**Welcome to E-XFL.COM**

## What is "Embedded - Microcontrollers"?

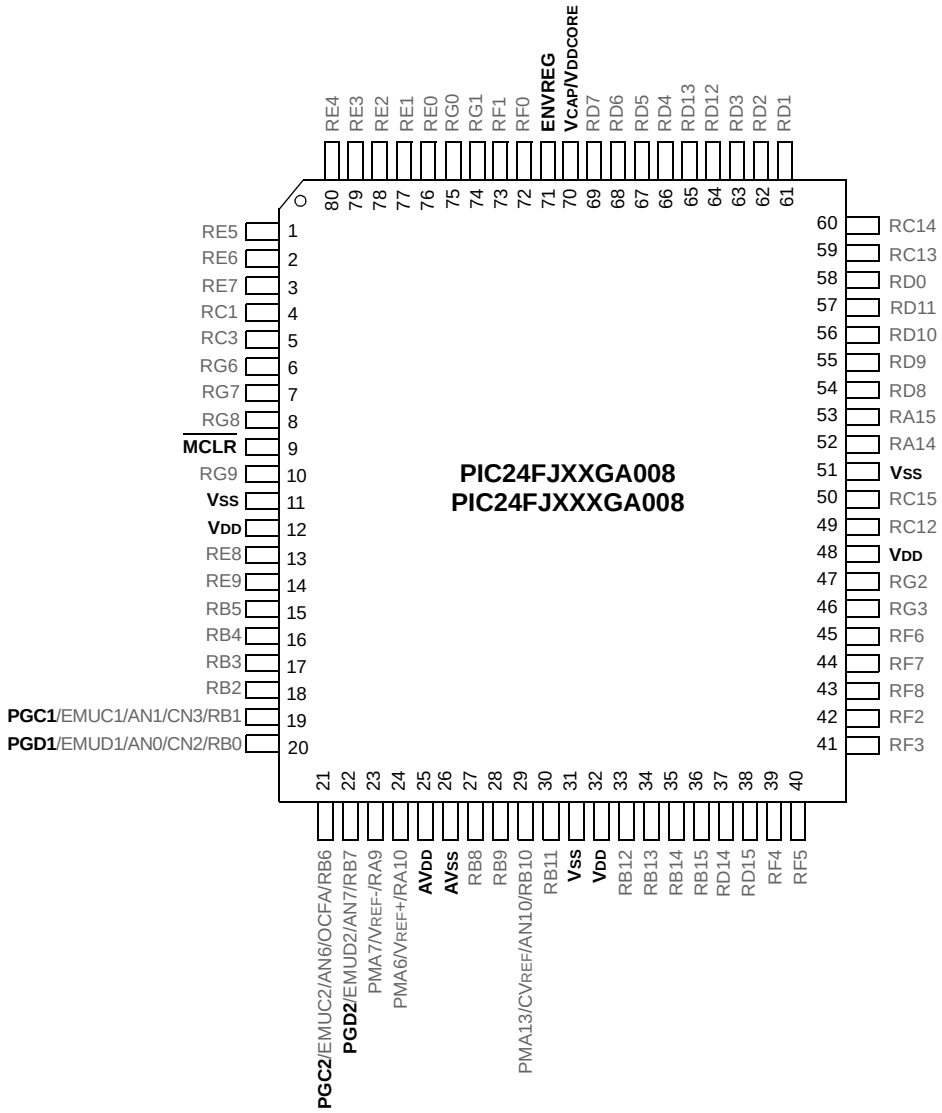
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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 | 16-Bit |
| Speed | 16MHz |
| Connectivity | I²C, SPI, UART/USART |
| Peripherals | Brown-out Detect/Reset, POR, PWM, WDT |
| Number of I/O | 69 |
| Program Memory Size | 128KB (43K x 24) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 8K x 8 |
| Voltage - Supply (Vcc/Vdd) | 2V ~ 3.6V |
| Data Converters | A/D 16x10b |
| Oscillator Type | Internal |
| 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/pic24fj128ga008t-i-pt |

## Pin Diagrams (Continued)

**80-Pin TQFP**



## 3.0 DEVICE PROGRAMMING – ICSP

ICSP mode is a special programming protocol that allows you to read and write to PIC24FJXXXGA0XX device family memory. The ICSP mode is the most direct method used to program the device; note, however, that Enhanced ICSP is faster. ICSP mode also has the ability to read the contents of executive memory to determine if the programming executive is present. This capability is accomplished by applying control codes and instructions, serially to the device, using pins, PGCx and PGDx.

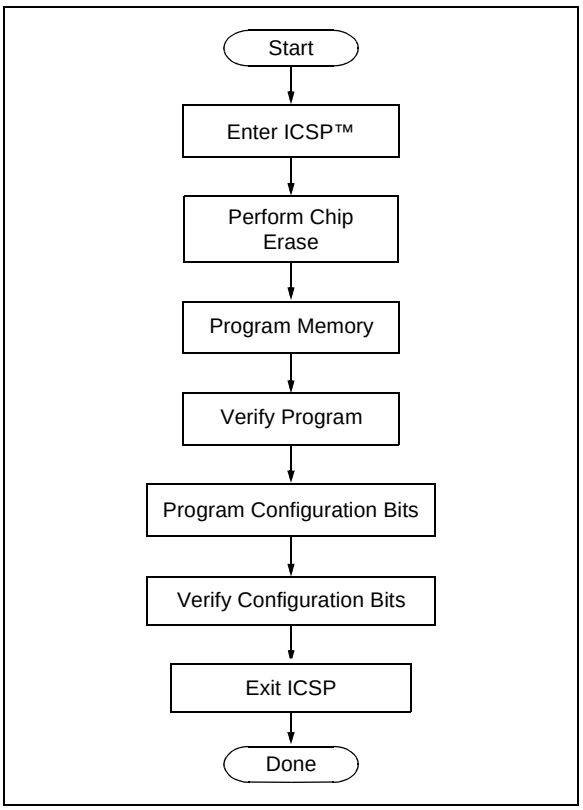
In ICSP mode, the system clock is taken from the PGCx pin, regardless of the device's oscillator Configuration bits. All instructions are shifted serially into an internal buffer, then loaded into the Instruction Register (IR) and executed. No program fetching occurs from internal memory. Instructions are fed in 24 bits at a time. PGDx is used to shift data in and PGCx is used as both the serial shift clock and the CPU execution clock.

| | |
|---|---|
| **Note:** | During ICSP operation, the operating frequency of PGCx must not exceed 10 MHz. |

### 3.1 Overview of the Programming Process

Figure 3-1 shows the high-level overview of the programming process. After entering ICSP mode, the first action is to Chip Erase the device. Next, the code memory is programmed, followed by the device Configuration registers. Code memory (including the Configuration registers) is then verified to ensure that programming was successful. Then, program the code-protect Configuration bits, if required.

**FIGURE 3-1: HIGH-LEVEL ICSP™ PROGRAMMING FLOW**




### 3.2 ICSP Operation

Upon entry into ICSP mode, the CPU is Idle. Execution of the CPU is governed by an internal state machine. A 4-bit control code is clocked in using PGCx and PGDx, and this control code is used to command the CPU (see Table 3-1).

The SIX control code is used to send instructions to the CPU for execution, and the REGOUT control code is used to read data out of the device via the VISI register.

**TABLE 3-1: CPU CONTROL CODES IN ICSP™ MODE**

| 4-Bit Control Code | Mnemonic | Description |
|---|---|---|
| 0000b | SIX | Shift in 24-bit instruction and execute. |
| 0001b | REGOUT | Shift out the VISI (0784h) register. |
| 0010b-1111b | N/A | Reserved. |

### 3.2.2 REGOUT SERIAL INSTRUCTION EXECUTION

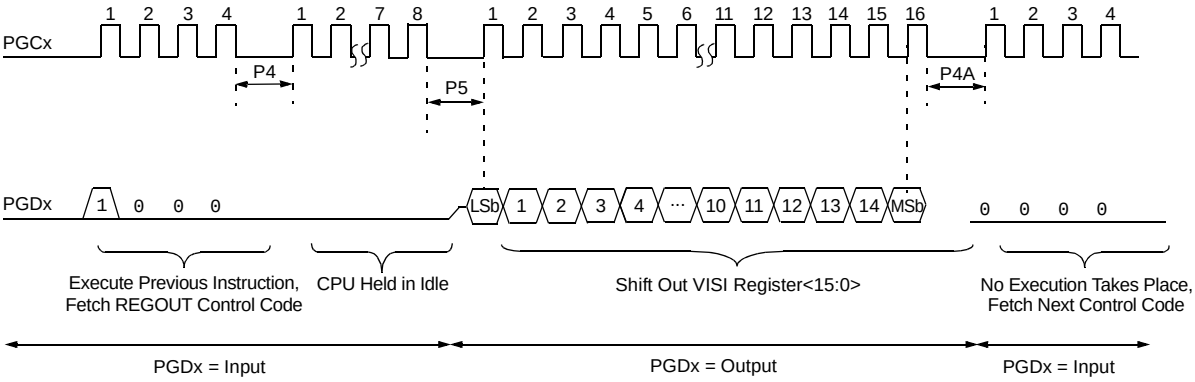
The REGOUT control code allows for data to be extracted from the device in ICSP mode. It is used to clock the contents of the VISI register, out of the device, over the PGDx pin. After the REGOUT control code is received, the CPU is held Idle for 8 cycles. After these 8 cycles, an additional 16 cycles are required to clock the data out (see Figure 3-3).

The REGOUT code is unique because the PGDx pin is an input when the control code is transmitted to the device. However, after the control code is processed, the PGDx pin becomes an output as the VISI register is shifted out.

**Note 1:** After the contents of VISI are shifted out, the PIC24FJXXXGA0XX device maintains PGDx as an output until the first rising edge of the next clock is received.

**2:** Data changes on the falling edge and latches on the rising edge of PGCx. For all data transmissions, the Least Significant bit (LSb) is transmitted first.

**FIGURE 3-3: REGOUT SERIAL EXECUTION**



## 3.4 Flash Memory Programming in ICSP Mode

### 3.4.1 PROGRAMMING OPERATIONS

Flash memory write and erase operations are controlled by the NVMCON register. Programming is performed by setting NVMCON to select the type of erase operation (Table 3-2) or write operation (Table 3-3) and initiating the programming by setting the WR control bit (NVMCON<15>).

In ICSP mode, all programming operations are self-timed. There is an internal delay between the user setting the WR control bit and the automatic clearing of the WR control bit when the programming operation is complete. Please refer to **Section 7.0 "AC/DC Characteristics and Timing Requirements"** for information about the delays associated with various programming operations.

**TABLE 3-2: NVMCON ERASE OPERATIONS**

| NVMCON Value | Erase Operation |
|---|---|
| 404Fh | Erase all code memory, executive memory and Configuration registers (does not erase Unit ID or Device ID registers). |
| 4042h | Erase a page of code memory or executive memory. |

**TABLE 3-3: NVMCON WRITE OPERATIONS**

| NVMCON Value | Write Operation |
|---|---|
| 4003h | Write a Configuration Word register. |
| 4001h | Program 1 row (64 instruction words) of code memory or executive memory. |

### 3.4.2 STARTING AND STOPPING A PROGRAMMING CYCLE

The WR bit (NVMCON<15>) is used to start an erase or write cycle. Setting the WR bit initiates the programming cycle.

All erase and write cycles are self-timed. The WR bit should be polled to determine if the erase or write cycle has been completed. Starting a programming cycle is performed as follows:

```
BSET    NVMCON, #WR
```

## 3.5 Erasing Program Memory

The procedure for erasing program memory (all of code memory, data memory, executive memory and code-protect bits) consists of setting NVMCON to 404Fh and executing the programming cycle.

A Chip Erase can erase all of user memory or all of both the user and configuration memory. A table write instruction should be executed prior to performing the Chip Erase to select which sections are erased.

When this table write instruction is executed:

- If the TBLPAG register points to user space (is less than 0x80), the Chip Erase will erase only user memory.
- If TBLPAG points to configuration space (is greater than or equal to 0x80), the Chip Erase will erase both user and configuration memory.

  If configuration memory is erased, the internal oscillator Calibration Word, located at 0x807FE, will be erased. This location should be stored prior to performing a whole Chip Erase and restored afterward to prevent internal oscillators from becoming uncalibrated.

Figure 3-5 shows the ICSP programming process for performing a Chip Erase. This process includes the ICSP command code, which must be transmitted (for each instruction), Least Significant bit first, using the PGCx and PGDx pins (see Figure 3-2).

| **Note:** | Program memory must be erased before writing any data to program memory. |
|---|---|

**FIGURE 3-5: CHIP ERASE FLOW**

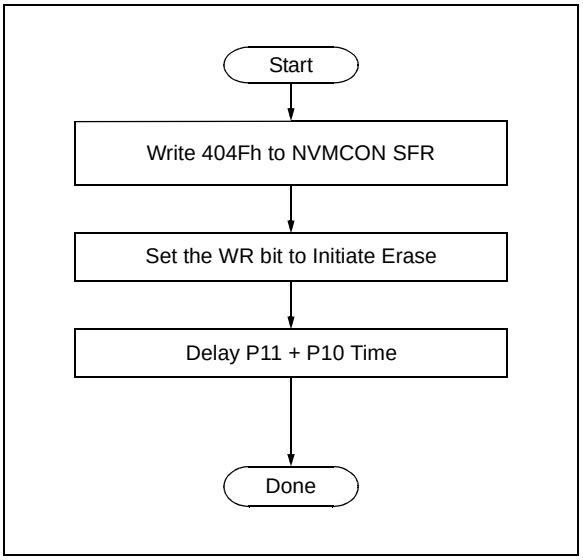


**TABLE 3-4: SERIAL INSTRUCTION EXECUTION FOR CHIP ERASE**

| Command (Binary) | Data (Hex) | Description |
|---|---|---|
| **Step 1:** Exit the Reset vector. | | |
| 0000 | 000000 | NOP |
| 0000 | 040200 | GOTO 0x200 |
| 0000 | 000000 | NOP |
| **Step 2:** Set the NVMCON to erase all program memory. | | |
| 0000 | 2404FA | MOV #0x404F, W10 |
| 0000 | 883B0A | MOV W10, NVMCON |
| **Step 3:** Set TBLPAG and perform dummy table write to select what portions of memory are erased. | | |
| 0000 | 200000 | MOV #<PAGEVAL>, W0 |
| 0000 | 880190 | MOV W0, TBLPAG |
| 0000 | 200000 | MOV #0x0000, W0 |
| 0000 | BB0800 | TBLWTL W0,[W0] |
| 0000 | 000000 | NOP |
| 0000 | 000000 | NOP |
| **Step 4:** Initiate the erase cycle. | | |
| 0000 | A8E761 | BSET NVMCON, #WR |
| 0000 | 000000 | NOP |
| 0000 | 000000 | NOP |
| **Step 5:** Repeat this step to poll the WR bit (bit 15 of NVMCON) until it is cleared by the hardware. | | |
| 0000 | 040200 | GOTO 0x200 |
| 0000 | 000000 | NOP |
| 0000 | 803B02 | MOV NVMCON, W2 |
| 0000 | 883C22 | MOV W2, VISI |
| 0000 | 000000 | NOP |
| 0001 | <VISI> | Clock out contents of the VISI register. |
| 0000 | 000000 | NOP |

## 3.7 Writing Configuration Words

The PIC24FJXXXGA0XX family configuration is stored in Flash Configuration Words at the end of the user space program memory and in multiple register Configuration Words located in the test space.

These registers reflect values read at any Reset from program memory locations. The values can be changed only by programming the content of the corresponding Flash Configuration Word and resetting the device. The Reset forces an automatic reload of the Flash stored configuration values by sequencing through the dedicated Flash Configuration Words and transferring the data into the Configuration registers. To change the values of the Flash Configuration Word once it has been programmed, the device must be Chip Erased, as described in **Section 3.5 “Erasing Program Memory”**, and reprogrammed to the desired value. It is not possible to program a ‘0’ to ‘1’, but they may be programmed from a ‘1’ to ‘0’ to enable code protection.

Table 3-7 shows the ICSP programming details for programming the Configuration Word locations, including the serial pattern with the ICSP command code which must be transmitted, Least Significant bit first, using the PGCx and PGDx pins (see Figure 3-2).

In Step 1, the Reset vector is exited. In Step 2, the NVMCON register is initialized for programming of code memory. In Step 3, the 24-bit starting destination address for programming is loaded into the TBLPAG register and W7 register.

The TBLPAG register must be loaded with the following:

- 96 and 64 Kbyte devices – 00h
- 128 Kbyte devices – 01h

To verify the data by reading the Configuration Words after performing the write in order, the code protection bits initially should be programmed to a ‘1’ to ensure that the verification can be performed properly. After verification is finished, the code protection bit can be programmed to a ‘0’ by using a word write to the appropriate Configuration Word.

**TABLE 3-6: DEFAULT CONFIGURATION REGISTER VALUES**

| Address | Name | Default Value |
|---|---|---|
| Last Word | CW1 | 7FFFh(1) |
| Last Word – 2 | CW2 | FFFFh |

**Note 1:** CW1<15> is reserved and must be programmed to ‘0’.

**TABLE 3-7: SERIAL INSTRUCTION EXECUTION FOR WRITING CONFIGURATION REGISTERS**

| Command (Binary) | Data (Hex) | Description |
|---|---|---|
| **Step 1:** Exit the Reset vector. | | |
| 0000 | 000000 | NOP |
| 0000 | 040200 | GOTO 0x200 |
| 0000 | 000000 | NOP |
| **Step 2:** Initialize the Write Pointer (W7) for the TBLWT instruction. | | |
| 0000 | 2xxxx7 | MOV <CW2Address15:0>, W7 |
| **Step 3:** Set the NVMCON register to program CW2. | | |
| 0000 | 24003A | MOV #0x4003, W10 |
| 0000 | 883B0A | MOV W10, NVMCON |
| **Step 4:** Initialize the TBLPAG register. | | |
| 0000 | 200xx0 | MOV <CW2Address23:16>, W0 |
| 0000 | 880190 | MOV W0, TBLPAG |
| **Step 5:** Load the Configuration register data to W6. | | |
| 0000 | 2xxxx6 | MOV #<CW2_VALUE>, W6 |
| **Step 6:** Write the Configuration register data to the write latch and increment the Write Pointer. | | |
| 0000 | 000000 | NOP |
| 0000 | BB1B86 | TBLWTL W6, [W7++] |
| 0000 | 000000 | NOP |
| 0000 | 000000 | NOP |
| **Step 7:** Initiate the write cycle. | | |
| 0000 | A8E761 | BSET NVMCON, #WR |
| 0000 | 000000 | NOP |
| 0000 | 000000 | NOP |
| **Step 8:** Repeat this step to poll the WR bit (bit 15 of NVMCON) until it is cleared by the hardware. | | |
| 0000 | 040200 | GOTO 0x200 |
| 0000 | 000000 | NOP |
| 0000 | 803B02 | MOV NVMCON, W2 |
| 0000 | 883C22 | MOV W2, VISI |
| 0000 | 000000 | NOP |
| 0001 | <VISI> | Clock out contents of the VISI register. |
| 0000 | 000000 | NOP |
| **Step 9:** Reset device internal PC. | | |
| 0000 | 040200 | GOTO 0x200 |
| 0000 | 000000 | NOP |
| **Step 10:** Repeat Steps 5-9 to write CW1. | | |

## 3.8 Reading Code Memory

Reading from code memory is performed by executing a series of TBLRD instructions and clocking out the data using the REGOUT command.

Table 3-8 shows the ICSP programming details for reading code memory. In Step 1, the Reset vector is exited. In Step 2, the 24-bit starting source address for reading is loaded into the TBLPAG register and W6 register. The upper byte of the starting source address is stored in TBLPAG and the lower 16 bits of the source address are stored in W6.

To minimize the reading time, the packed instruction word format that was utilized for writing is also used for reading (see Figure 3-6). In Step 3, the Write Pointer, W7, is initialized. In Step 4, two instruction words are read from code memory and clocked out of the device, through the VISI register, using the REGOUT command. Step 4 is repeated until the desired amount of code memory is read.

**TABLE 3-8: SERIAL INSTRUCTION EXECUTION FOR READING CODE MEMORY**

| Command (Binary) | Data (Hex) | Description |
|---|---|---|
| **Step 1:** Exit Reset vector. | | |
| 0000 | 000000 | NOP |
| 0000 | 040200 | GOTO 0x200 |
| 0000 | 000000 | NOP |
| **Step 2:** Initialize TBLPAG and the Read Pointer (W6) for TBLRD instruction. | | |
| 0000 | 200xx0 | MOV #<SourceAddress23:16>, W0 |
| 0000 | 880190 | MOV W0, TBLPAG |
| 0000 | 2xxxx6 | MOV #<SourceAddress15:0>, W6 |
| **Step 3:** Initialize the Write Pointer (W7) to point to the VISI register. | | |
| 0000 | 207847 | MOV #VISI, W7 |
| 0000 | 000000 | NOP |
| **Step 4:** Read and clock out the contents of the next two locations of code memory, through the VISI register, using the REGOUT command. | | |
| 0000 | BA0B96 | TBLRDL [W6], [W7] |
| 0000 | 000000 | NOP |
| 0000 | 000000 | NOP |
| 0001 | <VISI> | Clock out contents of VISI register |
| 0000 | 000000 | NOP |
| 0000 | BADBB6 | TBLRDH.B [W6++], [W7++] |
| 0000 | 000000 | NOP |
| 0000 | 000000 | NOP |
| 0000 | BAD3D6 | TBLRDH.B [++W6], [W7--] |
| 0000 | 000000 | NOP |
| 0000 | 000000 | NOP |
| 0001 | <VISI> | Clock out contents of VISI register |
| 0000 | 000000 | NOP |
| 0000 | BA0BB6 | TBLRDL [W6++], [W7] |
| 0000 | 000000 | NOP |
| 0000 | 000000 | NOP |
| 0001 | <VISI> | Clock out contents of VISI register |
| 0000 | 000000 | NOP |
| **Step 5:** Reset device internal PC. | | |
| 0000 | 040200 | GOTO 0x200 |
| 0000 | 000000 | NOP |
| **Step 6:** Repeat Steps 4 and 5 until all desired code memory is read. | | |

### 4.5.2 PROGRAMMING VERIFICATION

After code memory is programmed, the contents of memory can be verified to ensure that programming was successful. Verification requires code memory to be read back and compared against the copy held in the programmer's buffer.

The READP command can be used to read back all of the programmed code memory.

Alternatively, you can have the programmer perform the verification after the entire device is programmed using a checksum computation.

## 4.6 Configuration Bits Programming

### 4.6.1 OVERVIEW

The PIC24FJXXXGA0XX family has Configuration bits stored in the last two locations of implemented program memory (see Table 2-2 for locations). These bits can be set or cleared to select various device configurations. There are three types of Configuration bits: system operation bits, code-protect bits and unit ID bits. The system operation bits determine the power-on settings for system level components, such as oscillator and Watchdog Timer. The code-protect bits prevent program memory from being read and written.

The register descriptions for the CW1 and CW2 Configuration registers are shown in Table 4-2.

**TABLE 4-2: PIC24FJXXXGA0XX FAMILY CONFIGURATION BITS DESCRIPTION**

| Bit Field | Register | Description |
|---|---|---|
| I2C1SEL(1) | CW2<2> | I2C1 Pin Mapping bit<br>1 = Default location for SCL1/SDA1 pins<br>0 = Alternate location for SCL1/SDA1 pins |
| $\overline{\text{DEBUG}}$ | CW1<11> | Background Debug Enable bit<br>1 = Device will reset in User mode<br>0 = Device will reset in Debug mode |
| FCKSM1:FCKSM0 | CW2<7:6> | Clock Switching Mode bits<br>1x = Clock switching is disabled, Fail-Safe Clock Monitor is disabled<br>01 = Clock switching is enabled, Fail-Safe Clock Monitor is disabled<br>00 = Clock switching is enabled, Fail-Safe Clock Monitor is enabled |
| FNOSC2:FNOSC0 | CW2<10:8> | Initial Oscillator Source Selection bits<br>111 = Internal Fast RC (FRCDIV) oscillator with postscaler<br>110 = Reserved<br>101 = Low-Power RC (LPRC) oscillator<br>100 = Secondary (SOSC) oscillator<br>011 = Primary (XTPLL, HSPLL, ECPLL) oscillator with PLL<br>010 = Primary (XT, HS, EC) oscillator<br>001 = Internal Fast RC (FRCPLL) oscillator with postscaler and PLL<br>000 = Fast RC (FRC) oscillator |
| FWDTEN | CW1<7> | Watchdog Timer Enable bit<br>1 = Watchdog Timer always enabled (LPRC oscillator cannot be disabled; clearing the SWDTEN bit in the RCON register will have no effect)<br>0 = Watchdog Timer enabled/disabled by user software (LPRC can be disabled by clearing the SWDTEN bit in the RCON register) |
| GCP | CW1<13> | General Segment Code-Protect bit<br>1 = User program memory is not code-protected<br>0 = User program memory is code-protected |
| GWRP | CW1<12> | General Segment Write-Protect bit<br>1 = User program memory is not write-protected<br>0 = User program memory is write-protected |
| ICS | CW1<8> | ICD Communication Channel Select bit<br>1 = Communicate on PGC2/EMUC2 and PGD2/EMUD2<br>0 = Communicate on PGC1/EMUC1 and PGD1/EMUD1 |

**Note 1:** Available on 28 and 44-pin packages only.

**2:** Available only on 28 and 44-pin devices with a silicon revision of 3042h or higher.

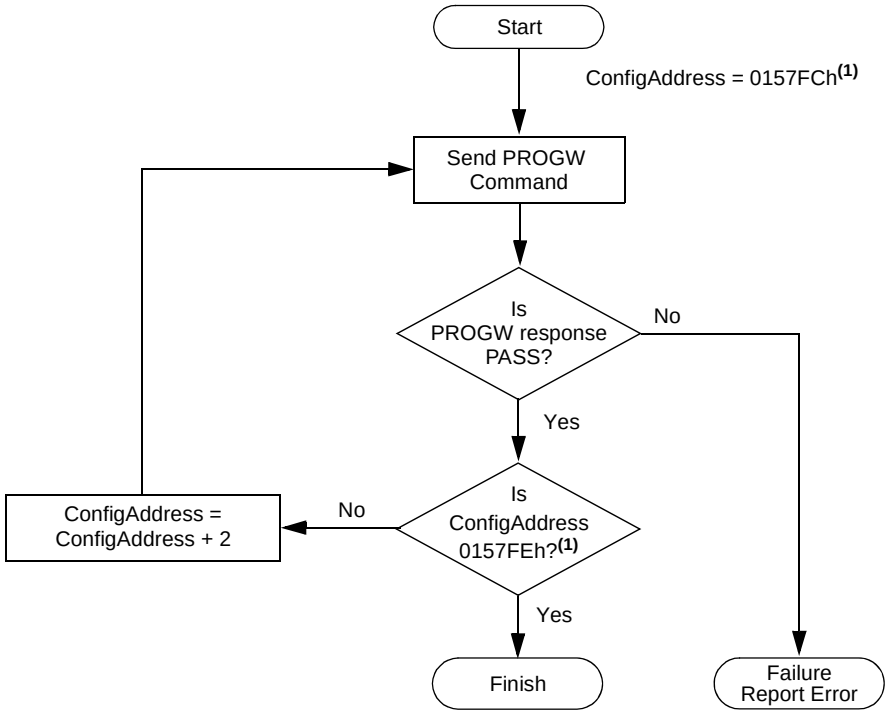
**TABLE 4-2: PIC24FJXXXGA0XX FAMILY CONFIGURATION BITS DESCRIPTION (CONTINUED)**

| Bit Field | Register | Description |
|---|---|---|
| ICS(1) | CW1<8> | ICD Pin Placement Select bit<br>11 = ICD EMUC/EMUD pins are shared with PGC1/PGD1<br>10 = ICD EMUC/EMUD pins are shared with PGC2/PGD2<br>01 = ICD EMUC/EMUD pins are shared with PGC3/PGD3<br>00 = Reserved; do not use |
| IESO | CW2<15> | Internal External Switchover bit<br>1 = Two-Speed Start-up enabled<br>0 = Two-Speed Start-up disabled |
| IOL1WAY(1) | CW2<4> | IOLOCK Bit One-Way Set Enable bit<br>0 = The OSCCON<IOLOCK> bit can be set and cleared as needed (provided an unlocking sequence is executed)<br>1 = The OSCCON<IOLOCK> bit can only be set once (provided an unlocking sequence is executed). Once IOLOCK is set, this prevents any possible future RP register changes |
| JTAGEN | CW1<14> | JTAG Enable bit<br>1 = JTAG enabled<br>0 = JTAG disabled |
| OSCIOFNC | CW2<5> | OSC2 Pin Function bit (except in XT and HS modes)<br>1 = OSC2 is clock output<br>0 = OSC2 is general purpose digital I/O pin |
| SOSCSEL1:<br>SOSCSEL0(2) | CW2<12:11> | Secondary Oscillator Power Mode Select bits<br>11 = Default (high drive strength) mode<br>01 = Low-Power (low drive strength) mode<br>x0 = Reserved; do not use |
| POSCMD1:<br>POSCMD0 | CW2<1:0> | Primary Oscillator Mode Select bits<br>11 = Primary oscillator disabled<br>10 = HS Crystal Oscillator mode<br>01 = XT Crystal Oscillator mode<br>00 = EC (External Clock) mode |
| WDTPOST3:<br>WDTPOST0 | CW1<3:0> | Watchdog Timer Prescaler bit<br>1111 = 1:32,768<br>1110 = 1:16,384<br>.<br>.<br>.<br>0001 = 1:2<br>0000 = 1:1 |
| WDTPRE | CW1<4> | Watchdog Timer Postscaler bit<br>1 = 1:128<br>0 = 1:32 |
| WINDIS | CW1<6> | Windowed WDT bit<br>1 = Watchdog Timer in Non-Window mode<br>0 = Watchdog Timer in Window mode; FWDTEN must be '1' |
| WUTSEL1:<br>WUTSEL0(2) | CW2<14:13> | Voltage Regulator Standby Mode Wake-up Time Select bits<br>11 = Default regulator wake time used<br>01 = Fast regulator wake time used<br>x0 = Reserved; do not use |

**Note 1:** Available on 28 and 44-pin packages only.

**2:** Available only on 28 and 44-pin devices with a silicon revision of 3042h or higher.

**FIGURE 4-5: CONFIGURATION BIT PROGRAMMING FLOW**




**Note 1:** Configuration Word addresses for PIC24FJ128GA devices are shown. Refer to Table 2-2 for others.

## 4.7 Exiting Enhanced ICSP Mode

Exiting Program/Verify mode is done by removing $V_{IH}$ from $\overline{MCLR}$, as shown in Figure 4-6. The only requirement for exit is that an interval, P16, should elapse between the last clock and program signals on PGCx and PGDx before removing $V_{IH}$.

**FIGURE 4-6: EXITING ENHANCED ICSP™ MODE**

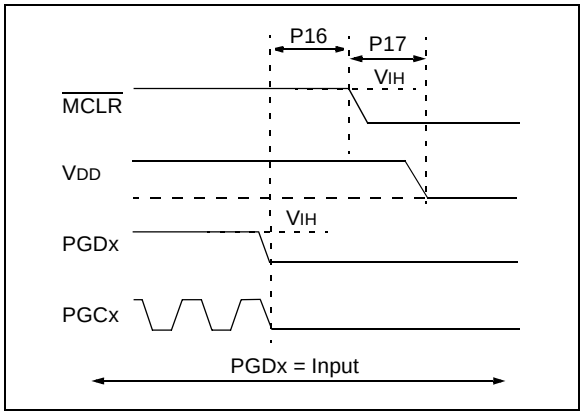
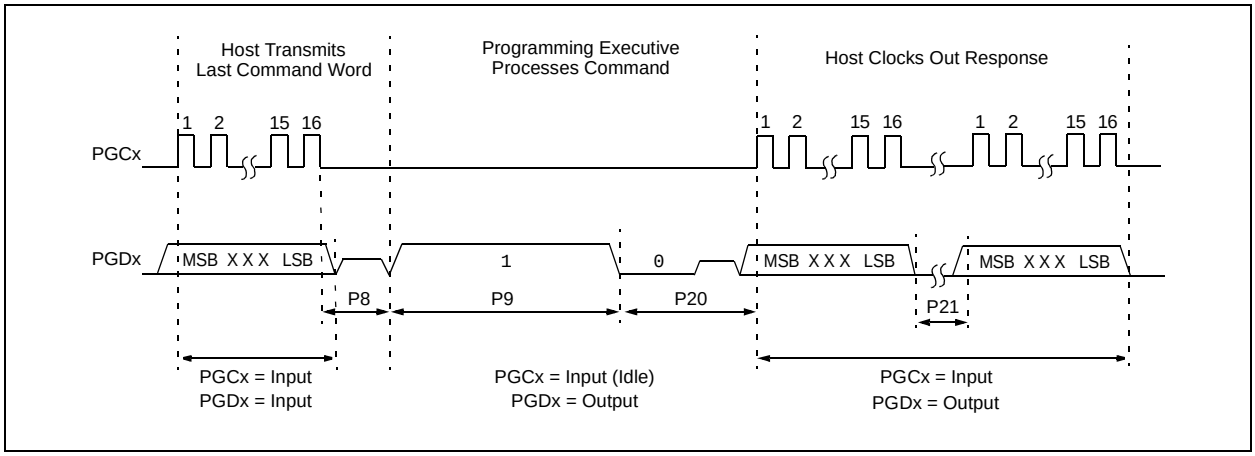


**FIGURE 5-3: PROGRAMMING EXECUTIVE – PROGRAMMER COMMUNICATION PROTOCOL**




## 5.2 Programming Executive Commands

The programming executive command set is shown in Table 5-1. This table contains the opcode, mnemonic, length, time-out and description for each command. Functional details on each command are provided in **Section 5.2.4 "Command Descriptions"**.

### 5.2.1 COMMAND FORMAT

All programming executive commands have a general format consisting of a 16-bit header and any required data for the command (see Figure 5-4). The 16-bit header consists of a 4-bit opcode field, which is used to identify the command, followed by a 12-bit command length field.

**FIGURE 5-4: COMMAND FORMAT**

| 15 12 | 11 0 |
|---|---|
| Opcode | Length |
| Command Data First Word (if required) | |
| • | |
| • | |
| Command Data Last Word (if required) | |

The command opcode must match one of those in the command set. Any command that is received which does not match the list in Table 5-1 will return a "NACK" response (see **Section 5.3.1.1 "Opcode Field"**).

The command length is represented in 16-bit words since the SPI operates in 16-bit mode. The programming executive uses the command length field to determine the number of words to read from the SPI port. If the value of this field is incorrect, the command will not be properly received by the programming executive.

### 5.2.2 PACKED DATA FORMAT

When 24-bit instruction words are transferred across the 16-bit SPI interface, they are packed to conserve space using the format shown in Figure 5-5. This format minimizes traffic over the SPI and provides the programming executive with data that is properly aligned for performing table write operations.

**FIGURE 5-5: PACKED INSTRUCTION WORD FORMAT**

| 15 8 | 7 0 |
|---|---|
| LSW1 | |
| MSB2 | MSB1 |
| LSW2 | |

LSWx: Least Significant 16 bits of instruction word
MSBx: Most Significant Bytes of instruction word

**Note:** When the number of instruction words transferred is odd, MSB2 is zero and LSW2 can not be transmitted.

### 5.2.3 PROGRAMMING EXECUTIVE ERROR HANDLING

The programming executive will "NACK" all unsupported commands. Additionally, due to the memory constraints of the programming executive, no checking is performed on the data contained in the programmer command. It is the responsibility of the programmer to command the programming executive with valid command arguments or the programming operation may fail. Additional information on error handling is provided in **Section 5.3.1.3 "QE_Code Field"**.

### 5.2.10 PROGW COMMAND

| 15 12 | 11 8 | 7 0 |
|---|---|---|
| Opcode | Length | |
| Data_MSB | | Addr_MSB |
| Addr_LS | | |
| Data_LS | | |

| Field | Description |
|---|---|
| Opcode | Dh |
| Length | 4h |
| Reserved | 0h |
| Addr_MSB | MSB of 24-bit destination address |
| Addr_LS | Least Significant 16 bits of 24-bit destination address |
| Data_MSB | MSB of 24-bit data |
| Data_LS | Least Significant 16 bits of 24-bit data |

The PROGW command instructs the programming executive to program one word of code memory (3 bytes) to the specific memory address.

After the word has been programmed to code memory, the programming executive verifies the programmed data against the data in the command.

**Expected Response (2 words):**

1600h

0002h

### 5.2.11 QBLANK COMMAND

| 15 12 | 11 0 |
|---|---|
| Opcode | Length |
| PSize_MSW | |
| PSize_LSW | |

| Field | Description |
|---|---|
| Opcode | Ah |
| Length | 3h |
| PSize | Length of program memory to check in 24-bit words plus one (max. of 49152) |

The QBLANK command queries the programming executive to determine if the contents of code memory and code-protect Configuration bits (GCP and GWRP) are blank (contain all ‘1’s). The size of code memory to check must be specified in the command.

The Blank Check for code memory begins at 0h and advances toward larger addresses for the specified number of instruction words.

QBLANK returns a QE_Code of F0h if the specified code memory and code-protect bits are blank; otherwise, QBLANK returns a QE_Code of 0Fh.

**Expected Response (2 words for blank device):**

1AF0h

0002h

**Expected Response (2 words for non-blank device):**

1A0Fh

0002h

| | |
|---|---|
| **Note:** | QBLANK does not check the system operation Configuration bits, since these bits are not set to ‘1’ when a Chip Erase is performed. |

### 5.2.12 QVER COMMAND

| 15 12 | 11 0 |
|---|---|
| Opcode | Length |

| Field | Description |
|---|---|
| Opcode | Bh |
| Length | 1h |

The QVER command queries the version of the programming executive software stored in test memory. The “version.revision” information is returned in the response’s QE_Code using a single byte with the following format: main version in upper nibble and revision in the lower nibble (i.e., 23h means version 2.3 of programming executive software).

**Expected Response (2 words):**

1BMNh (where “MN” stands for version M.N)

0002h

## 5.3 Programming Executive Responses

The programming executive sends a response to the programmer for each command that it receives. The response indicates if the command was processed correctly. It includes any required response data or error data.

The programming executive response set is shown in Table 5-2. This table contains the opcode, mnemonic and description for each response. The response format is described in **Section 5.3.1 “Response Format”**.

**TABLE 5-2: PROGRAMMING EXECUTIVE RESPONSE OP CODES**

| Opcode | Mnemonic | Description |
|---|---|---|
| 1h | PASS | Command successfully processed |
| 2h | FAIL | Command unsuccessfully processed |
| 3h | NACK | Command not known |

### 5.3.1 RESPONSE FORMAT

All programming executive responses have a general format consisting of a two-word header and any required data for the command.

| 15 12 | 11 8 | 7 0 |
|---|---|---|
| Opcode | Last_Cmd | QE_Code |
| Length | | |
| D_1 (if applicable) | | |
| ... | | |
| D_N (if applicable) | | |

| Field | Description |
|---|---|
| Opcode | Response opcode |
| Last_Cmd | Programmer command that generated the response |
| QE_Code | Query code or error code. |
| Length | Response length in 16-bit words (includes 2 header words) |
| D_1 | First 16-bit data word (if applicable) |
| D_N | Last 16-bit data word (if applicable) |

#### 5.3.1.1 Opcode Field

The opcode is a 4-bit field in the first word of the response. The opcode indicates how the command was processed (see Table 5-2). If the command was processed successfully, the response opcode is PASS. If there was an error in processing the command, the response opcode is FAIL and the QE_Code indicates the reason for the failure. If the command sent to the programming executive is not identified, the programming executive returns a NACK response.

#### 5.3.1.2 Last_Cmd Field

The Last_Cmd is a 4-bit field in the first word of the response and indicates the command that the programming executive processed. Since the programming executive can only process one command at a time, this field is technically not required. However, it can be used to verify that the programming executive correctly received the command that the programmer transmitted.

#### 5.3.1.3 QE_Code Field

The QE_Code is a byte in the first word of the response. This byte is used to return data for query commands and error codes for all other commands.

When the programming executive processes one of the two query commands (QBLANK or QVER), the returned opcode is always PASS and the QE_Code holds the query response data. The format of the QE_Code for both queries is shown in Table 5-3.

**TABLE 5-3: QE_Code FOR QUERIES**

| Query | QE_Code |
|---|---|
| QBLANK | 0Fh = Code memory is NOT blank<br>F0h = Code memory is blank |
| QVER | 0xMN, where programming executive software version = M.N (i.e., 32h means software version 3.2) |

When the programming executive processes any command other than a query, the QE_Code represents an error code. Supported error codes are shown in Table 5-4. If a command is successfully processed, the returned QE_Code is set to 0h, which indicates that there was no error in the command processing. If the verify of the programming for the PROGP or PROGC command fails, the QE_Code is set to 1h. For all other programming executive errors, the QE_Code is 2h.

**TABLE 5-4: QE_Code FOR NON-QUERY COMMANDS**

| QE_Code | Description |
|---|---|
| 0h | No error |
| 1h | Verify failed |
| 2h | Other error |

#### 5.3.1.4 Response Length

The response length indicates the length of the programming executive's response in 16-bit words. This field includes the 2 words of the response header.

With the exception of the response for the READP command, the length of each response is only 2 words.

The response to the READP command uses the packed instruction word format described in **Section 5.2.2 "Packed Data Format"**. When reading an odd number of program memory words (N odd), the response to the READP command is (3 * (N + 1)/2 + 2) words. When reading an even number of program memory words (N even), the response to the READP command is (3 * N/2 + 2) words.

**TABLE 5-5: PROGRAMMING THE PROGRAMMING EXECUTIVE (CONTINUED)**

| Command (Binary) | Data (Hex) | Description |
|---|---|---|
| **Step 15:** Set the Read Pointer (W6) and load the (next four write) latches. | | |
| 0000 | EB0300 | CLR W6 |
| 0000 | 000000 | NOP |
| 0000 | BB0BB6 | TBLWTL [W6++], [W7] |
| 0000 | 000000 | NOP |
| 0000 | 000000 | NOP |
| 0000 | BBDBB6 | TBLWTH.B [W6++], [W7++] |
| 0000 | 000000 | NOP |
| 0000 | 000000 | NOP |
| 0000 | BBEBB6 | TBLWTH.B [W6++], [++W7] |
| 0000 | 000000 | NOP |
| 0000 | 000000 | NOP |
| 0000 | BB1BB6 | TBLWTL [W6++], [W7++] |
| 0000 | 000000 | NOP |
| 0000 | 000000 | NOP |
| 0000 | BB0BB6 | TBLWTL [W6++], [W7] |
| 0000 | 000000 | NOP |
| 0000 | 000000 | NOP |
| 0000 | BBDBB6 | TBLWTH.B [W6++], [W7++] |
| 0000 | 000000 | NOP |
| 0000 | 000000 | NOP |
| 0000 | BBEBB6 | TBLWTH.B [W6++], [++W7] |
| 0000 | 000000 | NOP |
| 0000 | 000000 | NOP |
| 0000 | BB1BB6 | TBLWTL [W6++], [W7++] |
| 0000 | 000000 | NOP |
| 0000 | 000000 | NOP |
| **Step 16:** Repeat Steps 14-15, sixteen times, to load the write latches for the 64 instructions. | | |
| **Step 17:** Initiate the programming cycle. | | |
| 0000 | A8E761 | BSET NVMCON, #15 |
| 0000 | 000000 | NOP |
| 0000 | 000000 | NOP |
| **Step 18:** Repeat this step to poll the WR bit (bit 15 of NVMCON) until it is cleared by the hardware. | | |
| 0000 | 040200 | GOTO 0x200 |
| 0000 | 000000 | NOP |
| 0000 | 803B02 | MOV NVMCON, W2 |
| 0000 | 883C22 | MOV W2, VISI |
| 0000 | 000000 | NOP |
| 0001 | <VISI> | Clock out contents of the VISI register. |
| 0000 | 000000 | NOP |
| **Step 19:** Reset the device internal PC. | | |
| 0000 | 040200 | GOTO 0x200 |
| 0000 | 000000 | NOP |
| **Step 20:** Repeat Steps 14-19 until all 16 rows of executive memory have been programmed. On the final row, make sure to initialize the write latches at the Diagnostic and Calibration Words locations with 0xFFFFFF to ensure that the calibration is not overwritten. | | |

# 6.0 DEVICE DETAILS

## 6.1 Device ID

The Device ID region of memory can be used to determine mask, variant and manufacturing information about the chip. The Device ID region is 2 x 16 bits and it can be read using the READC command. This region of memory is read-only and can also be read when code protection is enabled.

Table 6-1 shows the Device ID for each device, Table 6-2 shows the Device ID registers and Table 6-3 describes the bit field of each register.

**TABLE 6-1: DEVICE IDs**

| Device | DEVID |
|---|---|
| PIC24FJ16GA002 | 0444h |
| PIC24FJ16GA004 | 044Ch |
| PIC24FJ32GA002 | 0445h |
| PIC24FJ32GA004 | 044Dh |
| PIC24FJ48GA002 | 0446h |
| PIC24FJ48GA004 | 044Eh |
| PIC24FJ64GA002 | 0447h |
| PIC24FJ64GA004 | 044Fh |
| PIC24FJ64GA006 | 0405h |
| PIC24FJ64GA008 | 0408h |
| PIC24FJ64GA010 | 040Bh |
| PIC24FJ96GA006 | 0406h |
| PIC24FJ96GA008 | 0409h |
| PIC24FJ96GA010 | 040Ch |
| PIC24FJ128GAGA006 | 0407h |
| PIC24FJ128GAGA008 | 040Ah |
| PIC24FJ128GAGA010 | 040Dh |

**TABLE 6-2: PIC24FJXXXGA0XX DEVICE ID REGISTERS**

| Address | Name | Bit | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| FF0000h | DEVID | — | | FAMID<7:0> | | | | | | | | DEV<5:0> | | | | | |
| FF0002h | DEVREV | — | | | | | | | MAJRV<2:0> | | | — | | | DOT<2:0> | | |

**TABLE 6-3: DEVICE ID BIT DESCRIPTIONS**

| Bit Field | Register | Description |
|---|---|---|
| FAMID<7:0> | DEVID | Encodes the family ID of the device |
| DEV<5:0> | DEVID | Encodes the individual ID of the device |
| MAJRV<2:0> | DEVREV | Encodes the major revision number of the device |
| DOT<2:0> | DEVREV | Encodes the minor revision number of the device |

**TABLE 6-4: CHECKSUM COMPUTATION (CONTINUED)**

| Device | Read Code Protection | Checksum Computation | Erased Checksum Value | Checksum with 0xAAAAAA at 0x0 and Last Code Address |
|---|---|---|---|---|
| PIC24FJ128GAGA006 | Disabled | CFGB + SUM(0:0157FB) | 0xF8CC | 0xF6CE |
| | Enabled | 0 | 0x0000 | 0x0000 |
| PIC24FJ128GAGA008 | Disabled | CFGB + SUM(0:0157FB) | 0xF8CC | 0xF6CE |
| | Enabled | 0 | 0x0000 | 0x0000 |
| PIC24FJ128GAGA010 | Disabled | CFGB + SUM(0:0157FB) | 0xF8CC | 0xF6CE |
| | Enabled | 0 | 0x0000 | 0x0000 |

**Legend:** Item Description
SUM[a:b] = Byte sum of locations, a to b inclusive (all 3 bytes of code memory)
CFGB = Configuration Block (masked),
64/80/100-Pin Devices = Byte sum of (CW1 & 0x7DDF + CW2 & 0x87E3)
28/44-Pin Devices = Byte sum of (CW1 & 0x7FDF + CW2 & 0xFFF7)

**Note:** CW1 address is last location of implemented program memory; CW2 is (last location – 2).

## 7.0 AC/DC CHARACTERISTICS AND TIMING REQUIREMENTS

**Standard Operating Conditions**
Operating Temperature: 0°C to +70°C. Programming at +25°C is recommended.

| Param No. | Symbol | Characteristic | Min | Max | Units | Conditions |
|---|---|---|---|---|---|---|
| D111 | VDD | Supply Voltage During Programming | VDDCORE + 0.1 | 3.60 | V | Normal programming(1,2) |
| D112 | IPP | Programming Current on $\overline{MCLR}$ | — | 5 | μA | |
| D113 | IDDP | Supply Current During Programming | — | 2 | 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 | — | 0.4 | V | IOL = 8.5 mA @ 3.6V |
| D090 | VOH | Output High Voltage | 3.0 | — | V | IOH = -3.0 mA @ 3.6V |
| D012 | CIO | Capacitive Loading on I/O pin (PGDx) | — | 50 | pF | To meet AC specifications |
| D013 | CF | Filter Capacitor Value on VCAP | 4.7 | 10 | μF | Required for controller core |
| P1 | TPGC | Serial Clock (PGCx) Period | 100 | — | ns | |
| P1A | TPGCL | Serial Clock (PGCx) Low Time | 40 | — | ns | |
| P1B | TPGCH | Serial Clock (PGCx) High Time | 40 | — | ns | |
| P2 | TSET1 | Input Data Setup Time to Serial Clock ↑ | 15 | — | ns | |
| P3 | THLD1 | Input Data Hold Time from PGCx ↑ | 15 | — | ns | |
| P4 | TDLY1 | Delay Between 4-Bit Command and Command Operand | 40 | — | ns | |
| P4A | TDLY1A | Delay Between 4-Bit Command Operand and Next 4-Bit Command | 40 | — | ns | |
| P5 | TDLY2 | Delay Between Last PGCx ↓ of Command Byte to First PGCx ↑ of Read of Data Word | 20 | — | ns | |
| P6 | TSET2 | VDD ↑ Setup Time to $\overline{MCLR}$ ↑ | 100 | — | ns | |
| P7 | THLD2 | Input Data Hold Time from $\overline{MCLR}$ ↑ | 25 | — | ms | |
| P8 | TDLY3 | Delay Between Last PGCx ↓ of Command Byte to PGDx ↑ by Programming Executive | 12 | — | μs | |
| P9 | TDLY4 | Programming Executive Command Processing Time | 40 | — | μs | |
| P10 | TDLY6 | PGCx Low Time After Programming | 400 | — | ns | |
| P11 | TDLY7 | Chip Erase Time | 400 | — | ms | |
| P12 | TDLY8 | Page Erase Time | 40 | — | ms | |
| P13 | TDLY9 | Row Programming Time | 2 | — | ms | |
| P14 | TR | $\overline{MCLR}$ Rise Time to Enter ICSP™ mode | — | 1.0 | μs | |
| P15 | TVALID | Data Out Valid from PGCx ↑ | 10 | — | ns | |
| P16 | TDLY10 | Delay Between Last PGCx ↓ and $\overline{MCLR}$ ↓ | 0 | — | s | |
| P17 | THLD3 | $\overline{MCLR}$ ↓ to VDD ↓ | 100 | — | ns | |
| P18 | TKEY1 | Delay from First $\overline{MCLR}$ ↓ to First PGCx ↑ for Key Sequence on PGDx | 40 | — | ns | |
| P19 | TKEY2 | Delay from Last PGCx ↓ for Key Sequence on PGDx to Second $\overline{MCLR}$ ↑ | 1 | — | ms | |
| P20 | TDLY11 | Delay Between PGDx ↓ by Programming Executive to PGDx Driven by Host | 23 | — | μs | |
| P21 | TDLY12 | Delay Between Programming Executive Command Response Words | 8 | — | ns | |

**Note 1:** VDDCORE must be supplied to the VDDCORE/VCAP pin if the on-chip voltage regulator is disabled. See **Section 2.1 "Power Requirements"** for more information. (Minimum VDDCORE allowing Flash programming is 2.25V.)

**2:** VDD must also be supplied to the AVDD pins during programming. AVDD and AVSS should always be within ±0.3V of VDD and VSS, respectively.

# 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

**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: 91-80-4182-8400
Fax: 91-80-4182-8422

**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: 886-3-572-9526
Fax: 886-3-572-6459

**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

01/02/08