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	40MHz
Connectivity	UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	16
Program Memory Size	4KB (2K x 16)
Program Memory Type	FLASH
EEPROM Size	128 x 8
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 5.5V
Data Converters	A/D 4x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	20-SSOP (0.209", 5.30mm Width)
Supplier Device Package	20-SSOP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf1230-i-ss

18/20/28-Pin Enhanced Flash Microcontrollers with nanoWatt Technology, High-Performance PWM and A/D

Power-Managed Modes:

- Run: CPU on, peripherals on
- Idle: CPU off, peripherals on
- Sleep: CPU off, peripherals off
- Ultra Low 50 nA Input Leakage
- Run mode currents down to 15 μA, typical
- Idle mode currents down to 3.7 μA, typical
- Sleep mode current down to 100 nA, typical
- Timer1 Oscillator: 1.8 μA, typical; 32 kHz; 2V
- Watchdog Timer (WDT): 1.4 μA, typical; 2V
- Two-Speed Oscillator Start-up

14-Bit Power Control PWM Module:

- Up to 6 PWM Channel Outputs
  - Complementary or independent outputs
- Edge or Center-Aligned Operation
- Flexible Dead-Band Generator
- Hardware Fault Protection Input
- Simultaneous Update of Duty Cycle and Period:
  - Flexible Special Event Trigger output

Flexible Oscillator Structure:

- Four Crystal modes, up to 40 MHz
- 4x Phase Lock Loop (PLL) – Available for Crystal and Internal Oscillators
- Two External RC modes, up to 4 MHz
  - Fast wake-up from Sleep and Idle, 1 μs, typical
- Two External Clock modes, up to 40 MHz
- Internal Oscillator Block:
  - 8 user-selectable frequencies from 31 kHz to 8 MHz
  - Provides a complete range of clock speeds from 31 kHz to 32 MHz when used with PLL
  - User-tunable to compensate for frequency drift
- Secondary Oscillator using Timer1 @ 32 kHz
- Fail-Safe Clock Monitor:
  - Allows for safe shutdown if peripheral clock stops

Peripheral Highlights:

- High-Current Sink/Source 25 mA/25 mA
- Up to 4 Programmable External Interrupts
- Four Input Change Interrupts
- Enhanced Addressable USART module:
  - Supports RS-485, RS-232 and LIN/J2602
  - RS-232 operation using internal oscillator block (no external crystal required)
  - Auto-wake-up on Start bit
  - Auto-Baud Detect
- 10-Bit, up to 4-Channel Analog-to-Digital Converter module (A/D):
  - Auto-acquisition capability
  - Conversion available during Sleep
- Up to 3 Analog Comparators
- Programmable Reference Voltage for Comparators
- Programmable, 15-Level Low-Voltage Detection (LVD) module:
  - Supports interrupt on Low-Voltage Detection

Special Microcontroller Features:

- C Compiler Optimized Architecture with Optional Extended Instruction Set
- Flash Memory Retention: > 40 years
- Self-Programmable under Software Control
- Priority Levels for Interrupts
- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
  - Programmable period from 4 ms to 131s
- Programmable Code Protection
- Single-Supply In-Circuit Serial Programming™ (ICSP™) via Two Pins
- In-Circuit Debug (ICD) via Two Pins
- Wide Operating Voltage Range (2.0V to 5.5V)

Device	Program Memory		Data Memory		I/O	10-Bit ADC Channel	EUSART	Analog Comparator	14-Bit PWM (ch)	Timers 16-Bit
	Flash (bytes)	# Single-Word Instructions	SRAM (bytes)	EEPROM (bytes)						
PIC18F1230	4096	2048	256	128	16	4	Yes	3	6	2
PIC18F1330	8192	4096	256	128	16	4	Yes	3	6	2

2.0 GUIDELINES FOR GETTING STARTED WITH PIC18F MICROCONTROLLERS

2.1 Basic Connection Requirements

Getting started with the PIC18F1230/1330 family of 8-bit microcontrollers requires attention to a minimal set of device pin connections before proceeding with development.

The following pins must always be connected:

- All VDD and VSS pins
  (see **Section 2.2 "Power Supply Pins"**)
- All AVDD and AVSS pins, regardless of whether or not the analog device features are used
  (see **Section 2.2 "Power Supply Pins"**)
- $\overline{\text{MCLR}}$ pin
  (see **Section 2.3 "Master Clear (MCLR) Pin"**)

These pins must also be connected if they are being used in the end application:

- PGC/PGD pins used for In-Circuit Serial Programming™ (ICSP™) and debugging purposes
  (see **Section 2.4 "ICSP Pins"**)
- OSCI and OSCO pins when an external oscillator source is used
  (see **Section 2.5 "External Oscillator Pins"**)

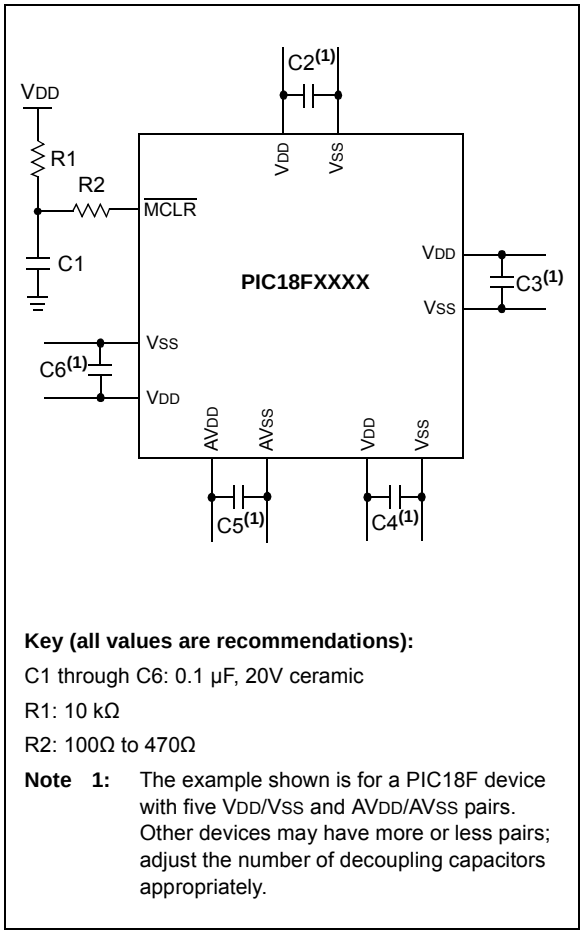
Additionally, the following pins may be required:

- VREF+/VREF- pins are used when external voltage reference for analog modules is implemented

Note:	The AVDD and AVSS pins must always be connected, regardless of whether any of the analog modules are being used.

The minimum mandatory connections are shown in Figure 2-1.

FIGURE 2-1: RECOMMENDED MINIMUM CONNECTIONS



Key (all values are recommendations):

C1 through C6: 0.1 µF, 20V ceramic

R1: 10 kΩ

R2: 100Ω to 470Ω

Note 1: The example shown is for a PIC18F device with five VDD/VSS and AVDD/AVSS pairs. Other devices may have more or less pairs; adjust the number of decoupling capacitors appropriately.

FIGURE 5-3: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ TIED TO V_{DD}, V_{DD} RISE < T_{PWRT})

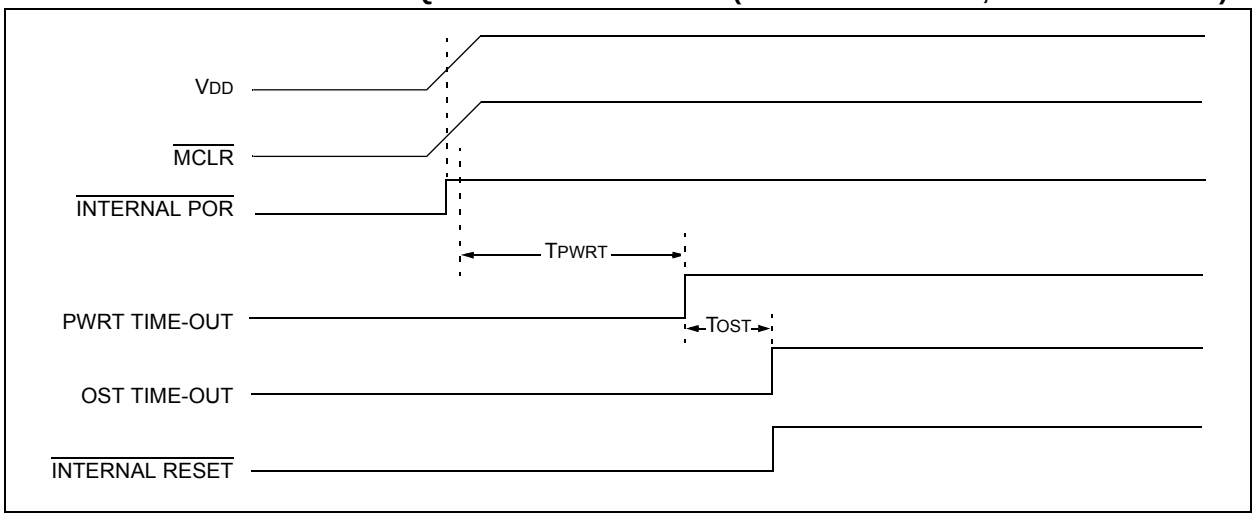


FIGURE 5-4: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ NOT TIED TO V_{DD}): CASE 1

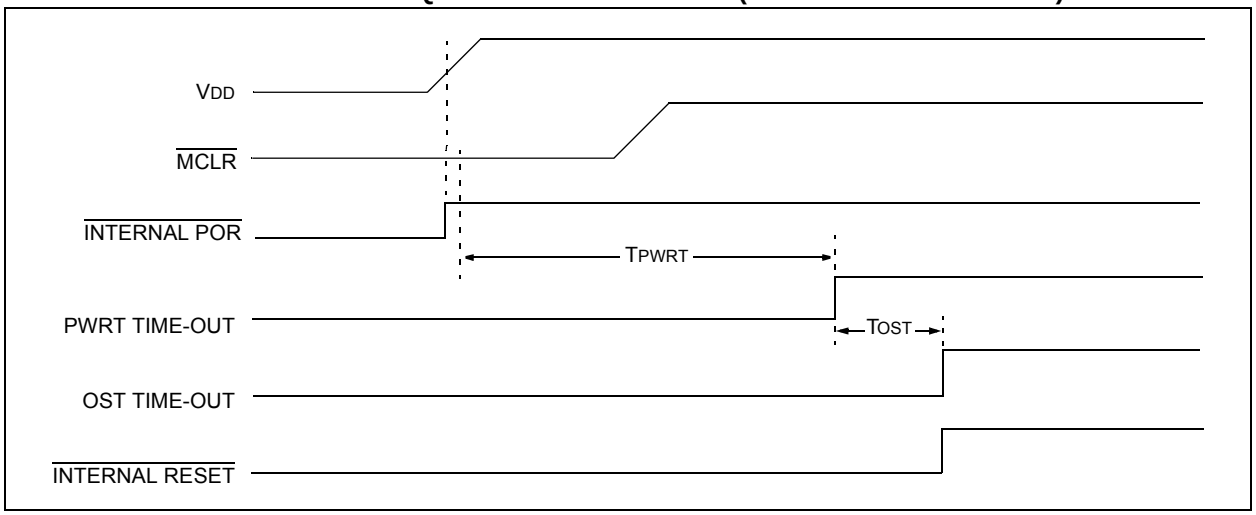


FIGURE 5-5: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ NOT TIED TO V_{DD}): CASE 2

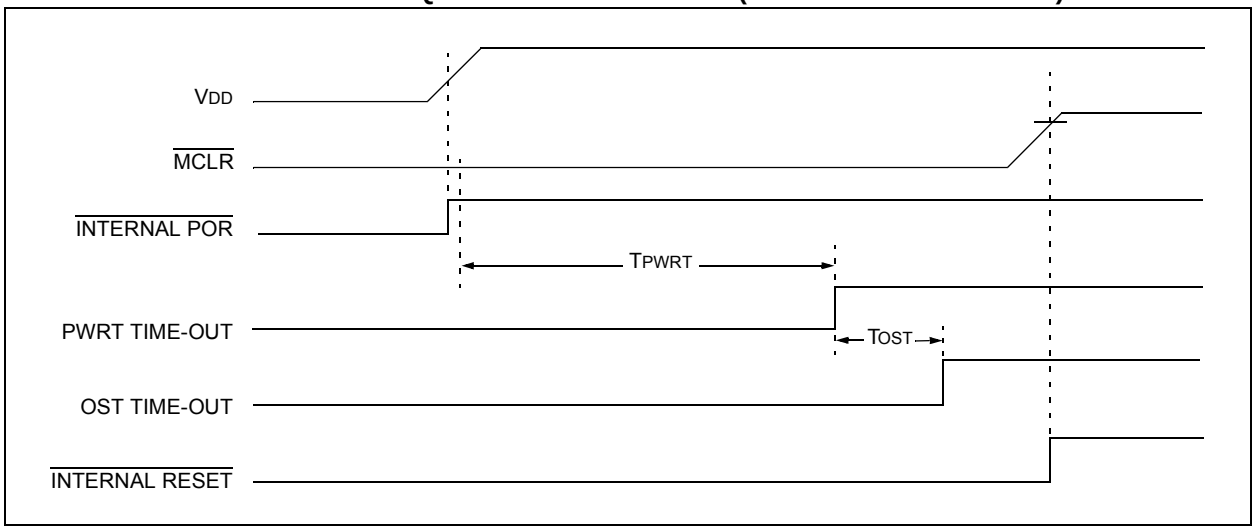
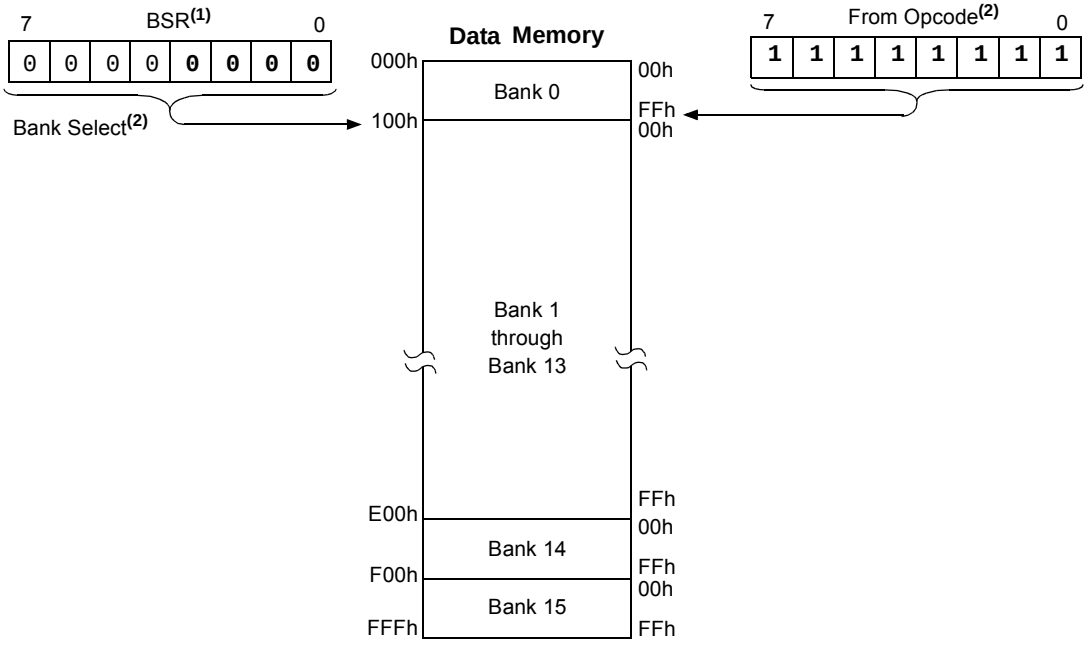


FIGURE 6-6: USE OF THE BANK SELECT REGISTER (DIRECT ADDRESSING)



Note 1: The Access RAM bit of the instruction can be used to force an override of the selected bank (BSR<3:0>) to the registers of the Access Bank.

2: The MOVFF instruction embeds the entire 12-bit address in the instruction.

6.3.2 ACCESS BANK

While the use of the BSR with an embedded 8-bit address allows users to address the entire range of data memory, it also means that the user must always ensure that the correct bank is selected. Otherwise, data may be read from or written to the wrong location. This can be disastrous if a GPR is the intended target of an operation, but an SFR is written to instead. Verifying and/or changing the BSR for each read or write to data memory can become very inefficient.

To streamline access for the most commonly used data memory locations, the data memory is configured with an Access Bank, which allows users to access a mapped block of memory without specifying a BSR. The Access Bank consists of the first 128 bytes of memory (00h-7Fh) in Bank 0 and the last 128 bytes of memory (80h-FFh) in Block 15. The lower half is known as the “Access RAM” and is composed of GPRs. The upper half is where the device’s SFRs are mapped. These two areas are mapped contiguously in the Access Bank and can be addressed in a linear fashion by an 8-bit address (Figure 6-5).

The Access Bank is used by core PIC18 instructions that include the Access RAM bit (the ‘a’ parameter in the instruction). When ‘a’ is equal to ‘1’, the instruction uses the BSR and the 8-bit address included in the opcode for the data memory address. When ‘a’ is ‘0’, however, the instruction is forced to use the Access Bank address map; the current value of the BSR is ignored entirely.

Using this “forced” addressing allows the instruction to operate on a data address in a single cycle without updating the BSR first. For 8-bit addresses of 80h and above, this means that users can evaluate and operate on SFRs more efficiently. The Access RAM below 80h is a good place for data values that the user might need to access rapidly, such as immediate computational results or common program variables. Access RAM also allows for faster and more code efficient context saving and switching of variables.

The mapping of the Access Bank is slightly different when the extended instruction set is enabled (XINST Configuration bit = 1). This is discussed in more detail in **Section 6.5.3 “Mapping the Access Bank in Indexed Literal Offset Addressing Mode”**.

6.3.3 GENERAL PURPOSE REGISTER FILE

PIC18 devices may have banked memory in the GPR area. This is data RAM which is available for use by all instructions. GPRs start at the bottom of Bank 0 (address 000h) and grow upwards towards the bottom of the SFR area. GPRs are not initialized by a Power-on Reset and are unchanged on all other Resets.

NOTES:

EXAMPLE 7-3: WRITING TO FLASH PROGRAM MEMORY

```
                MOVLW   D'88                    ; number of bytes in erase block
                MOVWF   COUNTER
                MOVLW   BUFFER_ADDR_HIGH        ; point to buffer
                MOVWF   FSR0H
                MOVLW   BUFFER_ADDR_LOW
                MOVWF   FSR0L
                MOVLW   CODE_ADDR_UPPER         ; Load TBLPTR with the base
                MOVWF   TBLPTRU                 ; address of the memory block
                MOVLW   CODE_ADDR_HIGH
                MOVWF   TBLPTRH
                MOVLW   CODE_ADDR_LOW
                MOVWF   TBLPTRL
READ_BLOCK
                TBLRD*+                         ; read into TABLAT, and inc
                MOVF    TABLAT, W               ; get data
                MOVWF   POSTINC0                ; store data
                DECFSZ  COUNTER                 ; done?
                BRA     READ_BLOCK              ; repeat
MODIFY_WORD
                MOVLW   DATA_ADDR_HIGH          ; point to buffer
                MOVWF   FSR0H
                MOVLW   DATA_ADDR_LOW
                MOVWF   FSR0L
                MOVLW   NEW_DATA_LOW            ; update buffer word
                MOVWF   POSTINC0
                MOVLW   NEW_DATA_HIGH
                MOVWF   INDF0
ERASE_BLOCK
                MOVLW   CODE_ADDR_UPPER         ; load TBLPTR with the base
                MOVWF   TBLPTRU                 ; address of the memory block
                MOVLW   CODE_ADDR_HIGH
                MOVWF   TBLPTRH
                MOVLW   CODE_ADDR_LOW
                MOVWF   TBLPTRL
                BSF     EECON1, EEPGD           ; point to Flash program memory
                BCF     EECON1, CFGS            ; access Flash program memory
                BSF     EECON1, WREN            ; enable write to memory
                BSF     EECON1, FREE            ; enable Row Erase operation
                BCF     INTCON, GIE             ; disable interrupts
                MOVLW   55h
Required        MOVWF   EECON2                  ; write 55h
Sequence        MOVLW   0AAh
                MOVWF   EECON2                  ; write 0AAh
                BSF     EECON1, WR              ; start erase (CPU stall)
                BSF     INTCON, GIE             ; re-enable interrupts
                TBLRD*-                         ; dummy read decrement
                MOVLW   BUFFER_ADDR_HIGH        ; point to buffer
                MOVWF   FSR0H
                MOVLW   BUFFER_ADDR_LOW
                MOVWF   FSR0L
WRITE_BUFFER_BACK
                MOVLW   D’8                     ; number of bytes in holding register
                MOVWF   COUNTER
WRITE_BYTE_TO_HREGS
                MOVFF   POSTINC0, WREG          ; get low byte of buffer data
                MOVWF   TABLAT                  ; present data to table latch
                TBLWT+*                         ; write data, perform a short write
                                                ; to internal TBLWT holding register.
                DECFSZ  COUNTER                 ; loop until buffers are full
                BRA     WRITE_WORD_TO_HREGS
```

REGISTER 14-3: PWMCON0: PWM CONTROL REGISTER 0

U-0	R/W-1(1)	R/W-1(1)	R/W-1(1)	U-0	R/W-0	R/W-0	R/W-0
—	PWMEN2	PWMEN1	PWMEN0	—	PMOD2	PMOD1	PMOD0
bit 7							bit 0

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

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

bit 6-4 **PWMEN2:PWMEN0:** PWM Module Enable bits(1)

111 = All odd PWM I/O pins enabled for PWM output
110 = PWM1, PWM3 pins enabled for PWM output
10x = All PWM I/O pins enabled for PWM output
011 = PWM0, PWM1, PWM2 and PWM3 I/O pins enabled for PWM output
010 = PWM0 and PWM1 pins enabled for PWM output
001 = PWM1 pin is enabled for PWM output
000 = PWM module disabled; all PWM I/O pins are general purpose I/O

bit 3 **Unimplemented:** Read as '0'

bit 2-0 **PMOD2:PMOD0:** PWM Output Pair Mode bits

For PMOD0:
1 = PWM I/O pin pair (PWM0, PWM1) is in the Independent mode
0 = PWM I/O pin pair (PWM0, PWM1) is in the Complementary mode

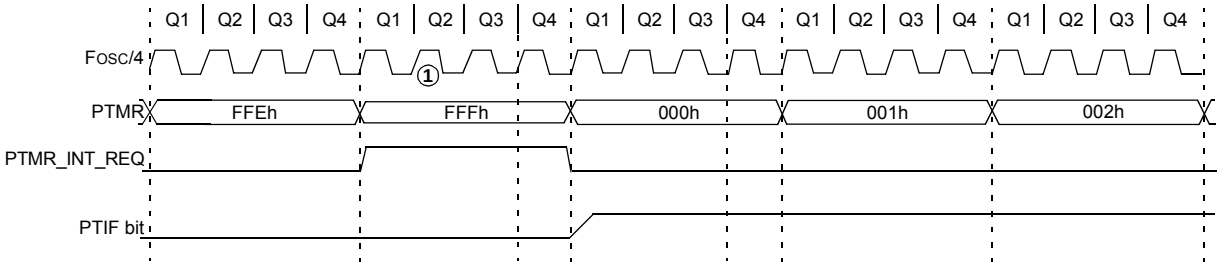
For PMOD1:
1 = PWM I/O pin pair (PWM2, PWM3) is in the Independent mode
0 = PWM I/O pin pair (PWM2, PWM3) is in the Complementary mode

For PMOD2:
1 = PWM I/O pin pair (PWM4, PWM5) is in the Independent mode
0 = PWM I/O pin pair (PWM4, PWM5) is in the Complementary mode

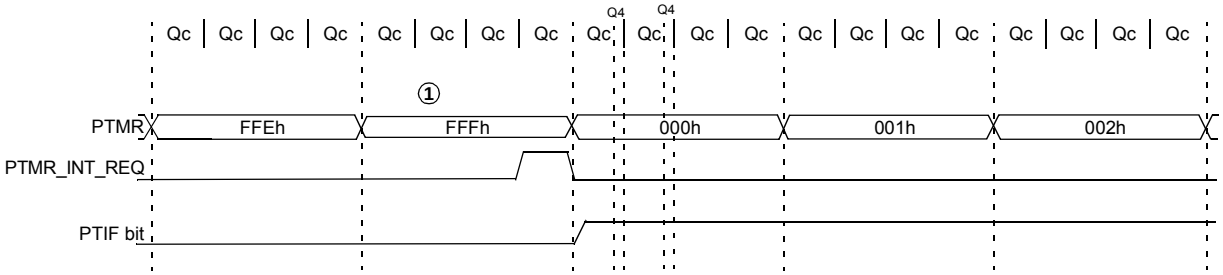
Note 1: Reset condition of PWMEN bits depends on the PWMPIN Configuration bit of CONFIG3L.

FIGURE 14-5: PWM TIME BASE INTERRUPT TIMING, FREE-RUNNING MODE

A: PRESCALER = 1:1



B: PRESCALER = 1:4



Note 1: PWM Time Base Period register, PTPER, is loaded with the value FFFh for this example.

14.4.2 INTERRUPTS IN SINGLE-SHOT MODE

When the PWM time base is in the Single-Shot mode (PTMOD<1:0> = 01), an interrupt event is generated when a match with the PTPER register occurs. The PWM Time Base register (PTMR) is reset to zero on the following input clock edge and the PTEN bit is cleared. The postscaler selection bits have no effect in this Timer mode.

14.4.3 INTERRUPTS IN CONTINUOUS UP/DOWN COUNT MODE

In the Continuous Up/Down Count mode (PTMOD<1:0> = 10), an interrupt event is generated each time the value of the PTMR register becomes zero and the PWM time base begins to count upwards. The postscaler selection bits may be used in this Timer mode to reduce the frequency of the interrupt events. Figure 14-7 shows the interrupts in Continuous Up/Down Count mode.

14.6.5 COMPLEMENTARY PWM OPERATION

The Complementary mode of PWM operation is useful to drive one or more power switches in half-bridge configuration, as shown in Figure 14-16. This inverter topology is typical for a 3-phase induction motor, brushless DC motor or 3-phase Uninterruptible Power Supply (UPS) control applications.

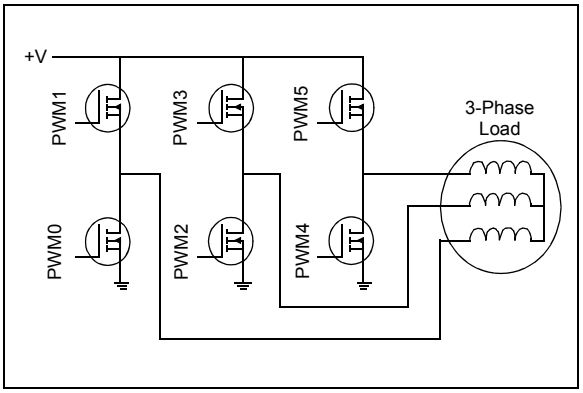
Each upper/lower power switch pair is fed by a complementary PWM signal. Dead time may be optionally inserted during device switching, where both outputs are inactive for a short period (see **Section 14.7 "Dead-Time Generators"**).

In Complementary mode, the duty cycle comparison units are assigned to the PWM outputs as follows:

- PDC0 register controls PWM1/PWM0 outputs
- PDC1 register controls PWM3/PWM2 outputs
- PDC2 register controls PWM5/PWM4 outputs

PWM1/3/5 are the main PWMs that are controlled by the PDCx registers and PWM0/2/4 are the complemented outputs. When using the PWMs to control the half-bridge, the odd number PWMs can be used to control the upper power switch and the even numbered PWMs can be used for the lower switches.

FIGURE 14-16: TYPICAL LOAD FOR COMPLEMENTARY PWM OUTPUTS



The Complementary mode is selected for each PWM I/O pin pair by clearing the appropriate PMODx bit in the PWMCON0 register. The PWM I/O pins are set to Complementary mode by default upon all kinds of device Resets.

The actual dead time is calculated from the DTCON register as follows:

Dead Time = Dead-Time Value/(FOSC/Prescaler)

Table 14-3 shows example dead-time ranges as a function of the input clock prescaler selected and the device operating frequency.

TABLE 14-3: EXAMPLE DEAD-TIME RANGES

Fosc (MHz)	MIPS	Prescaler Selection	Dead-Time Min	Dead-Time Max
40	10	FOSC/2	50 ns	3.2 μs
40	10	FOSC/4	100 ns	6.4 μs
40	10	FOSC/8	200 ns	12.8 μs
40	10	FOSC/16	400 ns	25.6 μs
32	8	FOSC/2	62.5 ns	4 μs
32	8	FOSC/4	125 ns	8 μs
32	8	FOSC/8	250 ns	16 μs
32	8	FOSC/16	500 ns	32 μs
25	6.25	FOSC/2	80 ns	5.12 μs
25	6.25	FOSC/4	160 ns	10.2 μs
25	6.25	FOSC/8	320 ns	20.5 μs
25	6.25	FOSC/16	640 ns	41 μs
20	5	FOSC/2	100 ns	6.4 μs
20	5	FOSC/4	200 ns	12.8 μs
20	5	FOSC/8	400 ns	25.6 μs
20	5	FOSC/16	800 ns	51.2 μs
10	2.5	FOSC/2	200 ns	12.8 μs
10	2.5	FOSC/4	400 ns	25.6 μs
10	2.5	FOSC/8	800 ns	51.2 μs
10	2.5	FOSC/16	1.6 μs	102.4 μs
5	1.25	FOSC/2	400 ns	25.6 μs
5	1.25	FOSC/4	800 ns	51.2 μs
5	1.25	FOSC/8	1.6 μs	102.4 μs
5	1.25	FOSC/16	3.2 μs	204.8 μs
4	1	FOSC/2	0.5 μs	32 μs
4	1	FOSC/4	1 μs	64 μs
4	1	FOSC/8	2 μs	128 μs
4	1	FOSC/16	4 μs	256 μs

14.7.4 DEAD-TIME DISTORTION

Note 1: For small PWM duty cycles, the ratio of dead time to the active PWM time may become large. In this case, the inserted dead time will introduce distortion into waveforms produced by the PWM module. The user can ensure that dead-time distortion is minimized by keeping the PWM duty cycle at least three times larger than the dead time. A similar effect occurs for duty cycles at or near 100%. The maximum duty cycle used in the application should be chosen such that the minimum inactive time of the signal is at least three times larger than the dead time. If the dead time is greater or equal to the duty cycle of one of the PWM output pairs, then that PWM pair will be inactive for the whole period.

2: Changing the dead-time values in DTCON when the PWM is enabled may result in an undesirable situation. Disable the PWM (PTEN = 0) before changing the dead-time value.

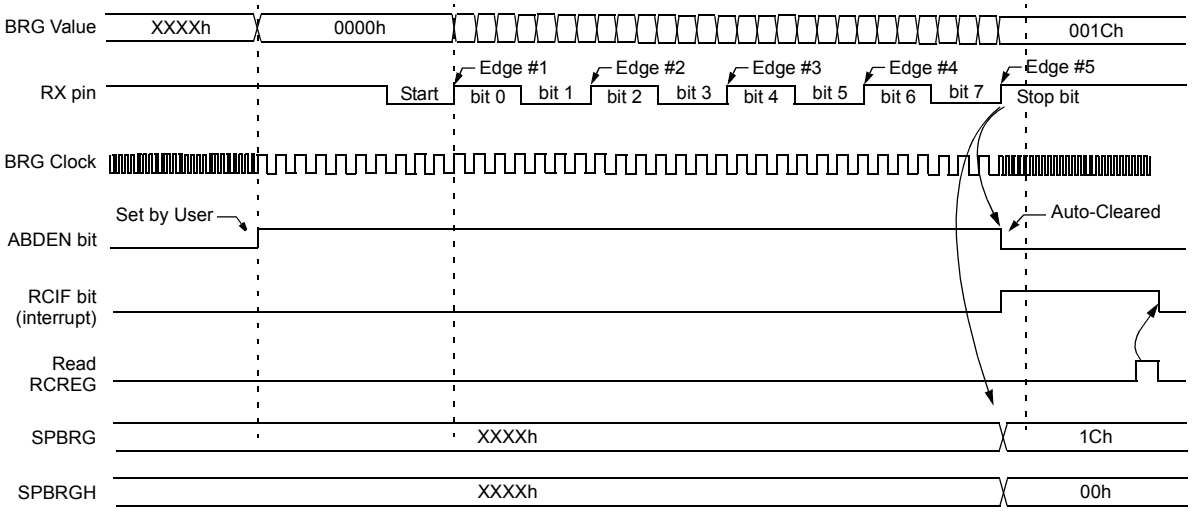
14.8 Independent PWM Output

Independent PWM mode is used for driving the loads (as shown in Figure 14-19) that drive one winding of a switched reluctance motor. A particular PWM output pair is configured in the Independent Output mode when the corresponding PMODx bit in the PWMCON0 register is set. No dead-time control is implemented between the PWM I/O pins when the module is operating in the Independent PWM mode and both I/O pins are allowed to be active simultaneously. This mode can also be used to drive stepper motors.

14.8.1 DUTY CYCLE ASSIGNMENT IN THE INDEPENDENT PWM MODE

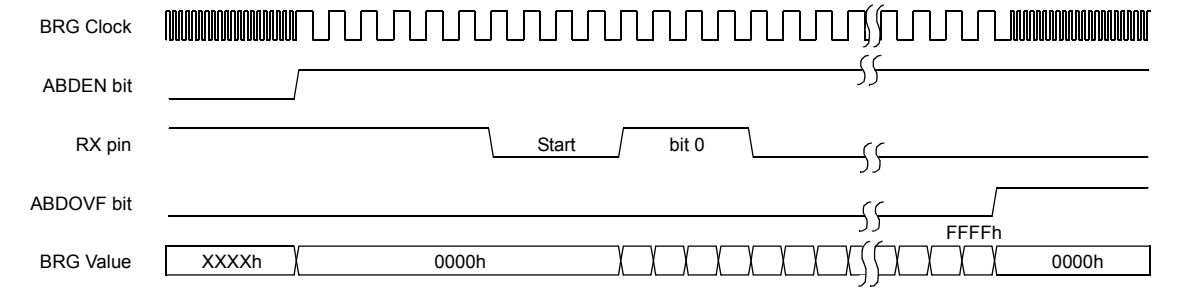
In the Independent PWM mode, each duty cycle generator is connected to both PWM output pins in a given PWM output pair. The odd and the even PWM output pins are driven with a single PWM duty cycle generator. PWM1 and PWM0 are driven by the PWM channel which uses the PDC0 register to set the duty cycle, PWM3 and PWM2 with PDC1, and PWM5 and PWM4 with PDC2 (see Figure 14-3 and Register 14-3).

FIGURE 15-1: AUTOMATIC BAUD RATE CALCULATION



Note: The ABD sequence requires the EUSART module to be configured in Asynchronous mode and WUE = 0.

FIGURE 15-2: BRG OVERFLOW SEQUENCE



15.2.5 BREAK CHARACTER SEQUENCE

The EUSART module has the capability of sending the special Break character sequences that are required by the LIN/J2602 bus standard. The Break character transmit consists of a Start bit, followed by twelve '0' bits and a Stop bit. The Frame Break character is sent whenever the SENDB and TXEN bits (TXSTA<3> and TXSTA<5>) are set while the Transmit Shift register is loaded with data. Note that the value of data written to TXREG will be ignored and all '0's will be transmitted.

The SENDB bit is automatically reset by hardware after the corresponding Stop bit is sent. This allows the user to preload the transmit FIFO with the next transmit byte following the Break character (typically, the Sync character in the LIN/J2602 specification).

Note that the data value written to the TXREG for the Break character is ignored. The write simply serves the purpose of initiating the proper sequence.

The TRMT bit indicates when the transmit operation is active or Idle, just as it does during normal transmission. See Figure 15-10 for the timing of the Break character sequence.

15.2.5.1 Break and Sync Transmit Sequence

The following sequence will send a message frame header made up of a Break, followed by an Auto-Baud Sync byte. This sequence is typical of a LIN/J2602 bus master.

1. Configure the EUSART for the desired mode.
2. Set the TXEN and SENDB bits to set up the Break character.
3. Load the TXREG with a dummy character to initiate transmission (the value is ignored).
4. Write '55h' to TXREG to load the Sync character into the transmit FIFO buffer.
5. After the Break has been sent, the SENDB bit is reset by hardware. The Sync character now transmits in the preconfigured mode.

When the TXREG becomes empty, as indicated by the TXIF, the next data byte can be written to TXREG.

15.2.6 RECEIVING A BREAK CHARACTER

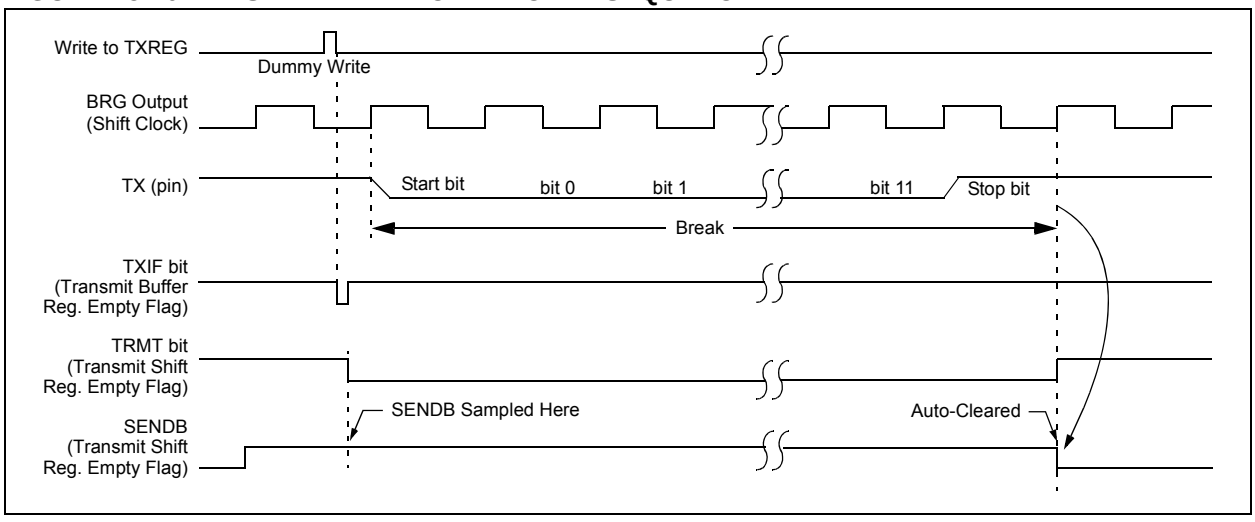
The Enhanced USART module can receive a Break character in two ways.

The first method forces configuration of the baud rate at a frequency of 9/13 the typical speed. This allows for the Stop bit transition to be at the correct sampling location (13 bits for Break versus Start bit and 8 data bits for typical data).

The second method uses the auto-wake-up feature described in **Section 15.2.4 "Auto-wake-up on Sync Break Character"**. By enabling this feature, the EUSART will sample the next two transitions on RX/DT, cause an RCIF interrupt and receive the next data byte followed by another interrupt.

Note that following a Break character, the user will typically want to enable the Auto-Baud Rate Detect feature. For both methods, the user can set the ABDEN bit once the TXIF interrupt is observed.

FIGURE 15-10: SEND BREAK CHARACTER SEQUENCE



REGISTER 16-2: ADCON1: A/D CONTROL REGISTER 1

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0
bit 7							bit 0

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

bit 7-5 **Unimplemented:** Read as '0'

bit 4 **VCFG0:** Voltage Reference Configuration bit (V_{REF+} source)
1 = Positive reference for the A/D is V_{REF+}
0 = Positive reference for the A/D is AV_{DD}

bit 3 **PCFG3:** A/D Port Configuration bit for RA6/AN3
0 = Port is configured as AN3
1 = Port is configured as RA6

bit 2 **PCFG2:** A/D Port Configuration bit for RA4/AN2
0 = Port is configured as AN2
1 = Port is configured as RA4

bit 1 **PCFG1:** A/D Port Configuration bit for RA1/AN1
0 = Port is configured as AN1
1 = Port is configured as RA1

bit 0 **PCFG0:** A/D Port Configuration bit for RA0/AN0
0 = Port is configured as AN0
1 = Port is configured as RA0

The analog reference voltage is software selectable to the device's positive supply voltage (V_{DD}), or the voltage level on the RA4/T0CKI/AN2/V_{REF+} pin.

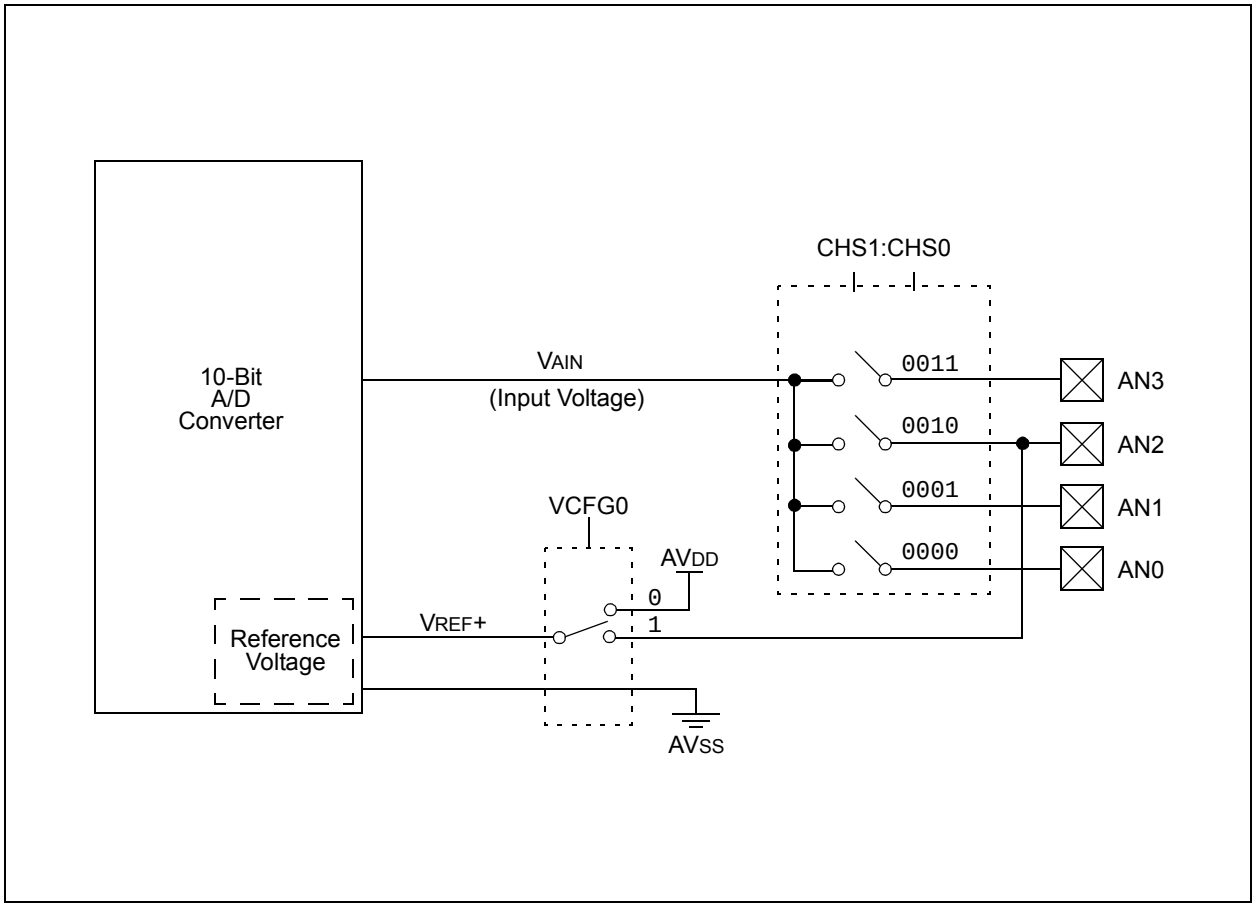
The A/D Converter has a unique feature of being able to operate while the device is in Sleep mode. To operate in Sleep, the A/D conversion clock must be derived from the A/D Converter's internal RC oscillator.

The output of the sample and hold is the input into the A/D Converter, which generates the result via successive approximation.

A device Reset forces all registers to their Reset state. This forces the A/D module to be turned off and any conversion in progress is aborted.

Each port pin associated with the A/D Converter can be configured as an analog input or as a digital I/O. The ADRESH and ADRESL registers contain the result of the A/D conversion. When the A/D conversion is complete, the result is loaded into the ADRESH:ADRESL register pair, the GO/$\overline{\text{DONE}}$ bit (ADCON0 register) is cleared and A/D Interrupt Flag bit, ADIF, is set. The block diagram of the A/D module is shown in Figure 16-1.

FIGURE 16-1: A/D BLOCK DIAGRAM



NOTES:

REGISTER 20-11: CONFIG7L: CONFIGURATION REGISTER 7 LOW (BYTE ADDRESS 30000Ch)

U-0	U-0	U-0	U-0	U-0	U-0	R/C-1	R/C-1
—	—	—	—	—	—	EBTR1(1)	EBTR0(1)
bit 7							bit 0

Legend:		
R = Readable bit	C = Clearable bit	U = Unimplemented bit, read as '0'
-n = Value when device is unprogrammed		u = Unchanged from programmed state

bit 7-2 **Unimplemented:** Read as '0'

bit 1 **EBTR1:** Table Read Protection bit (Block 1 Code Memory Area)
1 = Block 1 is not protected from table reads executed in other blocks
0 = Block 1 is protected from table reads executed in other blocks

bit 0 **EBTR0:** Table Read Protection bit (Block 0 Code Memory Area)
1 = Block 0 is not protected from table reads executed in other blocks
0 = Block 0 is protected from table reads executed in other blocks

Note 1: It is recommended to enable the corresponding CPx bit to protect block from external read operations.

REGISTER 20-12: CONFIG7H: CONFIGURATION REGISTER 7 HIGH (BYTE ADDRESS 30000Dh)

U-0	R/C-1	U-0	U-0	U-0	U-0	U-0	U-0
—	EBTRB(1)	—	—	—	—	—	—
bit 7							bit 0

Legend:		
R = Readable bit	C = Clearable bit	U = Unimplemented bit, read as '0'
-n = Value when device is unprogrammed		u = Unchanged from programmed state

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

bit 6 **EBTRB:** Table Read Protection bit (Boot Block Memory Area)
1 = Boot Block is not protected from table reads executed in other blocks
0 = Boot Block is protected from table reads executed in other blocks

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

Note 1: It is recommended to enable the corresponding CPx bit to protect block from external read operations.

20.3 Two-Speed Start-up

The Two-Speed Start-up feature helps to minimize the latency period from oscillator start-up to code execution by allowing the microcontroller to use the INTOSC oscillator as a clock source until the primary clock source is available. It is enabled by setting the IESO Configuration bit.

Two-Speed Start-up should be enabled only if the primary oscillator mode is LP, XT, HS or HSPLL (crystal-based modes). Other sources do not require an OST start-up delay; for these, Two-Speed Start-up should be disabled.

When enabled, Resets and wake-ups from Sleep mode cause the device to configure itself to run from the internal oscillator block as the clock source, following the time-out of the Power-up Timer, after a Power-on Reset is enabled. This allows almost immediate code execution while the primary oscillator starts and the OST is running. Once the OST times out, the device automatically switches to PRI_RUN mode.

To use a higher clock speed on wake-up, the INTOSC or postscaler clock sources can be selected to provide a higher clock speed by setting bits, IRCF2:IRCF0, immediately after Reset. For wake-ups from Sleep, the INTOSC or postscaler clock sources can be selected by setting the IRCF2:IRCF0 bits prior to entering Sleep mode.

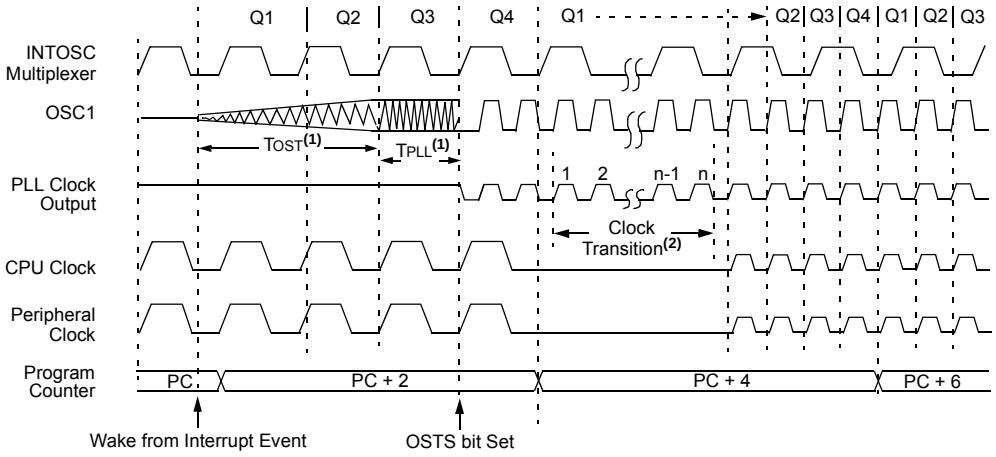
In all other power-managed modes, Two-Speed Start-up is not used. The device will be clocked by the currently selected clock source until the primary clock source becomes available. The setting of the IESO bit is ignored.

20.3.1 SPECIAL CONSIDERATIONS FOR USING TWO-SPEED START-UP

While using the INTOSC oscillator in Two-Speed Start-up, the device still obeys the normal command sequences for entering power-managed modes, including multiple SLEEP instructions (refer to **Section 4.1.4 "Multiple Sleep Commands"**). In practice, this means that user code can change the SCS1:SCS0 bit settings or issue SLEEP instructions before the OST times out. This would allow an application to briefly wake-up, perform routine "housekeeping" tasks and return to Sleep before the device starts to operate from the primary oscillator.

User code can also check if the primary clock source is currently providing the device clocking by checking the status of the OSTS bit (OSCCON<3>). If the bit is set, the primary oscillator is providing the clock. Otherwise, the internal oscillator block is providing the clock during wake-up from Reset or Sleep mode.

FIGURE 20-2: TIMING TRANSITION FOR TWO-SPEED START-UP (INTOSC TO HSPLL)



Note 1: TOST = 1024 TOSC; TPLL = 2 ms (approx). These intervals are not shown to scale.

2: Clock transition typically occurs within 2-4 TOSC.

NEGF **Negate f**

Syntax: NEGF f {,a}

Operands: $0 \le f \le 255$
$a \in [0,1]$

Operation: $(\bar{f}) + 1 \rightarrow f$

Status Affected: N, OV, C, DC, Z

Encoding:

0110	110a	ffff	ffff

Description: Location 'f' is negated using two's complement. The result is placed in the data memory location 'f'.
If 'a' is '0', the Access Bank is selected.
If 'a' is '1', the BSR is used to select the GPR bank.
If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \le 95$ (5Fh). See **Section 22.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write register 'f'

Example: NEGF REG, 1

Before Instruction
REG = 0011 1010 [3Ah]
After Instruction
REG = 1100 0110 [C6h]

NOP **No Operation**

Syntax: NOP

Operands: None

Operation: No operation

Status Affected: None

Encoding:

0000 1111	0000 xxxx	0000 xxxx	0000 xxxx

Description: No operation.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No operation	No operation	No operation

Example:

None.

POP **Pop Top of Return Stack**

Syntax: POP

Operands: None

Operation: (TOS) → bit bucket

Status Affected: None

Encoding:

0000	0000	0000	0110

Description: The TOS value is pulled off the return stack and is discarded. The TOS value then becomes the previous value that was pushed onto the return stack. This instruction is provided to enable the user to properly manage the return stack to incorporate a software stack.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No operation	POP TOS value	No operation

Example:
```
POP
GOTO    NEW
```
Before Instruction
TOS = 0031A2h
Stack (1 level down) = 014332h

After Instruction
TOS = 014332h
PC = NEW

PUSH **Push Top of Return Stack**

Syntax: PUSH

Operands: None

Operation: (PC + 2) → TOS

Status Affected: None

Encoding:

0000	0000	0000	0101

Description: The PC + 2 is pushed onto the top of the return stack. The previous TOS value is pushed down on the stack. This instruction allows implementing a software stack by modifying TOS and then pushing it onto the return stack.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	PUSH PC + 2 onto return stack	No operation	No operation

Example:
```
PUSH
```
Before Instruction
TOS = 345Ah
PC = 0124h

After Instruction
PC = 0126h
TOS = 0126h
Stack (1 level down) = 345Ah

NOTES: