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

"<u>Embedded - Microcontrollers</u>" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.
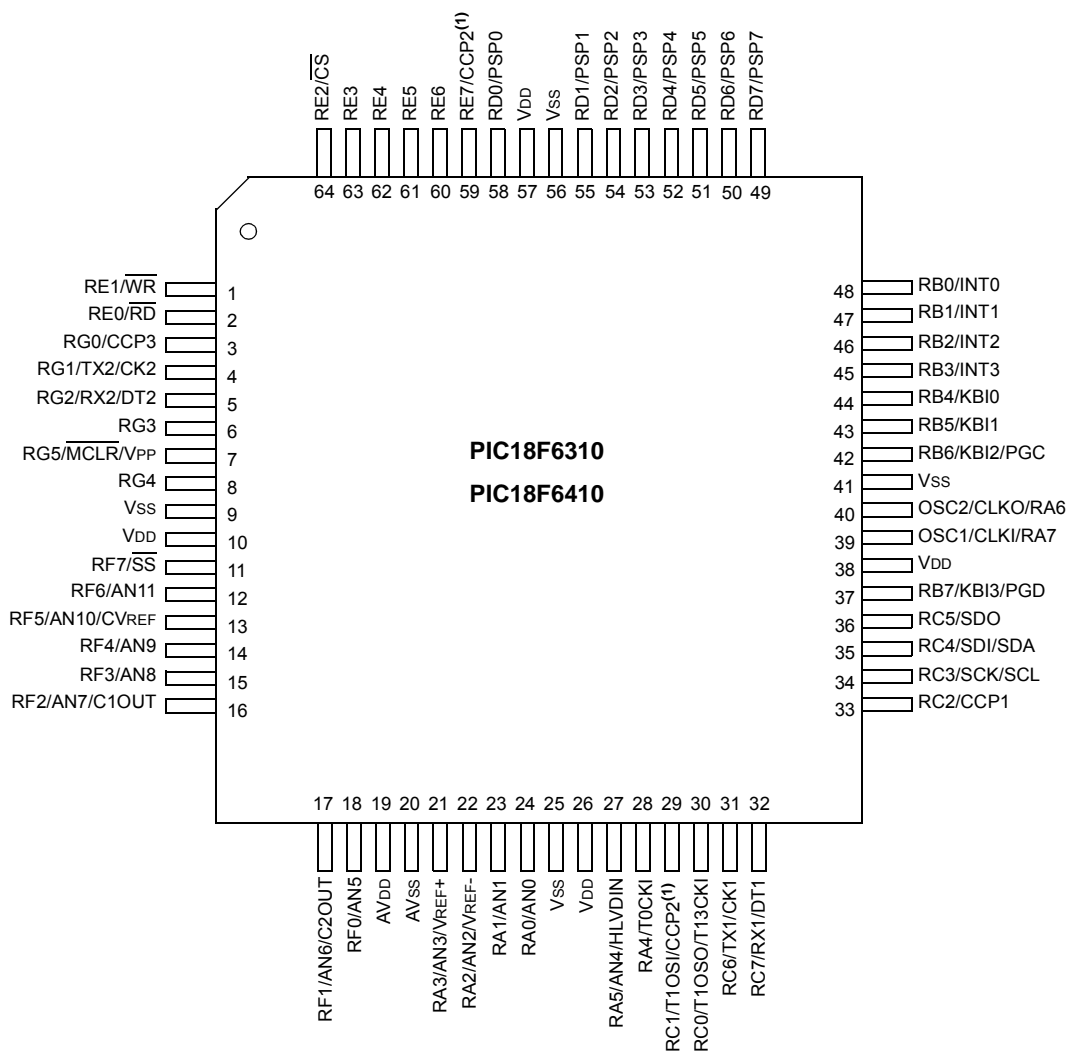
Applications of "<u>Embedded - Microcontrollers</u>"

Details

Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	40MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, HLVD, POR, PWM, WDT
Number of I/O	54
Program Memory Size	16KB (8K x 16)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	768 x 8
Voltage - Supply (Vcc/Vdd)	4.2V ~ 5.5V
Data Converters	A/D 12x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-TQFP
Supplier Device Package	64-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f6410-i-pt

PIC18F6310/6410/8310/8410

Pin Diagrams

64-Pin TQFP

Top pins (64–49): RE2/$\overline{\text{CS}}$, RE3, RE4, RE5, RE6, RE7/CCP2[(1)], RD0/PSP0, VDD, VSS, RD1/PSP1, RD2/PSP2, RD3/PSP3, RD4/PSP4, RD5/PSP5, RD6/PSP6, RD7/PSP7

Left pins (1–16):
1 — RE1/$\overline{\text{WR}}$
2 — RE0/$\overline{\text{RD}}$
3 — RG0/CCP3
4 — RG1/TX2/CK2
5 — RG2/RX2/DT2
6 — RG3
7 — RG5/$\overline{\text{MCLR}}$/VPP
8 — RG4
9 — VSS
10 — VDD
11 — RF7/$\overline{\text{SS}}$
12 — RF6/AN11
13 — RF5/AN10/CVREF
14 — RF4/AN9
15 — RF3/AN8
16 — RF2/AN7/C1OUT

Center label: **PIC18F6310** **PIC18F6410**

Right pins (48–33):
48 — RB0/INT0
47 — RB1/INT1
46 — RB2/INT2
45 — RB3/INT3
44 — RB4/KBI0
43 — RB5/KBI1
42 — RB6/KBI2/PGC
41 — VSS
40 — OSC2/CLKO/RA6
39 — OSC1/CLKI/RA7
38 — VDD
37 — RB7/KBI3/PGD
36 — RC5/SDO
35 — RC4/SDI/SDA
34 — RC3/SCK/SCL
33 — RC2/CCP1

Bottom pins (17–32):
17 — RF1/AN6/C2OUT
18 — RF0/AN5
19 — AVDD
20 — AVSS
21 — RA3/AN3/VREF+
22 — RA2/AN2/VREF-
23 — RA1/AN1
24 — RA0/AN0
25 — VSS
26 — VDD
27 — RA5/AN4/HLVDIN
28 — RA4/T0CKI
29 — RC1/T1OSI/CCP2[(1)]
30 — RC0/T1OSO/T13CKI
31 — RC6/TX1/CK1
32 — RC7/RX1/DT1

Note 1: RE7 is the alternate pin for CCP2 multiplexing.

TABLE 1-3: **PIC18F8310/8410 PINOUT I/O DESCRIPTIONS (CONTINUED)**

Pin Name	Pin Number TQFP	Pin Type	Buffer Type	Description
				PORTC is a bidirectional I/O port.
RC0/T1OSO/T13CKI	36			
RC0		I/O	ST	Digital I/O.
T1OSO		O	—	Timer1 oscillator output.
T13CKI		I	ST	Timer1/Timer3 external clock input.
RC1/T1OSI/CCP2	35			
RC1		I/O	ST	Digital I/O.
T1OSI		I	CMOS	Timer1 oscillator input.
CCP2[2]		I/O	ST	Capture 2 input/Compare 2 output/PWM2 output.
RC2/CCP1	43			
RC2		I/O	ST	Digital I/O.
CCP1		I/O	ST	Capture 1 input/Compare 1 output/PWM1 output.
RC3/SCK/SCL	44			
RC3		I/O	ST	Digital I/O.
SCK		I/O	ST	Synchronous serial clock input/output for SPI mode.
SCL		I/O	I^{2}C	Synchronous serial clock input/output for I^{2}C mode.
RC4/SDI/SDA	45			
RC4		I/O	ST	Digital I/O.
SDI		I	ST	SPI data in.
SDA		I/O	I^{2}C	I^{2}C data I/O.
RC5/SDO	46			
RC5		I/O	ST	Digital I/O.
SDO		O	—	SPI data out.
RC6/TX1/CK1	37			
RC6		I/O	ST	Digital I/O.
TX1		O	—	EUSART1 asynchronous transmit.
CK1		I/O	ST	EUSART1 synchronous clock (see related RX1/DT1).
RC7/RX1/DT1	38			
RC7		I/O	ST	Digital I/O.
RX1		I	ST	EUSART1 asynchronous receive.
DT1		I/O	ST	EUSART1 synchronous data (see related TX1/CK1).

Legend: TTL = TTL compatible input CMOS = CMOS compatible input or output
ST = Schmitt Trigger input with CMOS levels Analog = Analog input
I = Input O = Output
P = Power I^{2}C = ST with I^{2}C™ or SMB levels

Note 1: Alternate assignment for CCP2 when Configuration bit, CCP2MX, is cleared (all operating modes except Microcontroller mode).

2: Default assignment for CCP2 in all operating modes (CCP2MX is set).

3: Alternate assignment for CCP2 when CCP2MX is cleared (Microcontroller mode only).

TABLE 1-3: **PIC18F8310/8410 PINOUT I/O DESCRIPTIONS (CONTINUED)**

Pin Name	Pin Number TQFP	Pin Type	Buffer Type	Description
				PORTF is a bidirectional I/O port.
RF0/AN5	24			
RF0		I/O	ST	Digital I/O.
AN5		I	Analog	Analog Input 5.
RF1/AN6/C2OUT	23			
RF1		I/O	ST	Digital I/O.
AN6		I	Analog	Analog Input 6.
C2OUT		O	—	Comparator 2 output.
RF2/AN7/C1OUT	18			
RF2		I/O	ST	Digital I/O.
AN7		I	Analog	Analog Input 7.
C1OUT		O	—	Comparator 1 output.
RF3/AN8	17			
RF3		I/O	ST	Digital I/O.
AN8		I	Analog	Analog Input 8.
RF4/AN9	16			
RF4		I/O	ST	Digital I/O.
AN9		I	Analog	Analog Input 9.
RF5/AN10/CV$_{REF}$	15			
RF5		I/O	ST	Digital I/O.
AN10		I	Analog	Analog Input 10.
CV$_{REF}$		O	Analog	Comparator reference voltage output.
RF6/AN11	14			
RF6		I/O	ST	Digital I/O.
AN11		I	Analog	Analog Input 11.
RF7/$\overline{SS}$	13			
RF7		I/O	ST	Digital I/O.
$\overline{SS}$		I	TTL	SPI slave select input.

Legend: TTL = TTL compatible input   CMOS = CMOS compatible input or output
ST = Schmitt Trigger input with CMOS levels   Analog = Analog input
I = Input   O = Output
P = Power   I^{2}C = ST with I^{2}C™ or SMB levels

Note 1: Alternate assignment for CCP2 when Configuration bit, CCP2MX, is cleared (all operating modes except Microcontroller mode).

    2: Default assignment for CCP2 in all operating modes (CCP2MX is set).

    3: Alternate assignment for CCP2 when CCP2MX is cleared (Microcontroller mode only).

8.4 Operation in Power-Managed Modes

In alternate, power-managed Run modes, the external bus continues to operate normally. If a clock source with a lower speed is selected, bus operations will run at that speed. In these cases, excessive access times for the external memory may result if wait states have been enabled and added to external memory operations.

If operations in a lower power Run mode are antici-pated, users should provide in their applications for adjusting memory access times at the lower clock speeds.

In Sleep and Idle modes, the microcontroller core does not need to access data; bus operations are sus-pended. The state of the external bus is frozen with the address/data pins and most of the control pins holding at the same state they were in when the mode was invoked. The only potential changes are the CE, LB and UB pins which are held at logic high.

TABLE 8-2: REGISTERS ASSOCIATED WITH THE EXTERNAL MEMORY INTERFACE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page
MEMCON	EBDIS	—	WAIT1	WAIT0	—	—	WM1	WM0	65
CONFIG3L	WAIT	BW	—	—	—	—	PM1	PM0	285
CONFIG3H	MCLRE	—	—	—	—	LPT1OSC	—	CCP2MX	286

Legend: — = unimplemented, read as '0'. Shaded cells are not used for the external memory interface.

11.3 PORTC, TRISC and LATC Registers

PORTC is an 8-bit wide, bidirectional port. The corresponding Data Direction register is TRISC. Setting a TRISC bit (= 1) will make the corresponding PORTC pin an input (i.e., put the corresponding output driver in a High-Impedance mode). Clearing a TRISC bit (= 0) will make the corresponding PORTC pin an output (i.e., put the contents of the output latch on the selected pin).

The Output Latch register (LATC) is also memory mapped. Read-modify-write operations on the LATC register read and write the latched output value for PORTC.

PORTC is multiplexed with several peripheral functions (Table 11-5). The pins have Schmitt Trigger input buffers. RC1 is normally configured by Configuration bit, CCP2MX, as the default peripheral pin of the CCP2 module (default/erased state, CCP2MX = 1).

When enabling peripheral functions, care should be taken in defining TRIS bits for each PORTC pin. Some peripherals override the TRIS bit to make a pin an output, while other peripherals override the TRIS bit to make a pin an input. The user should refer to the corresponding peripheral section for the correct TRIS bit settings.

> **Note:** On a Power-on Reset, these pins are configured as digital inputs.

The contents of the TRISC register are affected by peripheral overrides. Reading TRISC always returns the current contents, even though a peripheral device may be overriding one or more of the pins.

EXAMPLE 11-3: INITIALIZING PORTC

```
CLRF    PORTC   ; Initialize PORTC by
                ; clearing output
                ; data latches
CLRF    LATC    ; Alternate method
                ; to clear output
                ; data latches
MOVLW   0CFh    ; Value used to
                ; initialize data
                ; direction
MOVWF   TRISC   ; Set RC<3:0> as inputs
                ; RC<5:4> as outputs
                ; RC<7:6> as inputs
```

16.2.3 SOFTWARE INTERRUPT

When the Capture mode is changed, a false capture interrupt may be generated. The user should keep bit, CCP2IE (PIE2<1>), clear to avoid false interrupts and should clear the flag bit, CCP2IF, following any such change in operating mode.

16.2.4 CCP PRESCALER

There are four prescaler settings in Capture mode; they are specified as part of the operating mode selected by the mode select bits (CCP2M<3:0>). Whenever the CCP module is turned off, or the CCP module is not in Capture mode, the prescaler counter is cleared. This means that any Reset will clear the prescaler counter.

Switching from one capture prescaler to another may generate an interrupt. Also, the prescaler counter will not be cleared, therefore, the first capture may be from a non-zero prescaler. Example 16-1 shows the recommended method for switching between capture prescalers. This example also clears the prescaler counter and will not generate the "false" interrupt.

EXAMPLE 16-1: CHANGING BETWEEN CAPTURE PRESCALERS

```
CLRF    CCP2CON       ; Turn CCP module off
MOVLW   NEW_CAPT_PS   ; Load WREG with the
                      ; new prescaler mode
                      ; value and CCP ON
MOVWF   CCP2CON       ; Load CCP2CON with
                      ; this value
```

16.3 Compare Mode

In Compare mode, the 16-bit CCPR2 register value is constantly compared against either the TMR1 or TMR3 register pair value. When a match occurs, the CCP2 pin can be:

- driven high
- driven low
- toggled (high-to-low or low-to-high)
- remain unchanged (that is, reflects the state of the I/O latch)

The action on the pin is based on the value of the mode select bits (CCP2M<3:0>). At the same time, the interrupt flag bit, CCP2IF, is set.

16.3.1 CCP PIN CONFIGURATION

The user must configure the CCPx pin as an output by clearing the appropriate TRIS bit.

> **Note:** Clearing the CCPxCON register will force the RC1 or RE7 compare output latch (depending on device configuration) to the default low level. This is not the PORTC or PORTE I/O data latch.

16.3.2 TIMER1/TIMER3 MODE SELECTION

Timer1 and/or Timer3 must be running in Timer mode, or Synchronized Counter mode, if the CCP module is using the compare feature. In Asynchronous Counter mode, the compare operation may not work.

16.3.3 SOFTWARE INTERRUPT MODE

When the Generate Software Interrupt mode is chosen (CCP2M<3:0> = 1010), the CCP2 pin is not affected. Only a CCP interrupt is generated if enabled and the CCP2IE bit is set.

16.3.4 SPECIAL EVENT TRIGGERS

CCP1 and CCP2 are both equipped with a Special Event Trigger. This is an internal hardware signal, generated in Compare mode, to trigger actions by other modules. The Special Event Trigger is enabled by selecting the Compare Special Event Trigger mode (CCP2M<3:0> = 1011).

For either CCP module, the Special Event Trigger resets the Timer register pair for whichever timer resource is currently assigned as the module's time base. This allows the CCPRx registers to serve as a programmable period register for either timer.

The Special Event Trigger for CCP2 can also start an A/D conversion. In order to do this, the A/D Converter must already be enabled.

> **Note:** The Special Event Trigger of CCP1 only resets Timer1/Timer3 and cannot start an A/D conversion even when the A/D Converter is enabled.

CCP3 is not equipped with a Special Event Trigger. Selecting the Compare Special Event Trigger mode for this device (CCP3M<3:0> = 1011) is functionally the same as selecting the Generate Software Interrupt mode (CCP3M<3:0> = 1010).

16.4 PWM Mode

In Pulse-Width Modulation (PWM) mode, the CCP2 pin produces up to a 10-bit resolution PWM output. Since the CCP2 pin is multiplexed with a PORTC or PORTE data latch, the appropriate TRIS bit must be cleared to make the CCP2 pin an output.

> **Note:** Clearing the CCP2CON register will force the RC1 or RE7 output latch (depending on device configuration) to the default low level. This is not the PORTC or PORTE I/O data latch.

Figure 16-4 shows a simplified block diagram of the CCP module in PWM mode.

For a step-by-step procedure on how to set up the CCP module for PWM operation, see **Section 16.4.3 "Setup for Pwm Operation"**.

FIGURE 16-4:   **SIMPLIFIED PWM BLOCK DIAGRAM**
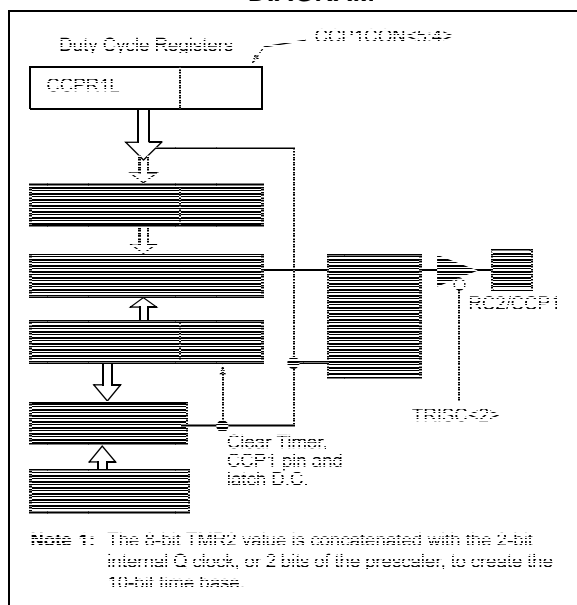
A PWM output (Figure 16-5) has a time base (period) and a time that the output stays high (duty cycle). The frequency of the PWM is the inverse of the period (1/period).

FIGURE 16-5:   **PWM OUTPUT**
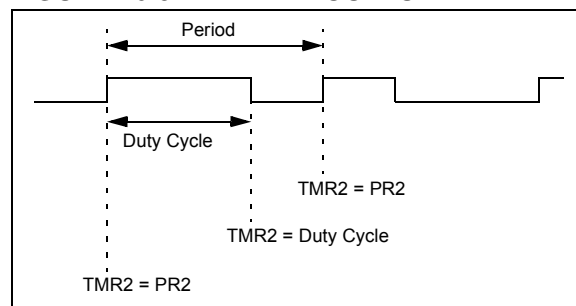
16.4.1 PWM PERIOD

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

EQUATION 16-1:

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

PWM frequency is defined as 1/[PWM period].

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

- TMR2 is cleared
- The CCP2 pin is set (exception: if PWM duty cycle = 0%, the CCP2 pin will not be set)
- The PWM duty cycle is latched from CCPR2L into CCPR2H

> **Note:** The Timer2 postscalers (see **Section 14.0 "Timer2 Module"**) are not used in the determination of the PWM frequency. The postscaler could be used to have a servo update rate at a different frequency than the PWM output.

FIGURE 17-9: I²C™ SLAVE MODE TIMING (TRANSMISSION, 7-BIT ADDRESS)
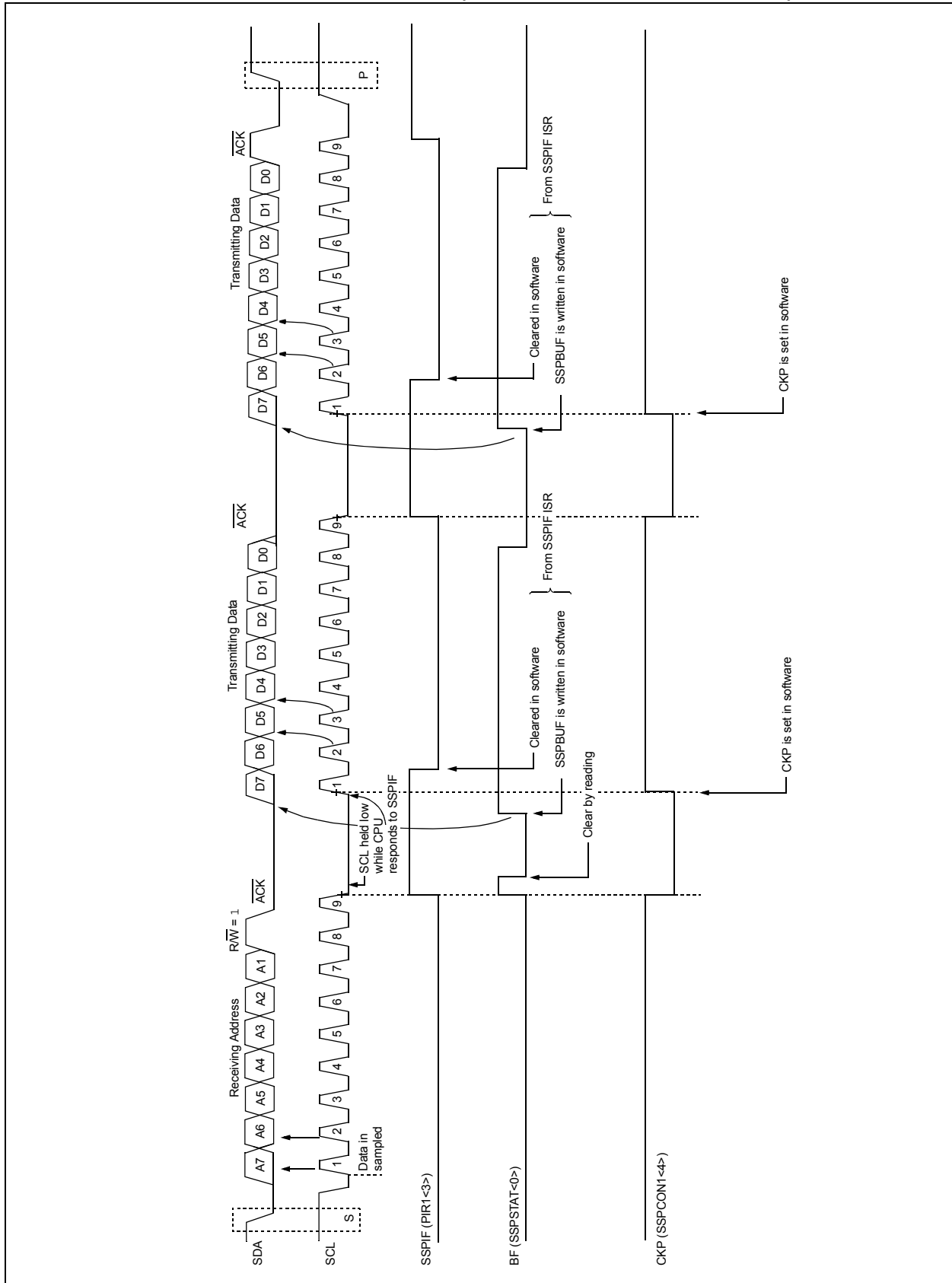
17.4.10   I²C MASTER MODE TRANSMISSION

Transmission of a data byte, a 7-bit address or the other half of a 10-bit address is accomplished by simply writing a value to the SSPBUF register. This action will set the Buffer Full flag bit, BF, and allow the Baud Rate Generator to begin counting and start the next transmission. Each bit of address/data will be shifted out onto the SDA pin after the falling edge of SCL is asserted (see data hold time specification Parameter #106). SCL is held low for one Baud Rate Generator rollover count (T$_{BRG}$). Data should be valid before SCL is released high (see data setup time specification Parameter #107). When the SCL pin is released high, it is held that way for T$_{BRG}$. The data on the SDA pin must remain stable for that duration and some hold time after the next falling edge of SCL. After the eighth bit is shifted out (the falling edge of the eighth clock), the BF flag is cleared and the master releases SDA. This allows the slave device being addressed to respond with an $\overline{ACK}$ bit during the ninth bit time if an address match occurred, or if data was received properly. The status of $\overline{ACK}$ is written into the ACKDT bit on the falling edge of the ninth clock. If the master receives an Acknowledge, the Acknowledge Status bit, ACKSTAT, is cleared. If not, the bit is set. After the ninth clock, the SSPIF bit is set and the master clock (Baud Rate Generator) is suspended until the next data byte is loaded into the SSPBUF, leaving SCL low and SDA unchanged (Figure 17-21).

After the write to the SSPBUF, each bit of address will be shifted out on the falling edge of SCL until all 7 address bits and the R/$\overline{W}$ bit are completed. On the falling edge of the eighth clock, the master will deassert the SDA pin, allowing the slave to respond with an Acknowledge. On the falling edge of the ninth clock, the master will sample the SDA pin to see if the address was recognized by a slave. The status of the $\overline{ACK}$ bit is loaded into the ACKSTAT status bit (SSPCON2<6>). Following the falling edge of the ninth clock transmission of the address, the SSPIF is set, the BF flag is cleared and the Baud Rate Generator is turned off until another write to the SSPBUF takes place, holding SCL low and allowing SDA to float.

17.4.10.1   BF Status Flag

In Transmit mode, the BF bit (SSPSTAT<0>) is set when the CPU writes to SSPBUF and is cleared when all 8 bits are shifted out.

17.4.10.2   WCOL Status Flag

If the user writes the SSPBUF when a transmit is already in progress (i.e., SSPSR is still shifting out a data byte), the WCOL is set and the contents of the buffer are unchanged (the write doesn't occur) after 2 T$_{CY}$ after the SSPBUF write. If SSPBUF is rewritten within 2 T$_{CY}$, the WCOL bit is set and SSPBUF is updated. This may result in a corrupted transfer. The user should verify that the WCOL flag is clear after each write to SSPBUF to ensure the transfer is correct.

17.4.10.3   ACKSTAT Status Flag

In Transmit mode, the ACKSTAT bit (SSPCON2<6>) is cleared when the slave has sent an Acknowledge ($\overline{ACK}$ = 0) and is set when the slave does not Acknowledge ($\overline{ACK}$ = 1). A slave sends an Acknowledge when it has recognized its address (including a general call), or when the slave has properly received its data.

17.4.11   I²C MASTER MODE RECEPTION

Master mode reception is enabled by programming the Receive Enable bit, RCEN (SSPCON2<3>).

> **Note:** The MSSP module must be in an Idle state before the RCEN bit is set or the RCEN bit will be disregarded.

The Baud Rate Generator begins counting and on each rollover, the state of the SCL pin changes (high-to-low/ low-to-high) and data is shifted into the SSPSR. After the falling edge of the eighth clock, the receive enable flag is automatically cleared, the contents of the SSPSR are loaded into the SSPBUF, the BF flag bit is set, the SSPIF flag bit is set and the Baud Rate Generator is suspended from counting, holding SCL low. The MSSP is now in Idle state awaiting the next command. When the buffer is read by the CPU, the BF flag bit is automatically cleared. The user can then send an Acknowledge bit at the end of reception by setting the Acknowledge Sequence Enable bit, ACKEN (SSPCON2<4>).

17.4.11.1   BF Status Flag

In receive operation, the BF bit is set when an address or data byte is loaded into SSPBUF from SSPSR. It is cleared when the SSPBUF register is read.

17.4.11.2   SSPOV Status Flag

In receive operation, the SSPOV bit is set when 8 bits are received into the SSPSR and the BF flag bit is already set from a previous reception.

17.4.11.3   WCOL Status Flag

If the user writes the SSPBUF when a receive is already in progress (i.e., SSPSR is still shifting in a data byte), the WCOL bit is set and the contents of the buffer are unchanged (the write doesn't occur).

17.4.14 SLEEP OPERATION

While in Sleep mode, the I^{2}C module can receive addresses or data and when an address match or complete byte transfer occurs, wake the processor from Sleep (if the MSSP interrupt is enabled).

17.4.15 EFFECT OF A RESET

A Reset disables the MSSP module and terminates the current transfer.

17.4.16 MULTI-MASTER MODE

In Multi-Master mode, the interrupt generation on the detection of the Start and Stop conditions allows the determination of when the bus is free. The Stop (P) and Start (S) bits are cleared from a Reset or when the MSSP module is disabled. Control of the I^{2}C bus may be taken when the P bit (SSPSTAT<4>) is set, or the bus is Idle, with both the S and P bits clear. When the bus is busy, enabling the MSSP interrupt will generate the interrupt when the Stop condition occurs.

In multi-master operation, the SDA line must be monitored for arbitration to see if the signal level is the expected output level. This check is performed in hardware with the result placed in the BCLIF bit.

The states where arbitration can be lost are:

• Address Transfer
• Data Transfer
• A Start Condition
• A Repeated Start Condition
• An Acknowledge Condition

17.4.17 MULTI -MASTER COMMUNICATION, BUS COLLISION AND BUS ARBITRATION

Multi-Master mode support is achieved by bus arbitration. When the master outputs address/data bits onto the SDA pin, arbitration takes place when the master outputs a '1' on SDA, by letting SDA float high and another master asserts a '0'. When the SCL pin floats high, data should be stable. If the expected data on SDA is a '1' and the data sampled on the SDA pin = 0, then a bus collision has taken place. The master will set the Bus Collision Interrupt Flag, BCLIF and reset the I^{2}C port to its Idle state (Figure 17-25).

If a transmit was in progress when the bus collision occurred, the transmission is halted, the BF flag is cleared, the SDA and SCL lines are deasserted and the SSPBUF can be written to. When the user services the bus collision Interrupt Service Routine and if the I^{2}C bus is free, the user can resume communication by asserting a Start condition.

If a Start, Repeated Start, Stop or Acknowledge condition was in progress when the bus collision occurred, the condition is aborted, the SDA and SCL lines are deasserted and the respective control bits in the SSPCON2 register are cleared. When the user services the bus collision Interrupt Service Routine and if the I^{2}C bus is free, the user can resume communication by asserting a Start condition.

The master will continue to monitor the SDA and SCL pins. If a Stop condition occurs, the SSPIF bit will be set.

A write to the SSPBUF will start the transmission of data at the first data bit regardless of where the transmitter left off when the bus collision occurred.

In Multi-Master mode, the interrupt generation on the detection of Start and Stop conditions allows the determination of when the bus is free. Control of the I^{2}C bus can be taken when the P bit is set in the SSPSTAT register, or the bus is Idle and the S and P bits are cleared.

FIGURE 17-25: BUS COLLISION TIMING FOR TRANSMIT AND ACKNOWLEDGE
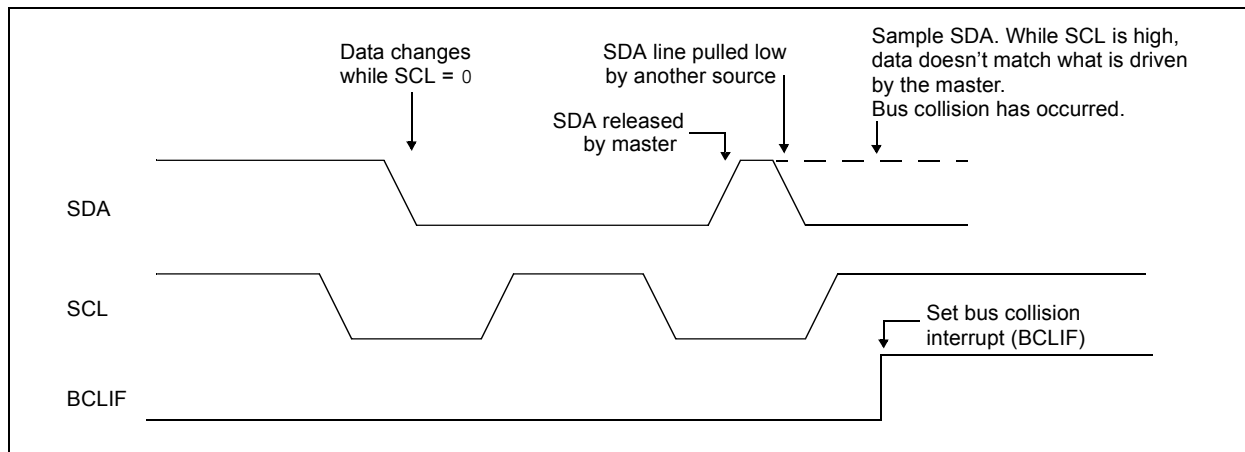
The module is enabled by setting the HLVDEN bit. Each time that the HLVD module is enabled, the circuitry requires some time to stabilize. The IRVST bit is a read-only bit and is used to indicate when the circuit is stable. The module can only generate an interrupt after the circuit is stable and IRVST is set.

The VDIRMAG bit determines the overall operation of the module. When VDIRMAG is cleared, the module monitors for drops in VDD below a predetermined set point. When the bit is set, the module monitors for rises in VDD above the set point.

23.1   Operation

When the HLVD module is enabled, a comparator uses an internally generated reference voltage as the set point. The set point is compared with the trip point, where each node in the resistor divider represents a trip point voltage. The "trip point" voltage is the voltage level at which the device detects a high or low-voltage event, depending on the configuration of the module. When the supply voltage is equal to the trip point, the voltage tapped off of the resistor array is equal to the internal reference voltage generated by the voltage reference module. The comparator then generates an interrupt signal by setting the HLVDIF bit.

The trip point voltage is software programmable to any one of 16 values. The trip point is selected by programming the HLVDL<3:0> bits (HLVDCON<3:0>).

The HLVD module has an additional feature that allows the user to supply the trip voltage to the module from an external source. This mode is enabled when bits, HLVDL<3:0>, are set to '1111'. In this state, the comparator input is multiplexed from the external input pin, HLVDIN. This gives users flexibility because it allows them to configure the High/Low-Voltage Detect interrupt to occur at any voltage in the valid operating range.

FIGURE 23-1:   HLVD MODULE BLOCK DIAGRAM (WITH EXTERNAL INPUT)
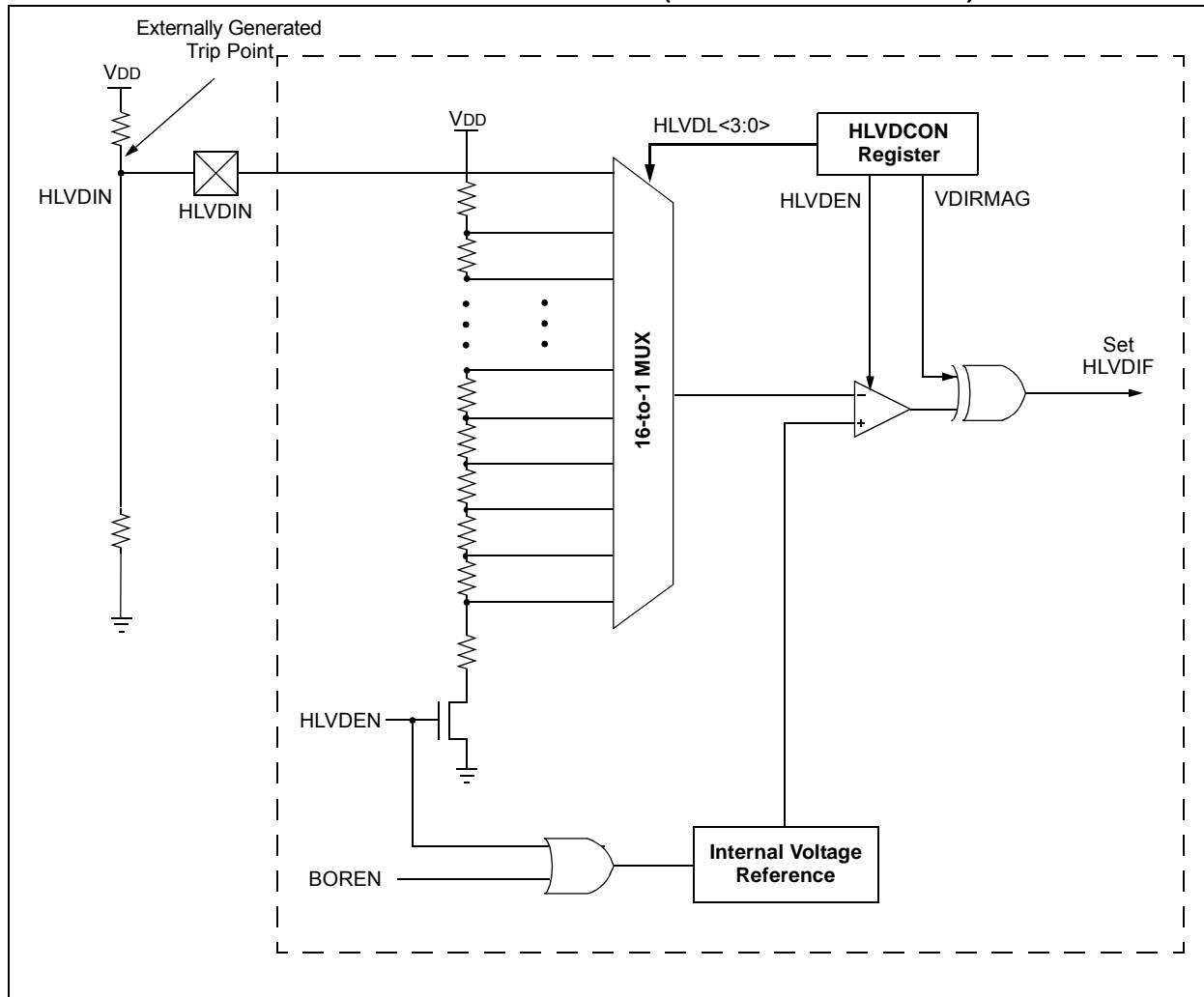
REGISTER 24-8: CONFIG7L: CONFIGURATION REGISTER 7 LOW (BYTE ADDRESS 30000Ch)[1]

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/C-1
—	—	—	—	—	—	—	EBTR[2,3]
bit 7							bit 0

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

bit 7-1   **Unimplemented:** Read as '0'

bit 0   **EBTR:** Table Read Protection bit[2,3]

    1= Internal program memory block is not protected from table reads executed from external memory block
    0= Internal program memory block is protected from table reads executed from external memory block

Note 1: Unimplemented on PIC18F6310/6410 devices; maintain the bit set.

    2: Valid for the entire internal program memory block in Extended Microcontroller mode and for only the boot block (0000h to 07FFh) in Microcontroller with Boot Block mode. This bit has no effect in Microcontroller and Microprocessor modes.

    3: It is recommended to enable the CP bit to protect the block from external read operations.

24.3 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 INTRC 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 a 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, IRCF<2:0>, immediately after Reset. For wake-ups from Sleep, the INTOSC or postscaler clock sources can be selected by setting the IRCF<2:0> 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.

24.3.1 SPECIAL CONSIDERATIONS FOR USING TWO-SPEED START-UP

While using the INTRC oscillator in Two-Speed Start-up, the device still obeys the normal command sequences for entering power-managed modes, including serial `SLEEP` instructions (refer to **Section 4.1.2 "Entering Power-Managed Modes"**). In practice, this means that user code can change the SCS<1:0> bits setting 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 24-2: TIMING TRANSITION FOR TWO-SPEED START-UP (INTOSC TO HSPLL)

Note 1: T_{OST} = 1024 T_{OSC}; T_{PLL} = 2 ms (approx). These intervals are not shown to scale.
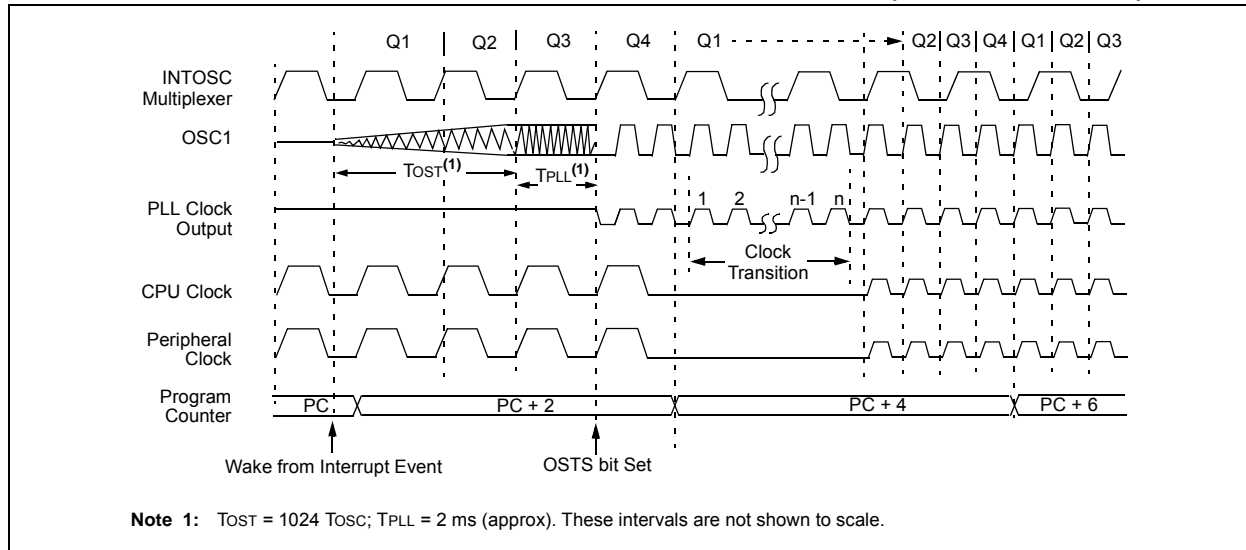
24.6 ID Locations

Eight memory locations (200000h-200007h) are designated as ID locations, where the user can store checksum or other code identification numbers. These locations are readable during normal execution through the `TBLRD` instruction. During program/verify, these locations are readable and writable. The ID locations can be read when the device is code-protected.

24.7 In-Circuit Serial Programming

PIC18F6310/6410/8310/8410 microcontrollers can be serially programmed while in the end application circuit. This is simply done with two lines for clock and data, and three other lines for power, ground and the programming voltage. This allows customers to manufacture boards with unprogrammed devices and then program the microcontroller just before shipping the product. This also allows the most recent firmware or a custom firmware to be programmed.

24.8 In-Circuit Debugger

When the $\overline{\text{DEBUG}}$ Configuration bit is programmed to a '0', the In-Circuit Debugger functionality is enabled. This function allows simple debugging functions when used with MPLAB® IDE. When the microcontroller has this feature enabled, some resources are not available for general use. Table 24-4 shows which resources are required by the background debugger.

TABLE 24-4: DEBUGGER RESOURCES

I/O Pins:	RB6, RB7
Stack:	2 levels
Program Memory:	<1 Kbyte
Data Memory:	<16 bytes

To use the In-Circuit Debugger function of the microcontroller, the design must implement In-Circuit Serial Programming connections to $\overline{\text{MCLR}}$/V$_{PP}$, V$_{DD}$, V$_{SS}$, RB7 and RB6. This will interface to the In-Circuit Debugger module available from Microchip or one of the third party development tool companies.

TABLE 25-1: OPCODE FIELD DESCRIPTIONS

Field	Description
a	RAM access bit a = 0: RAM location in Access RAM (BSR register is ignored) a = 1: RAM bank is specified by BSR register
bbb	Bit address within an 8-bit file register (0 to 7).
BSR	Bank Select Register. Used to select the current RAM bank.
C, DC, Z, OV, N	ALU Status bits: **C**arry, **D**igit **C**arry, **Z**ero, **Ov**erflow, **N**egative.
d	Destination select bit d = 0: store result in WREG d = 1: store result in file register f.
dest	Destination: either the WREG register or the specified register file location.
f	8-bit register file address (00h to FFh), or 2-bit FSR designator (0h to 3h).
f_s	12-bit register file address (000h to FFFh). This is the source address.
f_d	12-bit register file address (000h to FFFh). This is the destination address.
GIE	Global interrupt enable bit.
k	Literal field, constant data or label (may be either an 8-bit, 12-bit or a 20-bit value).
label	Label name.
mm	The mode of the TBLPTR register for the table read and table write instructions. Only used with table read and table write instructions:
*	No change to register (such as TBLPTR with table reads and writes).
*+	Post-Increment register (such as TBLPTR with table reads and writes).
*-	Post-Decrement register (such as TBLPTR with table reads and writes).
+*	Pre-Increment register (such as TBLPTR with table reads and writes).
n	The relative address (2's complement number) for relative branch instructions, or the direct address for call/branch and return instructions.
PC	Program Counter.
PCL	Program Counter Low Byte.
PCH	Program Counter High Byte.
PCLATH	Program Counter High Byte Latch.
PCLATU	Program Counter Upper Byte Latch.
$\overline{PD}$	Power-Down bit.
PRODH	Product of Multiply high byte.
PRODL	Product of Multiply low byte.
s	Fast Call/Return mode select bit s = 0: do not update into/from shadow registers s = 1: certain registers loaded into/from shadow registers (Fast mode)
TBLPTR	21-bit Table Pointer (points to a program memory location).
TABLAT	8-bit Table Latch.
$\overline{TO}$	Time-out bit.
TOS	Top-of-Stack.
u	Unused or Unchanged.
WDT	Watchdog Timer.
WREG	Working register (accumulator).
x	Don't care ('0' or '1'). The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all Microchip software tools.
z_s	7-bit offset value for indirect addressing of register files (source).
z_d	7-bit offset value for indirect addressing of register files (destination).
{ }	Optional argument.
[text]	Indicates an indexed address.
(text)	The contents of text.
[expr]<n>	Specifies bit n of the register indicated by the pointer expr.
→	Assigned to.
< >	Register bit field.
∈	In the set of.
italics	User-defined term (font is Courier New).

BTFSC	**Bit Test File, Skip if Clear**

Syntax:	BTFSC f, b {,a}
Operands:	$0 \leq f \leq 255$ $0 \leq b \leq 7$ $a \in [0,1]$
Operation:	skip if (f<b>) = 0
Status Affected:	None
Encoding:	1011	bbba	ffff	ffff
Description:	If bit 'b' in register 'f' is '0', then the next instruction is skipped. If bit 'b' is '0', then the next instruction fetched during the current instruction execution is discarded and a NOP is executed instead, making this a two-cycle instruction. 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 ≤ 95 (5Fh). See **Section 25.2.3** for details.
Words:	1
Cycles:	1(2) **Note:** 3 cycles if skip and followed by a 2-word instruction.

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	No operation

If skip:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation

If skip and followed by 2-word instruction:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation
No operation	No operation	No operation	No operation

Example:

```
HERE    BTFSC   FLAG, 1, 0
FALSE   :
TRUE    :
```

Before Instruction
    PC   =   address (HERE)
After Instruction
    If FLAG<1> = 0;
       PC   = address (TRUE)
    If FLAG<1> = 1;
       PC   = address (FALSE)

BTFSS	**Bit Test File, Skip if Set**

Syntax:	BTFSS f, b {,a}
Operands:	$0 \leq f \leq 255$ $0 \leq b < 7$ $a \in [0,1]$
Operation:	skip if (f<b>) = 1
Status Affected:	None
Encoding:	1010	bbba	ffff	ffff
Description:	If bit 'b' in register 'f' is '1', then the next instruction is skipped. If bit 'b' is '1', then the next instruction fetched during the current instruction execution is discarded and a NOP is executed instead, making this a two-cycle instruction. 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 ≤ 95 (5Fh). See **Section 25.2.3** for details.
Words:	1
Cycles:	1(2) **Note:** 3 cycles if skip and followed by a 2-word instruction.

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	No operation

If skip:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation

If skip and followed by 2-word instruction:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation
No operation	No operation	No operation	No operation

Example:

```
HERE    BTFSS   FLAG, 1, 0
FALSE   :
TRUE    :
```

Before Instruction
    PC   =   address (HERE)
After Instruction
    If FLAG<1> = 0;
       PC   = address (FALSE)
    If FLAG<1> = 1;
       PC   = address (TRUE)

XORWF	Exclusive OR W with f

Syntax:	XORWF f {,d {,a}}
Operands:	$0 \leq f \leq 255$ $d \in [0,1]$ $a \in [0,1]$
Operation:	(W) .XOR. (f) $\rightarrow$ dest
Status Affected:	N, Z
Encoding:	0001 | 10da | ffff | ffff
Description:	Exclusive OR the contents of W with register 'f'. If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in the register '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 $\leq$ 95 (5Fh). See **Section 25.2.3** for details.
Words:	1
Cycles:	1

Q Cycle Activity:

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

Example: XORWF   REG, 1, 0

Before Instruction
  REG = AFh
  W   = B5h
After Instruction
  REG = 1Ah
  W   = B5h

25.2.3 BYTE-ORIENTED AND BIT-ORIENTED INSTRUCTIONS IN INDEXED LITERAL OFFSET MODE

> **Note:** Enabling the PIC18 instruction set extension may cause legacy applications to behave erratically or fail entirely.

In addition to eight new commands in the extended set, enabling the extended instruction set also enables Indexed Literal Offset addressing (**Section 6.5.1 "Indexed Addressing with Literal Offset"**). This has a significant impact on the way that many commands of the standard PIC18 instruction set are interpreted.

When the extended set is disabled, addresses embedded in opcodes are treated as literal memory locations: either as a location in the Access Bank (a = 0) or in a GPR bank designated by the BSR (a = 1). When the extended instruction set is enabled and a = 0, however, a file register argument of 5Fh or less is interpreted as an offset from the pointer value in FSR2 and not as a literal address. For practical purposes, this means that all instructions that use the Access RAM bit as an argument – that is, all byte-oriented and bit-oriented instructions, or almost half of the core PIC18 instructions – may behave differently when the extended instruction set is enabled.

When the content of FSR2 is 00h, the boundaries of the Access RAM are essentially remapped to their original values. This may be useful in creating backward compatible code. If this technique is used, it may be necessary to save the value of FSR2 and restore it when moving back and forth between C and assembly routines in order to preserve the Stack Pointer. Users must also keep in mind the syntax requirements of the extended instruction set (see **Section 25.2.3.1 "Extended Instruction Syntax with Standard PIC18 Commands"**).

Although the Indexed Literal Offset mode can be very useful for dynamic stack and pointer manipulation, it can also be very annoying if a simple arithmetic operation is carried out on the wrong register. Users who are accustomed to the PIC18 programming must keep in mind that, when the extended instruction set is enabled, register addresses of 5Fh or less are used for Indexed Literal Offset Addressing.

Representative examples of typical byte-oriented and bit-oriented instructions in the Indexed Literal Offset mode are provided on the following page to show how execution is affected. The operand conditions shown in the examples are applicable to all instructions of these types.

25.2.3.1 Extended Instruction Syntax with Standard PIC18 Commands

When the extended instruction set is enabled, the file register argument 'f' in the standard byte-oriented and bit-oriented commands is replaced with the literal offset value 'k'. As already noted, this occurs only when f is less than or equal to 5Fh. When an offset value is used, it must be indicated by square brackets ("[ ]"). As with the extended instructions, the use of brackets indicates to the compiler that the value is to be interpreted as an index or an offset. Omitting the brackets, or using a value greater than 5Fh within brackets, will generate an error in the MPASM Assembler.

If the index argument is properly bracketed for Indexed Literal Offset addressing, the Access RAM argument is never specified; it will automatically be assumed to be '0'. This is in contrast to standard operation (extended instruction set disabled), when 'a' is set on the basis of the target address. Declaring the Access RAM bit in this mode will also generate an error in the MPASM assembler.

The destination argument 'd' functions as before.

In the latest versions of the MPASM assembler, language support for the extended instruction set must be explicitly invoked. This is done with either the command line option /y, or the PE directive in the source listing.

25.2.4 CONSIDERATIONS WHEN ENABLING THE EXTENDED INSTRUCTION SET

It is important to note that the extensions to the instruction set may not be beneficial to all users. In particular, users who are not writing code that uses a software stack may not benefit from using the extensions to the instruction set.

Additionally, the Indexed Literal Offset Addressing mode may create issues with legacy applications written to PIC18 assembler. This is because instructions in the legacy code may attempt to address registers in the Access Bank below 5Fh. Since these addresses are interpreted as literal offsets to FSR2 when the instruction set extension is enabled, the application may read or write to the wrong data addresses.

When porting an application to the PIC18F6310/6410/8310/8410, it is very important to consider the type of code. A large, re-entrant application that is written in C and would benefit from efficient compilation will do well when using the instruction set extensions. Legacy applications that heavily use the Access Bank will most likely not benefit from using the extended instruction set.

26.11 PICkit 2 Development Programmer/Debugger and PICkit 2 Debug Express

The PICkit™ 2 Development Programmer/Debugger is a low-cost development tool with an easy to use inter-face for programming and debugging Microchip's Flash families of microcontrollers. The full featured Windows® programming interface supports baseline (PIC10F, PIC12F5xx, PIC16F5xx), midrange (PIC12F6xx, PIC16F), PIC18F, PIC24, dsPIC30, dsPIC33, and PIC32 families of 8-bit, 16-bit, and 32-bit microcontrollers, and many Microchip Serial EEPROM products. With Microchip's powerful MPLAB Integrated Development Environment (IDE) the PICkit™ 2 enables in-circuit debugging on most PIC® microcon-trollers. In-Circuit-Debugging runs, halts and single steps the program while the PIC microcontroller is embedded in the application. When halted at a break-point, the file registers can be examined and modified.

The PICkit 2 Debug Express include the PICkit 2, demo board and microcontroller, hookup cables and CDROM with user's guide, lessons, tutorial, compiler and MPLAB IDE software.

26.12 MPLAB PM3 Device Programmer

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

26.13 Demonstration/Development Boards, Evaluation Kits, and Starter Kits

A wide variety of demonstration, development and evaluation boards for various PIC MCUs and dsPIC DSCs allows quick application development on fully func-tional systems. Most boards include prototyping areas for adding custom circuitry and provide application firmware and source code for examination and modification.

The boards support a variety of features, including LEDs, temperature sensors, switches, speakers, RS-232 interfaces, LCD displays, potentiometers and additional EEPROM memory.

The demonstration and development boards can be used in teaching environments, for prototyping custom circuits and for learning about various microcontroller applications.

In addition to the PICDEM™ and dsPICDEM™ demon-stration/development board series of circuits, Microchip has a line of evaluation kits and demonstration software for analog filter design, KEELOQ® security ICs, CAN, IrDA®, PowerSmart battery management, SEEVAL® evaluation system, Sigma-Delta ADC, flow rate sensing, plus many more.

Also available are starter kits that contain everything needed to experience the specified device. This usually includes a single application and debug capability, all on one board.

Check the Microchip web page (www.microchip.com) for the complete list of demonstration, development and evaluation kits.

27.2   DC Characteristics: Power-Down and Supply Current
PIC18F6310/6410/8310/8410 (Industrial, Extended)
PIC18LF6310/6410/8310/8410 (Industrial) (Continued)

PIC18LF6310/6410/8310/8410 (Industrial)	Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial
PIC18F6310/6410/8310/8410 (Industrial, Extended)	Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended

Param No.	Device	Typ	Max	Units	Conditions	
	Supply Current (IDD)[(2,3)]					
	PIC18LFX310/X410	59	117	μA	-40°C	
		59	108	μA	+25°C	VDD = 2.0V
		63	104	μA	+85°C	
	PIC18LFX310/X410	108	243	μA	-40°C	
		108	225	μA	+25°C	VDD = 3.0V
		117	216	μA	+85°C	FOSC = 1 MHz (**PRI_IDLE** mode, EC oscillator)
	All devices	270	432	μA	-40°C	
		216	405	μA	+25°C	VDD = 5.0V
		270	387	μA	+85°C	
		300	430	μA	+125°C	
	PIC18LFX310/X410	234	428	μA	-40°C	
		230	405	μA	+25°C	VDD = 2.0V
		243	387	μA	+85°C	
	PIC18LFX310/X410	378	810	μA	-40°C	
		387	765	μA	+25°C	VDD = 3.0V
		405	729	μA	+85°C	FOSC = 4 MHz (**PRI_IDLE** mode, EC oscillator)
	All devices	0.8	1.35	mA	-40°C	
		0.8	1.26	mA	+25°C	VDD = 5.0V
		0.8	1.17	mA	+85°C	
		1	1.4	mA	+125°C	
	All devices	5.4	14.4	mA	-40°C	
		5.6	14.4	mA	+25°C	VDD = 4.2 V
		5.9	14.4	mA	+85°C	
	All devices	7.3	16.2	mA	-40°C	FOSC = 40 MHz (**PRI_IDLE** mode, EC oscillator)
		8.2	16.2	mA	+25°C	
		7.5	16.2	mA	+85°C	VDD = 5.0V
		19	18	mA	+125°C	

Legend: Shading of rows is to assist in readability of the table.

Note 1: 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 VDD or VSS, and all features that add delta current disabled (such as WDT, Timer1 Oscillator, BOR, etc.).

2: The supply current is mainly a function of operating voltage, frequency and mode. Other factors, such as I/O pin loading and switching rate, oscillator type and circuit, internal code execution pattern and temperature, also have an impact on the current consumption. The test conditions for all IDD measurements in active operation mode are: OSC1 = external square wave, from rail-to-rail; all I/O pins are tri-stated, pulled to VDD or VSS; MCLR = VDD; WDT is enabled/disabled as specified.

3: When operation below -10°C is expected, use the T1OSC High-Power mode, where LPT1OSC (CONFIG3H<2>) = 0. When operation will always be above -10°C, then the low-power Timer1 oscillator may be selected.

4: BOR and HLVD enable internal band gap reference. With both modules enabled, current consumption will be less than the sum of both specifications.