



Welcome to [E-XFL.COM](https://www.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.

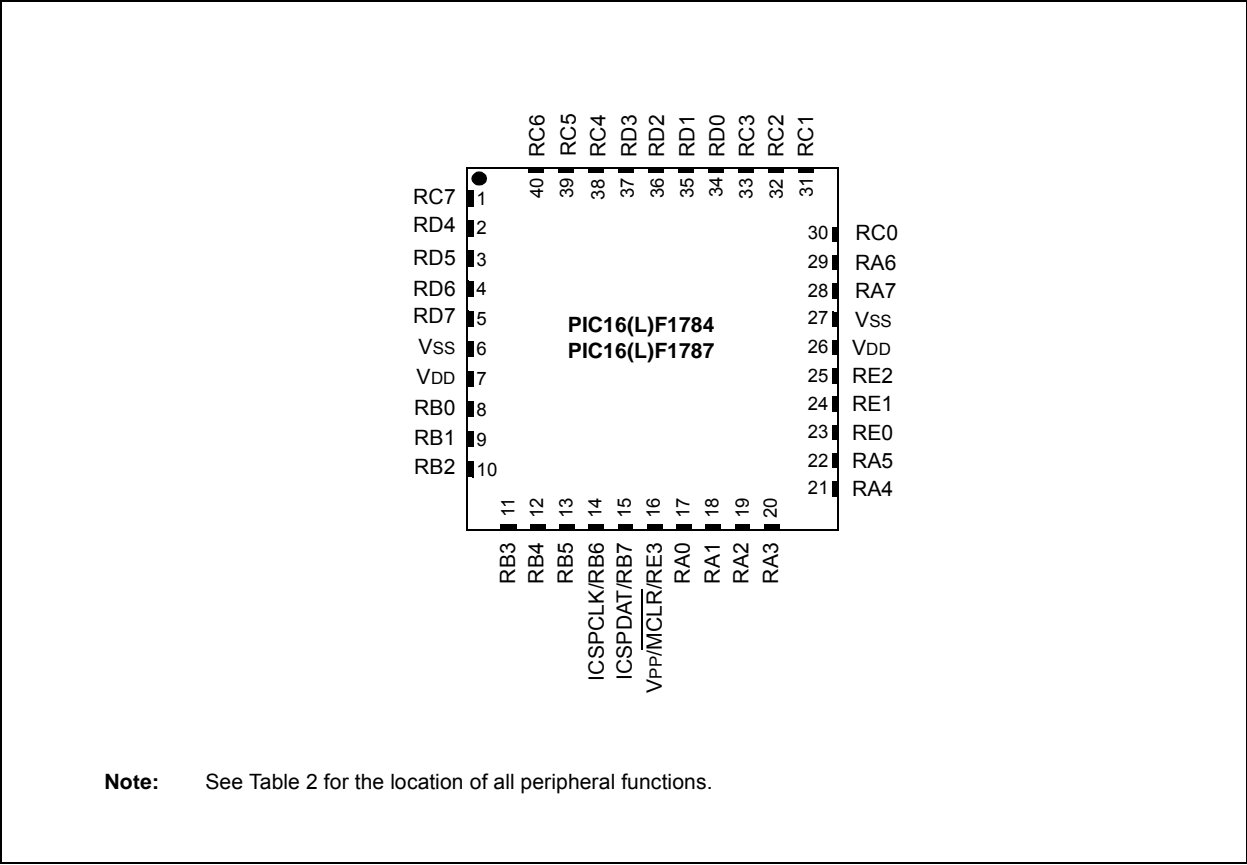
### Applications of "[Embedded - Microcontrollers](#)"

#### Details

|                            |   |
|----------------------------|---|
| Product Status             | Active  |
| Core Processor             | PIC   |
| Core Size                  | 8-Bit   |
| Speed                      | 32MHz   |
| Connectivity               | I <sup>2</sup> C, LINbus, SPI, UART/USART   |
| Peripherals                | Brown-out Detect/Reset, POR, PSMC, PWM, WDT   |
| Number of I/O              | 24  |
| Program Memory Size        | 14KB (8K x 14)  |
| Program Memory Type        | FLASH   |
| EEPROM Size                | 256 x 8   |
| RAM Size                   | 1K x 8  |
| Voltage - Supply (Vcc/Vdd) | 1.8V ~ 3.6V   |
| Data Converters            | A/D 11x12b; D/A 1x8b  |
| Oscillator Type            | Internal  |
| Operating Temperature      | -40°C ~ 85°C (TA)   |
| Mounting Type              | Surface Mount   |
| Package / Case             | 28-SSOP (0.209", 5.30mm Width)  |
| Supplier Device Package    | 28-SSOP   |
| Purchase URL               | <a href="https://www.e-xfl.com/product-detail/microchip-technology/pic16lf1786-i-ss">https://www.e-xfl.com/product-detail/microchip-technology/pic16lf1786-i-ss</a> |

# PIC16(L)F1784/6/7

Pin Diagram – 40-Pin UQFN (5x5)



# PIC16(L)F1784/6/7

## 3.2 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 3.6 “Indirect Addressing”** for more information.

Data memory uses a 12-bit address. The upper 5 bits of the address define the Bank address and the lower 7 bits select the registers/RAM in that bank.

### 3.2.1 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 3-11.

**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 |

# PIC16(L)F1784/6/7

**TABLE 3-12: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)**

| 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 |
|-------------------|-----------|--------------------------------|----------|-------------|-----------|--------------------------------------|----------|-------------|---------|-------------------|---------------------------|
| <b>Bank 16</b>    |           |                                |          |             |           |                                      |          |             |         |                   |                           |
| 80Ch<br>—<br>810h | —         | Unimplemented                  |          |             |           |                                      |          |             |         | —                 | —                         |
| 811h              | PSMC1CON  | PSMC1EN                        | PSMC1LD  | PSMC1DBFE   | PSMC1DBRE | P1MODE<3:0>                          |          |             |         | 0000 0000         | 0000 0000                 |
| 812h              | PSMC1MDL  | P1MDLEN                        | P1MDLPOL | P1MDLBIT    | —         | P1MSRC<3:0>                          |          |             |         | 000- 0000         | 000- 0000                 |
| 813h              | PSMC1SYNC | P1POFST                        | P1PRPOL  | P1DCPOL     | —         | —                                    | —        | P1SYNC<1:0> |         | 000- --00         | 000- --00                 |
| 814h              | PSMC1CLK  | —                              | —        | P1CPRE<1:0> |           | —                                    | —        | P1CSRC<1:0> |         | --00 --00         | --00 --00                 |
| 815h              | PSMC1OEN  | —                              | —        | P1OEF       | P1OEE     | P1OED                                | P1OEC    | P1OEB       | P1OEA   | --00 0000         | --00 0000                 |
| 816h              | PSMC1POL  | —                              | P1INPOL  | P1POLF      | P1POLE    | P1POLD                               | P1POLC   | P1POLB      | P1POLA  | -000 0000         | -000 0000                 |
| 817h              | PSMC1BLNK | —                              | —        | P1FEBM<1:0> |           | —                                    | —        | P1REBM<1:0> |         | --00 --00         | --00 --00                 |
| 818h              | PSMC1REBS | P1REBIN                        | —        | —           | P1REBSC4  | P1REBSC3                             | P1REBSC2 | P1REBSC1    | —       | 0--0 000-         | 0--0 000-                 |
| 819h              | PSMC1FEBS | P1FEBIN                        | —        | —           | P1FEBSC4  | P1FEBSC3                             | P1FEBSC2 | P1FEBSC1    | —       | 0--0 000-         | 0--0 000-                 |
| 81Ah              | PSMC1PHS  | P1PHSIN                        | —        | —           | P1PHSC4   | P1PHSC3                              | P1PHSC2  | P1PHSC1     | P1PHST  | 0--0 0000         | 0--0 0000                 |
| 81Bh              | PSMC1DCS  | P1DCSIN                        | —        | —           | P1DCSC4   | P1DCSC3                              | P1DCSC2  | P1DCSC1     | P1DCST  | 0--0 0000         | 0--0 0000                 |
| 81Ch              | PSMC1PRS  | P1PRSIN                        | —        | —           | P1PRSC4   | P1PRSC3                              | P1PRSC2  | P1PRSC1     | P1PRST  | 0--0 0000         | 0--0 0000                 |
| 81Dh              | PSMC1ASDC | P1ASE                          | P1ASDEN  | P1ARSEN     | —         | —                                    | —        | —           | P1ASDOV | 000- ---0         | 000- ---0                 |
| 81Eh              | PSMC1ASDL | —                              | —        | P1ASDLF     | P1ASDLE   | P1ASDLD                              | P1ASDLC  | P1ASDLB     | P1ASDLA | --00 0000         | --00 0000                 |
| 81Fh              | PSMC1ASDS | P1ASDSIN                       | —        | —           | P1ASDSC4  | P1ASDSC3                             | P1ASDSC2 | P1ASDSC1    | —       | 0--0 000-         | 0--0 000-                 |
| 820h              | PSMC1INT  | P1TOVIE                        | P1TPHIE  | P1TDCIE     | P1TPRIE   | P1TOVIF                              | P1TPHIF  | P1TDCIF     | P1TPRIF | 0000 0000         | 0000 0000                 |
| 821h              | PSMC1PHL  | Phase Low Count                |          |             |           |                                      |          |             |         | 0000 0000         | 0000 0000                 |
| 822h              | PSMC1PHH  | Phase High Count               |          |             |           |                                      |          |             |         | 0000 0000         | 0000 0000                 |
| 823h              | PSMC1DCL  | Duty Cycle Low Count           |          |             |           |                                      |          |             |         | 0000 0000         | 0000 0000                 |
| 824h              | PSMC1DCH  | Duty Cycle High Count          |          |             |           |                                      |          |             |         | 0000 0000         | 0000 0000                 |
| 825h              | PSMC1PRL  | Period Low Count               |          |             |           |                                      |          |             |         | 0000 0000         | 0000 0000                 |
| 826h              | PSMC1PRH  | Period High Count              |          |             |           |                                      |          |             |         | 0000 0000         | 0000 0000                 |
| 827h              | PSMC1TMRL | Time base Low Counter          |          |             |           |                                      |          |             |         | 0000 0001         | 0000 0001                 |
| 828h              | PSMC1TMRH | Time base High Counter         |          |             |           |                                      |          |             |         | 0000 0000         | 0000 0000                 |
| 829h              | PSMC1DBR  | rising Edge Dead-band Counter  |          |             |           |                                      |          |             |         | 0000 0000         | 0000 0000                 |
| 82Ah              | PSMC1DBF  | Falling Edge Dead-band Counter |          |             |           |                                      |          |             |         | 0000 0000         | 0000 0000                 |
| 82Bh              | PSMC1BLKR | rising Edge Blanking Counter   |          |             |           |                                      |          |             |         | 0000 0000         | 0000 0000                 |
| 82Ch              | PSMC1BLKF | Falling Edge Blanking Counter  |          |             |           |                                      |          |             |         | 0000 0000         | 0000 0000                 |
| 82Dh              | PSMC1FFA  | —                              | —        | —           | —         | Fractional Frequency Adjust Register |          |             |         | ---- 0000         | ---- 0000                 |
| 82Eh              | PSMC1STR0 | —                              | —        | P1STRF      | P1STRE    | P1STRD                               | P1STRC   | P1STRB      | P1STRA  | --00 0001         | --00 0001                 |
| 82Fh              | PSMC1STR1 | P1SYNC                         | —        | —           | —         | —                                    | —        | P1LSMEN     | P1HSMEN | 0--- --00         | 0--- --00                 |
| 830h              | —         | Unimplemented                  |          |             |           |                                      |          |             |         | —                 | —                         |

**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: These registers can be addressed from any bank.
  - 2: Unimplemented, read as '1'.
  - 3: PIC16(L)F1784/7 only.
  - 4: PIC16F1784/6/7 only.

# PIC16(L)F1784/6/7

## 4.6 Device ID and Revision ID

The memory location 8006h is where the Device ID and Revision ID are stored. The upper nine bits hold the Device ID. The lower five bits hold the Revision ID. See **Section 12.5 “User ID, Device ID and Configuration Word Access”** for more information on accessing these memory locations.

Development tools, such as device programmers and debuggers, may be used to read the Device ID and Revision ID.

## 4.7 Register Definitions: Device and Revision

### REGISTER 4-3: DEVID: DEVICE ID REGISTER

|          |   |   |       |   |   |
|----------|---|---|-------|---|---|
| R        | R | R | R     | R | R |
| DEV<8:3> |   |   |       |   |   |
| bit 13   |   |   | bit 8 |   |   |

|          |   |   |   |          |   |   |   |
|----------|---|---|---|----------|---|---|---|
| R        | R | R | R | R        | R | R | R |
| DEV<2:0> |   |   |   | REV<4:0> |   |   |   |
| bit 7    |   |   |   | bit 0    |   |   |   |

#### Legend:

R = Readable bit

'1' = Bit is set

'0' = Bit is cleared

bit 13-5      **DEV<8:0>**: Device ID bits

| Device      | DEVICEID<13:0> Values |          |
|-------------|-----------------------|----------|
|             | DEV<8:0>              | REV<4:0> |
| PIC16F1784  | 10 1010 010           | x xxxx   |
| PIC16LF1784 | 10 1010 111           | x xxxx   |
| PIC16F1786  | 10 1010 011           | x xxxx   |
| PIC16LF1786 | 10 1011 000           | x xxxx   |
| PIC16F1787  | 10 1010 100           | x xxxx   |
| PIC16LF1787 | 10 1011 001           | x xxxx   |

bit 4-0      **REV<4:0>**: Revision ID bits  
These bits are used to identify the revision (see Table under DEV<8:0> above).

## 6.2.2.7 Internal Oscillator Clock Switch Timing

When switching between the HFINTOSC, MFINTOSC and the LFINTOSC, the new oscillator may already be shut down to save power (see Figure 6-7). If this is the case, there is a delay after the IRCF<3:0> bits of the OSCCON register are modified before the frequency selection takes place. The OSCSTAT register will reflect the current active status of the HFINTOSC, MFINTOSC and LFINTOSC oscillators. The sequence of a frequency selection is as follows:

1. IRCF<3:0> bits of the OSCCON register are modified.
2. If the new clock is shut down, a clock start-up delay is started.
3. Clock switch circuitry waits for a falling edge of the current clock.
4. The current clock is held low and the clock switch circuitry waits for a rising edge in the new clock.
5. The new clock is now active.
6. The OSCSTAT register is updated as required.
7. Clock switch is complete.

See Figure 6-7 for more details.

If the internal oscillator speed is switched between two clocks of the same source, there is no start-up delay before the new frequency is selected. Clock switching time delays are shown in Table 6-1.

Start-up delay specifications are located in the oscillator tables of **Section 30.0 “Electrical Specifications”**.

# PIC16(L)F1784/6/7

## 6.4 Two-Speed Clock Start-up Mode

Two-Speed Start-up mode provides additional power savings by minimizing the latency between external oscillator start-up and code execution. In applications that make heavy use of the Sleep mode, Two-Speed Start-up will remove the external oscillator start-up time from the time spent awake and can reduce the overall power consumption of the device. This mode allows the application to wake-up from Sleep, perform a few instructions using the INTOSC internal oscillator block as the clock source and go back to Sleep without waiting for the external oscillator to become stable.

Two-Speed Start-up provides benefits when the oscillator module is configured for LP, XT or HS modes. The Oscillator Start-up Timer (OST) is enabled for these modes and must count 1024 oscillations before the oscillator can be used as the system clock source.

If the oscillator module is configured for any mode other than LP, XT or HS mode, then Two-Speed Start-up is disabled. This is because the external clock oscillator does not require any stabilization time after POR or an exit from Sleep.

If the OST count reaches 1024 before the device enters Sleep mode, the OSTS bit of the OSCSTAT register is set and program execution switches to the external oscillator. However, the system may never operate from the external oscillator if the time spent awake is very short.

**Note:** Executing a `SLEEP` instruction will abort the oscillator start-up time and will cause the OSTS bit of the OSCSTAT register to remain clear.

### 6.4.1 TWO-SPEED START-UP MODE CONFIGURATION

Two-Speed Start-up mode is configured by the following settings:

- IESO (of the Configuration Words) = 1; Internal/External Switchover bit (Two-Speed Start-up mode enabled).
- SCS (of the OSCCON register) = 00.
- FOSC<2:0> bits in the Configuration Words configured for LP, XT or HS mode.

Two-Speed Start-up mode is entered after:

- Power-on Reset (POR) and, if enabled, after Power-up Timer (PWRT) has expired, or
- Wake-up from Sleep.

**TABLE 6-1: OSCILLATOR SWITCHING DELAYS**

| Switch From      | Switch To   | Frequency                                       | Oscillator Delay                 |
|------------------|---|---|----------------------------------|
| Sleep/POR        | LFINTOSC <sup>(1)</sup><br>MFINTOSC <sup>(1)</sup><br>HFINTOSC <sup>(1)</sup> | 31 kHz<br>31.25 kHz-500 kHz<br>31.25 kHz-16 MHz | Oscillator Warm-up Delay (TWARM) |
| Sleep/POR        | EC, RC <sup>(1)</sup>   | DC – 32 MHz                                     | 2 cycles                         |
| LFINTOSC         | EC, RC <sup>(1)</sup>   | DC – 32 MHz                                     | 1 cycle of each                  |
| Sleep/POR        | Timer1 Oscillator<br>LP, XT, HS <sup>(1)</sup>                                | 32 kHz-20 MHz                                   | 1024 Clock Cycles (OST)          |
| Any clock source | MFINTOSC <sup>(1)</sup><br>HFINTOSC <sup>(1)</sup>                            | 31.25 kHz-500 kHz<br>31.25 kHz-16 MHz           | 2 $\mu$ s (approx.)              |
| Any clock source | LFINTOSC <sup>(1)</sup>   | 31 kHz  | 1 cycle of each                  |
| Any clock source | Timer1 Oscillator   | 32 kHz  | 1024 Clock Cycles (OST)          |
| PLL inactive     | PLL active  | 16-32 MHz                                       | 2 ms (approx.)                   |

**Note 1:** PLL inactive.

## 7.0 REFERENCE CLOCK MODULE

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

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

The reference clock module is controlled by the CLKRCON register (Register 7-1) and is enabled when setting the CLKREN bit. To output the divided clock signal to the CLKR port pin, the CLKROE bit must be set. The CLKRDIV<2:0> bits enable the selection of eight different clock divider options. The CLKRDC<1:0> bits can be used to modify the duty cycle of the output clock<sup>(1)</sup>. The CLKRSLR bit controls slew rate limiting.

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

### 7.1 Slew Rate

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

### 7.2 Effects of a Reset

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

## 7.3 Conflicts with the CLKR Pin

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

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

### 7.3.1 OSCILLATOR MODES

If LP, XT or HS oscillator modes are selected, the OSC2/CLKR pin must be used as an oscillator input pin and the CLKR output cannot be enabled. See **Section 6.2 "Clock Source Types"** for more information on different oscillator modes.

### 7.3.2 CLKOUT FUNCTION

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

## 7.4 Operation During Sleep

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



# PIC16(L)F1784/6/7

**TABLE 8-1: 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     | IOIE   | TMR0IF   | INTF     | IOCF     | 93               |
| OPTION_REG | WPUEN   | INTEDG   | TMR0CS   | TMR0SE   | PSA    | PS<2:0>  |          |          | 198              |
| PIE1       | TMR1GIE | ADIE     | RCIE     | TXIE     | SSP1IE | CCP1IE   | TMR2IE   | TMR1IE   | 94               |
| PIE2       | OSFIE   | C2IE     | C1IE     | EEIE     | BCL1IE | C4IE     | C3IE     | CCP2IE   | 95               |
| PIE3       | —       | —        | —        | CCP3IE   | —      | —        | —        | —        | 96               |
| PIE4       | —       | PSMC3TIE | PSMC2TIE | PSMC1TIE | —      | PSMC3SIE | PSMC2SIE | PSMC1SIE | 97               |
| PIR1       | TMR1GIF | ADIF     | RCIF     | TXIF     | SSP1IF | CCP1IF   | TMR2IF   | TMR1IF   | 98               |
| PIR2       | OSFIF   | C2IF     | C1IF     | EEIF     | BCL1IF | C4IF     | C3IF     | CCP2IF   | 99               |
| PIR3       | —       | —        | —        | CCP3IF   | —      | —        | —        | —        | 100              |
| PIR4       | —       | PSMC3TIF | PSMC2TIF | PSMC1TIF | —      | PSMC3SIF | PSMC2SIF | PSMC1SIF | 101              |

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

## 13.7 PORTC Registers

### 13.7.1 DATA REGISTER

PORTC is an 8-bit wide bidirectional port. The corresponding data direction register is TRISC (Register 13-20). Setting a TRISC bit (= 1) will make the corresponding PORTC pin an input (i.e., put the corresponding output driver in a High-Impedance mode). Clearing a TRISC bit (= 0) will make the corresponding PORTC pin an output (i.e., enable the output driver and put the contents of the output latch on the selected pin). Example 13-1 shows how to initialize an I/O port.

Reading the PORTC register (Register 13-19) 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 (LATC).

### 13.7.2 DIRECTION CONTROL

The TRISC register (Register 13-20) controls the PORTC pin output drivers, even when they are being used as analog inputs. The user should ensure the bits in the TRISC register are maintained set when using them as analog inputs. I/O pins configured as analog inputs always read '0'.

### 13.7.3 OPEN DRAIN CONTROL

The ODCONC register (Register 13-23) controls the open-drain feature of the port. Open drain operation is independently selected for each pin. When an ODCONC bit is set, the corresponding port output becomes an open drain driver capable of sinking current only. When an ODCONC bit is cleared, the corresponding port output pin is the standard push-pull drive capable of sourcing and sinking current.

### 13.7.4 SLEW RATE CONTROL

The SLRCONC register (Register 13-24) controls the slew rate option for each port pin. Slew rate control is independently selectable for each port pin. When an SLRCONC bit is set, the corresponding port pin drive is slew rate limited. When an SLRCONC bit is cleared, The corresponding port pin drive slews at the maximum rate possible.

### 13.7.5 INPUT THRESHOLD CONTROL

The INLVLC register (Register 13-25) controls the input voltage threshold for each of the available PORTC input pins. A selection between the Schmitt Trigger CMOS or the TTL Compatible thresholds is available. The input threshold is important in determining the value of a read of the PORTC register and also the

level at which an interrupt-on-change occurs, if that feature is enabled. See **Section TABLE 30-1: "Supply Voltage"** for more information on threshold levels.

**Note:** Changing the input threshold selection should be performed while all peripheral modules are disabled. Changing the threshold level during the time a module is active may inadvertently generate a transition associated with an input pin, regardless of the actual voltage level on that pin.

### 13.7.6 PORTC FUNCTIONS AND OUTPUT PRIORITIES

Each PORTC pin is multiplexed with other functions. The pins, their combined functions and their output priorities are shown in Table 13-7.

When multiple outputs are enabled, the actual pin control goes to the peripheral with the highest priority. Analog input and some digital input functions are not included in the list below. These input functions can remain active when the pin is configured as an output. Certain digital input functions override other port functions and are included in the priority list.

**TABLE 13-7: PORTC OUTPUT PRIORITY**

| Pin Name | Function Priority <sup>(1)</sup> |
|----------|----------------------------------|
| RC0      | T1OSO<br>PSMC1A<br>RC0           |
| RC1      | PSMC1B<br>CCP2<br>RC1            |
| RC2      | PSMC1C<br>CCP1<br>RC2            |
| RC3      | PSMC1D<br>SCL<br>SCK<br>RC3      |
| RC4      | PSMC1E<br>SDA<br>RC4             |
| RC5      | PSMC1F<br>SDO<br>RC5             |
| RC6      | PSMC2A<br>TX/CK<br>CCP3<br>RC6   |
| RC7      | PSMC2B<br>DT<br>RC7              |

**Note 1:** Priority listed from highest to lowest.

# PIC16(L)F1784/6/7

## REGISTER 13-38: WPUE: WEAK PULL-UP PORTE REGISTER

|       |     |     |     |         |                      |                      |                      |
|-------|-----|-----|-----|---------|----------------------|----------------------|----------------------|
| U-0   | U-0 | U-0 | U-0 | R/W-1/1 | R/W-1/1              | R/W-1/1              | R/W-1/1              |
| —     | —   | —   | —   | WPUE3   | WPUE2 <sup>(3)</sup> | WPUE1 <sup>(3)</sup> | WPUE0 <sup>(3)</sup> |
| 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                      **Unimplemented:** Read as '0'  
 bit 3-0                      **WPUE<3:0>:** Weak Pull-up Register bit<sup>(3)</sup>  
                                     1 = Pull-up enabled  
                                     0 = Pull-up disabled

- Note 1:** Global **WPUEN** bit of the **OPTION\_REG** register must be cleared for individual pull-ups to be enabled.
- Note 2:** The weak pull-up device is automatically disabled if the pin is in configured as an output.
- Note 3:** **WPUE<2:0>** are available on PIC16(L)F1784/7 only.

## REGISTER 13-39: ODCONE: PORTE OPEN DRAIN CONTROL REGISTER<sup>(1)</sup>

|       |     |     |     |     |         |         |         |
|-------|-----|-----|-----|-----|---------|---------|---------|
| U-0   | U-0 | U-0 | U-0 | U-0 | R/W-0/0 | R/W-0/0 | R/W-0/0 |
| —     | —   | —   | —   | —   | ODE2    | ODE1    | ODE0    |
| 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 7-3                      **Unimplemented:** Read as '0'  
 bit 2-0                      **ODE<2:0>:** PORTE Open Drain Enable bits  
                                     For RE<2:0> pins, respectively  
                                     1 = Port pin operates as open-drain drive (sink current only)  
                                     0 = Port pin operates as standard push-pull drive (source and sink current)

- Note 1:** **ODCONE<2:0>** are available on PIC16(L)F1784/7 only.

## REGISTER 14-3: IOCF: INTERRUPT-ON-CHANGE 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 |
| IOCF7      | IOCF6      | IOCF5      | IOCF4      | IOCF3      | IOCF2      | IOCF1      | IOCF0      |
| 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 **IOCF<7:0>**: Interrupt-on-Change Flag bits<sup>(1)</sup>

1 = An enabled change was detected on the associated pin.

Set when IOCF<sub>Px</sub> = 1 and a rising edge was detected RB<sub>x</sub>, or when IOCF<sub>Nx</sub> = 1 and a falling edge was detected on RB<sub>x</sub>.

0 = No change was detected, or the user cleared the detected change.

**Note 1:** For IOCF register, bit 3 (IOCF3) is the only implemented bit in the register.

**TABLE 14-1: SUMMARY OF REGISTERS ASSOCIATED WITH INTERRUPT-ON-CHANGE**

| Name   | Bit 7  | Bit 6  | Bit 5  | Bit 4  | Bit 3  | Bit 2  | Bit 1  | Bit 0  | Register on Page |
|--------|--------|--------|--------|--------|--------|--------|--------|--------|------------------|
| ANSELB | —      | ANSB6  | ANSB5  | ANSB4  | ANSB3  | ANSB2  | ANSB1  | ANSB0  | 138              |
| INTCON | GIE    | PEIE   | TMR0IE | INTE   | IOCIE  | TMR0IF | INTF   | IOCIF  | 93               |
| IOCAF  | IOCAF7 | IOCAF6 | IOCAF5 | IOCAF4 | IOCAF3 | IOCAF2 | IOCAF1 | IOCAF0 | 159              |
| IOCAN  | IOCAN7 | IOCAN6 | IOCAN5 | IOCAN4 | IOCAN3 | IOCAN2 | IOCAN1 | IOCAN0 | 158              |
| IOCAP  | IOCAP7 | IOCAP6 | IOCAP5 | IOCAP4 | IOCAP3 | IOCAP2 | IOCAP1 | IOCAP0 | 158              |
| IOCBF  | IOCBF7 | IOCBF6 | IOCBF5 | IOCBF4 | IOCBF3 | IOCBF2 | IOCBF1 | IOCBF0 | 159              |
| IOCBN  | IOCBN7 | IOCBN6 | IOCBN5 | IOCBN4 | IOCBN3 | IOCBN2 | IOCBN1 | IOCBN0 | 158              |
| IOCBP  | IOCBP7 | IOCBP6 | IOCBP5 | IOCBP4 | IOCBP3 | IOCBP2 | IOCBP1 | IOCBP0 | 158              |
| IOCCF  | IOCCF7 | IOCCF6 | IOCCF5 | IOCCF4 | IOCCF3 | IOCCF2 | IOCCF1 | IOCCF0 | 159              |
| IOCCN  | IOCCN7 | IOCCN6 | IOCCN5 | IOCCN4 | IOCCN3 | IOCCN2 | IOCCN1 | IOCCN0 | 158              |
| IOCCP  | IOCCP7 | IOCCP6 | IOCCP5 | IOCCP4 | IOCCP3 | IOCCP2 | IOCCP1 | IOCCP0 | 158              |
| IOCEF  | —      | —      | —      | —      | IOCEF3 | —      | —      | —      | 159              |
| IOCEN  | —      | —      | —      | —      | IOCEN3 | —      | —      | —      | 158              |
| IOCEP  | —      | —      | —      | —      | IOCEP3 | —      | —      | —      | 158              |
| TRISB  | TRISB7 | TRISB6 | TRISB5 | TRISB4 | TRISB3 | TRISB2 | TRISB1 | TRISB0 | 137              |

**Legend:** — = unimplemented location, read as '0'. Shaded cells are not used by interrupt-on-change.

## 24.7 Auto-Shutdown

Auto-shutdown is a method to immediately override the PSMC output levels with specific overrides that allow for safe shutdown of the application.

Auto-shutdown includes a mechanism to allow the application to restart under different conditions.

Auto-shutdown is enabled with the PxASDEN bit of the PSMC Auto-shutdown Control (PSMCxASDC) register (Register 24-15). All auto-shutdown features are enabled when PxASDEN is set and disabled when cleared.

### 24.7.1 SHUTDOWN

There are two ways to generate a shutdown event:

- Manual
- External Input

#### 24.7.1.1 Manual Override

The auto-shutdown control register can be used to manually override the pin functions. Setting the PxASE bit of the PSMC Auto-shutdown Control (PSMCxASDC) register (Register 24-15) generates a software shut-down event.

The auto-shutdown override will persist as long as PxASE remains set.

#### 24.7.1.2 External Input Source

Any of the given sources that are available for event generation are also available for system shut-down. This is so that external circuitry can monitor and force a shutdown without any software overhead. Auto-shutdown sources are selected with the PSMC Auto-shutdown Source (PSMCxASDS) register (Register 24-17).

When any of the selected external auto-shutdown sources go high, the PxASE bit is set and an auto-shutdown interrupt is generated.

**Note:** The external shutdown sources are level sensitive, not edge sensitive. The shutdown condition will persist as long as the circuit is driving the appropriate logic level.

### 24.7.2 PIN OVERRIDE LEVELS

The logic levels driven to the output pins during an auto-shutdown event are determined by the PSMC Auto-shutdown Output Level (PSMCxASDL) register (Register 24-16).

#### 24.7.2.1 PIN Override Enable

Setting the PxASDOV bit of the PSMC Auto-shutdown Control (PSMCxASDC) register (Register 24-15) will also force the override levels onto the pins, exactly like what happens when the auto-shutdown is used. However, whereas setting PxASE causes an auto-shutdown interrupt, setting PxASDOV does not generate an interrupt.

### 24.7.3 RESTART FROM AUTO-SHUTDOWN

After an auto-shutdown event has occurred, there are two ways for the module to resume operation:

- Manual restart
- Automatic restart

The restart method is selected with the PxARSEN bit of the PSMC Auto-shutdown Control (PSMCxASDC) register (Register 24-15).

#### 24.7.3.1 Manual Restart

When PxARSEN is cleared, and once the PxASDE bit is set, it will remain set until cleared by software.

The PSMC will restart on the period event after PxASDE bit is cleared in software.

#### 24.7.3.2 Auto-Restart

When PxARSEN is set, the PxASDE bit will clear automatically when the source causing the Reset and no longer asserts the shut-down condition.

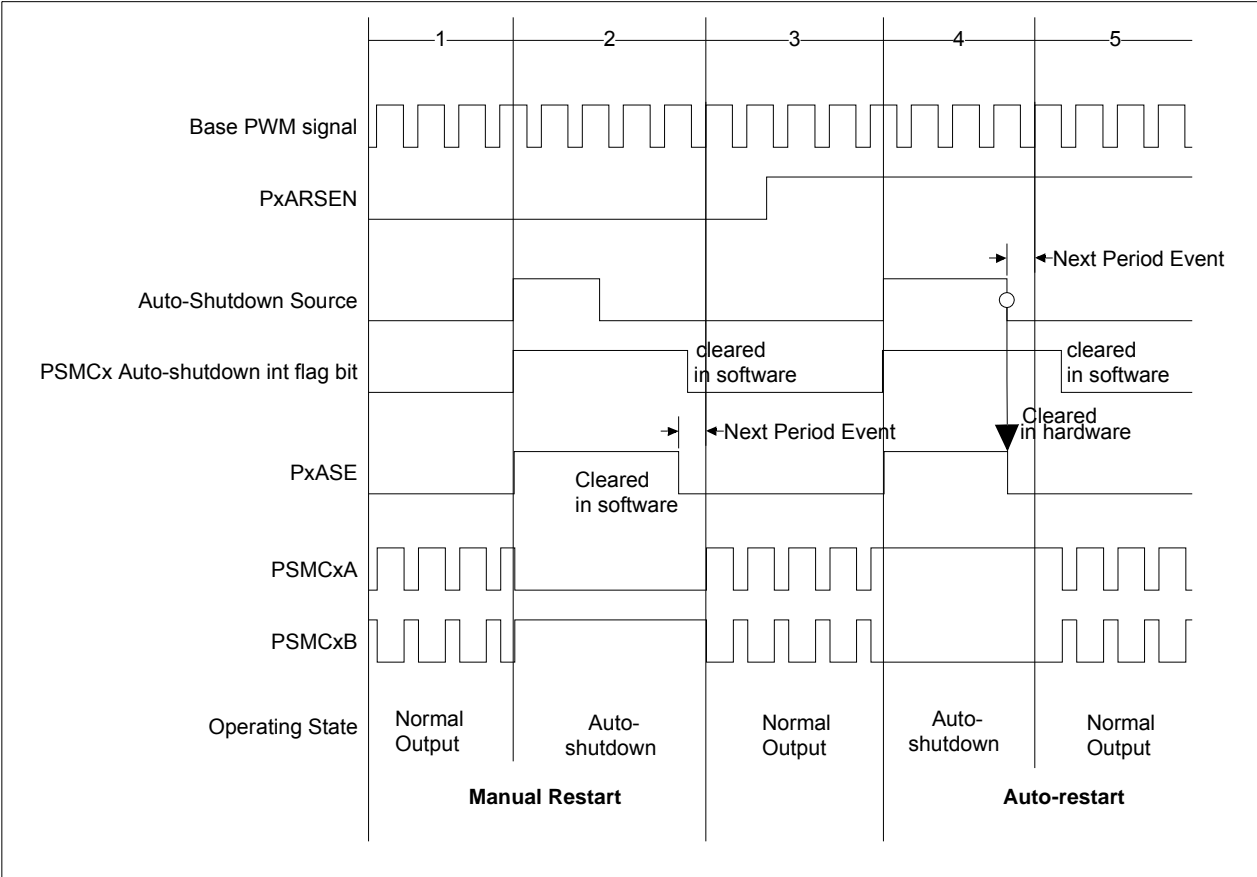
The PSMC will restart on the next period event after the auto-shutdown condition is removed.

Examples of manual and automatic restart are shown in Figure 24-20.

**Note:** Whether manual or auto-restart is selected, the PxASDE bit cannot be cleared in software when the auto-shutdown condition is still present.

# PIC16(L)F1784/6/7

FIGURE 24-20: AUTO-SHUTDOWN AND RESTART WAVEFORM



## REGISTER 24-24: PSMC<sub>x</sub>PRL: PSMC PERIOD COUNT LOW BYTE 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 |
| PSMC <sub>x</sub> PRL<7:0> |         |         |         |         |         |         |         |
| bit 7                      |         |         |         | bit 0   |         |         |         |

### Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

u = Bit is unchanged

x = Bit is unknown

-n/n = Value at POR and BOR/Value at all other Resets

'1' = Bit is set

'0' = Bit is cleared

bit 7-0      **PSMC<sub>x</sub>PRL<7:0>**: 16-bit Period Time Least Significant bits  
               = PSMC<sub>x</sub>PR<7:0>

## REGISTER 24-25: PSMC<sub>x</sub>PRH: PSMC PERIOD COUNT HIGH BYTE 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 |
| PSMC <sub>x</sub> PRH<7:0> |         |         |         |         |         |         |         |
| bit 7                      |         |         |         | bit 0   |         |         |         |

### Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

u = Bit is unchanged

x = Bit is unknown

-n/n = Value at POR and BOR/Value at all other Resets

'1' = Bit is set

'0' = Bit is cleared

bit 7-0      **PSMC<sub>x</sub>PRH<7:0>**: 16-bit Period Time Most Significant bits  
               = PSMC<sub>x</sub>PR<15:8>

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

## 26.3.1 CLOCK STRETCHING

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

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

## 26.3.2 ARBITRATION

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

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

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

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

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

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

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



## 26.6 I<sup>2</sup>C Master Mode

Master mode is enabled by setting and clearing the appropriate SSPM bits in the SSPCON1 register and by setting the SSPEN bit. In Master mode, the SDA and SCK pins must be configured as inputs. The MSSP peripheral hardware will override the output driver TRIS controls when necessary to drive the pins low.

Master mode of operation is supported by interrupt generation on the detection of the Start and Stop conditions. The Stop (P) and Start (S) bits are cleared from a Reset or when the MSSP module is disabled. Control of the I<sup>2</sup>C bus may be taken when the P bit is set, or the bus is Idle.

In Firmware Controlled Master mode, user code conducts all I<sup>2</sup>C bus operations based on Start and Stop bit condition detection. Start and Stop condition detection is the only active circuitry in this mode. All other communication is done by the user software directly manipulating the SDA and SCL lines.

The following events will cause the SSP Interrupt Flag bit, SSP1IF, to be set (SSP interrupt, if enabled):

- Start condition detected
- Stop condition detected
- Data transfer byte transmitted/received
- Acknowledge transmitted/received
- Repeated Start generated

**Note 1:** The MSSP module, when configured in I<sup>2</sup>C Master mode, does not allow queueing of events. For instance, the user is not allowed to initiate a Start condition and immediately write the SSPBUF register to initiate transmission before the Start condition is complete. In this case, the SSPBUF will not be written to and the WCOL bit will be set, indicating that a write to the SSPBUF did not occur

**2:** When in Master mode, Start/Stop detection is masked and an interrupt is generated when the SEN/PEN bit is cleared and the generation is complete.

### 26.6.1 I<sup>2</sup>C MASTER MODE OPERATION

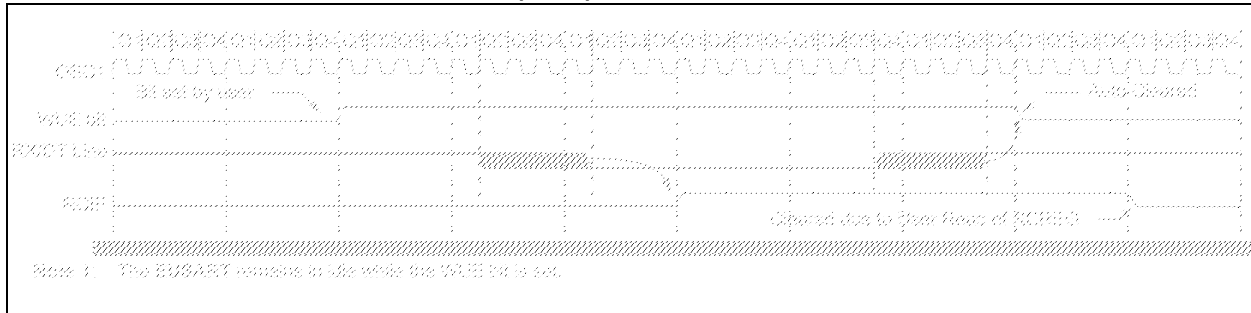
The master device generates all of the serial clock pulses and the Start and Stop conditions. A transfer is ended with a Stop condition or with a Repeated Start condition. Since the Repeated Start condition is also the beginning of the next serial transfer, the I<sup>2</sup>C bus will not be released.

In Master Transmitter mode, serial data is output through SDA, while SCL outputs the serial clock. The first byte transmitted contains the slave address of the receiving device (7 bits) and the Read/Write (R/W) bit. In this case, the R/W bit will be logic '0'. Serial data is transmitted 8 bits at a time. After each byte is transmitted, an Acknowledge bit is received. Start and Stop conditions are output to indicate the beginning and the end of a serial transfer.

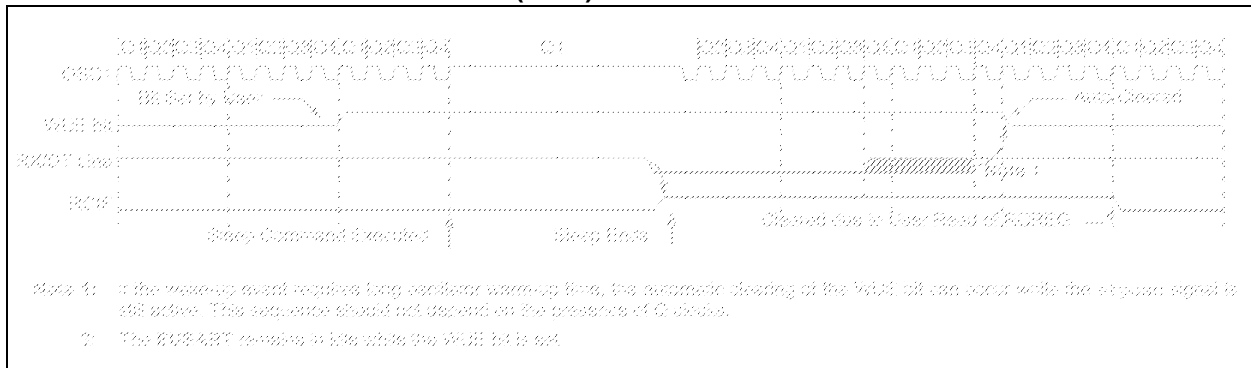
In Master Receive mode, the first byte transmitted contains the slave address of the transmitting device (7 bits) and the R/W bit. In this case, the R/W bit will be logic '1'. Thus, the first byte transmitted is a 7-bit slave address followed by a '1' to indicate the receive bit. Serial data is received via SDA, while SCL outputs the serial clock. Serial data is received 8 bits at a time. After each byte is received, an Acknowledge bit is transmitted. Start and Stop conditions indicate the beginning and end of transmission.

A Baud Rate Generator is used to set the clock frequency output on SCL. See **Section 26.7 “Baud Rate Generator”** for more detail.

**FIGURE 27-7: AUTO-WAKE-UP BIT (WUE) TIMING DURING NORMAL OPERATION**



**FIGURE 27-8: AUTO-WAKE-UP BIT (WUE) TIMINGS DURING SLEEP**



| <b>DECFSZ</b>    | <b>Decrement f, Skip if 0</b>  |
|------------------|--|
| Syntax:          | [ <i>label</i> ] DECFSZ f,d  |
| Operands:        | $0 \leq f \leq 127$<br>$d \in [0,1]$   |
| Operation:       | $(f) - 1 \rightarrow (\text{destination});$<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. |

| <b>INCFSZ</b>    | <b>Increment f, Skip if 0</b>   |
|------------------|---|
| Syntax:          | [ <i>label</i> ] INCFSZ f,d   |
| Operands:        | $0 \leq f \leq 127$<br>$d \in [0,1]$  |
| Operation:       | $(f) + 1 \rightarrow (\text{destination});$<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. |

| <b>GOTO</b>      | <b>Unconditional Branch</b>   |
|------------------|---|
| Syntax:          | [ <i>label</i> ] GOTO k   |
| Operands:        | $0 \leq k \leq 2047$  |
| Operation:       | $k \rightarrow \text{PC}<10:0>$<br>$\text{PCLATH}<6:3> \rightarrow \text{PC}<14:11>$  |
| 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. |

| <b>IORLW</b>     | <b>Inclusive OR literal with W</b>   |
|------------------|--|
| Syntax:          | [ <i>label</i> ] IORLW k   |
| Operands:        | $0 \leq k \leq 255$  |
| Operation:       | $(W) .OR. k \rightarrow (W)$   |
| 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. |

| <b>INCF</b>      | <b>Increment f</b>   |
|------------------|--|
| Syntax:          | [ <i>label</i> ] INCF f,d  |
| Operands:        | $0 \leq f \leq 127$<br>$d \in [0,1]$   |
| Operation:       | $(f) + 1 \rightarrow (\text{destination})$   |
| 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'. |

| <b>IORWF</b>     | <b>Inclusive OR W with f</b>  |
|------------------|---|
| Syntax:          | [ <i>label</i> ] IORWF f,d  |
| Operands:        | $0 \leq f \leq 127$<br>$d \in [0,1]$  |
| Operation:       | $(W) .OR. (f) \rightarrow (\text{destination})$   |
| Status Affected: | $\bar{Z}$   |
| 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'. |

# PIC16(L)F1784/6/7

**TABLE 30-13: ADC CONVERTER (ADC) 12-BIT DIFFERENTIAL CHARACTERISTICS:**

| Operating Conditions<br>V <sub>DD</sub> = 3V, Temp. = 25°C, Single-ended 2 $\mu$ s TAD, V <sub>REF+</sub> = 3V, V <sub>REF-</sub> = V <sub>SS</sub> |      |  |      |      |                 |            |   |
|---|------|--|------|------|-----------------|------------|---|
| Param No.   | Sym. | Characteristic                                 | Min. | Typ† | Max.            | Units      | Conditions  |
| AD01  | NR   | Resolution                                     | —    | —    | 10              | bit        |   |
| AD02  | EIL  | Integral Error                                 | —    | ±1   | ±1.6            | LSb        |   |
| AD03  | EDL  | Differential Error                             | —    | ±1   | ±1.4            | LSb        | No missing codes  |
| AD04  | EOFF | Offset Error                                   | —    | ±1   | ±3.5            | LSb        |   |
| AD05  | EGN  | Gain Error                                     | —    | ±1   | ±2              | LSb        |   |
| AD06  | VREF | Reference Voltage <sup>(3)</sup>               | 1.8  | —    | V <sub>DD</sub> | V          | VREF = (VREF+ minus VREF-)  |
| AD07  | VAIN | Full-Scale Range                               | —    | —    | VREF            | V          |   |
| AD08  | ZAIN | Recommended Impedance of Analog Voltage Source | —    | —    | 10              | k $\Omega$ | Can go higher if external 0.01 $\mu$ F capacitor is present on input pin. |
| AD09  | NR   | Resolution                                     | —    | —    | 12              | bit        |   |
| AD10  | EIL  | Integral Error                                 | —    | ±2   | —               | LSb        |   |
| AD11  | EDL  | Differential Error                             | —    | ±2   | —               | LSb        |   |
| AD12  | EOFF | Offset Error                                   | —    | ±1   | —               | LSb        |   |
| AD13  | EGN  | Gain Error                                     | —    | ±1   | —               | LSb        |   |
| AD14  | VREF | Reference Voltage <sup>(3)</sup>               | 1.8  | —    | V <sub>DD</sub> | V          | VREF = (VREF+ minus VREF-)  |
| AD15  | VAIN | Full-Scale Range                               | —    | —    | VREF            | V          |   |
| AD16  | ZAIN | Recommended Impedance of Analog Voltage Source | —    | —    | 10              | k $\Omega$ | Can go higher if external 0.01 $\mu$ F capacitor is present on input pin. |

\* 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:** Total Absolute Error includes integral, differential, offset and gain errors.

**2:** The ADC conversion result never decreases with an increase in the input voltage and has no missing codes.

**3:** ADC VREF is from external VREF, VDD pin or FVR, whichever is selected as reference input.

**4:** When ADC is off, it will not consume any current other than leakage current. The power-down current specification includes any such leakage from the ADC module.

**TABLE 30-14: ADC CONVERSION REQUIREMENTS**

| Standard Operating Conditions (unless otherwise stated) |      |   |      |                            |      |         |  |
|---|------|---|------|----------------------------|------|---------|--|
| Param No.   | Sym. | Characteristic  | Min. | Typ†                       | Max. | Units   | Conditions                             |
| AD130*  | TAD  | ADC Clock Period  | 1.0  | —                          | 9.0  | $\mu$ s | Tosc-based                             |
|   |      | ADC Internal RC Oscillator Period                               | 1.0  | 2.5                        | 6.0  | $\mu$ s | ADCS<1:0> = 11 (ADRC mode)             |
| AD131   | TCNV | Conversion Time (not including Acquisition Time) <sup>(1)</sup> | —    | 15 (12-bit)<br>13 (10-bit) | —    | TAD     | Set GO/DONE bit to conversion complete |
| AD132*  | TACQ | Acquisition Time  | —    | 5.0                        | —    | $\mu$ s |  |

\* These parameters are characterized but not tested.

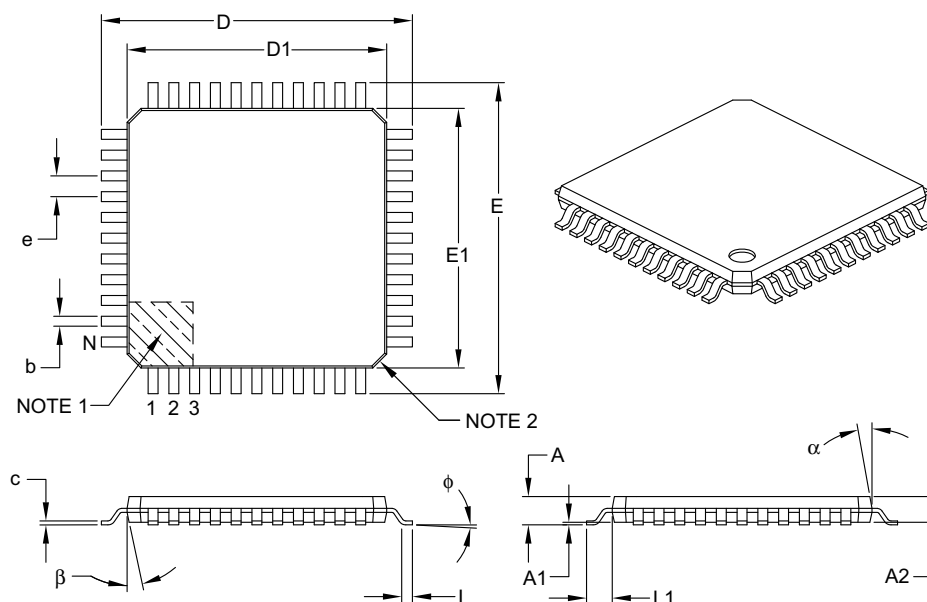
† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** The ADRES register may be read on the following T<sub>CY</sub> cycle.

# PIC16(L)F1784/6/7

## 44-Lead Plastic Thin Quad Flatpack (PT) – 10x10x1 mm Body, 2.00 mm [TQFP]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



| Units                    |          | MILLIMETERS |      |      |
|--------------------------|----------|-------------|------|------|
| Dimension Limits         |          | MIN         | NOM  | MAX  |
| Number of Leads          | N        | 44          |      |      |
| Lead Pitch               | e        | 0.80 BSC    |      |      |
| Overall Height           | A        | –           | –    | 1.20 |
| Molded Package Thickness | A2       | 0.95        | 1.00 | 1.05 |
| Standoff                 | A1       | 0.05        | –    | 0.15 |
| Foot Length              | L        | 0.45        | 0.60 | 0.75 |
| Footprint                | L1       | 1.00 REF    |      |      |
| Foot Angle               | $\phi$   | 0°          | 3.5° | 7°   |
| Overall Width            | E        | 12.00 BSC   |      |      |
| Overall Length           | D        | 12.00 BSC   |      |      |
| Molded Package Width     | E1       | 10.00 BSC   |      |      |
| Molded Package Length    | D1       | 10.00 BSC   |      |      |
| Lead Thickness           | c        | 0.09        | –    | 0.20 |
| Lead Width               | b        | 0.30        | 0.37 | 0.45 |
| Mold Draft Angle Top     | $\alpha$ | 11°         | 12°  | 13°  |
| Mold Draft Angle Bottom  | $\beta$  | 11°         | 12°  | 13°  |

### Notes:

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Chamfers at corners are optional; size may vary.
- Dimensions D1 and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.25 mm per side.
- Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-076B