PIC18F0XX

Programming for PIC18F0XX FLASH MCUs

1.0   DEVICE OVERVIEW

This document includes the programming specifications for the following devices:

- PIC18F010
- PIC18F020

2.0   PROGRAMMING THE PIC18F0XX

The PIC18F0XX can be programmed using a serial method while in users' system, allowing increased design flexibility. This programming specification applies to PIC18F0XX devices in all package types.

2.1   Hardware Requirements

The PIC18F0XX requires two programmable power supplies, one for VDD and one for VPP. Both supplies should have a minimum resolution of 0.25V.

2.2   Programming Mode

The Programming mode for the PIC18F0XX allows programming of user program memory, EEDATA memory, special locations used for ID, and the configuration words for the PIC18F0XX.

2.3   Pin Diagrams

The pin diagrams for the PIC18F0XX family are shown below in Figure 2-1.
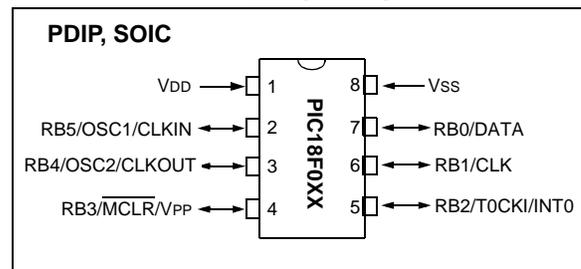
FIGURE 2-1:   **PIC18F0XX FAMILY PIN DIAGRAMS**

TABLE 2-1:   PIN DESCRIPTIONS (DURING PROGRAMMING)

Pin Name	During Programming		
	Pin Name	Pin Type	Pin Description
MCLR/VPP	VPP	P	Programming Enable
VDD	VDD	P	Power Supply
Vss	Vss	P	Ground
RB1/CLK	RB1	I	Serial Clock
RB0/DATA	RB0	I/O	Serial Data

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

PIC18F0XX

3.0   PROGRAMMING MODE

3.1   User Program Memory Map

The user memory space extends from 0x0000 to 0x0FFF (4 Kbytes), all of which is on-chip. Table 3-1 shows actual implementation of program memory in the PIC18F0XX family.

**TABLE 3-1:   IMPLEMENTATION OF PROGRAM MEMORY IN THE PIC18F0XX**

Device	Program Memory Size
PIC18F010	0x0000 - 0x07FF (2K)
PIC18F020	0x0000 - 0x0FFF (4K)

In Programming mode, the program memory space extends from 0x0000 to 0x0FFF for the PIC18F020, all of which is user program memory. All devices have ID memory starting at 200000h and configuration at 300000h.

A user may store identification information (ID) in eight ID registers. The ID registers are mapped in [0x200000h through 0x200007h]. The ID locations read out normally, even after code protection.

Location [0x3FFFFEh and 0x3FFFFFh] are reserved for the Device ID bits. The configuration bits are described in Section 4.0. These bits are used by the programmer to identify what device type is being programmed. From this, the programmer can retrieve specific information about the device, such as how much program memory is available.

FIGURE 3-1:   PROGRAM MEMORY MAPPING FOR PIC18F0XX
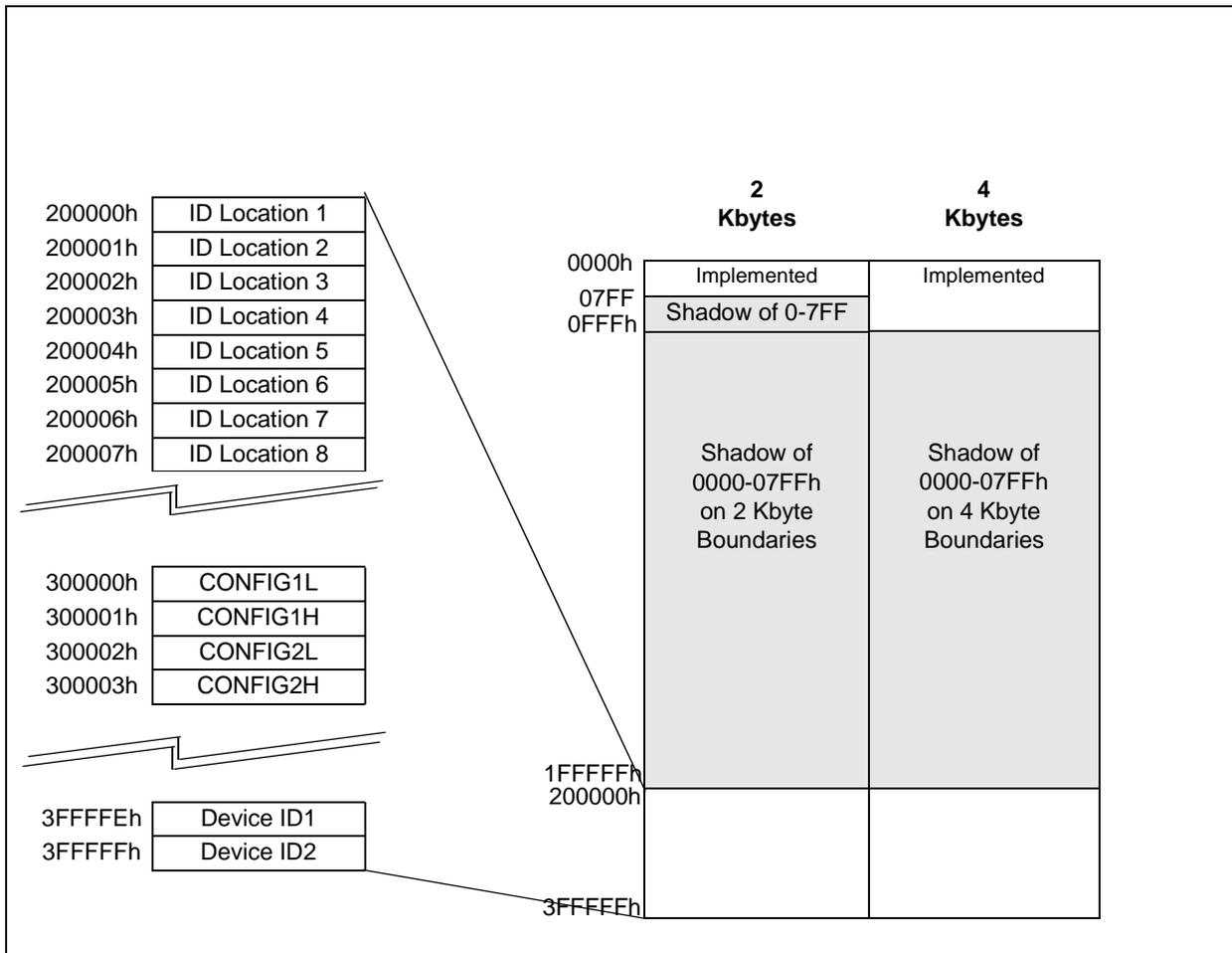
3.2   Program/Verify Mode

The Program/Verify mode is entered by holding pins RB0, RB1, and VDD low while raising $\overline{MCLR}$ pin from VIL to VIHH (high voltage). Once in this mode, the VDD pin is raised to VDD, and the user program memory and the configuration memory can be accessed and programmed in serial fashion.

The sequence that enters the device into the Programming/Verify mode places I/O in the high impedance state.

3.2.1   SERIAL PROGRAM/VERIFY OPERATION

The RB1 pin is used as a clock input pin, and the RB0 pin is used for entering command bits and data input/output during serial operation. To input a command, the clock pin (RB1) is cycled four times. Each command bit is latched on the falling edge of the clock with the Least Significant bit (LSb) of the command being input first. The data on pin RB0 is required to have a minimum setup and hold time (see AC/DC specifications), with respect to the falling edge of the clock. Then the clock pin is cycled 16 times to input or output data. Data is shifted LSb first.

The data is transmitted on the rising edge and latched on the falling edge of the clock.

The commands that are available are described in the following sections:

3.2.2   4-BIT COMMAND

All instructions are 20-bits consisting of a leading 4-bit command (LSb to MSb) followed by a 16-bit operand which depends on the type of command being executed. The device reads the 4-bit command and uses the 16-bit operand to operate on that command. The commands needed to program are shown in Table 3-2.

TABLE 3-2:   COMMANDS FOR PROGRAMMING

| Description	4-Bit Command
Core Instruction (shift in16-bit instruction)	`0000`
Table Read, post-increment	`1001`
Table Write	`1100`
Shift out data holding register	`0010`
Table Write, post-increment by 2	`0101`
Table Write, start programming, post-increment by 2	`0111`

3.2.3   CORE INSTRUCTION

The core instruction passes a 16-bit instruction to the CPU core for execution. This is needed to setup registers, as appropriate, to use the other commands. The address pointer used with these operations is TBLPTRU, TBLPTRH and TBLPTRL at addresses FF6-FF8. The core instruction is used to load the pointer prior to using the READ or WRITE operations.

PIC18F0XX

3.3 High Level Overview of the Programming Process

Figure 3-2 shows the high level overview of the Programming Process. First, blank check the part and, if not blank, perform a bulk erase. Next, program the code memory, data memory and ID locations. Verify the code memory, data memory and ID locations. If no errors are detected, program and verify the configuration bits.

FIGURE 3-2: **HIGH LEVEL PROGRAMMING FLOW CHART**
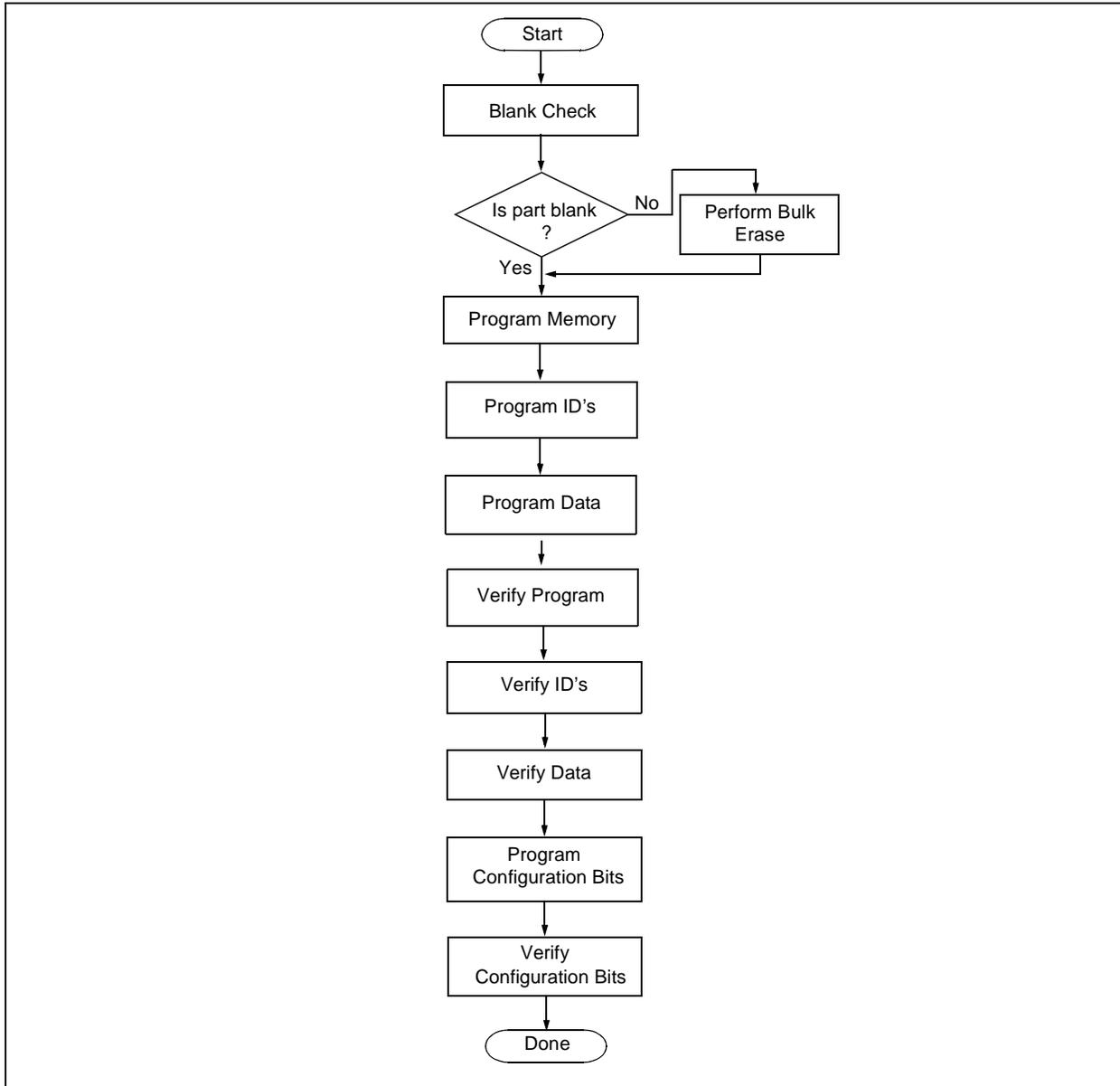
Note:	The following pages provide more detail for each step described above.

3.3.1   BLANK CHECK

Note:	As with any FLASH architecture device, the PIC18F0XX can be written and/or erased in blocks. The PIC18F0XX has two blocks of program memory. For simplicity's sake, this document describes only how to erase and program the entire device. This document is geared to users making the transition from programming EPROM PICmicro® MCUs.

The configuration bits should be checked first and, if either code memory or data memory is protected, these areas must be bulk erased. If both of these are not protected, the individual arrays should be checked one byte at a time. Blank checking the memory array is similar to the Verify Step, except that all bytes are compared against the erased state, with all bits 1. All memory areas must be checked, including any Program, Data, or ID locations. If any Program, Data, ID locations, or any code protect, or any data protect bits read '0', the device should be bulk erased.

**FIGURE 3-3:   BLANK CHECK FLOW CHART PIC18F0XX**
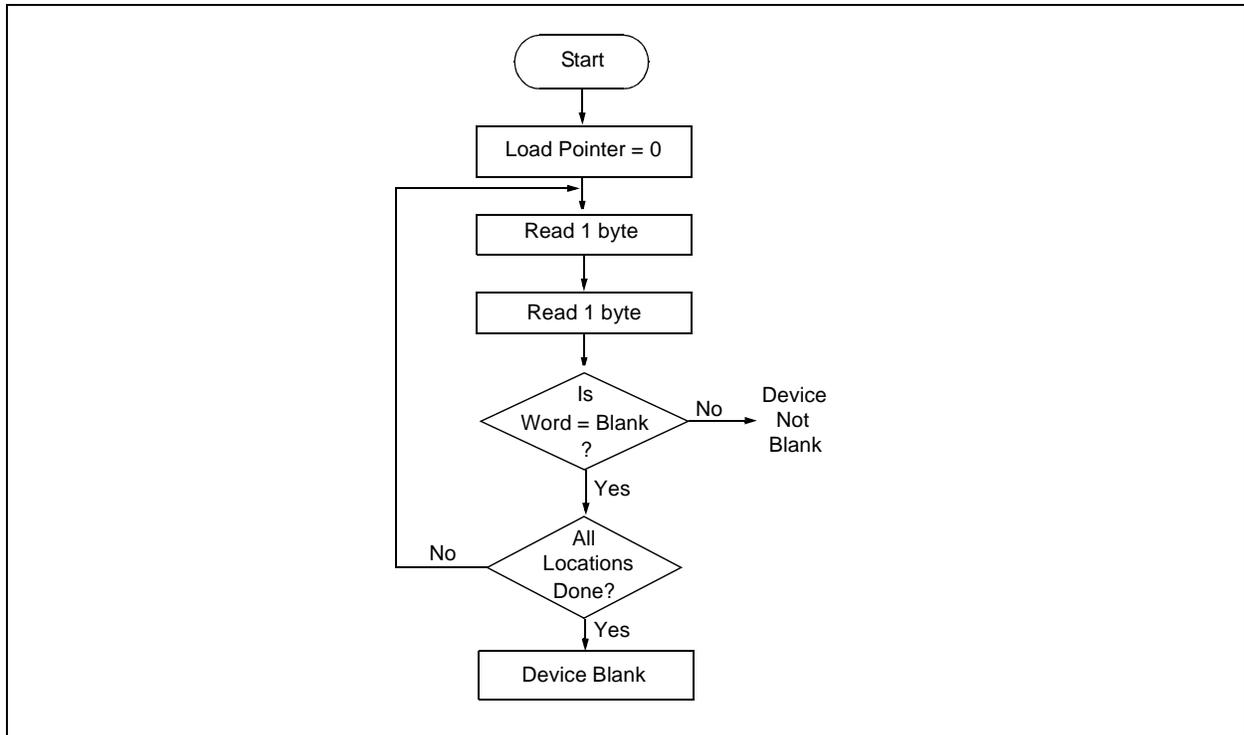
PIC18F0XX

3.4 Erasing Program and Data Memory

Erasing program memory is accomplished by writing 0x80 to address 3C0004h. Data memory is erased by writing 0x81 to the same address. The code sequence is shown in Table 3-3 and the flow chart is shown in Figure 3-4.

TABLE 3-3: ERASING MEMORY

4-Bit Command	Data Payload	Core Instruction
0	0E 3C	MOVLW 0X3C
0	6E F8	MOVWF TBLPTRU
0	0E 00	MOVLW 0X00
0	6E F7	MOVWF TBLPTRH
0	OE 04	MOVLW 0X04
0	6E F6	MOVWF TBLPTR
C	00 80	WRITE 0X80 TO 3C0004 TO ERASE PROGRAM MEMORY
0	00 00	Clock needed to start process
C	00 81	WRITE 0X81 TO 3C0004 TO ERASE PROGRAM MEMORY
0	00 00	Clock needed to start process

FIGURE 3-4: PROGRAM FLOW CHART - PIC18F0XX ERASING MEMORY

© 2001 Microchip Technology Inc.
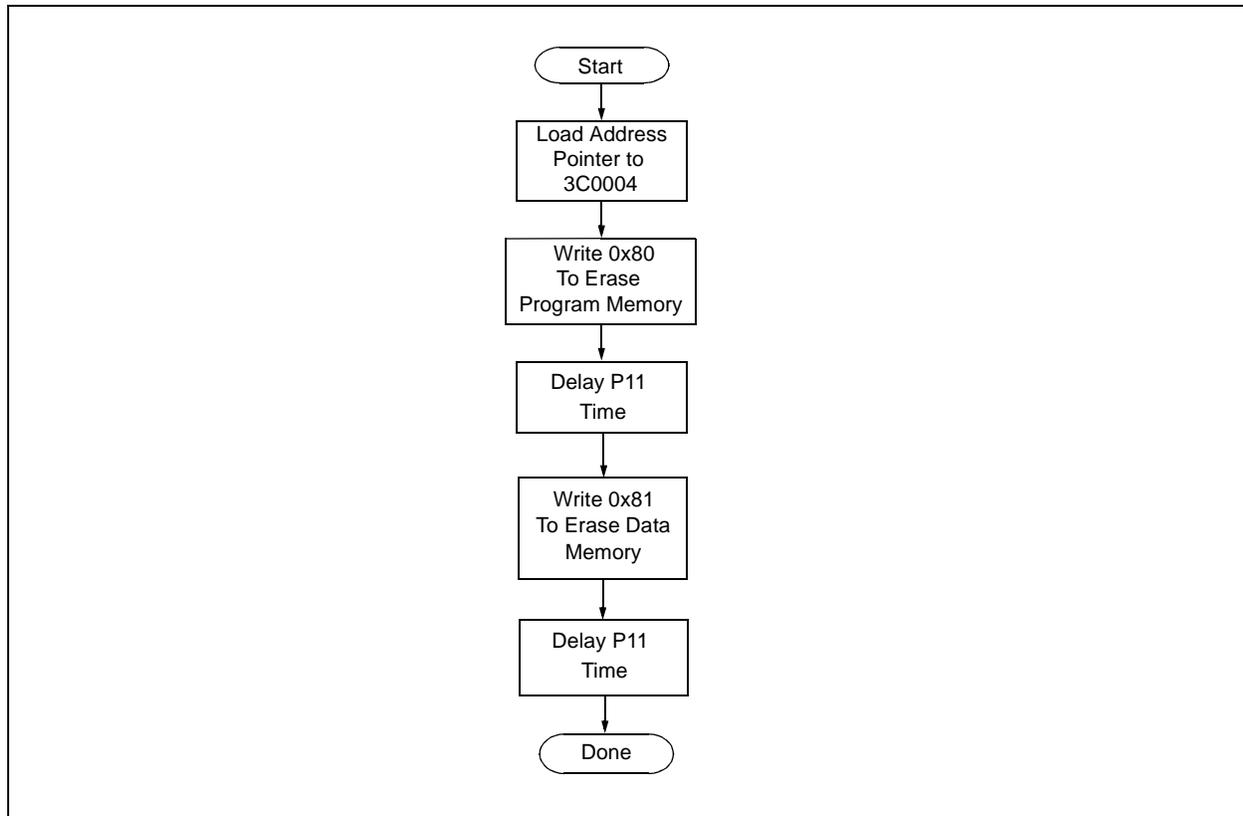
3.5 PROGRAMMING CODE MEMORY

Programming code memory, as shown in Table 3-4, requires first setting the memory pointer, then writing two words at a time to memory. Each memory write advances the memory pointer to the next word address, so additional words may be written without reloading the memory pointer. Continue writing two words at a time until the entire memory array has been written.

The memory pointer is 24-bits long and must be written as three separate bytes. The commands and command sequences shown in the following tables are the only ones recommended by Microchip:

**TABLE 3-4: LOADING MEMORY ADDRESS POINTER**

4-Bit Command	Data Payload	Core Instruction
0	0E<ADDRUpper>	MOVLW 00
0	6E F8	MOVWF TBLPTRU
0	0E<ADDRHigh	MOVLW 00
0	6E F7	MOVWF TBLPTRH
0	0E<ADDRLow	MOVLW 00
0	6E F6	MOVWF TBLPTRL

Once the memory pointer is set, the memory may be written in 2-word pairs. This command sequence must be repeated until all code words are written. The 2-word pairs are shown in the following table:

TABLE 3-5: PROGRAMMING 2-WORDS

4-Bit Command	Data Payload	Core Instruction
5	<Low Word, 16 bits>	—
7	<High Word, 16 bits>	—
0	00 00	NOP

FIGURE 3-5: PROGRAM MEMORY FLOW CHART PIC18F0XX
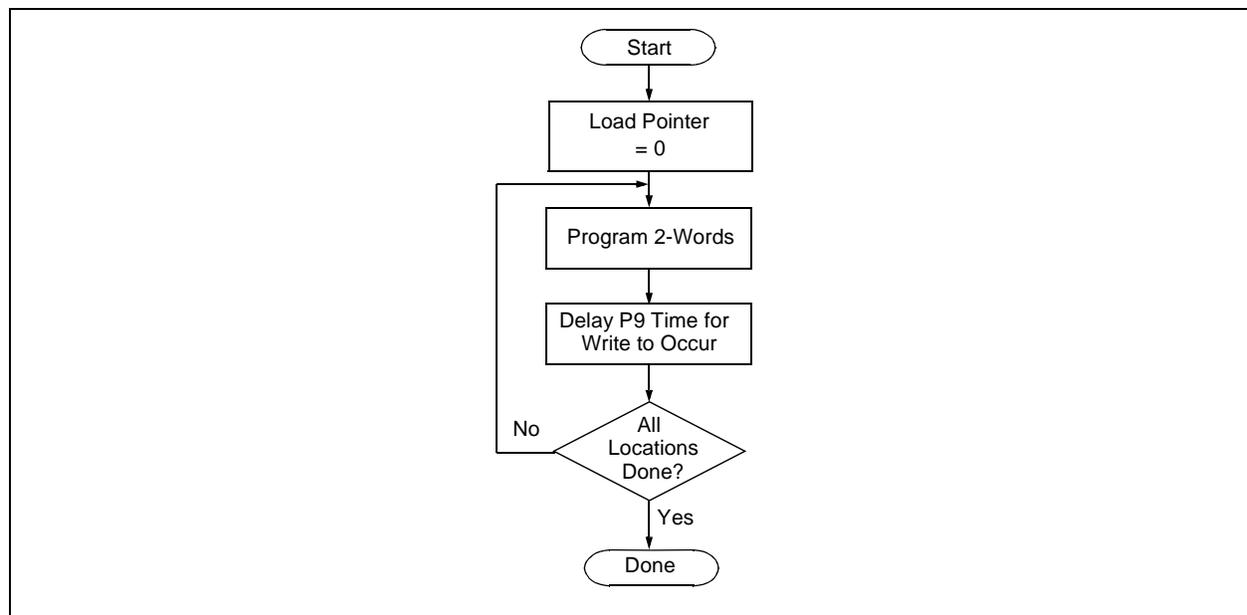
PIC18F0XX

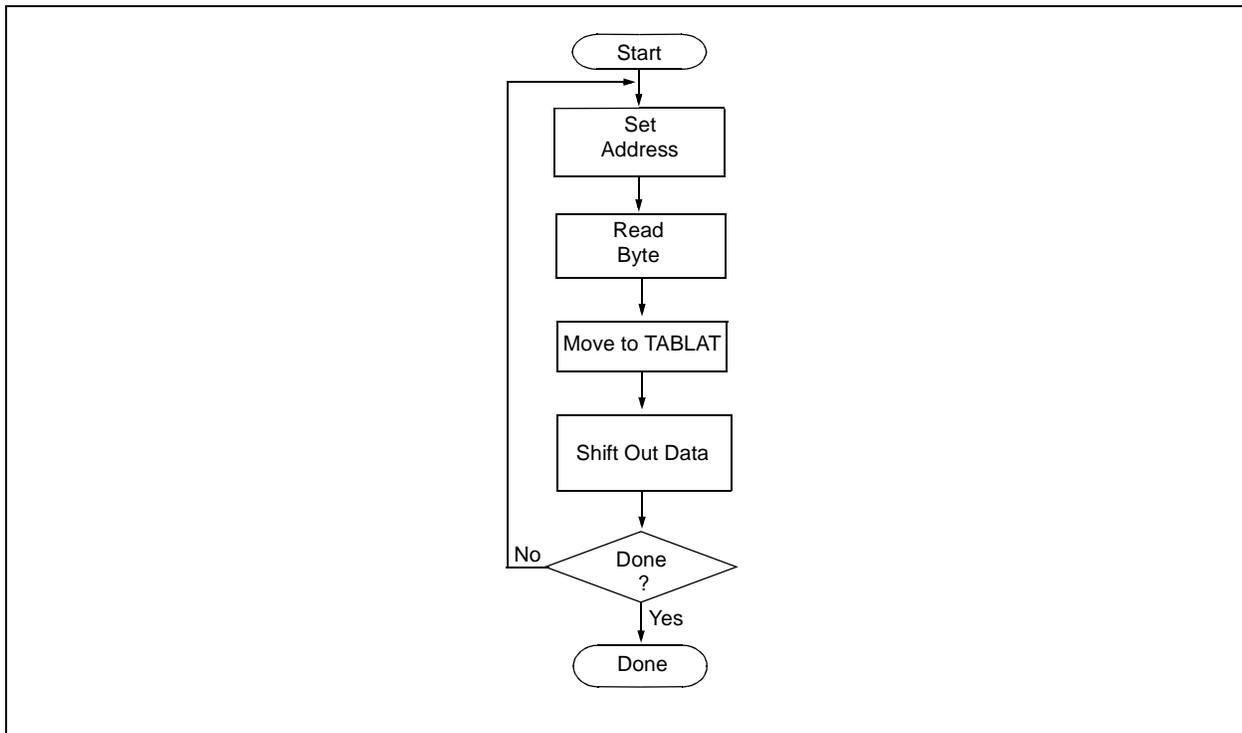### 3.5.1   READ EEDATA MEMORY

Data memory is read one byte at a time. Unlike the memory pointer used with code memory, the address pointer is not automatically incremented with each write so it must be explicitly set every cycle. The command sequence to read a single byte is shown in Table 3-6. This sequence must be repeated for each byte.

TABLE 3-6:   READ DATA MEMORY

4-Bit Command	Data Payload	Core Instruction
0	0E <DataADDR>	MOVLW <DataADDR>
0	6E A9	MOVWF EEADR
0	9E A6	BCF EECON1, EEPGD
0	80 A6	BSF EECON1, RD
0	00 00	NOP
0	CF A8 FF F5	MOVFF EEDATA, TABLAT
2	00 <byte>	Shift Out Data[1]

Note 1: The first Byte is undefined. The second Byte <byte> is the data.

**FIGURE 3-6:   PROGRAM DATA FLOW CHART - PIC18F0XX**

3.5.2 EEDATA MEMORY PROGRAMMING

Data memory is written one byte at a time. Unlike the memory pointer used with code memory, the address pointer is not automatically incremented with each write, so it must be explicitly set every cycle. The command sequence to write a single byte is shown in Table 3-7. This sequence must be repeated for each byte. Data memory is stored in the HEX file starting at F00000h.

TABLE 3-7: PROGRAMMING DATA MEMORY

4-Bit Command	Data Payload	Core Instruction
0	0E <DataADDR>	MOVLW <DataADDR>
0	6E A9	MOVWF EEADR
0	0E <DataByte>	MOVLW <DataByte>
0	6E A8	MOVWF EEDATA
0	84 A6	BSF EECON1, WREN
0	9E A6	BCF EECON1, EEPGD
0	0E 55	MOVLW 0X55
0	6E A7	MOVWF EECON2
0	0E AA	MOVLW 0XAA
0	6E A7	MOVWF EECON2
0	82 A6	BSF EECON1, WR
0	00 00	NOP

FIGURE 3-7: PROGRAM DATA FLOW CHART - PIC18F0XX
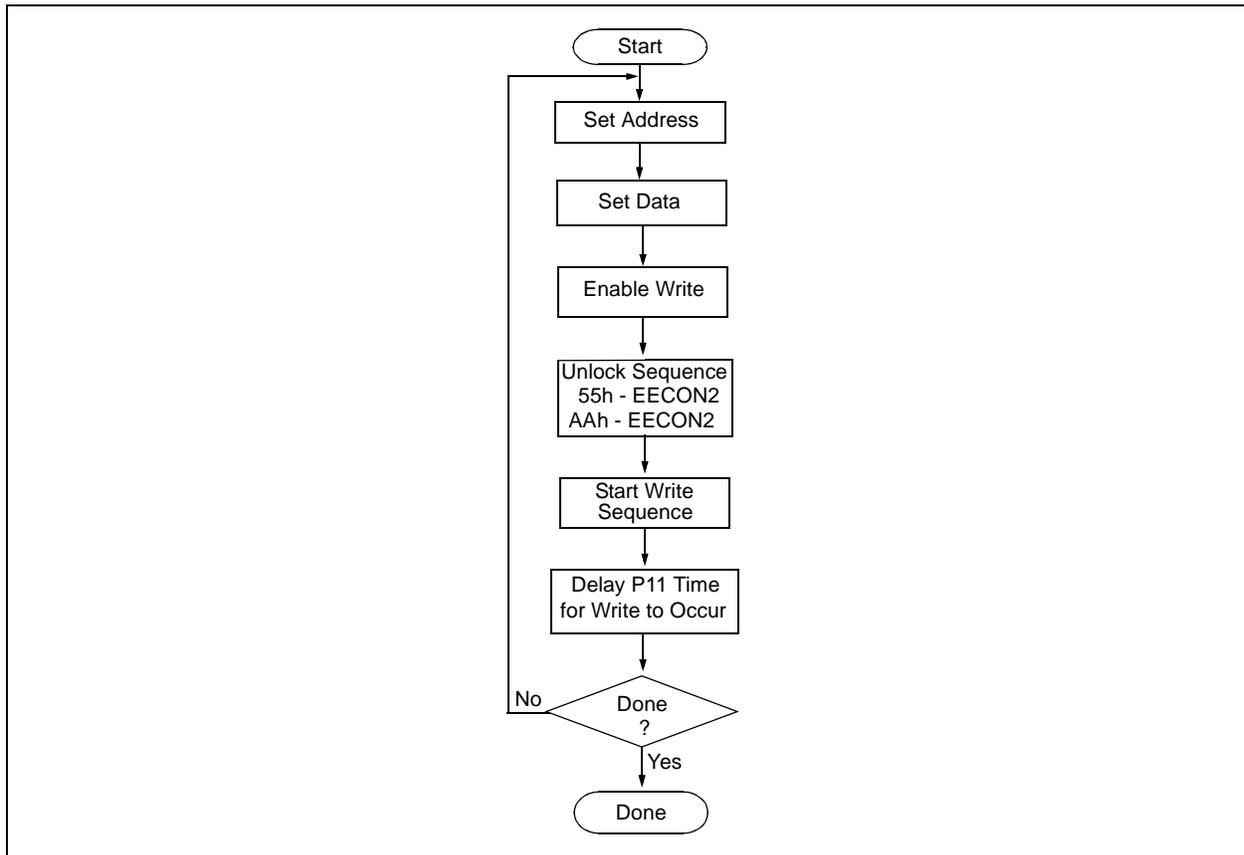
PIC18F0XX

3.5.3 PROGRAMMING THE USER ID LOCATIONS

The User ID Locations are programmed by setting the memory pointer to address 200000h, then programming two, 2-word pairs (see Table 3-8).

See Figure 3-13 and Figure 3-14 for timing of write instructions.

TABLE 3-8: SET ADDRESS POINTER TO ID LOCATION

4-Bit Command	Data Payload	Core Instruction
0	0E 20	MOVLW 0X20
0	6E F8	MOVWF TBLPTRU
0	0E 00	MOVLW 0X0
0	6E F7	MOVWF TBLPRTH
0	0E 00	MOVLW 0X00
0	6E F6	MOVWF TBLPTRL
5	<Byte 1> <Byte 0>	ID Bytes 0,1
7	<Byte 3> <Byte 2>	ID Bytes 2,3
0	00 00	NOP
5	<Byte 5> <Byte 4>	ID Bytes 4,5
7	<Byte 7> <Byte 6>	ID Bytes 6,7
0	00 00	NOP

### 3.5.4 PROGRAMMING THE CONFIGURATION LOCATIONS

The Configuration Locations are programmed by setting the memory pointer to address 300000h, then programming two words. Setting the memory pointer is performed by the command sequence shown in Table 3-9.

See Figure 3-14 for timing of write instructions.

TABLE 3-9:   SET ADDRESS POINTER TO CONFIGURATION LOCATION

4-Bit Command	Data Payload	Core Instruction
0	0E 30	MOVLW 0X30
0	6E F8	MOVWF TBLPTRU
0	0E 00	MOVLW 0X0
0	6E F7	MOVWF TBLPRTH
0	0E 00	MOVLW 0X00
0	6E F6	MOVWF TBLPTRL
7	<CONFIG1H> <CONFIG1L>	—
0	00 00	NOP
7	<CONFIG2H> <CONFIG2L>	—
0	00 00	NOP

**FIGURE 3-8:   PROGRAM MEMORY FLOW CHART PIC18F0XX**
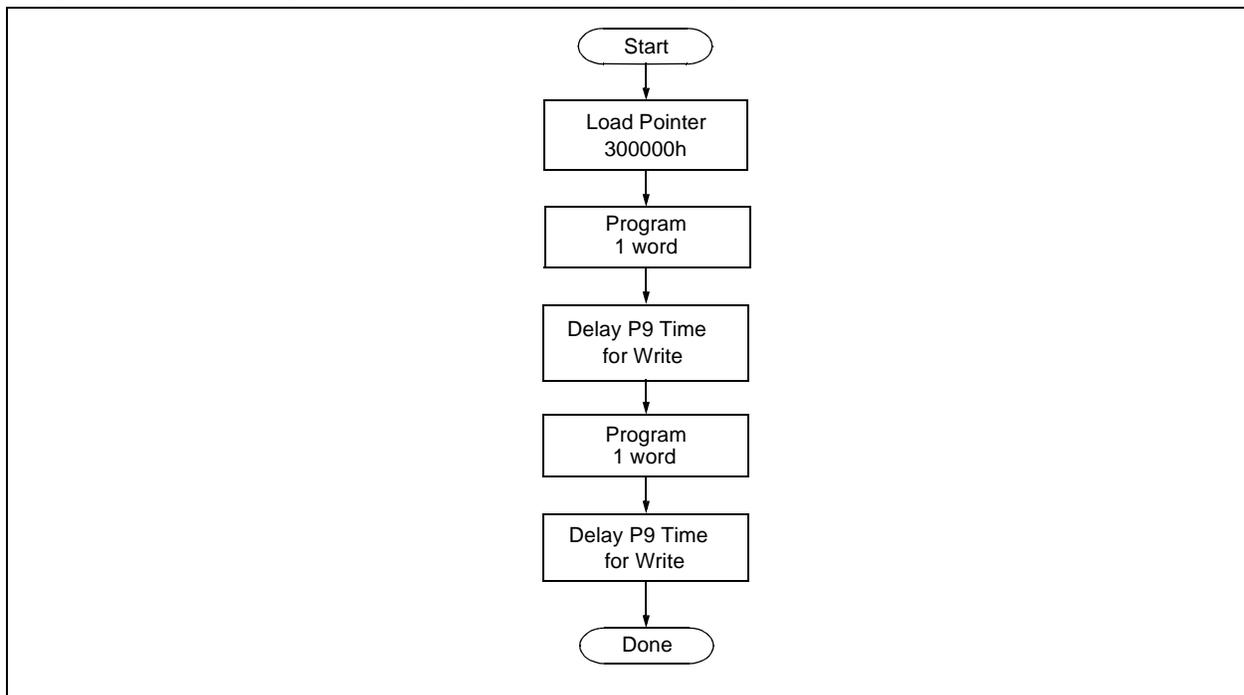
PIC18F0XX

3.5.5   VERIFY CODE MEMORY

The verify step involves reading back the memory space and comparing against the copy held in the programmer's buffer. First, however, the memory pointer must be set to the beginning of memory, as shown in Figure 3-9, and described in the programming code memory section.

Memory reads occur a single byte at a time. Two bytes must be read to compare against the word in the programmer's buffer. The memory pointer is advanced to the next address with each read, eliminating the need to change it manually.

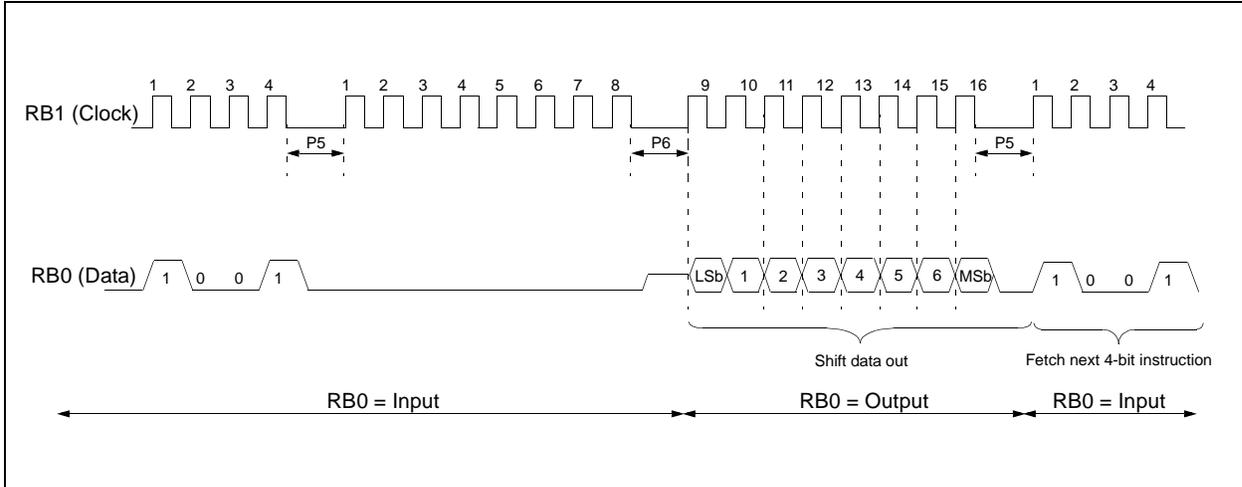**FIGURE 3-9:**   **VERIFY PROGRAM MEMORY FLOW CHART PIC18F0XX**
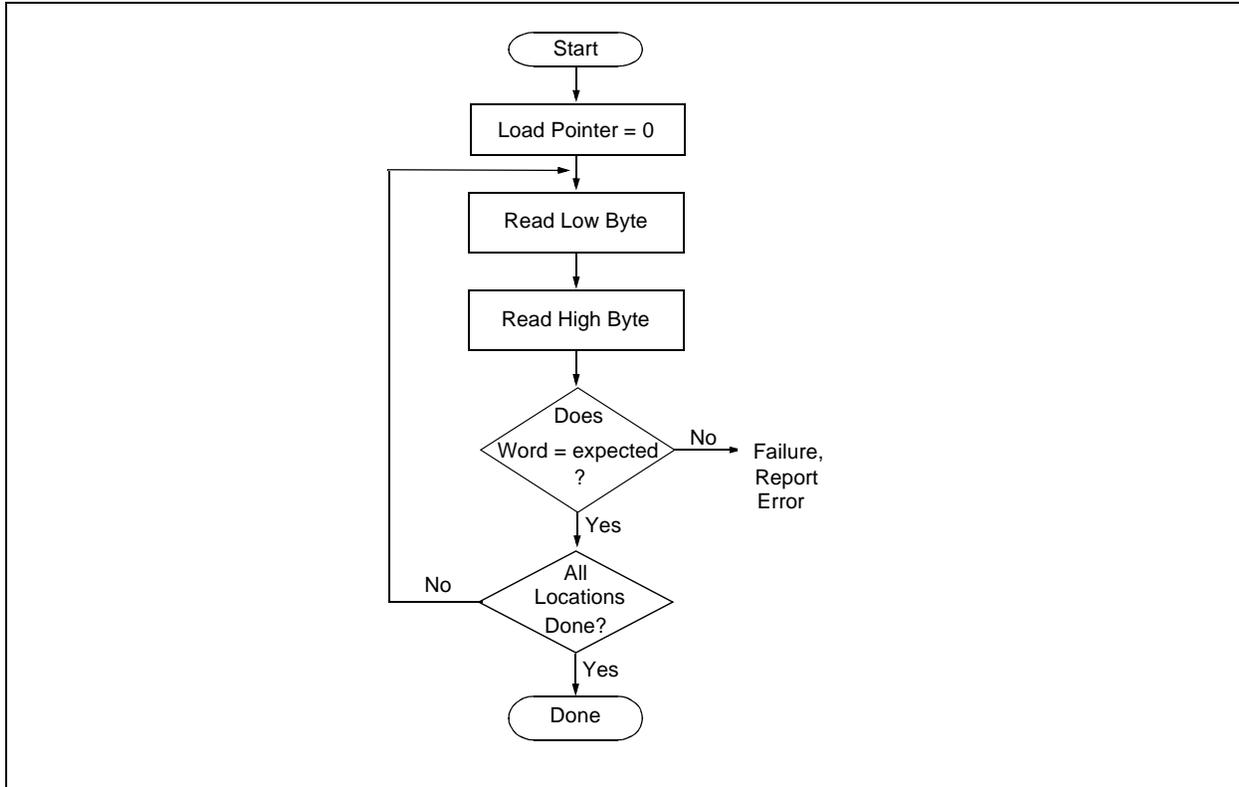
The command sequence to read a byte is as follows:

4-Bit Command	Data Payload	Core Instruction
9	00 <byte>	—

The memory read occurs while the first 8-bits are being clocked in, then clocked out in the second 8-bits of the payload phase, LSb first.

FIGURE 3-10: **TABLE READ POST-INCREMENT INSTRUCTION SEQUENCE (1001)**


3.5.6 READ INSTRUCTION

Figure 3-10 shows a Read operation. The 4-bit opcode is shifted in LSb first. The Read is executed during the next 8 clocks, then shifted out on RB0, during the last 8 clocks, LSB to MSB. The operation also increments the pointer by 1, pointing to the next byte in program memory, ready for the next read.

FIGURE 3-11: **SHIFT OUT DATA HOLDING REGISTER (0010)**
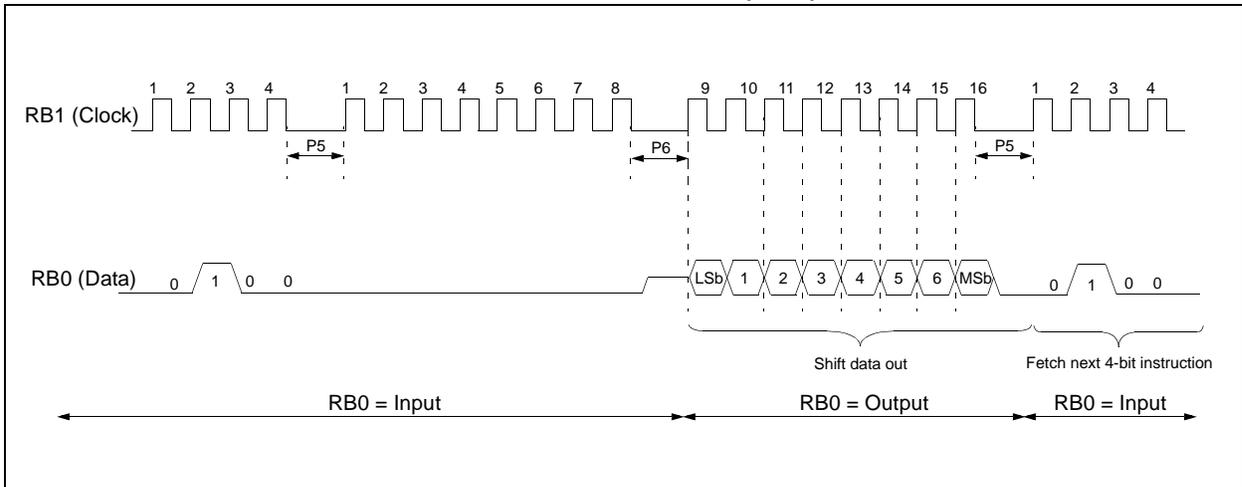
PIC18F0XX

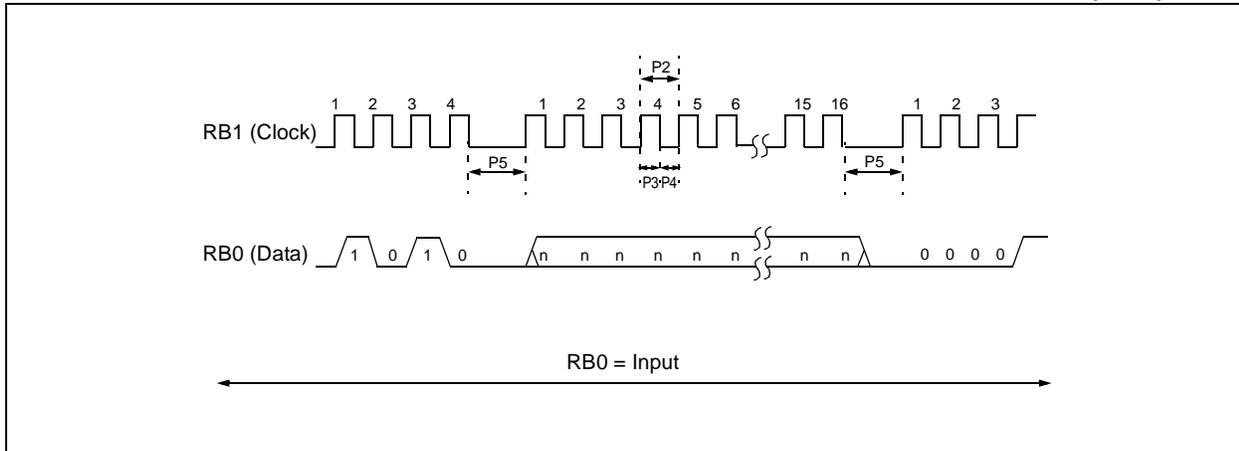## FIGURE 3-12: TABLE WRITE INSTRUCTION SEQUENCE (1100)
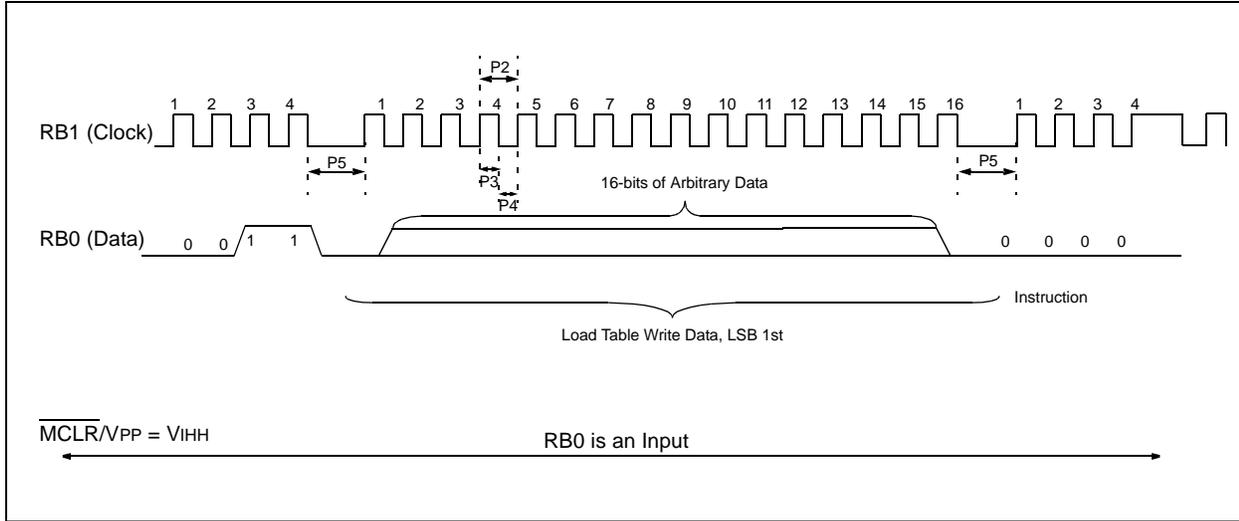
3.5.7 WRITE INSTRUCTIONS FOR PROGRAM MEMORY

The Write instruction is used repeatedly in ICSP™ mode. A 4-bit instruction will minimize the total number of clock cycles required to perform programming algorithms.

The timing diagram, Figure 3-12, applies to the table write, and Figure 3-13 applies to table write post-increment by 2.

FIGURE 3-13: TABLE WRITE POST-INCREMENT BY 2 INSTRUCTION SEQUENCE (0101)

FIGURE 3-14: **TABLE WRITE PROGRAM POST-INCREMENT BY 2 INSTRUCTION SEQUENCE (0111)**

3.5.8 WRITE/PROGRAM INSTRUCTIONS

The Write/Program (increment address pointer by 2 bytes and start programming) is similar to Write, but it also starts the programming cycle.

The Write/Program instruction loads the programming latches and starts a programming cycle, as shown in Figure 3-14. RB1 must be held high for the time specified by parameter P9.

When RB1 is asserted low, the device will cease programming the specified location.

After RB1 is asserted low, RB1 is held low for the time specified by parameter P10, to allow high voltage discharge of the program memory array.

PIC18F0XX

4.0 CONFIGURATION WORD

The PIC18F0XX has several configuration words. These bits can be set or cleared to select various device configurations. All other memory areas should be programmed and verified prior to setting configuration words.

4.1 Device ID Word

The device ID word for the PIC18F0XX is located at 3FFFFE.

Configuration bits and device IDs are not protected by any mechanism. This allows verification once protected.

4.2 ID Locations

A user may store identification information (ID) in eight ID locations mapped in [200000h:200007h].

TABLE 4-2: **PIC18F0XX CONFIGURATION BITS AND DEVICE IDS**

File Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default/ Unprogrammed Value
300000h	CONFIG1L	—	TR1	TW1	CP1	DP	TR0	TW0	CP0	`0111 1111`
300001h	CONFIG1H	—	—	OSCEN	MCLRE	—	FOSC2	FOSC1	FOSC0	`--11 -100`
300002h	CONFIG2L	—	—	—	—	—	—	BOREN	$\overline{\text{PWRTE}}$	`---- --11`
300003h	CONFIG2H	Reserved[1]	—	STVRE	WDTLE	WDTPS2	WDTPS1	WDTPS0	WDTE	`1-11 1111`
3FFFFEh	DEVID1	DEV2	DEV1	DEV0	REV4	REV3	REV2	REV1	REV0	`xxxx xxxx`
3FFFFFh	DEVID2	DEV10	DEV9	DEV8	DEV7	DEV6	DEV5	DEV4	DEV3	`xxxx xxxx`[2]

Legend: x = unknown, - = unimplemented, q = value depends on condition, r = reserved.
Grayed cells are unimplemented, read as 0.

Note 1: Reserved - This bit must be programmed as a 1.
     2: See Table 4-1.

TABLE 4-3: **PIC18F0XX BIT DESCRIPTION**

Bit Name	Configuration Words	Description
CP0	CONFIG1L	Code Protection bits (memory area 0000h - 03FFh) bytes `1` = Program memory code protection off `0` = All of program memory code protected
CP1	CONFIG1L	Code Protection bits (memory area > 0400h) bytes `1` = Program memory code protection off `0` = All of program memory code protected
DP	CONFIG1L	Data Protection bit `1` = External reads and writes are enabled `0` = External reads and writes are disabled

FOSC2:FOSC0	CONFIG1H	Oscillator Selection bits
		`111` = External RC oscillator/CLKOUT function on RB4/OSC2/CLKOUT pin
		`011` = External RC oscillator/RB4 function on RB4/OSC2/CLKOUT pin
		`101` = Internal RC oscillator/CLKOUT function on RB4/OSC2/CLKOUT pin,
		      RB5 function on RB5/OSC1/CLKIN pin
		`100` = Internal RC oscillator/RB4 function on RB4/OSC2/CLKOUT pin, RB5
		      function on RB5/OSC1/CLKIN pin
		`110` = EC external clock/CLKOUT function on RB4/OSC2/CLKOUT pin
		`010` = HS oscillator
		`001` = XT oscillator
		`000` = LP oscillator

BOREN	CONFIG2L	Brown-out Reset Enable bit `1` = Brown-out Reset enabled `0` = Brown-out Reset disabled
MCLRE	CONFIG1H	RB3/$\overline{\text{MCLR}}$ Function Select bit `1` = RB3/$\overline{\text{MCLR}}$ pin function is $\overline{\text{MCLR}}$ `0` = RB3/$\overline{\text{MCLR}}$ pin function is digital I/O, $\overline{\text{MCLR}}$ internally tied to V$_{DD}$
OSCEN	CONFIG1H	Oscillator Enable bit 1 = Switching to the internal oscillator is enabled 0 = Switching to the internal oscillator is disabled
$\overline{\text{PWRTE}}$	CONFIG2L	Power-up Timer Enable bit `1` = PWRT disabled `0` = PWRT enabled Enabling Brown-out Reset automatically enables the Power-up Timer (PWRT), regardless of the value of bit $\overline{\text{PWRTEN}}$. Ensure Power-up Timer is enabled when Brown-out Reset is enabled.
TR0	CONFIG1L	Table Read Protection bit (memory area 0000h - 03FFh) bytes `1` = Table reads are enabled `0` = Table reads are disabled from access outside of this block
TR1	CONFIG1L	Table Read Protection bit (memory area > 0400h) bytes `1` = Table reads are enabled `0` = Table reads are disabled from access outside of this block
TW0	CONFIG1L	Table Write Protection bit (memory area 0000h - 03FFh) bytes `1` = Table writes are enabled `0` = Table writes are disabled from access outside of this block
TW1	CONFIG1L	Table Write Protection bit (memory area > 0400h) bytes `1` = Table writes are enabled `0` = Table writes are disabled from access outside of this block

PIC18F0XX

TABLE 4-3: **PIC18F0XX BIT DESCRIPTION (CONTINUED)**

Bit Name	Configuration Words	Description
WDTPS2:WDTPS0	CONFIG2H	Watchdog Timer Postscale Select bits `111` = 1:128 `110` = 1:64 `101` = 1:32 `100` = 1:16 `011` = 1:8 `010` = 1:4 `001` = 1:2 `000` = 1:1
WDTE	CONFIG2H	Watchdog Timer Enable bit `1` = WDT enabled `0` = WDT disabled (control is placed on SWDTEN bit)
WDTLE	CONFIG2H	Watchdog Timer Long Delay Enable bit `1` = Use WDPS<2:0> bits to set delay `0` = Enable long postscaler divider, 16 X WDPS<2:0> bits
STVRE	CONFIG4L	Stack Overflow/Underflow Reset Enable bit `1` = Stack Overflow/Underflow will cause RESET `0` = Stack Overflow/Underflow will not cause RESET

PIC18F0XX

4.3   Embedding Configuration Word Information in the HEX File

To allow portability of code, a PIC18F0XX programmer is required to read the configuration word locations from the HEX file. If configuration word information was not present in the HEX file, then a simple warning message may be issued. Similarly, while saving a HEX file, all configuration word information must be included. An option to not include the configuration word information may be provided. When embedding configuration word information in the HEX file, it should be to address 300000h.

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

4.4   Checksum Computation

The checksum is calculated by summing the following:

• The contents of all program memory locations
• The configuration word, appropriately masked
• ID locations

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

Table 4-4 describes how to calculate the checksum for each device.

> **Note 1:** The checksum calculation differs depending on the code protect setting. Since the program memory locations read out differently, depending on the code protect setting, the table describes how to manipulate the actual program memory values to simulate the 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 word and ID locations can always be read.

TABLE 4-4:   CHECKSUM COMPUTATION

Device	Code Protect	Checksum	Blank Value	0xAA at 0 and Max Address
PIC18F010	Off	SUM[0000:07FF] + [CONFIG0 & 007F] + [CONFIG1 & 0037] + [CONFIG2 & 0003] + [CONFIG3 & 003F]	F8F8	F84E
	0000:03FF	SUM[0400:07FF] + [CONFIG0 & 007F] + [CONFIG1 & 0037] + [CONFIG2 & 0003] + [CONFIG3 & 003F] + SUM_ID	FD25	FCCB
	0400:07FF	SUM[0000:03FF] + [CONFIG0 & 007F] + [CONFIG1 & 0037] + [CONFIG2 & 0003] + [CONFIG3 & 003F] + SUM_ID	FD16	FCBC
	All	[CONFIG0 & 007F] + [CONFIG1 & 0037] + [CONFIG2 & 0003] + [CONFIG3 & 003F] + SUM_ID	0115	0110
PIC18F020	Off	SUM[0000:0FFF] + [CONFIG0 & 007F] + [CONFIG1 & 0037] + [CONFIG2 & 0003] + [CONFIG3 & 003F]	F0F8	F04E
	0000:03FF	SUM[0400:0FFF] + [CONFIG0 & 007F] + [CONFIG1 & 0037] + [CONFIG2 & 0003] + [CONFIG3 & 003F] + SUM_ID	F51D	F4C3
	0400:0FFF	SUM[0000:03FF] + [CONFIG0 & 007F] + [CONFIG1 & 0037] + [CONFIG2 & 0003] + [CONFIG3 & 003F] + SUM_ID	FD0E	FCB4
	All	[CONFIG0 & 007F] + [CONFIG1 & 0037] + [CONFIG2 & 0003] + [CONFIG3 & 003F] + SUM_ID	010D	0108

Legend: Item   Description
         CFGW =   Configuration Word
         SUM[a:b] =   Sum of locations a to b inclusive
         SUM_ID =   Byte-wise sum of lower four bits of all customer ID locations
         + =   Addition
         & =   Bitwise AND

PIC18F0XX

4.5 Embedding EEDATA Information in the HEX File

To allow portability of code, a PIC18F0XX programmer is required to read the EEDATA from the HEX file. If EEDATA information is not present, a simple warning message may be issued. Similarly, when saving a HEX file, all EEDATA must be included. An option to not include the EEDATA may be provided. When embedding EEDATA information in the HEX file, it should start at address F00000h.

Microchip Technology Inc. believes that this feature is important for the benefit of the end customer.

5.0 AC/DC CHARACTERISTICS
TIMING REQUIREMENTS FOR PROGRAM/VERIFY TEST MODE

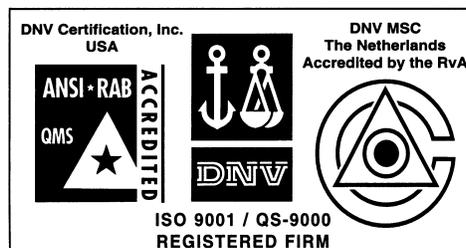|		**Standard Operating Conditions** Operating Temperature: -25°C is recommended Operating Voltage: 4.75V ≤ VDD ≤ 5.25V, unless otherwise stated				
Parameter No.	**Symbol**	**Characteristic**	**Min**	**Max**	**Units**	**Conditions**
	VIHH	Programming Voltage on VPP/MCLR pin	9.00	13.25	V	
	LIHH	Programming current on MCLR pin		300	μA	
	LIDDP	—	—	10	mA	
P1	Tser	Serial setup time	20	—	ns	
P2	Tsclk	Serial clock period	100	—	ns	
P3	Tset1	Input Data Setup Time to serial clock ↓	15	—	ns	
P4	Thld1	Input Data Hold Time from serial clock ↓	15	—	ns	
P5	Tdly1	Delay between last clock ↓ to first clock ↑ of next command	20	—	ns	
P6	Tdly2	Delay between last clock ↓ of command byte to first clock ↑ of read of data word	20	—	ns	
P8	Tdly4	Data input not driven to next clock input	1	—	ns	
P9	Tdly5	RB1 high time (minimum programming time)	2	—	ms	
P10	Tdly6	RB1 low time after programming (high voltage discharge time)	5	—	μs	
P11	Tdly7	Delay to allow self-timed data write to occur	5	—	ms	

NOTES:

Trademarks

The Microchip name, logo, PIC, PICmicro, PICMASTER, PIC-START, PRO MATE, KEELOQ, SEEVAL, MPLAB and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

Total Endurance, ICSP, In-Circuit Serial Programming, Filter-Lab, MXDEV, microID, FlexROM, fuzzyLAB, MPASM, MPLINK, MPLIB, PICDEM, ICEPIC, Migratable Memory, FanSense, ECONOMONITOR, Select Mode and microPort are trademarks of Microchip Technology Incorporated in the U.S.A.

Serialized Quick Term Programming (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.

© 2001, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

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: 480-792-7627
Web Address: http://www.microchip.com

Rocky Mountain
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7966 Fax: 480-792-7456

Atlanta
500 Sugar Mill Road, Suite 200B
Atlanta, GA 30350
Tel: 770-640-0034 Fax: 770-640-0307

Austin
Analog Product Sales
8303 MoPac Expressway North
Suite A-201
Austin, TX 78759
Tel: 512-345-2030 Fax: 512-345-6085

Boston
2 Lan Drive, Suite 120
Westford, MA 01886
Tel: 978-692-3848 Fax: 978-692-3821

Boston
Analog Product Sales
Unit A-8-1 Millbrook Tarry Condominium
97 Lowell Road
Concord, MA 01742
Tel: 978-371-6400 Fax: 978-371-0050

Chicago
333 Pierce Road, Suite 180
Itasca, IL 60143
Tel: 630-285-0071 Fax: 630-285-0075

Dallas
4570 Westgrove Drive, Suite 160
Addison, TX 75001
Tel: 972-818-7423 Fax: 972-818-2924

Dayton
Two Prestige Place, Suite 130
Miamisburg, OH 45342
Tel: 937-291-1654 Fax: 937-291-9175

Detroit
Tri-Atria Office Building
32255 Northwestern Highway, Suite 190
Farmington Hills, MI 48334
Tel: 248-538-2250 Fax: 248-538-2260

Los Angeles
18201 Von Karman, Suite 1090
Irvine, CA 92612
Tel: 949-263-1888 Fax: 949-263-1338

Mountain View
Analog Product Sales
1300 Terra Bella Avenue
Mountain View, CA 94043-1836
Tel: 650-968-9241 Fax: 650-967-1590

New York
150 Motor Parkway, Suite 202
Hauppauge, NY 11788
Tel: 631-273-5305 Fax: 631-273-5335

San Jose
Microchip Technology Inc.
2107 North First Street, Suite 590
San Jose, CA 95131
Tel: 408-436-7950 Fax: 408-436-7955

Toronto
6285 Northam Drive, Suite 108
Mississauga, Ontario L4V 1X5, Canada
Tel: 905-673-0699 Fax: 905-673-6509

ASIA/PACIFIC

Australia
Microchip Technology Australia Pty Ltd
Suite 22, 41 Rawson Street
Epping 2121, NSW
Australia
Tel: 61-2-9868-6733 Fax: 61-2-9868-6755

China - Beijing
Microchip Technology Beijing Office
Unit 915
New China Hong Kong Manhattan Bldg.
No. 6 Chaoyangmen Beidajie
Beijing, 100027, No. China
Tel: 86-10-85282100 Fax: 86-10-85282104

China - Shanghai
Microchip Technology Shanghai Office
Room 701, Bldg. B
Far East International Plaza
No. 317 Xian Xia Road
Shanghai, 200051
Tel: 86-21-6275-5700 Fax: 86-21-6275-5060

Hong Kong
Microchip Asia Pacific
RM 2101, Tower 2, Metroplaza
223 Hing Fong Road
Kwai Fong, N.T., Hong Kong
Tel: 852-2401-1200 Fax: 852-2401-3431

India
Microchip Technology Inc.
India Liaison Office
Divyasree Chambers
1 Floor, Wing A (A3/A4)
No. 11, O'Shaugnessey Road
Bangalore, 560 025, India
Tel: 91-80-2290061 Fax: 91-80-2290062

Japan
Microchip Technology Intl. Inc.
Benex S-1 6F
3-18-20, Shinyokohama
Kohoku-Ku, Yokohama-shi
Kanagawa, 222-0033, Japan
Tel: 81-45-471- 6166 Fax: 81-45-471-6122

ASIA/PACIFIC (continued)

Korea
Microchip Technology Korea
168-1, Youngbo Bldg. 3 Floor
Samsung-Dong, Kangnam-Ku
Seoul, Korea
Tel: 82-2-554-7200 Fax: 82-2-558-5934

Singapore
Microchip Technology Singapore Pte Ltd.
200 Middle Road
#07-02 Prime Centre
Singapore, 188980
Tel: 65-334-8870 Fax: 65-334-8850

Taiwan
Microchip Technology Taiwan
11F-3, No. 207
Tung Hua North Road
Taipei, 105, Taiwan
Tel: 886-2-2717-7175 Fax: 886-2-2545-0139

EUROPE

Denmark
Microchip Technology Denmark ApS
Regus Business Centre
Lautrup hoj 1-3
Ballerup DK-2750 Denmark
Tel: 45 4420 9895 Fax: 45 4420 9910

France
Arizona Microchip Technology SARL
Parc d'Activite du Moulin de Massy
43 Rue du Saule Trapu
Batiment A - ler Etage
91300 Massy, France
Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79

Germany
Arizona Microchip Technology GmbH
Gustav-Heinemann Ring 125
D-81739 Munich, Germany
Tel: 49-89-627-144 0 Fax: 49-89-627-144-44

Germany
Analog Product Sales
Lochhamer Strasse 13
D-82152 Martinsried, Germany
Tel: 49-89-895650-0 Fax: 49-89-895650-22

Italy
Arizona Microchip Technology SRL
Centro Direzionale Colleoni
Palazzo Taurus 1 V. Le Colleoni 1
20041 Agrate Brianza
Milan, Italy
Tel: 39-039-65791-1 Fax: 39-039-6899883

United Kingdom
Arizona Microchip Technology Ltd.
505 Eskdale Road
Winnersh Triangle
Wokingham
Berkshire, England RG41 5TU
Tel: 44 118 921 5869 Fax: 44-118 921-5820

01/30/01