1.0 OVERVIEW

This programming specification describes an SPI-compatible programming method for the PIC16(L)F1919X family of microcontrollers. Section 3.0 “Programming Algorithms” describes the programming commands, programming algorithms and electrical specifications which are used in that particular programming method. Appendix B contains individual part numbers, device identification and checksum values, pinout and packaging information, and Configuration Words.

Note: To enter LVP mode, the MSb of the Most Significant nibble must be shifted in first. This differs from entering the key sequence on some other device families.

1.1 Programming Data Flow

Nonvolatile Memory (NVM) programming data can be supplied by either the high-voltage In-Circuit Serial Programming™ (ICSP™) interface or the low-voltage In-Circuit Serial Programming (ICSP) interface. Data can be programmed into the Program Flash Memory (PFM), (EEPROM, if available), dedicated "User ID" locations and the Configuration Words.

1.2 Write and/or Erase Selection

Erasing or writing is selected according to the command used to begin operation (see Table 3-1). The terminologies used in this document related to erasing/writing to the Program Flash Memory are defined in Table 1-1 and are detailed below.

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Programmed Cell</td>
<td>A memory cell with a logic ‘0’</td>
</tr>
<tr>
<td>Erased Cell</td>
<td>A memory cell with a logic ‘1’</td>
</tr>
<tr>
<td>Erase</td>
<td>Change memory cell from a ‘0’ to a ‘1’</td>
</tr>
<tr>
<td>Write</td>
<td>Change memory cell from a ‘1’ to a ‘0’</td>
</tr>
<tr>
<td>Program</td>
<td>Generic erase and/or write</td>
</tr>
</tbody>
</table>

1.2.1 ERASING MEMORY

Program Flash Memory is erased by row or in bulk, where ‘bulk’ includes many subsets of the total memory space. The duration of the erase is always determined internally. Here, ‘row’ refers to the minimum erasable size and ‘bulk’ is one of the many possible subsets of all memory rows. All Bulk ICSP Erase commands have minimum VDD requirements, which are higher than the Row Erase and write requirements. Refer to Section 3.7 “Electrical Specifications”.

1.2.2 WRITING MEMORY

Program Flash Memory is written one row at a time. Multiple load data for NVM commands are used to fill the row data latches. The duration of the write is determined either internally or externally. Refer to Section 3.7 “Electrical Specifications”.

1.2.3 MULTI-WORD PROGRAMMING INTERFACE

Program Flash Memory (PFM) panels include a 64-word (one row) programming interface. The row to be programmed must first be erased either with a Bulk Erase or a Row Erase. Refer to Section 3.7 “Electrical Specifications”.

Note: To enter LVP mode, the MSb of the Most Significant nibble must be shifted in first. This differs from entering the key sequence on some other device families.

TABLE 1-1: PROGRAMMING TERMS

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Programmed</td>
<td>A memory cell with a logic ‘0’</td>
</tr>
<tr>
<td>Erased</td>
<td>A memory cell with a logic ‘1’</td>
</tr>
<tr>
<td>Erase</td>
<td>Change memory cell from a ‘0’ to a ‘1’</td>
</tr>
<tr>
<td>Write</td>
<td>Change memory cell from a ‘1’ to a ‘0’</td>
</tr>
<tr>
<td>Program</td>
<td>Generic erase and/or write</td>
</tr>
</tbody>
</table>
1.3 Hardware Requirements

1.3.1 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.

1.3.2 LOW-VOLTAGE ICSP PROGRAMMING

In Low-Voltage ICSP mode, the device 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.

1.3.2.1 Single-Supply ICSP Programming

The LVP Configuration bit 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 VIH. 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.

**Note 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.

1.4 Pin Utilization

Five pins are needed for ICSP programming. The pins are listed in Table 1-2. Refer to Table B-2 for pin locations and packaging information.

<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>I(1)</td>
<td>Program Mode Select</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 the MCLR input. Since the MCLR is used for a level source, MCLR does not draw any significant current.
## 2.0 MEMORY MAP

### FIGURE 2-1: PROGRAM MEMORY MAPPING

<table>
<thead>
<tr>
<th>PIC16(L)F19195</th>
<th>PIC16(L)F19196</th>
<th>PIC16(L)F19197</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC&lt;15:0&gt;</td>
<td>PC&lt;15:0&gt;</td>
<td>PC&lt;15:0&gt;</td>
</tr>
</tbody>
</table>

Note 1: Stack (16 levels)

<table>
<thead>
<tr>
<th>0000h</th>
<th>0000h</th>
<th>0000h</th>
</tr>
</thead>
<tbody>
<tr>
<td>07FFh</td>
<td>07FFh</td>
<td>07FFh</td>
</tr>
<tr>
<td>0800h</td>
<td>0800h</td>
<td>0800h</td>
</tr>
<tr>
<td>0FFFh</td>
<td>0FFFh</td>
<td>0FFFh</td>
</tr>
<tr>
<td>1000h</td>
<td>1000h</td>
<td>1000h</td>
</tr>
<tr>
<td>1FFFh</td>
<td>1FFFh</td>
<td>1FFFh</td>
</tr>
<tr>
<td>2000h</td>
<td>2000h</td>
<td>2000h</td>
</tr>
<tr>
<td>3FFFh</td>
<td>3FFFh</td>
<td>3FFFh</td>
</tr>
<tr>
<td>4000h</td>
<td>4000h</td>
<td>4000h</td>
</tr>
<tr>
<td>7FFFh</td>
<td>7FFFh</td>
<td>7FFFh</td>
</tr>
<tr>
<td>8000h</td>
<td>8000h</td>
<td>8000h</td>
</tr>
<tr>
<td>8003h</td>
<td>8003h</td>
<td>8003h</td>
</tr>
<tr>
<td>8004h</td>
<td>8004h</td>
<td>8004h</td>
</tr>
<tr>
<td>8005h</td>
<td>8005h</td>
<td>8005h</td>
</tr>
<tr>
<td>8006h</td>
<td>8006h</td>
<td>8006h</td>
</tr>
<tr>
<td>8007h</td>
<td>8007h</td>
<td>8007h</td>
</tr>
<tr>
<td>8008h</td>
<td>8008h</td>
<td>8008h</td>
</tr>
<tr>
<td>8009h</td>
<td>8009h</td>
<td>8009h</td>
</tr>
<tr>
<td>800Ah</td>
<td>800Ah</td>
<td>800Ah</td>
</tr>
<tr>
<td>800Bh</td>
<td>800Bh</td>
<td>800Bh</td>
</tr>
<tr>
<td>800Ch</td>
<td>800Ch</td>
<td>800Ch</td>
</tr>
<tr>
<td>80FFh</td>
<td>80FFh</td>
<td>80FFh</td>
</tr>
<tr>
<td>8100h</td>
<td>8100h</td>
<td>8100h</td>
</tr>
<tr>
<td>811Fh</td>
<td>811Fh</td>
<td>811Fh</td>
</tr>
<tr>
<td>8120h</td>
<td>8120h</td>
<td>8120h</td>
</tr>
<tr>
<td>81FFh</td>
<td>81FFh</td>
<td>81FFh</td>
</tr>
<tr>
<td>8200h</td>
<td>8200h</td>
<td>8200h</td>
</tr>
<tr>
<td>821Fh</td>
<td>821Fh</td>
<td>821Fh</td>
</tr>
<tr>
<td>8220h</td>
<td>8220h</td>
<td>8220h</td>
</tr>
<tr>
<td>F000h</td>
<td>F000h</td>
<td>F000h</td>
</tr>
<tr>
<td>F0FFh</td>
<td>F0FFh</td>
<td>F0FFh</td>
</tr>
<tr>
<td>FFFFh</td>
<td>FFFFh</td>
<td>FFFFh</td>
</tr>
</tbody>
</table>

Note 1: The stack is a separate SRAM panel, apart from all user memory panels.

2: Not code-protected.

3: Device Configuration Information, Device/Revision IDs are hard-coded in silicon.

4: The addresses do not roll over. The region is read as ‘0’.

5: For the purposes of instruction fetching during program execution, only 15 bits (PC<14:0>) are used. However, for the purposes of nonvolatile memory reading and writing through ICSP™ programming operations, the PC uses all 16 bits (PC<15:0>), and the “Load PC Address” command requires a full 16-bit data payload.
2.1 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.

2.2 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 2-1: DEVICEID: DEVICE ID REGISTER

<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>
<th>R</th>
<th>R</th>
<th>R</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>DEV&lt;11:0&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Legend:

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

bit 13-12 Read-Only bits
These bits are fixed with value ‘11’ for all devices included in this programming specification.

bit 11-0 **DEV<11:0>:** Device ID bits

**Note:** Refer to Table B-1 for a list of Device ID register values for the devices covered by this programming specification document.

REGISTER 2-2: REVISIONID: REVISION ID REGISTER

<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>
<th>R</th>
<th>R</th>
<th>R</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>MJRREV&lt;5:0&gt;</td>
<td>MNRREV&lt;5:0&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Legend:

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

bit 13-12 **Fixed Value:** Read-Only bits
These bits are fixed with value ‘10’ for all devices included in this programming specification.

bit 11-6 **MJRREV<5:0>:** Major Revision ID bits
These bits are used to identify a major revision. Major and minor revisions are assigned by Microchip.

bit 5-0 **MNRREV<5:0>:** Minor Revision ID bits
These bits are used to identify a minor revision.
2.3 Configuration Words

The devices have several Configuration Words starting at address 8007h. The individual bits within these Configuration Words are critical to the correct operation of the system. Configuration bits enable or disable specific features, placing these controls outside the normal software process, and they establish configured values prior to the execution of any software.

In terms of programming, these important Configuration bits should be considered:

1. **LVP: Low-Voltage Programming Enable bit**
   - 1 = ON – Low-Voltage Programming is enabled. MCLR/VPP pin function is MCLR. MCLRE Configuration bit is ignored.
   - 0 = OFF – HV on MCLR/VPP must be used for programming.

   It is important to note that the LVP bit cannot be written (to 0) while operating from the LVP programming interface. The purpose of this rule is to prevent the user from dropping out of LVP mode while programming from LVP mode, or accidentally eliminating LVP mode from the configuration state. For more information, see Section 3.1.2 “Low-Voltage Programming (LVP) Mode”.

2. **CP: User NVM Program Memory Code Protection bit**
   - 1 = OFF – User NVM code protection disabled
   - 0 = ON – User NVM code protection enabled

   For more information on code protection, see Section 3.3 “Code Protection”.

2.4 Device Information Area

The Device Information Area (DIA) is a dedicated region in the Program Flash Memory. The data is mapped from 8100h to 811Fh. These locations are read-only and cannot be erased or modified. The DIA holds the calibration data for the temperature indicator module and the FVR voltages, which are useful for temperature sensing applications and calibration.

2.5 Device Configuration Information

The Device Configuration Information (DCI) is a dedicated region in the Program Flash Memory mapped from 8200h to 821Fh. The data stored in the DCI memory is hard-coded into the device during manufacturing. Refer to Table D-1 in Appendix D: “Device Configuration Information (DCI)” for the complete DCI table address and description. The DCI holds information about the device which is useful for programming and bootloaders. These locations are read-only and cannot be erased or modified. For more information, refer to the product-specific data sheet.
3.0 PROGRAMMING ALGORITHMS

3.1 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 MSb 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. On entering the Program/Verify mode, the address is cleared.

3.1.1 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

3.1.1.1 VPP – First Entry Mode

To enter Program/Verify mode via the VPP-First mode, 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 V DD 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 MCLRE disabled (MCLRE = 0), the Power-up Timer is disabled (PWRT = 0), the internal oscillator is selected (Fosc = 100), and ICSPDAT and ICSPCLK are driven by the user application, the device will execute code and may drive the ICSPDAT and ISCPCLK I/O pins. Since code execution may prevent first entry, VPP-First Entry mode is strongly recommended, as it prevents user code from changing EEPROM contents or driving pins to affect Test mode entry. See the timing diagram in Figure 3-2.

3.1.1.2 VDD – First Entry Mode

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

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

The VDD-First mode is useful when programming the device when VDD is already applied, for it is not necessary to disconnect VDD to enter Program/Verify mode. During this cycle, any executing code will be interrupted and halted. See the timing diagram in Figure 3-1.
3.1.1.3 Program/Verify Mode Exit

To exit Program/Verify mode, lower MCLR from \( V_{IH} \) to \( V_{IL} \). VDD-First Entry mode should use VDD-Last Exit mode (see Figure 3-1). VPP-First Entry mode should use VPP-Last Exit mode (see Figure 3-2).

**FIGURE 3-1: PROGRAMMING ENTRY AND EXIT MODES – VPP-FIRST AND LAST**

**FIGURE 3-2: PROGRAMMING ENTRY AND EXIT MODES – VDD-FIRST AND LAST**
3.1.2 LOW-VOLTAGE PROGRAMMING (LVP) MODE

The Low-Voltage Programming mode allows the devices to be programmed using VDD only, without high voltage. When the LVP bit of the Configuration Word 4 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. The LSb of the pattern is a “don’t care x”. The Program/Verify mode entry pattern detect hardware verifies only the first 31 bits of the sequence and the last clock is required to activate the Program/Verify mode.

The key sequence is a specific 32-bit pattern, ‘32’h4d434850’ (more easily remembered as MCHP in ASCII). The device will enter Program/Verify mode only if the sequence is valid. The Most Significant bit of the Most 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 Figure 3-3 and Figure 3-4.

FIGURE 3-3: LVP ENTRY (POWERING-UP)

FIGURE 3-4: LVP ENTRY (POWERED)

Exiting Program/Verify mode is done by raising MCLR from below VIL to VIH level (or higher, up to VDD).

Note: To enter LVP mode, the MSb of the Most Significant nibble must be shifted in first. This differs from entering the key sequence on some other device families.
3.1.3 PROGRAM/VERIFY COMMANDS

Once a device has entered ICSP Program/Verify mode (using either high voltage or LVP entry), the programming host device may issue commands to the microcontroller, each eight bits in length. The commands are summarized in Table 3-1. The commands are used to erase and program the device. The commands load and use the Program Counter (PC).

Some of the 8-bit commands also have a data payload associated with it (such as Load Data for NVM and Read Data from NVM).

If the programming host device issues an 8-bit command byte that has a data payload associated with it, the host device is responsible for sending an additional 24 clock pulses (for example, three 8-bit bytes), in order to send or receive the payload data associated with the command.

The actual payload bits associated with a command are command-specific and will be fewer than 24 bits. However, the payload field is always padded with additional Start, Stop and Pad bits, to bring the total payload field size to 24 bits, so as to be compatible with many 8-bit SPI-based systems.

Within a 24-bit payload field, the first bit transmitted is always a Start bit, followed by a variable number of Pad bits, followed by the useful data payload bits and ending with one Stop bit. The useful data payload bits are always transmitted Most Significant bit (MSb) first.

When the programming device issues a command that involves a host to microcontroller payload (for example, Load PC Address), the Start, Stop and Pad bits should all be driven by the programmer to ‘0’. When the programming host device issues a command that involves microcontroller to host payload data (for example, Read Data from NVM), the Start, Stop and Pad bits should be treated as “don’t care” bits and the values should be ignored by the host.

When the programming host device issues an 8-bit command byte to the microcontroller, the host should wait a minimum amount of delay (see Table 3-1) prior to sending any additional clock pulses (associated with either a 24-bit data payload field or the next command byte).

### TABLE 3-1: ICSP™ COMMAND SET SUMMARY

<table>
<thead>
<tr>
<th>Command Name</th>
<th>Command Value</th>
<th>Payload</th>
<th>Delay after Command</th>
<th>Data/Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Load PC Address</td>
<td>1000 0000</td>
<td>80</td>
<td>TDLY</td>
<td>PC = payload value</td>
</tr>
<tr>
<td>Bulk Erase Program Memory</td>
<td>0001 1000</td>
<td>18</td>
<td>No</td>
<td>TERAB Depending on the current value of the PC, one or more memory regions.</td>
</tr>
<tr>
<td>Row Erase Program Memory</td>
<td>1111 0000</td>
<td>F0</td>
<td>No</td>
<td>TERAR The row addressed by the MSbs of the PC is erased; LSbs are ignored.</td>
</tr>
<tr>
<td>Load Data for NVM</td>
<td>0000 00J0</td>
<td>00/02</td>
<td>TDLY</td>
<td>J = 1: PC = PC + 1 after writing J = 0: PC is unchanged</td>
</tr>
<tr>
<td>Read Data from NVM</td>
<td>1111 11J0</td>
<td>FE/FC</td>
<td>Yes</td>
<td>TDLY J = 1: PC = PC + 1 after reading J = 0: PC is unchanged</td>
</tr>
<tr>
<td>Increment Address</td>
<td>1111 1000</td>
<td>F8</td>
<td>No</td>
<td>TDLY PC = PC + 1</td>
</tr>
<tr>
<td>Begin Internally Timed</td>
<td>1110 0000</td>
<td>E0</td>
<td>TPINT</td>
<td>Commits latched data to NVM (self timed)</td>
</tr>
<tr>
<td>Programming</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Begin Externally Timed</td>
<td>1100 0000</td>
<td>C0</td>
<td>TPEXT</td>
<td>Commits latched data to NVM (externally timed). After TPEXT, “End Externally Timed Programming” command must be issued.</td>
</tr>
<tr>
<td>Programming</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>End Externally Timed</td>
<td>1000 0010</td>
<td>82</td>
<td>TDIS</td>
<td>Should be issued within required time delay (TPEXT) after “Begin Externally Timed Programming” command.</td>
</tr>
</tbody>
</table>
Note: All clock pulses for both the 8-bit commands and the 24-bit payload fields are generated by the host programming device. The microcontroller does not drive the ICSPCLK line. The ICSPDAT signal is a bidirectional data line. For all commands and payload fields, except the Read Data from NVM payload, the host programming device continuously drives the ICSPDAT line. Both the host programmer device and the microcontroller should latch received ICSPDAT values on the falling edge of the ICSPCLK line. ISCPDAT timing will be met as per Figure 3-5.
3.1.3.1 Load Data for NVM

The Load Data for NVM command is used to load one programming data latch (for example, one 14-bit instruction word for program memory/configuration memory/User ID memory, or one 8-bit byte for an EEPROM data memory address). The Load Data for NVM command can be used to load data for Program Flash Memory (PFM) (see Figure 3-6) or the EEPROM, if available (see Figure 3-7). The word writes into program memory after the Begin Internally Timed Programming or Begin Externally Timed Programming commands write the entire row of data latches, not just one word. The lower five bits of the address are considered, while the other bits are ignored. Depending on the value of bit 1 of the command, the Program Counter (PC) may or may not be incremented (see Table 3-1). Refer to Section 3.1.3.9 “Row Erase Memory”.

FIGURE 3-6: LOAD DATA FOR NVM (PFM)

FIGURE 3-7: LOAD DATA FOR NVM (EEPROM, IF AVAILABLE)
3.1.3.2 Read Data from NVM

The Read Data from NVM command will transmit data bits out of the current PC address. The ICSPDAT pin will go into Output mode on the first falling edge of ICSPCLK, and it will revert to Input mode (high-impedance) after the 24th falling edge of the clock. The Start and Stop bits are only one half of a bit time wide, and should, therefore, be ignored by the host programmer device (since the latched value may be indeterminate). Additionally, the host programmer device should only consider the MSb to LSb payload bits as valid, and should ignore the values of the pad bits. If the program memory is code-protected (\(CP = 0\)), the data will be read as zeros (see Figure 3-10 and Figure 3-11). Depending on the value of bit ‘1’ of the command, the PC may or may not be incremented (see Table 3-1). The Read Data for NVM command can be used to read data for Program Flash Memory (PFM) (see Figure 3-10) or the EEPROM (see Figure 3-11).

3.1.3.3 Load PC Address

The PC value is set using the supplied data. The address implies the memory panel (PFM or EEPROM) to be accessed (see Figure 3-8).

![Figure 3-8: Load PC Address](image)

3.1.3.4 Increment Address

The PC is incremented by one when this command is received. It is not possible to decrement the address. To reset this counter, the user must use the Load PC Address command. This command performs the same action as the J bit in the Load/Read commands. See Figure 3-9.

![Figure 3-9: Increment Address](image)
3.1.3.5 Begin Internally Timed Programming

The write programming latches must already have been loaded using the Load Data for NVM command, prior to issuing the Begin Programming command. Programming of the addressed memory row will begin after this command is received. The lower LSBs of the address are ignored. An internal timing mechanism executes the write. The user must allow for the Erase/Write cycle time, TPINT, in order for the programming to complete, prior to issuing the next command (see Figure 3-12).

After the programming cycle is complete all the data latches are reset to ‘1’.
3.1.3.6 Begin Externally Timed Programming

Data to be programmed must be previously loaded by Load Data for NVM command before every Begin Programming command. To complete the programming, the End Externally Timed Programming command must be sent in the specified time window defined by TPEXT (see Figure 3-13). The lower LSBs of the address are ignored.

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

**FIGURE 3-13: BEGIN EXTERNALLY TIMED PROGRAMMING**

3.1.3.7 End Externally Timed Programming

This command is required to terminate the programming sequence after a Begin Externally Timed Programming command is given. If no programming command is in progress or if the programming cycle is internally timed, this command will execute as a No Operation (NOP) (Figure 3-14).

**FIGURE 3-14: END PROGRAM TIMING**
3.1.3.8 Bulk Erase Memory

The Bulk Erase Memory command performs different functions dependent on the current PC address. The Bulk Erase command affects specific portions of the memory depending on the initial value of the Program Counter. Whenever a Bulk Erase command is executed, the device will erase all bytes within the regions listed in Table 3-2. While a programming command is in progress, this command executes as a NOP.

After receiving the Bulk Erase Memory command, the erase will not complete until the time interval, TERAB, has expired (see Figure 3-15). The programming host device should not issue another 8-bit command until after the TERAB interval has fully elapsed.

### TABLE 3-2: BULK ERASE TABLE

<table>
<thead>
<tr>
<th>Address</th>
<th>Area(s) Erased</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CP = 1</td>
</tr>
<tr>
<td>0000h-7FFFh{(1)}</td>
<td>User Flash</td>
</tr>
<tr>
<td></td>
<td>Configuration Words</td>
</tr>
<tr>
<td>8000h-80FDh</td>
<td>User Flash</td>
</tr>
<tr>
<td></td>
<td>Configuration Words</td>
</tr>
<tr>
<td></td>
<td>User ID words</td>
</tr>
<tr>
<td>80FEh-80FFh</td>
<td>User Flash</td>
</tr>
<tr>
<td>8100h-E7FFh</td>
<td>No Operation</td>
</tr>
<tr>
<td>E800h-FFFFh</td>
<td>User Flash</td>
</tr>
<tr>
<td></td>
<td>Configuration Words</td>
</tr>
<tr>
<td></td>
<td>User ID words</td>
</tr>
</tbody>
</table>

**Note 1:** See Figure 2-1 for device specific Program Memory Size and locations.

### FIGURE 3-15: BULK ERASE MEMORY
3.1.3.9 Row Erase Memory

If the program memory is code-protected, the Row Erase Program Memory command will be ignored. When the address is 8000h-8004h, the Row Erase Program Memory command will only erase the User ID locations regardless of the setting of the CP Configuration bit. The Row Erase Memory command will erase an individual row. When write and erase operations are done on a row basis, the row size (number of 14-bit words) for erase operation is 64 and the row size (number of 14-bit latches) for the write operation is 64.

The Flash memory row defined by the current PC will be erased. The user must wait TERAR for erasing to complete (see Figure 3-16).

FIGURE 3-16: ROW ERASE MEMORY

3.2 Programming Algorithms

The device uses internal latches to temporarily store the 14-bit words used for programming. The data latches allow the user to program a full row with a single Begin Internally Timed Programming or Begin Externally Timed Programming command. The Load Data for NVM command is used to load a single data latch. The data latch will hold the data until the Begin Internally Timed Programming or Begin Externally Timed Programming command is given.

The data latches are aligned with the LSbs of the address. The address at the time the Begin Internally Timed Programming or Begin Externally 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 64-latch device will result in data being written to 0020h-003Fh.

If more than the maximum number of latches are written without a Begin Internally Timed Programming or Begin Externally Timed Programming command, the data in the data latches will be overwritten. Figure 3-17 through Figure 3-22 show the recommended flowcharts for programming.

Note: The Program Flash Memory region are programmed one row (64 words) at a time (Figure 3-20). User ID and Configuration Words are programmed one word at a time (Figure 3-19). While the EEPROM or Data Flash Memory is programmed one byte at a time. The value of the PC at the time of issuing the Begin Internally Timed Programming or Begin Externally Timed Programming command determines what row (of Program Flash Memory or EEPROM) or what word (of User ID or Configuration Word) will get programmed.
Figure 3-17: Device Program/Verify Flowchart

- Start
- Enter Programming Mode
- Bulk Erase Device
- Write Program Memory (1)
- Verify Program Memory
- Write EEPROM (3)
- Verify EEPROM Memory (3)
- Write User IDs
- Verify User IDs
- Write Configuration Words (2)
- Verify Configuration Words
- Exit Programming Mode
- Done

Note 1: See Figure 3-8.
Note 2: See Figure 3-16.
Note 3: The PIC16(L)F1919X family does not include EEPROM.
FIGURE 3-18: PROGRAM MEMORY FLOWCHART

Start

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

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

Read Data from NVM

Data Correct?

No

Report Programming Failure

Yes

Increment PC Address to Next Row

All Locations Done?

No

Yes

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 3-17.

3: See Figure 3-15.

FIGURE 3-19: ONE-WORD PROGRAM CYCLE

Program Cycle
(for Programming User ID and Configuration Words)

Load Data for NVM Command

Begin Programming Command (Internally Timed)

Wait TPINT
FIGURE 3-20: MULTIPLE-WORD PROGRAM CYCLE

Program Cycle
(for Writing to Program Flash Memory or Data Flash/EEPROM Memory)

- Load Data for NVM
- Increment Address
- Load Data for NVM
- Increment Address
- Load Data for NVM

- Begin Programming Command (Internally timed)
- Begin Programming Command (Externally timed)
- Wait TPINT
- Wait TPEXT
- End Externally Timed Programming Command
- Wait TDIS
FIGURE 3-21: USER ID AND CONFIGURATION MEMORY PROGRAM FLOWCHART

Start

Load PC Address (selects Bulk Erase regions)

Bulk Erase Program Memory(1)

Load PC Address (8000h)

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

Read from NVM Command

Data Correct?

Yes

Increment PC Address

No

Address = 8004h?

Yes

Load PC Address Command (8007h)

No

Address = 8004h?

Yes

Load PC Address Command (8007h)

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

Read Data from NVM Command

Data Correct?

No

Report Programming Failure

Yes

Increment PC Address

No

Address = 800Ch?

Yes

Done

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

2: See Figure 3-12.
3.3 Code Protection

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

The Revision ID, Device ID, Device Information Area, Device Configuration Information, User IDs and Configuration Words can be read out regardless of the code protection settings.

3.3.1 PROGRAM MEMORY

Code protection is enabled by programming the CP bit to ‘0’. The only way to disable code protection is to use the Bulk Erase Memory command (with the PC set to an address so as to Bulk Erase all program Flash contents).

**Note:** See Figure 2-1 for device specific Program Memory Size and locations.
3.4 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. For example, if the Configuration Word 1 is stored at 8007h, in the hex file this will be referenced as 1000Eh-1000Fh.

3.5 Configuration Words

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.

3.6 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.

3.6.1 PROGRAM CODE PROTECTION DISABLED CHECKSUM COMPUTATIONS

With the program code protection disabled, the checksum is computed by reading the contents of the program memory locations and summing up the program memory data starting at address 0000h, up to the maximum user addressable location. Any Carry bits exceeding 16 bits are ignored. Additionally, the relevant bits of the Configuration Words are added to the checksum. For PIC16 devices (14-bit program memory word), the two MSBs are taken as zero. All unimplemented Configuration bits are masked to '0' (see Appendix B: “PIC16(L)F1919X Device ID, Checksums and Pinout Descriptions”).

3.6.2 PROGRAM CODE PROTECTION ENABLED CHECKSUM COMPUTATIONS

When the MPLAB® X IDE check box for Project Properties → Building → Insert unprotected checksum in user ID memory 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 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'.
3.7 Electrical Specifications

Refer to device-specific data sheet for absolute maximum ratings.

### TABLE 3-3: AC/DC CHARACTERISTICS TIMING REQUIREMENTS FOR PROGRAM/VERIFY MODE

<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>
<tr>
<td>Sym.</td>
<td>Characteristics</td>
</tr>
<tr>
<td>VDD</td>
<td>Supply Voltage (VDDMIN(1), VDDMAX)</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>VPEW</td>
<td>Read/Write and Row Erase operations</td>
</tr>
<tr>
<td>VBE</td>
<td>Bulk Erase operations</td>
</tr>
<tr>
<td>IDD</td>
<td>Current on Vdd, Idle</td>
</tr>
<tr>
<td>IDDP</td>
<td>Current on Vdd, Programming</td>
</tr>
<tr>
<td>VPP</td>
<td>Current on MCLR/VPP</td>
</tr>
<tr>
<td>VIHH</td>
<td>High Voltage on MCLR/VPP for Program/Verify Mode Entry</td>
</tr>
<tr>
<td>TVHHR</td>
<td>MCLR Rise Time (Vil to ViHH) for Program/Verify Mode Entry</td>
</tr>
<tr>
<td>I/O pins</td>
<td></td>
</tr>
<tr>
<td>VIH</td>
<td>(ICSPCLK, ICSPDAT, MCLR/VPP) Input High Level</td>
</tr>
<tr>
<td>VIL</td>
<td>(ICSPCLK, ICSPDAT, MCLR/VPP) Input Low Level</td>
</tr>
<tr>
<td>VOH</td>
<td>ICSPDAT Output High Level</td>
</tr>
<tr>
<td></td>
<td>VDD-0.7</td>
</tr>
<tr>
<td></td>
<td>VDD-0.7</td>
</tr>
<tr>
<td>VOL</td>
<td>ICSPDAT Output Low Level</td>
</tr>
<tr>
<td></td>
<td>VSS+0.6</td>
</tr>
<tr>
<td></td>
<td>VSS+0.6</td>
</tr>
<tr>
<td>Programming Mode Entry and Exit</td>
<td></td>
</tr>
<tr>
<td>TENTS</td>
<td>Programing Mode Entry Setup Time: ICSPCLK, ICSPDAT Setup Time Before Vdd or MCLR↑</td>
</tr>
<tr>
<td>TENTH</td>
<td>Programing Mode Entry Hold Time: ICSPCLK, ICSPDAT Hold Time after Vdd or MCLR↑</td>
</tr>
<tr>
<td>Serial Program/Verify</td>
<td></td>
</tr>
<tr>
<td>TCKL</td>
<td>Clock Low Pulse Width</td>
</tr>
<tr>
<td>TCKH</td>
<td>Clock High Pulse Width</td>
</tr>
<tr>
<td>TDS</td>
<td>Data in SETUP TIME before Clock↓</td>
</tr>
<tr>
<td>TDH</td>
<td>Data in HOLD TIME after Clock↓</td>
</tr>
<tr>
<td>TCO</td>
<td>Clock↑ to DATA OUT VALID (during a Read Data Command)</td>
</tr>
</tbody>
</table>

**Note 1:** Bulk Erased devices default to brown-out enabled, with BORV = 1 (low trip point). VDDMIN is the VbOr threshold (with BORV = 1) when performing low-voltage programming on a Bulk Erased device, to ensure that the device is not held in Brown-out Reset.

**Note 2:** The hardware requires Vdd to be above the BOR threshold, at the ~2.4V nominal setting, in order to perform Bulk Erase operations. This threshold does not depend on the BORV Configuration bit settings. The threshold is the same for both F and LF devices, even though the LF devices may not have a user configurable ~2.4V nominal BOR trip point setting. Refer to the microcontroller device data sheet specifications for min./typ./max. limits of the VbOr level (at the BORV = 0 setting of F devices).

**Note 3:** Externally timed writes are not supported for Configuration bits.
<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>
<tr>
<td>Symbol</td>
<td>Characteristics</td>
</tr>
<tr>
<td>TLZD</td>
<td>Clock↓ to Data Low-impedance (during a Read Data Command)</td>
</tr>
<tr>
<td>THZD</td>
<td>Clock↓ to Data High-impedance (during a Read Data command)</td>
</tr>
<tr>
<td>TDLY</td>
<td>Data Input not Driven To Next Clock Input (delay required between command/data or command/command)</td>
</tr>
<tr>
<td>TERAB</td>
<td>Bulk Erase Cycle Time</td>
</tr>
<tr>
<td>TERAR</td>
<td>Row Erase Cycle Time</td>
</tr>
<tr>
<td>TPINT</td>
<td>Internally Timed Programming Operation Time</td>
</tr>
<tr>
<td>TPEXT</td>
<td>Delay Required between Begin Externally Timed Programming and End Externally Timed Programming Commands</td>
</tr>
<tr>
<td>TDIS</td>
<td>Delay Required after End Externally Timed Programming Command</td>
</tr>
<tr>
<td>TEXIT</td>
<td>Time Delay when Exiting Program/Verify Mode</td>
</tr>
</tbody>
</table>

**Note 1:** Bulk Erased devices default to brown-out enabled, with BORV = 1 (low trip point). VDDMIN is the VBOR threshold (with BORV = 1) when performing low-voltage programming on a Bulk Erased device, to ensure that the device is not held in Brown-out Reset.

**Note 2:** The hardware requires VDD to be above the BOR threshold, at the ~2.4V nominal setting, in order to perform Bulk Erase operations. This threshold does not depend on the BORV Configuration bit settings. The threshold is the same for both F and LF devices, even though the LF devices may not have a user configurable ~2.4V nominal BOR trip point setting. Refer to the microcontroller device data sheet specifications for min./typ./max. limits of the VBOR level (at the BORV = 1 setting of F devices).

**Note 3:** Externally timed writes are not supported for Configuration bits.
APPENDIX A:  REVISION HISTORY

Revision A (4/2016)
Initial release of this document.

Revision B (2/2017)
Updated Table B-1; Updated Examples B-1, B-2, B-3, B-4. Other minor corrections.

Revision C (6/2018)
Updated Example B-3. Updated Register 2-1.
# APPENDIX B: PIC16(L)F1919X DEVICE ID, CHECKSUMS AND PINOUT DESCRIPTIONS

## TABLE B-1: DEVICE IDs AND CHECKSUMS

<table>
<thead>
<tr>
<th>Device</th>
<th>Device ID</th>
<th>Config.1</th>
<th>Mask (HEX)</th>
<th>Config.2</th>
<th>Mask (HEX)</th>
<th>Config.3</th>
<th>Mask (HEX)</th>
<th>Config.4</th>
<th>Mask (HEX)</th>
<th>Config.5</th>
<th>Mask (HEX)</th>
<th>Checksum</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIC16F19195</td>
<td>309E</td>
<td>3FFF</td>
<td>2F77</td>
<td>3FFF</td>
<td>3EE7</td>
<td>3FFF</td>
<td>3F7F</td>
<td>3FFF</td>
<td>2F9F</td>
<td>3FFF</td>
<td>3F7F</td>
<td>0001 BD7D 3ED3 9AF5 1C4B</td>
</tr>
<tr>
<td>PIC16LF19195</td>
<td>309F</td>
<td>3FFF</td>
<td>2F77</td>
<td>3FFF</td>
<td>3EE7</td>
<td>3FFF</td>
<td>3F7F</td>
<td>3FFF</td>
<td>2F9F</td>
<td>3FFF</td>
<td>3F7F</td>
<td>0001 BD7D 3ED3 9AF5 1C4B</td>
</tr>
<tr>
<td>PIC16F19196</td>
<td>30A0</td>
<td>3FFF</td>
<td>2F77</td>
<td>3FFF</td>
<td>3EE7</td>
<td>3FFF</td>
<td>3F7F</td>
<td>3FFF</td>
<td>2F9F</td>
<td>3FFF</td>
<td>3F7F</td>
<td>0001 9D7D 1ED3 7AF5 FC4B</td>
</tr>
<tr>
<td>PIC16LF19196</td>
<td>30A1</td>
<td>3FFF</td>
<td>2F77</td>
<td>3FFF</td>
<td>3EE7</td>
<td>3FFF</td>
<td>3F7F</td>
<td>3FFF</td>
<td>2F9F</td>
<td>3FFF</td>
<td>3F7F</td>
<td>0001 9D7D 1ED3 7AF5 FC4B</td>
</tr>
<tr>
<td>PIC16F19197</td>
<td>30A2</td>
<td>3FFF</td>
<td>2F77</td>
<td>3FFF</td>
<td>3EE7</td>
<td>3FFF</td>
<td>3F7F</td>
<td>3FFF</td>
<td>2F9F</td>
<td>3FFF</td>
<td>3F7F</td>
<td>0001 5D7D DED3 3AF5 BC4B</td>
</tr>
<tr>
<td>PIC16LF19197</td>
<td>30A3</td>
<td>3FFF</td>
<td>2F77</td>
<td>3FFF</td>
<td>3EE7</td>
<td>3FFF</td>
<td>3F7F</td>
<td>3FFF</td>
<td>2F9F</td>
<td>3FFF</td>
<td>3F7F</td>
<td>0001 5D7D DED3 3AF5 BC4B</td>
</tr>
</tbody>
</table>
### EXAMPLE B-1: CHECKSUM COMPUTED WITH PROGRAM CODE PROTECTION DISABLED
**PIC16F19195, BLANK DEVICE**

<table>
<thead>
<tr>
<th>Device</th>
<th>Sum of Memory addresses</th>
<th>Configuration Word 1</th>
<th>Configuration Word 1 mask</th>
<th>Configuration Word 2</th>
<th>Configuration Word 2 mask</th>
<th>Configuration Word 3</th>
<th>Configuration Word 3 mask</th>
<th>Configuration Word 4</th>
<th>Configuration Word 4 mask</th>
<th>Configuration Word 5 Unprotected</th>
<th>Configuration Word 5 mask</th>
<th>Checksum</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIC16F19195</td>
<td>0000h-1FFFh</td>
<td>E000h</td>
<td>3FFFh</td>
<td>2F77h</td>
<td>3FFFh</td>
<td>3EEE7h</td>
<td>3FFFh</td>
<td>3FFFh</td>
<td>3FFFh</td>
<td>2F9Fh</td>
<td>3FFFh</td>
<td>0001h</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>F000h + 2F77h + 3EEE7h + 3FFFh + 2F9Fh + 0001h</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>BD7Dh</td>
</tr>
</tbody>
</table>

### EXAMPLE B-2: CHECKSUM COMPUTED WITH PROGRAM CODE PROTECTION DISABLED
**PIC16F19195, 00AAh AT FIRST AND LAST ADDRESS**

<table>
<thead>
<tr>
<th>Device</th>
<th>Sum of Memory addresses</th>
<th>Configuration Word 1</th>
<th>Configuration Word 1 mask</th>
<th>Configuration Word 2</th>
<th>Configuration Word 2 mask</th>
<th>Configuration Word 3</th>
<th>Configuration Word 3 mask</th>
<th>Configuration Word 4</th>
<th>Configuration Word 4 mask</th>
<th>Configuration Word 5 Unprotected</th>
<th>Configuration Word 5 mask</th>
<th>Checksum</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIC16F19195</td>
<td>0000h-1FFFh</td>
<td>6156h</td>
<td>3FFFh</td>
<td>2F77h</td>
<td>3FFFh</td>
<td>3EEE7h</td>
<td>3FFFh</td>
<td>3FFFh</td>
<td>3FFFh</td>
<td>2F9Fh</td>
<td>3FFFh</td>
<td>0001h</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6156h + 2F77h + 3EEE7h + 3FFFh + 2F9Fh + 0001h</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3ED3h</td>
</tr>
</tbody>
</table>
### EXAMPLE B-3: CHECKSUM COMPUTED WITH PROGRAM CODE PROTECTION ENABLED
PIC16F19195, BLANK DEVICE

<table>
<thead>
<tr>
<th>Configuration Word 1</th>
<th>3FFFh</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration Word 1 mask</td>
<td>2F77h</td>
</tr>
<tr>
<td>Configuration Word 2</td>
<td>3FFFh</td>
</tr>
<tr>
<td>Configuration Word 2 mask</td>
<td>3EE7h</td>
</tr>
<tr>
<td>Configuration Word 3</td>
<td>3FFFh</td>
</tr>
<tr>
<td>Configuration Word 3 mask</td>
<td>3F7Fh</td>
</tr>
<tr>
<td>Configuration Word 4</td>
<td>3FFFh</td>
</tr>
<tr>
<td>Configuration Word 4 mask</td>
<td>2F9Fh</td>
</tr>
<tr>
<td>Configuration Word 5 Unprotected</td>
<td>3FFEh</td>
</tr>
<tr>
<td>Configuration Word 5 mask</td>
<td>0001h</td>
</tr>
</tbody>
</table>

Sum of User IDs = (000Bh and 000Fh) << 12 + (000Dh and 000Fh) << 8 + (0007h and 000Fh) << 4 + (000Dh and 000Fh)

= B000h + 0D00h + 0070h + 000Dh

= BD7Dh

Checksum = (3FFFh and 2F77h) + (3FFFh and 3EE7h) + (3FFFh and 3F7Fh) + (3FFFh and 2F9Fh) + (3FFEh and 0001h) + BD7Dh

= 2F77h + 3EE7h + 3F7Fh + 2F9Fh + 0000h + BD7Dh

= 9AF9h

### EXAMPLE B-4: CHECKSUM COMPUTED WITH PROGRAM CODE PROTECTION ENABLED
PIC16F19195, 00AAh AT FIRST AND LAST ADDRESS

<table>
<thead>
<tr>
<th>Configuration Word 1</th>
<th>3FFFh</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration Word 1 mask</td>
<td>2F77h</td>
</tr>
<tr>
<td>Configuration Word 2</td>
<td>3FFFh</td>
</tr>
<tr>
<td>Configuration Word 2 mask</td>
<td>3EE7h</td>
</tr>
<tr>
<td>Configuration Word 3</td>
<td>3FFFh</td>
</tr>
<tr>
<td>Configuration Word 3 mask</td>
<td>3F7Fh</td>
</tr>
<tr>
<td>Configuration Word 4</td>
<td>3FFFh</td>
</tr>
<tr>
<td>Configuration Word 4 mask</td>
<td>2F9Fh</td>
</tr>
<tr>
<td>Configuration Word 5 Unprotected</td>
<td>3FFEh</td>
</tr>
<tr>
<td>Configuration Word 5 mask</td>
<td>0001h</td>
</tr>
</tbody>
</table>

Sum of User IDs = (0003h and 000Fh) << 12 + (000Eh and 000Fh) << 8 + (0007h and 000Fh) << 4 + (0003h and 000Fh)

= 3000h + 0E00h + 000Dh + 0003h

= 3ED3h

Checksum = (3FFFh and 2F77h) + (3FFFh and 3EE7h) + (3FFFh and 3F7Fh) + (3FFFh and 2F9Fh) + (3FFEh and 0001h) + 3ED3h

= 2F77h + 3EE7h + 3F7Fh + 2F9Fh + 0000h + 3ED3h

= 1C4Bh

**Note:** The Sum of User IDs in Example B-3 and Example B-4 is with the unprotected checksum inserted into the User ID memory. See Section 3.6.2, Program Code Protection Enabled Checksum Computations.
### TABLE B-2: PROGRAMMING PIN LOCATIONS BY PACKAGE TYPE

<table>
<thead>
<tr>
<th>Device</th>
<th>Package Code</th>
<th>Package Type</th>
<th>VDD PIN</th>
<th>VSS PIN</th>
<th>MCLR PIN</th>
<th>ICSPCLK PIN</th>
<th>ICSPDAT PIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIC16(L)F19195</td>
<td>64-pin TQFP</td>
<td>PT</td>
<td>10,38</td>
<td>9,41</td>
<td>7</td>
<td>42</td>
<td>37</td>
</tr>
<tr>
<td></td>
<td>64-pin QFN</td>
<td>NG</td>
<td>10,38</td>
<td>9,41</td>
<td>7</td>
<td>42</td>
<td>37</td>
</tr>
<tr>
<td>PIC16(L)F19196</td>
<td>64-pin TQFP</td>
<td>PT</td>
<td>10,38</td>
<td>9,41</td>
<td>7</td>
<td>42</td>
<td>37</td>
</tr>
<tr>
<td></td>
<td>64-pin QFN</td>
<td>NG</td>
<td>10,38</td>
<td>9,41</td>
<td>7</td>
<td>42</td>
<td>37</td>
</tr>
<tr>
<td>PIC16(L)F19197</td>
<td>64-pin TQFP</td>
<td>PT</td>
<td>10,38</td>
<td>9,41</td>
<td>7</td>
<td>42</td>
<td>37</td>
</tr>
<tr>
<td></td>
<td>64-pin QFN</td>
<td>NG</td>
<td>10,38</td>
<td>9,41</td>
<td>7</td>
<td>42</td>
<td>37</td>
</tr>
</tbody>
</table>
## REGISTER B-1: CONFIGURATION WORD 1: OSCILLATORS

<table>
<thead>
<tr>
<th>R/P-1</th>
<th>U-1</th>
<th>R/P-1</th>
<th>R/P-1</th>
<th>U-1</th>
<th>R/P-1</th>
<th>R/P-1</th>
<th>U-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>FCMEN</td>
<td>—</td>
<td>CSWEN</td>
<td>LCDPM</td>
<td>VBATEN</td>
<td>CLKOUTEN</td>
<td>—</td>
<td>RSTOSC2</td>
<td>RSTOSC1</td>
<td>RSTOSC0</td>
<td>—</td>
<td>FEXTOSC2</td>
</tr>
</tbody>
</table>

**Legend:**

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

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

### bit 12  Unimplemented: Read as ‘1’

### bit 11  CSWEN: Clock Switch Enable bit
- **1** = Writing to NOSC and NDIV is allowed
- **0** = The NOSC and NDIV bits cannot be changed by user software

### bit 10  LCDPM: LCD Charge Pump Mode bit
- **1** = User intends to enable LCD Charge Pump during LCD operation
- **0** = LCD Charge Pump forced off

### bit 9  VBATEN: VBAT Pin Enable bit
- **1** = VBAT functionality is disabled; VBAT pin becomes GPIO
- **0** = VBAT functionality is enabled; VBAT pin has a battery connected to it

### bit 8  CLKOUTEN: Clock Out Enable bit
- **1** = CLKOUT function is disabled; I/O or oscillator function on OSC2
- **0** = CLKOUT function is enabled; FOSC/4 clock appears at OSC2

- **Otherwise:** This bit is ignored.

### bit 7  Unimplemented: Read as ‘1’

### bit 6-4  RSTOSC<2:0>: Power-Up Default Value for COSC bits
- This value is the Reset-default value for COSC and selects the oscillator first used by user software.
- **111** = EXTOSC operating per FEXTOSC bits (device manufacturing default)
- **110** = HFINTOSC with HFFRQ = 4’b0000 4 MHz and CDIV = 4:1
- **101** = LFINTOSC
- **100** = SOSC
- **011** = Reserved (Defaults to HFINTOSC, OSCFRQ = 4 MHz, CDIV = 1:1)
- **010** = EXTOSC with 4x PLL, with EXTOSC operating per FEXTOSC bits
- **001** = EXTOSC with 2x PLL, with EXTOSC operating per FEXTOSC bits
- **000** = HFINTOSC with 2x PLL and HFFRQ = 4’b1111 32 MHz and CDIV = 1:1

### bit 3  Unimplemented: Read as ‘1’

### bit 2-0  FEXTOSC<2:0>: FEXTOSC External Oscillator Mode Selection bits
- **111** = EC (External Clock) above 8 MHz; PFM set to high power (device manufacturing default)
- **110** = EC (External Clock) for 100 kHz to 8 MHz; PFM set to medium power
- **101** = EC (External Clock) below 100 kHz; PFM set to low power
- **100** = Oscillator not enabled
- **011** = Oscillator not enabled
- **010** = Oscillator not enabled
- **001** = Oscillator not enabled
- **000** = Oscillator not enabled
## REGISTER B-2: CONFIGURATION WORD 2: SUPERVISORS

<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 7-6</th>
<th>Bit 5</th>
<th>Bit 4-3</th>
<th>Bit 2-1</th>
<th>Bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEBUG</td>
<td>STVREN</td>
<td>PPS1WAY</td>
<td>ZCDDIS</td>
<td>BORV</td>
<td>—</td>
<td>BOREN1</td>
<td>BOREN0</td>
<td>LPBOREN</td>
<td>—</td>
<td>PWRTS1</td>
</tr>
<tr>
<td>bit 13</td>
<td>bit 12</td>
<td>bit 11</td>
<td>bit 10</td>
<td>bit 9</td>
<td>bit 8</td>
<td>bit 7-6</td>
<td>bit 5</td>
<td>bit 4-3</td>
<td>bit 2-1</td>
<td>bit 0</td>
</tr>
</tbody>
</table>

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

#### bit 13: **DEBUG** - Debugger Enable bit
- **1** = Background debugger disabled
- **0** = Background debugger enabled

#### bit 12: **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 11: **PPS1WAY** - PPSLOCK One-Way Set Enable bit
- **1** = The PPSLOCK bit can be cleared and set only once; PPS registers remain locked after one clear/set cycle
- **0** = The PPSLOCK bit can be set and cleared repeatedly (subject to the unlock sequence)

#### bit 10: **ZCDDIS** - Zero-Cross Detect Disable bit
- **1** = ZCD disabled. ZCD can be enabled by setting the ZCDSEN bit of the ZCDCON register
- **0** = ZCD always enabled (ZCDSEN bit is ignored)

#### bit 9: **BORV** - Brown-out Reset Voltage Selection bit
- **1** = Brown-out Reset voltage (V_BOR) set to lower trip point level
- **0** = Brown-out Reset voltage (V_BOR) set to higher trip point level
- 
  The higher voltage setting is recommended for operation at or above 16 MHz.

#### bit 8: Unimplemented: Read as ‘1’

#### bit 7-6: **BOREN<1:0>** - Brown-out Reset Enable bits
- When enabled, Brown-out Reset Voltage (V_BOR) is set by the BORV bit
- **11** = Brown-out Reset is enabled; SBOREN bit is ignored
- **10** = Brown-out Reset is enabled while running, disabled in Sleep; SBOREN bit is ignored
- **01** = Brown-out Reset is enabled according to SBOREN
- **00** = Brown-out Reset is disabled

#### bit 5: **LPBOREN** - Low-Power BOR Enable bit
- **1** = ULPBOR is disabled
- **0** = ULPBOR is enabled

#### bit 4-3: Unimplemented: Read as ‘1’

#### bit 2-1: **PWRTS<1:0>** - Power-up Timer selection bits
- **11** = PWRT disabled
- **10** = PWRT set at 64 ms
- **01** = PWRT set at 16 ms
- **00** = PWRT set at 1 ms

#### bit 0: **MCLRE** - Master Clear (MCLR) Enable bit
- **If LVP = 1:**
  - RG5 pin function is MCLR (it will reset the device when driven low)
- **If LVP = 0:**
  - **1** = MCLR pin is MCLR (it will the reset device when driven low)
  - **0** = MCLR pin may be used as general purpose RG5 input
REGISTER B-3: CONFIGURATION WORD 3: WINDOWED WATCHDOG

<table>
<thead>
<tr>
<th>bit 13</th>
<th>bit 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>WDTCSCS2</td>
<td>WDTCSCS1</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- **WDTCSCS<2:0>:** WDT Input Clock Selector bits
- 000 = WDT reference clock is the 31.25 kHz HFINTOSC (MFINTOSC) output
- 001 = WDT reference clock is the 31.0 kHz LFINTOSC (default value)
- 010 = Reserved
- 011 = Software Control
- 110 = Reserved
- 111 = Software Control

bit 10- **WDTCWS<2:0>:** WDT Window Select bits

<table>
<thead>
<tr>
<th>WDTWS at POR</th>
<th>Software Control of WDTWS?</th>
<th>Keyed Access Required?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Window Delay</td>
<td>Window opening Percent of</td>
<td>Yes</td>
</tr>
<tr>
<td>Percent of Time</td>
<td>Percent of Time</td>
<td></td>
</tr>
<tr>
<td>Value</td>
<td></td>
<td></td>
</tr>
<tr>
<td>000</td>
<td>000</td>
<td>87.5</td>
</tr>
<tr>
<td>001</td>
<td>001</td>
<td>75</td>
</tr>
<tr>
<td>010</td>
<td>010</td>
<td>62.5</td>
</tr>
<tr>
<td>011</td>
<td>011</td>
<td>50</td>
</tr>
<tr>
<td>100</td>
<td>100</td>
<td>37.5</td>
</tr>
<tr>
<td>101</td>
<td>101</td>
<td>25</td>
</tr>
<tr>
<td>110</td>
<td>111</td>
<td>n/a</td>
</tr>
<tr>
<td>111</td>
<td>111</td>
<td>n/a</td>
</tr>
</tbody>
</table>

bit 7 Unimplemented: Read as ‘1’

bit 6-5 **WDTE<1:0>:** WDT Operating mode:
- 00 = WDT disabled, SWDTEN is ignored
- 01 = WDT enabled/disabled by SWDTEN bit in WDTCON0
- 10 = WDT enabled while Sleep = 0, suspended when Sleep = 1; SWDTEN ignored
- 11 = WDT enabled regardless of Sleep; SWDTEN is ignored
### REGISTER B-3: CONFIGURATION WORD 3: WINDOWED WATCHDOG (CONTINUED)

**bit 4-0 WDTCPS<4:0>: WDT Period Select bits**

<table>
<thead>
<tr>
<th>WDTCPS</th>
<th>WDTPS</th>
<th>Value</th>
<th>Divider Ratio</th>
<th>Typical Time Out (FIN = 31 kHz)</th>
<th>Software Control of WDTPS?</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000</td>
<td>00000</td>
<td>1:32</td>
<td>2^0</td>
<td>1 ms</td>
<td>No</td>
</tr>
<tr>
<td>00001</td>
<td>00001</td>
<td>1:64</td>
<td>2^1</td>
<td>2 ms</td>
<td>No</td>
</tr>
<tr>
<td>00010</td>
<td>00010</td>
<td>1:128</td>
<td>2^2</td>
<td>4 ms</td>
<td>No</td>
</tr>
<tr>
<td>00011</td>
<td>00011</td>
<td>1:256</td>
<td>2^3</td>
<td>8 ms</td>
<td>No</td>
</tr>
<tr>
<td>00100</td>
<td>00100</td>
<td>1:512</td>
<td>2^4</td>
<td>16 ms</td>
<td>No</td>
</tr>
<tr>
<td>00101</td>
<td>00101</td>
<td>1:1024</td>
<td>2^5</td>
<td>32 ms</td>
<td>No</td>
</tr>
<tr>
<td>00110</td>
<td>00110</td>
<td>1:2048</td>
<td>2^6</td>
<td>64 ms</td>
<td>No</td>
</tr>
<tr>
<td>00111</td>
<td>00111</td>
<td>1:4096</td>
<td>2^7</td>
<td>128 ms</td>
<td>No</td>
</tr>
<tr>
<td>01000</td>
<td>01000</td>
<td>1:8192</td>
<td>2^8</td>
<td>256 ms</td>
<td>No</td>
</tr>
<tr>
<td>01001</td>
<td>01001</td>
<td>1:16384</td>
<td>2^9</td>
<td>512 ms</td>
<td>No</td>
</tr>
<tr>
<td>01010</td>
<td>01010</td>
<td>1:32768</td>
<td>2^10</td>
<td>1 s</td>
<td>No</td>
</tr>
<tr>
<td>01011</td>
<td>01011</td>
<td>1:65536</td>
<td>2^11</td>
<td>2 s</td>
<td>No</td>
</tr>
<tr>
<td>01100</td>
<td>01100</td>
<td>1:131072</td>
<td>2^12</td>
<td>4 s</td>
<td>No</td>
</tr>
<tr>
<td>01101</td>
<td>01101</td>
<td>1:262144</td>
<td>2^13</td>
<td>8 s</td>
<td>No</td>
</tr>
<tr>
<td>01110</td>
<td>01110</td>
<td>1:524299</td>
<td>2^14</td>
<td>16 s</td>
<td>No</td>
</tr>
<tr>
<td>01111</td>
<td>01111</td>
<td>1:1048576</td>
<td>2^15</td>
<td>32 s</td>
<td>No</td>
</tr>
<tr>
<td>10000</td>
<td>10000</td>
<td>1:2097152</td>
<td>2^16</td>
<td>64 s</td>
<td>No</td>
</tr>
<tr>
<td>10001</td>
<td>10001</td>
<td>1:4194304</td>
<td>2^17</td>
<td>128 s</td>
<td>No</td>
</tr>
<tr>
<td>10010</td>
<td>10010</td>
<td>1:8388608</td>
<td>2^18</td>
<td>256 s</td>
<td>No</td>
</tr>
<tr>
<td>10011</td>
<td>10011</td>
<td>1:16777216</td>
<td>2^19</td>
<td>512 s</td>
<td>No</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>No</td>
</tr>
<tr>
<td>11110</td>
<td>11110</td>
<td>1:32</td>
<td>2^5</td>
<td>1 ms</td>
<td>No</td>
</tr>
<tr>
<td>11111</td>
<td>11111</td>
<td>1:65536</td>
<td>2^16</td>
<td>2 s</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Note 1:** Default fuse - 5'b11111
**REGISTER B-4: CONFIGURATION WORD 4: MEMORY**

<table>
<thead>
<tr>
<th>R/W-1</th>
<th>U-1</th>
<th>R/W-1</th>
<th>R/W-1</th>
<th>R/W-1</th>
<th>R/W-1</th>
<th>R/W-1</th>
<th>bit 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>bit 13</td>
<td>12</td>
<td>11</td>
<td>10</td>
<td>9</td>
<td>R/W-1</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**  
**LVP:** Low Voltage Programming Enable bit  
1 = Low voltage programming enabled. MCLR/VPP pin function is MCLR. MCLRE Configuration bit is ignored.  
0 = HV on MCLR/VPP must be used for programming.  
The LVP bit cannot be written (to zero) while operating from the LVP programming interface. The purpose of this rule is to prevent the user from dropping out of LVP mode while programming from LVP mode, or accidentally eliminating LVP mode from the configuration state.  
The preconditioned (erased) state for this bit is critical.

**bit 12**  
**Unimplemented:** Read as ‘1’

**bit 11**  
**WRTSAF:** Storage Area Flash Write Protection bit  
1 = SAF NOT write-protected  
0 = SAF write-protected  
Unimplemented, if SAF is not supported in the device family and only applicable if SAFEN = 0.

**bit 10**  
**WRTD:** Data EEPROM Write Protection bit  
1 = Data EEPROM NOT write-protected  
0 = Data EEPROM write-protected  
Unimplemented if data EEPROM is not present.

**bit 9**  
**WRTC:** Configuration Register Write Protection bit  
1 = Configuration Register NOT write-protected  
0 = Configuration Register write-protected

**bit 8**  
**WRTB:** Boot Block Write Protection bit  
1 = Boot Block NOT write-protected  
0 = Boot Block write-protected  
Only applicable if BBEN = 0.

**bit 7**  
**WRTAPP:** Application Block Write Protection bit  
1 = Application Block NOT write-protected  
0 = Application Block write-protected

**bit 6-5**  
**Unimplemented:** Read as ‘1’.

**bit 4**  
**SAFEN:** SAF Enable bit  
1 = SAF disabled  
0 = SAF enabled

**bit 3**  
**BBEN:** Boot Block Enable bit  
1 = Boot Block disabled  
0 = Boot Block enabled

**bit 2-0**  
**BBSIZE[2:0]:** Boot Block Size Selection bits (Refer to Table C-2)  
BBSIZE is used only when BBEN = 0  
BBSIZE bits can only be written while BBEN = 1; after BBEN = 0, BBSIZ is write-protected.

**Note 1:** Bits are implemented as sticky bits. Once protection is enabled, it can only be reset through a Bulk Erase.
APPENDIX C: MEMORY PARTITIONING

User Flash is partitioned into:

- Application Block
- Boot Block
- SAF Block

according to BBEN, BBSIZE<2:0> and SAFEN (Register B-4). Default settings assign all memory in the User Flash area to Application Block.

Boot Block

Boot block, if enabled begins at the lowest address of memory and spans the size specified by BBSIZE<2:0>. The SAF block, if enabled, is placed at the end of memory and spans 128 words.

Memory Write Protection

All partitions have a corresponding write protection fuse (Register B-4). If write-protected locations are written from NVMCON, memory is not changed and WRERR is set.

Note: Partitioned memory replaces the write protection options (WTC) of previous devices. Partitioning and associated write protection can be cleared using bulk erase.

TABLE C-1: MEMORY MAP PARTITIONS AND PROTECTION

<table>
<thead>
<tr>
<th>Reg</th>
<th>Address</th>
<th>Partition</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>BBEN = 1 SAFEN = 1</td>
</tr>
<tr>
<td>PFM</td>
<td>0000h</td>
<td>WRTAPP = 0: Application Block write-protected</td>
</tr>
<tr>
<td></td>
<td>• • • End of Boot address</td>
<td>End of Boot address + 1</td>
</tr>
<tr>
<td></td>
<td>WRTAPP = 0: Application Block write-protected</td>
<td>WRTAPP = 0: Application Block write-protected</td>
</tr>
<tr>
<td></td>
<td>End of PFM address - 128 • • • End of PFM address - 1</td>
<td>WRTSAF = 0: SAF write-protected</td>
</tr>
<tr>
<td>Config</td>
<td>Configuration Words</td>
<td>WRTC = 0: Configuration Registers write-protected</td>
</tr>
<tr>
<td>DFM</td>
<td>Data Flash Memory</td>
<td>WRTD = 0: Data EEPROM write-protected</td>
</tr>
</tbody>
</table>

Note 1: End of Boot address is based on BBSIZE<2:0>, see Register B-4 and Table C-2.

2: End of PFM address is based on Figure 2-1.

3: Configuration Words and Data Flash Memory address are based on Figure 2-1.
TABLE C-2:  BOOT BLOCK SIZE BITS

<table>
<thead>
<tr>
<th>BBEN</th>
<th>BBSIZE&lt;2:0&gt;</th>
<th>Preferred Boot Block Size (words)</th>
<th>Actual Boot Block Size</th>
<th>End of Boot Address</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>User Program Memory Size(^{(2)}) (words)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1k</td>
<td>2k</td>
</tr>
<tr>
<td>1</td>
<td>xxx</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>111</td>
<td>512</td>
<td>512</td>
<td>512</td>
</tr>
<tr>
<td>0</td>
<td>110</td>
<td>1024</td>
<td>1024</td>
<td>1024</td>
</tr>
<tr>
<td>0</td>
<td>101</td>
<td>2048</td>
<td>2048</td>
<td>2048</td>
</tr>
<tr>
<td>0</td>
<td>100</td>
<td>4096</td>
<td>4096</td>
<td>4096</td>
</tr>
<tr>
<td>0</td>
<td>011</td>
<td>8192</td>
<td>8192</td>
<td>8192</td>
</tr>
<tr>
<td>0</td>
<td>010</td>
<td>16384</td>
<td>16384</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>001</td>
<td>32768</td>
<td>32768</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>000</td>
<td>65536</td>
<td>65536</td>
<td></td>
</tr>
<tr>
<td>Note 1: This is generic information, and not all entries apply to this device.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Note 2: For each device, the user program memory size specification is listed in Figure 2-1.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Note 3: The maximum boot block size is half the user program memory size. All selections higher than the maximum are set to half size. For example, all BBSIZE = 000 - 100 produce a boot block size of 4kW on a 8kW device.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

REGISTER C-1:  CONFIGURATION WORD 5: CODE PROTECTION

<table>
<thead>
<tr>
<th></th>
<th>U-1</th>
<th>U-1</th>
<th>U-1</th>
<th>U-1</th>
<th>U-1</th>
<th>U-1</th>
<th>U-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>bit 13</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bit 8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>U-1</th>
<th>U-1</th>
<th>U-1</th>
<th>U-1</th>
<th>U-1</th>
<th>U-1</th>
<th>U-1</th>
<th>R/P-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>bit 7</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>CP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bit 0</td>
<td></td>
<td></td>
<td></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 13-1  Unimplemented: Read as ‘1’.

bit 0  CP: Program Flash Memory Code Protection bit

1 = Program Flash Memory code protection disabled
0 = Program Flash Memory code protection enabled
## APPENDIX D: DEVICE CONFIGURATION INFORMATION (DCI)

### TABLE D-1: DEVICE CONFIGURATION INFORMATION

<table>
<thead>
<tr>
<th>ADDRESS</th>
<th>DESCRIPTION</th>
<th>VALUE</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>8200h</td>
<td>Erase Row Size</td>
<td>64</td>
<td>Words</td>
</tr>
<tr>
<td>8201h</td>
<td>Number of write latches</td>
<td>64</td>
<td>—</td>
</tr>
<tr>
<td>8202h</td>
<td>Number of User Rows</td>
<td>See Table D-2</td>
<td>Rows</td>
</tr>
<tr>
<td>8203h</td>
<td>EE Data memory size</td>
<td>256</td>
<td>Bytes</td>
</tr>
<tr>
<td>8204h</td>
<td>Pin Count</td>
<td>See Table D-3</td>
<td>Pins</td>
</tr>
</tbody>
</table>

**Note 1:** These locations are read-only.

**Note 2:** Erase size is the minimum erasable unit in the PFM, expressed as rows.

**Note 3:** Total device Flash memory capacity is (row_size * num_rows).

### TABLE D-2: NUMBER OF USER ROWS

<table>
<thead>
<tr>
<th>Part Name</th>
<th>Memory Size</th>
<th>Number of User Rows</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIC16(L)F19195</td>
<td>8k</td>
<td>256</td>
</tr>
<tr>
<td>PIC16(L)F19196</td>
<td>16k</td>
<td>512</td>
</tr>
<tr>
<td>PIC16(L)F19197</td>
<td>32k</td>
<td>1024</td>
</tr>
</tbody>
</table>

### TABLE D-3: PIN COUNT

<table>
<thead>
<tr>
<th>Part Number</th>
<th>Pin Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIC16(L)F19195</td>
<td>64</td>
</tr>
<tr>
<td>PIC16(L)F19196</td>
<td>64</td>
</tr>
<tr>
<td>PIC16(L)F19197</td>
<td>64</td>
</tr>
</tbody>
</table>
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 unless otherwise stated.

Trademarks

The Microchip name and logo, the Microchip logo, AnyRate, AVR, AVR logo, AVR Freaks, BeaconThings, BitCloud, CryptoMemory, CryptoRF, dsPIC, FlashFlex, flexPWR, Helio, JukeBox, KEELOQ, KEELOQ logo, Kleer, LANCcheck, LINK MD, maXStylus, maXTouch, MediaLB, megaAVR, MOST, MOST logo, MPLAB, OptoLyzer, PIC, picoPower, PICSTART, PIC32 logo, Prochip Designer, QTouch, RightTouch, SAM-BA, SpyNIC, SST, SST Logo, SuperFlash, tinyAVR, UNI/O, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.


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.

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
ISO/TS 16949
# Worldwide Sales and Service

## AMERICAS
**Corporate Office**
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
**Technical Support:**
http://www.microchip.com/support
**Web Address:**
www.microchip.com

**Atlanta**
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

**Austin, TX**
Tel: 512-257-3370

**Boston**
Westborough, MA
Tel: 508-378-5000
Fax: 508-378-5001

**Chicago**
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

**Dallas**
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

**Detroit**
Novi, MI
Tel: 248-848-4000

**Houston, TX**
Tel: 281-894-5983

**Indianapolis**
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453
Tel: 317-536-2380

**Los Angeles**
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608
Tel: 951-273-7800

**Raleigh, NC**
Tel: 919-844-7510

**New York, NY**
Tel: 631-435-6000

**San Jose, CA**
Tel: 408-735-9110
Tel: 408-436-4270

**Canada - Toronto**
Tel: 905-695-1980
Fax: 905-695-2078

## ASIA/PACIFIC
**Australia - Sydney**
Tel: 61-2-9886-6733

**China - Beijing**
Tel: 86-10-8569-7000

**China - Chengdu**
Tel: 86-28-8665-5511

**China - Chongqing**
Tel: 86-23-8980-9588

**China - Dongguan**
Tel: 86-769-8702-9880

**China - Guangzhou**
Tel: 86-20-8755-8029

**China - Hangzhou**
Tel: 86-571-8792-8115

**China - Hong Kong SAR**
Tel: 852-2943-5100

**China - Nanjing**
Tel: 86-25-8473-2460

**China - Qingdao**
Tel: 86-532-8502-7355

**China - Shanghai**
Tel: 86-21-3326-8000

**China - Shenyang**
Tel: 86-24-2314-2829

**China - Shenzhen**
Tel: 86-755-8864-2200

**China - Suzhou**
Tel: 86-186-6233-1526

**China - Wuhan**
Tel: 86-27-5980-5300

**China - Xian**
Tel: 86-29-8833-7252

**China - Xiamen**
Tel: 86-592-2386138

**China - Zhuhai**
Tel: 86-756-8210040

**India - Bangalore**
Tel: 91-80-3090-4444

**India - New Delhi**
Tel: 91-11-4160-8631

**India - Pune**
Tel: 91-20-4121-0141

**Japan - Osaka**
Tel: 81-6-6152-7160

**Japan - Tokyo**
Tel: 81-3-6880-3770

**Korea - Daegu**
Tel: 82-53-744-4301

**Korea - Seoul**
Tel: 82-2-554-7200

**Malaysia - Kuala Lumpur**
Tel: 60-3-7651-7906

**Malaysia - Penang**
Tel: 60-4-227-8870

**Philippines - Manila**
Tel: 63-2-634-9065

**Singapore**
Tel: 65-6334-8870

**Taiwan - Hsin Chu**
Tel: 886-3-577-8366

**Taiwan - Kaohsiung**
Tel: 886-7-213-7830

**Taiwan - Taipei**
Tel: 886-2-2508-8600

**Thailand - Bangkok**
Tel: 66-2-694-1351

**Vietnam - Ho Chi Minh**
Tel: 84-28-5448-2100

## EUROPE
**Austria - Wels**
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

**Denmark - Copenhagen**
Tel: 45-4450-2828
Fax: 45-4485-2829

**Finland - Espoo**
Tel: 358-9-4520-820

**France - Paris**
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

**Germany - Garching**
Tel: 49-8931-9700

**Germany - Haan**
Tel: 49-2129-376400

**Germany - Heilbronn**
Tel: 49-7131-67-3636

**Germany - Karlsruhe**
Tel: 49-721-625370

**Germany - Munich**
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

**Germany - Rosenheim**
Tel: 49-8031-354-560

**Israel - Ra’anana**
Tel: 972-9-744-7705

**Italy - Milan**
Tel: 39-0331-742611
Fax: 39-0331-466781

**Italy - Padova**
Tel: 39-049-762528

**Netherlands - Drunen**
Tel: 31-416-690399
Fax: 31-416-690340

**Norway - Trondheim**
Tel: 47-7289-7561

**Poland - Warsaw**
Tel: 48-22-3325737

**Romania - Bucharest**
Tel: 40-21-407-87-50

**Spain - Madrid**
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

**Sweden - Gothenberg**
Tel: 46-31-704-60-40

**Sweden - Stockholm**
Tel: 46-8-5090-4654

**UK - Wokingham**
Tel: 44-118-921-5800
Fax: 44-118-921-5820