This document includes the programming specifications for the following devices:

- PIC16F1713 • PIC16LF1713
- PIC16F1716 • PIC16LF1716
- PIC16F1717 • PIC16LF1717
- PIC16F1718 • PIC16LF1718
- PIC16F1719 • PIC16LF1719

1.0 OVERVIEW

The device can be programmed using either the high-voltage In-Circuit Serial Programming™ (ICSP™) method or the low-voltage ICSP method.

1.1 Hardware Requirements

1.1.1 HIGH-VOLTAGE ICSP PROGRAMMING

In High-Voltage ICSP mode, the device requires two programmable power supplies: one for VDD and one for the MCLR/VPP pin.

1.1.2 LOW-VOLTAGE ICSP PROGRAMMING

In Low-Voltage ICSP mode, the PIC16(L)F171X devices can be programmed using a single VDD source in the operating range. The MCLR/VPP pin does not have to be brought to a different voltage, but can instead be left at the normal operating voltage.

1.1.2.1 Single-Supply ICSP Programming

The LVP bit in Configuration Word 2 enables single-supply (low-voltage) ICSP programming. The LVP bit defaults to a ‘1’ (enabled) from the factory. The LVP bit may only be programmed to ‘0’ by entering the High-Voltage ICSP mode, where the MCLR/VPP pin is raised to VIHH. Once the LVP bit is programmed to a ‘0’, only the High-Voltage ICSP mode is available and only the High-Voltage ICSP mode can be used to program the device.

Note 1: The High-Voltage ICSP mode is always available, regardless of the state of the LVP bit, by applying VIHH to the MCLR/VPP pin.

2: While in Low-Voltage ICSP mode, MCLR is always enabled, regardless of the MCLRE bit, and the port pin can no longer be used as a general purpose input.

1.2 Pin Utilization

Five pins are needed for ICSP programming. The pins are listed in Table 1-1.

### TABLE 1-1: PIN DESCRIPTIONS DURING PROGRAMMING

<table>
<thead>
<tr>
<th>Pin Name</th>
<th>During Programming</th>
<th>Pin Type</th>
<th>Pin Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICSPCLK</td>
<td>ICSPCLK</td>
<td>I</td>
<td>Clock Input – Schmitt Trigger Input</td>
</tr>
<tr>
<td>ICSPDAT</td>
<td>ICSPDAT</td>
<td>I/O</td>
<td>Data Input/Output – Schmitt Trigger Input</td>
</tr>
<tr>
<td>MCLR/VPP</td>
<td>Program/Verify mode</td>
<td>P⁽¹⁾</td>
<td>Program Mode Select/Programming Power Supply</td>
</tr>
<tr>
<td>VDD</td>
<td>VDD</td>
<td>P</td>
<td>Power Supply</td>
</tr>
<tr>
<td>VSS</td>
<td>VSS</td>
<td>P</td>
<td>Ground</td>
</tr>
</tbody>
</table>

Legend: I = Input, O = Output, P = Power

Note 1: The programming high voltage is internally generated. To activate the Program/Verify mode, high voltage needs to be applied to MCLR input. Since the MCLR is used for a level source, MCLR does not draw any significant current.
2.0 DEVICE PINOUTS

The pin diagrams for the PIC16L(F)1713/6/8 family are shown in Figure 2-1 and Figure 2-2.

The pin diagrams for the PIC16L(F)1717/9 family are shown in Figure 2-3, Figure 2-4 and Figure 2-5.

The pins that are required for programming are listed in Table 1-1 and shown in bold lettering in the pin diagrams.

FIGURE 2-1: 28-PIN DIAGRAM FOR PIC16L(F)1713/6/8

FIGURE 2-2: 28-PIN PACKAGE DIAGRAM FOR PIC16L(F)1713/6/8
FIGURE 2-3: 40-PIN DIP DIAGRAM FOR PIC16L(F)1717/9

FIGURE 2-4: 40-PIN UQFN (5X5) PACKAGE DIAGRAM FOR PIC16L(F)1717/9
FIGURE 2-5: 44-PIN TQFP (10X10) PACKAGE DIAGRAM FOR PIC16L(F)1717/9
3.0 MEMORY MAP

The memory is broken into two sections: program memory and configuration memory.

FIGURE 3-1: PIC16(L)F1713 PROGRAM MEMORY MAPPING
FIGURE 3-2: PIC16(L)F1716/7 PROGRAM MEMORY MAPPING

- 8000h: User ID Location
- 8001h: User ID Location
- 8002h: User ID Location
- 8003h: User ID Location
- 8004h: Reserved
- 8005h: Revision ID
- 8006h: Device ID
- 8007h: Configuration Word 1
- 8008h: Configuration Word 2
- 8009h: Calibration Word 1
- 800Ah: Calibration Word 2
- 800Bh: Calibration Word 3
- 800Ch: Calibration Word 4
- 800Dh: Reserved
- 800 Eh: Reserved
- 800Fh: Calibration Word 5
- 8010h: Calibration Word 6
- 8011h-81FFh: Reserved

Program Memory
Maps to 0-1FFFh
Implemented

Configuration Memory
Maps to 8000-81FF
Implemented

0000h-1FFFh
8 KW
FIGURE 3-3: PIC16(L)F1718/9 PROGRAM MEMORY MAPPING

- 0000h-3FFFh: Implemented
- 4000h-7FFFh: Implemented
- 8000h-81FF: Maps to 8000-81FF
- 8200h: Maps to 0-1FFFh
- FFFFh

- Program Memory
- Configuration Memory

- 8000h: User ID Location
- 8001h: User ID Location
- 8002h: User ID Location
- 8003h: User ID Location
- 8004h: Reserved
- 8005h: Revision ID
- 8006h: Device ID
- 8007h: Configuration Word 1
- 8008h: Configuration Word 2
- 8009h: Calibration Word 1
- 800Ah: Calibration Word 2
- 800Bh: Calibration Word 3
- 800Ch: Calibration Word 4
- 800Dh: Reserved
- 800 Eh: Reserved
- 800Fh: Calibration Word 5
- 8010h: Calibration Word 6
- 8011h-81FFh: Reserved
3.1 User ID Location

A user may store identification information (user ID) in four designated locations. The user ID locations are mapped to 8000h-8003h. Each location is 14 bits in length. Code protection has no effect on these memory locations. Each location may be read with code protection enabled or disabled.

Note: MPLAB® IDE only displays the seven Least Significant bits (LSb) of each user ID location; the upper bits are not read. It is recommended that only the seven LSbs be used if MPLAB IDE is the primary tool used to read these addresses.
3.2 Device/Revision ID

The 14-bit device ID word is located at 8006h and the 14-bit revision ID is located at 8005h. These locations are read-only and cannot be erased or modified.

REGISTER 3-1: DEVID: DEVICE ID REGISTER

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>R</th>
<th>R</th>
<th>R</th>
<th>R</th>
<th>R</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DEV&lt;13:8&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>bit 13</td>
<td></td>
<td></td>
<td></td>
<td>bit 8</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>R</th>
<th>R</th>
<th>R</th>
<th>R</th>
<th>R</th>
<th>R</th>
<th>R</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DEV&lt;7:0&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>bit 7</td>
<td></td>
<td></td>
<td></td>
<td>bit 0</td>
<td></td>
</tr>
</tbody>
</table>

Legend:

R = Readable bit
‘0’ = Bit is cleared ‘1’ = Bit is set x = Bit is unknown

bit 13-0 DEVID<13:0>: Device ID bits

Refer to Table 3-1 to determine what these bits will read on which device. A value of 3FFFh is invalid.

Note 1: This location cannot be written.

REGISTER 3-2: REVRID: REVISION ID REGISTER

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>R</th>
<th>R</th>
<th>R</th>
<th>R</th>
<th>R</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>REV&lt;13:8&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>bit 13</td>
<td></td>
<td></td>
<td></td>
<td>bit 8</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>R</th>
<th>R</th>
<th>R</th>
<th>R</th>
<th>R</th>
<th>R</th>
<th>R</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>REV&lt;7:0&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>bit 7</td>
<td></td>
<td></td>
<td></td>
<td>bit 0</td>
<td></td>
</tr>
</tbody>
</table>

Legend:

R = Readable bit
‘0’ = Bit is cleared ‘1’ = Bit is set x = Bit is unknown

bit 13-0 REVID<13:0>: Revision ID bits

These bits are used to identify the device revision.

Note 1: This location cannot be written.
<table>
<thead>
<tr>
<th>Device</th>
<th>Device ID</th>
<th>Revision ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIC16F1713</td>
<td>3049h</td>
<td>2xxxh</td>
</tr>
<tr>
<td>PIC16LF1713</td>
<td>304Bh</td>
<td>2xxxh</td>
</tr>
<tr>
<td>PIC16F1716</td>
<td>3048h</td>
<td>2xxxh</td>
</tr>
<tr>
<td>PIC16LF1716</td>
<td>304Ah</td>
<td>2xxxh</td>
</tr>
<tr>
<td>PIC16F1717</td>
<td>305Ch</td>
<td>2xxxh</td>
</tr>
<tr>
<td>PIC16LF1717</td>
<td>305Fh</td>
<td>2xxxh</td>
</tr>
<tr>
<td>PIC16F1718</td>
<td>305Bh</td>
<td>2xxxh</td>
</tr>
<tr>
<td>PIC16LF1718</td>
<td>305Eh</td>
<td>2xxxh</td>
</tr>
<tr>
<td>PIC16F1719</td>
<td>305Ah</td>
<td>2xxxh</td>
</tr>
<tr>
<td>PIC16LF1719</td>
<td>305Dh</td>
<td>2xxxh</td>
</tr>
</tbody>
</table>
3.3 Configuration Words

The device has two Configuration Words, Configuration Word 1 (8007h) and Configuration Word 2 (8008h). The individual bits within these Configuration Words are used to enable or disable device functions such as the Brown-out Reset, code protection and Power-up Timer.

REGISTER 3-3: CONFIGURATION WORD 1

<table>
<thead>
<tr>
<th>Bit 13</th>
<th>Bit 12</th>
<th>Bit 11</th>
<th>Bit 10-9</th>
<th>Bit 9</th>
<th>Bit 8</th>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>FCMEN</td>
<td>IESO</td>
<td>CLKOUT</td>
<td>BOREN&lt;1:0&gt;</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Legend:
- R = Readable bit
- P = Programmable bit
- U = Unimplemented bit, read as ‘1’
- ‘0’ = Bit is cleared
- ‘1’ = Bit is set
- n = Value when blank or after Bulk Erase

- **bit 13** FCMEN: Fail-Safe Clock Monitor Enable bit
  - 1 = ON Fail-Safe Clock Monitor is enabled
  - 0 = OFF Fail-Safe Clock Monitor is disabled

- **bit 12** IESO: Internal External Switchover bit
  - 1 = ON Internal/External Switchover mode is enabled
  - 0 = OFF Internal/External Switchover mode is disabled

- **bit 11** CLKOUTEN: Clock Out Enable bit
  - 1 = OFF CLKOUT function is disabled. I/O or oscillator function on CLKOUT
  - 0 = ON CLKOUT function is enabled on CLKOUT

- **bit 10-9** BOREN<1:0>: Brown-out Reset Enable bits\(^{(1)}\)
  - 11 = ON BOR enabled
  - 10 = SLEEP BOR enabled during operation and disabled in Sleep
  - 01 = SBODEN BOR controlled by SBOREN bit of the BORCON register
  - 00 = OFF BOR disabled

- **bit 8** Unimplemented: Read as ‘1’

- **bit 7** CP: Code Protection bit\(^{(2)}\)
  - 1 = OFF Program memory code protection is disabled
  - 0 = ON Program memory code protection is enabled

- **bit 6** MCLRE: MCLR/VPP Pin Function Select bit
  - If LVP bit = 1 (ON):
    - This bit is ignored.
  - If LVP bit = 0 (OFF):
    - 1 = ON MCLR/VPP pin function is MCLR; Weak pull-up enabled.
    - 0 = OFF MCLR/VPP pin function is digital input; MCLR internally disabled; Weak pull-up under control of port pin’s WPU control bit.

- **bit 5** PWRTE: Power-up Timer Enable bit\(^{(1)}\)
  - 1 = OFF PWRT disabled
  - 0 = ON PWRT enabled

**Note 1:** Enabling Brown-out Reset does not automatically enable Power-up Timer.

**Note 2:** The entire program memory will be erased when the code protection is turned off.
REGISTER 3-3: CONFIGURATION WORD 1 (CONTINUED)

bit 4-3  WDTE<1:0>: Watchdog Timer Enable bit
        11 = ON  WDT enabled
        10 = SLEEP  WDT enabled while running and disabled in Sleep
        01 = SWDTEN  WDT controlled by the SWDTEN bit in the WDTCON register
        00 = OFF  WDT disabled

bit 2-0  FOSC<2:0>: Oscillator Selection bits
        111 = ECH  External Clock, High-Power mode: CLkin on OSC1/CLKIN
        110 = ECM  External Clock, Medium-Power mode: CLkin on OSC1/CLKIN
        101 = ECL  External Clock, Low-Power mode: CLkin on OSC1/CLKIN
        100 = INTOSC  Internal HFINTOSC, I/O function on OSC1/CLKIN
        011 = EXTRC  External RC oscillator, RC function on OSC1/CLKIN
        010 = HS  High-speed crystal/resonator on OSC2/CLKOUT pin and OSC1/CLKIN
        001 = XT  Crystal/resonator on OSC2/CLKOUT pin and OSC1/CLKIN
        000 = LP  Low-power crystal on OSC2/CLKOUT pin and OSC1/CLKIN

Note 1: Enabling Brown-out Reset does not automatically enable Power-up Timer.
2: The entire program memory will be erased when the code protection is turned off.
## REGISTER 3-4: CONFIGURATION WORD 2

<table>
<thead>
<tr>
<th>R/P-1</th>
<th>R/P-1</th>
<th>R/P-1</th>
<th>R/P-1</th>
<th>R/P-1</th>
<th>R/P-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>LVP</td>
<td>DEBUG</td>
<td>LPBOR</td>
<td>BORV</td>
<td>STVREN</td>
<td>PLLen</td>
</tr>
</tbody>
</table>

### Legend:
- **R** = Readable bit
- **P** = Programmable bit
- **U** = Unimplemented bit, read as ‘1’
- ‘0’ = Bit is cleared
- ‘1’ = Bit is set
- **n** = Value when blank or after Bulk Erase

<table>
<thead>
<tr>
<th>R/P-1</th>
<th>U-1</th>
<th>U-1</th>
<th>U-1</th>
<th>U-1</th>
<th>R/P-1</th>
<th>R/P-1</th>
<th>R/P-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZCDDIS</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>PPS1WAY</td>
<td>WRT&lt;1:0&gt;</td>
<td></td>
</tr>
</tbody>
</table>

### Bit Descriptions:

**bit 13**
- **LVP**: Low-Voltage Programming Enable bit
  - **1** = ON Low-voltage programming enabled
  - **0** = OFF MCLR/VPP must be used for programming high voltage

**bit 12**
- **DEBUG**: In-Circuit Debugger Mode bit
  - **1** = OFF In-Circuit Debugger disabled, ICSPCLK and ICSPDAT are general purpose I/O pins
  - **0** = ON In-Circuit Debugger enabled, ICSPCLK and ICSPDAT are dedicated to the debugger

**bit 11**
- **LPBOR**: Low-Power Brown-out Reset Enable bit
  - **1** = ON Low-Power Brown-out Reset is disabled
  - **0** = ON Low-Power Brown-out Reset is enabled

**bit 10**
- **BORV**: Brown-out Reset Voltage Selection bit
  - **1** = LOW Brown-out Reset voltage (VBOR), low trip point selected
  - **0** = HIGH Brown-out Reset voltage (VBOR), high trip point selected

**bit 9**
- **STVREN**: Stack Overflow/Underflow Reset Enable bit
  - **1** = ON Stack Overflow or Underflow will cause a Reset
  - **0** = OFF Stack Overflow or Underflow will not cause a Reset

**bit 8**
- **PLLEN**: PLL Enable bit
  - **1** = ON 4xPLL enabled
  - **0** = OFF 4xPLL disabled

**bit 7**
- **ZCDDIS**: Zero-Cross Detect Disable bit
  - **1** = ON Zero-cross detection is disabled on POR. Zero-cross detection can be controlled by software.
  - **0** = OFF Zero-cross detection is always enabled. Software cannot disable zero-cross detection.

**bit 6-3**
- **Unimplemented**: Read as ‘1’

**bit 2**
- **PPS1WAY**: PPSLOCK One-Way Set Enable bit
  - **1** = ON The PPSLOCK bit is permanently set after the first access sequence that sets it.
  - **0** = OFF The PPSLOCK bit can be set and cleared as needed by the PPSLOCK access sequence.

### Notes:
1. The LVP bit cannot be programmed to ‘0’ when Programming mode is entered via LVP.
2. See **VBOR** parameter for specific trip point voltages.
REGISTER 3-4:  CONFIGURATION WORD 2 (CONTINUED)

bit 1-0  

WRT<1:0>: Flash Memory Self-Write Protection bits

4 kW Flash memory: (PIC16(L)F1713):

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>OFF</td>
</tr>
<tr>
<td>10</td>
<td>BOOT</td>
</tr>
<tr>
<td>01</td>
<td>HALF</td>
</tr>
<tr>
<td>00</td>
<td>ALL</td>
</tr>
</tbody>
</table>

8 kW Flash memory: (PIC16(L)F1716/7):

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>OFF</td>
</tr>
<tr>
<td>10</td>
<td>BOOT</td>
</tr>
<tr>
<td>01</td>
<td>HALF</td>
</tr>
<tr>
<td>00</td>
<td>ALL</td>
</tr>
</tbody>
</table>

16 kW Flash memory: (PIC16(L)F1718/9):

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>OFF</td>
</tr>
<tr>
<td>10</td>
<td>BOOT</td>
</tr>
<tr>
<td>01</td>
<td>HALF</td>
</tr>
<tr>
<td>00</td>
<td>ALL</td>
</tr>
</tbody>
</table>

Note 1: The LVP bit cannot be programmed to ‘0’ when Programming mode is entered via LVP.

2: See VbOR parameter for specific trip point voltages.
4.0 PROGRAM/VERIFY MODE

In Program/Verify mode, the program memory and the configuration memory can be accessed and programmed in serial fashion. ICSPDAT and ICSPCLK are used for the data and the clock, respectively. All commands and data words are transmitted LSb first. Data changes on the rising edge of the ICSPCLK and is latched on the falling edge. In Program/Verify mode, both the ICSPDAT and ICSPCLK are Schmitt Trigger inputs. The sequence that enters the device into Program/Verify mode places all other logic into the Reset state. Upon entering Program/Verify mode, all I/Os are automatically configured as high-impedance inputs and the address is cleared.

4.1 High-Voltage Program/Verify Mode Entry and Exit

There are two different methods of entering Program/Verify mode via high-voltage:

- **VPP** – First entry mode
- **VDD** – First entry mode

### 4.1.1 VPP – FIRST ENTRY MODE

To enter Program/Verify mode via the VPP-first method, the following sequence must be followed:

1. Hold ICSPCLK and ICSPDAT low. All other pins should be unpowered.
2. Raise the voltage on MCLR from 0V to VIHH.
3. Raise the voltage on VDD from 0V to the desired operating voltage.

The VPP-first entry prevents the device from executing code prior to entering Program/Verify mode. For example, when the Configuration Word has MCLR disabled (MCLRE = 0), the power-up time is disabled (PWRT = 0), the internal oscillator is selected (FOSC = 100), and RA0 and RA1 are driven by the user application, the device will execute code. Since this may prevent entry, VPP-first entry mode is strongly recommended. See the timing diagram in Figure 8-2.

### 4.1.2 VDD – FIRST ENTRY MODE

To enter Program/Verify mode via the VDD-first method, the following sequence must be followed:

1. Hold ICSPCLK and ICSPDAT low.
2. Raise the voltage on VDD from 0V to the desired operating voltage.
3. Raise the voltage on MCLR from VDD or below to VIHH.

The VDD-first method is useful when programming the device when VDD is already applied, for it is not necessary to disconnect VDD to enter Program/Verify mode. See the timing diagram in Figure 8-1.

4.1.3 PROGRAM/VERIFY MODE EXIT

To exit Program/Verify mode take MCLR to VDD or lower (VIL). See Figures 8-3 and 8-4.

**Note:** In systems where the VDD and MCLR/VPP signals can be controlled independently the VPP last method of exit should be used to keep the device in Reset, thereby preventing any issues that may be caused by program execution.

4.2 Low-Voltage Programming (LVP) Mode

The Low-Voltage Programming mode allows the PIC16(L)F171X devices to be programmed using VDD only, without high voltage. When the LVP bit of the Configuration Word 2 register is set to ‘1’, the low-voltage ICSP programming entry is enabled. To disable the Low-Voltage ICSP mode, the LVP bit must be programmed to ‘0’. This can only be done while in the High-Voltage Entry mode.

Entry into the Low-Voltage ICSP Program/Verify mode requires the following steps:

1. MCLR is brought to VIL.
2. A 32-bit key sequence is presented on ICSPDAT, while clocking ICSPCLK.

The key sequence is a specific 32-bit pattern, '0100 1101 0100 0011 0100 1000 0101 0000' (more easily remembered as MCHP in ASCII). The device will enter Program/Verify mode only if the sequence is valid. The Least Significant bit of the Least Significant nibble must be shifted in first.

Once the key sequence is complete, MCLR must be held at VIL for as long as Program/Verify mode is to be maintained.

For low-voltage programming timing, see Figures 8-8 and 8-9.

Exiting Program/Verify mode is done by no longer driving MCLR to VIL. See Figures 8-8 and 8-9.

**Note:** To enter LVP mode, the LSb of the Least Significant nibble must be shifted in first. This differs from entering the key sequence on other parts.
4.3 Program/Verify Commands

These devices implement 13 programming commands, each six bits in length. The commands are summarized in Table 4-1.

Commands that have data associated with them are specified to have a minimum delay of $T_{DLY}$ between the command and the data. After this delay, 16 clocks are required to either clock in or clock out the 14-bit data word. The first clock is for the Start bit and the last clock is for the Stop bit.

<table>
<thead>
<tr>
<th>Command</th>
<th>Mapping</th>
<th>Hex</th>
<th>Data/Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Load Configuration</td>
<td>x 0 0 0 0 0</td>
<td>00h</td>
<td>0, data (14), 0</td>
</tr>
<tr>
<td>Load Data For Program Memory</td>
<td>x 0 0 0 1 0</td>
<td>02h</td>
<td>0, data (14), 0</td>
</tr>
<tr>
<td>Read Data From Program Memory</td>
<td>x 0 0 1 0 0</td>
<td>04h</td>
<td>0, data (14), 0</td>
</tr>
<tr>
<td>Increment Address</td>
<td>x 0 0 1 1 0</td>
<td>06h</td>
<td>—</td>
</tr>
<tr>
<td>Reset Address</td>
<td>x 1 0 1 1 0</td>
<td>16h</td>
<td>—</td>
</tr>
<tr>
<td>Begin Internally Timed Programming</td>
<td>x 0 1 0 0 0</td>
<td>08h</td>
<td>—</td>
</tr>
<tr>
<td>Begin Externally Timed Programming</td>
<td>x 1 1 0 0 0</td>
<td>18h</td>
<td>—</td>
</tr>
<tr>
<td>End Externally Timed Programming</td>
<td>x 0 1 0 1 0</td>
<td>0Ah</td>
<td>—</td>
</tr>
<tr>
<td>Bulk Erase Program Memory</td>
<td>x 0 1 0 0 1</td>
<td>09h</td>
<td>Internally Timed</td>
</tr>
<tr>
<td>Row Erase Program Memory</td>
<td>x 1 0 0 0 1</td>
<td>11h</td>
<td>Internally Timed</td>
</tr>
</tbody>
</table>
4.3.1 LOAD CONFIGURATION

The Load Configuration command is used to access the configuration memory (User ID Locations, Configuration Words, Calibration Words). The Load Configuration command sets the address to 8000h and loads the data latches with one word of data (see Figure 4-1).

After issuing the Load Configuration command, use the Increment Address command until the proper address to be programmed is reached. The address is then programmed by issuing either the Begin Internally Timed Programming or Begin Externally Timed Programming command.

Note: Externally timed writes are not supported for Configuration and Calibration bits. Any externally timed write to the Configuration or Calibration Word will have no effect on the targeted word.

The only way to get back to the program memory (address 0) is to exit Program/Verify mode or issue the Reset Address command after the configuration memory has been accessed by the Load Configuration command.

FIGURE 4-1: LOAD CONFIGURATION

![Diagram of LOAD CONFIGURATION]

4.3.2 LOAD DATA FOR PROGRAM MEMORY

The Load Data for Program Memory command is used to load one 14-bit word into the data latches. The word programs into program memory after the Begin Internally Timed Programming or Begin Externally Timed Programming command is issued (see Figure 4-2).

FIGURE 4-2: LOAD DATA FOR PROGRAM MEMORY

![Diagram of LOAD DATA FOR PROGRAM MEMORY]
4.3.3 READ DATA FROM PROGRAM MEMORY

The Read Data from Program Memory command will transmit data bits out of the program memory map currently accessed, starting with the second rising edge of the clock input. The ICSPDAT pin will go into Output mode on the first falling clock edge, and it will revert to Input mode (high-impedance) after the 16th falling edge of the clock. If the program memory is code-protected (CP), the data will be read as zeros (see Figure 4-3).

FIGURE 4-3: READ DATA FROM PROGRAM MEMORY

4.3.4 INCREMENT ADDRESS

The address is incremented when this command is received. It is not possible to decrement the address. To reset this counter, the user must use the Reset Address command or exit Program/Verify mode and re-enter it.

If the address is incremented from address 7FFFh, it will wrap-around to location 0000h. If the address is incremented from FFFFh, it will wrap-around to location 8000h (see Figure 4-4).

FIGURE 4-4: INCREMENT ADDRESS
4.3.5 RESET ADDRESS

The Reset Address command will reset the address to 0000h, regardless of the current value. The address is used in program memory or the configuration memory. See Figure 4-5.

FIGURE 4-5: RESET ADDRESS

4.3.6 BEGIN INTERNALLY TIMED PROGRAMMING

A Load Configuration or Load Data for Program Memory command must be given before every Begin Programming command. Programming of the addressed memory will begin after this command is received. An internal timing mechanism executes the write. The user must allow for the program cycle time, TPINT, in order for the programming to complete.

The End Externally Timed Programming command is not needed when the Begin Internally Timed Programming is used to start the programming.

The program memory address that is being programmed is not erased prior to being programmed. See Figure 4-6.

FIGURE 4-6: BEGIN INTERNALLY TIMED PROGRAMMING
4.3.7 BEGIN EXTERNALLY TIMED PROGRAMMING

A Load Configuration or Load Data for Program Memory command must be given before every Begin Programming command. Programming of the addressed memory will begin after this command is received. To complete the programming, the End Externally Timed Programming command must be sent in the specified time window defined by TPEXT. See Figure 4-7.

Externally timed writes are not supported for Configuration and Calibration bits. Any externally timed write to the Configuration or Calibration Word will have no effect on the targeted word.

FIGURE 4-7: BEGIN EXTERNALLY TIMED PROGRAMMING

4.3.8 END EXTERNALLY TIMED PROGRAMMING

This command is required after a Begin Externally Timed Programming command is given. This command must be sent within the time window specified by TPEXT after the Begin Externally Timed Programming command is sent.

After sending the End Externally Timed Programming command, an additional delay (TDIS) is required before sending the next command. This delay is longer than the delay ordinarily required between other commands. See Figure 4-8.

FIGURE 4-8: END EXTERNALLY TIMED PROGRAMMING
4.3.9  BULK ERASE PROGRAM MEMORY

The Bulk Erase Program Memory command performs two different functions, dependent on the current state of the address.

Address 0000h-7FFFh:
- Program Memory is erased
- Configuration Words are erased

Address 8000h-8008h:
- Program Memory is erased
- Configuration Words are erased
- User ID Locations are erased

A Bulk Erase Program Memory command should not be issued when the address is greater than 8008h.

Note: The code protection Configuration bit (CP) has no effect on the Bulk Erase Program Memory command.

4.3.10  ROW ERASE PROGRAM MEMORY

The Row Erase Program Memory command will erase an individual row. Refer to Table 4-2 for row sizes of specific devices and the PC bits used to address them. If the program memory is code-protected, the Row Erase Program Memory command will be ignored. When the address is 8000h-8008h, the Row Erase Program Memory command will only erase the user ID locations regardless of the setting of the CP Configuration bit.

After receiving the Row Erase Program Memory command, the erase will not complete until the time interval, TERAR, has expired. See Figure 4-10.

FIGURE 4-9:  BULK ERASE PROGRAM MEMORY

FIGURE 4-10:  ROW ERASE PROGRAM MEMORY
## TABLE 4-2: PROGRAMMING ROW AND LATCH SIZES

<table>
<thead>
<tr>
<th>Devices</th>
<th>PC</th>
<th>Erase Row Size (Number of 14-bit Words)</th>
<th>Write Row Size (Number of 14-bit Latches)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIC16F1713</td>
<td>&lt;15:5&gt;</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>PIC16F1716</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PIC16F1717</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PIC16F1718</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PIC16F1719</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PIC16LF1713</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PIC16LF1716</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PIC16LF1717</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PIC16LF1718</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PIC16LF1719</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
5.0 PROGRAMMING ALGORITHMS

The devices use internal latches to temporarily store the 14-bit words used for programming. Refer to Table 4-2 for specific latch information. The data latches allow the user to write the program words with a single Begin Externally Timed Programming or Begin Internally Timed Programming command. The Load Program Data or the Load Configuration command is used to load a single data latch. The data latch will hold the data until the Begin Externally Timed Programming or Begin Internally Timed Programming command is given.

The data latches are aligned with the LSbs of the address. The PS address bits indicated in Table 4-2 at the time the Begin Externally Timed Programming or Begin Internally Timed Programming command is given will determine which memory row is written. Writes cannot cross a physical row boundary. For example, attempting to write from address 0002h-0021h in a 32-latch device will result in data being written to 0020h-003Fh.

If more than the maximum number of latches are written without a Begin Externally Timed Programming or Begin Internally Timed Programming command, the data in the data latches will be overwritten. The following figures show the recommended flowcharts for programming.
FIGURE 5-1: DEVICE PROGRAM/VERIFY FLOWCHART

[Flowchart diagram showing the steps of the process: Start, Enter Programming Mode, Bulk Erase Device, Write Program Memory, Write User IDs, Verify Program Memory, Verify User IDs, Write Configuration Words, Verify Configuration Words, Exit Programming Mode, Done.]

Note 1: See Figure 5-2.
Note 2: See Figure 5-5.
FIGURE 5-2: PROGRAM MEMORY FLOWCHART

Start

Bulk Erase Program Memory\(^{(1, 2)}\)

Program Cycle\(^{(3)}\)

Read Data from Program Memory

Data Correct?

Yes

Yes

Increment Address Command

No

All Locations Done?

No

Report Programming Failure

Done

Note 1: This step is optional if the device has already been erased or has not been previously programmed.

2: If the device is code-protected or must be completely erased, then Bulk Erase the device per Figure 5-6.

3: See Figure 5-3 or Figure 5-4.
FIGURE 5-3: ONE-WORD PROGRAM CYCLE

Program Cycle

Load Data for Program Memory

Begin Programming Command (Internally timed)

Wait TPINT

Begin Programming Command (Externally timed)\(^{(1)}\)

Wait TPEXT

End Programming Command

Wait TDIS

Note 1: Externally timed writes are not supported for Configuration and Calibration bits.
FIGURE 5-4: MULTIPLE-WORD PROGRAM CYCLE

Program Cycle

Load Data for Program Memory

Latch 1

Increment Address Command

Load Data for Program Memory

Latch 2

Latch 1

Latch 2

Latch 32

Begin Programming Command (Internally timed)

Wait TPINT

Begin Programming Command (Externally timed)

Wait TPEXT

End Programming Command

Wait TDIS
FIGURE 5-5: CONFIGURATION MEMORY PROGRAM FLOWCHART

Start

Load Configuration

Bulk Erase Program Memory(1)

One-word Program Cycle(2)
(User ID)

Read Data From Program Memory Command

Data Correct? Yes

Increment Address Command

Report Programming Failure

Address = 8004h?

No

Increment Address Command

Yes

One-word Program Cycle(2)
(Config. Word 1)

Read Data From Program Memory Command

Data Correct? No

Report Programming Failure

Yes

Increment Address Command

One-word Program Cycle(2)
(Config. Word 2)

Read Data From Program Memory Command

Data Correct? No

Report Programming Failure

Yes

Done

Note 1: This step is optional if the device is erased or not previously programmed.

2: See Figure 5-3.
FIGURE 5-6: ERASE FLOWCHART

Start

Load Configuration

Bulk Erase Program Memory

Done

Note: This sequence does not erase the Calibration Words.
6.0 CODE PROTECTION

Code protection is controlled using the CP bit in Configuration Word 1. When code protection is enabled, all program memory locations (0000h-7FFFh) read as ‘0’. Further programming is disabled for the program memory (0000h-7FFFh). Program memory can still be programmed and read during program execution.

The user ID locations and Configuration Words can be programmed and read out regardless of the code protection settings.

6.1 Program Memory

Code protection is enabled by programming the CP bit in Configuration Word 1 register to ‘0’.

The only way to disable code protection is to use the Bulk Erase Program Memory command.

7.0 HEX FILE USAGE

In the hex file there are two bytes per program word stored in the Intel® INHX32 hex format. Data is stored LSB first, MSB second. Because there are two bytes per word, the addresses in the hex file are 2x the address in program memory. (Example: The Configuration Word 1 is stored at 8007h. In the hex file this will be referenced as 1000Eh-1000Fh).

7.1 Configuration Word

To allow portability of code, it is strongly recommended that the programmer is able to read the Configuration Words and user ID locations from the hex file. If the Configuration Words information was not present in the hex file, a simple warning message may be issued. Similarly, while saving a hex file, Configuration Words and user ID information should be included.

7.2 Device ID

If a device ID is present in the hex file at 1000Ch-1000Dh (8006h on the part), the programmer should verify the device ID against the value read from the part. On a mismatch condition, the programmer should generate a warning message.

7.3 Checksum Computation

The checksum is calculated by two different methods dependent on the setting of the CP Configuration bit.

<table>
<thead>
<tr>
<th>TABLE 7-1: CONFIGURATION WORD MASK VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device</td>
</tr>
<tr>
<td>--------</td>
</tr>
<tr>
<td>PIC16F1713</td>
</tr>
<tr>
<td>PIC16LF1713</td>
</tr>
<tr>
<td>PIC16F1716</td>
</tr>
<tr>
<td>PIC16LF1716</td>
</tr>
<tr>
<td>PIC16F1717</td>
</tr>
<tr>
<td>PIC16LF1717</td>
</tr>
<tr>
<td>PIC16F1718</td>
</tr>
<tr>
<td>PIC16LF1718</td>
</tr>
<tr>
<td>PIC16F1719</td>
</tr>
<tr>
<td>PIC16LF1719</td>
</tr>
</tbody>
</table>

7.3.1 PROGRAM CODE PROTECTION DISABLED

With the program code protection disabled, the checksum is computed by reading the contents of the PIC16(L)F171X program memory locations and adding up the program memory data starting at address 0000h, up to the maximum user addressable location (e.g., FFFh for the PIC16F1713). Any Carry bits exceeding 16 bits are ignored. Additionally, the relevant bits of the Configuration Words are added to the checksum. All unimplemented Configuration bits are masked to ‘0’.
7.3.2 PROGRAM CODE PROTECTION ENABLED

When the MPLAB® IDE check box for Configure->ID Memory...-> Use Unprotected Checksum is checked, then the 16-bit checksum of the equivalent unprotected device is computed and stored in the user ID. Each nibble of the unprotected checksum is stored in the Least Significant nibble of each of the four user ID locations. The Most Significant checksum nibble is stored in the user ID at location 8000h, the second Most Significant nibble is stored at location 8001h, and so forth for the remaining nibbles and ID locations. The protected checksums in Table 7-2 assume that the Use Unprotected Checksum box is checked.

The checksum of a code-protected device is computed in the following manner: the Least Significant nibble of each user ID is used to create a 16-bit value. The Least Significant nibble of user ID location 8000h is the Most Significant nibble of the 16-bit value. The Least Significant nibble of user ID location 8001h is the second Most Significant nibble, and so forth for the remaining user IDs and 16-bit value nibbles. The resulting 16-bit value is summed with the Configuration Words. All unimplemented Configuration bits are masked to ‘0’.

<table>
<thead>
<tr>
<th>Device</th>
<th>Config1</th>
<th>Config2</th>
<th>Checksum</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Unprotected</td>
<td>Protected</td>
<td>Mask</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PIC16F1713</td>
<td>3FFh</td>
<td>3F7Fh</td>
<td>3EFFh</td>
</tr>
<tr>
<td>PIC16F1716</td>
<td>3FFh</td>
<td>3F7Fh</td>
<td>3EFFh</td>
</tr>
<tr>
<td>PIC16F1717</td>
<td>3FFh</td>
<td>3F7Fh</td>
<td>3EFFh</td>
</tr>
<tr>
<td>PIC16F1718</td>
<td>3FFh</td>
<td>3F7Fh</td>
<td>3EFFh</td>
</tr>
<tr>
<td>PIC16F1719</td>
<td>3FFh</td>
<td>3F7Fh</td>
<td>3EFFh</td>
</tr>
<tr>
<td>PIC16LF1713</td>
<td>3FFh</td>
<td>3F7Fh</td>
<td>3EFFh</td>
</tr>
<tr>
<td>PIC16LF1716</td>
<td>3FFh</td>
<td>3F7Fh</td>
<td>3EFFh</td>
</tr>
<tr>
<td>PIC16LF1717</td>
<td>3FFh</td>
<td>3F7Fh</td>
<td>3EFFh</td>
</tr>
<tr>
<td>PIC16LF1718</td>
<td>3FFh</td>
<td>3F7Fh</td>
<td>3EFFh</td>
</tr>
<tr>
<td>PIC16LF1719</td>
<td>3FFh</td>
<td>3F7Fh</td>
<td>3EFFh</td>
</tr>
</tbody>
</table>
8.0 ELECTRICAL SPECIFICATIONS

Refer to device specific data sheet for absolute maximum ratings.

### TABLE 8-1: AC/DC CHARACTERISTICS TIMING REQUIREMENTS FOR PROGRAM/VERIFY MODE

<table>
<thead>
<tr>
<th>AC/DC CHARACTERISTICS</th>
<th>Standard Operating Conditions</th>
<th>Production tested at 25°C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sym.</td>
<td>Characteristics</td>
<td>Min.</td>
</tr>
<tr>
<td>VDD</td>
<td>Supply Voltage <em>(VDDMIN(2), VDDMAX)</em></td>
<td>PIC16LF171X: 1.80, 2.50</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PIC16F171X: 2.30, 2.50</td>
</tr>
<tr>
<td>VPEW</td>
<td>Read/Write and Row Erase operations</td>
<td>VDDMIN: —</td>
</tr>
<tr>
<td>VBE</td>
<td>Bulk Erase operations</td>
<td>2.7</td>
</tr>
<tr>
<td>IDD</td>
<td>Current on VDD, Idle</td>
<td>—</td>
</tr>
<tr>
<td>IDDP</td>
<td>Current on VDD, Programming</td>
<td>—</td>
</tr>
<tr>
<td>IPP</td>
<td>VPP Current on MCLR/VPP</td>
<td>—</td>
</tr>
<tr>
<td>VIH</td>
<td>(ICSPCLK, ICSPDAT, MCLR/VPP) input high level</td>
<td>0.8 VDD: —</td>
</tr>
<tr>
<td>VIL</td>
<td>(ICSPCLK, ICSPDAT, MCLR/VPP) input low level</td>
<td>—</td>
</tr>
<tr>
<td>VOH</td>
<td>ICSPDAT output high level</td>
<td>VDD-0.7: —</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VDD-0.7: —</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VDD-0.7: —</td>
</tr>
<tr>
<td>VOL</td>
<td>ICSPDAT output low level</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Vss+0.6: V</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Vss+0.6: V</td>
</tr>
<tr>
<td>VBOR</td>
<td>Brown-out Reset Voltage:</td>
<td>BORV = 0 (high trip): —</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BORV = 1 (low trip): —</td>
</tr>
<tr>
<td></td>
<td></td>
<td>—</td>
</tr>
<tr>
<td>Programming Mode Entry and Exit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TENTS</td>
<td>Programming mode entry setup time: ICSPCLK, ICSPDAT setup time before VDD or MCLR↑</td>
<td>100</td>
</tr>
<tr>
<td>TENTH</td>
<td>Programming mode entry hold time: ICSPCLK, ICSPDAT hold time after VDD or MCLR↑</td>
<td>250</td>
</tr>
<tr>
<td>Serial Program/Verify</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TCKL</td>
<td>Clock Low Pulse Width</td>
<td>100</td>
</tr>
<tr>
<td>TCKH</td>
<td>Clock High Pulse Width</td>
<td>100</td>
</tr>
<tr>
<td>TDS</td>
<td>Data in setup time before clock↓</td>
<td>100</td>
</tr>
<tr>
<td>TDH</td>
<td>Data in hold time after clock↓</td>
<td>100</td>
</tr>
<tr>
<td>TCO</td>
<td>Clock↑ to data out valid (during a Read Data command)</td>
<td>0</td>
</tr>
<tr>
<td>TLZD</td>
<td>Clock↓ to data low-impedance (during a Read Data command)</td>
<td>0</td>
</tr>
</tbody>
</table>

**NOTES:**
1. Specified values are guaranteed by design.
2. Specified values are guaranteed by test.
3. Values in ( ) are for internal use only.

**FIGURE:**

- FIGURE 8-1: PIC16(L)F171X Device Diagram

© 2013 Microchip Technology Inc.
TABLE 8-1: AC/DC CHARACTERISTICS TIMING REQUIREMENTS FOR PROGRAM/VERIFY MODE (CONTINUED)

<table>
<thead>
<tr>
<th>AC/DC CHARACTERISTICS</th>
<th>Standard Operating Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Production tested at 25°C</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sym.</th>
<th>Characteristics</th>
<th>Min.</th>
<th>Typ.</th>
<th>Max.</th>
<th>Units</th>
<th>Conditions/Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>THZD</td>
<td>Clock↓ to data high-impedance (during a Read Data command)</td>
<td>0</td>
<td>—</td>
<td>80</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>TDLY</td>
<td>Data input not driven to next clock input (delay required between command/data or command/command)</td>
<td>1.0</td>
<td>—</td>
<td>—</td>
<td>μs</td>
<td></td>
</tr>
<tr>
<td>TERAB</td>
<td>Bulk Erase cycle time</td>
<td>—</td>
<td>—</td>
<td>5</td>
<td>ms</td>
<td></td>
</tr>
<tr>
<td>TERAR</td>
<td>Row Erase cycle time</td>
<td>—</td>
<td>—</td>
<td>2.5</td>
<td>ms</td>
<td></td>
</tr>
<tr>
<td>TPINT</td>
<td>Internally timed programming operation time</td>
<td>—</td>
<td>—</td>
<td>2.5</td>
<td>ms</td>
<td>Program memory</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td>ms</td>
<td>Configuration Words</td>
</tr>
<tr>
<td>TPEXT</td>
<td>Externally timed programming pulse</td>
<td>1.0</td>
<td>—</td>
<td>2.1</td>
<td>ms</td>
<td>Note 1</td>
</tr>
<tr>
<td>TDIS</td>
<td>Time delay from program to compare (HV discharge time)</td>
<td>300</td>
<td>—</td>
<td>—</td>
<td>μs</td>
<td></td>
</tr>
<tr>
<td>TXT</td>
<td>Time delay when exiting Program/Verify mode</td>
<td>1</td>
<td>—</td>
<td>—</td>
<td>μs</td>
<td></td>
</tr>
</tbody>
</table>

Note 1: Externally timed writes are not supported for Configuration and Calibration bits.

2: Bulk-erased devices default to brown-out enabled. VDDMIN is 2.85 volts when performing low-voltage programming on a bulk-erased device, to ensure that the device is not held in Brown-out Reset.

8.1 AC Timing Diagrams

FIGURE 8-1: PROGRAMMING MODE ENTRY – Vdd FIRST

FIGURE 8-2: PROGRAMMING MODE ENTRY – Vpp FIRST

FIGURE 8-3: PROGRAMMING MODE EXIT – Vpp LAST

FIGURE 8-4: PROGRAMMING MODE EXIT – Vdd LAST
FIGURE 8-5: CLOCK AND DATA TIMING

ICSPCLK

ICSPDAT as input

ICSPDAT as output

ICSPDAT from input to output

ICSPDAT from output to input

FIGURE 8-6: WRITE COMMAND – PAYLOAD TIMING

FIGURE 8-7: READ COMMAND – PAYLOAD TIMING
FIGURE 8-8: LVP ENTRY (POWERING UP)

FIGURE 8-9: LVP ENTRY (POWERED)

Note: Sequence matching can start with no edge on MCLR first.
APPENDIX A: REVISION HISTORY

Revision A (06/2013)
Initial release of this document.

Revision B (08/2013)
Updated pin diagrams.

Revision C (12/2013)
Changed the family name to PIC16(L)F171X; Added PIC16(L)F1717, PIC16(L)F1718 and PIC16(L)F1719; Updated Table 8-1 in Chapter 8.0 (Electrical Specifications); Other minor corrections.
Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.

- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.

- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip’s Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.

- Microchip is willing to work with the customer who is concerned about the integrity of their code.

- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as “unbreakable.”

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip’s code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications.

Trademarks

- The Microchip name and logo, the Microchip logo, dsPIC, FlashFlex, KEELoo, KEELoo logo, MPLAB, PIC, PICmicro, PICSTART, PIC32® logo, rPIC, SST, SST Logo, SuperFlash and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

- FilterLab, Hampshire, Hi-TECH C, Linear Active Thermistor, MTP, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

- Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

- Analog-for-the-Digital Age, Application Maestro, BodyCom, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, Hi-TIDE, In-Circuit Serial Programming, ICSP, Mindi, MiWi, MPASM, MPP, MPLAB Certified logo, MPLIB, MPLINK, mTouch, Omniscient Code Generation, PICC, PICC-18, PICDEM, PICDEM.net, PICKit, PICtail, REAL ICE, rFLAB, Select Mode, SQI, Serial Quad I/O, Total Endurance, TSHARC, UniWinDriver, WiperLock, ZENA and Z-Scale are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

- SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

- GestIC and ULPP are registered trademarks of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

- All other trademarks mentioned herein are property of their respective companies.

© 2013, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

Printed on recycled paper.

isbn: 9781620777152

QUALITY MANAGEMENT SYSTEM CERTIFIED BY DNV

ISO/TS 16949

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company’s quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip’s quality system for the design and manufacture of development systems is ISO 9001:2000 certified.