Welcome to <u>E-XFL.COM</u>

What is "<u>Embedded - Microcontrollers</u>"?

"<u>Embedded - Microcontrollers</u>" 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 "<u>Embedded - Microcontrollers</u>"

Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	20MHz
Connectivity	LINbus, UART/USART
Peripherals	Brown-out Detect/Reset, LCD, POR, PWM, WDT
Number of I/O	36
Program Memory Size	7KB (4K x 14)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 14x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Through Hole
Package / Case	40-DIP (0.600", 15.24mm)
Supplier Device Package	40-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lf1904-e-p

3.1.1 READING PROGRAM MEMORY AS DATA

There are two methods of accessing constants in program memory. The first method is to use tables of RETLW instructions. The second method is to set an FSR to point to the program memory.

3.1.1.1 RETLW Instruction

The RETLW instruction can be used to provide access to tables of constants. The recommended way to create such a table is shown in Example 3-1.

EXAMPLE 3-1: **RETLW INSTRUCTION**

```
constants
    BRW                 ;Add Index in W to
                        ;program counter to
                        ;select data
    RETLW DATA0         ;Index0 data
    RETLW DATA1         ;Index1 data
    RETLW DATA2
    RETLW DATA3


my_function
    ;… LOTS OF CODE…
    MOVLW       DATA_INDEX
    call constants
    ;… THE CONSTANT IS IN W
```

The BRW instruction makes this type of table very simple to implement. If your code must remain portable with previous generations of microcontrollers, then the BRW instruction is not available so the older table read method must be used.

3.1.1.2 Indirect Read with FSR

The program memory can be accessed as data by setting bit 7 of the FSRxH register and reading the matching INDFx register. The MOVIW instruction will place the lower eight bits of the addressed word in the W register. Writes to the program memory cannot be performed via the INDF registers. Instructions that access the program memory via the FSR require one extra instruction cycle to complete. Example 3-2 demonstrates accessing the program memory via an FSR.

The HIGH directive will set bit<7> if a label points to a location in program memory.

EXAMPLE 3-2: **ACCESSING PROGRAM MEMORY VIA FSR**

```
constants
    RETLW DATA0         ;Index0 data
    RETLW DATA1         ;Index1 data
    RETLW DATA2
    RETLW DATA3
my_function
    ;… LOTS OF CODE…
    MOVLW   LOW constants
    MOVWF   FSR1L
    MOVLW   HIGH constants
    MOVWF   FSR1H
    MOVIW 0[FSR1]
;THE PROGRAM MEMORY IS IN W
```

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
10Eh	LATD[3]	PORTD Data Latch								xxxx xxxx	uuuu uuuu
10Eh	LATE[3]	—	—	—	—	—	LATE2	LATE1	LATE0	---- -xxx	---- -uuu
111h 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	ANSELE[3]	—	—	—	—	—	ANSE2	ANSE1	ANSE0	---- -111	---- -111
191h	PMADRL	Program Memory Address Register Low Byte								0000 0000	0000 0000
192h	PMADRH	—[2]	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	—[2]	CFGS	LWLO	FREE	WRERR	WREN	WR	RD	1000 x000	1000 q000
196h	PMCON2	Program Memory Control Register 2								0000 0000	0000 0000
197h	—	Unimplemented								—	—
198h	—	Unimplemented								—	—
199h	RCREG	USART Receive Data Register								0000 0000	0000 0000
19Ah	TXREG	USART Transmit Data Register								0000 0000	0000 0000
19Bh	SPBRG	BRG<7:0>								0000 0000	0000 0000
19Ch	SPBRGH	BRG<15:8>								0000 0000	0000 0000
19Dh	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
19Eh	TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010	0000 0010
19Fh	BAUD1CON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	01-0 0-00	01-0 0-00
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.
2: Unimplemented, read as '1'.
3: PIC16LF1904/7 only.

5.3 Low-Power Brown-out Reset (LPBOR)

The Low-Power Brown-Out Reset (LPBOR) is an essential part of the Reset subsystem. Refer to Figure 5-1 to see how the BOR interacts with other modules.

The LPBOR is used to monitor the external V_{DD} pin. When too low of a voltage is detected, the device is held in Reset. When this occurs, a register bit ($\overline{BOR}$) is changed to indicate that a BOR Reset has occurred. The same bit is set for both the BOR and the LPBOR. Refer to Register 5-2.

5.3.1 ENABLING LPBOR

The LPBOR is controlled by the $\overline{LPBOR}$ bit of Configuration Word 2. When the device is erased, the LPBOR module defaults to disabled.

5.3.1.1 LPBOR Module Output

The output of the LPBOR module is a signal indicating whether or not a Reset is to be asserted. This signal is to be OR'd together with the Reset signal of the BOR module to provide the generic $\overline{BOR}$ signal, which goes to the PCON register and to the power control block.

5.4 $\overline{MCLR}$

The $\overline{MCLR}$ is an optional external input that can reset the device. The $\overline{MCLR}$ function is controlled by the MCLRE bit of Configuration Word 1 and the LVP bit of Configuration Word 2 (Table 5-2).

TABLE 5-2: $\overline{MCLR}$ CONFIGURATION

MCLRE	LVP	$\overline{MCLR}$
0	0	Disabled
1	0	Enabled
x	1	Enabled

5.4.1 $\overline{MCLR}$ ENABLED

When $\overline{MCLR}$ is enabled and the pin is held low, the device is held in Reset. The $\overline{MCLR}$ pin is connected to V_{DD} through an internal weak pull-up.

The device has a noise filter in the $\overline{MCLR}$ Reset path. The filter will detect and ignore small pulses.

> **Note:** A Reset does not drive the $\overline{MCLR}$ pin low.

5.4.2 $\overline{MCLR}$ DISABLED

When $\overline{MCLR}$ is disabled, the pin functions as a general purpose input and the internal weak pull-up is under software control. See **Section 11.5 "PORTE Registers"** for more information.

5.5 Watchdog Timer (WDT) Reset

The Watchdog Timer generates a Reset if the firmware does not issue a `CLRWDT` instruction within the time-out period. The $\overline{TO}$ and $\overline{PD}$ bits in the STATUS register are changed to indicate the WDT Reset. See **Section 9.0 "Watchdog Timer"** for more information.

5.6 `RESET` Instruction

A `RESET` instruction will cause a device Reset. The $\overline{RI}$ bit in the PCON register will be set to '0'. See Table 5-4 for default conditions after a `RESET` instruction has occurred.

5.7 Stack Overflow/Underflow Reset

The device can reset when the Stack Overflows or Underflows. The STKOVF or STKUNF bits of the PCON register indicate the Reset condition. These Resets are enabled by setting the STVREN bit in Configuration Word 2. See **Section 5.7 "Stack Overflow/Underflow Reset"** for more information.

5.8 Programming Mode Exit

Upon exit of Programming mode, the device will behave as if a POR had just occurred.

5.9 Power-Up Timer

The Power-up Timer optionally delays device execution after a BOR or POR event. This timer is typically used to allow V_{DD} to stabilize before allowing the device to start running.

The Power-up Timer is controlled by the $\overline{PWRTE}$ bit of Configuration Word 1.

5.10 Start-up Sequence

Upon the release of a POR or BOR, the following must occur before the device will begin executing:

1. Power-up Timer runs to completion (if enabled).
2. Oscillator start-up timer runs to completion (if required for oscillator source).
3. $\overline{MCLR}$ must be released (if enabled).

The total time-out will vary based on oscillator configuration and Power-up Timer configuration. See **Section 6.0 "Oscillator Module"** for more information.

The Power-up Timer and oscillator start-up timer run independently of $\overline{MCLR}$ Reset. If $\overline{MCLR}$ is kept low long enough, the Power-up Timer and oscillator start-up timer will expire. Upon bringing $\overline{MCLR}$ high, the device will begin execution immediately (see Figure 5-3). This is useful for testing purposes or to synchronize more than one device operating in parallel.
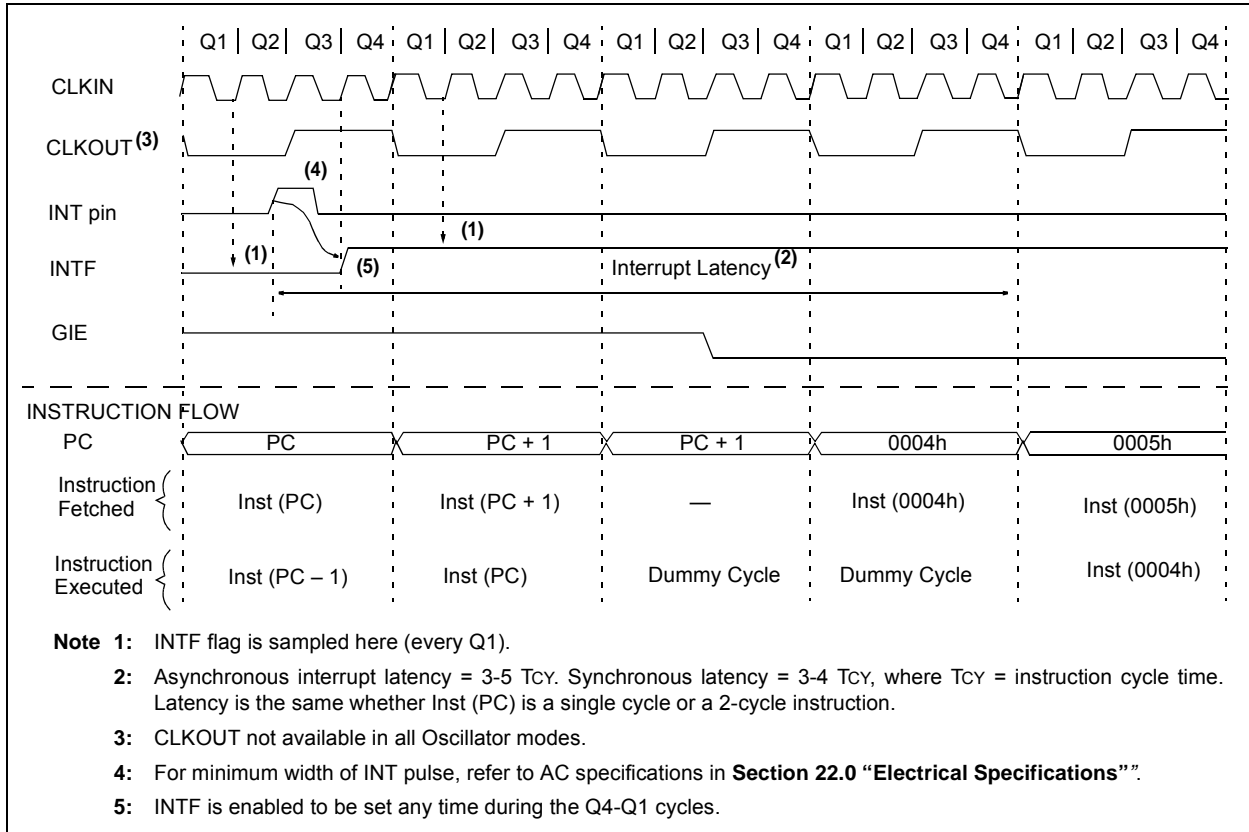
FIGURE 7-3: **INT PIN INTERRUPT TIMING**

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 22.0 "Electrical Specifications"**.

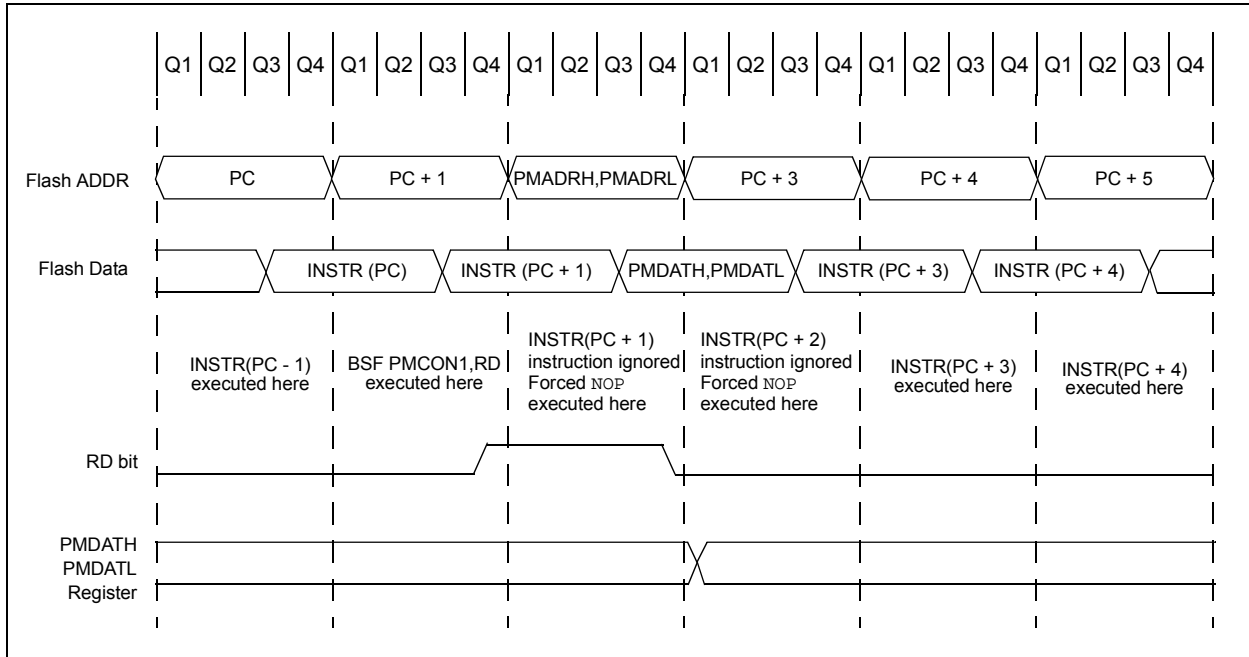   **5:** INTF is enabled to be set any time during the Q4-Q1 cycles.

FIGURE 10-2: **FLASH PROGRAM MEMORY READ CYCLE EXECUTION**

EXAMPLE 10-1: **FLASH PROGRAM MEMORY READ**

```
* This code block will read 1 word of program
* memory at the memory address:
    PROG_ADDR_HI : PROG_ADDR_LO
*   data will be returned in the variables;
*   PROG_DATA_HI, PROG_DATA_LO

    BANKSEL   PMADRL          ; Select Bank for PMCON registers
    MOVLW     PROG_ADDR_LO    ;
    MOVWF     PMADRL          ; Store LSB of address
    MOVLW     PROG_ADDR_HI    ;
    MOVWL     PMADRH          ; Store MSB of address

    BCF       PMCON1,CFGS     ; Do not select Configuration Space
    BSF       PMCON1,RD       ; Initiate read
    NOP                       ; Ignored (Figure 10-1)
    NOP                       ; Ignored (Figure 10-1)

    MOVF      PMDATL,W        ; Get LSB of word
    MOVWF     PROG_DATA_LO    ; Store in user location
    MOVF      PMDATH,W        ; Get MSB of word
    MOVWF     PROG_DATA_HI    ; Store in user location
```
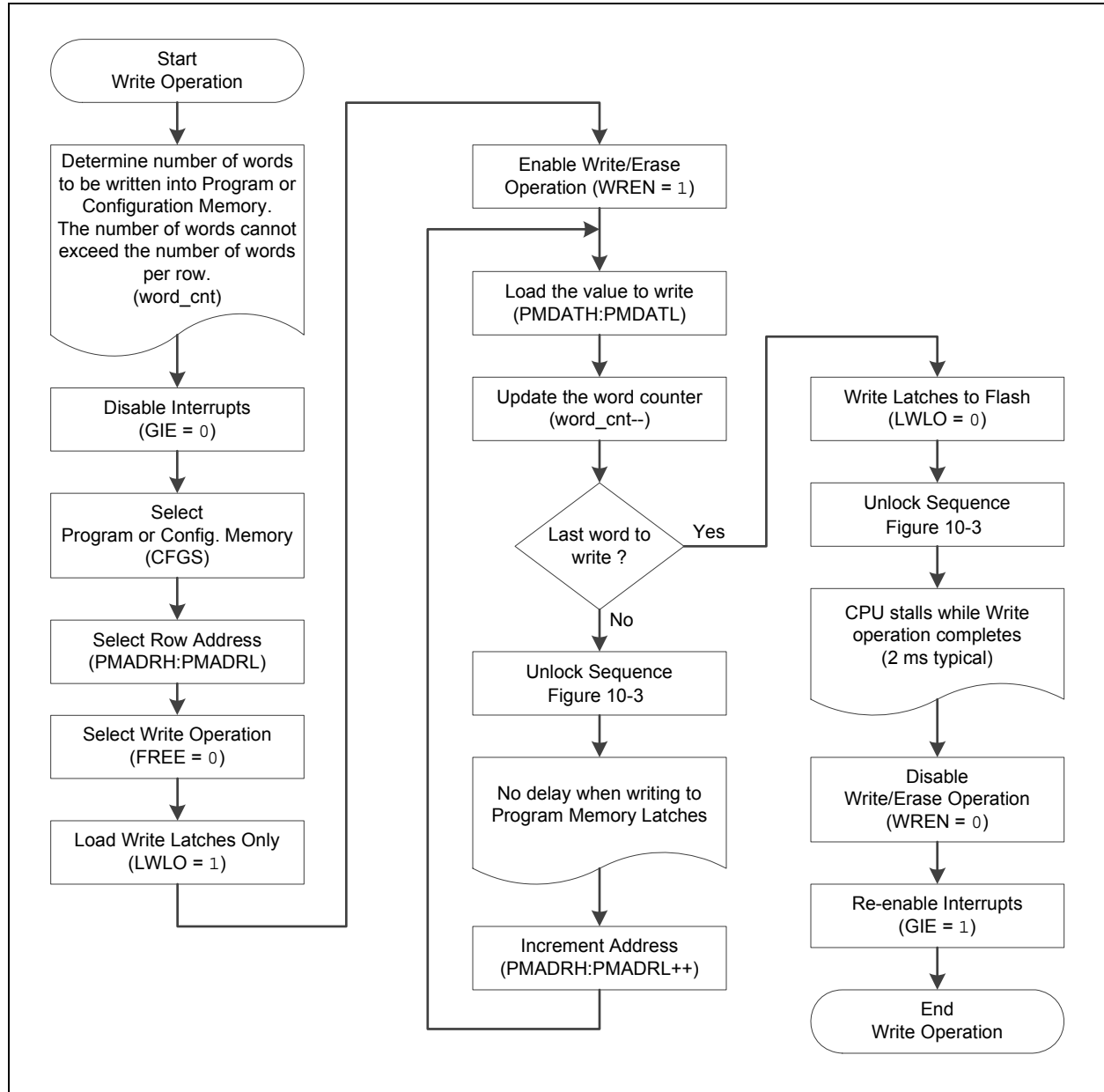
FIGURE 10-6: **FLASH PROGRAM MEMORY WRITE FLOWCHART**

14.0 TEMPERATURE INDICATOR MODULE

This family of devices is equipped with a temperature circuit designed to measure the operating temperature of the silicon die. The circuit's range of operating temperature falls between of -40°C and +85°C. The output is a voltage that is proportional to the device temperature. The output of the temperature indicator is internally connected to the device ADC.

The circuit may be used as a temperature threshold detector or a more accurate temperature indicator, depending on the level of calibration performed. A one-point calibration allows the circuit to indicate a temperature closely surrounding that point. A two-point calibration allows the circuit to sense the entire range of temperature more accurately. Reference Application Note AN1333, "*Use and Calibration of the Internal Temperature Indicator*" (DS01333) for more details regarding the calibration process.

14.1 Circuit Operation

Figure 14-1 shows a simplified block diagram of the temperature circuit. The proportional voltage output is achieved by measuring the forward voltage drop across multiple silicon junctions.

Equation 14-1 describes the output characteristics of the temperature indicator.

EQUATION 14-1: V_{OUT} RANGES

High Range: $V_{OUT} = V_{DD} - 4V_T$

Low Range: $V_{OUT} = V_{DD} - 2V_T$

The temperature sense circuit is integrated with the Fixed Voltage Reference (FVR) module. See **Section 13.0 "Fixed Voltage Reference (FVR)"** for more information.

The circuit is enabled by setting the TSEN bit of the FVRCON register. When disabled, the circuit draws no current.

The circuit operates in either high or low range. The high range, selected by setting the TSRNG bit of the FVRCON register, provides a wider output voltage. This provides more resolution over the temperature range, but may be less consistent from part to part. This range requires a higher bias voltage to operate and thus, a higher V_{DD} is needed.

The low range is selected by clearing the TSRNG bit of the FVRCON register. The low range generates a lower voltage drop and thus, a lower bias voltage is needed to operate the circuit. The low range is provided for low voltage operation.

FIGURE 14-1: TEMPERATURE CIRCUIT DIAGRAM
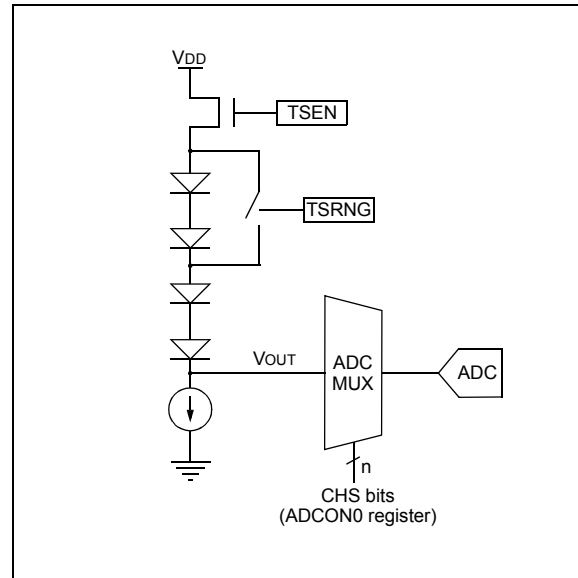
14.2 Minimum Operating V_{DD} vs. Minimum Sensing Temperature

When the temperature circuit is operated in low range, the device may be operated at any operating voltage that is within specifications.

When the temperature circuit is operated in high range, the device operating voltage, V_{DD}, must be high enough to ensure that the temperature circuit is correctly biased.

Table 14-1 shows the recommended minimum V_{DD} vs. range setting.

TABLE 14-1: RECOMMENDED V_{DD} VS. RANGE

Min. V_{DD}, TSRNG = 1	Min. V_{DD}, TSRNG = 0
3.6V	1.8V

14.3 Temperature Output

The output of the circuit is measured using the internal Analog-to-Digital Converter. A channel is reserved for the temperature circuit output. Refer to **Section 15.0 "Analog-to-Digital Converter (ADC) Module"** for detailed information.

14.4 ADC Acquisition Time

To ensure accurate temperature measurements, the user must wait at least 200 μs after the ADC input multiplexer is connected to the temperature indicator output before the conversion is performed. In addition, the user must wait 200 μs between sequential conversions of the temperature indicator output.

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 = F$_{OSC}$/2
001 = F$_{OSC}$/8
010 = F$_{OSC}$/32
011 = F$_{RC}$ (clock supplied from a dedicated RC oscillator)
100 = F$_{OSC}$/4
101 = F$_{OSC}$/16
110 = F$_{OSC}$/64
111 = F$_{RC}$ (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 = V$_{REF}$+ is connected to V$_{DD}$
01 = Reserved
10 = V$_{REF}$+ is connected to external V$_{REF}$+ pin[1]
11 = Reserved

Note 1: When selecting the FVR or the V$_{REF}$+ pin as the source of the positive reference, be aware that a minimum voltage specification exists. See **Section 22.0 "Electrical Specifications"** for details.

REGISTER 15-5:   ADRESH: ADC RESULT REGISTER HIGH (ADRESH) ADFM = 1

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
—	—	—	—	—	—	ADRES<9:8>	
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 7-2   **Reserved**: Do not use.

bit 1-0   **ADRES<9:8>**: ADC Result Register bits
Upper two bits of 10-bit conversion result

REGISTER 15-6:   ADRESL: ADC RESULT REGISTER LOW (ADRESL) ADFM = 1

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
ADRES<7:0>							
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 7-0   **ADRES<7:0>**: ADC Result Register bits
Lower eight bits of 10-bit conversion result

16.0 TIMER0 MODULE

The Timer0 module is an 8-bit timer/counter with the following features:

- 8-bit timer/counter register (TMR0)
- 8-bit prescaler (independent of Watchdog Timer)
- Programmable internal or external clock source
- Programmable external clock edge selection
- Interrupt on overflow
- TMR0 can be used to gate Timer1

Figure 16-1 is a block diagram of the Timer0 module.

16.1 Timer0 Operation

The Timer0 module can be used as either an 8-bit timer or an 8-bit counter.

16.1.1 8-BIT TIMER MODE

The Timer0 module will increment every instruction cycle, if used without a prescaler. 8-Bit Timer mode is selected by clearing the TMR0CS bit of the OPTION_REG register.

When TMR0 is written, the increment is inhibited for two instruction cycles immediately following the write.

Note: The value written to the TMR0 register can be adjusted, in order to account for the two instruction cycle delay when TMR0 is written.

16.1.2 8-BIT COUNTER MODE
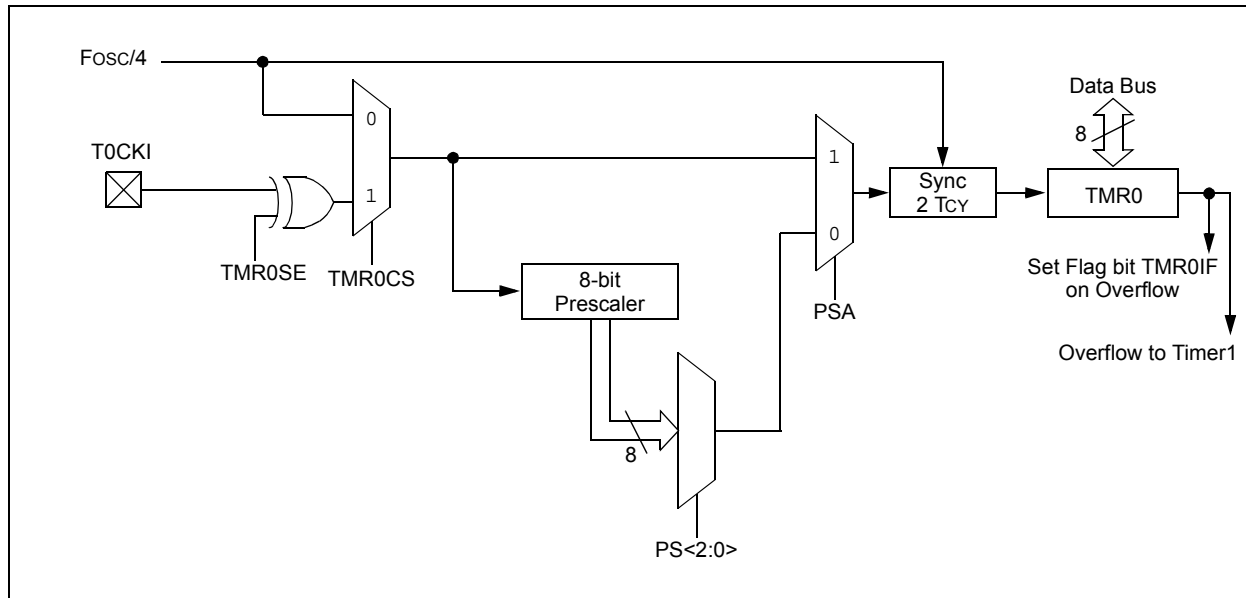
In 8-Bit Counter mode, the Timer0 module will increment on every rising or falling edge of the T0CKI pin.

8-Bit Counter mode using the T0CKI pin is selected by setting the TMR0CS bit in the OPTION_REG register to '1'.

The rising or falling transition of the incrementing edge is determined by the TMR0SE bit in the OPTION_REG register.

FIGURE 16-1: BLOCK DIAGRAM OF THE TIMER0

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

<u>If TMR1ON = 0</u>:
This bit is ignored
<u>If TMR1ON = 1</u>:
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

PIC16LF1904/6/7

18.0 ENHANCED UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER (EUSART)

The Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART) module is a serial I/O communications peripheral. It contains all the clock generators, shift registers and data buffers necessary to perform an input or output serial data transfer independent of device program execution. The EUSART, also known as a Serial Communications Interface (SCI), can be configured as a full-duplex asynchronous system or half-duplex synchronous system. Full-Duplex mode is useful for communications with peripheral systems, such as CRT terminals and personal computers. Half-Duplex Synchronous mode is intended for communications with peripheral devices, such as A/D or D/A integrated circuits, serial EEPROMs or other microcontrollers. These devices typically do not have internal clocks for baud rate generation and require the external clock signal provided by a master synchronous device.

The EUSART module includes the following capabilities:

- Full-duplex asynchronous transmit and receive
- Two-character input buffer
- One-character output buffer
- Programmable 8-bit or 9-bit character length
- Address detection in 9-bit mode
- Input buffer overrun error detection
- Received character framing error detection
- Half-duplex synchronous master
- Half-duplex synchronous slave
- Programmable clock and data polarity

The EUSART module implements the following additional features, making it ideally suited for use in Local Interconnect Network (LIN) bus systems:

- Automatic detection and calibration of the baud rate
- Wake-up on Break reception
- 13-bit Break character transmit

Block diagrams of the EUSART transmitter and receiver are shown in Figure 18-1 and Figure 18-2.

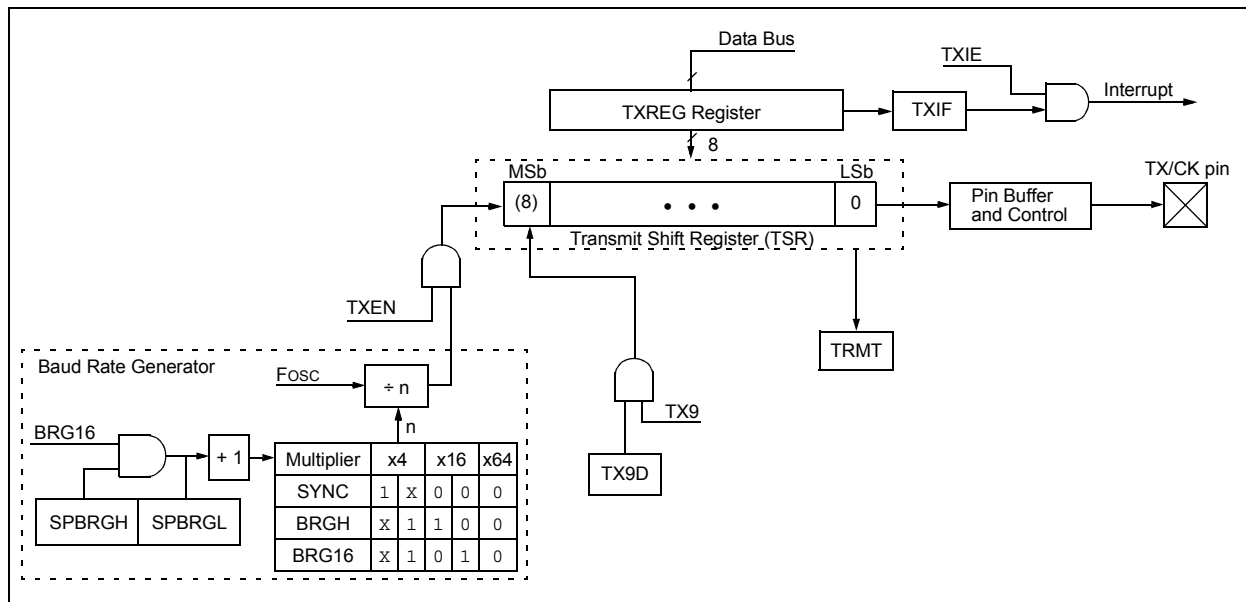**FIGURE 18-1: EUSART TRANSMIT BLOCK DIAGRAM**

PIC16LF1904/6/7

TABLE 18-5: BAUD RATES FOR ASYNCHRONOUS MODES

BAUD RATE	SYNC = 0, BRGH = 0, BRG16 = 0											
	Fosc = 20.000 MHz			Fosc = 18.432 MHz			Fosc = 16.000 MHz			Fosc = 11.0592 MHz		
	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)
300	—	—	—	—	—	—	—	—	—	—	—	—
1200	1221	1.73	255	1200	0.00	239	1202	0.16	207	1200	0.00	143
2400	2404	0.16	129	2400	0.00	119	2404	0.16	103	2400	0.00	71
9600	9470	-1.36	32	9600	0.00	29	9615	0.16	25	9600	0.00	17
10417	10417	0.00	29	10286	-1.26	27	10417	0.00	23	10165	-2.42	16
19.2k	19.53k	1.73	15	19.20k	0.00	14	19.23k	0.16	12	19.20k	0.00	8
57.6k	—	—	—	57.60k	0.00	7	—	—	—	57.60k	0.00	2
115.2k	—	—	—	—	—	—	—	—	—	—	—	—

BAUD RATE	SYNC = 0, BRGH = 0, BRG16 = 0											
	Fosc = 8.000 MHz			Fosc = 4.000 MHz			Fosc = 3.6864 MHz			Fosc = 1.000 MHz		
	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)
300	—	—	—	300	0.16	207	300	0.00	191	300	0.16	51
1200	1202	0.16	103	1202	0.16	51	1200	0.00	47	1202	0.16	12
2400	2404	0.16	51	2404	0.16	25	2400	0.00	23	—	—	—
9600	9615	0.16	12	—	—	—	9600	0.00	5	—	—	—
10417	10417	0.00	11	10417	0.00	5	—	—	—	—	—	—
19.2k	—	—	—	—	—	—	19.20k	0.00	2	—	—	—
57.6k	—	—	—	—	—	—	57.60k	0.00	0	—	—	—
115.2k	—	—	—	—	—	—	—	—	—	—	—	—

BAUD RATE	SYNC = 0, BRGH = 1, BRG16 = 0											
	Fosc = 20.000 MHz			Fosc = 18.432 MHz			Fosc = 16.000 MHz			Fosc = 11.0592 MHz		
	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)
300	—	—	—	—	—	—	—	—	—	—	—	—
1200	—	—	—	—	—	—	—	—	—	—	—	—
2400	—	—	—	—	—	—	—	—	—	—	—	—
9600	9615	0.16	129	9600	0.00	119	9615	0.16	103	9600	0.00	71
10417	10417	0.00	119	10378	-0.37	110	10417	0.00	95	10473	0.53	65
19.2k	19.23k	0.16	64	19.20k	0.00	59	19.23k	0.16	51	19.20k	0.00	35
57.6k	56.82k	-1.36	21	57.60k	0.00	19	58.82k	2.12	16	57.60k	0.00	11
115.2k	113.64k	-1.36	10	115.2k	0.00	9	111.1k	-3.55	8	115.2k	0.00	5

19.0 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 PIC16LF1904/6/7 device, the module drives the panels of up to four commons and up to 116 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 (PIC16LF1906 only)
- 29 Segment pins (PIC16LF1904/7 only)
- Static, 1/2 or 1/3 LCD Bias

> **Note:** COM3 and SEG15 share the same physical pin on the PIC16LF1906, therefore SEG15 is not available when using 1/4 multiplex displays.

19.1 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 (LCDCST)
- LCD Reference Voltage Control register (LCDREF)
- Up to 4 LCD Segment Enable registers (LCDSEn)
- Up to 16 LCD data registers (LCDDATAn)

FIGURE 19-1: LCD DRIVER MODULE BLOCK DIAGRAM

Note 1: These are not directly connected to the I/O pads, but may be tri-stated, depending on the configuration of the LCD module.

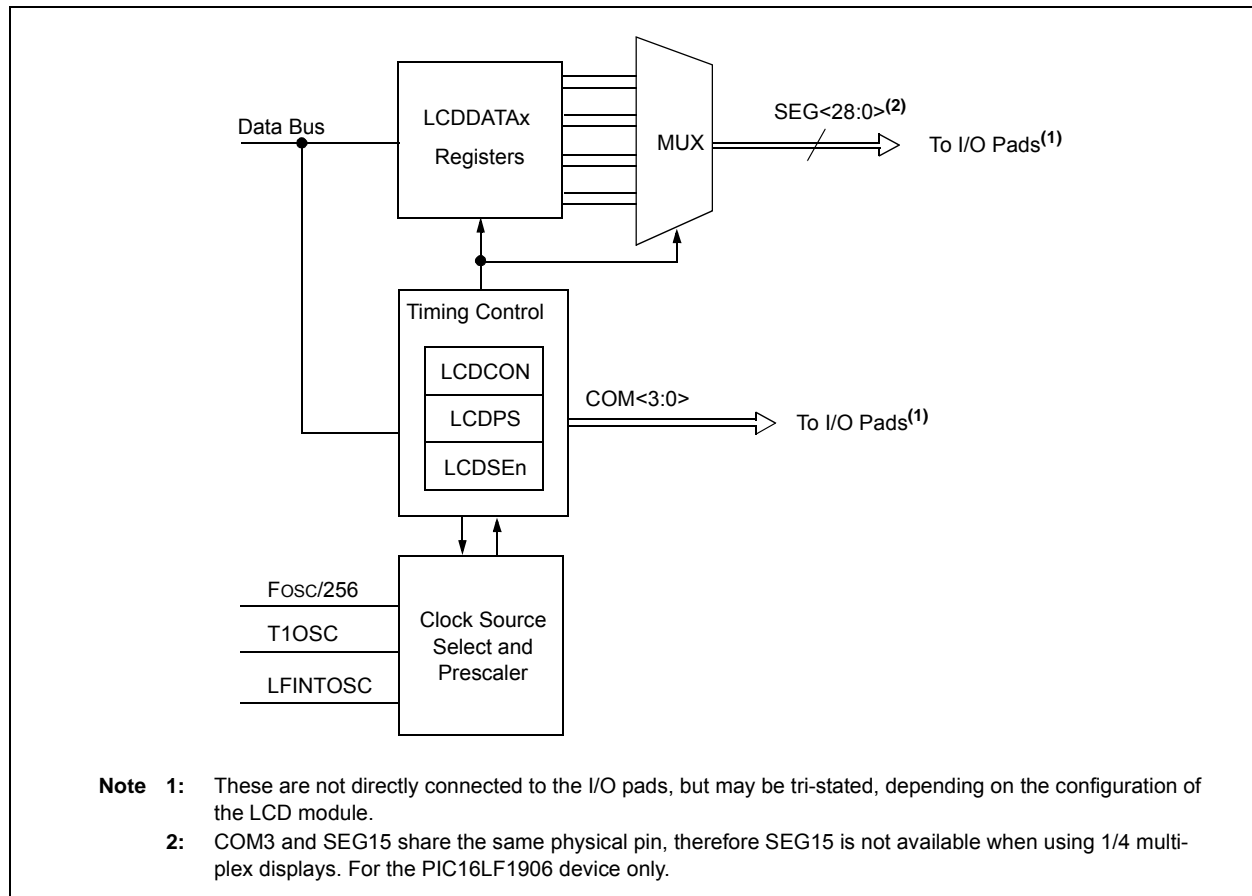2: COM3 and SEG15 share the same physical pin, therefore SEG15 is not available when using 1/4 multiplex displays. For the PIC16LF1906 device only.

19.2 LCD Clock Source Selection

The LCD module has three possible clock sources:

- FOSC/256
- T1OSC
- LFINTOSC

The first clock source is the system clock divided by 256 (FOSC/256). This divider ratio is chosen to provide about 1 kHz output when the system clock is 8 MHz. The divider is not programmable. Instead, the LCD prescaler bits LP<3:0> of the LCDPS register are used to set the LCD frame clock rate.

The second clock source is the T1OSC. This also gives about 1 kHz when a 32.768 kHz crystal is used with the Timer1 oscillator. To use the Timer1 oscillator as a clock source, the T1OSCEN bit of the T1CON register should be set.

The third clock source is the 31 kHz LFINTOSC, which provides approximately 1 kHz output.

The second and third clock sources may be used to continue running the LCD while the processor is in Sleep.

Using bits CS<1:0> of the LCDCON register can select any of these clock sources.

19.2.1 LCD PRESCALER

A 4-bit counter is available as a prescaler for the LCD clock. The prescaler is not directly readable or writable; its value is set by the LP<3:0> bits of the LCDPS register, which determine the prescaler assignment and prescale ratio.

The prescale values are selectable from 1:1 through 1:16.

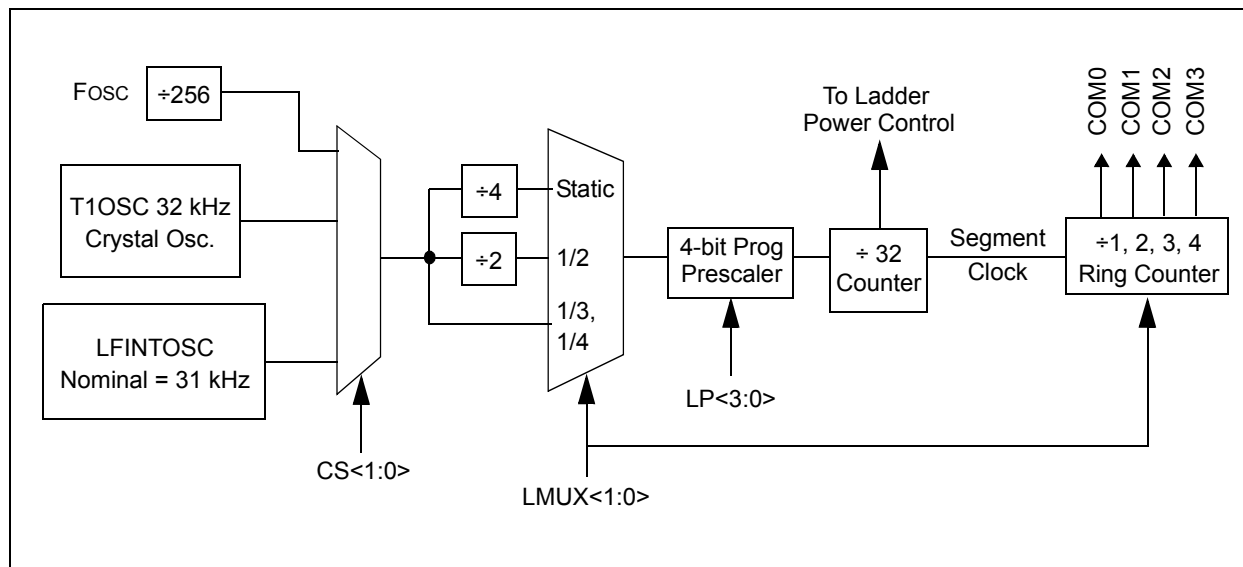**FIGURE 19-2: LCD CLOCK GENERATION**

TABLE 19-8: **SUMMARY OF REGISTERS ASSOCIATED WITH LCD OPERATION**

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	65
LCDCON	LCDEN	SLPEN	WERR	—	CS1	CS0	LMUX<1:0>		173
LCDCST	—	—	—	—	—	LCDCST<2:0>			176
LCDDATA0	SEG7 COM0	SEG6 COM0	SEG5 COM0	SEG4 COM0	SEG3 COM0	SEG2 COM0	SEG1 COM0	SEG0 COM0	177
LCDDATA1	SEG15 COM0	SEG14 COM0	SEG13 COM0	SEG12 COM0	SEG11 COM0	SEG10 COM0	SEG9 COM0	SEG8 COM0	177
LCDDATA2[1]	SEG23 COM0	SEG22 COM0	SEG21 COM0	SEG20 COM0	SEG19 COM0	SEG18 COM0	SEG17 COM0	SEG16 COM0	177
LCDDATA3	SEG7 COM1	SEG6 COM1	SEG5 COM1	SEG4 COM1	SEG3 COM1	SEG2 COM1	SEG1 COM1	SEG0 COM1	177
LCDDATA4	SEG15 COM1	SEG14 COM1	SEG13 COM1	SEG12 COM1	SEG11 COM1	SEG10 COM1	SEG9 COM1	SEG8 COM1	177
LCDDATA5[1]	SEG23 COM1	SEG22 COM1	SEG21 COM1	SEG20 COM1	SEG19 COM1	SEG18 COM1	SEG17 COM1	SEG16 COM1	177
LCDDATA6	SEG7 COM2	SEG6 COM2	SEG5 COM2	SEG4 COM2	SEG3 COM2	SEG2 COM2	SEG1 COM2	SEG0 COM2	177
LCDDATA7	SEG15 COM2	SEG14 COM2	SEG13 COM2	SEG12 COM2	SEG11 COM2	SEG10 COM2	SEG9 COM2	SEG8 COM2	177
LCDDATA8[1]	SEG23 COM2	SEG22 COM2	SEG21 COM2	SEG20 COM2	SEG19 COM2	SEG18 COM2	SEG17 COM2	SEG16 COM2	177
LCDDATA9	SEG7 COM3	SEG6 COM3	SEG5 COM3	SEG4 COM3	SEG3 COM3	SEG2 COM3	SEG1 COM3	SEG0 COM3	177
LCDDATA10	SEG15 COM3	SEG14 COM3	SEG13 COM3	SEG12 COM3	SEG11 COM3	SEG10 COM3	SEG9 COM3	SEG8 COM3	177
LCDDATA11[1]	SEG23 COM3	SEG22 COM3	SEG20 COM3	SEG19 COM3	SEG18 COM3	SEG17 COM3	SEG16 COM3	SEG15 COM3	177
LCDDATA12	—	—	—	SEG28 COM0	SEG27 COM0	SEG26 COM0	SEG25 COM0	SEG24 COM0	177
LCDDATA15	—	—	—	SEG28 COM1	SEG27 COM1	SEG26 COM1	SEG25 COM1	SEG24 COM1	177
LCDDATA18	—	—	—	SEG28 COM2	SEG27 COM2	SEG26 COM2	SEG25 COM2	SEG24 COM2	177
LCDDATA21	—	—	—	SEG28 COM3	SEG27 COM3	SEG26 COM3	SEG25 COM3	SEG24 COM3	177
LCDPS	WFT	BIASMD	LCDA	WA	LP<3:0>				174
LCDREF	LCDIRE	—	LCDIRI	—	VLCD3PE	VLCD2PE	VLCD1PE	—	175
LCDRL	LRLAP<1:0>		LRLBP<1:0>		—	LRLAT<2:0>			184
LCDSE0	SE<7:0>								177
LCDSE1	SE<15:8>								177
LCDSE2	SE<23:16>								177
LCDSE3	—	—	—	SE<28:24>					177
PIE2	—	—	—	—	—	LCDIE	—	—	67
PIR2	—	—	—	—	—	LCDIF	—	—	69
T1CON	TMR1CS1	TMR1CS0	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	—	TMR1ON	139

Legend: — = unimplemented location, read as '0'. Shaded cells are not used by the LCD module.
Note 1: PIC16LF1904/7 only.
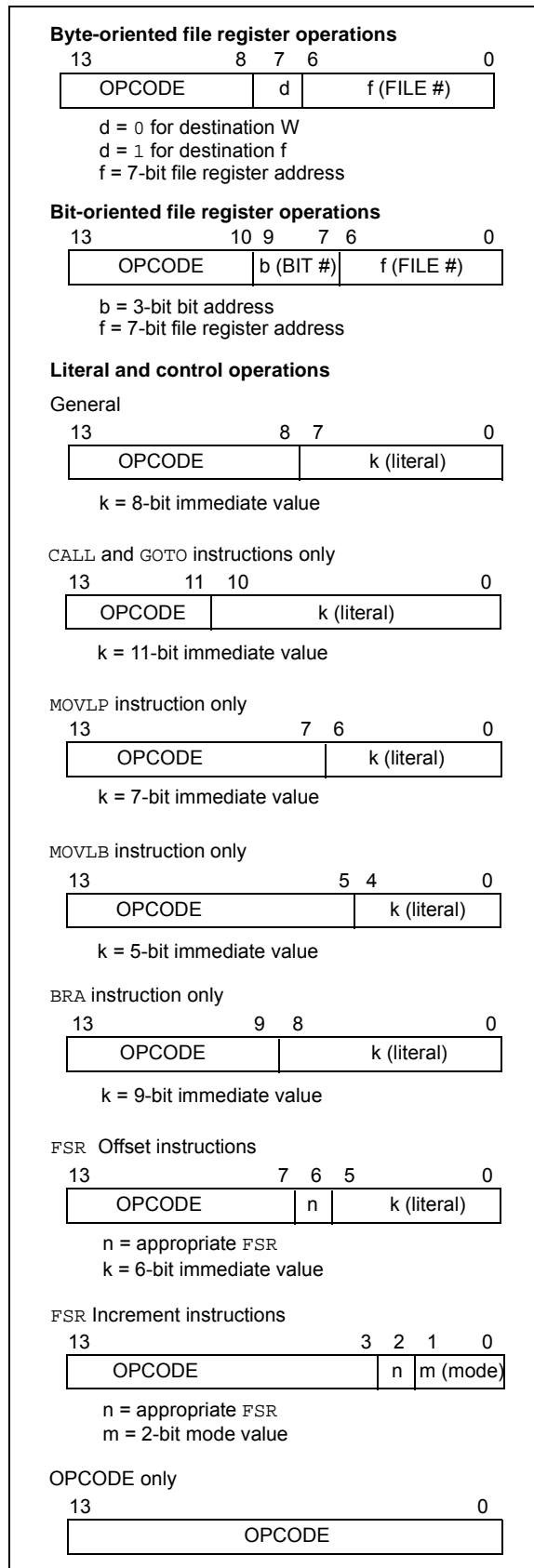
FIGURE 21-1: GENERAL FORMAT FOR INSTRUCTIONS

Byte-oriented file register operations

```
13              8  7  6              0
┌──────────────┬───┬────────────────┐
│   OPCODE     │ d │   f (FILE #)    │
└──────────────┴───┴────────────────┘
```

d = 0 for destination W
d = 1 for destination f
f = 7-bit file register address

Bit-oriented file register operations

```
13           10 9    7  6              0
┌─────────────┬───────┬────────────────┐
│   OPCODE    │b (BIT#)│  f (FILE #)    │
└─────────────┴───────┴────────────────┘
```

b = 3-bit bit address
f = 7-bit file register address

Literal and control operations

General

```
13                8  7              0
┌─────────────────┬─────────────────┐
│    OPCODE       │   k (literal)    │
└─────────────────┴─────────────────┘
```

k = 8-bit immediate value

CALL and GOTO instructions only

```
13           11 10                   0
┌─────────────┬───────────────────────┐
│   OPCODE    │     k (literal)        │
└─────────────┴───────────────────────┘
```

k = 11-bit immediate value

MOVLP instruction only

```
13                  7  6              0
┌───────────────────┬─────────────────┐
│     OPCODE        │   k (literal)    │
└───────────────────┴─────────────────┘
```

k = 7-bit immediate value

MOVLB instruction only

```
13                    5  4            0
┌─────────────────────┬────────────────┐
│      OPCODE         │   k (literal)   │
└─────────────────────┴────────────────┘
```

k = 5-bit immediate value

BRA instruction only

```
13               9  8                 0
┌────────────────┬───────────────────┐
│    OPCODE      │    k (literal)     │
└────────────────┴───────────────────┘
```

k = 9-bit immediate value

FSR Offset instructions

```
13               7  6  5              0
┌────────────────┬───┬─────────────────┐
│    OPCODE      │ n │   k (literal)    │
└────────────────┴───┴─────────────────┘
```

n = appropriate FSR
k = 6-bit immediate value

FSR Increment instructions

```
13                    3  2  1  0
┌─────────────────────┬───┬──────┐
│      OPCODE         │ n │m(mode)│
└─────────────────────┴───┴──────┘
```

n = appropriate FSR
m = 2-bit mode value

OPCODE only

```
13                                   0
┌─────────────────────────────────────┐
│              OPCODE                  │
└─────────────────────────────────────┘
```

FIGURE 22-8: **MINIMUM PULSE WIDTH FOR LPBOR DETECTION**
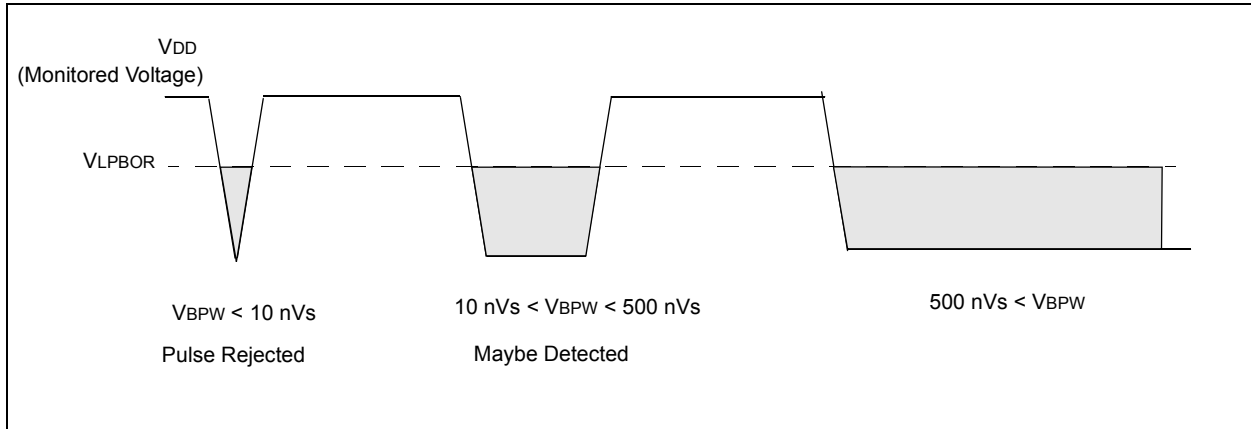
PIC16LF1904/6/7

TABLE 22-10: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, POWER-UP TIMER AND BROWN-OUT RESET PARAMETERS

| Standard Operating Conditions (unless otherwise stated) |

Param No.	Sym.	Characteristic	Min.	Typ†	Max.	Units	Conditions
30	TMCL	$\overline{MCLR}$ Pulse Width (low)	2 5	— —	— —	μs μs	VDD = 3.0V, -40°C to +85°C VDD = 3.0V
31	FWDTLP	Low Frequency Internal Oscillator Frequency	19	33	52	kHz	
32	TOST	Oscillator Start-up Timer Period[1]	—	1024	—	Tosc	**(Note 2)**
33*	TPWRT	Power-up Timer Period, $\overline{PWRTE}$ = 0	—	2048	—	Tosc	Clocked by LFINTOSC
34*	TIOZ	I/O High-Impedance from $\overline{MCLR}$ Low or Watchdog Timer Reset	—	—	2.0	μs	
35	VBOR	Brown-out Reset Voltage: BORV = 0 BORV = 1	2.55 1.80	2.70 1.90	2.85 2.05	V V	
35A*	VHYST	Brown-out Reset Hysteresis	25 —	50 —	75 100	mV mV	-40°C to +85°C -40°C to +125°C
35B*	TBORDC	Brown-out Reset DC Response Time	1 —	3 —	5 10	μs μs	VDD ≤ VBOR, -40°C to +85°C VDD ≤ VBOR
35C	TBORAC	Brown-out Reset AC Response Time	—	100	—	ns	Transient Response immunity for a noise spike that goes from VDD to VSS and back with 10 ns rise and fall times. Guidance only.
36	TFVRS	Fixed Voltage Reference Turn-on Time	—	—	5	μs	Turn on to specified stability
37	VLPBOR	Low-Power Brown-out Reset Voltage	1.85	1.95	2.10	V	-40°C to +85°C
38*	VZPHYST	Zero-Power Brown-out Reset Hysteresis	0	25	60	mV	-40°C to +85°C
39*	TZPBPW	Zero-Power Brown-out Reset AC Response Time for BOR detection	10	—	500	nVs	VDD ≤ VBOR, -40°C to +85°C

* 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: Instruction cycle period (TCY) equals four times the input oscillator time base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at "min" values with an external clock applied to the OSC1 pin. When an external clock input is used, the "max" cycle time limit is "DC" (no clock) for all devices.

2: Period of the slower clock.

3: To ensure these voltage tolerances, VDD and VSS must be capacitively decoupled as close to the device as possible. 0.1 μF and 0.01 μF values in parallel are recommended.

24.6 MPLAB X SIM Software Simulator

The MPLAB X SIM Software Simulator allows code development in a PC-hosted environment by simulating the PIC MCUs and dsPIC DSCs on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a comprehensive stimulus controller. Registers can be logged to files for further run-time analysis. The trace buffer and logic analyzer display extend the power of the simulator to record and track program execution, actions on I/O, most peripherals and internal registers.

The MPLAB X SIM Software Simulator fully supports symbolic debugging using the MPLAB XC Compilers, and the MPASM and MPLAB Assemblers. The software simulator offers the flexibility to develop and debug code outside of the hardware laboratory environment, making it an excellent, economical software development tool.

24.7 MPLAB REAL ICE In-Circuit Emulator System

The MPLAB REAL ICE In-Circuit Emulator System is Microchip's next generation high-speed emulator for Microchip Flash DSC and MCU devices. It debugs and programs all 8, 16 and 32-bit MCU, and DSC devices with the easy-to-use, powerful graphical user interface of the MPLAB X IDE.

The emulator is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with either a connector compatible with in-circuit debugger systems (RJ-11) or with the new high-speed, noise tolerant, Low-Voltage Differential Signal (LVDS) interconnection (CAT5).

The emulator is field upgradable through future firmware downloads in MPLAB X IDE. MPLAB REAL ICE offers significant advantages over competitive emulators including full-speed emulation, run-time variable watches, trace analysis, complex breakpoints, logic probes, a ruggedized probe interface and long (up to three meters) interconnection cables.

24.8 MPLAB ICD 3 In-Circuit Debugger System

The MPLAB ICD 3 In-Circuit Debugger System is Microchip's most cost-effective, high-speed hardware debugger/programmer for Microchip Flash DSC and MCU devices. It debugs and programs PIC Flash microcontrollers and dsPIC DSCs with the powerful, yet easy-to-use graphical user interface of the MPLAB IDE.

The MPLAB ICD 3 In-Circuit Debugger probe is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with a connector compatible with the MPLAB ICD 2 or MPLAB REAL ICE systems (RJ-11). MPLAB ICD 3 supports all MPLAB ICD 2 headers.

24.9 PICkit 3 In-Circuit Debugger/ Programmer

The MPLAB PICkit 3 allows debugging and programming of PIC and dsPIC Flash microcontrollers at a most affordable price point using the powerful graphical user interface of the MPLAB IDE. The MPLAB PICkit 3 is connected to the design engineer's PC using a full-speed USB interface and can be connected to the target via a Microchip debug (RJ-11) connector (compatible with MPLAB ICD 3 and MPLAB REAL ICE). The connector uses two device I/O pins and the Reset line to implement in-circuit debugging and In-Circuit Serial Programming™ (ICSP™).

24.10 MPLAB PM3 Device Programmer

The MPLAB PM3 Device Programmer is a universal, CE compliant device programmer with programmable voltage verification at V_{DDMIN} and V_{DDMAX} for maximum reliability. It features a large LCD display (128 x 64) for menus and error messages, and a modular, detachable socket assembly to support various package types. The ICSP cable assembly is included as a standard item. In Stand-Alone mode, the MPLAB PM3 Device Programmer can read, verify and program PIC devices without a PC connection. It can also set code protection in this mode. The MPLAB PM3 connects to the host PC via an RS-232 or USB cable. The MPLAB PM3 has high-speed communications and optimized algorithms for quick programming of large memory devices, and incorporates an MMC card for file storage and data applications.