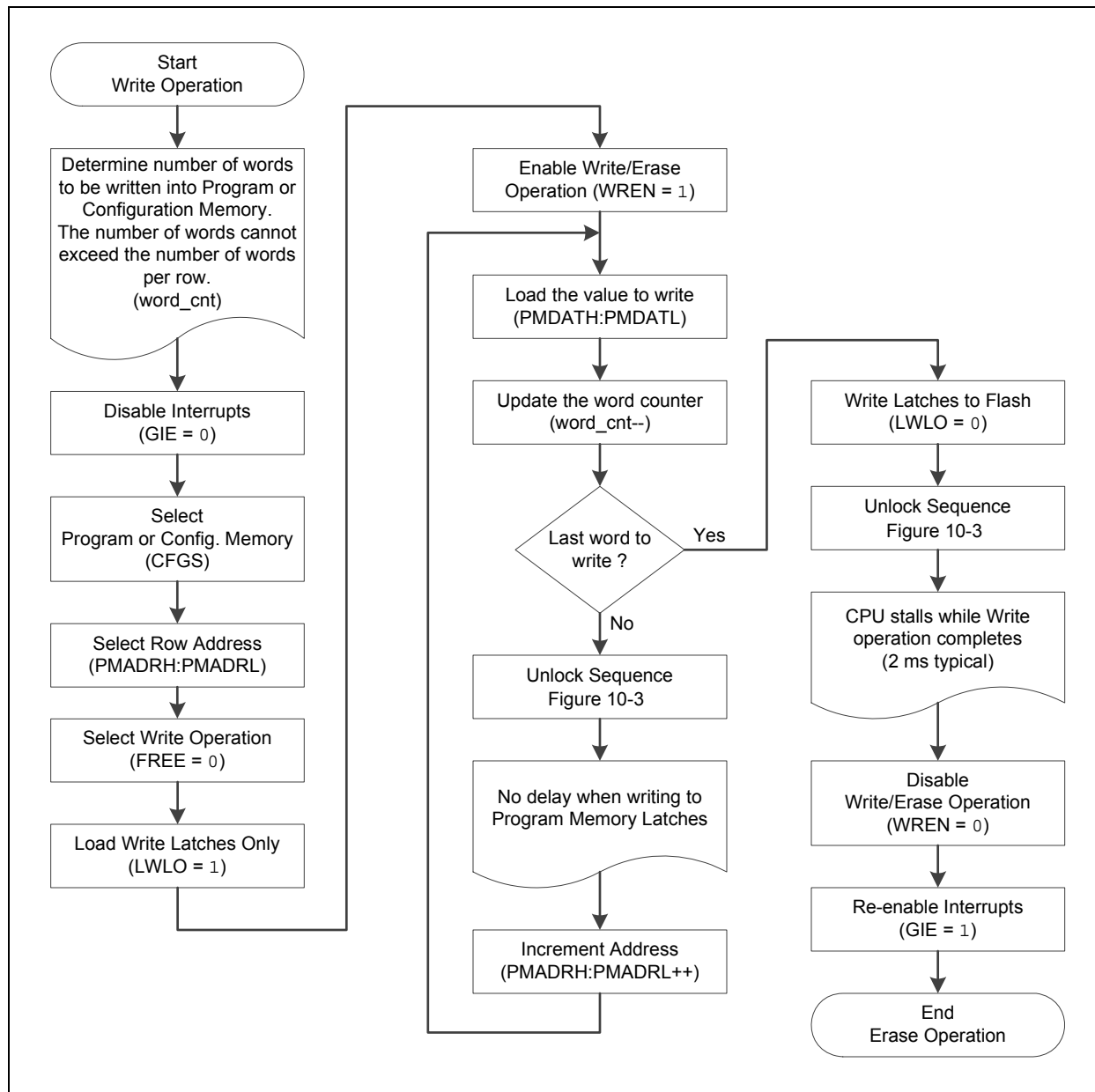
-n/n = Value at POR and BOR/Value at all other Resets

'1' = Bit is set

'0' = Bit is cleared

- bit 7      **TMR1GIF:** Timer1 Gate Interrupt Flag bit  
             1 = Interrupt is pending  
             0 = Interrupt is not pending
- bit 6      **ADIF:** A/D Converter Interrupt Flag bit  
             1 = Interrupt is pending  
             0 = Interrupt is not pending
- bit 5-1    **Unimplemented:** Read as '0'
- bit 0      **TMR1IF:** Timer1 Overflow Interrupt Flag bit  
             1 = Interrupt is pending  
             0 = Interrupt is not pending

**FIGURE 10-6: FLASH PROGRAM MEMORY WRITE FLOWCHART**



# PIC16LF1902/3

## REGISTER 11-10: PORTC: PORTC REGISTER

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0
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

bit 7-0                  **RC<7:0>**: PORTC General Purpose I/O Pin bits<sup>(1)</sup>  
1 = Port pin is  $\geq V_{IH}$   
0 = Port pin is  $\leq V_{IL}$

**Note 1:** Writes to PORTC are actually written to corresponding LATC register. Reads from PORTC register is return of actual I/O pin values.

## REGISTER 11-11: TRISC: PORTC TRI-STATE REGISTER

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0
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

bit 7-0                  **TRISC<7:0>**: PORTC Tri-State Control bits<sup>(1)</sup>  
1 = PORTC pin configured as an input (tri-stated)  
0 = PORTC pin configured as an output

## REGISTER 11-12: LATC: PORTC DATA LATCH REGISTER

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
LATC7	LATC6	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0
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

bit 7-0                  **LATC<7:0>**: PORTC Output Latch Value bits<sup>(1)</sup>

**Note 1:** Writes to PORTC are actually written to corresponding LATC register. Reads from PORTC register is return of actual I/O pin values.



## 12.0 INTERRUPT-ON-CHANGE

The PORTB pins can be configured to operate as Interrupt-on-Change (IOC) pins. An interrupt can be generated by detecting a signal that has either a rising edge or a falling edge. Any individual PORTB pin, or combination of PORTB pins, can be configured to generate an interrupt. The interrupt-on-change module has the following features:

- Interrupt-on-Change enable (Master Switch)
- Individual pin configuration
- Rising and falling edge detection
- Individual pin interrupt flags

Figure 12-1 is a block diagram of the IOC module.

### 12.1 Enabling the Module

To allow individual PORTB pins to generate an interrupt, the IOCIE bit of the INTCON register must be set. If the IOCIE bit is disabled, the edge detection on the pin will still occur, but an interrupt will not be generated.

### 12.2 Individual Pin Configuration

For each PORTB pin, a rising edge detector and a falling edge detector are present. To enable a pin to detect a rising edge, the associated IOCBP<sub>x</sub> bit of the IOCBP register is set. To enable a pin to detect a falling edge, the associated IOCBN<sub>x</sub> bit of the IOCBN register is set.

A pin can be configured to detect rising and falling edges simultaneously by setting both the IOCBP<sub>x</sub> bit and the IOCBN<sub>x</sub> bit of the IOCBP and IOCBN registers, respectively.

## 12.3 Interrupt Flags

The IOCBF<sub>x</sub> bits located in the IOCBF register are status flags that correspond to the interrupt-on-change pins of PORTB. If an expected edge is detected on an appropriately enabled pin, then the status flag for that pin will be set, and an interrupt will be generated if the IOCIE bit is set. The IOCIF bit of the INTCON register reflects the status of all IOCBF<sub>x</sub> bits.

### 12.4 Clearing Interrupt Flags

The individual status flags, (IOCBF<sub>x</sub> bits), can be cleared by resetting them to zero. If another edge is detected during this clearing operation, the associated status flag will be set at the end of the sequence, regardless of the value actually being written.

In order to ensure that no detected edge is lost while clearing flags, only AND operations masking out known changed bits should be performed. The following sequence is an example of what should be performed.

#### EXAMPLE 12-1:

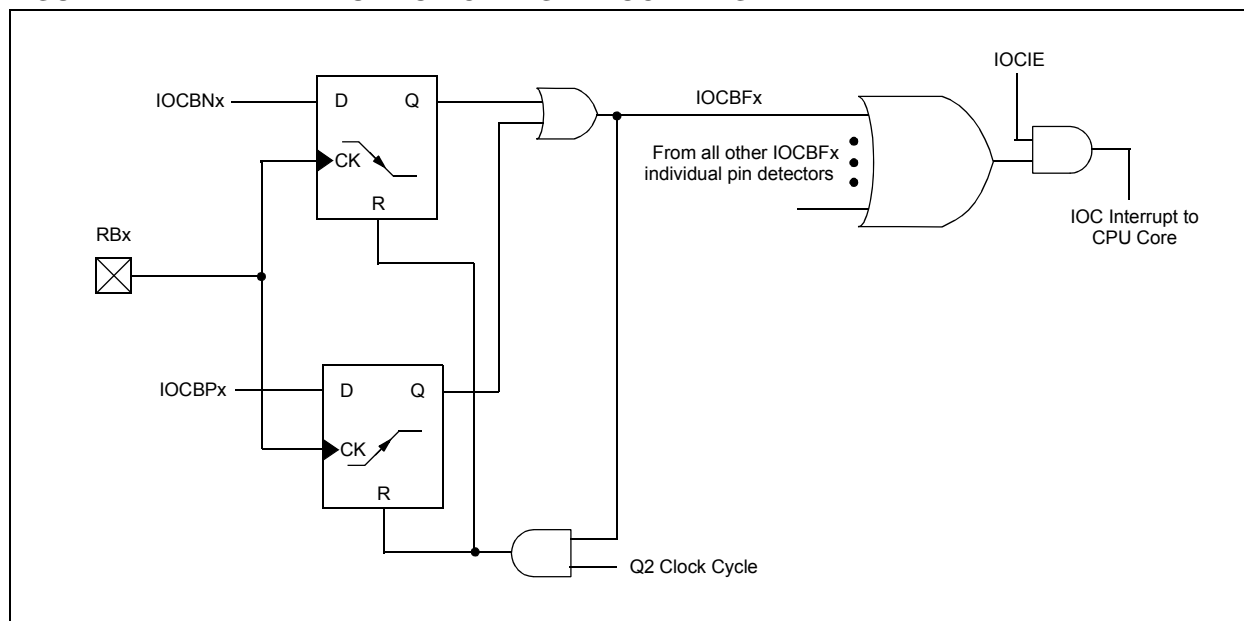
```
MOVLW  0xff
XORWF  IOCBF, W
ANDWF  IOCBF, F
```

### 12.5 Operation in Sleep

The interrupt-on-change interrupt sequence will wake the device from Sleep mode, if the IOCIE bit is set.

If an edge is detected while in Sleep mode, the IOCBF register will be updated prior to the first instruction executed out of Sleep.

**FIGURE 12-1: INTERRUPT-ON-CHANGE BLOCK DIAGRAM**



# PIC16LF1902/3

**TABLE 12-1: SUMMARY OF REGISTERS ASSOCIATED WITH INTERRUPT-ON-CHANGE**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELB	—	—	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0	94
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	60
IOCBF	IOCBF7	IOCBF6	IOCBF5	IOCBF4	IOCBF3	IOCBF2	IOCBF1	IOCBF0	101
IOCBN	IOCBN7	IOCBN6	IOCBN5	IOCBN4	IOCBN3	IOCBN2	IOCBN1	IOCBN0	101
IOCBP	IOCBP7	IOCBP6	IOCBP5	IOCBP4	IOCBP3	IOCBP2	IOCBP1	IOCBP0	101
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	93

**Legend:** — = unimplemented location, read as '0'. Shaded cells are not used by interrupt-on-change.

## 18.0 LIQUID CRYSTAL DISPLAY (LCD) DRIVER MODULE

The Liquid Crystal Display (LCD) Driver module generates the timing control to drive a static or multiplexed LCD panel. In the PIC16LF1902/3 device, the module drives the panels of up to four commons and up to 72 total segments. The LCD module also provides control of the LCD pixel data.

The LCD Driver module supports:

- Direct driving of LCD panel
- Three LCD clock sources with selectable prescaler
- Up to four common pins:
  - Static (1 common)
  - 1/2 multiplex (2 commons)
  - 1/3 multiplex (3 commons)
  - 1/4 multiplex (4 commons)
- 19 Segment pins
- Static, 1/2 or 1/3 LCD Bias

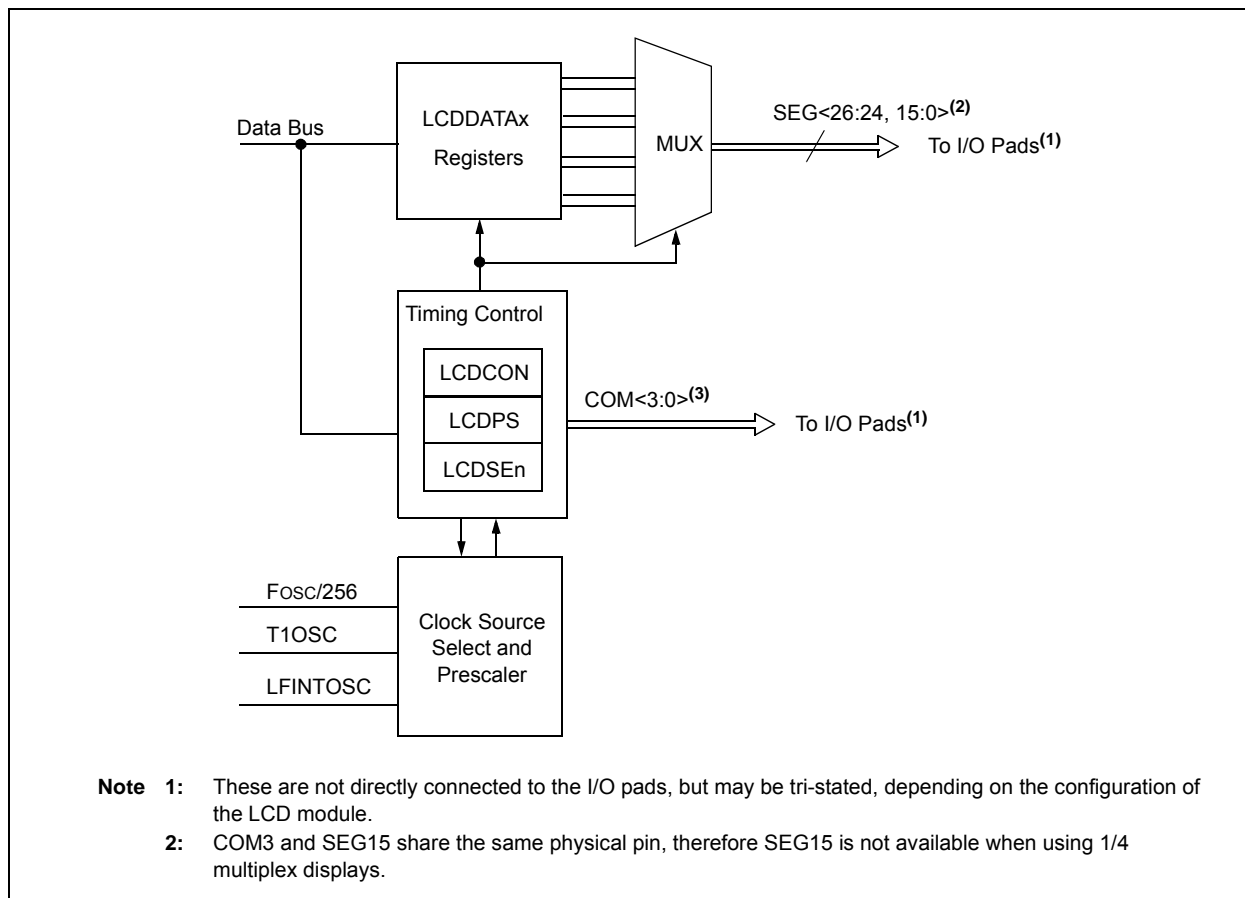
**Note:** COM3 and SEG15 share the same physical pin on the PIC16LF1902/3, therefore SEG15 is not available when using 1/4 multiplex displays.

## 18.1 LCD Registers

The module contains the following registers:

- LCD Control register (LCDCON)
- LCD Phase register (LCDPS)
- LCD Reference Ladder register (LCDRL)
- LCD Contrast Control register (LDCST)
- LCD Reference Voltage Control register (LCDREF)
- Up to 3 LCD Segment Enable registers (LCDSEn)
- Up to 12 LCD data registers (LCDDATAn)

**FIGURE 18-1: LCD DRIVER MODULE BLOCK DIAGRAM**



# PIC16LF1902/3

## REGISTER 18-4: LCDCST: LCD CONTRAST CONTROL REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	—	LCDCST<2:0>		
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

C = Only clearable bit

bit 7-3

**Unimplemented:** Read as '0'

bit 2-0

**LCDCST<2:0>:** LCD Contrast Control bits

Selects the resistance of the LCD contrast control resistor ladder

Bit Value = Resistor ladder

000 = Minimum Resistance (Maximum contrast). Resistor ladder is shorted.

001 = Resistor ladder is at 1/7th of maximum resistance

010 = Resistor ladder is at 2/7th of maximum resistance

011 = Resistor ladder is at 3/7th of maximum resistance

100 = Resistor ladder is at 4/7th of maximum resistance

101 = Resistor ladder is at 5/7th of maximum resistance

110 = Resistor ladder is at 6/7th of maximum resistance

111 = Resistor ladder is at maximum resistance (Minimum contrast).

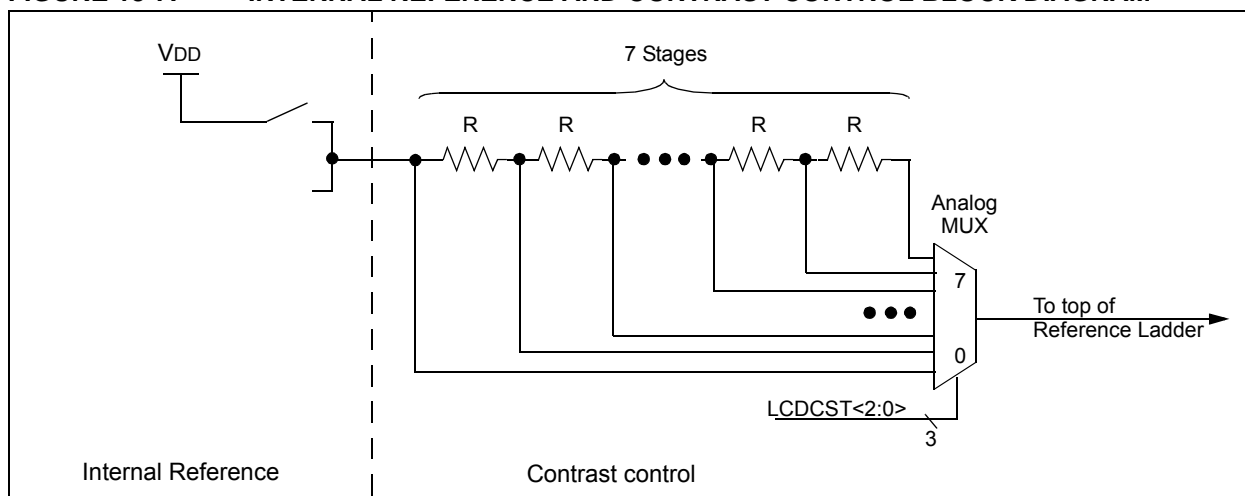
## 18.4.4 CONTRAST CONTROL

The LCD contrast control circuit consists of a 7-tap resistor ladder, controlled by the LCDCST bits. Refer to Figure 18-7.

The contrast control circuit is used to decrease the output voltage of the signal source by a total of approximately 10%, when  $LCDCST = 111$ .

Whenever the LCD module is inactive ( $LCDA = 0$ ), the contrast control ladder will be turned off (open).

**FIGURE 18-7: INTERNAL REFERENCE AND CONTRAST CONTROL BLOCK DIAGRAM**



## 18.4.5 INTERNAL REFERENCE

Under firmware control, an internal reference for the LCD bias voltages can be enabled. When enabled, the source of this voltage can be VDD. When no internal reference is selected, the LCD contrast control circuit is disabled and LCD bias must be provided externally.

Whenever the LCD module is inactive ( $LCDA = 0$ ), the internal reference will be turned off.

When the internal reference is enabled and the Fixed Voltage Reference is selected, the LCDIRI bit can be used to minimize power consumption by tying into the LCD reference ladder automatic power mode switching. When  $LCDIRI = 1$  and the LCD reference ladder is in Power mode 'B', the LCD internal FVR buffer is disabled.

**Note:** The LCD module automatically turns on the Fixed Voltage Reference when needed.

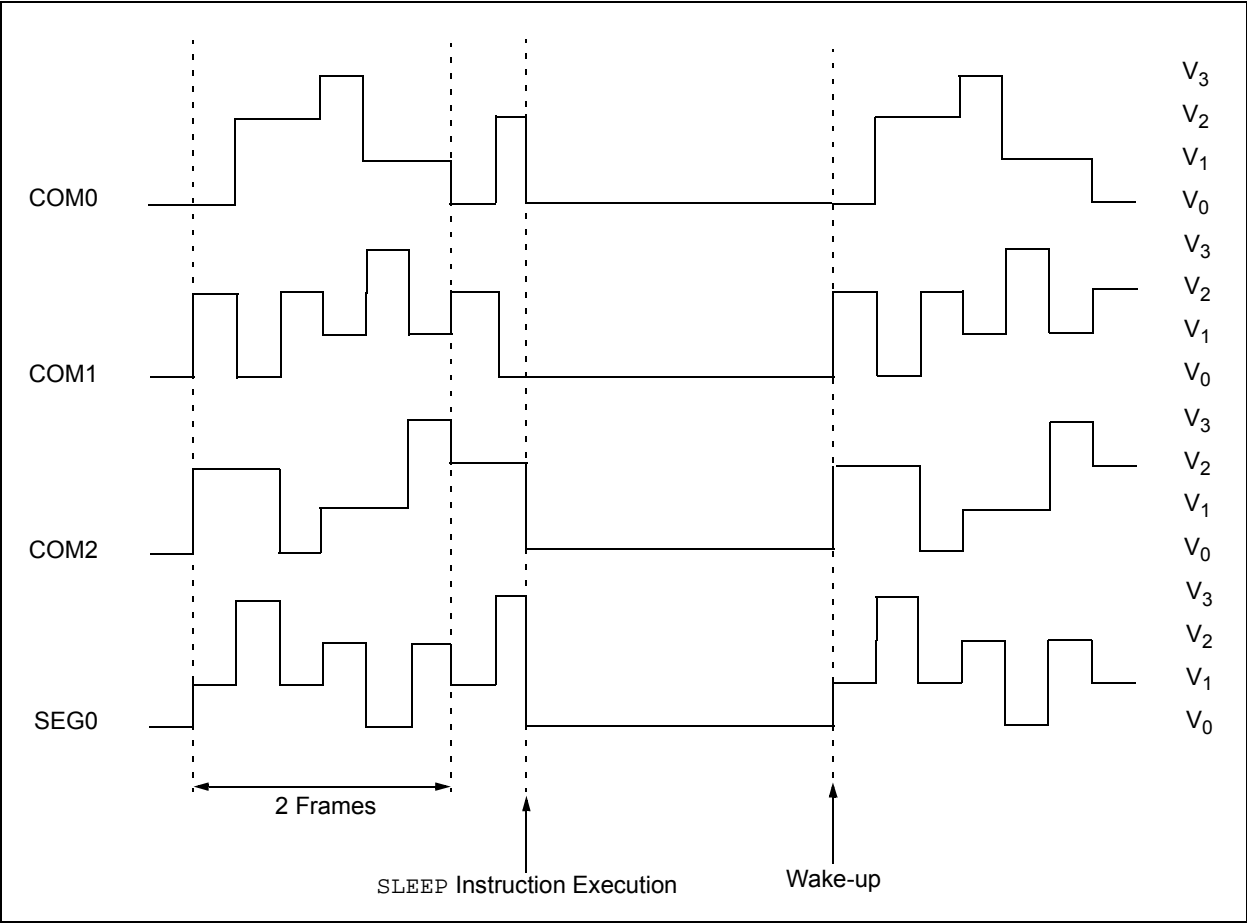
## 18.4.6 VLCD<3:1> PINS

The VLCD<3:1> pins provide the ability for an external LCD bias network to be used instead of the internal ladder. Use of the VLCD<3:1> pins does not prevent use of the internal ladder. Each VLCD pin has an independent control in the LCDREF register (Register 18-3), allowing access to any or all of the LCD Bias signals. This architecture allows for maximum flexibility in different applications.

For example, the VLCD<3:1> pins may be used to add capacitors to the internal reference ladder, increasing the drive capacity.

For applications where the internal contrast control is insufficient, the firmware can choose to only enable the VLCD3 pin, allowing an external contrast control circuit to use the internal reference divider.

FIGURE 18-20: SLEEP ENTRY/EXIT WHEN SLPEN = 1



# PIC16LF1902/3

---

## **SWAPF**      **Swap Nibbles in f**

---

Syntax:      [ *label* ]   SWAPF f,d

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

Operation:       $(f<3:0>) \rightarrow (\text{destination}<7:4>)$ ,  
                     $(f<7:4>) \rightarrow (\text{destination}<3:0>)$

Status Affected:      None

Description:      The upper and lower nibbles of register 'f' are exchanged. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed in register 'f'.

## **TRIS**      **Load TRIS Register with W**

---

Syntax:      [ *label* ]   TRIS f

Operands:       $5 \leq f \leq 7$

Operation:       $(W) \rightarrow \text{TRIS register 'f'}$

Status Affected:      None

Description:      Move data from W register to TRIS register.  
                    When 'f' = 5, TRISA is loaded.  
                    When 'f' = 6, TRISB is loaded.  
                    When 'f' = 7, TRISC is loaded.

## **XORLW**      **Exclusive OR literal with W**

---

Syntax:      [ *label* ]   XORLW k

Operands:       $0 \leq k \leq 255$

Operation:       $(W) .XOR. k \rightarrow (W)$

Status Affected:      Z

Description:      The contents of the W register are XOR'ed with the 8-bit literal 'k'. The result is placed in the W register.

## **XORWF**      **Exclusive OR W with f**

---

Syntax:      [ *label* ]   XORWF f,d

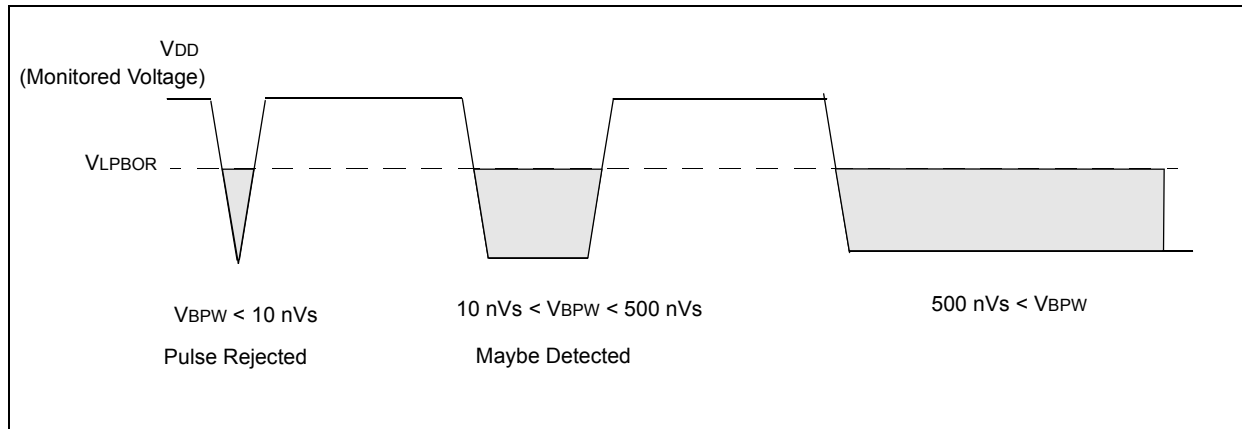
Operands:       $0 \leq f \leq 127$   
                     $d \in [0,1]$

Operation:       $(W) .XOR. (f) \rightarrow (\text{destination})$

Status Affected:      Z

Description:      Exclusive OR the contents of the W register with 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'.

**FIGURE 21-8: MINIMUM PULSE WIDTH FOR LPBOR DETECTION**





## 23.0 DEVELOPMENT SUPPORT

The PIC® microcontrollers (MCU) and dsPIC® digital signal controllers (DSC) are supported with a full range of software and hardware development tools:

- Integrated Development Environment
  - MPLAB® X IDE Software
- Compilers/Assemblers/Linkers
  - MPLAB XC Compiler
  - MPASM™ Assembler
  - MPLINK™ Object Linker/  
MPLIB™ Object Librarian
  - MPLAB Assembler/Linker/Librarian for  
Various Device Families
- Simulators
  - MPLAB X SIM Software Simulator
- Emulators
  - MPLAB REAL ICE™ In-Circuit Emulator
- In-Circuit Debuggers/Programmers
  - MPLAB ICD 3
  - PICKit™ 3
- Device Programmers
  - MPLAB PM3 Device Programmer
- Low-Cost Demonstration/Development Boards,  
Evaluation Kits and Starter Kits
- Third-party development tools

## 23.1 MPLAB X Integrated Development Environment Software

The MPLAB X IDE is a single, unified graphical user interface for Microchip and third-party software, and hardware development tool that runs on Windows®, Linux and Mac OS® X. Based on the NetBeans IDE, MPLAB X IDE is an entirely new IDE with a host of free software components and plug-ins for high-performance application development and debugging. Moving between tools and upgrading from software simulators to hardware debugging and programming tools is simple with the seamless user interface.

With complete project management, visual call graphs, a configurable watch window and a feature-rich editor that includes code completion and context menus, MPLAB X IDE is flexible and friendly enough for new users. With the ability to support multiple tools on multiple projects with simultaneous debugging, MPLAB X IDE is also suitable for the needs of experienced users.

### Feature-Rich Editor:

- Color syntax highlighting
- Smart code completion makes suggestions and provides hints as you type
- Automatic code formatting based on user-defined rules
- Live parsing

### User-Friendly, Customizable Interface:

- Fully customizable interface: toolbars, toolbar buttons, windows, window placement, etc.
- Call graph window

### Project-Based Workspaces:

- Multiple projects
- Multiple tools
- Multiple configurations
- Simultaneous debugging sessions

### File History and Bug Tracking:

- Local file history feature
- Built-in support for Bugzilla issue tracker

## 23.2 MPLAB XC Compilers

The MPLAB XC Compilers are complete ANSI C compilers for all of Microchip's 8, 16, and 32-bit MCU and DSC devices. These compilers provide powerful integration capabilities, superior code optimization and ease of use. MPLAB XC Compilers run on Windows, Linux or MAC OS X.

For easy source level debugging, the compilers provide debug information that is optimized to the MPLAB X IDE.

The free MPLAB XC Compiler editions support all devices and commands, with no time or memory restrictions, and offer sufficient code optimization for most applications.

MPLAB XC Compilers include an assembler, linker and utilities. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. MPLAB XC Compiler uses the assembler to produce its object file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command-line interface
- Rich directive set
- Flexible macro language
- MPLAB X IDE compatibility

## 23.3 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for PIC10/12/16/18 MCUs.

The MPASM Assembler generates relocatable object files for the MPLINK Object Linker, Intel® standard HEX files, MAP files to detail memory usage and symbol reference, absolute LST files that contain source lines and generated machine code, and COFF files for debugging.

The MPASM Assembler features include:

- Integration into MPLAB X IDE projects
- User-defined macros to streamline assembly code
- Conditional assembly for multipurpose source files
- Directives that allow complete control over the assembly process

## 23.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler. It can link relocatable objects from precompiled libraries, using directives from a linker script.

The MPLIB Object Librarian manages the creation and modification of library files of precompiled code. When a routine from a library is called from a source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications.

The object linker/librarian features include:

- Efficient linking of single libraries instead of many smaller files
- Enhanced code maintainability by grouping related modules together
- Flexible creation of libraries with easy module listing, replacement, deletion and extraction

## 23.5 MPLAB Assembler, Linker and Librarian for Various Device Families

MPLAB Assembler produces relocatable machine code from symbolic assembly language for PIC24, PIC32 and dsPIC DSC devices. MPLAB XC Compiler uses the assembler to produce its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

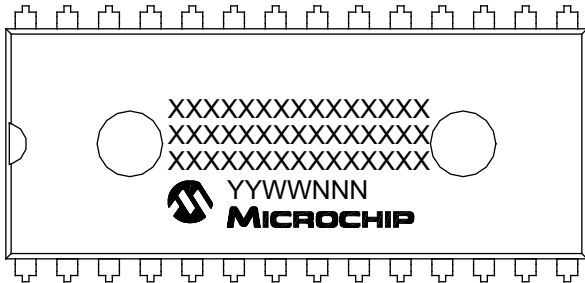
- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command-line interface
- Rich directive set
- Flexible macro language
- MPLAB X IDE compatibility

# PIC16LF1902/3

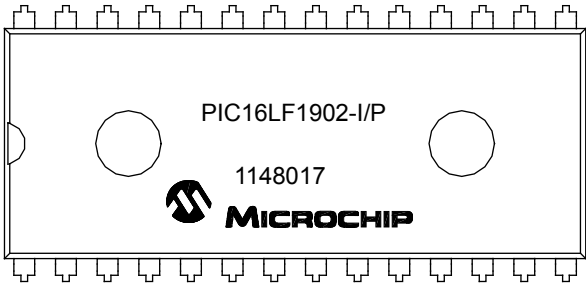
## 24.0 PACKAGING INFORMATION

### 24.1 Package Marking Information

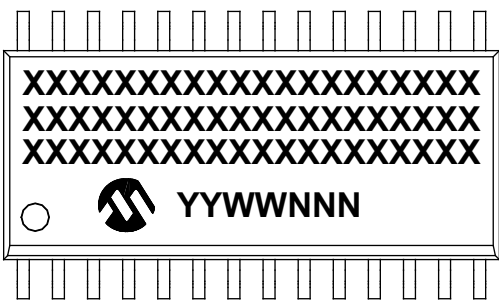
28-Lead PDIP (600 mil)



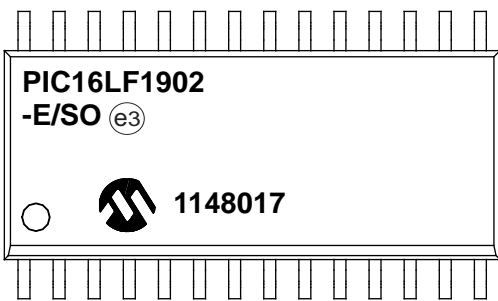
Example



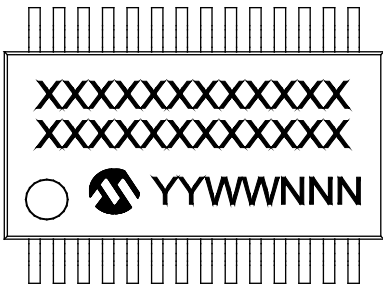
28-Lead SOIC (7.50 mm)



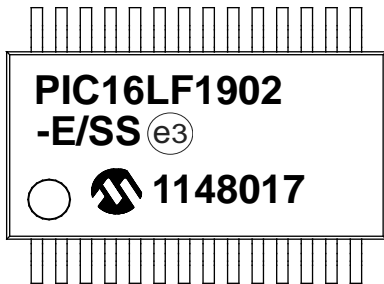
Example



28-Lead SSOP (5.30 mm)



Example

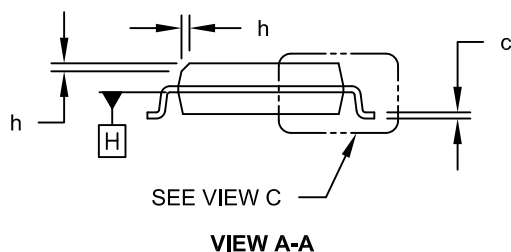
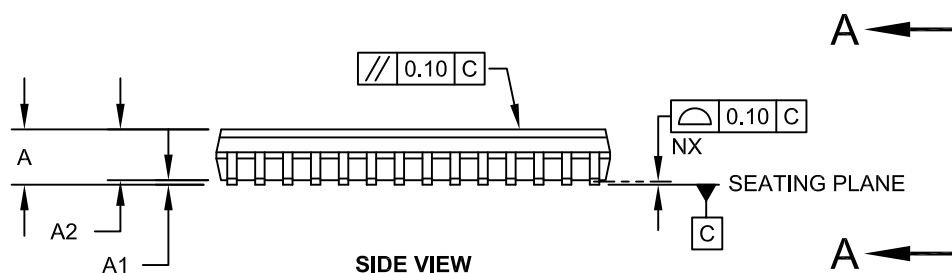
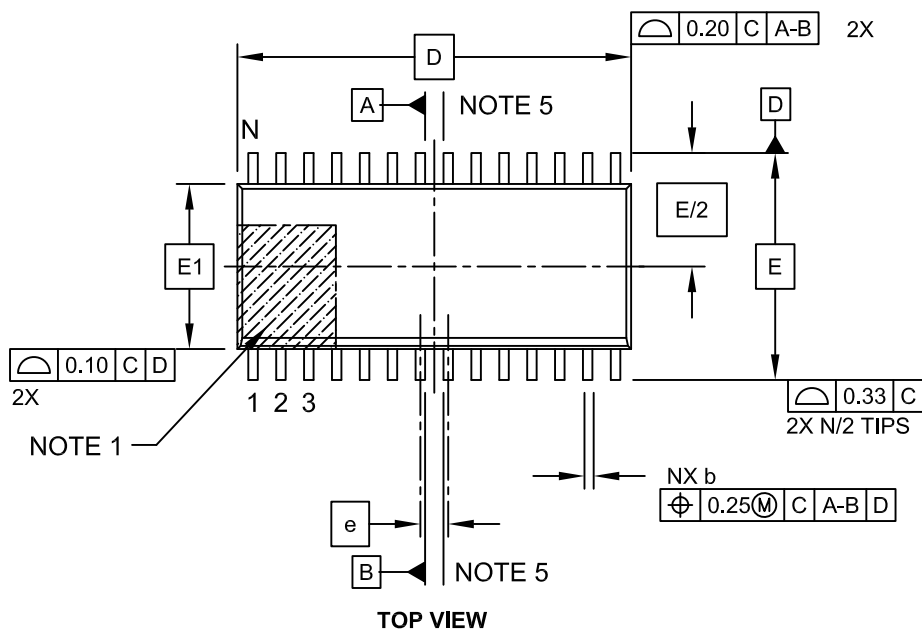


<b>Legend:</b>	XX...X	Customer-specific information
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week '01')
	NNN	Alphanumeric traceability code
	(e3)	Pb-free JEDEC® designator for Matte Tin (Sn)
	*	This package is Pb-free. The Pb-free JEDEC® designator (e3) can be found on the outer packaging for this package.

**Note:** In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.

## 28-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

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



---

**Note the following details of the code protection feature on Microchip devices:**

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

---

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

*Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELoq® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.*

**QUALITY MANAGEMENT SYSTEM**  
**CERTIFIED BY DNV**  
**== ISO/TS 16949 ==**

### Trademarks

The Microchip name and logo, the Microchip logo, AnyRate, dsPIC, FlashFlex, flexPWR, Helder, JukeBlox, KeeLoq, KeeLoq logo, Klear, LANCheck, LINK MD, MediaLB, MOST, MOST logo, MPLAB, OptoLyzer, PIC, PICSTART, PIC32 logo, RightTouch, SpyNIC, SST, SST Logo, SuperFlash and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

ClockWorks, The Embedded Control Solutions Company, ETHERSYNCH, Hyper Speed Control, HyperLight Load, IntelliMOS, mTouch, Precision Edge, and QUIET-WIRE are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, BodyCom, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, EtherGREEN, In-Circuit Serial Programming, ICSP, Inter-Chip Connectivity, JitterBlocker, KlearNet, KlearNet logo, MiWi, motorBench, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PureSilicon, RightTouch logo, REAL ICE, Ripple Blocker, Serial Quad I/O, SQL, SuperSwitcher, SuperSwitcher II, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

GestIC is a registered trademarks of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

© 2011-2016, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

ISBN: 978-1-5224-0546-7