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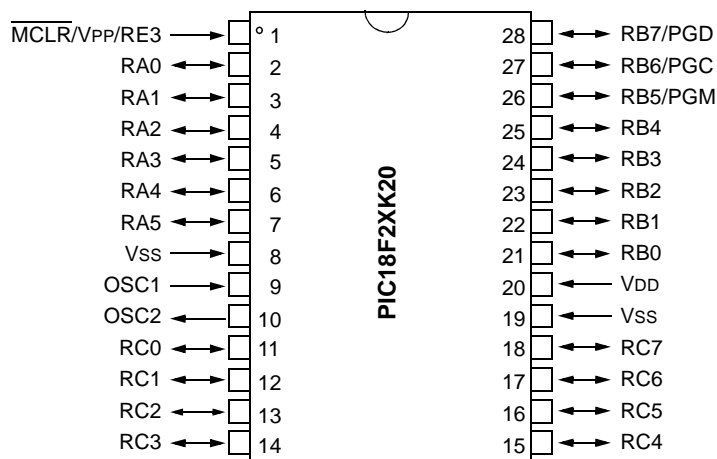
Applications of "**Embedded - Microcontrollers**"

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-SSOP (0.209", 5.30mm Width)
Supplier Device Package	28-SSOP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f23k20t-i-ss

PIC18F2XK20/4XK20

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

SDIP, SSOP, SOIC (300 MIL)

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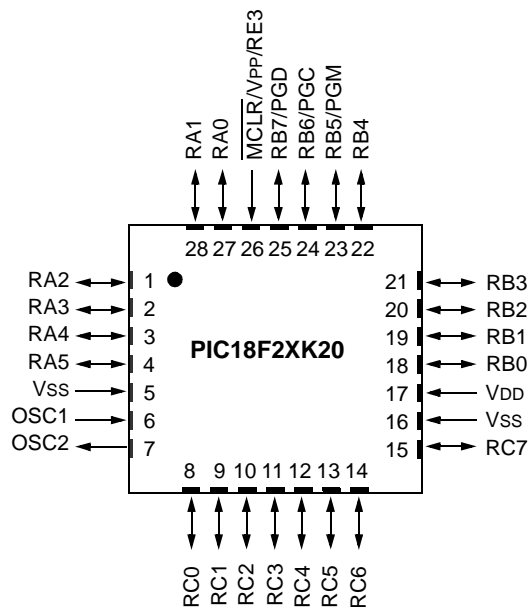
Note: The following devices are included in 28-pin QFN parts: PIC18F23K20, PIC18F24K20, PIC18F25K20, PIC18F26K20.
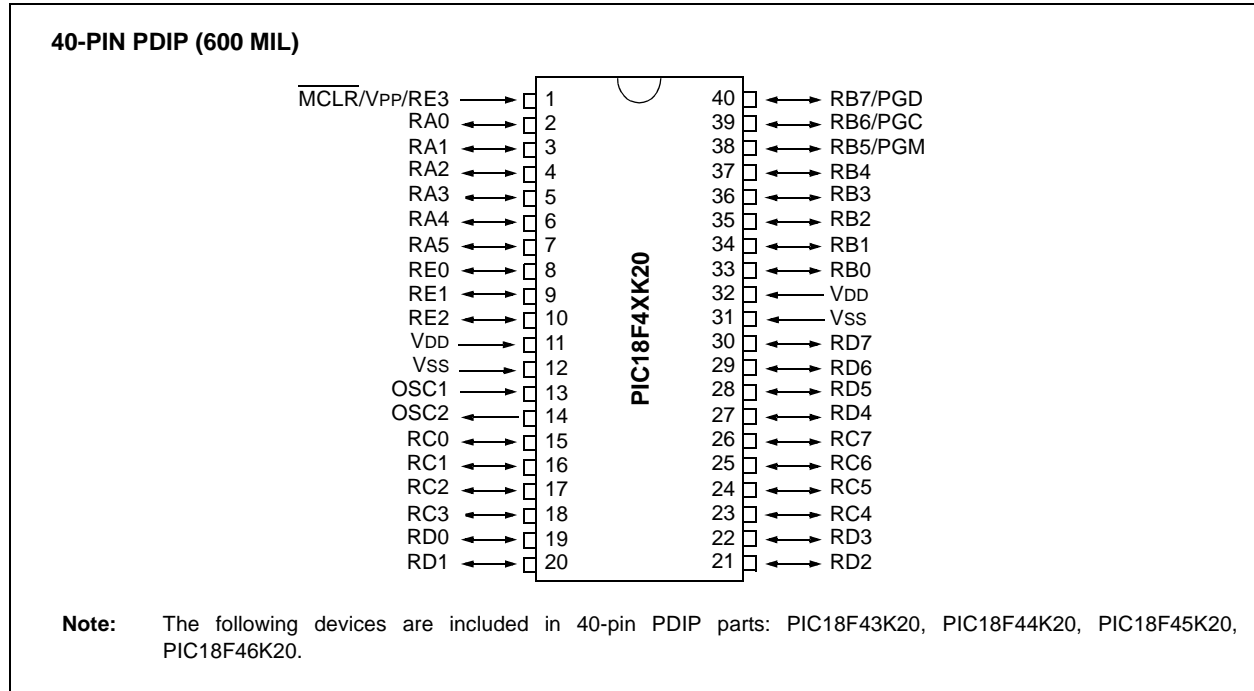
Advance Information

FIGURE 2-3: **40-PIN PDIP PIN DIAGRAMS**

40-PIN PDIP (600 MIL)

PIC18F4XK20

Pin	Signal		Pin	Signal
1	$\overline{MCLR}$/VPP/RE3		40	RB7/PGD
2	RA0		39	RB6/PGC
3	RA1		38	RB5/PGM
4	RA2		37	RB4
5	RA3		36	RB3
6	RA4		35	RB2
7	RA5		34	RB1
8	RE0		33	RB0
9	RE1		32	VDD
10	RE2		31	VSS
11	VDD		30	RD7
12	VSS		29	RD6
13	OSC1		28	RD5
14	OSC2		27	RD4
15	RC0		26	RC7
16	RC1		25	RC6
17	RC2		24	RC5
18	RC3		23	RC4
19	RD0		22	RD3
20	RD1		21	RD2

Note: The following devices are included in 40-pin PDIP parts: PIC18F43K20, PIC18F44K20, PIC18F45K20, PIC18F46K20.

FIGURE 2-4: **44-PIN TQFP PIN DIAGRAMS**

44-PIN TQFP

PIC18F4XK20

Note: The following devices are included in 44-pin TQFP parts: PIC18F43K20, PIC18F44K20, PIC18F45K20, PIC18F46K20.

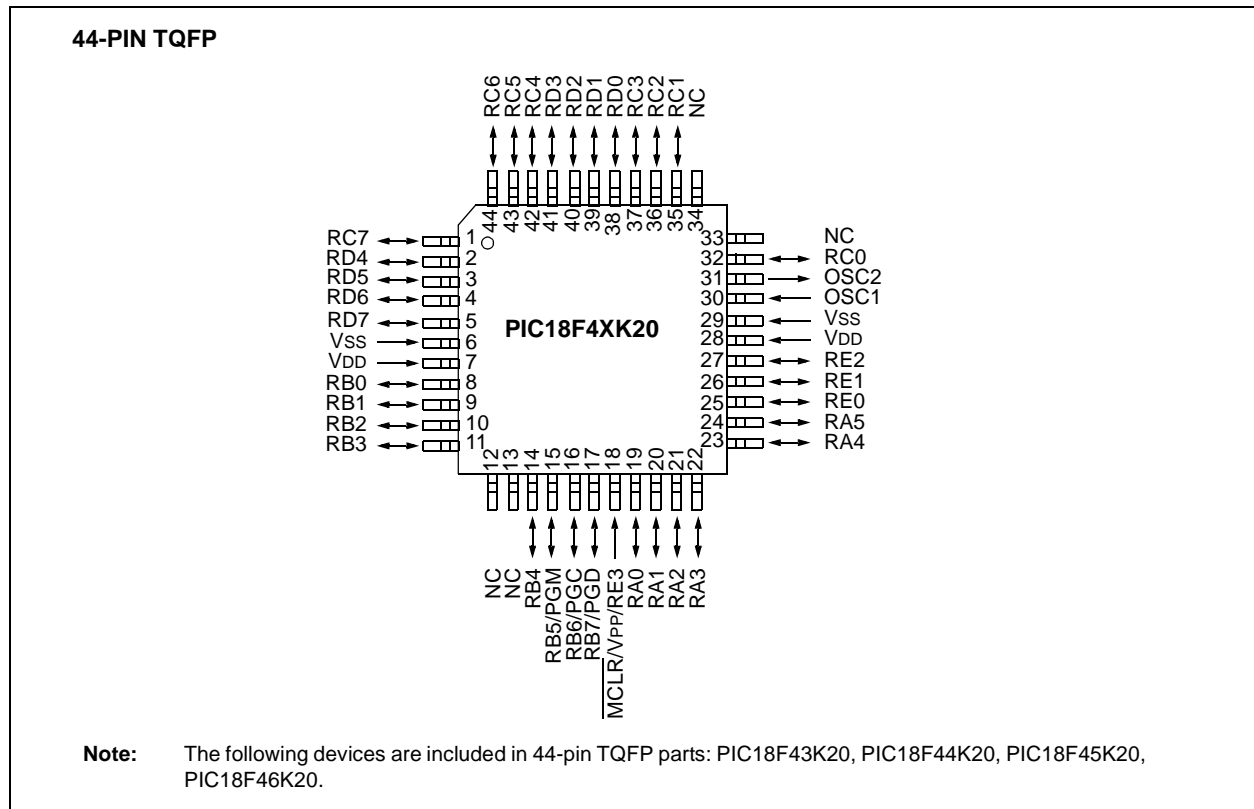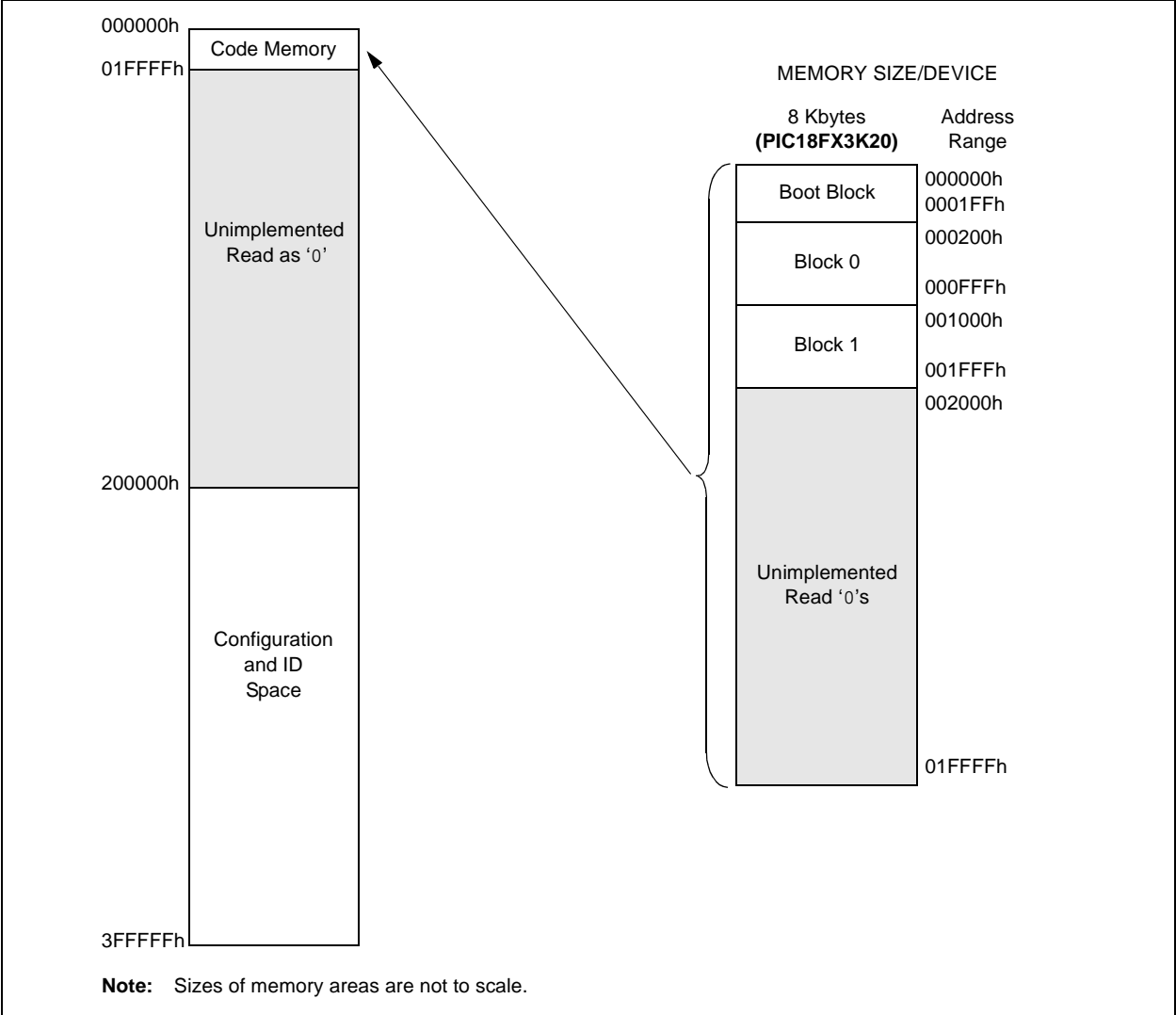## 2.3 Memory Maps

For the PIC18FX3K20 devices, the code memory space extends from 0000h to 01FFFh (8 Kbytes) in two 4-Kbyte blocks. Addresses 0000h through 01FFh, 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 2-2: IMPLEMENTATION OF CODE MEMORY

Device	Code Memory Size (Bytes)
PIC18F23K20	000000h-001FFFh (8K)
PIC18F43K20	

FIGURE 2-6: MEMORY MAP AND THE CODE MEMORY SPACE FOR PIC18FX3K20 DEVICES

Note: Sizes of memory areas are not to scale.

In addition to the code memory space, there are three blocks in the configuration and ID space that are accessible to the user through table reads and table writes. Their locations in the memory map are shown in Figure 2-10.

Users may store identification information (ID) in eight ID registers. These ID registers are mapped in addresses 200000h through 200007h. The ID locations read out normally, even after code protection is applied.

Locations 300000h through 30000Dh are reserved for the Configuration bits. These bits select various device options and are described in **Section 5.0 "Configuration Word"**. These Configuration bits read out normally, even after code protection.

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 5.0 "Configuration Word"**. These device ID bits read out normally, even after code protection.

### 2.3.1   MEMORY ADDRESS POINTER

Memory in the address space, 0000000h to 3FFFFFh, is addressed via the Table Pointer register, which is comprised of three Pointer 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 any read or write operations.

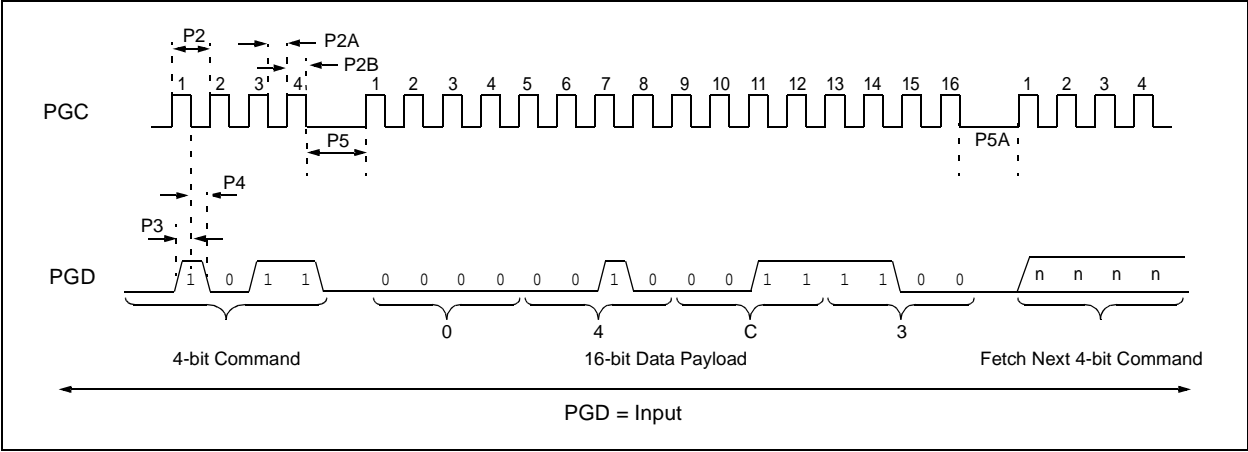**FIGURE 2-10:   CONFIGURATION AND ID LOCATIONS FOR PIC18F2XK20/4XK20 DEVICES**

Note: Sizes of memory areas are not to scale.
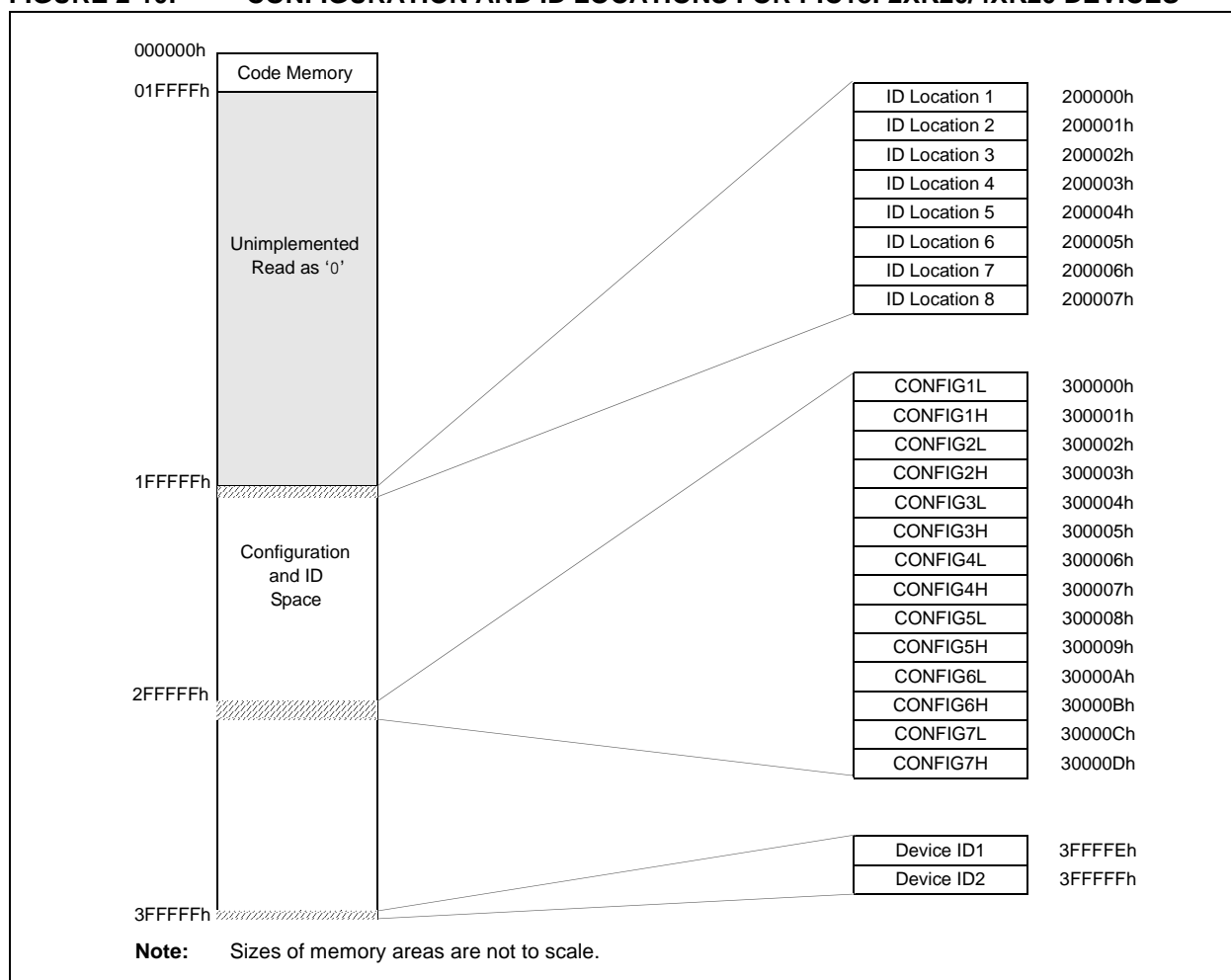
TABLE 2-7: **SAMPLE COMMAND SEQUENCE**

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

FIGURE 2-16: **TABLE WRITE, POST-INCREMENT TIMING DIAGRAM (1101)**

PIC18F2XK20/4XK20

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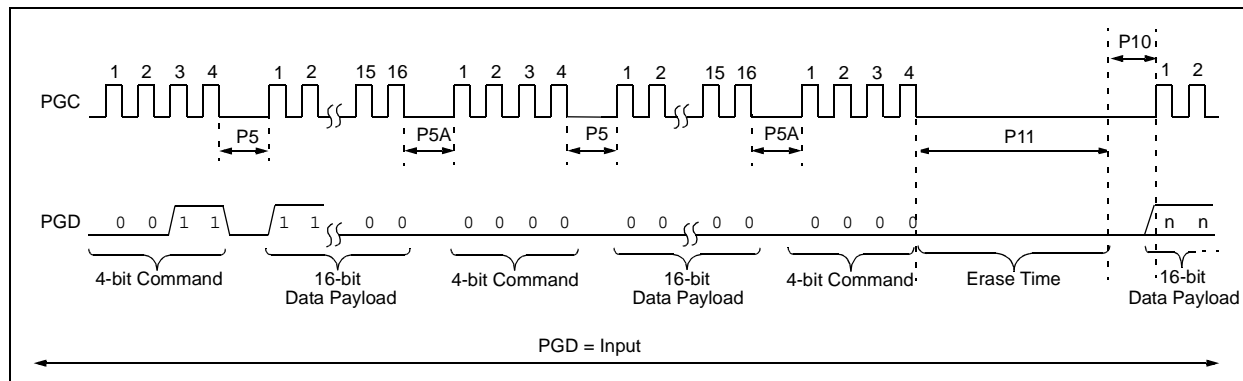
3.1.2 LOW-VOLTAGE ICSP BULK ERASE

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

FIGURE 3-3: **SINGLE ROW ERASE CODE MEMORY FLOW**
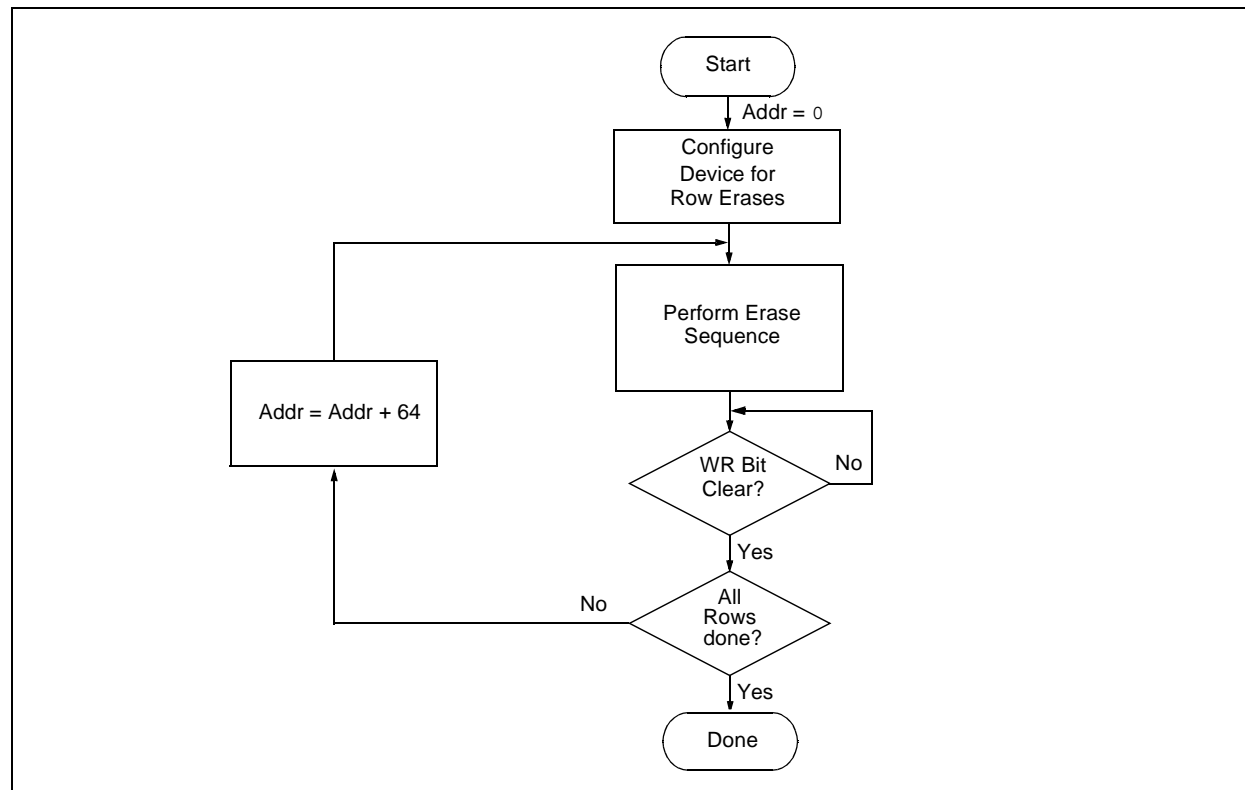
3.2 Code Memory Programming

Programming code memory is accomplished by first loading data into the write buffer and then initiating a programming sequence. The write and erase buffer sizes shown in Table 3-4 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.

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 PIC18F2XK20/4XK20 device is shown in Table 3-5. The flowchart shown in Figure 3-4 depicts the logic necessary to completely write a PIC18F2XK20/4XK20 device. The timing diagram that details the Start Programming command and parameters P9 and P10 is shown in Figure 3-5.

> **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 3-4: WRITE AND ERASE BUFFER SIZES

Devices (Arranged by Family)	Write Buffer Size (bytes)	Erase Size (bytes)
PIC18F26K20, PIC18F46K20	64	64
PIC18F24K20, PIC18F25K20, PIC18F44K20, PIC18F45K20	32	64
PIC18F23K20, PIC18F43K20	16	64

TABLE 3-5: WRITE CODE MEMORY CODE SEQUENCE

4-bit Command	Data Payload	Core Instruction
Step 1: Direct access to code memory.		

0000
0000
0000
	8E A6
	9C A6
	84 A6
		`BSF   EECON1, EEPGD`
		`BCF   EECON1, CFGS`
		`BSF   EECON1, WREN`

Step 2: Point to row to write.		

0000
0000
0000
0000
0000
0000
	0E <Addr[21:16]>
	6E F8
	0E <Addr[15:8]>
	6E F7
	0E <Addr[7:0]>
	6E F6
		`MOVLW <Addr[21:16]>`
		`MOVWF TBLPTRU`
		`MOVLW <Addr[15:8]>`
		`MOVWF TBLPTRH`
		`MOVLW <Addr[7:0]>`
		`MOVWF TBLPTRL`

Step 3: Load write buffer. Repeat for all but the last two bytes.		
1101	<MSB><LSB>	`Write 2 bytes and post-increment address by 2.`
Step 4: Load write buffer for last two bytes and start programming.		
1111 0000	<MSB><LSB> 00 00	`Write 2 bytes and start programming.` `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.		

PIC18F2XK20/4XK20

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.
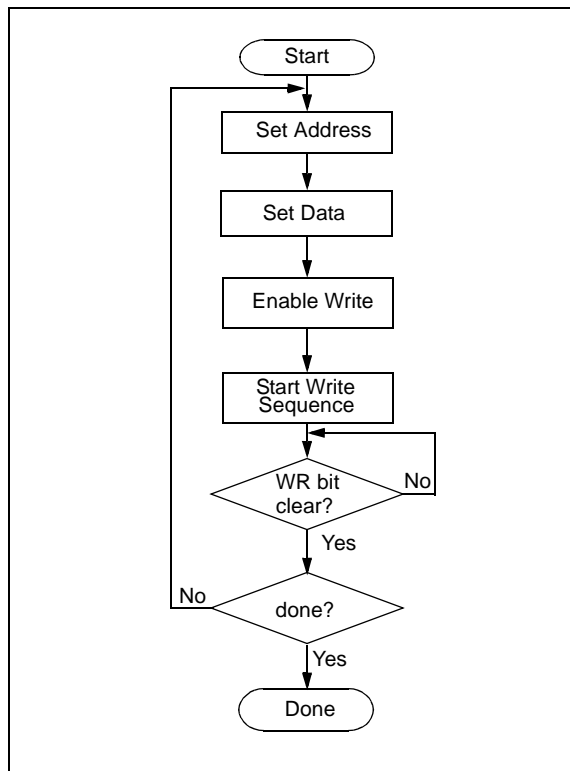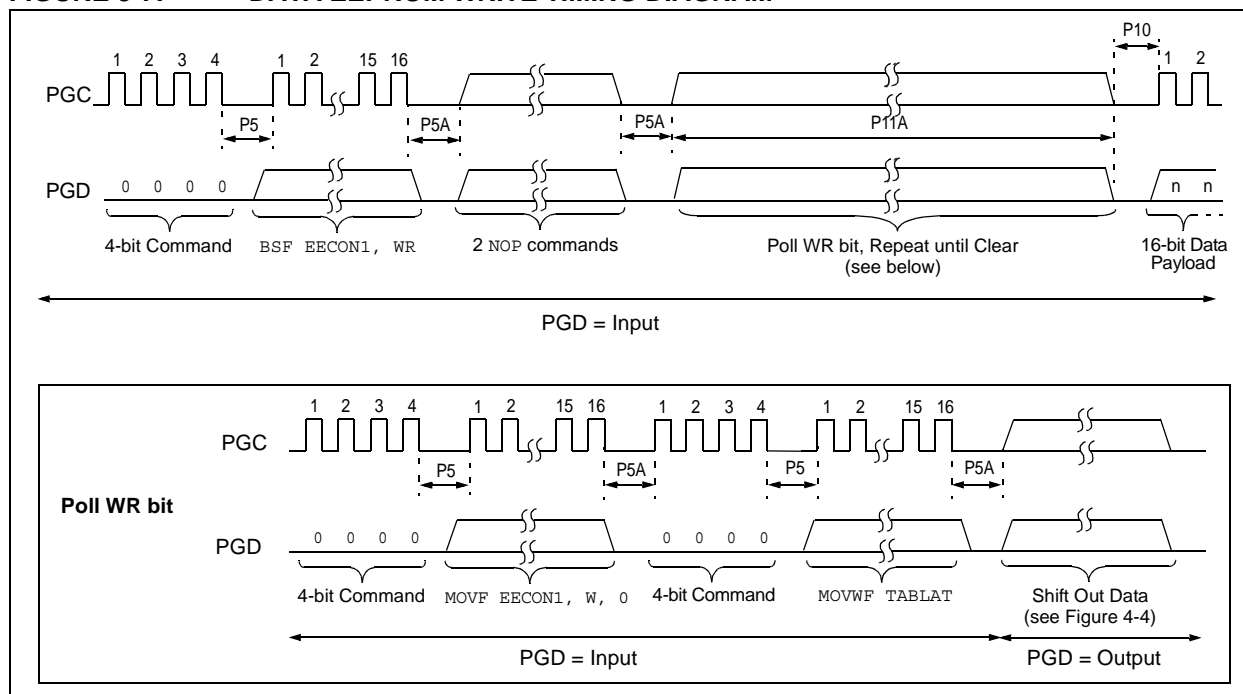
FIGURE 3-6: PROGRAM DATA FLOW

FIGURE 3-7: DATA EEPROM WRITE TIMING DIAGRAM

Advance Information © 2009 Microchip Technology Inc.

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.

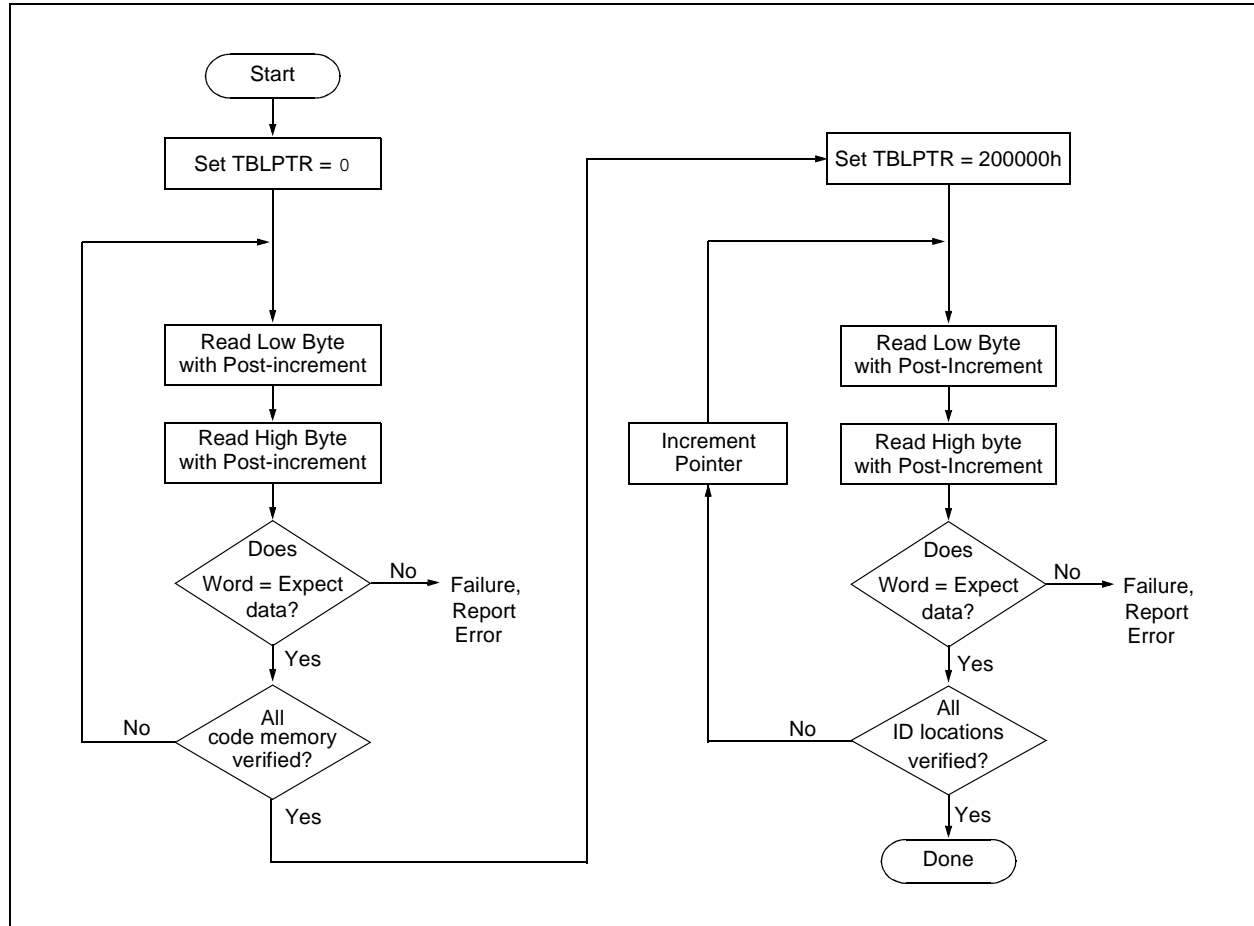**FIGURE 4-2:   VERIFY CODE MEMORY FLOW**

FIGURE 4-4: **SHIFT OUT DATA HOLDING REGISTER TIMING DIAGRAM (`0010`)**

Note 1: Magnification of the High-Impedance delay between PGC and PGD is shown in Figure 4-5.
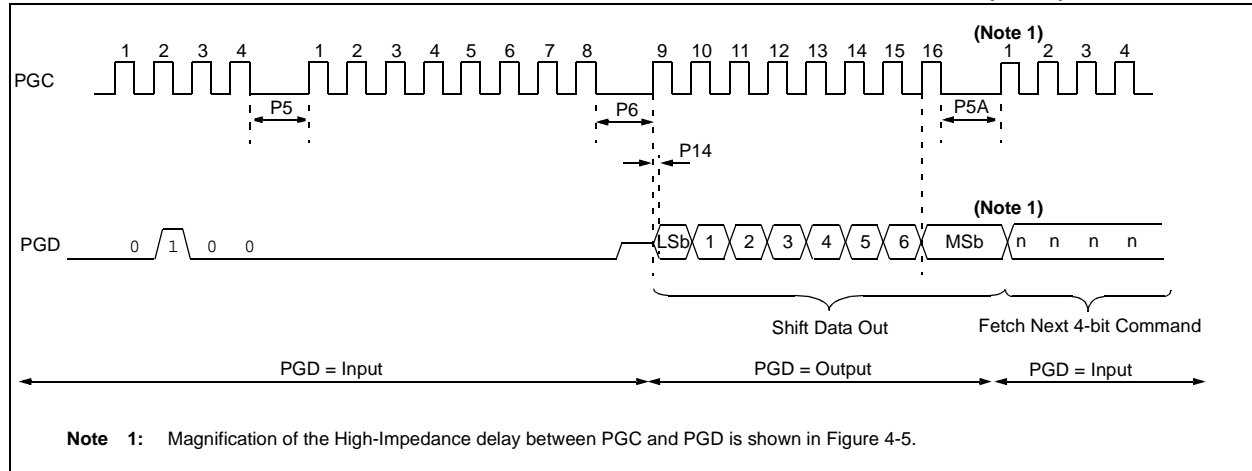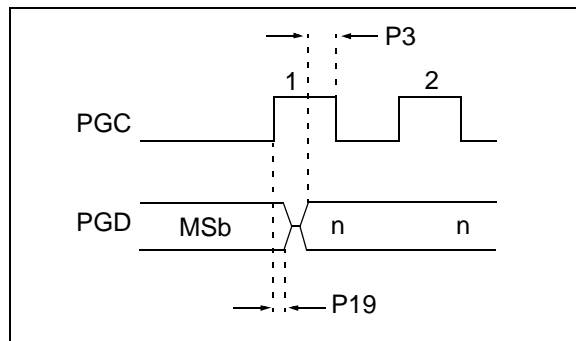
FIGURE 4-5: **HIGH-IMPEDANCE DELAY**

4.5 Verify Data EEPROM

A data EEPROM address may be read via a sequence of core instructions (4-bit command, '`0000`') and then output on PGD via the 4-bit command, '`0010`' (TABLAT register). The result may then be immediately compared to the appropriate data in the programmer's memory for verification. Refer to **Section 4.4 "Read Data EEPROM Memory"** for implementation details of reading data EEPROM.

4.6 Blank Check

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

A "blank" or "erased" memory cell will read as a '`1`'. Therefore, Blank Checking a device merely means to verify that all bytes read as FFh except the Configuration bits. Unused (reserved) Configuration bits will read '`0`' (programmed). Refer to Table 5-1 for blank configuration expect data for the various PIC18F2XK20/4XK20 devices.

Given that Blank Checking is merely code and data EEPROM verification with FFh expect data, refer to **Section 4.4 "Read Data EEPROM Memory"** and **Section 4.2 "Verify Code Memory and ID Locations"** for implementation details.

FIGURE 4-6: **BLANK CHECK FLOW**
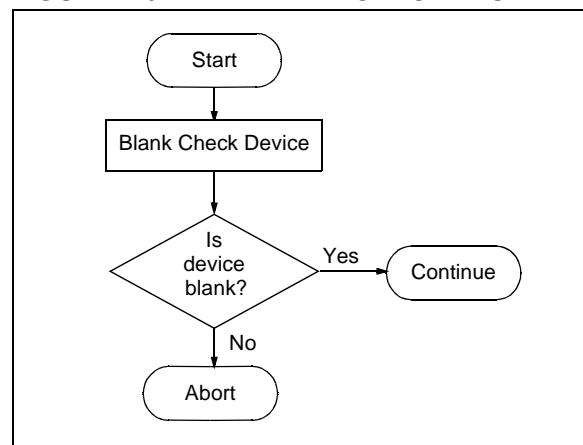
PIC18F2XK20/4XK20

5.0 CONFIGURATION WORD

The PIC18F2XK20/4XK20 devices have several Configuration Words. These bits can be set or cleared to select various device configurations. All other memory areas should be programmed and verified prior to setting Configuration Words. These bits may be read out normally, even after read or code protection. See Table 5-1 for a list of Configuration bits and device IDs and Table 5-3 for the Configuration bit descriptions.

5.1 User ID Locations

A user may store identification information (ID) in eight ID locations mapped in 200000h:200007h. It is recommended that the Most Significant nibble of each ID be Fh. In doing so, if the user code inadvertently tries to execute from the ID space, the ID data will execute as a `NOP`.

5.2 Device ID Word

The device ID word for the PIC18F2XK20/4XK20 devices is located at 3FFFFEh:3FFFFFh. These bits may be used by the programmer to identify what device type is being programmed and read out normally, even after code or read protection. See Table 5-2 for a complete list of device ID values.
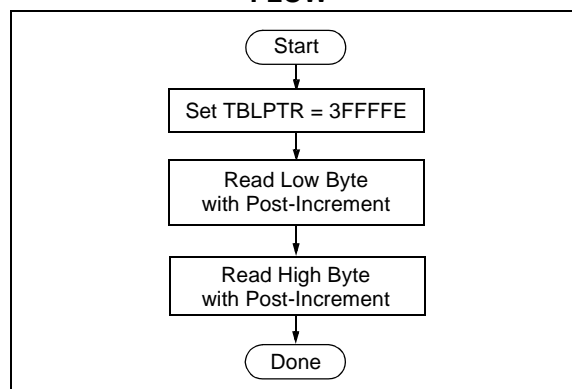
FIGURE 5-1: **READ DEVICE ID WORD FLOW**

TABLE 5-1: 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
300001h	CONFIG1H	IESO	FCMEN	—	—	FOSC3	FOSC2	FOSC1	FOSC0	`00-- 0111`
300002h	CONFIG2L	—	—	—	BORV1	BORV0	BOREN1	BOREN0	$\overline{\text{PWRTEN}}$	`---1 1111`
300003h	CONFIG2H	—	—	—	WDTPS3	WDTPS2	WDTPS1	WDTPS0	WDTEN	`---1 1111`
300005h	CONFIG3H	MCLRE	—	—	—	HFOFST	LPT1OSC	PBADEN	CCP2MX	`1--- 1011`
300006h	CONFIG4L	DEBUG	XINST	—	—	—	LVP	—	STVREN	`10-- -1-1`
300008h	CONFIG5L	—	—	—	—	CP3[1]	CP2[1]	CP1	CP0	`---- 1111`
300009h	CONFIG5H	CPD	CPB	—	—	—	—	—	—	`11-- ----`
30000Ah	CONFIG6L	—	—	—	—	WRT3[1]	WRT2[1]	WRT1	WRT0	`---- 1111`
30000Bh	CONFIG6H	WRTD	WRTB	WRTC	—	—	—	—	—	`111- ----`
30000Ch	CONFIG7L	—	—	—	—	EBTR3[1]	EBTR2[1]	EBTR1	EBTR0	`---- 1111`
30000Dh	CONFIG7H	—	EBTRB	—	—	—	—	—	—	`-1-- ----`
3FFFFEh	DEVID1[2]	DEV2	DEV1	DEV0	REV4	REV3	REV2	REV1	REV0	See Table 5-2
3FFFFFh	DEVID2[2]	DEV10	DEV9	DEV8	DEV7	DEV6	DEV5	DEV4	DEV3	See Table 5-2

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

Note 1: These bits are only implemented on specific devices. Refer to **Section 2.3 "Memory Maps"** to determine which bits apply based on available memory.

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

Advance Information

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.

PIC18F2XK20/4XK20

TABLE 5-3: PIC18F2XK20/4XK20 BIT DESCRIPTIONS

Bit Name	Configuration Words	Description
IESO	CONFIG1H	Internal External Switchover bit `1` = Internal External Switchover mode enabled `0` = Internal External Switchover mode disabled
FCMEN	CONFIG1H	Fail-Safe Clock Monitor Enable bit `1` = Fail-Safe Clock Monitor enabled `0` = Fail-Safe Clock Monitor disabled
FOSC<3:0>	CONFIG1H	Oscillator Selection bits `11xx` = External RC oscillator, CLKOUT function on RA6 `101x` = External RC oscillator, CLKOUT function on RA6 `1001` = HFINTOSC, CLKOUT function on RA6, port function on RA7 `1000` = HFINTOSC, port function on RA6, port function on RA7 `0111` = External RC oscillator, port function on RA6 `0110` = HS oscillator, PLL enabled (clock frequency = 4 x FOSC1) `0101` = EC oscillator, port function on RA6 `0100` = EC oscillator, CLKOUT function on RA6 `0011` = External RC oscillator, CLKOUT function on RA6 `0010` = HS oscillator `0001` = XT oscillator `0000` = LP oscillator
BORV<1:0>	CONFIG2L	Brown-out Reset Voltage bits `11` = V$_{BOR}$ set to 1.8V `10` = V$_{BOR}$ set to 2.2V `01` = V$_{BOR}$ set to 2.7V `00` = V$_{BOR}$ set to 3.0V
BOREN<1:0>	CONFIG2L	Brown-out Reset Enable bits `11` = Brown-out Reset enabled in hardware only (SBOREN is disabled) `10` = Brown-out Reset enabled in hardware only and disabled in Sleep mode (SBOREN is disabled) `01` = Brown-out Reset enabled and controlled by software (SBOREN is enabled) `00` = Brown-out Reset disabled in hardware and software
PWRTEN	CONFIG2L	Power-up Timer Enable bit `1` = PWRT disabled `0` = PWRT enabled
WDPS<3:0>	CONFIG2H	Watchdog Timer Postscaler 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

.

Advance Information

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

.

PIC18F2XK20/4XK20

5.3 Single-Supply ICSP Programming

The LVP bit in Configuration register, CONFIG4L, enables Single-Supply (Low-Voltage) ICSP Programming. The LVP bit defaults to a '1' (enabled) from the factory.

If Single-Supply Programming mode is not used, the LVP bit can be programmed to a '0' and RB5/PGM becomes a digital I/O pin. However, the LVP bit may only be programmed by entering the High-Voltage ICSP mode, where $\overline{MCLR}$/VPP/RE3 is raised to VIHH. Once the LVP bit is programmed to a '0', only the High-Voltage ICSP mode is available and only the High-Voltage ICSP mode can be used to program the device.

Note 1:	The High-Voltage ICSP mode is always available, regardless of the state of the LVP bit, by applying VIHH to the $\overline{MCLR}$/VPP/RE3 pin.
2:	While in Low-Voltage ICSP mode, the RB5 pin can no longer be used as a general purpose I/O.

5.4 Embedding Configuration Word Information in the HEX File

To allow portability of code, a PIC18F2XK20/4XK20 programmer is required to read the Configuration Word locations from the hex file. If Configuration Word information is not present in the hex file, then a simple warning message should be issued. Similarly, while saving a hex file, all Configuration Word information must be included. An option to not include the Configuration Word information may be provided. When embedding Configuration Word information in the hex file, it should start at address 300000h.

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

5.5 Embedding Data EEPROM Information In the HEX File

To allow portability of code, a PIC18F2XK20/4XK20 programmer is required to read the data EEPROM information from the hex file. If data EEPROM information is not present, a simple warning message should be issued. Similarly, when saving a hex file, all data EEPROM information must be included. An option to not include the data EEPROM information may be provided. When embedding data EEPROM information in the hex file, it should start at address F00000h.

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

5.6 Checksum Computation

The checksum is calculated by summing the following:

* The contents of all code memory locations
* The Configuration Word, appropriately masked
* ID locations (Only if any portion of program memory is code-protected)

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

Code protection limits access to program memory by both external programmer (code-protect) and code execution (table read protect). The ID locations, when included in a code protected checksum, contain the checksum of an unprotected part. The unprotected checksum is distributed: one nibble per ID location. Each nibble is right justified.

Table 5-4 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 5-4: 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

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 $\overline{MCLR}/V_{PP}/RE3$	V_{DD} + 4.5	9	V	
D110A	V_{IHL}	Low-Voltage Programming Voltage on $\overline{MCLR}/V_{PP}/RE3$	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 $\overline{MCLR}/V_{PP}/RE3$	—	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	$\overline{MCLR}/V_{PP}/RE3$ 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 $\overline{MCLR}$ 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.

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.

NOTES: