This document includes the programming specifications for the following devices:

- PIC12F1822 • PIC12LF1822
- PIC16F1823 • PIC16LF1823
- PIC16F1824 • PIC16LF1824
- PIC16F1825 • PIC16LF1825
- PIC16F1826 • PIC16LF1826
- PIC16F1827 • PIC16LF1827
- PIC16F1828 • PIC16LF1828
- PIC16F1829 • PIC16LF1829

1.0 OVERVIEW

The PIC12(L)F1822 and PIC16(L)F182X devices 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, these devices require 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, these 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 and Table 1-2.

**TABLE 1-1: PIN DESCRIPTIONS DURING PROGRAMMING – PIC16(L)F1826, PIC16(L)F1827**

<table>
<thead>
<tr>
<th>Pin Name</th>
<th>Function</th>
<th>Pin Type</th>
<th>Pin Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RB6</td>
<td>ICSPCLK</td>
<td>I</td>
<td>Clock Input – Schmitt Trigger Input</td>
</tr>
<tr>
<td>RB7</td>
<td>ICSPDAT</td>
<td>I/O</td>
<td>Data Input/Output – Schmitt Trigger Input</td>
</tr>
<tr>
<td>RA5/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:** In the PIC12(L)F1822 and PIC16(L)F182X, 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.

**TABLE 1-2: PIN DESCRIPTIONS DURING PROGRAMMING – PIC12(L)F1822, PIC16(L)F1823, PIC16(L)F1824, PIC16(L)F1825, PIC16(L)F1828 and PIC16(L)F1829**

<table>
<thead>
<tr>
<th>Pin Name</th>
<th>Function</th>
<th>Pin Type</th>
<th>Pin Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RA1</td>
<td>ICSPCLK</td>
<td>I</td>
<td>Clock Input – Schmitt Trigger Input</td>
</tr>
<tr>
<td>RA0</td>
<td>ICSPDAT</td>
<td>I/O</td>
<td>Data Input/Output – Schmitt Trigger Input</td>
</tr>
<tr>
<td>RA3/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:** In the PIC12(L)F1822 and PIC16(L)F182X, 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 PIC12(L)F1822 and PIC16(L)F182X family are shown in Figure 2-1 through Figure 2-9. 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: 18-PIN DIAGRAM FOR PIC16(L)F1826/1827**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RA2</td>
<td></td>
</tr>
<tr>
<td>RA3</td>
<td></td>
</tr>
<tr>
<td>RA4</td>
<td></td>
</tr>
<tr>
<td>RA5/MCLR/VPP</td>
<td></td>
</tr>
<tr>
<td>VSS</td>
<td></td>
</tr>
<tr>
<td>RB0</td>
<td></td>
</tr>
<tr>
<td>RB1</td>
<td></td>
</tr>
<tr>
<td>RB2</td>
<td></td>
</tr>
<tr>
<td>RB3</td>
<td></td>
</tr>
</tbody>
</table>

**FIGURE 2-2: 20-PIN DIAGRAM FOR PIC16(L)F1826/1827**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RA2</td>
<td></td>
</tr>
<tr>
<td>RA3</td>
<td></td>
</tr>
<tr>
<td>RA4</td>
<td></td>
</tr>
<tr>
<td>RA5/MCLR/VPP</td>
<td></td>
</tr>
<tr>
<td>VSS</td>
<td></td>
</tr>
<tr>
<td>RB0</td>
<td></td>
</tr>
<tr>
<td>RB1</td>
<td></td>
</tr>
<tr>
<td>RB2</td>
<td></td>
</tr>
<tr>
<td>RB3</td>
<td></td>
</tr>
<tr>
<td>VDD</td>
<td></td>
</tr>
</tbody>
</table>

**FIGURE 2-3: 28-PIN DIAGRAM FOR PIC16(L)F1826/1827**

**FIGURE 2-4: 8-PIN DIAGRAM FOR PIC12(L)F1822**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RA2</td>
<td></td>
</tr>
<tr>
<td>RA3</td>
<td></td>
</tr>
<tr>
<td>RA4</td>
<td></td>
</tr>
<tr>
<td>RA5/MCLR/VPP</td>
<td></td>
</tr>
<tr>
<td>VSS</td>
<td></td>
</tr>
</tbody>
</table>

**FIGURE 2-5: 8-PIN DIAGRAM FOR PIC12(L)F1822**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RA2</td>
<td></td>
</tr>
<tr>
<td>RA3</td>
<td></td>
</tr>
<tr>
<td>RA4</td>
<td></td>
</tr>
<tr>
<td>RA5/MCLR/VPP</td>
<td></td>
</tr>
</tbody>
</table>
FIGURE 2-6: 14-PIN DIAGRAM FOR PIC16(L)F1823/1824/1825

PDIP, SOIC, TSSOP

VDD  14  Vss
RA5  13
RA4  12
RA3/MCLR/VPP  11  RA2
RC5  10  RC0
RC4  9  RC1
RC3  8

FIGURE 2-7: 16-PIN DIAGRAM FOR PIC16(L)F1823/1824/1825

QFN (3x3 or 4x4)

VDD  14  Vss
VDD  13  VSS
VDD  12  VSS
VDD  11  VSS
RA5  10  RA2
RA4  9  RC0
RA3/MCLR/VPP  8  RC1
RC5  7
RC4  6
RC3  5
RC2  4
RC1  3
RC0  2
RA2  1

FIGURE 2-8: 20-PIN DIAGRAM FOR PIC16(L)F1828/1829

PDIP, SOIC, TSSOP

VDD  20  Vss
RA5  19  RA2
RA4  18  RA1/ICSPCLK
RA3/MCLR/VPP  17  RC1
RC5  16  RC0
RC4  15  RC1
RC3  14  RC2
RC2  13  RB4
RC1  12  RB5
RC0  11  RB6

FIGURE 2-9: 20-PIN DIAGRAM FOR PIC16(L)F1828/1829

QFN 4x4

RA4  15  RA5  14  VDD  13  Vss  12  RA5  11  RA4  10  RA3/MCLR/VPP  9  RC6  8  RC5  7  RC4  6  RC3  5  RC2  4  RC1  3  RC0  2  RA2  1  RA1/ICSPCLK  0  2019181716
3.0 MEMORY MAP

The memory for the PIC12(L)F1822 and PIC16(L)F182X devices is broken into two sections: program memory and configuration memory. Only the size of the program memory changes between devices, the configuration memory remains the same.

FIGURE 3-1: PIC12(L)F1822, PIC16(L)F1823 AND PIC16(L)F1826 PROGRAM MEMORY MAPPING
FIGURE 3-2: PIC16(L)F1827, PIC16(L)F1824 AND PIC16(L)F1828 PROGRAM MEMORY MAPPING

- 8000h - User ID Location
- 8001h - User ID Location
- 8002h - User ID Location
- 8003h - User ID Location
- 8004h - Reserved
- 8005h - Reserved
- 8006h - Device ID
- 8007h - Configuration Word 1
- 8008h - Configuration Word 2
- 8009h - Calibration Word 1
- 800ah - Calibration Word 2
- 800bh-81ffh - Reserved

4 KW Implemented

Maps to 0-0FFFh

Maps to 8000-81FFh

Reserved

Device ID

Configuration Word 1

Configuration Word 2

Calibration Word 1

Calibration Word 2

Reserved

User ID Location

User ID Location

User ID Location

User ID Location

User ID Location

Program Memory

Configuration Memory

0000h

0FFFh

7FFFh

8000h

8200h

FFFFh

User ID Location
FIGURE 3-3: PIC16(L)F1825 AND PIC16(L)F1829 PROGRAM MEMORY MAPPING

- 0000h to 1FFFh: Implemented
- 7FFFh: Implemented
- 8000h to 81FFh: Maps to Program Memory
- 8200h to FFFFh: Maps to Configuration Memory

- 8000h: User ID Location
- 8001h: User ID Location
- 8002h: User ID Location
- 8003h: User ID Location
- 8004h: Reserved
- 8005h: Reserved
- 8006h: Device ID
- 8007h: Configuration Word 1
- 8008h: Configuration Word 2
- 8009h: Calibration Word 1
- 800Ah: Calibration Word 2
- 800Bh-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 LSb’s be used if MPLAB IDE is the primary tool used to read these addresses.

**REGISTER 3-1: DEVICE ID: DEVICE ID REGISTER(1)**

<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>DEV8</td>
<td>DEV7</td>
<td>DEV6</td>
<td>DEV5</td>
<td>DEV4</td>
<td>DEV3</td>
<td>bit 13</td>
<td>bit 8</td>
</tr>
<tr>
<td>DEV2</td>
<td>DEV1</td>
<td>DEV0</td>
<td>REV4</td>
<td>REV3</td>
<td>REV2</td>
<td>REV1</td>
<td>REV0</td>
</tr>
</tbody>
</table>

**Legend:**

- **P** = Programmable bit
- **U** = Unimplemented bit, read as ‘0’
- **R** = Readable bit
- **W** = Writable bit
- ‘0’ = Bit is cleared
- ‘1’ = Bit is set
- -n = Value at POR
- x = Bit is unknown

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

These bits are used to identify the part number.

bit 4-0 **REV<4:0>:** Revision ID bits

These bits are used to identify the revision.

**Note:** This location cannot be written.

3.2 Device ID

The device ID word is located at 8006h. This location is read-only and cannot be erased or modified.
3.3 Configuration Words

There are 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.

3.4 Calibration Words

The internal calibration values are factory calibrated and stored in Calibration Words 1 and 2 (8009h, 800Ah).

The Calibration Words do not participate in erase operations. The device can be erased without affecting the Calibration Words.

TABLE 3-1: DEVICE ID VALUES

<table>
<thead>
<tr>
<th>DEVICE ID VALUES</th>
<th>DEV</th>
<th>REV</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIC16F1826</td>
<td>10 0111 100</td>
<td>x xxxx</td>
</tr>
<tr>
<td>PIC16F1827</td>
<td>10 0111 101</td>
<td>x xxxx</td>
</tr>
<tr>
<td>PIC16LF1826</td>
<td>10 1000 100</td>
<td>x xxxx</td>
</tr>
<tr>
<td>PIC16LF1827</td>
<td>10 1000 101</td>
<td>x xxxx</td>
</tr>
<tr>
<td>PIC16F1823</td>
<td>10 0111 001</td>
<td>x xxxx</td>
</tr>
<tr>
<td>PIC16LF1823</td>
<td>10 1000 001</td>
<td>x xxxx</td>
</tr>
<tr>
<td>PIC12F1822</td>
<td>10 0111 000</td>
<td>x xxxx</td>
</tr>
<tr>
<td>PIC12LF1822</td>
<td>10 1000 000</td>
<td>x xxxx</td>
</tr>
<tr>
<td>PIC16F1824</td>
<td>10 0111 010</td>
<td>x xxxx</td>
</tr>
<tr>
<td>PIC16LF1824</td>
<td>10 1000 010</td>
<td>x xxxx</td>
</tr>
<tr>
<td>PIC16F1825</td>
<td>10 0111 011</td>
<td>x xxxx</td>
</tr>
<tr>
<td>PIC16LF1825</td>
<td>10 1000 011</td>
<td>x xxxx</td>
</tr>
<tr>
<td>PIC16F1828</td>
<td>10 0111 110</td>
<td>x xxxx</td>
</tr>
<tr>
<td>PIC16LF1828</td>
<td>10 1000 110</td>
<td>x xxxx</td>
</tr>
<tr>
<td>PIC16F1829</td>
<td>10 0111 111</td>
<td>x xxxx</td>
</tr>
<tr>
<td>PIC16LF1829</td>
<td>10 1000 111</td>
<td>x xxxx</td>
</tr>
</tbody>
</table>
**REGISTER 3-2: CONFIGURATION WORD 1**

<table>
<thead>
<tr>
<th>Bit</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td><strong>FCMEN</strong>: Fail-Safe Clock Monitor Enable bit</td>
</tr>
<tr>
<td>12</td>
<td><strong>IESO</strong>: Internal External Switchover bit</td>
</tr>
<tr>
<td>11</td>
<td><strong>CLKOUTEN</strong>: Clock Out Enable bit</td>
</tr>
<tr>
<td>10-9</td>
<td><strong>BOREN&lt;1:0&gt;</strong>: Brown-out Reset Enable bits</td>
</tr>
<tr>
<td>8</td>
<td><strong>CPD</strong>: Data Code Protection bit</td>
</tr>
<tr>
<td>7</td>
<td><strong>CP</strong>: Code Protection bit</td>
</tr>
<tr>
<td>6</td>
<td><strong>MCLRE</strong>: MCLR/VPP Pin Function Select bit</td>
</tr>
<tr>
<td>4-3</td>
<td><strong>WDTE&lt;1:0&gt;</strong>: Watchdog Timer Enable bit</td>
</tr>
<tr>
<td>2-0</td>
<td><strong>FOSC&lt;2:0&gt;</strong>: Oscillator Selection bits</td>
</tr>
</tbody>
</table>

**Legend:**
- **W** = Writable bit
- **R** = Readable bit
- **U** = Unimplemented bit, read as '0'
- **P** = Programmable Bit
- **x** = Bit is unknown
- **'0'** = Bit is cleared
- **'1'** = Bit is set
- **Value at POR**

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

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

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

**bit 10-9**
- **BOREN<1:0>**: Brown-out Reset Enable bits
  - 11 = BOR enabled
  - 10 = BOR enabled during operation and disabled in Sleep
  - 01 = BOR controlled by SBOREN bit of the PCON register
  - 00 = BOR disabled

**bit 8**
- **CPD**: Data Code Protection bit
  - 1 = Data memory code protection is disabled
  - 0 = Data memory code protection is enabled

**bit 7**
- **CP**: Code Protection bit
  - 1 = Program memory code protection is disabled
  - 0 = Program memory code protection is enabled

**bit 6**
- **MCLRE**: MCLR/VPP Pin Function Select bit
  - If LVP bit = 1:
    - This bit is ignored.
  - If LVP bit = 0:
    - 1 = MCLR/VPP pin function is MCLR; Weak pull-up enabled.
    - 0 = MCLR/VPP pin function is digital input; MCLR internally disabled; Weak pull-up under control of WPUA register.

**bit 5**
- **PWRTE**: Power-up Timer Enable bit
  - 1 = PWRT disabled
  - 0 = PWRT enabled

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

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

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

<table>
<thead>
<tr>
<th>Bit 13</th>
<th>Bit 12</th>
<th>Bit 11</th>
<th>Bit 10</th>
<th>Bit 9</th>
<th>Bit 8</th>
<th>Bit 7-5</th>
<th>Bit 4</th>
<th>Bit 3-2</th>
<th>Bit 1-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>LVP</td>
<td>DEBUG</td>
<td>—</td>
<td>LVP</td>
<td>STVREN</td>
<td>PLLEN</td>
<td>Reserved(2)</td>
<td>Reserved(2)</td>
<td>Reserved(2)</td>
<td>WRT&lt;1:0&gt;</td>
</tr>
<tr>
<td>1 = Low-voltage programming enabled</td>
<td>1 = In-Circuit Debugger disabled, ICSPCLK and ICSPDAT are general purpose I/O pins</td>
<td>0 = In-Circuit Debugger enabled, ICSPCLK and ICSPDAT are dedicated to the debugger</td>
<td>1 = Brown-out Reset voltage (VBOR), low trip point selected</td>
<td>0 = Stack overflow or underflow will cause a Reset</td>
<td>1 = 4xPLL enabled</td>
<td>x = Bit is unknown</td>
<td>x = Bit is unknown</td>
<td>x = Bit is unknown</td>
<td>x = Bit is unknown</td>
</tr>
</tbody>
</table>

### Legend:
- **W** = Writable bit
- **R** = Readable bit
- **U** = Unimplemented bit, read as ‘0’
- **P** = Programmable Bit
- ‘0’ = Bit is cleared
- ‘1’ = Bit is set
- **x** = Bit is unknown

### Bit Descriptions:
- **bit 13**  
  **LVP**: Low-Voltage Programming Enable bit\(^{(1)}\)  
  1 = Low-voltage programming enabled  
  0 = HV on MCLR/VPP must be used for programming
- **bit 12**  
  **DEBUG**: In-Circuit Debugger Mode bit  
  1 = In-Circuit Debugger disabled, ICSPCLK and ICSPDAT are general purpose I/O pins  
  0 = In-Circuit Debugger enabled, ICSPCLK and ICSPDAT are dedicated to the debugger
- **bit 11**  
  **Unimplemented**: Read as ‘1’
- **bit 10**  
  **BORV**: Brown-out Reset Voltage Selection bit  
  1 = Brown-out Reset voltage (VBOR), low trip point selected  
  0 = Brown-out Reset voltage (VBOR), high trip point selected
- **bit 9**  
  **STVREN**: Stack Overflow/Underflow Reset Enable bit  
  1 = Stack overflow or underflow will cause a Reset  
  0 = Stack overflow or underflow will not cause a Reset
- **bit 8**  
  **PLLEN**: PLL Enable bit  
  1 = 4xPLL enabled  
  0 = 4xPLL disabled
- **bit 7-5**  
  **Unimplemented**: Read as ‘1’
- **bit 4**  
  **Reserved**: Read as ‘1’\(^{(2)}\)
- **bit 3-2**  
  **Unimplemented**: Read as ‘1’
- **bit 1-0**  
  **WRT<1:0>**: Flash Memory Self-Write Protection bits  

#### 2 kW Flash memory (PIC12(L)F1822/PIC16(L)F1823/1826):

- 11 = Write protection off  
- 10 = 000h to 1FFh write-protected, 200h to 7FFh may be modified by EECON control  
- 01 = 000h to 3FFh write-protected, 400h to 7FFh may be modified by EECON control  
- 00 = 000h to 7FFh write-protected, no addresses may be modified by EECON control

#### 4 kW Flash memory (PIC16(L)F1824/1827/1828):

- 11 = Write protection off  
- 10 = 000h to 1FFh write-protected, 200h to FFFh may be modified by EECON control  
- 01 = 000h to 7FFh write-protected, 800h to FFFh may be modified by EECON control  
- 00 = 000h to FFFh write-protected, no addresses may be modified by EECON control

#### 8 kW Flash memory (PIC16(L)F1825/1829):

- 11 = Write protection off  
- 10 = 000h to 1FFh write-protected, 200h to 1FFFh may be modified by EECON control  
- 01 = 000h to FFFh write-protected, 1000h to 1FFFh may be modified by EECON control  
- 00 = 000h to 1FFFh write-protected, no addresses may be modified by EECON control

### Notes:
1. The LVP bit cannot be programmed to ‘0’ when Programming mode is entered via LVP.
2. This bit must be programmed as a ‘1’.
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 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 Configuration Word 1 has MCLR disabled (MCLRE = 0), the power-up time is disabled (PWRT = 0), the internal oscillator is selected (FOSC = 100), and ICSPCLK and ICSPDAT pins 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 (VIH). See Figures 8-3 and 8-4.

4.2 Low-Voltage Programming (LVP) Mode

The Low-Voltage Programming mode allows the PIC12(L)F1822 and PIC16(L)F182X devices to be programmed using VDD only, without high voltage. When the LVP bit of 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 modes requires the following steps:
1. MCLR is brought to VIH.
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 VIH for as long as Program/Verify mode is to be maintained.

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

Exiting Program/Verify mode is done by no longer driving MCLR to VIH. See Figure 8-8 and Figure 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

The PIC12(L)F1822 and PIC16(L)F182X 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 TdLY 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 4-1: COMMAND MAPPING**

<table>
<thead>
<tr>
<th>Command</th>
<th>Mapping Data/Note</th>
<th>Binary (MSb ... LSb)</th>
<th>Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>Load Configuration</td>
<td></td>
<td>x 0 0 0 0 0</td>
<td>00h</td>
</tr>
<tr>
<td>Load Data For Program Memory</td>
<td></td>
<td>x 0 0 0 1 0</td>
<td>02h</td>
</tr>
<tr>
<td>Load Data For Data Memory</td>
<td></td>
<td>x 0 0 0 1 1</td>
<td>03h</td>
</tr>
<tr>
<td>Read Data From Program Memory</td>
<td></td>
<td>x 0 0 1 0 0</td>
<td>04h</td>
</tr>
<tr>
<td>Read Data From Data Memory</td>
<td></td>
<td>x 0 0 1 0 1</td>
<td>05h</td>
</tr>
<tr>
<td>Increment Address</td>
<td></td>
<td>x 0 0 1 1 1</td>
<td>06h</td>
</tr>
<tr>
<td>Reset Address</td>
<td></td>
<td>x 1 0 1 1 0</td>
<td>16h</td>
</tr>
<tr>
<td>Begin Internally Timed Programming</td>
<td></td>
<td>x 0 1 0 0 0</td>
<td>08h</td>
</tr>
<tr>
<td>Begin Externally Timed Programming</td>
<td></td>
<td>x 1 1 0 0 0</td>
<td>18h</td>
</tr>
<tr>
<td>End Externally Timed Programming</td>
<td></td>
<td>x 0 1 0 1 0</td>
<td>0Ah</td>
</tr>
<tr>
<td>Bulk Erase Program Memory</td>
<td></td>
<td>x 0 1 0 0 1</td>
<td>09h</td>
</tr>
<tr>
<td>Bulk Erase Data Memory</td>
<td></td>
<td>x 0 1 0 1 1</td>
<td>0Bh</td>
</tr>
<tr>
<td>Row Erase Program Memory</td>
<td></td>
<td>x 1 0 0 0 1</td>
<td>11h</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.

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.

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).
4.3.3 LOAD DATA FOR DATA MEMORY

The Load Data for Data Memory command will load a 14-bit “data word” when 16 cycles are applied. However, the data memory is only eight bits wide and thus, only the first eight bits of data after the Start bit will be programmed into the data memory. It is still necessary to cycle the clock the full 16 cycles in order to allow the internal circuitry to reset properly (see Figure 4-3).

FIGURE 4-3: LOAD DATA FOR DATA MEMORY COMMAND

![Diagram showing the load data for data memory command]

4.3.4 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 4-4).

FIGURE 4-4: READ DATA FROM PROGRAM MEMORY

![Diagram showing the read data from program memory command]
4.3.5 READ DATA FROM DATA MEMORY
The Read Data from Data Memory command will transmit data bits out of the data memory starting with the second rising edge of the clock input. The ICSPDAT pin will go into Output mode on the second rising edge, and it will revert to Input mode (high-impedance) after the 16th rising edge. The data memory is eight bits wide, and therefore, only the first eight bits that are output are actual data. If the data memory is code-protected, the data is read as all zeros. A timing diagram of this command is shown in Figure 4-5.

4.3.6 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.
4.3.7 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.

FIGURE 4-7: RESET ADDRESS

4.3.8 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, 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. However, the EEPROM memory address that is being programmed is erased prior to being programmed with internally timed programming.

FIGURE 4-8: BEGIN INTERNALLY TIMED PROGRAMMING
4.3.9 BEGIN EXTERNALLY TIMED PROGRAMMING

A Load Configuration, Load Data for Program Memory or Load Data for Data 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. No internal erase is performed for the data EEPROM, therefore, the device should be erased prior to executing this command.

The Begin Externally Timed Programming command cannot be used for programming the Configuration Words (see Figure 4-9).

FIGURE 4-9: BEGIN EXTERNALLY TIMED PROGRAMMING

4.3.10 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 4-10).

FIGURE 4-10: END EXTERNALLY TIMED PROGRAMMING
4.3.11 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
- If CPD = 0, Data Memory is erased

Address 8000h-8008h:
- Program Memory is erased
- Configuration Words are erased
- User ID Locations are erased
- If CPD = 0, Data Memory is erased

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

FIGURE 4-11: BULK ERASE PROGRAM MEMORY

4.3.12 BULK ERASE DATA MEMORY

To perform an erase of the data memory, after a Bulk Erase Data Memory command, wait a minimum of TERAB to complete Bulk Erase.

To erase data memory when data code-protect is active (CPD = 0), the Bulk Erase Program Memory command should be used.

FIGURE 4-12: BULK ERASE DATA MEMORY COMMAND

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

Note: Data memory will not erase if code-protected (CPD = 0).
4.3.13 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.

<table>
<thead>
<tr>
<th>TABLE 4-2: PROGRAMMING ROW SIZE AND LATCHES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Devices</td>
</tr>
<tr>
<td>---------</td>
</tr>
<tr>
<td>PIC16F1826/1827</td>
</tr>
<tr>
<td>PIC12F1822/16F1823</td>
</tr>
<tr>
<td>PIC16F1824/1825</td>
</tr>
<tr>
<td>PIC16F1828/1829</td>
</tr>
</tbody>
</table>

FIGURE 4-13: ROW ERASE PROGRAM MEMORY
5.0 PROGRAMMING ALGORITHMS

The PIC12F1822/16F182X 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 PC’s address at the time the Begin Externally Timed Programming or Begin Internally Timed Programming command is given will determine which location(s) in memory are written. Writes cannot cross the physical boundary. For example, with the PIC16F1827, attempting to write from address 0002h-0009h will result in data being written to 0008h-000Fh.

If more than the maximum number of data 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

Start

Enter Programming Mode

Bulk Erase Device

Write Program Memory(1)

Write User IDs

Write Data Memory(2)

Verify Program Memory

Verify User IDs

Verify Data Memory

Write Configuration Words(2)

Verify Configuration Words

Exit Programming Mode

Done

Note 1: See Figure 5-2.
2: See Figure 5-5.
3: See Figure 5-6.
FIGURE 5-2: PROGRAM MEMORY FLOWCHART

Start

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

Program Cycle\(^3\)

Read Data from Program Memory

Data Correct?

Yes

Increment Address Command

No

All Locations Done?

Yes

Done

No

Report Programming Failure

Note 1: This step is optional if 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 device per Figure 5-8.
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)

Wait TPEXT

End Programming Command

Wait TDIS
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

- Increment Address Command

- Load Data for Program Memory
  - Latch n

- 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?

No

Report Programming Failure

Yes

Increment Address Command

No

Address = 8004h?

Yes

Increment Address Command

Yes

Increment Address Command

Increment Address Command

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

Data Correct?

Report Programming Failure

Done

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

2: See Figure 5-3.
FIGURE 5-6: DATA MEMORY PROGRAM FLOWCHART

Start

Bulk Erase Data Memory

Data Program Cycle(1)

Read Data From Data Memory Command

Data Correct?

Yes

Done

No

Increment Address Command

All Locations Done?

No

Report Programming Failure

Yes

Note 1: See Figure 5-7.
FIGURE 5-7: DATA MEMORY PROGRAM CYCLE

Program Cycle

Load Data for Data Memory

Begin Programming Command (Internally timed)

Wait TINT

Begin Programming Command (Externally timed)

Wait TEXT

End Programming Command

Wait TDIS

FIGURE 5-8: ERASE FLOWCHART

Start

Load Configuration

Bulk Erase Program Memory

Bulk Erase Data 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 all '0'. Further programming is disabled for the program memory (0000h-7FFFh).

Data memory is protected with its own code-protect bit (CPD). When data code-protection is enabled (CPD = 0), all data memory locations read as '0'. Further programming is disabled for the data memory. Data 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.

6.2 Data Memory

Data memory protection is enabled by programming the CPD bit in Configuration Word 1 register to '0'.

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

Note: To ensure system security, if CPD bit = 0, the Bulk Erase Program Memory command will also erase data memory.

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: Configuration Word 1 is stored at 8007h on the PIC12(L)F1822 and PIC16(L)F182X. 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 and Revision

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

7.3 Data EEPROM

The programmer should be able to read data memory information from a hex file and write data memory contents to a hex file.

The physical address range of the 256 byte data memory is 0000h-00FFh. However, these addresses are logically mapped to address 1E000h-1E1FFh in the hex file. This provides a way of differentiating between the data and program memory locations in this range. The format for data memory storage is one data byte per address location, LSb aligned.
7.4 Checksum Computation

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

7.4.1 PROGRAM CODE PROTECTION DISABLED

With the program code protection disabled, the checksum is computed by reading the contents of the PIC12(L)F1822 and PIC16(L)F182X program memory locations and adding up the program memory data starting at address 0000h, up to the maximum user addressable location. Any Carry bit 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’.

**Note:** Data memory does not effect the checksum.

### EXAMPLE 7-1: CHECKSUM COMPUTED WITH PROGRAM CODE PROTECTION DISABLED

**PIC16F1827, BLANK DEVICE**

<table>
<thead>
<tr>
<th>Device</th>
<th>Config. Word 1</th>
<th>Config. Word 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIC16F1826</td>
<td>3FFFh</td>
<td>3713h</td>
</tr>
<tr>
<td>PIC16F1827</td>
<td>3FFFh</td>
<td>3713h</td>
</tr>
<tr>
<td>PIC16LF1826</td>
<td>3FFFh</td>
<td>3073h</td>
</tr>
<tr>
<td>PIC16LF1827</td>
<td>3FFFh</td>
<td>3073h</td>
</tr>
<tr>
<td>PIC12F1822</td>
<td>3FFFh</td>
<td>3713h</td>
</tr>
<tr>
<td>PIC12LF1822</td>
<td>3FFFh</td>
<td>3713h</td>
</tr>
<tr>
<td>PIC16F1823</td>
<td>3FFFh</td>
<td>3713h</td>
</tr>
<tr>
<td>PIC16LF1823</td>
<td>3FFFh</td>
<td>3713h</td>
</tr>
<tr>
<td>PIC16F1824</td>
<td>3FFFh</td>
<td>3713h</td>
</tr>
<tr>
<td>PIC16LF1824</td>
<td>3FFFh</td>
<td>3713h</td>
</tr>
<tr>
<td>PIC16F1825</td>
<td>3FFFh</td>
<td>3713h</td>
</tr>
<tr>
<td>PIC16LF1825</td>
<td>3FFFh</td>
<td>3713h</td>
</tr>
<tr>
<td>PIC16F1826</td>
<td>3FFFh</td>
<td>3713h</td>
</tr>
<tr>
<td>PIC16F1827</td>
<td>3FFFh</td>
<td>3713h</td>
</tr>
<tr>
<td>PIC16LF1827</td>
<td>3FFFh</td>
<td>3713h</td>
</tr>
</tbody>
</table>

**TABLE 7-1: CONFIGURATION WORD MASK VALUES**

**Note:** Data memory does not effect the checksum.

**EXAMPLE 7-2: CHECKSUM COMPUTED WITH PROGRAM CODE PROTECTION DISABLED
PIC16LF1827, 00AAh AT FIRST AND LAST ADDRESS**

<table>
<thead>
<tr>
<th>Device</th>
<th>Config. Word 1</th>
<th>Config. Word 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIC16LF1827</td>
<td>3FFFh</td>
<td>3713h</td>
</tr>
</tbody>
</table>

**PIC16F1827**

Sum of Memory addresses 0000h-0FFFh  F000h
Configuration Word 1  3FFFh
Configuration Word 1 mask  3FFFh
Configuration Word 2  3FFFh
Configuration Word 2 mask  3713h
Checksum  = F000h + (3FFFh and 3FFFh) + (3FFFh and 3713h)
= F000h + 3FFFh + 3713h
= 6712h

**PIC16LF1827**

Sum of Memory addresses 0000h-0FFFh  7156h
Configuration Word 1  3FFFh
Configuration Word 1 mask  3FFFh
Configuration Word 2  3FFFh
Configuration Word 2 mask  3703h
Checksum  = 7156h + (3FFFh and 3FFFh) + (3FFFh and 3703h)
= 7156h + 3FFFh + 3703h
= E858h
7.4.2 PROGRAM CODE PROTECTION ENABLED

With the program code protection enabled, the checksum is computed in the following manner: The Least Significant nibble of each User ID is used to create a 16-bit value. The masked value of User ID location 8000h is the Most Significant nibble. This Sum of User IDs is summed with the Configuration Words (all unimplemented Configuration bits are masked to ‘0’).

Note: Data memory does not effect the checksum.

EXAMPLE 7-3: CHECKSUM COMPUTED WITH PROGRAM CODE PROTECTION ENABLED
PIC16F1827, BLANK DEVICE

<table>
<thead>
<tr>
<th>Configuration Word 1</th>
<th>3F7Fh</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration Word 1 mask</td>
<td>3FFFh</td>
</tr>
<tr>
<td>Configuration Word 2</td>
<td>3FFFh</td>
</tr>
<tr>
<td>Configuration Word 2 mask</td>
<td>3713h</td>
</tr>
<tr>
<td>User ID (8000h)</td>
<td>0006h</td>
</tr>
<tr>
<td>User ID (8001h)</td>
<td>0007h</td>
</tr>
<tr>
<td>User ID (8002h)</td>
<td>0001h</td>
</tr>
<tr>
<td>User ID (8003h)</td>
<td>0002h</td>
</tr>
</tbody>
</table>

Sum of User IDs = (0006h and 000Fh) << 12 + (0007h and 000Fh) << 8 + (0001h and 000Fh) << 4 + (0002h and 000Fh)
= 6000h + 0700h + 0010h + 0002h
= 6712h

Checksum = (3F7Fh and 3FFFh) + (3FFFh and 3713h) + Sum of User IDs
= 3F7Fh + 3713h + 6712h
= DDA4h

EXAMPLE 7-4: CHECKSUM COMPUTED WITH PROGRAM CODE PROTECTION ENABLED
PIC16LF1827, 00AAh AT FIRST AND LAST ADDRESS

<table>
<thead>
<tr>
<th>Configuration Word 1</th>
<th>3F7Fh</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration Word 1 mask</td>
<td>3FFFh</td>
</tr>
<tr>
<td>Configuration Word 2</td>
<td>3FFFh</td>
</tr>
<tr>
<td>Configuration Word 2 mask</td>
<td>3703h</td>
</tr>
<tr>
<td>User ID (8000h)</td>
<td>000Eh</td>
</tr>
<tr>
<td>User ID (8001h)</td>
<td>0008h</td>
</tr>
<tr>
<td>User ID (8002h)</td>
<td>0005h</td>
</tr>
<tr>
<td>User ID (8003h)</td>
<td>0008h</td>
</tr>
</tbody>
</table>

Sum of User IDs = (000Eh and 000Fh) << 12 + (0008h and 000Fh) << 8 + (0005h and 000Fh) << 4 + (0008h and 000Fh)
= E000h + 0800h + 0050h + 0008h
= E858h

Checksum = (3F7Fh and 3FFFh) + (3FFFh and 3703h) + Sum of User IDs
= 3F7Fh + 3703h + E858h
= 5EDAh
### 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 (unless otherwise stated)</th>
<th>Operating Temperature</th>
<th>-40°C ≤ TA ≤ +85°C</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sym.</strong></td>
<td><strong>Characteristics</strong></td>
<td><strong>Min.</strong></td>
<td><strong>Typ.</strong></td>
</tr>
<tr>
<td><strong>VDD</strong></td>
<td>Vdd</td>
<td>PIC12F1822</td>
<td>2.1</td>
</tr>
<tr>
<td></td>
<td>Read/Write and Row Erase operations</td>
<td>PIC16F182X</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>PIC12LF1822</td>
<td>2.1</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>PIC16LF182X</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>Bulk Erase operations</td>
<td>PIC12F1822</td>
<td>2.7</td>
</tr>
<tr>
<td></td>
<td>PIC16F182X</td>
<td>2.7</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>PIC12LF1822</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>PIC16LF182X</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td><strong>IDD</strong></td>
<td>Current on VDD, Idle</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td><strong>IDDP</strong></td>
<td>Current on VDD, Programming</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td><strong>IPP</strong></td>
<td>VPP</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td><strong>VIH</strong></td>
<td>(ICSPCLK, ICSPDAT, MCLR/VPP) input high level</td>
<td>0.8 VDD</td>
<td>—</td>
</tr>
<tr>
<td><strong>VIL</strong></td>
<td>(ICSPCLK, ICSPDAT, MCLR/VPP) input low level</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td><strong>VOH</strong></td>
<td>ICSPDAT output high level</td>
<td>VDD-0.7</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>VDD-0.7</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>VDD-0.7</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td><strong>VOL</strong></td>
<td>ICSPDAT output low level</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>VSS+0.6</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>VSS+0.6</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td><strong>Programming Mode Entry and Exit</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>TENTS</strong></td>
<td>Programing mode entry setup time: ICSPCLK, ICSPDAT setup time before VDD or MCLR↑</td>
<td>100</td>
<td>—</td>
</tr>
<tr>
<td><strong>TENTH</strong></td>
<td>Programing mode entry hold time: ICSPCLK, ICSPDAT hold time after VDD or MCLR↑</td>
<td>250</td>
<td>—</td>
</tr>
<tr>
<td><strong>Serial Program/Verify</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>TCKL</strong></td>
<td>Clock Low Pulse Width</td>
<td>100</td>
<td>—</td>
</tr>
<tr>
<td><strong>TCKH</strong></td>
<td>Clock High Pulse Width</td>
<td>100</td>
<td>—</td>
</tr>
<tr>
<td><strong>TDS</strong></td>
<td>Data in setup time before clock↑</td>
<td>100</td>
<td>—</td>
</tr>
<tr>
<td><strong>TDH</strong></td>
<td>Data in hold time after clock↓</td>
<td>100</td>
<td>—</td>
</tr>
<tr>
<td><strong>TCO</strong></td>
<td>Clock↑ to data out valid (during a Read Data command)</td>
<td>0</td>
<td>—</td>
</tr>
<tr>
<td><strong>TLZD</strong></td>
<td>Clock↑ to data low-impedance (during a Read Data command)</td>
<td>0</td>
<td>—</td>
</tr>
<tr>
<td><strong>THZD</strong></td>
<td>Clock↑ to data high-impedance (during a Read Data command)</td>
<td>0</td>
<td>—</td>
</tr>
<tr>
<td><strong>TDLY</strong></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>
</tr>
<tr>
<td><strong>TERAB</strong></td>
<td>Bulk Erase cycle time</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td><strong>TERAR</strong></td>
<td>Row Erase cycle time</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>
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

---

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

<table>
<thead>
<tr>
<th>AC/DC CHARACTERISTICS</th>
<th>Standard Operating Conditions (unless otherwise stated)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Operating Temperature (-40°C \leq TA \leq +85°C)</td>
</tr>
<tr>
<td>Sym.</td>
<td>Characteristics</td>
</tr>
<tr>
<td>VPP</td>
<td>TENTH</td>
</tr>
<tr>
<td>VDD</td>
<td>TENTS</td>
</tr>
<tr>
<td>ICSPDAT</td>
<td>VIHH</td>
</tr>
<tr>
<td>ICSPCLK</td>
<td>VIL</td>
</tr>
<tr>
<td>TPEXT</td>
<td>Externally timed programming pulse</td>
</tr>
<tr>
<td>TDIS</td>
<td>Time delay from program to compare (HV discharge time)</td>
</tr>
<tr>
<td>TEXIT</td>
<td>Time delay when exiting Program/Verify mode</td>
</tr>
</tbody>
</table>

---

**TABLE 8-1:**

- **TPINT**: Internally timed programming operation time
- **TPEXT**: Externally timed programming pulse
- **TDIS**: Time delay from program to compare (HV discharge time)
- **TEXIT**: Time delay when exiting Program/Verify mode

---

**Sym.** is the symbol for each characteristic in the table.

**Characteristics** describe the timing requirements for programming and verifying in various modes.

**Min.**, **Typ.**, and **Max.** represent the minimum, typical, and maximum values of the timing requirements, respectively.

**Units** specify the unit of measurement for each time requirement.

**Conditions/Comments** provide additional details about the conditions under which these timing requirements are applicable.

---

**TABLE 8-1** illustrates the AC/DC characteristics and their timing requirements for the PIC12(L)F1822/PIC16(L)F182X family of microcontrollers, specifically focusing on programming and verifying modes.
FIGURE 8-5: CLOCK AND DATA TIMING

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

Revision A (06/2009)
Original release of this document.

Revision B (10/2009)
Added PIC12F/LF1822 and PIC16F/LF1823 devices.

Revision C (03/2010)
Added PIC12(L)F1824, PIC16(L)F1825, PIC16(L)F1828 and PIC16(L)F1829 devices; Added Figure 2-8, Figure 2-9 and Figure 3-3.

Revision D (09/2012)
Updated document to new format; Updated Figure 3-3, Register 3-3 and Table 8-1; 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. Microchip makes no representations or warranties of any kind, whether express or implied, written or oral, statutory or otherwise, related to the information, including but not limited to its condition, quality, performance, merchantability or fitness for purpose. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer’s risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

Trademarks
The Microchip name and logo, the Microchip logo, dsPIC, FlashFlex, KEELoo, KEELoo logo, MPLAB, PIC, PICmicro, PICSTART, PIC32 logo, rFIC, 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, MFP, MPLAB Certified logo, MPLIB, MPLINK, mTouch, Omniscent 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.

© 2010-2012, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

Printed on recycled paper.

ISBN: 9781620765968

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, KeelOo® 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.
## Worldwide Sales and Service

<table>
<thead>
<tr>
<th>AMERICAS</th>
<th>ASIA/PACIFIC</th>
<th>ASIA/PACIFIC</th>
<th>EUROPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Corporate Office</td>
<td>Asia Pacific Office</td>
<td>India - Bangalore</td>
<td>Austria - Wels</td>
</tr>
<tr>
<td>2355 West Chandler Blvd.</td>
<td>Suites 3707-14, 37th Floor</td>
<td>Tel: 91-80-3090-4444</td>
<td>Tel: 43-7242-2244-39</td>
</tr>
<tr>
<td>Chandler, AZ 85224-6199</td>
<td>Tower 6, The Gateway</td>
<td>Fax: 91-80-3090-4123</td>
<td>Fax: 43-7242-2244-393</td>
</tr>
<tr>
<td>Tel: 480-792-7200</td>
<td>Harbour City, Kowloon</td>
<td>India - New Delhi</td>
<td>Denmark - Copenhagen</td>
</tr>
<tr>
<td>Fax: 480-792-7277</td>
<td>Hong Kong</td>
<td>Tel: 91-11-4160-8631</td>
<td>Tel: 45-4450-2828</td>
</tr>
<tr>
<td>Technical Support</td>
<td>Tel: 852-2401-1200</td>
<td>Fax: 91-11-4160-8632</td>
<td>Fax: 45-4485-2829</td>
</tr>
<tr>
<td><a href="http://www.microchip.com/support">http://www.microchip.com/support</a></td>
<td>Fax: 852-2401-3431</td>
<td>India - Pune</td>
<td>France - Paris</td>
</tr>
<tr>
<td>Web Address:</td>
<td></td>
<td>Tel: 91-20-2566-1512</td>
<td>Tel: 33-1-69-53-63-20</td>
</tr>
<tr>
<td><a href="http://www.microchip.com">www.microchip.com</a></td>
<td></td>
<td>Fax: 91-20-2566-1513</td>
<td>Fax: 33-1-69-30-90-79</td>
</tr>
<tr>
<td>Atlanta</td>
<td>China - Beijing</td>
<td>Japan - Osaka</td>
<td>Germany - Munich</td>
</tr>
<tr>
<td>Duluth, GA</td>
<td>Tel: 86-10-8569-7000</td>
<td>Tel: 81-66-152-7160</td>
<td>Tel: 49-89-627-144-0</td>
</tr>
<tr>
<td>Tel: 678-957-9614</td>
<td>Fax: 86-10-8528-2104</td>
<td>Fax: 81-66-152-9310</td>
<td>Fax: 49-89-627-144-44</td>
</tr>
<tr>
<td>Fax: 678-957-1455</td>
<td>China - Chengdu</td>
<td>Japan - Yokohama</td>
<td>Italy - Milan</td>
</tr>
<tr>
<td></td>
<td>Tel: 86-28-8665-5511</td>
<td>Tel: 81-45-471-6166</td>
<td>Tel: 39-0331-742611</td>
</tr>
<tr>
<td></td>
<td>Fax: 86-28-8665-7889</td>
<td>Fax: 81-45-471-6122</td>
<td>Fax: 39-0331-466781</td>
</tr>
<tr>
<td></td>
<td>China - Chongqing</td>
<td>Korea - Daegu</td>
<td>Netherlands - Drunen</td>
</tr>
<tr>
<td></td>
<td>Tel: 86-23-8980-9588</td>
<td>Tel: 82-53-744-4301</td>
<td>Tel: 31-416-690399</td>
</tr>
<tr>
<td></td>
<td>Fax: 86-23-8980-9500</td>
<td>Fax: 82-53-744-4302</td>
<td>Fax: 31-416-690340</td>
</tr>
<tr>
<td></td>
<td>China - Hangzhou</td>
<td>Korea - Seoul</td>
<td>Spain - Madrid</td>
</tr>
<tr>
<td></td>
<td>Tel: 86-571-2819-3187</td>
<td>Tel: 82-2-554-7200</td>
<td>Tel: 34-91-708-08-90</td>
</tr>
<tr>
<td></td>
<td>Fax: 86-571-2819-3189</td>
<td>Fax: 82-2-558-5932 or</td>
<td>Fax: 34-91-708-08-91</td>
</tr>
<tr>
<td></td>
<td>China - Hong Kong SAR</td>
<td>82-2-558-5934</td>
<td>UK - Wokingham</td>
</tr>
<tr>
<td></td>
<td>Tel: 852-2401-1200</td>
<td></td>
<td>Tel: 44-118-921-5869</td>
</tr>
<tr>
<td></td>
<td>Fax: 852-2401-3431</td>
<td></td>
<td>Fax: 44-118-921-5820</td>
</tr>
<tr>
<td></td>
<td>China - Nanjing</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tel: 86-25-8473-2460</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fax: 86-25-8473-2470</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>China - Qingdao</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tel: 86-532-8502-7355</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fax: 86-532-8502-7205</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>China - Shanghai</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tel: 86-21-5407-5533</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fax: 86-21-5407-5066</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>China - Shenyang</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tel: 86-24-2334-2829</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fax: 86-24-2334-2393</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>China - Shenzhen</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tel: 86-755-8203-2660</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fax: 86-755-8203-1760</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>China - Wuhan</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tel: 86-27-5980-5300</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fax: 86-27-5980-5118</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>China - Xian</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tel: 86-29-8833-7252</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fax: 86-29-8833-7256</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>China - Xiamen</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tel: 86-592-2388138</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fax: 86-592-2388130</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>China - Zhuhai</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tel: 86-756-3210040</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fax: 86-756-3210049</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>