Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.
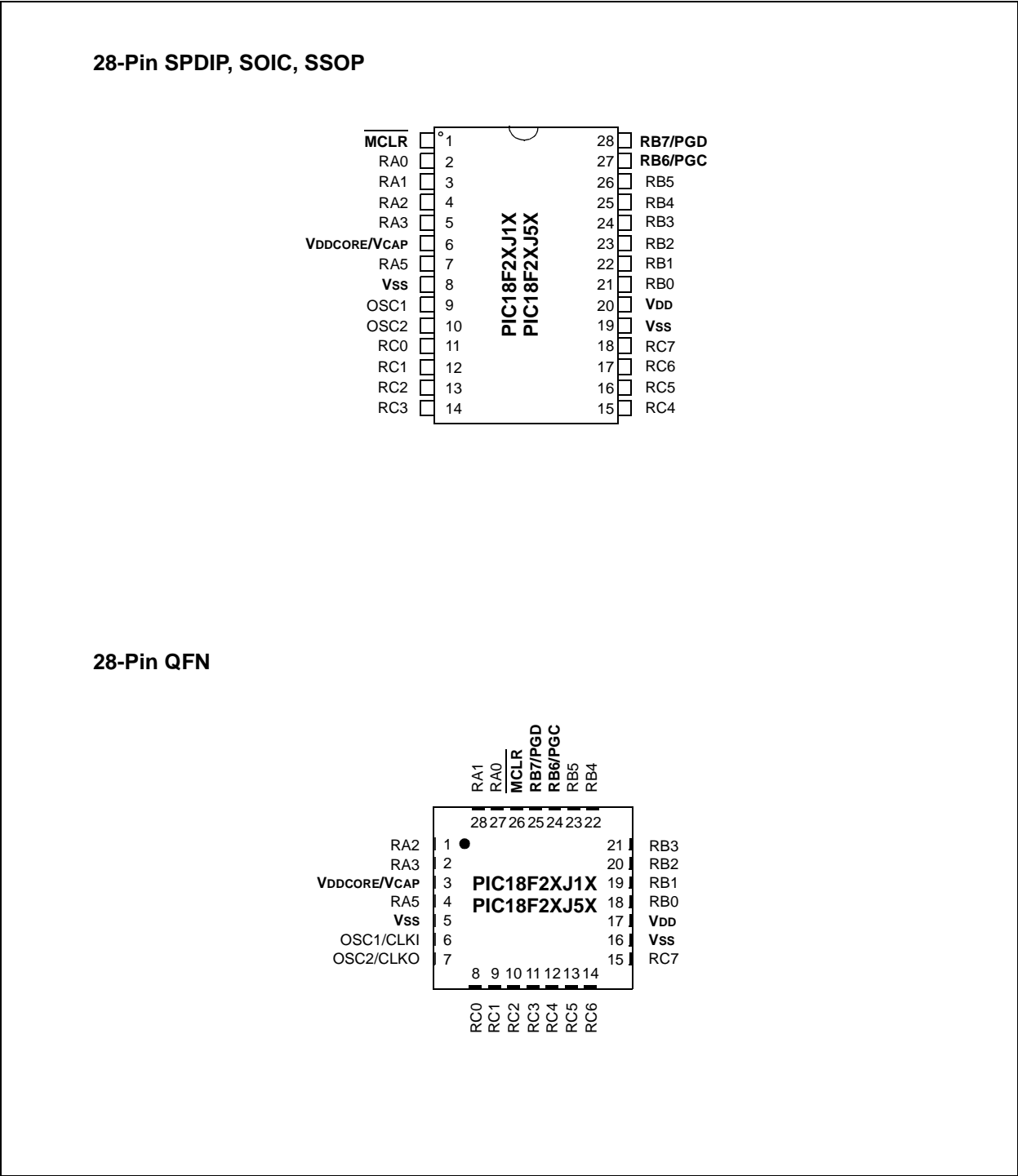
Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	48MHz
Connectivity	I²C, SPI, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	34
Program Memory Size	64KB (32K x 16)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	3.8K x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 3.6V
Data Converters	A/D 13x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-TQFP
Supplier Device Package	44-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf46j50-i-pt

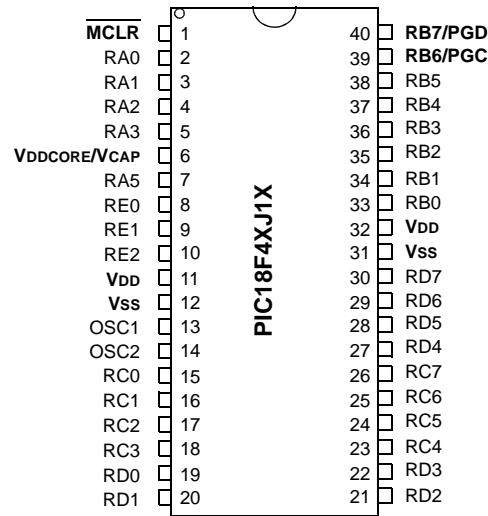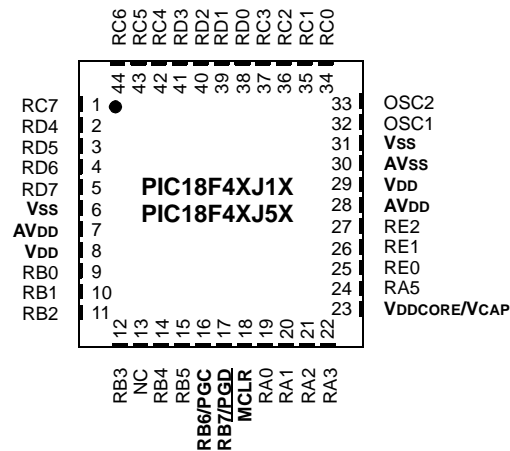**FIGURE 2-1:** **PIC18F2XJXX/4XJXX FAMILY PIN DIAGRAMS**

28-Pin SPDIP, SOIC, SSOP

			PIC18F2XJ1X / PIC18F2XJ5X		
MCLR	1			28	RB7/PGD
RA0	2			27	RB6/PGC
RA1	3			26	RB5
RA2	4			25	RB4
RA3	5			24	RB3
VDDCORE/VCAP	6			23	RB2
RA5	7			22	RB1
Vss	8			21	RB0
OSC1	9			20	VDD
OSC2	10			19	Vss
RC0	11			18	RC7
RC1	12			17	RC6
RC2	13			16	RC5
RC3	14			15	RC4

28-Pin QFN

Top pins (28–22): RA1, RA0, MCLR, RB7/PGD, RB6/PGC, RB5, RB4

PIC18F2XJ1X / PIC18F2XJ5X

RA2	1	21	RB3
RA3	2	20	RB2
VDDCORE/VCAP	3	19	RB1
RA5	4	18	RB0
Vss	5	17	VDD
OSC1/CLKI	6	16	Vss
OSC2/CLKO	7	15	RC7

Bottom pins (8–14): RC0, RC1, RC2, RC3, RC4, RC5, RC6

FIGURE 2-2: **PIC18F2XJXX/4XJXX FAMILY PIN DIAGRAMS (CONTINUED)**

40-Pin PDIP

```
                    ┌───────┐
         MCLR ──┤ 1      40 ├── RB7/PGD
          RA0 ──┤ 2      39 ├── RB6/PGC
          RA1 ──┤ 3      38 ├── RB5
          RA2 ──┤ 4      37 ├── RB4
          RA3 ──┤ 5      36 ├── RB3
  VDDCORE/VCAP ──┤ 6      35 ├── RB2
          RA5 ──┤ 7      34 ├── RB1
          RE0 ──┤ 8      33 ├── RB0
          RE1 ──┤ 9  P   32 ├── VDD
          RE2 ──┤ 10 I    31 ├── VSS
          VDD ──┤ 11 C    30 ├── RD7
          VSS ──┤ 12 1    29 ├── RD6
         OSC1 ──┤ 13 8    28 ├── RD5
         OSC2 ──┤ 14 F    27 ├── RD4
          RC0 ──┤ 15 4    26 ├── RC7
          RC1 ──┤ 16 X    25 ├── RC6
          RC2 ──┤ 17 J    24 ├── RC5
          RC3 ──┤ 18 1    23 ├── RC4
          RD0 ──┤ 19 X    22 ├── RD3
          RD1 ──┤ 20     21 ├── RD2
                    └───────┘
```

44-Pin QFN

PIC18F4XJ1X
PIC18F4XJ5X

44-Pin TQFP

PIC18F4XJ1X
PIC18F4XJ5X
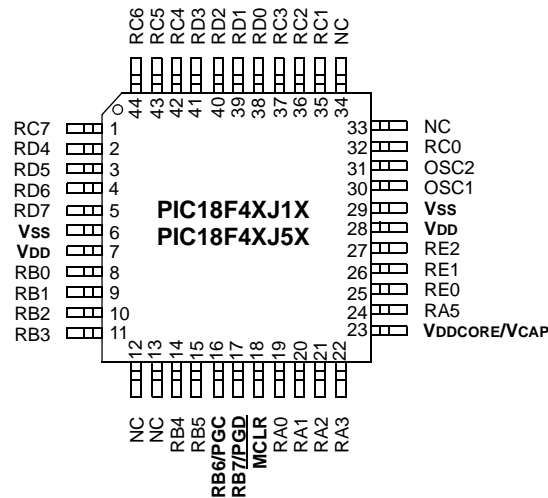
PIC18F2XJXX/4XJXX FAMILY

FIGURE 2-4: **MEMORY MAPS FOR PIC18F2XJXX/4XJXX FAMILY DEVICES**[1]

| | PIC18FX4JXX (16 Kbytes) | PIC18FX5JXX (32 Kbytes) | PIC18FX6JXX (64 Kbytes) | PIC18FX7JXX (128 Kbytes) | |

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: Addresses 300006h and 300007h are unimplemented in PIC18F45J10 Family devices.

2.3 Overview of the Programming Process

Figure 2-5 shows the high-level overview of the programming process in which a Bulk Erase is performed first, then the code memory is programmed. Since only nonvolatile Configuration Words are within the code memory space, the Configuration Words are also programmed as code. Code memory (including the Configuration Words) is then verified to ensure that programming was successful.

FIGURE 2-5: HIGH-LEVEL PROGRAMMING FLOW
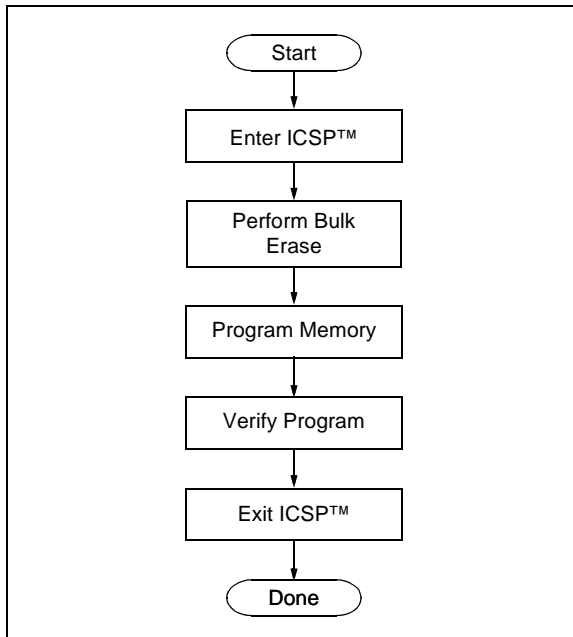
2.4 Entering and Exiting ICSP™ Program/Verify Mode

Entry into ICSP modes for PIC18F2XJXX/4XJXX Family devices is somewhat different than previous PIC18 devices. As shown in Figure 2-6, entering ICSP Program/Verify mode requires three steps:

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

The programming voltage applied to $\overline{\text{MCLR}}$ is VIH, or essentially, VDD. There is no minimum time requirement for holding at VIH. After VIH 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', which is 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, VIH must be applied to $\overline{\text{MCLR}}$ 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 may 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 VIH from $\overline{\text{MCLR}}$, as shown in Figure 2-7. The only requirement for exit is that an interval, P16, should elapse between the last clock and program signals on PGC and PGD before removing VIH.
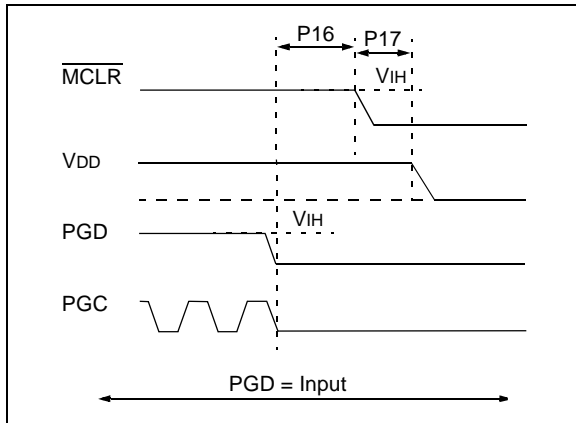
When VIH is reapplied to $\overline{\text{MCLR}}$, the device will enter the ordinary operational mode and begin executing the application instructions.

FIGURE 2-6: ENTERING PROGRAM/VERIFY MODE
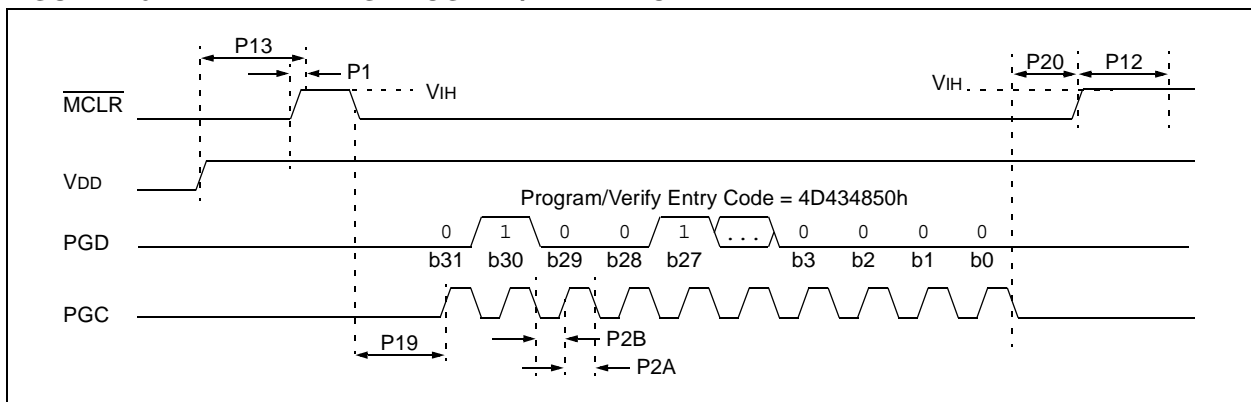
FIGURE 2-7: **EXITING PROGRAM/
VERIFY MODE**

PGD = Input

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 FOUR-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 eight bits of input data and eight 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 2-8 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 2-8: **TABLE WRITE, POST-INCREMENT TIMING (1101)**

PGD = Input
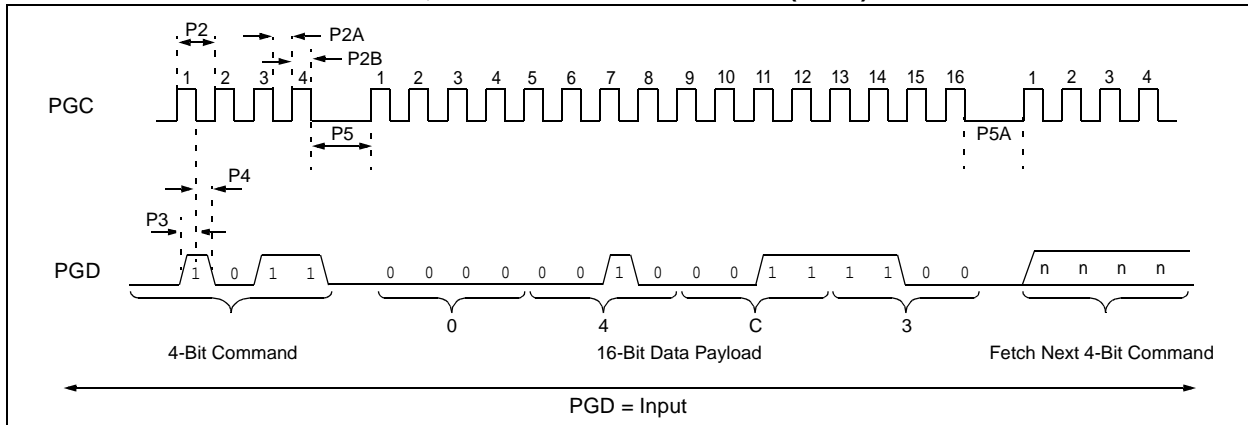
PIC18F2XJXX/4XJXX FAMILY

3.1.2 ICSP™ ROW ERASE

It is possible to erase one row (1024 bytes of data), provided the block is not code-protected or erase/write-protected. Rows are located at static boundaries beginning at program memory address 000000h, extending to the internal program memory limit. Refer to **Section 2.2 "Memory Maps"**).

The Row Erase duration is internally timed. After the WR bit in EECON1 is set, a NOP instruction is issued, where the 4th PGC is held high for the duration of the Row Erase time, P10.

The code sequence to Row Erase a PIC18F2XJXX/4XJXX Family device is shown in Table 3-2. The flowchart shown in Figure 3-4 depicts the logic necessary to completely erase a PIC18F2XJXX/4XJXX

Family device. The timing diagram that details the Row Erase command and parameter P10 is shown in Figure 3-3.

Note 1: If the last row of program memory is erased, bit 3 of CONFIG1H must also be programmed as '0'.

    2: The TBLPTR register can point at any byte within the row intended for erase.

    3: If code protection has been enabled, ICSP Bulk Erase (all program memory erased) operations can be used to disable code protection. ICSP Row Erase operations cannot be used to disable code protection.


TABLE 3-2: ERASE CODE MEMORY CODE SEQUENCE

4-Bit Command	Data Payload	Core Instruction
Step 1: Enable memory writes.		
0000	84 A6	BSF EECON1, WREN
Step 2: Point to first row in code memory.		
0000	6A F8	CLRF TBLPTRU
0000	6A F7	CLRF TBLPTRH
0000	6A F6	CLRF TBLPTRL
Step 3: Enable erase and erase single row.		
0000	88 A6	BSF EECON1, FREE
0000	82 A6	BSF EECON1, WR
0000	00 00	NOP – hold PGC high for time P10.
Step 4: Repeat Step 3, with Address Pointer incremented by 1024, until all rows are erased.		

FIGURE 3-3: SET WR AND START ROW ERASE TIMING

© 2008-2015 Microchip Technology Inc.
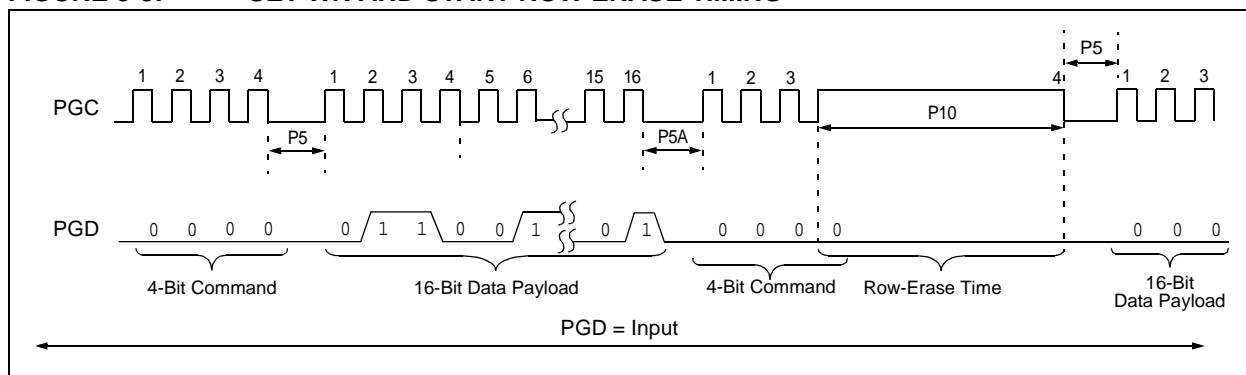
FIGURE 3-4: **SINGLE ROW ERASE CODE MEMORY FLOW**
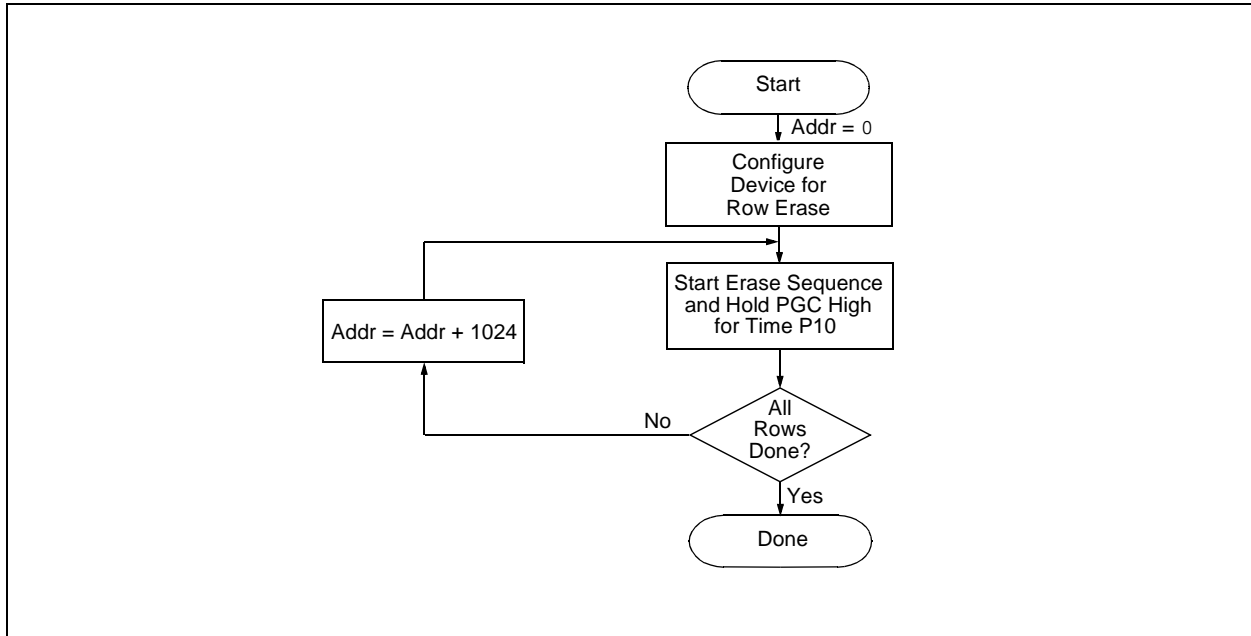
3.2 Code Memory Programming

Programming code memory is accomplished by first loading data into the write buffer and then initiating a programming sequence. The write buffer for all devices in the PIC18F2XJXX/4XJXX Family is 64 bytes. It can be mapped to any 64-byte block beginning at 000000h. The actual memory write sequence takes the contents of this buffer and programs the 64-byte block of code memory indicated by 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 internally timed. 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.

The code sequence to program a PIC18F2XJXX/4XJXX Family device is shown in Table 3-3. The flowchart shown in Figure 3-5 depicts the logic necessary to completely write a PIC18F2XJXX/4XJXX Family device. The timing diagram that details the Start Programming command and parameter P9 is shown in Figure 3-6.

> **Note 1:** The TBLPTR register must point to the same region when initiating the programming sequence as it did when the write buffers were loaded.

TABLE 3-3: 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.
To continue writing data, repeat Steps 2 through 4, where the Address Pointer is incremented by 2 at each iteration of the loop.		

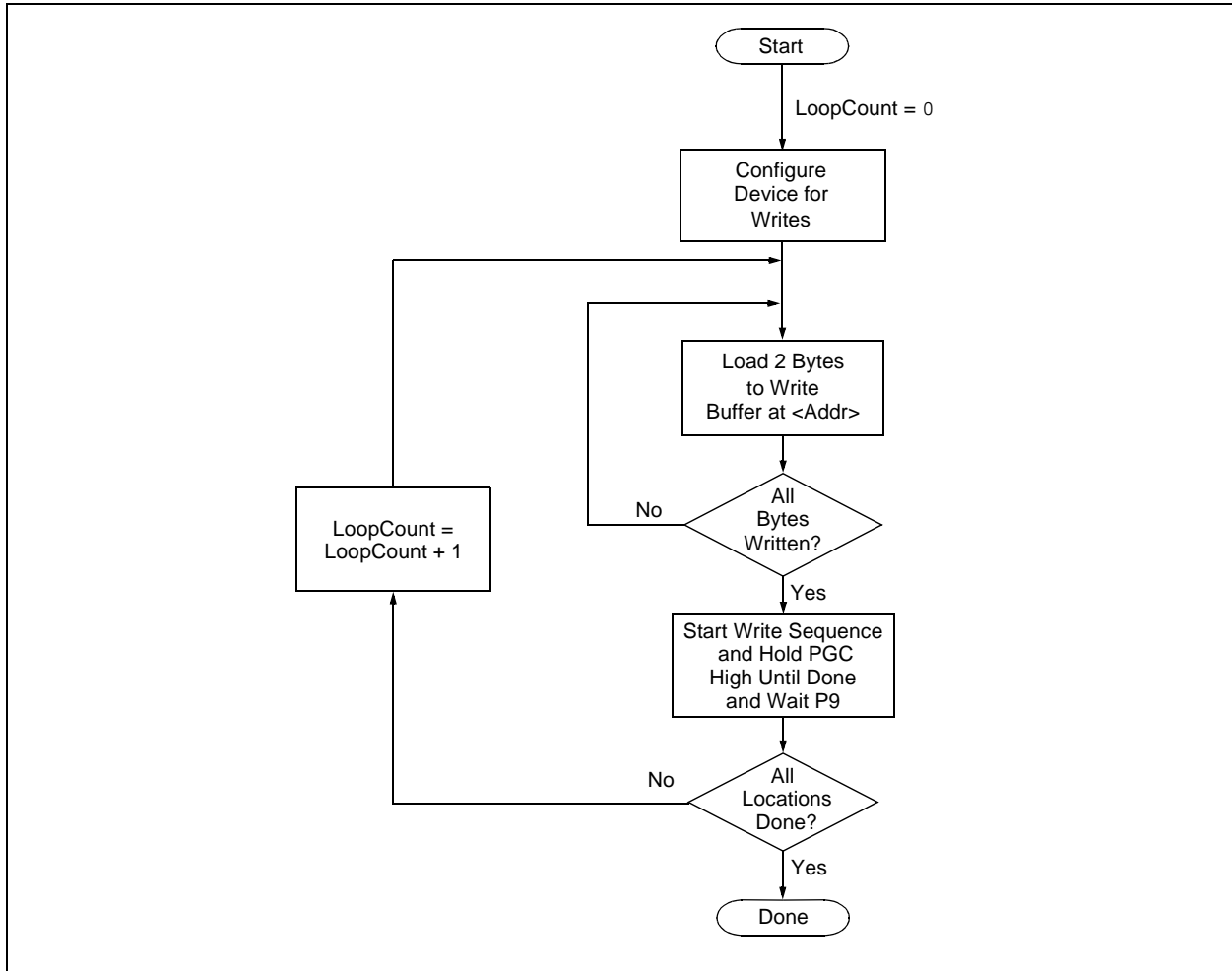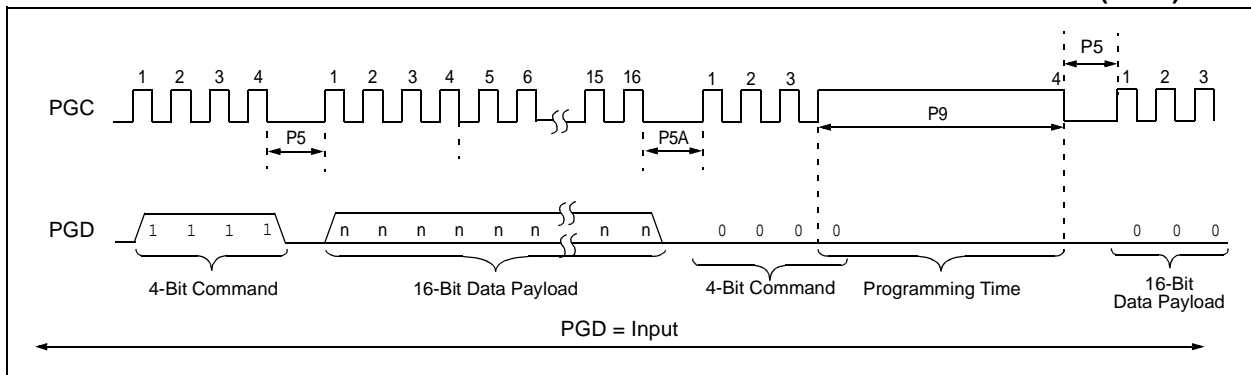**FIGURE 3-5:** **PROGRAM CODE MEMORY FLOW**

FIGURE 3-6: **TABLE WRITE AND START PROGRAMMING INSTRUCTION TIMING (1111)**

3.3 Endurance and Retention

To maintain the endurance specification of the Flash program memory cells, each byte should never be programmed more than once between erase operations. Before attempting to modify the contents of a specific byte of Flash memory a second time, an erase operation (either a Bulk Erase or a Row Erase which includes that byte) should be performed.
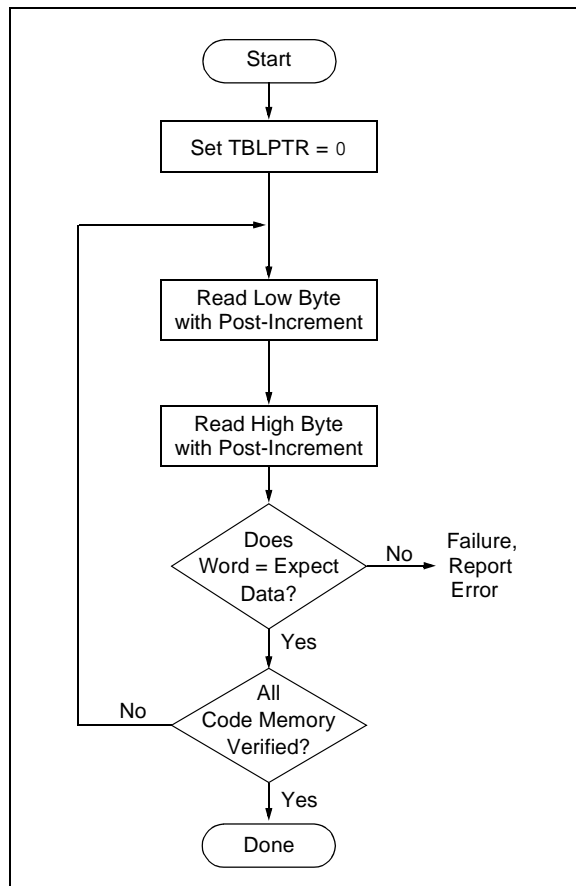
4.2 Verify Code Memory and Configuration Word

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 at the end 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 1:** 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 Words (and the CP0 bit) have been cleared.

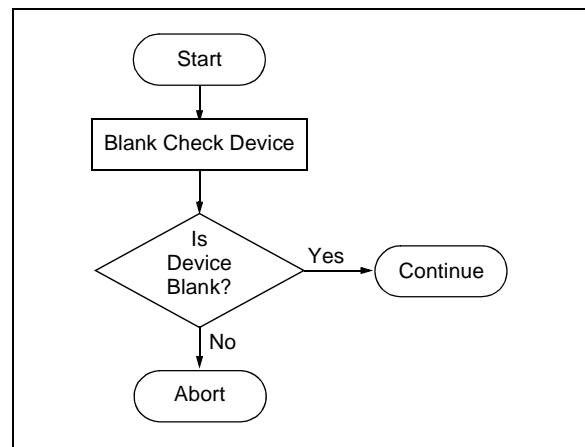**FIGURE 4-2: VERIFY CODE MEMORY FLOW**

4.3 Blank Check

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

A "blank" or "erased" memory cell will read as a '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.

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

FIGURE 4-3: BLANK CHECK FLOW

TABLE 5-3: **PIC18F45J10 FAMILY DEVICES: BIT DESCRIPTIONS**

Bit Name	Configuration Words	Description
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
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
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	Default Oscillator Select bit
		`1` = Clock designated by FOSC<1:0> is enabled as system clock when OSCCON<1:0> = `00`
		`0` = INTRC is enabled as system clock when OSCCON<1:0> = `00`

FOSC<1:0>	CONFIG2L	Primary Oscillator Select 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
WDTPS<3:0>	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
CCP2MX	CONFIG3H	CCP2 MUX bit `1` = CCP2 is multiplexed with RC1 `0` = CCP2 is multiplexed with RB3

TABLE 5-4: **PIC18F46J11 AND PIC18F46J50 FAMILY DEVICES: 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(1)
300000h	CONFIG1L	$\overline{\text{DEBUG}}$	XINST	STVREN	—	PLLDIV2(3)	PLLDIV1(3)	PLLDIV0(3)	WDTEN	111- 1111
300001h	CONFIG1H	—(2)	—(2)	—(2)	—(2)	—(4)	CP0	CPDIV1(3)	CPDIV0(3)	---- 0111
300002h	CONFIG2L	IESO	FCMEN	—	LPT1OSC	T1DIG	FOSC2	FOSC1	FOSC0	11-1 1111
300003h	CONFIG2H	—(2)	—(2)	—(2)	—(2)	WDTPS3	WDTPS2	WDTPS1	WDTPS0	---- 1111
300004h	CONFIG3L	DSWDTPS3	DSWDTPS2	DSWDTPS1	DSWDTPS0	DSWDTEN	DSBOREN	RTCOSC	DSWDTOSC	1111 1111
300005h	CONFIG3H	—(2)	—(2)	—(2)	—(2)	MSSPMSK	—	—	IOL1WAY	---- 1--1
300006h	CONFIG4L	WPCFG	WPEND	WPFP5(5)	WPFP4(6)	WPFP3	WPFP2	WPFP1	WPFP0	1111 1111
300007h	CONFIG4H	—(2)	—(2)	—(2)	—(2)	—	—	—	WPDIS	---- ---1
3FFFFEh	DEVID1	DEV2	DEV1	DEV0	REV4	REV3	REV2	REV1	REV0	xxxx xxxx
3FFFFFh	DEVID2	DEV10	DEV9	DEV8	DEV7	DEV6	DEV5	DEV4	DEV3	0100 00xx

Legend: x = unknown, u = unchanged, - = unimplemented. Shaded cells are unimplemented, read as '0'.

Note 1: Values reflect the unprogrammed state as received from the factory and following Power-on Resets. In all other Reset states, the configuration bytes maintain their previously programmed states.

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: These bits are not implemented in PIC18F46J11 family devices.

4: This bit should always be maintained at '0'.

5: This bit is not available on 32K and 16K memory devices (X4J11, X4J50, X5J11, and X5J50 devices) and should always be maintained at '0' on those devices.

6: This bit is not available on 16K memory devices (X4J11 and X4J50 devices) and should always be maintained at '0' on those devices.

TABLE 5-5: **PIC18F46J11 AND PIC18F46J50 FAMILY DEVICES: 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	Enhanced 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
PLLDIV<2:0>(3)	CONFIG1L	PLL Input Divider bits Divider must be selected to provide a 4 MHz input into the 96 MHz PLL. 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)

Note 1: The Configuration bits can only be programmed indirectly by programming the Flash Configuration Word.

2: The Configuration bits are reset to '1' only on VDD Reset; it is reloaded with the programmed value at any device Reset.

3: These bits are not implemented in PIC18F46J11 family devices.

4: Once this bit is cleared, all the Configuration registers which reside in the last page are also protected. To disable code protection, perform an ICSP™ Bulk Erase operation.

TABLE 5-5:   PIC18F46J11 AND PIC18F46J50 FAMILY DEVICES: BIT DESCRIPTIONS (CONTINUED)

Bit Name	Configuration Words	Description
CP0[4]	CONFIG1H	Code Protection bit `1` = Program memory is not code-protected `0` = Program memory is code-protected
CPDIV<1:0>[3]	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[1,2]	Two-Speed Start-up (Internal/External Oscillator Switchover) Control bit `1` = Oscillator Switchover mode enabled `0` = Oscillator Switchover mode disabled
FCMEN	CONFIG2L[1,2]	Fail-Safe Clock Monitor Enable bit `1` = Fail-Safe Clock Monitor enabled `0` = Fail-Safe Clock Monitor disabled
LPT1OSC	CONFIG2L[1,2]	Low-Power Timer1 Oscillator Enable bit `1` = Timer1 oscillator configured for low-power operation `0` = Timer1 oscillator configured for higher-power operation
T1DIG	CONFIG2L[1,2]	Secondary Clock Source T1OSCEN Enforcement bit[1] `1` = Secondary oscillator clock source may be selected (OSCCON <1:0> = `01`) regardless of T1OSCEN state `0` = Secondary oscillator clock source may not be selected unless T1CON <3> = `1`
FOSC<2:0>	CONFIG2L[1,2]	Oscillator Selection bits `111` = EC+PLL (S/W controlled by PLLEN bit), CLKO on RA6 `110` = EC oscillator (PLL always disabled) with CLKO on RA6 `101` = HS+PLL (S/W controlled by PLLEN bit) `100` = HS oscillator (PLL always disabled) `011` = INTOSCPLLO, internal oscillator with PLL (S/W controlled by PLLEN bit), CLKO on RA6, port function on RA7 `010` = INTOSCPLL, internal oscillator with PLL (S/W controlled by PLLEN bit), port function on RA6 and RA7 `001` = INTOSCO, internal oscillator, INTOSC or INTRC (PLL always disabled), CLKO on RA6, port function on RA7 `000` = INTOSC, internal oscillator INTOSC or INTRC (PLL always disabled), port function on RA6 and RA7
WDTPS<3:0>	CONFIG2H[1,2]	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: The Configuration bits can only be programmed indirectly by programming the Flash Configuration Word.
   2: The Configuration bits are reset to '`1`' only on VDD Reset; it is reloaded with the programmed value at any device Reset.
   3: These bits are not implemented in PIC18F46J11 family devices.
   4: Once this bit is cleared, all the Configuration registers which reside in the last page are also protected. To disable code protection, perform an ICSP™ Bulk Erase operation.

TABLE 5-5:   PIC18F46J11 AND PIC18F46J50 FAMILY DEVICES: BIT DESCRIPTIONS (CONTINUED)

Bit Name	Configuration Words	Description
DSWTPS<3:0>	CONFIG3L	Deep Sleep Watchdog Timer Postscale Select bits The DSWDT prescaler is 32; this creates an approximate base time unit of 1 ms. `1111` = 1:2,147,483,648 (25.7 days) `1110` = 1:536,870,912 (6.4 days) `1101` = 1:134,217,728 (38.5 hours) `1100` = 1:33,554,432 (9.6 hours) `1011` = 1:8,388,608 (2.4 hours) `1010` = 1:2,097,152 (36 minutes) `1001` = 1:524,288 (9 minutes) `1000` = 1:131,072 (135 seconds) `0111` = 1:32,768 (34 seconds) `0110` = 1:8,192 (8.5 seconds) `0101` = 1:2,048 (2.1 seconds) `0100` = 1:512 (528 ms) `0011` = 1:128 (132 ms) `0010` = 1:32 (33 ms) `0001` = 1:8 (8.3 ms) `0000` = 1:2 (2.1 ms)
DSWDTEN	CONFIG3L	Deep Sleep Watchdog Timer Enable bit `1` = DSWDT enabled `0` = DSWDT disabled
DSBOREN	CONFIG3L	Deep Sleep BOR Enable bit `1` = BOR enabled in Deep Sleep `0` = BOR disabled in Deep Sleep (does not affect operation in non Deep Sleep modes)
RTCOSC	CONFIG3L	RTCC Reference Clock Select bit `1` = RTCC uses T1OSC/T1CKI as reference clock `0` = RTCC uses INTRC as reference clock
DSWDTOSC	CONFIG3L	DSWDT Reference Clock Select bit `1` = DSWDT uses INTRC as reference clock `0` = DSWDT uses T1OSC/T1CKI as reference clock
MSSPMSK[1,2]	CONFIG3H	MSSP 7-Bit Address Masking Mode Enable bit `1` = 7-Bit Address Masking mode enable `0` = 5-Bit Address Masking mode enable
IOL1WAY	CONFIG3H	IOLOCK Bit One-Way Set Enable bit `1` = The IOLOCK bit (PPSCON<0>) can be set once, provided the unlock sequence has been completed. Once set, the Peripheral Pin Select registers cannot be written to a second time. `0` = The IOLOCK bit (PPSCON<0>) can be set and cleared as needed, provided the unlock sequence has been completed.
WPCFG[4]	CONFIG4L	Write/Erase Protect Configuration Words Page bit (valid when WPDIS = `0`) `1` = Configuration Words page is not erase/write-protected unless WPEND and WPFP<5:0> settings include the Configuration Words page `0` = Configuration Words page is erase/write-protected, regardless of WPEND and WPFP<5:0> settings
WPEND	CONFIG4L	Write/Erase Protect Region Select bit (valid when WPDIS = `0`) `1` = Flash pages, WPFP<5:0> to Configuration Words page, are write/erase-protected `0` = Flash pages, 0 to WPFP<5:0> are write/erase-protected

Note 1: The Configuration bits can only be programmed indirectly by programming the Flash Configuration Word.
  2: The Configuration bits are reset to '`1`' only on VDD Reset; it is reloaded with the programmed value at any device Reset.
  3: These bits are not implemented in PIC18F46J11 family devices.
  4: Once this bit is cleared, all the Configuration registers which reside in the last page are also protected. To disable code protection, perform an ICSP™ Bulk Erase operation.

TABLE 5-5: **PIC18F46J11 AND PIC18F46J50 FAMILY DEVICES: BIT DESCRIPTIONS (CONTINUED)**

Bit Name	Configuration Words	Description
WPFP<5:0>	CONFIG4L	Write/Erase Protect Page Start/End Location bits Used with WPEND bit to define which pages in Flash will be write/erase-protected.
WPDIS	CONFIG4H	Write Protect Disable bit 1 = WPFP<5:0>, WPEND and WPCFG bits ignored; all Flash memory may be erased or written 0 = WPFP<5:0>, WPEND and WPCFG bits enabled; write/erase-protect active for the selected region(s)
DEV<2:0>	DEVID1	Device ID bits Used with the DEV<10:3> bits in the Device ID Register 2 to identify the part number.
REV<4:0>	DEVID1	Revision ID bits Indicate the device revision.
DEV<10:3>	DEVID2	Device ID bits Used with the DEV<2:0> bits in the Device ID Register 1 to identify the part number.

Note 1: The Configuration bits can only be programmed indirectly by programming the Flash Configuration Word.
2: The Configuration bits are reset to '1' only on V_{DD} Reset; it is reloaded with the programmed value at any device Reset.
3: These bits are not implemented in PIC18F46J11 family devices.
4: Once this bit is cleared, all the Configuration registers which reside in the last page are also protected. To disable code protection, perform an ICSP™ Bulk Erase operation.

TABLE 5-6: **PIC18F47J13 AND PIC18F47J53 FAMILY DEVICES: 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[1]
300000h	CONFIG1L	DEBUG	XINST	STVREN	CFGPLLEN	PLLDIV2	PLLDIV1	PLLDIV0	WDTEN	111- 1111
300001h	CONFIG1H	—[2]	—[2]	—[2]	—[2]	—[4]	CP0	CPDIV1[3]	CPDIV0[3]	---- 0111
300002h	CONFIG2L	IESO	FCMEN	CLKOEC	SOSCSEL1	SOSCSEL0	FOSC2	FOSC1	FOSC0	1111 1111
300003h	CONFIG2H	—[2]	—[2]	—[2]	—[2]	WDTPS3	WDTPS2	WDTPS1	WDTPS0	---- 1111
300004h	CONFIG3L	DSWDTPS3	DSWDTPS2	DSWDTPS1	DSWDTPS0	DSWDTEN	DSBOREN	RTCOSC	DSWDTOSC	1111 1111
300005h	CONFIG3H	—[2]	—[2]	—[2]	—[2]	MSSPMSK	PLLSEL	ADCSEL	IOL1WAY	---- 1111
300006h	CONFIG4L	WPCFG	WPFP6	WPFP5	WPFP4	WPFP3	WPFP2	WPFP1	WPFP0	1111 1111
300007h	CONFIG4H	—[2]	—[2]	—[2]	—[2]	LS48MHZ[3]	—	WPEND	WPDIS	---- 1-11
3FFFFEh	DEVID1	DEV2	DEV1	DEV0	REV4	REV3	REV2	REV1	REV0	xxxx xxxx
3FFFFFh	DEVID2	DEV10	DEV9	DEV8	DEV7	DEV6	DEV5	DEV4	DEV3	0101 10xx

Legend: x = unknown, u = unchanged, - = unimplemented. Shaded cells are unimplemented, read as '0'.
Note 1: Values reflect the unprogrammed state as received from the factory and following Power-on Resets. In all other Reset states, the configuration bytes maintain their previously programmed states.
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: These bits are not implemented in PIC18F47J13 family devices.
4: This bit should always be maintained at '0'.

TABLE 5-7: PIC18F47J13 AND PIC18F47J53 FAMILY DEVICES: BIT DESCRIPTIONS

Bit Name	Configuration Words	Description
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	Enhanced 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
CFGPLLEN	CONFIG1L	Enable PLL on Start-up bit `1 =` PLL enabled on start-up. Not recommended for low-voltage designs. `0 =` PLL disabled on start-up. Firmware may later enable PLL through OSCTUNE<6>.
PLLDIV<2:0>	CONFIG1L	96 MHz PLL Input Divider bits Divider must be selected to provide a 4 MHz input into the 96 MHz PLL. `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 the SWDTEN bit)
CP0[4]	CONFIG1H	Code Protection bit `1 =` Program memory is not code-protected `0 =` Program memory is code-protected
CPDIV<1:0>[3]	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[1,2]	Two-Speed Start-up (Internal/External Oscillator Switchover) Control bit `1 =` Oscillator Switchover mode enabled `0 =` Oscillator Switchover mode disabled
FCMEN	CONFIG2L[1,2]	Fail-Safe Clock Monitor Enable bit `1 =` Fail-Safe Clock Monitor enabled `0 =` Fail-Safe Clock Monitor disabled
CLKOEC	CONFIG2L	EC Mode Clock Output Enable bit `1 =` CLKO output signal active on the RA6 pin (EC mode only) `0 =` CLKO output disabled
SOSCSEL<1:0>	CONFIG2L	Secondary Oscillator Circuit Selection bits `11 =` High-power SOSC circuit selected `10 =` Digital Input mode (SCLKI) `01 =` Low-power SOSC circuit selected `00 =` Reserved

Note 1: The Configuration bits can only be programmed indirectly by programming the Flash Configuration Word.

2: The Configuration bits are reset to '`1`' only on V_{DD} Reset; it is reloaded with the programmed value at any device Reset.

3: These bits are not implemented in PIC18F47J13 family devices.

4: Once this bit is cleared, all the Configuration registers which reside in the last page are also protected. To disable code protection, perform an ICSP™ Bulk Erase operation.

5: Not implemented on PIC18F47J53 family devices.

5.1 Device ID Word

The Device ID Word for the PIC18F2XJXX/4XJXX Family 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 protection has been enabled. The process for reading the Device IDs is shown in Figure 5-1. A complete list of Device ID values for the PIC18F2XJXX/4XJXX Family is presented in Table 5-8.
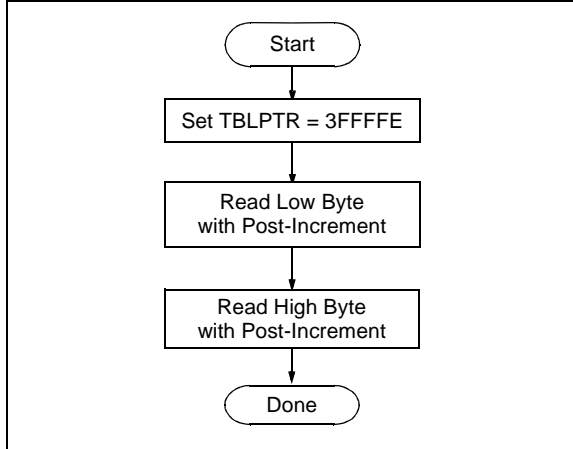
FIGURE 5-1: **READ DEVICE ID WORD FLOW**

TABLE 5-8: **DEVICE ID VALUE**

Device	Device ID Value	
	DEVID2	DEVID1
PIC18F24J10	1Dh	000x xxxx
PIC18F25J10	1Ch	000x xxxx
PIC18F44J10	1Dh	001x xxxx
PIC18F45J10	1Ch	001x xxxx
PIC18LF24J10	1Dh	010x xxxx
PIC18LF25J10	1Ch	010x xxxx
PIC18LF44J10	1Dh	011x xxxx
PIC18LF45J10	1Ch	011x xxxx
PIC18F25J11	4Dh	101x xxxx
PIC18F24J11	4Dh	100x xxxx
PIC18F26J11	4Dh	110x xxxx
PIC18F45J11	4Eh	000x xxxx
PIC18F44J11	4Dh	111x xxxx
PIC18F46J11	4Eh	001x xxxx
PIC18F24J50	4Ch	000x xxxx
PIC18F25J50	4Ch	001x xxxx
PIC18F26J50	4Ch	010x xxxx
PIC18F44J50	4Ch	011x xxxx
PIC18F45J50	4Ch	100x xxxx

TABLE 5-8: **DEVICE ID VALUE (CONTINUED)**

Device	Device ID Value	
	DEVID2	DEVID1
PIC18F46J50	4Ch	101x xxxx
PIC18LF2450	4Ch	110x xxxx
PIC18LF25J50	4Ch	111x xxxx
PIC18LF26J50	4Dh	000x xxxx
PIC18LF44J50	4Dh	001x xxxx
PIC18LF45J50	4Dh	010x xxxx
PIC18LF46J50	4Dh	011x xxxx
PIC18LF24J11	4Eh	010x xxxx
PIC18LF25J11	4Eh	011x xxxx
PIC18LF26J11	4Eh	100x xxxx
PIC18LF44J11	4Eh	101x xxxx
PIC18LF45J11	4Eh	110x xxxx
PIC18LF46J11	4Eh	111x xxxx
PIC18F26J13	59h	001x xxxx
PIC18F27J13	59h	011x xxxx
PIC18F46J13	59h	101x xxxx
PIC18F47J13	59h	111x xxxx
PIC18LF26J13	5Bh	001x xxxx
PIC18LF27J13	5Bh	011x xxxx
PIC18LF46J13	5Bh	101x xxxx
PIC18LF47J13	5Bh	111x xxxx
PIC18F26J53	58h	001x xxxx
PIC18F27J53	58h	011x xxxx
PIC18F46J53	58h	101x xxxx
PIC18F47J53	58h	111x xxxx
PIC18LF26J53	5Ah	001x xxxx
PIC18LF27J53	5Ah	011x xxxx
PIC18LF46J53	5Ah	101x xxxx
PIC18LF47J53	5Ah	111x xxxx

5.2 Checksum Computation

The checksum is calculated by summing the contents of all code memory locations and the device Configuration Words, appropriately masked. The Least Significant 16 bits of this sum are the checksum.

The checksum calculation differs depending on whether or not code protection is enabled. 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 Words can always be read.

PIC18F2XJXX/4XJXX FAMILY

Table 5-9 describes how to calculate the checksum for
each device.

TABLE 5-9:   CHECKSUM COMPUTATION

Device	Code Protection	Checksum
PIC18F24J10 PIC18F44J10	Off	SUM[000000:003FF7] + ([003FF8] & E1h) + ([003FF9] & 04h) + ([003FFA] & C7h) + ([003FFB] & 0Fh) + ([003FFD] & 01h)
	On	0000h
PIC18F24J11 PIC18F44J11	Off	SUM[000000:003FF7] + ([003FF8] & E1h) + ([003FF9] & FCh) + ([003FFA] & DFh) + ([003FFB] & FFh) + ([003FFC] & FFh) + ([003FFD] & F9h) + ([003FFE] & FFh) + ([003FFF] & F1h)
	On	0000h
PIC18F24J50 PIC18F44J50	Off	SUM[000000:003FF7] + ([003FF8] & EFh) + ([003FF9] & FFh) + ([003FFA] & DFh) + ([003FFB] & FFh) + ([003FFC] & FFh) + ([003FFD] & F9h) + ([003FFE] & FFh) + ([003FFF] & F1h)
	On	0000h
PIC18F25J10 PIC18F45J10	Off	SUM[000000:007FF7] + ([007FF8] & E1h) + ([007FF9] & 04h) + ([007FFA] & C7h) + ([007FFB] & 0Fh) + ([007FFD] & 01h)
	On	0000h
PIC18F25J11 PIC18F45J11	Off	SUM[000000:007FF7] + ([007FF8] & E1h) + ([007FF9] & FCh) + ([007FFA] & DFh) + ([007FFB] & FFh) + ([007FFC] & FFh) + ([007FFD] & F9h) + ([007FFE] & FFh) + ([007FFF] & F1h)
	On	0000h
PIC18F25J50 PIC18F45J50	Off	SUM[000000:007FF7] + ([007FF8] & EFh) + ([007FF9] & FFh) + ([007FFA] & DFh) + ([007FFB] & FFh) + ([007FFC] & FFh) + ([007FFD] & F9h) + ([007FFE] & FFh) + ([007FFF] & F1h)
	On	0000h
PIC18F26J11 PIC18F46J11	Off	SUM[000000:00FFF7] + ([00FFF8] & E1h) + ([00FFF9] & FCh) + ([00FFFA] & DFh) + ([00FFFB] & FFh) + ([00FFFC] & FFh) + ([00FFFD] & F9h) + ([00FFFE] & FFh) + ([00FFFF] & F1h)
	On	0000h
PIC18F26J50 PIC18F46J50	Off	SUM[000000:00FFF7] + ([00FFF8] & EFh) + ([00FFF9] & FFh) + ([00FFFA] & DFh) + ([00FFFB] & FFh) + ([00FFFC] & FFh) + ([00FFFD] & F9h) + ([00FFFE] & FFh) + ([00FFFF] & F1h)
	On	0000h
PIC18F26J13 PIC18F46J13	Off	SUM[000000:00FFF7] + ([00FFF8] & FFh) + ([00FFF9] & FCh) +([00FFFA] & FFh) + ([00FFFB] & FFh) + ([00FFFC] & FFh) + ([00FFFD] & FFh) + ([00FFFE] & BFh) + ([00FFFF] & F3h)
	On	0000h
PIC18F26J53 PIC18F46J53	Off	SUM[000000:00FFF7] + ([00FFF8] & FFh) + ([00FFF9] & FFh) +([00FFFA] & FFh) + ([00FFFB] & FFh) + ([00FFFC] & FFh) + ([00FFFD] & FBh) + ([00FFFE] & BFh) + ([00FFFF] & FBh)
	On	0000h
PIC18F27J13 PIC18F47J13	Off	SUM[000000:01FFF7] + ([01FFF8] & FFh) + ([01FFF9] & FCh) + ([01FFFA] & FFh) + ([01FFFB] & FFh) + ([01FFFC] & FFh) + ([01FFFD] & FFh) + ([01FFFE] & FFh) + ([01FFFF] & F3h)
	On	0000h
PIC18F27J53 PIC18F47J53	Off	SUM[000000:01FFF7] + ([01FFF8] & FFh) + ([01FFF9] & FFh) + ([01FFFA] & FFh) + ([01FFFB] & FFh) + ([01FFFC] & FFh) + ([01FFFD] & FBh) + ([01FFFE] & FFh) + ([01FFFF] & FBh)
	On	0000h

Legend: [a] = Value at address a; SUM[a:b] = Sum of locations a to b inclusive; + = Addition; & = Bitwise AND.
All addresses are hexadecimal.

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 ═