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.
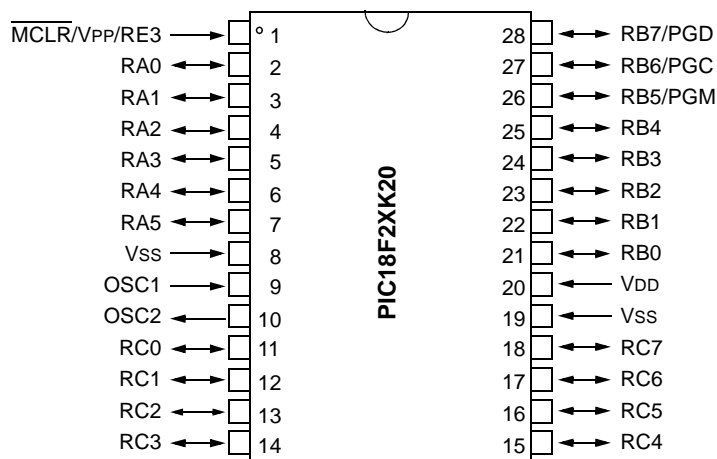
Applications of "**Embedded - Microcontrollers**"

Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	64MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, HLVD, POR, PWM, WDT
Number of I/O	24
Program Memory Size	8KB (4K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 11x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-UFQFN Exposed Pad
Supplier Device Package	28-UQFN (4x4)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f23k20-i-mv

PIC18F2XK20/4XK20

**FIGURE 2-1:** **28-PIN SDIP, SSOP AND SOIC PIN DIAGRAMS**

SDIP, SSOP, SOIC (300 MIL)

Left pin	#		#	Right pin
$\overline{MCLR}$/VPP/RE3	1		28	RB7/PGD
RA0	2		27	RB6/PGC
RA1	3		26	RB5/PGM
RA2	4		25	RB4
RA3	5		24	RB3
RA4	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

PIC18F2XK20

Note: The following devices are included in 28-pin SDIP, SSOP and SOIC parts: PIC18F23K20, PIC18F24K20, PIC18F25K20, PIC18F26K20.

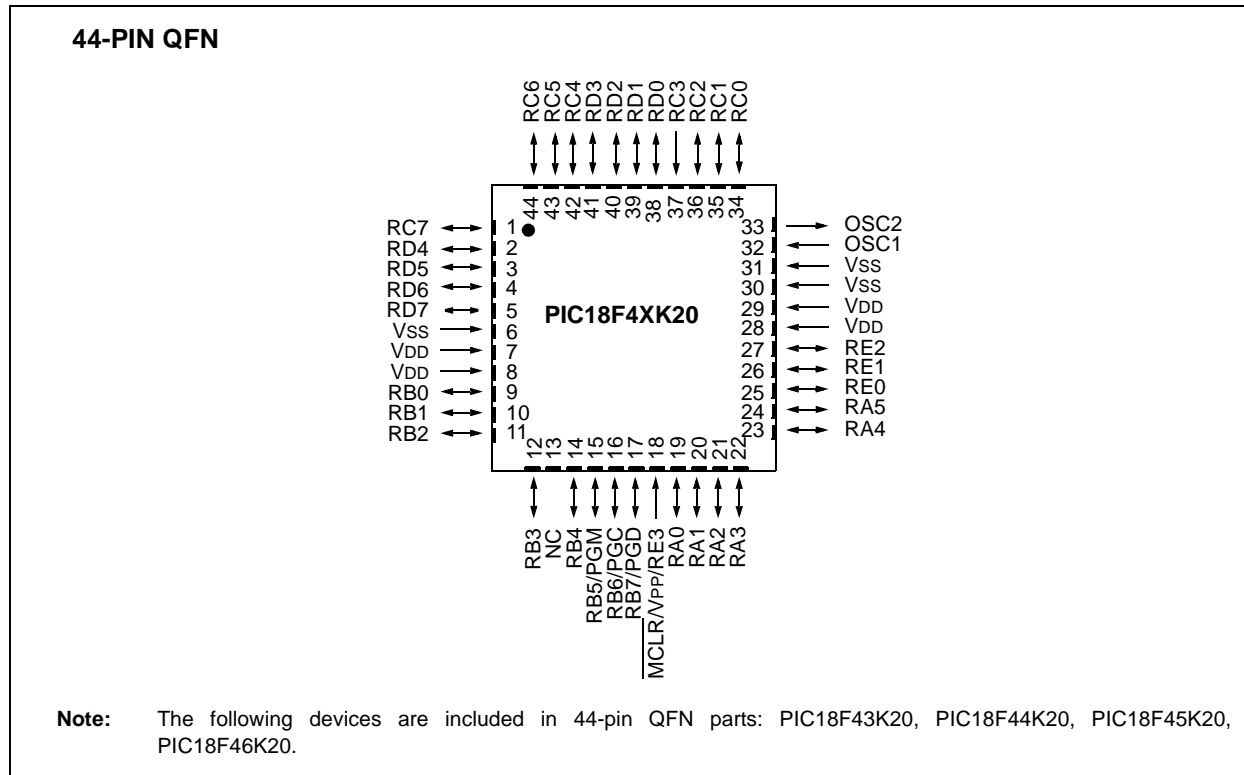**FIGURE 2-2:** **28-PIN QFN PIN DIAGRAMS**

28-Pin QFN
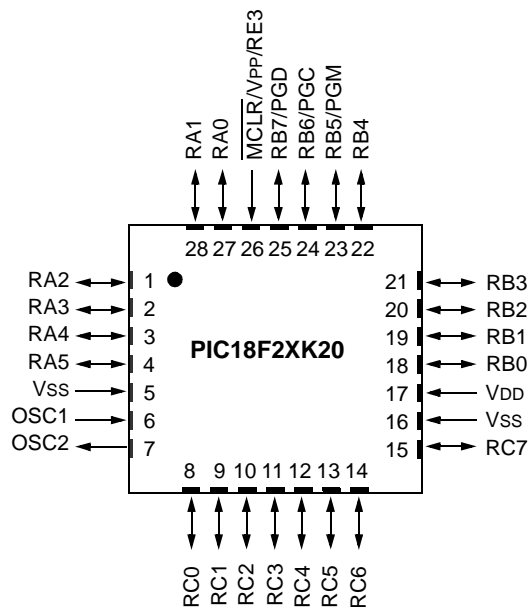
PIC18F2XK20

Note: The following devices are included in 28-pin QFN parts: PIC18F23K20, PIC18F24K20, PIC18F25K20, PIC18F26K20.

Advance Information

FIGURE 2-5: **44-PIN QFN PIN DIAGRAMS**

Note: The following devices are included in 44-pin QFN parts: PIC18F43K20, PIC18F44K20, PIC18F45K20, PIC18F46K20.
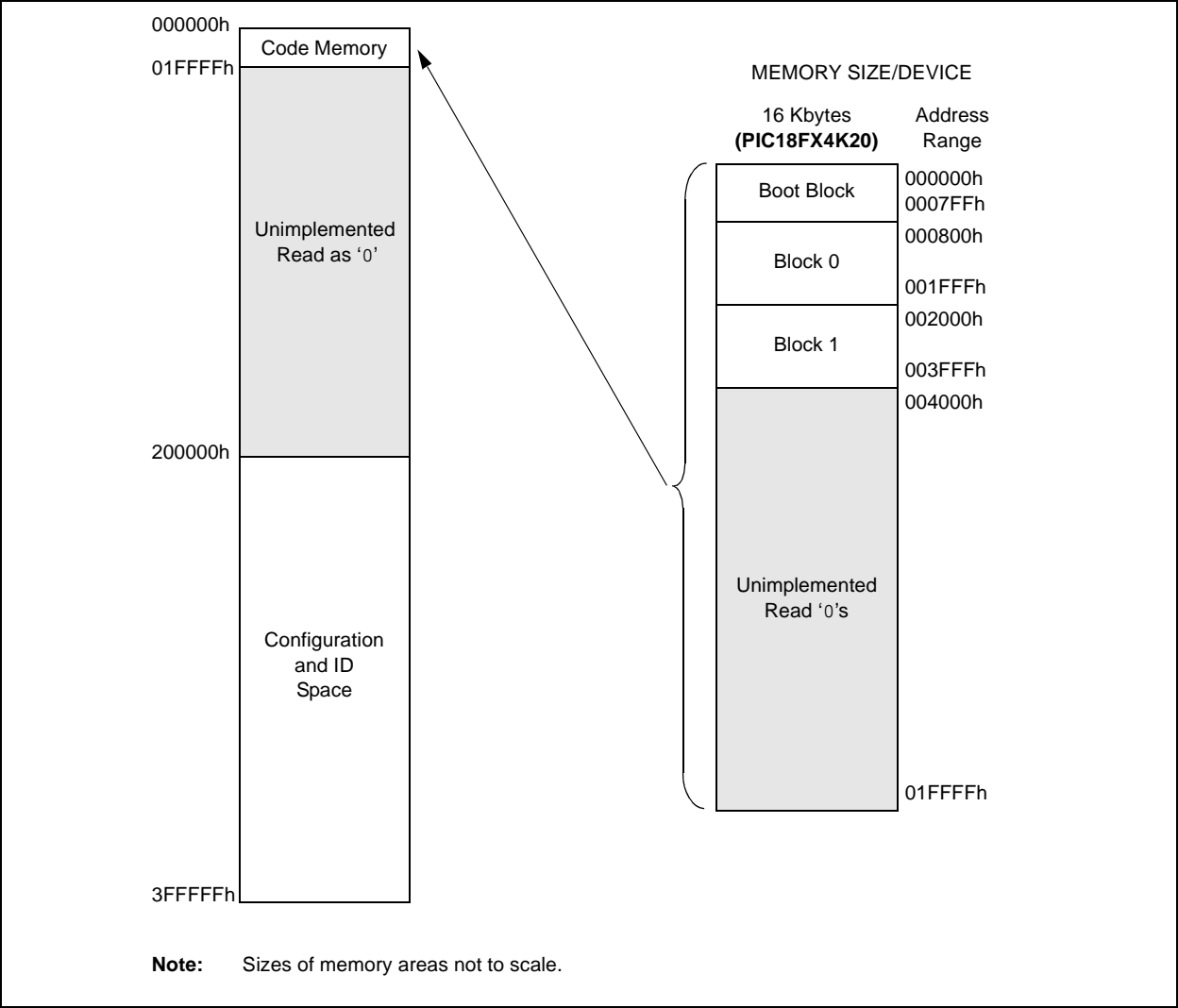
PIC18F2XK20/4XK20

For PIC18FX4K20 devices, the code memory space extends from 000000h to 003FFFh (16 Kbytes) in two 8-Kbyte blocks. Addresses 000000h through 0007FFh, however, define a "Boot Block" region that is treated separately from Block 0. All of these blocks define code protection boundaries within the code memory space.

TABLE 2-3: **IMPLEMENTATION OF CODE MEMORY**

Device	Code Memory Size (Bytes)
PIC18F24K20	000000h-003FFFh (16K)
PIC18F44K20	

FIGURE 2-7: **MEMORY MAP AND THE CODE MEMORY SPACE FOR PIC18FX4K20 DEVICES**

Note: Sizes of memory areas not to scale.

For PIC18FX5K20 devices, the code memory space extends from 000000h to 007FFFh (32 Kbytes) in four 8-Kbyte blocks. Addresses 000000h through 0007FFh, however, define a "Boot Block" region that is treated separately from Block 0. All of these blocks define code protection boundaries within the code memory space.

TABLE 2-4: **IMPLEMENTATION OF CODE MEMORY**

Device	Code Memory Size (Bytes)
PIC18F25K20	000000h-007FFFh (32K)
PIC18F45K20	

FIGURE 2-8: **MEMORY MAP AND THE CODE MEMORY SPACE FOR PIC18FX5K20 DEVICES**

**Note:** Sizes of memory areas not to scale.
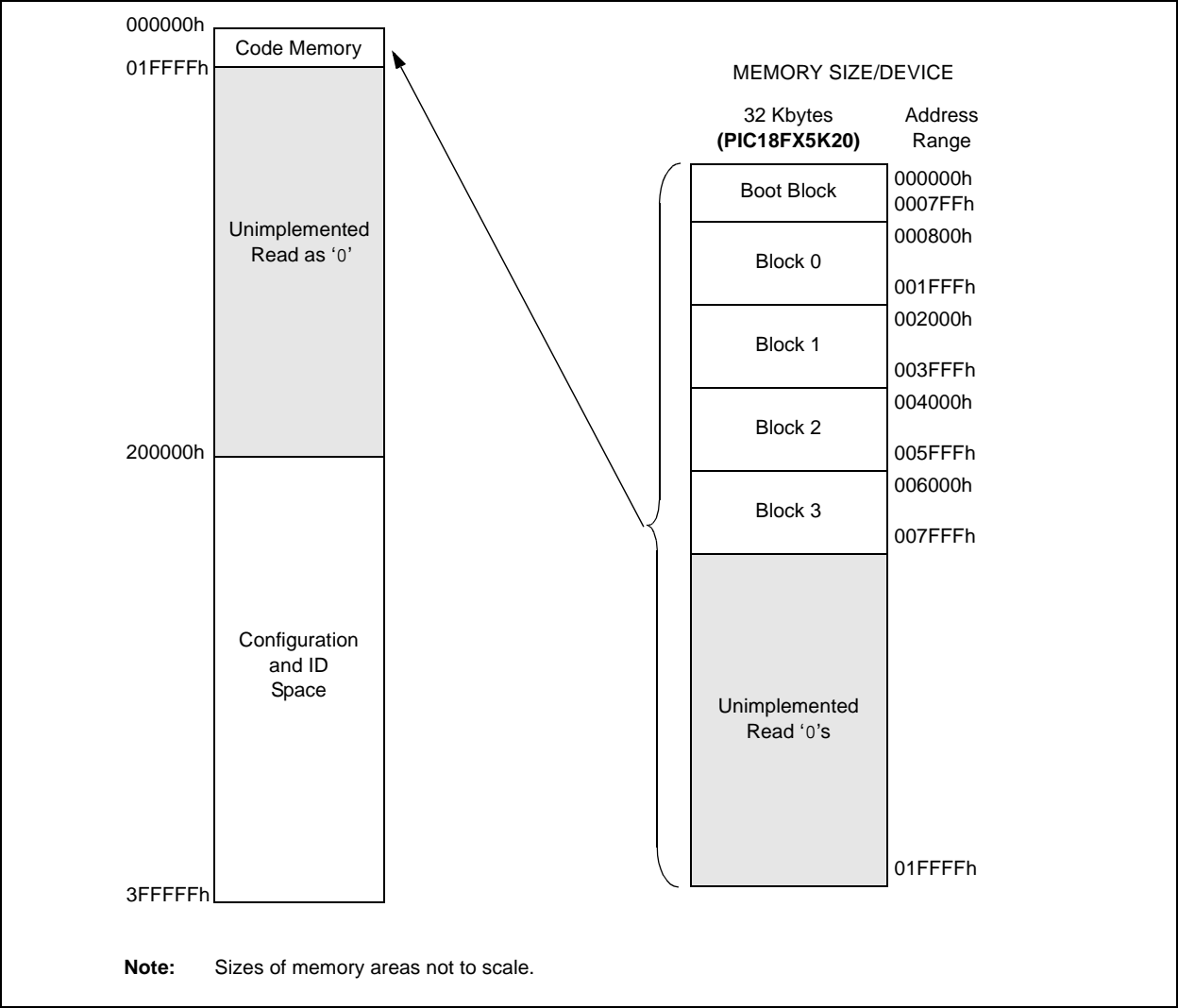
## 2.6 Entering and Exiting Low-Voltage ICSP Program/Verify Mode

When the LVP Configuration bit is '1' (see **Section 5.3 "Single-Supply ICSP Programming"**), the Low-Voltage ICSP mode is enabled. As shown in Figure 2-14, Low-Voltage ICSP Program/Verify mode is entered by holding PGC and PGD low, placing a logic high on PGM and then raising MCLR/VPP/RE3 to VIH. In this mode, the RB5/PGM pin is dedicated to the programming function and ceases to be a general purpose I/O pin. Figure 2-15 shows the exit sequence.

The sequence that enters the device into the Program/Verify mode places all unused I/Os in the high-impedance state.

FIGURE 2-14: ENTERING LOW-VOLTAGE PROGRAM/VERIFY MODE
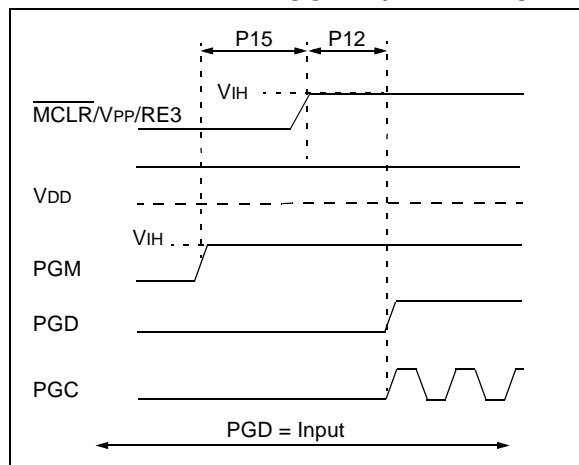
FIGURE 2-15: EXITING LOW-VOLTAGE PROGRAM/VERIFY MODE
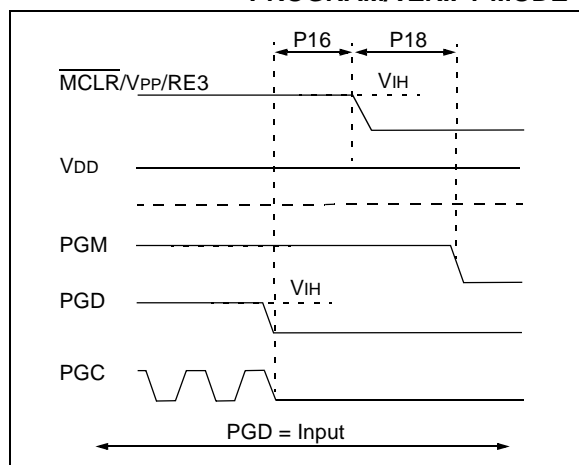
2.7 Serial Program/Verify Operation

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.

2.7.1 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 2-6.

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 2-7. The 4-bit command is shown Most Significant bit (MSb) first. The command operand, or "Data Payload", is shown <MSB><LSB>. Figure 2-16 demonstrates how to serially present a 20-bit command/operand to the device.

2.7.2 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 2-6: COMMANDS FOR PROGRAMMING

Description	4-Bit Command
Core Instruction (Shift in16-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

3.0 DEVICE PROGRAMMING

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

In all cases, except high-voltage ICSP Bulk Erase, the EECON1 register must be configured in order to operate on a particular memory region.

When using the EECON1 register to act on code memory, the EEPGD bit must be set (EECON1<7> = 1) and the CFGS bit must be cleared (EECON1<6> = 0). The WREN bit must be set (EECON1<2> = 1) to enable writes of any sort (e.g., erases) and this must be done prior to initiating a write sequence. The FREE bit must be set (EECON1<4> = 1) in order to erase the program space being pointed to by the Table Pointer. The erase or write sequence is initiated by setting the WR bit (EECON1<1> = 1). It is strongly recommended that the WREN bit only be set immediately prior to a program or erase.

3.1 ICSP Erase

3.1.1 HIGH-VOLTAGE ICSP BULK ERASE

Erasing code or data EEPROM is accomplished by configuring two Bulk Erase Control registers located at 3C0004h and 3C0005h. Code memory may be erased portions at a time, or the user may erase the entire device in one action. Bulk Erase operations will also clear any code-protect settings associated with the memory block erased. Erase options are detailed in Table 3-1. If data EEPROM is code-protected (CPD = 0), the user must request an erase of data EEPROM (e.g., 0084h as shown in Table 3-1).

TABLE 3-1:   BULK ERASE OPTIONS

Description	Data (3C0005h:3C0004h)
Chip Erase	0F8Fh
Erase User ID	0088h
Erase Data EEPROM	0084h
Erase Boot Block	0081h
Erase Config Bits	0082h
Erase Code EEPROM Block 0	0180h
Erase Code EEPROM Block 1	0280h
Erase Code EEPROM Block 2	0480h
Erase Code EEPROM Block 3	0880h

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.

The code sequence to erase the entire device is shown in Table 3-2 and the flowchart is shown in Figure 3-1.

> **Note:** A Bulk Erase is the only way to reprogram code-protect bits from an "on" state to an "off" state.

**TABLE 3-2:   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	0F 0F	Write 0Fh 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	8F 8F	Write 8F8Fh 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
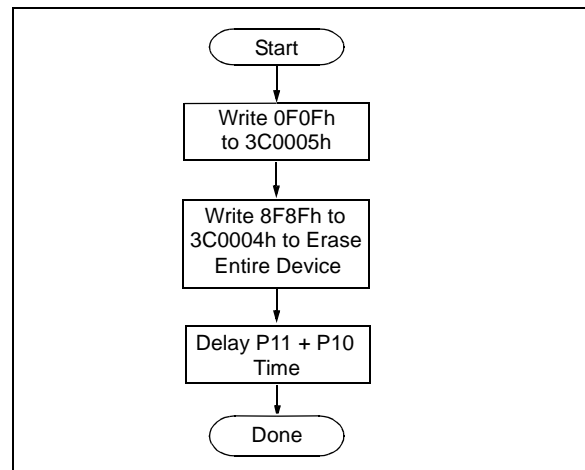
PIC18F2XK20/4XK20

FIGURE 3-2: **BULK ERASE TIMING DIAGRAM**
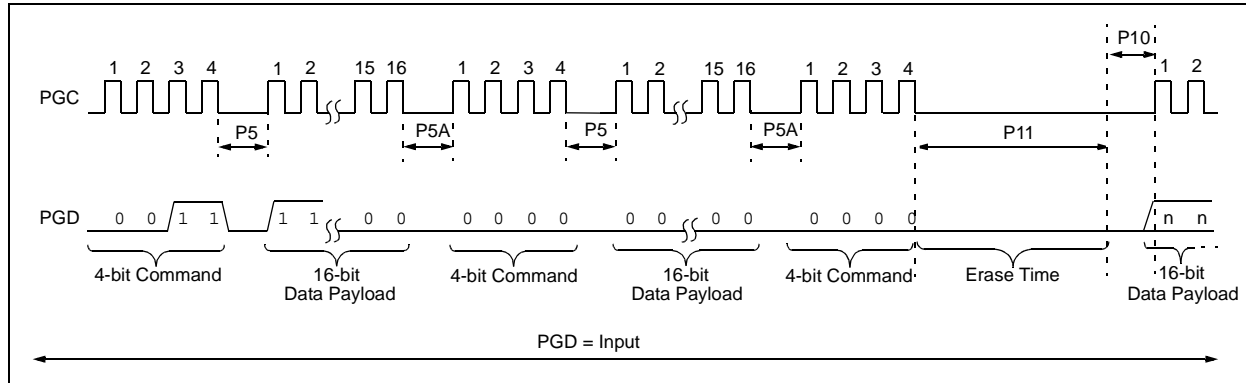
3.1.2   LOW-VOLTAGE ICSP BULK ERASE

When using low-voltage ICSP, the part must be supplied by the voltage specified in parameter D111 if a Bulk Erase is to be executed. All other Bulk Erase details as described above apply.

If it is determined that a program memory erase must be performed at a supply voltage below the Bulk Erase limit, refer to the erase methodology described in **Section 3.1.3 "ICSP Row Erase"** and **Section 3.2.1 "Modifying Code Memory"**.

If it is determined that a data EEPROM erase must be performed at a supply voltage below the Bulk Erase limit, follow the methodology described in **Section 3.3 "Data EEPROM Programming"** and write '1's to the array.

3.1.3   ICSP ROW ERASE

Regardless of whether high or low-voltage ICSP is used, it is possible to erase one row (64 bytes of data), provided the block is not code or write-protected. Rows are located at static boundaries beginning at program memory address 000000h, extending to the internal program memory limit (see **Section 2.3 "Memory Maps"**).

The Row Erase duration is self-timed. After the WR bit in EECON1 is set, two NOPs are issued. Erase starts upon the 4th PGC of the second NOP. It ends when the WR bit is cleared by hardware.

The code sequence to Row Erase a PIC18F2XK20/ 4XK20 device is shown in Table 3-3. The flowchart shown in Figure 3-3 depicts the logic necessary to completely erase a PIC18F2XK20/4XK20 device. The timing diagram for Row Erase is identical to the data EEPROM write timing shown in Figure 3-7.

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

Advance Information

PIC18F2XK20/4XK20

FIGURE 3-3: **SINGLE ROW ERASE CODE MEMORY FLOW**

Advance Information
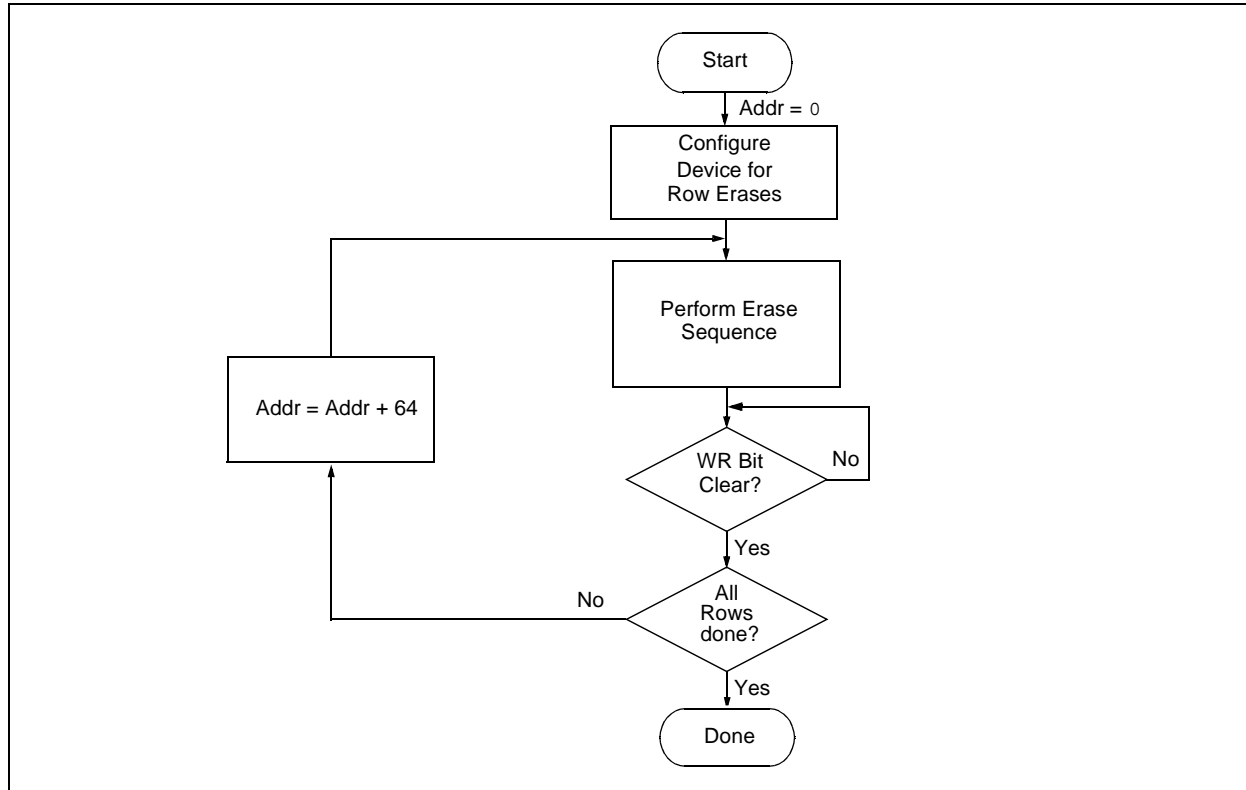
3.2.1 MODIFYING CODE MEMORY

The previous programming example assumed that the device has been Bulk Erased prior to programming (see **Section 3.1.1 "High-Voltage ICSP Bulk Erase"**). It may be the case, however, that the user wishes to modify only a section of an already programmed device.

The appropriate number of bytes required for the erase buffer must be read out of code memory (as described in **Section 4.2 "Verify Code Memory and ID Locations"**) and buffered. Modifications can be made on this buffer. Then, the block of code memory that was read out must be erased and rewritten with the modified data.

The WREN bit must be set if the WR bit in EECON1 is used to initiate a write sequence.

TABLE 3-6:   MODIFYING CODE MEMORY

4-bit Command	Data Payload	Core Instruction
Step 1: Direct access to code memory.		
0000	8E A6	BSF EECON1, EEPGD
0000	9C A6	BCF EECON1, CFGS
Step 2: Read code memory into buffer (**Section 4.1 "Read Code Memory, ID Locations and Configuration Bits"**).		
Step 3: Set the Table Pointer for the block to be erased.		
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
Step 4: Enable memory writes and setup an erase.		
0000	84 A6	BSF EECON1, WREN
0000	88 A6	BSF EECON1, FREE
Step 5: Initiate erase.		
0000	88 A6	BSF EECON1, FREE
0000	82 A6	BSF EECON1, WR
0000	00 00	NOP
0000	00 00	NOP Erase starts on the 4th clock of this instruction
Step 6: Poll WR bit. Repeat until bit is clear.		
0000	50 A6	MOVF EECON1, W, 0
0000	6E F5	MOVWF TABLAT
0000	00 00	NOP
0000	<MSB><LSB>	Shift out data[1]
Step 7: 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
•	•	Write 2 bytes and start programming.
1111	<MSB><LSB>	NOP - hold PGC high for time P9 and low for time P10.
0000	00 00	
To continue modifying data, repeat Steps 2 through 6, where the Address Pointer is incremented by the appropriate number of bytes (see Table 3-4) at each iteration of the loop. The write cycle must be repeated enough times to completely rewrite the contents of the erase buffer.		
Step 8: Disable writes.		
0000	94 A6	BCF EECON1, WREN

3.3 Data EEPROM Programming

Data EEPROM is accessed one byte at a time via an Address Pointer (register pair EEADRH:EEADR) and a data latch (EEDATA). Data EEPROM is written by loading EEADRH:EEADR with the desired memory location, EEDATA with the data to be written and initiating a memory write by appropriately configuring the EECON1 register. A byte write automatically erases the location and writes the new data (erase-before-write).

When using the EECON1 register to perform a data EEPROM write, both the EEPGD and CFGS bits must be cleared (EECON1<7:6> = 00). The WREN bit must be set (EECON1<2> = 1) to enable writes of any sort and this must be done prior to initiating a write sequence. The write sequence is initiated by setting the WR bit (EECON1<1> = 1).

The write begins on the falling edge of the 24th PGC after the WR bit is set. It ends when the WR bit is cleared by hardware.

After the programming sequence terminates, PGC must be held low for the time specified by parameter P10 to allow high-voltage discharge of the memory array.
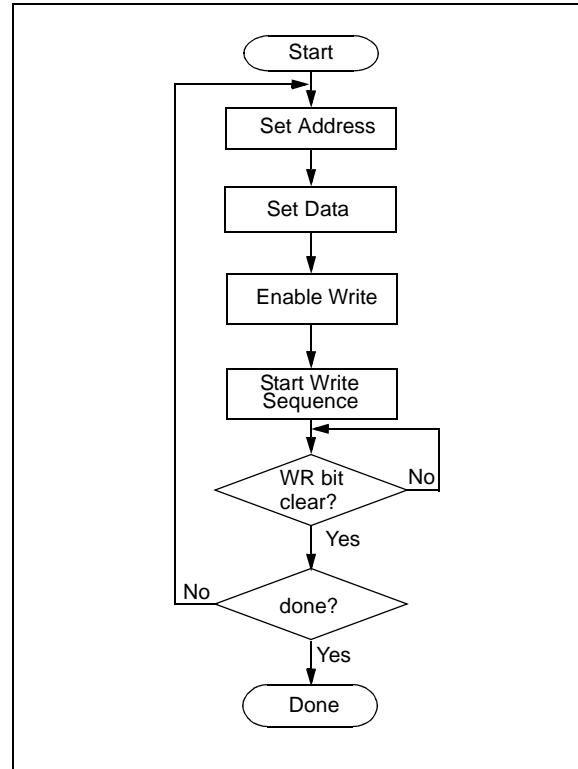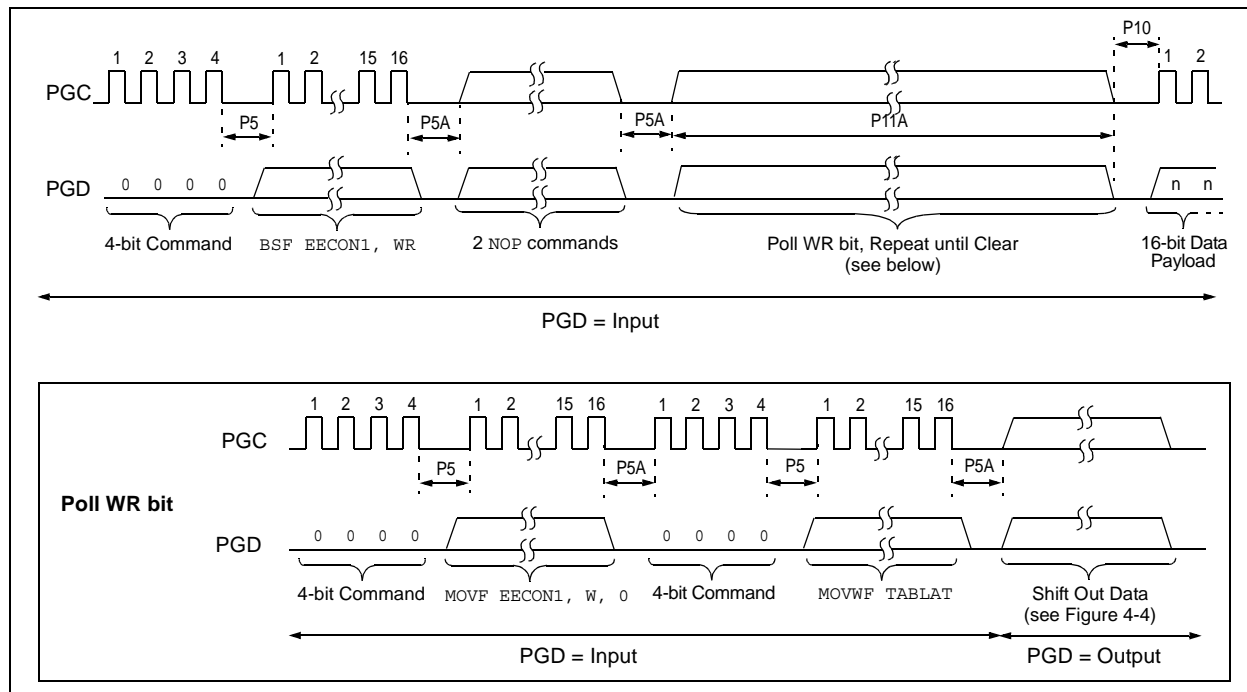
FIGURE 3-6:   PROGRAM DATA FLOW

FIGURE 3-7:   DATA EEPROM WRITE TIMING DIAGRAM

PIC18F2XK20/4XK20

4.0 READING THE DEVICE

4.1 Read Code Memory, ID Locations and Configuration Bits

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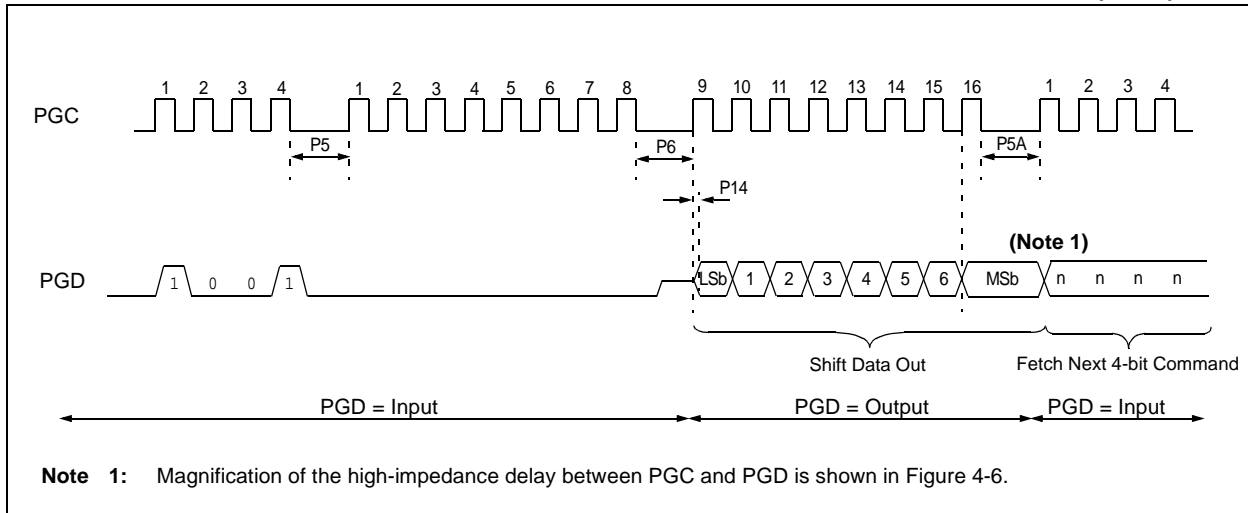
This technique will work to read any memory in the 000000h to 3FFFFFh address space, so it also applies to the reading of the ID and Configuration registers.

> **Note:** When table read protection is enabled, the first read access to a protected block should be discarded and the read repeated to retrieve valid data. Subsequent reads of the same block can be performed normally.

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 DIAGRAM (1001)

Note 1: Magnification of the high-impedance delay between PGC and PGD is shown in Figure 4-6.

Advance Information
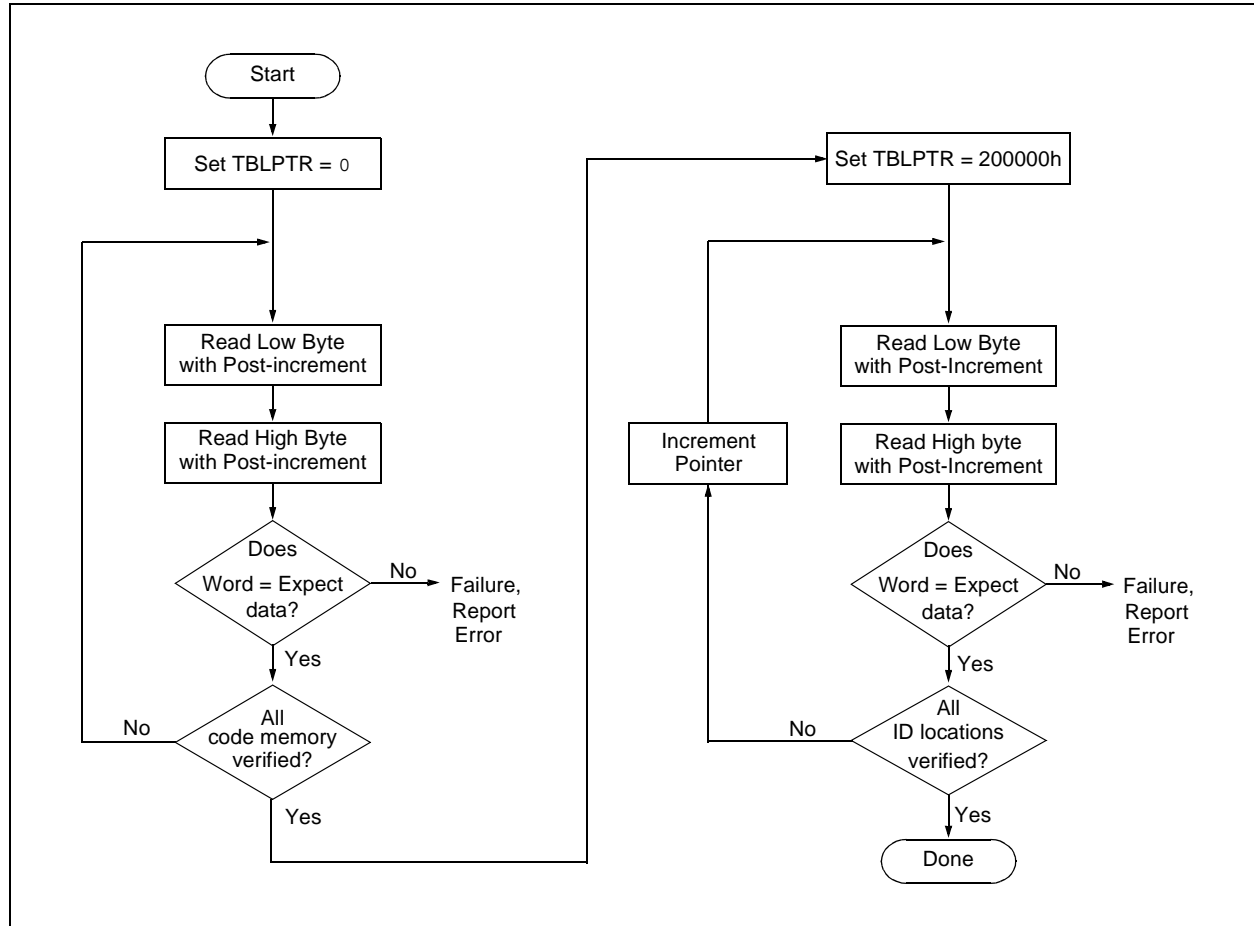
PIC18F2XK20/4XK20

4.2 Verify Code Memory and ID Locations

The verify step involves reading back the code memory space and comparing it against the copy held in the programmer's buffer. 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, ID Locations and Configuration Bits"** for implementation details of reading code memory.

The Table Pointer must be manually set to 200000h (base address of the ID locations) once the code memory has been verified. The post-increment feature of the table read 4-bit command can not be used to increment the Table Pointer beyond the code memory space. In a 64-Kbyte device, for example, a post-increment read of address FFFFh will wrap the Table Pointer back to 000000h, rather than point to unimplemented address 010000h.

FIGURE 4-2: VERIFY CODE MEMORY FLOW

© 2009 Microchip Technology Inc.   **Advance Information**   DS41297F-page 25

4.3 Verify Configuration Bits

A configuration address may be read and output on PGD via the 4-bit command, '1001'. Configuration data is read and written in a byte-wise fashion, so it is not necessary to merge two bytes into a word prior to a compare. The result may then be immediately compared to the appropriate configuration data in the programmer's memory for verification. Refer to **Section 4.1 "Read Code Memory, ID Locations and Configuration Bits"** for implementation details of reading configuration data.

4.4 Read Data EEPROM Memory

Data EEPROM is accessed one byte at a time via an Address Pointer (register pair EEADRH:EEADR) and a data latch (EEDATA). Data EEPROM is read by loading EEADRH:EEADR with the desired memory location and initiating a memory read by appropriately configuring the EECON1 register. The data will be loaded into EEDATA, where it may be serially output on PGD via the 4-bit command, '0010' (Shift Out Data Holding register). 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 4-4).

The command sequence to read a single byte of data is shown in Table 4-2.

FIGURE 4-3:   READ DATA EEPROM FLOW
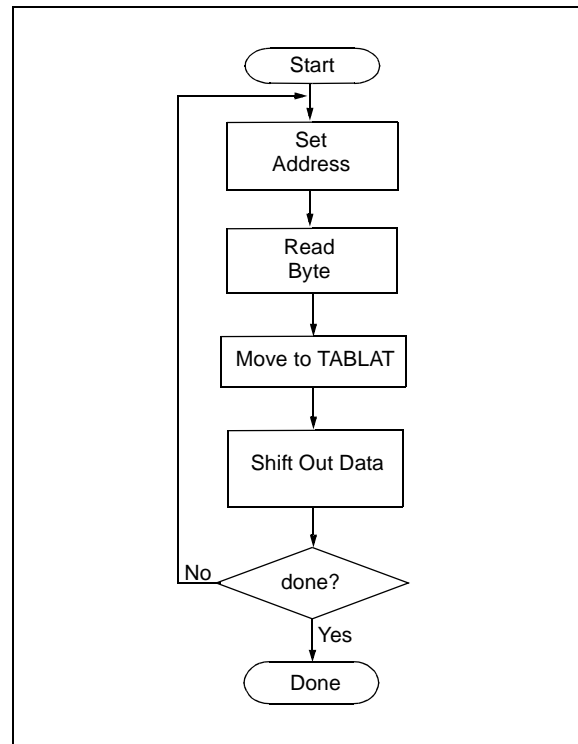
TABLE 4-2:   READ DATA EEPROM MEMORY

4-bit Command	Data Payload	Core Instruction
Step 1: Direct access to data EEPROM.		
0000	9E A6	BCF EECON1, EEPGD
0000	9C A6	BCF EECON1, CFGS
Step 2: Set the data EEPROM Address Pointer.		
0000	0E <Addr>	MOVLW <Addr>
0000	6E A9	MOVWF EEADR
0000	OE <AddrH>	MOVLW <AddrH>
0000	6E AA	MOVWF EEADRH
Step 3: Initiate a memory read.		
0000	80 A6	BSF EECON1, RD
Step 4: Load data into the Serial Data Holding register.		
0000	50 A8	MOVF EEDATA, W, 0
0000	6E F5	MOVWF TABLAT
0000	00 00	NOP
0010	<MSB><LSB>	Shift Out Data[1]

  Note   1:   The <LSB> is undefined. The <MSB> is the data.

TABLE 5-2: **DEVICE ID VALUE**

Device	Device ID Value	
	DEVID2	**DEVID1**
PIC18F23K20	20h	`111x xxxx`
PIC18F24K20	20h	`101x xxxx`
PIC18F25K20	20h	`011x xxxx`
PIC18F26K20	20h	`001x xxxx`
PIC18F43K20	20h	`110x xxxx`
PIC18F44K20	20h	`100x xxxx`
PIC18F45K20	20h	`010x xxxx`
PIC18F46K20	20h	`000x xxxx`

Note: The 'x's in DEVID1 contain the device revision code.

TABLE 5-3: **PIC18F2XK20/4XK20 BIT DESCRIPTIONS (CONTINUED)**

Bit Name	Configuration Words	Description
WDTEN	CONFIG2H	Watchdog Timer Enable bit `1` = WDT enabled `0` = WDT disabled (control is placed on SWDTEN bit)
MCLRE	CONFIG3H	$\overline{\text{MCLR}}$ Pin Enable bit `1` = $\overline{\text{MCLR}}$ pin enabled, RE3 input pin disabled `0` = RE3 input pin enabled, $\overline{\text{MCLR}}$ pin disabled
HFOFST	CONFIG3H	HFINTOSC Fast Start `1` = HFINTOSC output is not delayed `0` = HFINTOSC output is delayed until oscillator is stable (IOFS = `1`)
LPT1OSC	CONFIG3H	Low-Power Timer1 Oscillator Enable bit `1` = Timer1 configured for low-power operation `0` = Timer1 configured for higher power operation
PBADEN	CONFIG3H	PORTB A/D Enable bit `1` = PORTB A/D<4:0> pins are configured as analog input channels on Reset `0` = PORTB A/D<4:0> pins are configured as digital I/O on Reset
CCP2MX	CONFIG3H	CCP2 MUX bit `1` = CCP2 input/output is multiplexed with RC1 `0` = CCP2 input/output is multiplexed with RB3
DEBUG	CONFIG4L	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	CONFIG4L	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)
LVP	CONFIG4L	Low-Voltage Programming Enable bit `1` = Low-Voltage Programming enabled, RB5 is the PGM pin `0` = Low-Voltage Programming disabled, RB5 is an I/O pin
STVREN	CONFIG4L	Stack Overflow/Underflow Reset Enable bit `1` = Reset on stack overflow/underflow enabled `0` = Reset on stack overflow/underflow disabled

.

TABLE 5-4: **CHECKSUM COMPUTATION**

Device	Code-Protect	Checksum	Blank Value	0xAA at 0 and Max Address
PIC18FX3K20	None	SUM[0000:01FF]+SUM[0200:0FFF]+SUM[1000:1FFF]+ (CONFIG1L & 00h)+(CONFIG1H & CFh)+(CONFIG2L & 1Fh)+ (CONFIG2H & 1F)+(CONFIG3L & 00h)+(CONFIG3H & 8Fh)+ (CONFIG4L & C5h)+(CONFIG4H & 00h)+(CONFIG5L & 03h)+ (CONFIG5H & C0h)+(CONFIG6L & 03h)+(CONFIG6H & E0h)+ (CONFIG7L & 03h)+(CONFIG7H & 40h)	E33Eh	E294h
	Boot Block	SUM[0200:0FFF]+SUM[1000:1FFF]+(CONFIG1L & 00h)+ (CONFIG1H & CFh)+(CONFIG2L & 1Fh)+(CONFIG2H & 1F)+ (CONFIG3L & 00h)+(CONFIG3H & 8Fh)+(CONFIG4L & C5h)+ (CONFIG4H & 00h)+(CONFIG5L & 03h)+(CONFIG5H & C0h)+ (CONFIG6L & 03h)+(CONFIG6H & E0h)+(CONFIG7L & 03h)+ (CONFIG7H & 40h)+SUM_ID	E520h	E4C6h
	Boot/Block 0	SUM[1000:1FFF]+(CONFIG1L & 00h)+(CONFIG1H & CFh)+ (CONFIG2L & 1Fh)+(CONFIG2H & 1F)+(CONFIG3L & 00h)+ (CONFIG3H & 8Fh)+(CONFIG4L & C5h)+(CONFIG4H & 00h)+ (CONFIG5L & 03h)+(CONFIG5H & C0h)+(CONFIG6L & 03h)+ (CONFIG6H & E0h)+(CONFIG7L & 03h)+(CONFIG7H & 40h)+SUM_ID	F31Fh	F2C5h
	All	(CONFIG1L & 00h)+(CONFIG1H & CFh)+(CONFIG2L & 1Fh)+ (CONFIG2H & 1F)+(CONFIG3L & 00h)+(CONFIG3H & 8Fh)+ (CONFIG4L & C5h)+(CONFIG4H & 00h)+(CONFIG5L & 03h)+ (CONFIG5H & C0h)+(CONFIG6L & 03h)+(CONFIG6H & E0h)+ (CONFIG7L & 03h)+(CONFIG7H & 40h)+SUM_ID	031Dh	0318h
PIC18FX4K20	None	SUM[0000:07FF]+SUM[0800:1FFF]+SUM[2000:3FFF]+ (CONFIG1L & 00h)+(CONFIG1H & CFh)+(CONFIG2L & 1Fh)+ (CONFIG2H & 1F)+(CONFIG3L & 00h)+(CONFIG3H & 8Fh)+ (CONFIG4L & C5h)+(CONFIG4H & 00h)+(CONFIG5L & 03h)+ (CONFIG5H & C0h)+(CONFIG6L & 03h)+(CONFIG6H & E0h)+ (CONFIG7L & 03h)+(CONFIG7H & 40h)	C33Eh	C294h
	Boot Block	SUM[0800:1FFF]+SUM[2000:3FFF]+(CONFIG1L & 00h)+ (CONFIG1H & CFh)+(CONFIG2L & 1Fh)+(CONFIG2H & 1F)+ (CONFIG3L & 00h)+(CONFIG3H & 8Fh)+(CONFIG4L & C5h)+ (CONFIG4H & 00h)+(CONFIG5L & 03h)+(CONFIG5H & C0h)+ (CONFIG6L & 03h)+(CONFIG6H & E0h)+(CONFIG7L & 03h)+ (CONFIG7H & 40h)+SUM_ID	CB1Eh	CAC4h
	Boot/Block 0	SUM[2000:3FFF]+(CONFIG1L & 00h)+(CONFIG1H & CFh)+ (CONFIG2L & 1Fh)+(CONFIG2H & 1F)+(CONFIG3L & 00h)+ (CONFIG3H & 8Fh)+(CONFIG4L & C5h)+(CONFIG4H & 00h)+ (CONFIG5L & 03h)+(CONFIG5H & C0h)+(CONFIG6L & 03h)+ (CONFIG6H & E0h)+(CONFIG7L & 03h)+(CONFIG7H & 40h)+SUM_ID	E31Dh	E2C3h
	All	(CONFIG1L & 00h)+(CONFIG1H & CFh)+(CONFIG2L & 1Fh)+ (CONFIG2H & 1F)+(CONFIG3L & 00h)+(CONFIG3H & 8Fh)+ (CONFIG4L & C5h)+(CONFIG4H & 00h)+(CONFIG5L & 03h)+ (CONFIG5H & C0h)+(CONFIG6L & 03h)+(CONFIG6H & E0h)+ (CONFIG7L & 03h)+(CONFIG7H & 40h)+SUM_ID	031Bh	0316h

Legend:

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

PIC18F2XK20/4XK20

TABLE 5-4: **CHECKSUM COMPUTATION (CONTINUED)**

Device	Code-Protect	Checksum	Blank Value	0xAA at 0 and Max Address
PIC18FX5K20	None	SUM[0000:07FF]+SUM[0800:1FFF]+SUM[2000:3FFF]+ SUM[4000:5FFF]+SUM[6000:7FFF]+(CONFIG1L & 00h)+ (CONFIG1H & CFh)+(CONFIG2L & 1Fh)+(CONFIG2H & 1F)+ (CONFIG3L & 00h)+(CONFIG3H & 8Fh)+(CONFIG4L & C5h)+ (CONFIG4H & 00h)+(CONFIG5L & 0Fh)+(CONFIG5H & C0h)+ (CONFIG6L & 0Fh)+(CONFIG6H & E0h)+(CONFIG7L & 0Fh)+ (CONFIG7H & 40h)	8362h	82B8h
	Boot Block	SUM[0800:1FFF]+SUM[2000:3FFF]+SUM[4000:5FFF]+SUM[6000:7FFF]+ (CONFIG1L & 00h)+(CONFIG1H & CFh)+(CONFIG2L & 1Fh)+ (CONFIG2H & 1F)+(CONFIG3L & 00h)+(CONFIG3H & 8Fh)+ (CONFIG4L & C5h)+(CONFIG4H & 00h)+(CONFIG5L & 0Fh)+ (CONFIG5H & C0h)+(CONFIG6L & 0Fh)+(CONFIG6H & E0h)+ (CONFIG7L & 0Fh)+(CONFIG7H & 40h)+SUM_ID	8B35h	8AEAh
	Boot/ Block 0/ Block 1	SUM[4000:5FFF]+SUM[6000:7FFF]+(CONFIG1L & 00h)+ (CONFIG1H & CFh)+(CONFIG2L & 1Fh)+(CONFIG2H & 1F)+ (CONFIG3L & 00h)+(CONFIG3H & 8Fh)+(CONFIG4L & C5h)+ (CONFIG4H & 00h)+(CONFIG5L & 0Fh)+(CONFIG5H & C0h)+ (CONFIG6L & 0Fh)+(CONFIG6H & E0h)+(CONFIG7L & 0Fh)+ (CONFIG7H & 40h)+SUM_ID	C332h	C2E7h
	All	(CONFIG1L & 00h)+(CONFIG1H & CFh)+(CONFIG2L & 1Fh)+ (CONFIG2H & 1F)+(CONFIG3L & 00h)+(CONFIG3H & 8Fh)+ (CONFIG4L & C5h)+(CONFIG4H & 00h)+(CONFIG5L & 0Fh)+ (CONFIG5H & C0h)+(CONFIG6L & 0Fh)+(CONFIG6H & E0h)+ (CONFIG7L & 0Fh)+(CONFIG7H & 40h)+SUM_ID	0326h	0330h

Legend:

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

TABLE 5-4: CHECKSUM COMPUTATION (CONTINUED)

Device	Code-Protect	Checksum	Blank Value	0xAA at 0 and Max Address
PIC18FX6K20	None	SUM[0000:07FF]+SUM[0800:3FFF]+SUM[4000:7FFF]+ SUM[8000:BFFF]+SUM[C000:FFFF]+(CONFIG1L & 00h)+ (CONFIG1H & CFh)+(CONFIG2L & 1Fh)+(CONFIG2H & 1F)+ (CONFIG3L & 00h)+(CONFIG3H & 8Fh)+(CONFIG4L & C5h)+ (CONFIG4H & 00h)+(CONFIG5L & 0Fh)+(CONFIG5H & C0h)+ (CONFIG6L & 0Fh)+(CONFIG6H & E0h)+(CONFIG7L & 0Fh)+ (CONFIG7H & 40h)	0362h	02B8h
	Boot Block	SUM[0800:3FFF]+SUM[4000:7FFF]+SUM[8000:BFFF]+SUM[C000:FFFF]+ (CONFIG1L & 00h)+(CONFIG1H & CFh)+(CONFIG2L & 1Fh)+ (CONFIG2H & 1F)+(CONFIG3L & 00h)+(CONFIG3H & 8Fh)+ (CONFIG4L & C5h)+(CONFIG4H & 00h)+(CONFIG5L & 0Fh)+ (CONFIG5H & C0h)+(CONFIG6L & 0Fh)+(CONFIG6H & E0h)+ (CONFIG7L & 0Fh)+(CONFIG7H & 40h)+SUM_ID	0B2Dh	0AE2h
	Boot/ Block 0/ Block 1	SUM[3000:BFFF]+SUM[C000:FFFF]+(CONFIG1L & 00h)+ (CONFIG1H & CFh)+(CONFIG2L & 1Fh)+(CONFIG2H & 1F)+ (CONFIG3L & 00h)+(CONFIG3H & 8Fh)+(CONFIG4L & C5h)+ (CONFIG4H & 00h)+(CONFIG5L & 0Fh)+(CONFIG5H & C0h)+ (CONFIG6L & 0Fh)+(CONFIG6H & E0h)+(CONFIG7L & 0Fh)+ (CONFIG7H & 40h)+SUM_ID	832Ah	82DFh
	All	(CONFIG1L & 00h)+(CONFIG1H & CFh)+(CONFIG2L & 1Fh)+ (CONFIG2H & 1F)+(CONFIG3L & 00h)+(CONFIG3H & 8Fh)+ (CONFIG4L & C5h)+(CONFIG4H & 00h)+(CONFIG5L & 0Fh)+ (CONFIG5H & C0h)+(CONFIG6L & 0Fh)+(CONFIG6H & E0h)+ (CONFIG7L & 0Fh)+(CONFIG7H & 40h)+SUM_ID	031Eh	0328h

Legend:

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

PIC18F2XK20/4XK20

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

Standard Operating Conditions
Operating Temperature: 25°C is recommended

Param No.	Sym.	Characteristic	Min.	Max.	Units	Conditions
D110	V$_{IHH}$	High-Voltage Programming Voltage on MCLR/V$_{PP}$/RE3	V$_{DD}$ + 4.5	9	V	
D110A	V$_{IHL}$	Low-Voltage Programming Voltage on MCLR/V$_{PP}$/RE3	1.80	3.60	V	
D111	V$_{DD}$	Supply Voltage During Programming	1.80	3.60	V	Row Erase/Write
			2.7	3.60	V	Bulk Erase operations
D112	I$_{PP}$	Programming Current on MCLR/V$_{PP}$/RE3	—	300	µA	
D113	I$_{DDP}$	Supply Current During Programming	—	10	mA	
D031	V$_{IL}$	Input Low Voltage	V$_{SS}$	0.2 V$_{DD}$	V	
D041	V$_{IH}$	Input High Voltage	0.8 V$_{DD}$	V$_{DD}$	V	
D080	V$_{OL}$	Output Low Voltage	—	0.6	V	I$_{OL}$ = X.X mA @ 2.7V
D090	V$_{OH}$	Output High Voltage	V$_{DD}$ – 0.7	—	V	I$_{OH}$ = -Y.Y mA @ 2.7V
D012	C$_{IO}$	Capacitive Loading on I/O pin (PGD)	—	50	pF	To meet AC specifications
P1	T$_R$	MCLR/V$_{PP}$/RE3 Rise Time to enter Program/Verify mode	—	1.0	µs	**(Note 1)**
P2	T$_{PGC}$	Serial Clock (PGC) Period	100	—	ns	V$_{DD}$ = 3.6V
			1	—	µs	V$_{DD}$ = 1.8V
P2A	T$_{PGCL}$	Serial Clock (PGC) Low Time	40	—	ns	V$_{DD}$ = 3.6V
			400	—	ns	V$_{DD}$ = 1.8V
P2B	T$_{PGCH}$	Serial Clock (PGC) High Time	40	—	ns	V$_{DD}$ = 3.6V
			400	—	ns	V$_{DD}$ = 1.8V
P3	T$_{SET1}$	Input Data Setup Time to Serial Clock ↓	15	—	ns	
P4	T$_{HLD1}$	Input Data Hold Time from PGC ↓	15	—	ns	
P5	T$_{DLY1}$	Delay between 4-bit Command and Command Operand	40	—	ns	
P5A	T$_{DLY1A}$	Delay between 4-bit Command Operand and next 4-bit Command	40	—	ns	
P6	T$_{DLY2}$	Delay between Last PGC ↓ of Command Byte to First PGC ↑ of Read of Data Word	20	—	ns	
P9	T$_{DLY5}$	PGC High Time (minimum programming time)	1	—	ms	Externally Timed
P9A	T$_{DLY5A}$	PGC High Time	5		ms	Configuration Word programming time
P10	T$_{DLY6}$	PGC Low Time after Programming (high-voltage discharge time)	200	—	µs	
P11	T$_{DLY7}$	Delay to allow Self-Timed Data Write or Bulk Erase to occur	5	—	ms	
P11A	T$_{DRWT}$	Data Write Polling Time	4	—	ms	

Note 1: Do not allow excess time when transitioning MCLR between V$_{IL}$ and V$_{IHH}$; this can cause spurious program executions to occur. The maximum transition time is:
1 T$_{CY}$ + T$_{PWRT}$ (if enabled) + 1024 T$_{OSC}$ (for LP, HS, HS/PLL and XT modes only) + 2 ms (for HS/PLL mode only) + 1.5 µs (for EC mode only) where T$_{CY}$ is the instruction cycle time, T$_{PWRT}$ is the Power-up Timer period and T$_{OSC}$ is the oscillator period. For specific values, refer to the Electrical Characteristics section of the device data sheet for the particular device.

Advance Information

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

Standard Operating Conditions
Operating Temperature: 25°C is recommended

Param No.	Sym.	Characteristic	Min.	Max.	Units	Conditions
P12	T$_{HLD2}$	Input Data Hold Time from $\overline{MCLR}$/V$_{PP}$/RE3 ↑	2	—	μs	
P13	T$_{SET2}$	V$_{DD}$ ↑ Setup Time to $\overline{MCLR}$/V$_{PP}$/RE3 ↑	100	—	ns	
P14	T$_{VALID}$	Data Out Valid from PGC ↑	10	—	ns	
P15	T$_{SET3}$	PGM ↑ Setup Time to $\overline{MCLR}$/V$_{PP}$/RE3 ↑	2	—	μs	
P16	T$_{DLY8}$	Delay between Last PGC ↓ and $\overline{MCLR}$/V$_{PP}$/RE3 ↓	0	—	s	
P17	T$_{HLD3}$	$\overline{MCLR}$/V$_{PP}$/RE3 ↓ to V$_{DD}$ ↓	—	100	ns	
P18	T$_{HLD4}$	$\overline{MCLR}$/V$_{PP}$/RE3 ↓ to PGM ↓	0	—	s	
P19	T$_{HIZ}$	Delay from PGC ↑ to PGD High-Z	3	10	nS	
P20	T$_{PPDP}$	Hold time after V$_{PP}$ changes	5	—	μs	

Note 1: Do not allow excess time when transitioning $\overline{MCLR}$ between V$_{IL}$ and V$_{IHH}$; this can cause spurious program executions to occur. The maximum transition time is:
1 T$_{CY}$ + T$_{PWRT}$ (if enabled) + 1024 T$_{OSC}$ (for LP, HS, HS/PLL and XT modes only) + 2 ms (for HS/PLL mode only) + 1.5 μs (for EC mode only) where T$_{CY}$ is the instruction cycle time, T$_{PWRT}$ is the Power-up Timer period and T$_{OSC}$ is the oscillator period. For specific values, refer to the Electrical Characteristics section of the device data sheet for the particular device.