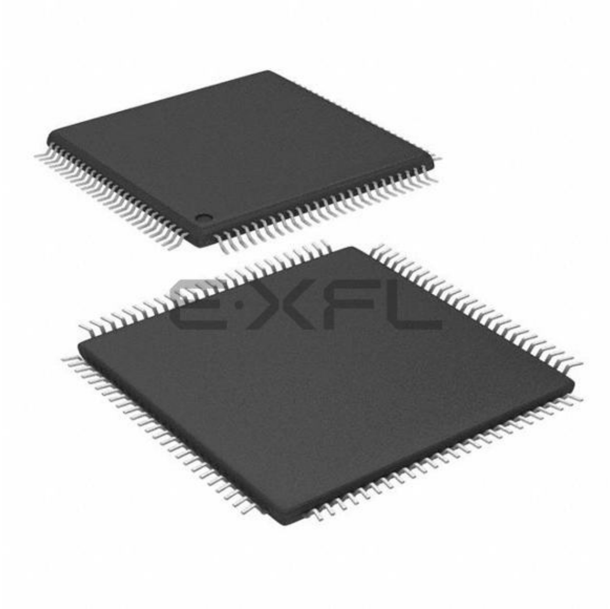
Microchip Technology - PIC24FJ128GA010-I/PT Datasheet



What is "Embedded - Microcontrollers"?

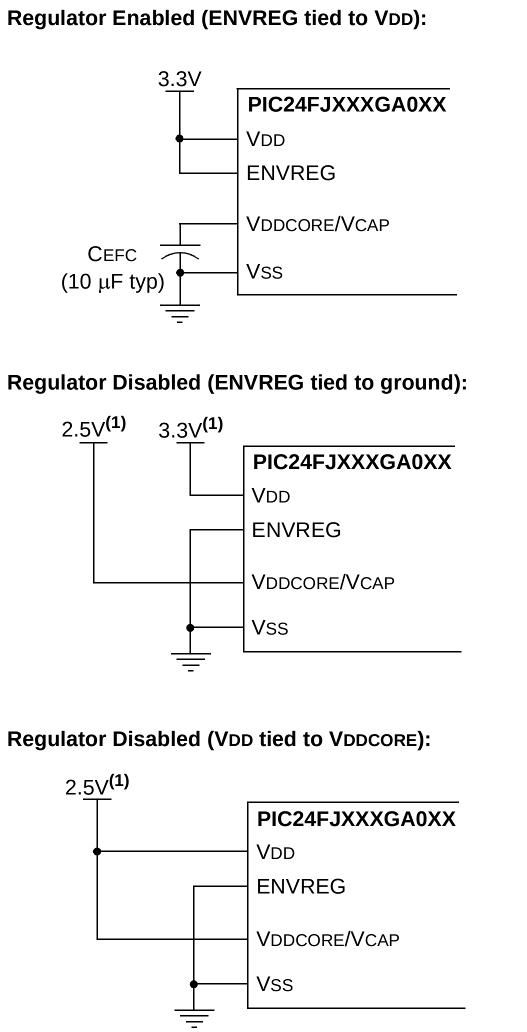
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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, PMP, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	84
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	100-TQFP
Supplier Device Package	100-TQFP (12x12)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic24fj128ga010-i-pt

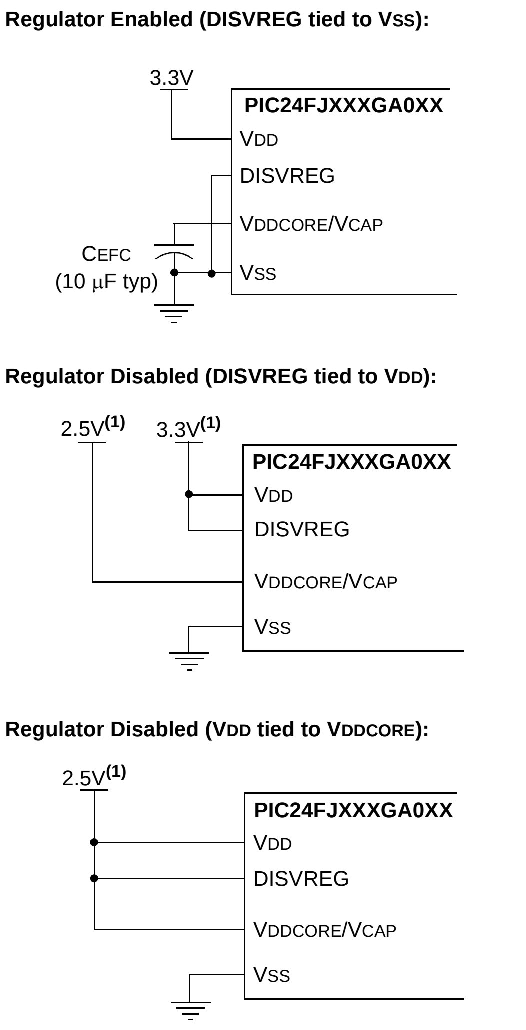
The regulator provides power to the core from the other VDD pins. A low-ESR capacitor (such as tantalum) must be connected to the VDDCORE pin (Figure 2-2 and Figure 2-3). This helps to maintain the stability of the regulator. The specifications for core voltage and capacitance are listed in **Section 7.0 “AC/DC Characteristics and Timing Requirements”**.

FIGURE 2-2: CONNECTIONS FOR THE ON-CHIP REGULATOR (64/80/100-PIN DEVICES)



Note 1: These are typical operating voltages. Refer to **Section 7.0 “AC/DC Characteristics and Timing Requirements”** for the full operating ranges of VDD and VDDCORE.

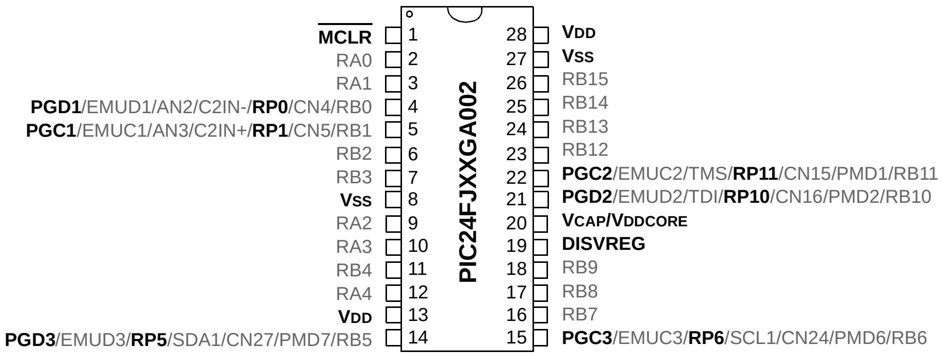
FIGURE 2-3: CONNECTIONS FOR THE ON-CHIP REGULATOR (28/44-PIN DEVICES)



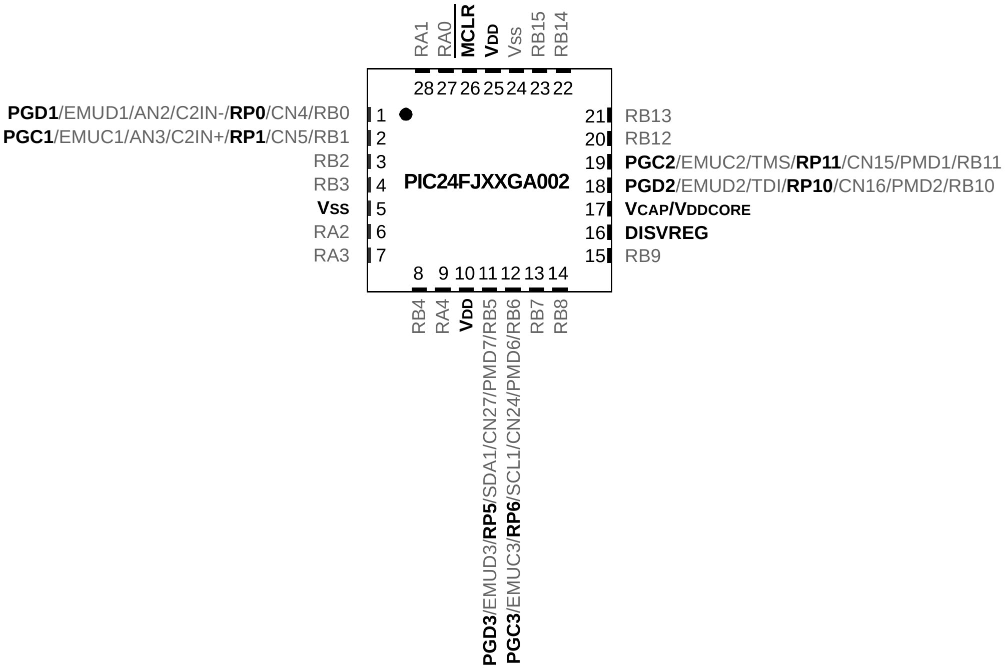
Note 1: These are typical operating voltages. Refer to **Section 7.0 “AC/DC Characteristics and Timing Requirements”** for the full operating ranges of VDD and VDDCORE.

Pin Diagrams

28-Pin PDIP, SSOP, SOIC


PIC24FJXXGA002
MCLR 1
RA0 2
RA1 3
PGD1/EMUD1/AN2/C2IN-/RP0/CN4/RB0 4
PGC1/EMUC1/AN3/C2IN+/RP1/CN5/RB1 5
RB2 6
RB3 7
VSS 8
RA2 9
RA3 10
RB4 11
RA4 12
VDD 13
PGD3/EMUD3/RP5/SDA1/CN27/PMD7/RB5 14
28 VDD
27 VSS
26 RB15
25 RB14
24 RB13
23 RB12
22 PGC2/EMUC2/TMS/RP11/CN15/PMD1/RB11
21 PGD2/EMUD2/TDI/RP10/CN16/PMD2/RB10
20 VCAP/VDDCORE
19 DISVREG
18 RB9
17 RB8
16 RB7
15 PGC3/EMUC3/RP6/SCL1/CN24/PMD6/RB6

28-Pin QFN$^{(1)}$

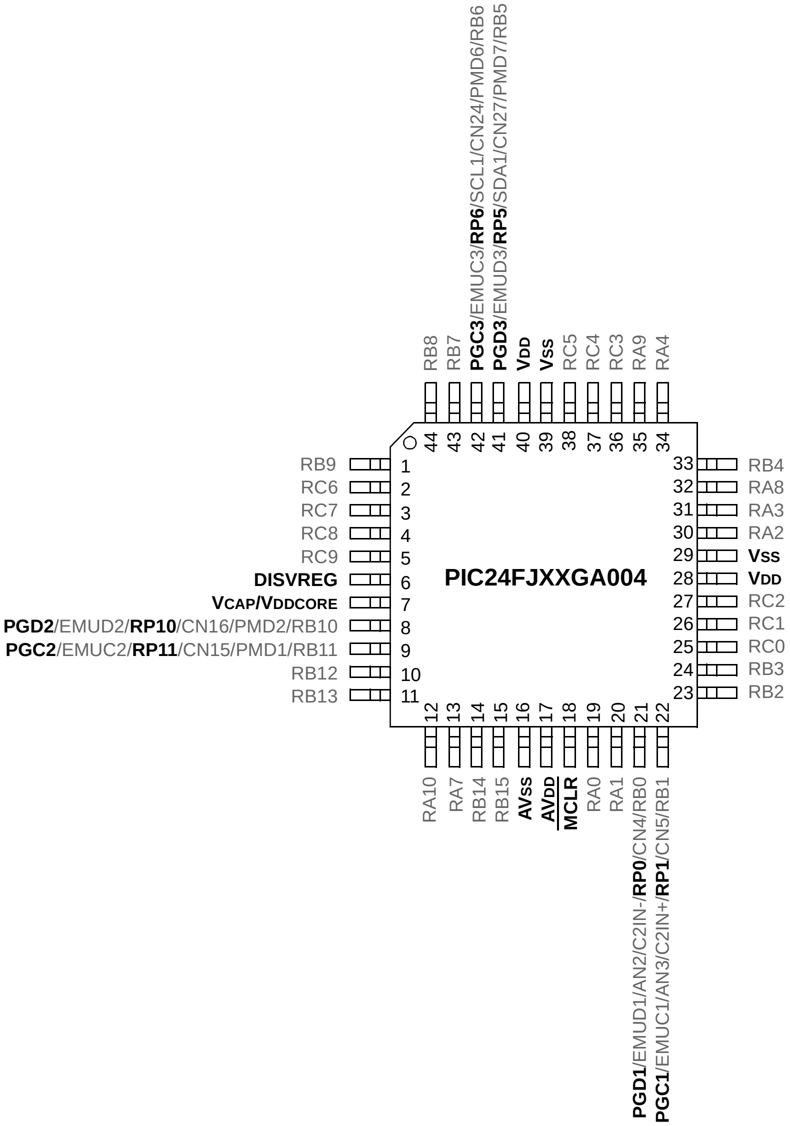

RA1 28
RA0 27
MCLR 26
VDD 25
VSS 24
RB15 23
RB14 22
PGD1/EMUD1/AN2/C2IN-/RP0/CN4/RB0 1
PGC1/EMUC1/AN3/C2IN+/RP1/CN5/RB1 2
RB2 3
RB3 4
VSS 5
RA2 6
RA3 7
PIC24FJXXGA002
21 RB13
20 RB12
19 PGC2/EMUC2/TMS/RP11/CN15/PMD1/RB11
18 PGD2/EMUD2/TDI/RP10/CN16/PMD2/RB10
17 VCAP/VDDCORE
16 DISVREG
15 RB9
8 RB4
9 RA4
10 VDD
11 PGD3/EMUD3/RP5/SDA1/CN27/PMD7/RB5
12 PGC3/EMUC3/RP6/SCL1/CN24/PMD6/RB6
13 RB7
14 RB8

Legend: **RPx** represents remappable peripheral pins.

Note 1: The bottom pad of QFN packages should be connected to VSS.

Pin Diagrams (Continued)

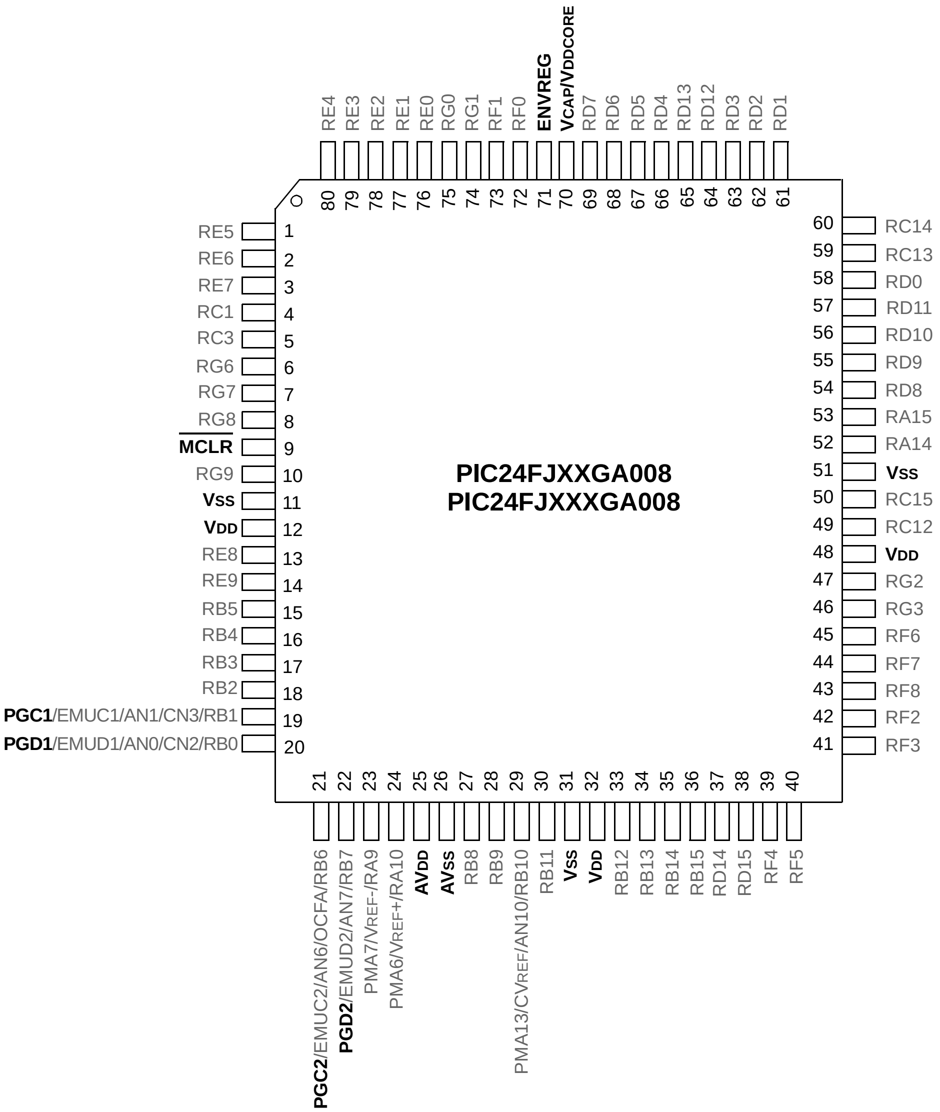
44-Pin TQFP



Legend: **RPx** represents remappable peripheral pins.

Pin Diagrams (Continued)

80-Pin TQFP



2.4 Memory Map

The program memory map extends from 000000h to FFFFFEh. Code storage is located at the base of the memory map and supports up to 44K instruction words (about 128 Kbytes). Table 2-3 shows the program memory size and number of erase and program blocks present in each device variant. Each erase block, or page, contains 512 instructions, and each program block, or row, contains 64 instructions.

Locations 800000h through 8007FEh are reserved for executive code memory. This region stores the programming executive and the debugging executive. The programming executive is used for device programming and the debugging executive is used for in-circuit debugging. This region of memory can not be used to store user code.

The last two implemented program memory locations are reserved for the device Configuration registers.

TABLE 2-2: FLASH CONFIGURATION WORD LOCATIONS FOR PIC24FJXXXGA0XX DEVICES

Device	Configuration Word Addresses	
	1	2
PIC24FJ16GA	002BFEh	002BFCh
PIC24FJ32GA	0057FEh	0057FCh
PIC24FJ48GA	0083FEh	0083FCh
PIC24FJ64GA	00ABFEh	00ABFCh
PIC24FJ96GA	00FFFEh	00FFFCh
PIC24FJ128GAGA	0157FEh	0157FCh

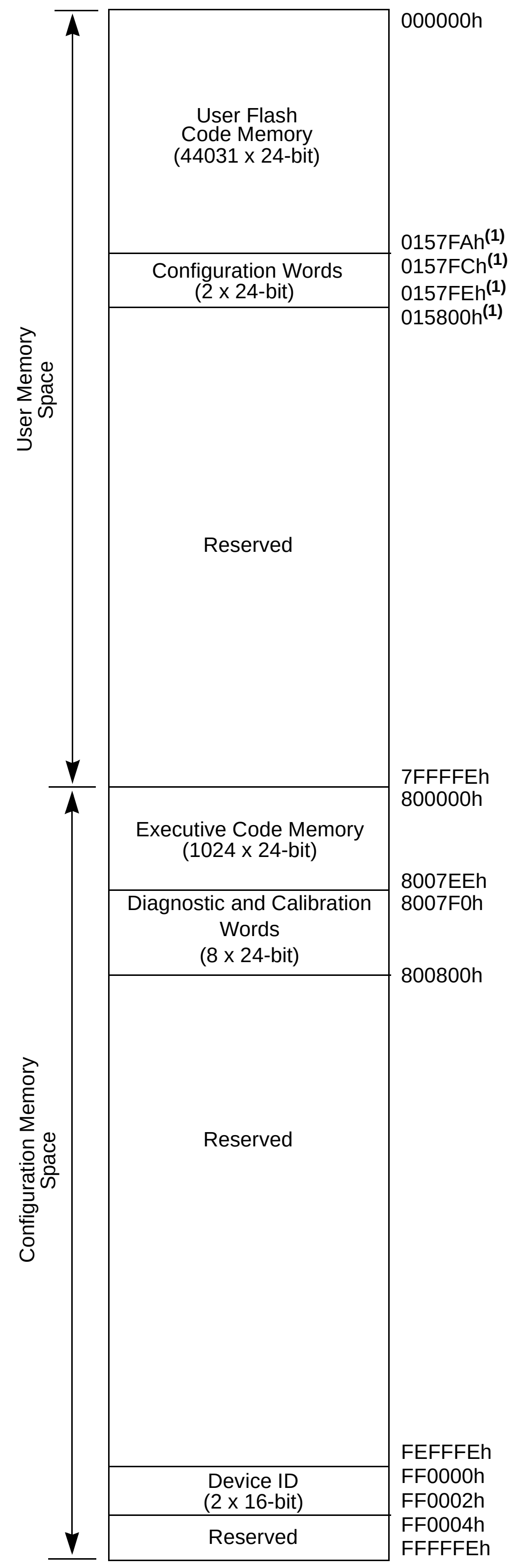
Locations, FF0000h and FF0002h, are reserved for the Device ID registers. These bits can be used by the programmer to identify what device type is being programmed. They are described in **Section 6.1 “Device ID”**. The Device ID registers read out normally, even after code protection is applied.

Figure 2-4 shows the memory map for the PIC24FJXXXGA0XX family variants.

TABLE 2-3: CODE MEMORY SIZE

Device	User Memory Address Limit (Instruction Words)	Write Blocks	Erase Blocks
PIC24FJ16GA	002BFEh (5.5K)	88	11
PIC24FJ32GA	0057FEh (11K)	176	22
PIC24FJ48GA	0083FEh (16.5K)	264	33
PIC24FJ64GA	00ABFEh (22K)	344	43
PIC24FJ96GA	00FFFEh (32K)	512	64
PIC24FJ128GA	0157FEh (44K)	688	86

FIGURE 2-4: PROGRAM MEMORY MAP



Note 1: The address boundaries for user Flash code memory are device dependent (see Table 2-3).

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.

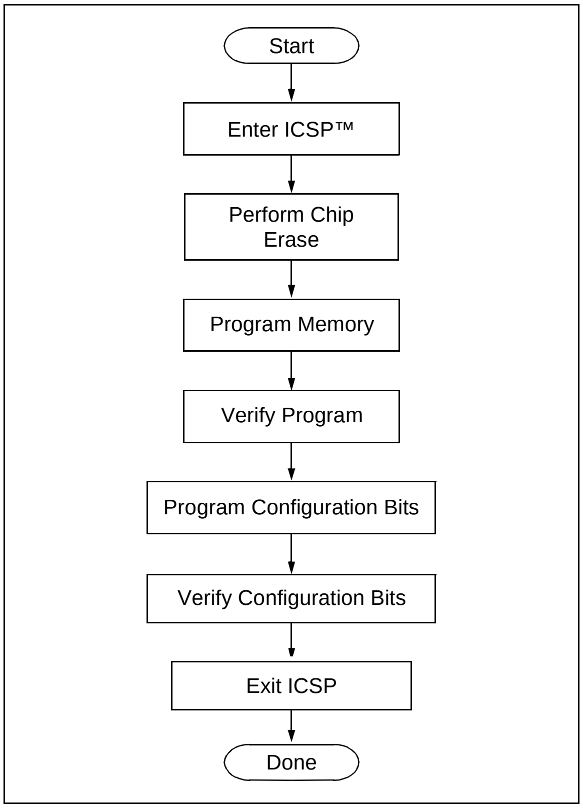
In ICSP mode, the system clock is taken from the PGCx pin, regardless of the device's oscillator Configuration bits. All instructions are shifted serially into an internal buffer, then loaded into the Instruction Register (IR) and executed. No program fetching occurs from internal memory. Instructions 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.

TABLE 3-4: SERIAL INSTRUCTION EXECUTION FOR CHIP ERASE

Command (Binary)	Data (Hex)	Description
Step 1: Exit the Reset vector.		

0000
0000
0000
	000000
	040200
	000000
		`NOP`
		`GOTO    0x200`
		`NOP`

Step 2: Set the NVMCON to erase all program memory.		

0000
0000
	2404FA
	883B0A
		`MOV     #0x404F, W10`
		`MOV     W10, NVMCON`

Step 3: Set TBLPAG and perform dummy table write to select what portions of memory are erased.		

0000
0000
0000
0000
0000
0000
	200000
	880190
	200000
	BB0800
	000000
	000000
		`MOV     #<PAGEVAL>, W0`
		`MOV     W0, TBLPAG`
		`MOV     #0x0000, W0`
		`TBLWTL  W0,[W0]`
		`NOP`
		`NOP`

Step 4: Initiate the erase cycle.		

0000
0000
0000
	A8E761
	000000
	000000
		`BSET    NVMCON, #WR`
		`NOP`
		`NOP`

Step 5: Repeat this step to poll the WR bit (bit 15 of NVMCON) until it is cleared by the hardware.		

0000
0000
0000
0000
0000
0001
0000
	040200
	000000
	803B02
	883C22
	000000
	<VISI>
	000000
		`GOTO    0x200`
		`NOP`
		`MOV     NVMCON, W2`
		`MOV     W2, VISI`
		`NOP`
		`Clock out contents of the VISI register.`
		`NOP`

3.6 Writing Code Memory

The procedure for writing code memory is the same as the procedure for writing the Configuration registers, except that 64 instruction words are programmed at a time. To facilitate this operation, working registers, W0:W5, are used as temporary holding registers for the data to be programmed.

Table 3-5 shows the ICSP programming details, 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 a full row of code memory. In Step 3, the 24-bit starting destination address for programming is loaded into the TBLPAG register and W7 register. (The upper byte of the starting destination address is stored in TBLPAG and the lower 16 bits of the destination address are stored in W7.)

To minimize the programming time, A packed instruction format is used (Figure 3-6).

In Step 4, four packed instruction words are stored in working registers, W0:W5, using the `MOV` instruction, and the Read Pointer, W6, is initialized. The contents of W0:W5 (holding the packed instruction word data) are shown in Figure 3-6.

In Step 5, eight `TBLWT` instructions are used to copy the data from W0:W5 to the write latches of code memory. Since code memory is programmed 64 instruction words at a time, Steps 4 and 5 are repeated 16 times to load all the write latches (Step 6).

After the write latches are loaded, programming is initiated by writing to the NVMCON register in Steps 7 and 8. In Step 9, the internal PC is reset to 200h. This is a precautionary measure to prevent the PC from incrementing into unimplemented memory when large devices are being programmed. Lastly, in Step 10, Steps 3-9 are repeated until all of code memory is programmed.

FIGURE 3-6: PACKED INSTRUCTION WORDS IN W<0:5>

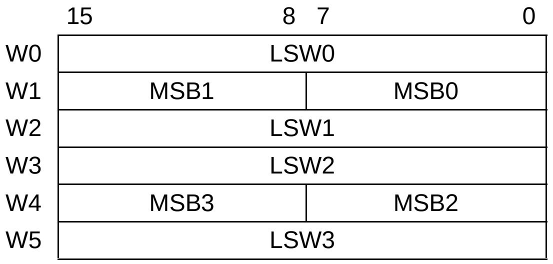


TABLE 3-5: SERIAL INSTRUCTION EXECUTION FOR WRITING CODE MEMORY

Command (Binary)	Data (Hex)	Description
Step 1: Exit the Reset vector.		

0000
0000
0000
	000000
	040200
	000000
		`NOP`
		`GOTO 0x200`
		`NOP`

Step 2: Set the NVMCON to program 64 instruction words.		

0000
0000
	24001A
	883B0A
		`MOV #0x4001, W10`
		`MOV W10, NVMCON`

Step 3: Initialize the Write Pointer (W7) for `TBLWT` instruction.		

0000
0000
0000
	200xx0
	880190
	2xxxx7
		`MOV #<DestinationAddress23:16>, W0`
		`MOV W0, TBLPAG`
		`MOV #<DestinationAddress15:0>, W7`

Step 4: Load W0:W5 with the next 4 instruction words to program.		

0000
0000
0000
0000
0000
0000
	2xxxx0
	2xxxx1
	2xxxx2
	2xxxx3
	2xxxx4
	2xxxx5
		`MOV #<LSW0>, W0`
		`MOV #<MSB1:MSB0>, W1`
		`MOV #<LSW1>, W2`
		`MOV #<LSW2>, W3`
		`MOV #<MSB3:MSB2>, W4`
		`MOV #<LSW3>, W5`

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'.

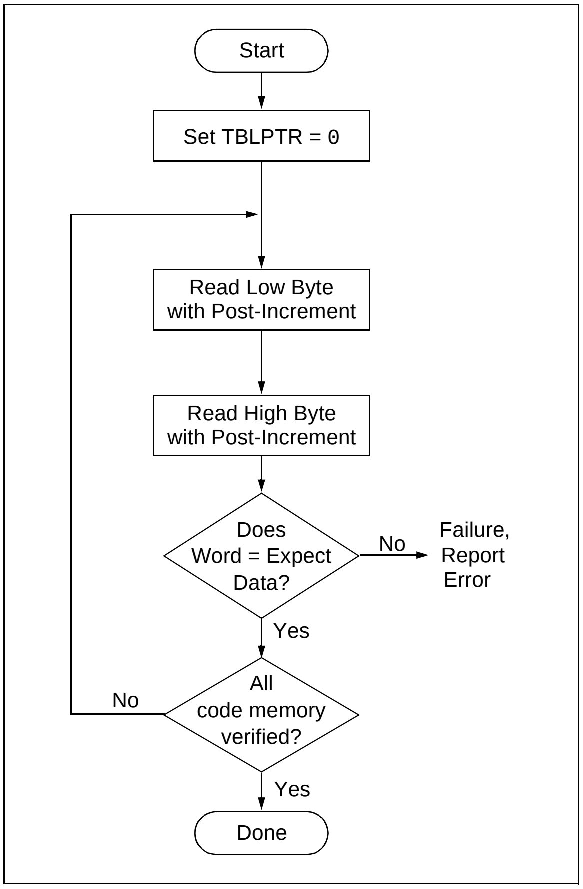
3.10 Verify Code Memory and Configuration Word

The verify step involves reading back the code memory space and comparing it against the copy held in the programmer's buffer. The Configuration registers are verified with the rest of the code.

The verify process is shown in the flowchart in Figure 3-8. Memory reads occur a single byte at a time, so two bytes must be read to compare against the word in the programmer's buffer. Refer to **Section 3.8 "Reading Code Memory"** for implementation details of reading code memory.

Note:	Because the Configuration registers include the device code protection bit, code memory should be verified immediately after writing if code protection is enabled. This is because the device will not be readable or verifiable if a device Reset occurs after the code-protect bit in CW1 has been cleared.

FIGURE 3-8: VERIFY CODE MEMORY FLOW


Start
Set TBLPTR = 0
Read Low Byte with Post-Increment
Read High Byte with Post-Increment
Does Word = Expect Data?
No
Failure, Report Error
Yes
All code memory verified?
No
Yes
Done

3.11 Reading the Application ID Word

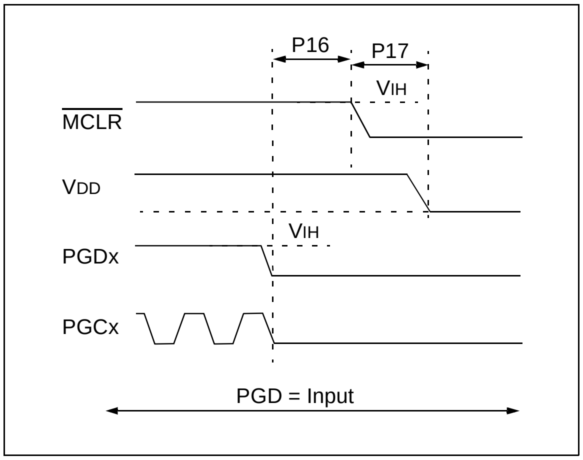
The Application ID Word is stored at address 8005BEh in executive code memory. To read this memory location, you must use the SIX control code to move this program memory location to the VISI register. Then, the REGOUT control code must be used to clock the contents of the VISI register out of the device. The corresponding control and instruction codes that must be serially transmitted to the device to perform this operation are shown in Table 3-10.

After the programmer has clocked out the Application ID Word, it must be inspected. If the Application ID has the value, BBh, the programming executive is resident in memory and the device can be programmed using the mechanism described in **Section 4.0 "Device Programming – Enhanced ICSP"**. However, if the Application ID has any other value, the programming executive is not resident in memory; it must be loaded to memory before the device can be programmed. The procedure for loading the programming executive to memory is described in **Section 5.4 "Programming the Programming Executive to Memory"**.

3.12 Exiting ICSP Mode

Exiting Program/Verify mode is done by removing V_{IH} from $\overline{MCLR}$, as shown in Figure 3-9. 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 3-9: EXITING ICSP™ MODE



4.6.2 PROGRAMMING METHODOLOGY

Configuration bits may be programmed a single byte at a time using the PROGW command. This command specifies the configuration data and Configuration register address. When Configuration bits are programmed, any unimplemented or reserved bits must be programmed with a '1'.

Two PROGW commands are required to program the Configuration bits. A flowchart for Configuration bit programming is shown in Figure 4-5.

Note:	If the General Segment Code-Protect bit (GCP) is programmed to '0', code memory is code-protected and can not be read. Code memory must be verified before enabling read protection. See **Section 4.6.4 "Code-Protect Configuration Bits"** for more information about code-protect Configuration bits.

4.6.3 PROGRAMMING VERIFICATION

After the Configuration bits are programmed, the contents of memory should be verified to ensure that the programming was successful. Verification requires the Configuration bits to be read back and compared against the copy held in the programmer's buffer. The READP command reads back the programmed Configuration bits and verifies that the programming was successful.

4.6.4 CODE-PROTECT CONFIGURATION BITS

CW1 Configuration register controls code protection for the PIC24FJXXXGA0XX family. Two forms of code protection are provided. One form prevents code memory from being written (write protection) and the other prevents code memory from being read (read protection).

GWRP (CW1<12>) controls write protection and GCP (CW1<13>) controls read protection. Protection is enabled when the respective bit is '0'.

Erasing sets GWRP and GCP to '1', which allows the device to be programmed.

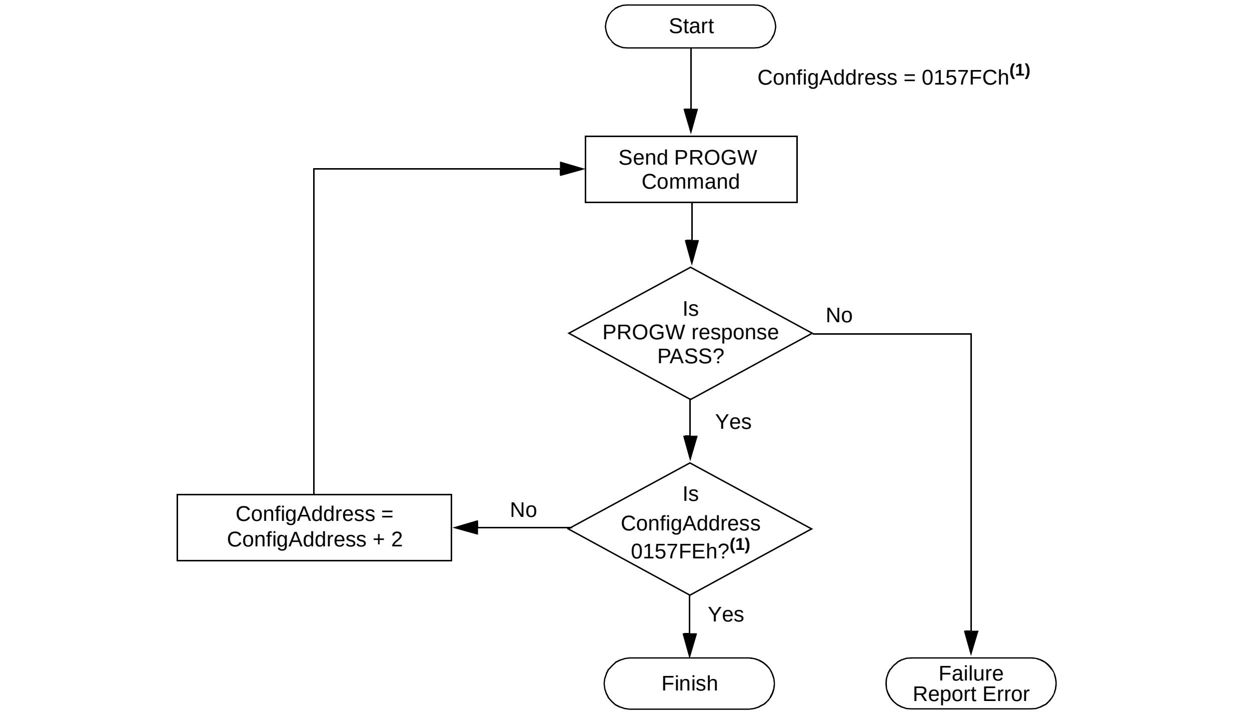
When write protection is enabled (GWRP = 0), any programming operation to code memory will fail.

When read protection is enabled (GCP = 0), any read from code memory will cause a 0h to be read, regardless of the actual contents of code memory. Since the programming executive always verifies what it programs, attempting to program code memory with read protection enabled also will result in failure.

It is imperative that both GWRP and GCP are '1' while the device is being programmed and verified. Only after the device is programmed and verified should either GWRP or GCP be programmed to '0' (see **Section 4.6 "Configuration Bits Programming"**).

Note:	Bulk Erasing in ICSP mode is the only way to reprogram code-protect bits from an ON state ('0') to an Off state ('1').

FIGURE 4-5: CONFIGURATION BIT PROGRAMMING FLOW

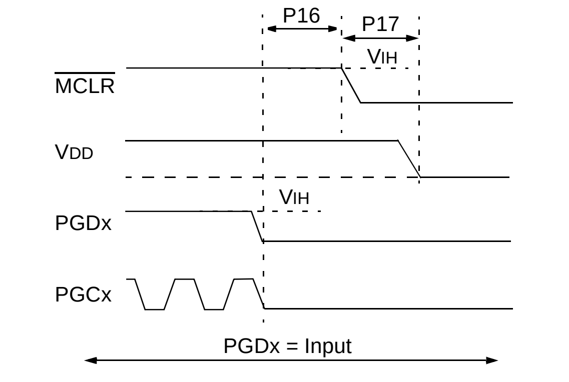

Start
ConfigAddress = 0157FCh(1)
Send PROGW Command
Is PROGW response PASS?
No
Yes
Is ConfigAddress 0157FEh?(1)
No
ConfigAddress = ConfigAddress + 2
Yes
Finish
Failure Report Error

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 VIH from $\overline{\text{MCLR}}$, 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 VIH.

FIGURE 4-6: EXITING ENHANCED ICSP™ MODE



5.2.6 READC COMMAND

15 12	11 8	7 0
Opcode	Length	
N		Addr_MSB
Addr_LS		

Field	Description
Opcode	1h
Length	3h
N	Number of 8-bit Device ID registers to read (max. of 256)
Addr_MSB	MSB of 24-bit source address
Addr_LS	Least Significant 16 bits of 24-bit source address

The READC command instructs the programming executive to read N or Device ID registers, starting from the 24-bit address specified by Addr_MSB and Addr_LS. This command can only be used to read 8-bit or 16-bit data.

When this command is used to read Device ID registers, the upper byte in every data word returned by the programming executive is 00h and the lower byte contains the Device ID register value.

Expected Response (4 + 3 * (N – 1)/2 words for N odd):

1100h

2 + N

Device ID Register 1

...

Device ID Register N

Note: Reading unimplemented memory will cause the programming executive to reset. Please ensure that only memory locations present on a particular device are accessed.

5.2.7 READP COMMAND

15 12	11 8	7 0
Opcode	Length	
N		
Reserved		Addr_MSB
Addr_LS		

Field	Description
Opcode	2h
Length	4h
N	Number of 24-bit instructions to read (max. of 32768)
Reserved	0h
Addr_MSB	MSB of 24-bit source address
Addr_LS	Least Significant 16 bits of 24-bit source address

The READP command instructs the programming executive to read N 24-bit words of code memory, including Configuration Words, starting from the 24-bit address specified by Addr_MSB and Addr_LS. This command can only be used to read 24-bit data. All data returned in response to this command uses the packed data format described in **Section 5.2.2 "Packed Data Format"**.

Expected Response (2 + 3 * N/2 words for N even):

1200h

2 + 3 * N/2

Least significant program memory word 1

...

Least significant data word N

Expected Response (4 + 3 * (N – 1)/2 words for N odd):

1200h

4 + 3 * (N – 1)/2

Least significant program memory word 1

...

MSB of program memory word N (zero padded)

Note: Reading unimplemented memory will cause the programming executive to reset. Please ensure that only memory locations present on a particular device are accessed.

5.2.8 PROGC COMMAND

<table>
<tr><td>15 12</td><td>11 8</td><td>7 0</td></tr>
<tr><td>Opcode</td><td colspan="2">Length</td></tr>
<tr><td colspan="2">Reserved</td><td>Addr_MSB</td></tr>
<tr><td colspan="3">Addr_LS</td></tr>
<tr><td colspan="3">Data</td></tr>
</table>

Field	Description
Opcode	4h
Length	4h
Reserved	0h
Addr_MSB	MSB of 24-bit destination address
Addr_LS	Least Significant 16 bits of 24-bit destination address
Data	8-bit data word

The PROGC command instructs the programming executive to program a single Device ID register located at the specified memory address.

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

Expected Response (2 words):

1400h
0002h

5.2.9 PROGP COMMAND

<table>
<tr><td>15 12</td><td>11 8</td><td>7 0</td></tr>
<tr><td>Opcode</td><td colspan="2">Length</td></tr>
<tr><td colspan="2">Reserved</td><td>Addr_MSB</td></tr>
<tr><td colspan="3">Addr_LS</td></tr>
<tr><td colspan="3">D_1</td></tr>
<tr><td colspan="3">D_2</td></tr>
<tr><td colspan="3">...</td></tr>
<tr><td colspan="3">D_96</td></tr>
</table>

Field	Description
Opcode	5h
Length	63h
Reserved	0h
Addr_MSB	MSB of 24-bit destination address
Addr_LS	Least Significant 16 bits of 24-bit destination address
D_1	16-bit data word 1
D_2	16-bit data word 2
...	16-bit data word 3 through 95
D_96	16-bit data word 96

The PROGP command instructs the programming executive to program one row of code memory, including Configuration Words (64 instruction words), to the specified memory address. Programming begins with the row address specified in the command. The destination address should be a multiple of 80h.

The data to program to memory, located in command words, D_1 through D_96, must be arranged using the packed instruction word format shown in Figure 5-5.

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

Expected Response (2 words):

1500h
0002h

Note:	Refer to Table 2-3 for code memory size information.

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 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 7: Repeat Steps 5 and 6 to erase the second page of executive memory. The W1 Pointer should be incremented by 400h to point to the second page.		
Step 8: Initialize TBLPAG and NVMCON to write stored diagnostic and calibration as single words. Initialize W1 and W2 as Write and Read Pointers to rewrite stored Diagnostic and Calibration Words.		
0000	200800	MOV #0x80, W0
0000	880190	MOV W0, TBLPAG
0000	240031	MOV #0x4003, W1
0000	883B01	MOV W1, NVMCON
0000	207F00	MOV #0x07F0, W1
0000	2000C2	MOV #0xC, W2
0000	000000	NOP
Step 9: Perform write of a single word of calibration data and initiate single-word write cycle.		
0000	BB18B2	TBLWTL [W2++], [W1++]
0000	000000	NOP
0000	000000	NOP
0000	A8E761	BSET NVMCON, #15
0000	000000	NOP
0000	000000	NOP
Step 10: 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	803B00	MOV NVMCON, W0
0000	883C20	MOV W0, VISI
0000	000000	NOP
0001	<VISI>	Clock out contents of VISI register.
0000	000000	NOP
Step 11: Repeat steps 9-10 seven more times to program the remainder of the Diagnostic and Calibration Words back into program memory.		
Step 12: Initialize the NVMCON to program 64 instruction words.		
0000	240010	MOV #0x4001, W0
0000	883B00	MOV W0, NVMCON
Step 13: Initialize TBLPAG and the Write Pointer (W7).		
0000	200800	MOV #0x80, W0
0000	880190	MOV W0, TBLPAG
0000	EB0380	CLR W7
0000	000000	NOP
Step 14: Load W0:W5 with the next four words of packed programming executive code and initialize W6 for programming. Programming starts from the base of executive memory (800000h) using W6 as a Read Pointer and W7 as a Write Pointer.		
0000	2<LSW0>0	MOV #<LSW0>, W0
0000	2<MSB1:MSB0>1	MOV #<MSB1:MSB0>, W1
0000	2<LSW1>2	MOV #<LSW1>, W2
0000	2<LSW2>3	MOV #<LSW2>, W3
0000	2<MSB3:MSB2>4	MOV #<MSB3:MSB2>, W4
0000	2<LSW3>5	MOV #<LSW3>, W5

5.4.2 PROGRAMMING VERIFICATION

After the programming executive has been programmed to executive memory using ICSP, it must be verified. Verification is performed by reading out the contents of executive memory and comparing it with the image of the programming executive stored in the programmer.

Reading the contents of executive memory can be performed using the same technique described in **Section 3.8 “Reading Code Memory”**. A procedure for reading executive memory is shown in Table 5-6. Note that in Step 2, the TBLPAG register is set to 80h, such that executive memory may be read. The last eight words of executive memory should be verified with stored values of the Diagnostic and Calibration Words to ensure accuracy.

TABLE 5-6: READING EXECUTIVE MEMORY

Command (Binary)	Data (Hex)	Description
Step 1: Exit the 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	200800	MOV #0x80, W0
0000	880190	MOV W0, TBLPAG
0000	EB0300	CLR 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 executive 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 the device internal PC.		
0000	040200	GOTO 0x200
0000	000000	NOP
Step 6: Repeat Steps 4 and 5 until all desired code memory is read.		

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 $\overline{\text{MCLR}}$	—	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 $\overline{\text{MCLR}}$ ↑	100	—	ns	
P7	THLD2	Input Data Hold Time from $\overline{\text{MCLR}}$ ↑	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	$\overline{\text{MCLR}}$ 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 $\overline{\text{MCLR}}$ ↓	0	—	s	
P17	THLD3	$\overline{\text{MCLR}}$ ↓ to VDD ↓	100	—	ns	
P18	TKEY1	Delay from First $\overline{\text{MCLR}}$ ↓ to First PGCx ↑ for Key Sequence on PGDx	40	—	ns	
P19	TKEY2	Delay from Last PGCx ↓ for Key Sequence on PGDx to Second $\overline{\text{MCLR}}$ ↑	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.

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

FilterLab, Linear Active Thermistor, MXDEV, MXLAB, SEEVAL, SmartSensor 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, In-Circuit Serial Programming, ICSP, ICEPIC, Mindi, MiWi, MPASM, MPLAB Certified logo, MPLIB, MPLINK, mTouch, PICkit, PICDEM, PICDEM.net, PICtail, PIC^{32} logo, PowerCal, PowerInfo, PowerMate, PowerTool, REAL ICE, rfLAB, Select Mode, Total Endurance, 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.

♻ Printed on recycled paper.

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
== ISO/TS 16949:2002 ==

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.