PIC16(L)F153XX Memory Programming Specification

1.0 OVERVIEW

This programming specification describes an SPI-compatible programming method for the PIC16(L)F153XX 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 1-1: PROGRAMMING TERMS

Term	Definition
Programmed Cell	A memory cell with a logic '0'
Erased Cell	A memory cell with a logic '1'
Erase	Change memory cell from a '0' to a '1'
Write	Change memory cell from a '1' to a '0'
Program	Generic erase and/or write

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 3.5 "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 3.5 "Electrical Specifications"**.

1.2.3 MULTI-WORD PROGRAMMING INTERFACE

Program Flash Memory (PFM) panels include a 32-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 3.5 "Electrical Specifications"**.

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 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.

1.4 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 1-2: PIN DESCRIPTIONS DURING PROGRAMMING

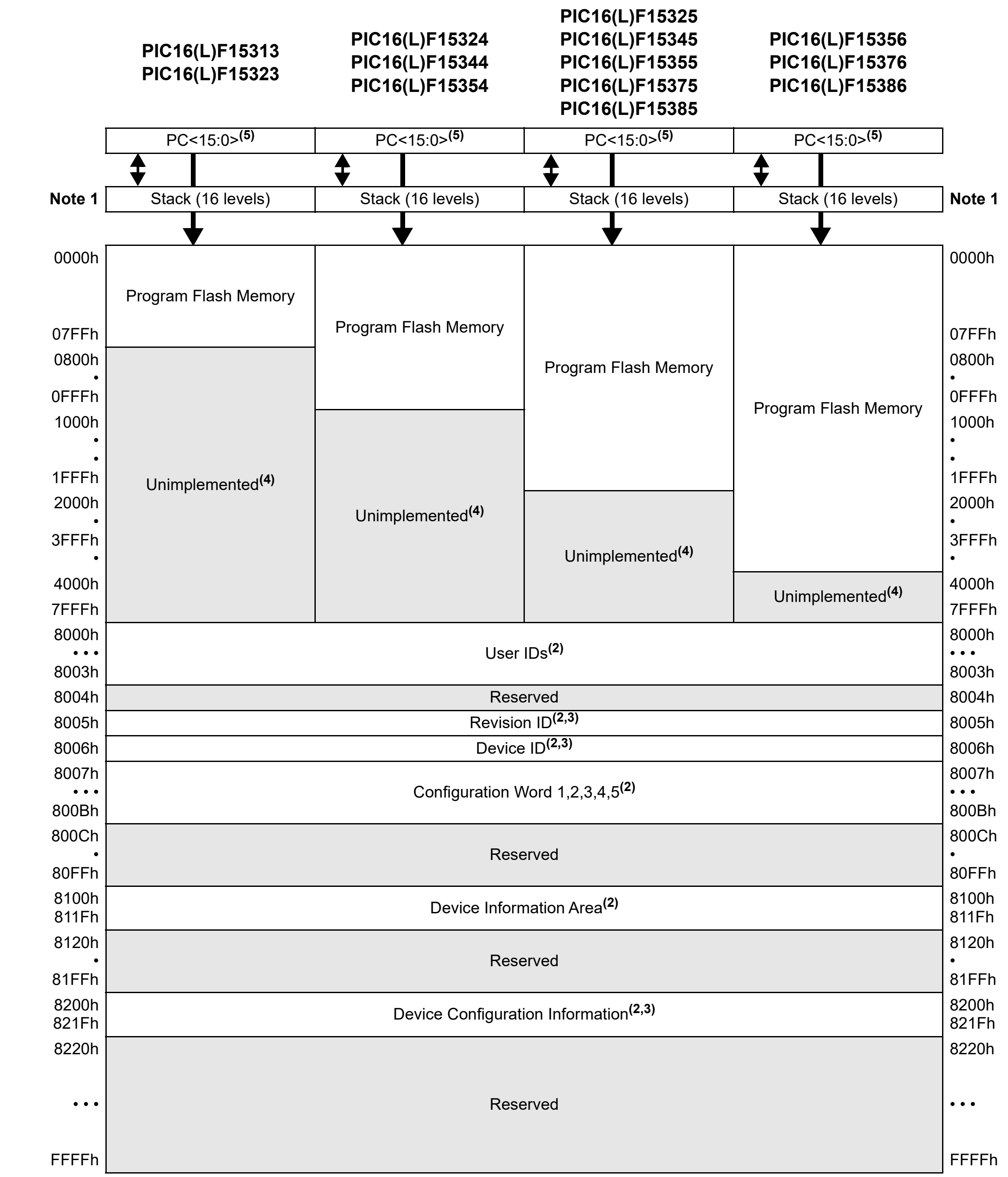
Pin Name	During Programming		
	Function	Pin Type	Pin Description
ICSPCLK	ICSPCLK	I	Clock Input – Schmitt Trigger Input
ICSPDAT	ICSPDAT	I/O	Data Input/Output – Schmitt Trigger Input
MCLR/VPP	Program/Verify mode	I(1)	Program Mode Select
VDD	VDD	P	Power Supply
VSS	VSS	P	Ground

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



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

R	R	R	R	R	R	R	R	R	R	R	R	R	R
1	1	DEV11	DEV10	DEV9	DEV8	DEV7	DEV6	DEV5	DEV4	DEV3	DEV2	DEV1	DEV0
bit 13													bit 0

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

R	R	R	R	R	R	R	R	R	R	R	R	R	R
1	0	MJRREV<5:0>						MNRREV<5:0>					
bit 13													bit 0

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. $\overline{\text{MCLR}}$/VPP pin function is $\overline{\text{MCLR}}$. MCLRE Configuration bit is ignored.
- 0 = OFF – HV on $\overline{\text{MCLR}}$/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. **$\overline{\text{CP}}$: 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 C-1 in **Appendix C: "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 $\overline{\text{MCLR}}$ from 0V to VIHH.
3. Raise the voltage on VDD from 0V to the desired operating voltage.

The VPP-first entry prevents the device from executing code prior to entering Program/Verify mode. For example, when the Configuration Word has $\overline{\text{MCLR}}$ disabled (MCLRE = 0), the Power-up Timer is disabled ($\overline{\text{PWRTE}}$ = 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 VDD from 0V to the desired operating voltage.
3. Raise the voltage on $\overline{\text{MCLR}}$ 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 $\overline{\text{MCLR}}$ from VIHH to VIL. 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

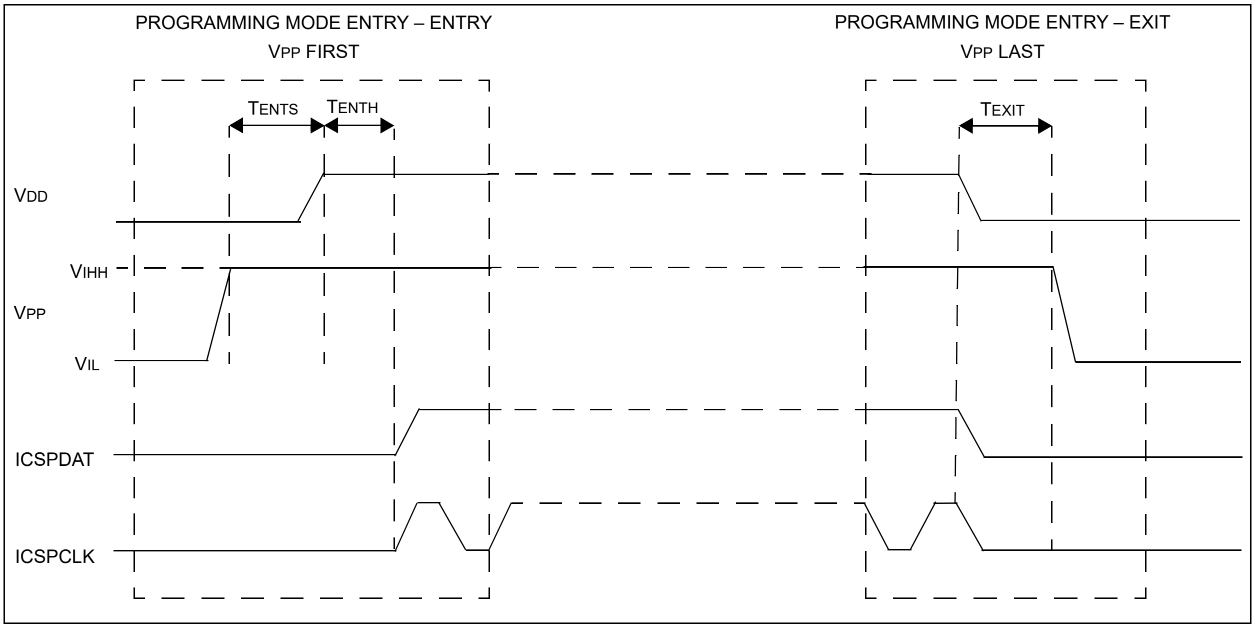
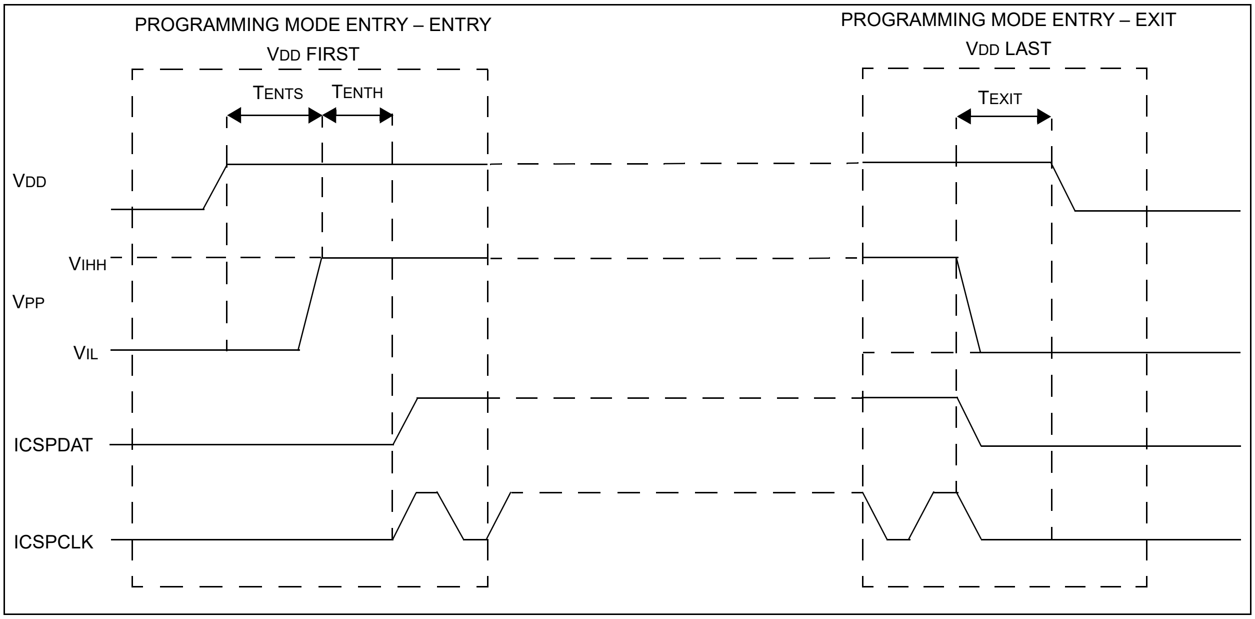


FIGURE 3-2: PROGRAMMING ENTRY AND EXIT MODES – VDD FIRST AND LAST



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)

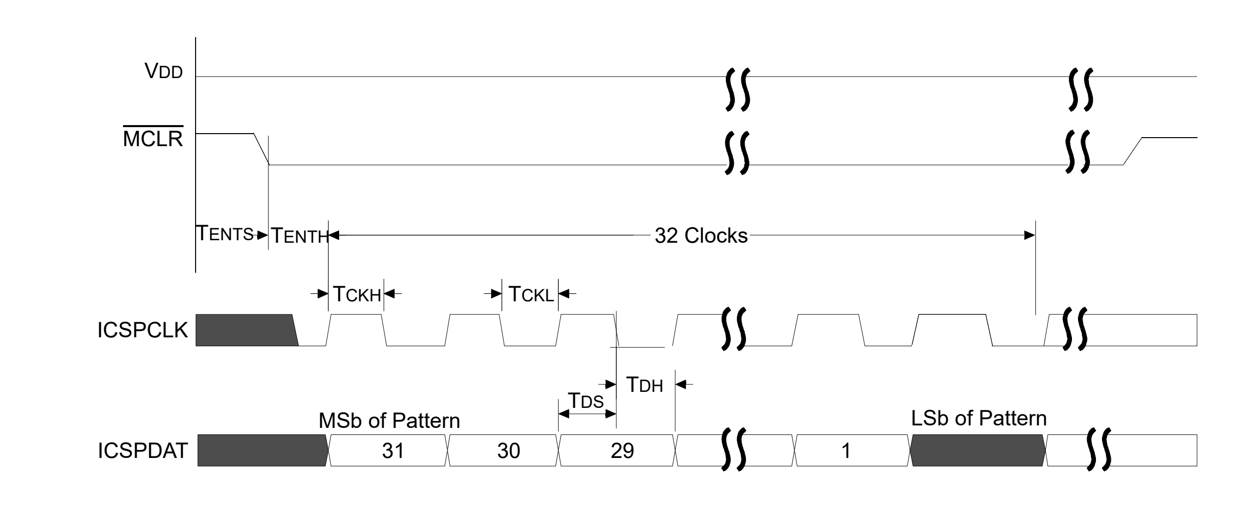
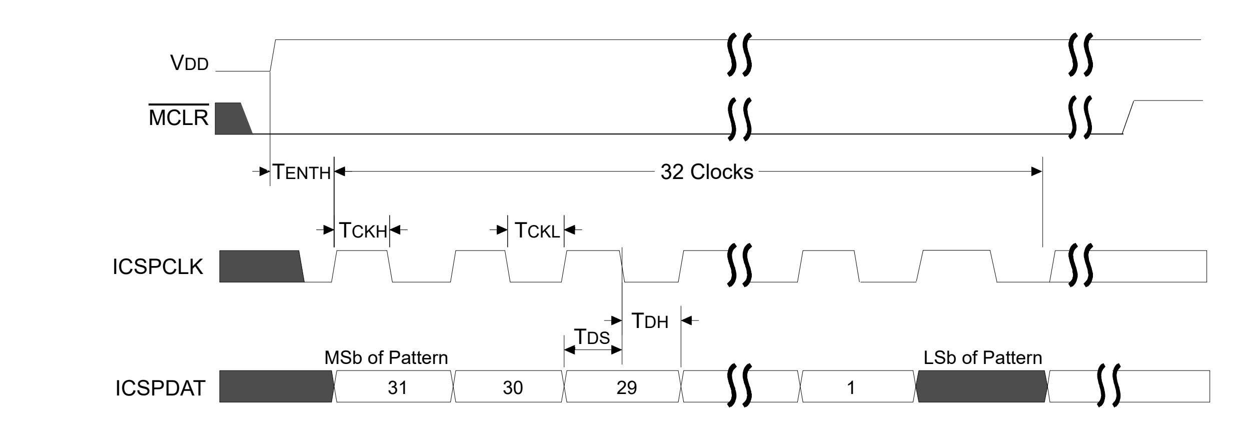


FIGURE 3-4: LVP ENTRY (POWERED)



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.

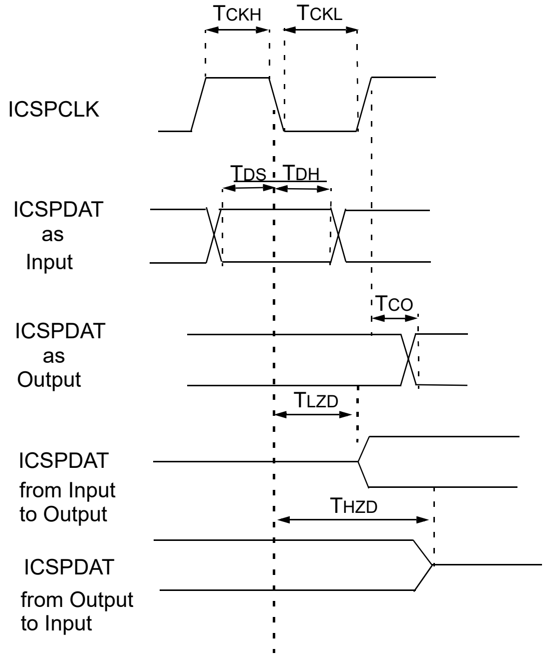
When the programming host device issues an 8-bit command byte to the microcontroller, the host should wait a minimum amount of delay (see Table 3-1) prior to sending any additional clock pulses (associated with either a 24-bit data payload field or the next command byte).

TABLE 3-1: ICSP™ COMMAND SET SUMMARY

Command Name	Command Value		Payload Expected	Delay after Command	Data/Note
	Binary (MSb … LSb)	Hex			
Load PC Address	`1000 000x`	80	Yes	TDLY	PC = payload value
Bulk Erase Program Memory	`0001 1000`	18	No	TERAB	Depending on the current value of the PC, one or more memory regions.
Row Erase Program Memory	`1111 0000`	F0	No	TERAR	The row addressed by the MSbs of the PC is erased; LSbs are ignored.
Load Data for NVM	`0000 00J0`	00/02	Yes	TDLY	J = 1: PC = PC + 1 after writing J = 0: PC is unchanged
Read Data from NVM	`1111 11J0`	FE/FC	Yes	TDLY	J = 1: PC = PC + 1 after reading J = 0: PC is unchanged
Increment Address	`1111 1000`	F8	No	TDLY	PC = PC + 1
Begin Internally Timed Programming	`1110 0000`	E0	No	TPINT	Commits latched data to NVM (self timed)
Begin Externally Timed Programming	`1100 0000`	C0	No	TPEXT	Commits latched data to NVM (externally timed). After TPEXT, “End Externally Timed Programming” command must be issued.
End Externally Timed Programming	`1000 0010`	82	No	TDIS	Should be issued within required time delay (TPEXT) after “Begin Externally Timed Programming” command.

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.

FIGURE 3-5: CLOCK AND DATA TIMING



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)

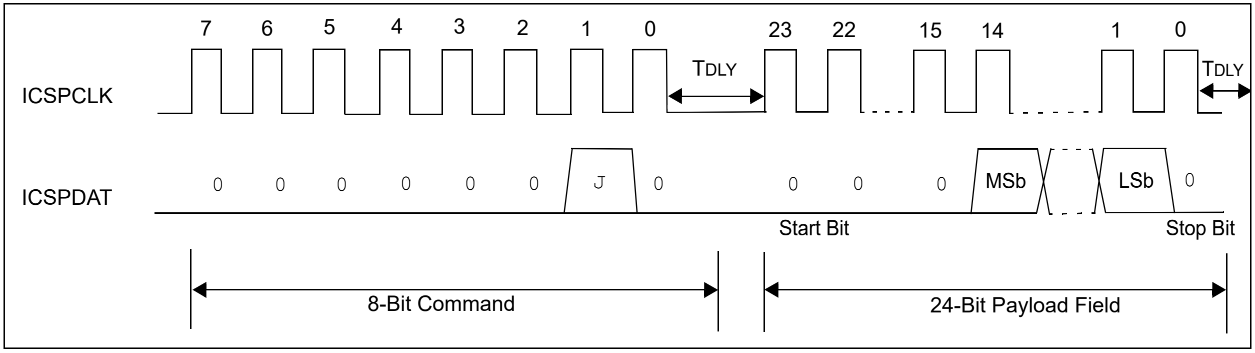
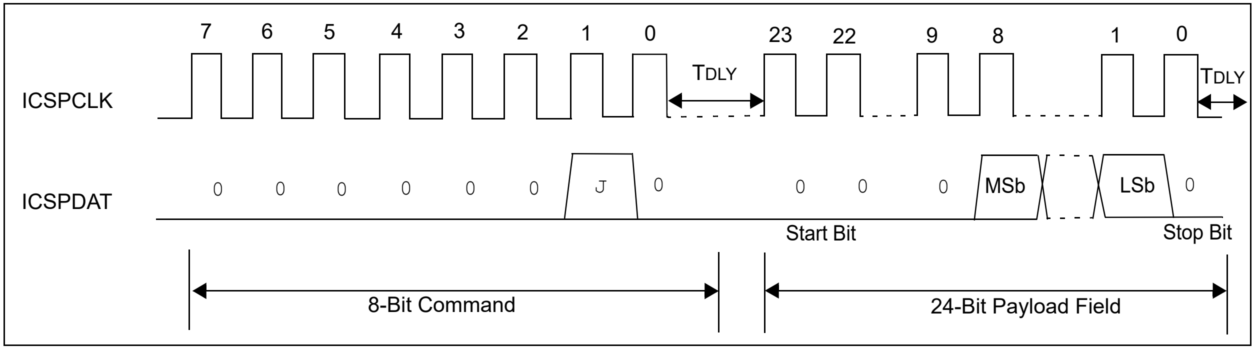


FIGURE 3-7: LOAD DATA FOR NVM (EEPROM, IF AVAILABLE)



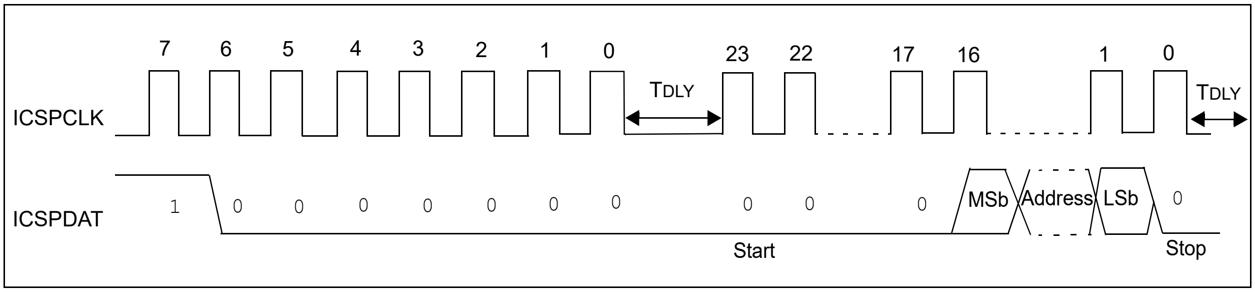
3.1.3.2 Read Data from NVM

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 ($\overline{CP}$ = `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



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

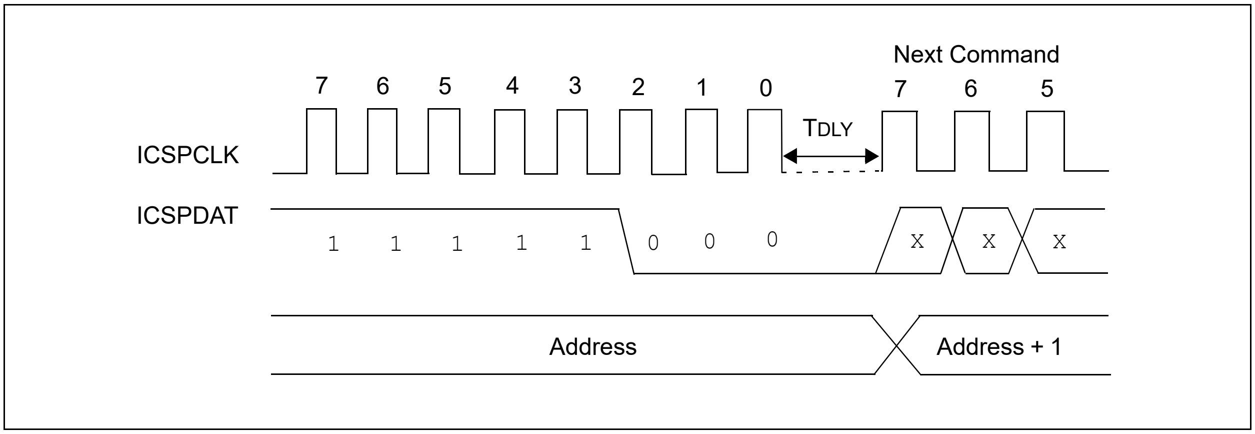


FIGURE 3-10: READ DATA FROM NVM (PFM OR CONFIGURATION WORDS)

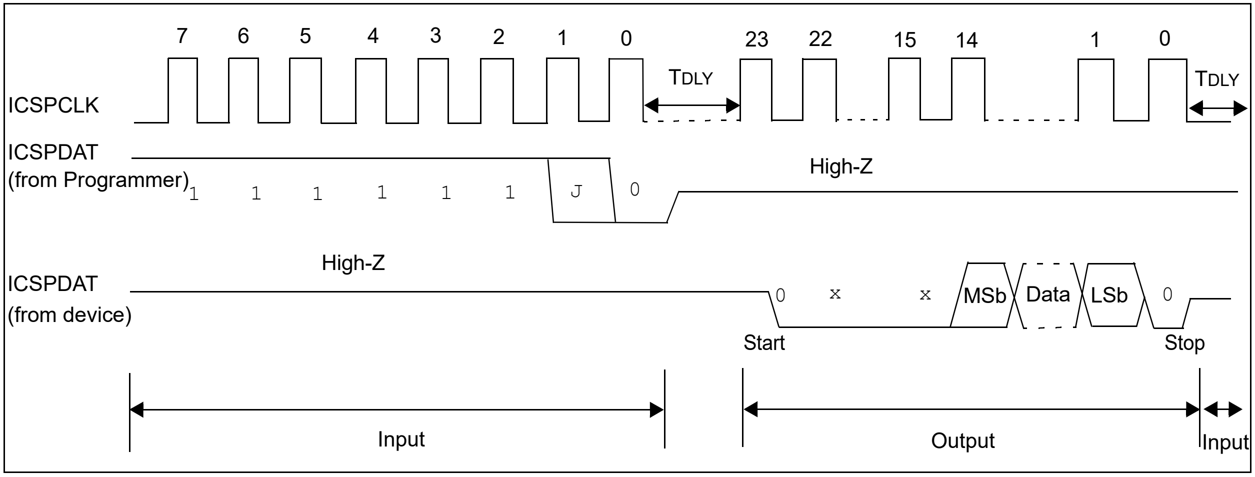
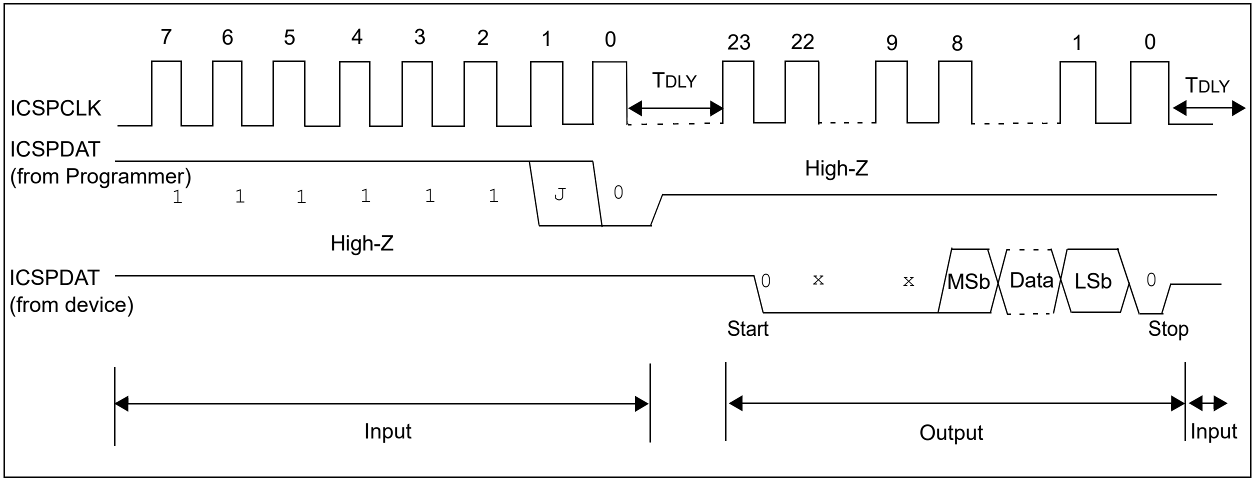


FIGURE 3-11: READ DATA FROM NVM (EEPROM, IF AVAILABLE)

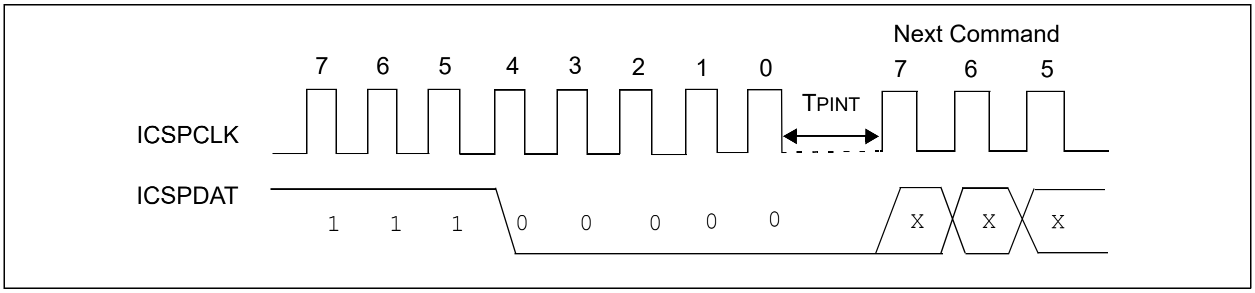


3.1.3.5 Begin Internally Timed Programming

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’.

FIGURE 3-12: BEGIN INTERNALLY TIMED PROGRAMMING

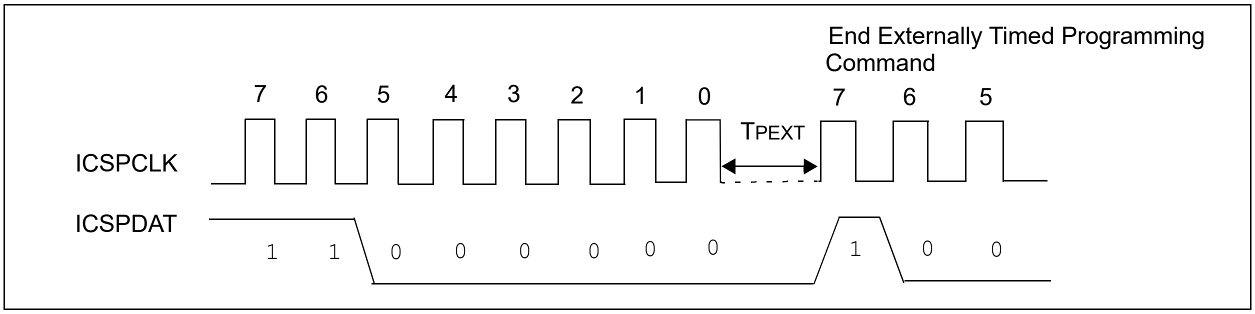


3.1.3.6 Begin Externally Timed Programming

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.

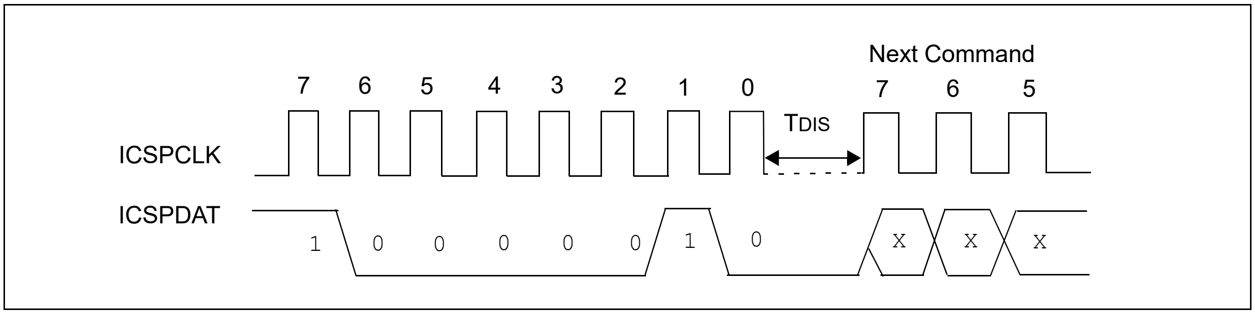
FIGURE 3-13: BEGIN EXTERNALLY TIMED PROGRAMMING



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

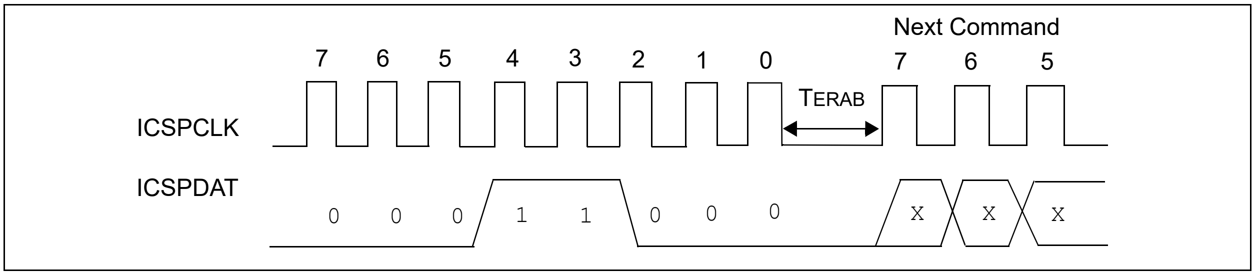
The Bulk Erase Memory command performs different functions dependent on the current PC address. The Bulk Erase command affects specific portions of the memory depending on the initial value of the Program Counter. Whenever a Bulk Erase command is executed, the device will erase all bytes within the regions 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

Address	Area(s) Erased	
	$\overline{\text{CP}}$ = 1	**$\overline{\text{CP}}$ = 0**
0000h-7FFFh	User Flash	User Flash
	Configuration Words	Configuration Words
8000h-80FDh	User Flash	User Flash
	Configuration Words	Configuration Words
	User ID words	User ID words
80FEh-80FFh	User Flash	User Flash
8100h-E7FFh	No Operation	No Operation
E800h-FFFFh	User Flash	User Flash
	Configuration Words	Configuration Words
	User ID words	User ID words

FIGURE 3-15: BULK ERASE MEMORY

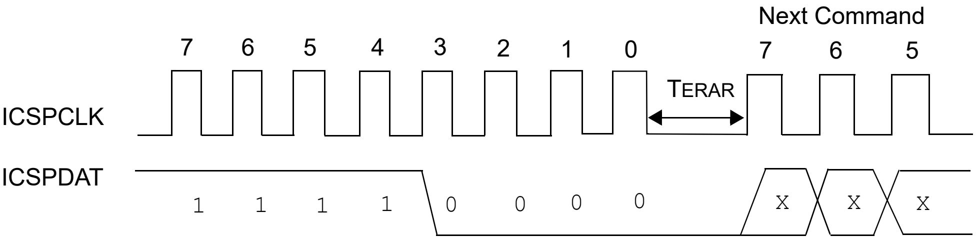


3.1.3.9 Row Erase Memory

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 $\overline{\text{CP}}$ 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 32 and the row size (number of 14-bit latches) for the write operation is 32.

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

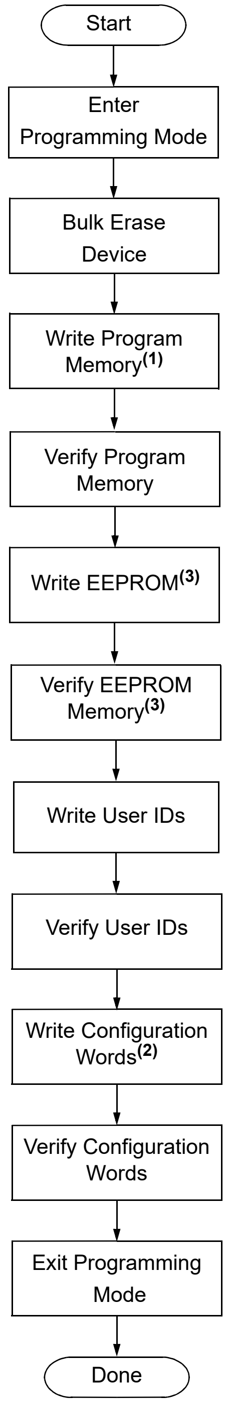
The device uses internal latches to temporarily store the 14-bit words used for programming. The data latches allow the user to program a full row with a single Begin Internally Timed Programming or Begin Externally Timed Programming command. The Load Data for NVM command is used to load a single data 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 32-latch device will result in data being written to 0020h-003Fh.

If more than the maximum number of latches are written without a Begin 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 and EEPROM memory regions are programmed one row (32 words) at a time (Figure 3-20), while the User ID and Configuration words are programmed one word at a time (Figure 3-19).
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

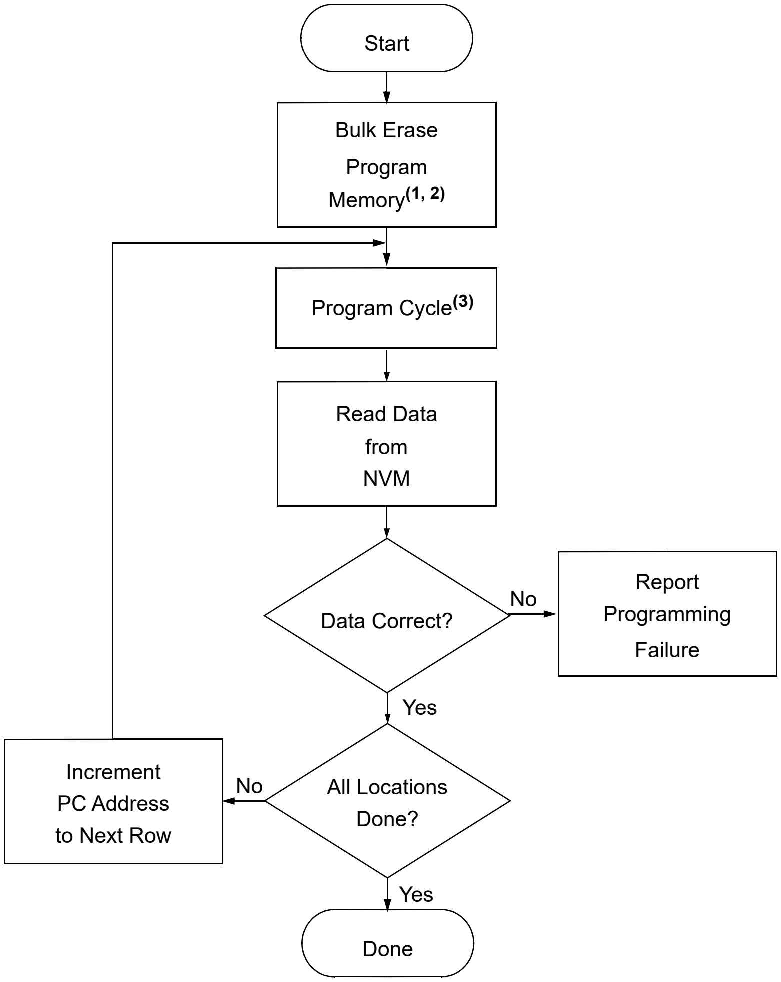


Note 1: See Figure 3-8.

2: See Figure 3-16.

3: The PIC16(L)F153XX 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
All Locations Done?
No
Increment PC Address to Next Row
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

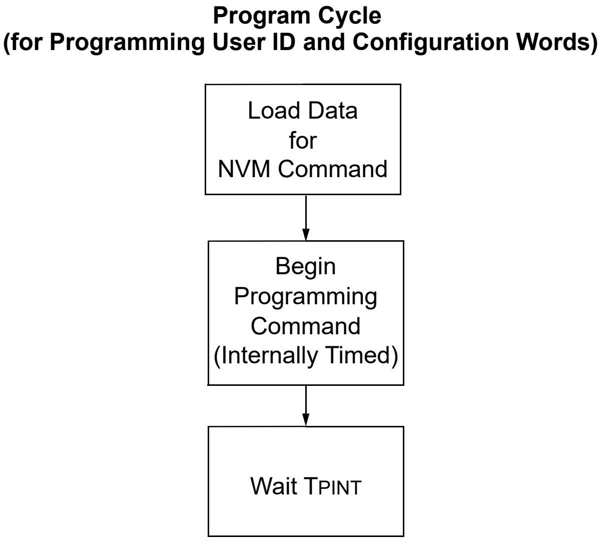

Program Cycle
(for Programming User ID and Configuration Words)
Load Data for NVM Command
Begin Programming Command (Internally Timed)
Wait TPINT

FIGURE 3-20: MULTIPLE-WORD PROGRAM CYCLE

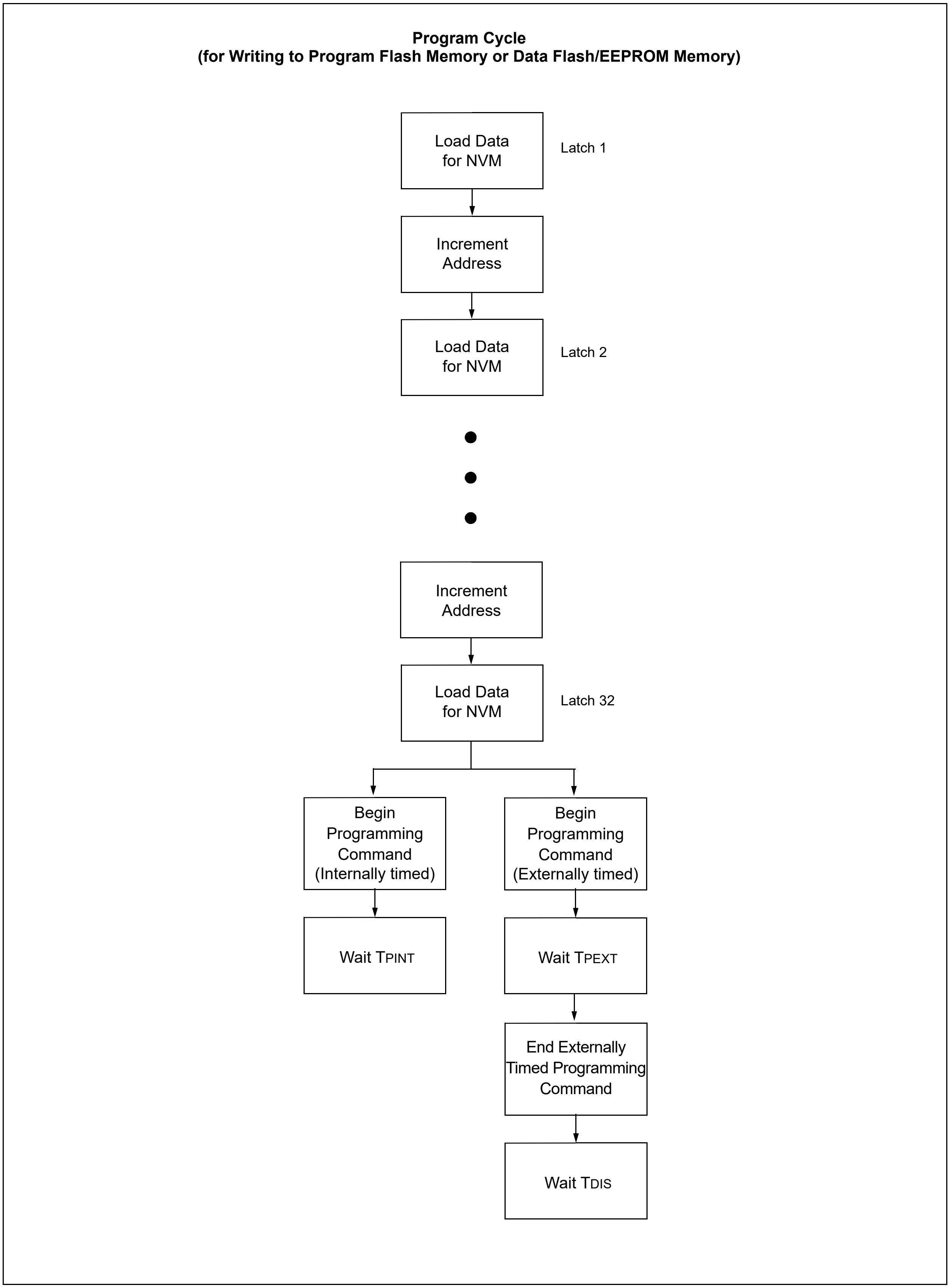
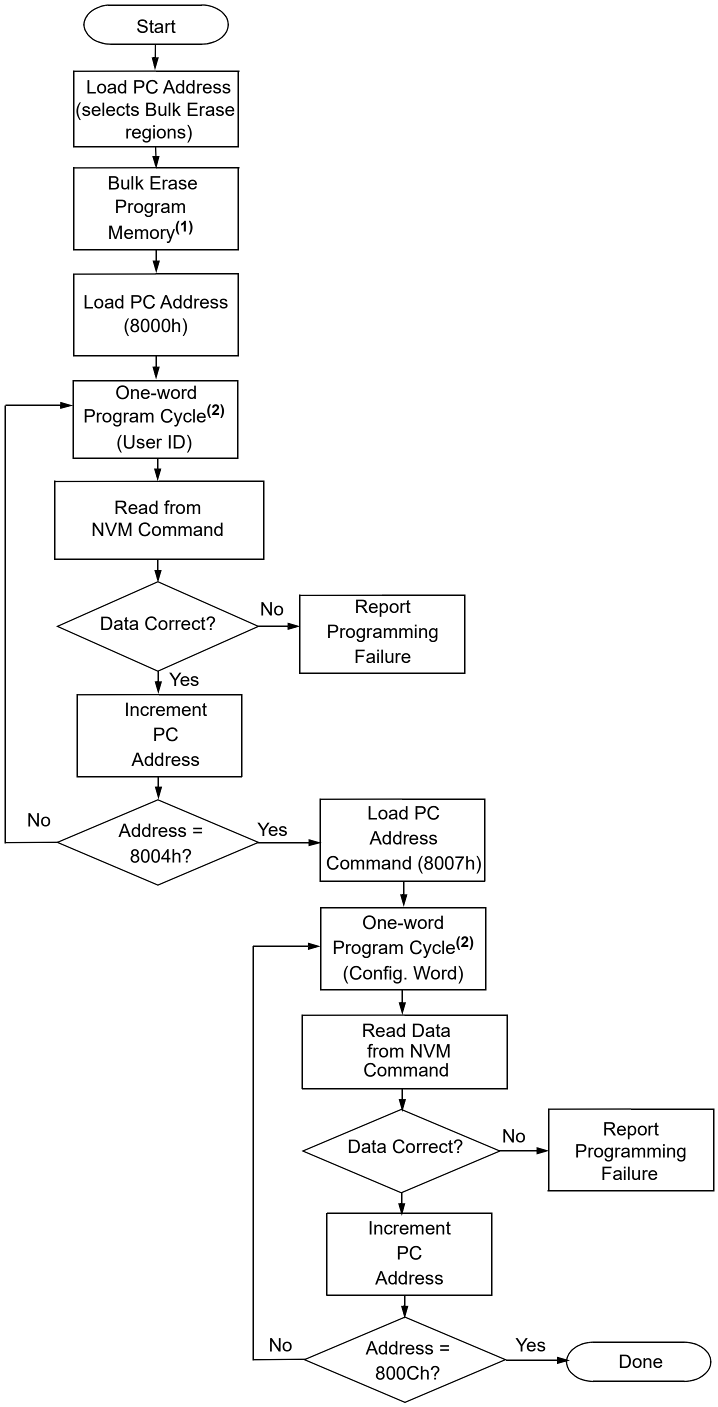

Program Cycle
(for Writing to Program Flash Memory or Data Flash/EEPROM Memory)
Load Data for NVM
Latch 1
Increment Address
Load Data for NVM
Latch 2
Increment Address
Load Data for NVM
Latch 32
Begin Programming Command (Internally timed)
Wait TPINT
Begin Programming Command (Externally timed)
Wait TPEXT
End Externally Timed Programming Command
Wait TDIS

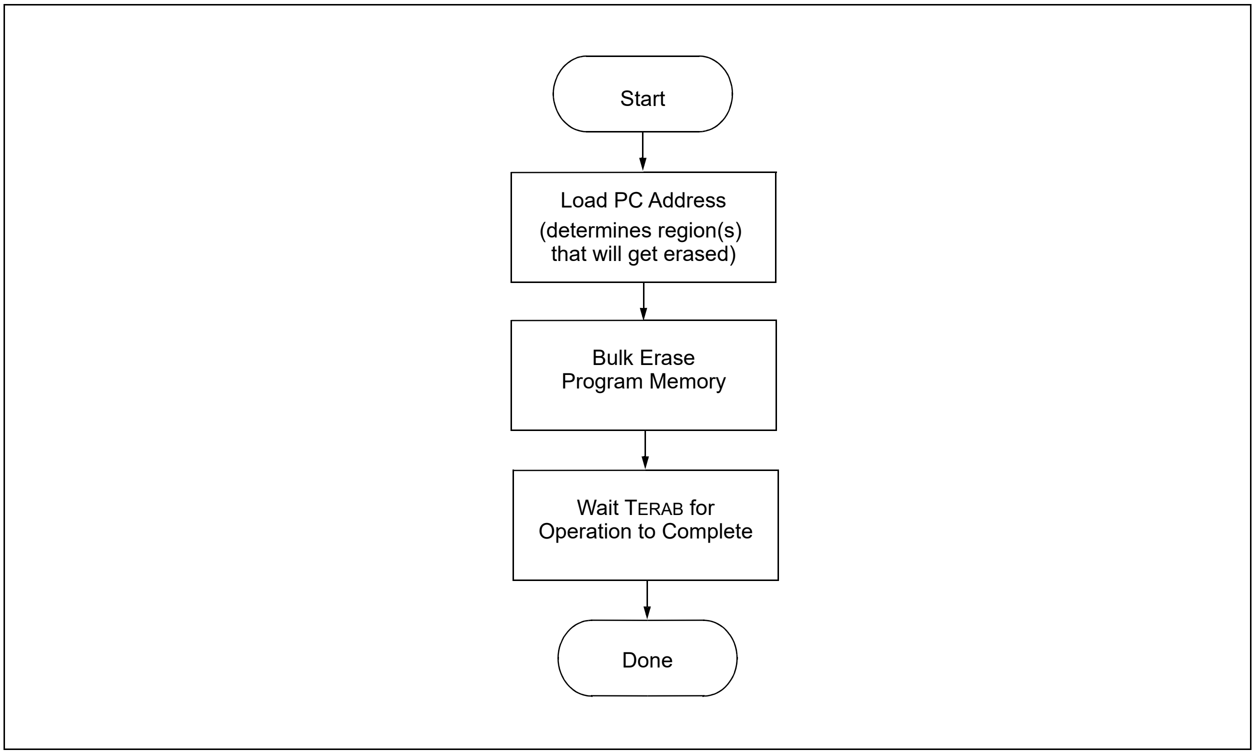
FIGURE 3-21: USER ID AND CONFIGURATION MEMORY PROGRAM FLOWCHART


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?
No
Report Programming Failure
Yes
Increment PC Address
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
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.

FIGURE 3-22: BULK ERASE FLOWCHART


Start
Load PC Address (determines region(s) that will get erased)
Bulk Erase Program Memory
Wait TERAB for Operation to Complete
Done

3.3 Code Protection

Code protection is controlled using the $\overline{\text{CP}}$ 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 $\overline{\text{CP}}$ 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).

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.

3.4.1 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.

3.4.2 DEVICE ID

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

3.4.3 CHECKSUM COMPUTATION

The checksum is calculated by two different methods dependent on the setting of the $\overline{\text{CP}}$ Configuration bit.

3.4.3.1 Program Code Protection Disabled

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 (e.g., 0FFFh). 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)F153XX Device ID, Checksums and Pinout Descriptions”**).

3.4.3.2 Program Code Protection Enabled

When the MPLABX® 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’.

3.5 Electrical Specifications

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

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

AC/DC CHARACTERISTICS			**Standard Operating Conditions** Production tested at 25°C				
Sym.	**Characteristics**		**Min.**	**Typ.**	**Max.**	**Units**	**Conditions/Comments**
	Programming Supply Voltages and Currents						
VDD	Supply Voltage (VDDMIN(1), VDDMAX)	PICXXLF1XXXX	1.80	—	3.60	V	
		PICXXF1XXXX	2.30	—	5.50	V	
VPEW	Read/Write and Row Erase operations		VDDMIN	—	VDDMAX	V	
VBE	Bulk Erase operations		VBOR(2)	—	VDDMAX	V	
IDDI	Current on VDD, Idle		—	—	1.0	mA	
IDDP	Current on VDD, Programming		—	—	5.0	mA	
	VPP						
IPP	Current on $\overline{\text{MCLR}}$/VPP		—	—	600	μA	
VIHH	High Voltage on $\overline{\text{MCLR}}$/VPP for Program/Verify Mode Entry		8.0	—	9.0	V	
TVHHR	$\overline{\text{MCLR}}$ Rise Time (VIL to VIHH) for Program/Verify Mode Entry		—	—	1.0	μs	
	I/O pins						
VIH	(ICSPCLK, ICSPDAT, $\overline{\text{MCLR}}$/VPP) Input High Level		0.8 VDD	—	—	V	
VIL	(ICSPCLK, ICSPDAT, $\overline{\text{MCLR}}$/VPP) Input Low Level		—	—	0.2 VDD	V	
VOH	ICSPDAT Output High Level		VDD-0.7 VDD-0.7 VDD-0.7	—	—	V	IOH = -3.5 mA, VDD = 5V IOH = -3 mA, VDD = 3.3V IOH = -1 mA, VDD = 1.8V
VOL	ICSPDAT Output Low Level		—	—	VSS+0.6 VSS+0.6 VSS+0.6	V	IOL = 8 mA, VDD = 5V IOL = 6 mA, VDD = 3.3V IOL = 1.8 mA, VDD = 1.8V
	Programming Mode Entry and Exit						
TENTS	Programing Mode Entry Setup Time: ICSPCLK, ICSPDAT Setup Time Before VDD or $\overline{\text{MCLR}}$↑		100	—	—	ns	
TENTH	Programing Mode Entry Hold Time: ICSPCLK, ICSPDAT Hold Time after VDD or $\overline{\text{MCLR}}$↑		250	—	—	μs	
	Serial Program/Verify						
TCKL	Clock Low Pulse Width		100	—	—	ns	
TCKH	Clock High Pulse Width		100	—	—	ns	
TDS	Data in SETUP TIME before Clock↓		100	—	—	ns	
TDH	Data in HOLD TIME after Clock↓		100	—	—	ns	
TCO	Clock↑ to DATA OUT VALID (during a Read Data Command)		0	—	80	ns	

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.

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).

3: Externally timed writes are not supported for Configuration bits.

TABLE 3-3: AC/DC CHARACTERISTICS TIMING REQUIREMENTS FOR PROGRAM/VERIFY MODE (CONTINUED)

AC/DC CHARACTERISTICS		Standard Operating Conditions Production tested at 25°C				
Sym.	Characteristics	Min.	Typ.	Max.	Units	Conditions/Comments
TLZD	Clock↓ to Data Low-impedance (during a Read Data Command)	0	—	80	ns	
THZD	Clock↓ to Data High-impedance (during a Read Data command)	0	—	80	ns	
TDLY	Data Input not Driven To Next Clock Input (delay required between command/data or command/command)	1.0	—	—	µs	
TERAB	Bulk Erase Cycle Time	—	—	8.4	ms	PIC16(L)F153XX devices
TERAR	Row Erase Cycle Time	—	—	2.8	ms	
TPINT	Internally Timed Programming Operation Time	— —	— —	2.8 5.6	ms ms	Program memory Configuration Words
TPEXT	Delay Required between Begin Externally Timed Programming and End Externally Timed Programming Commands	1.0	—	2.1	ms	(**Note 3**)
TDIS	Delay Required after End Externally Timed Programming Command	300	—	—	µs	
TEXIT	Time Delay when Exiting Program/Verify Mode	1	—	—	µs	

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.

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).

3: Externally timed writes are not supported for Configuration bits.

APPENDIX A: REVISION HISTORY

Revision D (3/2021)

Updated Table B-1. Updated Example B-1, B-2, B-3, and B-4. Other minor editorial corrections.

Revision C (4/2017)

Updated Table B-1.

Revision B (4/2016)

Updated Section 3.4.3.2. Updated Table B-1. Updated Example B-3. Added Note under Example B-4.

Revision A (3/2016)

Initial release.

APPENDIX B: PIC16(L)F153XX DEVICE ID, CHECKSUMS AND PINOUT DESCRIPTIONS

TABLE B-1: DEVICE IDs AND CHECKSUMS

Device	Device ID	Config.1		Config.2		Config.3		Config.4		Config.5			Checksum			
													Unprotected		Code-protected	
		Word (HEX)	Mask (HEX)	Word (HEX)	Mask (HEX)	Word (HEX)	Mask (HEX)	Word (HEX)	Mask (HEX)	Word (unprotected) (HEX)	Word (protected) (HEX)	Mask (HEX)	Blank (HEX)	00AAh First and Last (HEX)	Blank (HEX)	00AAh First and Last (HEX)
PIC16F15313	30BE	3FFF	2977	3FFF	3EE3	3FFF	3F7F	3FFF	2B9F	3FFF	3FFE	0001	CB79	4CCF	A2F1	2447
PIC16LF15313	30BF	3FFF	2977	3FFF	3EE3	3FFF	3F7F	3FFF	2B9F	3FFF	3FFE	0001	CB79	4CCF	A2F1	2447
PIC16F15323	30C0	3FFF	2977	3FFF	3EE3	3FFF	3F7F	3FFF	2B9F	3FFF	3FFE	0001	CB79	4CCF	A2F1	2447
PIC16LF15323	30C1	3FFF	2977	3FFF	3EE3	3FFF	3F7F	3FFF	2B9F	3FFF	3FFE	0001	CB79	4CCF	A2F1	2447
PIC16F15324	30C2	3FFF	2977	3FFF	3EE3	3FFF	3F7F	3FFF	2B9F	3FFF	3FFE	0001	C379	44CF	9AF1	1C47
PIC16LF15324	30C3	3FFF	2977	3FFF	3EE3	3FFF	3F7F	3FFF	2B9F	3FFF	3FFE	0001	C379	44CF	9AF1	1C47
PIC16F15344	30C4	3FFF	2977	3FFF	3EE3	3FFF	3F7F	3FFF	2B9F	3FFF	3FFE	0001	C379	44CF	9AF1	1C47
PIC16LF15344	30C5	3FFF	2977	3FFF	3EE3	3FFF	3F7F	3FFF	2B9F	3FFF	3FFE	0001	C379	44CF	9AF1	1C47
PIC16F15325	30C6	3FFF	2977	3FFF	3EE3	3FFF	3F7F	3FFF	2B9F	3FFF	3FFE	0001	B379	34CF	8AF1	0C47
PIC16LF15325	30C7	3FFF	2977	3FFF	3EE3	3FFF	3F7F	3FFF	2B9F	3FFF	3FFE	0001	B379	34CF	8AF1	0C47
PIC16F15345	30C8	3FFF	2977	3FFF	3EE3	3FFF	3F7F	3FFF	2B9F	3FFF	3FFE	0001	B379	34CF	8AF1	0C47
PIC16LF15345	30C9	3FFF	2977	3FFF	3EE3	3FFF	3F7F	3FFF	2B9F	3FFF	3FFE	0001	B379	34CF	8AF1	0C47
PIC16F15354	30AC	3FFF	2977	3FFF	3EE3	3FFF	3F7F	3FFF	2B9F	3FFF	3FFE	0001	C379	44CF	9AF1	1C47
PIC16LF15354	30AD	3FFF	2977	3FFF	3EE3	3FFF	3F7F	3FFF	2B9F	3FFF	3FFE	0001	C379	44CF	9AF1	1C47
PIC16F15355	30AE	3FFF	2977	3FFF	3EE3	3FFF	3F7F	3FFF	2B9F	3FFF	3FFE	0001	B379	34CF	8AF1	0C47
PIC16LF15355	30AF	3FFF	2977	3FFF	3EE3	3FFF	3F7F	3FFF	2B9F	3FFF	3FFE	0001	B379	34CF	8AF1	0C47
PIC16F15356	30B0	3FFF	2977	3FFF	3EE3	3FFF	3F7F	3FFF	2B9F	3FFF	3FFE	0001	9379	14CF	6AF1	EC47
PIC16LF15356	30B1	3FFF	2977	3FFF	3EE3	3FFF	3F7F	3FFF	2B9F	3FFF	3FFE	0001	9379	14CF	6AF1	EC47
PIC16F15375	30B2	3FFF	2977	3FFF	3EE3	3FFF	3F7F	3FFF	2B9F	3FFF	3FFE	0001	B379	34CF	8AF1	0C47
PIC16LF15375	30B3	3FFF	2977	3FFF	3EE3	3FFF	3F7F	3FFF	2B9F	3FFF	3FFE	0001	B379	34CF	8AF1	0C47
PIC16F15376	30B4	3FFF	2977	3FFF	3EE3	3FFF	3F7F	3FFF	2B9F	3FFF	3FFE	0001	9379	14CF	6AF1	EC47
PIC16LF15376	30B5	3FFF	2977	3FFF	3EE3	3FFF	3F7F	3FFF	2B9F	3FFF	3FFE	0001	9379	14CF	6AF1	EC47

TABLE B-1: DEVICE IDs AND CHECKSUMS (CONTINUED)

Device	Device ID	Config.1		Config.2		Config.3		Config.4		Config.5			Checksum			
													Unprotected		Code-protected	
		Word (HEX)	Mask (HEX)	Word (HEX)	Mask (HEX)	Word (HEX)	Mask (HEX)	Word (HEX)	Mask (HEX)	Word (unprotected) (HEX)	Word (protected) (HEX)	Mask (HEX)	Blank (HEX)	00AAh First and Last (HEX)	Blank (HEX)	00AAh First and Last (HEX)
PIC16F15385	30B6	3FFF	2977	3FFF	3EE3	3FFF	3F7F	3FFF	2B9F	3FFF	3FFE	0001	B379	34CF	8AF1	0C47
PIC16LF15385	30B7	3FFF	2977	3FFF	3EE3	3FFF	3F7F	3FFF	2B9F	3FFF	3FFE	0001	B379	34CF	8AF1	0C47
PIC16F15386	30B8	3FFF	2977	3FFF	3EE3	3FFF	3F7F	3FFF	2B9F	3FFF	3FFE	0001	9379	14CF	6AF1	EC47
PIC16LF15386	30B9	3FFF	2977	3FFF	3EE3	3FFF	3F7F	3FFF	2B9F	3FFF	3FFE	0001	9379	14CF	6AF1	EC47

EXAMPLE B-1: CHECKSUM COMPUTED WITH PROGRAM CODE PROTECTION DISABLED PIC16F15354, BLANK DEVICE

PIC16F15354	Sum of Memory addresses 0000h-0FFFh	F000h (1000h*3FFFh)
	Configuration Word 1	3FFFh
	Configuration Word 1 mask	2977h
	Configuration Word 2	3FFFh
	Configuration Word 2 mask	3EE3h
	Configuration Word 3	3FFFh
	Configuration Word 3 mask	3F7Fh
	Configuration Word 4	3FFFh
	Configuration Word 4 mask	2F9Fh
	Configuration Word 5 Unprotected	3FFFh
	Configuration Word 5 mask	0001h
	Checksum	= F000h + (3FFFh and 2977h) + (3FFFh and 3EE3h) + (3FFFh and 3F7Fh) + (3FFFh and 2F9Fh) + (3FFFh and 0001h) = F000h + 2977h + 3EE3h + 3F7Fh + 2B9Fh + 0001h = C379h

EXAMPLE B-2: CHECKSUM COMPUTED WITH PROGRAM CODE PROTECTION DISABLED PIC16F15354, 00AAh AT FIRST AND LAST ADDRESS

PIC16F15354	Sum of Memory addresses 0000h-0FFFh	7156h (AAh + (FFEh*3FFFh) + AAh)
	Configuration Word 1	3FFFh
	Configuration Word 1 mask	2977h
	Configuration Word 2	3FFFh
	Configuration Word 2 mask	3EE3h
	Configuration Word 3	3FFFh
	Configuration Word 3 mask	3F7Fh
	Configuration Word 4	3FFFh
	Configuration Word 4 mask	2F9Fh
	Configuration Word 5 Unprotected	3FFFh
	Configuration Word 5 mask	0001h
	Checksum	= 7156h + (3FFFh and 2977h) + (3FFFh and 3EE3h) + (3FFFh and 3F7Fh) + (3FFFh and 2F9Fh) + (3FFFh and 0001h) = 7156h + 2977h + 3EE3h + 3F7Fh + 2B9Fh + 0001h = 44CFh

EXAMPLE B-3: CHECKSUM COMPUTED WITH PROGRAM CODE PROTECTION ENABLED PIC16F15354, BLANK DEVICE

PIC16F15354	Configuration Word 1	3FFFh
	Configuration Word 1 mask	2977h
	Configuration Word 2	3FFFh
	Configuration Word 2 mask	3EE3h
	Configuration Word 3	3FFFh
	Configuration Word 3 mask	3F7Fh
	Configuration Word 4	3FFFh
	Configuration Word 4 mask	2F9Fh
	Configuration Word 5 Unprotected	3FFEh
	Configuration Word 5 mask	0001h
	Sum of User IDs	= (000Ch and 000Fh) << 12 + (0007h and 000Fh) << 8 + (0007h and 000Fh) << 4 + (0009h and 000Fh)
		= C000h + 0700h + 0070h + 0009h
		= C779h
	Checksum	= (3FFFh and 2977h) + (3FFFh and 3EE3h) + (3FFFh and 3F7Fh) + (3FFFh and 2F9Fh) + (3FFEh and 0001h) + C779h
		= 2977h + 3EE3h + 3F7Fh + 2B9Fh + 0000h + C779h
		= 9AF1h

EXAMPLE B-4: CHECKSUM COMPUTED WITH PROGRAM CODE PROTECTION ENABLED PIC16F15354, 00AAh AT FIRST AND LAST ADDRESS

PIC16F15354	Configuration Word 1	3FFFh
	Configuration Word 1 mask	2977h
	Configuration Word 2	3FFFh
	Configuration Word 2 mask	3EE3h
	Configuration Word 3	3FFFh
	Configuration Word 3 mask	3F7Fh
	Configuration Word 4	3FFFh
	Configuration Word 4 mask	2F9Fh
	Configuration Word 5 Unprotected	3FFEh
	Configuration Word 5 mask	0001h
	Sum of User IDs	= (0004h and 000Fh) << 12 + (0008h and 000Fh) << 8 + (000Ch and 000Fh) << 4 + (000Fh and 000Fh)
		= 4000h + 0800h + 00C0h + 000Fh
		= 48CFh
	Checksum	= (3FFFh and 2977h) + (3FFFh and 3EE3h) + (3FFFh and 3F7Fh) + (3FFFh and 2F9Fh) + (3FFEh and 0001h) + 48CFh
		= 2977h + 3EE3h + 3F7Fh + 2B9Fh + 0000h + 48CFh
		= 1C47h

Note: The sum of User IDs shown in examples B-3 and B-4 are the values inserted in the User ID location when the user enables that option in the IDE as explained in **Section 3.4.3.2 “Program Code Protection Enabled”**

TABLE B-2: PROGRAMMING PIN LOCATIONS BY PACKAGE TYPE

Device	Package	Package Code	V_{DD} PIN	V_{SS} PIN	$\overline{\text{MCLR}}$		ICSPCLK		ICSPDAT	
					PIN	PORT	PIN	PORT	PIN	PORT
PIC16(L)F15313	8-pin PDIP	(P)	1	8	4	RA3	6	RA1	7	RA0
	8 pin SOIC	(SO)	1	8	4	RA3	6	RA1	7	RA0
	8-pin DFN	(MH)	1	8	4	RA3	6	RA1	7	RA0
	8-pin MSOP	(MS)	1	8	4	RA3	6	RA1	7	RA0
PIC16(L)F15323	14-pin PDIP	(P)	1	14	4	RA3	12	RA1	13	RA0
	14- pin SOIC	(SO)	1	14	4	RA3	12	RA1	13	RA0
	16-pin QFN	(ML)	16	13	3	RA3	11	RA1	12	RA0
	14-pin TSSOP	(ST)	1	14	4	RA3	12	RA1	13	RA0
PIC16(L)F15324	14-pin PDIP	(P)	1	14	4	RA3	12	RA1	13	RA0
	14- pin SOIC	(SO)	1	14	4	RA3	12	RA1	13	RA0
	16-pin QFN	(ML)	16	13	3	RA3	11	RA1	12	RA0
	14-pin TSSOP	(ST)	1	14	4	RA3	12	RA1	13	RA0
PIC16(L)F15344	20-pin PDIP	(P)	1	20	4	RA3	18	RA1	19	RA0
	20- pin SOIC	(SO)	1	20	4	RA3	18	RA1	19	RA0
	20-pin QFN	(ML)	18	17	1	RA3	15	RA1	16	RA0
	20-pin SSOP	(SS)	1	20	4	RA3	18	RA1	19	RA0
PIC16(L)F15325	14-pin PDIP	(P)	1	14	4	RA3	12	RA1	13	RA0
	14- pin SOIC	(SO)	1	14	4	RA3	12	RA1	13	RA0
	16-pin QFN	(ML)	16	13	3	RA3	11	RA1	12	RA0
	14-pin TSSOP	(ST)	1	14	4	RA3	12	RA1	13	RA0
PIC16(L)F15345	20-pin PDIP	(P)	1	20	4	RA3	18	RA1	19	RA0
	20- pin SOIC	(SO)	1	20	4	RA3	18	RA1	19	RA0
	20-pin QFN	(ML)	18	17	1	RA3	15	RA1	16	RA0
	20-pin SSOP	(SS)	1	20	4	RA3	18	RA1	19	RA0
PIC16(L)F15354 PIC16(L)F15355	28-pin SPDIP	(SP)	20	19,8	1	RE3	27	RB6	28	RB7
	28- pin SOIC	(SO)	20	19,8	1	RE3	27	RB6	28	RB7
	28-pin QFN	(ML)	17	5,16	26	RE3	24	RB6	25	RB7
	28-pin SSOP	(SS)	20	19,8	1	RE3	27	RB6	28	RB7
	28-pin UQFN	(MV)	17	5,16	26	RE3	24	RB6	25	RB7
PIC16(L)F15356	28-pin SPDIP	(SP)	20	19,8	1	RE3	27	RB6	28	RB7
	28- pin SOIC	(SO)	20	19,8	1	RE3	27	RB6	28	RB7
	28-pin QFN	(ML)	17	5,16	26	RE3	24	RB6	25	RB7
	28-pin SSOP	(SS)	20	19,8	1	RE3	27	RB6	28	RB7
	28-pin UQFN	(MV)	17	5,16	26	RE3	24	RB6	25	RB7
PIC16(L)F15375 PIC16(L)F15376	40-pin PDIP	(P)	11,32	12,31	1	RE3	39	RB6	40	RB7
	40-pin UQFN	(MV)	7,26	6,27	16	RE3	14	RB6	15	RB7
	44-pin QFN	(ML)	7,28	6,30	18	RE3	16	RB6	17	RB7
	44-pin TQFP	(PT)	7,28	6,29	18	RE3	16	RB6	17	RB7
PIC16(L)F15385 PIC16(L)F15386	48-pin UQFN	(MV)	7,30	6,31	20	RE3	18	RB6	19	RB7
	48-pin TQFP	(PT)	7,30	6,31	20	RE3	18	RB6	19	RB7

REGISTER B-1: CONFIGURATION WORD 1: OSCILLATORS

R/P-1	U-1	R/P-1	U-1	U-1	R/P-1	U-1	R/P-1	R/P-1	R/P-1	U-1	R/P-1	R/P-1	R/P-1
FCMEN	—	CSWEN	—	—	$\overline{\text{CLKOUTEN}}$	—	RSTOSC2	RSTOSC1	RSTOSC0	—	FEXTOSC2	FEXTOSC1	FEXTOSC0
bit 13													bit 0

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 10-9 **Unimplemented**: Read as ‘1’

bit 8 **$\overline{\text{CLKOUTEN}}$**: Clock Out Enable bit
If FEXTOSC = EC (high, mid or low) or Not Enabled:
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
101 = Reserved
100 = LFINTOSC
011 = SOSC
010 = EXTOSC with 2x PLL, with EXTOSC operating per FEXTOSC bits
001 = EXTOSC with 4x PLL, with EXTOSC operating per FEXTOSC bits
000 = HFINTOSC with 2x PLL and HFFRQ = 4’b1111

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 = Reserved (do not use)
010 = HS (Crystal oscillator) above 4 MHz; PFM set to high power
001 = XT (Crystal oscillator) above 100 kHz, below 4 MHz; PFM set to medium power
000 = LP (Crystal oscillator) optimized for 32.768 kHz; PFM set to low power

REGISTER B-2: CONFIGURATION WORD 2: SUPERVISORS

R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	U-1	R/P-1	R/P-1	R/P-1	U-1	U-1	U-1	R/P-1	R/P-1
$\overline{\text{DEBUG}}$	STVREN	PPS1WAY	ZCDDIS	BORV	—	BOREN1	BOREN0	$\overline{\text{LPBOREN}}$	—	—	—	$\overline{\text{PWRTE}}$	MCLRE
bit 13													bit 0

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 **$\overline{\text{DEBUG}}$**: 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 **$\overline{\text{LPBOREN}}$**: Low-Power BOR Enable bit
1 = ULPBOR is disabled
0 = ULPBOR is enabled

bit 4-2 **Unimplemented**: Read as '1'

bit 1 **$\overline{\text{PWRTE}}$**: Power-up Timer Enable bit
1 = PWRT is disabled
0 = PWRT is enabled

bit 0 **MCLRE**: Master Clear ($\overline{\text{MCLR}}$) Enable bit
If LVP = 1:
RE3 pin function is $\overline{\text{MCLR}}$ (it will reset the device when driven low)
If LVP = 0:
1 = $\overline{\text{MCLR}}$ pin is $\overline{\text{MCLR}}$ (it will the reset device when driven low)
0 = $\overline{\text{MCLR}}$ pin may be used as general purpose RE3 input

REGISTER B-3: CONFIGURATION WORD 3: WINDOWED WATCHDOG

R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
WDTCCS2	WDTCCS1	WDTCCS0	WDTCWS2	WDTCWS1	WDTCWS0
bit 13					bit 8

U-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
	WDTE1	WDTE0	WDTCPS4	WDTCPS3	WDTCPS2	WDTCPS1	WDTCPS0
bit 7							bit 0

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

bit 13-11 **WDTCCS<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
.
.
.
.
110 = Reserved
111 = Software Control

bit 10-8 **WDTCWS<2:0>**: WDT Window Select bits

WDTCWS	WDTWS at POR			Software control of WDTWS?	Keyed access required?
	Value	Window delay Percent of time	Window opening Percent of time		
000	000	87.5	12.5	No	Yes
001	001	75	25		
010	010	62.5	37.5		
011	011	50	50		
100	100	37.5	62.5		
101	101	25	75		
110	111	n/a	100		
111	111	n/a	100	Yes	No

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

WDTCPS	WDTPS at POR				Software Control of WDTPS?
	Value	Divider Ratio		Typical Time Out (FIN = 31 kHz)	
00000	00000	1:32	2^{5}	1 ms	No
00001	00001	1:64	2^{6}	2 ms	
00010	00010	1:128	2^{7}	4 ms	
00011	00011	1:256	2^{8}	8 ms	
00100	00100	1:512	2^{9}	16 ms	
00101	00101	1:1024	2^{10}	32 ms	
00110	00110	1:2048	2^{11}	64 ms	
00111	00111	1:4096	2^{12}	128 ms	
01000	01000	1:8192	2^{13}	256 ms	
01001	01001	1:16384	2^{14}	512 ms	
01010	01010	1:32768	2^{15}	1 s	
01011	01011	1:65536	2^{16}	2 s	
01100	01100	1:131072	2^{17}	4 s	
01101	01101	1:262144	2^{18}	8 s	
01110	01110	1:524299	2^{19}	16 s	
01111	01111	1:1048576	2^{20}	32 s	
10000	10000	1:2097152	2^{21}	64 s	
10001	10001	1:4194304	2^{22}	128 s	
10010	10010	1:8388608	2^{23}	256 s	
10011 . . . 11110	10011 . . . 11110	1:32	2^{5}	1 ms	No

REGISTER B-4: CONFIGURATION WORD 4: MEMORY

R/W-1	U-1	R/W-1	U-1	R/W-1	R/W-1
LVP	—	$\overline{\text{WRTSAF}}$[1]	—	$\overline{\text{WRTC}}$[1]	$\overline{\text{WRTB}}$[1]
bit 13	12	11	10	9	bit 8

R/W-1	U-1	U-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
$\overline{\text{WRTAPP}}$[1]	—	—	$\overline{\text{SAFEN}}$[1]	$\overline{\text{BBEN}}$[1]	BBSIZE2	BBSIZE1	BBSIZE0
bit 7	6	5	4	3	2	1	bit 0

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

bit 13 **LVP:** Low Voltage Programming Enable bit
1 = Low voltage programming enabled. $\overline{\text{MCLR}}$/VPP pin function is $\overline{\text{MCLR}}$. MCLRE Configuration bit is ignored.
0 = HV on $\overline{\text{MCLR}}$/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 **$\overline{\text{WRTSAF}}$:** 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 $\overline{\text{SAFEN}}$ = 0.

bit 10 **Unimplemented:** Read as ‘1’

bit 9 **$\overline{\text{WRTC}}$:** Configuration Register Write Protection bit
1 = Configuration Register NOT write-protected
0 = Configuration Register write-protected

bit 8 **$\overline{\text{WRTB}}$:** Boot Block Write Protection bit
1 = Boot Block NOT write-protected
0 = Boot Block write-protected
Only applicable if $\overline{\text{BBEN}}$ = 0.

bit 7 **$\overline{\text{WRTAPP}}$:** 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 **$\overline{\text{SAFEN}}$:** SAF Enable bit
1 = SAF disabled
0 = SAF enabled

bit 3 **$\overline{\text{BBEN}}$:** Boot Block Enable bit
1 = Boot Block disabled
0 = Boot Block enabled

bit 2-0 **BBSIZE[2:0]:** Boot Block Size Selection bits
BBSIZE is used only when $\overline{\text{BBEN}}$ = 0
BBSIZ bits can only be written while $\overline{\text{BBEN}}$ = 1; after $\overline{\text{BBEN}}$ = 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.

TABLE B-3: BOOT BLOCK SIZE BITS

$\overline{\text{BBEN}}$	BBSIZE[2:0]	Preferred Boot Block Size (words)	Actual Boot Block Size User Program Memory Size[2] (words)						END_ADDRESS_BOOT
			1k	2k	4k	8k	16k	32k	
1	xxx	0	0	0	0	0	0	0	—
0	111	512	512	512	512	512	512	512	01FFh
0	110	1024		1024	1024	1024	1024	1024	03FFh
0	101	2048			2048	2048	2048	2048	07FFh
0	100	4096				4096	4096	4096	0FFFh
0	011	8192					8192	8192	1FFFh
0	010	16384			**Note 3**			16384	3FFFh
0	001	32768							3FFFh
0	000	65536							3FFFh

Note 1: This is generic information, and not all entries apply to this device.

2: For each device, the user program memory size specification is listed in Figure 2-1.

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.

REGISTER B-5: CONFIGURATION WORD 5: CODE PROTECTION

U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—
bit 13					bit 8

U-1	U-1	U-1	U-1	U-1	U-1	U-1	R/P-1
—	—	—	—	—	—	—	$\overline{\text{CP}}$
bit 7							bit 0

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

bit 13-1 **Unimplemented:** Read as '1'

bit 0 **$\overline{\text{CP}}$:** Program Flash Memory Code Protection bit
1 = Program Flash Memory code protection disabled
0 = Program Flash Memory code protection enabled

APPENDIX C: DEVICE CONFIGURATION INFORMATION (DCI)

TABLE C-1: DEVICE CONFIGURATION INFORMATION

ADDRESS	DESCRIPTION	VALUE	UNITS
PIC16			
8200h	Erase Row Size	32	Words
8201h	Number of write latches	32	—
8202h	Number of User Rows	See Table C-2	Rows
8203h	EE Data memory size	0	Bytes
8204h	Pin Count	See Table C-3	Pins

Note 1: These locations are read-only.

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

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

TABLE C-2: NUMBER OF USER ROWS

Part Name	Memory size	Number of User rows
PIC16(L)F15313/23	2k	64
PIC16(L)F15324/44/54	4k	128
PIC16(L)F15325/45/55/75/85	8k	256
PIC16(L)F15356/76/86	16k	512

TABLE C-3: PIN COUNT

Part Number	Pin Count
PIC16(L)F15313	8
PIC16(L)F15323/24/25	14,16
PIC16(L)F15344/45	20
PIC16(L)F15354/55/56	28
PIC16(L)F15375/76	40,44

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods being used in attempts to breach the code protection features of the Microchip devices. We believe that these methods require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Attempts to breach these code protection features, most likely, cannot be accomplished without violating Microchip's intellectual property rights.
- Microchip is willing to work with any customer who is concerned about the integrity of its code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is "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 is provided for the sole purpose of designing with and using Microchip products. Information 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.

THIS INFORMATION IS PROVIDED BY MICROCHIP "AS IS". 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 ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE.

IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL OR CONSEQUENTIAL LOSS, DAMAGE, COST OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP'S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION. 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.

For information regarding Microchip's Quality Management Systems, please visit www.microchip.com/quality.

Trademarks

ISBN: 978-1-5224-7866-9

MICROCHIP

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: 774-760-0087
Fax: 774-760-0088

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: 61-2-9868-6733

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: 86-24-2334-2829

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: 86-592-2388138

China - Zhuhai
Tel: 86-756-3210040

ASIA/PACIFIC

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: 45-4485-5910
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: 49-2129-3766400

Germany - Heilbronn
Tel: 49-7131-72400

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: 39-049-7625286

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

Norway - Trondheim
Tel: 47-7288-4388

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

02/28/20