PIC18F6XJXX/8XJXX

Flash Microcontroller Programming Specification

1.0 DEVICE OVERVIEW

This document includes the programming specifications for the following devices:

- PIC18F67J50
- PIC18F67J11
- PIC18F67J10
- PIC18F66J55
- PIC18F66J50
- PIC18F66J16
- PIC18F66J15
- PIC18F66J11
- PIC18F66J10
- PIC18F65J90
- PIC18F65J50
- PIC18F65J15
- PIC18F65J11
- PIC18F65J10
- PIC18F64J90
- PIC18F64J11
- PIC18F63J90
- PIC18F63J11
- PIC18F87J50
- PIC18F87J11
- PIC18F87J10
- PIC18F86J55
- PIC18F86J50
- PIC18F86J16
- PIC18F86J15
- PIC18F86J11
- PIC18F86J10
- PIC18F85J90
- PIC18F85J50
- PIC18F85J15
- PIC18F85J11
- PIC18F85J10
- PIC18F84J90
- PIC18F84J11
- PIC18F83J90
- PIC18F83J11

2.0 PROGRAMMING OVERVIEW OF THE PIC18F6XJXX/8XJXX

The PIC18F6XJXX/8XJXX devices are programmed using In-Circuit Serial Programming™ (ICSP™). This programming specification applies to PIC18F6XJXX/8XJXX devices in all package types.

2.1 Pin Diagrams

The pin diagrams for the PIC18F6XJXX/8XJXX are shown in Figure 2-1 and Figure 2-2. The pins that are required for programming are listed in Table 2-1 and shown in darker lettering in the figures.

TABLE 2-1: PIN DESCRIPTIONS (DURING PROGRAMMING): PIC18F6XJXX/8XJXX

Pin Name	During Programming		
	Pin Name	Pin Type	Pin Description
$\overline{MCLR}$	$\overline{MCLR}$	P	Programming Enable
VDD and AVDD(1)	VDD	P	Power Supply
VSS and AVSS(1)	VSS	P	Ground
ENVREG	ENVREG	P	Internal Voltage Regulator Enable
VDDCORE/VCAP	VDDCORE	P	Regulated Power Supply for Microcontroller Core
	VCAP	I	Filter Capacitor for On-Chip Voltage Regulator
RB6	PGC	I	Serial Clock
RB7	PGD	I/O	Serial Data
VUSB(2)	VUSB	P	Internal USB 3.3V Voltage Regulator

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

Note 1: All power supply and ground pins must be connected, including analog supplies (AVDD) and ground (AVSS).

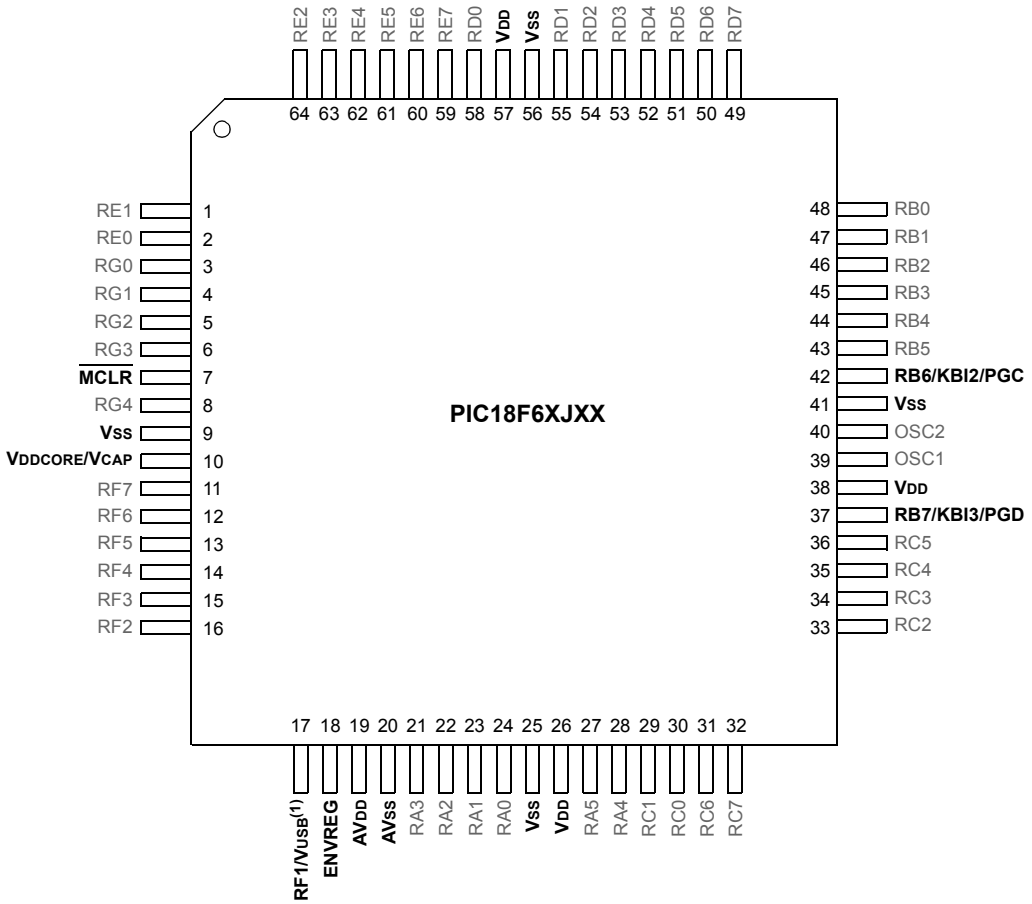
2: Valid only for PIC18F6XJ5X/8XJ5X families. This pin should be connected to VDD during programming.

FIGURE 2-1: PIC18F6XJXX PIN DIAGRAMS

64-Pin TQFP

The following devices are included in 64-pin TQFP parts:

- PIC18F67J50
- PIC18F67J11
- PIC18F67J10
- PIC18F66J55
- PIC18F66J50
- PIC18F66J16
- PIC18F66F15
- PIC18F66J11
- PIC18F66J10
- PIC18F65J90
- PIC18F65J50
- PIC18F65J15
- PIC18F65J11
- PIC18F65J10
- PIC18F64J90
- PIC18F64J11
- PIC18F63J90
- PIC18F63J11



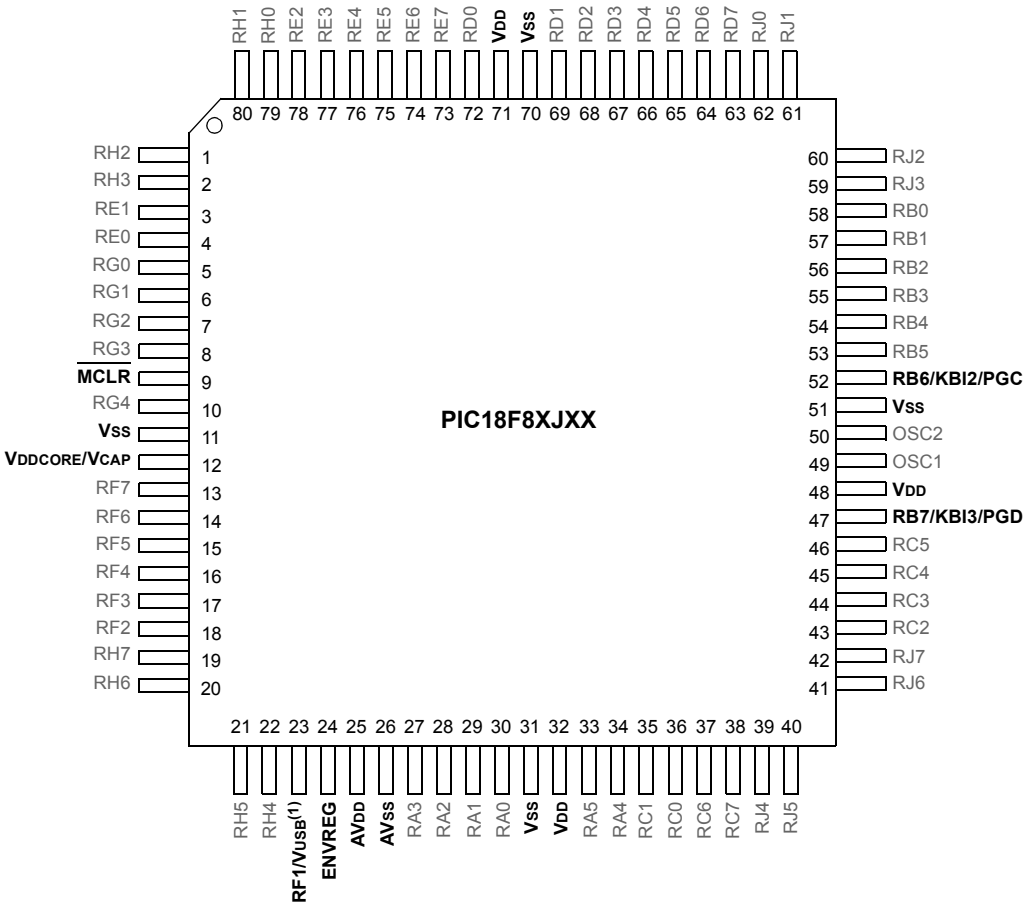
Note 1: Valid only for PIC18F6XJ5X/8XJ5X families.

FIGURE 2-2: PIC18F8XJXX PIN DIAGRAMS

80-Pin TQFP

The following devices are included in 80-pin TQFP parts:

- PIC18F87J50
- PIC18F87J11
- PIC18F87J10
- PIC18F86J55
- PIC18F86J50
- PIC18F86J16
- PIC18F86F15
- PIC18F86J11
- PIC18F86J10
- PIC18F85J90
- PIC18F85J50
- PIC18F85J15
- PIC18F85J11
- PIC18F85J10
- PIC18F84J90
- PIC18F84J11
- PIC18F83J90
- PIC18F83J11



Note 1: Valid only for PIC18F6XJ5X/8XJ5X families.

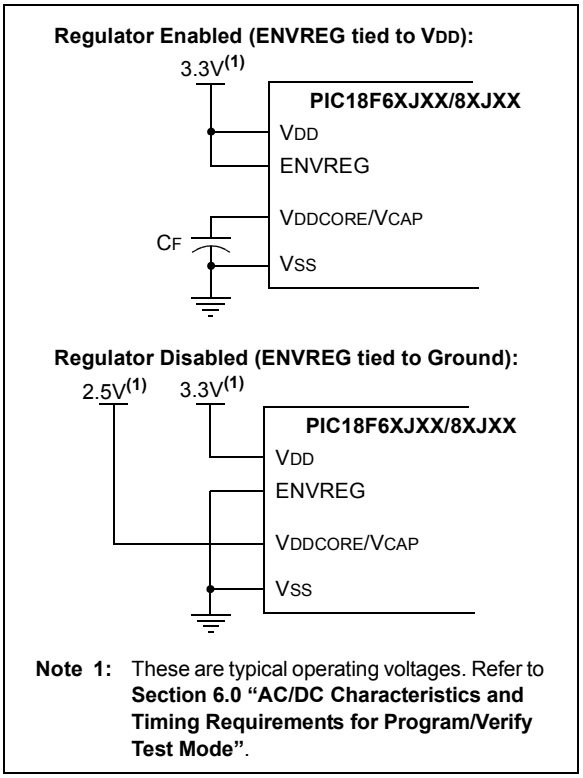
2.1.1 ON-CHIP VOLTAGE REGULATOR

All of the PIC18F6XJXX/8XJXX devices have dual power requirements. The microcontroller core can be powered from an external source that is separate from VDD, or it can be powered from an on-chip regulator which derives power from VDD. Both sources use the common VDDCORE/VCAP pin.

The regulator is enabled by connecting VDD to the ENVREG pin. In this case, a low ESR capacitor must be connected to the VDDCORE/VCAP pin for proper device operation. If the regulator is disabled by connecting VSS to the ENVREG pin, power to the core must be supplied on VDDCORE/VCAP. Whether or not the regulator is used, it is always good design practice to have sufficient capacitance on all supply pins. Examples are shown in Figure 2-3.

The specifications for core voltage and capacitance are listed in **Section 6.0 "AC/DC Characteristics and Timing Requirements for Program/Verify Test Mode"**.

FIGURE 2-3: CONNECTIONS FOR THE ON-CHIP REGULATOR



Note 1: These are typical operating voltages. Refer to **Section 6.0 "AC/DC Characteristics and Timing Requirements for Program/Verify Test Mode"**.

2.2 Memory Maps

The PIC18F6XJXX/8XJXX devices offer a total of eight program memory sizes, ranging from 8 Kbytes to 128 Kbytes. The memory sizes for different members of the family are shown in Table 2-2. The overall memory maps for all devices are shown in Figure 2-4, Figure 2-5, Figure 2-6 and Figure 2-7.

For purposes of code protection, the program memory for every device is treated as a single block. Enabling code protection thus protects the entire code memory and not individual segments.

The Configuration Words for these devices are located at addresses 300000h through 300005h. These are implemented as three pairs of volatile memory registers. 300006h-300007h are reserved for a fourth pair of Configuration registers that are not implemented in PIC18F6XJXX/8XJXX devices. Each register is automatically loaded from a copy stored at the end of program memory. For this reason, the top four words of the code space (also called the Flash Configuration Words) should be written with configuration data and not executable code. The addresses of the Flash Configuration Words are also listed in Table 2-2. Refer to section **Section 5.0 "Configuration Word"** for more information.

Locations 3FFFFEh and 3FFFFFh are reserved for the Device ID bits. These bits may be used by the programmer to identify what device type is being programmed and are described in **Section 5.1 "Device ID Word"**. These Device ID bits read out normally, even after code protection.

2.2.1 MEMORY ADDRESS POINTER

Memory in the device address space (000000h to 3FFFFFh) is addressed via the Table Pointer register, which in turn is comprised of three registers:

- TBLPTRU at RAM address 0FF8h
- TBLPTRH at RAM address 0FF7h
- TBLPTRL at RAM address 0FF6h

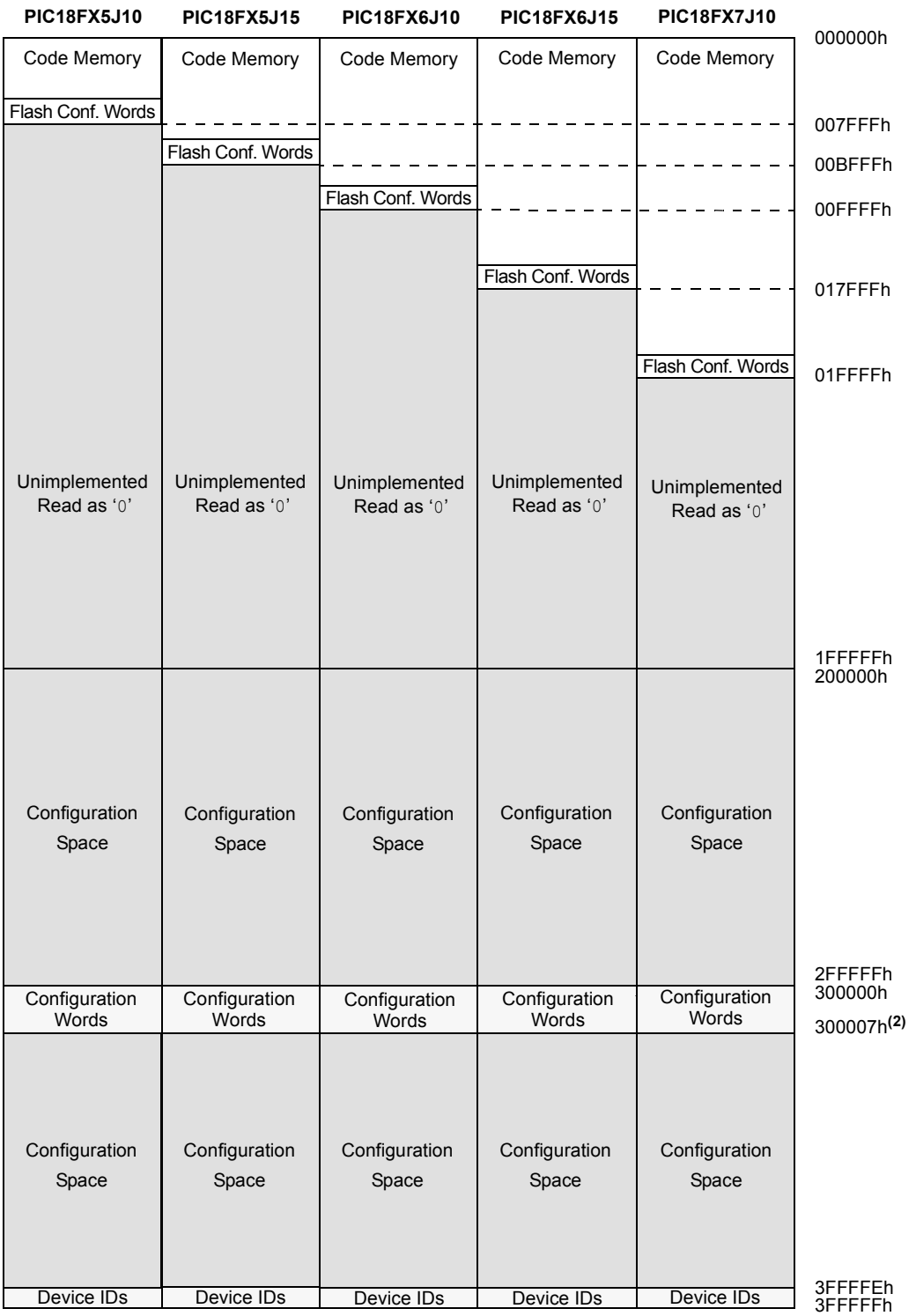
TBLPTRU	TBLPTRH	TBLPTRL
Addr[21:16]	Addr[15:8]	Addr[7:0]

The 4-bit command, '0000' (core instruction), is used to load the Table Pointer prior to using many read or write operations.

TABLE 2-2: PROGRAM MEMORY SIZES FOR PIC18F6XJXX/8XJXX DEVICES

Device	Program Memory (Kbytes)	Location of Flash Configuration Words
PIC18F63J11	8	1FF8h:1FFFh
PIC18F63J90		
PIC18F83J11		
PIC18F83J90		
PIC18F64J11	16	3FF8h:3FFFh
PIC18F64J90		
PIC18F84J11		
PIC18F84J90		
PIC18F65J10	32	7FF8h:7FFFh
PIC18F65J11		
PIC18F65J50		
PIC18F65J90		
PIC18F85J10		
PIC18F85J11		
PIC18F85J50		
PIC18F85J90		
PIC18F65J15	48	BFF8h:BFFFh
PIC18F85J15		
PIC18F66J10	64	FFF8h:FFFFh
PIC18F66J11		
PIC18F66J50		
PIC18F86J10		
PIC18F86J11		
PIC18F86J50		
PIC18F66J15	96	17FF8h:17FFFh
PIC18F66J16		
PIC18F66J55		
PIC18F86J15		
PIC18F86J16		
PIC18F86J55		
PIC18F67J10	128	1FFF8h:1fFFFh
PIC18F67J11		
PIC18F67J50		
PIC18F87J11		
PIC18F87J10		
PIC18F87J50		

FIGURE 2-4: MEMORY MAPS FOR PIC18FXXJ10/XXJ15 DEVICES(1)



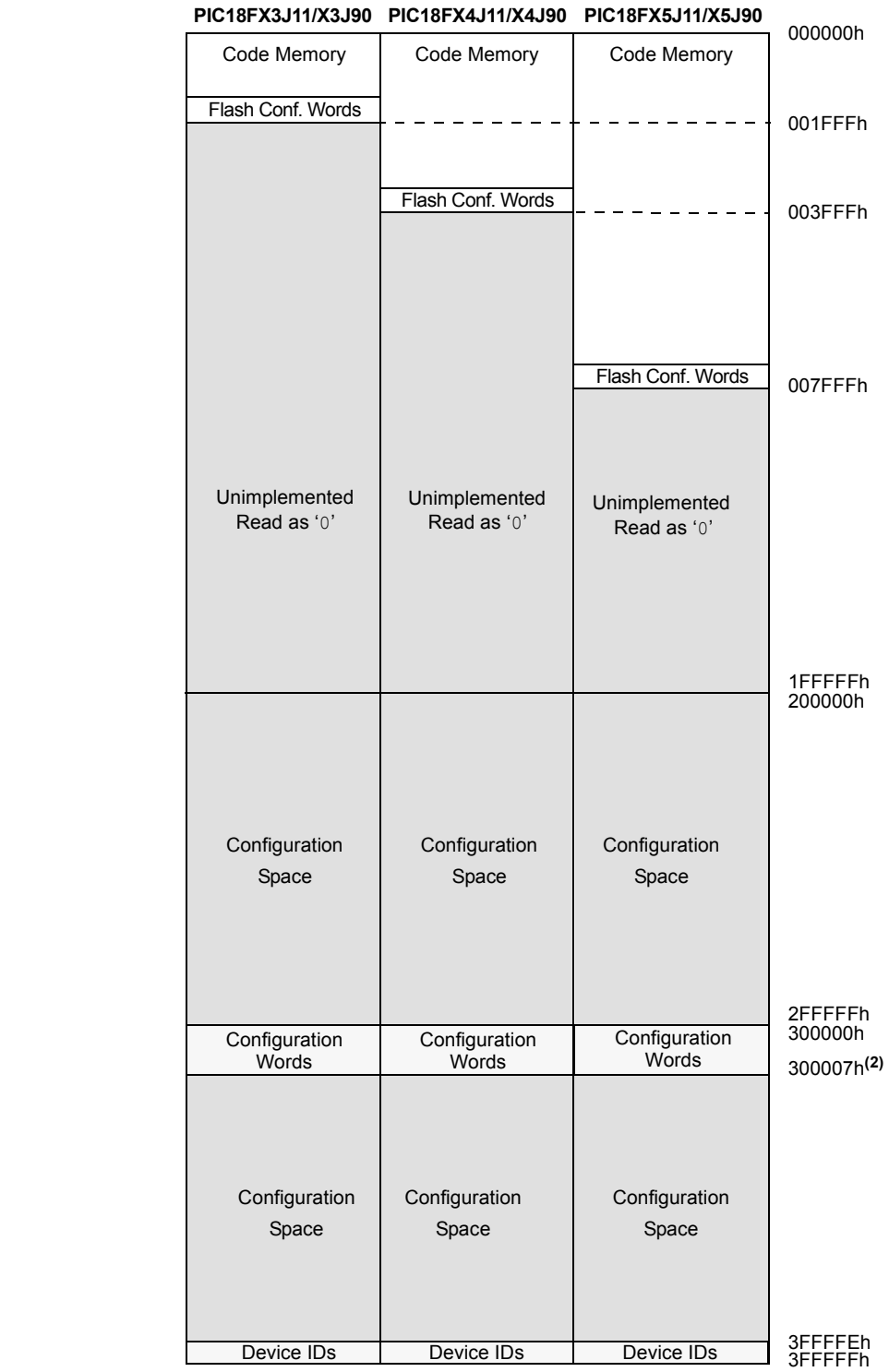
Memory spaces are unimplemented or unavailable in normal execution mode and read as '0'.

Memory spaces are read-only (Device IDs) or cannot be directly programmed by ICSP™ (Configuration Words).

Note 1: Sizes of memory areas are not to scale. Sizes of accessible memory areas are enhanced to show detail.

2: Configuration Words at 300006h and 300007h are not implemented on PIC18FXXJ10/XXJ15 devices.

FIGURE 2-5: MEMORY MAPS FOR PIC18FXXJ11/XXJ90 DEVICES(1)



Memory spaces are unimplemented or unavailable in normal execution mode and read as '0'.

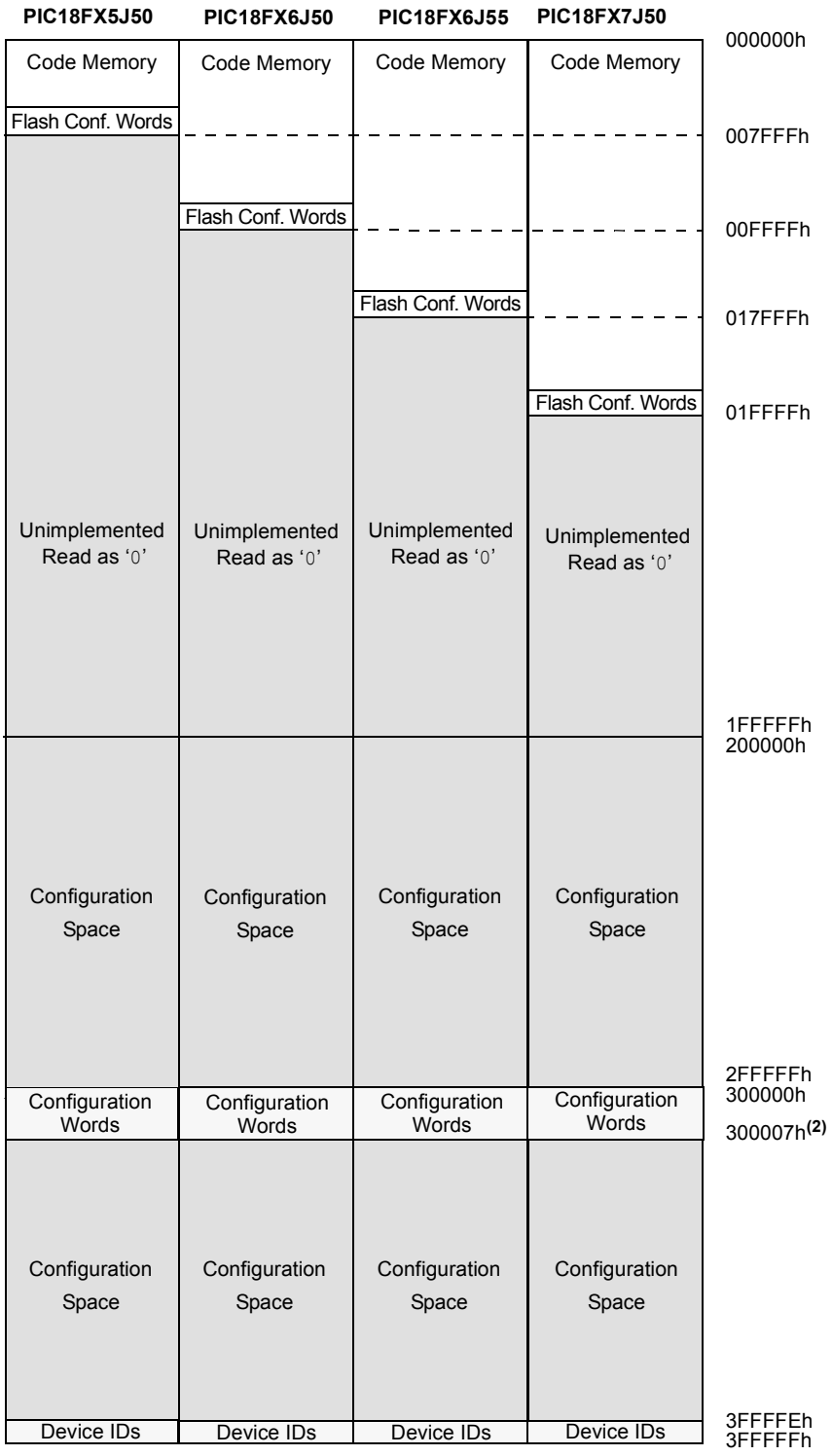
Memory spaces are read-only (Device IDs) or cannot be directly programmed by ICSP™ (Configuration Words).

Note 1: Sizes of memory areas are not to scale. Sizes of accessible memory areas are enhanced to show detail.

2: Configuration Words at 300006h and 300007h are not implemented on PIC18FXXJ11/XXJ90 devices.

3: PIC18F66J11/67J11/86J11/87J11 memory map is not included in this PIC18FXXJ11/XXJ90 memory map (see Figure 2-6).

FIGURE 2-6: MEMORY MAPS FOR PIC18F6XJ5X/8XJ5X DEVICES(1)



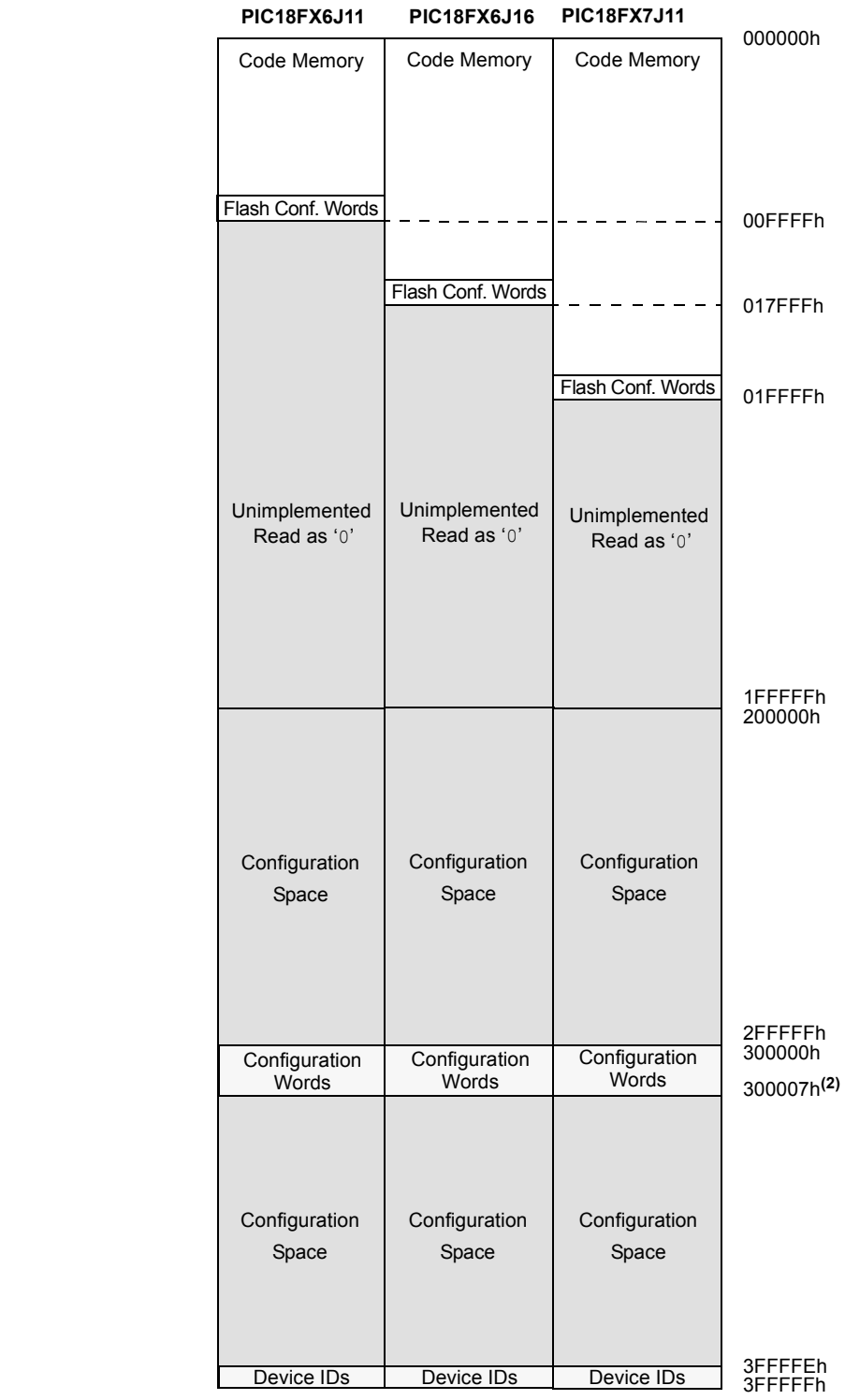
Memory spaces are unimplemented or unavailable in normal execution mode and read as '0'.

Memory spaces are read-only (Device IDs) or cannot be directly programmed by ICSP™ (Configuration Words).

Note 1: Sizes of memory areas are not to scale. Sizes of accessible memory areas are enhanced to show detail.

2: Configuration Words at 300006h and 300007h are not implemented on PIC18F6XJ50/8XJ5X devices.

FIGURE 2-7: MEMORY MAPS FOR PIC18F6XJ11/F6XJ16/F8XJ11/F8XJ16 DEVICES(1)



Memory spaces are unimplemented or unavailable in normal execution mode and read as '0'.

Memory spaces are read-only (Device IDs) or cannot be directly programmed by ICSP™ (Configuration Words).

Note 1: Sizes of memory areas are not to scale. Sizes of accessible memory areas are enhanced to show detail.

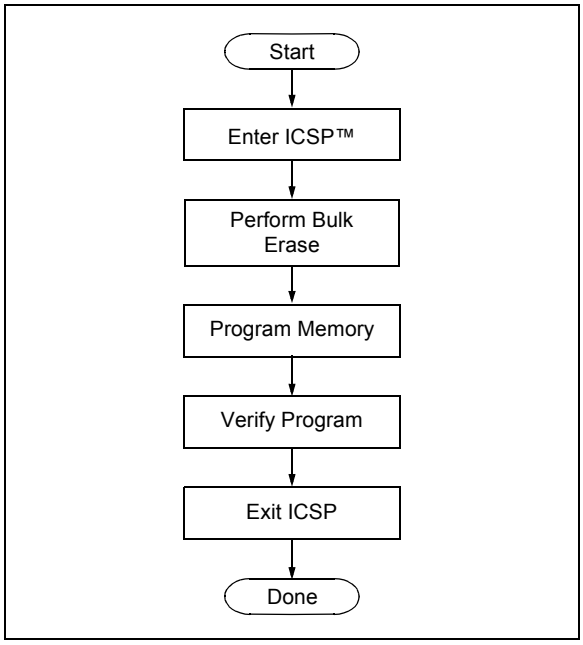
2: Configuration Words at 300006h and 300007h are not implemented on PIC18F6XJ50/8XJ5X devices.

2.3 Overview of the Programming Process

Figure 2-8 shows the high-level overview of the programming process. First, a Bulk Erase is performed. Next, the code memory is programmed. Since the only nonvolatile Configuration Words are within the code memory space, they too are programmed as if they were code. Code memory (including the Configuration Words) is then verified to ensure that programming was successful.

Note:	In order to maintain the endurance of the cells, each Flash byte should not be programmed more than twice between erase operations. A Bulk Erase of the device is required before attempting to modify the contents a third time.

FIGURE 2-8: HIGH-LEVEL PROGRAMMING FLOW



2.4 Entering and Exiting ICSP Program/Verify Mode

Entry into ICSP modes for PIC18F6XJXX/8XJXX devices is somewhat different than previous PIC18 devices. As shown in Figure 2-9, entering ICSP Program/Verify mode requires three steps:

1. Voltage is briefly applied to the $\overline{MCLR}$ pin.
2. A 32-bit key sequence is presented on PGD.
3. Voltage is reapplied to $\overline{MCLR}$ within a specific period of time and held.

The programming voltage applied to $\overline{MCLR}$ is V_{IH}, or essentially, V_{DD}. There is no minimum time requirement for holding at V_{IH}. After V_{IH} is removed, an interval of at least P19 must elapse before presenting the key sequence on PGD.

The key sequence is a specific 32-bit pattern, '0100 1101 0100 0011 0100 1000 0101 0000' (more easily remembered as 4D434850h in hexadecimal). 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, V_{IH} must be applied to $\overline{MCLR}$ and held at that level for as long as Program/Verify mode is to be maintained. An interval of at least time P20 and P12 must elapse before presenting data on PGD. Signals appearing on PGD before P12 has elapsed will not be interpreted as valid.

On successful entry, the program memory can be accessed and programmed in serial fashion. While in the Program/Verify mode, all unused I/Os are placed in the high-impedance state.

Exiting Program/Verify mode is done by removing V_{IH} from $\overline{MCLR}$, as shown in Figure 2-10. The only requirement for exit is that an interval P16 should elapse between the last clock and the program signals on PGC and PGD before removing V_{IH}.

FIGURE 2-9: ENTERING PROGRAM/VERIFY MODE

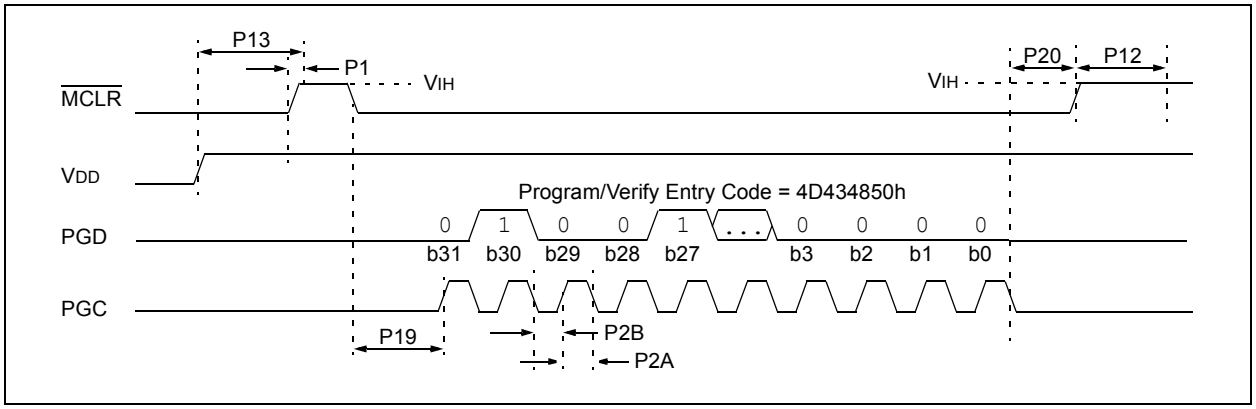
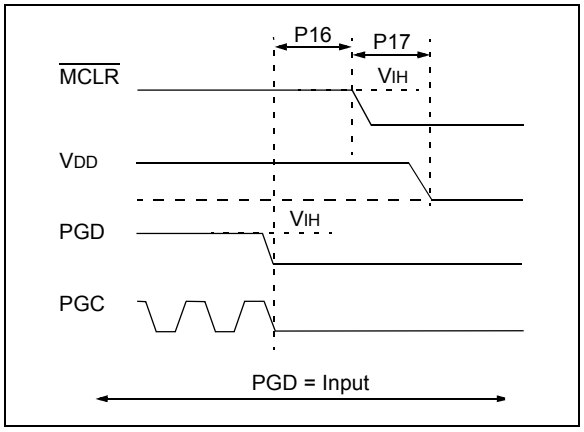


FIGURE 2-10: EXITING PROGRAM/VERIFY MODE



2.5 Serial Program/Verify Operation

The PGC pin is used as a clock input pin and the PGD pin is used for entering command bits and data input/output during serial operation. Commands and data are transmitted on the rising edge of PGC, latched on the falling edge of PGC and are Least Significant bit (LSb) first.

2.5.1 4-BIT COMMANDS

All instructions are 20 bits, consisting of a leading 4-bit command followed by a 16-bit operand, which depends on the type of command being executed. To input a command, PGC is cycled four times. The commands needed for programming and verification are shown in Table 2-3.

Depending on the 4-bit command, the 16-bit operand represents 16 bits of input data, or 8 bits of input data and 8 bits of output data.

Throughout this specification, commands and data are presented as illustrated in Table 2-4. The 4-bit command is shown Most Significant bit (MSb) first. The command operand, or "Data Payload", is shown <MSB><LSB>. Figure 2-11 demonstrates how to serially present a 20-bit command/operand to the device.

2.5.2 CORE INSTRUCTION

The core instruction passes a 16-bit instruction to the CPU core for execution. This is needed to set up registers as appropriate for use with other commands.

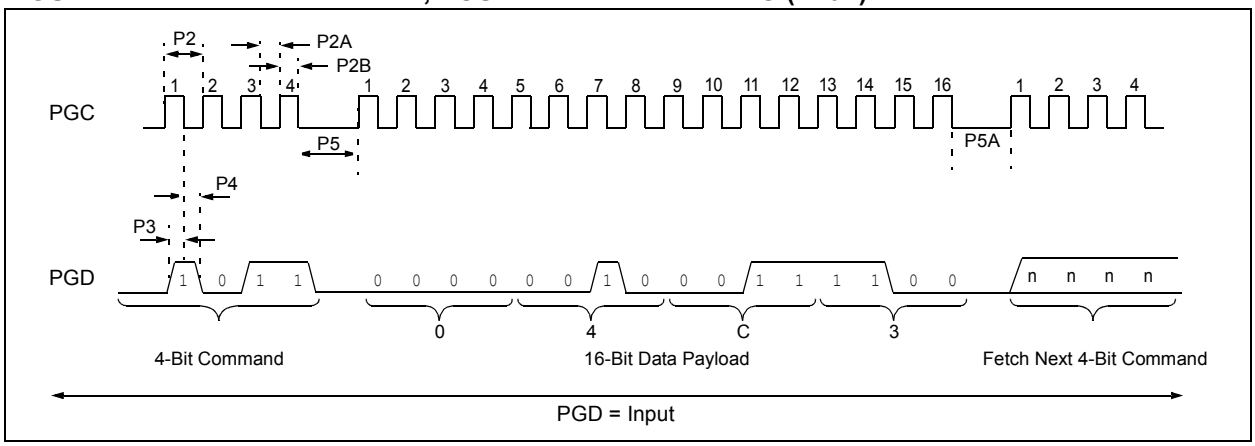
TABLE 2-3: COMMANDS FOR PROGRAMMING

Description	4-Bit Command
Core Instruction (Shift in 16-bit instruction)	0000
Shift out TABLAT register	0010
Table Read	1000
Table Read, post-increment	1001
Table Read, post-decrement	1010
Table Read, pre-increment	1011
Table Write	1100
Table Write, post-increment by 2	1101
Table Write, start programming, post-increment by 2	1110
Table Write, start programming	1111

TABLE 2-4: SAMPLE COMMAND SEQUENCE

4-Bit Command	Data Payload	Core Instruction
1101	3C 40	Table Write, post-increment by 2

FIGURE 2-11: TABLE WRITE, POST-INCREMENT TIMING (1101)



3.0 DEVICE PROGRAMMING

Programming includes the ability to erase or write the memory within the device.

The EECON1 register is used to control Write or Row Erase operations. The WREN bit (EECON1<2>) must be set to enable writes and this must be done prior to initiating a write sequence. It is strongly recommended that the WREN bit only be set immediately prior to a Program Erase.

Note:	The EECON1 register is available only in ICSP Programming mode. In normal operating modes, the corresponding SFR location (FA6h) is unimplemented. Writes to the register during code execution will have no effect; reading the location will return '0's.

3.1 ICSP Erase

3.1.1 ICSP BULK ERASE

The PIC18F6XJXX/8XJXX devices may be Bulk Erased by writing 0180h to the register pair, 3C0005h:3C0004h. The basic sequence is shown in Table 3-1 and demonstrated in Figure 3-1.

Since the code-protect Configuration bit is stored in the program code within code memory, a Bulk Erase operation will also clear any code-protect settings for the device.

The actual Bulk Erase function is a self-timed operation. Once the erase has started (falling edge of the 4th PGC after the NOP command), serial execution will cease until the erase completes (parameter P11). During this time, PGC may continue to toggle but PGD must be held low.

Note:	A Bulk Erase is the only way to reprogram the code-protect Configuration bit from an ON state to an OFF state.

TABLE 3-1: BULK ERASE COMMAND SEQUENCE

4-Bit Command	Data Payload	Core Instruction
0000	0E 3C	MOVLW 3Ch
0000	6E F8	MOVWF TBLPTRU
0000	0E 00	MOVLW 00h
0000	6E F7	MOVWF TBLPTRH
0000	0E 05	MOVLW 05h
0000	6E F6	MOVWF TBLPTRL
1100	01 01	Write 01h to 3C0005h
0000	0E 3C	MOVLW 3Ch
0000	6E F8	MOVWF TBLPTRU
0000	0E 00	MOVLW 00h
0000	6E F7	MOVWF TBLPTRH
0000	0E 04	MOVLW 04h
0000	6E F6	MOVWF TBLPTRL
1100	80 80	Write 80h TO 3C0004h to erase entire device.
0000	00 00	NOP
0000	00 00	Hold PGD low until erase completes.

FIGURE 3-1: BULK ERASE FLOW

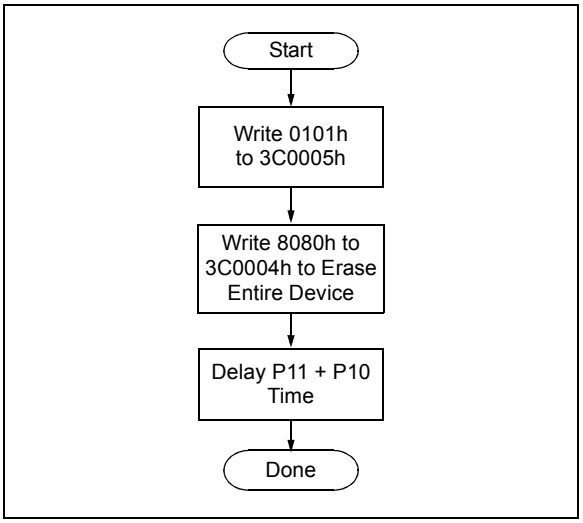
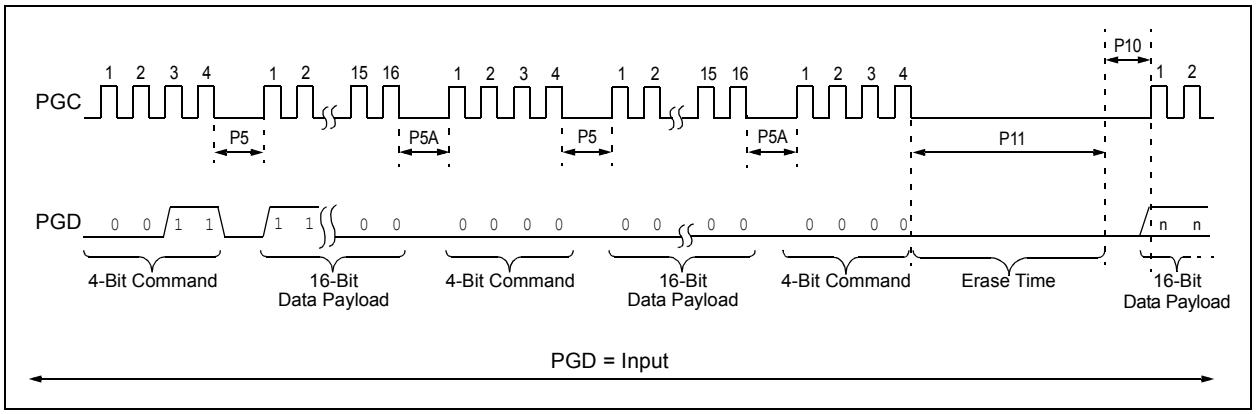


FIGURE 3-2: BULK ERASE TIMING



3.2 Code Memory Programming

Programming code memory is accomplished by first loading data into the write buffer and then initiating a programming sequence. The write buffer for all PIC18F6XJXX/8XJXX devices is 64 bytes. It can be mapped to any location of the same size beginning at 000000h. The actual memory write sequence takes the contents of this buffer and programs the proper amount of code memory that contains the Table Pointer.

Write buffer locations are not cleared following a write operation. The buffer retains its data after the write is complete. This means that the buffer must be written with 64 bytes on each operation. If there are locations in the code memory that are to remain empty, the corresponding locations in the buffer must be filled with FFFFh. This avoids rewriting old data from the previous cycle.

The programming duration is externally timed and is controlled by PGC. After a Start Programming command is issued (4-bit command, '1111'), a NOP is issued, where the 4th PGC is held high for the duration of the programming time, P9.

After PGC is brought low, the programming sequence is terminated. PGC must be held low for the time specified by parameter P10 to allow high-voltage discharge of the memory array.

The code sequence to program a PIC18F6XJXX/8XJXX device is shown in Table 3-2. The flowchart shown in Figure 3-3 depicts the logic necessary to completely write a PIC18F6XJXX/8XJXX device. The timing diagram that details the Start Programming command and parameters P9 and P10 is shown in Figure 3-4.

Note:	The TBLPTR register must point to the same region when initiating the programming sequence as it did when the write buffers were loaded.

TABLE 3-2: WRITE CODE MEMORY CODE SEQUENCE

4-Bit Command	Data Payload	Core Instruction
Step 1: Enable writes.		
0000	84 A6	BSF EECON1, WREN
Step 2: Load write buffer.		
0000	0E <Addr[21:16]>	MOVLW <Addr[21:16]>
0000	6E F8	MOVWF TBLPTRU
0000	0E <Addr[15:8]>	MOVLW <Addr[15:8]>
0000	6E F7	MOVWF TBLPTRH
0000	0E <Addr[7:0]>	MOVLW <Addr[7:0]>
0000	6E F6	MOVWF TBLPTRL
Step 3: Repeat for all but the last two bytes. Any unused locations should be filled with FFFFh.		
1101	<MSB><LSB>	Write 2 bytes and post-increment address by 2.
Step 4: Load write buffer for last two bytes.		
1111	<MSB><LSB>	Write 2 bytes and start programming.
0000	00 00	NOP - hold PGC high for time P9 and low for time P10.
To continue writing data, repeat steps 2 through 4, where the Address Pointer is incremented by 2 at each iteration of the loop.		

FIGURE 3-3: PROGRAM CODE MEMORY FLOW

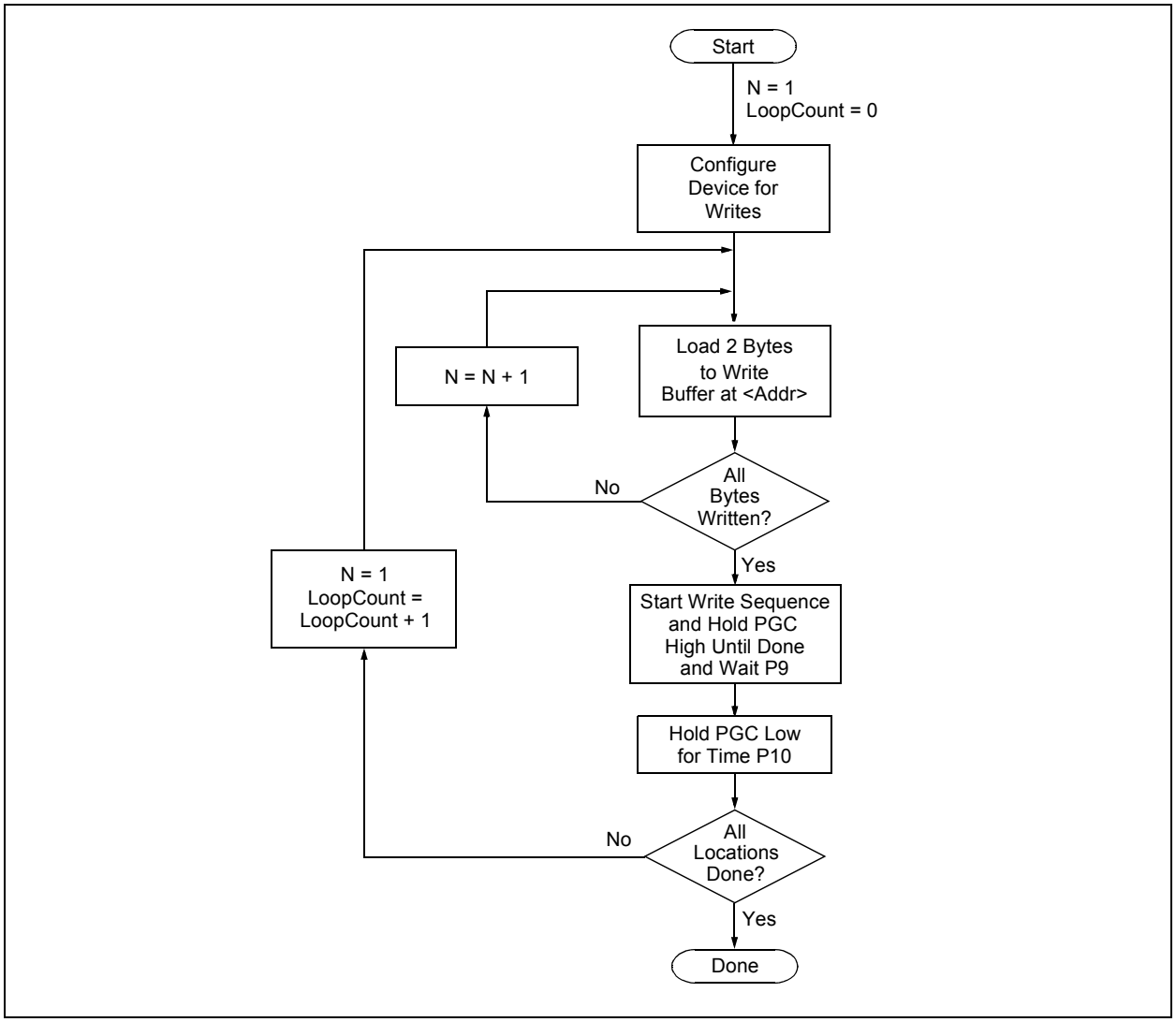
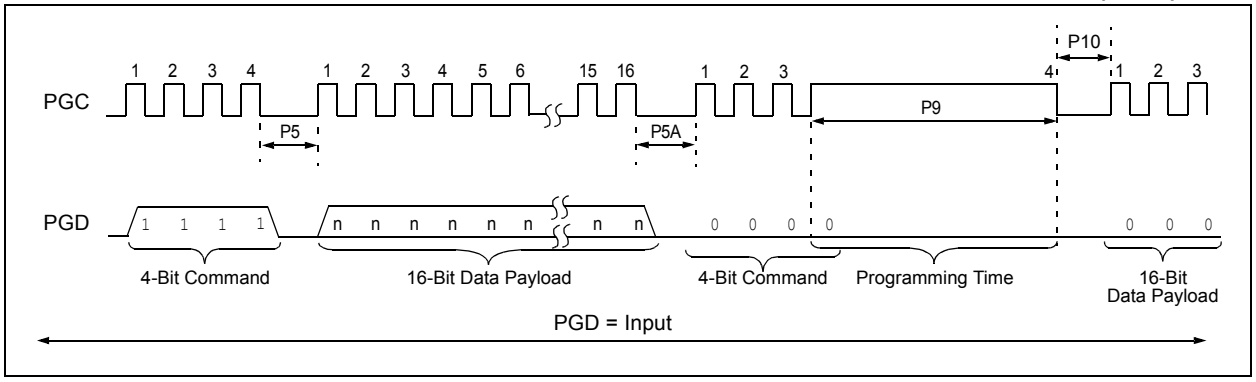


FIGURE 3-4: TABLE WRITE AND START PROGRAMMING INSTRUCTION TIMING (1111)



3.2.1 MODIFYING CODE MEMORY

The previous programming example assumed that the device had been Bulk Erased prior to programming. There may be situations where the user wishes to modify a section of memory in an already programmed device. Doing this is simply an extension of the three basic operations: read, erase and write.

Before a Bulk Erase is performed on the device, the entire code memory is read back (as described in **Section 4.2 "Verify Code Memory and Configuration Word"**) and buffered. It is at this point that the required changes are made on the buffered program. The device is then erased and reprogrammed with the corrected code. An overview of the process is shown in Table 3-3.

3.2.2 CONFIGURATION WORD PROGRAMMING

Since the Flash Configuration Words are stored in program memory, they are programmed as if they were program data. Refer to **Section 3.2 "Code Memory Programming"** and **Section 3.2.1 "Modifying Code Memory"** for methods and examples on programming or modifying program memory. See also **Section 5.0 "Configuration Word"** for additional information on the Configuration Words.

TABLE 3-3: MODIFYING CODE MEMORY

4-Bit Command	Data Payload	Core Instruction
Step 1: Direct access to code memory.		
Step 2: Read and modify code memory (see **Section 4.1 "Read Code Memory"**).		
Step 3: Perform a Bulk Erase (see **Section 3.1.1 "ICSP Bulk Erase"**).		
Step 4: Enable memory writes.		
0000	84 A6	BSF EECON1, WREN
Step 5: Load write buffer. The correct bytes will be selected based on the Table Pointer.		
0000	0E <Addr[21:16]>	MOVLW <Addr[21:16]>
0000	6E F8	MOVWF TBLPTRU
0000	0E <Addr[8:15]>	MOVLW <Addr[8:15]>
0000	6E F7	MOVWF TBLPTRH
0000	0E <Addr[7:0]>	MOVLW <Addr[7:0]>
0000	6E F6	MOVWF TBLPTRL
1101	<MSB><LSB>	Write 2 bytes and post-increment address by 2.
.	.	
.	.	Repeat as many times as necessary to fill the write buffer.
.	.	
1111	<MSB><LSB>	Write 2 bytes and start programming.
0000	00 00	NOP - hold PGC high for time P9 and low for time P10.
To continue modifying data, repeat Step 5, where the Address Pointer is incremented by the appropriate number of bytes at each iteration of the loop. The write cycle must be repeated enough times to completely rewrite the contents of the program memory.		
Step 7: Disable writes.		
0000	94 A6	BCF EECON1, WREN

4.0 READING THE DEVICE

4.1 Read Code Memory

Code memory is accessed one byte at a time via the 4-bit command, '1001' (table read, post-increment). The contents of memory pointed to by the Table Pointer (TBLPTRU:TBLPTRH:TBLPTRL) are serially output on PGD.

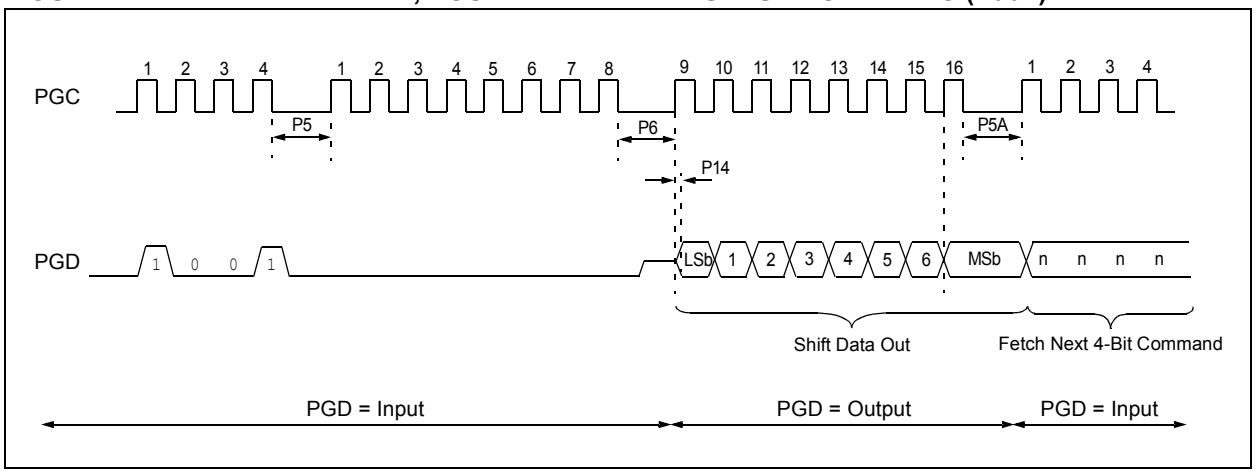
The 4-bit command is shifted in LSb first. The read is executed during the next 8 clocks, then shifted out on PGD during the last 8 clocks, LSb to MSb. A delay of P6 must be introduced after the falling edge of the 8th PGC of the operand to allow PGD to transition from an input to an output. During this time, PGC must be held low (see Figure 4-1). This operation also increments the Table Pointer by one, pointing to the next byte in code memory for the next read.

This technique will work to read any memory in the 000000h to 3FFFFFh address space, so it also applies to reading the Configuration registers.

TABLE 4-1: READ CODE MEMORY SEQUENCE

4-Bit Command	Data Payload	Core Instruction
Step 1: Set Table Pointer.		
0000	0E <Addr[21:16]>	MOVLW Addr[21:16]
0000	6E F8	MOVWF TBLPTRU
0000	0E <Addr[15:8]>	MOVLW <Addr[15:8]>
0000	6E F7	MOVWF TBLPTRH
0000	0E <Addr[7:0]>	MOVLW <Addr[7:0]>
0000	6E F6	MOVWF TBLPTRL
Step 2: Read memory and then shift out on PGD, LSb to MSb.		
1001	00 00	TBLRD *+

FIGURE 4-1: TABLE READ, POST-INCREMENT INSTRUCTION TIMING (1001)



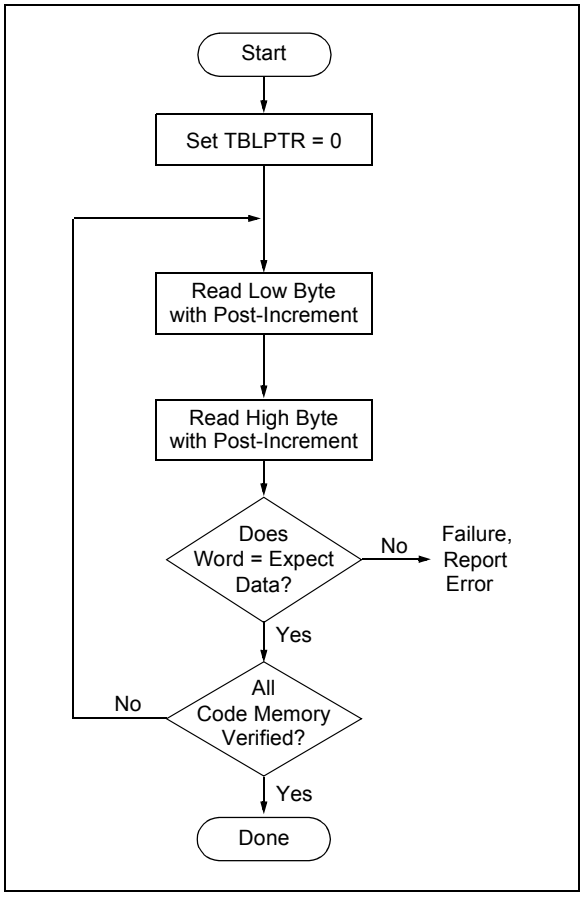
4.2 Verify Code Memory and Configuration Word

The verify step involves reading back the code memory space and comparing it against the copy held in the programmer's buffer. Because the Flash Configuration Words are stored in the top of program memory, it is verified with the rest of the code at this time.

The verify process is shown in the flowchart in Figure 4-2. Memory reads occur a single byte at a time, so two bytes must be read to compare against the word in the programmer's buffer. Refer to **Section 4.1 "Read Code Memory"** for implementation details of reading code memory.

Note:	Because the Flash Configuration Word contains the device code protection bit, code memory should be verified immediately after writing if code protection is enabled. This is because the device will not be readable or verifiable if a device Reset occurs after the Flash Configuration Word (and the CP0 bit) have been cleared.

FIGURE 4-2: VERIFY CODE MEMORY FLOW



4.3 Blank Check

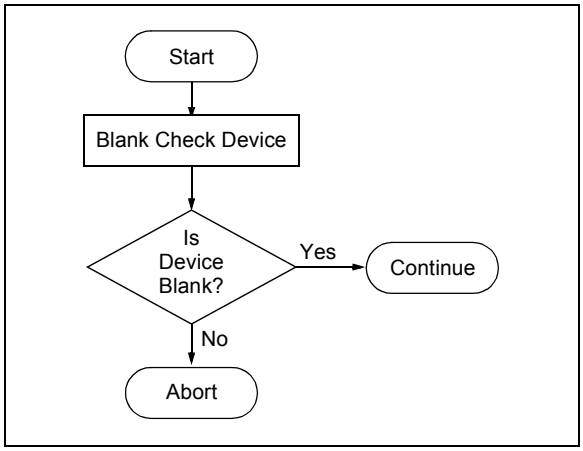
The term "Blank Check" means to verify that the device has no programmed memory cells. All memories must be verified: code memory and Configuration bits. The Device ID registers (3FFFFEh:3FFFFFh) should be ignored.

A "blank" or "erased" memory cell will read as '1', so Blank Checking a device merely means to verify that all bytes read as FFh. The overall process flow is shown in Figure 4-3.

Note:	Following a device Bulk Erase, the Configuration Words will read as shown in Table 5-2.

Given that Blank Checking is merely code verification with FFh expect data, refer to **Section 4.2 "Verify Code Memory and Configuration Word"** for implementation details.

FIGURE 4-3: BLANK CHECK FLOW



5.0 CONFIGURATION WORD

The Configuration Words of the PIC18F6XJXX/8XJXX devices are implemented as volatile memory registers, as opposed to the programmable nonvolatile memory used in other PIC18 devices. All of the Configuration registers (CONFIG1L, CONFIG1H, CONFIG2L, CONFIG2H, CONFIG3L and CONFIG3H) are automatically loaded following each device Reset.

The data for these registers is taken from the four Flash Configuration Words located at the top of program memory. Configuration data is stored in order, starting with CONFIG1L in the lowest Flash address and ending with CONFIG4H in the highest. The mapping to specific Configuration Words is shown in Table 5-1. While four words are reserved in program memory, only three words (CONFIG1L through CONFIG3H) are used for device configuration. Users should always reserve these locations for Configuration Word data and write their application code accordingly.

The upper four bits of each Configuration Word should always be stored in program memory as '1111'. This is done so these program memory addresses will always be '1111 xxxx xxxx xxxx' and interpreted as a NOP instruction if they were ever to be executed. Because the corresponding bits in the Configuration Word are unimplemented, they will not be written to.

The Configuration and Device ID registers are summarized in Table 5-2. A listing of the individual Configuration bits and their options is provided in Table 5-3.

TABLE 5-1: MAPPING OF THE FLASH CONFIGURATION WORDS TO THE CONFIGURATION REGISTERS

Configuration Byte	Code Space Address(1)	Configuration Register Address
CONFIG1L	XXXF8h	300000h
CONFIG1H	XXXF9h	300001h
CONFIG2L	XXXFAh	300002h
CONFIG2H	XXXFBh	300003h
CONFIG3L	XXXFCh	300004h
CONFIG3H	XXXFDh	300005h
CONFIG4L(2)	XXXFEh	300006h
CONFIG4H(2)	XXXFFh	300007h

Note 1: See Table 2-2 for the complete addresses within code space for specific devices and memory sizes.

2: Unimplemented in PIC18F6XJXX/8XJXX devices.

TABLE 5-2: 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	$\overline{\text{DEBUG}}$	XINST	STVREN	—	—	—	—	WDTEN	111- ---1
		$\overline{\text{DEBUG}}$	XINST	STVREN	—	PLLDIV2(1)	PLLDIV1(1)	PLLDIV0(1)	WDTEN	111- 1111(1)
300001h	CONFIG1H	—(2)	—(2)	—(2)	—(2)	—(3)	CP0	—	—	---- 01--
		—(2)	—(2)	—(2)	—(2)	—(3)	CP0	CPUDIV1(1)	CPUDIV0(1)	---- 0111(1)
300002h	CONFIG2L	IESO	FCMEN	—	—	—	FOSC2	FOSC1	FOSC0	11-- -111
300003h	CONFIG2H	—(2)	—(2)	—(2)	—(2)	WDTPS3	WDTPS2	WDTPS1	WDTPS0	---- 1111
300004h	CONFIG3L	WAIT(4)	BW(4)	EMB1(4)	EMB0(4)	EASHFT(4)	—	—	—	---- ----
300005h	CONFIG3H	—(2)	—(2)	—(2)	—(2)	—	—	ECCPMX(4,7,8)	CCP2MX	---- --11
		—(2)	—(2)	—(2)	—(2)	MSSPSEL(6)	PMPMX(7)	ECCPMX(4,7,8)	CCP2MX	---- 1111(7)
		—(2)	—(2)	—(2)	—(2)	MSSPSEL(6)	—	ECCPMX(4,7,8)	CCP2MX	---- 1-11(6)
3FFFFEh	DEVID1(5)	DEV2	DEV1	DEV0	REV4	REV3	REV2	REV1	REV0	See Table 5-4
3FFFFFh	DEVID2(5)	DEV10	DEV9	DEV8	DEV7	DEV6	DEV5	DEV4	DEV3	See Table 5-4

Legend: x = unknown, u = unchanged, – = unimplemented, q = value depends on condition.
Shaded cells are unimplemented, read as '0'.

Note 1: Implemented in PIC18F6XJ5X/8XJ5X devices only.

2: The value of these bits in program memory should always be '1'. This ensures that the location is executed as a NOP if it is accidentally executed.

3: This bit should always be maintained '0'.

4: Implemented in 80-pin devices only.

5: DEVID registers are read-only and cannot be programmed by the user.

6: Implemented in PIC18F6XJ5X/8XJ5X and PIC18F66J11/66J16/67J11/86J11/86J16/87J11 only.

7: Implemented in PIC18F8XJ5X and PIC18F86J11/86J16/87J11 only.

8: Implemented in PIC18FXXJ10/8XJ15 devices only.

TABLE 5-3: PIC18F6XJXX/8XJXX BIT DESCRIPTIONS

Bit Name	Configuration Words	Description
$\overline{\text{DEBUG}}$	CONFIG1L	Background Debugger Enable bit 1 = Background debugger disabled, RB6 and RB7 configured as general purpose I/O pins 0 = Background debugger enabled, RB6 and RB7 are dedicated to In-Circuit Debug
XINST	CONFIG1L	Extended Instruction Set Enable bit 1 = Instruction set extension and Indexed Addressing mode enabled 0 = Instruction set extension and Indexed Addressing mode disabled (Legacy mode)
STVREN	CONFIG1L	Stack Overflow/Underflow Reset Enable bit 1 = Reset on stack overflow/underflow enabled 0 = Reset on stack overflow/underflow disabled
PLLDIV2:PLLDIV0(1)	CONFIG1L	Oscillator Selection bits 111 = No divide – oscillator used directly (4 MHz input) 110 = Oscillator divided by 2 (8 MHz input) 101 = Oscillator divided by 3 (12 MHz input) 100 = Oscillator divided by 4 (16 MHz input) 011 = Oscillator divided by 5 (20 MHz input) 010 = Oscillator divided by 6 (24 MHz input) 001 = Oscillator divided by 10 (40 MHz input) 000 = Oscillator divided by 12 (48 MHz input)
WDTEN	CONFIG1L	Watchdog Timer Enable bit 1 = WDT enabled 0 = WDT disabled (control is placed on SWDTEN bit)
CP0	CONFIG1H	Code Protection bit 1 = Program memory is not code-protected 0 = Program memory is code-protected
CPUDIV1:CPUDIV0(1)	CONFIG1H	CPU System Clock Selection bits 11 = No CPU system clock divide 10 = CPU system clock divided by 2 01 = CPU system clock divided by 3 00 = CPU system clock divided by 6
IESO	CONFIG2L	Internal/External Oscillator Switchover bit 1 = Oscillator Switchover mode enabled 0 = Oscillator Switchover mode disabled
FCMEN	CONFIG2L	Fail-Safe Clock Monitor Enable bit 1 = Fail-Safe Clock Monitor enabled 0 = Fail-Safe Clock Monitor disabled
FOSC2	CONFIG2L	Primary Oscillator Select bit 1 = Default primary oscillator on start-up is EC or HS, depending on the settings of FOSC1:FOSC0; INTRC selected when OSCCON<1:0> = 11 0 = Default primary oscillator on start-up is INTRC; INTRC is also selected when OSCCON<1:0> = 11 or 00

Note 1: Implemented in PIC18FXXJ5X devices only.
2: Implemented in PIC18F66J11/66J16/67J11/86J11/86J16/87J11 devices only.
3: Implemented in 80-pin devices only.

TABLE 5-3: PIC18F6XJXX/8XJXX BIT DESCRIPTIONS (CONTINUED)

Bit Name	Configuration Words	Description
FOSC1:FOSC0	CONFIG2L	Oscillator Selection bits 11 = EC oscillator, PLL enabled and under software control, CLKO function on OSC2 10 = EC oscillator, CLKO function on OSC2 01 = HS oscillator, PLL enabled and under software control 00 = HS oscillator
FOSC1:FOSC0	CONFIG2L	Oscillator Selection bits(1) 111 = ECPLL oscillator with PLL enabled, CLKO on RA6 and port function on RA7, ECPLL oscillator used by USB 110 = EC oscillator with CLKO on RA6 and port function on RA7, EC oscillator used by USB 101 = HSPLL oscillator with PLL enabled 100 = HS oscillator, HS oscillator used by USB 011 = INTOSCPLLO oscillator with INTOSC and PLL enabled, CLKO on RA6 and port function on RA7 010 = INTOSCPLL oscillator, port function on RA6 and RA7 001 = INTOSCO internal oscillator block (INTRC/INTOSC) with CLKO on RA6, port function on RA7 000 = INTOSC internal oscillator block (INTRC/INTOSC), port function on RA6 and RA7 Oscillator Selection bits(2) 111 = ECPLL oscillator with 4xPLL enabled 110 = EC oscillator 101 = HSPLL oscillator with 4xPLL enabled 100 = HS oscillator 011 = INTOSCPLLO, INTOSC with 4xPLL, CLKO on RA6 and port function on RA7 010 = INTOSCPLL, INTOSC with 4xPLL oscillator, port function on RA6 and RA7 001 = INTOSCO internal oscillator block (INTRC/INTOSC) with CLKO on RA6, port function on RA7 000 = INTOSC internal oscillator block (INTRC/INTOSC), port function on RA6 and RA7
WDTPS3:WDTPS0	CONFIG2H	Watchdog Timer Postscale Select bits 1111 = 1:32,768 1110 = 1:16,384 1101 = 1:8,192 1100 = 1:4,096 1011 = 1:2,048 1010 = 1:1,024 1001 = 1:512 1000 = 1:256 0111 = 1:128 0110 = 1:64 0101 = 1:32 0100 = 1:16 0011 = 1:8 0010 = 1:4 0001 = 1:2 0000 = 1:1

Note 1: Implemented in PIC18FXXJ5X devices only.
2: Implemented in PIC18F66J11/66J16/67J11/86J11/86J16/87J11 devices only.
3: Implemented in 80-pin devices only.

TABLE 5-3: PIC18F6XJXX/8XJXX BIT DESCRIPTIONS (CONTINUED)

Bit Name	Configuration Words	Description
WAIT	CONFIG3L	External Bus Wait Enable bit 1 = Wait states for operations on external memory bus disabled 0 = Wait states for operations on external memory bus enabled
BW	CONFIG3L	Data Bus Width Select bit 1 = 16-Bit External Bus mode 0 = 8-Bit External Bus mode
EMB1:EMB0	CONFIG3L	External Memory Bus Configuration bits 00 = Extended Microcontroller mode, 20-Bit Address mode 01 = Extended Microcontroller mode,16-Bit Address mode 10 = Extended Microcontroller mode,12-Bit Address mode 11 = Microcontroller mode – external bus disabled
EASHFT	CONFIG3L	External Address Bus Shift Enable bit 1 = Address shifting enabled; address on external bus is offset to start at 000000h 0 = Address shifting disabled; address on external bus reflects the PC value
MSSPSEL(1,2)	CONFIG3H	MSSP Address Select bit 1 = 7-Bit Address Mask mode 0 = 5-Bit Address Mask mode
PMPMX(1,2,3)	CONFIG3H	PMP Pin Select bit 1 = PMP port pins connected to EMB 0 = PMP port pins not connected to EMB
ECCPMX	CONFIG3H	ECCP MUX bit 1 = ECCP1 outputs (P1B/P1C) are multiplexed with RE6 and RE5; ECCP3 outputs (P3B/P3C) are multiplexed with RE4 and RE3 0 = ECCP1 outputs (P1B/P1C) are multiplexed with RH7 and RH6; ECCP3 outputs (P3B/P3C) are multiplexed with RH5 and RH4
CCP2MX	CONFIG3H	CCP2 MUX bit 1 = ECCP2/P2A is multiplexed with RC1 0 = ECCP2/P2A is multiplexed with RE7 in Microcontroller mode (all devices) or with RB3 in Extended Microcontroller mode (80-pin devices only)

Note 1: Implemented in PIC18FXXJ5X devices only.
2: Implemented in PIC18F66J11/66J16/67J11/86J11/86J16/87J11 devices only.
3: Implemented in 80-pin devices only.

5.1 Device ID Word

The Device ID word for PIC18F6XJXX/8XJXX devices is located at 3FFFFEh:3FFFFFh. These read-only bits may be used by the programmer to identify what device type is being programmed and read out normally, even after code or read-protected. The process for reading the Device IDs is shown in Figure 5-1. A complete list of Device ID values for PIC18F6XJXX/8XJXX devices is presented in Table 5-4.

FIGURE 5-1: READ DEVICE ID WORD FLOW

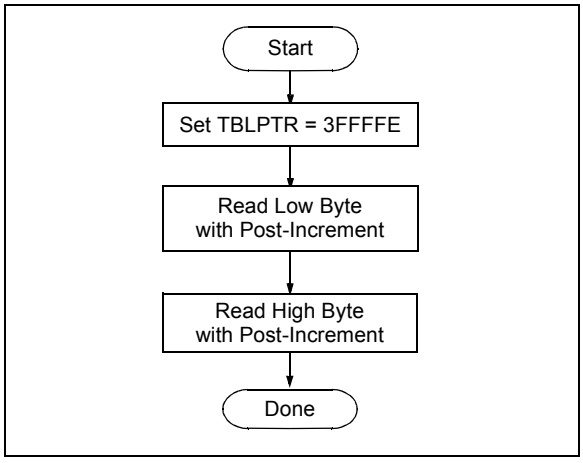


TABLE 5-4: DEVICE ID VALUE

Device	Device ID Value	
	DEVID2	DEVID1
PIC18F63J11	39h	000x xxxx
PIC18F63J90	38h	000x xxxx
PIC18F64J11	39h	001x xxxx
PIC18F64J90	38h	001x xxxx
PIC18F65J10	15h	001x xxxx
PIC18F65J11	39h	011x xxxx
PIC18F65J15	15h	010x xxxx
PIC18F65J50	41h	000x xxxx
PIC18F65J90	38h	011x xxxx
PIC18F66J10	15h	011x xxxx
PIC18F66J11	44h	010x xxxx
PIC18F66J15	15h	100x xxxx
PIC18F66J16	44h	011x xxxx
PIC18F66J50	41h	010x xxxx
PIC18F66J55	41h	011x xxxx
PIC18F67J10	15h	101x xxxx
PIC18F67J11	44h	100x xxxx
PIC18F67J50	41h	100x xxxx
PIC18F83J11	39h	100x xxxx
PIC18F83J90	38h	100x xxxx
PIC18F84J11	39h	101x xxxx
PIC18F84J90	38h	101x xxxx
PIC18F85J10	15h	111x xxxx
PIC18F85J11	39h	111x xxxx
PIC18F85J15	17h	000x xxxx
PIC18F85J50	41h	101x xxxx
PIC18F85J90	38h	111x xxxx
PIC18F86J10	17h	001x xxxx
PIC18F86J11	44h	111x xxxx
PIC18F86J15	17h	010x xxxx
PIC18F86J16	45h	000x xxxx
PIC18F86J50	41h	111x xxxx
PIC18F86J55	42h	000x xxxx
PIC18F87J10	17h	011x xxxx
PIC18F87J11	45h	001x xxxx
PIC18F87J50	42h	001x xxxx

Legend: The 'x's in DEVID1 are reserved for the device revision code.

5.2 Checksum Computation

The checksum is calculated by summing the following:

- The contents of all code memory locations
- The Configuration Block (CFGB), appropriately masked
- ID locations

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

Table 5-5 (pages 23 through 25) describes how to calculate the checksum for each device.

Note: The checksum calculation differs depending on the code-protect setting. Since the code memory locations read out differently depending on the code-protect setting, the table describes how to manipulate the actual code memory values to simulate the values that would be read from a protected device. When calculating a checksum by reading a device, the entire code memory can simply be read and summed. The Configuration Word and ID locations can always be read.

TABLE 5-5: CHECKSUM EQUATION FOR PIC18F6XJXX/8XJXX

Family	Device	Read Code Protection	Checksum Computation
PIC18F87J10	PIC18F65J10	Disabled	CFGB60 + SUM(0000:7FF7h)
		Enabled	0000h
	PIC18F65J15	Disabled	CFGB60 + SUM(0000:BFF7h)
		Enabled	0000h
	PIC18F66J10	Disabled	CFGB60 + SUM(0000:FFF7h)
		Enabled	0000h
	PIC18F66J15	Disabled	CFGB60 + SUM(00000:17FF7h)
		Enabled	0000h
	PIC18F67J10	Disabled	CFGB60 + SUM(00000:1FFF7h)
		Enabled	0000h
	PIC18F85J10	Disabled	CFGB80 + SUM(0000:7FF7h)
		Enabled	0000h
	PIC18F85J15	Disabled	CFGB80 + SUM(0000:BFF7h)
		Enabled	0000h
	PIC18F86J10	Disabled	CFGB80 + SUM(0000:FFF7h)
		Enabled	0000h
	PIC18F86J15	Disabled	CFGB80 + SUM(00000:17FF7h)
		Enabled	0000h
	PIC18F87J10	Disabled	CFGB80 + SUM(00000:1FFF7h)
		Enabled	0000h
CFGB80 = Byte sum of [(CW1 & 04E1h) + (CW2 & 0FC7h) + (CW3 & 03F8h)] CFGB60 = Byte sum of [(CW1 & 04E1h) + (CW2 & 0FC7h) + (CW3 & 0100h)]			

Legend:

Item		Description
SUM(a:b)	=	Byte sum of locations a to b inclusive (all 3 bytes of code memory)
+	=	Addition
CW	=	Configuration Word
CFGB	=	Configuration Block (Masked)

Note: CW3 address is (last location – 2) of implemented program memory; CW2 is (last location – 4); CW1 is (last location – 6).

TABLE 5-5: CHECKSUM EQUATION FOR PIC18F6XJXX/8XJXX (CONTINUED)

Family	Device	Read Code Protection	Checksum Computation
PIC18F85J90	PIC18F63J90	Disabled	CFGB + SUM(0000:1FF7h)
		Enabled	0000h
	PIC18F64J90	Disabled	CFGB + SUM(0000:3FF7h)
		Enabled	0000h
	PIC18F64J95	Disabled	CFGB + SUM(0000:5FF7h)
		Enabled	0000h
	PIC18F65J90	Disabled	CFGB + SUM(0000:7FF7h)
		Enabled	0000h
	PIC18F83J90	Disabled	CFGB + SUM(0000:1FF7h)
		Enabled	0000h
	PIC18F84J90	Disabled	CFGB + SUM(0000:3FF7h)
		Enabled	0000h
	PIC18F84J95	Disabled	CFGB + SUM(0000:5FF7h)
		Enabled	0000h
	PIC18F85J90	Disabled	CFGB + SUM(0000:7FF7h)
		Enabled	0000h
CFGB = Byte sum of [(CW1 & 0CE1h) + (CW2 & 0FC7h) + (CW3 & 0100h)]			
PIC18F85J11	PIC18F63J11	Disabled	CFGB60 + SUM(0000:1FF7h)
		Enabled	0000h
	PIC18F64J11	Disabled	CFGB60 + SUM(0000:3FF7h)
		Enabled	0000h
	PIC18F64J16	Disabled	CFGB60 + SUM(0000:5FF7h)
		Enabled	0000h
	PIC18F65J11	Disabled	CFGB60 + SUM(0000:7FF7h)
		Enabled	0000h
	PIC18F83J11	Disabled	CFGB80 + SUM(0000:1FF7h)
		Enabled	0000h
	PIC18F84J11	Disabled	CFGB80 + SUM(0000:3FF7h)
		Enabled	0000h
	PIC18F84J16	Disabled	CFGB80 + SUM(0000:5FF7h)
		Enabled	0000h
	PIC18F85J11	Disabled	CFGB80 + SUM(0000:7FF7h)
		Enabled	0000h
CFGB80 = Byte sum of [(CW1 & 0CE1h) + (CW2 & 0FC7h) + (CW3 & 01F8h)] CFGB60 = Byte sum of [(CW1 & 0CE1h) + (CW2 & 0FC7h) + (CW3 & 0100h)]			

Legend:

Item		Description
SUM(a:b)	=	Byte sum of locations a to b inclusive (all 3 bytes of code memory)
+	=	Addition
CW	=	Configuration Word
CFGB	=	Configuration Block (Masked)

Note: CW3 address is (last location – 2) of implemented program memory; CW2 is (last location – 4); CW1 is (last location – 6).

TABLE 5-5: CHECKSUM EQUATION FOR PIC18F6XJXX/8XJXX (CONTINUED)

Family	Device	Read Code Protection	Checksum Computation
PIC18F87J50	PIC18F65J50	Disabled	CFGB60 + SUM(0000:7FF7h)
		Enabled	0000h
	PIC18F65J55	Disabled	CFGB60 + SUM(0000:BFF7h)
		Enabled	0000h
	PIC18F66J50	Disabled	CFGB60 + SUM(0000:FFF7h)
		Enabled	0000h
	PIC18F66J55	Disabled	CFGB60 + SUM(00000:17FF7h)
		Enabled	0000h
	PIC18F67J50	Disabled	CFGB60 + SUM(00000:1FFF7h)
		Enabled	0000h
	PIC18F85J50	Disabled	CFGB80 + SUM(0000:7FF7h)
		Enabled	0000h
	PIC18F85J55	Disabled	CFGB80 + SUM(0000:BFF7h)
		Enabled	0000h
	PIC18F86J50	Disabled	CFGB80 + SUM(0000:FFF7h)
		Enabled	0000h
	PIC18F86J55	Disabled	CFGB80 + SUM(00000:17FF7h)
		Enabled	0000h
	PIC18F87J50	Disabled	CFGB80 + SUM(00000:1FFF7h)
		Enabled	0000h
CFGB80 = Byte sum of [(CW1 & 07FFh) + (CW2 & 0FC7h) + (CW3 & 0FF8h)] CFGB60 = Byte sum of [(CW1 & 07FFh) + (CW2 & 0FC7h) + (CW3 & 0900h)]			
PIC18F87J11	PIC18F65J16	Disabled	CFGB60 + SUM(0000:BFF7h)
		Enabled	0000h
	PIC18F66J11	Disabled	CFGB60 + SUM(0000:FFF7h)
		Enabled	0000h
	PIC18F66J16	Disabled	CFGB60 + SUM(00000:17FF7h)
		Enabled	0000h
	PIC18F67J11	Disabled	CFGB60 + SUM(00000:1FFF7h)
		Enabled	0000h
	PIC18F85J16	Disabled	CFGB80 + SUM(0000:BFF7h)
		Enabled	0000h
	PIC18F86J11	Disabled	CFGB80 + SUM(0000:FFF7h)
		Enabled	0000h
	PIC18F86J16	Disabled	CFGB80 + SUM(00000:17FF7h)
		Enabled	0000h
	PIC18F87J11	Disabled	CFGB80 + SUM(00000:1FFF7h)
		Enabled	0000h
CFGB80 = Byte sum of [(CW1 & 07E1h) + (CW2 & 0FC7h) + (CW3 & 0FF8h)] CFGB60 = Byte sum of [(CW1 & 07E1h) + (CW2 & 0FC7h) + (CW3 & 0900h)]			

Legend:

Item		Description
SUM(a:b)	=	Byte sum of locations a to b inclusive (all 3 bytes of code memory)
+	=	Addition
CW	=	Configuration Word
CFGB	=	Configuration Block (Masked)

Note: CW3 address is (last location – 2) of implemented program memory; CW2 is (last location – 4); CW1 is (last location – 6).

6.0 AC/DC CHARACTERISTICS AND TIMING REQUIREMENTS FOR PROGRAM/VERIFY TEST MODE

Standard Operating Conditions
Operating Temperature: 25°C is recommended

Param No.	Symbol	Characteristic		Min	Max	Units	Conditions
	VDDCORE	External Supply Voltage for Microcontroller Core		2.25	2.70	V	**(Note 1)**
D111	VDD	Supply Voltage During Programming	ENVREG = VSS	VDDCORE	3.60	V	Normal programming **(Note 2)**
			ENVREG = VDD	2.65	3.60		
D112	IPP	Programming Current on $\overline{\text{MCLR}}$		—	5	μA	
D113	IDDP	Supply Current During Programming		—	10	mA	
D031	VIL	Input Low Voltage		VSS	0.2 VDD	V	
D041	VIH	Input High Voltage		0.8 VDD	VDD	V	
D080	VOL	Output Low Voltage		—	0.6	V	IOL = 8.5 mA @ 3.6V
D090	VOH	Output High Voltage		VDD – 0.7	—	V	IOH = -3.0 mA @ 3.6V
D012	CIO	Capacitive Loading on I/O pin (PGD)		—	50	pF	To meet AC specifications
	CF	Filter Capacitor Value on VCAP		1	10	μF	Required for controller core operation when voltage regulator is enabled

Note 1: VDDCORE must be supplied to the VDDCORE/VCAP pin if the on-chip voltage regulator is disabled. See **Section 2.1.1 "On-Chip Voltage Regulator"** for more information.

2: VDD must also be supplied to the AVDD pins during programming and to the ENVREG pin if the on-chip voltage regulator is used. AVDD and AVSS should always be within ±0.3V of VDD and VSS, respectively.

6.0 AC/DC CHARACTERISTICS AND TIMING REQUIREMENTS FOR PROGRAM/VERIFY TEST MODE (CONTINUED)

Standard Operating Conditions
Operating Temperature: 25°C is recommended

Param No.	Symbol	Characteristic		Min	Max	Units	Conditions
P1	TR	$\overline{MCLR}$ Rise Time to Enter Program/Verify mode		—	1.0	μs	
P2	TPGC	Serial Clock (PGC) Period		100	—	ns	
P2A	TPGCL	Serial Clock (PGC) Low Time		40	—	ns	
P2B	TPGCH	Serial Clock (PGC) High Time		40	—	ns	
P3	TSET1	Input Data Setup Time to Serial Clock ↓		15	—	ns	
P4	THLD1	Input Data Hold Time from PGC ↓		15	—	ns	
P5	TDLY1	Delay between 4-bit Command and Command Operand		40	—	ns	
P5A	TDLY1A	Delay between 4-bit Command Operand and Next 4-bit Command		40	—	ns	
P6	TDLY2	Delay between Last PGC ↓ of Command Byte to First PGC ↑ of Read of Data Word		20	—	ns	
P9	TDLY5	PGC High Time (minimum programming time)		2.8	—	ms	
P10	TDLY6	PGC Low Time after Programming		400	—	ns	
P11	TDLY7	Delay to allow Bulk Erase to Occur		400	—	ms	
P12	THLD2	Input Data Hold Time from $\overline{MCLR}$ ↑		300	—	μs	
P13	TSET2	VDD ↑ Setup Time to $\overline{MCLR}$ ↑		100	—	ns	
P14	TVALID	Data Out Valid from PGC ↑		10	—	ns	
P16	TDLY8	Delay between Last PGC ↓ and $\overline{MCLR}$ ↓		0	—	s	
P17	THLD3	$\overline{MCLR}$ ↓ to VDD ↓		—	100	ns	
P19	TKEY1	Delay from First $\overline{MCLR}$ ↓ to First PGC ↑ for Key Sequence on PGD		10	—	μs	
P20	TKEY2	Delay from Last PGC ↓ for Key Sequence on PGD to Second $\overline{MCLR}$ ↑		40	—	ns	

Note 1: VDDCORE must be supplied to the VDDCORE/VCAP pin if the on-chip voltage regulator is disabled. See **Section 2.1.1 "On-Chip Voltage Regulator"** for more information.

2: VDD must also be supplied to the AVDD pins during programming and to the ENVREG pin if the on-chip voltage regulator is used. AVDD and AVSS should always be within ±0.3V of VDD and VSS, respectively.

NOTES:

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.

Trademarks

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

Microchip received ISO/TS-16949:2002 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

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

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

Santa Clara
Santa Clara, CA
Tel: 408-961-6444
Fax: 408-961-6445

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

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

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: 86-28-8665-5511
Fax: 86-28-8665-7889

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: 86-29-8833-7252
Fax: 86-29-8833-7256

ASIA/PACIFIC

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

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

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 - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

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: 43-7242-2244-39
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

06/25/07