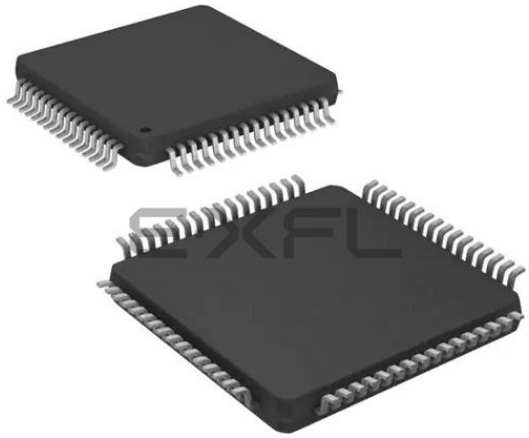
# Microchip Technology - PIC18F6490T-I/PT Datasheet



**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 | I²C, SPI, UART/USART |
| Peripherals | Brown-out Detect/Reset, HLVD, LCD, POR, PWM, WDT |
| Number of I/O | 50 |
| 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/pic18f6490t-i-pt |

## 2.4 RC Oscillator

For timing insensitive applications, the “RC” and “RCIO” device options offer additional cost savings. The actual oscillator frequency is a function of several factors:

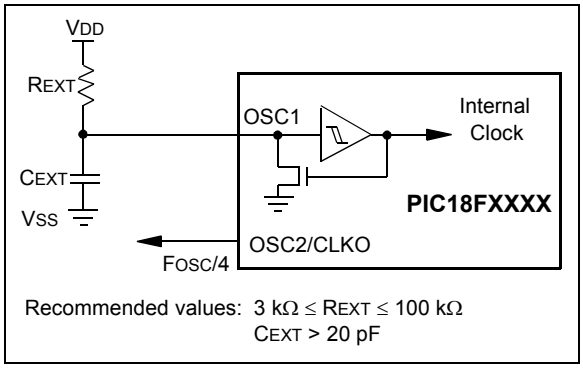
- Supply voltage
- Values of the external resistor (REXT) and capacitor (CEXT)
- Operating temperature

Given the same device, operating voltage and temperature and component values, there will also be unit-to-unit frequency variations. These are due to factors such as:

- Normal manufacturing variation
- Difference in lead frame capacitance between package types (especially for low CEXT values)
- Variations within the tolerance of limits of REXT and CEXT

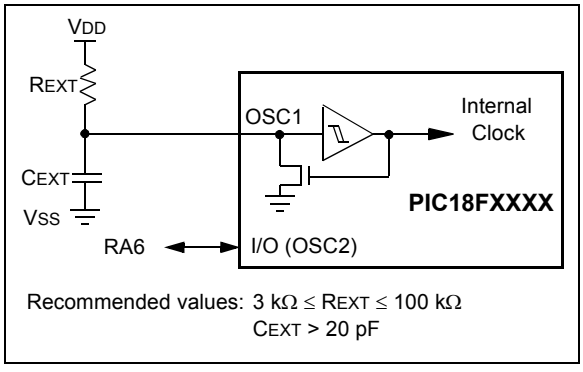
In the RC Oscillator mode, the oscillator frequency divided by 4 is available on the OSC2 pin. This signal may be used for test purposes or to synchronize other logic. Figure 2-5 shows how the R/C combination is connected.

**FIGURE 2-5: RC OSCILLATOR MODE**




The RCIO Oscillator mode (Figure 2-6) functions like the RC mode, except that the OSC2 pin becomes an additional general purpose I/O pin. The I/O pin becomes bit 6 of PORTA (RA6).

**FIGURE 2-6: RCIO OSCILLATOR MODE**




## 2.5 PLL Frequency Multiplier

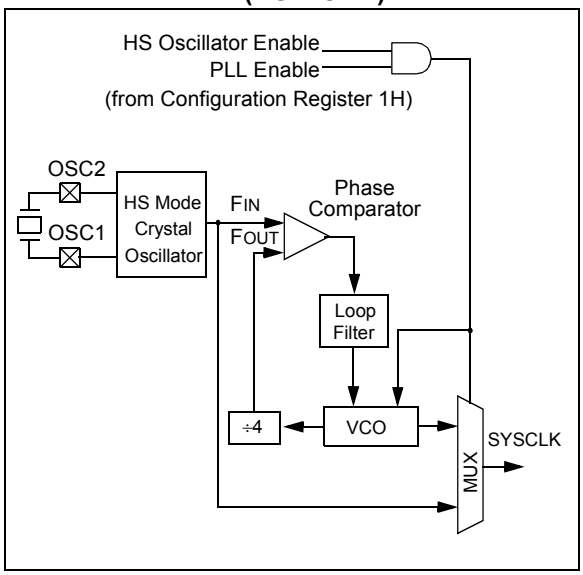
A Phase Locked Loop (PLL) circuit is provided as an option for users who want to use a lower frequency oscillator circuit, or to clock the device up to its highest rated frequency from a crystal oscillator. This may be useful for customers who are concerned with EMI due to high-frequency crystals, or users who require higher clock speeds from an internal oscillator.

### 2.5.1 HSPLL OSCILLATOR MODE

The HSPLL mode makes use of the HS mode oscillator for frequencies up to 10 MHz. A PLL then multiplies the oscillator output frequency by 4 to produce an internal clock frequency up to 40 MHz.

The PLL is only available to the crystal oscillator when the FOSC3:FOSC0 Configuration bits are programmed for HSPLL mode (= 0110).

**FIGURE 2-7: PLL BLOCK DIAGRAM (HS MODE)**




### 2.5.2 PLL AND INTOSC

The PLL is also available to the internal oscillator block in selected oscillator modes. In this configuration, the PLL is enabled in software and generates a clock output of up to 32 MHz. The operation of INTOSC with the PLL is described in **Section 2.6.4 “PLL in INTOSC Modes”**.

## 2.6 Internal Oscillator Block

The PIC18F6390/6490/8390/8490 devices include an internal oscillator block, which generates two different clock signals; either can be used as the microcontroller's clock source. This may eliminate the need for external oscillator circuits on the OSC1 and/or OSC2 pins.

The main output (INTOSC) is an 8 MHz clock source, which can be used to directly drive the device clock. It also drives a postscaler, which can provide a range of clock frequencies from 31 kHz to 4 MHz. The INTOSC output is enabled when a clock frequency from 125 kHz to 8 MHz is selected.

The other clock source is the internal RC oscillator (INTRC), which provides a nominal 31 kHz output. INTRC is enabled if it is selected as the device clock source; it is also enabled automatically when any of the following are enabled:

- Power-up Timer
- Fail-Safe Clock Monitor
- Watchdog Timer
- Two-Speed Start-up
- LCD with INTRC as its clock source

These features are discussed in greater detail in **Section 23.0 "Special Features of the CPU"**.

The clock source frequency (INTOSC direct, INTRC direct or INTOSC postscaler) is selected by configuring the IRCF bits of the OSCCON register (Register 2-2).

### 2.6.1 INTIO MODES

Using the internal oscillator as the clock source eliminates the need for up to two external oscillator pins, which can then be used for digital I/O. Two distinct configurations are available:

- In INTIO1 mode, the OSC2 pin outputs FOSC/4, while OSC1 functions as RA7 for digital input and output.
- In INTIO2 mode, OSC1 functions as RA7 and OSC2 functions as RA6, both for digital input and output.

### 2.6.2 INTOSC OUTPUT FREQUENCY

The internal oscillator block is calibrated at the factory to produce an INTOSC output frequency of 8.0 MHz.

The INTRC oscillator operates independently of the INTOSC source. Any changes in INTOSC across voltage and temperature are not necessarily reflected by changes in INTRC and vice versa.

### 2.6.3 OSCTUNE REGISTER

The internal oscillator's output has been calibrated at the factory, but can be adjusted in the user's application. This is done by writing to the OSCTUNE register (Register 2-1). The tuning sensitivity is constant throughout the tuning range.

When the OSCTUNE register is modified, the INTOSC and INTRC frequencies will begin shifting to the new frequency. The INTRC clock will reach the new frequency within 8 clock cycles (approximately 8 * 32 μs = 256 μs). The INTOSC clock will stabilize within 1 ms. Code execution continues during this shift. There is no indication that the shift has occurred.

The OSCTUNE register also implements the INTSRC and PLLEN bits, which control certain features of the internal oscillator block. The INTSRC bit allows users to select which internal oscillator provides the clock source when the 31 kHz frequency option is selected. This is covered in greater detail in **Section 2.7.1 "Oscillator Control Register"**.

The PLLEN bit controls the operation of the frequency multiplier, PLL, in internal oscillator modes.

### 2.6.4 PLL IN INTOSC MODES

The 4x frequency multiplier can be used with the internal oscillator block to produce faster device clock speeds than are normally possible with an internal oscillator. When enabled, the PLL produces a clock speed of up to 32 MHz.

Unlike HSPLL mode, the PLL is controlled through software. The control bit, PLLEN (OSCTUNE<6>), is used to enable or disable its operation.

The PLL is available when the device is configured to use the internal oscillator block as its primary clock source (FOSC3:FOSC0 = 1001 or 1000). Additionally, the PLL will only function when the selected output frequency is either 4 MHz or 8 MHz (OSCCON<6:4> = 111 or 110). If both of these conditions are not met, the PLL is disabled.

The PLLEN control bit is only functional in those internal oscillator modes where the PLL is available. In all other modes, it is forced to '0' and is effectively unavailable.

### 2.6.5 INTOSC FREQUENCY DRIFT

The factory calibrates the internal oscillator block output (INTOSC) for 8 MHz. However, this frequency may drift as VDD or temperature changes, which can affect the controller operation in a variety of ways. It is possible to adjust the INTOSC frequency by modifying the value in the OSTUNE register. This has no effect on the INTRC clock source frequency.

Tuning the INTOSC source requires knowing when to make the adjustment, in which direction it should be made and in some cases, how large a change is needed. Three compensation techniques are discussed in **Section 2.6.5.1 "Compensating with the AUSART"**, **Section 2.6.5.2 "Compensating with the Timers"** and **Section 2.6.5.3 "Compensating with the Timers"**, but other techniques may be used.

## 2.7 Clock Sources and Oscillator Switching

Like previous PIC18 devices, the PIC18F6390/6490/8390/8490 family includes a feature that allows the device clock source to be switched from the main oscillator to an alternate low-frequency clock source. PIC18F6390/6490/8390/8490 devices offer two alternate clock sources. When an alternate clock source is enabled, the various power-managed operating modes are available.

Essentially, there are three clock sources for these devices:

- Primary oscillators
- Secondary oscillators
- Internal oscillator block

The **primary oscillators** include the External Crystal and Resonator modes, the External RC modes, the External Clock modes and the internal oscillator block. The particular mode is defined by the FOSC3:FOSC0 Configuration bits. The details of these modes are covered earlier in this chapter.

The **secondary oscillators** are those external sources not connected to the OSC1 or OSC2 pins. These sources may continue to operate even after the controller is placed in a power-managed mode.

PIC18F6390/6490/8390/8490 devices offer the Timer1 oscillator as a secondary oscillator. This oscillator, in all power-managed modes, is often the time base for functions such as a Real-Time Clock.

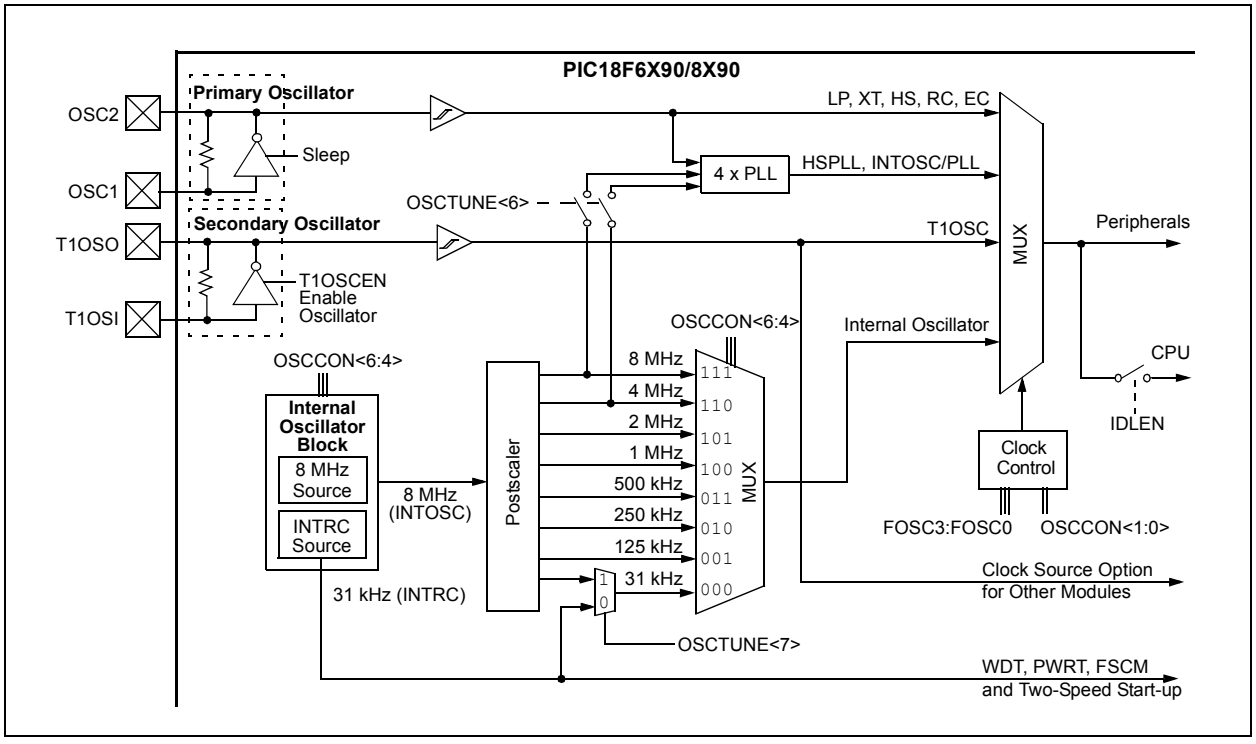
Most often, a 32.768 kHz watch crystal is connected between the RC0/T1OSO/T13CKI and RC1/T1OSI pins. Like the LP Oscillator mode circuit, loading capacitors are also connected from each pin to ground.

The Timer1 oscillator is discussed in greater detail in **Section 11.3 "Timer1 Oscillator"**.

In addition to being a primary clock source, the **internal oscillator block** is available as a power-managed mode clock source. The INTRC source is also used as the clock source for several special features, such as the WDT and Fail-Safe Clock Monitor.

The clock sources for the PIC18F6390/6490/8390/8490 devices are shown in Figure 2-8. See **Section 23.0 "Special Features of the CPU"** for Configuration register details.

**FIGURE 2-8: PIC18F6390/6490/8390/8490 CLOCK DIAGRAM**



**REGISTER 4-1: RCON: RESET CONTROL REGISTER**

| R/W-0 | R/W-1(1) | U-0 | R/W-1 | R-1 | R-1 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| IPEN | SBOREN | — | $\overline{RI}$ | $\overline{TO}$ | $\overline{PD}$ | $\overline{POR}$ | $\overline{BOR}$ |
| 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 **IPEN:** Interrupt Priority Enable bit
1 = Enable priority levels on interrupts
0 = Disable priority levels on interrupts (PIC16CXXX Compatibility mode)

bit 6 **SBOREN:** BOR Software Enable bit(1)
If BOREN1:BOREN0 = 01:
1 = BOR is enabled
0 = BOR is disabled
If BOREN1:BOREN0 = 00, 10 or 11:
Bit is disabled and read as '0'.

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

bit 4 **$\overline{RI}$:** RESET Instruction Flag bit
1 = The RESET instruction was not executed (set by firmware only)
0 = The RESET instruction was executed causing a device Reset (must be set in software after a Brown-out Reset occurs)

bit 3 **$\overline{TO}$:** Watchdog Time-out Flag bit
1 = Set by power-up, CLRWDT instruction or SLEEP instruction
0 = A WDT time-out occurred

bit 2 **$\overline{PD}$:** Power-Down Detection Flag bit
1 = Set by power-up or by the CLRWDT instruction
0 = Set by execution of the SLEEP instruction

bit 1 **$\overline{POR}$:** Power-on Reset Status bit
1 = A Power-on Reset has not occurred (set by firmware only)
0 = A Power-on Reset occurred (must be set in software after a Power-on Reset occurs)

bit 0 **$\overline{BOR}$:** Brown-out Reset Status bit
1 = A Brown-out Reset has not occurred (set by firmware only)
0 = A Brown-out Reset occurred (must be set in software after a Brown-out Reset occurs)

**Note 1:** If SBOREN is enabled, its Reset state is '1'; otherwise, it is '0'.

**Note 1:** It is recommended that the $\overline{POR}$ bit be set after a Power-on Reset has been detected, so that subsequent Power-on Resets may be detected.

**2:** Brown-out Reset is said to have occurred when $\overline{BOR}$ is '0' and $\overline{POR}$ is '1' (assuming that $\overline{POR}$ was set to '1' by software immediately after a Power-on Reset).

**TABLE 4-4: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)**

| Register | Applicable Devices | | Power-on Reset, Brown-out Reset | $\overline{\text{MCLR}}$ Resets WDT Reset RESET Instruction Stack Resets | Wake-up via WDT or Interrupt |
|---|---|---|---|---|---|
| LCDSE5 | 6X90 | 8X90 | 0000 0000 | 0000 0000(6) | uuuu uuuu |
| LCDSE4 | 6X90 | 8X90 | 0000 0000 | 0000 0000(6) | uuuu uuuu |
| LCDSE3 | 6X90 | 8X90 | 0000 0000 | 0000 0000(6) | uuuu uuuu |
| LCDSE2 | 6X90 | 8X90 | 0000 0000 | 0000 0000(6) | uuuu uuuu |
| LCDSE1 | 6X90 | 8X90 | 0000 0000 | 0000 0000(6) | uuuu uuuu |
| LCDSE0 | 6X90 | 8X90 | 0000 0000 | 0000 0000(6) | uuuu uuuu |
| LCDCON | 6X90 | 8X90 | 000- 0000 | 000- 0000 | uuu- uuuu |
| LCDPS | 6X90 | 8X90 | 0000 0000 | 0000 0000 | uuuu uuuu |

**Legend:** u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition. Shaded cells indicate conditions do not apply for the designated device.

**Note 1:** One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).

**2:** When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the PC is loaded with the interrupt vector (0008h or 0018h).

**3:** When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the TOSU, TOSH and TOSL are updated with the current value of the PC. The STKPTR is modified to point to the next location in the hardware stack.

**4:** See Table 4-3 for Reset value for specific condition.

**5:** Bits 6 and 7 of PORTA, LATA and TRISA are enabled depending on the oscillator mode selected. When not enabled as PORTA pins, they are disabled and read '0'.

**6:** These registers are cleared on POR and unchanged on BOR.

**TABLE 5-2: PIC18F6390/6490/8390/8490 REGISTER FILE SUMMARY (CONTINUED)**

| File Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOR | Details on page: |
|---|---|---|---|---|---|---|---|---|---|---|
| SPBRGH1 | EUSART1 Baud Rate Generator Register High Byte | | | | | | | | 0000 0000 | 62, 201 |
| BAUDCON1 | ABDOVF | RCIDL | — | SCKP | BRG16 | — | WUE | ABDEN | 01-0 0-00 | 62, 200 |
| LCDDATA23(6) | S47C3 | S46C3 | S45C3 | S44C3 | S43C3 | S42C3 | S41C3 | S40C3 | xxxx xxxx | 63, 261 |
| LCDDATA22(6) | S39C3 | S38C3 | S37C3 | S36C3 | S35C3 | S34C3 | S33C3 | S32C3 | xxxx xxxx | 63, 261 |
| LCDDATA21 | S31C3 | S30C3 | S29C3 | S28C3 | S27C3 | S26C3 | S25C3 | S24C3 | xxxx xxxx | 63, 261 |
| LCDDATA20 | S23C3 | S22C3 | S21C3 | S20C3 | S19C3 | S18C3 | S17C3 | S16C3 | xxxx xxxx | 63, 261 |
| LCDDATA19 | S15C3 | S14C3 | S13C3 | S12C3 | S11C3 | S10C3 | S09C3 | S08C3 | xxxx xxxx | 63, 261 |
| LCDDATA18 | S07C3 | S06C3 | S05C3 | S04C3 | S03C3 | S02C3 | S01C3 | S00C3 | xxxx xxxx | 63, 261 |
| LCDDATA17(6) | S47C2 | S46C2 | S45C2 | S44C2 | S43C2 | S42C2 | S41C2 | S40C2 | xxxx xxxx | 63, 261 |
| LCDDATA16(6) | S39C2 | S38C2 | S37C2 | S36C2 | S35C2 | S34C2 | S33C2 | S32C2 | xxxx xxxx | 63, 261 |
| LCDDATA15 | S31C2 | S30C2 | S29C2 | S28C2 | S27C2 | S26C2 | S25C2 | S24C2 | xxxx xxxx | 63, 261 |
| LCDDATA14 | S23C2 | S22C2 | S21C2 | S20C2 | S19C2 | S18C2 | S17C2 | S16C2 | xxxx xxxx | 63, 261 |
| LCDDATA13 | S15C2 | S14C2 | S13C2 | S12C2 | S11C2 | S10C2 | S09C2 | S08C2 | xxxx xxxx | 63, 261 |
| LCDDATA12 | S07C2 | S06C2 | S05C2 | S04C2 | S03C2 | S02C2 | S01C2 | S00C2 | xxxx xxxx | 63, 261 |
| LCDDATA11(6) | S47C1 | S46C1 | S45C1 | S44C1 | S43C1 | S42C1 | S41C1 | S40C1 | xxxx xxxx | 63, 261 |
| SPBRG2 | AUSART2 Baud Rate Generator Register | | | | | | | | 0000 0000 | 63, 220 |
| RCREG2 | AUSART2 Receive Register | | | | | | | | 0000 0000 | 63, 224 |
| TXREG2 | AUSART2 Transmit Register | | | | | | | | 0000 0000 | 63, 222 |
| TXSTA2 | CSRC | TX9 | TXEN | SYNC | — | BRGH | TRMT | TX9D | 0000 -010 | 63, 218 |
| RCSTA2 | SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D | 0000 000x | 63, 219 |
| LCDDATA10(6) | S39C1 | S38C1 | S37C1 | S36C1 | S35C1 | S34C1 | S33C1 | S32C1 | xxxx xxxx | 63, 261 |
| LCDDATA9 | S31C1 | S30C1 | S29C1 | S28C1 | S27C1 | S26C1 | S25C1 | S24C1 | xxxx xxxx | 63, 261 |
| LCDDATA8 | S23C1 | S22C1 | S21C1 | S20C1 | S19C1 | S18C1 | S17C1 | S16C1 | xxxx xxxx | 63, 261 |
| LCDDATA7 | S15C1 | S14C1 | S13C1 | S12C1 | S11C1 | S10C1 | S09C1 | S08C1 | xxxx xxxx | 63, 261 |
| LCDDATA6 | S07C1 | S06C1 | S05C1 | S04C1 | S03C1 | S02C1 | S01C1 | S00C1 | xxxx xxxx | 63, 261 |
| LCDDATA5(6) | S47C0 | S46C0 | S45C0 | S44C0 | S43C0 | S42C0 | S41C0 | S40C0 | xxxx xxxx | 63, 261 |
| LCDDATA4(6) | S39C0 | S38C0 | S37C0 | S36C0 | S35C0 | S34C0 | S33C0 | S32C0 | xxxx xxxx | 63, 261 |
| LCDDATA3 | S31C0 | S30C0 | S29C0 | S28C0 | S27C0 | S26C0 | S25C0 | S24C0 | xxxx xxxx | 63, 261 |
| LCDDATA2 | S23C0 | S22C0 | S21C0 | S20C0 | S19C0 | S18C0 | S17C0 | S16C0 | xxxx xxxx | 63, 261 |
| LCDDATA1 | S15C0 | S14C0 | S13C0 | S12C0 | S11C0 | S10C0 | S09C0 | S08C0 | xxxx xxxx | 63, 261 |
| LCDDATA0 | S07C0 | S06C0 | S05C0 | S04C0 | S03C0 | S02C0 | S01C0 | S00C0 | xxxx xxxx | 63, 261 |
| LCDSE5(2) | SE47 | SE46 | SE45 | SE44 | SE43 | SE42 | SE41 | SE40 | 0000 0000 | 64, 261 |
| LCDSE4(2) | SE39 | SE38 | SE37 | SE36 | SE35 | SE34 | SE33 | SE32 | 0000 0000 | 64, 260 |
| LCDSE3 | SE31 | SE30 | SE29 | SE28 | SE27 | SE26 | SE25 | SE24 | 0000 0000 | 64, 260 |
| LCDSE2 | SE23 | SE22 | SE21 | SE20 | SE19 | SE18 | SE17 | SE16 | 0000 0000 | 64, 260 |
| LCDSE1 | SE15 | SE14 | SE13 | SE12 | SE11 | SE10 | SE9 | SE8 | 0000 0000 | 64, 260 |
| LCDSE0 | SE7 | SE6 | SE5 | SE4 | SE3 | SE2 | SE1 | SE0 | 0000 0000 | 64, 260 |
| LCDCON | LCDEN | SLPEN | WERR | — | CS1 | CS0 | LMUX1 | LMUX0 | 000- 0000 | 64, 258 |
| LCDPS | WFT | BIASMD | LCDA | WA | LP3 | LP2 | LP1 | LP0 | 0000 0000 | 64, 259 |

**Legend:** x = unknown, u = unchanged, – = unimplemented, q = value depends on condition

**Note 1:** The SBOREN bit is only available when the BOREN1:BOREN0 Configuration bits = 01; otherwise it is disabled and reads as '0'. See **Section 4.4 "Brown-out Reset (BOR)"**.

**2:** These registers and/or bits are not implemented on 64-pin devices; read as '0'.

**3:** The PLLEN bit is only available in specific oscillator configurations; otherwise, it is disabled and reads as '0'. See **Section 2.6.4 "PLL in INTOSC Modes"**.

**4:** The RG5 bit is only available when Master Clear is disabled (MCLRE Configuration bit = 0); otherwise, RG5 reads as '0'. This bit is read-only.

**5:** RA6/RA7 and their associated latch and direction bits are individually configured as port pins based on various primary oscillator modes. When disabled, these bits read as '0'.

**6:** These registers are implemented but unused in 64-pin devices and may be used as general-purpose data RAM if required.

## 6.2 Control Registers

Two control registers are used in conjunction with the TBLRD instruction: the TABLAT register and the TBLPTR register set.

### 6.2.1 TABLE LATCH REGISTER (TABLAT)

The Table Latch (TABLAT) is an 8-bit register mapped into the SFR space. The Table Latch register is used to hold 8-bit data during data transfers between program memory and data RAM.

### 6.2.2 TABLE POINTER REGISTER (TBLPTR)

The Table Pointer register (TBLPTR) addresses a byte within the program memory. It is comprised of three SFR registers: Table Pointer Upper Byte, Table Pointer High Byte and Table Pointer Low Byte (TBLPTRU:TBLPTRH:TBLPTRL). Only the lower six bits of TBLPTRU are used with TBLPTRH and TBLPTRL to form a 22-bit wide pointer.

The contents of TBLPTR indicates a location in program memory space. The low-order 21 bits allow the device to address the full 2 Mbytes of program memory space. The 22nd bit allows access to the configuration space, including the device ID, user ID locations and the Configuration bits.

The TBLPTR register set is updated when executing a TBLRD in one of four ways, based on the instruction's arguments. These are detailed in Table 6-1. These operations on the TBLPTR only affect the low-order 21 bits.

When a TBLRD is executed, all 22 bits of the TBLPTR determine which byte is read from program memory into TABLAT.

**TABLE 6-1: TABLE POINTER OPERATIONS WITH TBLRD INSTRUCTIONS**

| Example | Operation on Table Pointer |
|---|---|
| TBLRD* | TBLPTR is not modified |
| TBLRD*+ | TBLPTR is incremented after the read |
| TBLRD*- | TBLPTR is decremented after the read |
| TBLRD+* | TBLPTR is incremented before the read |

## 6.3 Reading the Flash Program Memory

The TBLRD instruction is used to retrieve data from program memory and places it into data RAM. Table reads from program memory are performed one byte at a time.

TBLPTR points to a byte address in program space. Executing TBLRD places the byte pointed to into TABLAT. In addition, TBLPTR can be modified automatically for the next table read operation.

The internal program memory is typically organized by words. The Least Significant bit of the address selects between the high and low bytes of the word. Figure 6-2 shows the interface between the internal program memory and the TABLAT.

A typical method for reading data from program memory is shown in Example 6-1.

**FIGURE 6-2: READS FROM FLASH PROGRAM MEMORY**

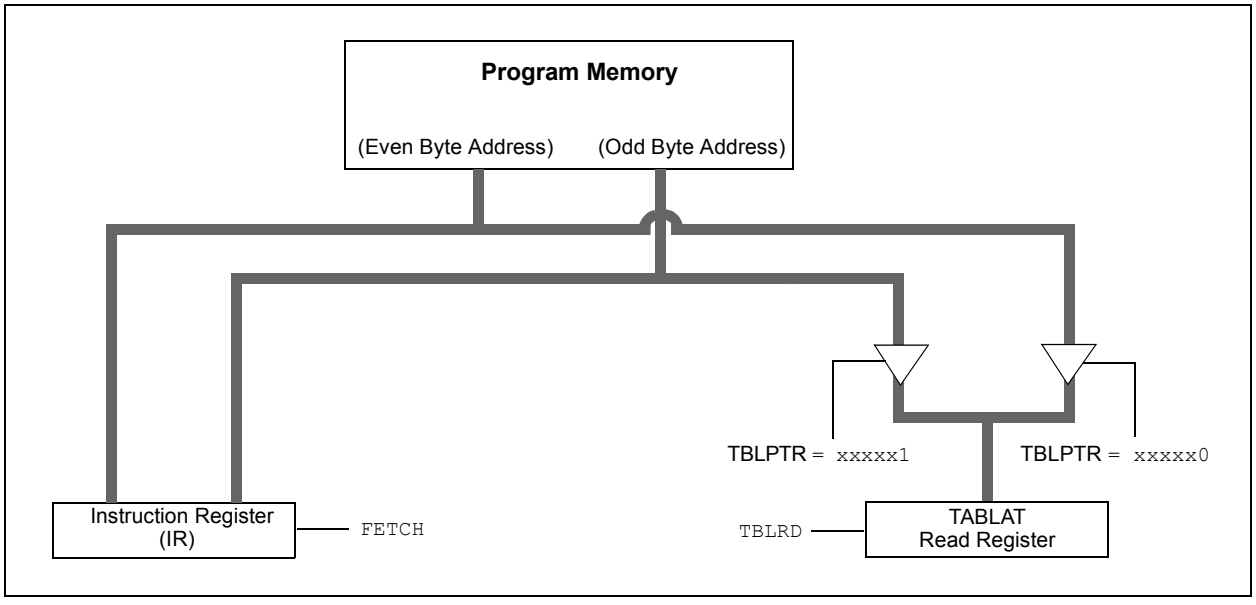
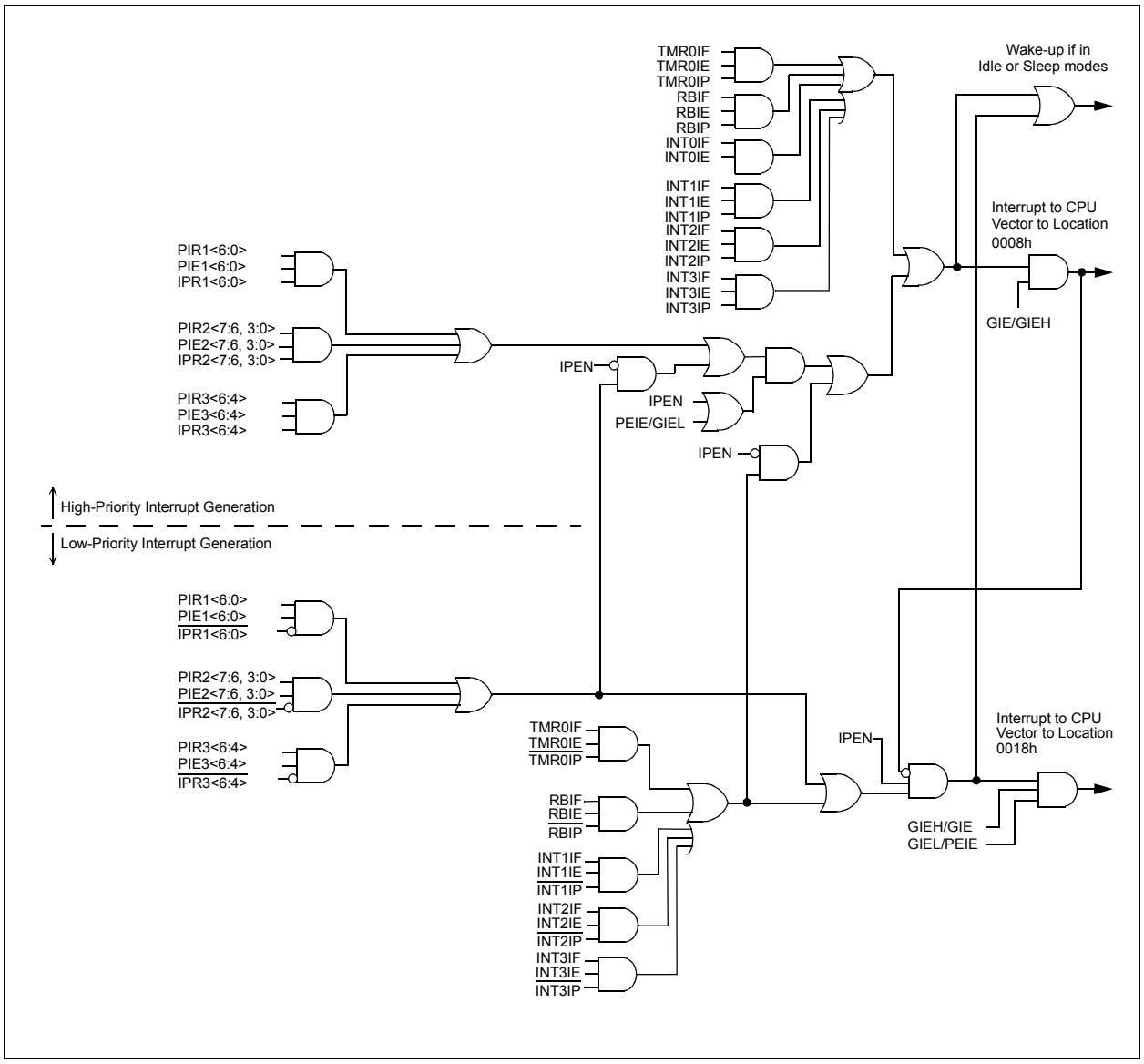


**FIGURE 8-1: PIC18F6X90/8X90 INTERRUPT LOGIC**



## 9.2 PORTB, TRISB and LATB Registers

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

The Data Latch register (LATB) is also memory mapped. Read-modify-write operations on the LATB register read and write the latched output value for PORTB.

**EXAMPLE 9-2: INITIALIZING PORTB**

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

Each of the PORTB pins has a weak internal pull-up. A single control bit can turn on all the pull-ups. This is performed by clearing bit $\overline{\text{RBPU}}$ (INTCON2<7>). The weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on a Power-on Reset.

Four of the PORTB pins (RB7:RB4) have an interrupt-on-change feature. Only pins configured as inputs can cause this interrupt to occur (i.e., any RB7:RB4 pin configured as an output is excluded from the interrupt-on-change comparison). The input pins (of RB7:RB4) are compared with the old value latched on the last read of PORTB. The “mismatch” outputs of RB7:RB4 are ORed together to generate the RB Port Change Interrupt with Flag bit, RBIF (INTCON<0>).

This interrupt can wake the device from power-managed modes. The user, in the Interrupt Service Routine, can clear the interrupt in the following manner:

a) Any read or write of PORTB (except with the MOVFF (ANY), PORTB instruction). This will end the mismatch condition.
b) Clear flag bit, RBIF.

A mismatch condition will continue to set flag bit, RBIF. Reading PORTB will end the mismatch condition and allow flag bit RBIF to be cleared.

The interrupt-on-change feature is recommended for wake-up on key depression operation and operations where PORTB is only used for the interrupt-on-change feature. Polling of PORTB is not recommended while using the interrupt-on-change feature.

RB4:RB1 are also multiplexed with LCD segment drives controlled by bits in the LCDSE1 register. I/O port functions are only available when the segments are disabled.

**TABLE 9-16: PORTH FUNCTIONS**

| Pin Name | Function | TRIS Setting | I/O | Buffer | Description |
|---|---|---|---|---|---|
| RH0/SEG47 | RH0 | 0 | O | DIG-4 | LATH<0> data output; disabled when LCD segment enabled. |
| | | 1 | I | ST | PORTH<0> data input. |
| | SEG47 | x | O | ANA | Segment 47 analog output for LCD. |
| RH1/SEG46 | RH1 | 0 | O | DIG | LATH<1> data output; disabled when LCD segment enabled. |
| | | 1 | I | ST | PORTH<1> data input. |
| | SEG46 | x | O | ANA | Segment 46 analog output for LCD. |
| RH2/SEG45 | RH2 | 0 | O | DIG | LATH<2> data output; disabled when LCD segment enabled. |
| | | 1 | I | ST | PORTH<2> data input. |
| | SEG45 | x | O | ANA | Segment 45 analog output for LCD. |
| RH3/SEG44 | RH3 | 0 | O | DIG | LATH<3> data output; disabled when LCD segment enabled. |
| | | 1 | I | ST | PORTH<3> data input. |
| | SEG44 | x | O | ANA | Segment 44 analog output for LCD. |
| RH4/SEG40 | RH4 | 0 | O | DIG | LATH<4> data output; disabled when LCD segment enabled. |
| | | 1 | I | ST | PORTH<4> data input. |
| | SEG40 | x | O | ANA | Segment 40 analog output for LCD |
| RH5/SEG41 | RH5 | 0 | O | DIG | LATH<5> data output; disabled when LCD segment enabled. |
| | | 1 | I | ST | PORTH<5> data input. |
| | SEG41 | x | O | ANA | Segment 41 analog output for LCD. |
| RH6/SEG42 | RH6 | 0 | O | DIG | LATH<6> data output; disabled when LCD segment enabled. |
| | | 1 | I | ST | PORTH<6> data input. |
| | SEG42 | x | O | ANA | Segment 42 analog output for LCD. |
| RH7/SEG43 | RH7 | 0 | O | DIG | LATH<7> data output; disabled when LCD segment enabled. |
| | | 1 | I | ST | PORTH<7> data input. |
| | SEG43 | x | O | ANA | Segment 43 analog output for LCD. |

**Legend:** O = Output, I = Input, ANA = Analog Signal, DIG = Digital Output, ST = Schmitt Buffer Input, TTL = TTL Buffer Input, x = Don't care (TRIS bit does not affect port direction or is overridden for this option).

**TABLE 9-17: SUMMARY OF REGISTERS ASSOCIATED WITH PORTH**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Reset Values on Page |
|---|---|---|---|---|---|---|---|---|---|
| TRISH | PORTH Data Direction Register | | | | | | | | 62 |
| PORTH | Read PORTH pin/Write PORTH Data Latch | | | | | | | | 62 |
| LATH | LATH Data Output Register | | | | | | | | 62 |
| LCDSE5 | SE47 | SE46 | SE45 | SE44 | SE43 | SE42 | SE41 | SE40 | 64 |

## 13.1 Timer3 Operation

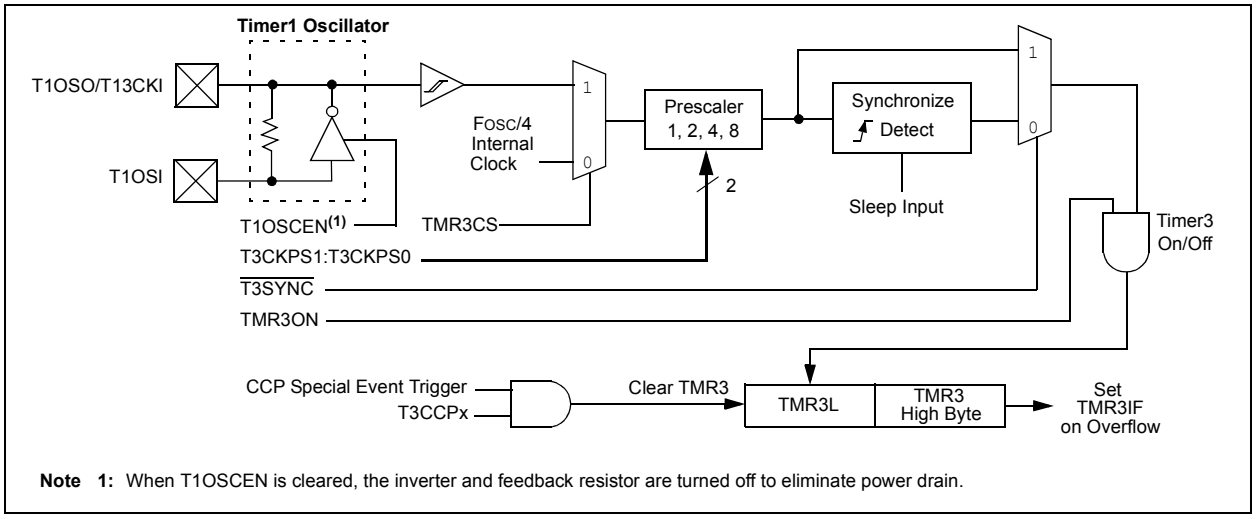
Timer3 can operate in one of three modes:

- Timer
- Synchronous counter
- Asynchronous counter

The operating mode is determined by the clock select bit, TMR3CS (T3CON<1>). When TMR3CS is cleared (= 0), Timer3 increments on every internal instruction cycle (FOSC/4). When the bit is set, Timer3 increments on every rising edge of the Timer1 external clock input or the Timer1 oscillator, if enabled.

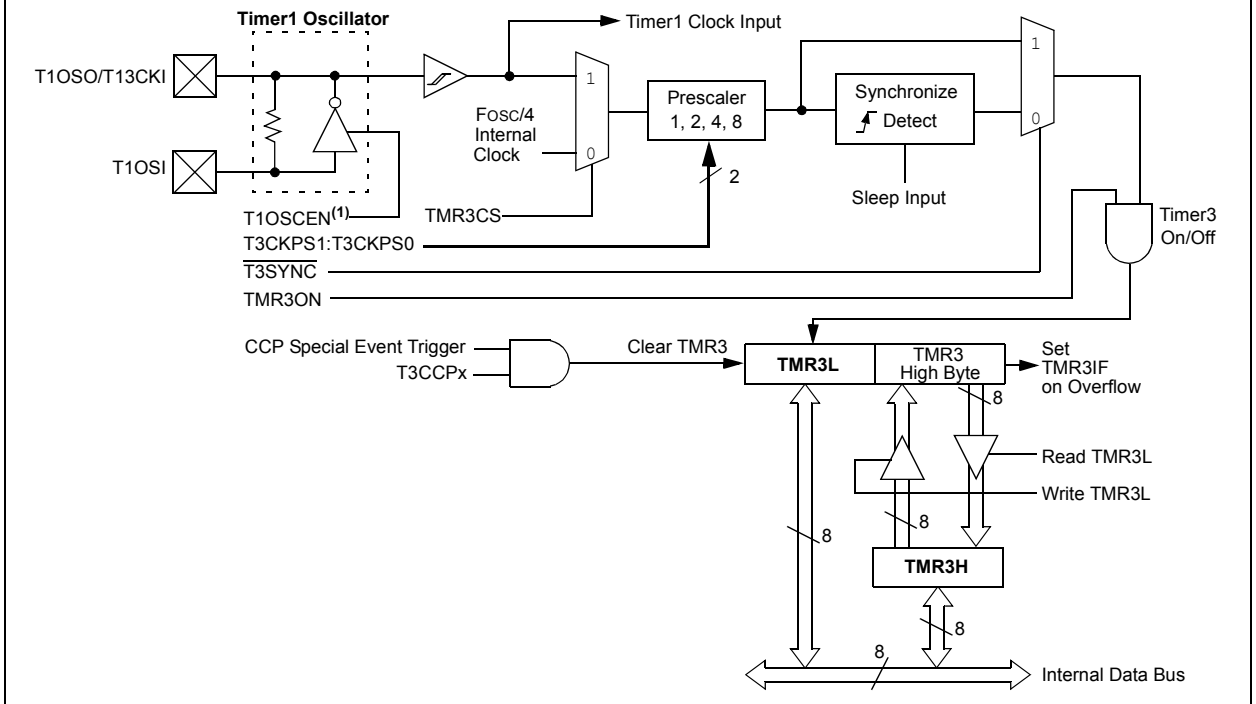
As with Timer1, the RC1/T1OSI and RC0/T1OSO/T13CKI pins become inputs when the Timer1 oscillator is enabled. This means the values of TRISC<1:0> are ignored and the pins are read as '0'.

**FIGURE 13-1: TIMER3 BLOCK DIAGRAM**




**Note 1:** When T1OSCEN is cleared, the inverter and feedback resistor are turned off to eliminate power drain.

**FIGURE 13-2: TIMER3 BLOCK DIAGRAM (16-BIT READ/WRITE MODE)**




**Note 1:** When T1OSCEN is cleared, the inverter and feedback resistor are turned off to eliminate power drain.

#### 15.4.17.1 Bus Collision During a Start Condition

During a Start condition, a bus collision occurs if:

a) SDA or SCL are sampled low at the beginning of the Start condition (Figure 15-26).

b) SCL is sampled low before SDA is asserted low (Figure 15-27).

During a Start condition, both the SDA and the SCL pins are monitored.

If the SDA pin is already low, or the SCL pin is already low, then all of the following occur:

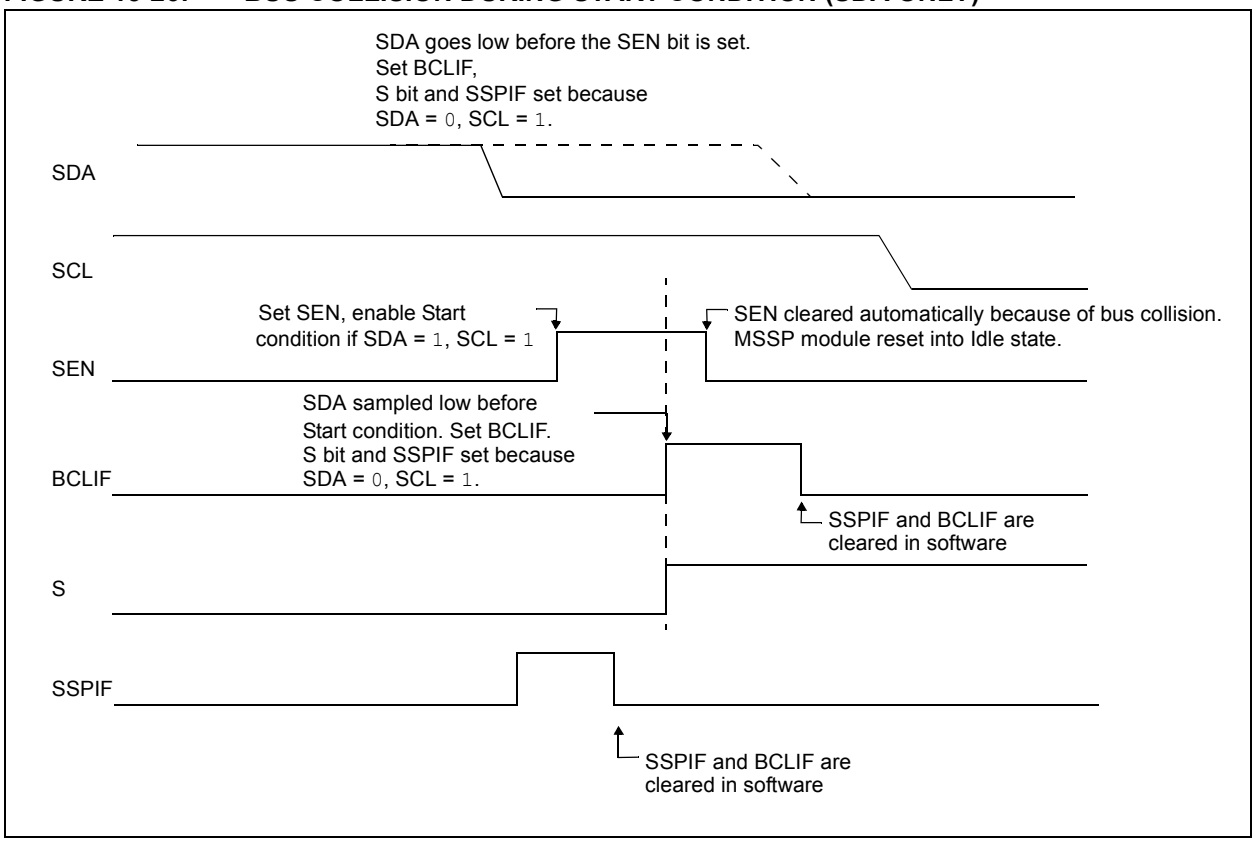
- the Start condition is aborted,
- the BCLIF flag is set and
- the MSSP module is reset to its Idle state (Figure 15-26).

The Start condition begins with the SDA and SCL pins deasserted. When the SDA pin is sampled high, the Baud Rate Generator is loaded from SSPADD<6:0> and counts down to '0'. If the SCL pin is sampled low while SDA is high, a bus collision occurs because it is assumed that another master is attempting to drive a data '1' during the Start condition.

If the SDA pin is sampled low during this count, the BRG is reset and the SDA line is asserted early (Figure 15-28). If, however, a '1' is sampled on the SDA pin, the SDA pin is asserted low at the end of the BRG count. The Baud Rate Generator is then reloaded and counts down to 0 and during this time, if the SCL pins are sampled as '0', a bus collision does not occur. At the end of the BRG count, the SCL pin is asserted low.

| **Note:** | The reason that bus collision is not a factor during a Start condition is that no two bus masters can assert a Start condition at the exact same time. Therefore, one master will always assert SDA before the other. This condition does not cause a bus collision because the two masters must be allowed to arbitrate the first address following the Start condition. If the address is the same, arbitration must be allowed to continue into the data portion, Repeated Start or Stop conditions. |
|---|---|

**FIGURE 15-26: BUS COLLISION DURING START CONDITION (SDA ONLY)**



**REGISTER 16-3: BAUDCON1: BAUD RATE CONTROL REGISTER 1**

| R/W-0 | R-1 | U-0 | R/W-0 | R/W-0 | U-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| ABDOVF | RCIDL | — | SCKP | BRG16 | — | WUE | ABDEN |
| 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 **ABDOVF:** Auto-Baud Acquisition Rollover Status bit
1 = A BRG rollover has occurred during Auto-Baud Rate Detect mode (must be cleared in software)
0 = No BRG rollover has occurred

bit 6 **RCIDL:** Receive Operation Idle Status bit
1 = Receive operation is Idle
0 = Receive operation is active

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

bit 4 **SCKP:** Synchronous Clock Polarity Select bit
Asynchronous mode:
Unused in this mode.
Synchronous mode:
1 = Idle state for clock (CK1) is a high level
0 = Idle state for clock (CK1) is a low level

bit 3 **BRG16:** 16-Bit Baud Rate Register Enable bit
1 = 16-bit Baud Rate Generator – SPBRGH1 and SPBRG1
0 = 8-bit Baud Rate Generator – SPBRG1 only (Compatible mode), SPBRGH1 value ignored

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

bit 1 **WUE:** Wake-up Enable bit
Asynchronous mode:
1 = EUSART will continue to sample the RX1 pin – interrupt generated on falling edge; bit cleared in hardware on following rising edge
0 = RX1 pin not monitored or rising edge detected
Synchronous mode:
Unused in this mode.

bit 0 **ABDEN:** Auto-Baud Detect Enable bit
Asynchronous mode:
1 = Enable baud rate measurement on the next character. Requires reception of a Sync field (55h); cleared in hardware upon completion.
0 = Baud rate measurement disabled or completed
Synchronous mode:
Unused in this mode.

### 16.1.3 AUTO-BAUD RATE DETECT

The Enhanced USART module supports the automatic detection and calibration of baud rate. This feature is active only in Asynchronous mode and while the WUE bit is clear.

The automatic baud rate measurement sequence (Figure 16-1) begins whenever a Start bit is received and the ABDEN bit is set. The calculation is self-averaging.

In the Auto-Baud Rate Detect (ABD) mode, the clock to the BRG is reversed. Rather than the BRG clocking the incoming RX1 signal, the RX1 signal is timing the BRG. In ABD mode, the internal Baud Rate Generator is used as a counter to time the bit period of the incoming serial byte stream.

Once the ABDEN bit is set, the state machine will clear the BRG and look for a Start bit. The Auto-Baud Rate Detect must receive a byte with the value, 55h (ASCII "U", which is also the LIN bus Sync character), in order to calculate the proper bit rate. The measurement is taken over both a low and a high bit time in order to minimize any effects caused by asymmetry of the incoming signal. After a Start bit, the SPBRG1 begins counting up, using the preselected clock source on the first rising edge of RX1. After eight bits on the RX1 pin or the fifth rising edge, an accumulated value totalling the proper BRG period is left in the SPBRGH1:SPBRG1 register pair. Once the 5th edge is seen (this should correspond to the Stop bit), the ABDEN bit is automatically cleared.

If a rollover of the BRG occurs (an overflow from FFFFh to 0000h), the event is trapped by the ABDOVF status bit (BAUDCON1<7>). It is set in hardware by BRG rollovers and can be set or cleared by the user in software. ABD mode remains active after rollover events and the ABDEN bit remains set (Figure 16-2).

While calibrating the baud rate period, the BRG registers are clocked at 1/8th the preconfigured clock rate. Note that the BRG clock will be configured by the BRG16 and BRGH bits. This allows the user to verify that no carry occurred for 8-bit modes by checking for 00h in the SPBRGH1 register. Refer to Table 16-4 for counter clock rates to the BRG.

While the ABD sequence takes place, the EUSART state machine is held in Idle. The RC1IF interrupt is set once the fifth rising edge on RX1 is detected. The value in the RCREG1 needs to be read to clear the RC1IF interrupt. The contents of RCREG1 should be discarded.

**Note 1:** If the WUE bit is set with the ABDEN bit, Auto-Baud Rate Detection will occur on the byte *following* the Break character.

**2:** It is up to the user to determine that the incoming character baud rate is within the range of the selected BRG clock source. Some combinations of oscillator frequency and EUSART baud rates are not possible due to bit error rates. Overall system timing and communication baud rates must be taken into consideration when using the Auto-Baud Rate Detection feature.

**3:** When the auto-baud feature is enabled, the BRG16 bit (BAUDCON<3>) must be set.

**TABLE 16-4: BRG COUNTER CLOCK RATES**

| BRG16 | BRGH | BRG Counter Clock |
|---|---|---|
| 0 | 0 | FOSC/512 |
| 0 | 1 | FOSC/128 |
| 1 | 0 | FOSC/128 |
| 1 | 1 | FOSC/32 |

#### 16.1.3.1 ABD and EUSART Transmission

Since the BRG clock is reversed during ABD acquisition, the EUSART transmitter cannot be used during ABD. This means that whenever the ABDEN bit is set, TXREG1 cannot be written to. Users should also ensure that ABDEN does not become set during a transmit sequence. Failing to do this may result in unpredictable EUSART operation.

**FIGURE 22-12: TYPE-B WAVEFORMS IN 1/3 MUX, 1/3 BIAS DRIVE**

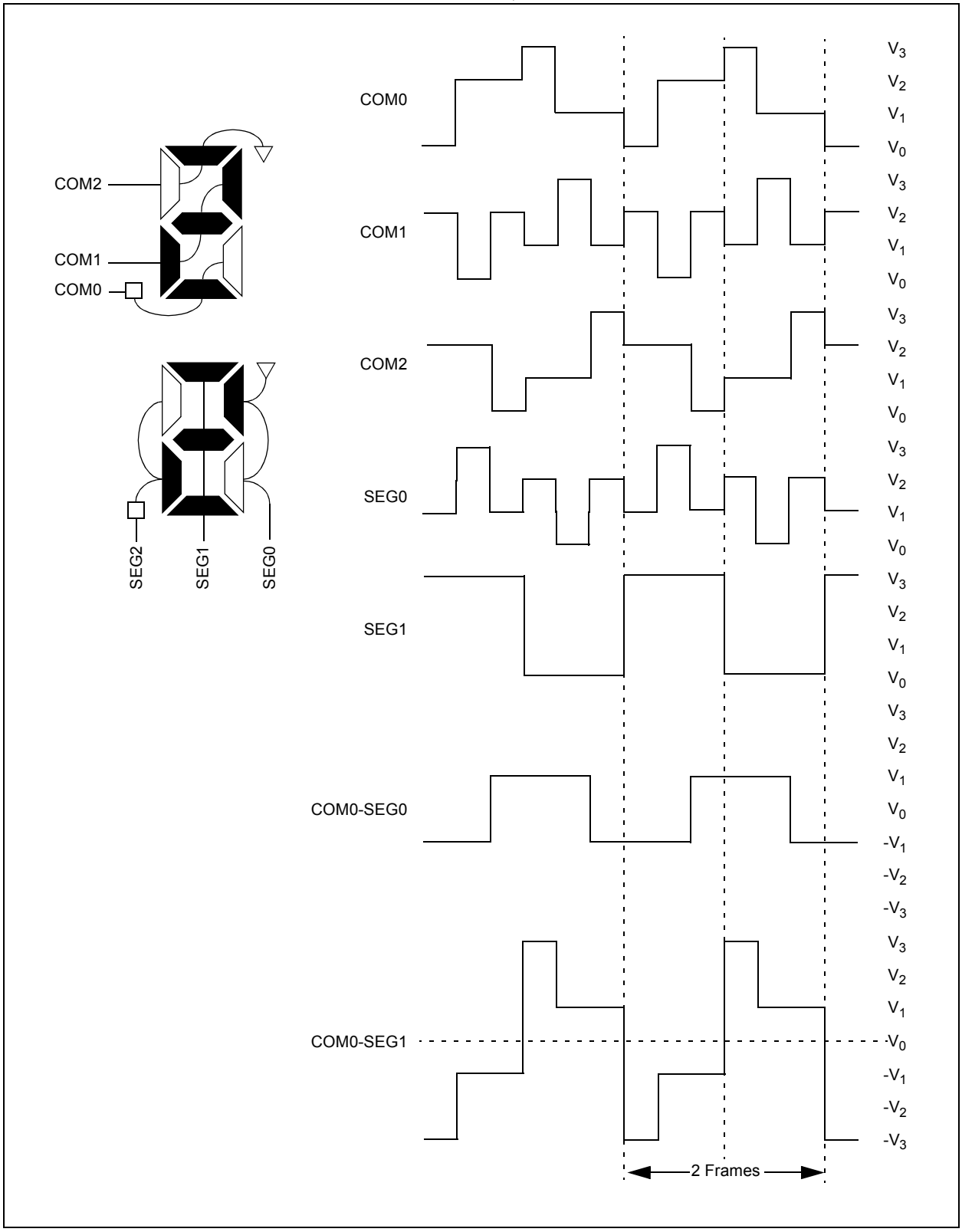
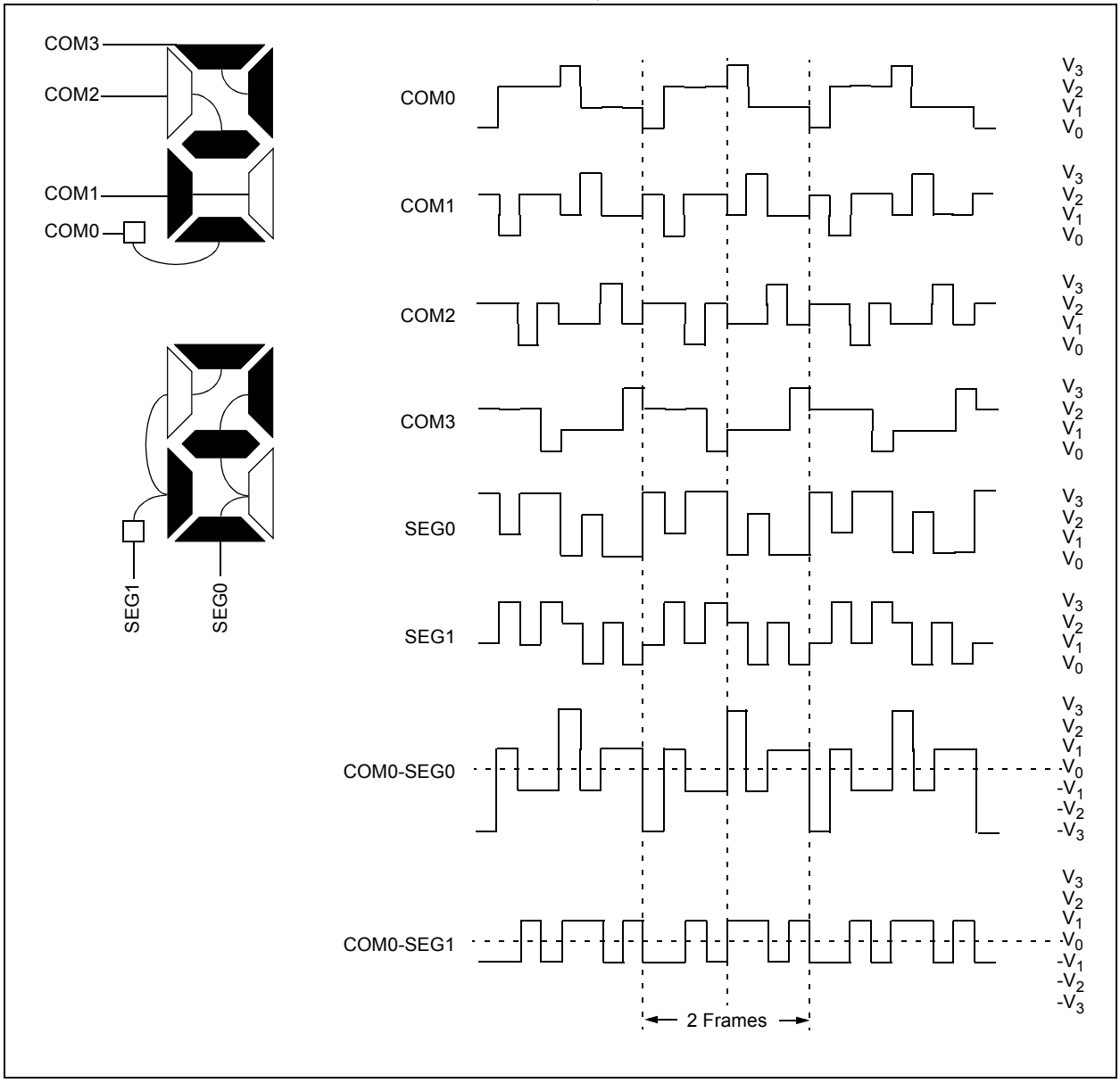


**FIGURE 22-14: TYPE-B WAVEFORMS IN 1/4 MUX, 1/3 BIAS DRIVE**



## 23.6 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.

## 23.7 In-Circuit Serial Programming

PIC18F6390/6490/8390/8490 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.

## 23.8 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 23-4 shows which resources are required by the background debugger.

**TABLE 23-4: DEBUGGER RESOURCES**

| | |
|---|---|
| I/O pins: | RB6, RB7 |
| Stack: | 2 levels |
| Program Memory: | 512 bytes |
| Data Memory: | 10 bytes |

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

**DAW** **Decimal Adjust W Register**

Syntax: DAW

Operands: None

Operation: If [W<3:0> >9] or [DC = 1] then,
(W<3:0>) + 6 → W<3:0>;
else,
(W<3:0>) → W<3:0>;

If [W<7:4> >9] or [C = 1] then,
(W<7:4>) + 6 → W<7:4>,
C = 1;
else,
(W<7:4>) → W<7:4>

Status Affected: C

Encoding:

| | | | |
|---|---|---|---|
| 0000 | 0000 | 0000 | 0111 |

Description: DAW adjusts the eight-bit value in W, resulting from the earlier addition of two variables (each in packed BCD format) and produces a correct packed BCD result.

Words: 1

Cycles: 1

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| Decode | Read register W | Process Data | Write W |

Example 1:

```
DAW
```

Before Instruction
W = A5h
C = 0
DC = 0
After Instruction
W = 05h
C = 1
DC = 0

Example 2:

Before Instruction
W = CEh
C = 0
DC = 0
After Instruction
W = 34h
C = 1
DC = 0

**DECF** **Decrement f**

Syntax: DECF f {,d {,a}}

Operands: 0 ≤ f ≤ 255
d ∈ [0,1]
a ∈ [0,1]

Operation: (f) – 1 → dest

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

Encoding:

| | | | |
|---|---|---|---|
| 0000 | 01da | ffff | ffff |

Description: Decrement register 'f'. If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f' (default).

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 24.2.3 "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 to destination |

Example: DECF CNT, 1, 0

Before Instruction
CNT = 01h
Z = 0
After Instruction
CNT = 00h
Z = 1

**DECFSZ** **Decrement f, Skip if 0**

Syntax: DECFSZ f {,d {,a}}

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

Operation: (f) – 1 → dest,
skip if result = 0

Status Affected: None

Encoding:

| 0010 | 11da | ffff | ffff |
|---|---|---|---|

Description: The contents of register 'f' are decremented. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed back in register 'f' (default).

If the result is '0', the next instruction which is already fetched is discarded and a NOP is executed instead, making it 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 24.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** 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 | Write to destination |

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    DECFSZ  CNT, 1, 1
        GOTO    LOOP
CONTINUE
```

Before Instruction
PC = Address (HERE)
After Instruction
CNT = CNT – 1
If CNT = 0;
PC = Address (CONTINUE)
If CNT ≠ 0;
PC = Address (HERE + 2)

**DCFSNZ** **Decrement f, Skip if not 0**

Syntax: DCFSNZ f {,d {,a}}

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

Operation: (f) – 1 → dest,
skip if result ≠ 0

Status Affected: None

Encoding:

| 0100 | 11da | ffff | ffff |
|---|---|---|---|

Description: The contents of register 'f' are decremented. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed back in register 'f' (default).

If the result is not '0', the next instruction which is already fetched is discarded and a NOP is executed instead, making it 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 24.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** 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 | Write to destination |

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    DCFSNZ  TEMP, 1, 0
ZERO    :
NZERO   :
```

Before Instruction
TEMP = ?
After Instruction
TEMP = TEMP – 1,
If TEMP = 0;
PC = Address (ZERO)
If TEMP ≠ 0;
PC = Address (NZERO)

## 26.4 AC (Timing) Characteristics

### 26.4.1 TIMING PARAMETER SYMBOLOGY

The timing parameter symbols have been created following one of the following formats:

1. TppS2ppS
2. TppS
3. TCC:ST ($I^2C$ specifications only)
4. Ts ($I^2C$ specifications only)

| | | | |
|---|---|---|---|
| T | | | |
| F | Frequency | T | Time |

Lowercase letters (pp) and their meanings:

| | | | |
|---|---|---|---|
| pp | | | |
| cc | CCP1 | osc | OSC1 |
| ck | CLKO | rd | $\overline{RD}$ |
| cs | $\overline{CS}$ | rw | $\overline{RD}$ or $\overline{WR}$ |
| di | SDI | sc | SCK |
| do | SDO | ss | $\overline{SS}$ |
| dt | Data in | t0 | T0CKI |
| io | I/O port | t1 | T13CKI |
| mc | $\overline{MCLR}$ | wr | $\overline{WR}$ |

Uppercase letters and their meanings:

| | | | |
|---|---|---|---|
| S | | | |
| F | Fall | P | Period |
| H | High | R | Rise |
| I | Invalid (High-impedance) | V | Valid |
| L | Low | Z | High-impedance |
| $I^2C$ only | | | |
| AA | output access | High | High |
| BUF | Bus free | Low | Low |

TCC:ST ($I^2C$ specifications only)

| | | | |
|---|---|---|---|
| CC | | | |
| HD | Hold | SU | Setup |
| ST | | | |
| DAT | DATA input hold | STO | Stop condition |
| STA | Start condition | | |