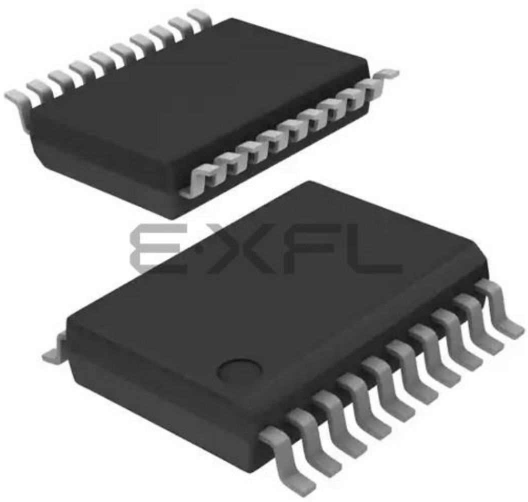




**What is "Embedded - Microcontrollers"?**

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

**Applications of "Embedded - Microcontrollers"**

| Details | |
|---|---|
| Product Status | Active |
| Core Processor | PIC |
| Core Size | 8-Bit |
| Speed | 32MHz |
| Connectivity | I²C, LINbus, SPI, UART/USART |
| Peripherals | Brown-out Detect/Reset, POR, PWM, WDT |
| Number of I/O | 15 |
| Program Memory Size | 14KB (8K x 14) |
| Program Memory Type | FLASH |
| EEPROM Size | 256 x 8 |
| RAM Size | 1K x 8 |
| Voltage - Supply (Vcc/Vdd) | 1.8V ~ 5.5V |
| Data Converters | A/D 12x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 20-SSOP (0.209", 5.30mm Width) |
| Supplier Device Package | 20-SSOP |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic16f1847t-i-ss |

**TABLE 1-2: PIC16(L)F1847 PINOUT DESCRIPTION**

| Name | Function | Input Type | Output Type | Description |
|---|---|---|---|---|
| RA0/AN0/CPS0/C12IN0-/SDO2 | RA0 | TTL | CMOS | General purpose I/O. |
| | AN0 | AN | — | ADC Channel 0 input. |
| | CPS0 | AN | — | Capacitive sensing input 0. |
| | C12IN0- | AN | — | Comparator C1 or C2 negative input. |
| | SDO2 | — | CMOS | SPI data output. |
| RA1/AN1/CPS1/C12IN1-/$\overline{\text{SS2}}$ | RA1 | TTL | CMOS | General purpose I/O. |
| | AN1 | AN | — | ADC Channel 1 input. |
| | CPS1 | AN | — | Capacitive sensing input 1. |
| | C12IN1- | AN | — | Comparator C1 or C2 negative input. |
| | $\overline{\text{SS2}}$ | ST | — | Slave Select input 2. |
| RA2/AN2/CPS2/C12IN2-/C12IN+/VREF-/DACOUT | RA2 | TTL | CMOS | General purpose I/O. |
| | AN2 | AN | — | ADC Channel 2 input. |
| | CPS2 | AN | — | Capacitive sensing input 2. |
| | C12IN2- | AN | — | Comparator C1 or C2 negative input. |
| | C12IN+ | AN | — | Comparator C1 or C2 positive input. |
| | VREF- | AN | — | ADC Negative Voltage Reference input. |
| | DACOUT | — | AN | Voltage Reference output. |
| RA3/AN3/CPS3/C12IN3-/C1IN+/VREF+/C1OUT/CCP3/SRQ | RA3 | TTL | CMOS | General purpose I/O. |
| | AN3 | AN | — | ADC Channel 3 input. |
| | CPS3 | AN | — | Capacitive sensing input 3. |
| | C12IN3- | AN | — | Comparator C1 or C2 negative input. |
| | C1IN+ | AN | — | Comparator C1 positive input. |
| | VREF+ | AN | — | ADC Voltage Reference input. |
| | C1OUT | — | CMOS | Comparator C1 output. |
| | CCP3 | ST | CMOS | Capture/Compare/PWM3. |
| | SRQ | — | CMOS | SR latch non-inverting output. |
| RA4/AN4/CPS4/C2OUT/T0CKI/CCP4/SRNQ | RA4 | TTL | CMOS | General purpose I/O. |
| | AN4 | AN | — | ADC Channel 4 input. |
| | CPS4 | AN | — | Capacitive sensing input 4. |
| | C2OUT | — | CMOS | Comparator C2 output. |
| | T0CKI | ST | — | Timer0 clock input. |
| | CCP4 | ST | CMOS | Capture/Compare/PWM4. |
| | SRNQ | — | CMOS | SR latch inverting output. |
| RA5/$\overline{\text{MCLR}}$/VPP/$\overline{\text{SS1}}$ | RA5 | TTL | CMOS | General purpose I/O. |
| | $\overline{\text{MCLR}}$ | ST | — | Master Clear with internal pull-up. |
| | VPP | HV | — | Programming voltage. |
| | $\overline{\text{SS1}}$ | ST | — | Slave Select input 1. |

**Legend:** AN = Analog input or output; CMOS = CMOS compatible input or output; OD = Open Drain
TTL = TTL compatible input; ST = Schmitt Trigger input with CMOS levels; I$^2$C™ = Schmitt Trigger input with I$^2$C levels
HV = High Voltage; XTAL = Crystal

**Note 1:** Pin functions can be moved using the APFCON0 or APFCON1 register.
**2:** Default function location.

**TABLE 3-5: PIC16(L)F1847 MEMORY MAP, BANKS 16-23**

| | BANK 16 | | BANK 17 | | BANK 18 | | BANK 19 | | BANK 20 | | BANK 21 | | BANK 22 | | BANK 23 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 800h | INDF0 | 880h | INDF0 | 900h | INDF0 | 980h | INDF0 | A00h | INDF0 | A80h | INDF0 | B00h | INDF0 | B80h | INDF0 |
| 801h | INDF1 | 881h | INDF1 | 901h | INDF1 | 981h | INDF1 | A01h | INDF1 | A81h | INDF1 | B01h | INDF1 | B81h | INDF1 |
| 802h | PCL | 882h | PCL | 902h | PCL | 982h | PCL | A02h | PCL | A82h | PCL | B02h | PCL | B82h | PCL |
| 803h | STATUS | 883h | STATUS | 903h | STATUS | 983h | STATUS | A03h | STATUS | A83h | STATUS | B03h | STATUS | B83h | STATUS |
| 804h | FSR0L | 884h | FSR0L | 904h | FSR0L | 984h | FSR0L | A04h | FSR0L | A84h | FSR0L | B04h | FSR0L | B84h | FSR0L |
| 805h | FSR0H | 885h | FSR0H | 905h | FSR0H | 985h | FSR0H | A05h | FSR0H | A85h | FSR0H | B05h | FSR0H | B85h | FSR0H |
| 806h | FSR1L | 886h | FSR1L | 906h | FSR1L | 986h | FSR1L | A06h | FSR1L | A86h | FSR1L | B06h | FSR1L | B86h | FSR1L |
| 807h | FSR1H | 887h | FSR1H | 907h | FSR1H | 987h | FSR1H | A07h | FSR1H | A87h | FSR1H | B07h | FSR1H | B87h | FSR1H |
| 808h | BSR | 888h | BSR | 908h | BSR | 988h | BSR | A08h | BSR | A88h | BSR | B08h | BSR | B88h | BSR |
| 809h | WREG | 889h | WREG | 909h | WREG | 989h | WREG | A09h | WREG | A89h | WREG | B09h | WREG | B89h | WREG |
| 80Ah | PCLATH | 88Ah | PCLATH | 90Ah | PCLATH | 98Ah | PCLATH | A0Ah | PCLATH | A8Ah | PCLATH | B0Ah | PCLATH | B8Ah | PCLATH |
| 80Bh | INTCON | 88Bh | INTCON | 90Bh | INTCON | 98Bh | INTCON | A0Bh | INTCON | A8Bh | INTCON | B0Bh | INTCON | B8Bh | INTCON |
| 80Ch | — | 88Ch | — | 90Ch | — | 98Ch | — | A0Ch | — | A8Ch | — | B0Ch | — | B8Ch | — |
| 80Dh | — | 88Dh | — | 90Dh | — | 98Dh | — | A0Dh | — | A8Dh | — | B0Dh | — | B8Dh | — |
| 80Eh | — | 88Eh | — | 90Eh | — | 98Eh | — | A0Eh | — | A8Eh | — | B0Eh | — | B8Eh | — |
| 80Fh | — | 88Fh | — | 90Fh | — | 98Fh | — | A0Fh | — | A8Fh | — | B0Fh | — | B8Fh | — |
| 810h | — | 890h | — | 910h | — | 990h | — | A10h | — | A90h | — | B10h | — | B90h | — |
| 811h | — | 891h | — | 911h | — | 991h | — | A11h | — | A91h | — | B11h | — | B91h | — |
| 812h | — | 892h | — | 912h | — | 992h | — | A12h | — | A92h | — | B12h | — | B92h | — |
| 813h | — | 893h | — | 913h | — | 993h | — | A13h | — | A93h | — | B13h | — | B93h | — |
| 814h | — | 894h | — | 914h | — | 994h | — | A14h | — | A94h | — | B14h | — | B94h | — |
| 815h | — | 895h | — | 915h | — | 995h | — | A15h | — | A95h | — | B15h | — | B95h | — |
| 816h | — | 896h | — | 916h | — | 996h | — | A16h | — | A96h | — | B16h | — | B96h | — |
| 817h | — | 897h | — | 917h | — | 997h | — | A17h | — | A97h | — | B17h | — | B97h | — |
| 818h | — | 898h | — | 918h | — | 998h | — | A18h | — | A98h | — | B18h | — | B98h | — |
| 819h | — | 899h | — | 919h | — | 999h | — | A19h | — | A99h | — | B19h | — | B99h | — |
| 81Ah | — | 89Ah | — | 91Ah | — | 99Ah | — | A1Ah | — | A9Ah | — | B1Ah | — | B9Ah | — |
| 81Bh | — | 89Bh | — | 91Bh | — | 99Bh | — | A1Bh | — | A9Bh | — | B1Bh | — | B9Bh | — |
| 81Ch | — | 89Ch | — | 91Ch | — | 99Ch | — | A1Ch | — | A9Ch | — | B1Ch | — | B9Ch | — |
| 81Dh | — | 89Dh | — | 91Dh | — | 99Dh | — | A1Dh | — | A9Dh | — | B1Dh | — | B9Dh | — |
| 81Eh | — | 89Eh | — | 91Eh | — | 99Eh | — | A1Eh | — | A9Eh | — | B1Eh | — | B9Eh | — |
| 81Fh | — | 89Fh | — | 91Fh | — | 99Fh | — | A1Fh | — | A9Fh | — | B1Fh | — | B9Fh | — |
| 820h–86Fh | Unimplemented Read as '0' | 8A0h–8EFh | Unimplemented Read as '0' | 920h–96Fh | Unimplemented Read as '0' | 9A0h–9EFh | Unimplemented Read as '0' | A20h–A6Fh | Unimplemented Read as '0' | AA0h–AEFh | Unimplemented Read as '0' | B20h–B6Fh | Unimplemented Read as '0' | BA0h–BEFh | Unimplemented Read as '0' |
| 870h–87Fh | Accesses 70h – 7Fh | 8F0h–8FFh | Accesses 70h – 7Fh | 970h–97Fh | Accesses 70h – 7Fh | 9F0h–9FFh | Accesses 70h – 7Fh | A70h–A7Fh | Accesses 70h – 7Fh | AF0h–AFFh | Accesses 70h – 7Fh | B70h–B7Fh | Accesses 70h – 7Fh | BF0h–BFFh | Accesses 70h – 7Fh |

# 6.0 REFERENCE CLOCK MODULE

The reference clock module provides the ability to send a divided clock to the clock output pin of the device (CLKR) and provide a secondary internal clock source to the modulator module. This module is available in all oscillator configurations and allows the user to select a greater range of clock submultiples to drive external devices in the application. The reference clock module includes the following features:

- System clock is the source
- Available in all oscillator configurations
- Programmable clock divider
- Output enable to a port pin
- Selectable duty cycle
- Slew rate control

The reference clock module is controlled by the CLKRCON register (Register 6-1) and is enabled when setting the CLKREN bit. To output the divided clock signal to the CLKR port pin, the CLKROE bit must be set. The CLKRDIV<2:0> bits enable the selection of eight different clock divider options. The CLKRDC<1:0> bits can be used to modify the duty cycle of the output clock[(1)]. The CLKRSLR bit controls slew rate limiting.

| | |
|---|---|
| **Note 1:** | If the base clock rate is selected without a divider, the output clock will always have a duty cycle equal to that of the source clock, unless a 0% duty cycle is selected. If the clock divider is set to base clock/2, then 25% and 75% duty cycle accuracy will be dependent upon the source clock. |

For information on using the reference clock output with the modulator module, see **Section 23.0 "Data Signal Modulator"**.

## 6.1 Slew Rate

The slew rate limitation on the output port pin can be disabled. The Slew Rate limitation can be removed by clearing the CLKRSLR bit in the CLKRCON register.

## 6.2 Effects of a Reset

Upon any device Reset, the reference clock module is disabled. The user's firmware is responsible for initializing the module before enabling the output. The registers are reset to their default values.

## 6.3 Conflicts with the CLKR pin

There are two cases when the reference clock output signal cannot be output to the CLKR pin, if:

- LP, XT or HS oscillator mode is selected.
- CLKOUT function is enabled.

Even if either of these cases are true, the module can still be enabled and the reference clock signal may be used in conjunction with the modulator module.

### 6.3.1 OSCILLATOR MODES

If LP, XT or HS oscillator modes are selected, the OSC2/CLKR pin must be used as an oscillator input pin and the CLKR output cannot be enabled. See **Section 5.2 "Clock Source Types"** for more information on different oscillator modes.

### 6.3.2 CLKOUT FUNCTION

The CLKOUT function has a higher priority than the reference clock module. Therefore, if the CLKOUT function is enabled by the $\overline{\text{CLKOUTEN}}$ bit in Configuration Words, FOSC/4 will always be output on the port pin. Reference **Section 4.0 "Device Configuration"** for more information.

## 6.4 Operation During Sleep

As the reference clock module relies on the system clock as its source, and the system clock is disabled in Sleep, the module does not function in Sleep, even if an external clock source or the Timer1 clock source is configured as the system clock. The module outputs will remain in their current state until the device exits Sleep.

## 8.3 Interrupts During Sleep

Some interrupts can be used to wake from Sleep. To wake from Sleep, the peripheral must be able to operate without the system clock. The interrupt source must have the appropriate Interrupt Enable bit(s) set prior to entering Sleep.

On waking from Sleep, if the GIE bit is also set, the processor will branch to the interrupt vector. Otherwise, the processor will continue executing instructions after the `SLEEP` instruction. The instruction directly after the `SLEEP` instruction will always be executed before branching to the ISR. Refer to the **Section 9.0 "Power-Down Mode (Sleep)"** for more details.

## 8.4 INT Pin

The INT pin can be used to generate an asynchronous edge-triggered interrupt. This interrupt is enabled by setting the INTE bit of the INTCON register. The INTEDG bit of the OPTION_REG register determines on which edge the interrupt will occur. When the INTEDG bit is set, the rising edge will cause the interrupt. When the INTEDG bit is clear, the falling edge will cause the interrupt. The INTF bit of the INTCON register will be set when a valid edge appears on the INT pin. If the GIE and INTE bits are also set, the processor will redirect program execution to the interrupt vector.

## 8.5 Automatic Context Saving

Upon entering an interrupt, the return PC address is saved on the stack. Additionally, the following registers are automatically saved in the Shadow registers:

- W register
- STATUS register (except for $\overline{\text{TO}}$ and $\overline{\text{PD}}$)
- BSR register
- FSR registers
- PCLATH register

Upon exiting the Interrupt Service Routine, these registers are automatically restored. Any modifications to these registers during the ISR will be lost. If modifications to any of these registers are desired, the corresponding shadow register should be modified and the value will be restored when exiting the ISR. The shadow registers are available in Bank 31 and are readable and writable. Depending on the user's application, other registers may also need to be saved.

**EXAMPLE 11-3: FLASH PROGRAM MEMORY READ**

```
* This code block will read 1 word of program
* memory at the memory address:
    PROG_ADDR_HI : PROG_ADDR_LO
*   data will be returned in the variables;
*   PROG_DATA_HI, PROG_DATA_LO

    BANKSEL  EEADRL          ; Select Bank for EEPROM registers
    MOVLW    PROG_ADDR_LO    ;
    MOVWF    EEADRL          ; Store LSB of address
    MOVLW    PROG_ADDR_HI    ;
    MOVWL    EEADRH          ; Store MSB of address

    BCF      EECON1,CFGS     ; Do not select Configuration Space
    BSF      EECON1,EEPGD    ; Select Program Memory
    BCF      INTCON,GIE      ; Disable interrupts
    BSF      EECON1,RD       ; Initiate read
    NOP                      ; Executed (Figure 11-1)
    NOP                      ; Ignored (Figure 11-1)
    BSF      INTCON,GIE      ; Restore interrupts

    MOVF     EEDATL,W        ; Get LSB of word
    MOVWF    PROG_DATA_LO    ; Store in user location
    MOVF     EEDATH,W        ; Get MSB of word
    MOVWF    PROG_DATA_HI    ; Store in user location
```

### REGISTER 11-5: EECON1: EEPROM CONTROL 1 REGISTER

| R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W/HC-0/0 | R/W-x/q | R/W-0/0 | R/S/HC-0/0 | R/S/HC-0/0 |
|---|---|---|---|---|---|---|---|
| EEPGD | CFGS | LWLO | FREE | WRERR | WREN | WR | RD |
| bit 7 | | | | | | | bit 0 |

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

bit 7 **EEPGD:** Flash Program/Data EEPROM Memory Select bit
1 = Accesses program space Flash memory
0 = Accesses data EEPROM memory

bit 6 **CFGS:** Flash Program/Data EEPROM or Configuration Select bit
1 = Accesses Configuration, User ID and Device ID Registers
0 = Accesses Flash Program or data EEPROM Memory

bit 5 **LWLO:** Load Write Latches Only bit
If CFGS = 1 (Configuration space) **OR** CFGS = 0 and EEPGD = 1 (program Flash):
1 = The next WR command does not initiate a write; only the program memory latches are updated.
0 = The next WR command writes a value from EEDATH:EEDATL into program memory latches and initiates a write of all the data stored in the program memory latches.

If CFGS = 0 and EEPGD = 0: (Accessing data EEPROM)
LWLO is ignored. The next WR command initiates a write to the data EEPROM.

bit 4 **FREE:** Program Flash Erase Enable bit
If CFGS = 1 (Configuration space) **OR** CFGS = 0 and EEPGD = 1 (program Flash):
1 = Performs an erase operation on the next WR command (cleared by hardware after completion of erase).
0 = Performs a write operation on the next WR command.

If EEPGD = 0 and CFGS = 0: (Accessing data EEPROM)
FREE is ignored. The next WR command will initiate both a erase cycle and a write cycle.

bit 3 **WRERR:** EEPROM Error Flag bit
1 = Condition indicates an improper program or erase sequence attempt or termination (bit is set automatically on any set attempt (write ‘1’) of the WR bit).
0 = The program or erase operation completed normally.

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

bit 1 **WR:** Write Control bit
1 = Initiates a program Flash or data EEPROM program/erase operation.
The operation is self-timed and the bit is cleared by hardware once operation is complete.
The WR bit can only be set (not cleared) in software.
0 = Program/erase operation to the Flash or data EEPROM is complete and inactive.

bit 0 **RD:** Read Control bit
1 = Initiates an program Flash or data EEPROM read. Read takes one cycle. RD is cleared in hardware. The RD bit can only be set (not cleared) in software.
0 = Does not initiate a program Flash or data EEPROM data read.

## 16.3 ADC Acquisition Requirements

For the ADC to meet its specified accuracy, the charge holding capacitor (CHOLD) must be allowed to fully charge to the input channel voltage level. The Analog Input model is shown in Figure 16-4. The source impedance (RS) and the internal sampling switch (RSS) impedance directly affect the time required to charge the capacitor CHOLD. The sampling switch (RSS) impedance varies over the device voltage (VDD), refer to Figure 16-4. **The maximum recommended impedance for analog sources is 10 kΩ.** As the source impedance is decreased, the acquisition time may be decreased. After the analog input channel is selected (or changed), an ADC acquisition must be done before the conversion can be started. To calculate the minimum acquisition time, Equation 16-1 may be used. This equation assumes that 1/2 LSb error is used (1,024 steps for the ADC). The 1/2 LSb error is the maximum error allowed for the ADC to meet its specified resolution.

**EQUATION 16-1: ACQUISITION TIME EXAMPLE**

*Assumptions:* $Temperature = 50°C$ *and external impedance of* $10k\Omega$ $5.0V$ $V_{DD}$

$$\begin{aligned} T_{ACQ} &= \text{Amplifier Settling Time} + \text{Hold Capacitor Charging Time} + \text{Temperature Coefficient} \\ &= T_{AMP} + T_C + T_{COFF} \\ &= 2\mu s + T_C + [(Temperature - 25°C)(0.05\mu s/°C)] \end{aligned}$$

*The value for* $T_C$ *can be approximated with the following equations:*

$$V_{APPLIED}\left(1 - \frac{1}{(2^{n+1}) - 1}\right) = V_{CHOLD} \quad ;[1]\ V_{CHOLD} \text{ charged to within 1/2 lsb}$$

$$V_{APPLIED}\left(1 - e^{\frac{-T_C}{RC}}\right) = V_{CHOLD} \quad ;[2]\ V_{CHOLD} \text{ charge response to } V_{APPLIED}$$

$$V_{APPLIED}\left(1 - e^{\frac{-T_C}{RC}}\right) = V_{APPLIED}\left(1 - \frac{1}{(2^{n+1}) - 1}\right) \quad ;\text{combining [1] and [2]}$$

*Note: Where n = number of bits of the ADC.*

*Solving for* $T_C$:

$$\begin{aligned} T_C &= -C_{HOLD}(R_{IC} + R_{SS} + R_S)\ \ln(1/2047) \\ &= -10pF(1k\Omega + 7k\Omega + 10k\Omega)\ \ln(0.0004885) \\ &= 1.37\mu s \end{aligned}$$

*Therefore:*

$$\begin{aligned} T_{ACQ} &= 2\mu s + 1.37\mu s + [(50°C - 25°C)(0.05\mu s/°C)] \\ &= 4.62\mu s \end{aligned}$$

**Note 1:** The reference voltage (VREF) has no effect on the equation, since it cancels itself out.

**2:** The charge holding capacitor (CHOLD) is not discharged after each conversion.

**3:** The maximum recommended impedance for analog sources is 10 kΩ. This is required to meet the pin leakage specification.

### 20.1.3 SOFTWARE PROGRAMMABLE PRESCALER

A software programmable prescaler is available for exclusive use with Timer0. The prescaler is enabled by clearing the PSA bit of the OPTION_REG register.

| **Note:** | The Watchdog Timer (WDT) uses its own independent prescaler. |
|---|---|

There are eight prescaler options for the Timer0 module ranging from 1:2 to 1:256. The prescale values are selectable via the PS<2:0> bits of the OPTION_REG register. In order to have a 1:1 prescaler value for the Timer0 module, the prescaler must be disabled by setting the PSA bit of the OPTION_REG register.

The prescaler is not readable or writable. All instructions writing to the TMR0 register will clear the prescaler.

### 20.1.4 TIMER0 INTERRUPT

Timer0 will generate an interrupt when the TMR0 register overflows from FFh to 00h. The TMR0IF interrupt flag bit of the INTCON register is set every time the TMR0 register overflows, regardless of whether or not the Timer0 interrupt is enabled. The TMR0IF bit can only be cleared in software. The Timer0 interrupt enable is the TMR0IE bit of the INTCON register.

| **Note:** | The Timer0 interrupt cannot wake the processor from Sleep since the timer is frozen during Sleep. |
|---|---|

### 20.1.5 8-BIT COUNTER MODE SYNCHRONIZATION

When in 8-Bit Counter mode, the incrementing edge on the T0CKI pin must be synchronized to the instruction clock. Synchronization can be accomplished by sampling the prescaler output on the Q2 and Q4 cycles of the instruction clock. The high and low periods of the external clocking source must meet the timing requirements as shown in **Section 30.0 "Electrical Specifications"**.

### 20.1.6 OPERATION DURING SLEEP

Timer0 cannot operate while the processor is in Sleep mode. The contents of the TMR0 register will remain unchanged while the processor is in Sleep mode.

## 22.1 Timer2/4/6 Operation

The clock input to the Timer2/4/6 modules is the system instruction clock (FOSC/4).

TMRx increments from 00h on each clock edge.

A 4-bit counter/prescaler on the clock input allows direct input, divide-by-4 and divide-by-16 prescale options. These options are selected by the prescaler control bits, TxCKPS<1:0> of the TxCON register. The value of TMRx is compared to that of the Period register, PRx, on each clock cycle. When the two values match, the comparator generates a match signal as the timer output. This signal also resets the value of TMRx to 00h on the next cycle and drives the output counter/postscaler (see **Section 22.2 "Timer2/4/6 Interrupt"**).

The TMRx and PRx registers are both directly readable and writable. The TMRx register is cleared on any device Reset, whereas the PRx register initializes to FFh. Both the prescaler and postscaler counters are cleared on the following events:

- a write to the TMRx register
- a write to the TxCON register
- Power-on Reset (POR)
- Brown-out Reset (BOR)
- $\overline{\text{MCLR}}$ Reset
- Watchdog Timer (WDT) Reset
- Stack Overflow Reset
- Stack Underflow Reset
- `RESET` Instruction

| **Note:** | TMRx is not cleared when TxCON is written. |
|---|---|

## 22.2 Timer2/4/6 Interrupt

Timer2/4/6 can also generate an optional device interrupt. The Timer2/4/6 output signal (TMRx-to-PRx match) provides the input for the 4-bit counter/postscaler. This counter generates the TMRx match interrupt flag which is latched in TMRxIF of the PIRx register. The interrupt is enabled by setting the TMRx Match Interrupt Enable bit, TMRxIE of the PIEx register.

A range of 16 postscale options (from 1:1 through 1:16 inclusive) can be selected with the postscaler control bits, TxOUTPS<3:0>, of the TxCON register.

## 22.3 Timer2/4/6 Output

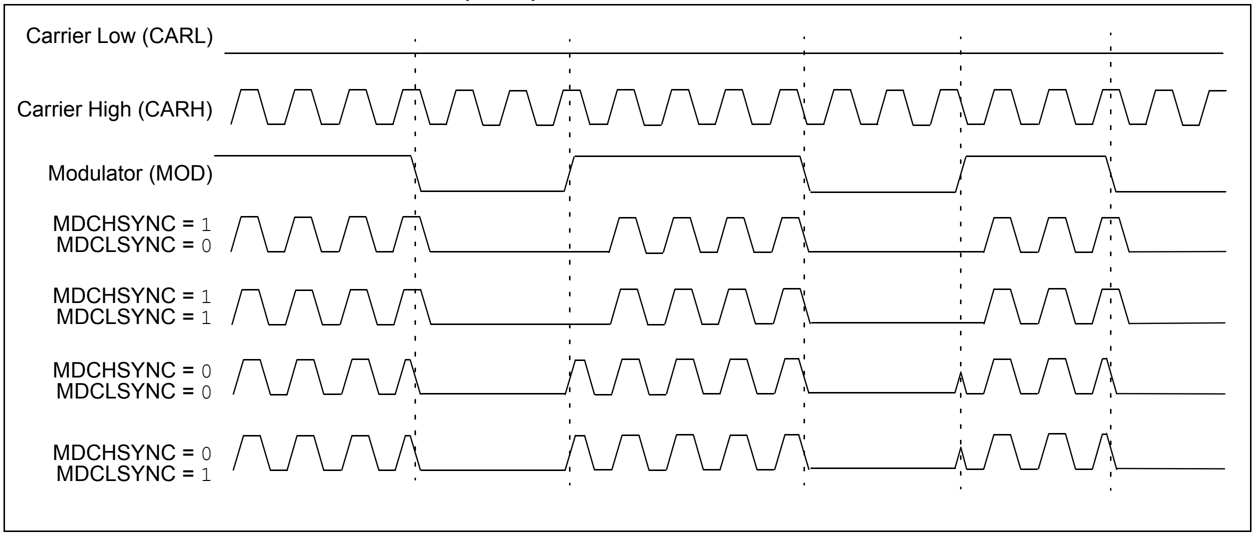
The unscaled output of TMRx is available primarily to the CCP modules, where it is used as a time base for operations in PWM mode.

Timer2 can be optionally used as the shift clock source for the MSSPx modules operating in SPI mode. Additional information is provided in **Section 25.0 "Master Synchronous Serial Port (MSSP1 and MSSP2) Module"**.

## 22.4 Timer2/4/6 Operation During Sleep

The Timer2/4/6 timers cannot be operated while the processor is in Sleep mode. The contents of the TMRx and PRx registers will remain unchanged while the processor is in Sleep mode.

**FIGURE 23-2: ON OFF KEYING (OOK) SYNCHRONIZATION**




**EXAMPLE 23-1: NO SYNCHRONIZATION (MDSHSYNC = 0, MDCLSYNC = 0)**

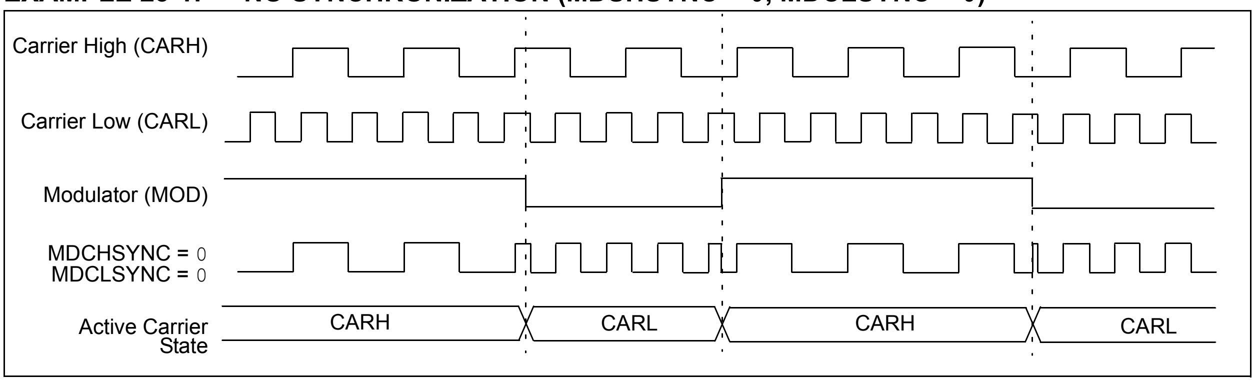
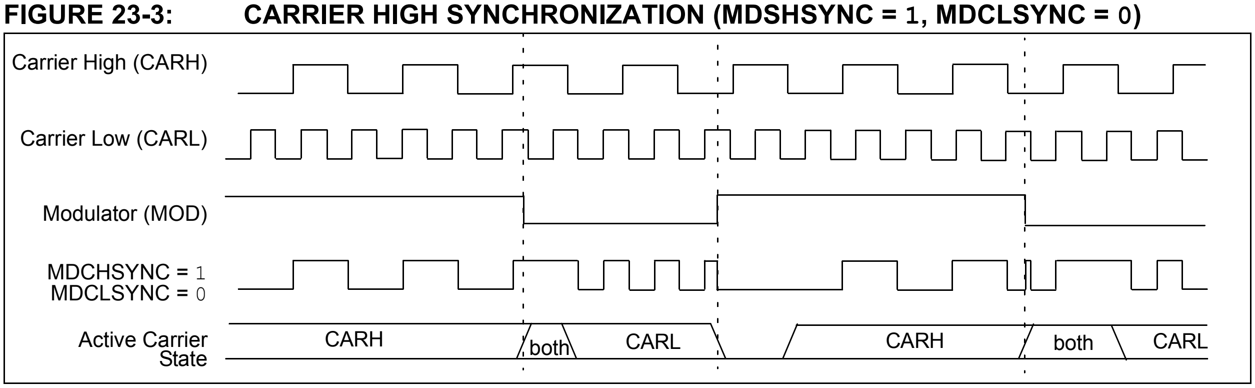



**FIGURE 23-3: CARRIER HIGH SYNCHRONIZATION (MDSHSYNC = 1, MDCLSYNC = 0)**



### REGISTER 23-4: MDCARL: MODULATION LOW CARRIER CONTROL REGISTER

| R/W-x/u | R/W-x/u | R/W-x/u | U-0 | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u |
|---|---|---|---|---|---|---|---|
| MDCLODIS | MDCLPOL | MDCLSYNC | — | MDCL<3:0> | | | |
| bit 7 | | | | | | | bit 0 |

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

bit 7 **MDCLODIS:** Modulator Low Carrier Output Disable bit
- 1 = Output signal driving the peripheral output pin (selected by MDCL<3:0> of the MDCARL register) is disabled
- 0 = Output signal driving the peripheral output pin (selected by MDCL<3:0> of the MDCARL register) is enabled

bit 6 **MDCLPOL:** Modulator Low Carrier Polarity Select bit
- 1 = Selected low carrier signal is inverted
- 0 = Selected low carrier signal is not inverted

bit 5 **MDCLSYNC:** Modulator Low Carrier Synchronization Enable bit
- 1 = Modulator waits for a falling edge on the low time carrier signal before allowing a switch to the high time carrier
- 0 = Modulator Output is not synchronized to the low time carrier signal(1)

bit 4 **Unimplemented:** Read as ‘0’

bit 3-0 **MDCL<3:0>** Modulator Data High Carrier Selection bits (1)
- 1111 = Reserved. No channel connected.
- •
- •
- •
- 1000 = Reserved. No channel connected.
- 0111 = CCP4 output (PWM Output mode only)
- 0110 = CCP3 output (PWM Output mode only)
- 0101 = CCP2 output (PWM Output mode only)
- 0100 = CCP1 output (PWM Output mode only)
- 0011 = Reference Clock module signal
- 0010 = MDCIN2 port pin
- 0001 = MDCIN1 port pin
- 0000 = Vss

**Note 1:** Narrowed carrier pulse widths or spurs may occur in the signal stream if the carrier is not synchronized.

### TABLE 23-1: SUMMARY OF REGISTERS ASSOCIATED WITH DATA SIGNAL MODULATOR MODE

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| MDCARH | MDCHODIS | MDCHPOL | MDCHSYNC | — | MDCH<3:0> | | | | 200 |
| MDCARL | MDCLODIS | MDCLPOL | MDCLSYNC | — | MDCL<3:0> | | | | 201 |
| MDCON | MDEN | MDOE | MDSLR | MDOPOL | MDOUT | — | — | MDBIT | 198 |
| MDSRC | MDMSODIS | — | — | — | MDMS<3:0> | | | | 199 |

**Legend:** — = unimplemented, read as ‘0’. Shaded cells are not used in the Data Signal Modulator mode.

### 24.2.5 COMPARE DURING SLEEP

The Compare mode is dependent upon the system clock (FOSC) for proper operation. Since FOSC is shut down during Sleep mode, the Compare mode will not function properly during Sleep.

### 24.2.6 ALTERNATE PIN LOCATIONS

This module incorporates I/O pins that can be moved to other locations with the use of the alternate pin function registers, APFCON0 and APFCON1. To determine which pins can be moved and what their default locations are upon a Reset, see **Section 12.1 "Alternate Pin Function"** for more information.

**TABLE 24-4: SUMMARY OF REGISTERS ASSOCIATED WITH COMPARE**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| APFCON0 | RXDTSEL | SDO1SEL | SS1SEL | P2BSEL | CCP2SEL | P1DSEL | P1CSEL | CCP1SEL | 118 |
| CCP1CON | P1M<1:0> | | DC1B<1:0> | | CCP1M<3:0> | | | | 226 |
| CCPR1L | Capture/Compare/PWM Register Low Byte (LSB) | | | | | | | | 204* |
| CCPR1H | Capture/Compare/PWM Register High Byte (MSB) | | | | | | | | 204* |
| CM1CON0 | C1ON | C1OUT | C1OE | C1POL | — | C1SP | C1HYS | C1SYNC | 170 |
| CM1CON1 | C1INTP | C1INTN | C1PCH<1:0> | | — | — | C1NCH<1:0> | | 171 |
| CM2CON0 | C2ON | C2OUT | C2OE | C2POL | — | C2SP | C2HYS | C2SYNC | 170 |
| CM2CON1 | C2INTP | C2INTN | C2PCH<1:0> | | — | — | C2NCH<1:0> | | 171 |
| CCP2CON | P2M<1:0> | | DC2B<1:0> | | CCP2M<3:0> | | | | 226 |
| CCPR2L | Capture/Compare/PWM Register Low Byte (LSB) | | | | | | | | 226 |
| CCPR2H | Capture/Compare/PWM Register High Byte (MSB) | | | | | | | | 226 |
| CCP3CON | — | — | DC3B<1:0> | | CCP3M<3:0> | | | | 226 |
| CCPR3L | Capture/Compare/PWM Register Low Byte (LSB) | | | | | | | | 226 |
| CCPR3H | Capture/Compare/PWM Register High Byte (MSB) | | | | | | | | 226 |
| CCP4CON | — | — | DC4B<1:0> | | CCP4M<3:0)> | | | | 226 |
| CCPR4L | Capture/Compare/PWM Register Low Byte (LSB) | | | | | | | | 226 |
| CCPR4H | Capture/Compare/PWM Register High Byte (MSB) | | | | | | | | 226 |
| INTCON | GIE | PEIE | TMR0IE | INTE | IOCE | TMR0IF | INTF | IOCF | 88 |
| PIE1 | TMR1GIE | ADIE | RCIE | TXIE | SSP1IE | CCP1IE | TMR2IE | TMR1IE | 89 |
| PIE2 | OSFIE | C2IE | C1IE | EEIE | BCL1IE | — | — | CCP2IE | 90 |
| PIE3 | — | — | CCP4IE | CCP3IE | TMR6IE | — | TMR4IE | — | 91 |
| PIR1 | TMR1GIF | ADIF | RCIF | TXIF | SSP1IF | CCP1IF | TMR2IF | TMR1IF | 93 |
| PIR2 | OSFIF | C2IF | C1IF | EEIF | BCL1IF | — | — | CCP2IF | 94 |
| PIR3 | — | — | CCP4IF | CCP3IF | TMR6IF | — | TMR4IF | — | 95 |
| T1CON | TMR1CS<1:0> | | T1CKPS<1:0> | | T1OSCEN | $\overline{\text{T1SYNC}}$ | — | TMR1ON | 185 |
| T1GCON | TMR1GE | T1GPOL | T1GTM | T1GSPM | T1GGO/$\overline{\text{DONE}}$ | T1GVAL | T1GSS<1:0> | | 186 |
| TMR1L | Holding Register for the Least Significant Byte of the 16-bit TMR1 Register | | | | | | | | 177* |
| TMR1H | Holding Register for the Most Significant Byte of the 16-bit TMR1 Register | | | | | | | | 177* |
| TRISA | TRISA7 | TRISA6 | TRISA5 | TRISA4 | TRISA3 | TRISA2 | TRISA1 | TRISA0 | 120 |
| TRISB | TRISB7 | TRISB6 | TRISB5 | TRISB4 | TRISB3 | TRISB2 | TRISB1 | TRISB0 | 126 |

**Legend:** — = Unimplemented locations, read as '0'. Shaded cells are not used by Compare mode.
* Page provides register information.

When one device is transmitting a logical one, or letting the line float, and a second device is transmitting a logical zero, or holding the line low, the first device can detect that the line is not a logical one. This detection, when used on the SCLx line, is called clock stretching. Clock stretching gives slave devices a mechanism to control the flow of data. When this detection is used on the SDAx line, it is called arbitration. Arbitration ensures that there is only one master device communicating at any single time.

### 25.3.1 CLOCK STRETCHING

When a slave device has not completed processing data, it can delay the transfer of more data through the process of Clock Stretching. An addressed slave device may hold the SCLx clock line low after receiving or sending a bit, indicating that it is not yet ready to continue. The master that is communicating with the slave will attempt to raise the SCLx line in order to transfer the next bit, but will detect that the clock line has not yet been released. Because the SCLx connection is open-drain, the slave has the ability to hold that line low until it is ready to continue communicating.

Clock stretching allows receivers that cannot keep up with a transmitter to control the flow of incoming data.

### 25.3.2 ARBITRATION

Each master device must monitor the bus for Start and Stop bits. If the device detects that the bus is busy, it cannot begin a new message until the bus returns to an Idle state.

However, two master devices may try to initiate a transmission on or about the same time. When this occurs, the process of arbitration begins. Each transmitter checks the level of the SDAx data line and compares it to the level that it expects to find. The first transmitter to observe that the two levels do not match, loses arbitration, and must stop transmitting on the SDAx line.

For example, if one transmitter holds the SDAx line to a logical one (lets it float) and a second transmitter holds it to a logical zero (pulls it low), the result is that the SDAx line will be low. The first transmitter then observes that the level of the line is different than expected and concludes that another transmitter is communicating.

The first transmitter to notice this difference is the one that loses arbitration and must stop driving the SDAx line. If this transmitter is also a master device, it also must stop driving the SCLx line. It then can monitor the lines for a Stop condition before trying to reissue its transmission. In the meantime, the other device that has not noticed any difference between the expected and actual levels on the SDAx line continues with its original transmission. It can do so without any complications, because so far, the transmission appears exactly as expected with no other transmitter disturbing the message.

Slave Transmit mode can also be arbitrated, when a master addresses multiple slaves, but this is less common.

If two master devices are sending a message to two different slave devices at the address stage, the master sending the lower slave address always wins arbitration. When two master devices send messages to the same slave address, and addresses can sometimes refer to multiple slaves, the arbitration process must continue into the data stage.

Arbitration usually occurs very rarely, but it is a necessary process for proper multi-master support.

#### 25.5.3.3 7-Bit Transmission with Address Hold Enabled

Setting the AHEN bit of the SSPxCON3 register enables additional clock stretching and interrupt generation after the 8th falling edge of a received matching address. Once a matching address has been clocked in, CKP is cleared and the SSPxIF interrupt is set.

Figure 25-19 displays a standard waveform of a 7-bit Address Slave Transmission with AHEN enabled.

1. Bus starts Idle.
2. Master sends Start condition; the S bit of SSPxSTAT is set; SSPxIF is set if interrupt on Start detect is enabled.
3. Master sends matching address with R/$\overline{\text{W}}$ bit set. After the 8th falling edge of the SCLx line the CKP bit is cleared and SSPxIF interrupt is generated.
4. Slave software clears SSPxIF.
5. Slave software reads ACKTIM bit of SSPxCON3 register, and R/$\overline{\text{W}}$ and D/$\overline{\text{A}}$ of the SSPxSTAT register to determine the source of the interrupt.
6. Slave reads the address value from the SSPxBUF register clearing the BF bit.
7. Slave software decides from this information if it wishes to $\overline{\text{ACK}}$ or not $\overline{\text{ACK}}$ and sets the ACKDT bit of the SSPxCON2 register accordingly.
8. Slave sets the CKP bit releasing SCLx.
9. Master clocks in the $\overline{\text{ACK}}$ value from the slave.
10. Slave hardware automatically clears the CKP bit and sets SSPxIF after the $\overline{\text{ACK}}$ if the R/$\overline{\text{W}}$ bit is set.
11. Slave software clears SSPxIF.
12. Slave loads value to transmit to the master into SSPxBUF setting the BF bit.

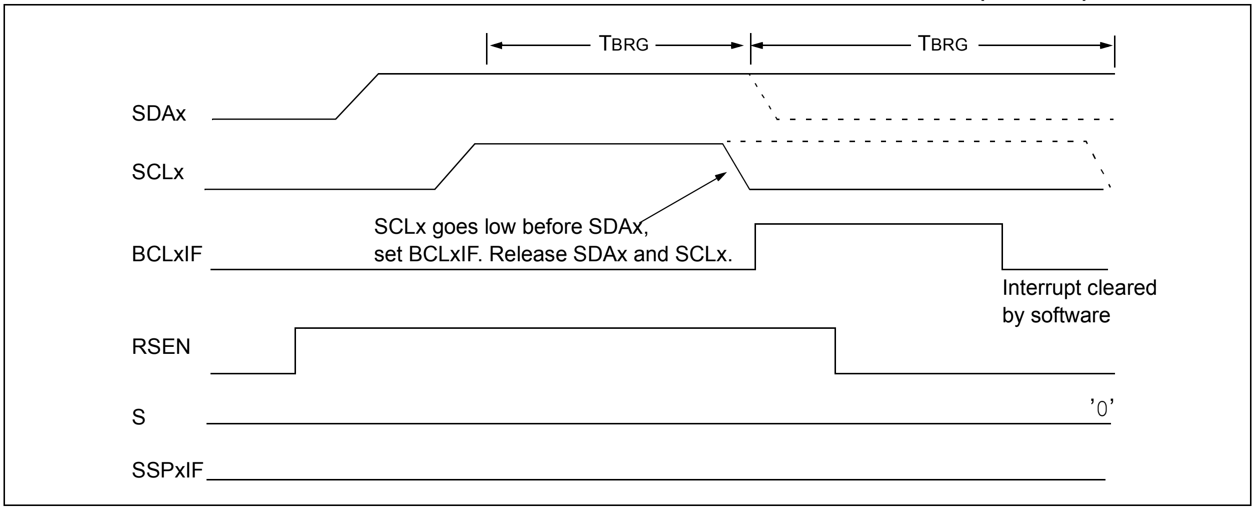
| **Note:** | SSPxBUF cannot be loaded until after the $\overline{\text{ACK}}$. |
|---|---|

13. Slave sets CKP bit releasing the clock.
14. Master clocks out the data from the slave and sends an $\overline{\text{ACK}}$ value on the 9th SCLx pulse.
15. Slave hardware copies the $\overline{\text{ACK}}$ value into the ACKSTAT bit of the SSPxCON2 register.
16. Steps 10-15 are repeated for each byte transmitted to the master from the slave.
17. If the master sends a not $\overline{\text{ACK}}$ the slave releases the bus allowing the master to send a Stop and end the communication.

| **Note:** | Master must send a not $\overline{\text{ACK}}$ on the last byte to ensure that the slave releases the SCLx line to receive a Stop. |
|---|---|

**FIGURE 25-37: BUS COLLISION DURING REPEATED START CONDITION (CASE 2)**



## 25.8 Register Definitions: MSSP Control

### REGISTER 25-1: SSPxSTAT: SSPx STATUS REGISTER

| R/W-0/0 | R/W-0/0 | R-0/0 | R-0/0 | R-0/0 | R-0/0 | R-0/0 | R-0/0 |
|---|---|---|---|---|---|---|---|
| SMP | CKE | $D/\overline{A}$ | P | S | $R/\overline{W}$ | UA | BF |
| bit 7 | | | | | | | bit 0 |

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

bit 7 **SMP:** SPI Data Input Sample bit

SPI Master mode:
1 = Input data sampled at end of data output time
0 = Input data sampled at middle of data output time

SPI Slave mode:
SMP must be cleared when SPI is used in Slave mode

In $I^2C$ Master or Slave mode:
1 = Slew rate control disabled for standard speed mode (100 kHz and 1 MHz)
0 = Slew rate control enabled for high speed mode (400 kHz)

bit 6 **CKE:** SPI Clock Edge Select bit (SPI mode only)

In SPI Master or Slave mode:
1 = Transmit occurs on transition from active to Idle clock state
0 = Transmit occurs on transition from Idle to active clock state

In $I^2C$™ mode only:
1 = Enable input logic so that thresholds are compliant with SMBus specification
0 = Disable SMBus specific inputs

bit 5 **$D/\overline{A}$:** Data/$\overline{\text{Address}}$ bit ($I^2C$ mode only)
1 = Indicates that the last byte received or transmitted was data
0 = Indicates that the last byte received or transmitted was address

bit 4 **P:** Stop bit
($I^2C$ mode only. This bit is cleared when the MSSPx module is disabled, SSPEN is cleared.)
1 = Indicates that a Stop bit has been detected last (this bit is '0' on Reset)
0 = Stop bit was not detected last

bit 3 **S:** Start bit
($I^2C$ mode only. This bit is cleared when the MSSPx module is disabled, SSPEN is cleared.)
1 = Indicates that a Start bit has been detected last (this bit is '0' on Reset)
0 = Start bit was not detected last

bit 2 **$R/\overline{W}$:** Read/$\overline{\text{Write}}$ bit information ($I^2C$ mode only)
This bit holds the $R/\overline{W}$ bit information following the last address match. This bit is only valid from the address match to the next Start bit, Stop bit, or not $\overline{\text{ACK}}$ bit.

In $I^2C$ Slave mode:
1 = Read
0 = Write

In $I^2C$ Master mode:
1 = Transmit is in progress
0 = Transmit is not in progress
OR-ing this bit with SEN, RSEN, PEN, RCEN or ACKEN will indicate if the MSSPx is in Idle mode.

bit 1 **UA:** Update Address bit (10-bit $I^2C$ mode only)
1 = Indicates that the user needs to update the address in the SSPxADD register
0 = Address does not need to be updated

**TABLE 30-2: SUPPLY CURRENT (IDD)(1,2)**

| PIC16LF1847 | | Standard Operating Conditions (unless otherwise stated) | | | | | |
|---|---|---|---|---|---|---|---|
| PIC16F1847 | | | | | | | |
| Param. No. | Device Characteristics | Min. | Typ† | Max. | Units | Conditions: VDD | Conditions: Note |
| D010 | | — | 9.5 | 14 | μA | 1.8 | FOSC = 32 kHz<br>LP Oscillator<br>-40°C ≤ TA ≤ +85°C |
| | | — | 12.5 | 17 | μA | 3.0 | |
| D010 | | — | 22 | 29 | μA | 1.8 | FOSC = 32 kHz<br>LP Oscillator<br>-40°C ≤ TA ≤ +85°C |
| | | — | 27 | 35 | μA | 3.0 | |
| | | — | 30 | 38 | μA | 5.0 | |
| D010A | | — | 9.5 | 14 | μA | 1.8 | FOSC = 32 kHz<br>LP Oscillator<br>-40°C ≤ TA ≤ +125°C |
| | | — | 12.5 | 17 | μA | 3.0 | |
| D010A | | — | 22 | 29 | μA | 1.8 | FOSC = 32 kHz<br>LP Oscillator<br>-40°C ≤ TA ≤ +125°C |
| | | — | 27 | 35 | μA | 3.0 | |
| | | — | 30 | 38 | μA | 5.0 | |
| D011 | | — | 105 | 110 | μA | 1.8 | FOSC = 1 MHz<br>XT Oscillator |
| | | — | 160 | 190 | μA | 3.0 | |
| D011 | | — | 132 | 154 | μA | 1.8 | FOSC = 1 MHz<br>XT Oscillator |
| | | — | 186 | 220 | μA | 3.0 | |
| | | — | 216 | 290 | μA | 5.0 | |
| D012 | | — | 264 | 370 | μA | 1.8 | FOSC = 4 MHz<br>XT Oscillator |
| | | — | 491 | 620 | μA | 3.0 | |
| D012 | | — | 285 | 300 | μA | 1.8 | FOSC = 4 MHz<br>XT Oscillator |
| | | — | 408 | 600 | μA | 3.0 | |
| | | — | 490 | 700 | μA | 5.0 | |
| D013 | | — | 55 | 160 | μA | 1.8 | FOSC = 1 MHz<br>EC Oscillator<br>Medium-Power mode |
| | | — | 90 | 230 | μA | 3.0 | |
| D013 | | — | 75 | 95 | μA | 1.8 | FOSC = 1 MHz<br>EC Oscillator<br>Medium-Power mode |
| | | — | 116 | 130 | μA | 3.0 | |
| | | — | 145 | 185 | μA | 5.0 | |

* These parameters are characterized but not tested.

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** The test conditions for all IDD measurements in active operation mode are: OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD; $\overline{MCLR}$ = VDD; WDT disabled.

**2:** The supply current is mainly a function of the operating voltage and frequency. Other factors, such as I/O pin loading and switching rate, oscillator type, internal code execution pattern and temperature, also have an impact on the current consumption.

**3:** 8 MHz internal oscillator with 4x PLL enabled.

**4:** 8 MHz crystal oscillator with 4x PLL enabled.

**5:** For RC oscillator configurations, current through REXT is not included. The current through the resistor can be extended by the formula IR = VDD/2REXT (mA) with REXT in kΩ.

**FIGURE 31-15: IDD TYPICAL, HFINTOSC MODE, PIC16LF1847 ONLY**

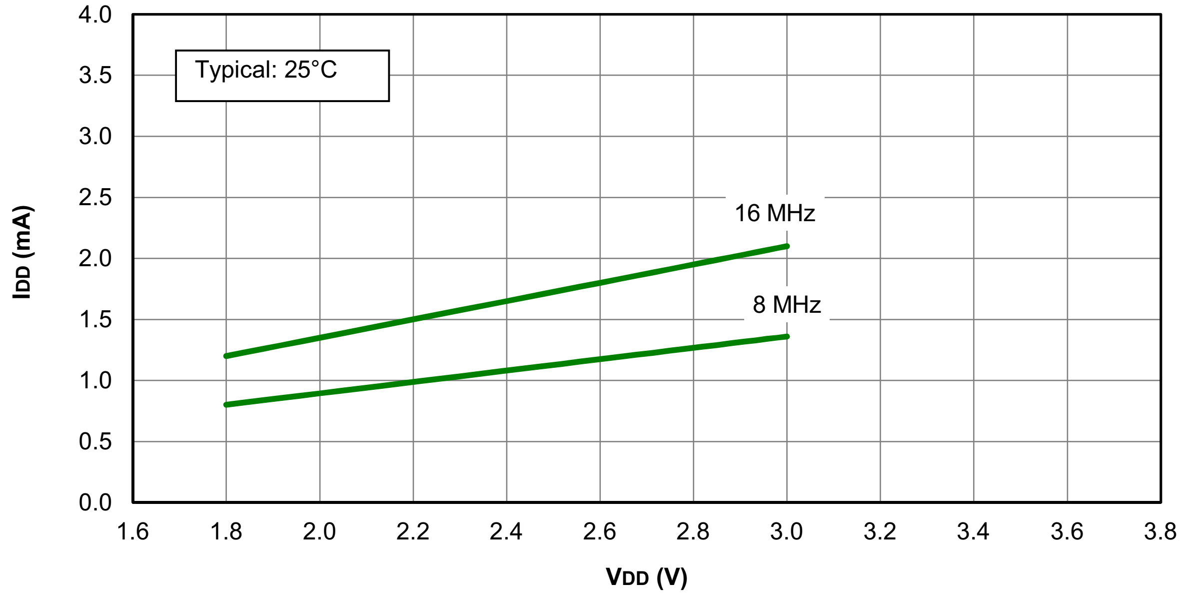



**FIGURE 31-16: IDD MAXIMUM, HFINTOSC MODE, PIC16LF1847 ONLY**

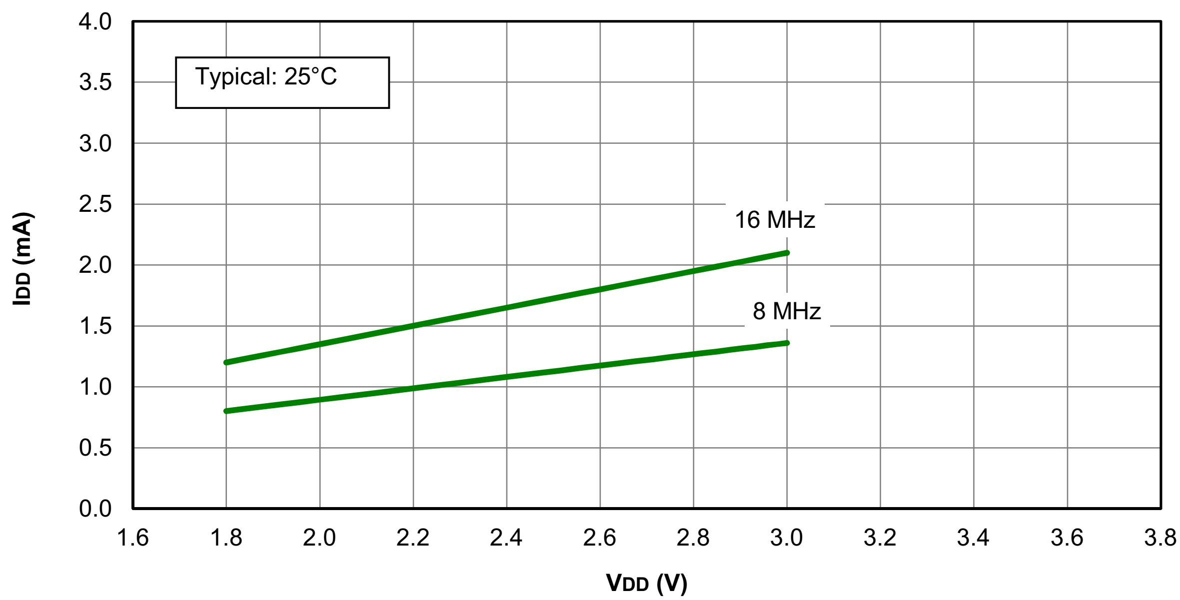


**FIGURE 31-41: VOH vs. IOH OVER TEMPERATURE, VDD = 5.0V, PIC16F1847 ONLY**

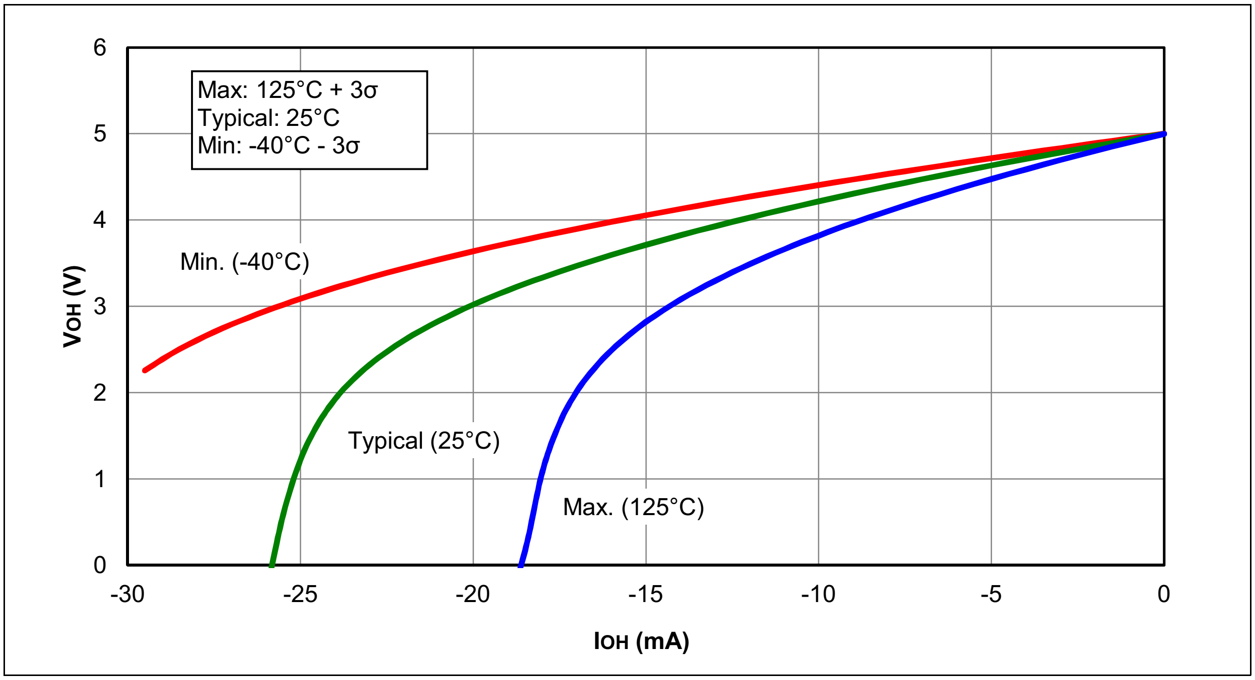



**FIGURE 31-42: VOL vs. IOL OVER TEMPERATURE, VDD = 5.0V, PIC16F1847 ONLY**

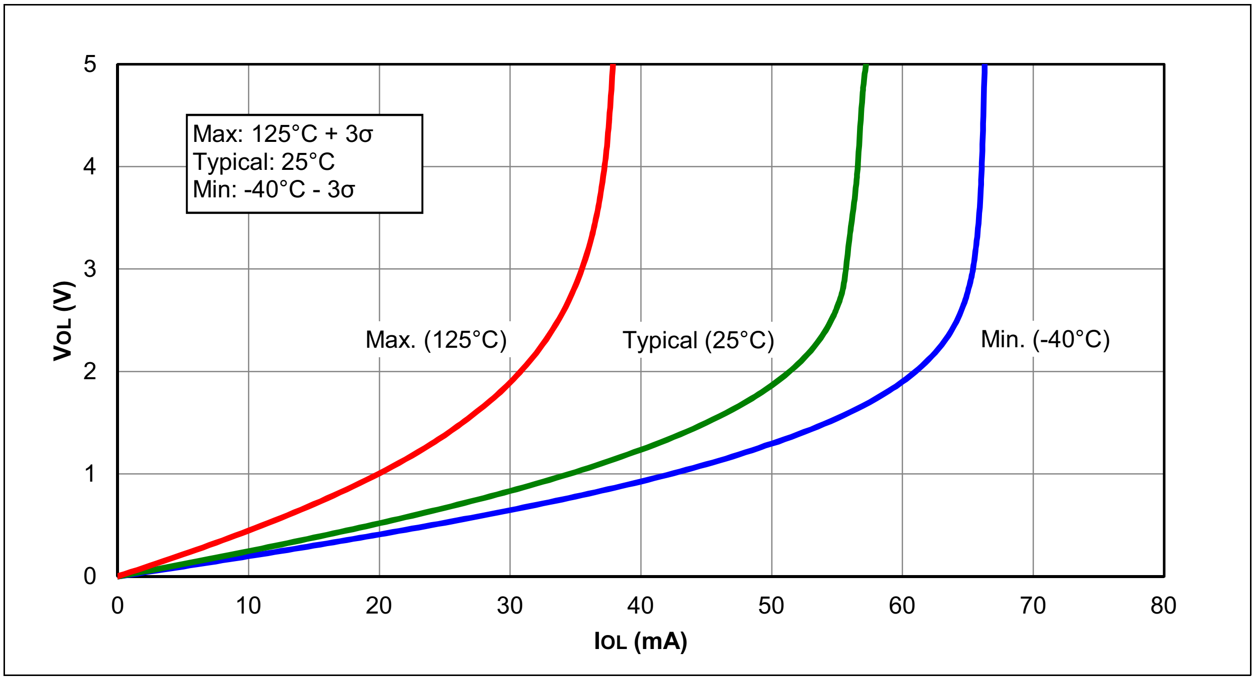


**FIGURE 31-47: POR RELEASE VOLTAGE**

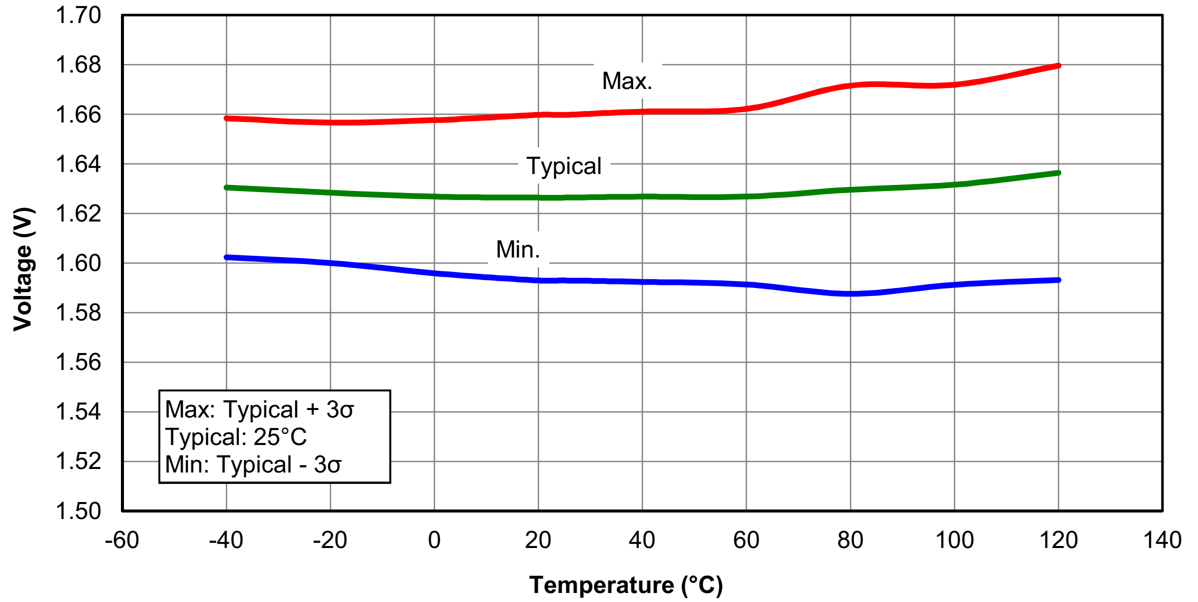



**FIGURE 31-48: POR REARM VOLTAGE, PIC16F1847 ONLY**