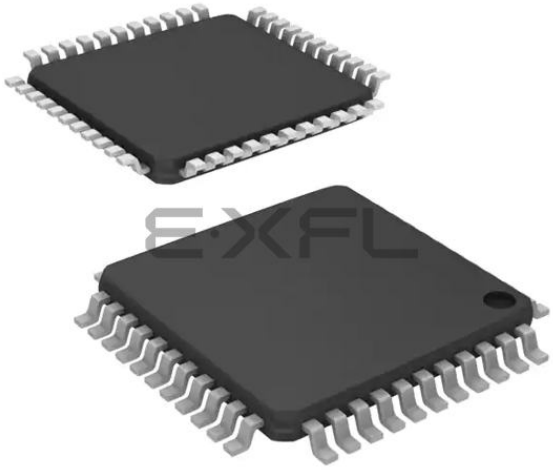


**Welcome to E-XFL.COM**

## 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 | 36 |
| Program Memory Size | 14KB (8K x 14) |
| Program Memory Type | FLASH |
| EEPROM Size | 128 x 8 |
| RAM Size | 1K x 8 |
| Voltage - Supply (Vcc/Vdd) | 2.3V ~ 5.5V |
| Data Converters | A/D 28x10b; D/A 4x5b, 4x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 125°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 44-TQFP |
| Supplier Device Package | 44-TQFP (10x10) |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic16f1777-e-pt |

## 3.0 MEMORY ORGANIZATION

These devices contain the following types of memory:

- Program Memory
  - Configuration Words
  - Device ID
  - User ID
  - Flash Program Memory
- Data Memory
  - Core Registers
  - Special Function Registers
  - General Purpose RAM
  - Common RAM

**Note 1:** The method to access Flash memory through the PMCON registers is described in **Section 10.0 "Flash Program Memory Control"**.

The following features are associated with access and control of program memory and data memory:

- PCL and PCLATH
- Stack
- Indirect Addressing

### 3.1 Program Memory Organization

The enhanced mid-range core has a 15-bit program counter capable of addressing a 32K x 14 program memory space. Table 3-1 shows the memory sizes implemented for the PIC16(L)F1777/8/9 family. Accessing a location above these boundaries will cause a wrap-around within the implemented memory space. The Reset vector is at 0000h and the interrupt vector is at 0004h (see Figures 3-1 and 3-2).

### 3.2 High-Endurance Flash

This device has a 128-byte section of high-endurance program Flash memory (PFM) in lieu of data EEPROM. This area is especially well suited for nonvolatile data storage that is expected to be updated frequently over the life of the end product. See **Section 10.2 "Flash Program Memory Overview"** for more information on writing data to PFM. See **Section 3.2.1.2 "Indirect Read with FSR"** for more information about using the FSR registers to read byte data stored in PFM.

**TABLE 3-1: DEVICE SIZES AND ADDRESSES**

| Device | Program Memory Space (Words) | Last Program Memory Address | High-Endurance Flash Memory Address Range(1) |
|---|---|---|---|
| PIC16(L)F1777 | 8,192 | 1FFFh | 1F80h-1FFFh |
| PIC16(L)F1778/9 | 16,384 | 3FFFh | 3F80h-3FFFh |

**Note 1:** High-endurance Flash applies to the low byte of each address in the range.

## 3.3 Data Memory Organization

The data memory is partitioned in 32 memory banks with 128 bytes in a bank. Each bank consists of (Figure 3-3):

- 12 core registers
- 20 Special Function Registers (SFR)
- Up to 80 bytes of General Purpose RAM (GPR)
- 16 bytes of common RAM

The active bank is selected by writing the bank number into the Bank Select Register (BSR). Unimplemented memory will read as '0'. All data memory can be accessed either directly (via instructions that use the file registers) or indirectly via the two File Select Registers (FSR). See **Section 3.7 "Indirect Addressing"** for more information.

Data memory uses a 12-bit address. The upper five bits of the address define the Bank address and the lower seven bits select the registers/RAM in that bank.

### 3.3.1 CORE REGISTERS

The core registers contain the registers that directly affect the basic operation. The core registers occupy the first 12 addresses of every data memory bank (addresses x00h/x08h through x0Bh/x8Bh). These registers are listed below in Table 3-2. For detailed information, see Table 3-17.

**TABLE 3-2: CORE REGISTERS**

| Addresses | BANKx |
|---|---|
| x00h or x80h | INDF0 |
| x01h or x81h | INDF1 |
| x02h or x82h | PCL |
| x03h or x83h | STATUS |
| x04h or x84h | FSR0L |
| x05h or x85h | FSR0H |
| x06h or x86h | FSR1L |
| x07h or x87h | FSR1H |
| x08h or x88h | BSR |
| x09h or x89h | WREG |
| x0Ah or x8Ah | PCLATH |
| x0Bh or x8Bh | INTCON |

#### 3.3.1.1 STATUS Register

The STATUS register, shown in Register 3-1, contains:

- the arithmetic status of the ALU
- the Reset status

The STATUS register can be the destination for any instruction, like any other register. If the STATUS register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. These bits are set or cleared according to the device logic. Furthermore, the $\overline{TO}$ and $\overline{PD}$ bits are not writable. Therefore, the result of an instruction with the STATUS register as destination may be different than intended.

For example, `CLRF STATUS` will clear the upper three bits and set the Z bit. This leaves the STATUS register as '`000u u1uu`' (where u = unchanged).

It is recommended, therefore, that only `BCF`, `BSF`, `SWAPF` and `MOVWF` instructions are used to alter the STATUS register, because these instructions do not affect any Status bits. For other instructions not affecting any Status bits (Refer to **Section 35.0 "Instruction Set Summary"**).

| **Note:** | The C and DC bits operate as $\overline{\text{Borrow}}$ and $\overline{\text{Digit Borrow}}$ out bits, respectively, in subtraction. |
|---|---|

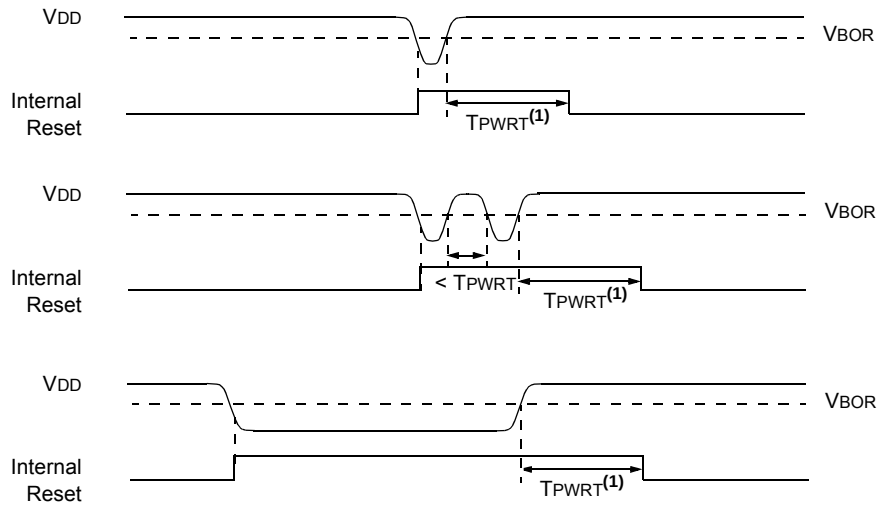
## TABLE 3-18: SPECIAL FUNCTION REGISTER SUMMARY

| Addr | 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 0** | | | | | | | | | | | |
| 00Ch | PORTA | RA7 | RA6 | RA5 | RA4 | RA3 | RA2 | RA1 | RA0 | xxxx xxxx | uuuu uuuu |
| 00Dh | PORTB | RB7 | RB6 | RB5 | RB4 | RB3 | RB2 | RB1 | RB0 | xxxx xxxx | uuuu uuuu |
| 00Eh | PORTC | RC7 | RC6 | RC5 | RC4 | RC3 | RC2 | RC1 | RC0 | xxxx xxxx | uuuu uuuu |
| 00Fh | PORTD(3) | RD7 | RD6 | RD5 | RD4 | RD3 | RD2 | RD1 | RD0 | xxxx xxxx | uuuu uuuu |
| 010h | PORTE | — | — | — | — | RE3 | RE2(3) | RE1(3) | RE0(3) | ---- xxxx | ---- uuuu |
| 011h | PIR1 | TMR1GIF | ADIF | RCIF | TXIF | SSP1IF | CCP1IF | TMR2IF | TMR1IF | 0000 0000 | 0000 0000 |
| 012h | PIR2 | OSFIF | C2IF | C1IF | COG1IF | BCL1IF | C4IF | C3IF | CCP2IF | 0000 0000 | 0000 0000 |
| 013h | PIR3 | — | — | COG2IF | ZCDIF | CLC4IF | CLC3IF | CLC2IF | CLC1IF | --00 0000 | --00 0000 |
| 014h | PIR4 | — | TMR8IF | TMR5GIF | TMR5IF | TMR3GIF | TMR3IF | TMR6IF | TRM4IF | -000 0000 | -000 0000 |
| 015h | PIR5 | CCP8IF(3) | CCP7IF | COG4IF(3) | COG3IF | C8IF(3) | C7IF(3) | C6IF | C5IF | 0000 --00 | 0000 --00 |
| 016h | PIR6 | — | — | — | — | PWM12IF(3) | PWM11IF | PWM6IF | PWM5IF | ---- 0000 | ---- 0000 |
| 017h | TMR0 | Timer0 Module Register | | | | | | | | 0000 0000 | 0000 0000 |
| 018h | TMR1L | Holding Register for the Least Significant Byte of the 16-bit TMR1 Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 019h | TMR1H | Holding Register for the Most Significant Byte of the 16-bit TMR1 Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 01Ah | T1CON | CS<1:0> | | CKPS<1:0> | | OSCEN | $\overline{\text{SYNC}}$ | — | ON | 0000 00-0 | uuuu uu-u |
| 01Bh | T1GCON | GE | GPOL | GTM | GSPM | GGO/DONE | GVAL | GSS<1:0> | | 0000 0x00 | uuuu uxuu |
| 01Ch | TMR3L | Holding Register for the Least Significant Byte of the 16-bit TMR3 Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 01Dh | TMR3H | Holding Register for the Most Significant Byte of the 16-bit TMR3 Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 01Eh | T3CON | CS<1:0> | | CKPS<1:0> | | OSCEN | $\overline{\text{SYNC}}$ | — | ON | 0000 00-0 | uuuu uu-u |
| 01Fh | T3GCON | GE | GPOL | GTM | GSPM | GGO/$\overline{\text{DONE}}$ | GVAL | GSS<1:0> | | 0000 0x00 | uuuu uxuu |

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

**Note 1:** Unimplemented, read as '1'.
**2:** Unimplemented on PIC16LF1777/8/9.
**3:** Unimplemented on PIC16(L)F1778.

**FIGURE 6-2: BROWN-OUT SITUATIONS**




**Note 1:** TPWRT delay only if $\overline{\text{PWRTE}}$ bit is programmed to '0'.

## 6.3 Register Definitions: BOR Control

**REGISTER 6-1: BORCON: BROWN-OUT RESET CONTROL REGISTER**

| R/W-1/u | R/W-0/u | U-0 | U-0 | U-0 | U-0 | U-0 | R-q/u |
|---|---|---|---|---|---|---|---|
| SBOREN | BORFS(1) | — | — | — | — | — | BORRDY |
| 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 | q = Value depends on condition |

bit 7 **SBOREN:** Software Brown-out Reset Enable bit
If BOREN <1:0> in Configuration Words ≠ 01:
SBOREN is read/write, but has no effect on the BOR.
If BOREN <1:0> in Configuration Words = 01:
1 = BOR Enabled
0 = BOR Disabled

bit 6 **BORFS:** Brown-out Reset Fast Start bit(1)
If BOREN<1:0> = 11 (Always on) or BOREN<1:0> = 00 (Always off)
BORFS is Read/Write, but has no effect.
If BOREN <1:0> = 10 (Disabled in Sleep) or BOREN<1:0> = 01 (Under software control):
1 = Band gap is forced on always (covers sleep/wake-up/operating cases)
0 = Band gap operates normally, and may turn off

bit 5-1 **Unimplemented:** Read as '0'

bit 0 **BORRDY:** Brown-out Reset Circuit Ready Status bit
1 = The Brown-out Reset circuit is active
0 = The Brown-out Reset circuit is inactive

**Note 1:** BOREN<1:0> bits are located in Configuration Words.

**EXAMPLE 10-2: ERASING ONE ROW OF PROGRAM MEMORY**

```
; This row erase routine assumes the following:
; 1. A valid address within the erase row is loaded in ADDRH:ADDRL
; 2. ADDRH and ADDRL are located in shared data memory 0x70 - 0x7F (common RAM)


        BCF         INTCON,GIE      ; Disable ints so required sequences will execute properly
        BANKSEL     PMADRL
        MOVF        ADDRL,W         ; Load lower 8 bits of erase address boundary
        MOVWF       PMADRL
        MOVF        ADDRH,W         ; Load upper 6 bits of erase address boundary
        MOVWF       PMADRH
        BCF         PMCON1,CFGS     ; Not configuration space
        BSF         PMCON1,FREE     ; Specify an erase operation
        BSF         PMCON1,WREN     ; Enable writes

Required
Sequence
        MOVLW       55h             ; Start of required sequence to initiate erase
        MOVWF       PMCON2          ; Write 55h
        MOVLW       0AAh            ;
        MOVWF       PMCON2          ; Write AAh
        BSF         PMCON1,WR       ; Set WR bit to begin erase
        NOP                         ; NOP instructions are forced as processor starts
        NOP                         ; row erase of program memory.
                                    ;
                                    ; The processor stalls until the erase process is complete
                                    ; after erase processor continues with 3rd instruction

        BCF         PMCON1,WREN     ; Disable writes
        BSF         INTCON,GIE      ; Enable interrupts
```

**REGISTER 11-32: SLRCOND: PORTD SLEW RATE CONTROL REGISTER**

| R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 |
|---|---|---|---|---|---|---|---|
| SLRD7 | SLRD6 | SLRD5 | SLRD4 | SLRD3 | SLRD2 | SLRD1 | SLRD0 |
| 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 7-0 **SLRD<7:0>:** PORTD Slew Rate Enable bits
For RD<7:0> pins
1 = Port pin slew rate is limited
0 = Port pin slews at maximum rate

**REGISTER 11-33: INLVLD: PORTD INPUT LEVEL CONTROL REGISTER**

| R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 |
|---|---|---|---|---|---|---|---|
| INLVLD7 | INLVLD6 | INLVLD5 | INLVLD4 | INLVLD3 | INLVLD2 | INLVLD1 | INLVLD0 |
| 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 7-0 **INLVLD<7:0>:** PORTD Input Level Select bits
For RD<7:0> pins
1 = Port pin digital input operates with ST thresholds
0 = Port pin digital input operates with TTL thresholds

## 12.0 PERIPHERAL PIN SELECT (PPS) MODULE

The Peripheral Pin Select (PPS) module connects peripheral inputs and outputs to the device I/O pins. Only digital signals are included in the selections. All analog inputs and outputs remain fixed to their assigned pins. Input and output selections are independent as shown in the simplified block diagram Figure 12-1.

### 12.1 PPS Inputs

Each peripheral has a PPS register with which the inputs to the peripheral are selected. Inputs include the device pins.

Multiple peripherals can operate from the same source simultaneously. Port reads always return the pin level regardless of peripheral PPS selection. If a pin also has associated analog functions, the ANSEL bit for that pin must be cleared to enable the digital input buffer.

Although every peripheral has its own PPS input selection register, the selections are identical for every peripheral as shown in Register 12-1.

| **Note:** | The notation “xxx” in the register name is a place holder for the peripheral identifier. For example, CLC1PPS. |
|---|---|

### 12.2 PPS Outputs

Each I/O pin has a PPS register with which the pin output source is selected. With few exceptions, the port TRIS control associated with that pin retains control over the pin output driver. Peripherals that control the pin output driver as part of the peripheral operation will override the TRIS control as needed. These peripherals include:

- EUSART (synchronous operation)
- MSSP ($I^2C$)
- COG (auto-shutdown)

Although every pin has its own PPS peripheral selection register, the selections are identical for every pin as shown in Register 12-2.

| **Note:** | The notation “Rxy” is a place holder for the pin identifier. For example, RA0PPS. |
|---|---|

**FIGURE 12-1: SIMPLIFIED PPS BLOCK DIAGRAM**

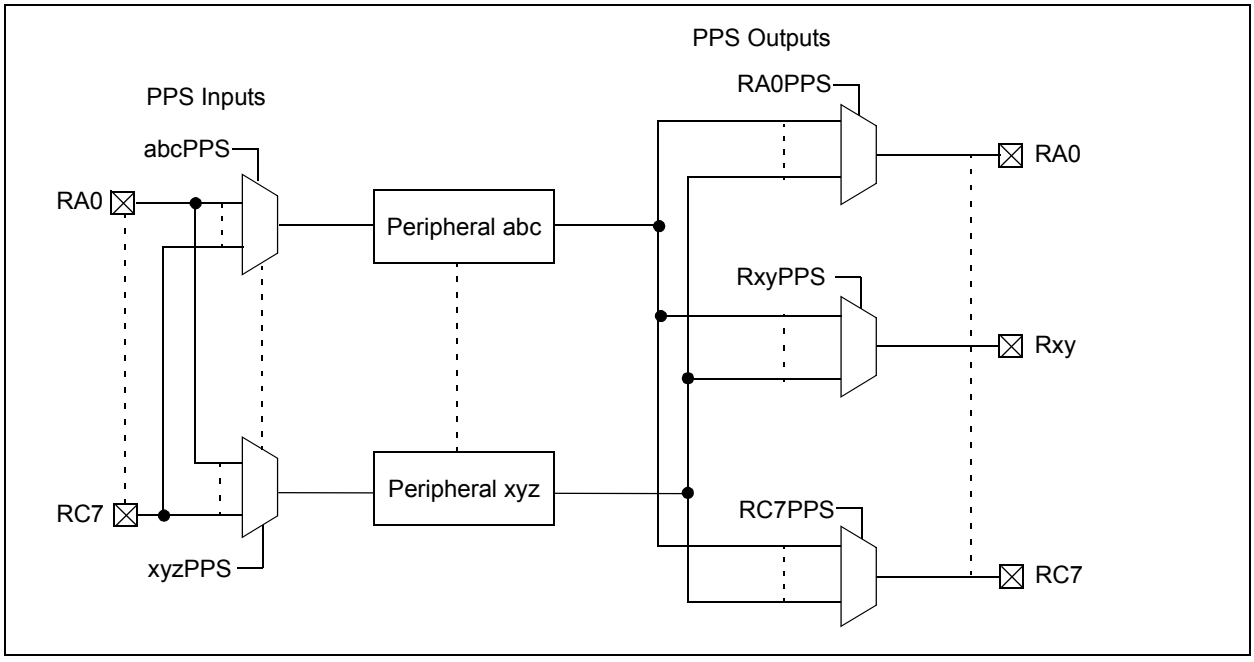
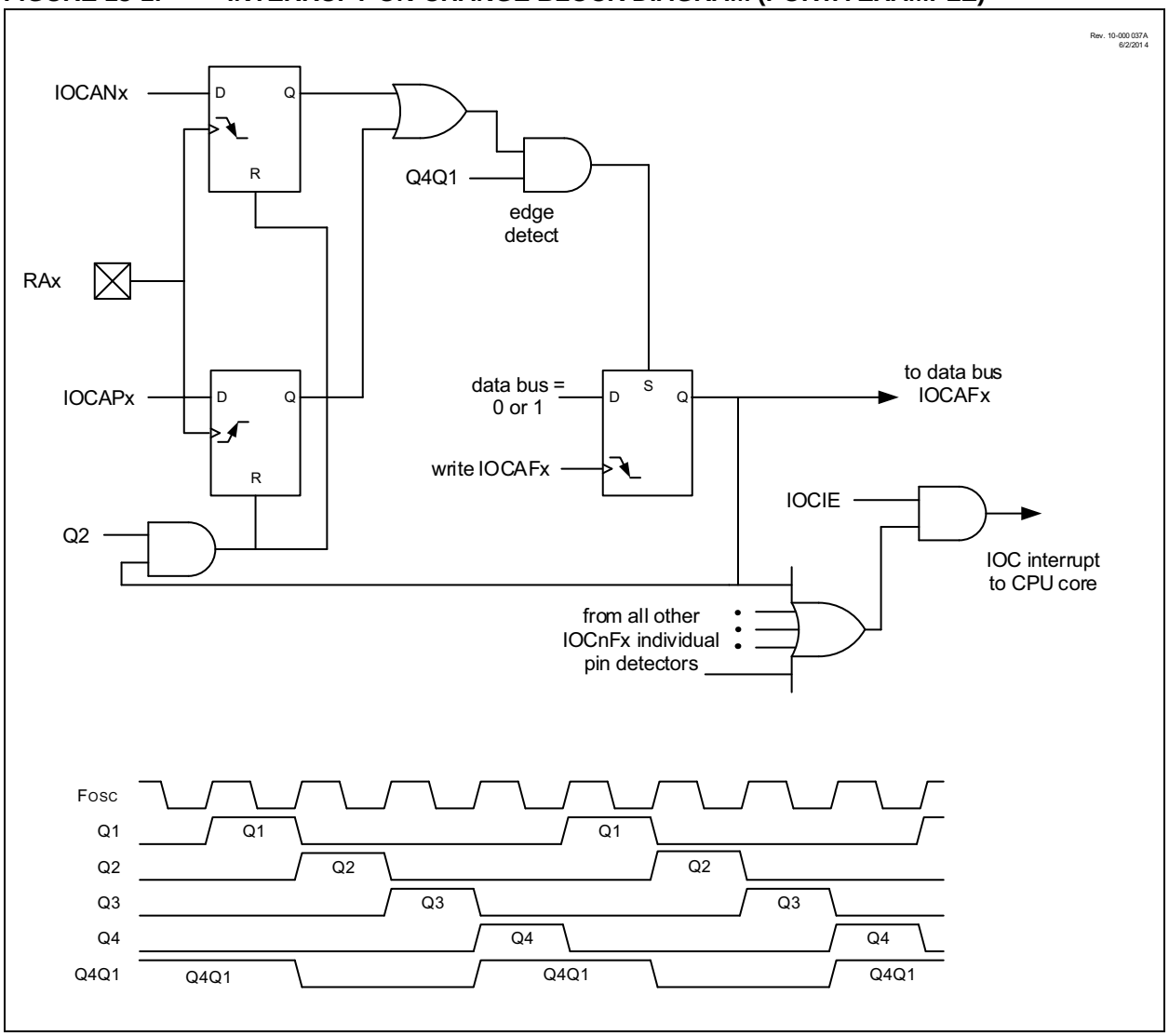


**FIGURE 13-1: INTERRUPT-ON-CHANGE BLOCK DIAGRAM (PORTA EXAMPLE)**



**REGISTER 13-7: IOCCP: INTERRUPT-ON-CHANGE PORTC POSITIVE EDGE REGISTER**

| R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| IOCCP7 | IOCCP6 | IOCCP5 | IOCCP4 | IOCCP3 | IOCCP2 | IOCCP1 | IOCCP0 |
| 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 7-0 **IOCCP<7:0>:** Interrupt-on-Change PORTC Positive Edge Enable bits
1 = Interrupt-on-Change enabled on the pin for a positive going edge. IOCCFx bit and IOCIF flag will be set upon detecting an edge.
0 = Interrupt-on-Change disabled for the associated pin.

**REGISTER 13-8: IOCCN: INTERRUPT-ON-CHANGE PORTC NEGATIVE EDGE REGISTER**

| R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| IOCCN7 | IOCCN6 | IOCCN5 | IOCCN4 | IOCCN3 | IOCCN2 | IOCCN1 | IOCCN0 |
| 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 7-0 **IOCCN<7:0>:** Interrupt-on-Change PORTC Negative Edge Enable bits
1 = Interrupt-on-Change enabled on the pin for a negative going edge. IOCCFx bit and IOCIF flag will be set upon detecting an edge.
0 = Interrupt-on-Change disabled for the associated pin.

**REGISTER 13-9: IOCCF: INTERRUPT-ON-CHANGE PORTC FLAG REGISTER**

| R/W/HS-0/0 | R/W/HS-0/0 | R/W/HS-0/0 | R/W/HS-0/0 | R/W/HS-0/0 | R/W/HS-0/0 | R/W/HS-0/0 | R/W/HS-0/0 |
|---|---|---|---|---|---|---|---|
| IOCCF7 | IOCCF6 | IOCCF5 | IOCCF4 | IOCCF3 | IOCCF2 | IOCCF1 | IOCCF0 |
| 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 | HS - Bit is set in hardware |

bit 7-0 **IOCCF<7:0>:** Interrupt-on-Change PORTC Flag bits
1 = An enabled change was detected on the associated pin.
Set when IOCCPx = 1 and a rising edge was detected on RCx, or when IOCCNx = 1 and a falling edge was detected on RCx.
0 = No change was detected, or the user cleared the detected change.

## 14.4 Register Definitions: FVR Control

**REGISTER 14-1: FVRCON: FIXED VOLTAGE REFERENCE CONTROL REGISTER**

| R/W-0/0 | R-q/q | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| FVREN | FVRRDY(1) | TSEN(3) | TSRNG(3) | CDAFVR<1:0> | | ADFVR<1:0> | |
| bit 7 | | | | | | | bit 0 |

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

bit 7 **FVREN:** Fixed Voltage Reference Enable bit
1 = Fixed Voltage Reference is enabled
0 = Fixed Voltage Reference is disabled

bit 6 **FVRRDY:** Fixed Voltage Reference Ready Flag bit(1)
1 = Fixed Voltage Reference output is ready for use
0 = Fixed Voltage Reference output is not ready or not enabled

bit 5 **TSEN:** Temperature Indicator Enable bit(3)
1 = Temperature Indicator is enabled
0 = Temperature Indicator is disabled

bit 4 **TSRNG:** Temperature Indicator Range Selection bit(3)
1 = $V_{OUT} = V_{DD} - 4V_T$ (High Range)
0 = $V_{OUT} = V_{DD} - 2V_T$ (Low Range)

bit 3-2 **CDAFVR<1:0>:** Comparator/DAC FVR Buffer Gain Selection bits
11 = Comparator/DAC FVR Buffer Gain is 4x, with output $V_{CDAFVR} = 4x\ V_{FVR}$(2)
10 = Comparator/DAC FVR Buffer Gain is 2x, with output $V_{CDAFVR} = 2x\ V_{FVR}$(2)
01 = Comparator/DAC FVR Buffer Gain is 1x, with output $V_{CDAFVR} = 1x\ V_{FVR}$
00 = Comparator/DAC FVR Buffer is off

bit 1-0 **ADFVR<1:0>:** ADC FVR Buffer Gain Selection bits
11 = ADC FVR Buffer Gain is 4x, with output $V_{ADFVR} = 4x\ V_{FVR}$(2)
10 = ADC FVR Buffer Gain is 2x, with output $V_{ADFVR} = 2x\ V_{FVR}$(2)
01 = ADC FVR Buffer Gain is 1x, with output $V_{ADFVR} = 1x\ V_{FVR}$
00 = ADC FVR Buffer is off

**Note 1:** FVRRDY is always '1' on PIC16F1773/6 only.
**2:** Fixed Voltage Reference output cannot exceed $V_{DD}$.
**3:** See **Section 15.0 "Temperature Indicator Module"** for additional information.

**TABLE 14-2: SUMMARY OF REGISTERS ASSOCIATED WITH FIXED VOLTAGE REFERENCE**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on page |
|---|---|---|---|---|---|---|---|---|---|
| FVRCON | FVREN | FVRRDY | TSEN | TSRNG | CDAFVR<1:0> | | ADFVR<1:0> | | 223 |

**Legend:** Shaded cells are not used with the Fixed Voltage Reference.

### 21.1.3 SOFTWARE PROGRAMMABLE PRESCALER

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

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

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

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

### 21.1.4 TIMER0 INTERRUPT

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

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

### 21.1.5 8-BIT COUNTER MODE SYNCHRONIZATION

When in 8-Bit Counter mode, the incrementing edge on the T0CKI pin must be synchronized to the instruction clock. Synchronization can be accomplished by sampling the prescaler output on the Q2 and Q4 cycles of the instruction clock. The high and low periods of the external clocking source must meet the timing requirements as shown in Table 36-12: Timer0 and Timer1 External Clock Requirements.

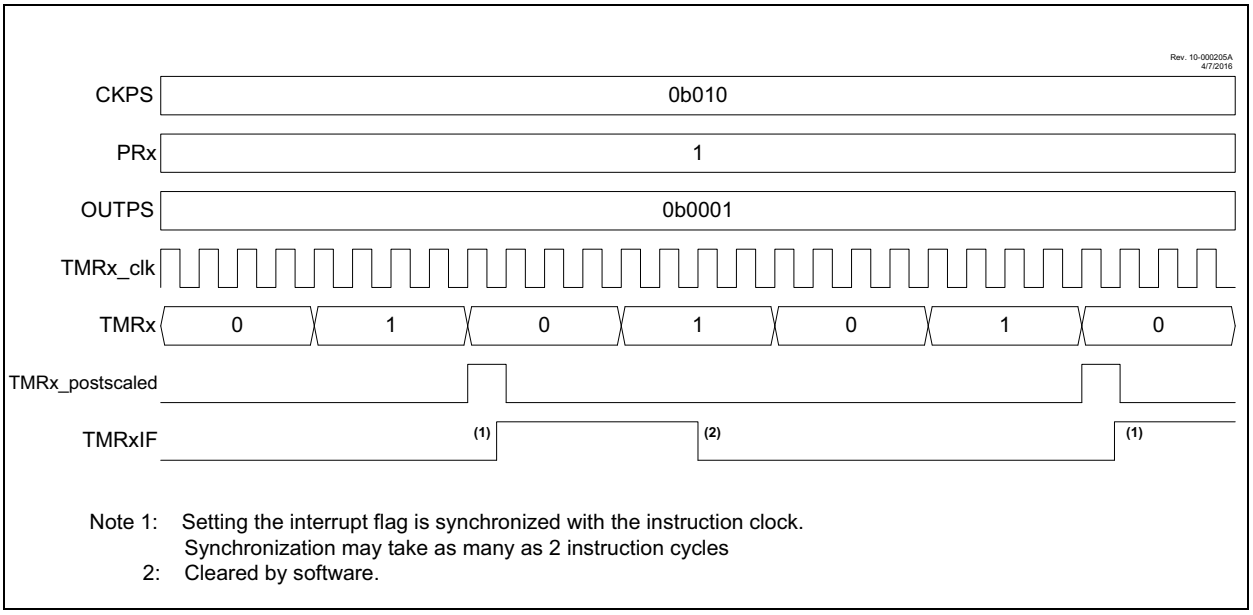
### 21.1.6 OPERATION DURING SLEEP

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

## 23.5 Timer2 Interrupt

Timer2 can also generate a device interrupt. The interrupt is generated when the postscaler counter matches one of 16 postscale options (from 1:1 through 1:16), which are selected with the postscaler control bits, OUTPS<3:0> of the T2CON register. The interrupt is enabled by setting the TMR2IE interrupt enable bit of the PIE1 register. Interrupt timing is illustrated in Figure 23-3.

**FIGURE 23-3: TIMER2 PRESCALER, POSTSCALER, AND INTERRUPT TIMING DIAGRAM**




Note 1: Setting the interrupt flag is synchronized with the instruction clock. Synchronization may take as many as 2 instruction cycles
2: Cleared by software.

## 24.3 PWM Overview

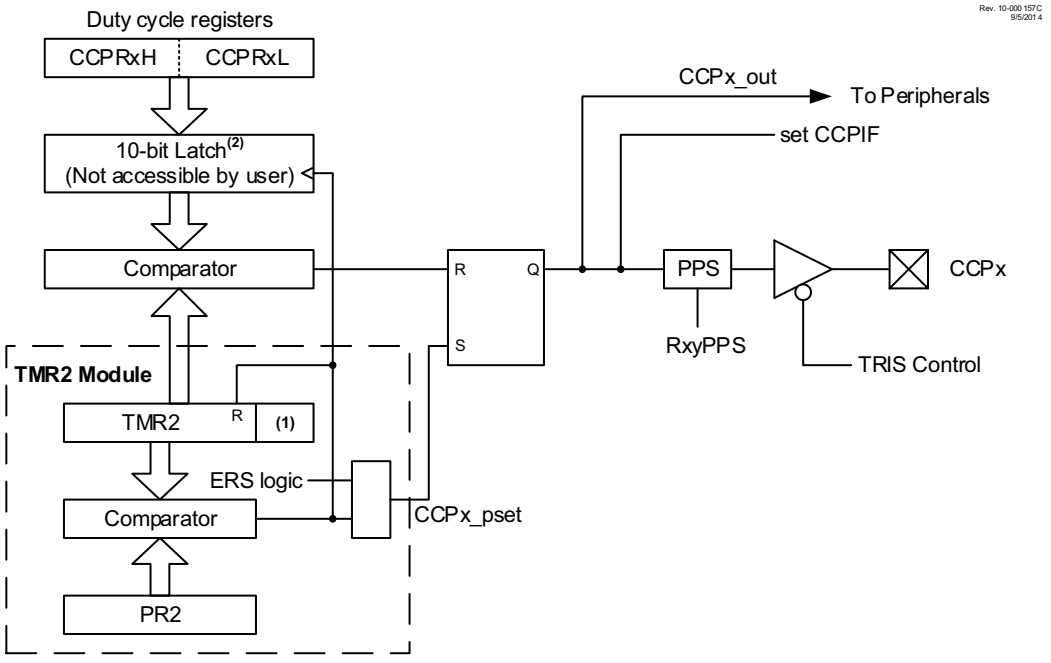
Pulse-Width Modulation (PWM) is a scheme that provides power to a load by switching quickly between fully on and fully off states. The PWM signal resembles a square wave where the high portion of the signal is considered the on state and the low portion of the signal is considered the off state. The high portion, also known as the pulse width, can vary in time and is defined in steps. A larger number of steps applied, which lengthens the pulse width, also supplies more power to the load. Lowering the number of steps applied, which shortens the pulse width, supplies less power. The PWM period is defined as the duration of one complete cycle or the total amount of on and off time combined.

PWM resolution defines the maximum number of steps that can be present in a single PWM period. A higher resolution allows for more precise control of the pulse width time and in turn the power that is applied to the load.

The term duty cycle describes the proportion of the on time to the off time and is expressed in percentages, where 0% is fully off and 100% is fully on. A lower duty cycle corresponds to less power applied and a higher duty cycle corresponds to more power applied.

Figure 24-3 shows a typical waveform of the PWM signal.

**FIGURE 24-3: SIMPLIFIED PWM BLOCK DIAGRAM**




Notes: 1. 8-bit timer is concatenated with two bits generated by Fosc or two bits of the internal prescaler to create 10-bit time-base.
2. The alignment of the 10 bits from the CCPR register is determined by the FMT bit.

#### 32.4.9 ACKNOWLEDGE SEQUENCE

The ninth SCL pulse for any transferred byte in $I^2C$ is dedicated as an Acknowledge. It allows receiving devices to respond back to the transmitter by pulling the SDA line low. The transmitter must release control of the line during this time to shift in the response. The Acknowledge ($\overline{ACK}$) is an active-low signal, pulling the SDA line low indicates to the transmitter that the device has received the transmitted data and is ready to receive more.

The result of an $\overline{ACK}$ is placed in the ACKSTAT bit of the SSPxCON2 register.

Slave software, when the AHEN and DHEN bits are set, allow the user to set the $\overline{ACK}$ value sent back to the transmitter. The ACKDT bit of the SSPxCON2 register is set/cleared to determine the response.

Slave hardware will generate an $\overline{ACK}$ response if the AHEN and DHEN bits of the SSPxCON3 register are clear.

There are certain conditions where an $\overline{ACK}$ will not be sent by the slave. If the BF bit of the SSPxSTAT register or the SSPOV bit of the SSPxCON1 register are set when a byte is received.

When the module is addressed, after the eighth falling edge of SCL on the bus, the ACKTIM bit of the SSPxCON3 register is set. The ACKTIM bit indicates the acknowledge time of the active bus. The ACKTIM Status bit is only active when the AHEN bit or DHEN bit is enabled.

## 32.5 $I^2C$ SLAVE MODE OPERATION

The MSSP Slave mode operates in one of four modes selected by the SSPM bits of SSPxCON1 register. The modes can be divided into 7-bit and 10-bit Addressing mode. 10-bit Addressing modes operate the same as 7-bit with some additional overhead for handling the larger addresses.

Modes with Start and Stop bit interrupts operate the same as the other modes with SSPxIF additionally getting set upon detection of a Start, Restart, or Stop condition.

#### 32.5.1 SLAVE MODE ADDRESSES

The SSPxADD register (Register 32-6) contains the Slave mode address. The first byte received after a Start or Restart condition is compared against the value stored in this register. If the byte matches, the value is loaded into the SSPxBUF register and an interrupt is generated. If the value does not match, the module goes idle and no indication is given to the software that anything happened.

The SSP Mask register (Register 32-5) affects the address matching process. See **Section 32.5.8 "SSP Mask Register"** for more information.

##### 32.5.1.1 $I^2C$ Slave 7-bit Addressing Mode

In 7-bit Addressing mode, the LSb of the received data byte is ignored when determining if there is an address match.

##### 32.5.1.2 $I^2C$ Slave 10-bit Addressing Mode

In 10-bit Addressing mode, the first received byte is compared to the binary value of '1 1 1 1 0 A9 A8 0'. A9 and A8 are the two MSb's of the 10-bit address and stored in bits 2 and 1 of the SSPxADD register.

After the acknowledge of the high byte the UA bit is set and SCL is held low until the user updates SSPxADD with the low address. The low address byte is clocked in and all eight bits are compared to the low address value in SSPxADD. Even if there is not an address match; SSPxIF and UA are set, and SCL is held low until SSPxADD is updated to receive a high byte again. When SSPxADD is updated the UA bit is cleared. This ensures the module is ready to receive the high address byte on the next communication.

A high and low address match as a write request is required at the start of all 10-bit addressing communication. A transmission can be initiated by issuing a Restart once the slave is addressed, and clocking in the high address with the R/$\overline{W}$ bit set. The slave hardware will then acknowledge the read request and prepare to clock out data. This is only valid for a slave after it has received a complete high and low address byte match.

**REGISTER 33-3: BAUD1CON: BAUD RATE CONTROL REGISTER**

| R-0/0 | R-1/1 | U-0 | R/W-0/0 | R/W-0/0 | U-0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| ABDOVF | RCIDL | — | SCKP | BRG16 | — | WUE | ABDEN |
| 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 **ABDOVF:** Auto-Baud Detect Overflow bit
Asynchronous mode:
1 = Auto-baud timer overflowed
0 = Auto-baud timer did not overflow
Synchronous mode:
Don't care

bit 6 **RCIDL**: Receive Idle Flag bit
Asynchronous mode:
1 = Receiver is Idle
0 = Start bit has been received and the receiver is receiving
Synchronous mode:
Don't care

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

bit 4 **SCKP**: Synchronous Clock Polarity Select bit
Asynchronous mode:
1 = Transmit inverted data to the TX/CK pin
0 = Transmit non-inverted data to the TX/CK pin
Synchronous mode:
1 = Data is clocked on rising edge of the clock
0 = Data is clocked on falling edge of the clock

bit 3 **BRG16:** 16-bit Baud Rate Generator bit
1 = 16-bit Baud Rate Generator is used
0 = 8-bit Baud Rate Generator is used

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

bit 1 **WUE:** Wake-up Enable bit
Asynchronous mode:
1 = Receiver is waiting for a falling edge. No character will be received, byte RCIF will be set. WUE will automatically clear after RCIF is set.
0 = Receiver is operating normally
Synchronous mode:
Don't care

bit 0 **ABDEN**: Auto-Baud Detect Enable bit
Asynchronous mode:
1 = Auto-Baud Detect mode is enabled (clears when auto-baud is complete)
0 = Auto-Baud Detect mode is disabled
Synchronous mode:
Don't care

**RETFIE** **Return from Interrupt**

| | |
|---|---|
| Syntax: | [ *label* ] RETFIE k |
| Operands: | None |
| Operation: | TOS → PC,<br>1 → GIE |
| Status Affected: | None |
| Description: | Return from Interrupt. Stack is POPed and Top-of-Stack (TOS) is loaded in the PC. Interrupts are enabled by setting Global Interrupt Enable bit, GIE (INTCON<7>). This is a 2-cycle instruction. |
| Words: | 1 |
| Cycles: | 2 |
| Example: | `RETFIE`<br>After Interrupt<br>PC = TOS<br>GIE = 1 |

**RETLW** **Return with literal in W**

| | |
|---|---|
| Syntax: | [ *label* ] RETLW k |
| Operands: | 0 ≤ k ≤ 255 |
| Operation: | k → (W);<br>TOS → PC |
| Status Affected: | None |
| Description: | The W register is loaded with the 8-bit literal 'k'. The program counter is loaded from the top of the stack (the return address). This is a 2-cycle instruction. |
| Words: | 1 |
| Cycles: | 2 |
| Example: | `CALL TABLE;W contains table`<br>`;offset value`<br>`;W now has table value`<br>`TABLE`<br>`ADDWF PC ;W = offset`<br>`RETLW k1 ;Begin table`<br>`RETLW k2 ;`<br>`RETLW kn ; End of table`<br>Before Instruction<br>W = 0x07<br>After Instruction<br>W = value of k8 |

**RETURN** **Return from Subroutine**

| | |
|---|---|
| Syntax: | [ *label* ] RETURN |
| Operands: | None |
| Operation: | TOS → PC |
| Status Affected: | None |
| Description: | Return from subroutine. The stack is POPed and the top of the stack (TOS) is loaded into the program counter. This is a 2-cycle instruction. |

**RLF** **Rotate Left f through Carry**

| | |
|---|---|
| Syntax: | [ *label* ] RLF f,d |
| Operands: | 0 ≤ f ≤ 127<br>d ∈ [0,1] |
| Operation: | See description below |
| Status Affected: | C |
| Description: | The contents of register 'f' are rotated one bit to the left through the Carry flag. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is stored back in register 'f'. |
| Words: | 1 |
| Cycles: | 1 |
| Example: | `RLF REG1,0`<br>Before Instruction<br>REG1 = 1110 0110<br>C = 0<br>After Instruction<br>REG1 = 1110 0110<br>W = 1100 1100<br>C = 1 |

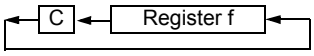


**TABLE 36-3: POWER-DOWN CURRENTS ($I_{PD}$)(1,2)**

| PIC16LF1777/8/9 | | Operating Conditions: (unless otherwise stated)<br>Low-Power Sleep Mode | | | | | | |
|---|---|---|---|---|---|---|---|---|
| PIC16F1777/8/9 | | Low-Power Sleep Mode, VREGPM = 1 | | | | | | |
| Param No. | Device Characteristics | Min. | Typ† | Max. +85°C | Max. +125°C | Units | Conditions | |
| | | | | | | | $V_{DD}$ | Note |
| D023 | Base $I_{PD}$ | — | 0.05 | 1.0 | 8.0 | µA | 1.8 | WDT, BOR, FVR, and SOSC disabled, all Peripherals Inactive |
| | | — | 0.08 | 2.0 | 9.0 | µA | 3.0 | |
| D023 | Base $I_{PD}$ | — | 0.3 | 2.4 | 10 | µA | 2.3 | WDT, BOR, FVR, and SOSC disabled, all Peripherals Inactive, Low-Power Sleep mode |
| | | — | 0.4 | 4 | 12 | µA | 3.0 | |
| | | — | 0.5 | 6 | 15 | µA | 5.0 | |
| D023A | Base $I_{PD}$ | — | 9.8 | 17 | 28 | µA | 2.3 | WDT, BOR, FVR and SOSC disabled, all Peripherals inactive, Normal Power Sleep mode VREGPM = 0 |
| | | — | 10.3 | 20 | 40 | µA | 3.0 | |
| | | — | 11.5 | 22 | 44 | µA | 5.0 | |
| D024 | | — | 0.5 | 6 | 14 | µA | 1.8 | WDT Current |
| | | — | 0.8 | 7 | 17 | µA | 3.0 | |
| D024 | | — | 0.8 | 6 | 15 | µA | 2.3 | WDT Current |
| | | — | 0.9 | 7 | 20 | µA | 3.0 | |
| | | — | 1.0 | 8 | 22 | µA | 5.0 | |
| D025 | | — | 15 | 28 | 30 | µA | 1.8 | FVR Current (ADC) |
| | | — | 24 | 35 | 38 | µA | 3.0 | |
| D025 | | — | 18 | 33 | 35 | µA | 2.3 | FVR Current (ADC) |
| | | — | 24 | 35 | 40 | µA | 3.0 | |
| | | — | 26 | 37 | 44 | µA | 5.0 | |
| D025A | | — | 25 | 50 | 55 | µA | 1.8 | FVR Current (DAC) |
| | | — | 30 | 65 | 70 | µA | 3.0 | |
| D025A | | — | 30 | 55 | 66 | µA | 2.3 | FVR Current (DAC) |
| | | — | 32 | 68 | 82 | µA | 3.0 | |
| | | — | 35 | 77 | 90 | µA | 5.0 | |
| D026 | | — | 7.5 | 25 | 28 | µA | 3.0 | BOR Current |
| D026 | | — | 10 | 25 | 28 | µA | 3.0 | BOR Current |
| | | — | 12 | 28 | 31 | µA | 5.0 | |
| D027 | | — | 0.5 | 4 | 10 | µA | 3.0 | LPBOR Current |
| D027 | | — | 0.8 | 6 | 15 | µA | 3.0 | LPBOR Current |
| | | — | 1 | 8 | 17 | µA | 5.0 | |
| D028 | | — | 0.5 | 5 | 9 | µA | 1.8 | SOSC Current |
| | | — | 0.8 | 8.5 | 12 | µA | 3.0 | |
| D028 | | — | 1.1 | 6 | 10 | µA | 2.3 | SOSC Current |
| | | — | 1.3 | 8.5 | 20 | µA | 3.0 | |
| | | — | 1.4 | 10 | 25 | 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 peripheral current is the sum of the base $I_{PD}$ and the additional current consumed when this peripheral is enabled. The peripheral Δ current can be determined by subtracting the base $I_{DD}$ or $I_{PD}$ current from this limit. Max values should be used when calculating total current consumption.

**2:** The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to $V_{SS}$.

**3:** ADC clock source is FRC.

**TABLE 36-9: PLL CLOCK TIMING SPECIFICATIONS**

| Standard Operating Conditions (unless otherwise stated) | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym.** | **Characteristic** | **Min.** | **Typ†** | **Max.** | **Units** | **Conditions** |
| F10 | FOSC | Oscillator Frequency Range | 4 | — | 8 | MHz | |
| F11 | FSYS | On-Chip VCO System Frequency | 16 | — | 32 | MHz | |
| F12 | TRC | PLL Start-up Time (Lock Time) | — | — | 2 | ms | |
| F13* | ΔCLK | CLKOUT Stability (Jitter) | -0.25% | — | +0.25% | % | |

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**FIGURE 36-7: CLKOUT AND I/O TIMING**

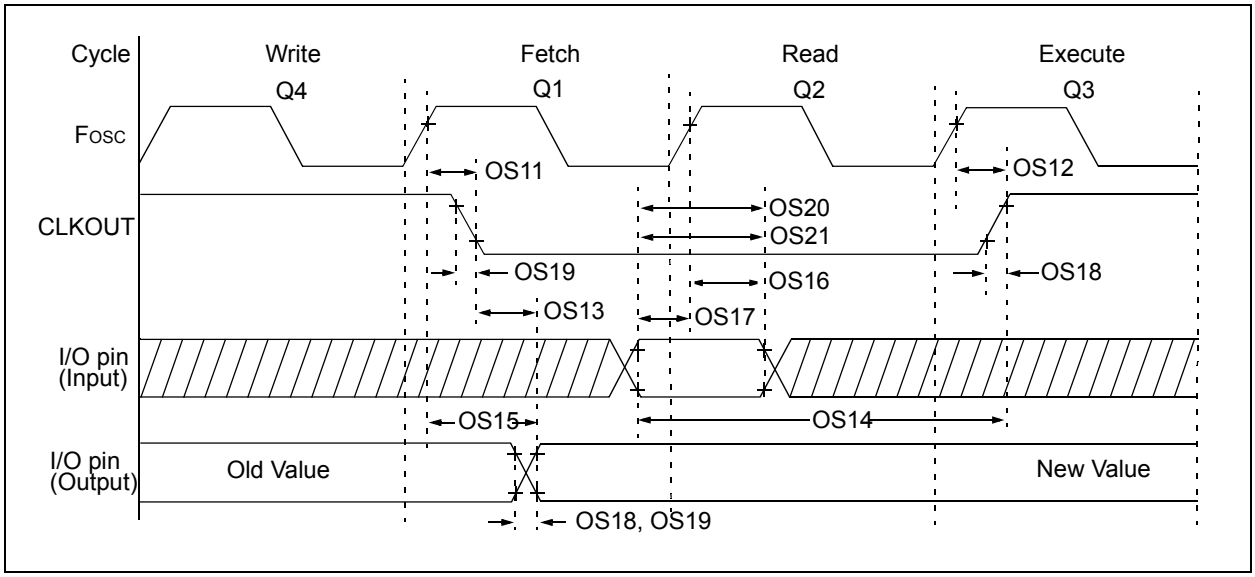



**TABLE 36-10: CLKOUT AND I/O TIMING PARAMETERS**

| Standard Operating Conditions (unless otherwise stated) | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym.** | **Characteristic** | **Min.** | **Typ†** | **Max.** | **Units** | **Conditions** |
| OS11 | TosH2ckL | FOSC↑ to CLKOUT↓ (1) | — | — | 70 | ns | 3.3V ≤ VDD ≤ 5.0V |
| OS12 | TosH2ckH | FOSC↑ to CLKOUT↑ (1) | — | — | 72 | ns | 3.3V ≤ VDD ≤ 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 | 3.3V ≤ VDD ≤ 5.0V |
| OS16 | TosH2ioI | Fosc↑ (Q2 cycle) to Port input invalid (I/O in hold time) | 50 | — | — | ns | 3.3V ≤ VDD ≤ 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 = 1.8V<br>3.3V ≤ VDD ≤ 5.0V |
| OS19* | TioF | Port output fall time(2) | —<br>— | 28<br>15 | 55<br>30 | ns | VDD = 1.8V<br>3.3V ≤ VDD ≤ 5.0V |
| OS20* | Tinp | INT pin input high or low time | 25 | — | — | ns | |
| OS21* | Tioc | Interrupt-on-change new input level time | 25 | — | — | 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 EXTRC mode where CLKOUT output is 4 x TOSC.

**2:** Slew rate limited.

**Note:** Unless otherwise noted, VIN = 5V, FOSC = 300 kHz, CIN = 0.1 µF, TA = 25°C.

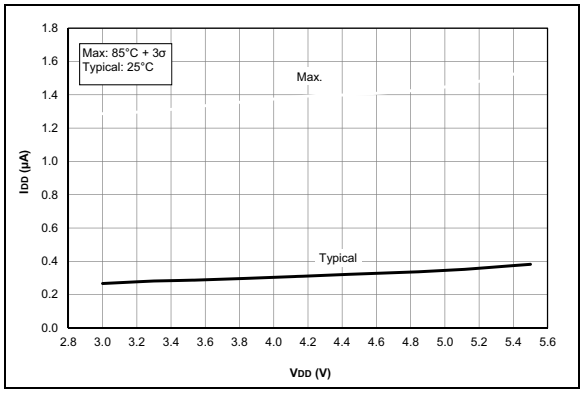



***FIGURE 37-43:*** *IPD, LP Brown-Out Reset (LPBOR = 0), PIC16F1777/8/9 Only.*

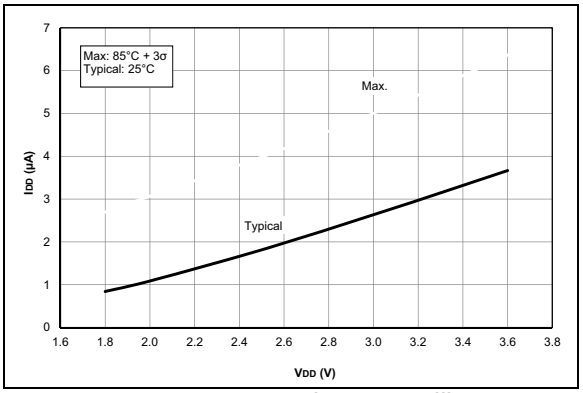



***FIGURE 37-44:*** *IPD, Timer1 Oscillator, FOSC = 32 kHz, PIC16LF1777/8/9 Only.*

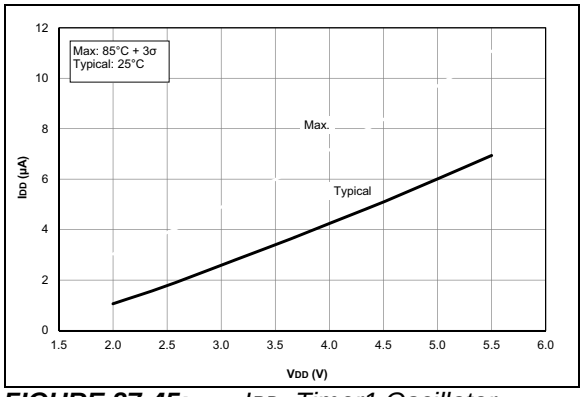



***FIGURE 37-45:*** *IPD, Timer1 Oscillator, FOSC = 32 kHz, PIC16F1777/8/9 Only.*

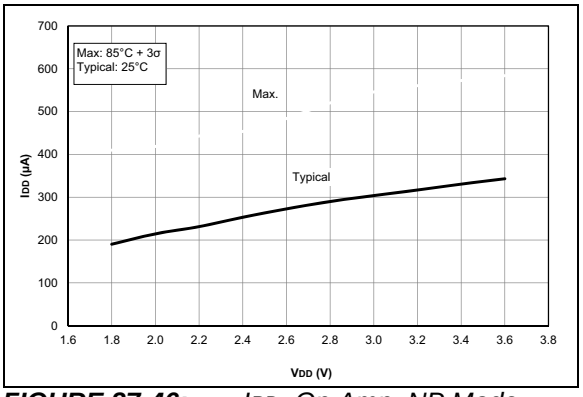



***FIGURE 37-46:*** *IPD, Op Amp, NP Mode (VREFPM = 0), PIC16LF1777/8/9 Only.*

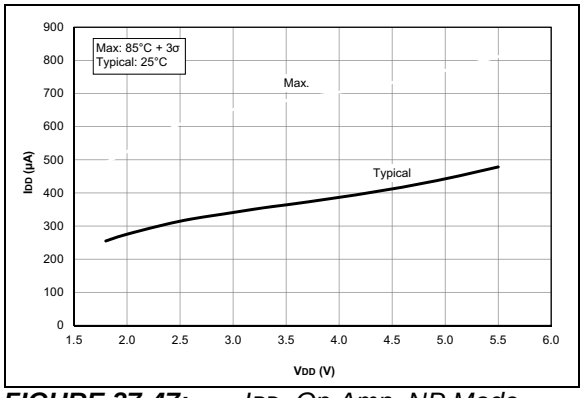



***FIGURE 37-47:*** *IPD, Op Amp, NP Mode (VREFPM = 0), PIC16F1777/8/9 Only.*

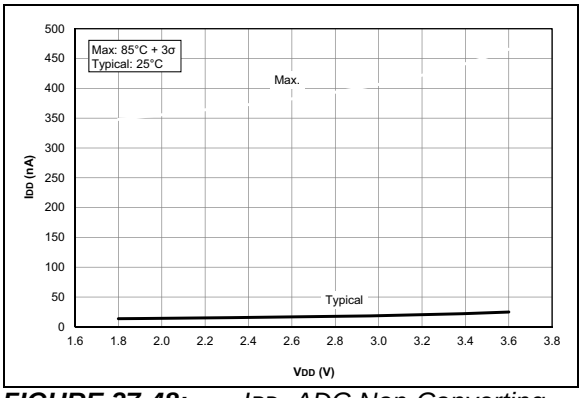



***FIGURE 37-48:*** *IPD, ADC Non-Converting, PIC16LF1777/8/9 Only.*