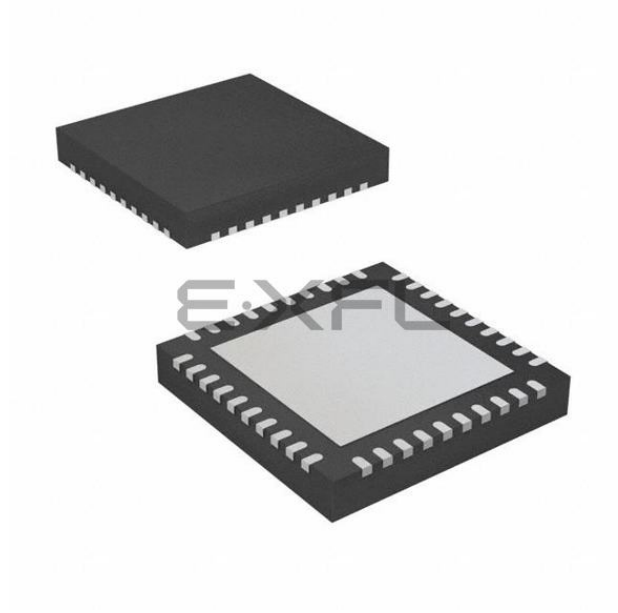


**Welcome to E-XFL.COM**

## 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 | 20MHz |
| Connectivity | I²C, SPI, UART/USART |
| Peripherals | Brown-out Detect/Reset, POR, PWM, WDT |
| Number of I/O | 36 |
| Program Memory Size | 14KB (8K x 14) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 363 x 8 |
| Voltage - Supply (Vcc/Vdd) | 1.8V ~ 5.5V |
| Data Converters | A/D 14x8b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 40-UFQFN Exposed Pad |
| Supplier Device Package | 40-UQFN (5x5) |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic16f707-i-mv |

## TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at **docerrors@microchip.com** or fax the **Reader Response Form** in the back of this data sheet to (480) 792-4150. We welcome your feedback.

### Most Current Data Sheet

To obtain the most up-to-date version of this data sheet, please register at our Worldwide Website at:

**http://www.microchip.com**

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number, (e.g., DS30000000A is version A of document DS30000000).

### Errata

An errata sheet, describing minor operational differences from the data sheet and recommended workarounds, may exist for current devices. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

- Microchip's Worldwide Website; **http://www.microchip.com**
- Your local Microchip sales office (see last page)

When contacting a sales office, please specify which device, revision of silicon and data sheet (include literature number) you are using.

### Customer Notification System

Register on our website at **www.microchip.com** to receive the most current information on all of our products.

## 1.0 DEVICE OVERVIEW

The PIC16(L)F707 devices are covered by this data sheet. They are available in 40/44-pin packages. Figure 1-1 shows a block diagram of the PIC16(L)F707 devices. Table 1-1 shows the pinout descriptions.

#### 6.2.2.4 RA3/AN3/VREF+/CPSA2

The RA3 pin is configurable to function as one of the following:

- General purpose I/O
- Analog input for the A/D
- Voltage Reference input for the A/D
- Capacitive sensing input

#### 6.2.2.5 RA4/CPSA3/T0CKI/TACKI

The RA4 pin is configurable to function as one of the following:

- General purpose I/O
- Capacitive sensing input
- Clock input for Timer0
- Clock input for TimerA

The Timer0 clock input function works independently of any TRIS register setting. Effectively, if TRISA4 = 0, the PORTA4 register bit will output to the pad and clock Timer0 at the same time.

#### 6.2.2.6 RA5/AN4/CPSA4/$\overline{SS}$/VCAP

The RA5 pin is configurable to function as one of the following:

- General purpose I/O
- Capacitive sensing input
- Analog input for the A/D
- Slave Select input for the SSP(1)
- Voltage Regulator Capacitor pin (PIC16F707 only)

**Note 1:** $\overline{SS}$ pin location may be selected as RA5 or RA0.

#### 6.2.2.7 RA6/CPSB1/OSC2/CLKOUT/VCAP

The RA6 pin is configurable to function as one of the following:

- General purpose I/O
- Crystal/resonator connection
- Clock Output
- Voltage Regulator Capacitor pin (PIC16F707 only)
- Capacitive sensing input

#### 6.2.2.8 RA7/CPSB0/OSC1/CLKIN

The RA7 pin is configurable to function as one of the following:

- General purpose I/O
- Crystal/resonator connection
- Clock Input
- Capacitive sensing input.

**TABLE 6-1: SUMMARY OF REGISTERS ASSOCIATED WITH PORTA**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOR | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|
| ADCON0 | — | — | CHS3 | CHS2 | CHS1 | CHS0 | GO/$\overline{DONE}$ | ADON | --00 0000 | --00 0000 |
| ADCON1 | — | ADCS2 | ADCS1 | ADCS0 | — | — | ADREF1 | ADREF0 | -000 --00 | -000 --00 |
| ANSELA | ANSA7 | ANSA6 | ANSA5 | ANSA4 | ANSA3 | ANSA2 | ANSA1 | ANSA0 | 1111 1111 | 1111 1111 |
| APFCON | — | — | — | — | — | — | SSSEL | CCP2SEL | ---- --00 | ---- --00 |
| CPSACON0 | CPSAON | CPSARM | — | — | CPSARNG1 | CPSARNG0 | CPSAOUT | TAXCS | 00-- 0000 | 00-- 0000 |
| CPSACON1 | — | — | — | — | CPSACH3 | CPSACH2 | CPSACH1 | CPSACH0 | ---- 0000 | ---- 0000 |
| CPSBCON0 | CPSBON | CPSBRM | — | — | CPSBRNG1 | CPSBRNG0 | CPSBOUT | TBXCS | 00-- 0000 | 00-- 0000 |
| CPSBCON1 | — | — | — | — | CPSBCH3 | CPSBCH2 | CPSBCH1 | CPSBCH0 | ---- 0000 | ---- 0000 |
| CONFIG2(1) | — | — | VCAPEN1 | VCAPEN0 | — | — | — | — | — | — |
| OPTION_REG | $\overline{RBPU}$ | INTEDG | TMR0CS | TMR0SE | PSA | PS2 | PS1 | PS0 | 1111 1111 | 1111 1111 |
| PORTA | RA7 | RA6 | RA5 | RA4 | RA3 | RA2 | RA1 | RA0 | xxxx xxxx | xxxx xxxx |
| SSPCON | WCOL | SSPOV | SSPEN | CKP | SSPM3 | SSPM2 | SSPM1 | SSPM0 | 0000 0000 | 0000 0000 |
| TRISA | TRISA7 | TRISA6 | TRISA5 | TRISA4 | TRISA3 | TRISA2 | TRISA1 | TRISA0 | 1111 1111 | 1111 1111 |
| TACON | TMRAON | — | TACS | TASE | TAPSA | TAPS2 | TAPS1 | TAPS0 | 0-00 0000 | 0-00 0000 |
| DACCON0 | DACEN | DACLPS | DACOE | — | DACPSS1 | DACPSS0 | — | — | 000- 00-- | 000- 00-- |

**Legend:** x = unknown, u = unchanged, – = unimplemented locations read as '0'. Shaded cells are not used by PORTA.
**Note 1:** PIC16F707 only.

## 6.3 PORTB and TRISB Registers

PORTB is an 8-bit wide, bidirectional port. The corresponding data direction register is TRISB (Register 6-6). 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., enable the output driver and put the contents of the output latch on the selected pin). Example 6-2 shows how to initialize PORTB.

Reading the PORTB register (Register 6-5) reads the status of the pins, whereas writing to it will write to the PORT latch. All write operations are read-modify-write operations. Therefore, a write to a port implies that the port pins are read, this value is modified and then written to the PORT data latch.

The TRISB register (Register 6-6) controls the PORTB pin output drivers, even when they are being used as analog inputs. The user should ensure the bits in the TRISB register are maintained set when using them as analog inputs. I/O pins configured as analog input always read '0'. Example 6-2 shows how to initialize PORTB.

**EXAMPLE 6-2: INITIALIZING PORTB**

```
BANKSEL  PORTB       ;
CLRF     PORTB       ;Init PORTB
BANKSEL  ANSELB
CLRF     ANSELB      ;Make RB<7:0> digital
BANKSEL  TRISB       ;
MOVLW    B'11110000' ;Set RB<7:4> as inputs
                     ;and RB<3:0> as outputs
MOVWF    TRISB       ;
```

| **Note:** | The ANSELB register must be initialized to configure an analog channel as a digital input. Pins configured as analog inputs will read '0'. |
|---|---|

### 6.3.1 ANSELB REGISTER

The ANSELB register (Register 6-9) is used to configure the Input mode of an I/O pin to analog. Setting the appropriate ANSELB bit high will cause all digital reads on the pin to be read as '0' and allow analog functions on the pin to operate correctly.

The state of the ANSELB bits has no affect on digital output functions. A pin with TRIS clear and ANSELB set will still operate as a digital output, but the Input mode will be analog. This can cause unexpected behavior when executing read-modify-write instructions on the affected port.

### 6.3.2 WEAK PULL-UPS

Each of the PORTB pins has an individually configurable internal weak pull-up. Control bits WPUB<7:0> enable or disable each pull-up (see Register 6-7). Each weak pull-up is automatically turned off when the port pin is configured as an output. All pull-ups are disabled on a Power-on Reset by the $\overline{\text{RBPU}}$ bit of the OPTION register.

### 6.3.3 INTERRUPT-ON-CHANGE

All of the PORTB pins are individually configurable as an interrupt-on-change pin. Control bits IOCB<7:0> enable or disable the interrupt function for each pin. Refer to Register 6-8. The interrupt-on-change feature is disabled on a Power-on Reset.

For enabled interrupt-on-change pins, the present value is compared with the old value latched on the last read of PORTB to determine which bits have changed or mismatched the old value. The 'mismatch' outputs of the last read are OR'd together to set the PORTB Change Interrupt Flag bit (RBIF) in the INTCON register.

This interrupt can wake the device from Sleep. The user, in the Interrupt Service Routine, clears the interrupt by:

a) Any read or write of PORTB. This will end the mismatch condition.
b) Clear the flag bit RBIF.

A mismatch condition will continue to set flag bit RBIF. Reading or writing PORTB will end the mismatch condition and allow flag bit RBIF to be cleared. The latch holding the last read value is not affected by a $\overline{\text{MCLR}}$ nor Brown-out Reset. After these Resets, the RBIF flag will continue to be set if a mismatch is present.

| **Note:** | When a pin change occurs at the same time as a read operation on PORTB, the RBIF flag will always be set. If multiple PORTB pins are configured for the interrupt-on-change, the user may not be able to identify which pin changed state. |
|---|---|

#### 6.3.4.4 RB3/AN9/CPSB11/CCP2

These pins are configurable to function as one of the following:

- General purpose I/O
- Analog input for the ADC
- Capacitive sensing input
- Capture 2 input, Compare 2 output, and PWM2 output

**Note:** CCP2 pin location may be selected as RB3 or RC1.

#### 6.3.4.5 RB4/AN11/CPSB12

These pins are configurable to function as one of the following:

- General purpose I/O
- Analog input for the ADC
- Capacitive sensing input

#### 6.3.4.6 RB5/AN13/CPSB13/T1G/T3CKI

These pins are configurable to function as one of the following:

- General purpose I/O
- Analog input for the ADC
- Capacitive sensing input
- Timer1 gate input
- Timer3 clock input

#### 6.3.4.7 RB6/ICSPCLK/CPSB14

These pins are configurable to function as one of the following:

- General purpose I/O
- In-Circuit Serial Programming clock
- Capacitive sensing input

#### 6.3.4.8 RB7/ICSPDAT/CPSB15

These pins are configurable to function as one of the following:

- General purpose I/O
- In-Circuit Serial Programming data
- Capacitive sensing input

**TABLE 6-2: SUMMARY OF REGISTERS ASSOCIATED WITH PORTB**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOR | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|
| ADCON0 | — | — | CHS3 | CHS2 | CHS1 | CHS0 | GO/$\overline{\text{DONE}}$ | ADON | --00 0000 | --00 0000 |
| ANSELB | ANSB7 | ANSB6 | ANSB5 | ANSB4 | ANSB3 | ANSB2 | ANSB1 | ANSB0 | 1111 1111 | 1111 1111 |
| APFCON | — | — | — | — | — | — | SSSEL | CCP2SEL | ---- --00 | ---- --00 |
| CCP2CON | — | — | DC2B1 | DC2B0 | CCP2M3 | CCP2M2 | CCP2M1 | CCP2M0 | --00 0000 | --00 0000 |
| CPSBCON0 | CPSBON | CPSBRM | — | — | CPSBRNG1 | CPSBRNG0 | CPSBOUT | TBXCS | 00-- 0000 | 00-- 0000 |
| CPSBCON1 | — | — | — | — | CPSBCH3 | CPSBCH2 | CPSBCH1 | CPSBCH0 | ---- 0000 | ---- 0000 |
| INTCON | GIE | PEIE | TMR0IE | INTE | RBIE | TMR0IF | INTF | RBIF | 0000 000x | 0000 000X |
| IOCB | IOCB7 | IOCB6 | IOCB5 | IOCB4 | IOCB3 | IOCB2 | IOCB1 | IOCB0 | 0000 0000 | 0000 0000 |
| OPTION_REG | $\overline{\text{RBPU}}$ | INTEDG | TMR0CS | TMR0SE | PSA | PS2 | PS1 | PS0 | 1111 1111 | 1111 1111 |
| PORTB | RB7 | RB6 | RB5 | RB4 | RB3 | RB2 | RB1 | RB0 | xxxx xxxx | xxxx xxxx |
| T3CON | TMR3CS1 | TMR3CS0 | T3CKPS1 | T3CKPS0 | — | T3SYNC | — | TMR3ON | 0000 -0-0 | 0000 -0-0 |
| T1GCON | TMR1GE | T1GPOL | T1GTM | T1GSPM | T1GGO/$\overline{\text{DONE}}$ | T1GVAL | T1GSS1 | T1GSS0 | 0000 0x00 | uuuu uxuu |
| TRISB | TRISB7 | TRISB6 | TRISB5 | TRISB4 | TRISB3 | TRISB2 | TRISB1 | TRISB0 | 1111 1111 | 1111 1111 |
| WPUB | WPUB7 | WPUB6 | WPUB5 | WPUB4 | WPUB3 | WPUB2 | WPUB1 | WPUB0 | 1111 1111 | 1111 1111 |

**Legend:** x = unknown, u = unchanged, - = unimplemented locations read as '0'. Shaded cells are not used by PORTB.

## 14.1 TimerA/B Operation

The TimerA/B modules can be used as either 8-bit timers or 8-bit counters. Additionally, the modules can also be used to set Timer1's/Timer3's period of measurement for the capacitive sensing modules via Timer1's or Timer3's gate feature.

**TABLE 14-1: CPSOSC/TIMER ASSOCIATION**

| Cap Sense Oscillator | Divider Timer | Period Measurement |
|---|---|---|
| CPS A | TimerA | Timer1 |
| CPS B | TimerB | Timer3 |

### 14.1.1 8-BIT TIMER MODE

The TimerA/B modules will increment every instruction cycle, if used without a prescaler. 8-bit Timer mode is selected by clearing the TMRxCS bit of the TxCON registers.

When TMRx is written, the increment is inhibited for two instruction cycles immediately following the write.

| **Note:** | The value written to the TMRx register can be adjusted, in order to account for the two instruction cycle delay when TMRx is written. |
|---|---|

### 14.1.2 8-BIT COUNTER MODE

In 8-bit Counter mode, the TimerA/B modules will increment on every rising or falling edge of the TxCKI pin or the Capacitive Sensing Oscillator (CPSxOSC) signal. 8-bit Counter mode using the TxCKI pin is selected by setting the TMRxCS bit of the TxCON register to '1' and resetting the TxXCS bit in the CPSxCON0 register to '0'. 8-bit Counter mode using the Capacitive Sensing Oscillator (CPSxOSC) signal is selected by setting the TMRxCS bit in the TxCON register to '1' and setting the TxXCS bit in the CPSxCON0 register to '1'.

The rising or falling transition of the incrementing edge for either input source is determined by the TMRxSE bit in the TxCON register.

### 14.1.3 SOFTWARE PROGRAMMABLE PRESCALER

For TimerA/B modules, the software programmable prescaler is exclusive to the Timer. The prescaler is enabled by clearing the TMRxPSA bit of the TxCON register.

There are eight prescaler options for TimerA/B modules ranging from 1:2 to 1:256. The prescale values are selectable via the TMRxPS<2:0> bits of the TxCON register for TimerA/B. In order to have a 1:1 prescaler value for the TimerA/B modules, the prescaler must be disabled.

The prescaler is not readable or writable. When the prescaler is enabled or assigned to the Timer module, all instructions writing to the TMRx register will clear the prescaler. Enabling the TimerA/B modules also clears the prescaler.

### 14.1.4 TIMERA/B INTERRUPT

TimerA/B will generate an interrupt when the corresponding TMR register overflows from FFh to 00h. The TMRxIF interrupt flag bit of the PIR2 register is set every time the TMRx register overflows. These interrupt flag bits are set regardless of whether or not the relative Timer interrupt is enabled. The interrupt flag bits can only be cleared in software. The TimerA/B interrupt enable bits are the TMRxIE in the PIE2 register.

| **Note:** | TimerA/B interrupts cannot wake the processor from Sleep since the timer is frozen during Sleep. |
|---|---|

### 14.1.5 USING TIMERA/B WITH AN EXTERNAL CLOCK

When TimerA/B is in Counter mode, the synchronization of the TxCKI input and the TMRx register is accomplished by sampling the prescaler output on the Q2 and Q4 cycles of the internal phase clocks. Therefore, the high and low periods of the external clock source must meet the timing requirements as shown in **Section 25.0 "Electrical Specifications"**.

### 14.1.6 TIMER ENABLE

Operation of TimerA/B is enabled by setting the TMRxON bit of the TxCON register. When the module is disabled, the value in the TMRx register is maintained. Enabling the TMRx module will reset the prescaler used by the counter.

### 14.1.7 OPERATION DURING SLEEP

TimerA and TimerB cannot operate while the processor is in Sleep mode. The contents of the TMRx registers will remain unchanged while the processor is in Sleep mode.

## 18.0 ADDRESSABLE UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER (AUSART)

The Addressable Universal Synchronous Asynchronous Receiver Transmitter (AUSART) module is a serial I/O communications peripheral. It contains all the clock generators, shift registers and data buffers necessary to perform an input or output serial data transfer independent of device program execution. The AUSART, also known as a Serial Communications Interface (SCI), can be configured as a full-duplex asynchronous system or half-duplex synchronous system. Full-Duplex mode is useful for communications with peripheral systems, such as CRT terminals and personal computers. Half-Duplex Synchronous mode is intended for communications with peripheral devices, such as A/D or D/A integrated circuits, serial EEPROMs or other microcontrollers. These devices typically do not have internal clocks for baud rate generation and require the external clock signal provided by a master synchronous device.

The AUSART module includes the following capabilities:

- Full-duplex asynchronous transmit and receive
- Two-character input buffer
- One-character output buffer
- Programmable 8-bit or 9-bit character length
- Address detection in 9-bit mode
- Input buffer overrun error detection
- Received character framing error detection
- Half-duplex synchronous master
- Half-duplex synchronous slave
- Sleep operation

Block diagrams of the AUSART transmitter and receiver are shown in Figure 18-1 and Figure 18-2.

**FIGURE 18-1: AUSART TRANSMIT BLOCK DIAGRAM**

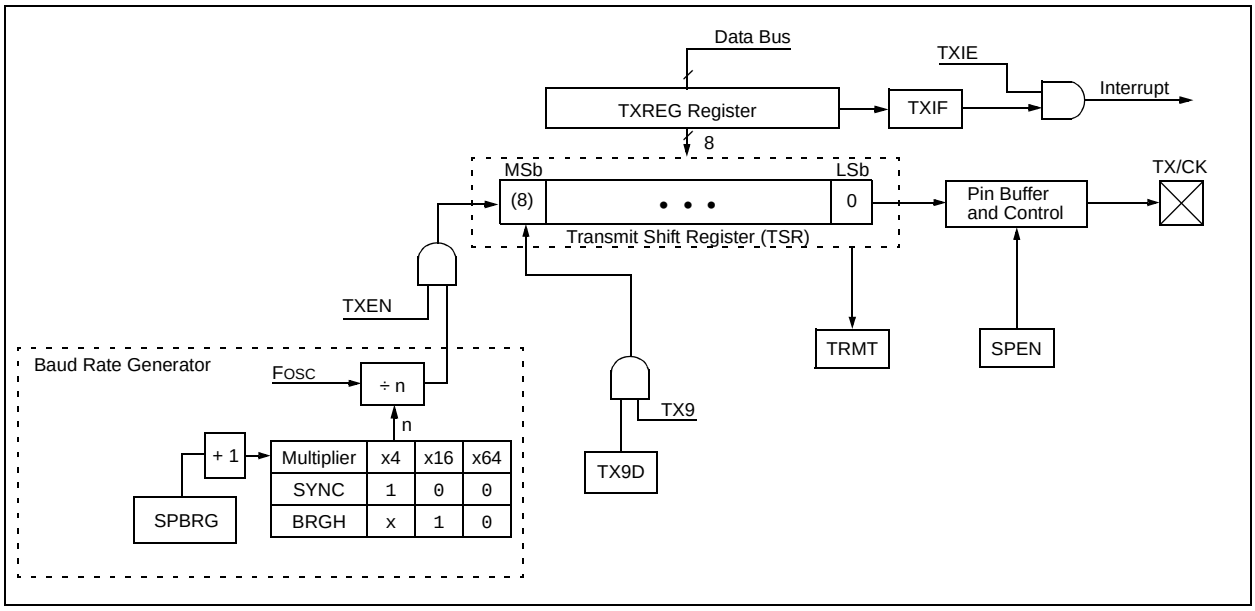


**FIGURE 18-3: ASYNCHRONOUS TRANSMISSION**

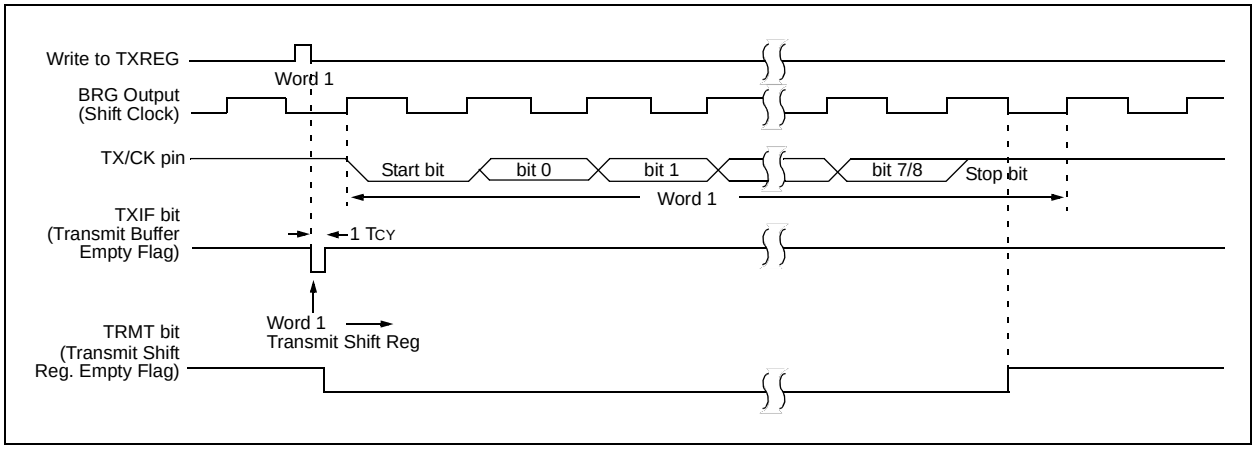
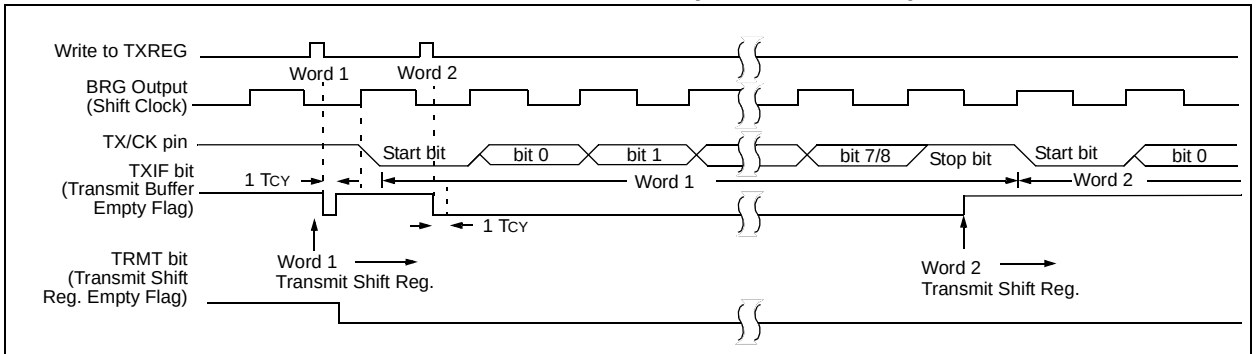



**FIGURE 18-4: ASYNCHRONOUS TRANSMISSION (BACK-TO-BACK)**




**Note:** This timing diagram shows two consecutive transmissions.

**TABLE 18-1: REGISTERS ASSOCIATED WITH ASYNCHRONOUS TRANSMISSION**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOR | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|
| INTCON | GIE | PEIE | TMR0IE | INTE | RBIE | TMR0IF | INTF | RBIF | 0000 000x | 0000 000x |
| PIE1 | TMR1GIE | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | 0000 0000 | 0000 0000 |
| PIR1 | TMR1GIF | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 0000 0000 | 0000 0000 |
| RCSTA | SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D | 0000 000x | 0000 000x |
| SPBRG | BRG7 | BRG6 | BRG5 | BRG4 | BRG3 | BRG2 | BRG1 | BRG0 | 0000 0000 | 0000 0000 |
| TRISC | TRISC7 | TRISC6 | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 | 1111 1111 | 1111 1111 |
| TXREG | AUSART Transmit Data Register | | | | | | | | 0000 0000 | 0000 0000 |
| TXSTA | CSRC | TX9 | TXEN | SYNC | — | BRGH | TRMT | TX9D | 0000 -010 | 0000 -010 |

**Legend:** x = unknown, - = unimplemented read as '0'. Shaded cells are not used for asynchronous transmission.

### 18.1.2 AUSART ASYNCHRONOUS RECEIVER

The Asynchronous mode is typically used in RS-232 systems. The receiver block diagram is shown in Figure 18-2. The data is received on the RX/DT pin and drives the data recovery block. The data recovery block is actually a high-speed shifter operating at 16 times the baud rate, whereas the serial Receive Shift Register (RSR) operates at the bit rate. When all eight or nine bits of the character have been shifted in, they are immediately transferred to a two character First-In First-Out (FIFO) memory. The FIFO buffering allows reception of two complete characters and the start of a third character before software must start servicing the AUSART receiver. The FIFO and RSR registers are not directly accessible by software. Access to the received data is via the RCREG register.

**REGISTER 18-2: RCSTA: RECEIVE STATUS AND CONTROL REGISTER**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R-0 | R-0 | R-x |
|---|---|---|---|---|---|---|---|
| SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D |
| 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 **SPEN:** Serial Port Enable bit(1)
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 ninth bit can be used as parity bit
Asynchronous mode 8-bit (RX9 = 0):
Don't care
Synchronous mode:
Must be set to '0'

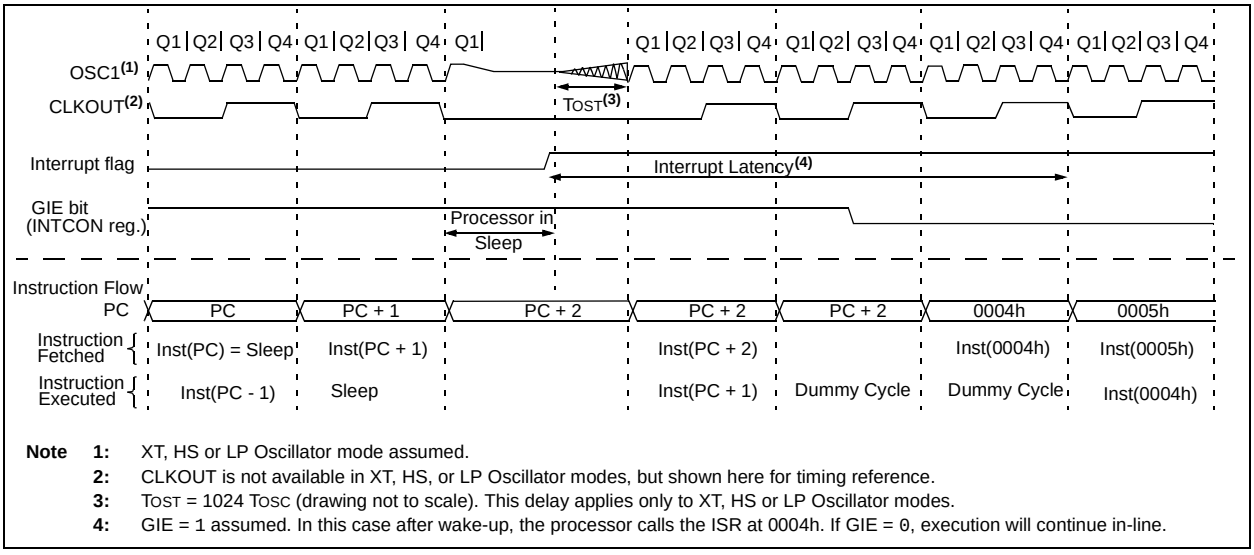
bit 2 **FERR:** Framing Error bit
1 = Framing error (can be updated by reading RCREG register and receive next valid byte)
0 = No framing error

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:** The AUSART module automatically changes the pin from tri-state to drive as needed. Configure TRISx = 1.

**FIGURE 21-1: WAKE-UP FROM SLEEP THROUGH INTERRUPT**




**Note 1:** XT, HS or LP Oscillator mode assumed.
**2:** CLKOUT is not available in XT, HS, or LP Oscillator modes, but shown here for timing reference.
**3:** TOST = 1024 TOSC (drawing not to scale). This delay applies only to XT, HS or LP Oscillator modes.
**4:** GIE = 1 assumed. In this case after wake-up, the processor calls the ISR at 0004h. If GIE = 0, execution will continue in-line.

**TABLE 21-1: SUMMARY OF REGISTERS ASSOCIATED WITH POWER-DOWN MODE**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOR | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|
| INTCON | GIE | PEIE | TMR0IE | INTE | IOCIE | TMR0IF | INTF | IOCIF | 0000 000x | 0000 000x |
| IOCBF | IOCBF7 | IOCBF6 | IOCBF5 | IOCBF4 | IOCBF3 | IOCBF2 | IOCBF1 | IOCBF0 | 0000 0000 | 0000 0000 |
| PIE1 | TMR1GIE | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | 0000 0000 | 0000 0000 |
| PIE2 | TMR3GIE | TMR3IE | TMRBIE | TMRAIE | — | — | — | CCP2IE | 0000 ---0 | 0000 ---0 |
| PIR1 | TMR1GIF | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 0000 0000 | 0000 0000 |
| PIR2 | TMR3GIF | TMR3IF | TMRBIF | TMRAIF | — | — | — | CCP2IF | 0000 ---0 | 0000 ---0 |
| STATUS | IRP | RP1 | RP0 | $\overline{TO}$ | $\overline{PD}$ | Z | DC | C | 0001 1xxx | 000q quuu |

**Legend:** — = unimplemented location, read as '0'. Shaded cells are not used in Power-Down mode.

## 23.2 Instruction Descriptions

| ADDLW | Add literal and W |
|---|---|
| Syntax: | [ *label* ] ADDLW k |
| Operands: | $0 \le k \le 255$ |
| Operation: | (W) + k → (W) |
| Status Affected: | C, DC, Z |
| Description: | The contents of the W register are added to the 8-bit literal 'k' and the result is placed in the W register. |

| ADDWF | Add W and f |
|---|---|
| Syntax: | [ *label* ] ADDWF f,d |
| Operands: | $0 \le f \le 127$<br>$d \in [0,1]$ |
| Operation: | (W) + (f) → (destination) |
| Status Affected: | C, DC, Z |
| Description: | Add the contents of the W register with register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'. |

| ANDLW | AND literal with W |
|---|---|
| Syntax: | [ *label* ] ANDLW k |
| Operands: | $0 \le k \le 255$ |
| Operation: | (W) .AND. (k) → (W) |
| Status Affected: | Z |
| Description: | The contents of W register are AND'ed with the 8-bit literal 'k'. The result is placed in the W register. |

| ANDWF | AND W with f |
|---|---|
| Syntax: | [ *label* ] ANDWF f,d |
| Operands: | $0 \le f \le 127$<br>$d \in [0,1]$ |
| Operation: | (W) .AND. (f) → (destination) |
| Status Affected: | Z |
| Description: | AND the W register with register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'. |

| BCF | Bit Clear f |
|---|---|
| Syntax: | [ *label* ] BCF f,b |
| Operands: | $0 \le f \le 127$<br>$0 \le b \le 7$ |
| Operation: | 0 → (f<b>) |
| Status Affected: | None |
| Description: | Bit 'b' in register 'f' is cleared. |

| BSF | Bit Set f |
|---|---|
| Syntax: | [ *label* ] BSF f,b |
| Operands: | $0 \le f \le 127$<br>$0 \le b \le 7$ |
| Operation: | 1 → (f<b>) |
| Status Affected: | None |
| Description: | Bit 'b' in register 'f' is set. |

| BTFSC | Bit Test f, Skip if Clear |
|---|---|
| Syntax: | [ *label* ] BTFSC f,b |
| Operands: | $0 \le f \le 127$<br>$0 \le b \le 7$ |
| Operation: | skip if (f<b>) = 0 |
| Status Affected: | None |
| Description: | If bit 'b' in register 'f' is '1', the next instruction is executed.<br>If bit 'b', in register 'f', is '0', the next instruction is discarded, and a NOP is executed instead, making this a 2-cycle instruction. |

## 24.2 MPLAB XC Compilers

The MPLAB XC Compilers are complete ANSI C compilers for all of Microchip's 8, 16, and 32-bit MCU and DSC devices. These compilers provide powerful integration capabilities, superior code optimization and ease of use. MPLAB XC Compilers run on Windows, Linux or MAC OS X.

For easy source level debugging, the compilers provide debug information that is optimized to the MPLAB X IDE.

The free MPLAB XC Compiler editions support all devices and commands, with no time or memory restrictions, and offer sufficient code optimization for most applications.

MPLAB XC Compilers include an assembler, linker and utilities. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. MPLAB XC Compiler uses the assembler to produce its object file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command-line interface
- Rich directive set
- Flexible macro language
- MPLAB X IDE compatibility

## 24.3 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for PIC10/12/16/18 MCUs.

The MPASM Assembler generates relocatable object files for the MPLINK Object Linker, Intel® standard HEX files, MAP files to detail memory usage and symbol reference, absolute LST files that contain source lines and generated machine code, and COFF files for debugging.

The MPASM Assembler features include:

- Integration into MPLAB X IDE projects
- User-defined macros to streamline assembly code
- Conditional assembly for multipurpose source files
- Directives that allow complete control over the assembly process

## 24.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler. It can link relocatable objects from precompiled libraries, using directives from a linker script.

The MPLIB Object Librarian manages the creation and modification of library files of precompiled code. When a routine from a library is called from a source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications.

The object linker/library features include:

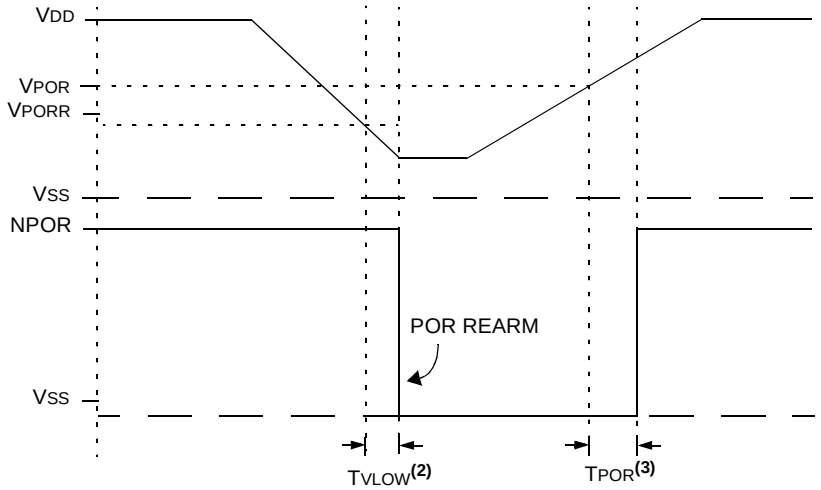
- Efficient linking of single libraries instead of many smaller files
- Enhanced code maintainability by grouping related modules together
- Flexible creation of libraries with easy module listing, replacement, deletion and extraction

## 24.5 MPLAB Assembler, Linker and Librarian for Various Device Families

MPLAB Assembler produces relocatable machine code from symbolic assembly language for PIC24, PIC32 and dsPIC DSC devices. MPLAB XC Compiler uses the assembler to produce its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command-line interface
- Rich directive set
- Flexible macro language
- MPLAB X IDE compatibility

**FIGURE 25-1: POR AND POR REARM WITH SLOW RISING VDD**




**Note 1:** When NPOR is low, the device is held in Reset.
**2:** TPOR 1 μs typical.
**3:** TVLOW 2.7 μs typical.

## 25.6 Timing Parameter Symbology

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

1. TppS2ppS
2. TppS

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

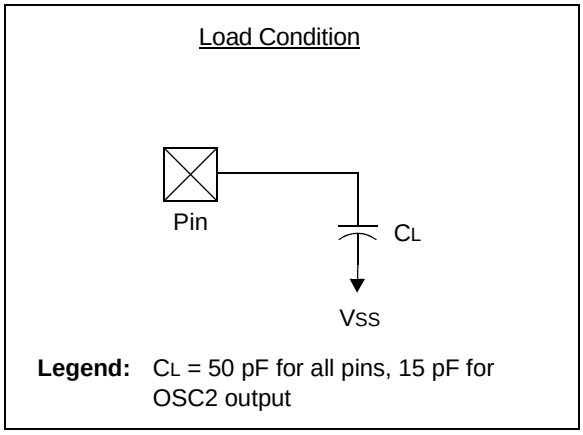
Lowercase letters (pp) and their meanings:

| | | | |
|---|---|---|---|
| **pp** | | | |
| cc | CCP1 | osc | OSC1 |
| ck | CLKOUT | 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 | T1CKI |
| mc | $\overline{MCLR}$ | wr | $\overline{WR}$ |

Uppercase letters and their meanings:

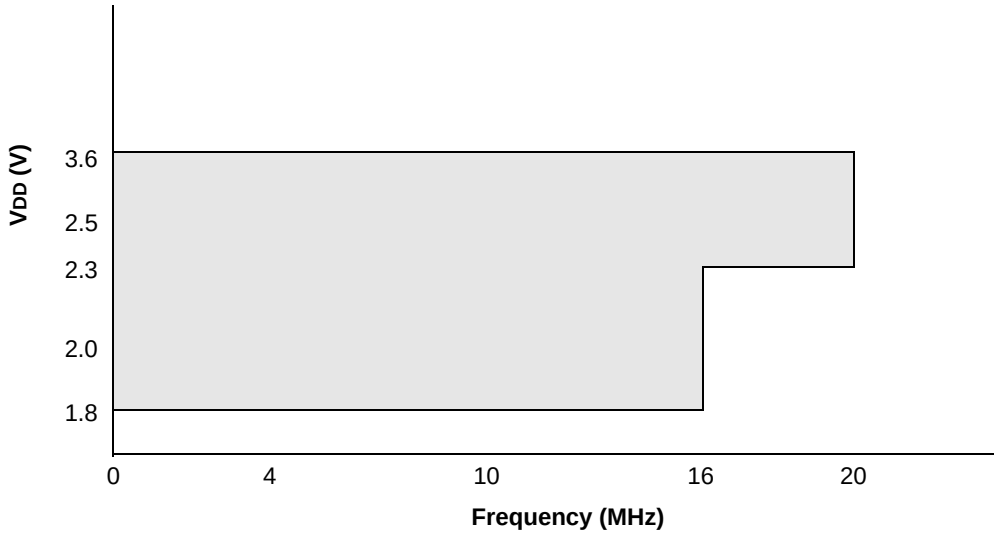
| | | | |
|---|---|---|---|
| **S** | | | |
| F | Fall | P | Period |
| H | High | R | Rise |
| I | Invalid (High-impedance) | V | Valid |
| L | Low | Z | High-impedance |

**FIGURE 25-2: LOAD CONDITIONS**



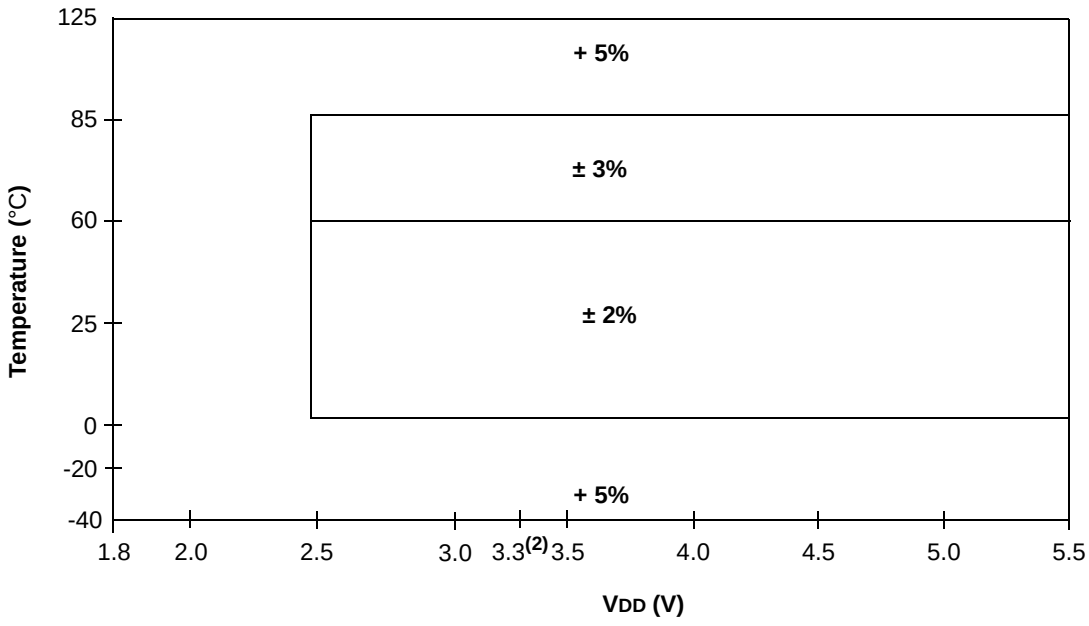

**Legend:** $C_L$ = 50 pF for all pins, 15 pF for OSC2 output

**FIGURE 25-5: PIC16LF707 VOLTAGE FREQUENCY GRAPH, -40°C ≤ TA ≤ +125°C**




**Note 1:** The shaded region indicates the permissible combinations of voltage and frequency.
**2:** Refer to Table 25-1 for each Oscillator mode's supported frequencies.

**FIGURE 25-6: HFINTOSC FREQUENCY ACCURACY OVER DEVICE VDD AND TEMPERATURE**




**Note 1:** This chart covers both regulator enabled and regulator disabled states.
**2:** Regulator Nominal voltage.

**TABLE 25-1: CLOCK OSCILLATOR TIMING REQUIREMENTS**

| **Standard Operating Conditions (unless otherwise stated)**<br>Operating temperature -40°C ≤ TA ≤ +125°C | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym.** | **Characteristic** | **Min.** | **Typ†** | **Max.** | **Units** | **Conditions** |
| OS01 | FOSC | External CLKIN Frequency(1) | DC | — | 37 | kHz | LP Oscillator mode |
| | | | DC | — | 4 | MHz | XT Oscillator mode |
| | | | DC | — | 20 | MHz | HS Oscillator mode |
| | | | DC | — | 20 | MHz | EC Oscillator mode |
| | | Oscillator Frequency(1) | — | 32.768 | — | kHz | LP Oscillator mode |
| | | | 0.1 | — | 4 | MHz | XT Oscillator mode |
| | | | 1 | — | 4 | MHz | HS Oscillator mode, VDD ≤ 2.7V |
| | | | 1 | — | 20 | MHz | HS Oscillator mode, VDD > 2.7V |
| | | | DC | — | 4 | MHz | RC Oscillator mode |
| OS02 | TOSC | External CLKIN Period(1) | 27 | — | ∞ | µs | LP Oscillator mode |
| | | | 250 | — | ∞ | ns | XT Oscillator mode |
| | | | 50 | — | ∞ | ns | HS Oscillator mode |
| | | | 50 | — | ∞ | ns | EC Oscillator mode |
| | | Oscillator Period(1) | — | 30.5 | — | µs | LP Oscillator mode |
| | | | 250 | — | 10,000 | ns | XT Oscillator mode |
| | | | 250 | — | 1,000 | ns | HS Oscillator mode, VDD ≤ 2.7V |
| | | | 50 | — | 1,000 | ns | HS Oscillator mode, VDD > 2.7V |
| | | | 250 | — | — | ns | RC Oscillator mode |
| OS03 | TCY | Instruction Cycle Time(1) | 200 | TCY | DC | ns | TCY = 4/FOSC |
| OS04* | TosH, TosL | External CLKIN High, External CLKIN Low | 2 | — | — | µs | LP oscillator |
| | | | 100 | — | — | ns | XT oscillator |
| | | | 20 | — | — | ns | HS oscillator |
| OS05* | TosR, TosF | External CLKIN Rise, External CLKIN Fall | 0 | — | ∞ | ns | LP oscillator |
| | | | 0 | — | ∞ | ns | XT oscillator |
| | | | 0 | — | ∞ | ns | HS oscillator |

* 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:** Instruction cycle period (TCY) equals four times the input oscillator time base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at "min" values with an external clock applied to OSC1 pin. When an external clock input is used, the "max" cycle time limit is "DC" (no clock) for all devices.

**TABLE 25-3: CLKOUT AND I/O TIMING PARAMETERS**

| **Standard Operating Conditions (unless otherwise stated)**<br>Operating Temperature -40°C ≤ TA ≤ +125°C | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym.** | **Characteristic** | **Min.** | **Typ†** | **Max.** | **Units** | **Conditions** |
| OS11 | TosH2ckL | Fosc↑ to CLKOUT↓ **(1)** | — | — | 70 | ns | VDD = 3.3-5.0V |
| OS12 | TosH2ckH | Fosc↑ to CLKOUT↑ **(1)** | — | — | 72 | ns | VDD = 3.3-5.0V |
| OS13 | TckL2ioV | CLKOUT↓ to Port out valid**(1)** | — | — | 20 | ns | |
| OS14 | TioV2ckH | Port input valid before CLKOUT↑**(1)** | TOSC + 200 ns | — | — | ns | |
| OS15 | TosH2ioV | Fosc↑ (Q1 cycle) to Port out valid | — | 50 | 70* | ns | VDD = 3.3-5.0V |
| OS16 | TosH2ioI | Fosc↑ (Q2 cycle) to Port input invalid (I/O in hold time) | 50 | — | — | ns | VDD = 3.3-5.0V |
| OS17 | TioV2osH | Port input valid to Fosc↑ (Q2 cycle) (I/O in setup time) | 20 | — | — | ns | |
| OS18 | TioR | Port output rise time**(2)** | —<br>— | 40<br>15 | 72<br>32 | ns | VDD = 2.0V<br>VDD = 3.3-5.0V |
| OS19 | TioF | Port output fall time**(2)** | —<br>— | 28<br>15 | 55<br>30 | ns | VDD = 2.0V<br>VDD = 3.3-5.0V |
| OS20* | Tinp | INT pin input high or low time | 25 | — | — | ns | |
| OS21* | Trbp | PORTB interrupt-on-change new input level time | TCY | — | — | ns | |

* These parameters are characterized but not tested.

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated.

**Note 1:** Measurements are taken in RC mode where CLKOUT output is 4 x TOSC.

**2:** Includes OSC2 in CLKOUT mode.

**TABLE 25-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, POWER-UP TIMER, AND BROWN-OUT RESET PARAMETERS**

| **Standard Operating Conditions (unless otherwise stated)**<br>Operating Temperature -40°C ≤ TA ≤ +125°C | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym.** | **Characteristic** | **Min.** | **Typ†** | **Max.** | **Units** | **Conditions** |
| 30 | TMCL | $\overline{\text{MCLR}}$ Pulse Width (low) | 2<br>5 | —<br>— | —<br>— | μs<br>μs | VDD = 3.3-5V, -40°C to +85°C<br>VDD = 3.3-5V |
| 31 | TWDTLP | Low Power Watchdog Timer Time-out Period (No Prescaler) | 10 | 18 | 27 | ms | VDD = 3.3V-5V |
| 32 | TOST | Oscillator Start-up Timer Period(1), (2) | — | 1024 | — | Tosc | **(Note 3)** |
| 33* | TPWRT | Power-up Timer Period, $\overline{\text{PWRTE}}$ = 0 | 40 | 65 | 140 | ms | |
| 34* | TIOZ | I/O high-impedance from $\overline{\text{MCLR}}$ Low or Watchdog Timer Reset | — | — | 2.0 | μs | |
| 35 | VBOR | Brown-out Reset Voltage | 2.38<br>1.80 | 2.5<br>1.9 | 2.73<br>2.11 | V | BORV=2.5V<br>BORV=1.9V |
| 36* | VHYST | Brown-out Reset Hysteresis | 0 | 25 | 50 | mV | -40°C to +85°C |
| 37* | TBORDC | Brown-out Reset DC Response Time | 1 | 3 | 5<br>10 | μs | VDD ≤ VBOR, -40°C to +85°C<br>VDD ≤ VBOR |

* 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:** Instruction cycle period (TCY) equals four times the input oscillator time base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at "min" values with an external clock applied to the OSC1 pin. When an external clock input is used, the "max" cycle time limit is "DC" (no clock) for all devices.

**2:** By design.

**3:** Period of the slower clock.

**4:** To ensure these voltage tolerances, VDD and VSS must be capacitively decoupled as close to the device as possible. 0.1 μF and 0.01 μF values in parallel are recommended.

**FIGURE 26-4: PIC16LF707 TYPICAL IDD vs. FOSC OVER VDD, EC MODE**

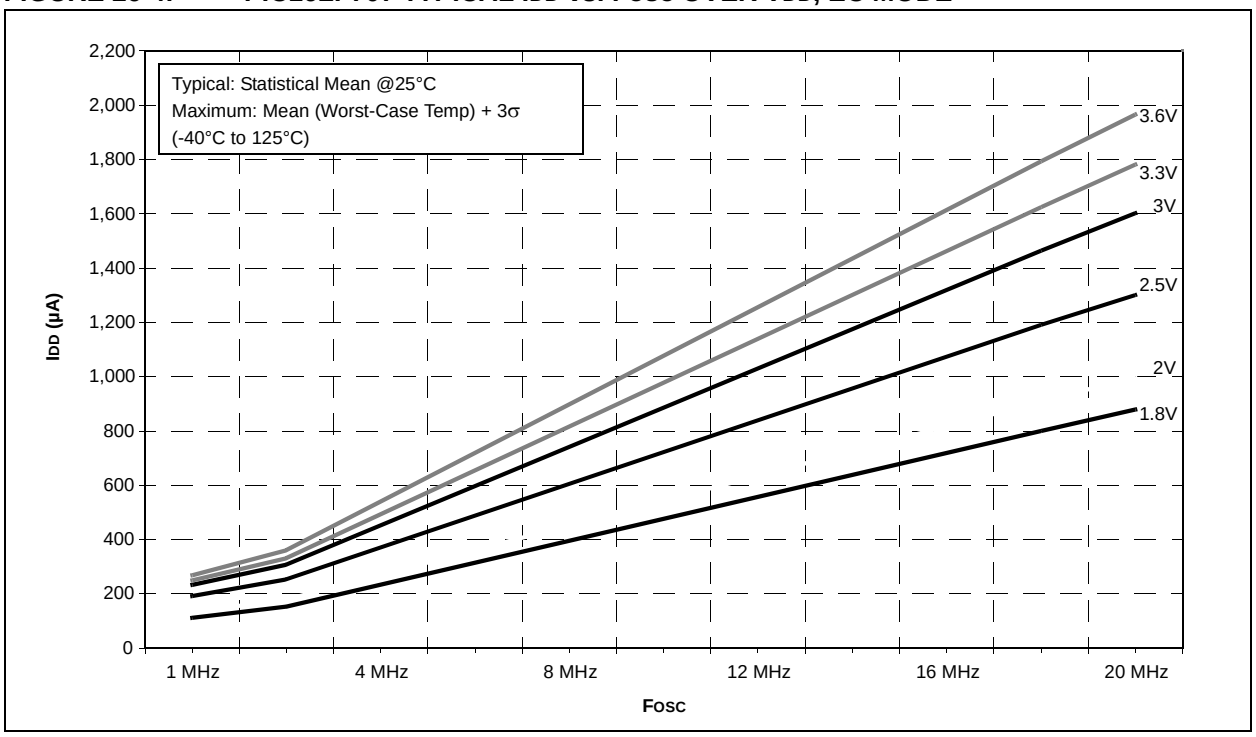



**FIGURE 26-5: PIC16F707 MAXIMUM IDD vs. VDD OVER FOSC, EXTRC MODE, VCAP = 0.1µF**

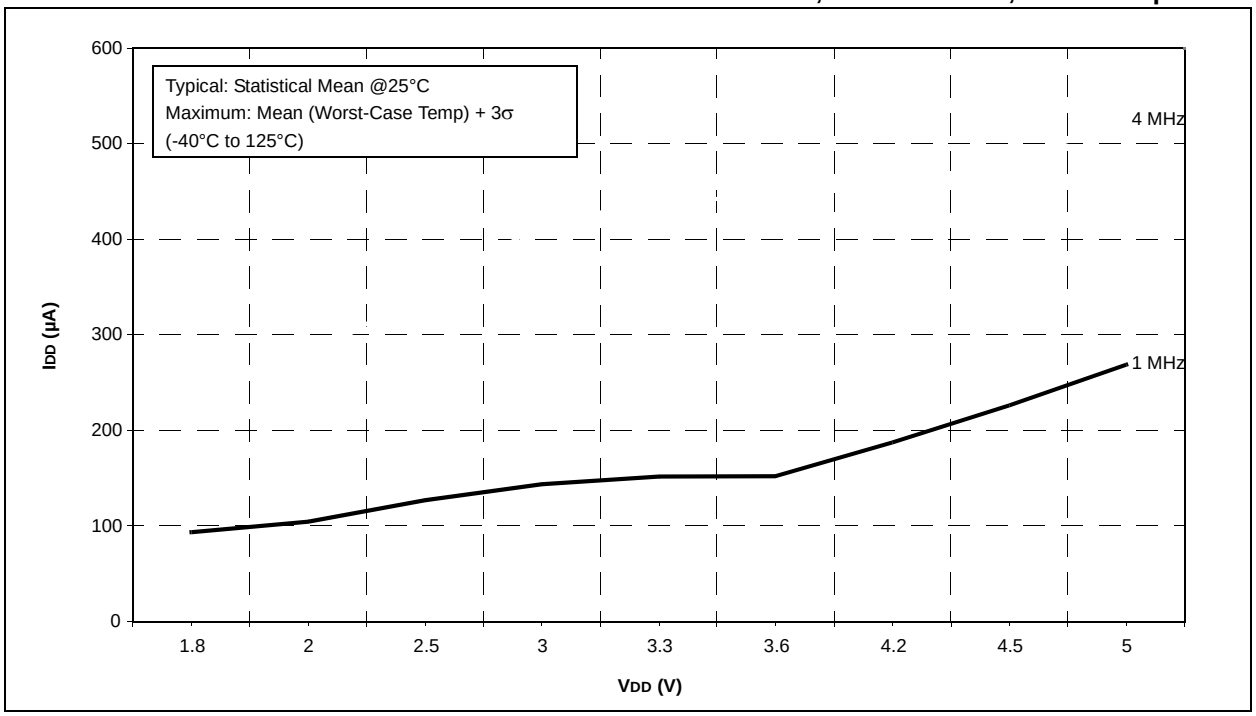


**FIGURE 26-6: PIC16LF707 MAXIMUM IDD vs. VDD OVER FOSC, EXTRC MODE**

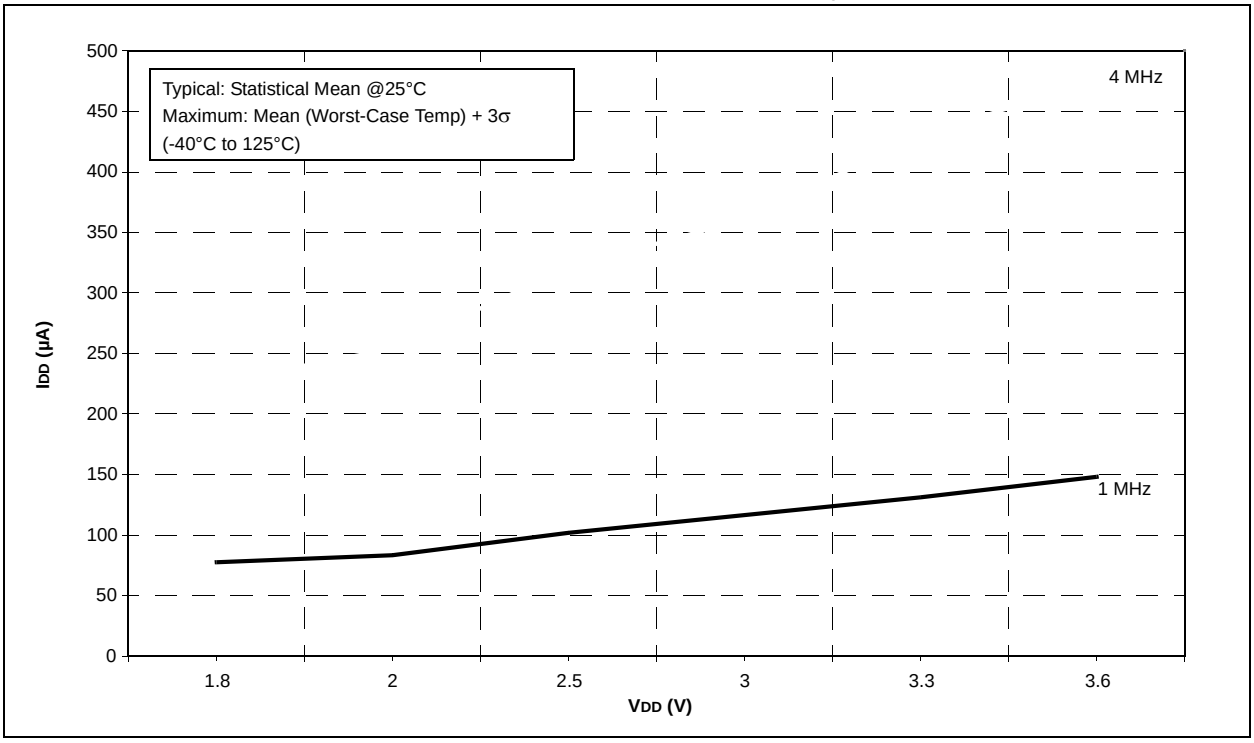



**FIGURE 26-7: PIC16F707 TYPICAL IDD vs. VDD OVER FOSC, EXTRC MODE, VCAP = 0.1µF**