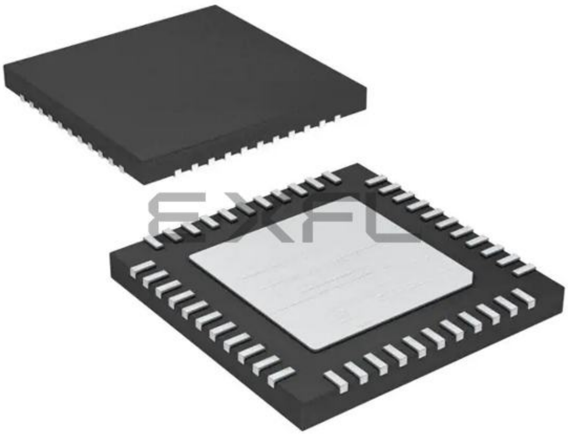


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

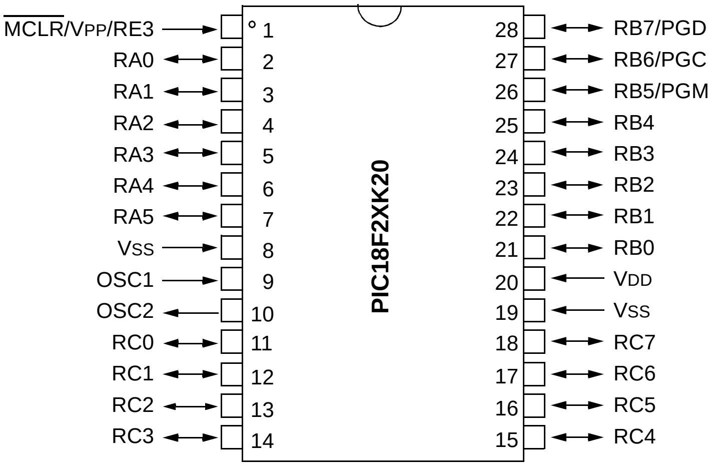
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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	35
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 14x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-VQFN Exposed Pad
Supplier Device Package	44-QFN (8x8)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f43k20-i-ml

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

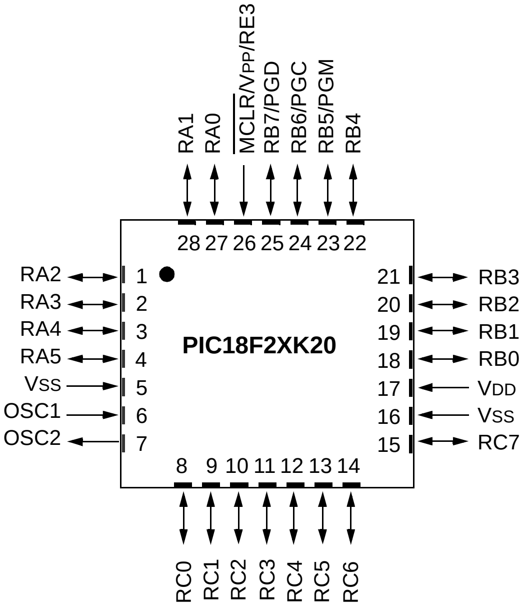
SDIP, SSOP, SOIC (300 MIL)



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

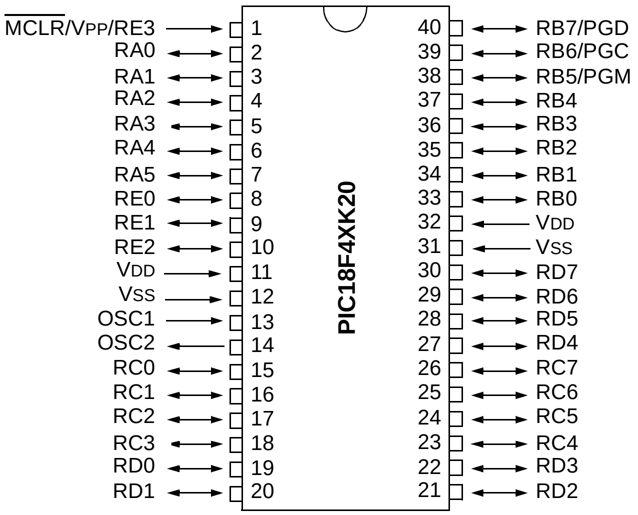
28-Pin QFN



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

FIGURE 2-3: 40-PIN PDIP PIN DIAGRAMS

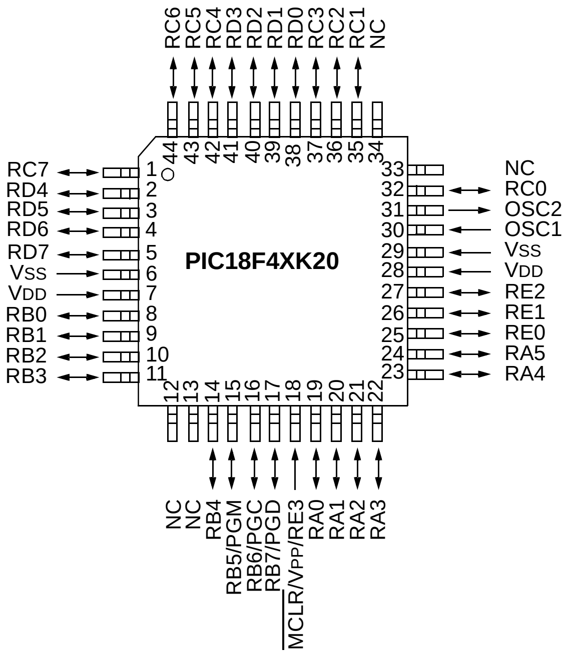
40-PIN PDIP (600 MIL)



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

FIGURE 2-4: 44-PIN TQFP PIN DIAGRAMS

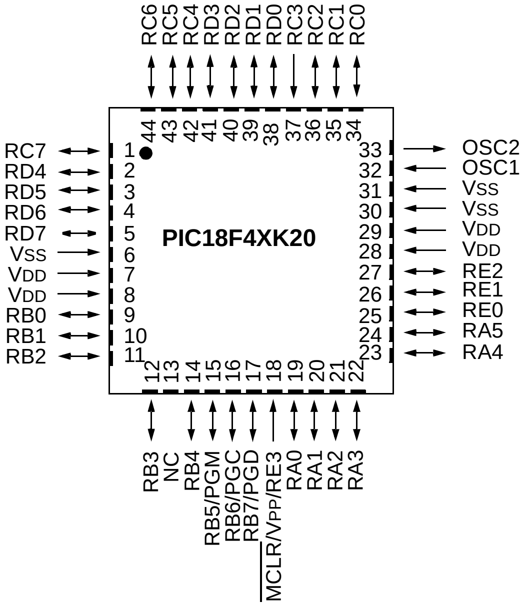
44-PIN TQFP



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

FIGURE 2-5: 44-PIN QFN PIN DIAGRAMS

44-PIN QFN



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

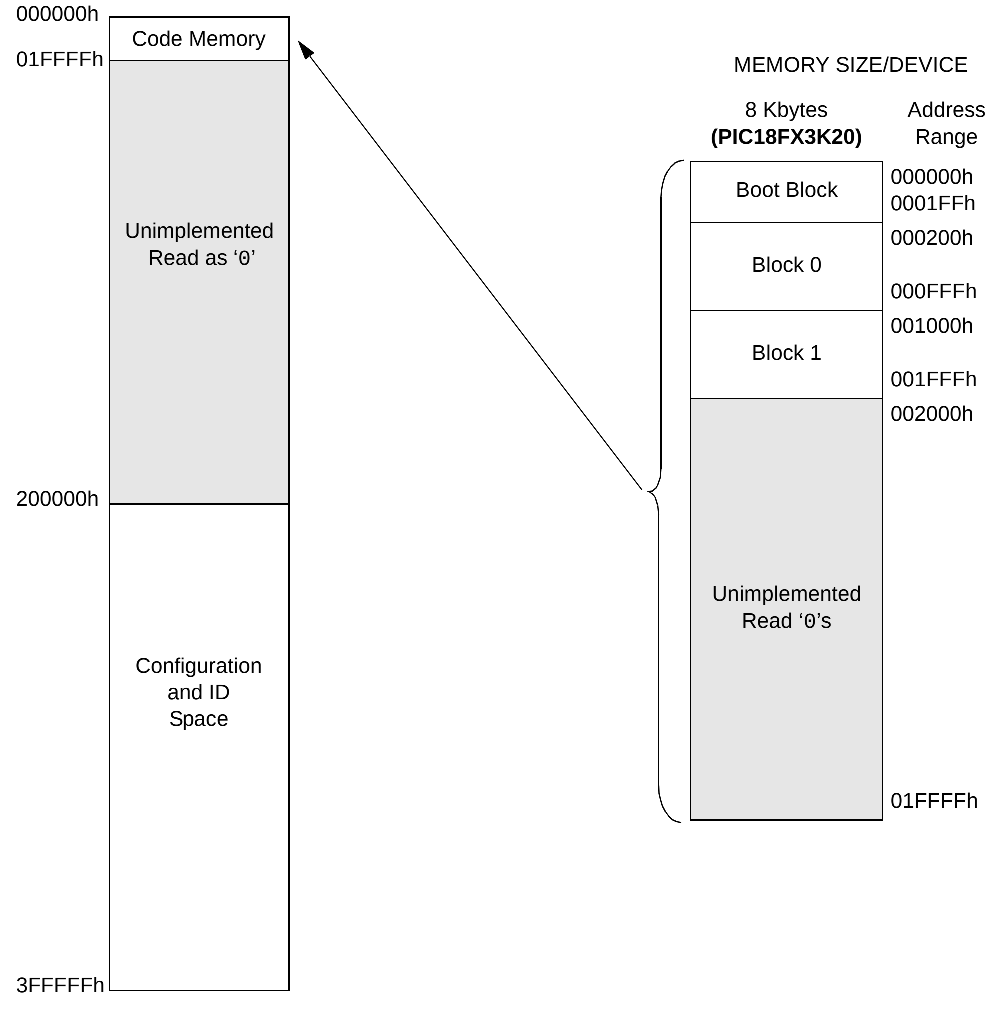
2.3 Memory Maps

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



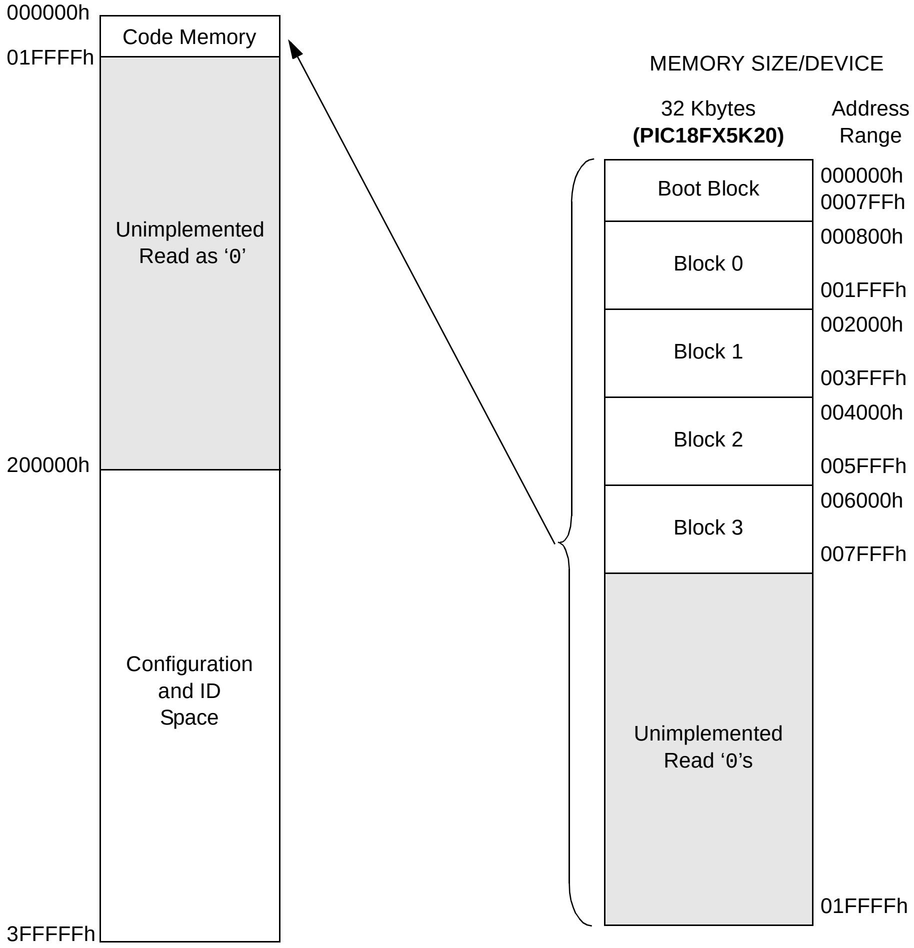
Note: Sizes of memory areas are not to scale.

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

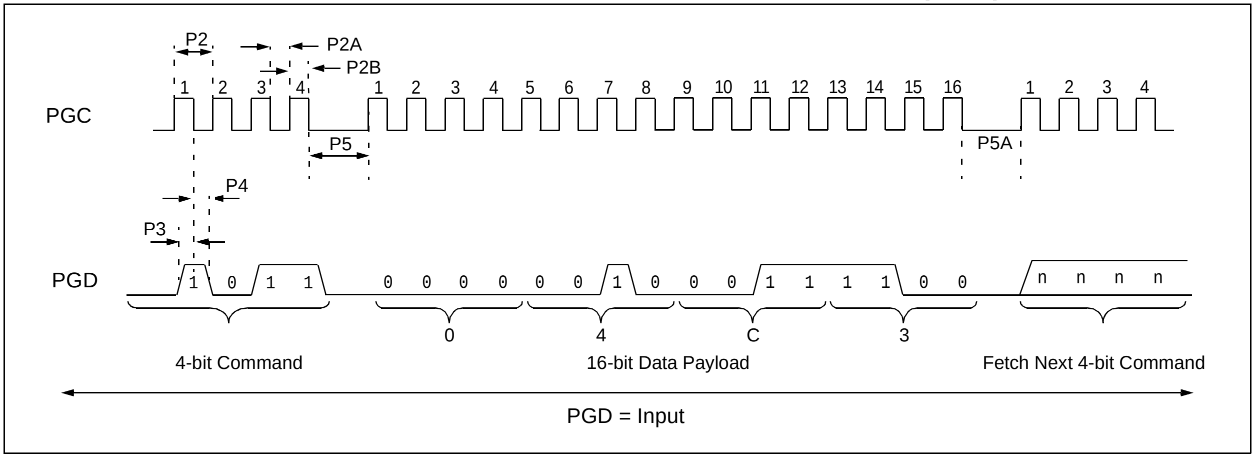


Note: Sizes of memory areas not to scale.

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`)



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

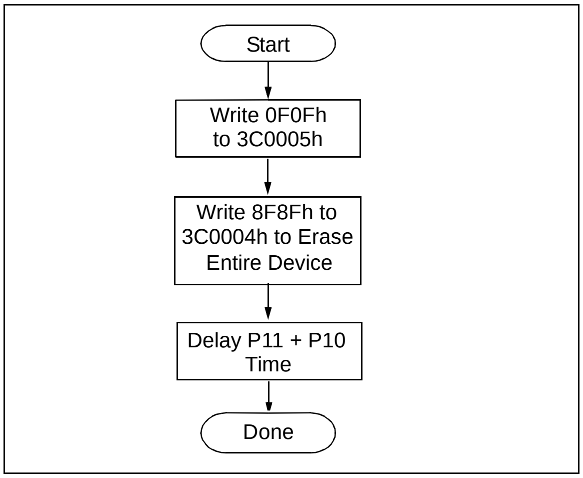
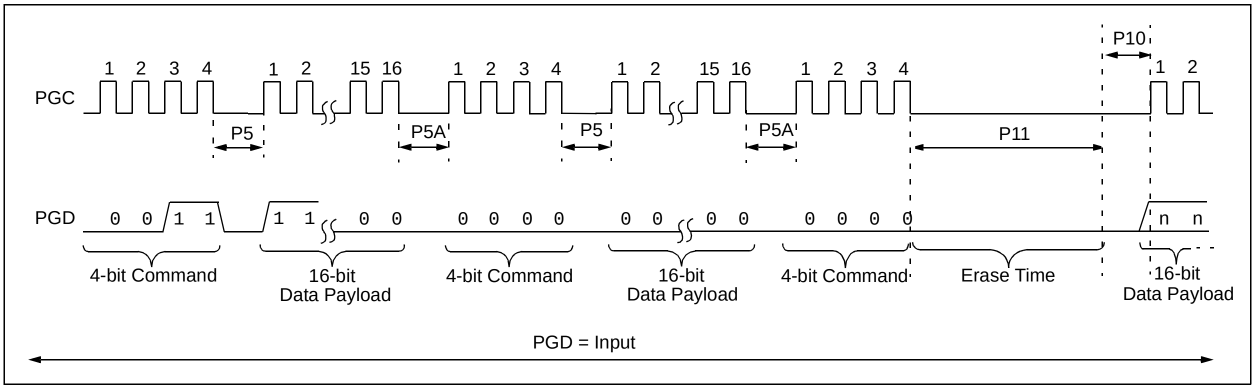


FIGURE 3-2: BULK ERASE TIMING DIAGRAM



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 `NOP`s 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.

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.		

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

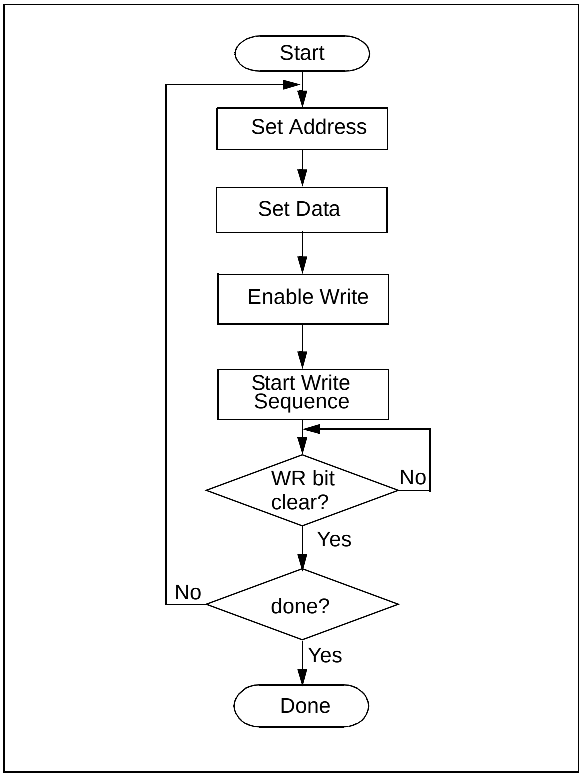
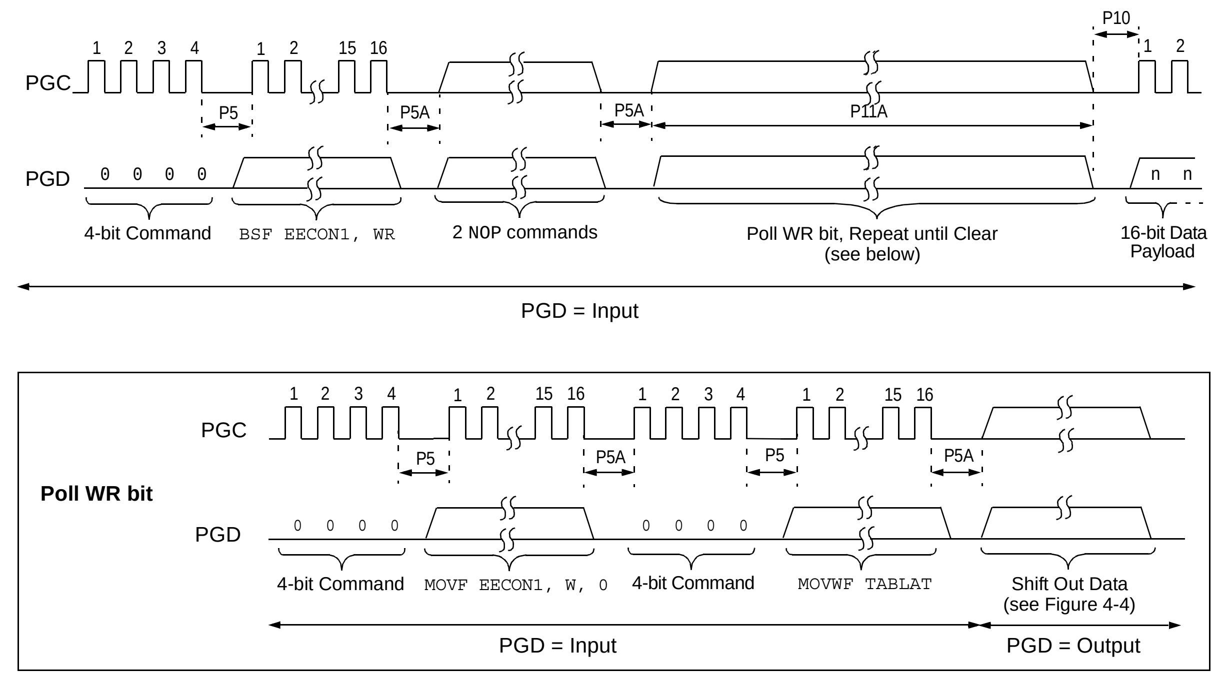


FIGURE 3-7: DATA EEPROM WRITE TIMING DIAGRAM



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.

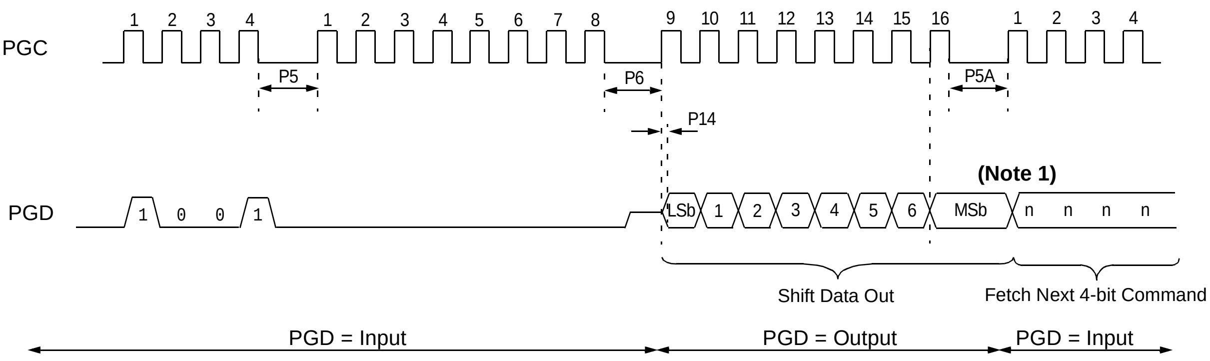
TABLE 4-1: READ CODE MEMORY SEQUENCE

4-bit Command	Data Payload	Core Instruction
Step 1: Set Table Pointer		

`0000`
`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 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.

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

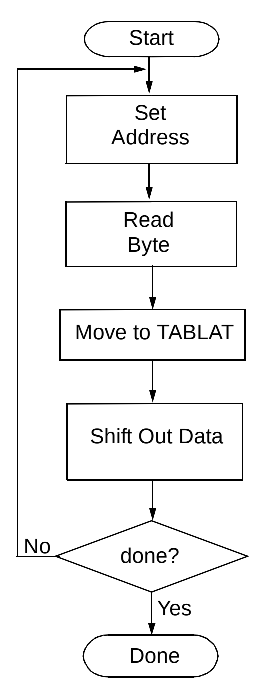


TABLE 4-2: READ DATA EEPROM MEMORY

4-bit Command	Data Payload	Core Instruction
Step 1: Direct access to data EEPROM.		
`0000` `0000`	9E A6 9C A6	`BCF EECON1, EEPGD` `BCF EECON1, CFGS`
Step 2: Set the data EEPROM Address Pointer.		

`0000`
`0000`
`0000`
`0000`
	0E <Addr>
	6E A9
	OE <AddrH>
	6E AA
		`MOVLW <Addr>`
		`MOVWF EEADR`
		`MOVLW <AddrH>`
		`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`
`0000`
`0000`
`0010`
	50 A8
	6E F5
	00 00
	<MSB><LSB>
		`MOVF EEDATA, W, 0`
		`MOVWF TABLAT`
		`NOP`
		`Shift Out Data`[1]

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

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 $\overline{\text{MCLR}}$/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 $\overline{\text{MCLR}}$/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

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

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

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 $\overline{\text{MCLR}}/V_{PP}$/RE3 ↑	2	—	μs	
P13	T_{SET2}	V_{DD} ↑ Setup Time to $\overline{\text{MCLR}}/V_{PP}$/RE3 ↑	100	—	ns	
P14	T_{VALID}	Data Out Valid from PGC ↑	10	—	ns	
P15	T_{SET3}	PGM ↑ Setup Time to $\overline{\text{MCLR}}/V_{PP}$/RE3 ↑	2	—	μs	
P16	T_{DLY8}	Delay between Last PGC ↓ and $\overline{\text{MCLR}}/V_{PP}$/RE3 ↓	0	—	s	
P17	T_{HLD3}	$\overline{\text{MCLR}}/V_{PP}$/RE3 ↓ to V_{DD} ↓	—	100	ns	
P18	T_{HLD4}	$\overline{\text{MCLR}}/V_{PP}$/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 $\overline{\text{MCLR}}$ 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:

WORLDWIDE SALES AND SERVICE

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://support.microchip.com
Web Address:
www.microchip.com

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

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Cleveland
Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Farmington Hills, MI
Tel: 248-538-2250
Fax: 248-538-2260

Kokomo
Kokomo, IN
Tel: 765-864-8360
Fax: 765-864-8387

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

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

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

ASIA/PACIFIC

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

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8528-2100
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Hong Kong SAR
Tel: 852-2401-1200
Fax: 852-2401-3431

China - Nanjing
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8203-2660
Fax: 86-755-8203-1760

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4080

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

India - Pune
Tel: 91-20-2566-1512
Fax: 91-20-2566-1513

Japan - Yokohama
Tel: 81-45-471- 6166
Fax: 81-45-471-6122

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

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-6578-300
Fax: 886-3-6578-370

Taiwan - Kaohsiung
Tel: 886-7-536-4818
Fax: 886-7-536-4803

Taiwan - Taipei
Tel: 886-2-2500-6610
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

UK - Wokingham
Tel: 44-118-921-5869
Fax: 44-118-921-5820

03/26/09