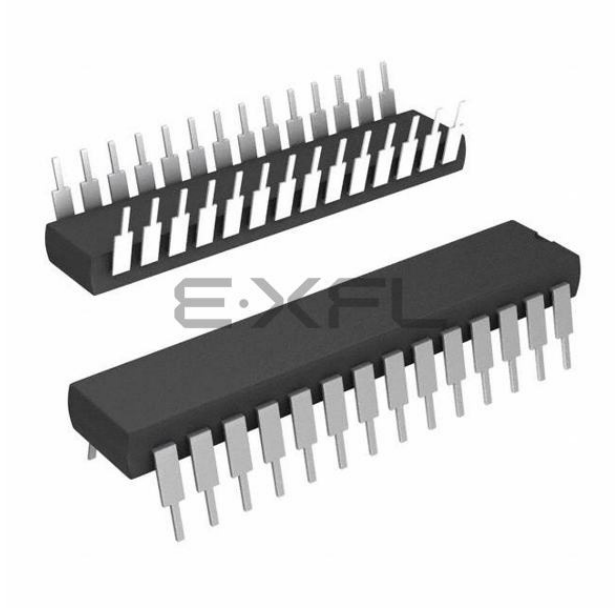


Welcome to E-XFL.COM

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, 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 ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	28-DIP (0.300", 7.62mm)
Supplier Device Package	28-SPDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lf1902-i-sp

3.0 MEMORY ORGANIZATION

These devices contain the following types of memory:

- Program Memory
  - Configuration Words
  - Device ID
  - User ID
  - Flash Program Memory
- Data Memory
  - Core Registers
  - Special Function Registers
  - General Purpose RAM
  - Common RAM

The following features are associated with access and control of program memory and data memory:

- PCL and PCLATH
- Stack
- Indirect Addressing

3.1 Program Memory Organization

The enhanced mid-range core has a 15-bit program counter capable of addressing 32K x 14 program memory space. Table 3-1 shows the memory sizes implemented for the PIC16LF1902/3 family. Accessing a location above these boundaries will cause a wrap-around within the implemented memory space. The Reset vector is at 0000h and the interrupt vector is at 0004h (see Figures 3-1, and 3-2).

TABLE 3-1: DEVICE SIZES AND ADDRESSES

Device	Program Memory Space (Words)	Last Program Memory Address	High-Endurance Flash Memory Address Range(1)
PIC16LF1902	2,048	07FFh	0780h-07FFh
PIC16LF1903	4,096	0FFFh	0F80h-0FFFh

Note 1: High-endurance Flash applies to low byte of each address in the range.

TABLE 3-3: PIC16LF1902/3 MEMORY MAP (CONTINUED)

BANK 8		BANK 9		BANK 10		BANK 11		BANK 12		BANK 13		BANK 14	
400h	Core Registers (Table 3-2)	480h	Core Registers (Table 3-2)	500h	Core Registers (Table 3-2)	580h	Core Registers (Table 3-2)	600h	Core Registers (Table 3-2)	680h	Core Registers (Table 3-2)	700h	Core Registers (Table 3-2)
40Bh		48Bh		50Bh		58Bh		60Bh		68Bh		70Bh	
40Ch	Unimplemented Read as '0'	48Ch	Unimplemented Read as '0'	50Ch	Unimplemented Read as '0'	58Ch	Unimplemented Read as '0'	60Ch	Unimplemented Read as '0'	68Ch	Unimplemented Read as '0'	70Ch	Unimplemented Read as '0'
46Fh		4EFh		56Fh		5EFh		66Fh		6EFh		76Fh	
470h	Common RAM (Accesses 70h – 7Fh)	4F0h	Common RAM (Accesses 70h – 7Fh)	570h	Common RAM (Accesses 70h – 7Fh)	5F0h	Common RAM (Accesses 70h – 7Fh)	670h	Common RAM (Accesses 70h – 7Fh)	6F0h	Common RAM (Accesses 70h – 7Fh)	770h	Common RAM (Accesses 70h – 7Fh)
47Fh		4FFh		57Fh		5FFh		67Fh		6FFh		77Fh	

BANK 16		BANK 17		BANK 18		BANK 19		BANK 20		BANK 21		BANK 22		BANK 23	
800h	Core Registers (Table 3-2)Table 3-2	880h	Core Registers (Table 3-2)	900h	Core Registers (Table 3-2)	980h	Core Registers (Table 3-2)	A00h	Core Registers (Table 3-2)	A80h	Core Registers (Table 3-2)	B00h	Core Registers (Table 3-2)	B80h	Core Registers (Table 3-2)
80Bh		88Bh		90Bh		98Bh		A0Bh		A8Bh		B0Bh		B8Bh	
80Ch	Unimplemented Read as '0'	88Ch	Unimplemented Read as '0'	90Ch	Unimplemented Read as '0'	98Ch	Unimplemented Read as '0'	A0Ch	Unimplemented Read as '0'	A8Ch	Unimplemented Read as '0'	B0Ch	Unimplemented Read as '0'	B8Ch	Unimplemented Read as '0'
86Fh		8EFh		96Fh		9EFh		A6Fh		AEFh		B6Fh		BEFh	
870h	Common RAM (Accesses 70h – 7Fh)	8F0h	Common RAM (Accesses 70h – 7Fh)	970h	Common RAM (Accesses 70h – 7Fh)	9F0h	Common RAM (Accesses 70h – 7Fh)	A70h	Common RAM (Accesses 70h – 7Fh)	AF0h	Common RAM (Accesses 70h – 7Fh)	B70h	Common RAM (Accesses 70h – 7Fh)	BF0h	Common RAM (Accesses 70h – 7Fh)
87Fh		8FFh		97Fh		9FFh		A7Fh		AFFh		B7Fh		BFFh	

BANK 24		BANK 25		BANK 26		BANK 27		BANK 28		BANK 29		BANK 30	
C00h	Core Registers (Table 3-2)	C80h	Core Registers (Table 3-2)	D00h	Core Registers (Table 3-2)	D80h	Core Registers (Table 3-2)	E00h	Core Registers (Table 3-2)	E80h	Core Registers (Table 3-2)	F00h	Core Registers (Table 3-2)
C0Bh		C8Bh		D0Bh		D8Bh		E0Bh		E8Bh		F0Bh	
C0Ch	Unimplemented Read as '0'	C8Ch	Unimplemented Read as '0'	D0Ch	Unimplemented Read as '0'	D8Ch	Unimplemented Read as '0'	E0Ch	Unimplemented Read as '0'	E8Ch	Unimplemented Read as '0'	F0Ch	Unimplemented Read as '0'
C6Fh		CEFh		D6Fh		DEFh		E6Fh		EEFh		F6Fh	
C70h	Common RAM (Accesses 70h – 7Fh)	CF0h	Common RAM (Accesses 70h – 7Fh)	D70h	Common RAM (Accesses 70h – 7Fh)	DF0h	Common RAM (Accesses 70h – 7Fh)	E70h	Common RAM (Accesses 70h – 7Fh)	EF0h	Common RAM (Accesses 70h – 7Fh)	F70h	Common RAM (Accesses 70h – 7Fh)
C7Fh		CFFh		D7Fh		DFFh		E7Fh		EFFh		F7Fh	

Legend: = Unimplemented data memory locations, read as '0'

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 15 (Continued)											
7ABh	—	Unimplemented								—	—
7ACh	LCDDATA12	—	—	—	—	—	SEG26 COM0	SEG25 COM0	SEG24 COM0	---- -xxx	---- -uuu
7ADh	—	Unimplemented								—	—
7AEh	—	Unimplemented								—	—
7AFh	LCDDATA15	—	—	—	—	—	SEG26 COM1	SEG25 COM1	SEG24 COM1	---- -xxx	---- -uuu
7B0h	—	Unimplemented								—	—
7B1h	—	Unimplemented								—	—
7B2h	LCDDATA18	—	—	—	—	—	SEG26 COM2	SEG25 COM2	SEG24 COM2	---- -xxx	---- -uuu
7B3h	—	Unimplemented								—	—
7B4h	—	Unimplemented								—	—
7B5h	LCDDATA21	—	—	—	—	—	SEG26 COM3	SEG25 COM3	SEG24 COM3	---- -xxx	---- -uuu
7B6h — 7EFh	—	Unimplemented								—	—
Bank 16-30											
x0Ch or x8Ch to x1Fh or x9Fh	—	Unimplemented								—	—
Bank 31											
F8Ch — FE3h	—	Unimplemented								—	—
FE4h	STATUS_SHAD	—	—	—	—	—	Z_SHAD	DC_SHAD	C_SHAD	---- -xxx	---- -uuu
FE5h	WREG_SHAD	Working Register Normal (Non-ICD) Shadow								xxxx xxxx	uuuu uuuu
FE6h	BSR_SHAD	—	—	—	Bank Select Register Normal (Non-ICD) Shadow					---x xxxx	---u uuuu
FE7h	PCLATH_SHAD	—	Program Counter Latch High Register Normal (Non-ICD) Shadow							-xxx xxxx	uuuu uuuu
FE8h	FSR0L_SHAD	Indirect Data Memory Address 0 Low Pointer Normal (Non-ICD) Shadow								xxxx xxxx	uuuu uuuu
FE9h	FSR0H_SHAD	Indirect Data Memory Address 0 High Pointer Normal (Non-ICD) Shadow								xxxx xxxx	uuuu uuuu
FEAh	FSR1L_SHAD	Indirect Data Memory Address 1 Low Pointer Normal (Non-ICD) Shadow								xxxx xxxx	uuuu uuuu
FEBh	FSR1H_SHAD	Indirect Data Memory Address 1 High Pointer Normal (Non-ICD) Shadow								xxxx xxxx	uuuu uuuu
FECh	—	Unimplemented								—	—
FEDh	STKPTR	—	—	—	Current Stack Pointer					---1 1111	---1 1111
FEEh	TOSL	Top of Stack Low byte								xxxx xxxx	uuuu uuuu
FEFh	TOSH	—	Top of Stack High byte							-xxx xxxx	-uuu uuuu

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.

2: Unimplemented, read as '1'.

FIGURE 6-4: INTERNAL OSCILLATOR SWITCH TIMING

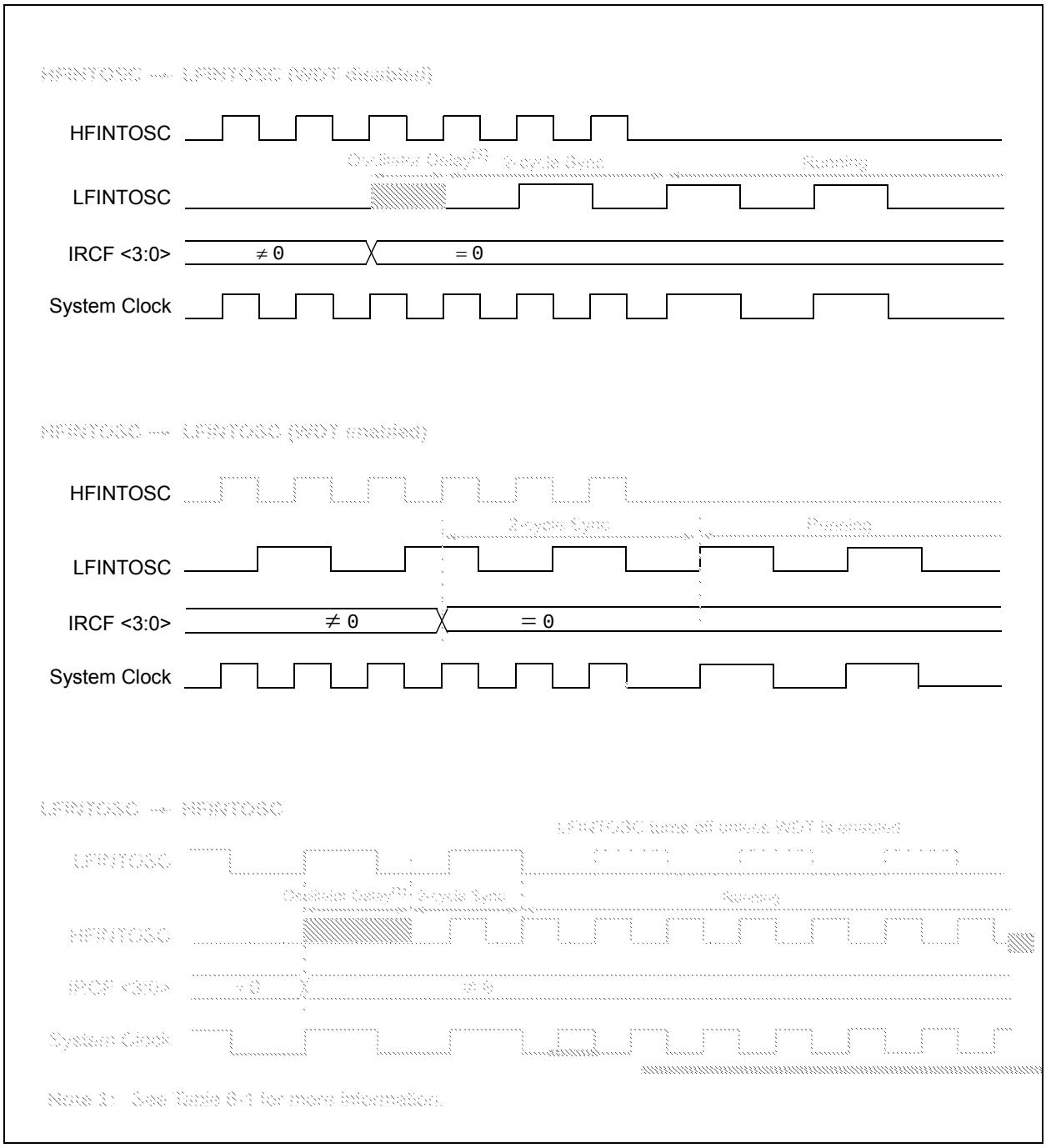


TABLE 6-1: OSCILLATOR SWITCHING DELAYS

Switch From	Switch To	Oscillator Delay
Any clock source	LFINTOSC	1 cycle of each clock source
	HFINTOSC	2 μs (approx.)
	ECH, ECM, ECL	2 cycles
	Secondary Oscillator	1024 Secondary Oscillator Cycles

6.3 Clock Switching

The system clock source can be switched between external and internal clock sources via software using the System Clock Select (SCS) bits of the OSCCON register. The following clock sources can be selected using the SCS bits:

- Default system oscillator determined by FOSC bits in Configuration Word 1
- Secondary oscillator 32 kHz crystal
- Internal Oscillator Block (INTOSC)

6.3.1 SYSTEM CLOCK SELECT (SCS) BITS

The System Clock Select (SCS) bits of the OSCCON register selects the system clock source that is used for the CPU and peripherals.

- When the SCS bits of the OSCCON register = `00`, the system clock source is determined by value of the FOSC<1:0> bits in the Configuration Word 1.
- When the SCS bits of the OSCCON register = `01`, the system clock source is the secondary oscillator.
- When the SCS bits of the OSCCON register = `1x`, the system clock source is chosen by the internal oscillator frequency selected by the IRCF<3:0> bits of the OSCCON register. After a Reset, the SCS bits of the OSCCON register are always cleared.

When switching between clock sources, a delay is required to allow the new clock to stabilize. These oscillator delays are shown in Table 6-2.

6.3.2 OSCILLATOR START-UP TIME-OUT STATUS (OSTS) BIT

The Oscillator Start-up Time-out Status (OSTS) bit of the OSCSTAT register indicates whether the system clock is running from the external clock source, as defined by the FOSC<1:0> bits in the Configuration Word 1, or from the internal clock source.

6.3.3 SECONDARY OSCILLATOR

The secondary oscillator is a separate crystal oscillator associated with the Timer1 peripheral. It is optimized for timekeeping operations with a 32.768 kHz crystal connected between the T1OSI and T1CKI/T1OSO device pins.

The secondary oscillator is enabled using the T1OSCEN control bit in the T1CON register. See **Section 17.0 “Timer1 Module with Gate Control”** for more information about the Timer1 peripheral.

6.3.4 SECONDARY OSCILLATOR READY (T1OSCR) BIT

The user must ensure that the secondary oscillator is ready to be used before it is selected as a system clock source. The Secondary Oscillator Ready (T1OSCR) bit of the OSCSTAT register indicates whether the secondary oscillator is ready to be used. After the T1OSCR bit is set, the SCS bits can be configured to select the secondary oscillator.

7.1 Operation

Interrupts are disabled upon any device Reset. They are enabled by setting the following bits:

- GIE bit of the INTCON register
- Interrupt Enable bit(s) for the specific interrupt event(s)
- PEIE bit of the INTCON register (if the Interrupt Enable bit of the interrupt event is contained in the PIE1 and PIE2 registers)

The INTCON, PIR1 and PIR2 registers record individual interrupts via interrupt flag bits. Interrupt flag bits will be set, regardless of the status of the GIE, PEIE and individual interrupt enable bits.

The following events happen when an interrupt event occurs while the GIE bit is set:

- Current prefetched instruction is flushed
- GIE bit is cleared
- Current Program Counter (PC) is pushed onto the stack
- Critical registers are automatically saved to the shadow registers (See **Section 7.5 "Automatic Context Saving"**)
- PC is loaded with the interrupt vector 0004h

The firmware within the Interrupt Service Routine (ISR) should determine the source of the interrupt by polling the interrupt flag bits. The interrupt flag bits must be cleared before exiting the ISR to avoid repeated interrupts. Because the GIE bit is cleared, any interrupt that occurs while executing the ISR will be recorded through its interrupt flag, but will not cause the processor to redirect to the interrupt vector.

The `RETFIE` instruction exits the ISR by popping the previous address from the stack, restoring the saved context from the shadow registers and setting the GIE bit.

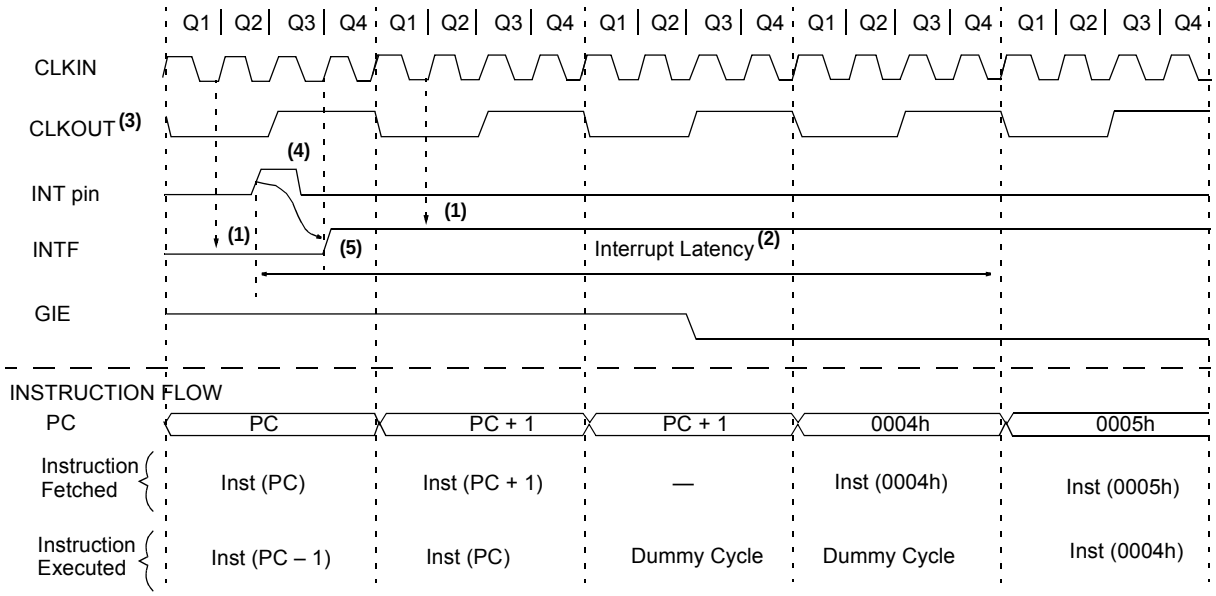
For additional information on a specific interrupt's operation, refer to its peripheral chapter.

Note 1:	Individual interrupt flag bits are set, regardless of the state of any other enable bits.
2:	All interrupts will be ignored while the GIE bit is cleared. Any interrupt occurring while the GIE bit is clear will be serviced when the GIE bit is set again.

7.2 Interrupt Latency

Interrupt latency is defined as the time from when the interrupt event occurs to the time code execution at the interrupt vector begins. The latency for synchronous interrupts is three or four instruction cycles. For asynchronous interrupts, the latency is three to five instruction cycles, depending on when the interrupt occurs. See Figure 7-2 and Figure 7.3 for more details.

FIGURE 7-3: INT PIN INTERRUPT TIMING



Note 1: INTF flag is sampled here (every Q1).

2: Asynchronous interrupt latency = 3-5 TCY. Synchronous latency = 3-4 TCY, where TCY = instruction cycle time. Latency is the same whether Inst (PC) is a single cycle or a 2-cycle instruction.

3: CLKOUT not available in all Oscillator modes.

4: For minimum width of INT pulse, refer to AC specifications in **Section 21.0 "Electrical Specifications"**.

5: INTF is enabled to be set any time during the Q4-Q1 cycles.

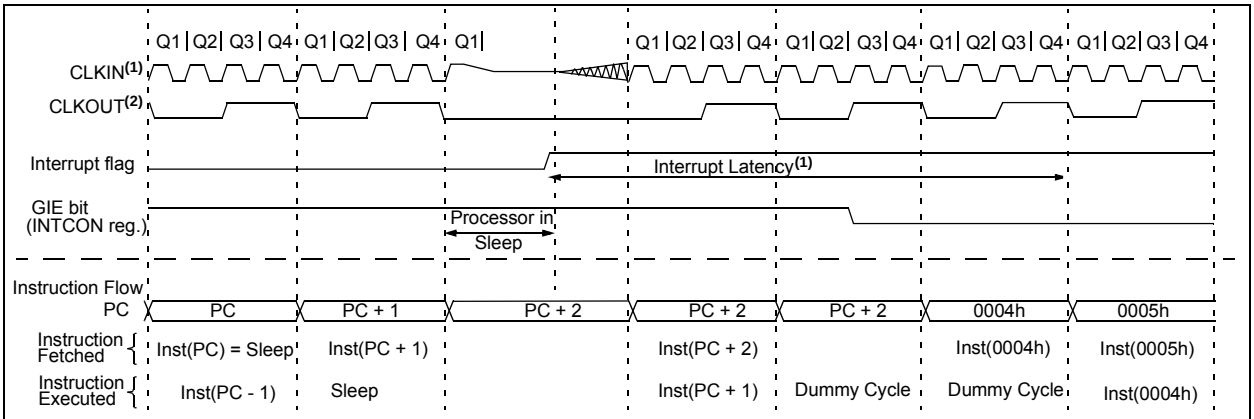
8.1.1 WAKE-UP USING INTERRUPTS

When global interrupts are disabled (GIE cleared) and any interrupt source has both its interrupt enable bit and interrupt flag bit set, one of the following will occur:

- If the interrupt occurs **before** the execution of a SLEEP instruction
  - SLEEP instruction will execute as a NOP.
  - WDT and WDT prescaler will not be cleared
  - $\overline{TO}$ bit of the STATUS register will not be set
  - $\overline{PD}$ bit of the STATUS register will not be cleared.
- If the interrupt occurs **during or after** the execution of a SLEEP instruction
  - SLEEP instruction will be completely executed
  - Device will immediately wake-up from Sleep
  - WDT and WDT prescaler will be cleared
  - $\overline{TO}$ bit of the STATUS register will be set
  - $\overline{PD}$ bit of the STATUS register will be cleared.

Even if the flag bits were checked before executing a SLEEP instruction, it may be possible for flag bits to become set before the SLEEP instruction completes. To determine whether a SLEEP instruction executed, test the $\overline{PD}$ bit. If the $\overline{PD}$ bit is set, the SLEEP instruction was executed as a NOP.

FIGURE 8-1: WAKE-UP FROM SLEEP THROUGH INTERRUPT



Note 1: GIE = 1 assumed. In this case after wake-up, the processor calls the ISR at 0004h. If GIE = 0, execution will continue in-line.

TABLE 8-1: SUMMARY OF REGISTERS ASSOCIATED WITH POWER-DOWN MODE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
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
PIE1	TMR1GIE	ADIE	—	—	—	—	—	TMR1IE	61
PIE2	—	—	—	—	—	LCDIE	—	—	62
PIR1	TMR1GIF	ADIF	—	—	—	—	—	TMR1IF	63
PIR2	—	—	—	—	—	LCDIF	—	—	64
STATUS	—	—	—	$\overline{TO}$	$\overline{PD}$	Z	DC	C	16
WDTCON	—	—	WDTPS<4:0>					SWDTEN	70

Legend: — = unimplemented location, read as '0'. Shaded cells are not used in Power-Down mode.

REGISTER 11-1: PORTA: PORTA REGISTER

R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R-x/x	R/W-x/x	R/W-x/x	R/W-x/x
RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0
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 **RA<7:0>**: PORTA I/O Value bits(1)
1 = Port pin is ≥ VIH
0 = Port pin is ≤ VIL

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

REGISTER 11-2: TRISA: PORTA TRI-STATE REGISTER

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R-1/1	R/W-1/1	R/W-1/1	R/W-1/1
TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0
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 7-4 **TRISA<7:4>:** PORTA Tri-State Control bits
1 = PORTA pin configured as an input (tri-stated)
0 = PORTA pin configured as an output

bit 3 **TRISA3:** RA3 Port Tri-State Control bit
This bit is always '1' as RA3 is an input only

bit 2-0 **TRISA<2:0>:** PORTA Tri-State Control bits
1 = PORTA pin configured as an input (tri-stated)
0 = PORTA pin configured as an output

REGISTER 11-3: LATA: PORTA 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
LATA7	LATA6	LATA5	LATA4	LATA3	LATA2	LATA1	LATA0
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 7-4 **LATA<7:0>**: PORTA Output Latch Value bits(1)

Note 1: Writes to PORTA are actually written to corresponding LATA register. Reads from PORTA 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 IOCBPx bit of the IOCBP register is set. To enable a pin to detect a falling edge, the associated IOCBNx bit of the IOCBN register is set.

A pin can be configured to detect rising and falling edges simultaneously by setting both the IOCBPx bit and the IOCBNx bit of the IOCBP and IOCBN registers, respectively.

12.3 Interrupt Flags

The IOCBFx 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 IOCBFx bits.

12.4 Clearing Interrupt Flags

The individual status flags, (IOCBFx 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:

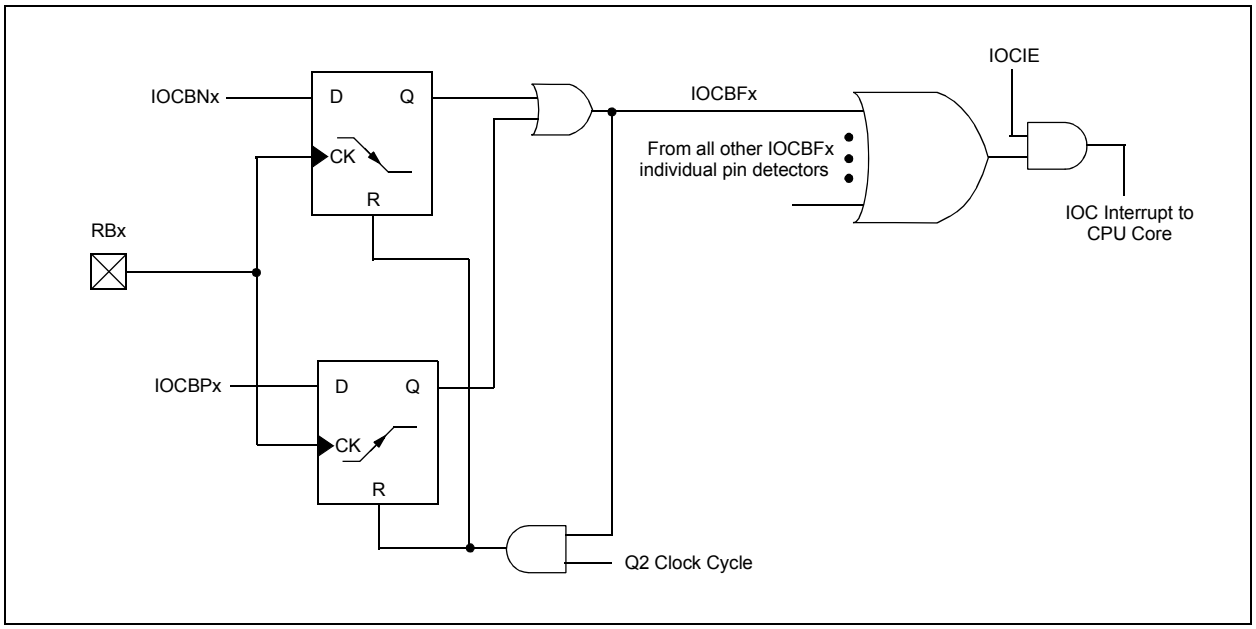
```
MOVLW  0xff
XORWF  IOCBF, W
ANDWF  IOCBF, F
```

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



REGISTER 15-2: ADCON1: A/D CONTROL REGISTER 1

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0
ADFM	ADCS<2:0>			—	—	ADPREF<1:0>	
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	

bit 7 **ADFM:** A/D Result Format Select bit
1 = Right justified. Six Most Significant bits of ADRESH are set to ‘0’ when the conversion result is loaded.
0 = Left justified. Six Least Significant bits of ADRESL are set to ‘0’ when the conversion result is loaded.

bit 6-4 **ADCS<2:0>:** A/D Conversion Clock Select bits
000 = FOSC/2
001 = FOSC/8
010 = FOSC/32
011 = FRC (clock supplied from a dedicated RC oscillator)
100 = FOSC/4
101 = FOSC/16
110 = FOSC/64
111 = FRC (clock supplied from a dedicated RC oscillator)

bit 3-2 **Unimplemented:** Read as ‘0’

bit 1-0 **ADPREF<1:0>:** A/D Positive Voltage Reference Configuration bits
00 = VREF+ is connected to VDD
01 = Reserved
10 = VREF+ is connected to external VREF+ pin(1)
11 = Reserved

Note 1: When selecting the FVR or the VREF+ pin as the source of the positive reference, be aware that a minimum voltage specification exists. See **Section 21.0 “Electrical Specifications”** for details.

17.10 Timer1 Gate Control Register

The Timer1 Gate Control register (T1GCON), shown in Register 17-2, is used to control Timer1 gate.

REGISTER 17-2: T1GCON: TIMER1 GATE CONTROL REGISTER

R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	R/W/HC-0/u	R-x/x	R/W-0/u	R/W-0/u
TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/$\overline{\text{DONE}}$	T1GVAL	T1GSS<1:0>	
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	HC = Bit is cleared by hardware

bit 7 **TMR1GE:** Timer1 Gate Enable bit

If TMR1ON = 0:
This bit is ignored
If TMR1ON = 1:
1 = Timer1 counting is controlled by the Timer1 gate function
0 = Timer1 counts regardless of Timer1 gate function

bit 6 **T1GPOL:** Timer1 Gate Polarity bit

1 = Timer1 gate is active-high (Timer1 counts when gate is high)
0 = Timer1 gate is active-low (Timer1 counts when gate is low)

bit 5 **T1GTM:** Timer1 Gate Toggle Mode bit

1 = Timer1 Gate Toggle mode is enabled
0 = Timer1 Gate Toggle mode is disabled and toggle flip-flop is cleared
Timer1 gate flip-flop toggles on every rising edge.

bit 4 **T1GSPM:** Timer1 Gate Single-Pulse Mode bit

1 = Timer1 gate Single-Pulse mode is enabled and is controlling Timer1 gate
0 = Timer1 gate Single-Pulse mode is disabled

bit 3 **T1GGO/$\overline{\text{DONE}}$:** Timer1 Gate Single-Pulse Acquisition Status bit

1 = Timer1 gate single-pulse acquisition is ready, waiting for an edge
0 = Timer1 gate single-pulse acquisition has completed or has not been started

bit 2 **T1GVAL:** Timer1 Gate Current State bit

Indicates the current state of the Timer1 gate that could be provided to TMR1H:TMR1L.
Unaffected by Timer1 Gate Enable (TMR1GE).

bit 1-0 **T1GSS<1:0>:** Timer1 Gate Source Select bits

00 = Timer1 gate pin
01 = Timer0 overflow output
10 = Reserved
11 = Reserved

TABLE 18-1: LCD SEGMENT AND DATA REGISTERS

Device	# of LCD Registers	
	Segment Enable	Data
PIC16LF1902/3	3	12

The LCDCON register (Register 18-1) controls the operation of the LCD Driver module. The LCDPS register (Register 18-2) configures the LCD clock source prescaler and the type of waveform; Type-A or Type-B. The LCDSEn registers (Register 18-5) configure the functions of the port pins.

The following LCDSEn registers are available:

- LCDSE0 SE<7:0>
- LCDSE1 SE<15:8>
- LCDSE3 SE<26:24>

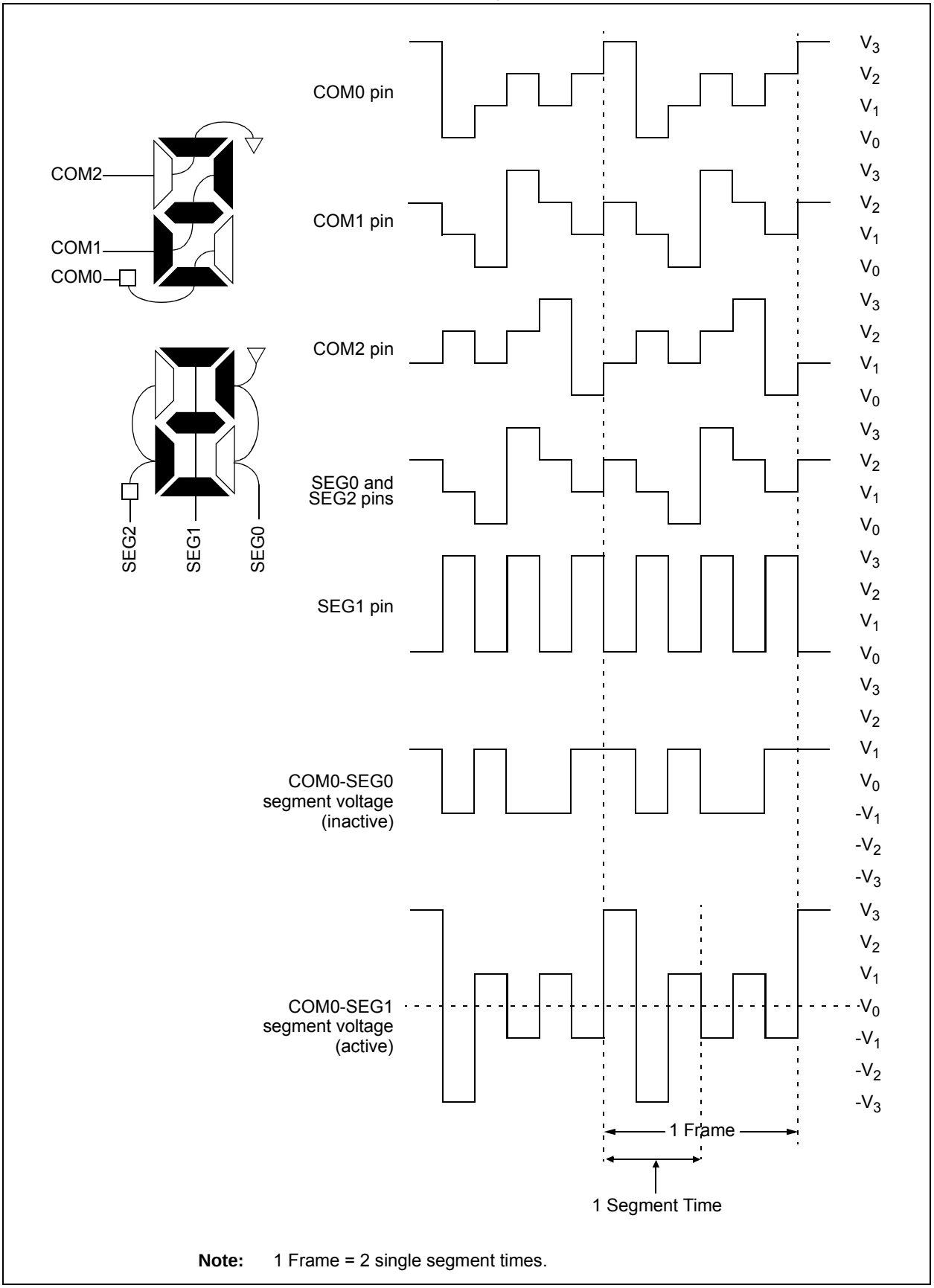
Once the module is initialized for the LCD panel, the individual bits of the LCDDATAn registers are cleared/set to represent a clear/dark pixel, respectively:

- LCDDATA0 SEG<7:0>COM0
- LCDDATA1 SEG<15:8>COM0
- LCDDATA3 SEG<7:0>COM1
- LCDDATA4 SEG<15:8>COM1
- LCDDATA6 SEG<7:0>COM2
- LCDDATA7 SEG<15:8>COM2
- LCDDATA9 SEG<7:0>COM3
- LCDDATA10 SEG<15:8>COM3
- LCDDATA12 SEG<26:24>COM0
- LCDDATA15 SEG<26:24>COM1
- LCDDATA18 SEG<26:24>COM2
- LCDDATA21 SEG<26:24>COM3

As an example, LCDDATAn is detailed in Register 18-6.

Once the module is configured, the LCDEN bit of the LCDCON register is used to enable or disable the LCD module. The LCD panel can also operate during Sleep by clearing the SLPEN bit of the LCDCON register.

FIGURE 18-15: TYPE-A WAVEFORMS IN 1/3 MUX, 1/3 BIAS DRIVE



Note: 1 Frame = 2 single segment times.

19.0 IN-CIRCUIT SERIAL PROGRAMMING™ (ICSP™)

ICSP™ programming allows customers to manufacture circuit boards with unprogrammed devices. Programming can be done after the assembly process allowing the device to be programmed with the most recent firmware or a custom firmware. Five pins are needed for ICSP™ programming:

- ICSPCLK
- ICSPDAT
- $\overline{MCLR}$/VPP
- VDD
- VSS

In Program/Verify mode the Program Memory, User IDs and the Configuration Words are programmed through serial communications. The ICSPDAT pin is a bidirectional I/O used for transferring the serial data and the ICSPCLK pin is the clock input. For more information on ICSP™ refer to the **"***PIC16F193X/LF193X/PIC16F194X/LF194X/PIC16LF190X Memory Programming Specification***"** (DS41397).

19.1 High-Voltage Programming Entry Mode

The device is placed into High-Voltage Programming Entry mode by holding the ICSPCLK and ICSPDAT pins low then raising the voltage on $\overline{MCLR}$/VPP to VIHH.

19.2 Low-Voltage Programming Entry Mode

The Low-Voltage Programming Entry mode allows the PIC® Flash MCUs to be programmed using VDD only, without high voltage. When the LVP bit of Configuration Word 2 is set to '1', the low-voltage ICSP programming entry is enabled. To disable the Low-Voltage ICSP mode, the LVP bit must be programmed to '0'.

Entry into the Low-Voltage Programming Entry mode requires the following steps:

1. $\overline{MCLR}$ is brought to VIL.
2. A 32-bit key sequence is presented on ICSPDAT, while clocking ICSPCLK.

Once the key sequence is complete, $\overline{MCLR}$ must be held at VIL for as long as Program/Verify mode is to be maintained.

If low-voltage programming is enabled (LVP = 1), the $\overline{MCLR}$ Reset function is automatically enabled and cannot be disabled. See **Section 5.3 "Low-Power Brown-out Reset (LPBOR)"** for more information.

The LVP bit can only be reprogrammed to '0' by using the High-Voltage Programming mode.

19.3 Common Programming Interfaces

Connection to a target device is typically done through an ICSP™ header. A commonly found connector on development tools is the RJ-11 in the 6P6C (6-pin, 6 connector) configuration. See Figure 19-1.

FIGURE 19-1: ICD RJ-11 STYLE CONNECTOR INTERFACE

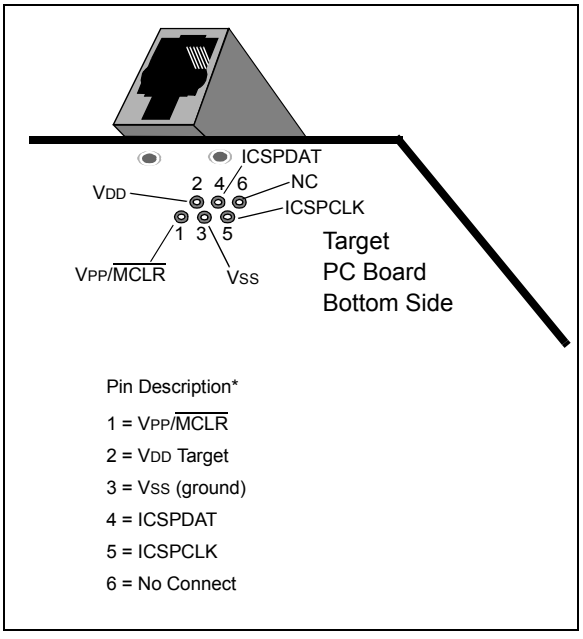


TABLE 20-3: PIC16LF1902/3 ENHANCED INSTRUCTION SET

Mnemonic, Operands		Description	Cycles	14-Bit Opcode MSb			LSb	Status Affected	Notes
BYTE-ORIENTED FILE REGISTER OPERATIONS									
ADDWF	f, d	Add W and f	1	00	0111	dfff	ffff	C, DC, Z	2
ADDWFC	f, d	Add with Carry W and f	1	11	1101	dfff	ffff	C, DC, Z	2
ANDWF	f, d	AND W with f	1	00	0101	dfff	ffff	Z	2
ASRF	f, d	Arithmetic Right Shift	1	11	0111	dfff	ffff	C, Z	2
LSLF	f, d	Logical Left Shift	1	11	0101	dfff	ffff	C, Z	2
LSRF	f, d	Logical Right Shift	1	11	0110	dfff	ffff	C, Z	2
CLRF	f	Clear f	1	00	0001	lfff	ffff	Z	2
CLRW	–	Clear W	1	00	0001	0000	00xx	Z	
COMF	f, d	Complement f	1	00	1001	dfff	ffff	Z	2
DECF	f, d	Decrement f	1	00	0011	dfff	ffff	Z	2
INCF	f, d	Increment f	1	00	1010	dfff	ffff	Z	2
IORWF	f, d	Inclusive OR W with f	1	00	0100	dfff	ffff	Z	2
MOVF	f, d	Move f	1	00	1000	dfff	ffff	Z	2
MOVWF	f	Move W to f	1	00	0000	1fff	ffff		2
RLF	f, d	Rotate Left f through Carry	1	00	1101	dfff	ffff	C	2
RRF	f, d	Rotate Right f through Carry	1	00	1100	dfff	ffff	C	2
SUBWF	f, d	Subtract W from f	1	00	0010	dfff	ffff	C, DC, Z	2
SUBWFB	f, d	Subtract with Borrow W from f	1	11	1011	dfff	ffff	C, DC, Z	2
SWAPF	f, d	Swap nibbles in f	1	00	1110	dfff	ffff		2
XORWF	f, d	Exclusive OR W with f	1	00	0110	dfff	ffff	Z	2
BYTE ORIENTED SKIP OPERATIONS									
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	00	1011	dfff	ffff		1, 2
INCFSZ	f, d	Increment f, Skip if 0	1(2)	00	1111	dfff	ffff		1, 2
BIT-ORIENTED FILE REGISTER OPERATIONS									
BCF	f, b	Bit Clear f	1	01	00bb	bfff	ffff		2
BSF	f, b	Bit Set f	1	01	01bb	bfff	ffff		2
BIT-ORIENTED SKIP OPERATIONS									
BTFSC	f, b	Bit Test f, Skip if Clear	1 (2)	01	10bb	bfff	ffff		1, 2
BTFSS	f, b	Bit Test f, Skip if Set	1 (2)	01	11bb	bfff	ffff		1, 2
LITERAL OPERATIONS									
ADDLW	k	Add literal and W	1	11	1110	kkkk	kkkk	C, DC, Z	
ANDLW	k	AND literal with W	1	11	1001	kkkk	kkkk	Z	
IORLW	k	Inclusive OR literal with W	1	11	1000	kkkk	kkkk	Z	
MOVLB	k	Move literal to BSR	1	00	0000	001k	kkkk		
MOVLP	k	Move literal to PCLATH	1	11	0001	1kkk	kkkk		
MOVLW	k	Move literal to W	1	11	0000	kkkk	kkkk		
SUBLW	k	Subtract W from literal	1	11	1100	kkkk	kkkk	C, DC, Z	
XORLW	k	Exclusive OR literal with W	1	11	1010	kkkk	kkkk	Z	

Note 1: If the Program Counter (PC) is modified, or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.

2: If this instruction addresses an INDF register and the MSb of the corresponding FSR is set, this instruction will require one additional instruction cycle.

MOVWI **Move W to INDFn**

Syntax: [*label*] MOVWI ++FSRn
[*label*] MOVWI --FSRn
[*label*] MOVWI FSRn++
[*label*] MOVWI FSRn--
[*label*] MOVWI k[FSRn]

Operands: n ∈ [0,1]
mm ∈ [00,01, 10, 11]
-32 ≤ k ≤ 31

Operation: W → INDFn
Effective address is determined by
- FSR + 1 (preincrement)
- FSR - 1 (predecrement)
- FSR + k (relative offset)

After the Move, the FSR value will be either:
- FSR + 1 (all increments)
- FSR - 1 (all decrements)

Unchanged

Status Affected: None

Mode	Syntax	mm
Preincrement	++FSRn	00
Predecrement	--FSRn	01
Postincrement	FSRn++	10
Postdecrement	FSRn--	11

Description: This instruction is used to move data between W and one of the indirect registers (INDFn). Before/after this move, the pointer (FSRn) is updated by pre/post incrementing/decrementing it.

Note: The INDFn registers are not physical registers. Any instruction that accesses an INDFn register actually accesses the register at the address specified by the FSRn.

FSRn is limited to the range 0000h - FFFFh. Incrementing/decrementing it beyond these bounds will cause it to wrap-around.

The increment/decrement operation on FSRn WILL NOT affect any Status bits.

NOP **No Operation**

Syntax: [*label*] NOP

Operands: None

Operation: No operation

Status Affected: None

Description: No operation.

Words: 1

Cycles: 1

Example:
```
NOP
```

OPTION **Load OPTION_REG Register with W**

Syntax: [*label*] OPTION

Operands: None

Operation: (W) → OPTION_REG

Status Affected: None

Description: Move data from W register to OPTION_REG register.

Words: 1

Cycles: 1

Example:
```
OPTION
```
Before Instruction
OPTION_REG = 0xFF
W = 0x4F
After Instruction
OPTION_REG = 0x4F
W = 0x4F

RESET **Software Reset**

Syntax: [*label*] RESET

Operands: None

Operation: Execute a device Reset. Resets the nRI flag of the PCON register.

Status Affected: None

Description: This instruction provides a way to execute a hardware Reset by software.

TABLE 21-4: I/O PORTS

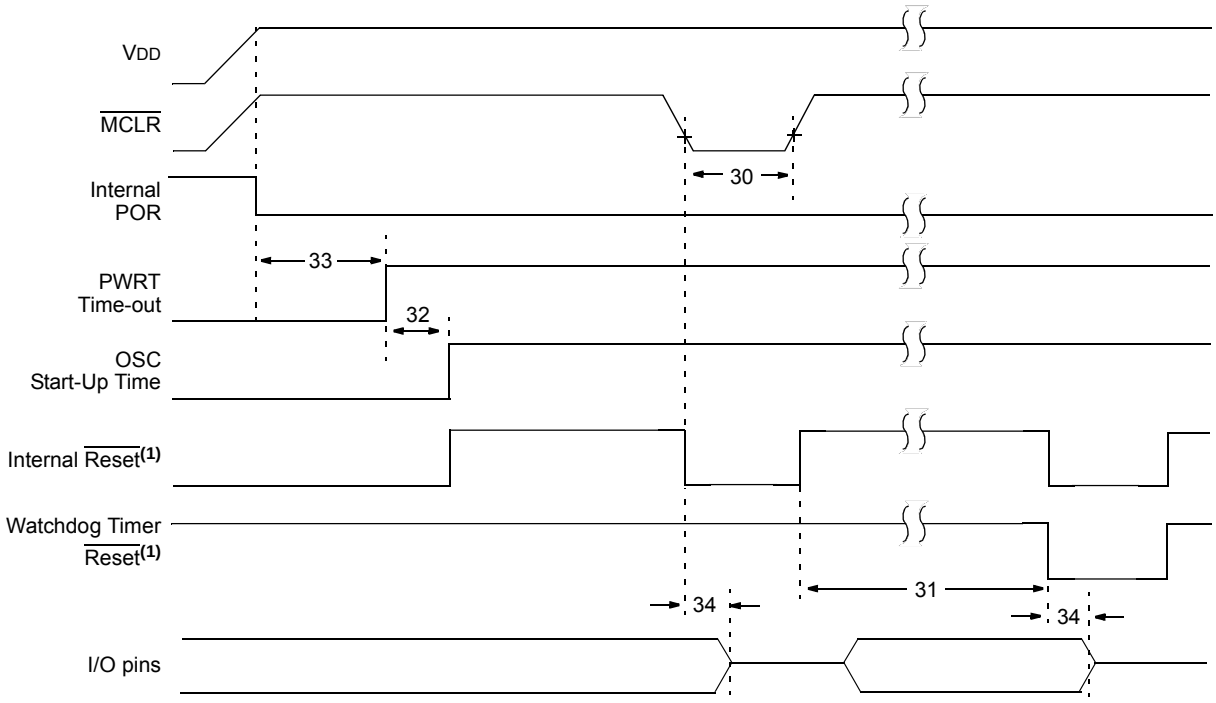
DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise stated)				
Param No.	Sym.	Characteristic	Min.	Typ†	Max.	Units	Conditions
	V_{IL}	**Input Low Voltage**					
		I/O PORT:					
D032		with TTL buffer	—	—	0.15 V_{DD}	V	1.8V ≤ V_{DD} ≤ 3.6V
D033		with Schmitt Trigger buffer	—	—	0.2 V_{DD}	V	1.8V ≤ V_{DD} ≤ 3.6V
D034		$\overline{MCLR}$, OSC1	—	—	0.2 V_{DD}	V	
	V_{IH}	**Input High Voltage**					
		I/O ports:					
D040		with TTL buffer	0.25 V_{DD} + 0.8	—	—	V	1.8V ≤ V_{DD} ≤ 3.6V
D041		with Schmitt Trigger buffer	0.8 V_{DD}	—	—	V	1.8V ≤ V_{DD} ≤ 3.6V
D042		$\overline{MCLR}$	0.8 V_{DD}	—	—	V	
	I_{IL}	**Input Leakage Current(2)**					
D060		I/O ports	—	± 5	± 125	nA	V_{SS} ≤ V_{PIN} ≤ V_{DD}, Pin at high-impedance @ 85°C
				± 5	± 1000	nA	125°C
D061		$\overline{MCLR}$(3)	—	± 50	± 200	nA	V_{SS} ≤ V_{PIN} ≤ V_{DD} @ 85°C
	I_{PUR}	**Weak Pull-up Current**					
D070*			25	100	200	µA	V_{DD} = 3.3V, V_{PIN} = V_{SS}
	V_{OL}	**Output Low Voltage**					
D080		I/O ports	—	—	0.6	V	I_{OL} = 6mA, V_{DD} = 3.3V I_{OL} = 1.8mA, V_{DD} = 1.8V
	V_{OH}	**Output High Voltage**					
D090		I/O ports	V_{DD} - 0.7	—	—	V	I_{OH} = 3mA, V_{DD} = 3.3V I_{OH} = 1mA, V_{DD} = 1.8V
		Capacitive Loading Specs on Output Pins					
D101*	C_{IO}	All I/O pins	—	—	50	pF	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

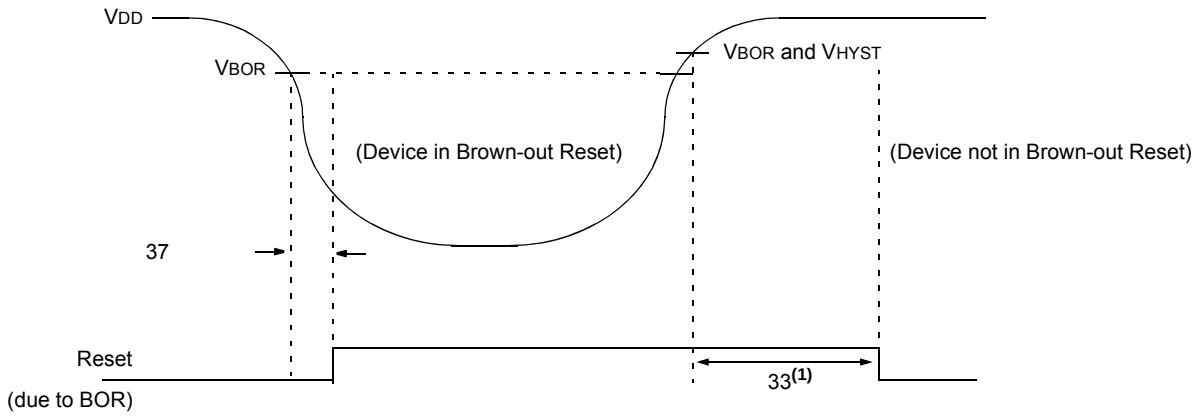
Note 1: Negative current is defined as current sourced by the pin.

FIGURE 21-6: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING



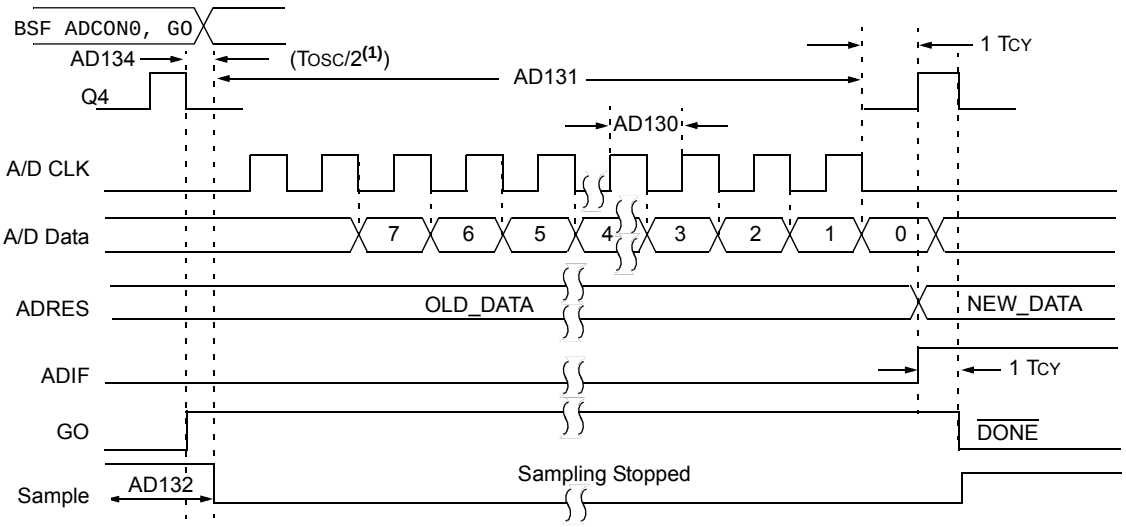
Note 1: Asserted low.

FIGURE 21-7: BROWN-OUT RESET TIMING AND CHARACTERISTICS



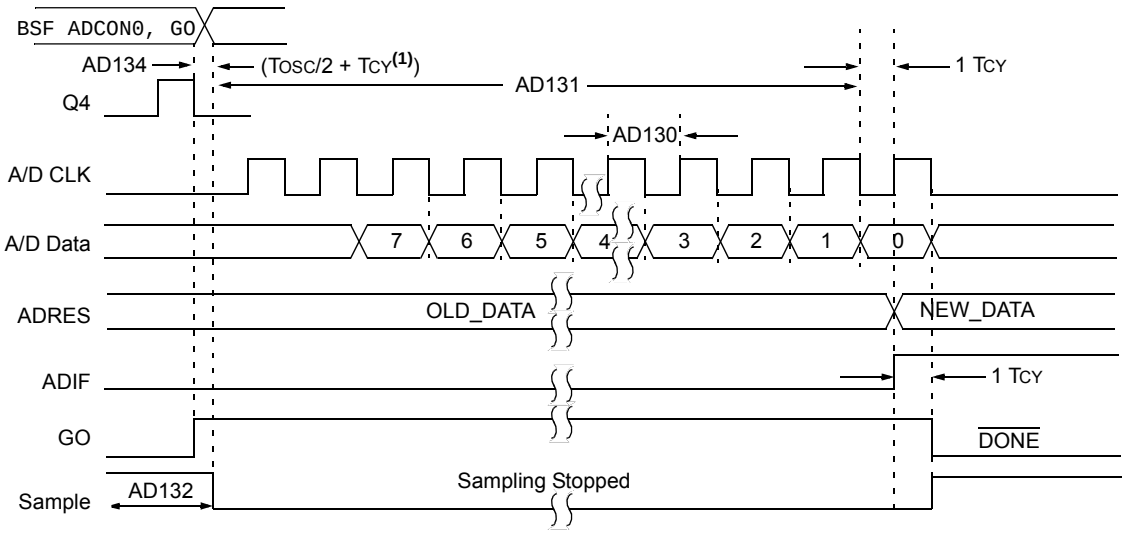
Note 1: 64 ms delay only if PWRTE bit in the Configuration Word register is programmed to '0'.
2 ms delay if PWRTE = 0 and VREGEN = 1.

FIGURE 21-10: PIC16LF1902/3 A/D CONVERSION TIMING (NORMAL MODE)



Note 1: If the A/D clock source is selected as RC, a time of TCY is added before the A/D clock starts. This allows the SLEEP instruction to be executed.

FIGURE 21-11: PIC16LF1902/3 A/D CONVERSION TIMING (SLEEP MODE)



Note 1: If the A/D clock source is selected as RC, a time of TCY is added before the A/D clock starts. This allows the SLEEP instruction to be executed.