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	25MHz
Connectivity	EBI/EMI, I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	68
Program Memory Size	64KB (32K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	3.75K x 8
Voltage - Supply (Vcc/Vdd)	4.2V ~ 5.5V
Data Converters	A/D 16x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	80-TQFP
Supplier Device Package	80-TQFP (12x12)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f8620-i-pt

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 2-3.

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. 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. These Device ID bits read out normally, even after code protection.

2.3.1 MEMORY ADDRESS POINTER

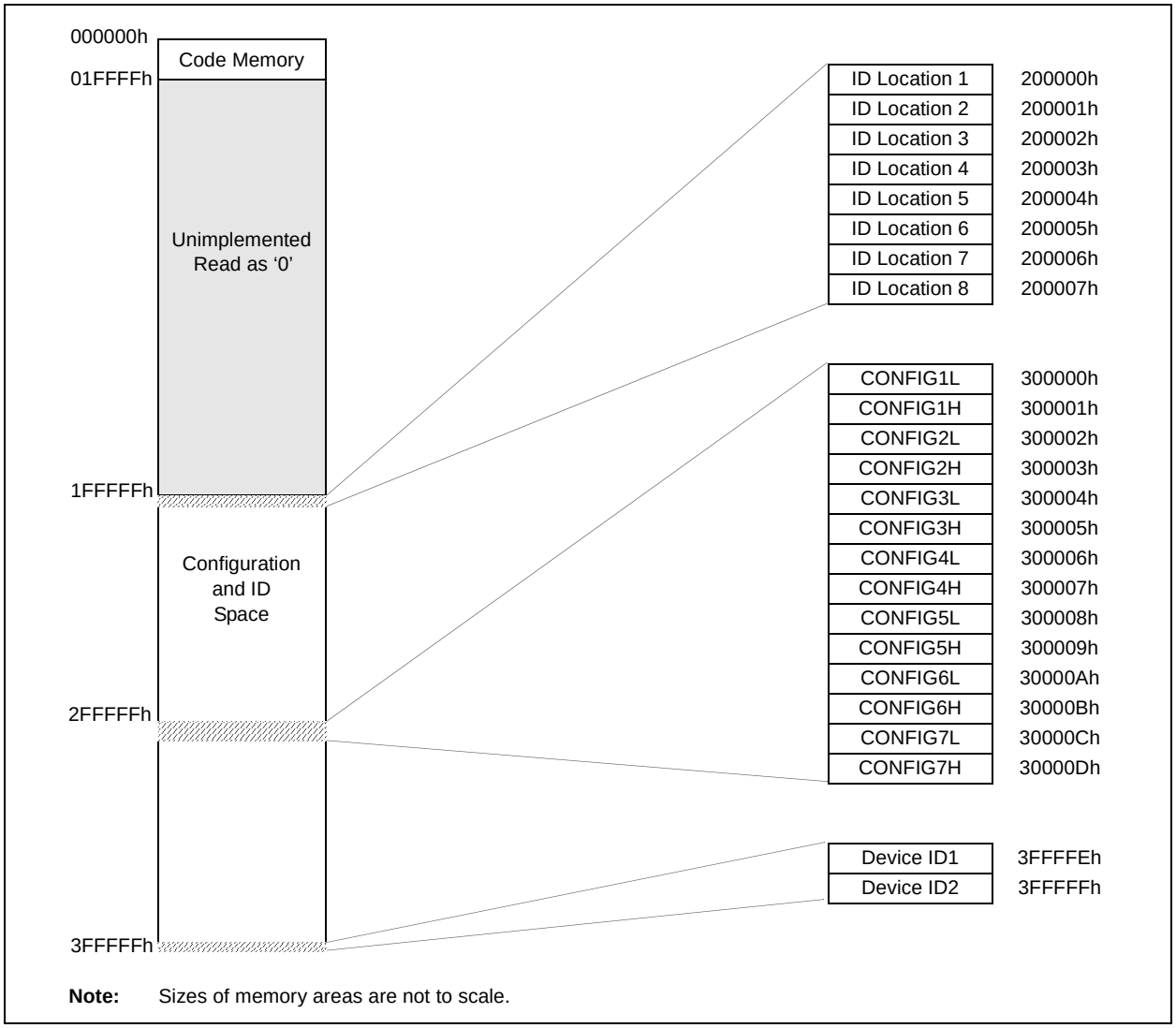
Memory in the address space 0000000h to 3FFFFFh is addressed via the Table Pointer, which is comprised of three pointer registers:

- 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 many Read or Write operations.

FIGURE 2-3: CONFIGURATION AND ID LOCATIONS FOR PIC18FXX20 DEVICES



Note: Sizes of memory areas are not to scale.

© 2010 Microchip Technology Inc.

3.0 DEVICE PROGRAMMING

3.1 High Voltage ICSP Bulk Erase

Erasing Code or Data EEPROM is accomplished by writing an “erase option” to address 3C0004h. 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.

TABLE 3-1: BULK ERASE OPTIONS

Description	Data
Chip Erase	80h
Erase Data EEPROM	81h
Erase Boot Block	83h
Erase Block 1	88h
Erase Block 2	89h
Erase Block 3	8Ah
Erase Block 4	8Bh
Erase Block 5	8Ch
Erase Block 6	8Dh
Erase Block 7	8Eh
Erase Block 8	8Fh

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

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

Note: A bulk erase is the only way to reprogram code protect bits from an on-state to an off-state.
Non-code protect bits are not returned to default settings by a bulk erase. These bits should be programmed to ones, as outlined in Section 3.6, "Configuration Bits Programming".

FIGURE 3-1: BULK ERASE COMMAND SEQUENCE

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

FIGURE 3-2: BULK ERASE FLOW

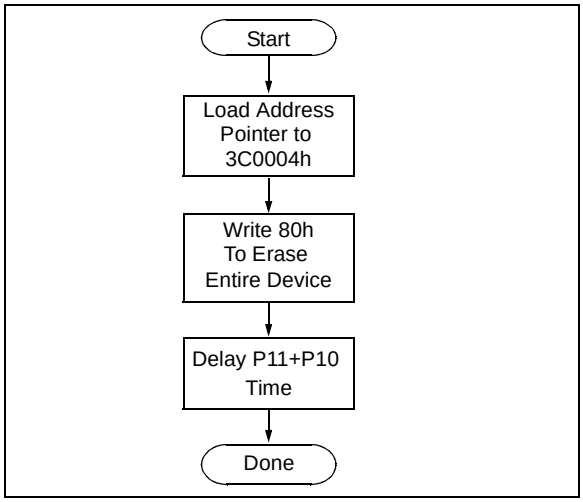


FIGURE 3-3: BULK ERASE TIMING

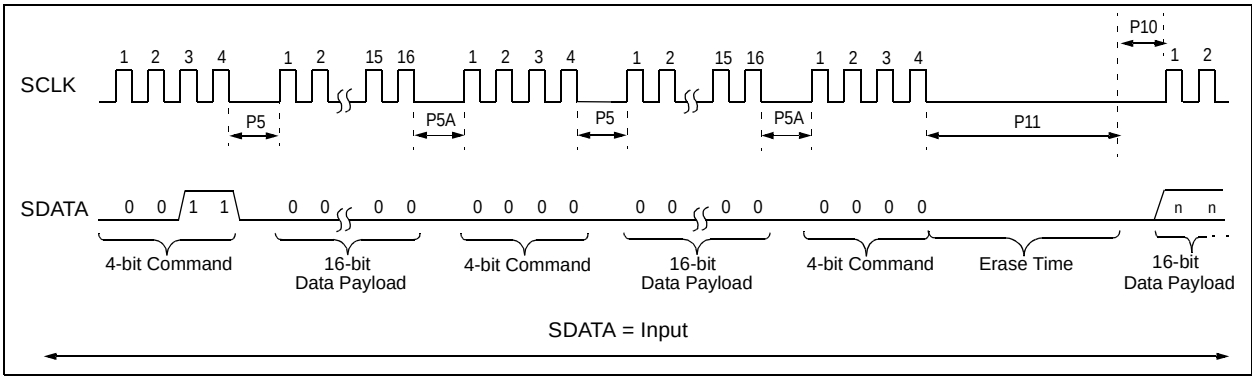
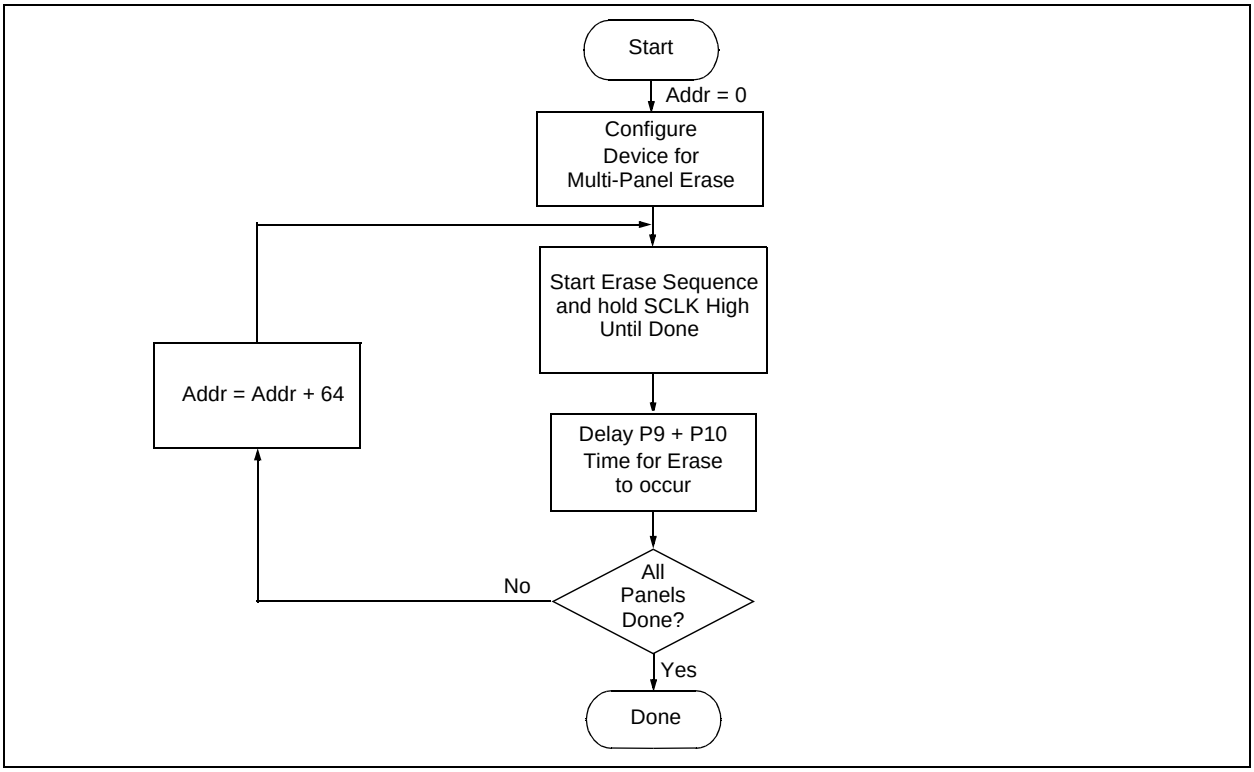


TABLE 3-2: ERASE CODE MEMORY CODE SEQUENCE

4-Bit Command	Data Payload	Core Instruction
Step 1: Direct access to config memory.		
0000	8E A6	BSF EECON1, EEPGD
0000	8C A6	BSF EECON1, CFGS
0000	86 A6	BSF EECON1, WREN
Step 2: Configure device for multi-panel writes.		
0000	0E 3C	MOVLW 3Ch
0000	6E F8	MOVWF TBLPTRU
0000	0E 00	MOVLW 00h
0000	6E F7	MOVWF TBLPTRH
0000	0E 06	MOVLW 06h
0000	6E F6	MOVWF TBLPTRL
1100	00 40	Write 40h to 3C0006h to enable multi-panel erase.
Step 3: Direct access to code memory and enable erase.		
0000	8E A6	BSF EECON1, EEPGD
0000	9C A6	BCF EECON1, CFGS
0000	88 A6	BSF EECON1, FREE
0000	6A F8	CLRF TBLPTRU
0000	6A F7	CLRF TBLPTRH
0000	6A F6	CLRF TBLPTRL
Step 4: Erase single row of all panels at an offset.		
1111	<DummyLSB> <DummyMSB>	Write 2 dummy bytes and start programming.
0000	00 00	NOP - hold SCLK high for time P9.
Step 5: Repeat step 4, with Address Pointer incremented by 64 until all panels are erased.		

FIGURE 3-4: MULTI-PANEL SINGLE ROW ERASE CODE MEMORY FLOW



3.2 Code Memory Programming

Programming code memory is accomplished by first loading data into the appropriate write buffers and then initiating a programming sequence. Each panel in the code memory space (see Figure 2-2) has an 8-byte deep write buffer that must be loaded prior to initiating a write sequence. The actual memory write sequence takes the contents of these buffers and programs the associated EEPROM code memory.

Typically, all of the program buffers are written in parallel (Multi-Panel Write mode). In other words, in the case of a 128-Kbyte device (16 panels with an 8-byte buffer per panel), 128 bytes will be simultaneously programmed during each programming sequence. In this case, the offset of the write within each panel is the same (see Figure 3-5). Multi-Panel Write mode is enabled by appropriately configuring the Programming Control register located at 3C0006h.

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

After SCLK is brought low, the programming sequence is terminated. SCLK 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 PIC18FXX20 device is shown in Figure 3-3. The flowchart shown in Figure 3-6 depicts the logic necessary to completely write a PIC18FXX20 device. The timing diagram that details the “Start Programming” command, and parameters P9 and P10, is shown in Figure 3-7.

Note:	The TBLPTR register must contain the same offset value when initiating the programming sequence as it did when the write buffers were loaded.

© 2010 Microchip Technology Inc.

TABLE 3-3: WRITE CODE MEMORY CODE SEQUENCE

4-Bit Command	Data Payload	Core Instruction
Step 1: Direct access to config memory.		
0000	8E A6	BSF EECON1, EEPGD
0000	8C A6	BSF EECON1, CFGS
0000	86 A6	BSF EECON1, WREN
Step 2: Configure device for multi-panel writes.		
0000	0E 3C	MOVLW 3Ch
0000	6E F8	MOVWF TBLPTRU
0000	0E 00	MOVLW 00h
0000	6E F7	MOVWF TBLPTRH
0000	0E 06	MOVLW 06h
0000	6E F6	MOVWF TBLPTRL
1100	00 40	Write 40h to 3C0006h to enable multi-panel writes.
Step 3: Direct access to code memory.		
0000	8E A6	BSF EECON1, EEPGD
0000	9C A6	BCF EECON1, CFGS
Step 4: Load write buffer for Panel 1.		
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
1101	<LSB><MSB>	Write 2 bytes and post-increment address by 2
1101	<LSB><MSB>	Write 2 bytes and post-increment address by 2
1101	<LSB><MSB>	Write 2 bytes and post-increment address by 2
1100	<LSB><MSB>	Write 2 bytes
Step 5: Repeat for Panel 2.		
Step 6: Repeat for all but the last panel (N – 1).		
Step 7: Load write buffer for last panel.		
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
1101	<LSB><MSB>	Write 2 bytes and post-increment address by 2
1101	<LSB><MSB>	Write 2 bytes and post-increment address by 2
1101	<LSB><MSB>	Write 2 bytes and post-increment address by 2
1111	<LSB><MSB>	Write 2 bytes and start programming
0000	00 00	NOP - hold SCLK high for time P9
To continue writing data, repeat steps 2 through 5, where the Address Pointer is incremented by 8 in each panel at each iteration of the loop.		

© 2010 Microchip Technology Inc.

FIGURE 3-6: PROGRAM CODE MEMORY FLOW

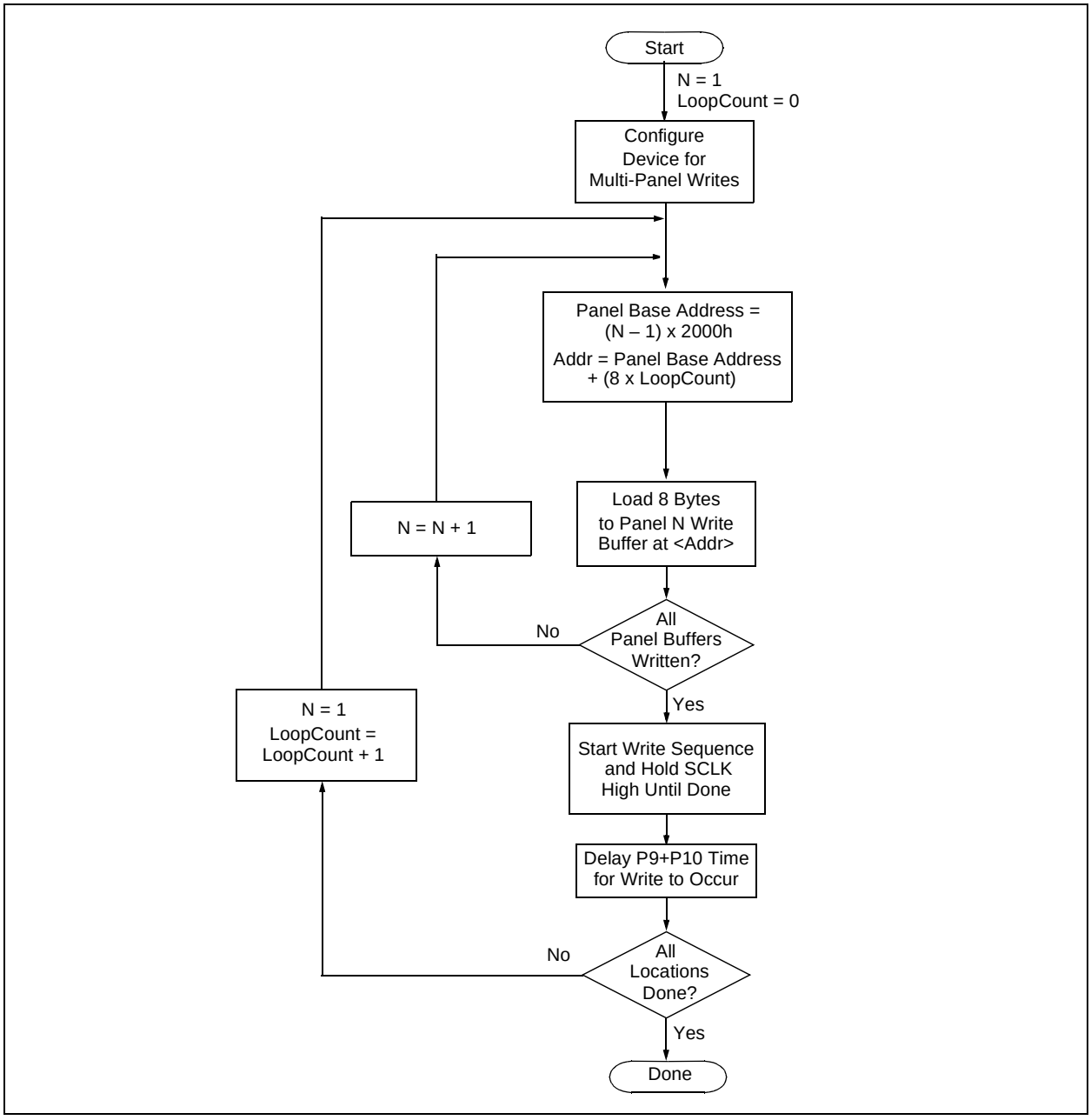
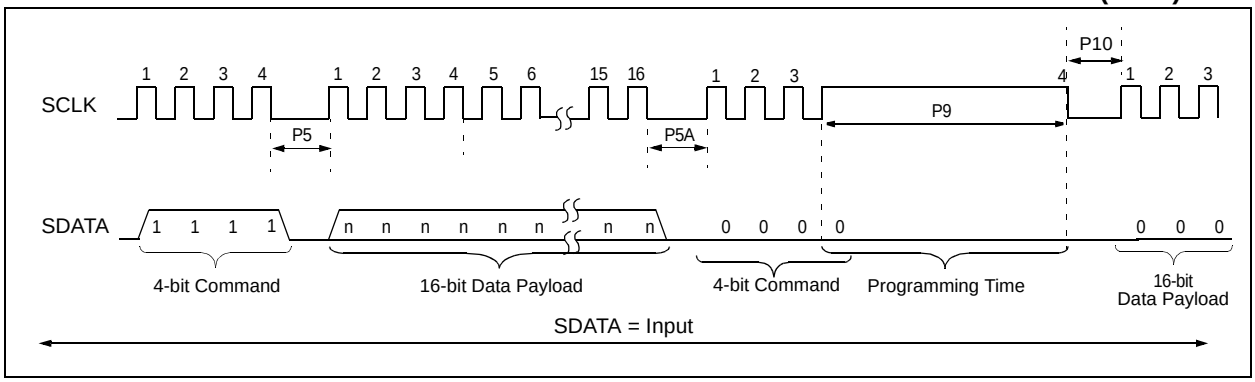


FIGURE 3-7: TABLE WRITE AND START PROGRAMMING INSTRUCTION TIMING (1111)



3.3 Data EEPROM Programming

Data EEPROM is accessed one byte at a time via an Address Pointer (register pair EEADR:EEADRH) and a data latch (EEDATA). Data EEPROM is written by loading EEADR:EEADRH with the desired memory location, EEDATA with the data to be written, and initiating a memory write by appropriately configuring the EECON1 and EECON2 registers. 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). It is strongly recommended that the WREN bit be set only when absolutely necessary.

To help prevent inadvertent writes when using the EECON1 register, EECON2 is used to “enable” the WR bit. This register must be sequentially loaded with 55h and then AAh, immediately prior to asserting the WR bit in order for the write to occur.

The write begins on the falling edge of the 4th SCLK after the WR bit is set. It ends when the WR bit is cleared by hardware.

After the programming sequence terminates, SCLK must still be held low for the time specified by parameter P10 to allow high voltage discharge of the memory array.

FIGURE 3-8: PROGRAM DATA FLOW

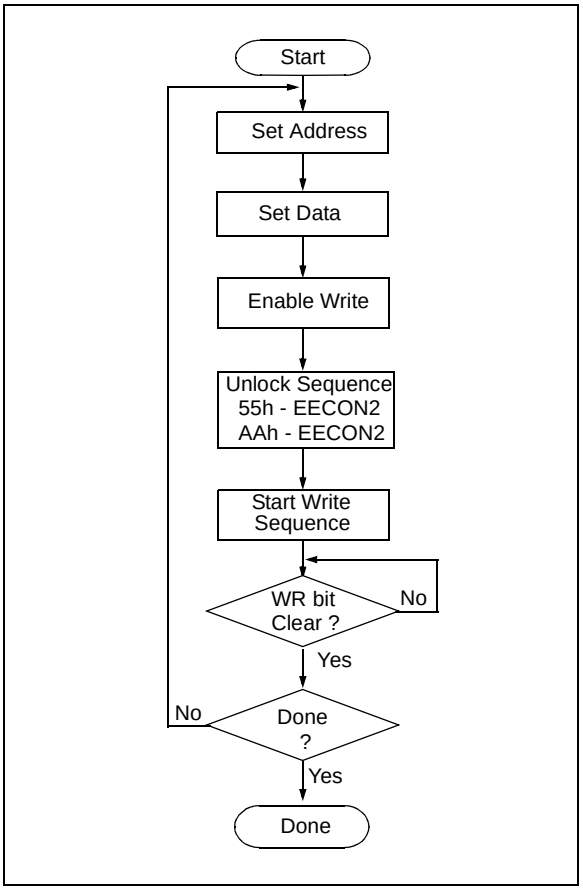
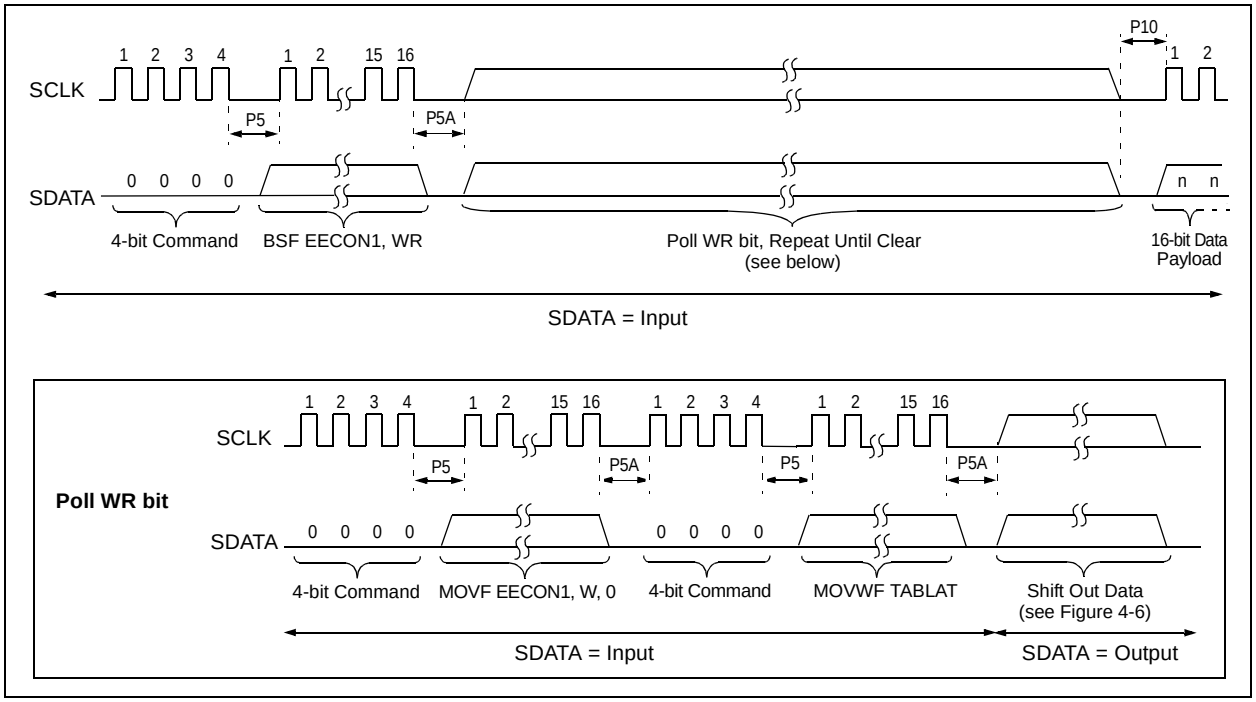


FIGURE 3-9: DATA EEPROM WRITE TIMING



© 2010 Microchip Technology Inc.

TABLE 3-5: PROGRAMMING DATA 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: Load the data to be written.		
0000	0E <Data>	MOVLW <Data>
0000	6E A8	MOVWF EEDATA
Step 4: Enable memory writes.		
0000	84 A6	BSF EECON1, WREN
Step 5: Perform required sequence.		
0000	0E 55	MOVLW 0X55
0000	6E A7	MOVWF EECON2
0000	0E AA	MOVLW 0XAA
0000	6E A7	MOVWF EECON2
Step 6: Initiate write.		
0000	82 A6	BSF EECON1, WR
Step 7: Poll WR bit, repeat until the bit is clear.		
0000	50 A6	MOVF EECON1, W, 0
0000	6E F5	MOVWF TABLAT
0010	<LSB><MSB>	Shift out data(1)
Step 8: Disable writes.		
0000	94 A6	BCF EECON1, WREN
Repeat steps 2 through 8 to write more data.		

Note 1: See Figure 4-4 for details on Shift Out Data timing.

3.4 ID Location Programming

The ID Locations are programmed much like the code memory, except that multi-panel writes must be disabled. The single panel that will be written will automatically be enabled, based on the value of the Table Pointer. The ID registers are mapped in addresses 200000h through 200007h. These locations read out normally, even after code protection.

Note:	Even though multi-panel writes are disabled, the user must still fill the 8-byte data buffer for the panel.

Figure 3-6 demonstrates the code sequence required to write the ID locations.

TABLE 3-6: WRITE ID SEQUENCE

4-Bit Command	Data Payload	Core Instruction
Step 1: Direct access to config memory.		
0000	8E A6	BSF EECON1, EEPGD
0000	8C A6	BSF EECON1, CFGS
Step 2: Configure device for single panel writes.		
0000	0E 3C	MOVLW 3Ch
0000	6E F8	MOVWF TBLPTRU
0000	0E 00	MOVLW 00h
0000	6E F7	MOVWF TBLPTRH
0000	0E 06	MOVLW 06h
0000	6E F6	MOVWF TBLPTRL
1100	00 00	Write 00h to 3C0006h to enable single panel writes.
Step 3: Direct access to code memory.		
0000	8E A6	BSF EECON1, EEPGD
0000	9C A6	BCF EECON1, CFGS
Step 4: Load write buffer. Panel will be automatically determined by address.		
0000	0E 20	MOVLW 20h
0000	6E F8	MOVWF TBLPTRU
0000	0E 00	MOVLW 00h
0000	6E F7	MOVWF TBLPTRH
0000	0E 00	MOVLW 00h
0000	6E F6	MOVWF TBLPTRL
1101	<LSB><MSB>	Write 2 bytes and post-increment address by 2
1101	<LSB><MSB>	Write 2 bytes and post-increment address by 2
1101	<LSB><MSB>	Write 2 bytes and post-increment address by 2
1111	<LSB><MSB>	Write 2 bytes and start programming
0000	00 00	NOP - hold SCLK high for time P9

In order to modify the ID locations, refer to the methodology described in Section 3.2.2, “Modifying Code Memory”. As with code memory, the ID locations must be erased before modified.

© 2010 Microchip Technology Inc.

3.5 Boot Block Programming

The Boot Block segment is programmed in exactly the same manner as the ID locations (see Section 3.4). Multi-panel writes must be disabled so that only addresses in the range 0000h to 01FFh will be written.

The code sequence detailed in Figure 3-6 should be used, except that the address data used in “Step 2” will be in the range 000000h to 0001FFh.

3.6 Configuration Bits Programming

Unlike code memory, the configuration bits are programmed a byte at a time. The “Table Write, Begin Programming” 4-bit command (1111) is used, but only 8 bits of the following 16-bit payload will be written. The LSB of the payload will be written to even addresses, and the MSB will be written to odd addresses. The code sequence to program two consecutive configuration locations is shown in Figure 3-7.

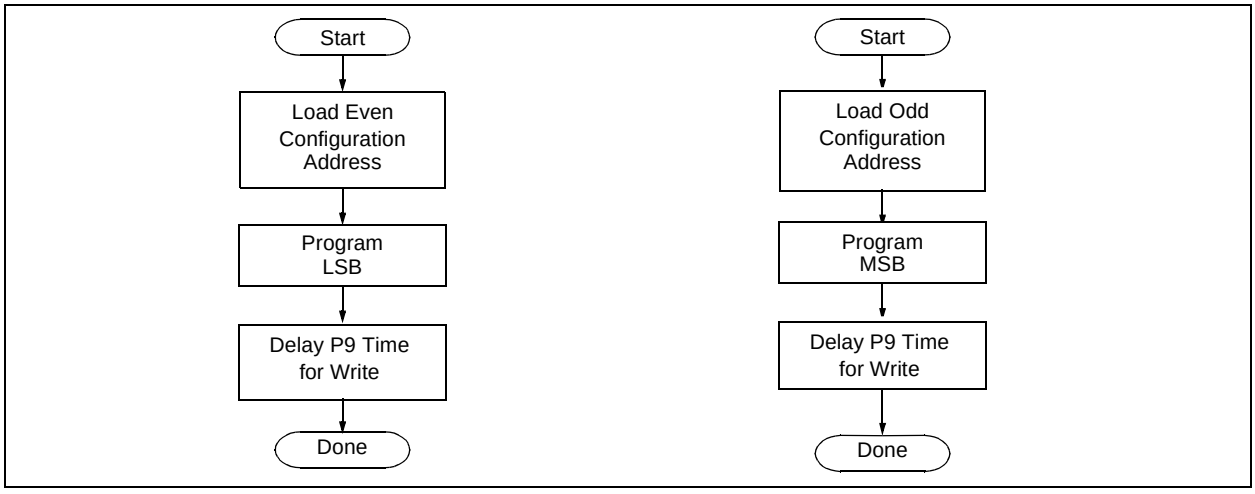
TABLE 3-7: SET ADDRESS POINTER TO CONFIGURATION LOCATION

4-Bit Command	Data Payload	Core Instruction
Step 1: Direct access to config memory.		
0000	8E A6	BSF EECON1, EEPGD
0000	8C A6	BSF EECON1, CFGS
Step 2: Position the program counter(1).		
0000	EF 00	GOTO 100000h
0000	F8 00	
Step 3(2): Set Table Pointer for config byte to be written. Write even/odd addresses.		
0000	0E 30	MOVLW 30h
0000	6E F8	MOVWF TBLPTRU
0000	0E 00	MOVLW 00h
0000	6E F7	MOVWF TBLPRTH
0000	0E 00	MOVLW 00h
0000	6E F6	MOVWF TBLPTRL
1111	<LSB><MSB ignored>	Load 2 bytes and start programming
0000	00 00	NOP - hold SCLK high for time P9
0000	2A F6	INCF TBLPTRL
1111	<LSB ignored><MSB>	Load 2 bytes and start programming
0000	00 00	NOP - hold SCLK high for time P9

Note 1: If the code protection bits are programmed while the program counter resides in the same block, then the interaction of code protection logic may prevent further table write. To avoid this situation, move the program counter outside the code protection area (e.g., GOTO 100000h).

2: Enabling the write protection of configuration bits (WRTC = 0 in CONFIG6H) will prevent further writing of configuration bits. Always write all the configuration bits before enabling the write protection for configuration bits.

FIGURE 3-10: CONFIGURATION PROGRAMMING FLOW



4.3 Verify Configuration Bits

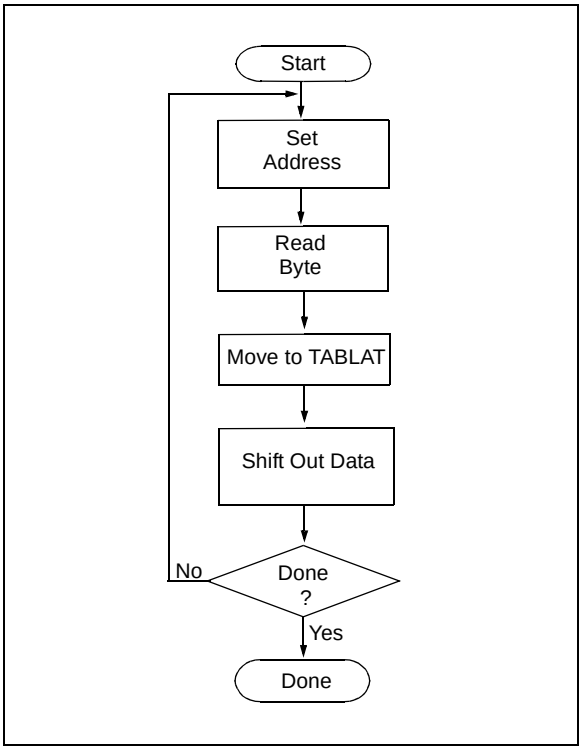
A configuration address may be read and output on SDATA via the 4-bit command, '1001'. Configuration data is read and written in a byte-wise fashion, so it is not necessary to merge two bytes into a word prior to a compare. The result may then be immediately compared to the appropriate configuration data in the programmer's memory for verification. Refer to Section 4.1 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 EEADR:EEADRH) and a data latch (EEDATA). Data EEPROM is read by loading EEADR:EEADRH 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 SDATA 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 SCLK of the operand to allow SDATA to transition from an input to an output. During this time, SCLK must be held low (see Figure 4-4).

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

FIGURE 4-3: READ DATA EEPROM FLOW



© 2010 Microchip Technology Inc.

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 SDATA via the 4-bit command, '0010' (Shift Out Data Holding register). The result may then be immediately compared to the appropriate data in the programmer's memory for verification. Refer to Section 4.4 for implementation details of reading data EEPROM.

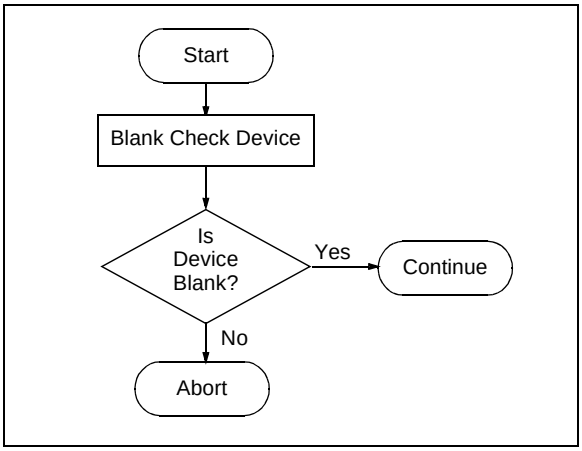
4.6 Blank Check

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'. So, "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 5-2 for blank configuration expect data for the various PIC18FXX20 devices.

Given that "Blank Checking" is merely code and data EEPROM verification with FFh expect data, refer to Section 4.4 and Section 4.2 for implementation details.

FIGURE 4-5: BLANK CHECK FLOW



© 2010 Microchip Technology Inc.

5.0 CONFIGURATION WORD

The PIC18FXX20 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.

5.1 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 0Fh. In doing so, if the user code inadvertently tries to execute from the ID space, the ID data will execute as NOP.

5.2 Device ID Word

The device ID word for the PIC18FXX20 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.

5.3 Low Voltage Programming (LVP) Bit

The LVP bit in Configuration register, CONFIG4L, enables low voltage ICSP programming. The LVP bit defaults to a '1' from the factory.

If Low Voltage 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 $\overline{MCLR}/V_{PP}$ is raised to V_{IHH}. 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 normal ICSP mode is always available, regardless of the state of the LVP bit, by applying V_{IHH} to the $\overline{MCLR}/V_{PP}$ pin.

2: While in low voltage ICSP mode, the RB5 pin can no longer be used as a general purpose I/O.

TABLE 5-1: DEVICE ID VALUES

Device	Device ID Value	
	DEVID2	DEVID1
PIC18F6520	0Bh	001x xxxx
PIC18F6620	06h	011x xxxx
PIC18F6720	06h	001x xxxx
PIC18F8520	0Bh	000x xxxx
PIC18F8620	06h	010x xxxx
PIC18F8720	06h	000x xxxx

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

TABLE 5-3: PIC18FXX20 CONFIGURATION BIT DESCRIPTIONS

Bit Name	Configuration Words	Description
$\overline{\text{OSCEN}}$	CONFIG1H	Low Power System Clock Option (Timer1) Enable bit 1 = Disabled 0 = Timer1 oscillator system clock option enabled
FOSC2:FOSC0	CONFIG1H	Oscillator Selection bits 111 = RC oscillator w/ OSC2 configured as RA6 110 = HS oscillator w/ PLL enabled 101 = EC oscillator w/ OSC2 configured as RA6 100 = RC oscillator w/ OSC2 configured as “divide by 4 clock output” 011 = RC oscillator 010 = HS oscillator 001 = XT oscillator 000 = LP oscillator
BORV1:BORV0	CONFIG2L	Brown-out Reset Voltage bits 11 = V_{BOR} set to 2.0V 10 = V_{BOR} set to 2.7V 01 = V_{BOR} set to 4.2V 00 = V_{BOR} set to 4.5V
BOREN	CONFIG2L	Brown-out Reset Enable bit 1 = Brown-out Reset enabled 0 = Brown-out Reset disabled
$\overline{\text{PWRTEN}}$	CONFIG2L	Power-up Timer Enable bit 1 = PWRT disabled 0 = PWRT enabled
WDTPS2:WDTPS0	CONFIG2H	Watchdog Timer Postscaler Select bits 111 = 1:128 110 = 1:64 101 = 1:32 100 = 1:16 011 = 1:8 010 = 1:4 001 = 1:2 000 = 1:1
WDTEN	CONFIG2H	Watchdog Timer Enable bit 1 = WDT enabled 0 = WDT disabled (control is placed on SWDTEN bit)
WAIT(1)	CONFIG3L	External Bus Data Wait Enable bit 1 = Wait selections unavailable 0 = Wait selections determined by WAIT1:WAIT0 bits of MEMCOM register
PM1:PM0(1)	CONFIG3L	Processor Mode Select bits 11 = Microcontroller mode 10 = Microprocessor mode 01 = Microprocessor with Boot Block mode 00 = Extended Microcontroller mode

Note 1: Unimplemented in PIC18F6X20 (64-pin) devices; maintain this bit set.

2: Unimplemented in PIC18FX620 devices; maintain this bit set.

3: PIC18F8520/8620 devices only.

TABLE 5-3: PIC18FXX20 CONFIGURATION BIT DESCRIPTIONS (CONTINUED)

Bit Name	Configuration Words	Description
T1OSCMX(3)	CONFIG3H	Timer1 Oscillator MUX bit 1 = Legacy Timer1 oscillator selected 0 = Low power Timer1 oscillator selected
CCP2MX	CONFIG3H	CCP2 MUX bit 1 = CCP2 input/output is multiplexed with RC1 0 = CCP2 input/output is multiplexed with RB3
$\overline{\text{DEBUG}}$	CONFIG4L	Background Debugger Enable bit 1 = Background debugger disabled 0 = Background debugger enabled
LVP	CONFIG4L	Low Voltage Programming Enable bit 1 = Low voltage programming enabled 0 = Low voltage programming disabled
STVREN	CONFIG4L	Stack Overflow/Underflow Reset Enable bit 1 = Stack overflow/underflow will cause RESET 0 = Stack overflow/underflow will not cause RESET
CP0	CONFIG5L	Code Protection bits (Block 0) 1 = Code memory not code protected 0 = Code memory code protected
CP1	CONFIG5L	Code Protection bits (Block 1) 1 = Code memory not code protected 0 = Code memory code protected
CP2	CONFIG5L	Code Protection bits (Block 2) 1 = Code memory not code protected 0 = Code memory code protected
CP3	CONFIG5L	Code Protection bits (Block 3) 1 = Code memory not code protected 0 = Code memory code protected
CP4(2)	CONFIG5L	Code Protection bits (Block 4) 1 = Code memory not code protected 0 = Code memory code protected
CP5(2)	CONFIG5L	Code Protection bits (Block 5) 1 = Code memory not code protected 0 = Code memory code protected
CP6(2)	CONFIG5L	Code Protection bits (Block 6) 1 = Code memory not code protected 0 = Code memory code protected
CP7(2)	CONFIG5L	Code Protection bits (Block 7) 1 = Code memory not code protected 0 = Code memory code protected

Note 1: Unimplemented in PIC18F6X20 (64-pin) devices; maintain this bit set.

2: Unimplemented in PIC18FX620 devices; maintain this bit set.

3: PIC18F8520/8620 devices only.

© 2010 Microchip Technology Inc.

TABLE 5-3: PIC18FXX20 CONFIGURATION BIT DESCRIPTIONS (CONTINUED)

Bit Name	Configuration Words	Description
CPD	CONFIG5H	Code Protection bits (Data EEPROM) 1 = Data EEPROM not code protected 0 = Data EEPROM code protected
CPB	CONFIG5H	Code Protection bits (Boot Block) 1 = Boot block not code protected 0 = Boot block code protected
WRT0	CONFIG6L	Table Write Protection bit (Block 0) 1 = Code memory not write protected 0 = Code memory write protected
WRT1	CONFIG6L	Table Write Protection bit (Block 1) 1 = Code memory not write protected 0 = Code memory write protected
WRT2	CONFIG6L	Table Write Protection bit (Block 2) 1 = Code memory not write protected 0 = Code memory write protected
WRT3	CONFIG6L	Table Write Protection bit (Block 3) 1 = Code memory not write protected 0 = Code memory write protected
WRT4(2)	CONFIG6L	Table Write Protection bit (Block 4) 1 = Code memory not write protected 0 = Code memory write protected
WRT5(2)	CONFIG6L	Table Write Protection bit (Block 5) 1 = Code memory not write protected 0 = Code memory write protected
WRT6(2)	CONFIG6L	Table Write Protection bit (Block 6) 1 = Code memory not write protected 0 = Code memory write protected
WRT7(2)	CONFIG6L	Table Write Protection bit (Block 7) 1 = Code memory not write protected 0 = Code memory write protected
WRTD	CONFIG6H	Table Write Protection bit (Data EEPROM) 1 = Data EEPROM not write protected 0 = Data EEPROM write protected
WRTB	CONFIG6H	Table Write Protection bit (Boot Block) 1 = Boot block not write protected 0 = Boot block write protected
WRTC	CONFIG6H	Table Write Protection bit (Configuration registers) 1 = Configuration registers not write protected 0 = Configuration registers write protected

Note 1: Unimplemented in PIC18F6X20 (64-pin) devices; maintain this bit set.

2: Unimplemented in PIC18FX620 devices; maintain this bit set.

3: PIC18F8520/8620 devices only.

TABLE 5-4: CHECKSUM COMPUTATION (CONTINUED)

Device	Code Protect	Checksum	Blank Value	0xAA at 0 and Max Address
PIC18F6720	None	SUM(0000:01FF)+SUM(0200:3FFF)+SUM(4000:7FFF)+ SUM(8000:BFFF)+SUM(C000:FFFF)+SUM(10000:13FFF)+ SUM(14000:17FFF)+SUM(18000:1BFFF)+SUM(1C000:1FFFF)+ (CFGW1L & 0000)+(CFGW1H & 0027)+(CFGW2L & 000F)+ (CFGW2H & 000F)+(CFGW3L & 0000)+(CFGW3H & 0001)+ (CFGW4L & 0085)+(CFGW4H & 0000)+(CFGW5L & 00FF)+ (CFGW5H & 00C0)+(CFGW6L & 00FF)+(CFGW6H & 00E0)+ (CFGW7L & 00FF)+(CFGW7H & 0040)	05A8	04FE
	Boot Block	SUM(0200:3FFF)+SUM(4000:7FFF)+SUM(8000:BFFF)+ SUM(C000:FFFF)+SUM(10000:13FFF)+SUM(14000:17FFF)+ SUM(18000:1BFFF)+SUM(1C000:1FFFF)+(CFGW1L & 0000)+ (CFGW1H & 0027)+(CFGW2L & 000F)+(CFGW2H & 000F)+ (CFGW3L & 0000)+(CFGW3H & 0001)+(CFGW4L & 0085)+ (CFGW4H & 0000)+(CFGW5L & 00FF)+(CFGW5H & 00C0)+ (CFGW6L & 00FF)+(CFGW6H & 00E0)+(CFGW7L & 00FF)+ (CFGW7H & 0040)+SUM(IDs)	077F	0734
	Boot/ Block1/ Block2	SUM(8000:BFFF)+SUM(C000:FFFF)+SUM(10000:13FFF)+ SUM(14000:17FFF)+SUM(18000:1BFFF)+SUM(1C000:1FFFF)+ (CFGW1L & 0000)+(CFGW1H & 0027)+(CFGW2L & 000F)+ (CFGW2H & 000F)+(CFGW3L & 0000)+(CFGW3H & 0001)+ (CFGW4L & 0085)+(CFGW4H & 0000)+(CFGW5L & 00FF)+ (CFGW5H & 00C0)+(CFGW6L & 00FF)+(CFGW6H & 00E0)+ (CFGW7L & 00FF)+(CFGW7H & 0040)+SUM(IDs)	857C	8531
	All	(CFGW1L & 0000)+(CFGW1H & 0027)+(CFGW2L & 000F)+ (CFGW2H & 000F)+(CFGW3L & 0000)+(CFGW3H & 0001)+ (CFGW4L & 0085)+(CFGW4H & 0000)+(CFGW5L & 00FF)+ (CFGW5H & 00C0)+(CFGW6L & 00FF)+(CFGW6H & 00E0)+ (CFGW7L & 00FF)+(CFGW7H & 0040)+SUM(IDs)	480	048A

Legend:

Item		Description
CFGW	=	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

© 2010 Microchip Technology Inc.

TABLE 5-4: CHECKSUM COMPUTATION (CONTINUED)

Device	Code Protect	Checksum	Blank Value	0xAA at 0 and Max Address
PIC18F8520	None	SUM(0000:07FF)+SUM(0800:1FFF)+SUM(2000:3FFF)+ SUM(4000:5FFF)+SUM(6000:7FFF)+(CFGW1L & 0000)+ (CFGW1H & 0027)+(CFGW2L & 000F)+(CFGW2H & 000F)+ (CFGW3L & 0000)+(CFGW3H & 0002)+(CFGW4L & 0085)+ (CFGW4H & 0000)+(CFGW5L & 00FF)+(CFGW5H & 00C0)+ (CFGW6L & 00FF)+(CFGW6H & 00E0)+(CFGW7L & 00FF)+ (CFGW7H & 0040)	05AA	500
	Boot Block	SUM(0800:1FFF)+SUM(2000:3FFF)+SUM(4000:5FFF)+ SUM(6000:7FFF)+(CFGW1L & 0000)+(CFGW1H & 0027)+ (CFGW2L & 000F)+(CFGW2H & 000F)+(CFGW3L & 0000)+ (CFGW3H & 0002)+(CFGW4L & 0085)+(CFGW4H & 0000)+ (CFGW5L & 00FF)+(CFGW5H & 00C0)+(CFGW6L & 00FF)+ (CFGW6H & 00E0)+(CFGW7L & 00FF)+(CFGW7H & 0040)+ SUM(IDs)	783	071A
	Boot/ Block1/ Block2	SUM(4000:5FFF)+SUM(6000:7FFF)+(CFGW1L & 0000)+ (CFGW1H & 0027)+(CFGW2L & 000F)+(CFGW2H & 000F)+ (CFGW3L & 0000)+(CFGW3H & 0002)+(CFGW4L & 0085)+ (CFGW4H & 0000)+(CFGW5L & 00FF)+(CFGW5H & 00C0)+ (CFGW6L & 00FF)+(CFGW6H & 00E0)+(CFGW7L & 00FF)+ (CFGW7H & 0040)+SUM(IDs)	8580	8517
	All	(CFGW1L & 0000)+(CFGW1H & 0027)+(CFGW2L & 000F)+ (CFGW2H & 000F)+(CFGW3L & 0000)+(CFGW3H & 0002)+ (CFGW4L & 0085)+(CFGW4H & 0000)+(CFGW5L & 00FF)+ (CFGW5H & 00C0)+(CFGW6L & 00FF)+(CFGW6H & 00E0)+ (CFGW7L & 00FF)+(CFGW7H & 0040)+SUM(IDs)	484	470

Legend:

Item		Description
CFGW	=	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	VIHH	High Voltage Programming Voltage on MCLR/VPP	9.00	13.25	V	
D110A	VIHL	Low Voltage Programming Voltage on MCLR/VPP	2.00	5.50	V	
D111	VDD	Supply Voltage During Programming	2.00	5.50	V	Normal programming
			4.50	5.50	V	Bulk erase operations
D112	IPP	Programming Current on MCLR/VPP	—	300	µA	
D113	IDDP	Supply Current During Programming	—	10	mA	
D031	VIL	Input Low Voltage	VSS	0.2 VDD	V	
D041	VIH	Input High Voltage	0.8 VDD	VDD	V	
D080	VOL	Output Low Voltage	—	0.6	V	IOL = 8.5 mA @ 4.5V
D090	VOH	Output High Voltage	VDD – 0.7	—	V	IOH = -3.0 mA @ 4.5V
D012	CIO	Capacitive Loading on I/O pin (SDATA)	—	50	pF	To meet AC specifications
P1	TR	MCLR/VPP Rise Time to enter Program/Verify mode	—	1.0	µs	**(Note 1)**
P2	Tsclk	Serial Clock (SCLK) Period	100	—	ns	
P2A	TsclkL	Serial Clock (SCLK) Low Time	40	—	ns	
P2B	TsclkH	Serial Clock (SCLK) High Time	40	—	ns	
P3	Tset1	Input Data Setup Time to Serial Clock ↓	15	—	ns	
P4	Thld1	Input Data Hold Time from SCLK ↓	15	—	ns	
P5	Tdly1	Delay between 4-bit Command and Command Operand	40	—	ns	
P5A	Tdly1a	Delay between 4-bit Command Operand and next 4-bit Command	40	—	ns	
P6	Tdly2	Delay between Last SCLK ↓ of Command Byte to First SCLK ↑ of Read of Data Word	20	—	ns	
P9	Tdly5	SCLK High Time (minimum programming time)	1	—	ms	
P10	Tdly6	SCLK Low Time after Programming (high voltage discharge time)	5	—	µs	
P11	Tdly7	Delay to allow Self-Timed Data Write or Bulk Erase to occur	10	—	ms	
P11A	Tdrwt	Data Write Polling Time	4	—	ms	
P12	Thld2	Input Data Hold Time from MCLR/VPP ↑	2	—	µs	
P13	Tset2	VDD ↑ Setup Time to MCLR/VPP ↑	100	—	ns	
P14	Tvalid	Data Out Valid from SCLK ↑	10	—	ns	
P15	Tset3	PGM ↑ Setup Time to MCLR/VPP ↑	2	—	µs	

Note 1: Do not allow excess time when transitioning MCLR between VIL and VIHH; this can cause spurious program executions to occur. The maximum transition time is:
1 TCY + TPWRT (if enabled) + 1024 TOSC (for LP, HS, HS/PLL, and XT modes only)
+ 2 ms (for HS/PLL mode only) + 1.5 µs (for EC mode only)
where TCY is the Instruction Cycle Time, TPWRT is the Power-up Timer Period, and TOSC is the Oscillator Period.

For specific values, refer to the Electrical Characteristics section of the Device Data Sheet for the particular device.

© 2010 Microchip Technology Inc.

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE**.** Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

Trademarks

The Microchip name and logo, the Microchip logo, dsPIC, KEELOQ, KEELOQ logo, MPLAB, PIC, PICmicro, PICSTART, rfPIC and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

FilterLab, Hampshire, HI-TECH C, Linear Active Thermistor, MXDEV, MXLAB, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, HI-TIDE, In-Circuit Serial Programming, ICSP, Mindi, MiWi, MPASM, MPLAB Certified logo, MPLIB, MPLINK, mTouch, Octopus, Omniscient Code Generation, PICC, PICC-18, PICDEM, PICDEM.net, PICkit, PICtail, PIC^{32} logo, REAL ICE, rfLAB, Select Mode, Total Endurance, TSHARC, UniWinDriver, WiperLock and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.

© 2010, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

Printed on recycled paper.

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