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	32MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LCD, POR, PWM, WDT
Number of I/O	43
Program Memory Size	14KB (8K x 14)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 39x12b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	48-TQFP
Supplier Device Package	48-TQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lf19185-i-pt

9.0 OSCILLATOR MODULE (WITH FAIL-SAFE CLOCK MONITOR)

9.1 Overview

The oscillator module has a wide variety of clock sources and selection features that allow it to be used in a wide range of applications while maximizing performance and minimizing power consumption. Figure 9-1 illustrates a block diagram of the oscillator module.

Clock sources can be supplied from external oscillators. In addition, the system clock source can be supplied from one of two internal oscillators and PLL circuits, with a choice of speeds selectable via software. Additional clock features include:

- Selectable system clock source between external or internal sources via software.
- Fail-Safe Clock Monitor (FSCM) designed to detect a failure of the external clock source (ECH, ECM, ECL) and switch automatically to the internal oscillator.
- Oscillator Start-up Timer (OST) ensures stability of SOSC.

The RSTOSC bits of Configuration Word 1 determine the type of oscillator that will be used when the device is reset, including when it is first powered up.

The internal clock configuration, LFINTOSC, HFINTOSC (set at 1 MHz), or HFINTOSC (set at 32 MHz) can be set through the RSTOSC bits.

If an external clock source is selected, the FEXTOSC bits of Configuration Word 1 must be used to select the external clock mode.

The external oscillator module can be configured in one of the following clock modes, by setting the FEXTOSC<2:0> bits of Configuration Word 1:

1. ECL – External Clock Low-Power mode ECL<= 500 kHz
2. ECM – External Clock Medium Power mode ECM <= 8 MHz
3. ECH – External Clock High-Power mode ECH <= 32 MHz

The ECH, ECM, and ECL Clock modes rely on an external logic level signal as the device clock source. Each mode is optimized for a different frequency range.

The INTOSC internal oscillator block produces low and high-frequency clock sources, designated LFINTOSC and HFINTOSC. (see Internal Oscillator Block, Figure 9-1). A wide selection of device clock frequencies may be derived from these clock sources.

TABLE 9-3: SUMMARY OF REGISTERS ASSOCIATED WITH CLOCK SOURCES

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
OSCCON1	—	NOSC<2:0>			NDIV<3:0>				152
OSCCON2	—	COSC<2:0>			CDIV<3:0>				152
OSCCON3	CWSHOLD	SOSCPWR	—	ORDY	NOSCR	—	—	—	154
OSCFRQ	—	—	—	—	—	HFFRQ<2:0>			157
OSCSTAT	EXTOR	HFOR	MFOR	LFOR	SOR	ADOR	—	PLLR	155
OSCTUNE	—	—	HFTUN<5:0>						157
OSCEN	EXTOEN	HFOEN	MFOEN	LFOEN	SOSCEN	ADOEN	—	—	156
ACTCON	ACTEN	ACTUD	—	—	ACTLOCK	—	ACTORS		158

Legend: — = unimplemented location, read as '0'. Shaded cells are not used by clock sources.

TABLE 9-4: SUMMARY OF CONFIGURATION WORD WITH CLOCK SOURCES

Name	Bits	Bit -/7	Bit -/6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	Register on Page
CONFIG1	13:8	—	—	FCMEN	—	CSWEN	LCDPEN	$\overline{\text{VBATEN}}$	$\overline{\text{CLKOUTEN}}$	120
	7:0	—	RSTOSC<2:0>			—	FEXTOSC<2:0>			

Legend: — = unimplemented location, read as '0'. Shaded cells are not used by clock sources.

EXAMPLE 13-4: WRITING TO PROGRAM FLASH MEMORY (PFM)

```
; This write routine assumes the following:
; 1. 32 bytes of data are loaded, starting at the address in DATA_ADDR
; 2. Each word of data to be written is made up of two adjacent bytes in DATA_ADDR,
;    stored in little endian format
; 3. A valid starting address (the least significant bits = 00000) is loaded in ADDRH:ADDRL
; 4. ADDRH and ADDRL are located in common RAM (locations 0x70 - 0x7F)
; 5. NVM interrupts are not taken into account

        BANKSEL     NVMADRH
        MOVF        ADDRH,W
        MOVWF       NVMADRH             ; Load initial address
        MOVF        ADDRL,W
        MOVWF       NVMADRL
        MOVLW       LOW DATA_ADDR       ; Load initial data address
        MOVWF       FSR0L
        MOVLW       HIGH DATA_ADDR
        MOVWF       FSR0H
        BCF         NVMCON1,NVMREGS     ; Set Program Flash Memory as write location
        BSF         NVMCON1,WREN        ; Enable writes
        BSF         NVMCON1,LWLO        ; Load only write latches

LOOP
        MOVIW       FSR0++
        MOVWF       NVMDATL             ; Load first data byte
        MOVIW       FSR0++
        MOVWF       NVMDATH             ; Load second data byte

        MOVF        NVMADRL,W
        XORLW       0x1F                ; Check if lower bits of address are 00000
        ANDLW       0x1F                ; and if on last of 32 addresses
        BTFSC       STATUS,Z            ; Last of 32 words?
        GOTO        START_WRITE         ; If so, go write latches into memory

        CALL        UNLOCK_SEQ          ; If not, go load latch
        INCF        NVMADRL,F           ; Increment address
        GOTO        LOOP

START_WRITE
        BCF         NVMCON1,LWLO        ; Latch writes complete, now write memory
        CALL        UNLOCK_SEQ          ; Perform required unlock sequence
        BCF         NVMCON1,WREN        ; Disable writes

UNLOCK_SEQ
        MOVLW       55h
        BCF         INTCON,GIE          ; Disable interrupts
        MOVWF       NVMCON2             ; Begin unlock sequence
        MOVLW       AAh
        MOVWF       NVMCON2
        BSF         NVMCON1,WR
        BSF         INTCON,GIE          ; Unlock sequence complete, re-enable interrupts
        return
```

REGISTER 13-5: NVMCON1: NONVOLATILE MEMORY CONTROL 1 REGISTER

U-0	R/W-0/0	R/W-0/0	R/W/HC-0/0	R/W/HC-x/q	R/W-0/0	R/S/HC-0/0	R/S/HC-0/0
—	NVMREGS	LWLO	FREE	WRERR(1,2,3)	WREN	WR(4,5,6)	RD
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
S = Bit can only be set	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 **Unimplemented:** Read as '0'

bit 6 **NVMREGS:** Configuration Select bit
1 = Access EEPROM, DIA, DCI, Configuration, User ID and Device ID Registers
0 = Access PFM

bit 5 **LWLO:** Load Write Latches Only bit
When FREE = 0:
1 = The next WR command updates the write latch for this word within the row; no memory operation is initiated.
0 = The next WR command writes data or erases
Otherwise: The bit is ignored

bit 4 **FREE:** PFM Erase Enable bit
When NVMREGS:NVMADR points to a PFM location:
1 = Performs an erase operation with the next WR command; the 32-word pseudo-row containing the indicated address is erased (to all 1s) to prepare for writing.
0 = All erase operations have completed normally

bit 3 **WRERR:** Program/Erase Error Flag bit(1,2,3)
This bit is normally set by hardware.
1 = A write operation was interrupted by a Reset, interrupted unlock sequence, or WR was written to one while NVMADR points to a write-protected address.
0 = The program or erase operation completed normally

bit 2 **WREN:** Program/Erase Enable bit
1 = Allows program/erase cycles
0 = Inhibits programming/erasing of program Flash

bit 1 **WR:** Write Control bit(4,5,6)
When NVMREG:NVMADR points to a PFM location:
1 = Initiates the operation indicated by Table 13-4
0 = NVM program/erase operation is complete and inactive.

bit 0 **RD:** Read Control bit(7)
1 = Initiates a read at address = NVMADR1, and loads data to NVMDAT Read takes one instruction cycle and the bit is cleared when the operation is complete. The bit can only be set (not cleared) in software.
0 = NVM read operation is complete and inactive

Note 1: Bit is undefined while WR = 1.
2: Bit must be cleared by software; hardware will not clear this bit.
3: Bit may be written to '1' by software in order to implement test sequences.
4: This bit can only be set by following the unlock sequence of **Section 13.4.2 "NVM Unlock Sequence"**.
5: Operations are self-timed, and the WR bit is cleared by hardware when complete.
6: Once a write operation is initiated, setting this bit to zero will have no effect.
7: Reading from EEPROM loads only NVMDATL<7:0> (Register 13-1).

REGISTER 14-49: HIDRVF: PORTF HIGH DRIVE CONTROL REGISTER

R/W-0/0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
HIDF7	—	—	—	—	—	—	—
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 **HIDF7:** PORTF High Drive Enable bit
For RF7 pin
1 = High current source and sink enabled
0 = Standard current source and sink

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

TABLE 14-7: SUMMARY OF REGISTERS ASSOCIATED WITH PORTF

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
PORTF	RF7	RF6	RF5	RF4	RF3	RF2	RF1	RF0	255
TRISF	TRISF7	TRISF6	TRISF5	TRISF4	TRISF3	TRISF2	TRISF1	TRISF0	255
LATF	LATF7	LATF6	LATF5	LATF4	LATF3	LATF2	LATF1	LATF0	256
ANSELF	ANSF7	ANSF6	ANSF5	ANSF4	ANSF3	ANSF2	ANSF1	ANSF0	256
WPUF	WPUF7	WPUF6	WPUF5	WPUF4	WPUF3	WPUF2	WPUF1	WPUF0	256
ODCONF	ODCF7	ODCF6	ODCF5	ODCF4	ODCF3	ODCF2	ODCF1	ODCF0	257
SLRCONF	SLRF7	SLRF6	SLRF5	SLRF4	SLRF3	SLRF2	SLRF1	SLRF0	257
INLVLF	INLVLF7	INLVLF6	INLVLF5	INLVLF4	INLVLF3	INLVLF2	INLVLF1	INLVLF0	257
HIDRVF	HIDF7	—	—	—	—	—	—	—	258

Legend: x = unknown, u = unchanged, – = unimplemented locations read as '0'. Shaded cells are not used by PORTF.

TABLE 18-1: SUMMARY OF REGISTERS ASSOCIATED WITH FIXED VOLTAGE REFERENCE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR<1:0>		ADFVR<1:0>		285
ADCON0	ON	CONT	—	CS	—	FM	—	GO	305
ADCON1	PPOL	IPEN	GPOL	—	—	—	—	DSEN	306
DAC1CON0	DAC1EN	—	DAC1OE1	DAC1OE2	DAC1PSS<1:0>		—	—	332

Legend: – = unimplemented locations read as '0'. Shaded cells are not used with the Fixed Voltage Reference.

24.6 RTCC Control Registers

REGISTER 24-1: RTCCON: RTC CONTROL REGISTER

R/W-0/u	U-0	R/W-0/u	R-0/u	R-0/u	U-0	R/W-0/u	R/W-0/u
RTCEN(1)	—	RTCWREN	RTCSYNC	HALFSEC(2)	—	RTCCLKSEL1	RTCCLKSEL0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	u = Bit is unchanged
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **RTCEN:** RTC Enable bit(1)
- 1 = RTCC module is enabled
- 0 = RTCC module is disabled

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

bit 5 **RTCWREN:** RTCC Value Registers Write Enable bit
- 1 = RTCC registers can be written to by the user
- 0 = RTCC registers are locked out from being written to by the user

bit 4 **RTCSYNC:** RTCC Value Registers Read Synchronization bit
- 1 = RTCC registers can change while reading due to a rollover ripple resulting in an invalid data read; if the register is read twice and results in the same data, the data can be assumed to be valid.
- 0 = RTCC registers can be read without concern over a rollover ripple

bit 3 **HALFSEC:** Half-Second Status bit(2)
- 1 = Second half period of a second
- 0 = First half period of a second

bit 2 **Unimplemented:** Read as '0'

bit 1-0 **RTCCLKSEL<1:0>:** RTC Clock Source Selection bits
- 00 = SOSC (expected to 32.768 kHz)
- 01 = MFINTOSC (31.25 kHz)
- 10 = 50 Hz Powerline Clock (Zero-Cross Detect)
- 11 = 60 Hz Powerline Clock (Zero-Cross Detect)

Note 1: A write to the RTCEN bit is only allowed when RTCWREN = 1.

2: This bit is read-only. It is cleared to '0' on a write to the SECONDS register.

REGISTER 26-3: T1CLK TIMER1 CLOCK SELECT REGISTER

U-0	U-0	U-0	U-0	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u
—	—	—	—	CS<3:0>			
bit 7							bit 0

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

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

bit 3-0 **CS<3:0>:** Timer1 Clock Select bits

1111 = Reserved
1110 = Reserved
1101 = Reserved
1100 = LC4_out
1011 = LC3_out
1010 = LC2_out
1001 = LC1_out
1000 = Timer0 Overflow Output
0111 = SOSC
0110 = MFINTOSC (32 kHz)
0101 = MFINTOSC (500 kHz)
0100 = LFINTOSC
0011 = HFINTOSC
0010 = FOSC
0001 = FOSC/4
0000 = TxCKIPPS

TABLE 27-1: TIMER2/4 OPERATING MODES

Mode	MODE<4:0>		Output Operation	Operation	Timer Control		
	<4:3>	<2:0>			Start	Reset	Stop
Free Running Period	00	000	Period Pulse	Software gate (Figure 27-4)	ON = 1	—	ON = 0
		001		Hardware gate, active-high (Figure 27-5)	ON = 1 and TMRx_ers = 1	—	ON = 0 or TMRx_ers = 0
		010		Hardware gate, active-low	ON = 1 and TMRx_ers = 0	—	ON = 0 or TMRx_ers = 1
		011	Period Pulse with Hardware Reset	Rising or falling edge Reset	ON = 1	TMRx_ers ↕	ON = 0
		100		Rising edge Reset (Figure 27-6)		TMRx_ers ↑	
		101		Falling edge Reset		TMRx_ers ↓	
		110		Low level Reset		TMRx_ers = 0	ON = 0 or TMRx_ers = 0
		111		High level Reset (Figure 27-7)		TMRx_ers = 1	ON = 0 or TMRx_ers = 1
One-shot	01	000	One-shot	Software start (Figure 27-8)	ON = 1	—	ON = 0 or Next clock after TMRx = PRx (**Note 2**)
		001	Edge triggered start (**Note 1**)	Rising edge start (Figure 27-9)	ON = 1 and TMRx_ers ↑	—	
		010		Falling edge start	ON = 1 and TMRx_ers ↓	—	
		011		Any edge start	ON = 1 and TMRx_ers ↕	—	
		100	Edge triggered start and hardware Reset (**Note 1**)	Rising edge start and Rising edge Reset (Figure 27-10)	ON = 1 and TMRx_ers ↑	TMRx_ers ↑	
		101		Falling edge start and Falling edge Reset	ON = 1 and TMRx_ers ↓	TMRx_ers ↓	
		110		Rising edge start and Low level Reset (Figure 27-11)	ON = 1 and TMRx_ers ↑	TMRx_ers = 0	
		111		Falling edge start and High level Reset	ON = 1 and TMRx_ers ↓	TMRx_ers = 1	
Mono-stable	10	000	Reserved				
		001	Edge triggered start (**Note 1**)	Rising edge start (Figure 27-12)	ON = 1 and TMRx_ers ↑	—	ON = 0 or Next clock after TMRx = PRx (**Note 3**)
		010		Falling edge start	ON = 1 and TMRx_ers ↓	—	
		011		Any edge start	ON = 1 and TMRx_ers ↕	—	
Reserved		100	Reserved				
Reserved		101	Reserved				
One-shot		110	Level triggered start and hardware Reset	High level start and Low level Reset (Figure 27-13)	ON = 1 and TMRx_ers = 1	TMRx_ers = 0	ON = 0 or Held in Reset (**Note 2**)
		111		Low level start & High level Reset	ON = 1 and TMRx_ers = 0	TMRx_ers = 1	
Reserved	11	xxx	Reserved				

Note 1: If ON = 0 then an edge is required to restart the timer after ON = 1.

2: When TMRx = PRx then the next clock clears ON and stops TMRx at 00h.

3: When TMRx = PRx then the next clock stops TMRx at 00h but does not clear ON.

REGISTER 27-2: TxCON: TIMER2/4 CONTROL REGISTER

R/W/HC-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
ON(1)	CKPS<2:0>			OUTPS<3:0>			
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	HC = Bit is cleared by hardware

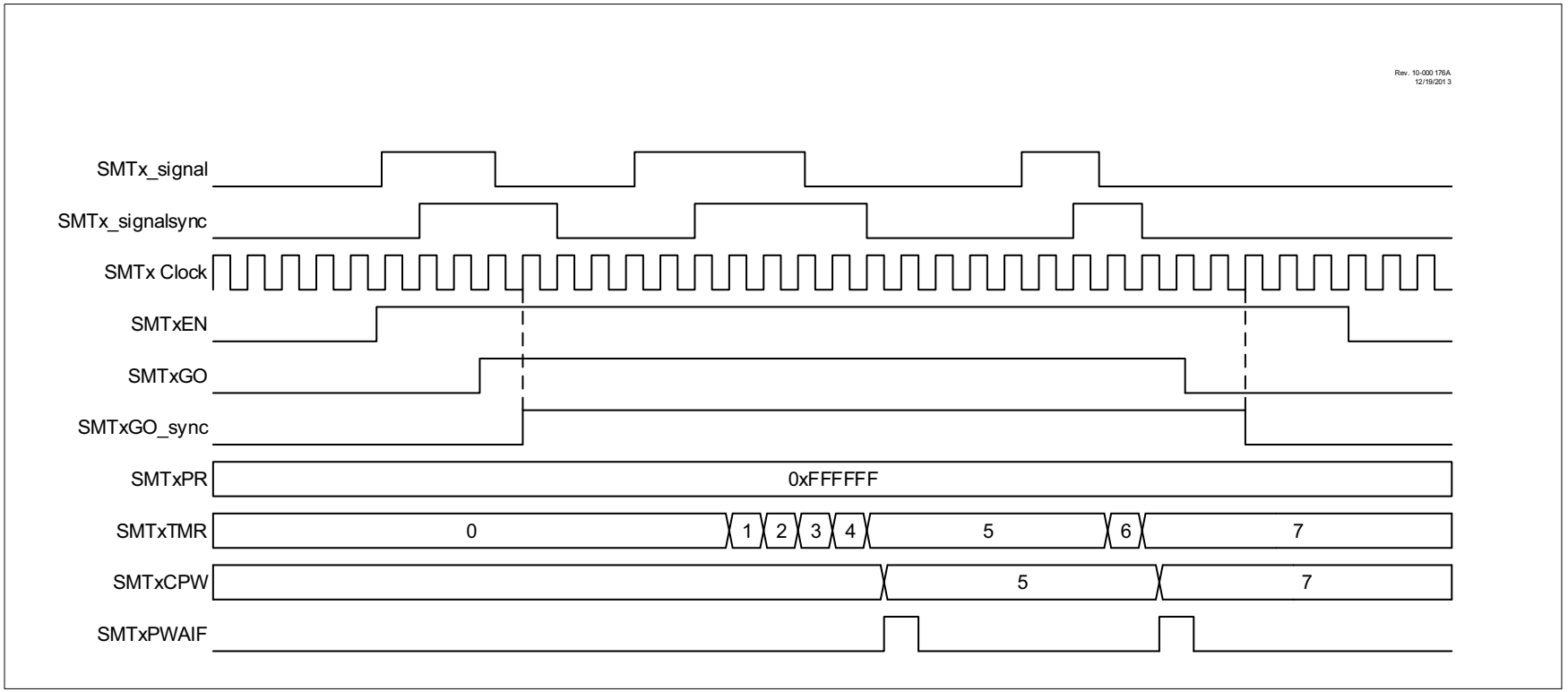
bit 7 **ON:** Timer2/4 On bit
1 = Timer2/4 is on
0 = Timer2/4 is off: all counters and state machines are reset

bit 6-4 **CKPS<2:0>:** Timer2/4-type Clock Prescale Select bits
111 = 1:128 Prescaler
110 = 1:64 Prescaler
101 = 1:32 Prescaler
100 = 1:16 Prescaler
011 = 1:8 Prescaler
010 = 1:4 Prescaler
001 = 1:2 Prescaler
000 = 1:1 Prescaler

bit 3-0 **OUTPS<3:0>:** Timer2/4 Output Postscaler Select bits
1111 = 1:16 Postscaler
1110 = 1:15 Postscaler
1101 = 1:14 Postscaler
1100 = 1:13 Postscaler
1011 = 1:12 Postscaler
1010 = 1:11 Postscaler
1001 = 1:10 Postscaler
1000 = 1:9 Postscaler
0111 = 1:8 Postscaler
0110 = 1:7 Postscaler
0101 = 1:6 Postscaler
0100 = 1:5 Postscaler
0011 = 1:4 Postscaler
0010 = 1:3 Postscaler
0001 = 1:2 Postscaler
0000 = 1:1 Postscaler

Note 1: In certain modes, the ON bit will be auto-cleared by hardware. See **Section 27.5 "Operation Examples"**.

FIGURE 28-4: GATED TIMER MODE REPEAT ACQUISITION TIMING DIAGRAM



28.7.4 HIGH AND LOW-MEASURE MODE

This mode measures the high and low-pulse time of the SMTSIGx relative to the SMT clock. It begins incrementing the SMTxTMR on a rising edge on the SMTSIGx input, then updates the SMTxCPW register with the value and resets the SMTxTMR on a falling edge, starting to increment again. Upon observing another rising edge, it updates the SMTxCPR register with its current value and once again resets the SMTxTMR value and begins incrementing again. See Figure 28-8 and Figure 28-9.

29.3.4 TIMER2 TIMER RESOURCE

This device has a newer version of the Timer2 module that has many new modes, which allow for greater customization and control of the PWM signals than on older parts. Refer to **Section 27.5 "Operation Examples"** for examples of PWM signal generation using the different modes of Timer2. The CCP operation requires that the timer used as the PWM time base has the FOSC/4 clock source selected

29.3.5 PWM PERIOD

The PWM period is specified by the PR2 register of Timer2. The PWM period can be calculated using the formula of Equation 29-1.

EQUATION 29-1: PWM PERIOD

$$PWM\ Period = [(PR2) + 1] \bullet 4 \bullet T_{OSC} \bullet (TMR2\ Prescale\ Value)$$

Note 1: TOSC = 1/FOSC

When TMR2 is equal to PR2, the following three events occur on the next increment cycle:

- TMR2 is cleared
- The CCPx pin is set. (Exception: If the PWM duty cycle = 0%, the pin will not be set.)
- The PWM duty cycle is transferred from the CCPRxL/H register pair into a 10-bit buffer.

Note:	The Timer postscaler (see **Section 27.4 "Timer2/4 Interrupt"**) is not used in the determination of the PWM frequency.

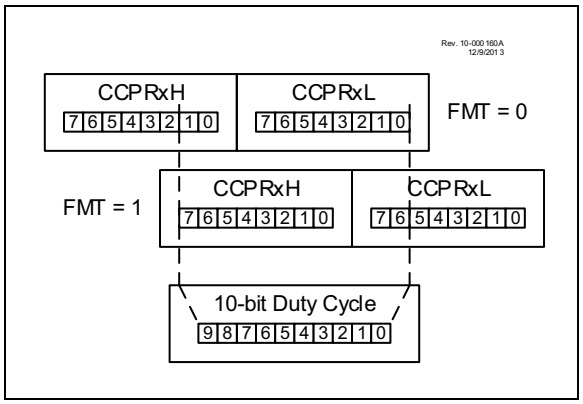
29.3.6 PWM DUTY CYCLE

The PWM duty cycle is specified by writing a 10-bit value to the CCPRxH:CCPRxL register pair. The alignment of the 10-bit value is determined by the CCPRxFMT bit of the CCPxCON register (see Figure 29-5). The CCPRxH:CCPRxL register pair can be written to at any time; however the duty cycle value is not latched into the 10-bit buffer until after a match between PR2 and TMR2.

Equation 29-2 is used to calculate the PWM pulse width.

Equation 29-3 is used to calculate the PWM duty cycle ratio.

FIGURE 29-5: PWM 10-BIT ALIGNMENT



EQUATION 29-2: PULSE WIDTH

$$Pulse\ Width = (CCPRxH{:}CCPRxL\ register\ pair) \bullet T_{OSC} \bullet (TMR2\ Prescale\ Value)$$

EQUATION 29-3: DUTY CYCLE RATIO

$$Duty\ Cycle\ Ratio = \frac{(CCPRxH{:}CCPRxL\ register\ pair)}{4(PR2 + 1)}$$

CCPRxH:CCPRxL register pair are used to double buffer the PWM duty cycle. This double buffering provides for glitchless PWM operation.

The 8-bit timer TMR2 register is concatenated with either the 2-bit internal system clock (FOSC), or two bits of the prescaler, to create the 10-bit time base. The system clock is used if the Timer2 prescaler is set to 1:1.

When the 10-bit time base matches the CCPRxH:CCPRxL register pair, then the CCPx pin is cleared (see Figure 29-4).

29.3.7 PWM RESOLUTION

The resolution determines the number of available duty cycles for a given period. For example, a 10-bit resolution will result in 1024 discrete duty cycles, whereas an 8-bit resolution will result in 256 discrete duty cycles.

The maximum PWM resolution is ten bits when PR2 is 255. The resolution is a function of the PR2 register value as shown by Equation 29-4.

EQUATION 29-4: PWM RESOLUTION

$$Resolution = \frac{log[4(PR2 + 1)]}{log(2)}\ bits$$

Note:	If the pulse-width value is greater than the period, the assigned PWM pin(s) will remain unchanged.

33.6.8 ACKNOWLEDGE SEQUENCE TIMING

An Acknowledge sequence is enabled by setting the Acknowledge Sequence Enable bit, ACKEN bit of the SSPxCON2 register. When this bit is set, the SCL pin is pulled low and the contents of the Acknowledge data bit are presented on the SDA pin. If the user wishes to generate an Acknowledge, then the ACKDT bit should be cleared. If not, the user should set the ACKDT bit before starting an Acknowledge sequence. The Baud Rate Generator then counts for one rollover period (TBRG) and the SCL pin is deasserted (pulled high). When the SCL pin is sampled high (clock arbitration), the Baud Rate Generator counts for TBRG. The SCL pin is then pulled low. Following this, the ACKEN bit is automatically cleared, the Baud Rate Generator is turned off and the MSSP module then goes into IDLE mode (Figure 33-30).

33.6.8.1 WCOL Status Flag

If the user writes the SSPxBUF when an Acknowledge sequence is in progress, then WCOL bit is set and the contents of the buffer are unchanged (the write does not occur).

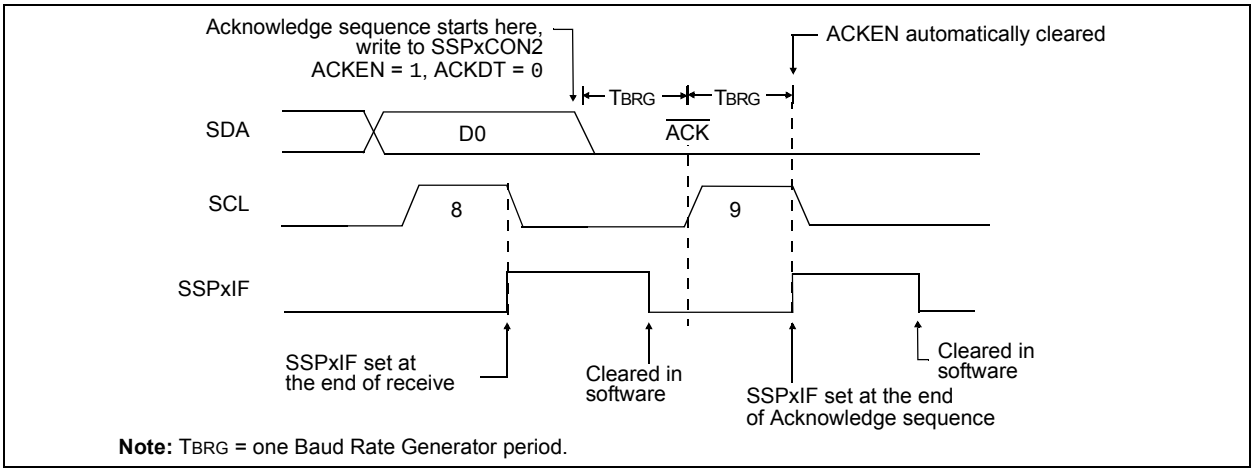
33.6.9 STOP CONDITION TIMING

A Stop bit is asserted on the SDA pin at the end of a receive/transmit by setting the Stop Sequence Enable bit, PEN bit of the SSPxCON2 register. At the end of a receive/transmit, the SCL line is held low after the falling edge of the ninth clock. When the PEN bit is set, the master will assert the SDA line low. When the SDA line is sampled low, the Baud Rate Generator is reloaded and counts down to '0'. When the Baud Rate Generator times out, the SCL pin will be brought high and one TBRG (Baud Rate Generator rollover count) later, the SDA pin will be deasserted. When the SDA pin is sampled high while SCL is high, the P bit of the SSPxSTAT register is set. A TBRG later, the PEN bit is cleared and the SSPxIF bit is set (Figure 33-31).

33.6.9.1 WCOL Status Flag

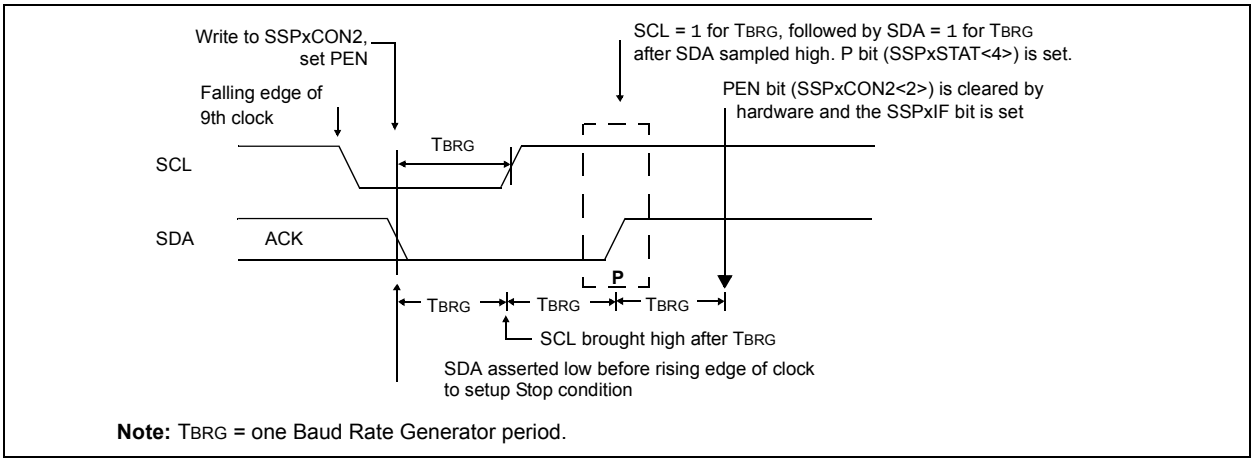
If the user writes the SSPxBUF when a Stop sequence is in progress, then the WCOL bit is set and the contents of the buffer are unchanged (the write does not occur).

FIGURE 33-30: ACKNOWLEDGE SEQUENCE WAVEFORM



Note: TBRG = one Baud Rate Generator period.

FIGURE 33-31: STOP CONDITION RECEIVE OR TRANSMIT MODE



Note: TBRG = one Baud Rate Generator period.

FIGURE 35-1: LCD CONTROLLER MODULE BLOCK DIAGRAM

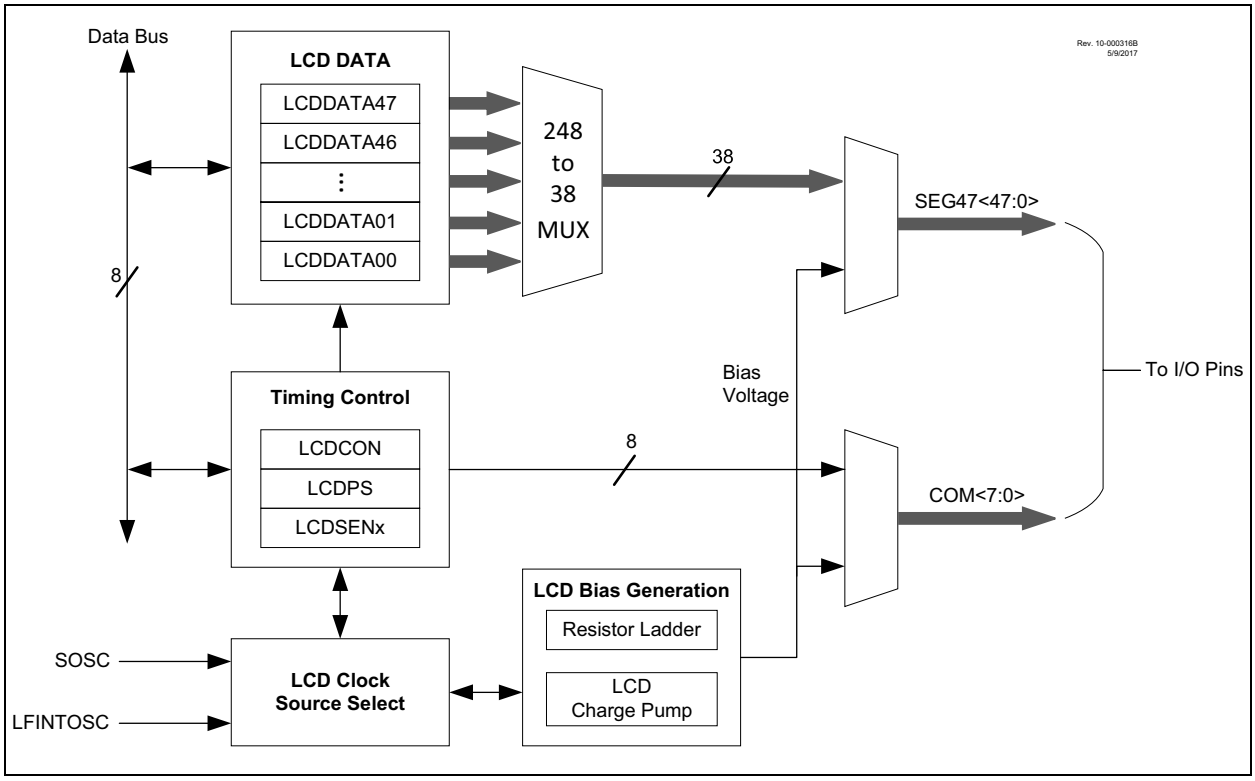
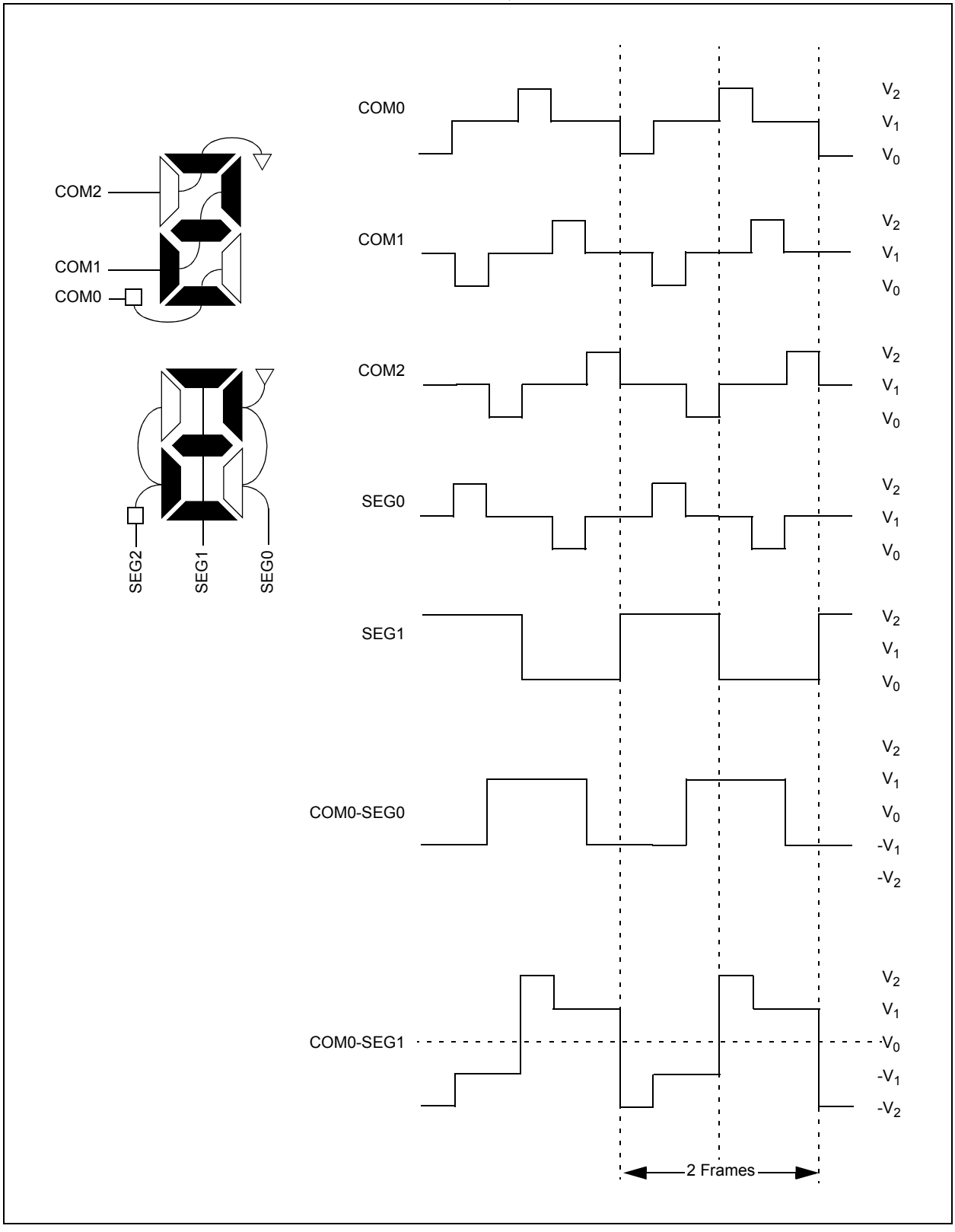


FIGURE 35-14: TYPE-B WAVEFORMS IN 1/3 MUX, 1/2 BIAS DRIVE



37.3 Instruction Descriptions

ADDFSR — Add Literal to FSRn

Syntax:	[*label*] ADDFSR FSRn, k
Operands:	-32 ≤ k ≤ 31 n ∈ [0, 1]
Operation:	FSR(n) + k → FSR(n)
Status Affected:	None
Description:	The signed 6-bit literal 'k' is added to the contents of the FSRnH:FSRnL register pair. FSRn is limited to the range 0000h-FFFFh. Moving beyond these bounds will cause the FSR to wrap-around.

ADDLW — Add literal and W

Syntax:	[*label*] ADDLW k
Operands:	0 ≤ k ≤ 255
Operation:	(W) + k → (W)
Status Affected:	C, DC, Z
Description:	The contents of the W register are added to the 8-bit literal 'k' and the result is placed in the W register.

ADDWF — Add W and f

Syntax:	[*label*] ADDWF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(W) + (f) → (destination)
Status Affected:	C, DC, Z
Description:	Add 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'.

ADDWFC — ADD W and CARRY bit to f

Syntax:	[*label*] ADDWFC f {,d}
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(W) + (f) + (C) → dest
Status Affected:	C, DC, Z
Description:	Add W, the Carry flag and data memory location 'f'. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed in data memory location 'f'.

ANDLW — AND literal with W

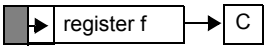
Syntax:	[*label*] ANDLW k
Operands:	0 ≤ k ≤ 255
Operation:	(W) .AND. (k) → (W)
Status Affected:	Z
Description:	The contents of W register are AND'ed with the 8-bit literal 'k'. The result is placed in the W register.

ANDWF — AND W with f

Syntax:	[*label*] ANDWF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(W) .AND. (f) → (destination)
Status Affected:	Z
Description:	AND 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'.

ASRF — Arithmetic Right Shift

Syntax:	[*label*] ASRF f {,d}
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f<7>)→ dest<7> (f<7:1>) → dest<6:0>, (f<0>) → C,
Status Affected:	C, Z
Description:	The contents of register 'f' are shifted one bit to the right through the Carry flag. The MSb remains unchanged. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is stored back in register 'f'.



TRIS **Load TRIS Register with W**

Syntax:	[*label*] TRIS f
Operands:	$5 \le f \le 7$
Operation:	(W) → 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:	$0 \le k \le 255$
Operation:	(W) .XOR. k → (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:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(W) .XOR. (f) → (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'.

TABLE 38-1: REGISTER FILE SUMMARY FOR PIC16(L)F19155/56/75/76/85/86 DEVICES

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
40Ch	—	Unimplemented								
40Dh	HIDRVB	—	—	—	—	—	—	HIDB1	—	232
40Eh	—	Unimplemented								
40Fh	—	Unimplemented								
410h	—	Unimplemented								
411h	—	Unimplemented								
412h	—	Unimplemented								
413h	—	Unimplemented								
414h	—	Unimplemented								
415h	—	Unimplemented								
416h	—	Unimplemented								
417h	—	Unimplemented								
418h	—	Unimplemented								
419h	—	Unimplemented								
41Ah	—	Unimplemented								
41Bh	—	Unimplemented								
41Ch	—	Unimplemented								
41Dh	—	Unimplemented								
41Eh	—	Unimplemented								
41Fh	—	Unimplemented								
48Ch	SMT1TMRL	SMT1TMR								416
48Dh	SMT1TMRH	SMT1TMR								416
48Eh	SMT1TMRU	SMT1TMR								416
48Fh	SMT1CPRL	CPR								417
490h	SMT1CPRH	CPR								417
491h	SMT1CPRU	CPR								417
492h	SMT1CPWL	CPW								418
493h	SMT1CPWH	CPW								418
494h	SMT1CPWU	CPW								418
495h	SMT1PRL	SMT1PR								419
496h	SMT1PRH	SMT1PR								419
497h	SMT1PRU	SMT1PR								419
498h	SMT1CON0	EN	—	STP	WPOL	SPOL	CPOL	SMT1PS<1:0>		410
499h	SMT1CON1	SMT1GO	REPEAT	—	—	MODE<3:0>				411
49Ah	SMT1STAT	CPRUP	CPWUP	RST	—	—	TS	WS	AS	412
49Bh	SMT1CLK	—	—	—	—	—	CSEL<2:0>			413
49Ch	SMT1SIG	—	—	—	SSEL<4:0>					415
49Dh	SMT1WIN	—	—	—	WSEL<4:0>					414

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

Note 1: Unimplemented data memory locations, read as '0'.

TABLE 39-3: POWER-DOWN CURRENT (I_{PD})(1,2)

PIC16(L)F19155/56/75/76/85/86				Standard Operating Conditions (unless otherwise stated)					
PIC16F19155/56/75/76/85/86				Standard Operating Conditions (unless otherwise stated) VREGPM = 1					
Param. No.	Symbol	Device Characteristics	Min.	Typ.†	Max. +85°C	Max. +125°C	Units	Conditions	
								V_{DD}	Note
D200	I_{PD}	I_{PD} Base	—	0.05	—	—	µA	3.0V	
D200 D200A	I_{PD}	I_{PD} Base	—	0.4	—	—	µA	3.0V	
			—	18	—	—	µA	3.0V	VREGPM = 0
D201	I_{PD_WDT}	Low-Frequency Internal Oscillator/WDT	—	0.4	—	—	µA	3.0V	
D201	I_{PD_WDT}	Low-Frequency Internal Oscillator/WDT	—	0.6	—	—	µA	3.0V	
D202	I_{PD_SOSC}	Secondary Oscillator (SOSC)	—	0.6	—	—	µA	3.0V	
D202	I_{PD_SOSC}	Secondary Oscillator (SOSC)	—	0.8	—	—	µA	3.0V	
D203	I_{PD_FVR}	FVR	—	28	—	—	µA	3.0V	
D203	I_{PD_FVR}	FVR	—	33	—	—	µA	3.0V	
D204	I_{PD_BOR}	Brown-out Reset (BOR)	—	10	—	—	µA	3.0V	
D204	I_{PD_BOR}	Brown-out Reset (BOR)	—	14	—	—	µA	3.0V	
D205	I_{PD_LPBOR}	Low-Power Brown-out Reset (LPBOR)	—	0.5	—	—	µA	3.0V	
D205	I_{PD_LPBOR}	Low-Power Brown-out Reset (LPBOR)	—	0.7	—	—	µA	3.0V	
D206	I_{PD_ADCA}	ADC - Active	—	250	—	—	µA	3.0V	ADC is converting (4)
D206	I_{PD_ADCA}	ADC - Active	—	280	—	—	µA	3.0V	ADC is converting (4)
D207	$I_{PD\_ADCA\_CHGPUMP}$	ADC - Active with Charge Pump	—	315	—	—	µA	3.0V	
D207	$I_{PD\_ADCA\_CHGPUMP}$	ADC - Active with Charge Pump	—	345	—	—	µA	3.0V	
D208	I_{PD_CMP}	Comparator C1	—	30	—	—	µA	3.0V	
D208	I_{PD_CMP}	Comparator C1	—	33	—	—	µA	3.0V	
D209	$I_{PD\_CMP\_LP}$	LP Comparator C2 (6)	—	0.25	—	—	µA	3.0V	
D209	$I_{PD\_CMP\_LP}$	LP Comparator C2 (6)	—	0.6	—	—	µA	3.0V	
D210	$I_{PD\_LCD\_PUMP\_3V}$	3V Pump Output	—	2.5	—	—	µA	2.5V	
D210	$I_{PD\_LCD\_PUMP\_5V}$	5V Pump Output	—	3	—	—	µA	2.5V	

† 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: The peripheral current is the sum of the base I_{DD} and the additional current consumed when this peripheral is enabled. The peripheral Δ current can be determined by subtracting the base I_{DD} or I_{PD} current from this limit. Max. values should be used when calculating total current consumption.

2: The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode with all I/O pins in high-impedance state and tied to V_{SS}.

3: All peripheral currents listed are on a per-peripheral basis if more than one instance of a peripheral is available.

4: ADC clock source is FRC.

5: = LF device

6: The IPD spec for the LP comparator does not include current consumed by the supporting clock.