**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 | - |
| Peripherals | Brown-out Detect/Reset, POR, PWM, WDT |
| Number of I/O | 17 |
| Program Memory Size | 3.5KB (2K x 14) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 128 x 8 |
| Voltage - Supply (Vcc/Vdd) | 1.8V ~ 3.6V |
| Data Converters | A/D 12x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Through Hole |
| Package / Case | 20-DIP (0.300", 7.62mm) |
| Supplier Device Package | 20-PDIP |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic16lf1507-i-p |

**TABLE 1-2: PIC16(L)F1507 PINOUT DESCRIPTION**

| Name | Function | Input Type | Output Type | Description |
|---|---|---|---|---|
| RA0/AN0/ICSPDAT | RA0 | TTL | CMOS | General purpose I/O. |
| | AN0 | AN | — | A/D Channel input. |
| | ICSPDAT | ST | CMOS | ICSP™ Data I/O. |
| RA1/AN1/VREF+/ICSPCLK | RA1 | TTL | CMOS | General purpose I/O. |
| | AN1 | AN | — | A/D Channel input. |
| | VREF+ | AN | — | A/D Positive Voltage Reference input. |
| | ICSPCLK | ST | — | Serial Programming Clock. |
| RA2/AN2/T0CKI/INT/PWM3/CLC1(1)/$\overline{\text{CWG1FLT}}$ | RA2 | ST | CMOS | General purpose I/O. |
| | AN2 | AN | — | A/D Channel input. |
| | T0CKI | ST | — | Timer0 clock input. |
| | INT | ST | — | External interrupt. |
| | PWM3 | — | CMOS | Pulse Width Module source output. |
| | CLC1 | — | CMOS | Configurable Logic Cell source output. |
| | $\overline{\text{CWG1FLT}}$ | ST | — | Complementary Waveform Generator Fault input. |
| RA3/CLC1IN0/VPP/$\overline{\text{MCLR}}$ | RA3 | TTL | — | General purpose input. |
| | CLC1IN0 | ST | — | Configurable Logic Cell source input. |
| | VPP | HV | — | Programming voltage. |
| | $\overline{\text{MCLR}}$ | ST | — | Master Clear with internal pull-up. |
| RA4/AN3/CLKOUT/T1G | RA4 | TTL | CMOS | General purpose I/O. |
| | AN3 | AN | — | A/D Channel input. |
| | CLKOUT | — | CMOS | FOSC/4 output. |
| | T1G | ST | — | Timer1 Gate input. |
| RA5/CLKIN/T1CKI/NCO1CLK | RA5 | TTL | CMOS | General purpose I/O. |
| | CLKIN | CMOS | — | External clock input (EC mode). |
| | T1CKI | ST | — | Timer1 clock input. |
| | NCO1CLK | ST | — | Numerically Controlled Oscillator Clock source input. |
| RB4/AN10 | RB4 | TTL | CMOS | General purpose I/O. |
| | AN10 | AN | — | A/D Channel input. |
| RB5/AN11 | RB5 | TTL | CMOS | General purpose I/O. |
| | AN11 | AN | — | A/D Channel input. |
| RB6 | RB6 | TTL | CMOS | General purpose I/O. |
| RB7 | RB7 | TTL | CMOS | General purpose I/O. |
| RC0/AN4/CLC2 | RC0 | TTL | CMOS | General purpose I/O. |
| | AN4 | AN | — | A/D Channel input. |
| | CLC2 | — | CMOS | Configurable Logic Cell source output. |
| RC1/AN5/PWM4/NCO1(1) | RC1 | TTL | CMOS | General purpose I/O. |
| | AN5 | AN | — | A/D Channel input. |
| | PWM4 | — | CMOS | Pulse Width Module source output. |
| | NCO1 | — | CMOS | Numerically Controlled Oscillator is source output. |
| RC2/AN6 | RC2 | TTL | CMOS | General purpose I/O. |
| | AN6 | AN | — | A/D Channel input. |

**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; $I^2C$™ = Schmitt Trigger input with $I^2C$ levels
HV = High Voltage; XTAL = Crystal

**Note 1:** Alternate pin function selected with the APFCON (Register 11-1) register.

**TABLE 3-5: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)**

| Address | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Bank 4** | | | | | | | | | | | |
| 20Ch | WPUA | — | — | WPUA5 | WPUA4 | WPUA3 | WPUA2 | WPUA1 | WPUA0 | --11 1111 | --11 1111 |
| 20Dh | WPUB | WPUB7 | WPUB6 | WPUB5 | WPUB4 | — | — | — | — | 1111 ---- | 1111 ---- |
| 20Eh to 21Fh | — | Unimplemented | | | | | | | | — | — |
| **Bank 5** | | | | | | | | | | | |
| 28Ch to 29Fh | — | Unimplemented | | | | | | | | — | — |
| **Bank 6** | | | | | | | | | | | |
| 30Ch to 31Fh | — | Unimplemented | | | | | | | | — | — |
| **Bank 7** | | | | | | | | | | | |
| 38Ch to 390h | — | Unimplemented | | | | | | | | — | — |
| 391h | IOCAP | — | — | IOCAP5 | IOCAP4 | IOCAP3 | IOCAP2 | IOCAP1 | IOCAP0 | --00 0000 | --00 0000 |
| 392h | IOCAN | — | — | IOCAN5 | IOCAN4 | IOCAN3 | IOCAN2 | IOCAN1 | IOCAN0 | --00 0000 | --00 0000 |
| 393h | IOCAF | — | — | IOCAF5 | IOCAF4 | IOCAF3 | IOCAF2 | IOCAF1 | IOCAF0 | --00 0000 | --00 0000 |
| 394h | IOCBP | IOCBP7 | IOCBP6 | IOCBP5 | IOCBP4 | — | — | — | — | 0000 ---- | 0000 ---- |
| 395h | IOCBN | IOCBN7 | IOCBN6 | IOCBN5 | IOCBN4 | — | — | — | — | 0000 ---- | 0000 ---- |
| 396h | IOCBF | IOCBF7 | IOCBF6 | IOCBF5 | IOCBF4 | — | — | — | — | 0000 ---- | 0000 ---- |
| 397h to 39Fh | — | Unimplemented | | | | | | | | — | — |
| **Bank 8** | | | | | | | | | | | |
| 40Ch to 41Fh | — | Unimplemented | | | | | | | | — | — |
| **Bank 9** | | | | | | | | | | | |
| 48Ch to 497h | — | Unimplemented | | | | | | | | — | — |
| 498h | NCO1ACCL | NCO1ACC<7:0> | | | | | | | | 0000 0000 | 0000 0000 |
| 499h | NCO1ACCH | NCO1ACC<15:8> | | | | | | | | 0000 0000 | 0000 0000 |
| 49Ah | NCO1ACCU | NCO1ACC<19:16> | | | | | | | | 0000 0000 | 0000 0000 |
| 49Bh | NCO1INCL | NCO1INC<7:0> | | | | | | | | 0000 0001 | 0000 0001 |
| 49Ch | NCO1INCH | NCO1INC<15:8> | | | | | | | | 0000 0000 | 0000 0000 |
| 49Dh | — | Unimplemented | | | | | | | | — | — |
| 49Eh | NCO1CON | N1EN | N1OE | N1OUT | N1POL | — | — | — | N1PFM | 0000 ---0 | 0000 ---0 |
| 49Fh | NCO1CLK | N1PWS<2:0> | | | — | — | — | N1CKS<1:0> | | 0000 --00 | 0000 --00 |

**Legend:** x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, r = reserved. Shaded locations are unimplemented, read as '0'.
**Note 1:** PIC16F1507 only.
**2:** Unimplemented, read as '1'.

**FIGURE 3-5: ACCESSING THE STACK EXAMPLE 2**

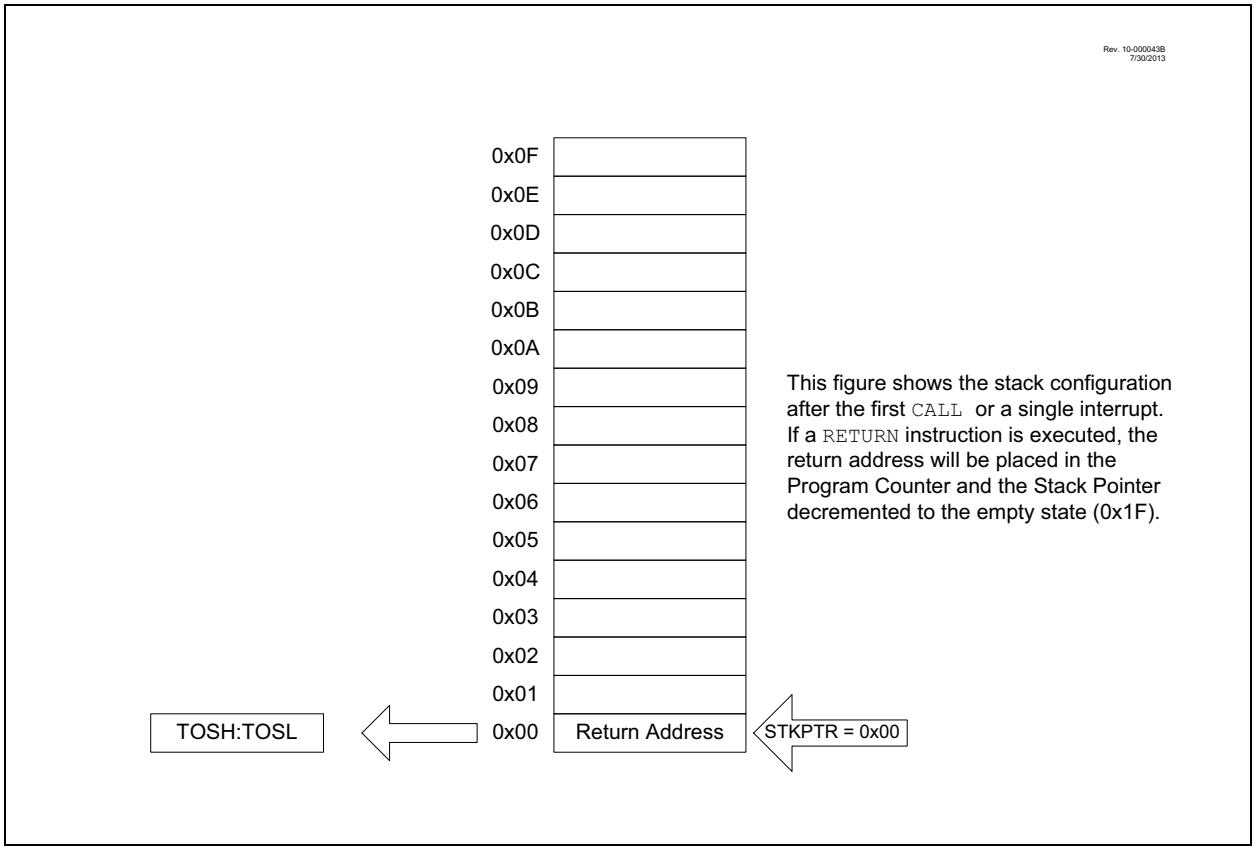
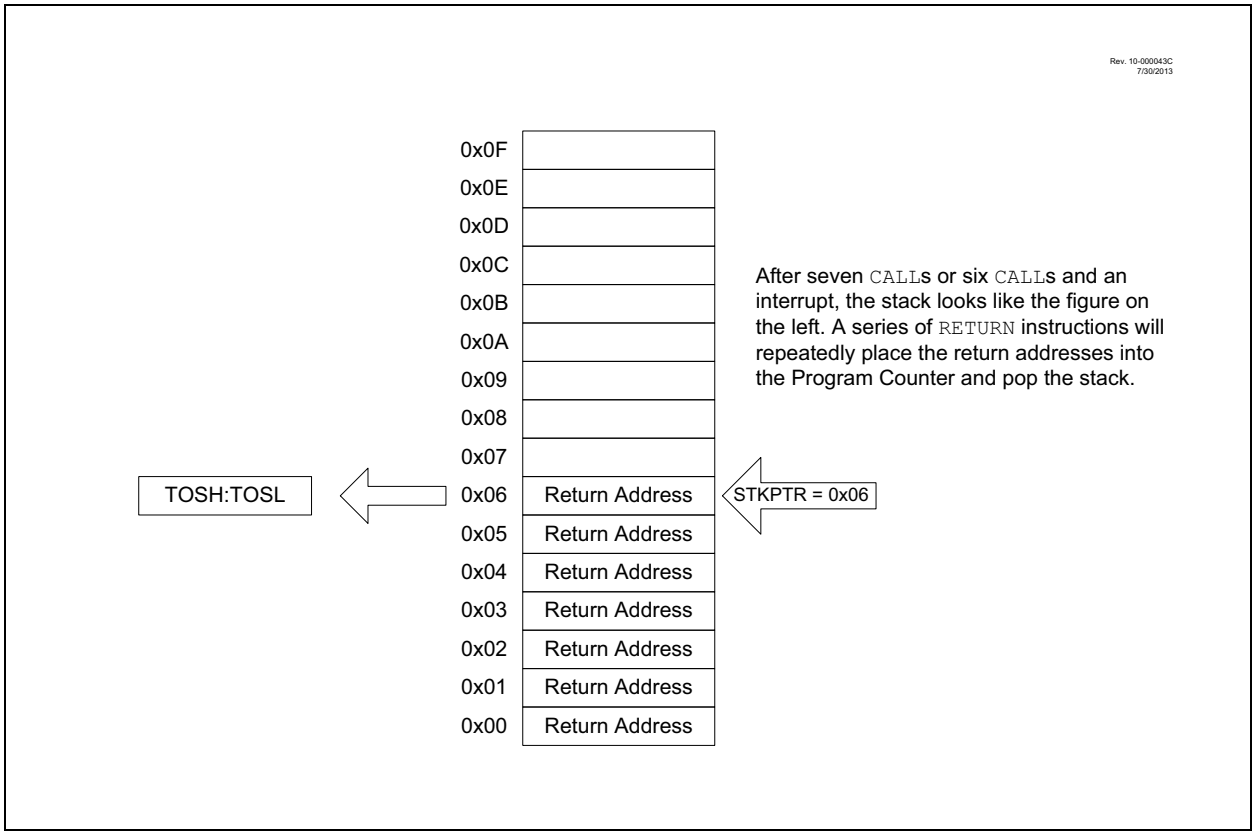



**FIGURE 3-6: ACCESSING THE STACK EXAMPLE 3**



# 5.0 OSCILLATOR MODULE

## 5.1 Overview

The oscillator module has a wide variety of clock sources and selection features that allow it to be used in a wide range of applications while maximizing performance and minimizing power consumption. Figure 5-1 illustrates a block diagram of the oscillator module.

Clock sources can be supplied from an external clock or from one of two internal oscillators, with a choice of speeds selectable via software. Additional clock features include:

- Selectable system clock source between external or internal sources via software.
- Fast start-up oscillator allows internal circuits to power-up and stabilize before switching to the 16 MHz HFINTOSC

The oscillator module can be configured in one of the following clock modes.

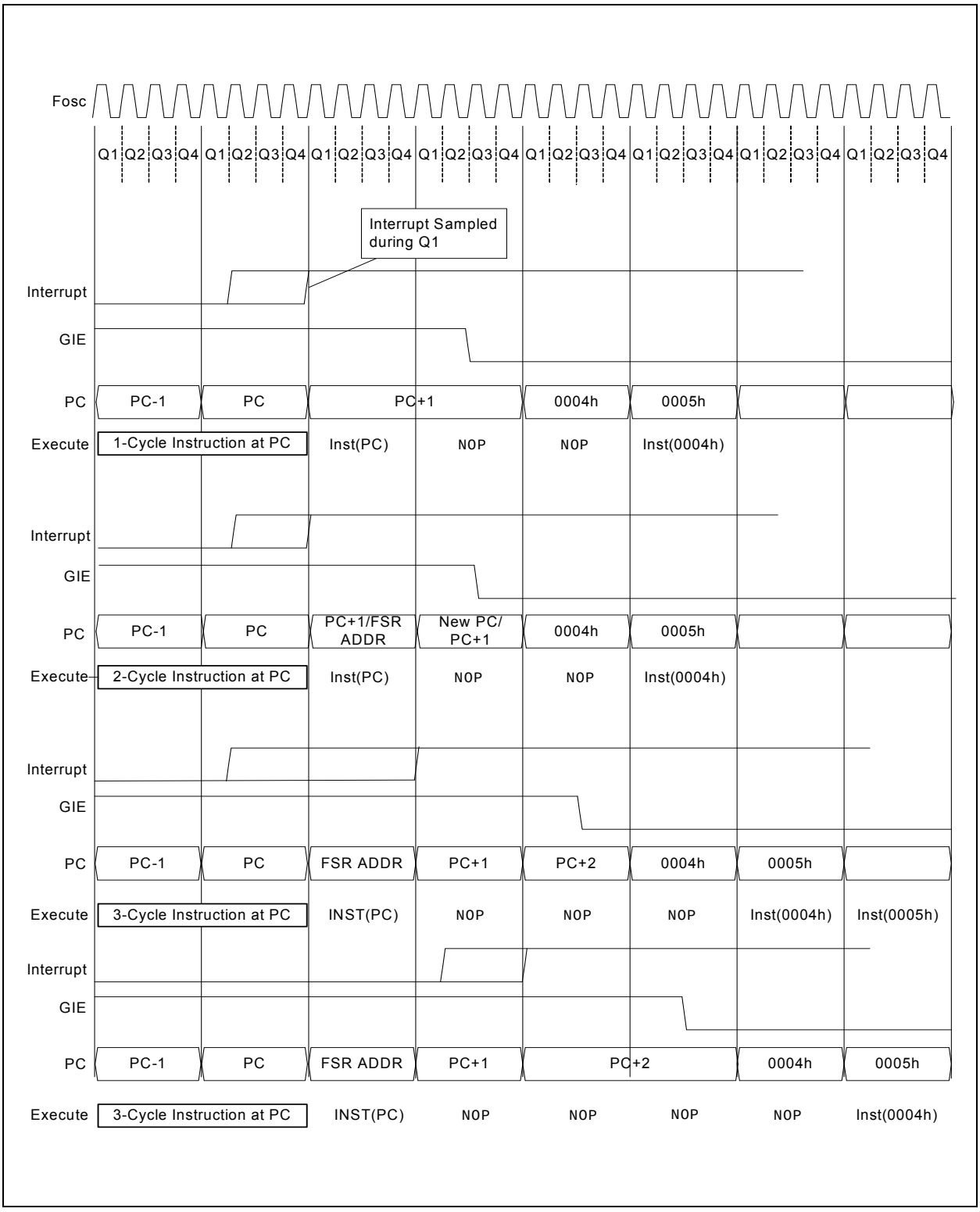
1. ECL – External Clock Low-Power mode (0 MHz to 0.5 MHz)
2. ECM – External Clock Medium Power mode (0.5 MHz to 4 MHz)
3. ECH – External Clock High-Power mode (4 MHz to 20 MHz)
4. INTOSC – Internal oscillator (31 kHz to 16 MHz)

Clock Source modes are selected by the FOSC<1:0> bits in the Configuration Words. The FOSC bits determine the type of oscillator that will be used when the device is first powered.

The ECH, ECM, and ECL clock modes rely on an external logic level signal as the device clock source.

The INTOSC internal oscillator block produces a low and high-frequency clock source, designated LFINTOSC and HFINTOSC. (See Internal Oscillator Block, Figure 5-1). A wide selection of device clock frequencies may be derived from these two clock sources.

**FIGURE 7-2: INTERRUPT LATENCY**



## 7.3 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 **Section 8.0 "Power-Down Mode (Sleep)"** for more details.

## 7.4 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.

## 7.5 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 $\overline{TO}$ and $\overline{PD}$)
- 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.

**TABLE 7-1: SUMMARY OF REGISTERS ASSOCIATED WITH INTERRUPTS**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| INTCON | GIE | PEIE | TMR0IE | INTE | IOCIE | TMR0IF | INTF | IOCIF | 64 |
| OPTION_REG | $\overline{\text{WPUEN}}$ | INTEDG | TMR0CS | TMR0SE | PSA | PS<2:0> | | | 133 |
| PIE1 | TMR1GIE | ADIE | — | — | — | — | TMR2IE | TMR1IE | 65 |
| PIE2 | — | — | — | — | — | NCO1IE | — | — | 66 |
| PIE3 | — | — | — | — | — | — | CLC2IE | CLC1IE | 67 |
| PIR1 | TMR1GIF | ADIF | — | — | — | — | TMR2IF | TMR1IF | 68 |
| PIR2 | — | — | — | — | — | NCO1IF | — | — | 69 |
| PIR3 | — | — | — | — | — | — | CLC2IF | CLC1IF | 70 |

**Legend:** — = unimplemented location, read as '0'. Shaded cells are not used by interrupts.

**REGISTER 11-9: LATB: PORTB DATA LATCH REGISTER**

| R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u | U-0 | U-0 | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| LATB7 | LATB6 | LATB5 | LATB4 | — | — | — | — |
| 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 7-4 **LATB<7:4>**: RB<7:4> Output Latch Value bits(1)

bit 3-0 **Unimplemented:** Read as ‘0’

**Note 1:** Writes to PORTB are actually written to corresponding LATB register. Reads from PORTB register is return of actual I/O pin values.

**REGISTER 11-10: ANSELB: PORTB ANALOG SELECT REGISTER**

| U-0 | U-0 | R/W-1/1 | R/W-1/1 | U-0 | U-0 | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| — | — | ANSB5 | ANSB4 | — | — | — | — |
| 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 5-4 **ANSB<5:4>**: Analog Select between Analog or Digital Function on pins RB<5:4>, respectively
1 = Analog input. Pin is assigned as analog input(1). Digital input buffer disabled.
0 = Digital I/O. Pin is assigned to port or digital special function.

bit 3-0 **Unimplemented:** Read as ‘0’

**Note 1:** When setting a pin to an analog input, the corresponding TRIS bit must be set to Input mode in order to allow external control of the voltage on the pin.

**REGISTER 11-15: ANSELC: PORTC ANALOG SELECT REGISTER**

| R/W-1/1 | R/W-1/1 | U-0 | U-0 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 |
|---|---|---|---|---|---|---|---|
| ANSC7 | ANSC6 | — | — | ANSC3 | ANSC2 | ANSC1 | ANSC0 |
| 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 **ANSC<7:6>**: Analog Select between Analog or Digital Function on pins RC<7:6>, respectively
1 = Analog input. Pin is assigned as analog input(1). Digital input buffer disabled.
0 = Digital I/O. Pin is assigned to port or digital special function.

bit 5-4 **Unimplemented:** Read as '0'

bit 3-0 **ANSC<3:0>**: Analog Select between Analog or Digital Function on pins RC<3:0>, respectively
1 = Analog input. Pin is assigned as analog input(1). Digital input buffer disabled.
0 = Digital I/O. Pin is assigned to port or digital special function.

**Note 1:** When setting a pin to an analog input, the corresponding TRIS bit must be set to Input mode in order to allow external control of the voltage on the pin.

**TABLE 11-9: SUMMARY OF REGISTERS ASSOCIATED WITH PORTC**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| ANSELC | ANSC7 | ANSC6 | — | — | ANSC3 | ANSC2 | ANSC1 | ANSC0 | 107 |
| LATC | LATC7 | LATC6 | LATC5 | LATC4 | LATC3 | LATC2 | LATC1 | LATC0 | 106 |
| PORTC | RC7 | RC6 | RC5 | RC4 | RC3 | RC2 | RC1 | RC0 | 106 |
| TRISC | TRISC7 | TRISC6 | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 | 106 |

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

**TABLE 15-3: SUMMARY OF REGISTERS ASSOCIATED WITH ADC**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| ADCON0 | — | CHS<4:0> | | | | | GO/$\overline{\text{DONE}}$ | ADON | 123 |
| ADCON1 | ADFM | ADCS<2:0> | | | — | — | ADPREF<1:0> | | 124 |
| ADCON2 | TRIGSEL<3:0> | | | | — | — | — | — | 125 |
| ADRESH | ADC Result Register High | | | | | | | | 126, 127 |
| ADRESL | ADC Result Register Low | | | | | | | | 126, 127 |
| ANSELA | — | — | — | ANSA4 | — | ANSA2 | ANSA1 | ANSA0 | 99 |
| ANSELB | — | — | ANSB5 | ANSB4 | — | — | — | — | 103 |
| ANSELC | ANSC7 | ANSC6 | — | — | ANSC3 | ANSC2 | ANSC1 | ANSC0 | 107 |
| INTCON | GIE | PEIE | TMR0IE | INTE | IOCIE | TMR0IF | INTF | IOCIF | 64 |
| PIE1 | TMR1GIE | ADIE | — | — | — | — | TMR2IE | TMR1IE | 65 |
| PIR1 | TMR1GIF | ADIF | — | — | — | — | TMR2IF | TMR1IF | 68 |
| TRISA | — | — | TRISA5 | TRISA4 | —(1) | TRISA2 | TRISA1 | TRISA0 | 98 |
| TRISB | TRISB7 | TRISB6 | TRISB5 | TRISB4 | — | — | — | — | 102 |
| TRISC | TRISC7 | TRISC6 | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 | 106 |
| FVRCON | FVREN | FVRRDY | TSEN | TSRNG | — | — | ADFVR<1:0> | | 114 |

**Legend:** x = unknown, u = unchanged, – = unimplemented read as '0', q = value depends on condition. Shaded cells are not used for ADC module.

**Note 1:** Unimplemented, read as '1'.

**TABLE 17-5: SUMMARY OF REGISTERS ASSOCIATED WITH TIMER1**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| ANSELA | — | — | — | ANSA4 | — | ANSA2 | ANSA1 | ANSA0 | 99 |
| INTCON | GIE | PEIE | TMR0IE | INTE | IOCIE | TMR0IF | INTF | IOCIF | 64 |
| PIE1 | TMR1GIE | ADIE | — | — | — | — | TMR2IE | TMR1IE | 65 |
| PIR1 | TMR1GIF | ADIF | — | — | — | — | TMR2IF | TMR1IF | 68 |
| TMR1H | Holding Register for the Most Significant Byte of the 16-bit TMR1 Count | | | | | | | | 138* |
| TMR1L | Holding Register for the Least Significant Byte of the 16-bit TMR1 Count | | | | | | | | 138* |
| TRISA | — | — | TRISA5 | TRISA4 | —(1) | TRISA2 | TRISA1 | TRISA0 | 98 |
| T1CON | TMR1CS<1:0> | | T1CKPS<1:0> | | — | $\overline{\text{T1SYNC}}$ | — | TMR1ON | 142 |
| T1GCON | TMR1GE | T1GPOL | T1GTM | T1GSPM | T1GGO/$\overline{\text{DONE}}$ | T1GVAL | — | T1GSS0 | 143 |

**Legend:** — = unimplemented location, read as '0'. Shaded cells are not used by the Timer1 module.
* Page provides register information.
**Note 1:** Unimplemented, read as '1'.

## 18.1 Timer2 Operation

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

TMR2 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, T2CKPS<1:0> of the T2CON register. The value of TMR2 is compared to that of the Period register, PR2, 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 TMR2 to 00h on the next cycle and drives the output counter/ postscaler (see **Section 18.2 "Timer2 Interrupt"**).

The TMR2 and PR2 registers are both directly readable and writable. The TMR2 register is cleared on any device Reset, whereas the PR2 register initializes to FFh. Both the prescaler and postscaler counters are cleared on the following events:

- a write to the TMR2 register
- a write to the T2CON 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:** | TMR2 is not cleared when T2CON is written. |
|---|---|

## 18.2 Timer2 Interrupt

Timer2 can also generate an optional device interrupt. The Timer2 output signal (T2_match) provides the input for the 4-bit counter/postscaler. This counter generates the TMR2 match interrupt flag which is latched in TMR2IF of the PIR1 register. The interrupt is enabled by setting the TMR2 Match Interrupt Enable bit, TMR2IE of the PIE1 register.

A range of 16 postscale options (from 1:1 through 1:16 inclusive) can be selected with the postscaler control bits, T2OUTPS<3:0>, of the T2CON register.

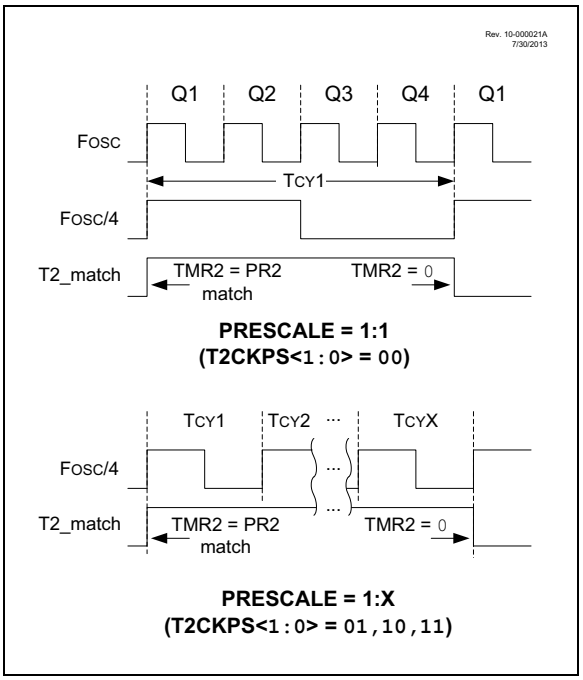
## 18.3 Timer2 Output

The output of TMR2 is T2_match. T2_match is available to the following peripherals:

- Configurable Logic Cell (CLC)
- Numerically Controlled Oscillator (NCO)
- Pulse Width Modulator (PWM)

The T2_match signal is synchronous with the system clock. Figure 18-3 shows two examples of the timing of the T2_match signal relative to FOSC and prescale value, T2CKPS<1:0>. The upper diagram illustrates 1:1 prescale timing and the lower diagram, 1:X prescale timing.

**FIGURE 18-3: T2_MATCH TIMING DIAGRAM**




## 18.4 Timer2 Operation During Sleep

Timer2 cannot be operated while the processor is in Sleep mode. The contents of the TMR2 and PR2 registers will remain unchanged while the processor is in Sleep mode.

## 20.1 CLCx Setup

Programming the CLCx module is performed by configuring the four stages in the logic signal flow. The four stages are:

- Data selection
- Data gating
- Logic function selection
- Output polarity

Each stage is setup at run time by writing to the corresponding CLCx Special Function Registers. This has the added advantage of permitting logic reconfiguration on-the-fly during program execution.

### 20.1.1 DATA SELECTION

There are 16 signals available as inputs to the configurable logic. Four 8-input multiplexers are used to select the inputs to pass on to the next stage. The 16 inputs to the multiplexers are arranged in groups of four. Each group is available to two of the four multiplexers, in each case, paired with a different group. This arrangement makes possible selection of up to two from a group without precluding a selection from another group.

Data selection is through four multiplexers as indicated on the left side of Figure 20-2. Data inputs in the figure are identified by a generic numbered input name.

Table 20-1 correlates the generic input name to the actual signal for each CLC module. The columns labeled lcxd1 through lcxd4 indicate the MUX output for the selected data input. D1S through D4S are abbreviations for the MUX select input codes: LCxD1S<2:0> through LCxD4S<2:0>, respectively. Selecting a data input in a column excludes all other inputs in that column.

Data inputs are selected with CLCxSEL0 and CLCxSEL1 registers (Register 20-3 and Register 20-5, respectively).

| **Note:** | Data selections are undefined at power-up. |
|---|---|

**TABLE 20-1: CLCx DATA INPUT SELECTION**

| Data Input | lcxd1 D1S | lcxd2 D2S | lcxd3 D3S | lcxd4 D4S | CLC 1 | CLC 2 |
|---|---|---|---|---|---|---|
| LCx_in[0] | 000 | — | — | 100 | CLC1IN0 | CLC2IN0 |
| LCx_in[1] | 001 | — | — | 101 | CLC1IN1 | CLC2IN1 |
| LCx_in[2] | 010 | — | — | 110 | Reserved | Reserved |
| LCx_in[3] | 011 | — | — | 111 | Reserved | Reserved |
| LCx_in[4] | 100 | 000 | — | — | FOSC | FOSC |
| LCx_in[5] | 101 | 001 | — | — | T0_overflow | T0_overflow |
| LCx_in[6] | 110 | 010 | — | — | T1_overflow | T1_overflow |
| LCx_in[7] | 111 | 011 | — | — | T2_match | T2_match |
| LCx_in[8] | — | 100 | 000 | — | LC1_out | LC1_out |
| LCx_in[9] | — | 101 | 001 | — | LC2_out | LC2_out |
| LCx_in[10] | — | 110 | 010 | — | Reserved | Reserved |
| LCx_in[11] | — | 111 | 011 | — | Reserved | Reserved |
| LCx_in[12] | — | — | 100 | 000 | NCO1_out | LFINTOSC |
| LCx_in[13] | — | — | 101 | 001 | HFINTOSC | FRC |
| LCx_in[14] | — | — | 110 | 010 | PWM3_out | PWM1_out |
| LCx_in[15] | — | — | 111 | 011 | PWM4_out | PWM2_out |

**REGISTER 20-6: CLCxGLS1: GATE 2 LOGIC SELECT REGISTER**

| R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u |
|---|---|---|---|---|---|---|---|
| LCxG2D4T | LCxG2D4N | LCxG2D3T | LCxG2D3N | LCxG2D2T | LCxG2D2N | LCxG2D1T | LCxG2D1N |
| 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 **LCxG2D4T:** Gate 2 Data 4 True (non-inverted) bit
1 = lcxd4T is gated into lcxg2
0 = lcxd4T is not gated into lcxg2

bit 6 **LCxG2D4N:** Gate 2 Data 4 Negated (inverted) bit
1 = lcxd4N is gated into lcxg2
0 = lcxd4N is not gated into lcxg2

bit 5 **LCxG2D3T:** Gate 2 Data 3 True (non-inverted) bit
1 = lcxd3T is gated into lcxg2
0 = lcxd3T is not gated into lcxg2

bit 4 **LCxG2D3N:** Gate 2 Data 3 Negated (inverted) bit
1 = lcxd3N is gated into lcxg2
0 = lcxd3N is not gated into lcxg2

bit 3 **LCxG2D2T:** Gate 2 Data 2 True (non-inverted) bit
1 = lcxd2T is gated into lcxg2
0 = lcxd2T is not gated into lcxg2

bit 2 **LCxG2D2N:** Gate 2 Data 2 Negated (inverted) bit
1 = lcxd2N is gated into lcxg2
0 = lcxd2N is not gated into lcxg2

bit 1 **LCxG2D1T:** Gate 2 Data 1 True (non-inverted) bit
1 = lcxd1T is gated into lcxg2
0 = lcxd1T is not gated into lcxg2

bit 0 **LCxG2D1N:** Gate 2 Data 1 Negated (inverted) bit
1 = lcxd1N is gated into lcxg2
0 = lcxd1N is not gated into lcxg2

**REGISTER 20-7: CLCxGLS2: GATE 3 LOGIC SELECT REGISTER**

| R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u |
|---|---|---|---|---|---|---|---|
| LCxG3D4T | LCxG3D4N | LCxG3D3T | LCxG3D3N | LCxG3D2T | LCxG3D2N | LCxG3D1T | LCxG3D1N |
| 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 **LCxG3D4T:** Gate 3 Data 4 True (non-inverted) bit
1 = lcxd4T is gated into lcxg3
0 = lcxd4T is not gated into lcxg3

bit 6 **LCxG3D4N:** Gate 3 Data 4 Negated (inverted) bit
1 = lcxd4N is gated into lcxg3
0 = lcxd4N is not gated into lcxg3

bit 5 **LCxG3D3T:** Gate 3 Data 3 True (non-inverted) bit
1 = lcxd3T is gated into lcxg3
0 = lcxd3T is not gated into lcxg3

bit 4 **LCxG3D3N:** Gate 3 Data 3 Negated (inverted) bit
1 = lcxd3N is gated into lcxg3
0 = lcxd3N is not gated into lcxg3

bit 3 **LCxG3D2T:** Gate 3 Data 2 True (non-inverted) bit
1 = lcxd2T is gated into lcxg3
0 = lcxd2T is not gated into lcxg3

bit 2 **LCxG3D2N:** Gate 3 Data 2 Negated (inverted) bit
1 = lcxd2N is gated into lcxg3
0 = lcxd2N is not gated into lcxg3

bit 1 **LCxG3D1T:** Gate 3 Data 1 True (non-inverted) bit
1 = lcxd1T is gated into lcxg3
0 = lcxd1T is not gated into lcxg3

bit 0 **LCxG3D1N:** Gate 3 Data 1 Negated (inverted) bit
1 = lcxd1N is gated into lcxg3
0 = lcxd1N is not gated into lcxg3

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

| | |
|---|---|
| Syntax: | [ *label* ] DECFSZ f,d |
| Operands: | 0 ≤ f ≤ 127<br>d ∈ [0,1] |
| Operation: | (f) - 1 → (destination);<br>skip if result = 0 |
| Status Affected: | None |
| Description: | The contents of register 'f' are decremented. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'.<br>If the result is '1', the next instruction is executed. If the result is '0', then a NOP is executed instead, making it a 2-cycle instruction. |

**GOTO** **Unconditional Branch**

| | |
|---|---|
| Syntax: | [ *label* ] GOTO k |
| Operands: | 0 ≤ k ≤ 2047 |
| Operation: | k → PC<10:0><br>PCLATH<6:3> → PC<14:11> |
| Status Affected: | None |
| Description: | GOTO is an unconditional branch. The 11-bit immediate value is loaded into PC bits <10:0>. The upper bits of PC are loaded from PCLATH<4:3>. GOTO is a 2-cycle instruction. |

**INCF** **Increment f**

| | |
|---|---|
| Syntax: | [ *label* ] INCF f,d |
| Operands: | 0 ≤ f ≤ 127<br>d ∈ [0,1] |
| Operation: | (f) + 1 → (destination) |
| Status Affected: | Z |
| Description: | The contents of register 'f' are incremented. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'. |

**INCFSZ** **Increment f, Skip if 0**

| | |
|---|---|
| Syntax: | [ *label* ] INCFSZ f,d |
| Operands: | 0 ≤ f ≤ 127<br>d ∈ [0,1] |
| Operation: | (f) + 1 → (destination),<br>skip if result = 0 |
| Status Affected: | None |
| Description: | The contents of register 'f' are incremented. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'.<br>If the result is '1', the next instruction is executed. If the result is '0', a NOP is executed instead, making it a 2-cycle instruction. |

**IORLW** **Inclusive OR literal with W**

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

**IORWF** **Inclusive OR W with f**

| | |
|---|---|
| Syntax: | [ *label* ] IORWF f,d |
| Operands: | 0 ≤ f ≤ 127<br>d ∈ [0,1] |
| Operation: | (W) .OR. (f) → (destination) |
| Status Affected: | Z |
| Description: | Inclusive OR the W register with register 'f'. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'. |

## 25.3 DC Characteristics

### TABLE 25-1: SUPPLY VOLTAGE

| PIC16LF1507 | | | Standard Operating Conditions (unless otherwise stated) | | | | |
|---|---|---|---|---|---|---|---|
| PIC16F1507 | | | | | | | |
| Param. No. | Sym. | Characteristic | Min. | Typ† | Max. | Units | Conditions |
| D001 | VDD | **Supply Voltage** | | | | | |
| | | | VDDMIN | | VDDMAX | | |
| | | | 1.8 | — | 3.6 | V | FOSC ≤ 16 MHz |
| | | | 2.5 | — | 3.6 | V | FOSC ≤ 20 MHz |
| D001 | | | 2.3 | — | 5.5 | V | FOSC ≤ 16 MHz |
| | | | 2.5 | — | 5.5 | V | FOSC ≤ 20 MHz |
| D002* | VDR | **RAM Data Retention Voltage(1)** | | | | | |
| | | | 1.5 | — | — | V | Device in Sleep mode |
| D002* | | | 1.7 | — | — | V | Device in Sleep mode |
| D002A* | VPOR | **Power-on Reset Release Voltage(2)** | | | | | |
| | | | — | 1.6 | — | V | |
| D002A* | | | — | 1.6 | — | V | |
| D002B* | VPORR* | **Power-on Reset Rearm Voltage(2)** | | | | | |
| | | | — | 0.8 | — | V | |
| D002B* | | | — | 1.5 | — | V | |
| D003 | VFVR | **Fixed Voltage Reference Voltage** | | | | | |
| | | 1x gain (1.024V nominal) | | | | | VDD ≥ 2.5V, -40°C ≤ TA ≤ +85°C |
| | | 2x gain (2.048V nominal) | -4 | — | +4 | % | VDD ≥ 2.5V, -40°C ≤ TA ≤ +85°C |
| | | 4x gain (4.096V nominal) | -3 | — | +7 | % | VDD ≥ 4.75V, -40°C ≤ TA ≤ +85°C |
| D004* | SVDD | **VDD Rise Rate(2)** | 0.05 | — | — | V/ms | Ensures that the Power-on Reset signal is released properly. |

* 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:** This is the limit to which VDD can be lowered in Sleep mode without losing RAM data.

**2:** See Figure 25-3, POR and POR REARM with Slow Rising VDD.

**FIGURE 26-3: IDD, EXTERNAL CLOCK (ECL), LOW-POWER MODE, FOSC = 500 kHz, PIC16LF1507 ONLY**

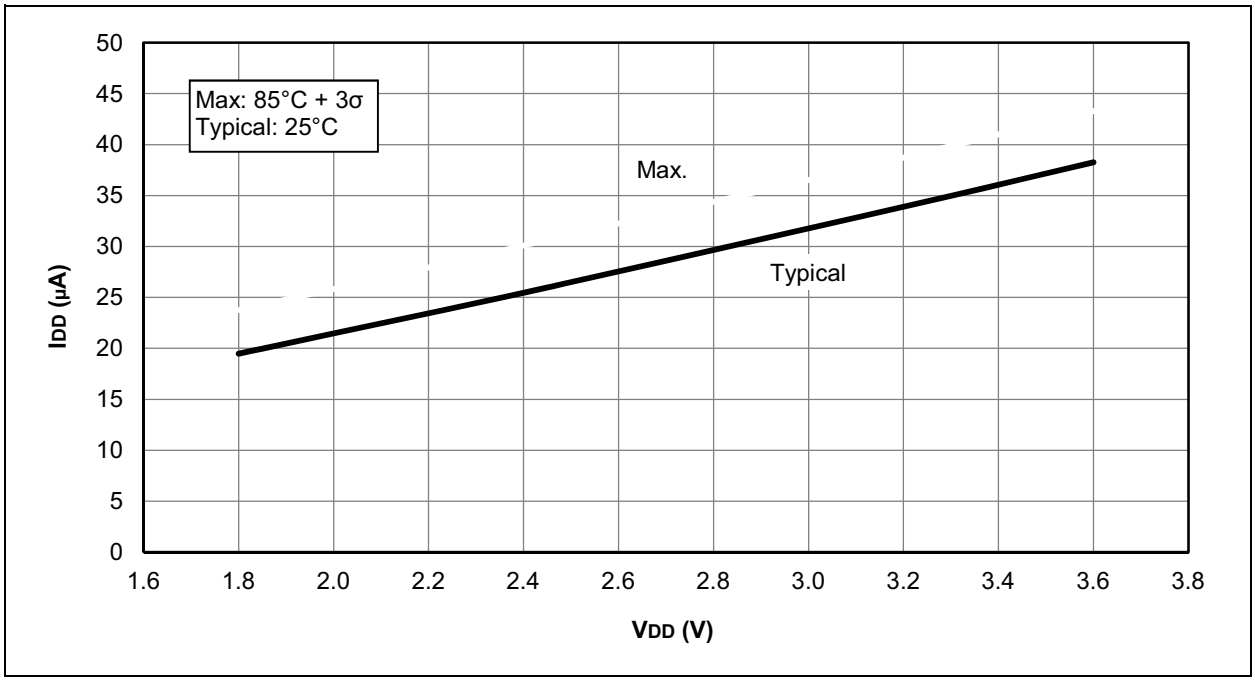



**FIGURE 26-4: IDD, EXTERNAL CLOCK (ECL), LOW-POWER MODE, FOSC = 500 kHz, PIC16F1507 ONLY**

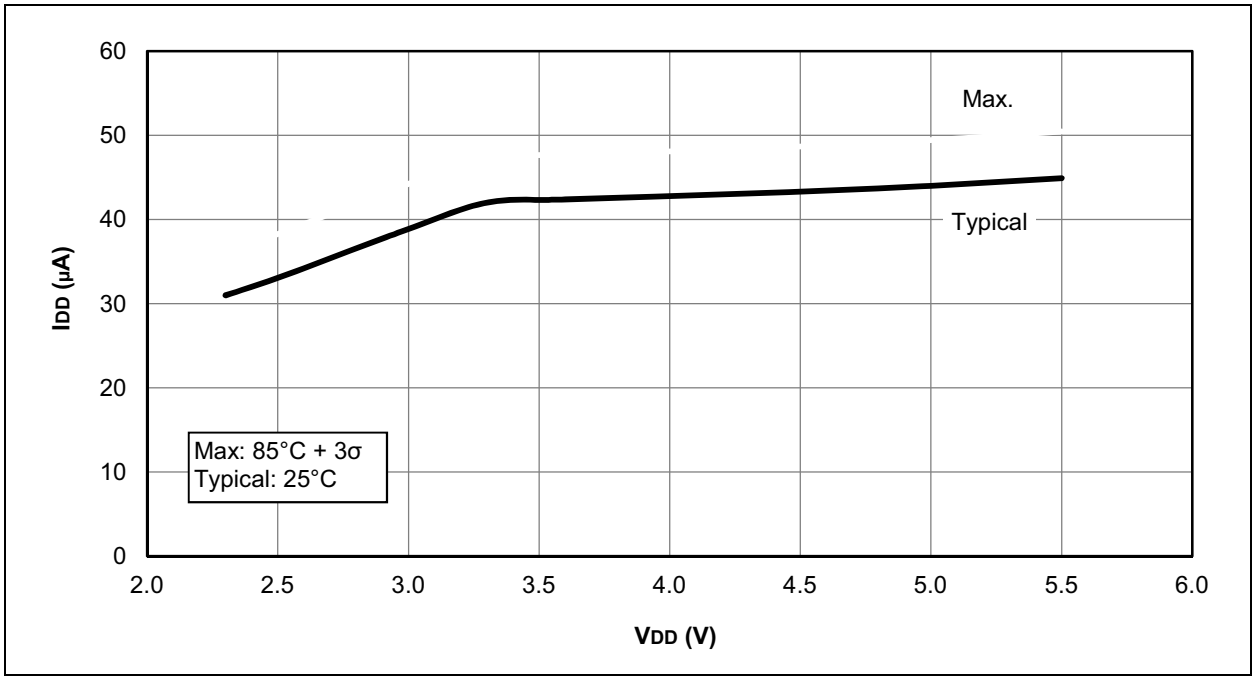


**FIGURE 26-5: IDD TYPICAL, EXTERNAL CLOCK (ECM), MEDIUM POWER MODE, PIC16LF1507 ONLY**

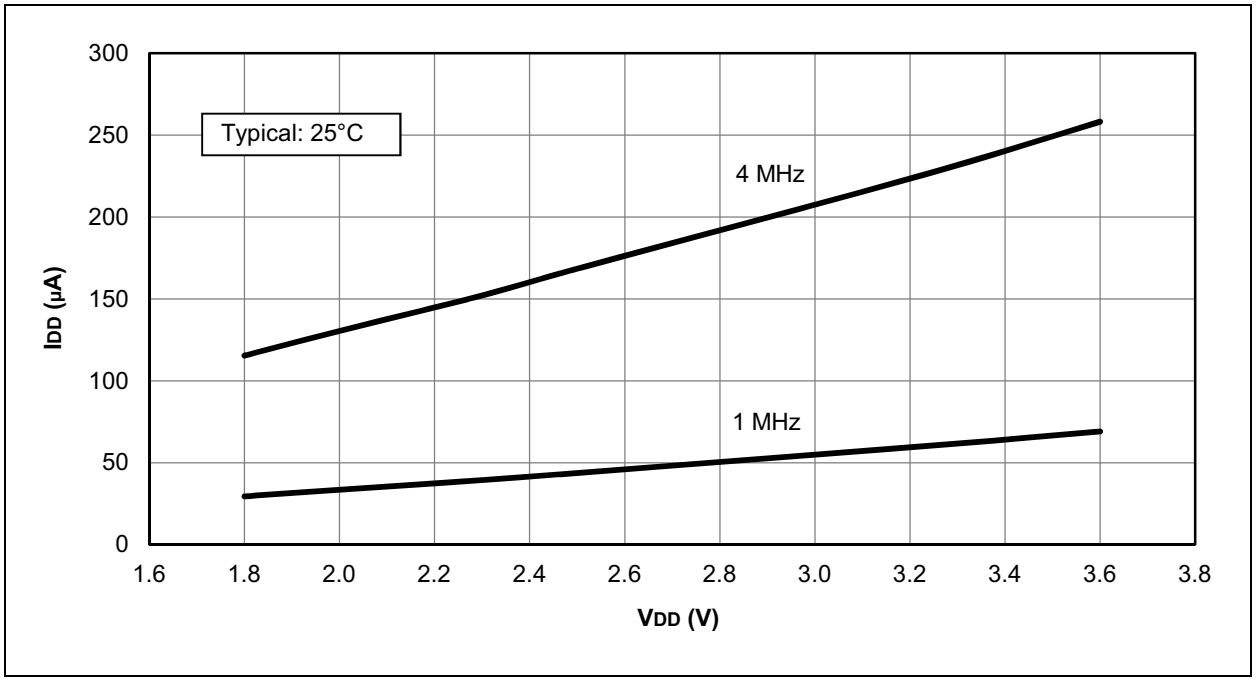



**FIGURE 26-6: IDD MAXIMUM, EXTERNAL CLOCK (ECM), MEDIUM POWER MODE, PIC16LF1507 ONLY**

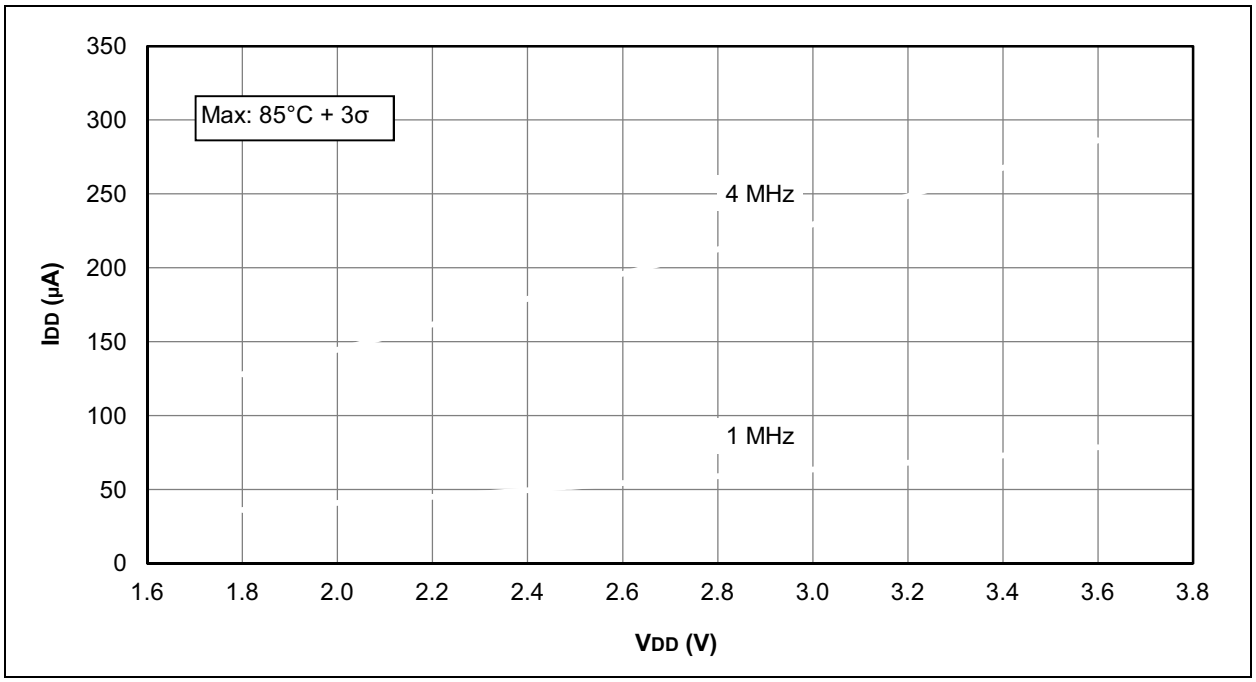


**FIGURE 26-49: LFINTOSC FREQUENCY OVER VDD AND TEMPERATURE, PIC16LF1507 ONLY**

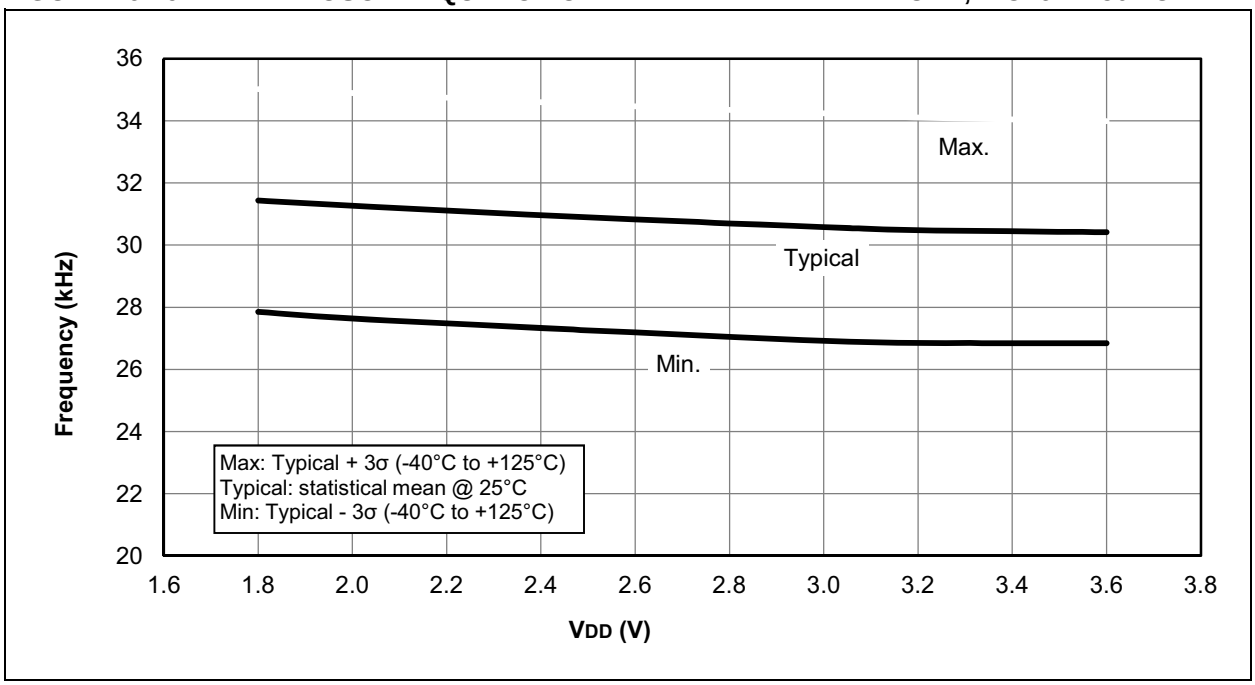



**FIGURE 26-50: LFINTOSC FREQUENCY OVER VDD AND TEMPERATURE, PIC16F1507 ONLY**