MICROCHIP

PIC16F88X

PIC16F88X Memory Programming Specification

This document includes the programming specifications for the following devices:

- PIC16F883
- PIC16F884
- PIC16F886
- PIC16F887

1.0 PROGRAMMING THE PIC16F88X DEVICES

The PIC16F88X can be programmed using the high voltage In-Circuit Serial Programming™ (ICSP™) method or the low voltage ICSP method. Both of these can be done with the device in the user's system. The low voltage ICSP method is slightly different than the high voltage method and these differences are noted where applicable. This programming specification applies to these devices in all package types.

1.1 Hardware Requirements

In high voltage ICSP mode, PIC16F88X devices require two programmable power supplies: one for VDD and one for $\overline{\text{MCLR}}$/VPP (see **Section 6.0 "Program/ Verify Mode Electrical Characteristics"** for more details).

1.2 Program/Verify Mode

The Program/Verify mode for the PIC16F88X devices allow programming of user program memory, data memory, user ID locations, Calibration Words and the Configuration Word.

Programming and verification can take place in any memory region, independent of the remaining regions. This allows independent programming of program and data memory regions.

TABLE 1-1: PIN DESCRIPTIONS IN PROGRAM/VERIFY MODE: PIC16F88X

Pin Name	During Programming		
	Function	**Pin Type**	**Pin Description**
RB3	PGM	I	Low voltage ICSP™ programming input if LVP Configuration bit equals '1'
RB6	ICSPCLK	I	Clock Input – Schmitt Trigger Input
RB7	ICSPDAT	I/O	Data Input/Output – Schmitt Trigger Input
$\overline{\text{MCLR}}$	Program/Verify mode	P(1)	Program Mode Select
VDD	VDD	P	Power Supply
VSS	VSS	P	Ground

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

Note 1: In the PIC16F88X, the programming high voltage is internally generated. To activate the Program/Verify mode, high voltage needs to be applied to $\overline{\text{MCLR}}$ input. Since the $\overline{\text{MCLR}}$ is used for a level source, $\overline{\text{MCLR}}$ does not draw any significant current.

FIGURE 1-1: PIC16F883/886 28-PIN PDIP, SOIC, SSOP

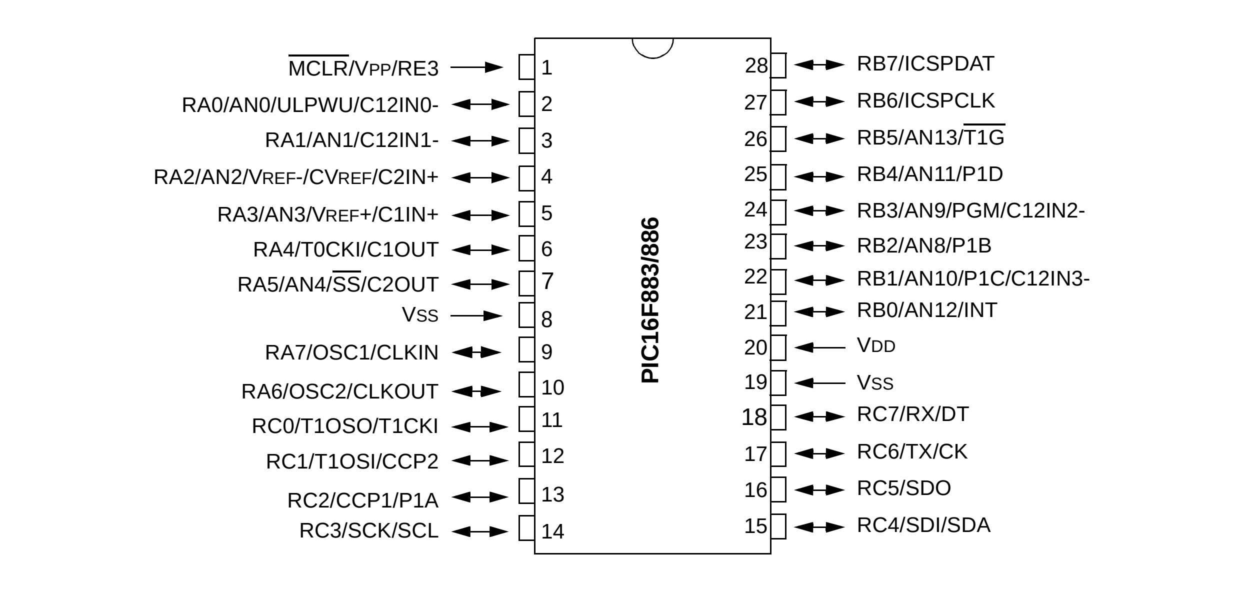


FIGURE 1-2: PIC16F883/886 28-PIN QFN

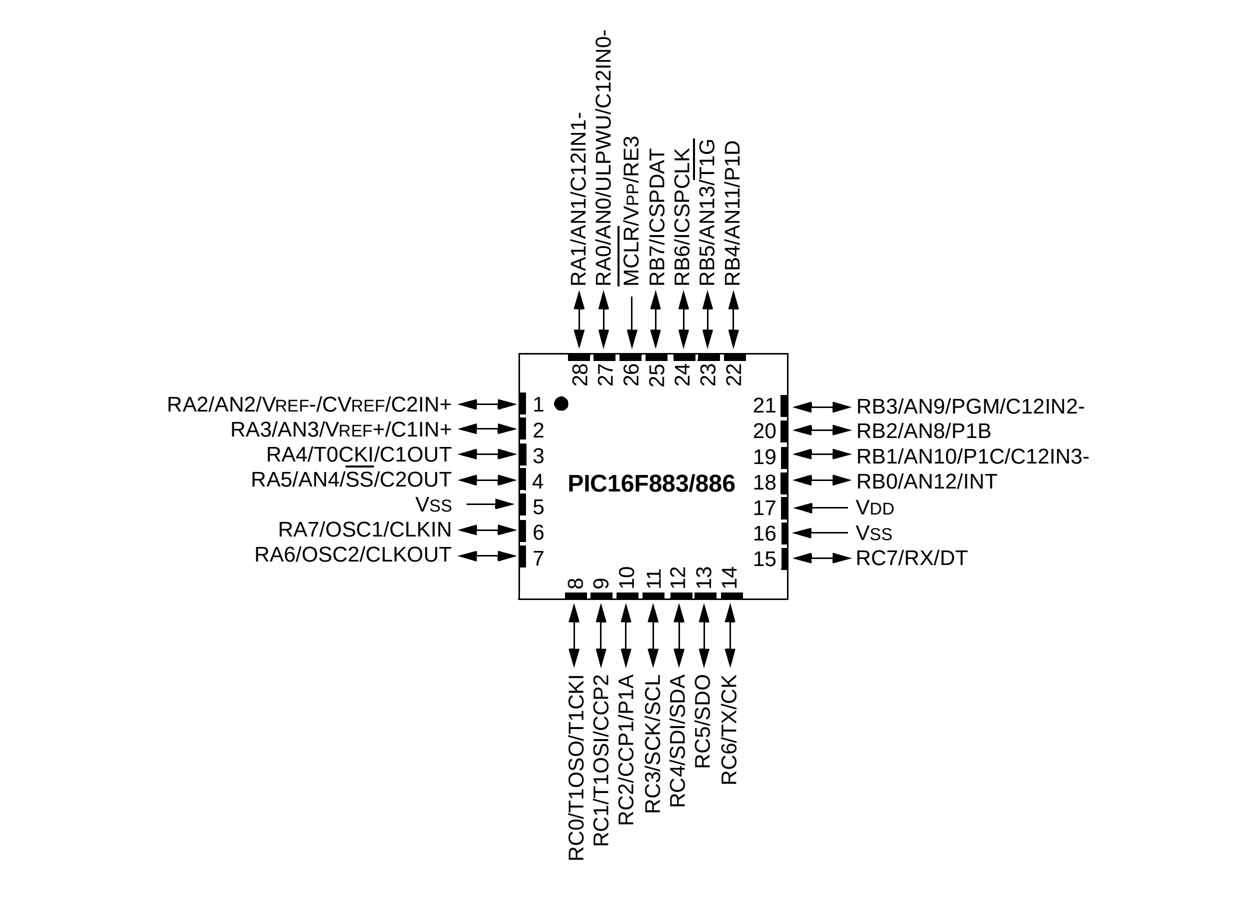


FIGURE 1-3: PIC16F884/887 44-PIN DIAGRAM TQFP, QFN

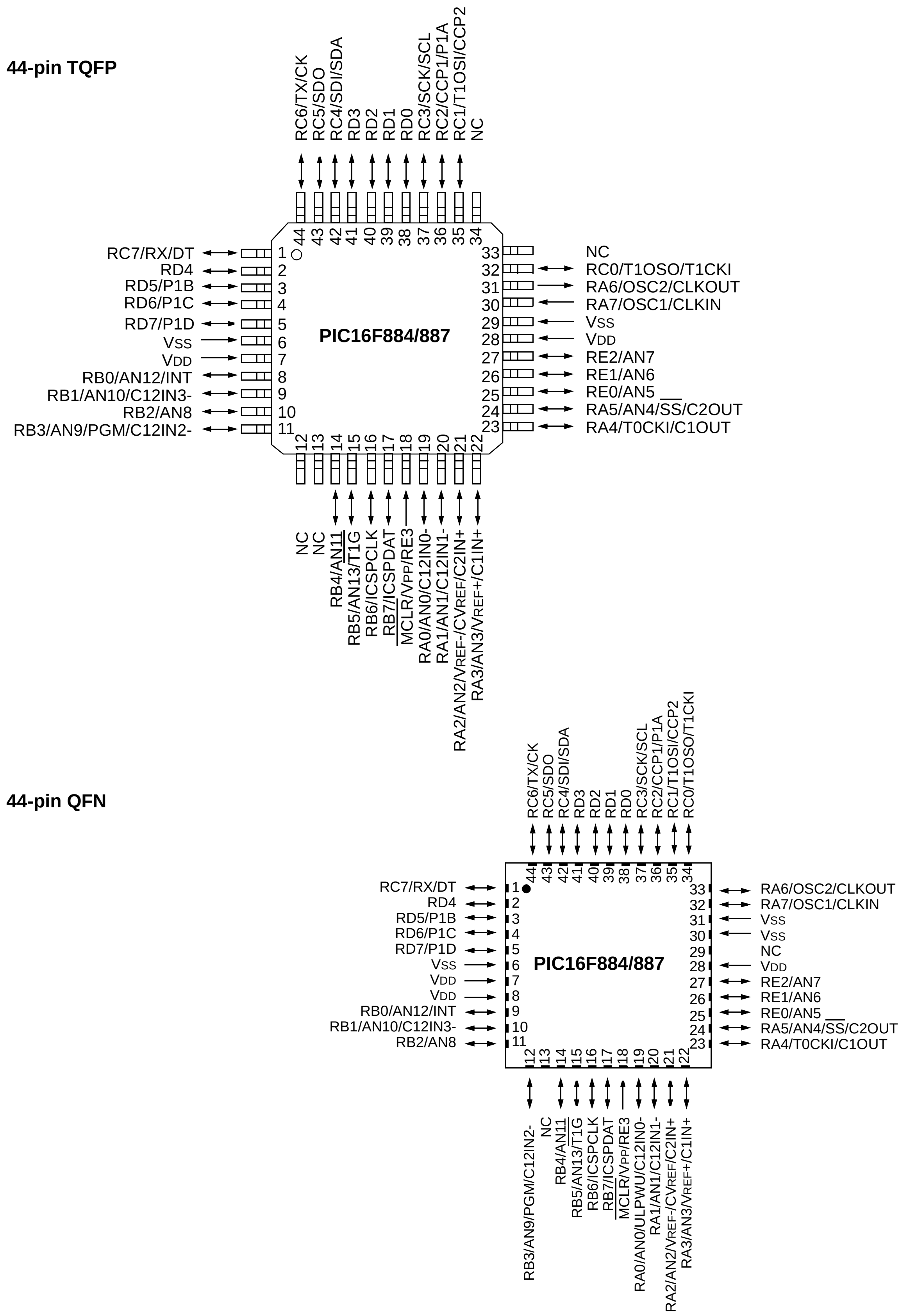
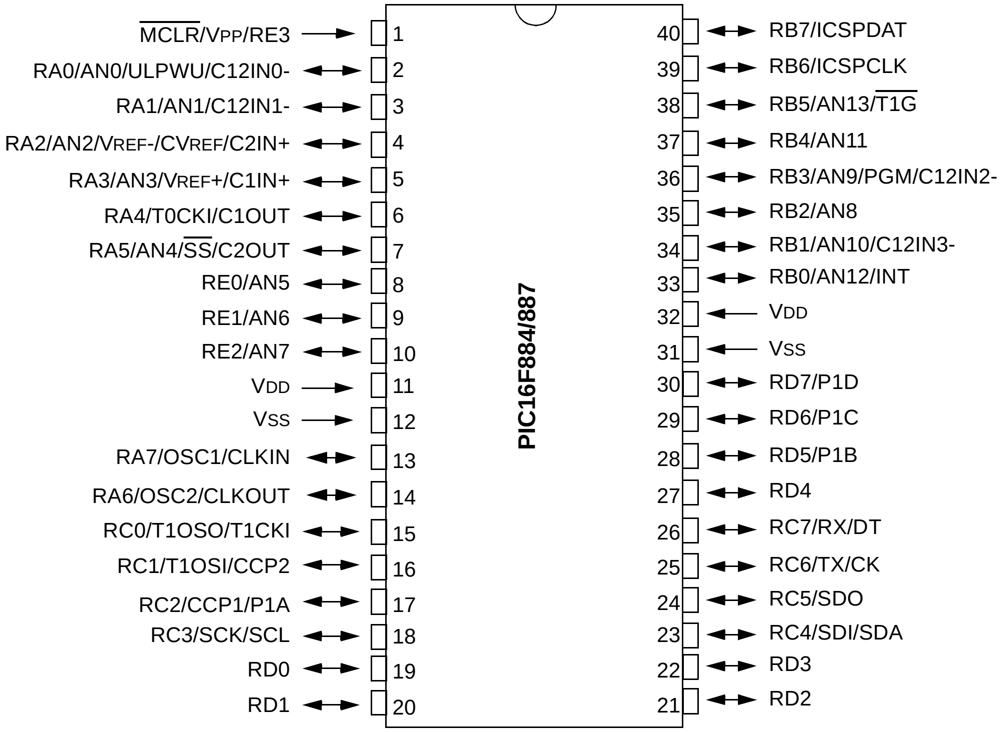


FIGURE 1-4: PIC16F884/887 40-PIN PDIP



2.0 MEMORY DESCRIPTION

2.1 Program Memory Map

The user memory space extends from 0x0000-0x0FFF for the PIC16F883/884, and from 0x0000-0x1FFF for PIC16F886/887. In Program/Verify mode, the program memory space extends from 0x0000 to 0x3FFF, with the first half being user program memory and the second half (0x2000-0x3FFF) being configuration memory. The PC will increment from 0x0000 to 0x1FFF and wrap to 0x0000, 0x2000 to 0x3FFF and wrap around to 0x2000 (not to 0x0000). Once in configuration memory, the highest bit of the PC stays a '1', thus always pointing to the configuration memory. The only way to point to user program memory is to reset the part and re-enter Program/Verify mode as described in **Section 3.0 "Program/Verify Mode"**.

For the PIC16F88X devices, the configuration memory space, 0x2000-0x2009, are physically implemented. However, only locations 0x2000-0x2003 and 0x2007-0x2009 are available. Other locations are reserved.

2.2 User ID Locations

A user may store identification information (user ID) in four designated locations. The user ID locations are mapped in 0x2000-0x2003. It is recommended that the user use only the seven Least Significant bits (LSb) of each user ID location. The user ID locations read out normally, even after code protection is enabled. It is recommended that ID locations are written as 'xx xxxx xbbb bbbb' where 'bbb bbbb' is user ID information.

The 14 bits may be programmed, but only the 7 LSbs are displayed by MPLAB® IDE. The xxxx's are "don't care" bits and are not read by MPLAB IDE.

2.3 Calibration Word

For the PIC16F88X devices, the 8 MHz Internal Oscillator (INTOSC), the Power-on Reset (POR), and the Brown-out Reset (BOR) modules are factory calibrated and stored in the Calibration Word (0x2009). See the applicable device data sheet for more information.

The Calibration Words do not necessarily participate in erase operation unless a specific procedure is executed. Therefore, the device can be erased without affecting the Calibration Words. This simplifies the erase procedure, for these values do not need to be read and restored after the device is erased. See **Section 3.2.6.10 "Bulk Erase Program Memory"** for more information on the various erase sequences.

FIGURE 2-1: PIC16F883/884 PROGRAM MEMORY MAPPING

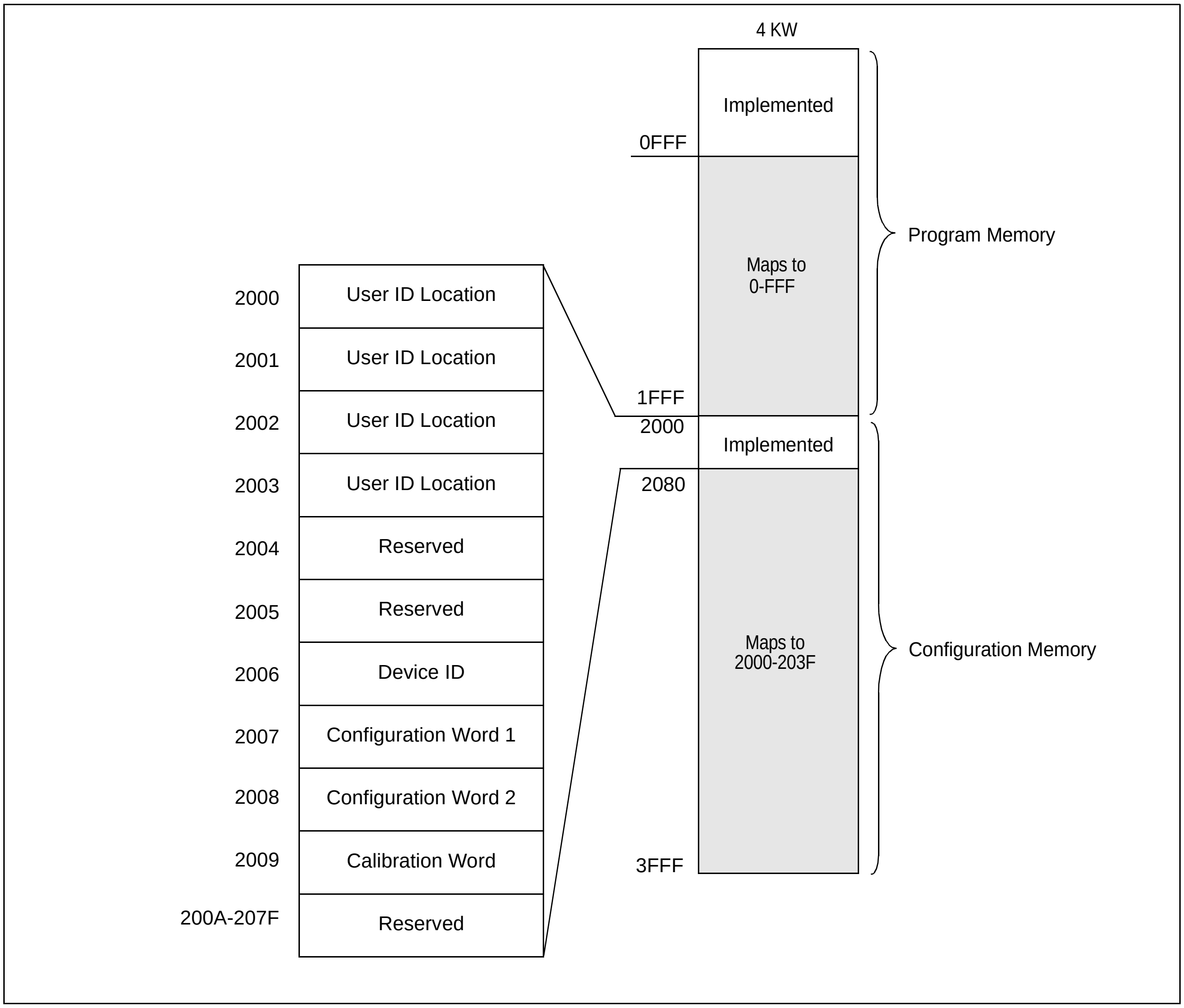
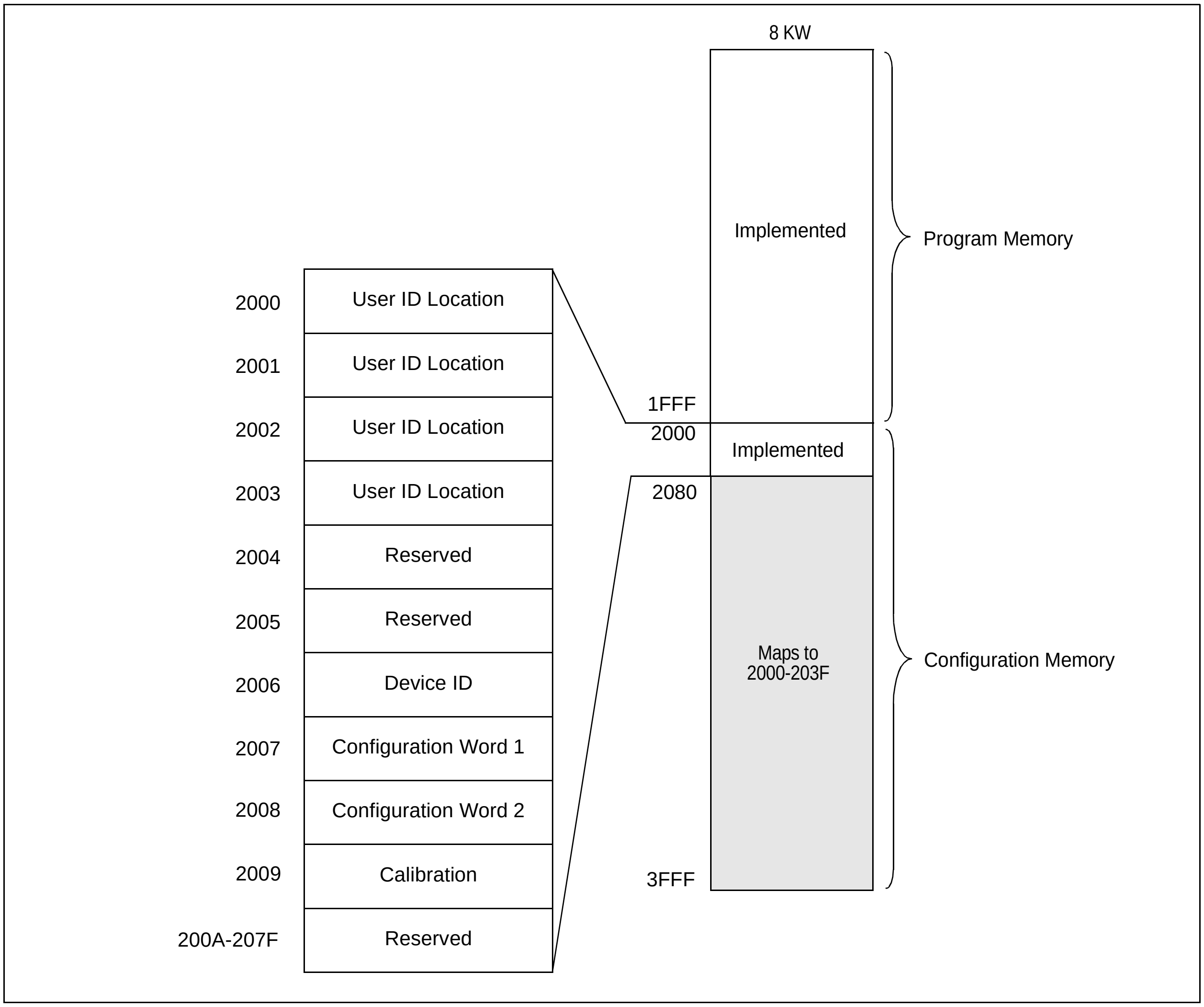


FIGURE 2-2: PIC16F886/887 PROGRAM MEMORY MAPPING



3.0 PROGRAM/VERIFY MODE

Two methods are available to enter Program/Verify mode. The “VPP-first” is entered by holding ICSPDAT and ICSPCLK low while raising $\overline{\text{MCLR}}$ pin from VIL to VIHH (high voltage), then applying VDD and data. This method can be used for any Configuration Word selection and **must** be used if the INTOSC and internal $\overline{\text{MCLR}}$ options are selected (FOSC<2:0> = 100 or 101 and MCLRE = 0). The VPP-first entry prevents the device from executing code prior to entering Program/Verify mode. See the timing diagram in Figure 3-1.

The second entry method, “VDD-first”, is entered by applying VDD, holding ICSPDAT and ICSPCLK low, then raising $\overline{\text{MCLR}}$ pin from VIL to VIHH (high voltage), followed by data. This technique is useful when programming the device when VDD is already applied, for it is not necessary to disconnect VDD to enter Program/Verify mode. See the timing diagram in Figure 3-2.

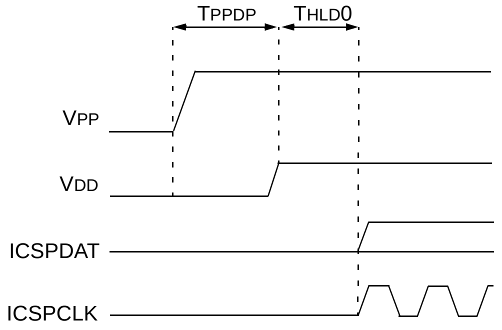
Once in this mode, the program memory, data memory, and configuration memory can be accessed and programmed in serial fashion. ICSPDAT and ICSPCLK are Schmitt Trigger inputs in this mode. RB6 is tri-state, regardless of fuse setting.

The sequence that enters the device into the Programming/Verify mode places all other logic into the Reset state (the $\overline{\text{MCLR}}$ pin was initially at VIL). Therefore, all I/O’s are in the Reset state (high-impedance inputs) and the Program Counter (PC) is cleared.

When powering-down VDD, make sure VDD does not undershoot VSS. If VDD undershoots VSS while VPP is applied, device damage can result. To prevent possible device damage, power-down VPP either before VDD or at the same time as VDD.

When programming a device with internal $\overline{\text{MCLR}}$ and INTOSC, care must be taken to prevent code execution during power-down. If VDD is powered down before VPP, there is a possibility for a VDD undershoot to cause device damage. If VPP is powered down before VDD, there is the possibility of code execution. If VDD is powered down at the same time as VPP or just slightly after VPP, code execution is prevented. See Figure 3-3 for the timing.

FIGURE 3-1: VPP-FIRST PROGRAM/ VERIFY MODE ENTRY



Note: This method of entry is valid, regardless of Configuration Word selected.

FIGURE 3-2: VDD-FIRST PROGRAM/ VERIFY MODE ENTRY

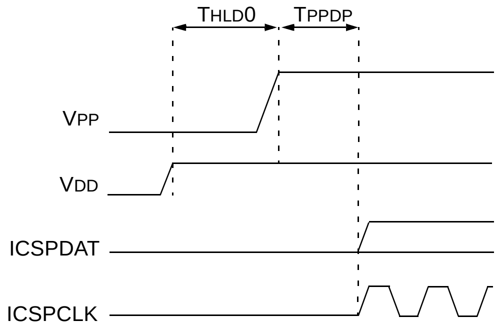
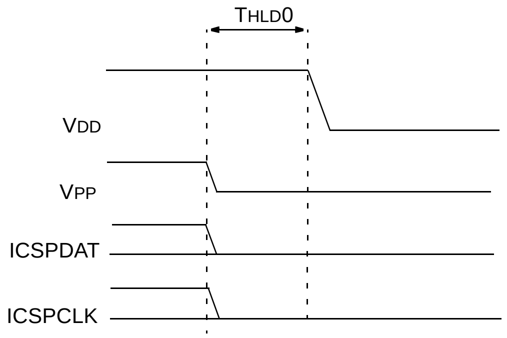


FIGURE 3-3: PROGRAM/VERIFY MODE EXIT



3.1 Low Voltage ICSP™ Mode

Low Voltage ICSP Programming mode allows the PIC16F88X device to be programmed using VDD only. However, when this mode is enabled by a Configuration bit (LVP), the PIC16F88X device dedicates RB3 to control entry/exit into Programming mode. When LVP bit is set to '1', the low voltage ICSP programming entry is enabled. Since the LVP Configuration bit allows low voltage ICSP programming entry in its erased state, an erased device will have the LVP bit enabled at the factory. While LVP is '1', RB3 is dedicated to low-voltage ICSP programming. Bring RB3 and then, $\overline{\text{MCLR}}$ to VDD to enter Programming mode. All other specifications for high voltage ICSP apply. To disable Low Voltage ICSP mode, the LVP bit must be programmed to '0'. This must be done while entered with the High Voltage Entry mode (LVP bit = '1'). RB3 is now a general purpose I/O pin.

3.2 Program/Erase Algorithms

The PIC16F88X program memory may be written in three ways. The PIC16F883/884 uses one-word and four-word writes. The PIC16F886/887 uses one-word, four-word and eight-word writes. The four-word or eight-word algorithm is used to program the program memory only. The one-word algorithm can write any available memory location (i.e., program memory, configuration memory and data memory).

After writing the array, the PC may be reset and read back to verify the write. It is not possible to verify immediately following the write because the PC can only increment, not decrement.

A device Reset will clear the PC and set the address to '0'. The Increment Address command will increment the PC. The Load Configuration command will set the PC to 0x2000. The available commands are shown in Table 3-1.

3.2.1 EIGHT-WORD PROGRAMMING

Only the program memory on PIC16F886/887 can be written using this algorithm. Data and configuration memory (>0x2000) must use the One-word Programming Algorithm (**Section 3.2.3 "One-Word Programming"**).

This algorithm writes eight sequential addresses in program memory. The eight addresses must point to an eight-word block with addresses modulo 8 of 0, 1, 2, 3, 4, 5, 6 and 7. For example, programming address 8 through 15 can be programmed together. Programming addresses 2 through 9 will create an unexpected result.

The sequence for programming eight words of program memory at a time is as follows:

1. Load a word at the current program memory address using Load Data For Program Memory command.
2. Issue an Increment Address command.
3. Load a word at the current program memory address using Load Data For Program Memory command.
4. Repeat Step 2 and Step 3 six times.
5. Issue a Begin Programming command either internally or externally timed.
6. Wait TPROG1 (internally timed) or TPROG2 (externally timed).
7. Issue End Programming if externally timed.
8. Issue an Increment Address command.
9. Repeat this sequence as required to write program memory.

See Figure 3-18 for more information.

3.2.2 FOUR-WORD PROGRAMMING

Four-word programming can be used on all devices in the PIC16F88X family. Only the program memory can be written using this algorithm. Data and configuration memory (>0x2000) must use the One-word Programming Algorithm (**Section 3.2.3 "One-Word Programming"**).

This algorithm writes four sequential addresses in program memory. The four addresses must point to a four-word block with addresses modulo 4 of 0, 1, 2 and 3. For example, programming address 4 through 7 can be programmed together. Programming addresses 2 through 5 will create an unexpected result.

The sequence for programming four words of program memory at a time is as follows:

1. Load a word at the current program memory address using Load Data For Program Memory command.
2. Issue an Increment Address command.
3. Load a word at the current program memory address using Load Data For Program Memory command.
4. Repeat Step 2 and Step 3 two times.
5. Issue a Begin Programming command either internally or externally timed.
6. Wait TPROG1 (internally timed) or TPROG2 (externally timed).
7. Issue End Programming if externally timed.
8. Issue an Increment Address command.
9. Repeat this sequence as required to write program memory.

See Figure 3-17 for more information.

3.2.3 ONE-WORD PROGRAMMING

The program memory may also be written one word at a time to allow compatibility with other 8-pin and 14-pin Flash PICmicro® devices. Configuration memory (>0x2000) and data memory must be written one word (or byte) at a time.

> **Note:** The write latches must be reset after programming the device ID (0x2006), Configuration Words (0x2007-0x2008) or Calibration Word (0x2009). See **Section 3.2.4 "Resetting Write Latches"**.

The sequence for programming one word of program memory at a time is as follows:

1. Load a word at the current program memory address using Load Data For Program Memory command.
2. Issue a Begin Programming command either internally or externally timed.
3. Wait TPROG1 (internally timed) or TPROG2 (externally timed).
4. Issue End Programming if externally timed.
5. Issue an Increment Address command.
6. Repeat this sequence as required to write program, data or configuration memory.

See Figure 3-16 for more information.

3.2.4 RESETTING WRITE LATCHES

The device ID (0x2006), Configuration Words (0x2007-0x2008) and Calibration Word (0x2009) are mapped into the configuration memory, but do not physically reside in it. As a result, the write latches are not reset when programming these locations and must be reset by the programmer. This can be done in two ways, either loading all eight latches with '1's or by exiting Program/Verify mode.

The sequence for manually resetting the write latches is as follows:

1. Load a word using Load Data For Program Memory or Load Data For Configuration Memory command with a data word of all '1's.
2. Issue an Increment Address command.
3. Repeat this sequence three times on PIC16F883/884 and seven times on PIC16F886/887 to reset all write latches.

3.2.5 ERASE ALGORITHMS

The PIC16F88X will erase different memory locations depending on the Program Counter (PC), $\overline{\text{CP}}$ and $\overline{\text{CPD}}$ values and which erase command is executed. The following sequences can be used to erase noted memory locations. In each sequence, the data memory will be erased if the $\overline{\text{CPD}}$ bit in the Configuration Word is programmed (clear).

To erase the program memory and Configuration Words (0x2007-0x2008), the following sequence must be performed. Note the Calibration Word (0x2009) and user ID (0x2000-0x2003) **will not** be erased.

1. Do a Bulk Erase Program Memory command.
2. Wait TERA to complete erase.

To erase the user ID (0x2000-0x2003), Configuration Words (0x2007-0x2008) and program memory, use the following sequence. Note that the Calibration Word (0x2009) **will not** be erased.

1. Perform Load Configuration with dummy data to point the Program Counter (PC) to 0x2000.
2. Perform a Bulk Erase Program Memory command.
3. Wait TERA to complete erase.

To erase the user ID (0x2000-0x2003), Configuration Words (0x2007-0x2008), Calibration Word (0x2009), and program memory, use the following sequence. Note that the Calibration Word (0x2009) will be erased.

1. Perform Load Configuration with dummy data to point the Program Counter (PC) to 0x2000.
2. Perform 9 Increment Address commands to point the PC to the Calibration Word at 0x2009.
3. Do a Bulk Erase Program Memory command.
4. Wait TERA to complete erase.

To erase the data memory, use the following sequence:

1. Perform a Bulk Erase Data Memory command.
2. Wait TERA to complete erase.

3.2.6 SERIAL PROGRAM/VERIFY OPERATION

The ICSPCLK pin is used as a clock input and the ICSPDAT pin is used for entering command bits and data input/output during serial operation. To input a command, ICSPCLK is cycled six times. Each command bit is latched on the falling edge of the clock with the LSb of the command being input first. The data input onto the ICSPDAT pin is required to have a minimum setup and hold time (see Table 6-1), with respect to the falling edge of the clock. Commands that have data associated with them (Read and Load) are specified to have a minimum delay of TDLY1 between the command and the data. After this delay, the clock pin is cycled 16 times with the first cycle being a Start bit and the last cycle being a Stop bit.

During a read operation, the LSb will be transmitted onto ICSPDAT pin on the rising edge of the second cycle. For a load operation, the LSb will be latched on the falling edge of the second cycle. A minimum TDLY1 delay is also specified between consecutive commands, except for the End Programming command, which requires a TDIS.

All commands and data words are transmitted LSb first. Data is transmitted on the rising edge and latched on the falling edge of the ICSPCLK. To allow for decoding of commands and reversal of data pin configuration, a time separation of at least TDLY1 is required between a command and a data word.

The commands that are available are described in Table 3-1.

TABLE 3-1: COMMAND MAPPING FOR PIC16F88X

Command	Mapping (MSb … LSb)						Data
Load Configuration	x	x	0	0	0	0	0, data (14), 0
Load Data For Program Memory	x	x	0	0	1	0	0, data (14), 0
Load Data For Data Memory	x	x	0	0	1	1	0, data (8), zero (6), 0
Read Data From Program Memory	x	x	0	1	0	0	0, data (14), 0
Read Data From Data Memory	x	x	0	1	0	1	0, data (8), zero (6), 0
Increment Address	x	x	0	1	1	0	
Begin Programming	x	0	1	0	0	0	Internally Timed
Begin Programming	x	1	1	0	0	0	Externally Timed
End Programming	x	0	1	0	1	0	
Bulk Erase Program Memory	x	x	1	0	0	1	Internally Timed
Bulk Erase Data Memory	x	x	1	0	1	1	Internally Timed
Row Erase Program Memory	x	1	0	0	0	1	Internally Timed

3.2.6.1 Load Configuration

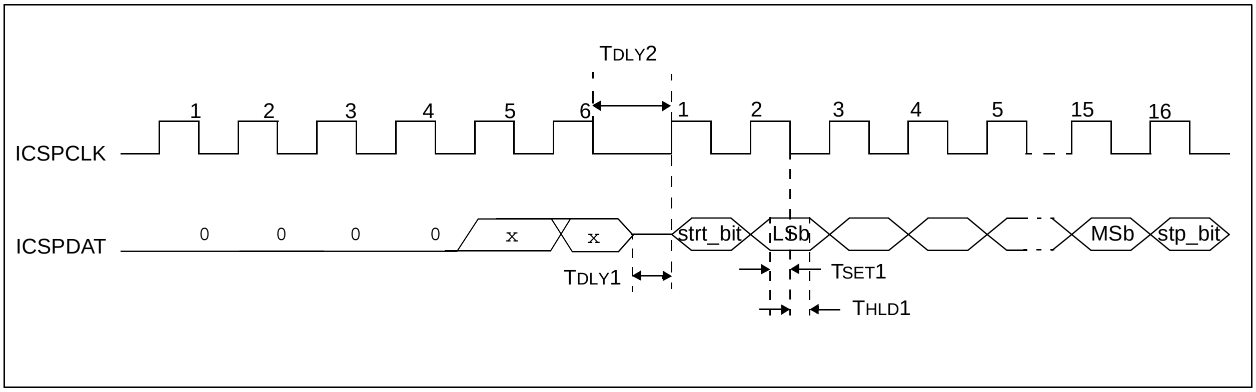
The Load Configuration command is used to access the Configuration Words (0x2007-0x2008) user ID (0x2000-0x2003) and Calibration Word (0x2009). This command sets the Program Counter (PC) to address 0x2000 and loads the data latches with one word of data.

After receiving a Load Configuration command, the Configuration Word is accessed by performing an Increment Address command 7 or 8 times to point the PC to Configuration Word 0x2007 or 0x2008. It can then be programmed with the loaded data using a Begin Programming command either internally or externally timed.

After the 6-bit command is input, ICSPCLK pin is cycled an additional 16 times for the Start bit, 14 bits of data and a Stop bit. See Figure 3-4.

After the configuration memory is entered, the only way to get back to the program memory is to exit the Program/Verify mode by taking $\overline{\text{MCLR}}$ low (VIL).

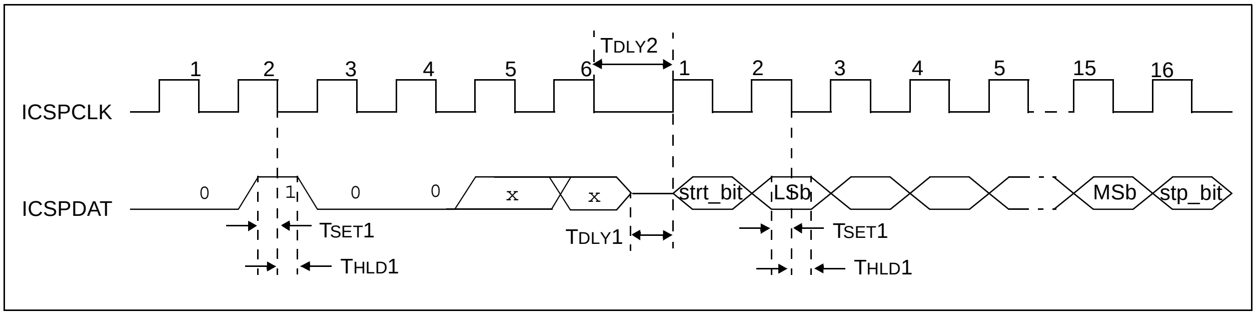
FIGURE 3-4: LOAD CONFIGURATION COMMAND



3.2.6.2 Load Data For Program Memory

After receiving this command, the chip will load in a 14-bit "data word" when 16 cycles are applied, as described previously. A timing diagram for the Load Data For Program Memory command is shown in Figure 3-5.

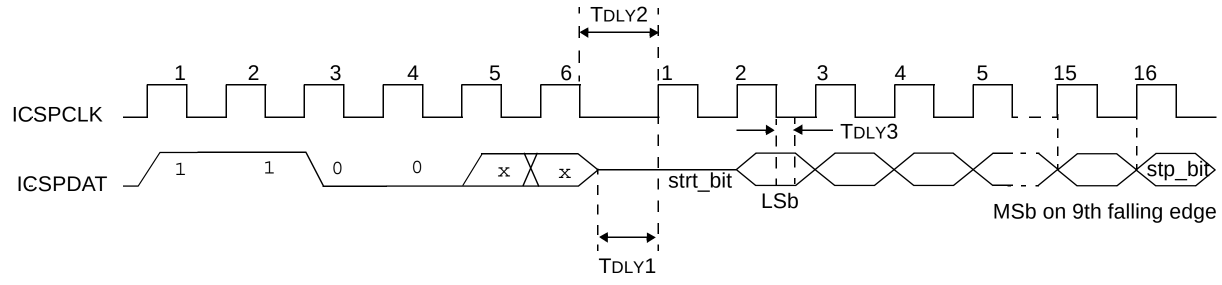
FIGURE 3-5: LOAD DATA FOR PROGRAM MEMORY COMMAND



3.2.6.3 Load Data For Data Memory

After receiving this command, the chip will load in a 14-bit "data word" when 16 cycles are applied. However, the data memory is only 8-bits wide and thus, only the first 8 bits of data after the Start bit will be programmed into the data memory. It is still necessary to cycle the clock the full 16 cycles in order to allow the internal circuitry to reset properly. The data memory contains 256 bytes.

FIGURE 3-6: LOAD DATA FOR DATA MEMORY COMMAND

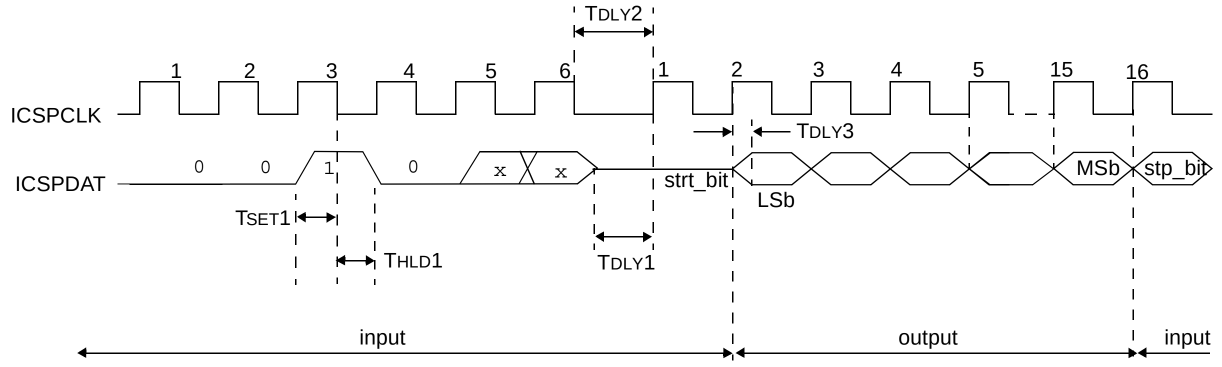


3.2.6.4 Read Data From Program Memory

After receiving this command, the chip will transmit data bits out of the program memory (user or configuration) currently accessed, starting with the second rising edge of the clock input. The data pin will go into Output mode on the second rising clock edge, and it will revert to Input mode (high-impedance) after the 16th rising edge.

If the program memory is code-protected ($\overline{\text{CP}}$ = 0), the data is read as zeros.

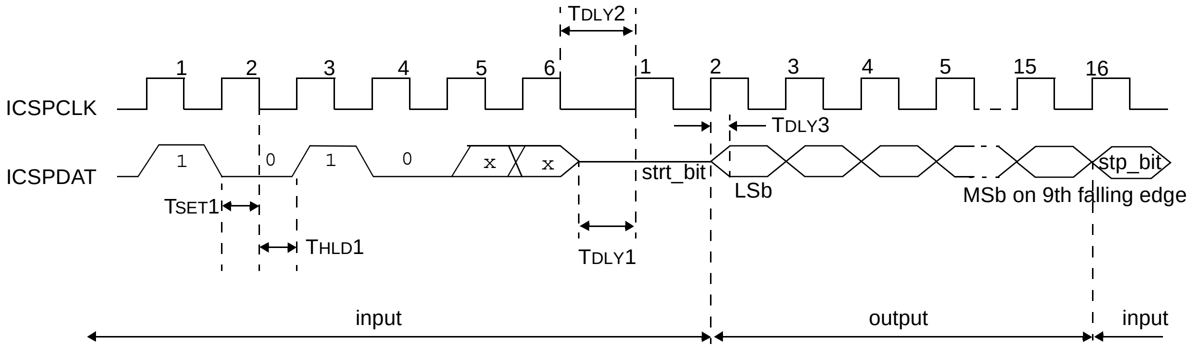
FIGURE 3-7: READ DATA FROM PROGRAM MEMORY COMMAND



3.2.6.5 Read Data From Data Memory

After receiving this command, the chip will transmit data bits out of the data memory starting with the second rising edge of the clock input. The ICSPDAT pin will go into Output mode on the second rising edge, and it will revert to Input mode (high-impedance) after the 16th rising edge. As previously stated, the data memory is 8-bits wide, and therefore, only the first 8 bits that are output are actual data. If the data memory is code-protected, the data is read as all zeros. A timing diagram of this command is shown in Figure 3-8.

FIGURE 3-8: READ DATA FROM PROGRAM MEMORY COMMAND

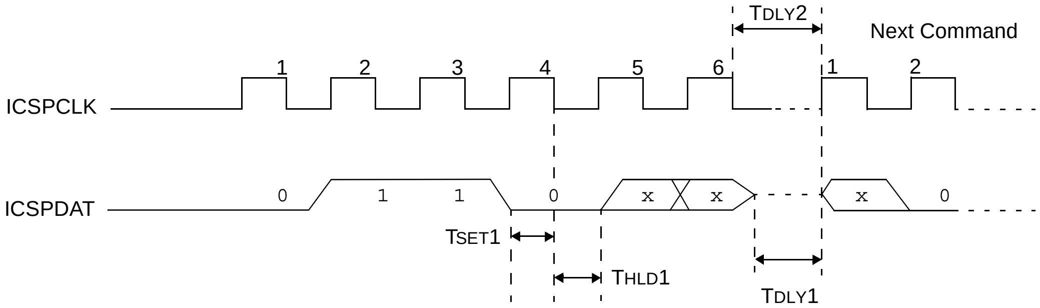


3.2.6.6 Increment Address

The PC is incremented when this command is received. A timing diagram of this command is shown in Figure 3-9.

It is not possible to decrement the address counter. To reset this counter, the user should exit and re-enter Program/Verify mode.

FIGURE 3-9: INCREMENT ADDRESS COMMAND (PROGRAM/VERIFY)



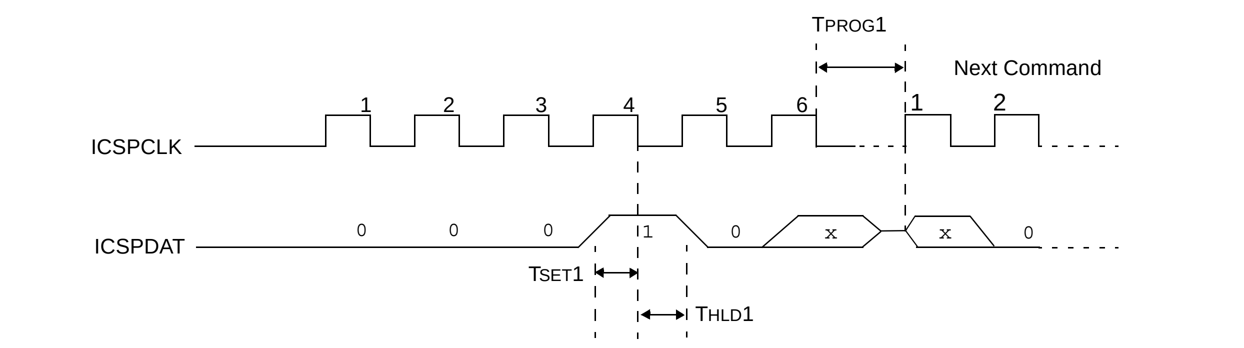
Advance Information

3.2.6.7 Begin Programming (Internally Timed)

A Load command must be given before every Begin Programming command. Programming of the appropriate memory (user program memory, configuration memory or data memory) will begin after this command is received and decoded. An internal timing mechanism executes a write. The user must allow for program cycle time for programming to complete. No End Programming command is required.

The addressed location is not erased before programming. However, the address location is erased if Data Memory is being programmed.

FIGURE 3-10: BEGIN PROGRAMMING (INTERNALLY TIMED)

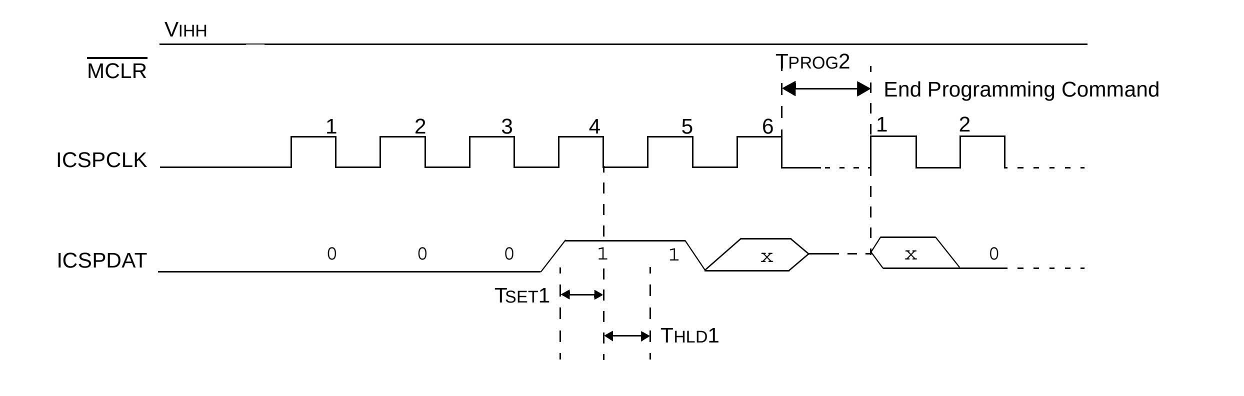


3.2.6.8 Begin Programming (Externally Timed)

A Load command must be given before every Begin Programming command**.** Programming of the appropriate memory (program memory, configuration or data memory) will begin after this command is received and decoded. Programming requires (TPROG2) time and is terminated using an End Programming command.

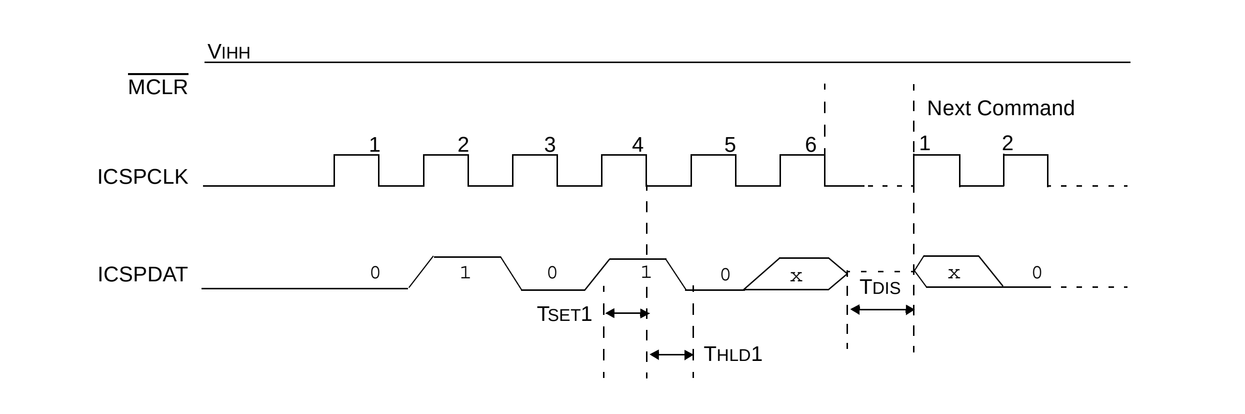
The addressed location is not erased before programming.

FIGURE 3-11: BEGIN PROGRAMMING COMMAND (EXTERNALLY TIMED



3.2.6.9 End Programming

FIGURE 3-12: END PROGRAMMING (SERIAL PROGRAM/VERIFY)

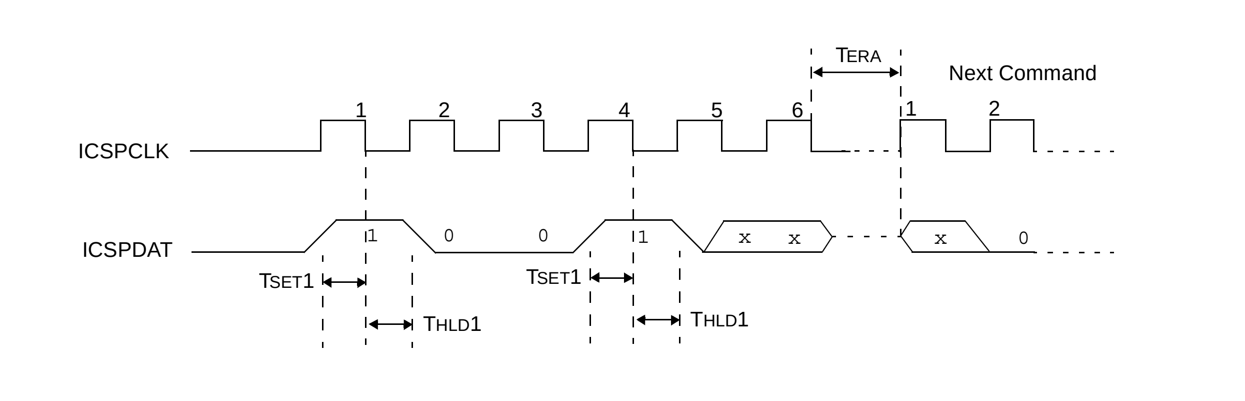


3.2.6.10 Bulk Erase Program Memory

After this command is performed, the entire program memory and Configuration Words (0x2007-0x2008) are erased. Data memory will also be erased if the $\overline{\text{CPD}}$ bit in the Configuration Word is programmed (clear). See **Section 3.2.5 “Erase Algorithms”** for erase sequences.

Note:	All bulk erase operations must take place between 4.5V and 5.5V VDD.

FIGURE 3-13: BULK ERASE PROGRAM MEMORY COMMAND



3.2.6.11 Bulk Erase Data Memory

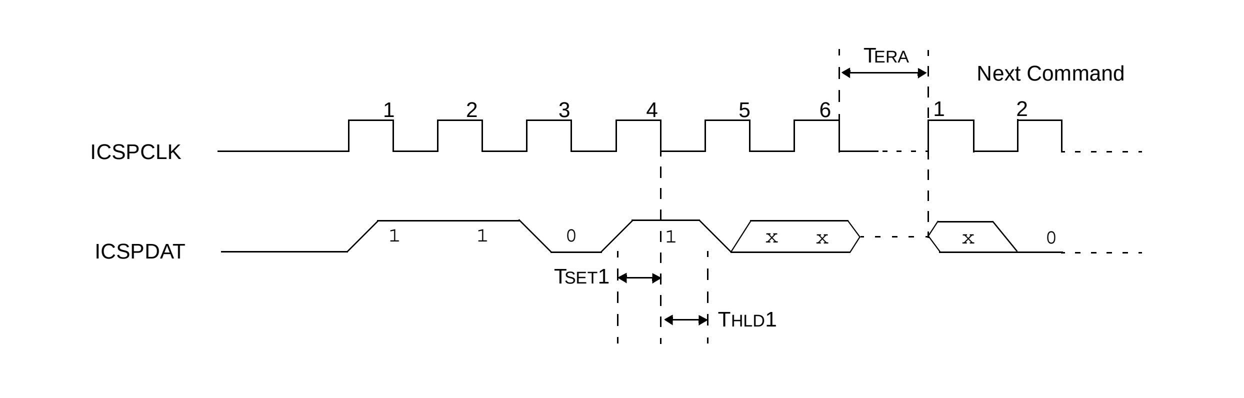
To perform an erase of the data memory, the following sequence must be performed.

1. Perform a Bulk Erase Data Memory command.
2. Wait TERA to complete bulk erase.

Data memory won't erase if code-protected ($\overline{\text{CPD}}$ = 0).

> **Note 1:** All bulk erase operations must take place between 4.5V and 5.5V VDD.
>
> **2:** Data memory won't erase if code-protected ($\overline{\text{CPD}}$ = 0) .

FIGURE 3-14: BULK ERASE DATA MEMORY COMMAND



3.2.6.12 Row Erase Program Memory

This command erases the 16-word row of program memory pointed to by PC<11:4>. If the program memory array is protected ($\overline{\text{CP}}$ = 0) or the PC points to configuration memory (>0x2000), the command is ignored.

To perform a Row Erase Program Memory, the following sequence must be performed.

1. Execute a Row Erase Program Memory command.
2. Wait TERA to complete a row erase.

FIGURE 3-15: ROW ERASE PROGRAM MEMORY COMMAND

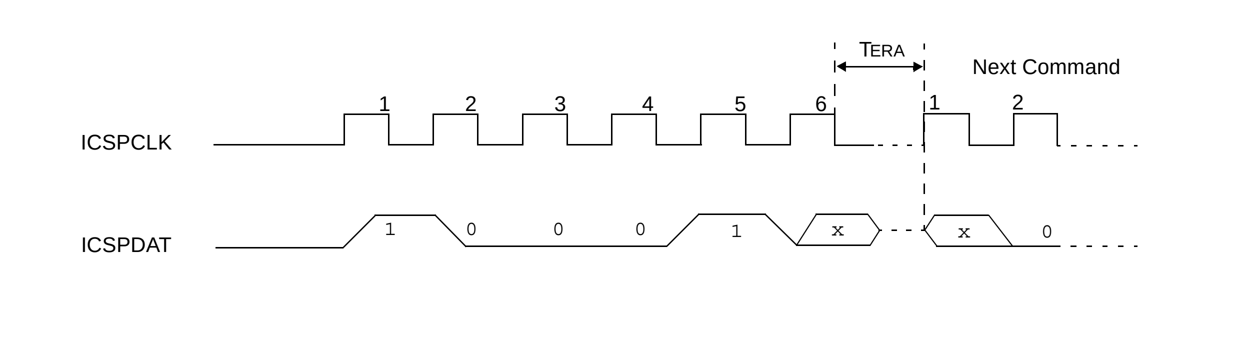
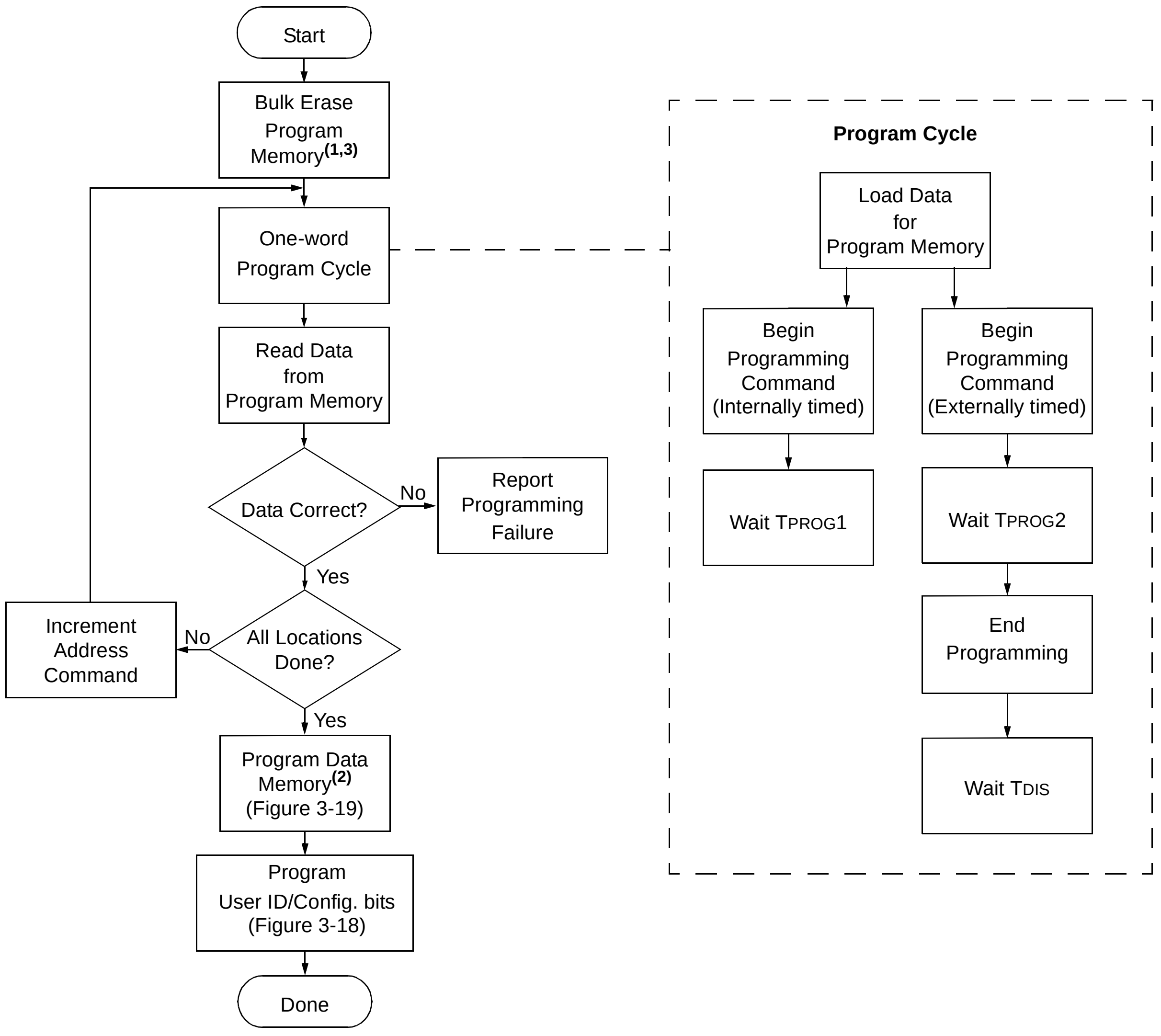


FIGURE 3-16: ONE-WORD PROGRAMMING FLOWCHART

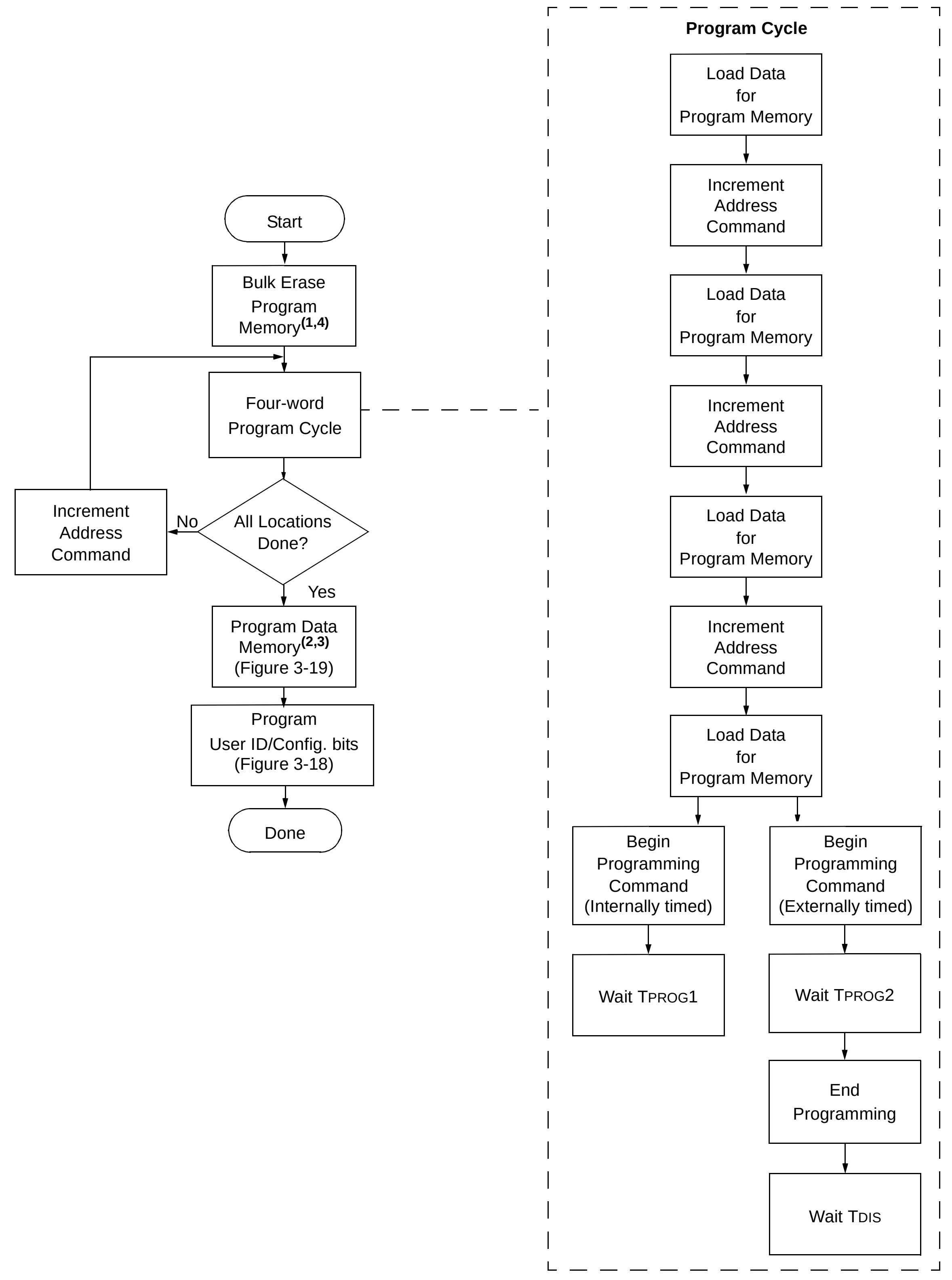

Start
Bulk Erase Program Memory(1,3)
One-word Program Cycle
Read Data from Program Memory
Data Correct?
No
Report Programming Failure
Yes
All Locations Done?
No
Increment Address Command
Yes
Program Data Memory(2) (Figure 3-19)
Program User ID/Config. bits (Figure 3-18)
Done
Program Cycle
Load Data for Program Memory
Begin Programming Command (Internally timed)
Begin Programming Command (Externally timed)
Wait TPROG1
Wait TPROG2
End Programming
Wait TDIS

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

2: This step is optional if the data memory does not require updates.

3: If the device is code-protected or must be completely erased, then bulk erase device per Figure 3-21.

FIGURE 3-17: FOUR-WORD PROGRAMMING FLOWCHART


Start
Bulk Erase Program Memory(1,4)
Four-word Program Cycle
All Locations Done?
No
Yes
Increment Address Command
Program Data Memory(2,3) (Figure 3-19)
Program User ID/Config. bits (Figure 3-18)
Done
Program Cycle
Load Data for Program Memory
Increment Address Command
Load Data for Program Memory
Increment Address Command
Load Data for Program Memory
Increment Address Command
Load Data for Program Memory
Begin Programming Command (Internally timed)
Begin Programming Command (Externally timed)
Wait TPROG1
Wait TPROG2
End Programming
Wait TDIS

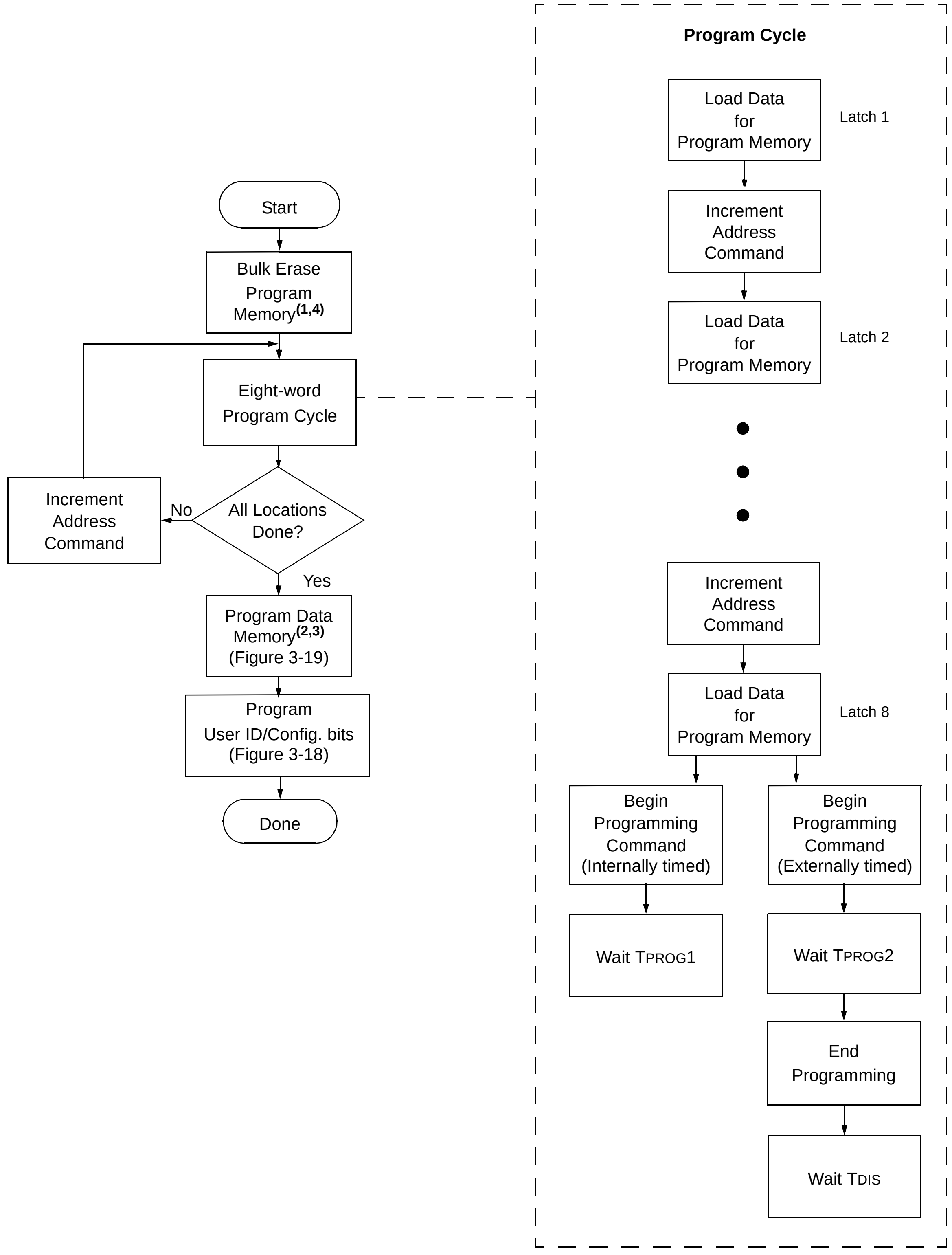
Note 1: This step is optional if device is erased or not previously programmed.

2: Verification in Four-Word mode is accomplished after programming by reading back the entire memory.

3: This step is optional if the data memory does not require updates.

4: If the device is code-protected or must be completely erased, then bulk erase device per Figure 3-21.

FIGURE 3-18: EIGHT-WORD PROGRAMMING FLOWCHART


Program Cycle
Start
Bulk Erase Program Memory(1,4)
Eight-word Program Cycle
All Locations Done?
No
Increment Address Command
Yes
Program Data Memory(2,3) (Figure 3-19)
Program User ID/Config. bits (Figure 3-18)
Done
Load Data for Program Memory
Latch 1
Increment Address Command
Load Data for Program Memory
Latch 2
Increment Address Command
Load Data for Program Memory
Latch 8
Begin Programming Command (Internally timed)
Wait TPROG1
Begin Programming Command (Externally timed)
Wait TPROG2
End Programming
Wait TDIS

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

2: Verification in Eight-Word mode is accomplished after programming by reading back the entire memory.

3: This step is optional if the data memory does not require updates.

4: If the device is code-protected or must be completely erased, then bulk erase device per Figure 3-21.

FIGURE 3-19: PROGRAM FLOWCHART – PIC16F88X CONFIGURATION MEMORY

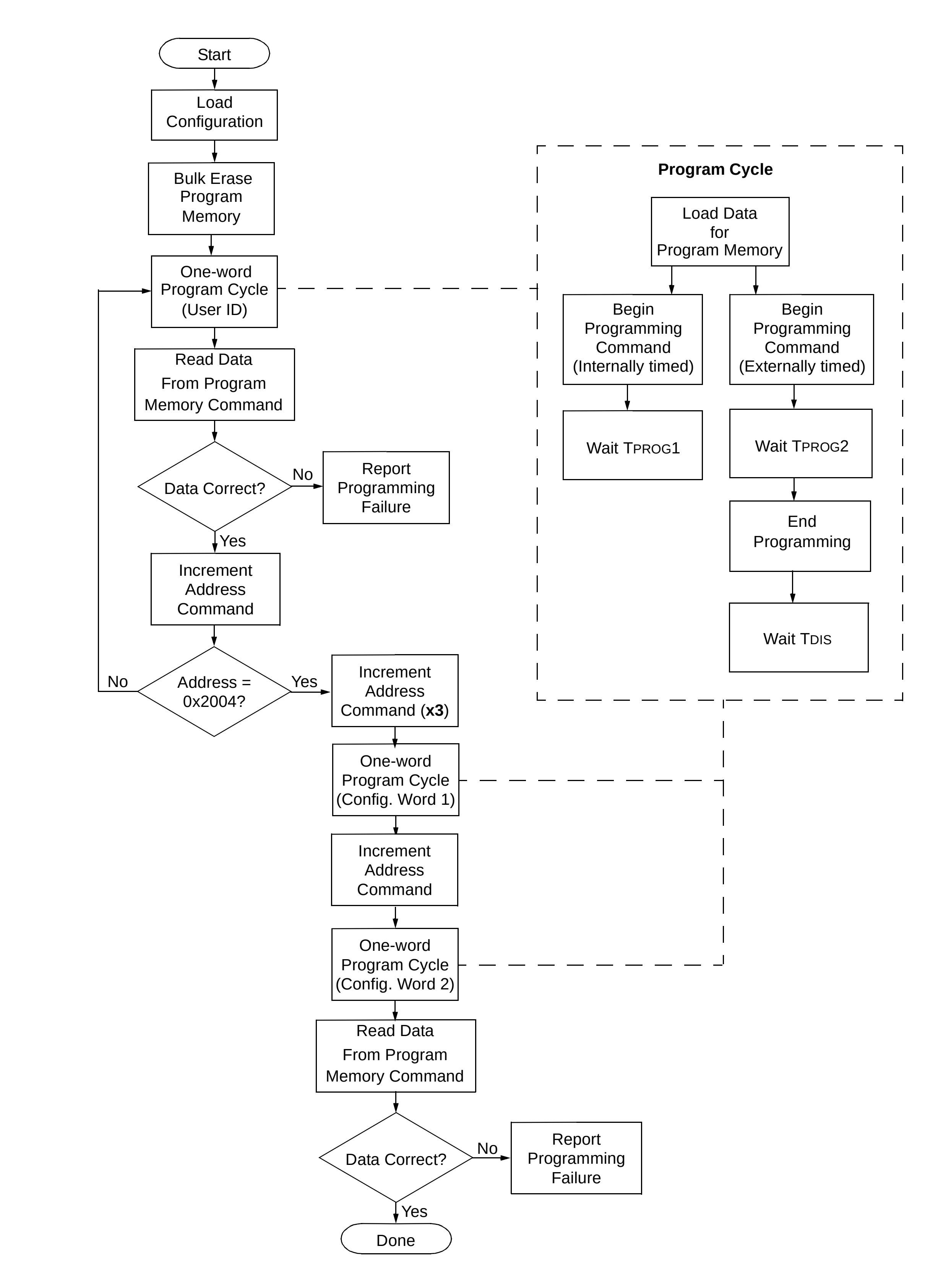

Start
Load Configuration
Bulk Erase Program Memory
One-word Program Cycle (User ID)
Read Data From Program Memory Command
Data Correct?
No
Report Programming Failure
Yes
Increment Address Command
Address = 0x2004?
No
Yes
Increment Address Command (x3)
One-word Program Cycle (Config. Word 1)
Increment Address Command
One-word Program Cycle (Config. Word 2)
Read Data From Program Memory Command
Data Correct?
No
Report Programming Failure
Yes
Done
Program Cycle
Load Data for Program Memory
Begin Programming Command (Internally timed)
Begin Programming Command (Externally timed)
Wait TPROG1
Wait TPROG2
End Programming
Wait TDIS

FIGURE 3-20: PROGRAM FLOWCHART – PIC16F88X DATA MEMORY

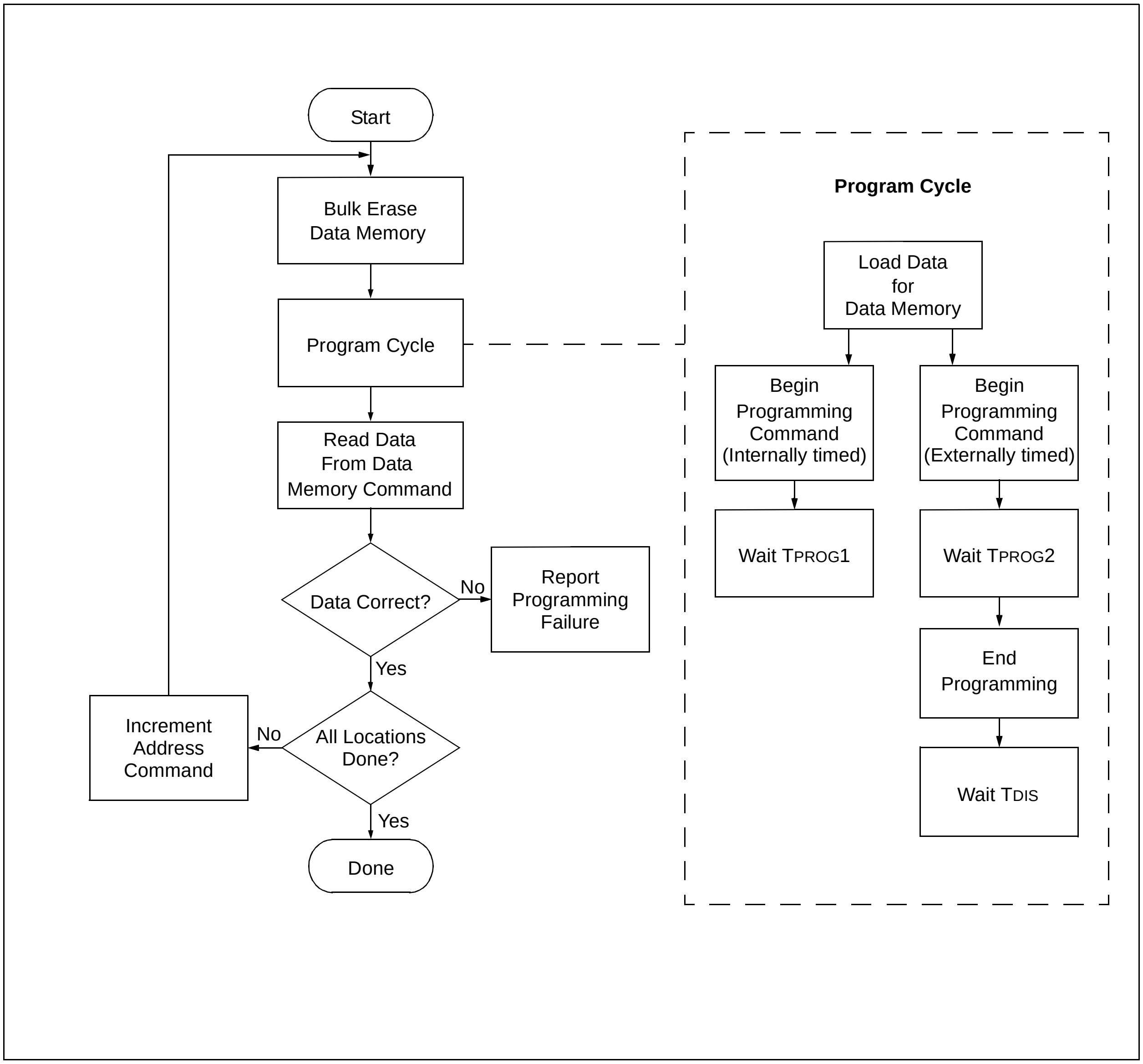
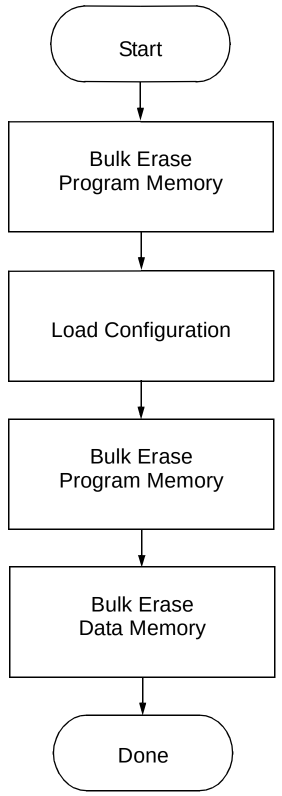

Start
Bulk Erase Data Memory
Program Cycle
Read Data From Data Memory Command
Data Correct?
No
Report Programming Failure
Yes
All Locations Done?
No
Increment Address Command
Yes
Done
Program Cycle
Load Data for Data Memory
Begin Programming Command (Internally timed)
Begin Programming Command (Externally timed)
Wait TPROG1
Wait TPROG2
End Programming
Wait TDIS

FIGURE 3-21: PROGRAM FLOWCHART – ERASE FLASH DEVICE



Note: This sequence does not erase the Calibration Word at address 2009h.

4.0 CONFIGURATION WORD

The PIC16F88X has several Configuration bits. These bits can be programmed (reads '0'), or left unchanged (reads '1'), to select various device configurations.

4.1 Low Voltage Programming (LVP) Bit

The LVP bit in Configuration register Configuration Word 1 enables low voltage ICSP programming. The LVP bit defaults to a '1' following an erase. If Low Voltage Programming mode is not used, the LVP bit can be programmed to a '0' and RB3/PGM becomes a digital I/O pin. However, the LVP bit may only be programmed by entering the high voltage ICSP mode, where $\overline{\text{MCLR}}$/VPP 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 normal high voltage ICSP mode is always available, regardless of the state of the LVP bit, by applying VIHH to the $\overline{\text{MCLR}}$/VPP pin.

2: While in Low Voltage ICSP mode, the RB3 pin can no longer be used as a general purpose I/O.

3: If the device Master Clear is disabled, verify that either of the following is done to ensure proper entry into ICSP mode:

a) disable Low Voltage Programming (Config Word 1<12> = 0); or

b) make certain that RB3/PGM is held low during entry into ICSP.

REGISTER 4-1: CONFIGURATION WORD 1 (ADDRESS: 2007h)

R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
$\overline{\text{DEBUG}}$	LVP	FCMEN	IESO	BOREN1	BOREN0	$\overline{\text{CPD}}$
bit 13						bit 7

R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
$\overline{\text{CP}}$	$\overline{\text{MCLRE}}$	$\overline{\text{PWRTE}}$	WDTEN	FOSC2	FOSC1	FOSC0
bit 6						bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘1’
-n = Default value	‘1’ = Bit is erased	‘0’ = Bit is programmed
P = Programmable bit	x = Bit is unknown	

bit 13 **$\overline{\text{DEBUG}}$:** Debugger Mode bit (this bit is documented as ***Reserved*** in the data sheet)
1 = Background debugger function not enabled
0 = Background debugger functional

bit 12 **LVP:** Low Voltage Programming Enable bit
1 = RB3/PGM pin has PGM function, low voltage programming enabled
0 = RB3 pin is digital I/O, HV on $\overline{\text{MCLR}}$ must be used for programming

bit 11 **FCMEN:** Fail-Safe Clock Monitor Enable bit
1 = Fail-Safe Clock is enabled
0 = Fail-Safe Clock is disabled

bit 10 **IESO:** Internal/External Switch Over bit
1 = Internal/External Switch Over mode enabled
0 = Internal/External Switch Over mode disabled

bit 9-8 **BOREN<1:0>:** Brown-out Reset Selection bits
11 = BOR enabled
10 = BOR enabled during operation and disabled in Sleep
01 = BOR controlled by SBOREN bit (PCON<4>)
00 = BOR disabled

bit 7 **$\overline{\text{CPD}}$:** Data EE Memory Code Protection bit
1 = Code protection off
0 = Data EE memory code-protected

bit 6 **$\overline{\text{CP}}$:** Flash Program Memory Code Protection bit
<u>PIC16F886/887</u>
1 = Code protection off
0 = 0000h to 1FFFh code protection on
<u>PIC16F883/884</u>
1 = Code protection off
0 = 0000h to 0FFFh code protection on

bit 5 **$\overline{\text{MCLRE}}$:** $\overline{\text{MCLR}}$/V_{PP}/RE3 Pin Function Select bit
1 = $\overline{\text{MCLR}}$/V_{PP}/RE3 pin function is digital input
0 = $\overline{\text{MCLR}}$/V_{PP}/RE3 pin function is $\overline{\text{MCLR}}$

bit 4 **$\overline{\text{PWRTE}}$:** Power-up Timer Enable bit
1 = PWRT disabled
0 = PWRT enabled

bit 3 **WDTEN:** Watchdog Timer Enable bit
1 = WDT enabled
0 = WDT disabled

REGISTER 4-1: CONFIGURATION WORD 1 (ADDRESS: 2007h) (CONTINUED)

bit 2-0 **FOSC<2:0>:** Oscillator Selection bits
111 = RC oscillator: CLKO function on RA6/OSC2/CLKO pin, RC on RA7/OSC1/CLKI
110 = RCIO oscillator: I/O function on RA6/OSC2/CLKO pin, RC on RA7/OSC1/CLKI
101 = INTOSC oscillator: CLKO function on RA6/OSC2/CLKO pin, I/O function on RA7/OSC1/CLKI
100 = INTOSCIO oscillator: I/O function on RA6/OSC2/CLKO pin, I/O function on RA7/OSC1/CLKI
011 = EC oscillator: I/O function on RA6/OSC2/CLKO pin, CLKI on RA7/OSC1/CLKI
010 = HS oscillator: High-speed crystal/resonator on RA6/OSC2/CLKO pin and RA7/OSC1/CLKI
001 = XT oscillator: Crystal/resonator on RA6/OSC2/CLKO pin and RA7/OSC1/CLKI
000 = LP oscillator: Low-power crystal on RA6/OSC2/CLKO pin and RA7/OSC1/CLKI

REGISTER 4-2: CONFIGURATION WORD 2 (ADDRESS: 2008h)

U-1	U-1	U-1	R/P-1	R/P-1	R/P-1	U-1
—	—	—	WRT1	WRT0	BOR4V	—
bit 13						bit7

U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—
bit 6						bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘1’
-n = Default value	‘1’ = Bit is erased	‘0’ = Bit is programmed
P = Programmable bit	x = Bit is unknown	

bit 13-11 **Unimplemented:** Read as ‘1’

bit 10-9 **WRT<1:0>:** Flash Program Memory Write Enable bits
PIC16F886/887
00 = 0000h to 0FFFh write protected, 1000h to 1FFFh may be modified by EECON control
01 = 0000h to 07FFh write protected, 0800h to 1FFFh may be modified by EECON control
10 = 0000h to 00FFh write protected, 0100h to 1FFFh may be modified by EECON control
11 = Write protection off
PIC16F883/884
00 = 0000h to 07FFh write protected, 0800h to 0FFFh may be modified by EECON control
01 = 0000h to 03FFh write protected, 0400h to 0FFFh may be modified by EECON control
10 = 0000h to 00FFh write protected, 0100h to 0FFFh may be modified by EECON control
11 = Write protection off

bit 8 BOR4V: Brown-out Reset Selection bit
1 = Brown-out Reset set to 4V
0 = Brown-out Reset set to 2.1V

bit 7-0 **Unimplemented:** Read as ‘1’

REGISTER 4-3: CALIBRATION WORD (CONFIG: 2009h)

U-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
—	FCAL6	FCAL5	FCAL4	FCAL3	FCAL2	FCAL1
bit 13						bit 7

R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
FCAL0	POR2	POR1	POR0	BOR2	BOR1	BOR0
bit 6						bit 0

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

bit 13 **Unimplemented**

bit 12-6 **FCAL<6:0>:** Internal Oscillator Calibration bits[(2)]
`0111111` = Maximum frequency
•
•
`0000001`
`0000000` = Center frequency. Oscillator is running at the calibrated frequency
`1111111`
•
•
`1000000` = Minimum frequency

bit 5-3 **POR<2:0>:** POR Calibration bits[(2)]
`111` = Adjust POR voltage *(7.5%)*
`110` = Adjust POR voltage *(5.0%)*
`101` = Target POR voltage (*2.5%*)
`100` = Adjust POR voltage *(center)*
`000` = Adjust POR voltage *(center)*
`001` = Target POR voltage (-2.5%)
`010` = Adjust POR voltage *(-5.0%)*
`011` = Adjust POR voltage *(-7.5%)*

bit 2-0 **BOR<2:0>:** BOR Calibration bits[(2)]
`000` = Target BOR voltage *(2.1V)*
`001` = Adjust BOR voltage (-50*mV)*
`010` = Adjust BOR voltage *(-100mV)*
`011` = Adjust BOR voltage *(-150mV)*
`100` = Target BOR voltage (*2.1*V)
`101` = Adjust BOR voltage *(50mV)*
`110` = Adjust BOR voltage *(100mV)*
`111` = Adjust BOR voltage *(150mV)*

Note 1: This location does not participate in bulk erase operations.

2: The calibration bits must be read, preserved, then replaced by the user during Program Memory Bulk Erase operation with PC = 2009h.

4.2 Device ID Word

The device ID word for the PIC16F88X is located at 2006h. This location can not be erased.

TABLE 4-1: DEVICE ID VALUES

Device	Device ID Values	
	Dev	Rev
PIC16F883	`10 0000 001`	`x xxxx`
PIC16F884	`10 0000 010`	`x xxxx`
PIC16F886	`10 0000 011`	`x xxxx`
PIC16F887	`10 0000 100`	`x xxxx`

5.0 CODE PROTECTION

For PIC16F88X, once the $\overline{\text{CP}}$ bit is programmed to '`0`', all program memory locations read all '`0`'s. Further programming is disabled for the entire program memory.

Data memory is protected with its own Code-Protect bit ($\overline{\text{CPD}}$). When enabled, the data memory can still be programmed and read using the EECON1 register (See the applicable data sheet for more information).

The user ID locations and the Configuration Word can be programmed and read out regardless of the state of the $\overline{\text{CP}}$ and $\overline{\text{CPD}}$ bits.

5.1 Disabling Code Protection

It is recommended to use the procedure in Figure 3-21 to disable code protection of the device. This sequence will erase the program memory, data memory, Configuration Word (0x2007-0x2008) and user ID locations (0x2000-0x2003). The Calibration Words (0x2009) **will not** be erased.

> **Note:** To ensure system security, if $\overline{\text{CPD}}$ bit = `0`, Bulk Erase Program Memory command will also erase data memory.

5.2 Embedding Configuration Words and User ID Information in the Hex File

To allow portability of code, the programmer is required to read the Configuration Words and user ID locations from the hex file when loading the hex file. If 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 must be included. An option to not include this information may be provided.

Specifically for the PIC16F88X, the data memory should also be embedded in the hex file (see **Section 5.3.2 "Embedding Data Memory Contents In Hex File"**).

Microchip Technology Incorporated feels strongly that this feature is important for the benefit of the end customer.

5.3 Checksum Computation

5.3.1 CHECKSUM

Checksum is calculated by reading the contents of the PIC16F88X memory locations and adding up the opcodes up to the maximum user addressable location, (e.g., 0x1FFF for DSTEMP). Any carry bits exceeding 16 bits are neglected. Finally, the Configuration Words (appropriately masked) are added to the checksum. Checksum computation for the PIC16F88X devices is shown in Table 5-1.

The checksum is calculated by summing the following:

- The contents of all program memory locations
- The Configuration Words, appropriately masked
- Masked user ID locations (when applicable)

The Least Significant 16 bits of this sum is the checksum.

The following table describes how to calculate the checksum for each device. Note that the checksum calculation differs depending on the code-protect setting. Since the program memory locations read out zeroes when code-protected, the table describes how to manipulate the actual program memory values to simulate values that would be read from a protected device. When calculating a checksum by reading a device, the entire program memory can simply be read and summed. The Configuration Words and user ID locations can always be read regardless of code-protect setting.

> **Note:** Some older devices have an additional value added in the checksum. This is to maintain compatibility with older device programmer checksums.

TABLE 5-1: CHECKSUM COMPUTATIONS

Device	Code Protect	Checksum*	Blank Value	0x25E6 at 0 and Max Address
PIC16F883	$\overline{CP}$ = 1	SUM[0x0000:0x0FFF] + (CFG1 & 0x3FFF) + (CFG2 & 0x0700)	0x36FF	0x02CD
	$\overline{CP}$ = 0	(CFG1 & 0x3FFF) + (CFG2 & 0x0700) + SUM_ID	0x7DBE	0x498C
PIC16F884	$\overline{CP}$ = 1	SUM[0x0000:0x0FFF] + (CFG1 & 0x3FFF) + (CFG2 & 0x0700)	0x36FF	0x02CD
	$\overline{CP}$ = 0	(CFG1 & 0x3FFF) + (CFG2 & 0x0700) + SUM_ID	0x7DBE	0x498C
PIC16F886	$\overline{CP}$ = 1	SUM[0x0000:0x1FFF] + (CFG1 & 0x3FFF) + (CFG2 & 0x0700)	0x26FF	0xF2CD
	$\overline{CP}$ = 0	(CFG1 & 0x3FFF) + (CFG2 & 0x0700) + SUM_ID	0x6DBE	0x398C
PIC16F887	$\overline{CP}$ = 1	SUM[0x0000:0x1FFF] + (CFG1 & 0x3FFF) + (CFG2 & 0x0700)	0x26FF	0xF2CD
	$\overline{CP}$ = 0	(CFG1 & 0x3FFF) + (CFG2 & 0x0700) + SUM_ID	0x6DBE	0x398C

Legend: CFG = Configuration Word. Example calculations assume Configuration Word is erased (all '1's).
SUM[a:b] = [Sum of locations a to b inclusive]
SUM_ID = User ID locations masked by 0xF then made into a 16-bit value with ID0 as the Most Significant nibble.
For example, ID0 = 0x1, ID1 = 0x2, ID3 = 0x3, ID4 = 0x4, then SUM_ID = 0x1234.
The 4 LSbs of the unprotected checksum is used for the example calculations.
*Checksum = [Sum of all the individual expressions] MODULO [0xFFFF]
+ = Addition
& = Bitwise AND

5.3.2 EMBEDDING DATA MEMORY CONTENTS IN HEX FILE

The programmer should be able to read data memory information from a hex file and conversely (as an option), write data memory contents to a hex file along with program memory information and Configuration Words (0x2007-0x2008) and user ID (0x2000-0x2003) information.

The physical address range of the 256 data memory is 0x0000-0x00FF. However, these addresses are logically mapped to address 0x2100-0x21FF for use in writing assembly code. This provides a way of differentiating between the data and program memory locations in this range. The format for data memory storage is one data byte per address location, LSb aligned. A simple example of data memory is given below:

```
org 0x2100
de “My Prgram, v1.0”, 0
```

6.0 PROGRAM/VERIFY MODE ELECTRICAL CHARACTERISTICS

TABLE 6-1: AC/DC CHARACTERISTICS TIMING REQUIREMENTS FOR PROGRAM/VERIFY MODE

AC/DC CHARACTERISTICS		**Standard Operating Conditions (unless otherwise stated)** Operating Temperature -40°C ≤ TA ≤ +85°C Operating Voltage 4.5V ≤ VDD ≤ 5.5V				
Sym	**Characteristics**	**Min**	**Typ**	**Max**	**Units**	**Conditions/Comments**
		General				
VDD	VDD level for read/write operations, program and data memory	2.0	—	5.5	V	
	VDD level for bulk erase operations, program and data memory	4.5	—	5.5	V	
VPP	High voltage on $\overline{\text{MCLR}}$ for Program/Verify mode entry	10	—	12	V	
TVHHR	$\overline{\text{MCLR}}$ rise time (VSS to VHH) for Program/Verify mode entry	—	—	1.0	µs	
TPPDP	Hold time after VPP changes	5	—	—	µs	
VIH1	(ICSPCLK, ICSPDAT) input high level	0.8 VDD	—	—	V	
VIL1	(ICSPCLK, ICSPDAT) input low level	0.2 VDD	—	—	V	
TSET0	ICSPCLK, ICSPDAT setup time before $\overline{\text{MCLR}}$↑ (Program/Verify mode selection pattern setup time)	100	—	—	ns	
THLD0	Hold time after VPP changes	0	—	1	µs	
		Serial Program/Verify				
TSET1	Data in setup time before clock↓	100	—	—	ns	
THLD1	Data in hold time after clock↓	100	—	—	ns	
TDLY1	Data input not driven to next clock input (delay required between command/data or command/command)	1.0	—	—	µs	
TDLY2	Delay between clock↓ to clock↑ of next command or data	1.0	—	—	µs	
TDLY3	Clock↑ to data out valid (during a Read Data command)		—	80	ns	
TERA	Erase cycle time	—	5	6	ms	
TPROG1	Programming cycle time (internally timed)	2 5	—	3 6	ms	Program memory Data memory
TPROG2	Programming cycle time (externally timed)	2	—	2.5	ms	10°C ≤ TA ≤ +40°C Program memory
TDIS	Time delay from program to compare (HV discharge time)	100	—	—	µs	

APPENDIX A: REVISION HISTORY

Revision A (3/06)

Original release.

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

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

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

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE**.** Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

Trademarks

The Microchip name and logo, the Microchip logo, Accuron, dsPIC, KEELOQ, microID, MPLAB, PIC, PICmicro, PICSTART, PRO MATE, PowerSmart, rfPIC, and SmartShunt are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AmpLab, FilterLab, Migratable Memory, MXDEV, MXLAB, SEEVAL, SmartSensor and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, dsPICDEM, dsPICDEM.net, dsPICworks, ECAN, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, Linear Active Thermistor, MPASM, MPLIB, MPLINK, MPSIM, PICkit, PICDEM, PICDEM.net, PICLAB, PICtail, PowerCal, PowerInfo, PowerMate, PowerTool, REAL ICE, rfLAB, rfPICDEM, Select Mode, Smart Serial, SmartTel, Total Endurance, UNI/O, WiperLock and Zena are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

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

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

Printed on recycled paper.

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
ISO/TS 16949:2002

Microchip received ISO/TS-16949:2002 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona and Mountain View, California in October 2003. The Company's quality system processes and procedures are for its PICmicro® 8-bit MCUs, 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.

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://support.microchip.com
Web Address:
www.microchip.com

Atlanta
Alpharetta, GA
Tel: 770-640-0034
Fax: 770-640-0307

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
Farmington Hills, MI
Tel: 248-538-2250
Fax: 248-538-2260

Kokomo
Kokomo, IN
Tel: 765-864-8360
Fax: 765-864-8387

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

San Jose
Mountain View, CA
Tel: 650-215-1444
Fax: 650-961-0286

Toronto
Mississauga, Ontario,
Canada
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8528-2100
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8676-6200
Fax: 86-28-8676-6599

China - Fuzhou
Tel: 86-591-8750-3506
Fax: 86-591-8750-3521

China - Hong Kong SAR
Tel: 852-2401-1200
Fax: 852-2401-3431

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8203-2660
Fax: 86-755-8203-1760

China - Shunde
Tel: 86-757-2839-5507
Fax: 86-757-2839-5571

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian
Tel: 86-29-8833-7250
Fax: 86-29-8833-7256

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-4182-8400
Fax: 91-80-4182-8422

India - New Delhi
Tel: 91-11-5160-8631
Fax: 91-11-5160-8632

India - Pune
Tel: 91-20-2566-1512
Fax: 91-20-2566-1513

Japan - Yokohama
Tel: 81-45-471- 6166
Fax: 81-45-471-6122

Korea - Gumi
Tel: 82-54-473-4301
Fax: 82-54-473-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Penang
Tel: 60-4-646-8870
Fax: 60-4-646-5086

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-572-9526
Fax: 886-3-572-6459

Taiwan - Kaohsiung
Tel: 886-7-536-4818
Fax: 886-7-536-4803

Taiwan - Taipei
Tel: 886-2-2500-6610
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-399
Fax: 43-7242-2244-393

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

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

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

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

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

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

UK - Wokingham
Tel: 44-118-921-5869
Fax: 44-118-921-5820

02/16/06