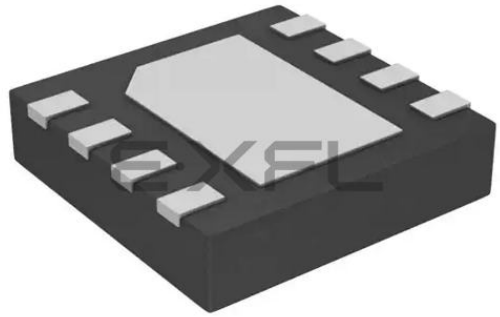


Welcome to E-XFL.COM

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, POR, PWM, WDT
Number of I/O	6
Program Memory Size	3.5KB (2K x 14)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 5x10b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	8-UDFN Exposed Pad
Supplier Device Package	8-UDFN (3x3)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f18313-e-rf

TABLE 4-4: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-31 (CONTINUED)

Address	Name	PIC16(L)F18313	PIC16(L)F18323	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 10-11													
CPU CORE REGISTERS; see Table 4-2 for specifics													
50Ch-51Fh	—	—		Unimplemented								—	—
58Ch-59Fh	—	—		Unimplemented								—	—
Bank 12													
60Ch	—	—		Unimplemented								—	—
60Dh	—	—		Unimplemented								—	—
60Eh	—	—		Unimplemented								—	—
60Fh	—	—		Unimplemented								—	—
610h	—	—		Unimplemented								—	—
611h	—	—		Unimplemented								—	—
612h	—	—		Unimplemented								—	—
613h	—	—		Unimplemented								—	—
614h	—	—		Unimplemented								—	—
615h	—	—		Unimplemented								—	—
616h	—	—		Unimplemented								—	—
617h	PWM5DCL			PWM5DC<1:0>		—	—	—	—	—	—	xx-- ----	uu-- ----
618h	PWM5DCH			PWM5DC<9:2>								xxxx xxxx	uuuu uuuu
619h	PWM5CON			PWM5EN	—	PWM5OUT	PWM5POL	—	—	—	—	0-00 ----	0-00 ----
61Ah	PWM6DCL			PWM6DC<1:0>		—	—	—	—	—	—	xx-- ----	uu-- ----
61Bh	PWM6DCH			PWM6DC<9:2>								xxxx xxxx	uuuu uuuu
61Ch	PWM6CON			PWM6EN	—	PWM6OUT	PWM6POL	—	—	—	—	0-00 ----	0-00 ----
61Dh-61Fh	—	—		Unimplemented								—	—

Legend: x = unknown, u = unchanged, q =depends on condition, - = unimplemented, read as '0', r = reserved. Shaded locations unimplemented, read as '0'.
Note 1: Only on PIC16F18313/18323.

TABLE 4-4: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-31 (CONTINUED)

Address	Name	PIC16(L)F18313	PIC16(L)F18323	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 18													
CPU CORE REGISTERS; see Table 4-2 for specifics													
90Ch	—	—		Unimplemented								—	—
90Dh	—	—		Unimplemented								—	—
90Eh	—	—		Unimplemented								—	—
90Fh	—	—		Unimplemented								—	—
910h	—	—		Unimplemented								—	—
911h	PMD0			SYSCMD	FVRMD	—	—	—	NVMMD	CLKRMD	IOCMD	00-- -000	00-- -000
912h	PMD1			NCOMD	—	—	—	—	TMR2MD	TMR1MD	TMR0MD	0--- -000	0--- -000
913h	PMD2	X	—	—	DACMD	ADCMD	—	—	—	CMP1MD	—	-00- --0-	-00- --0-
		—	X	—	DACMD	ADCMD	—	—	CMP2MD	CMP1MD	—	-00- -00-	-00- -00-
914h	PMD3			—	CWG1MD	PWM6MD	PWM5MD	—	—	CCP2MD	CCP1MD	-000 --00	-000 --00
915h	PMD4			—	—	UART1MD	—	—	—	MSSP1MD	—	--0- --0-	--0- --0-
916h	PMD5			—	—	—	—	—	CLC2MD	CLC1MD	DSMMD	---- -000	---- -000
917h	—	—		Unimplemented								—	—
918h	CPUDOZE			IDLEN	DOZEN	ROI	DOE	—	DOZE<2:0>			000- -000	000- -000
919h	OSCCON1			—	NOSC<2:0>			NDIV<3:0>				-qqq 0000	-qqq 0000
91Ah	OSCCON2			—	COSC<2:0>			CDIV<3:0>				-qqq 0000	-qqq 0000
91Bh	OSCCON3			CSWHOLD	SOSCPWR	SOSCBE	ORDY	NOSCR	—	—	—	0000 0---	0000 0---
91Ch	OSCSTAT1			EXTOR	HFOR	—	LFOR	SOR	ADOR	—	PLLR	qq-q qq-q	qq-q qq-q
91Dh	OSCEN			EXTOEN	HFOEN	—	LFOEN	SOSCEN	ADOEN	—	—	00-0 00--	00-0 00--
91Eh	OSCTUNE			—	—	HFTUN<5:0>						--10 0000	--10 0000
91Fh	OSCFRQ			—	—	—	—	HFFRQ<3:0>				---- 0110	---- 0110

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

Note 1: Only on PIC16F18313/18323.

7.0 OSCILLATOR MODULE

7.1 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 7-1 illustrates a block diagram of the oscillator module.

Clock sources can be supplied from external oscillators, quartz-crystal resonators and ceramic resonators. 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 (LP, XT, HS, ECH, ECM, ECL) and switch automatically to the internal oscillator.
- Oscillator Start-up Timer (OST) ensures stability of crystal oscillator sources.

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 modes, LFINTOSC, HFINTOSC (set at 1 MHz), or HFINTOSC (set at 32 MHz) can be set through the RSTOSC bits.

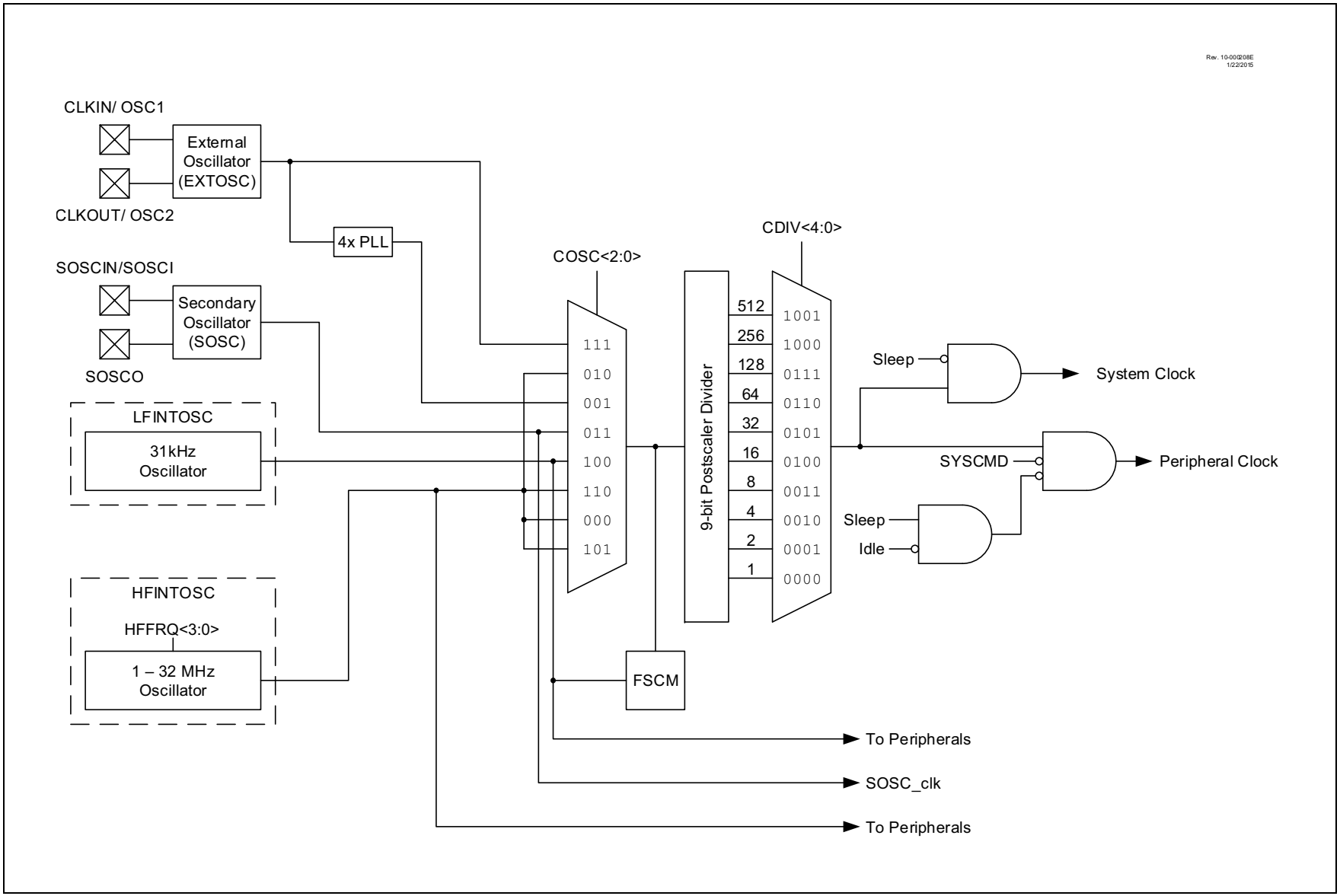
If an external clock source is selected, the FEXTOSC bits of Configuration Word 1 must be used in conjunction with the RSTOSC bits to select the External Clock mode.

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 (<= 100 kHz)
2. ECM – External Clock Medium-Power mode (<= 8 MHz)
3. ECH – External Clock High-Power mode (<= 32 MHz)
4. LP – 32 kHz Low-Power Crystal mode.
5. XT – Medium Gain Crystal or Ceramic Resonator Oscillator mode (between 100 MHz and 4 MHz)
6. HS – High Gain Crystal or Ceramic Resonator mode (above 4 MHz)

The ECH, ECM, and ECL clock modes rely on an external logic level signal as the device clock source. The LP, XT, and HS clock modes require an external crystal or resonator to be connected to the device. 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 7-1).

FIGURE 7-1: SIMPLIFIED PIC® MCU CLOCK SOURCE BLOCK DIAGRAM



7.2.2 INTERNAL CLOCK SOURCES

The device may be configured to use the internal oscillator block as the system clock by performing one of the following actions:

- Program the RSTOSC<2:0> bits in Configuration Words to select the INTOSC clock source, which will be used as the default system clock upon a device Reset.
- Write the NOSC<2:0> bits in the OSCCON1 register to switch the system clock source to the internal oscillator during run-time. See **Section 7.3 “Clock Switching”** for more information.

The function of the OSC2/CLKOUT pin is determined by the CLKOUTEN bit in Configuration Words.

The internal oscillator block has two independent oscillators that can produce two internal system clock sources.

1. The **HFINTOSC** (High-Frequency Internal Oscillator) is factory-calibrated and operates up to 32 MHz.
2. The **LFINTOSC** (Low-Frequency Internal Oscillator) is factory-calibrated and operates at 31 kHz.

7.2.2.1 HFINTOSC

The High-Frequency Internal Oscillator (HFINTOSC) is a precision digitally-controlled internal clock source that produces a stable clock up to 32 MHz. The HFINTOSC can be enabled through one of the following methods:

- Programming the RSTOSC<2:0> bits in Configuration Word 1 to ‘110’ (1 MHz) or ‘000’ (32 MHz) to set the oscillator upon device Power-up or Reset.
- Write to the NOSC<2:0> bits of the OSCCON1 register during run-time.

The HFINTOSC frequency can be selected by setting the HFFRQ<3:0> bits of the OSCFRQ register.

The NDIV<3:0> bits of the OSCCON1 register allow for the division of the output of the selected clock source by a range between 1:1 and 1:512.

7.2.2.2 2x PLL

The oscillator module contains a PLL that can be used with the HFINTOSC clock source to provide a system clock source. The input frequency to the PLL is limited to 8, 12, or 16 MHz, which will yield a system clock source of 16, 24, or 32 MHz, respectively. The PLL may be enabled for use by one of two methods:

1. Program the RSTOSC bits in the Configuration Word 1 to '000' to enable the HFINTOSC (32 MHz). This setting configures the HFFRQ<3:0> bits to '110' (16 MHz) and activates the 2x PLL.
2. Write '000' the NOSC<2:0> bits in the OSCCON1 register to enable the 2x PLL, and write the correct value into the HFFRQ<3:0> bits of the OSCFRQ register to select the desired system clock frequency. See Register 7-6 for more information.

REGISTER 7-5: OSCEN: OSCILLATOR MANUAL ENABLE REGISTER

R-q/q	R-q/q	U-0	R-q/q	R-q/q	R-q/q	U-0	U-0
EXTOEN	HFOEN	—	LFOEN	SOSCEN	ADOEN	—	—
bit 7							bit 0

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

bit 7 **EXTOEN:** External Oscillator Manual Request Enable bit
1 = EXTOSC is explicitly enabled, operating as specified by FEXTOSC
0 = EXTOSC could be enabled by another module

bit 6 **HFOEN:** HFINTOSC Oscillator Manual Request Enable bit
1 = HFINTOSC is explicitly enabled, operating as specified by OSCFRQ
0 = HFINTOSC could be enabled by another module

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

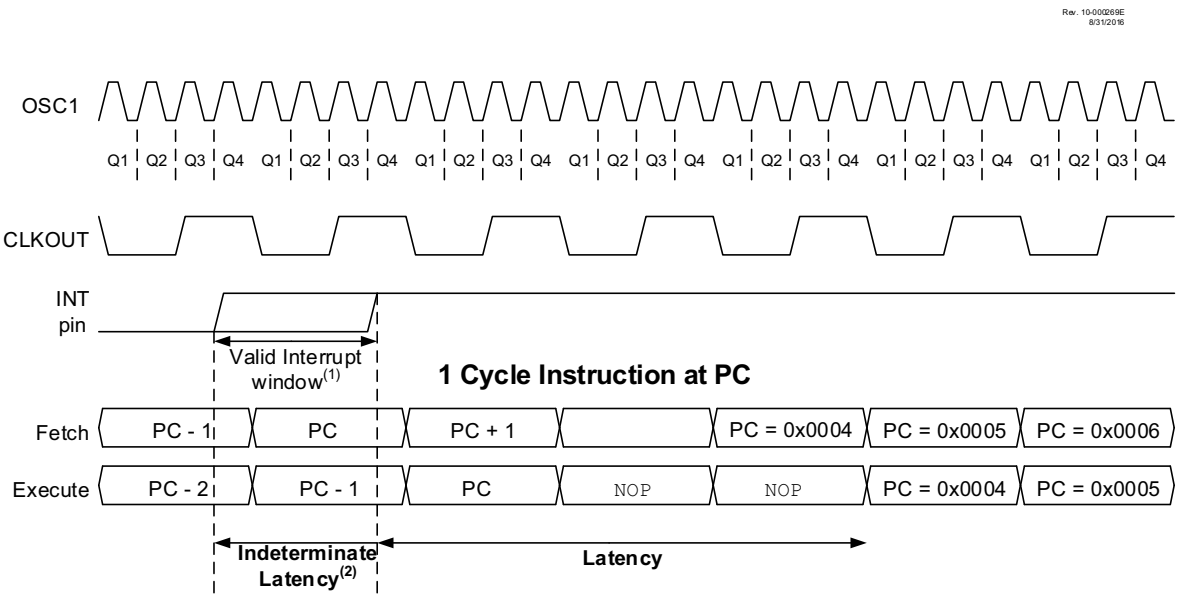
bit 4 **LFOEN:** LFINTOSC (31 kHz) Oscillator Manual Request Enable bit
1 = LFINTOSC is explicitly enabled
0 = LFINTOSC could be enabled by another module

bit 3 **SOSCEN:** Secondary Oscillator Manual Request Enable bit
1 = Secondary oscillator is explicitly enabled, operating as specified by SOSCBE and SOSCPWR
0 = Secondary oscillator could be enabled by another module

bit 2 **ADOEN:** ADCRC (600 kHz) Oscillator Manual Request Enable bit
1 = ADCRC is explicitly enabled
0 = ADCRC could be enabled by another module

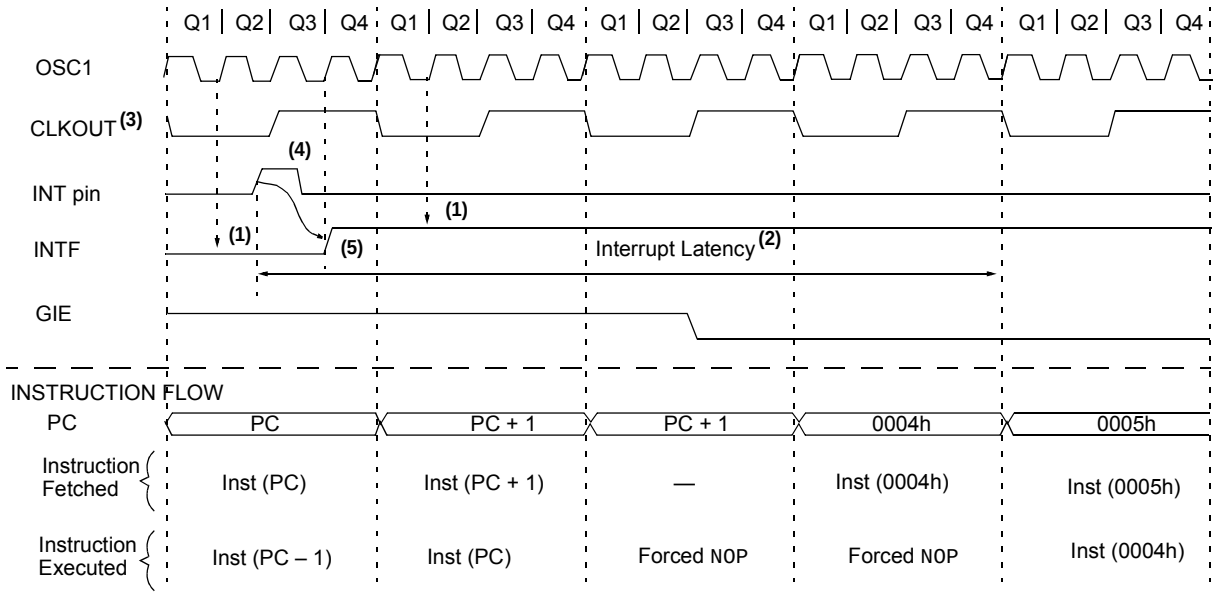
bit 1-0 **Unimplemented:** Read as '0'

FIGURE 8-2: INTERRUPT LATENCY



Note 1: An interrupt may occur at any time during the interrupt window.
2: Since an interrupt may occur any time during the interrupt window, the actual latency can vary.

FIGURE 8-3: 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 35.0 "Electrical Specifications"**.

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

10.6 Register Definitions: Watchdog Control

REGISTER 10-1: WDTCON: WATCHDOG TIMER CONTROL REGISTER

U-0	U-0	R/W-0/0	R/W-1/1	R/W-0/0	R/W-1/1	R/W-1/1	R/W-0/0
—	—	WDTPS<4:0>(1)					SWDTEN
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	-m/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7-6 **Unimplemented:** Read as '0'

bit 5-1 **WDTPS<4:0>:** Watchdog Timer Period Select bits(1)

Bit Value = Prescale Rate

11111 = Reserved. Results in minimum interval (1:32)
•
•
•
10011 = Reserved. Results in minimum interval (1:32)

10010 = 1:8388608 (2^{23}) (Interval 256s nominal)
10001 = 1:4194304 (2^{22}) (Interval 128s nominal)
10000 = 1:2097152 (2^{21}) (Interval 64s nominal)
01111 = 1:1048576 (2^{20}) (Interval 32s nominal)
01110 = 1:524288 (2^{19}) (Interval 16s nominal)
01101 = 1:262144 (2^{18}) (Interval 8s nominal)
01100 = 1:131072 (2^{17}) (Interval 4s nominal)
01011 = 1:65536 (Interval 2s nominal) (Reset value)
01010 = 1:32768 (Interval 1s nominal)
01001 = 1:16384 (Interval 512 ms nominal)
01000 = 1:8192 (Interval 256 ms nominal)
00111 = 1:4096 (Interval 128 ms nominal)
00110 = 1:2048 (Interval 64 ms nominal)
00101 = 1:1024 (Interval 32 ms nominal)
00100 = 1:512 (Interval 16 ms nominal)
00011 = 1:256 (Interval 8 ms nominal)
00010 = 1:128 (Interval 4 ms nominal)
00001 = 1:64 (Interval 2 ms nominal)
00000 = 1:32 (Interval 1 ms nominal)

bit 0 **SWDTEN:** Software Enable/Disable for Watchdog Timer bit

If WDTE<1:0> = 1x:
This bit is ignored.
If WDTE<1:0> = 01:
1 = WDT is turned on
0 = WDT is turned off
If WDTE<1:0> = 00:
This bit is ignored.

Note 1: Times are approximate. WDT time is based on 31 kHz LFINTOSC.

11.4.3 NVMREG WRITE TO EEPROM

Writing to the EEPROM is accomplished by the following steps:

1. Set the NVMREGS and WREN bits of the NVMCON1 register.
2. Write the desired address (address + 7000h) into the NVMADRH:NVMADRL register pair (Table 11-2).
3. Perform the unlock sequence as described in **Section 11.4.2 "NVM Unlock Sequence"**.

A single EEPROM byte is written with NVMDATA. The operation includes an implicit erase cycle for that byte (it is not necessary to set the FREE bit), and requires many instruction cycles to finish. CPU execution continues in parallel and, when complete, WR is cleared by hardware, NVMIF is set, and an interrupt will occur if NVMIE is also set. Software must poll the WR bit to determine when writing is complete, or wait for the interrupt to occur. WREN will remain unchanged.

Once the EEPROM write operation begins, clearing the WR bit will have no effect; the operation will run to completion.

11.4.4 NVMREG ERASE OF PROGRAM FLASH MEMORY

Program Flash Memory can only be erased one row at a time. No automatic erase occurs upon the initiation of the write to Program Flash Memory.

To erase a Program Flash Memory row:

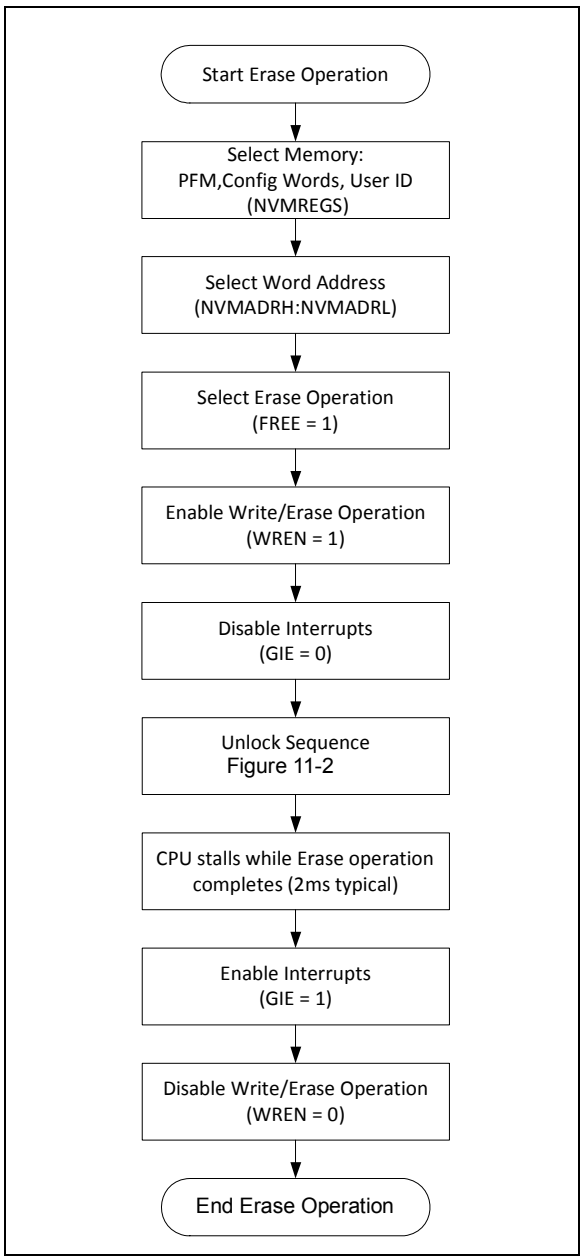
1. Clear the NVMREGS bit of the NVMCON1 register to erase Program Flash Memory locations, or set the NMVREGS bit to erase User ID locations.
2. Write the desired address into the NVMADRH:NVMADRL register pair (Table 11-2).
3. Set the FREE and WREN bits of the NVMCON1 register.
4. Perform the unlock sequence as described in **Section 11.4.2 "NVM Unlock Sequence"**.

If the Program Flash Memory address is write-protected, the WR bit will be cleared and the erase operation will not take place.

While erasing Program Flash Memory, CPU operation is suspended, and resumes when the operation is complete. Upon completion, the NVMIF is set, and an interrupt will occur if the NVMIE bit is also set.

Write latch data is not affected by erase operations, and WREN will remain unchanged.

FIGURE 11-3: NVM ERASE FLOWCHART



REGISTER 11-5: NVMCON1: NONVOLATILE MEMORY CONTROL 1 REGISTER

U-0	R/W-0/0	R/W-0/0	R/W/HC-0/0	R/W/HS-x/q	R/W-0/0	R/S/HC-0/0	R/S/HC-0/0
—	NVMREGS	LWLO	FREE	WRERR	WREN	WR	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, Configuration, User ID and Device ID Registers
0 = Access Program Flash Memory

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:** Program Flash Memory Erase Enable bit
When NVMREGS:NVMADR points to a Program Flash Memory 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, 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 EEPROM location:
1 = Initiates an erase/program cycle at the corresponding EEPROM location
0 = NVM program/erase operation is complete and inactive
When NVMREG:NVMADR points to a Program Flash Memory location:
1 = Initiates the operation indicated by Table 11-4
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 may change while WR = 1 (during the EEPROM write operation it may be '0' or '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 11.4.2 "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 11-1).

19.1.1 PWM PERIOD

Referring to Figure 19-1, the PWM output has a period and a pulse width. The frequency of the PWM is the inverse of the period (1/period).

The PWM period is specified by writing to the PR2 register. The PWM period can be calculated using the following formula:

EQUATION 19-1: PWM PERIOD

$$PWM\ Period = [(PR2) + 1] \cdot 4 \cdot TOSC \cdot (TMR2\ Prescale\ Value)$$

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 PWMx pin is set (Exception: If the PWM duty cycle = 0%, the pin will not be set.)
- The PWM pulse width is latched from PWMxDC.

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

19.1.2 PWM DUTY CYCLE

The PWM duty cycle is specified by writing a 10-bit value to the PWMxDC register. The PWMxDCH contains the eight MSbs and the PWMxDCL<7:6> bits contain the two LSbs.

The PWMDC register is double-buffered and can be updated at any time. This double buffering is essential for glitch-free PWM operation. New values take effect when TMR2 = PR2. Note that PWMDC is left-justified.

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.

Equation 19-2 is used to calculate the PWM pulse width.

Equation 19-3 is used to calculate the PWM duty cycle ratio.

EQUATION 19-2: PULSE WIDTH

$$Pulse\ Width = (PWMxDC) \cdot TOSC \cdot (TMR2\ Prescale\ Value)$$

EQUATION 19-3: DUTY CYCLE RATIO

$$Duty\ Cycle\ Ratio = \frac{(PWMxDC)}{4(PR2+1)}$$

19.1.3 PWM RESOLUTION

PWM resolution, expressed in number of bits, defines the maximum number of discrete steps that can be present in a single PWM period. For example, a 10-bit resolution will result in 1024 discrete steps, whereas an 8-bit resolution will result in 256 discrete steps.

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 19-4.

EQUATION 19-4: 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.

19.1.4 OPERATION IN SLEEP MODE

In Sleep mode, the TMR2 register will not increment and the state of the module will not change. If the PWMx pin is driving a value, it will continue to drive that value. When the device wakes up, TMR2 will continue from its previous state.

19.1.5 CHANGES IN SYSTEM CLOCK FREQUENCY

The PWM frequency is derived from the system clock frequency. Any changes in the system clock frequency will result in changes to the PWM frequency. See **Section 7.0, Oscillator Module** for additional details.

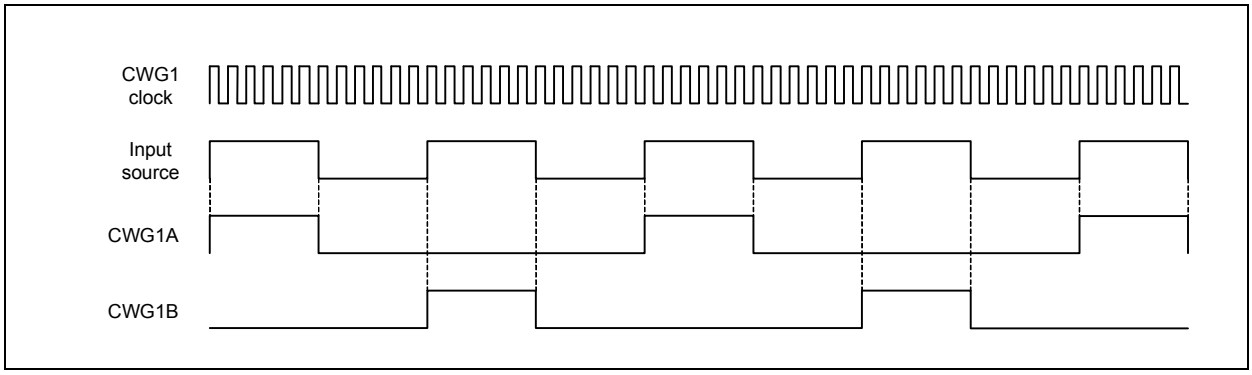
20.2.2 PUSH-PULL MODE

In Push-Pull mode, two output signals are generated, alternating copies of the input as illustrated in Figure 20-2. This alternation creates the push-pull effect required for driving some transformer-based power supply designs. Dead-band control is not used in Push-Pull mode. Steering modes are not used in Push-Pull mode.

The push-pull sequencer is reset whenever EN = 0 or if an auto-shutdown event occurs. The sequencer is clocked by the first input pulse, and the first output appears on CWG1A.

The unused outputs CWG1C and CWG1D drive copies of CWG1A and CWG1B, respectively, but with polarity controlled by POLC and POLD.

FIGURE 20-2: CWG1 PUSH-PULL MODE OPERATION



20.2.3 STEERING MODES

In both Synchronous and Asynchronous Steering modes, the modulated input signal can be steered to any combination of four CWG outputs and a fixed-value will be presented on all the outputs not used for the PWM output. Each output has independent polarity, steering, and shutdown options. Dead-band control is not used in either Steering mode.

When STRx = 0 (Register 20-5), then the corresponding pin is held at the level defined by DATx (Register 20-5). When STRx = 1, then the pin is driven by the modulated input signal.

The POLx bits (Register 20-2) control the signal polarity only when WGSTRx = 1.

The CWG auto-shutdown operation also applies to Steering modes as described in **Section 20.11 "Register Definitions: CWG Control"**.

Note:	Only the STRx bits are synchronized; the DATx (data) bits are not synchronized.

REGISTER 22-7: 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

TABLE 22-3: SUMMARY OF REGISTERS ASSOCIATED WITH ADC

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	90
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	BCL1IE	TMR2IE	TMR1IE	92
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	BCL1IF	TMR2IF	TMR1IF	97
TRISA	—	—	TRISA5	TRISA4	—(2)	TRISA2	TRISA1	TRISA0	132
TRISC(1)	—	—	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	138
ADCON0	CHS<5:0>						GO/$\overline{\text{DONE}}$	ADON	220
ADCON1	ADFM	ADCS<2:0>			—	ADNREF	ADPREF<1:0>		221
ADACT	—	—	—	—	ADACT<3:0>				222
ADRESH	ADRESH<7:0>								223, 223
ADRESL	ADRESL<7:0>								223, 224
ANSELA	—	—	ANSA5	ANSA4	—	ANSA2	ANSA1	ANSA0	133
ANSELC(1)	—	—	ANSC5	ANSC4	ANSC3	ANSC2	ANSC1	ANSC0	139
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR<1:0>		ADFVR<1:0>		157
DAC1CON1	—	—	—	DAC1R<4:0>					239
OSCSTAT1	EXTOR	HFOR	—	LFOR	SOR	ADOR	—	PLLR	82

Legend: – = unimplemented read as ‘0’. Shaded cells are not used for the ADC module.

Note 1: PIC16(L)F18323 only.

2: Unimplemented, read as ‘1’.

REGISTER 26-4: T0CON1: TIMER0 CONTROL REGISTER 1

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	R/W-0/0
T0CS<2:0>			T0ASYNC	T0CKPS<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	

bit 7-5 **T0CS<2:0>:** Timer0 Clock Source select bits
000 = T0CKIPPS (True)
001 = T0CKIPPS (Inverted)
010 = FOSC/4
011 = HFINTOSC
100 = LFINTOSC
101 = Reserved
110 = SOSC
111 = CLC1

bit 4 **T0ASYNC:** TMR0 Input Asynchronization Enable bit
1 = The input to the TMR0 counter is not synchronized to system clocks
0 = The input to the TMR0 counter is synchronized to FOSC/4

bit 3-0 **T0CKPS<3:0>:** Prescaler Rate Select bit
0000 = 1:1
0001 = 1:2
0010 = 1:4
0011 = 1:8
0100 = 1:16
0101 = 1:32
0110 = 1:64
0111 = 1:128
1000 = 1:256
1001 = 1:512
1010 = 1:1024
1011 = 1:2048
1100 = 1:4096
1101 = 1:8192
1110 = 1:16384
1111 = 1:32768

REGISTER 30-7: SSP1BUF: MSSP BUFFER 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
SSP1BUF<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 **SSP1BUF<7:0>:** MSSP Buffer bits

TABLE 30-3: SUMMARY OF REGISTERS ASSOCIATED WITH MSSP1

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
TRISA	—	—	TRISA5	TRISA4	—(3)	TRISA2	TRISA1	TRISA0	132
ANSELA	—	—	ANSA5	ANSA4	—	ANSA2	ANSA1	ANSA0	133
INLVLA(1)	—	—	INLVLA5	INLVLA4	INLVLA3	INLVLA2	INLVLA1	INLVLA0	135
TRISC(2)	—	—	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	138
ANSELC(2)	—	—	ANSC5(3)	ANSC4(3)	ANSC3	ANSC2	ANSC1	ANSC0	139
INLVLC(1, 2)	—	—	INLVLC5	INLVLC4	INLVLC3	INLVLC2	INLVLC1	INLVLC0	140
INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	90
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	BCL1IF	TMR2IF	TMR1IF	97
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	BCL1IE	TMR2IE	TMR1IE	92
SSP1STAT	SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF	331
SSP1CON1	WCOL	SSPOV	SSPEN	CKP	SSPM<3:0>				332
SSP1CON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	333
SSP1CON3	ACKTIM	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN	331
SSP1MSK	SSP1MSK<7:0>								335
SSP1ADD	SSP1ADD<7:0>								335
SSP1BUF	SSP1BUF<7:0>								336
SSP1CLKPPS	—	—	—	SSP1CLKPPS<4:0>					143
SSP1DATPPS	—	—	—	SSP1DATPPS<4:0>					143
SSP1SSPPS	—	—	—	SSP1SSPPS<4:0>					143

Legend: — = Unimplemented location, read as '0'. Shaded cells are not used by the MSSP module

Note 1: When using designated I^2C pins, the associated pin values in INLVLx will be ignored.

2: PIC16(L)F18323 only.

3: Unimplemented, read as '1'.

31.3.2 AUTO-BAUD OVERFLOW

During the course of automatic baud detection, the ABDOVF bit of the BAUD1CON register will be set if the baud rate counter overflows before the fifth rising edge is detected on the RX pin. The ABDOVF bit indicates that the counter has exceeded the maximum count that can fit in the 16 bits of the SP1BRGH:SP1BRGL register pair. The overflow condition will set the RCIF flag. The counter continues to count until the fifth rising edge is detected on the RX pin. The RCIDL bit will remain false ('0') until the fifth rising edge at which time the RCIDL bit will be set. If the RC1REG is read after the overflow occurs but before the fifth rising edge then the fifth rising edge will set the RCIF again.

Terminating the auto-baud process early to clear an overflow condition will prevent proper detection of the sync character fifth rising edge. If any falling edges of the sync character have not yet occurred when the ABDEN bit is cleared then those will be falsely detected as start bits. The following steps are recommended to clear the overflow condition:

1. Read RC1REG to clear RCIF
2. If RCIDL is zero then wait for RCIF and repeat step 1.
3. Clear the ABDOVF bit.

31.3.3 AUTO-WAKE-UP ON BREAK

During Sleep mode, all clocks to the EUSART1 are suspended. Because of this, the Baud Rate Generator is inactive and a proper character reception cannot be performed. The Auto-Wake-up feature allows the controller to wake-up due to activity on the RX/DT line. This feature is available only in Asynchronous mode.

The Auto-Wake-up feature is enabled by setting the WUE bit of the BAUD1CON register. Once set, the normal receive sequence on RX/DT is disabled, and the EUSART1 remains in an Idle state, monitoring for a wake-up event independent of the CPU mode. A wake-up event consists of a high-to-low transition on the RX/DT line. (This coincides with the start of a Sync Break or a wake-up signal character for the LIN protocol.)

The EUSART1 module generates an RCIF interrupt coincident with the wake-up event. The interrupt is generated synchronously to the Q clocks in normal CPU operating modes (Figure 31-7), and asynchronously if the device is in Sleep mode (Figure 31-8). The interrupt condition is cleared by reading the RC1REG register.

The WUE bit is automatically cleared by the low-to-high transition on the RX line at the end of the Break. This signals to the user that the Break event is over. At this point, the EUSART1 module is in Idle mode waiting to receive the next character.

31.3.3.1 Special Considerations

Break Character

To avoid character errors or character fragments during a wake-up event, the wake-up character must be all zeros.

When the wake-up is enabled the function works independent of the low time on the data stream. If the WUE bit is set and a valid non-zero character is received, the low time from the Start bit to the first rising edge will be interpreted as the wake-up event. The remaining bits in the character will be received as a fragmented character and subsequent characters can result in framing or overrun errors.

Therefore, the initial character in the transmission must be all '0's. This must be ten or more bit times, 13-bit times recommended for LIN bus, or any number of bit times for standard RS-232 devices.

Oscillator Start-up Time

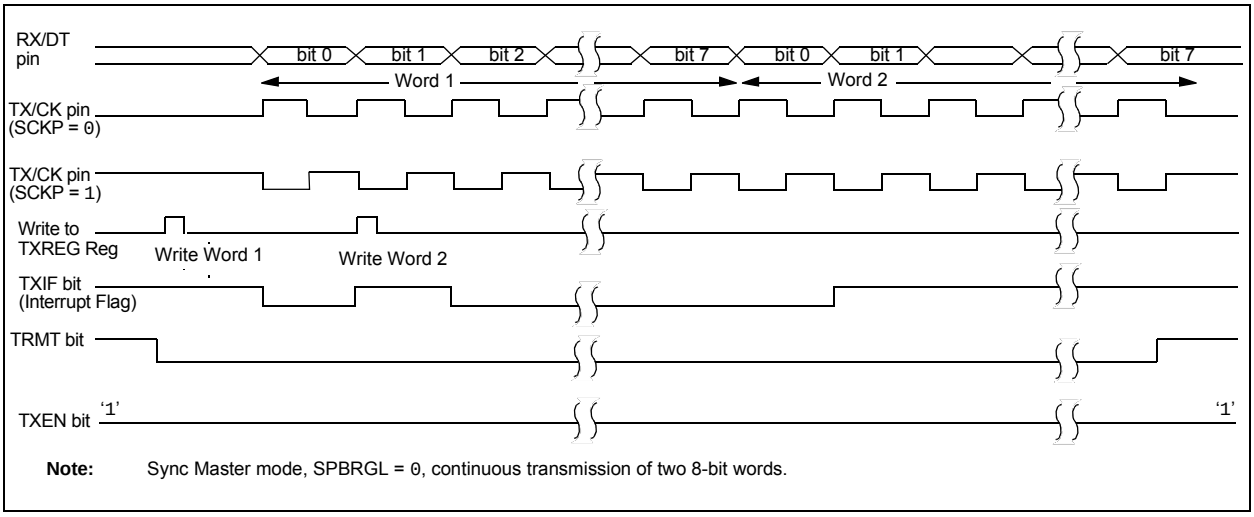
Oscillator start-up time must be considered, especially in applications using oscillators with longer start-up intervals (i.e., LP, XT or HS/PLL mode). The Sync Break (or wake-up signal) character must be of sufficient length, and be followed by a sufficient interval, to allow enough time for the selected oscillator to start and provide proper initialization of the EUSART1.

WUE Bit

The wake-up event causes a receive interrupt by setting the RCIF bit. The WUE bit is cleared in hardware by a rising edge on RX/DT. The interrupt condition is then cleared in software by reading the RC1REG register and discarding its contents.

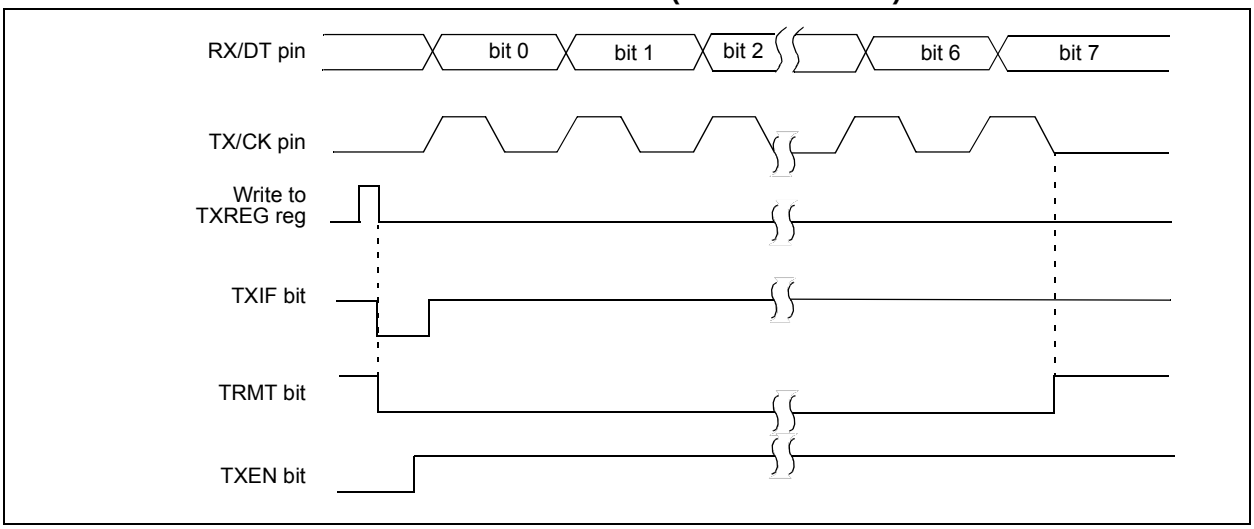
To ensure that no actual data is lost, check the RCIDL bit to verify that a receive operation is not in process before setting the WUE bit. If a receive operation is not occurring, the WUE bit may then be set just prior to entering the Sleep mode.

FIGURE 31-10: SYNCHRONOUS TRANSMISSION



Note: Sync Master mode, SPBRGL = 0, continuous transmission of two 8-bit words.

FIGURE 31-11: SYNCHRONOUS TRANSMISSION (THROUGH TXEN)



31.4.1.5 Synchronous Master Reception

Data is received at the RX/DT pin. The RX/DT pin output driver is automatically disabled when the EUSART1 is configured for synchronous master receive operation.

In Synchronous mode, reception is enabled by setting either the Single Receive Enable bit (SREN of the RC1STA register) or the Continuous Receive Enable bit (CREN of the RC1STA register).

When SREN is set and CREN is clear, only as many clock cycles are generated as there are data bits in a single character. The SREN bit is automatically cleared at the completion of one character. When CREN is set, clocks are continuously generated until CREN is cleared. If CREN is cleared in the middle of a character the CK clock stops immediately and the partial character is discarded. If SREN and CREN are both set, then SREN is cleared at the completion of the first character and CREN takes precedence.

To initiate reception, set either SREN or CREN. Data is sampled at the RX/DT pin on the trailing edge of the TX/CK clock pin and is shifted into the Receive Shift Register (RSR). When a complete character is received into the RSR, the RCIF bit is set and the character is automatically transferred to the two character receive FIFO. The Least Significant eight bits of the top character in the receive FIFO are available in RC1REG. The RCIF bit remains set as long as there are unread characters in the receive FIFO.

Note:	If the RX/DT function is on an analog pin, the corresponding ANSEL bit must be cleared for the receiver to function.

SLEEP **Enter Sleep mode**

Syntax: [*label*] SLEEP

Operands: None

Operation: 00h → WDT,
0 → WDT prescaler,
1 → $\overline{TO}$,
0 → $\overline{PD}$

Status Affected: $\overline{TO}$, $\overline{PD}$

Description: The power-down Status bit, $\overline{PD}$ is cleared. Time-out Status bit, $\overline{TO}$ is set. Watchdog Timer and its prescaler are cleared.
See **Section 9.3 "Sleep Mode"** for more information.

SUBLW **Subtract W from literal**

Syntax: [*label*] SUBLW k

Operands: 0 ≤ k ≤ 255

Operation: k - (W) → (W)

Status Affected: C, DC, Z

Description: The W register is subtracted (2's complement method) from the 8-bit literal 'k'. The result is placed in the W register.

C = 0	W > k
C = 1	W ≤ k
DC = 0	W<3:0> > k<3:0>
DC = 1	W<3:0> ≤ k<3:0>

SUBWF **Subtract W from f**

Syntax: [*label*] SUBWF f,d

Operands: 0 ≤ f ≤ 127
d ∈ [0,1]

Operation: (f) - (W) → (destination)

Status Affected: C, DC, Z

Description: Subtract (2's complement method) W register from 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.

C = 0	W > f
C = 1	W ≤ f
DC = 0	W<3:0> > f<3:0>
DC = 1	W<3:0> ≤ f<3:0>

SUBWFB **Subtract W from f with Borrow**

Syntax: SUBWFB f {,d}

Operands: 0 ≤ f ≤ 127
d ∈ [0,1]

Operation: (f) – (W) – ($\overline{B}$) → dest

Status Affected: C, DC, Z

Description: Subtract W and the BORROW flag (CARRY) from register 'f' (2's complement method). If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f'.

SWAPF **Swap Nibbles in f**

Syntax: [*label*] SWAPF f,d

Operands: 0 ≤ f ≤ 127
d ∈ [0,1]

Operation: (f<3:0>) → (destination<7:4>),
(f<7:4>) → (destination<3:0>)

Status Affected: None

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

TABLE 35-4: I/O PORTS

Standard Operating Conditions (unless otherwise stated)							
Param. No.	**Sym.**	**Characteristic**	**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
	VIL	**Input Low Voltage**					
		I/O PORT:					
D300		with TTL buffer	—	—	0.8	V	4.5V ≤ VDD ≤ 5.5V
D301			—	—	0.15 VDD	V	1.8V ≤ VDD ≤ 4.5V
D302		with Schmitt Trigger buffer	—	—	0.2 VDD	V	2.0V ≤ VDD ≤ 5.5V
D303		with I^2C levels	—	—	0.3 VDD	V	
D304		with SMBus levels	—	—	0.8	V	2.7V ≤ VDD ≤ 5.5V
D305		$\overline{MCLR}$	—	—	0.2 VDD	V	
	VIH	**Input High Voltage**					
		I/O PORT:					
D320		with TTL buffer	2.0	—	—	V	4.5V ≤ VDD ≤ 5.5V
D321			0.25 VDD + 0.8	—	—	V	1.8V ≤ VDD ≤ 4.5V
D322		with Schmitt Trigger buffer	0.8 VDD	—	—	V	2.0V ≤ VDD ≤ 5.5V
D323		with I^2C levels	0.7 VDD	—	—	V	
D324		with SMBus levels	2.1	—	—	V	2.7V ≤ VDD ≤ 5.5V
D325		$\overline{MCLR}$	0.7 VDD	—	—	V	
	IIL	**Input Leakage Current(2)**					
D340		I/O Ports	—	± 5	± 125	nA	VSS ≤ VPIN ≤ VDD, Pin at high-impedance, 85°C
D341			—	± 5	± 1000	nA	VSS ≤ VPIN ≤ VDD, Pin at high-impedance, 125°C
D342		$\overline{MCLR}$(2)	—	± 50	± 200	nA	VSS ≤ VPIN ≤ VDD, Pin at high-impedance, 85°C
	IPUR	**Weak Pull-up Current**					
D350			25	120	200	µA	VDD = 3.0V, VPIN = VSS
	VOL	**Output Low Voltage(4)**					
D360		I/O ports	—	—	0.6	V	IOL = 10.0mA, VDD = 3.0V
	VOH	**Output High Voltage(4)**					
D370		I/O ports	VDD - 0.7	—	—	V	IOH = 6.0 mA, VDD = 3.0V
D380	CIO	**All I/O pins**	—	5	50	pF	

† 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.

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

36.0 DC AND AC CHARACTERISTICS GRAPHS AND CHARTS

Charts and Graphs are not available at this time.