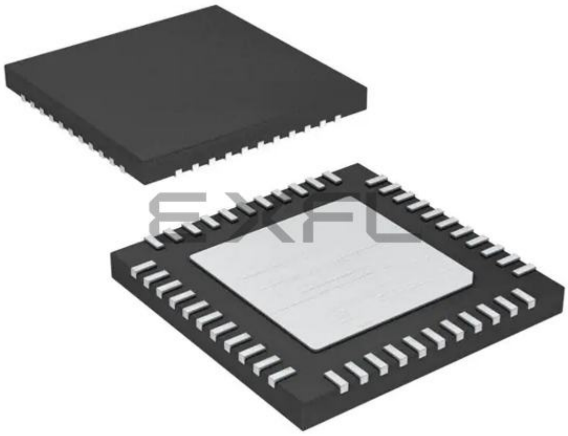


Welcome to E-XFL.COM

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.

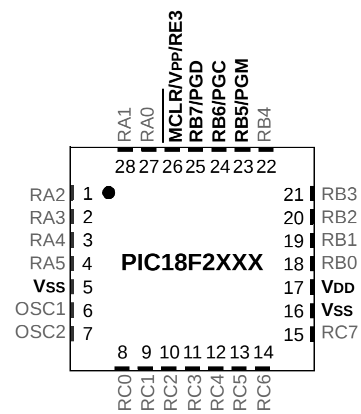
Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	PIC
Core Size	8-Bit
Speed	40MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, HLVD, POR, PWM, WDT
Number of I/O	36
Program Memory Size	64KB (32K x 16)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	3.8K x 8
Voltage - Supply (Vcc/Vdd)	4.2V ~ 5.5V
Data Converters	A/D 13x10b
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/pic18f4610t-i-ml

The following devices are included in 28-pin QFN parts:

- PIC18F2221
- PIC18F2321
- PIC18F2410
- PIC18F2420
- PIC18F2423
- PIC18F2450
- PIC18F2480
- PIC18F2510
- PIC18F2520
- PIC18F2523
- PIC18F2580
- PIC18F2682
- PIC18F2685

FIGURE 2-2: 28-Pin QFN



The following devices are included in 40-pin PDIP parts:

- PIC18F4221
- PIC18F4321
- PIC18F4410
- PIC18F4420
- PIC18F4423
- PIC18F4450
- PIC18F4455
- PIC18F4458
- PIC18F4480
- PIC18F4510
- PIC18F4515
- PIC18F4520
- PIC18F4523
- PIC18F4525
- PIC18F4550
- PIC18F4553
- PIC18F4580
- PIC18F4585
- PIC18F4610
- PIC18F4620
- PIC18F4680
- PIC18F4682
- PIC18F4685

FIGURE 2-3: 40-Pin PDIP

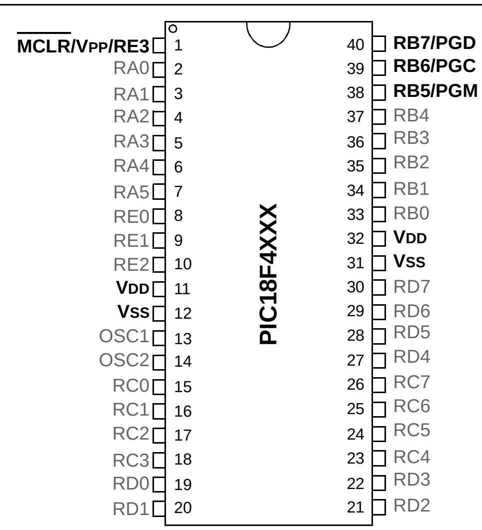
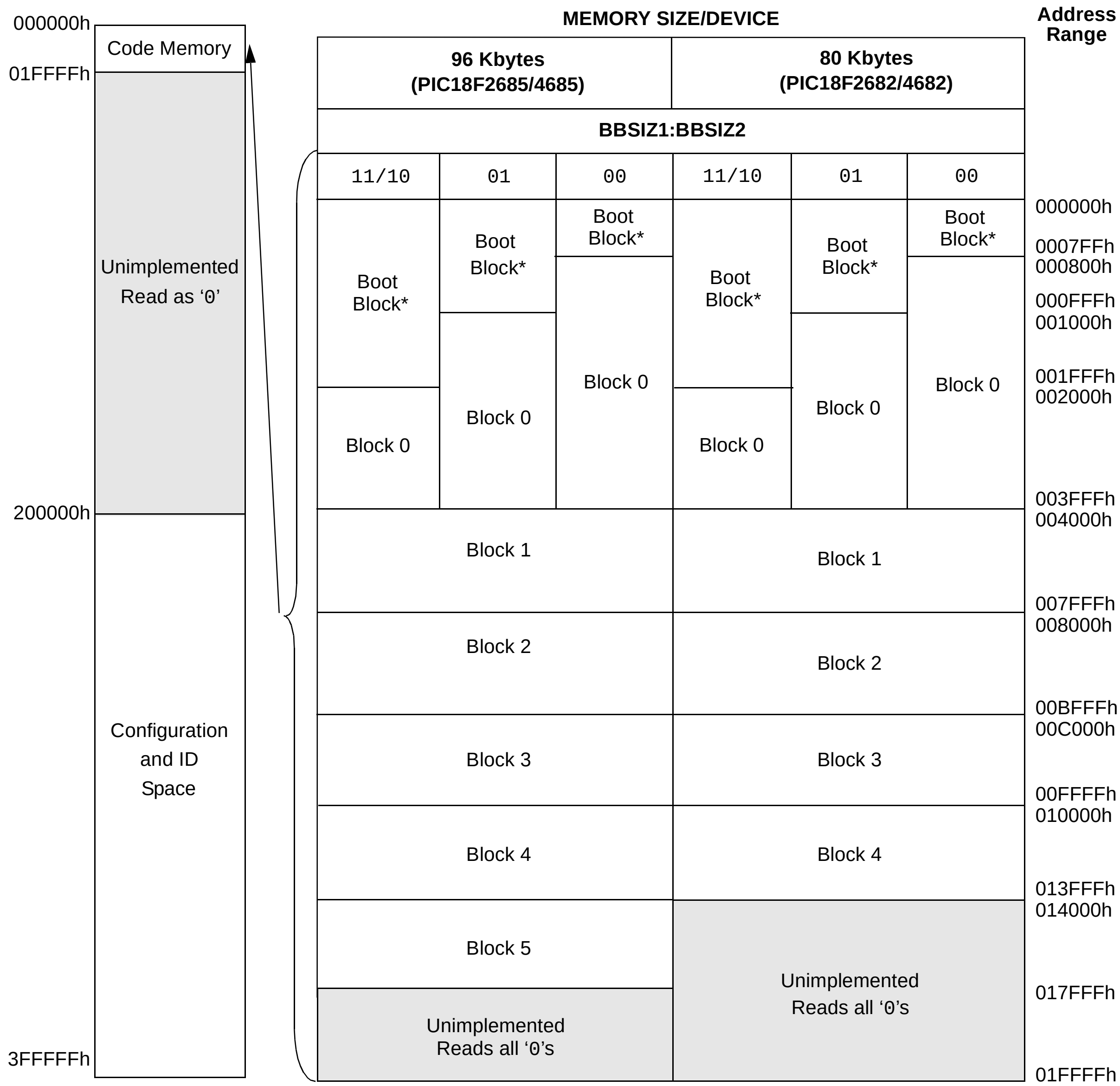


FIGURE 2-7: MEMORY MAP AND THE CODE MEMORY SPACE FOR PIC18F2685/4685 AND PIC18F2682/4682 DEVICES



Note: Sizes of memory areas are not to scale.
* Boot Block size is determined by the BBSIZ<1:2> bits in the CONFIG4L register.

For PIC18FX5X0/X5X3 devices, the code memory space extends from 000000h to 007FFFh (32 Kbytes) in four 8-Kbyte blocks. For PIC18FX4X5/X4X8 devices, the code memory space extends from 000000h to 005FFFh (24 Kbytes) in three 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.

In addition to the code memory space, there are three blocks that are accessible to the user through Table Reads and Table Writes. Their locations in the memory map are shown in Figure 2-12.

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

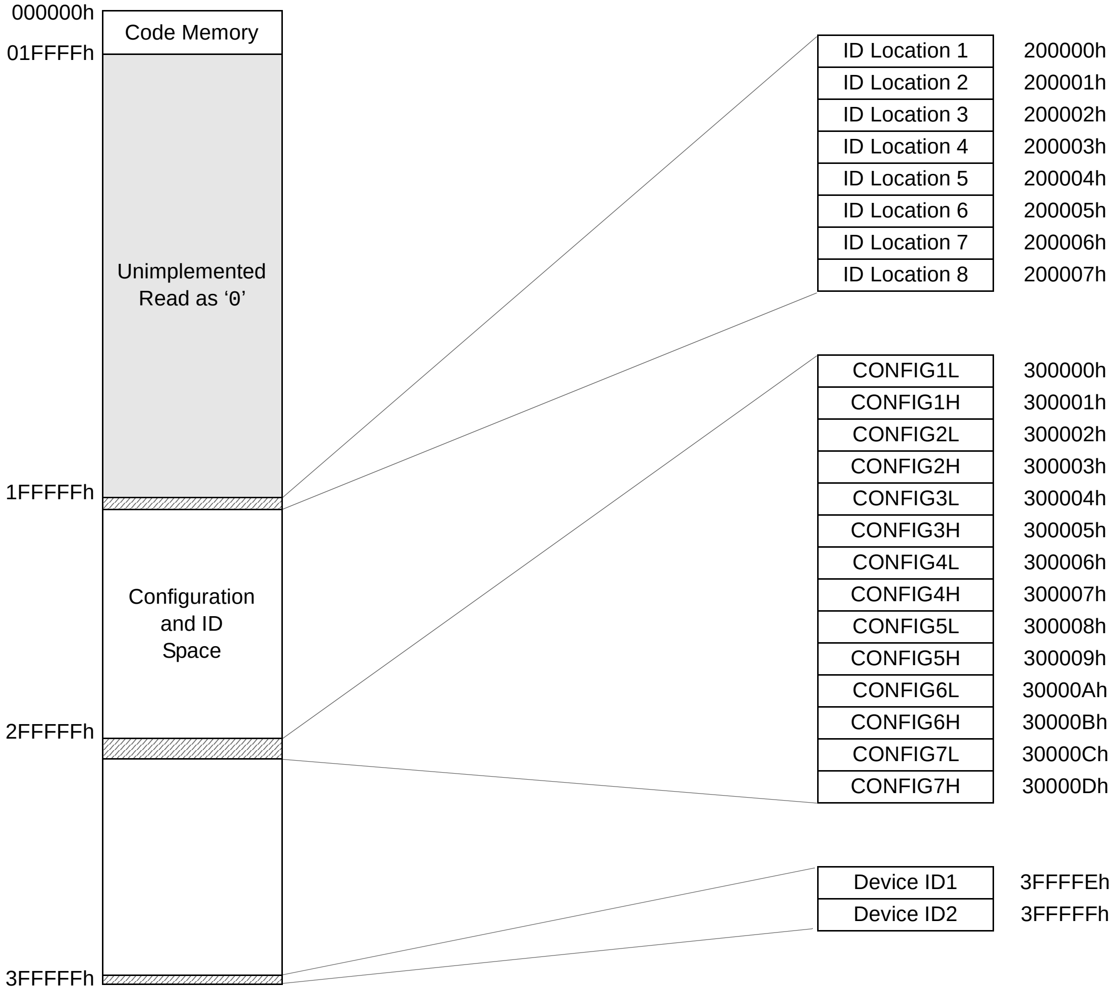
Memory in the address space, 0000000h to 3FFFFFh, is addressed via the Table Pointer register, which is comprised of three pointer registers:

- 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 many read or write operations.

FIGURE 2-12: CONFIGURATION AND ID LOCATIONS FOR PIC18F2XXX/4XXX FAMILY DEVICES



Note: Sizes of memory areas are not to scale.

2.5 Entering and Exiting High-Voltage ICSP Program/Verify Mode

As shown in Figure 2-14, the High-Voltage ICSP Program/Verify mode is entered by holding PGC and PGD low and then raising $\overline{MCLR}$/VPP/RE3 to VIHH (high voltage). Once in this mode, the code memory, data EEPROM (selected devices only, see **Section 3.3 "Data EEPROM Programming"**), ID locations and Configuration bits can be accessed and programmed in serial fashion. Figure 2-15 shows the exit sequence.

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

FIGURE 2-14: ENTERING HIGH-VOLTAGE PROGRAM/VERIFY MODE

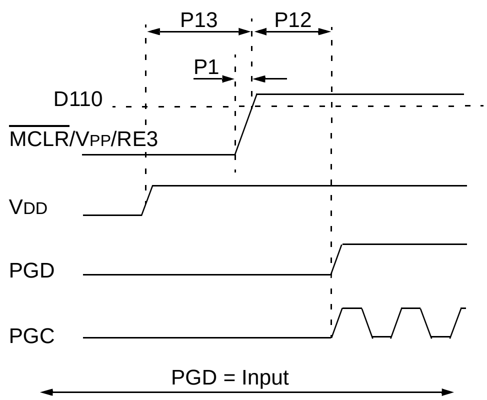
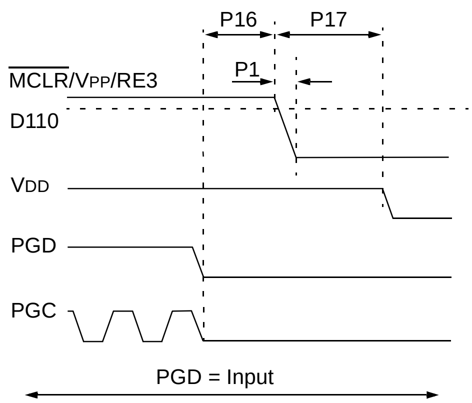


FIGURE 2-15: EXITING HIGH-VOLTAGE PROGRAM/VERIFY MODE



2.7 Serial Program/Verify Operation

The PGC pin is used as a clock input pin and the PGD pin is used for entering command bits and data input/output during serial operation. Commands and data are transmitted on the rising edge of PGC, latched on the falling edge of PGC and are Least Significant bit (LSb) first.

2.7.1 4-BIT COMMANDS

All instructions are 20 bits, consisting of a leading 4-bit command followed by a 16-bit operand, which depends on the type of command being executed. To input a command, PGC is cycled four times. The commands needed for programming and verification are shown in Table 2-8.

Depending on the 4-bit command, the 16-bit operand represents 16 bits of input data or 8 bits of input data and 8 bits of output data.

Throughout this specification, commands and data are presented as illustrated in Table 2-9. The 4-bit command is shown Most Significant bit (MSb) first. The command operand, or “Data Payload”, is shown as <MSB><LSB>. Figure 2-18 demonstrates how to serially present a 20-bit command/operand to the device.

2.7.2 CORE INSTRUCTION

The core instruction passes a 16-bit instruction to the CPU core for execution. This is needed to set up registers as appropriate for use with other commands.

TABLE 2-8: COMMANDS FOR PROGRAMMING

Description	4-Bit Command
Core Instruction (Shift in16-bit instruction)	`0000`
Shift Out TABLAT Register	`0010`
Table Read	`1000`
Table Read, Post-Increment	`1001`
Table Read, Post-Decrement	`1010`
Table Read, Pre-Increment	`1011`
Table Write	`1100`
Table Write, Post-Increment by 2	`1101`
Table Write, Start Programming, Post-Increment by 2	`1110`
Table Write, Start Programming	`1111`

TABLE 2-9: SAMPLE COMMAND SEQUENCE

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

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 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 being 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	3F8Fh
Erase Data EEPROM(1)	0084h
Erase Boot Block	0081h
Erase Configuration 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
Erase Code EEPROM Block 4	1080h
Erase Code EEPROM Block 5	2080h

Note 1: Selected devices only, see **Section 3.3 "Data EEPROM Programming"**.

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.

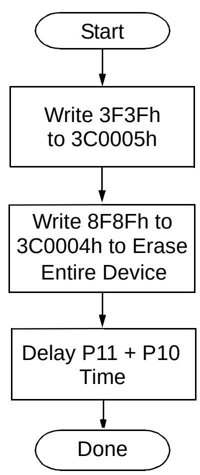
The code sequence to erase the entire device is shown in Table and the flowchart is shown in Figure 3-1.

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	3F 3F	Write 3F3Fh 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.
		NOP
		Hold PGD low until erase completes.
0000	00 00	
0000	00 00	

FIGURE 3-1: BULK ERASE FLOW



3.2.1 MODIFYING CODE MEMORY

The previous programming example assumed that the device had 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.		
Step 2: Read and modify code memory (see **Section 4.1 "Read Code Memory, ID Locations and Configuration Bits"**).		
0000 0000	8E A6 9C A6	BSF EECON1, EEPGD BCF EECON1, CFGS
Step 3: Set the Table Pointer for the block to be erased.		
0000 0000 0000 0000 0000 0000	0E <Addr[21:16]> 6E F8 0E <Addr[8:15]> 6E F7 0E <Addr[7:0]> 6E F6	MOVLW <Addr[21:16]> MOVWF TBLPTRU MOVLW <Addr[8:15]> MOVWF TBLPTRH MOVLW <Addr[7:0]> MOVWF TBLPTRL
Step 4: Enable memory writes and set up an erase.		
0000 0000	84 A6 88 A6	BSF EECON1, WREN BSF EECON1, FREE
Step 5: Initiate erase.		
0000 0000	82 A6 00 00	BSF EECON1, WR NOP - hold PGC high for time P9 and low for time P10.
Step 6: Load write buffer. The correct bytes will be selected based on the Table Pointer.		
0000 0000 0000 0000 0000 0000 1101 . . . 1111 0000	0E <Addr[21:16]> 6E F8 0E <Addr[8:15]> 6E F7 0E <Addr[7:0]> 6E F6 <MSB><LSB> . . . <MSB><LSB> 00 00	MOVLW <Addr[21:16]> MOVWF TBLPTRU MOVLW <Addr[8:15]> MOVWF TBLPTRH MOVLW <Addr[7:0]> MOVWF TBLPTRL 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. NOP - hold PGC high for time P9 and low for time P10.
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 7: Disable writes.		
0000	94 A6	BCF EECON1, WREN

3.3 Data EEPROM Programming

Note: Data EEPROM programming is **not** available on the following devices:	
PIC18F2410	PIC18F4410
PIC18F2450	PIC18F4450
PIC18F2510	PIC18F4510
PIC18F2515	PIC18F4515
PIC18F2610	PIC18F4610

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 4th PGC after the WR bit is set. It ends when the WR bit is cleared by hardware.

After the programming sequence terminates, PGC must still 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

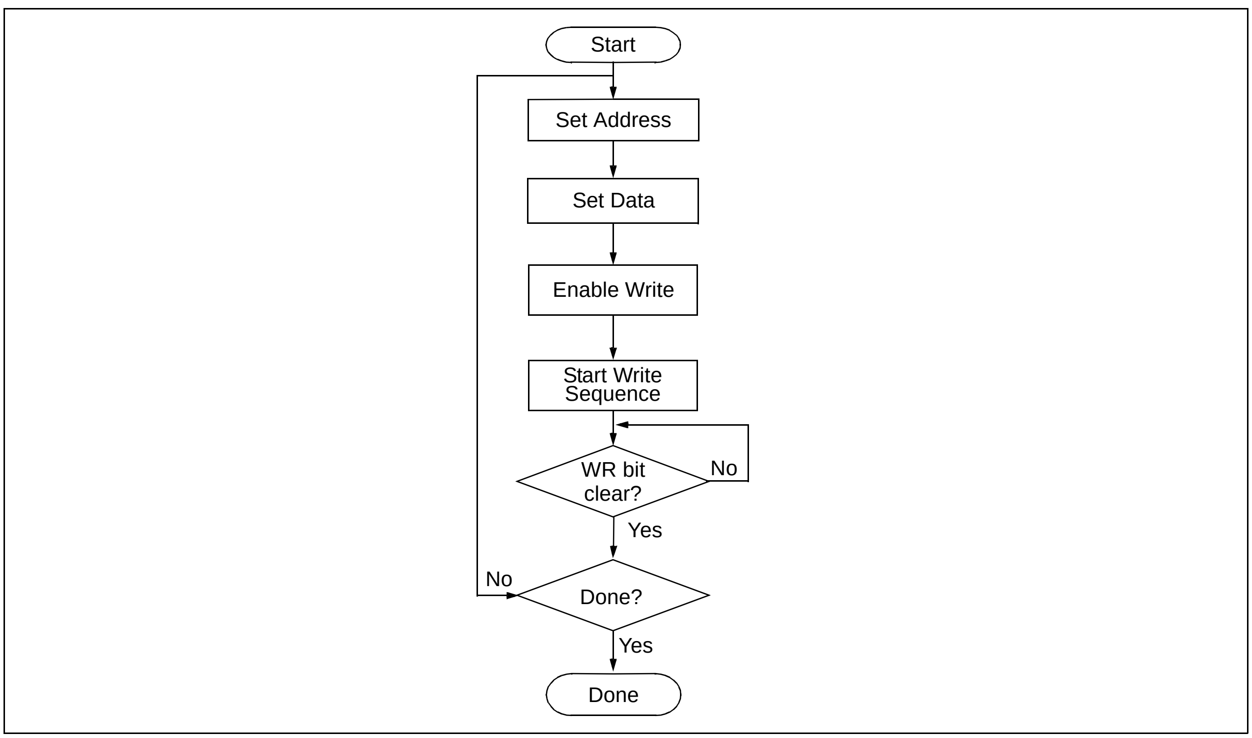


FIGURE 3-7: DATA EEPROM WRITE TIMING

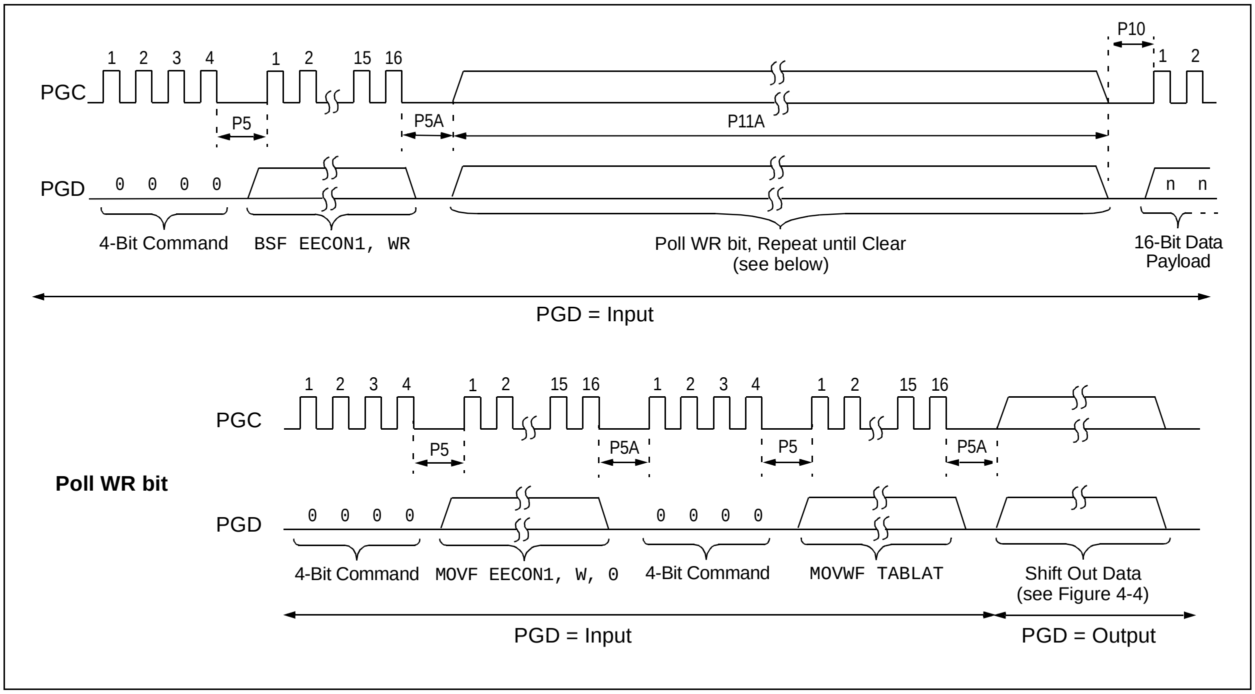


TABLE 3-7: PROGRAMMING DATA MEMORY

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: Load the data to be written.		
0000 0000	0E <Data> 6E A8	MOVLW <Data> MOVWF EEDATA
Step 4: Enable memory writes.		
0000	84 A6	BSF EECON1, WREN
Step 5: Initiate write.		
0000	82 A6	BSF EECON1, WR
Step 6: Poll WR bit, repeat until the bit is clear.		
0000 0000 0000 0010	50 A6 6E F5 00 00 <MSB><LSB>	MOVF EECON1, W, 0 MOVWF TABLAT NOP Shift out data(1)
Step 7: Hold PGC low for time P10.		
Step 8: Disable writes.		
0000	94 A6	BCF EECON1, WREN
Repeat Steps 2 through 8 to write more data.		

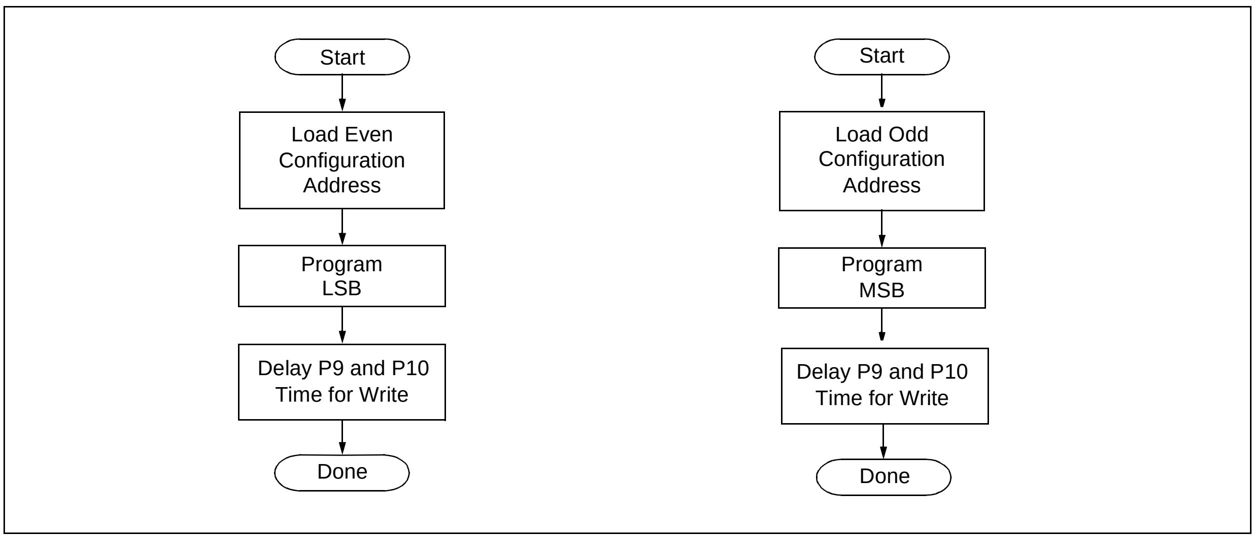
Note 1: See Figure 4-4 for details on shift out data timing.

TABLE 3-9: SET ADDRESS POINTER TO CONFIGURATION LOCATION

4-Bit Command	Data Payload	Core Instruction
Step 1: Enable writes and direct access to configuration memory.		
0000	8E A6	BSF EECON1, EEPGD
0000	8C A6	BSF EECON1, CFGS
Step 2: Set Table Pointer for configuration byte to be written. Write even/odd addresses.[(1)]		
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	<MSB ignored><LSB>	Load 2 bytes and start programming.
0000	00 00	NOP - hold PGC high for time P9 and low for time P10.
0000	0E 01	MOVLW 01h
0000	6E F6	MOVWF TBLPTRL
1111	<MSB><LSB ignored>	Load 2 bytes and start programming.
0000	00 00	NOP - hold PGC high for time P9 and low for time P10.

Note 1: Enabling the write protection of Configuration bits (WRTC = 0 in CONFIG6H) will prevent further writing of the Configuration bits. Always write all the Configuration bits before enabling the write protection for Configuration bits.

FIGURE 3-8: CONFIGURATION PROGRAMMING FLOW



5.0 CONFIGURATION WORD

The PIC18F2XXX/4XXX Family 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 the 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 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 PIC18F2XXX/4XXX Family 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.

In some cases, devices may share the same DEVID values. In such cases, the Most Significant bit of the device revision, REV4 (DEVID1<4>), will need to be examined to completely determine the device being accessed.

See Table 5-2 for a complete list of Device ID values.

FIGURE 5-1: READ DEVICE ID WORD FLOW

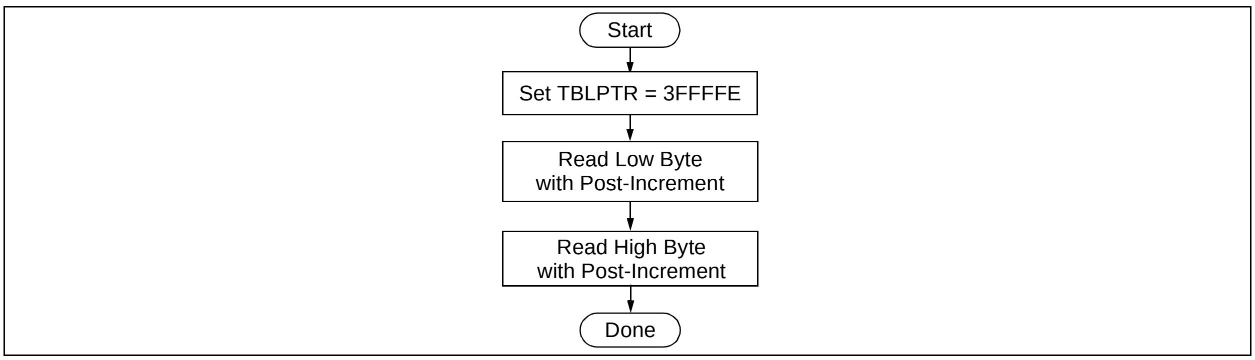


TABLE 5-3: PIC18F2XXX/4XXX FAMILY BIT DESCRIPTIONS

Bit Name	Configuration Words	Description
IESO	CONFIG1H	Internal External Switchover bit 1 = Internal External Switchover mode is enabled 0 = Internal External Switchover mode is disabled
FCMEN	CONFIG1H	Fail-Safe Clock Monitor Enable bit 1 = Fail-Safe Clock Monitor is enabled 0 = Fail-Safe Clock Monitor is disabled
FOSC<3:0>	CONFIG1H	Oscillator Selection bits 11xx = External RC oscillator, CLKO function on RA6 101x = External RC oscillator, CLKO function on RA6 1001 = Internal RC oscillator, CLKO function on RA6, port function on RA7 1000 = Internal RC oscillator, port function on RA6, port function on RA7 0111 = External RC oscillator, port function on RA6 0110 = HS oscillator, PLL is enabled (Clock Frequency = 4 x FOSC1) 0101 = EC oscillator, port function on RA6 0100 = EC oscillator, CLKO function on RA6 0011 = External RC oscillator, CLKO function on RA6 0010 = HS oscillator 0001 = XT oscillator 0000 = LP oscillator
FOSC<3:0>	CONFIG1H	Oscillator Selection bits **(PIC18F2455/2550/4455/4550, PIC18F2458/2553/4458/4553 and PIC18F2450/4450 devices only)** 111x = HS oscillator, PLL is enabled, HS is used by USB 110x = HS oscillator, HS is used by USB 1011 = Internal oscillator, HS is used by USB 1010 = Internal oscillator, XT is used by USB 1001 = Internal oscillator, CLKO function on RA6, EC is used by USB 1000 = Internal oscillator, port function on RA6, EC is used by USB 0111 = EC oscillator, PLL is enabled, CLKO function on RA6, EC is used by USB 0110 = EC oscillator, PLL is enabled, port function on RA6, EC is used by USB 0101 = EC oscillator, CLKO function on RA6, EC is used by USB 0100 = EC oscillator, port function on RA6, EC is used by USB 001x = XT oscillator, PLL is enabled, XT is used by USB 000x = XT oscillator, XT is used by USB
USBDIV	CONFIG1L	USB Clock Selection bit **(PIC18F2455/2550/4455/4550, PIC18F2458/2553/4458/4553 and PIC18F2450/4450 devices only)** Selects the clock source for full-speed USB operation: 1 = USB clock source comes from the 96 MHz PLL divided by 2 0 = USB clock source comes directly from the OSC1/OSC2 oscillator block; no divide
CPUDIV<1:0>	CONFIG1L	CPU System Clock Selection bits **(PIC18F2455/2550/4455/4550, PIC18F2458/2553/4458/4553 and PIC18F2450/4450 devices only)** 11 = CPU system clock divided by 4 10 = CPU system clock divided by 3 01 = CPU system clock divided by 2 00 = No CPU system clock divide

Note 1: The BBSIZ bits, BBSIZ<1:0> and BBSIZ<2:1> bits, cannot be changed once any of the following code-protect bits are enabled: CPB or CP0, WRTB or WRT0, EBTRB or EBTR0.

2: Not available in PIC18FXX8X and PIC18F2450/4450 devices.

TABLE 5-3: PIC18F2XXX/4XXX FAMILY BIT DESCRIPTIONS (CONTINUED)

Bit Name	Configuration Words	Description
WRT5	CONFIG6L	Write Protection bit (Block 5 code memory area) **(PIC18F2685 and PIC18F4685 devices only)** 1 = Block 5 is not write-protected 0 = Block 5 is write-protected
WRT4	CONFIG6L	Write Protection bit (Block 4 code memory area) **(PIC18F2682/2685 and PIC18F4682/4685 devices only)** 1 = Block 4 is not write-protected 0 = Block 4 is write-protected
WRT3	CONFIG6L	Write Protection bit (Block 3 code memory area) 1 = Block 3 is not write-protected 0 = Block 3 is write-protected
WRT2	CONFIG6L	Write Protection bit (Block 2 code memory area) 1 = Block 2 is not write-protected 0 = Block 2 is write-protected
WRT1	CONFIG6L	Write Protection bit (Block 1 code memory area) 1 = Block 1 is not write-protected 0 = Block 1 is write-protected
WRT0	CONFIG6L	Write Protection bit (Block 0 code memory area) 1 = Block 0 is not write-protected 0 = Block 0 is write-protected
WRTD	CONFIG6H	Write Protection bit (Data EEPROM) 1 = Data EEPROM is not write-protected 0 = Data EEPROM is write-protected
WRTB	CONFIG6H	Write Protection bit (Boot Block memory area) 1 = Boot Block is not write-protected 0 = Boot Block is write-protected
WRTC	CONFIG6H	Write Protection bit (Configuration registers) 1 = Configuration registers are not write-protected 0 = Configuration registers are write-protected
EBTR5	CONFIG7L	Table Read Protection bit (Block 5 code memory area) **(PIC18F2685 and PIC18F4685 devices only)** 1 = Block 5 is not protected from Table Reads executed in other blocks 0 = Block 5 is protected from Table Reads executed in other blocks
EBTR4	CONFIG7L	Table Read Protection bit (Block 4 code memory area) **(PIC18F2682/2685 and PIC18F4682/4685 devices only)** 1 = Block 4 is not protected from Table Reads executed in other blocks 0 = Block 4 is protected from Table Reads executed in other blocks
EBTR3	CONFIG7L	Table Read Protection bit (Block 3 code memory area) 1 = Block 3 is not protected from Table Reads executed in other blocks 0 = Block 3 is protected from Table Reads executed in other blocks
EBTR2	CONFIG7L	Table Read Protection bit (Block 2 code memory area) 1 = Block 2 is not protected from Table Reads executed in other blocks 0 = Block 2 is protected from Table Reads executed in other blocks
EBTR1	CONFIG7L	Table Read Protection bit (Block 1 code memory area) 1 = Block 1 is not protected from Table Reads executed in other blocks 0 = Block 1 is protected from Table Reads executed in other blocks

Note 1: The BBSIZ bits, BBSIZ<1:0> and BBSIZ<2:1> bits, cannot be changed once any of the following code-protect bits are enabled: CPB or CP0, WRTB or WRT0, EBTRB or EBTR0.

2: Not available in PIC18FXX8X and PIC18F2450/4450 devices.

TABLE 5-5: CONFIGURATION WORD MASKS FOR COMPUTING CHECKSUMS

Device	Configuration Word (CONFIGxx)													
	1L	1H	2L	2H	3L	3H	4L	4H	5L	5H	6L	6H	7L	7H
	Address (30000xh)													
	0h	1h	2h	3h	4h	5h	6h	7h	8h	9h	Ah	Bh	Ch	Dh
PIC18F2221	00	CF	1F	1F	00	87	F5	00	03	C0	03	E0	03	40
PIC18F2321	00	CF	1F	1F	00	87	F5	00	03	C0	03	E0	03	40
PIC18F2410	00	CF	1F	1F	00	87	C5	00	03	C0	03	E0	03	40
PIC18F2420	00	CF	1F	1F	00	87	C5	00	03	C0	03	E0	03	40
PIC18F2423	00	CF	1F	1F	00	87	C5	00	03	C0	03	E0	03	40
PIC18F2450	3F	CF	3F	1F	00	86	ED	00	03	40	03	60	03	40
PIC18F2455	3F	CF	3F	1F	00	87	E5	00	07	C0	07	E0	07	40
PIC18F2458	3F	CF	3F	1F	00	87	E5	00	07	C0	07	E0	07	40
PIC18F2480	00	CF	1F	1F	00	86	D5	00	03	C0	03	E0	03	40
PIC18F2510	00	1F	1F	1F	00	87	C5	00	0F	C0	0F	E0	0F	40
PIC18F2515	00	CF	1F	1F	00	87	C5	00	0F	C0	0F	E0	0F	40
PIC18F2520	00	CF	1F	1F	00	87	C5	00	0F	C0	0F	E0	0F	40
PIC18F2523	00	CF	1F	1F	00	87	C5	00	0F	C0	0F	E0	0F	40
PIC18F2525	00	CF	1F	1F	00	87	C5	00	0F	C0	0F	E0	0F	40
PIC18F2550	3F	CF	3F	1F	00	87	E5	00	0F	C0	0F	E0	0F	40
PIC18F2553	3F	CF	3F	1F	00	87	E5	00	0F	C0	0F	E0	0F	40
PIC18F2580	00	CF	1F	1F	00	86	D5	00	0F	C0	0F	E0	0F	40
PIC18F2585	00	CF	1F	1F	00	86	C5	00	0F	C0	0F	E0	0F	40
PIC18F2610	00	CF	1F	1F	00	87	C5	00	0F	C0	0F	E0	0F	40
PIC18F2620	00	CF	1F	1F	00	87	C5	00	0F	C0	0F	E0	0F	40
PIC18F2680	00	CF	1F	1F	00	86	C5	00	0F	C0	0F	E0	0F	40
PIC18F2682	00	CF	1F	1F	00	86	C5	00	3F	C0	3F	E0	3F	40
PIC18F2685	00	CF	1F	1F	00	86	C5	00	3F	C0	3F	E0	3F	40
PIC18F4221	00	CF	1F	1F	00	87	F5	00	03	C0	03	E0	03	40
PIC18F4321	00	CF	1F	1F	00	87	F5	00	03	C0	03	E0	03	40
PIC18F4410	00	CF	1F	1F	00	87	C5	00	03	C0	03	E0	03	40
PIC18F4420	00	CF	1F	1F	00	87	C5	00	03	C0	03	E0	03	40
PIC18F4423	00	CF	1F	1F	00	87	C5	00	03	C0	03	E0	03	40
PIC18F4450	3F	CF	3F	1F	00	86	ED	00	03	40	03	60	03	40
PIC18F4455	3F	CF	3F	1F	00	87	E5	00	07	C0	07	E0	07	40
PIC18F4458	3F	CF	3F	1F	00	87	E5	00	07	C0	07	E0	07	40
PIC18F4480	00	CF	1F	1F	00	86	D5	00	03	C0	03	E0	03	40
PIC18F4510	00	CF	1F	1F	00	87	C5	00	0F	C0	0F	E0	0F	40
PIC18F4515	00	CF	1F	1F	00	87	C5	00	0F	C0	0F	E0	0F	40
PIC18F4520	00	CF	1F	1F	00	87	C5	00	0F	C0	0F	E0	0F	40
PIC18F4523	00	CF	1F	1F	00	87	C5	00	0F	C0	0F	E0	0F	40
PIC18F4525	00	CF	1F	1F	00	87	C5	00	0F	C0	0F	E0	0F	40
PIC18F4550	3F	CF	3F	1F	00	87	E5	00	0F	C0	0F	E0	0F	40
PIC18F4553	3F	CF	3F	1F	00	87	E5	00	0F	C0	0F	E0	0F	40
PIC18F4580	00	CF	1F	1F	00	86	D5	00	0F	C0	0F	E0	0F	40
PIC18F4585	00	CF	1F	1F	00	86	C5	00	0F	C0	0F	E0	0F	40
PIC18F4610	00	CF	1F	1F	00	87	C5	00	0F	C0	0F	E0	0F	40

Legend: Shaded cells are unimplemented.

TABLE 5-5: CONFIGURATION WORD MASKS FOR COMPUTING CHECKSUMS (CONTINUED)

Device	Configuration Word (CONFIGxx)													
	1L	1H	2L	2H	3L	3H	4L	4H	5L	5H	6L	6H	7L	7H
	Address (30000xh)													
	0h	1h	2h	3h	4h	5h	6h	7h	8h	9h	Ah	Bh	Ch	Dh
PIC18F4620	00	CF	1F	1F	00	87	C5	00	0F	C0	0F	E0	0F	40
PIC18F4680	00	CF	1F	1F	00	86	C5	00	0F	C0	0F	E0	0F	40
PIC18F4682	00	CF	1F	1F	00	86	C5	00	3F	C0	3F	E0	3F	40
PIC18F4685	00	CF	1F	1F	00	86	C5	00	3F	C0	3F	E0	3F	40

Legend: Shaded cells are unimplemented.

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
P11A	TDRWT	Data Write Polling Time	4	—	ms	
P12	THLD2	Input Data Hold Time from $\overline{MCLR}$/VPP/RE3 ↑	2	—	μs	
P13	TSET2	VDD ↑ Setup Time to $\overline{MCLR}$/VPP/RE3 ↑	100	—	ns	**(Note 2)**
P14	TVALID	Data Out Valid from PGC ↑	10	—	ns	
P15	TSET3	PGM ↑ Setup Time to $\overline{MCLR}$/VPP/RE3 ↑	2	—	μs	**(Note 2)**
P16	TDLY8	Delay Between Last PGC ↓ and $\overline{MCLR}$/VPP/RE3 ↓	0	—	s	
P17	THLD3	$\overline{MCLR}$/VPP/RE3 ↓ to VDD ↓	—	100	ns	
P18	THLD4	$\overline{MCLR}$/VPP/RE3 ↓ to PGM ↓	0	—	s	

Note 1: Do not allow excess time when transitioning $\overline{MCLR}$ 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.

2: When ICPRT = 1, this specification also applies to ICVPP.

3: At 0°C-50°C.

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

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

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

Trademarks

ISBN: 978-1-63277-856-7

QUALITY MANAGEMENT SYSTEM CERTIFIED BY DNV

ISO/TS 16949

Microchip received ISO/TS-16949:2009 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.