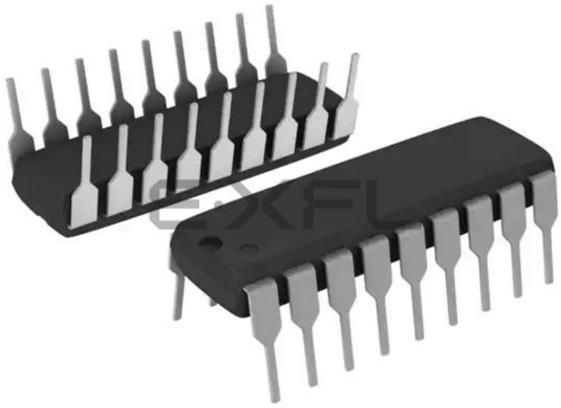


Welcome to E-XFL.COM

## What is "Embedded - Microcontrollers"?

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

## Applications of "Embedded - Microcontrollers"

| Details | |
|---|---|
| Product Status | Active |
| Core Processor | PIC |
| Core Size | 8-Bit |
| Speed | 32MHz |
| Connectivity | I²C, LINbus, SPI, UART/USART |
| Peripherals | Brown-out Detect/Reset, POR, PWM, WDT |
| Number of I/O | 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 | Through Hole |
| Package / Case | 18-DIP (0.300", 7.62mm) |
| Supplier Device Package | 18-PDIP |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic16f1847-i-p |

## 7.4 MCLR

The MCLR is an optional external input that can reset the device. The MCLR function is controlled by the MCLRE bit of Configuration Words and the LVP bit of Configuration Words (Table 7-2).

**TABLE 7-2: MCLR CONFIGURATION**

| MCLRE | LVP | MCLR |
|---|---|---|
| 0 | 0 | Disabled |
| 1 | 0 | Enabled |
| x | 1 | Enabled |

### 7.4.1 MCLR ENABLED

When MCLR is enabled and the pin is held low, the device is held in Reset. The MCLR pin is connected to VDD through an internal weak pull-up.

The device has a noise filter in the MCLR Reset path. The filter will detect and ignore small pulses.

**Note:** A Reset does not drive the MCLR pin low.

### 7.4.2 MCLR DISABLED

When MCLR is disabled, the pin functions as a general purpose input and the internal weak pull-up is under software control. See **Section 12.3 "PORTA Registers"** for more information.

## 7.5 Watchdog Timer (WDT) Reset

The Watchdog Timer generates a Reset if the firmware does not issue a `CLRWDT` instruction within the time-out period. The TO and PD bits in the STATUS register are changed to indicate the WDT Reset. See **Section 10.0 "Watchdog Timer"** for more information.

## 7.6 `RESET` Instruction

A `RESET` instruction will cause a device Reset. The RI bit in the PCON register will be set to '0'. See Table 7-4 for default conditions after a `RESET` instruction has occurred.

## 7.7 Stack Overflow/Underflow Reset

The device can reset when the Stack Overflows or Underflows. The STKOVF or STKUNF bits of the PCON register indicate the Reset condition. These Resets are enabled by setting the STVREN bit in Configuration Words. See **Section 3.5.2 "Overflow/Underflow Reset"** for more information.

## 7.8 Programming Mode Exit

Upon exit of Programming mode, the device will behave as if a POR had just occurred.

## 7.9 Power-Up Timer

The Power-up Timer optionally delays device execution after a BOR or POR event. This timer is typically used to allow VDD to stabilize before allowing the device to start running.

The Power-up Timer is controlled by the PWRTE bit of Configuration Words.

## 7.10 Start-up Sequence

Upon the release of a POR or BOR, the following must occur before the device will begin executing:

1. Power-up Timer runs to completion (if enabled).
2. Oscillator start-up timer runs to completion (if required for oscillator source).
3. MCLR must be released (if enabled).

The total time-out will vary based on oscillator configuration and Power-up Timer configuration. See **Section 5.0 "Oscillator Module (With Fail-Safe Clock Monitor)"** for more information.

The Power-up Timer and oscillator start-up timer run independently of MCLR Reset. If MCLR is kept low long enough, the Power-up Timer and oscillator start-up timer will expire. Upon bringing MCLR high, the device will begin execution immediately (see Figure 7-4). This is useful for testing purposes or to synchronize more than one device operating in parallel.

## 7.11 Determining the Cause of a Reset

Upon any Reset, multiple bits in the STATUS and PCON register are updated to indicate the cause of the Reset. Table 7-3 and Table 7-4 show the Reset conditions of these registers.

**TABLE 7-3: RESET STATUS BITS AND THEIR SIGNIFICANCE**

| STKOVF | STKUNF | $\overline{\text{RMCLR}}$ | $\overline{\text{RI}}$ | $\overline{\text{POR}}$ | $\overline{\text{BOR}}$ | $\overline{\text{TO}}$ | $\overline{\text{PD}}$ | Condition |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 0 | x | 1 | 1 | Power-on Reset |
| 0 | 0 | 1 | 1 | 0 | x | 0 | x | Illegal, $\overline{\text{TO}}$ is set on $\overline{\text{POR}}$ |
| 0 | 0 | 1 | 1 | 0 | x | x | 0 | Illegal, $\overline{\text{PD}}$ is set on $\overline{\text{POR}}$ |
| 0 | 0 | 1 | 1 | u | 0 | 1 | 1 | Brown-out Reset |
| u | u | u | u | u | u | 0 | u | WDT Reset |
| u | u | u | u | u | u | 0 | 0 | WDT Wake-up from Sleep |
| u | u | u | u | u | u | 1 | 0 | Interrupt Wake-up from Sleep |
| u | u | 0 | u | u | u | u | u | $\overline{\text{MCLR}}$ Reset during normal operation |
| u | u | 0 | u | u | u | 1 | 0 | $\overline{\text{MCLR}}$ Reset during Sleep |
| u | u | u | 0 | u | u | u | u | RESET Instruction Executed |
| 1 | u | u | u | u | u | u | u | Stack Overflow Reset (STVREN = 1) |
| u | 1 | u | u | u | u | u | u | Stack Underflow Reset (STVREN = 1) |

**TABLE 7-4: RESET CONDITION FOR SPECIAL REGISTERS(2)**

| Condition | Program Counter | STATUS Register | PCON Register |
|---|---|---|---|
| Power-on Reset | 0000h | ---1 1000 | 00-- 110x |
| $\overline{\text{MCLR}}$ Reset during normal operation | 0000h | ---u uuuu | uu-- 0uuu |
| $\overline{\text{MCLR}}$ Reset during Sleep | 0000h | ---1 0uuu | uu-- 0uuu |
| WDT Reset | 0000h | ---0 uuuu | uu-- uuuu |
| WDT Wake-up from Sleep | PC + 1 | ---0 0uuu | uu-- uuuu |
| Brown-out Reset | 0000h | ---1 1uuu | 00-- 11u0 |
| Interrupt Wake-up from Sleep | PC + 1(1) | ---1 0uuu | uu-- uuuu |
| RESET Instruction Executed | 0000h | ---u uuuu | uu-- u0uu |
| Stack Overflow Reset (STVREN = 1) | 0000h | ---u uuuu | 1u-- uuuu |
| Stack Underflow Reset (STVREN = 1) | 0000h | ---u uuuu | u1-- uuuu |

**Legend:** u = unchanged, x = unknown, - = unimplemented bit, reads as '0'.

**Note 1:** When the wake-up is due to an interrupt and Global Enable bit (GIE) is set, the return address is pushed on the stack and PC is loaded with the interrupt vector (0004h) after execution of PC + 1.

**2:** If a Status bit is not implemented, that bit will be read as '0'.

## 7.12 Power Control (PCON) Register

The Power Control (PCON) register contains flag bits to differentiate between a:

- Power-on Reset ($\overline{\text{POR}}$)
- Brown-out Reset ($\overline{\text{BOR}}$)
- Reset Instruction Reset ($\overline{\text{RI}}$)
- Stack Overflow Reset (STKOVF)
- Stack Underflow Reset (STKUNF)
- $\overline{\text{MCLR}}$ Reset ($\overline{\text{RMCLR}}$)

The PCON register bits are shown in Register 7-2.

## 7.13 Register Definitions: Power Control

### REGISTER 7-2: PCON: POWER CONTROL REGISTER

| R/W/HS-0/q | R/W/HS-0/q | U-0 | U-0 | R/W/HC-1/q | R/W/HC-1/q | R/W/HC-q/u | R/W/HC-q/u |
|---|---|---|---|---|---|---|---|
| STKOVF | STKUNF | — | — | $\overline{\text{RMCLR}}$ | $\overline{\text{RI}}$ | $\overline{\text{POR}}$ | $\overline{\text{BOR}}$ |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| HC = Bit is cleared by hardware | | HS = Bit is set by hardware |
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ |
| u = Bit is unchanged | x = Bit is unknown | -m/n = Value at POR and BOR/Value at all other Resets |
| ‘1’ = Bit is set | ‘0’ = Bit is cleared | q = Value depends on condition |

bit 7 **STKOVF:** Stack Overflow Flag bit
1 = A Stack Overflow occurred
0 = A Stack Overflow has not occurred or set to ‘0’ by firmware

bit 6 **STKUNF:** Stack Underflow Flag bit
1 = A Stack Underflow occurred
0 = A Stack Underflow has not occurred or set to ‘0’ by firmware

bit 5-4 **Unimplemented:** Read as ‘0’

bit 3 **$\overline{\text{RMCLR}}$:** $\overline{\text{MCLR}}$ Reset Flag bit
1 = A $\overline{\text{MCLR}}$ Reset has not occurred or set to ‘1’ by firmware
0 = A $\overline{\text{MCLR}}$ Reset has occurred (set to ‘0’ in hardware when a $\overline{\text{MCLR}}$ Reset occurs)

bit 2 **$\overline{\text{RI}}$:** RESET Instruction Flag bit
1 = A RESET instruction has not been executed or set to ‘1’ by firmware
0 = A RESET instruction has been executed (set to ‘0’ in hardware upon executing a RESET instruction)

bit 1 **$\overline{\text{POR}}$:** Power-on Reset Status bit
1 = No Power-on Reset occurred
0 = A Power-on Reset occurred (must be set in software after a Power-on Reset occurs)

bit 0 **$\overline{\text{BOR}}$:** Brown-out Reset Status bit
1 = No Brown-out Reset occurred
0 = A Brown-out Reset occurred (must be set in software after a Power-on Reset or Brown-out Reset occurs)

**NOTES:**

**REGISTER 8-4: PIE3: PERIPHERAL INTERRUPT ENABLE REGISTER 3**

| U-0 | U-0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | U-0 | R/W-0/0 | U-0 |
|---|---|---|---|---|---|---|---|
| — | — | CCP4IE | CCP3IE | TMR6IE | — | TMR4IE | — |
| 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 7-6 **Unimplemented:** Read as ‘0’

bit 5 **CCP4IE:** CCP4 Interrupt Enable bit
1 = Enables the CCP4 interrupt
0 = Disables the CCP4 interrupt

bit 4 **CCP3IE:** CCP3 Interrupt Enable bit
1 = Enables the CCP3 interrupt
0 = Disables the CCP3 interrupt

bit 3 **TMR6IE:** TMR6 to PR6 Match Interrupt Enable bit
1 = Enables the TMR6 to PR6 Match interrupt
0 = Disables the TMR6 to PR6 Match interrupt

bit 2 **Unimplemented:** Read as ‘0’

bit 1 **TMR4IE:** TMR4 to PR4 Match Interrupt Enable bit
1 = Enables the TMR4 to PR4 Match interrupt
0 = Disables the TMR4 to PR4 Match interrupt

bit 0 **Unimplemented:** Read as ‘0’

**Note 1:** Bit PEIE of the INTCON register must be set to enable any peripheral interrupt.

### 16.1.5 INTERRUPTS

The ADC module allows for the ability to generate an interrupt upon completion of an Analog-to-Digital conversion. The ADC Interrupt Flag is the ADIF bit in the PIR1 register. The ADC Interrupt Enable is the ADIE bit in the PIE1 register. The ADIF bit must be cleared in software.

> **Note 1:** The ADIF bit is set at the completion of every conversion, regardless of whether or not the ADC interrupt is enabled.
>
> **2:** The ADC operates during Sleep only when the FRC oscillator is selected.

This interrupt can be generated while the device is operating or while in Sleep. If the device is in Sleep, the interrupt will wake-up the device. Upon waking from Sleep, the next instruction following the `SLEEP` instruction is always executed. If the user is attempting to wake-up from Sleep and resume in-line code execution, the GIE and PEIE bits of the INTCON register must be disabled. If the GIE and PEIE bits of the INTCON register are enabled, execution will switch to the Interrupt Service Routine.

Please refer to **Section 16.1.5 "Interrupts"** for more information.

### 16.1.6 RESULT FORMATTING

The 10-bit ADC conversion result can be supplied in two formats, left justified or right justified. The ADFM bit of the ADCON1 register controls the output format.

Figure 16-3 shows the two output formats.

**FIGURE 16-3: 10-BIT ADC CONVERSION RESULT FORMAT**

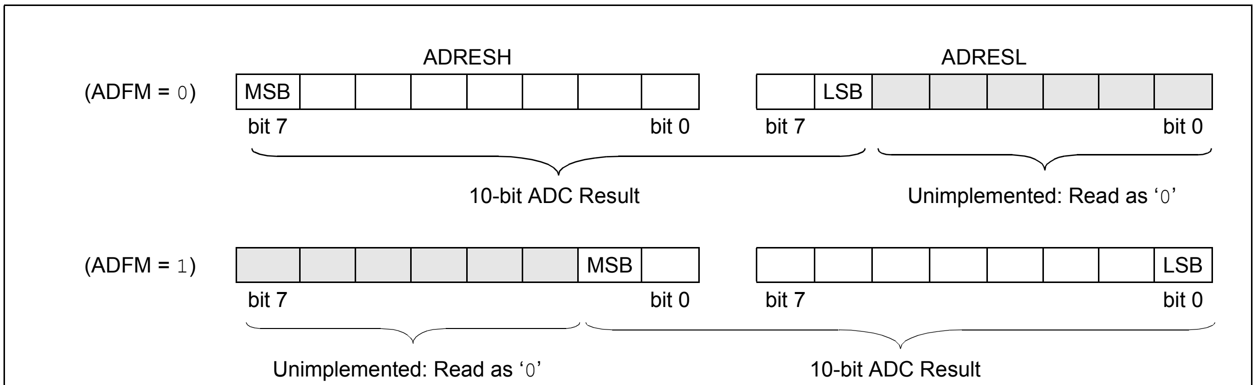
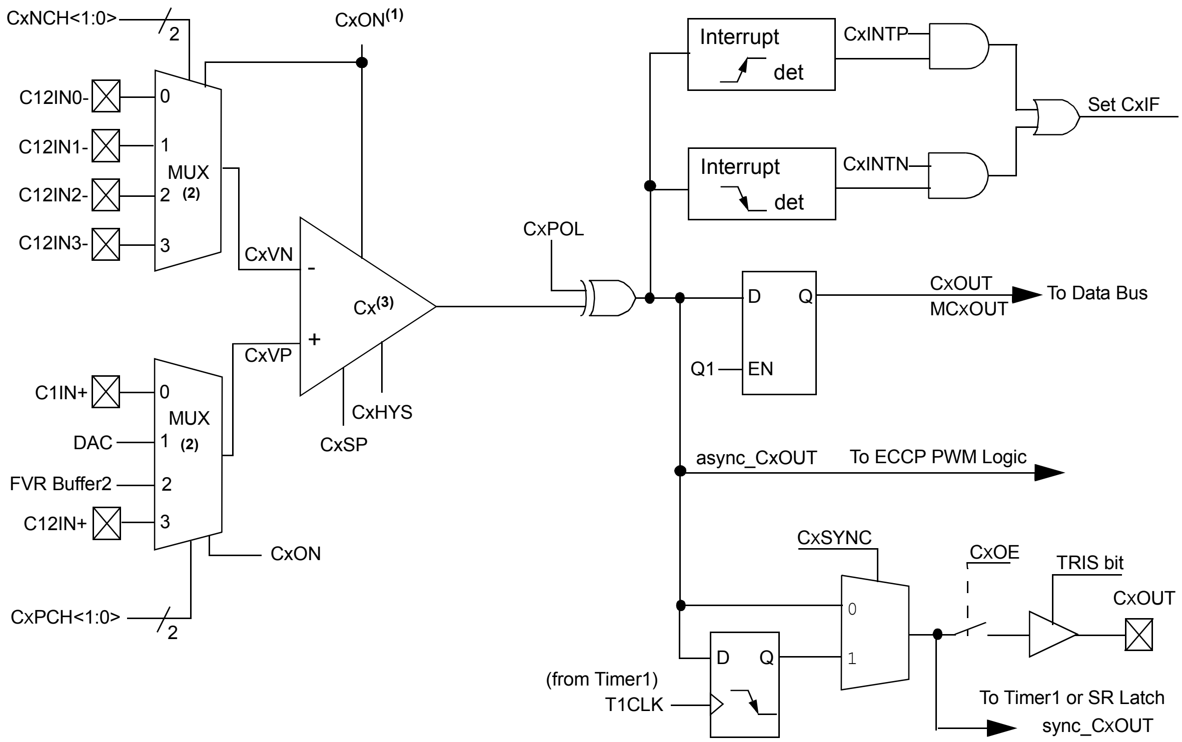


**FIGURE 19-2: COMPARATOR 1 MODULE SIMPLIFIED BLOCK DIAGRAM**




**Note 1:** When CxON = 0, the Comparator will produce a '0' at the output
**2:** When CxON = 0, all multiplexer inputs are disconnected.
**3:** Output of comparator can be frozen during debugging.

### 19.7 Comparator Negative Input Selection

The CxNCH<1:0> bits of the CMxCON0 register direct one of four analog pins to the comparator inverting input.

| | |
|---|---|
| **Note:** | To use CxIN+ and CxINx- pins as analog input, the appropriate bits must be set in the ANSEL register and the corresponding TRIS bits must also be set to disable the output drivers. |

### 19.8 Comparator Response Time

The comparator output is indeterminate for a period of time after the change of an input source or the selection of a new reference voltage. This period is referred to as the response time. The response time of the comparator differs from the settling time of the voltage reference. Therefore, both of these times must be considered when determining the total response time to a comparator input change. See the Comparator and Voltage Reference Specifications in **Section 30.0 "Electrical Specifications"** for more details.

### 19.9 Interaction with ECCP Logic

The C1 and C2 comparators can be used as general purpose comparators. Their outputs can be brought out to the C1OUT and C2OUT pins. When the ECCP Auto-Shutdown is active it can use one or both comparator signals. If auto-restart is also enabled, the comparators can be configured as a closed loop analog feedback to the ECCP, thereby, creating an analog controlled PWM.

| | |
|---|---|
| **Note:** | When the comparator module is first initialized the output state is unknown. Upon initialization, the user should verify the output state of the comparator prior to relying on the result, primarily when using the result in connection with other peripheral features, such as the ECCP Auto-Shutdown mode. |

### 19.10 Analog Input Connection Considerations

A simplified circuit for an analog input is shown in Figure 19-1. Since the analog input pins share their connection with a digital input, they have reverse biased ESD protection diodes to VDD and VSS. The analog input, therefore, must be between VSS and VDD. If the input voltage deviates from this range by more than 0.6V in either direction, one of the diodes is forward biased and a latch-up may occur.

A maximum source impedance of 10 kΩ is recommended for the analog sources. Also, any external component connected to an analog input pin, such as a capacitor or a Zener diode, should have very little leakage current to minimize inaccuracies introduced.

| | |
|---|---|
| **Note 1:** | When reading a PORT register, all pins configured as analog inputs will read as a '0'. Pins configured as digital inputs will convert as an analog input, according to the input specification. |
| **2:** | Analog levels on any pin defined as a digital input, may cause the input buffer to consume more current than is specified. |

## 21.11 Timer1 Control Register

The Timer1 Control register (T1CON), shown in Register 21-1, is used to control Timer1 and select the various features of the Timer1 module.

### REGISTER 21-1: T1CON: TIMER1 CONTROL REGISTER

| R/W-0/u | R/W-0/u | R/W-0/u | R/W-0/u | R/W-0/u | R/W-0/u | U-0 | R/W-0/u |
|---|---|---|---|---|---|---|---|
| TMR1CS<1:0> | | T1CKPS<1:0> | | T1OSCEN | $\overline{\text{T1SYNC}}$ | — | TMR1ON |
| 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 7-6 **TMR1CS<1:0>:** Timer1 Clock Source Select bits
11 = Timer1 clock source is Capacitive Sensing Oscillator (CAPOSC)
10 = Timer1 clock source is pin or oscillator:
If T1OSCEN = 0:
External clock from T1CKI pin (on the rising edge)
If T1OSCEN = 1:
Crystal oscillator on T1OSI/T1OSO pins
01 = Timer1 clock source is system clock (FOSC)
00 = Timer1 clock source is instruction clock (FOSC/4)

bit 5-4 **T1CKPS<1:0>:** Timer1 Input Clock Prescale Select bits
11 = 1:8 Prescale value
10 = 1:4 Prescale value
01 = 1:2 Prescale value
00 = 1:1 Prescale value

bit 3 **T1OSCEN:** LP Oscillator Enable Control bit
1 = Dedicated Timer1 oscillator circuit enabled
0 = Dedicated Timer1 oscillator circuit disabled

bit 2 **$\overline{\text{T1SYNC}}$:** Timer1 External Clock Input Synchronization Control bit
TMR1CS<1:0> = 1X
1 = Do not synchronize external clock input
0 = Synchronize external clock input with system clock (FOSC)

TMR1CS<1:0> = 0X
This bit is ignored. Timer1 uses the internal clock when TMR1CS<1:0> = 1X.

bit 1 **Unimplemented:** Read as ‘0’

bit 0 **TMR1ON:** Timer1 On bit
1 = Enables Timer1
0 = Stops Timer1
Clears Timer1 Gate flip-flop

**NOTES:**

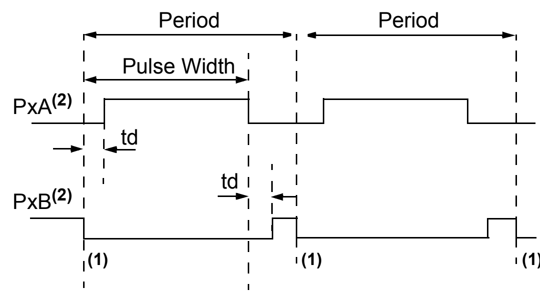
### 24.4.1 HALF-BRIDGE MODE

In Half-Bridge mode, two pins are used as outputs to drive push-pull loads. The PWM output signal is output on the CCPx/PxA pin, while the complementary PWM output signal is output on the PxB pin (see Figure 24-9). This mode can be used for Half-Bridge applications, as shown in Figure 24-9, or for Full-Bridge applications, where four power switches are being modulated with two PWM signals.

In Half-Bridge mode, the programmable dead-band delay can be used to prevent shoot-through current in Half-Bridge power devices. The value of the PDC<6:0> bits of the PWMxCON register sets the number of instruction cycles before the output is driven active. If the value is greater than the duty cycle, the corresponding output remains inactive during the entire cycle. See **Section 24.4.5 "Programmable Dead-Band Delay Mode"** for more details of the dead-band delay operations.

Since the PxA and PxB outputs are multiplexed with the PORT data latches, the associated TRIS bits must be cleared to configure PxA and PxB as outputs.

**FIGURE 24-8: EXAMPLE OF HALF-BRIDGE PWM OUTPUT**




td = Dead-Band Delay

**Note 1:** At this time, the TMRx register is equal to the PRx register.

**2:** Output signals are shown as active-high.

**FIGURE 24-9: EXAMPLE OF HALF-BRIDGE APPLICATIONS**

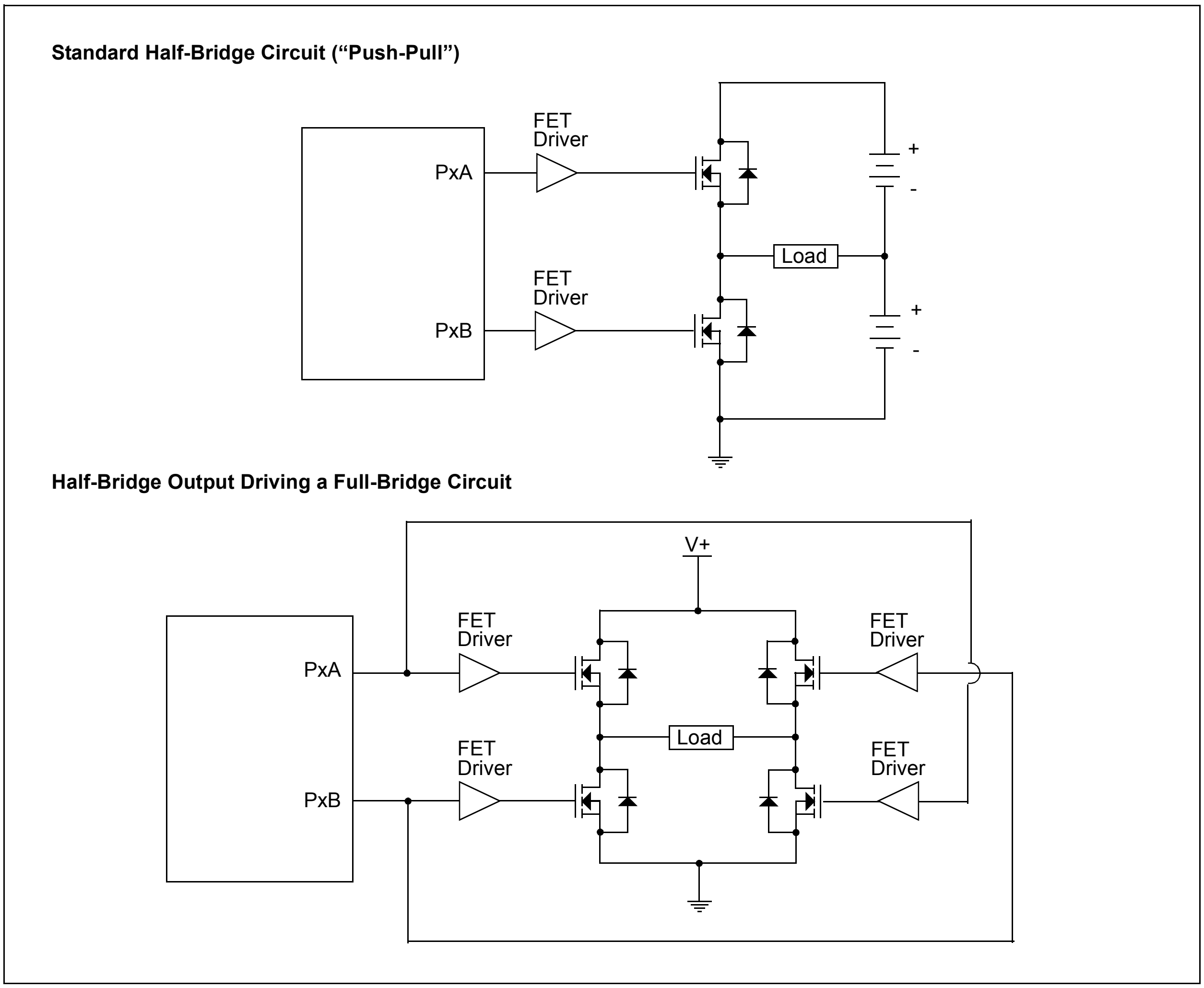
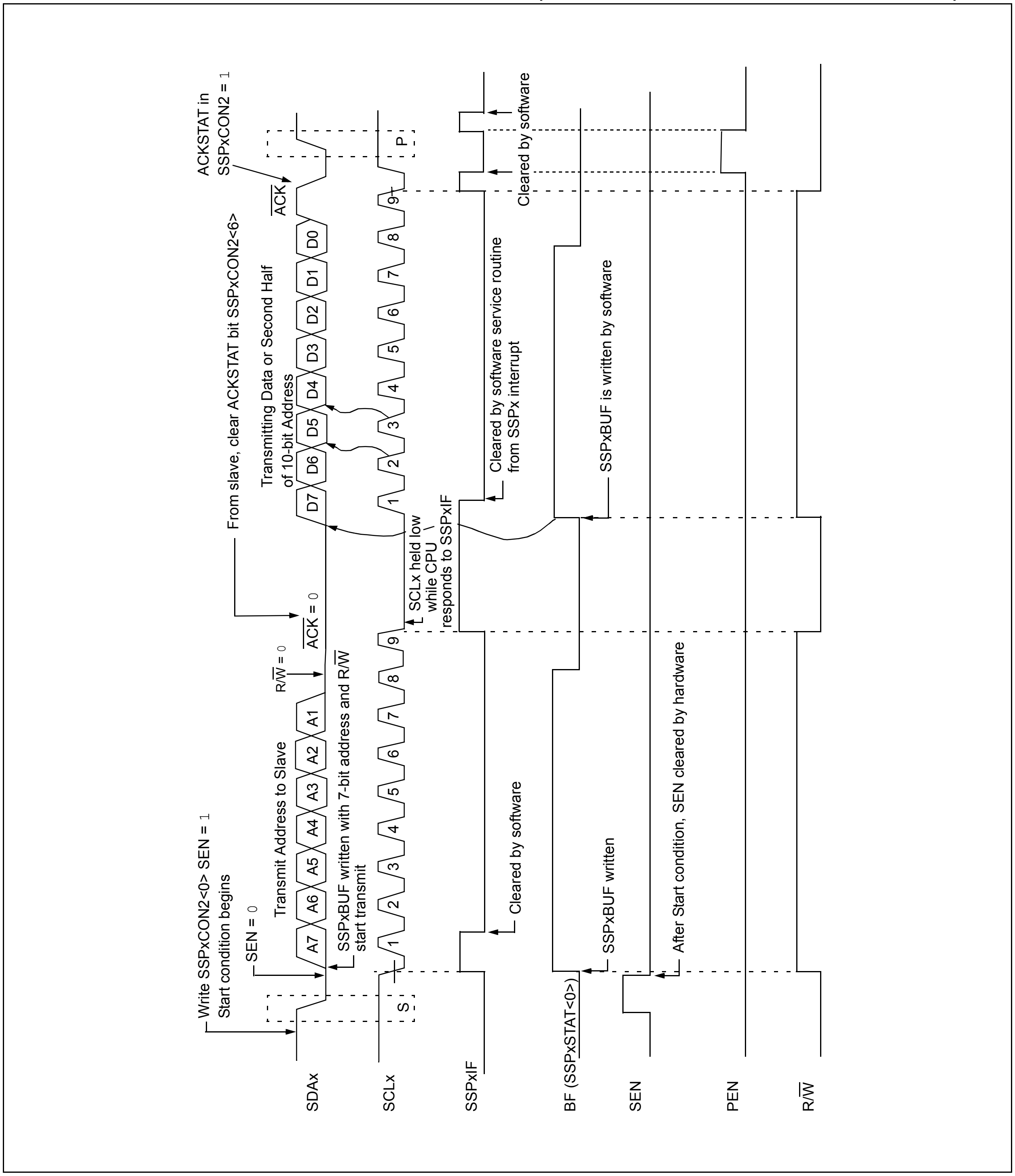


**FIGURE 25-28: I²C MASTER MODE WAVEFORM (TRANSMISSION, 7 OR 10-BIT ADDRESS)**



### REGISTER 25-3: SSPxCON2: SSPx CONTROL REGISTER 2

| R/W-0/0 | R-0/0 | R/W-0/0 | R/S/HS-0/0 | R/S/HS-0/0 | R/S/HS-0/0 | R/S/HS-0/0 | R/W/HS-0/0 |
|---|---|---|---|---|---|---|---|
| GCEN | ACKSTAT | ACKDT | ACKEN | RCEN | PEN | RSEN | SEN |
| 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 | HC = Cleared by hardware S = User set |

bit 7 **GCEN:** General Call Enable bit (in $I^2C$ Slave mode only)
1 = Enable interrupt when a general call address (0x00 or 00h) is received in the SSPxSR
0 = General call address disabled

bit 6 **ACKSTAT:** Acknowledge Status bit (in $I^2C$ mode only)
1 = Acknowledge was not received
0 = Acknowledge was received

bit 5 **ACKDT:** Acknowledge Data bit (in $I^2C$ mode only)
In Receive mode:
Value transmitted when the user initiates an Acknowledge sequence at the end of a receive
1 = Not Acknowledge
0 = Acknowledge

bit 4 **ACKEN:** Acknowledge Sequence Enable bit (in $I^2C$ Master mode only)
In Master Receive mode:
1 = Initiate Acknowledge sequence on SDAx and SCLx pins, and transmit ACKDT data bit. Automatically cleared by hardware.
0 = Acknowledge sequence idle

bit 3 **RCEN:** Receive Enable bit (in $I^2C$ Master mode only)
1 = Enables Receive mode for $I^2C$
0 = Receive idle

bit 2 **PEN:** Stop Condition Enable bit (in $I^2C$ Master mode only)
SCKx Release Control:
1 = Initiate Stop condition on SDAx and SCLx pins. Automatically cleared by hardware.
0 = Stop condition Idle

bit 1 **RSEN:** Repeated Start Condition Enable bit (in $I^2C$ Master mode only)
1 = Initiate Repeated Start condition on SDAx and SCLx pins. Automatically cleared by hardware.
0 = Repeated Start condition Idle

bit 0 **SEN:** Start Condition Enable/Stretch Enable bit
In Master mode:
1 = Initiate Start condition on SDAx and SCLx pins. Automatically cleared by hardware.
0 = Start condition Idle
In Slave mode:
1 = Clock stretching is enabled for both slave transmit and slave receive (stretch enabled)
0 = Clock stretching is disabled

**Note 1:** For bits ACKEN, RCEN, PEN, RSEN, SEN: If the $I^2C$ module is not in the Idle mode, this bit may not be set (no spooling) and the SSPxBUF may not be written (or writes to the SSPxBUF are disabled).

**REGISTER 26-2: RCSTA: RECEIVE STATUS AND CONTROL REGISTER**

| R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R-0/0 | R-0/0 | R-x/x |
|---|---|---|---|---|---|---|---|
| SPEN | RX9 | SREN(1) | CREN(1) | ADDEN | FERR | OERR | RX9D |
| 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 **SPEN:** Serial Port Enable bit
1 = Serial port enabled (configures RX/DT and TX/CK pins as serial port pins)
0 = Serial port disabled (held in Reset)

bit 6 **RX9:** 9-bit Receive Enable bit
1 = Selects 9-bit reception
0 = Selects 8-bit reception

bit 5 **SREN:** Single Receive Enable bit
Asynchronous mode:
Don’t care
Synchronous mode – Master:
1 = Enables single receive
0 = Disables single receive
This bit is cleared after reception is complete.
Synchronous mode – Slave
Don’t care

bit 4 **CREN:** Continuous Receive Enable bit
Asynchronous mode:
1 = Enables receiver
0 = Disables receiver
Synchronous mode:
1 = Enables continuous receive until enable bit CREN is cleared (CREN overrides SREN)
0 = Disables continuous receive

bit 3 **ADDEN:** Address Detect Enable bit
Asynchronous mode 9-bit (RX9 = 1):
1 = Enables address detection, enable interrupt and load the receive buffer when RSR<8> is set
0 = Disables address detection, all bytes are received and 9th bit can be used as parity bit
Asynchronous mode 8-bit (RX9 = 0):
Don’t care

bit 2 **FERR:** Framing Error bit
1 = Framing error (can be updated by reading RCREG register and receive next valid byte)
0 = No framing error

bit 1 **OERR:** Overrun Error bit
1 = Overrun error (can be cleared by clearing bit CREN)
0 = No overrun error

bit 0 **RX9D:** Ninth bit of Received Data
This can be address/data bit or a parity bit and must be calculated by user firmware.

**Note 1:** SREN/CREN overrides TXEN in Sync mode.

### MOVWI — Move W to INDFn

| | |
|---|---|
| Syntax: | [ *label* ] MOVWI ++FSRn<br>[ *label* ] MOVWI --FSRn<br>[ *label* ] MOVWI FSRn++<br>[ *label* ] MOVWI FSRn--<br>[ *label* ] MOVWI k[FSRn] |
| Operands: | $n \in [0,1]$<br>$mm \in [00,01, 10, 11]$<br>$-32 \leq k \leq 31$ |
| Operation: | W → INDFn<br>Effective address is determined by<br>• FSR + 1 (preincrement)<br>• FSR - 1 (predecrement)<br>• FSR + k (relative offset)<br>After the Move, the FSR value will be either:<br>• FSR + 1 (all increments)<br>• FSR - 1 (all decrements)<br>Unchanged |
| Status Affected: | None |

| Mode | Syntax | mm |
|---|---|---|
| Preincrement | ++FSRn | 00 |
| Predecrement | --FSRn | 01 |
| Postincrement | FSRn++ | 10 |
| Postdecrement | FSRn-- | 11 |

Description: This instruction is used to move data between W and one of the indirect registers (INDFn). Before/after this move, the pointer (FSRn) is updated by pre/post incrementing/decrementing it.

Note: The INDFn registers are not physical registers. Any instruction that accesses an INDFn register actually accesses the register at the address specified by the FSRn.

FSRn is limited to the range 0000h - FFFFh. Incrementing/decrementing it beyond these bounds will cause it to wrap-around.

The increment/decrement operation on FSRn WILL NOT affect any Status bits.

### NOP — No Operation

| | |
|---|---|
| Syntax: | [ *label* ] NOP |
| Operands: | None |
| Operation: | No operation |
| Status Affected: | None |
| Description: | No operation. |
| Words: | 1 |
| Cycles: | 1 |
| <u>Example:</u> | `NOP` |

### OPTION — Load OPTION_REG Register with W

| | |
|---|---|
| Syntax: | [ *label* ] OPTION |
| Operands: | None |
| Operation: | (W) → OPTION_REG |
| Status Affected: | None |
| Description: | Move data from W register to OPTION_REG register. |
| Words: | 1 |
| Cycles: | 1 |
| <u>Example:</u> | `OPTION`<br>Before Instruction<br>OPTION_REG = 0xFF<br>W = 0x4F<br>After Instruction<br>OPTION_REG = 0x4F<br>W = 0x4F |

### RESET — Software Reset

| | |
|---|---|
| Syntax: | [ *label* ] RESET |
| Operands: | None |
| Operation: | Execute a device Reset. Resets the nRI flag of the PCON register. |
| Status Affected: | None |
| Description: | This instruction provides a way to execute a hardware Reset by software. |

**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; $\overline{\text{MCLR}}$ = 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.

## 31.0 DC AND AC CHARACTERISTICS GRAPHS AND CHARTS

The graphs and tables provided in this section are for **design guidance** and are **not tested**.

In some graphs or tables, the data presented are **outside specified operating range** (i.e., outside specified $V_{DD}$ range). This is for **information only** and devices are ensured to operate properly only within the specified range.

| | |
|---|---|
| **Note:** | The graphs and tables provided following this note are a statistical summary based on a limited number of samples and are provided for informational purposes only. The performance characteristics listed herein are not tested or guaranteed. In some graphs or tables, the data presented may be outside the specified operating range (e.g., outside specified power supply range) and therefore, outside the warranted range. |

**“Typical” represents the mean of the distribution at 25°C. “MAXIMUM”, “Max.”, “MINIMUM” or “Min.” represents (mean + 3$\sigma$) or (mean - 3$\sigma$) respectively, where $\sigma$ is a standard deviation, over each temperature range.**

**FIGURE 31-13: IDD, MFINTOSC MODE, FOSC = 500 kHz, PIC16LF1847 ONLY**

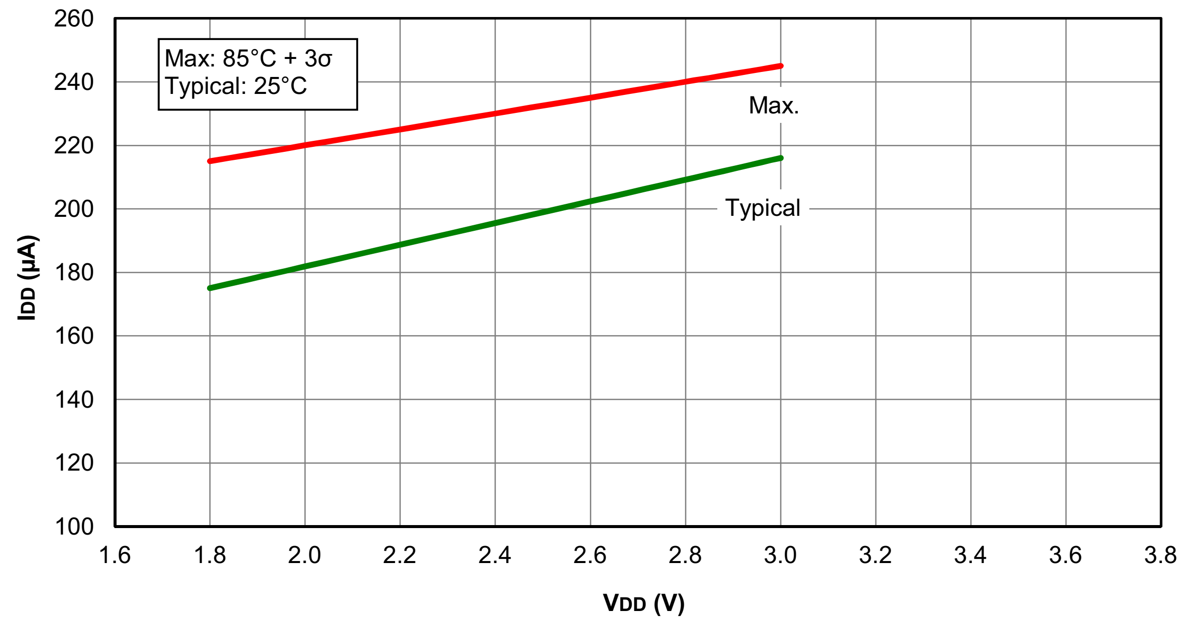



**FIGURE 31-14: IDD, MFINTOSC MODE, FOSC = 500 kHz, PIC16F1847 ONLY**

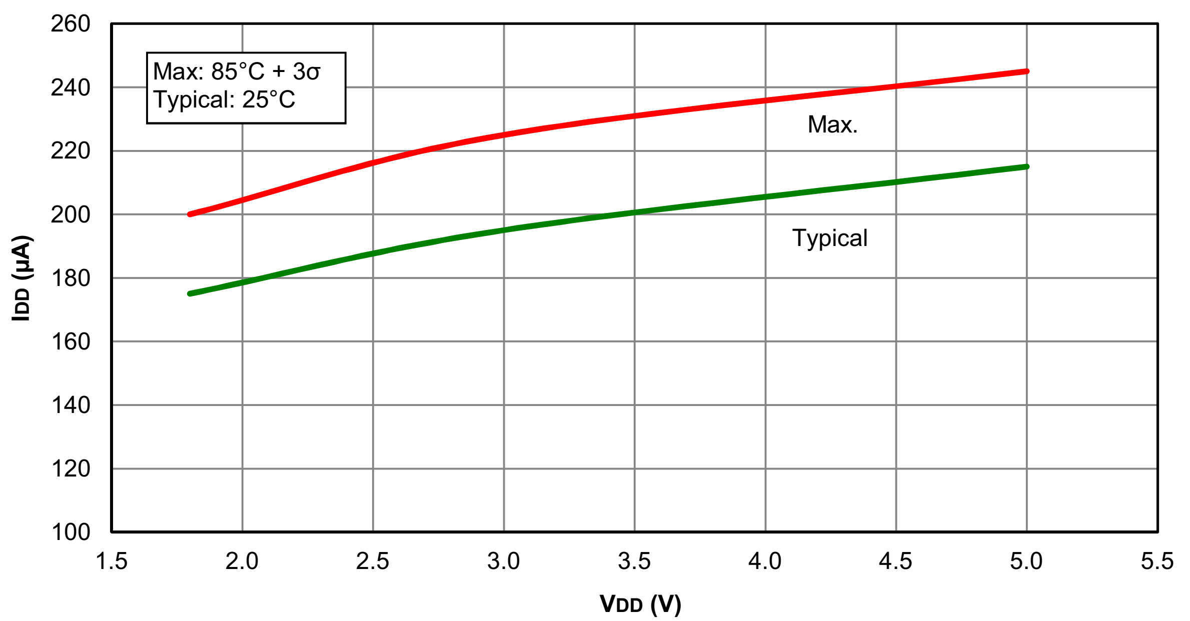
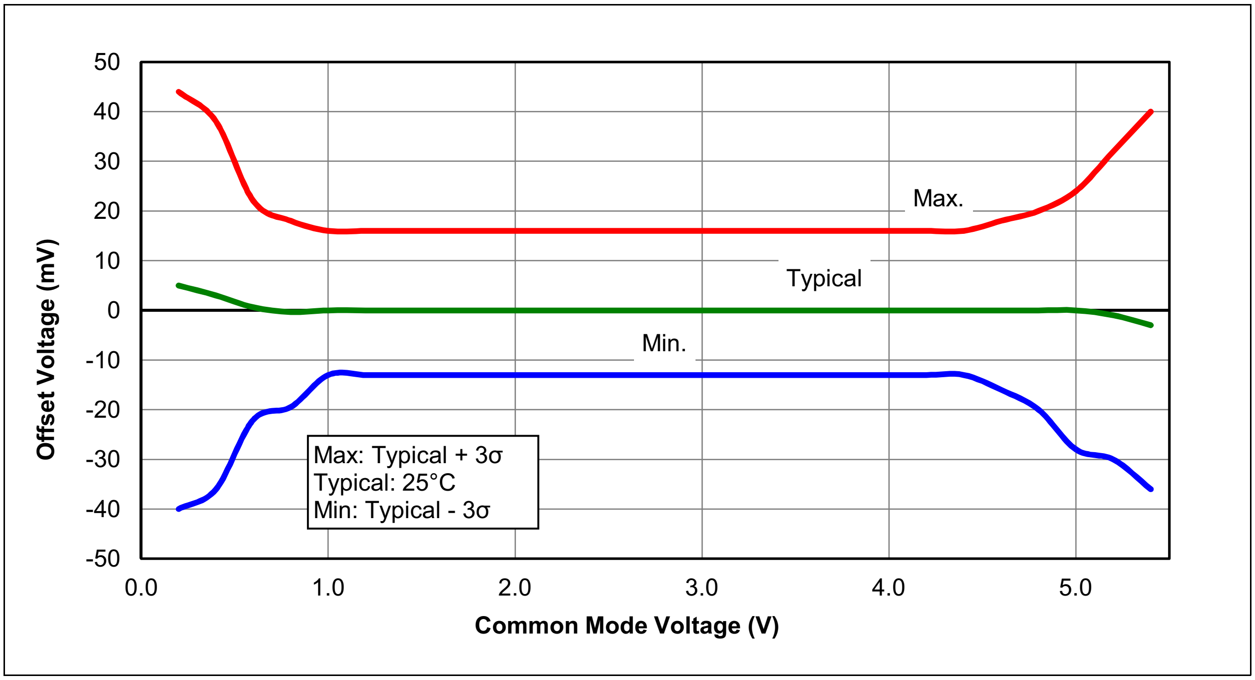


**FIGURE 31-59: COMPARATOR INPUT OFFSET AT 25°C, NORMAL-POWER MODE, CxSP = 1, PIC16F1847 ONLY**



**NOTES:**