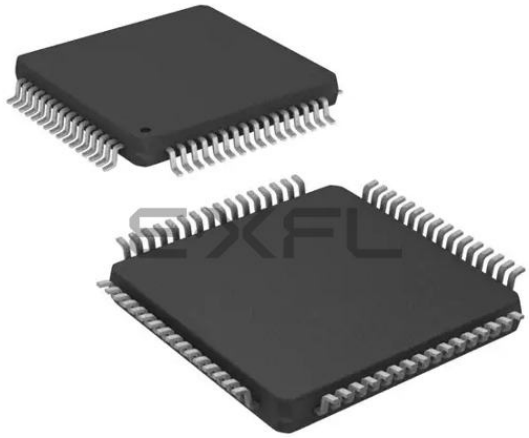
Microchip Technology - DSPIC30F6011AT-30I/PF Datasheet



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	dsPIC
Core Size	16-Bit
Speed	30 MIPs
Connectivity	CANbus, I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	52
Program Memory Size	132KB (44K x 24)
Program Memory Type	FLASH
EEPROM Size	2K x 8
RAM Size	6K x 8
Voltage - Supply (Vcc/Vdd)	2.5V ~ 5.5V
Data Converters	A/D 16x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-TQFP
Supplier Device Package	64-TQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/dspic30f6011at-30i-pf

5.6.3 PROGRAMMING VERIFICATION

Once the data EEPROM is programmed, the contents of memory can be verified to ensure that the programming was successful. Verification requires the data EEPROM to be read back and compared against the copy held in the programmer's buffer. The READD command reads back the programmed data EEPROM.

Alternatively, the programmer can perform the verification once the entire device is programmed using a checksum computation, as described in **Section 6.8 "Checksum Computation"**.

Note:	TBLRDL instructions executed within a REPEAT loop must not be used to read from Data EEPROM. Instead, it is recommended to use PSV access.

5.7 Configuration Bits Programming

5.7.1 OVERVIEW

The dsPIC30F has Configuration bits stored in seven 16-bit registers. These bits can be set or cleared to select various device configurations. There are two types of Configuration bits: system-operation bits and code-protect bits. The system-operation bits determine the power-on settings for system-level components such as the oscillator and Watchdog Timer. The code-protect bits prevent program memory from being read and written.

The FOSC Configuration register has three different register descriptions, based on the device. The FOSC Configuration register description for the dsPIC30F2010 and dsPIC30F6010/6011/6012/6013/6014 devices are shown in Table 5-4.

Note:	If user software performs an erase operation on the configuration fuse, it must be followed by a write operation to this fuse with the desired value, even if the desired value is the same as the state of the erased fuse.

The FOSC Configuration register description for the dsPIC30F4011/4012 and dsPIC30F5011/5013 devices is shown in Table 5-5.

The FOSC Configuration register description for all remaining devices (dsPIC30F2011/2012, dsPIC30F3010/3011/3012/3013, dsPIC30F3014/4013, dsPIC30F5015 and dsPIC30F6011A/6012A/6013A/ 6014A) is shown in Table 5-6. Always use the correct register descriptions for your target processor.

The FWDT, FBORPOR, FBS, FSS, FGS and FICD Configuration registers are not device-dependent. The register descriptions for these Configuration registers are shown in Table 5-7.

The Device Configuration register maps are shown in Table 5-8 through Table 5-11.

TABLE 5-4: FOSC CONFIGURATION BITS DESCRIPTION FOR dsPIC30F2010 AND dsPIC30F6010/6011/6012/6013/6014

Bit Field	Register	Description
FCKSM<1:0>	FOSC	**Clock Switching Mode** 1x = Clock switching is disabled, Fail-Safe Clock Monitor is disabled 01 = Clock switching is enabled, Fail-Safe Clock Monitor is disabled 00 = Clock switching is enabled, Fail-Safe Clock Monitor is enabled
FOS<1:0>	FOSC	**Oscillator Source Selection on POR** 11 = Primary Oscillator 10 = Internal Low-Power RC Oscillator 01 = Internal Fast RC Oscillator 00 = Low-Power 32 kHz Oscillator (Timer1 Oscillator)
FPR<3:0>	FOSC	**Primary Oscillator Mode** 1111 = ECIO w/PLL 16X – External Clock mode with 16X PLL. OSC2 pin is I/O 1110 = ECIO w/PLL 8X – External Clock mode with 8X PLL. OSC2 pin is I/O 1101 = ECIO w/PLL 4X – External Clock mode with 4X PLL. OSC2 pin is I/O 1100 = ECIO – External Clock mode. OSC2 pin is I/O 1011 = EC – External Clock mode. OSC2 pin is system clock output (FOSC/4) 1010 = Reserved (do not use) 1001 = ERC – External RC Oscillator mode. OSC2 pin is system clock output (FOSC/4) 1000 = ERCIO – External RC Oscillator mode. OSC2 pin is I/O 0111 = XT w/PLL 16X – XT Crystal Oscillator mode with 16X PLL 0110 = XT w/PLL 8X – XT Crystal Oscillator mode with 8X PLL 0101 = XT w/PLL 4X – XT Crystal Oscillator mode with 4X PLL 0100 = XT – XT Crystal Oscillator mode (4 MHz-10 MHz crystal) 001x = HS – HS Crystal Oscillator mode (10 MHz-25 MHz crystal) 000x = XTL – XTL Crystal Oscillator mode (200 kHz-4 MHz crystal)

TABLE 5-6: FOSC CONFIGURATION BITS DESCRIPTION FOR dsPIC30F2011/2012, dsPIC30F3010/3011/3012/3013/3014, dsPIC30F4013, dsPIC30F5015/5016, dsPIC30F6010A/6011A/6012A/6013A/6014A AND dsPIC30F6015 (CONTINUED)

Bit Field	Register	Description
FPR<4:0>	FOSC	**Alternate Oscillator Mode (when FOS<2:0> = 011b)** 1xxxx = Reserved (do not use) 0111x = Reserved (do not use) 01101 = Reserved (do not use) 01100 = ECIO – External clock. OSC2 pin is I/O 01011 = EC – External clock. OSC2 pin is system clock output (FOSC/4) 01010 = Reserved (do not use) 01001 = ERC – External RC oscillator. OSC2 pin is system clock output (FOSC/4) 01000 = ERCIO – External RC oscillator. OSC2 pin is I/O 00111 = Reserved (do not use) 00110 = Reserved (do not use) 00101 = Reserved (do not use) 00100 = XT – XT crystal oscillator (4 MHz-10 MHz crystal) 00010 = HS – HS crystal oscillator (10 MHz-25 MHz crystal) 00001 = Reserved (do not use) 00000 = XTL – XTL crystal oscillator (200 kHz-4 MHz crystal)

TABLE 5-10: dsPIC30F CONFIGURATION REGISTERS (FOR dsPIC30F2011/2012, dsPIC30F3010/3011/3012/3013/3014, dsPIC30F4013 AND dsPIC30F5015/5016)

Address	Name	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0xF80000	FOSC	FCKSM<1:0>		—	—	—	FOS<2:0>			—	—	—	FPR<4:0>				
0xF80002	FWDT	FWDTEN	—	—	—	—	—	—	—	—	—	FWPSA<1:0>		FWPSB<3:0>			
0xF80004	FBORPOR	MCLREN	—	—	—	—	PWMPIN(1)	HPOL(1)	LPOL(1)	BOREN	—	BORV<1:0>		—	—	FPWRT<1:0>	
0xF80006	FBS	—	—	Reserved(2)		—	—	—	Reserved(2)	—	—	—	—	Reserved(2)			
0xF80008	FSS	—	—	Reserved(2)		—	—	Reserved(2)		—	—	—	—	Reserved(2)			
0xF8000A	FGS	—	—	—	—	—	—	—	—	—	—	—	—	—	Reserved(3)	GCP	GWRP
0xF8000C	FICD	BKBUG	COE	—	—	—	—	—	—	—	—	—	—	—	—	ICS<1:0>	

Note 1: On the 2011, 2012, 3012, 3013, 3014 and 4013, these bits are reserved (read as '1' and must be programmed as '1').
2: Reserved bits read as '1' and must be programmed as '1'.
3: The FGS<2> bit is a read-only copy of the GCP bit (FGS<1>).

TABLE 5-11: dsPIC30F CONFIGURATION REGISTERS (FOR dsPIC30F6010A/6011A/6012A/6013A/6014A AND dsPIC30F6015)

Address	Name	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0xF80000	FOSC	FCKSM<1:0>		—	—	—	FOS<2:0>			—	—	—	FPR<4:0>				
0xF80002	FWDT	FWDTEN	—	—	—	—	—	—	—	—	—	FWPSA<1:0>		FWPSB<3:0>			
0xF80004	FBORPOR	MCLREN	—	—	—	—	PWMPIN(1)	HPOL(1)	LPOL(1)	BOREN	—	BORV<1:0>		—	—	FPWRT<1:0>	
0xF80006	FBS	—	—	RBS<1:0>		—	—	—	EBS	—	—	—	—	BSS<2:0>			BWRP
0xF80008	FSS	—	—	RSS<1:0>		—	—	ESS<1:0>		—	—	—	—	SSS<2:0>			SWRP
0xF8000A	FGS	—	—	—	—	—	—	—	—	—	—	—	—	—	GSS<1:0>		GWRP
0xF8000C	FICD	BKBUG	COE	—	—	—	—	—	—	—	—	—	—	—	—	ICS<1:0>	

Note 1: On the 6011A, 6012A, 6013A and 6014A, these bits are reserved (read as '1' and must be programmed as '1').

8.0 PROGRAMMING EXECUTIVE COMMANDS

8.1 Command Set

The programming executive command set is shown in Table 8-1. This table contains the opcode, mnemonic, length, time out and description for each command. Functional details on each command are provided in the command descriptions (see **Section 8.5 "Command Descriptions"**).

8.2 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 8-1). 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 8-1: 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 8-1 will return a "NACK" response (see **Section 9.2.1 "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.

8.3 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 8-2. This format minimizes traffic over the SPI and provides the programming executive with data that is properly aligned for performing table write operations.

FIGURE 8-2: PACKED INSTRUCTION WORD FORMAT

15 8	7 0
lsw1	
MSB2	MSB1
lsw2	

lswx: Least significant 16 bits of instruction word
MSBx: Most Significant Byte of instruction word

Note:	When the number of instruction words transferred is odd, MSB2 is zero and lsw2 cannot be transmitted.

8.4 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 9.2.3 "QE_Code Field"**.

8.5 Command Descriptions

All commands that are supported by the programming executive are described in **Section 8.5.1 "SCHECK Command"** through **Section 8.5.11 "QVER Command"**.

8.5.1 SCHECK COMMAND

15 12	11 0
Opcode	Length

Field	Description
Opcode	0x0
Length	0x1

The SCHECK command instructs the programming executive to do nothing, but generate a response. This command is used as a "sanity check" to verify that the programming executive is operational.

Expected Response (2 words):
0x1000
0x0002

Note:	This instruction is not required for programming, but is provided for development purposes only.

8.5.2 READD COMMAND

15 12	11 8	7 0
Opcode	Length	
Reserved0	N	
Reserved1		Addr_MSB
Addr_LS		

Field	Description
Opcode	0x1
Length	0x4
Reserved0	0x0
N	Number of 16-bit words to read (max of 2048)
Reserved1	0x0
Addr_MSB	MSB of 24-bit source address
Addr_LS	LS 16 bits of 24-bit source address

The READD command instructs the programming executive to read N 16-bit words of memory starting from the 24-bit address specified by Addr_MSB and Addr_LS. This command can only be used to read 16-bit data. It can be used to read data EEPROM, Configuration registers and the device ID.

Expected Response (2+N words):
0x1100
N + 2
Data word 1
...
Data word N

Note:	Reading unimplemented memory will cause the programming executive to reset.

8.5.3 READP COMMAND

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

Field	Description
Opcode	0x2
Length	0x4
N	Number of 24-bit instructions to read (max of 32768)
Reserved	0x0
Addr_MSB	MSB of 24-bit source address
Addr_LS	LS 16 bits of 24-bit source address

The READP command instructs the programming executive to read N 24-bit words of code memory 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 8.3 "Packed Data Format"**.

Expected Response (2 + 3 * N/2 words for N even):
0x1200
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):
0x1200
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.

8.5.4 PROGD COMMAND

15 12	11 8	7 0
Opcode	Length	
Reserved	Addr_MSB	
Addr_LS		
D_1		
D_2		
...		
D_16		

Field	Description
Opcode	0x4
Length	0x13
Reserved	0x0
Addr_MSB	MSB of 24-bit destination address
Addr_LS	LS 16 bits of 24-bit destination address
D_1	16-bit data word 1
D_2	16-bit data word 2
...	16-bit data words 3 through 15
D_16	16-bit data word 16

The PROGD command instructs the programming executive to program one row of data EEPROM. The data to be programmed is specified by the 16 data words (D_1, D_2,..., D_16) and is programmed to the destination address specified by Addr_MSB and Addr_LSB. The destination address should be a multiple of 0x20.

Once the row of data EEPROM has been programmed, the programming executive verifies the programmed data against the data in the command.

Expected Response (2 words):
0x1400
0x0002

Note:	Refer to Table 5-3 for data EEPROM size information.

8.5.9 ERASEP COMMAND

15 12	11 8	7 0
Opcode	Length	
Num_Rows		Addr_MSB
Addr_LS		

Field	Description
Opcode	0x9
Length	0x3
Num_Rows	Number of rows to erase
Addr_MSB	MSB of 24-bit base address
Addr_LS	LS 16 bits of 24-bit base address

The ERASEP command erases the specified number of rows of code memory from the specified base address. The specified base address must be a multiple of 0x40.

Once the erase is performed, all targeted words of code memory contain 0xFFFFFF.

Expected Response (2 words):
0x1900
0x0002

Note: The ERASEP command cannot be used to erase the Configuration registers or device ID. Code-protect Configuration registers can only be erased with the ERASEB command, while the device ID is read-only.

8.5.10 QBLANK COMMAND

15 12	11 0
Opcode	Length
PSize	
Reserved	DSize

Field	Description
Opcode	0xA
Length	0x3
PSize	Length of program memory to check (in 24-bit words), max of 49152
Reserved	0x0
DSize	Length of data memory to check (in 16-bit words), max of 2048

The QBLANK command queries the programming executive to determine if the contents of code memory and data EEPROM are blank (contains all '1's). The size of code memory and data EEPROM to check must be specified in the command.

The Blank Check for code memory begins at 0x0 and advances toward larger addresses for the specified number of instruction words. The Blank Check for data EEPROM begins at 0x7FFFFE and advances toward smaller addresses for the specified number of data words.

QBLANK returns a QE_Code of 0xF0 if the specified code memory and data EEPROM are blank. Otherwise, QBLANK returns a QE_Code of 0x0F.

Expected Response (2 words for blank device):
0x1AF0
0x0002

Expected Response (2 words for non-blank device):
0x1A0F
0x0002

Note: The QBLANK command does not check the system Configuration registers. The READD command must be used to determine the state of the Configuration registers.

11.0 ICSP™ MODE

11.1 ICSP Mode

ICSP mode is a special programming protocol that allows you to read and write to the dsPIC30F programming executive. The ICSP mode is the second (and slower) method used to program the device. This mode also has the ability to read the contents of executive memory to determine whether the programming executive is present. This capability is accomplished by applying control codes and instructions serially to the device using pins PGC and PGD.

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

Data is transmitted on the rising edge and latched on the falling edge of PGC. For all data transmissions, the Least Significant bit (LSb) is transmitted first.

Note 1: During ICSP operation, the operating frequency of PGC must not exceed 5 MHz.

2: Because ICSP is slower, it is recommended that only Enhanced ICSP (E-ICSP) mode be used for device programming, as described in **Section 5.1 "Overview of the Programming Process"**.

11.2 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 PGC and PGD, and this control code is used to command the CPU (see Table 11-1).

The SIX control code is used to send instructions to the CPU for execution, while the REGOUT control code is used to read data out of the device via the VISI register. The operation details of ICSP mode are provided in **Section 11.2.1 "SIX Serial Instruction Execution"** and **Section 11.2.2 "REGOUT Serial Instruction Execution"**.

TABLE 11-1: 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 register.
0010b-1111b	N/A	Reserved.

11.2.1 SIX SERIAL INSTRUCTION EXECUTION

The SIX control code allows execution of dsPIC30F assembly instructions. When the SIX code is received, the CPU is suspended for 24 clock cycles as the instruction is then clocked into the internal buffer. Once the instruction is shifted in, the state machine allows it to be executed over the next four clock cycles. While the received instruction is executed, the state machine simultaneously shifts in the next 4-bit command (see Figure 11-2).

Note 1: Coming out of the ICSP entry sequence, the first 4-bit control code is always forced to SIX and a forced NOP instruction is executed by the CPU. Five additional PGC clocks are needed on start-up, thereby resulting in a 9-bit SIX command instead of the normal 4-bit SIX command. After the forced SIX is clocked in, ICSP operation resumes as normal (the next 24 clock cycles load the first instruction word to the CPU). See Figure 11-1 for details.

2: TBLRDH, TBLRDL, TBLWTH and TBLWTL instructions must be followed by a NOP instruction.

11.2.2 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 PGD pin. Once the REGOUT control code is received, eight clock cycles are required to process the command. During this time, the CPU is held idle. After these eight cycles, an additional 16 cycles are required to clock the data out (see Figure 11-3).

The REGOUT instruction is unique because the PGD pin is an input when the control code is transmitted to the device. However, once the control code is processed, the PGD pin becomes an output as the VISI register is shifted out. After the contents of the VISI are shifted out, PGD becomes an input again as the state machine holds the CPU idle until the next 4-bit control code is shifted in.

Note:	Once the contents of VISI are shifted out, the dsPIC® DSC device maintains PGD as an output until the first rising edge of the next clock is received.

FIGURE 11-1: PROGRAM ENTRY AFTER RESET

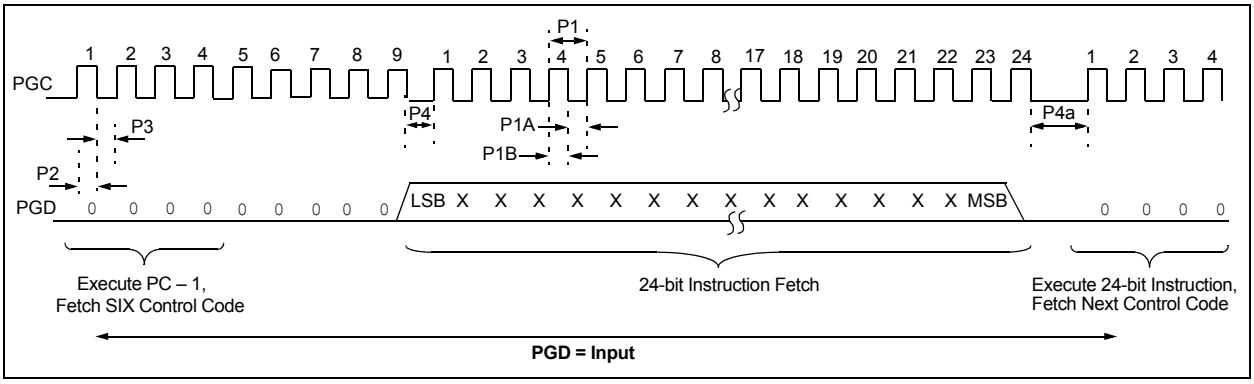


FIGURE 11-2: SIX SERIAL EXECUTION

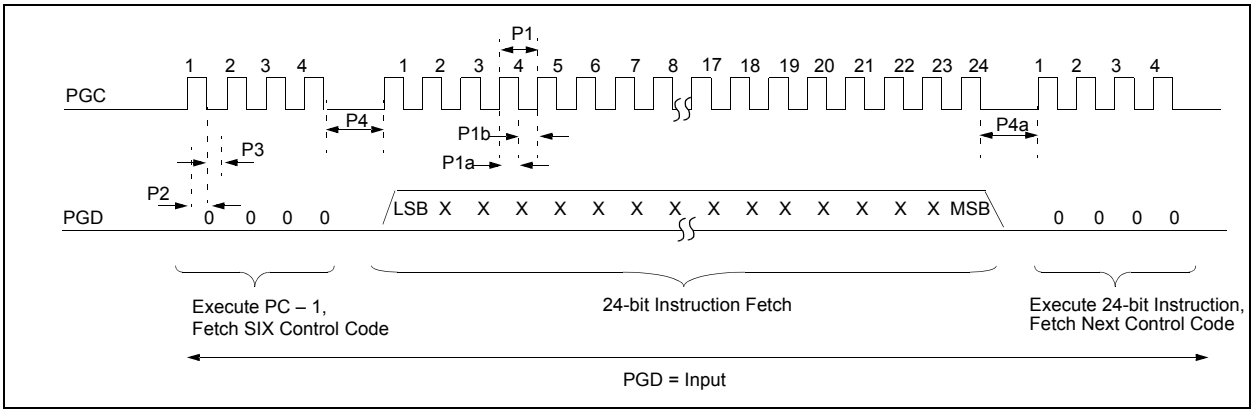
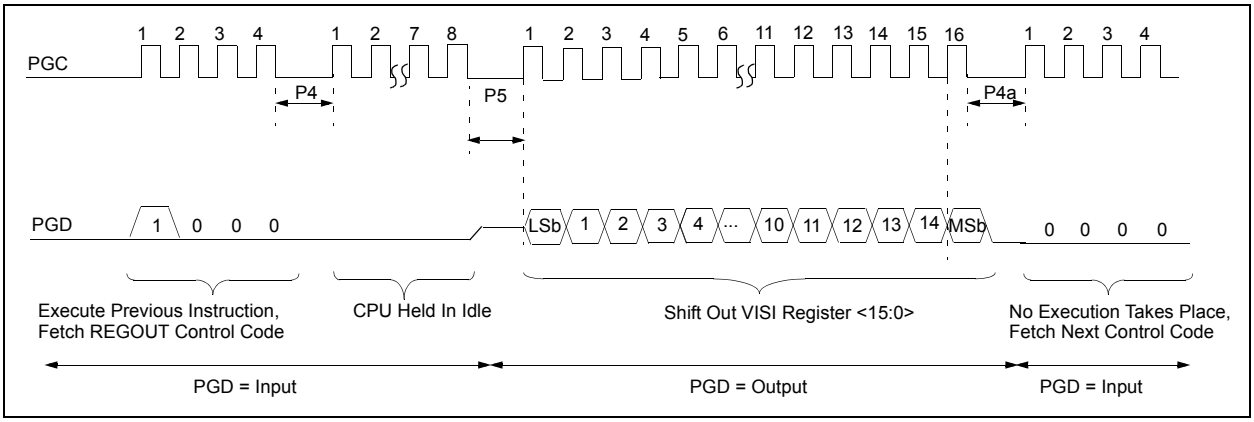


FIGURE 11-3: REGOUT SERIAL EXECUTION



11.4 Flash Memory Programming in ICSP Mode

Programming in ICSP mode is described in **Section 11.4.1 "Programming Operations"** through **Section 11.4.3 "Starting and Stopping a Programming Cycle"**. Step-by-step procedures are described in **Section 11.5 "Erasing Program Memory in Normal-Voltage Systems"** through **Section 11.13 "Reading the Application ID Word"**. All programming operations must use serial execution, as described in **Section 11.2 "ICSP Operation"**.

11.4.1 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 11-2) or write operation (Table 11-3), writing a key sequence to enable the programming and initiating the programming by setting the WR control bit, NVMCON<15>.

In ICSP mode, all programming operations are externally timed. An external 2 ms delay must be used between setting the WR control bit and clearing the WR control bit to complete the programming operation.

TABLE 11-2: NVMCON ERASE OPERATIONS

NVMCON Value	Erase Operation
0x407F	Erase all code memory, data memory (does not erase UNIT ID).
0x4075	Erase 1 row (16 words) of data EEPROM.
0x4074	Erase 1 word of data EEPROM.
0x4072	Erase all executive memory.
0x4071	Erase 1 row (32 instruction words) from 1 panel of code memory.
0x406E	Erase Boot Secure and General Segments, then erase FBS, FSS and FGS configuration registers.
0x4066	Erase all Data EEPROM allocated to Boot Segment.
0x405E	Erase Secure and General Segments, then erase FSS and FGS configuration registers.
0x4056	Erase all Data EEPROM allocated to Secure Segment.
0x404E	Erase General Segment, then erase FGS configuration register.
0x4046	Erase all Data EEPROM allocated to General Segment.

TABLE 11-3: NVMCON WRITE OPERATIONS

NVMCON Value	Write Operation
0x4008	Write 1 word to configuration memory.
0x4005	Write 1 row (16 words) to data memory.
0x4004	Write 1 word to data memory.
0x4001	Write 1 row (32 instruction words) into 1 panel of program memory.

11.4.2 UNLOCKING NVMCON FOR PROGRAMMING

Writes to the WR bit (NVMCON<15>) are locked to prevent accidental programming from taking place. Writing a key sequence to the NVMKEY register unlocks the WR bit and allows it to be written to. The unlock sequence is performed as follows:

```
MOV     #0x55, W8
MOV     W8, NVMKEY
MOV     #0xAA, W9
MOV     W9, NVMKEY
```

Note: Any working register, or working register pair, can be used to write the unlock sequence.

11.4.3 STARTING AND STOPPING A PROGRAMMING CYCLE

Once the unlock key sequence has been written to the NVMKEY register, the WR bit (NVMCON<15>) is used to start and stop an erase or write cycle. Setting the WR bit initiates the programming cycle. Clearing the WR bit terminates the programming cycle.

All erase and write cycles must be externally timed. An external delay must be used between setting and clearing the WR bit. Starting and stopping a programming cycle is performed as follows:

```
BSET    NVMCON, #WR
<Wait 2 ms>
BCLR    NVMCON, #WR
```

11.5 Erasing Program Memory in Normal-Voltage Systems

The procedure for erasing program memory (all code memory, data memory, executive memory and code-protect bits) consists of setting NVMCON to 0x407F, unlocking NVMCON for erasing and then executing the programming cycle. This method of bulk erasing program memory only works for systems where VDD is between 4.5 volts and 5.5 volts. The method for erasing program memory for systems with a lower VDD (3.0 volts-4.5 volts) is described in **Section 6.1 "Erasing Memory"**.

TABLE 11-4: SERIAL INSTRUCTION EXECUTION FOR BULK ERASING PROGRAM MEMORY (ONLY IN NORMAL-VOLTAGE SYSTEMS) (CONTINUED)

Command (Binary)	Data (Hexadecimal)	Description
0000	200558	MOV #0x55, W8
0000	883B38	MOV W8, NVMKEY
0000	200AA9	MOV #0xAA, W9
0000	883B39	MOV W9, NVMKEY
Step 11: Initiate the erase cycle.		
0000	A8E761	BSET NVMCON, #WR
0000	000000	NOP
0000	000000	NOP
—	—	Externally time 'P13a' ms (see **Section 13.0 "AC/DC Characteristics and Timing Requirements"**)
0000	000000	NOP
0000	000000	NOP
0000	A9E761	BCLR NVMCON, #WR
0000	000000	NOP
0000	000000	NOP

Note 1: Steps 2-8 are only required for the dsPIC30F5011/5013 devices. These steps may be skipped for all other devices in the dsPIC30F family.

11.6 Erasing Program Memory in Low-Voltage Systems

The procedure for erasing program memory (all code memory and data memory) in low-voltage systems (with VDD between 2.5 volts and 4.5 volts) is quite different than the procedure for erasing program memory in normal-voltage systems. Instead of using a Bulk Erase operation, each region of memory must be individually erased by row. Namely, all of the code memory, executive memory and data memory must be erased one row at a time. This procedure is detailed in Table 11-5.

Due to security restrictions, the FBS, FSS and FGS register cannot be erased in low-voltage systems. Once any bits in the FGS register are programmed to '0', they can only be set back to '1' by performing a Bulk Erase in a normal-voltage system. Alternatively, a Segment Erase operation can be performed instead of a Bulk Erase.

Normal-voltage systems can also be used to erase program memory as shown in Table 11-5. However, since this method is more time-consuming and does not clear the code-protect bits, it is not recommended.

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

TABLE 11-5: SERIAL INSTRUCTION EXECUTION FOR ERASING PROGRAM MEMORY (EITHER IN LOW-VOLTAGE OR NORMAL-VOLTAGE SYSTEMS)

Command (Binary)	Data (Hexadecimal)	Description
Step 1: Exit the Reset vector.		
0000	040100	GOTO 0x100
0000	040100	GOTO 0x100
0000	000000	NOP
Step 2: Initialize NVMADR and NVMADRU to erase code memory and initialize W7 for row address updates.		
0000	EB0300	CLR W6
0000	883B16	MOV W6, NVMADR
0000	883B26	MOV W6, NVMADRU
0000	200407	MOV #0x40, W7
Step 3: Set NVMCON to erase 1 row of code memory.		
0000	24071A	MOV #0x4071, W10
0000	883B0A	MOV W10, NVMCON
Step 4: Unlock the NVMCON to erase 1 row of code memory.		
0000	200558	MOV #0x55, W8
0000	883B38	MOV W8, NVMKEY
0000	200AA9	MOV #0xAA, W9
0000	883B39	MOV W9, NVMKEY
Step 5: Initiate the erase cycle.		
0000	A8E761	BSET NVMCON, #WR
0000	000000	NOP
0000	000000	NOP
–	–	Externally time 'P13a' ms (see **Section 13.0 "AC/DC Characteristics and Timing Requirements"**)
0000	000000	NOP
0000	000000	NOP
0000	A9E761	BCLR NVMCON, #WR
0000	000000	NOP
0000	000000	NOP

TABLE 11-5: SERIAL INSTRUCTION EXECUTION FOR ERASING PROGRAM MEMORY (EITHER IN LOW-VOLTAGE OR NORMAL-VOLTAGE SYSTEMS) (CONTINUED)

Command (Binary)	Data (Hexadecimal)	Description
Step 6: Update the row address stored in NVMADRU:NVMADR. When W6 rolls over to 0x0, NVMADRU must be incremented.		
0000	430307	ADD W6, W7, W6
0000	AF0042	BTSC SR, #C
0000	EC2764	INC NVMADRU
0000	883B16	MOV W6, NVMADR
Step 7: Reset device internal PC.		
0000	040100	GOTO 0x100
0000	000000	NOP
Step 8: Repeat Steps 3-7 until all rows of code memory are erased.		
Step 9: Initialize NVMADR and NVMADRU to erase executive memory and initialize W7 for row address updates.		
0000	EB0300	CLR W6
0000	883B16	MOV W6, NVMADR
0000	200807	MOV #0x80, W7
0000	883B27	MOV W7, NVMADRU
0000	200407	MOV #0x40, W7
Step 10: Set NVMCON to erase 1 row of executive memory.		
0000	24071A	MOV #0x4071, W10
0000	883B0A	MOV W10, NVMCON
Step 11: Unlock the NVMCON to erase 1 row of executive memory.		
0000	200558	MOV #0x55, W8
0000	883B38	MOV W8, NVMKEY
0000	200AA9	MOV #0xAA, W9
0000	883B39	MOV W9, NVMKEY
Step 12: Initiate the erase cycle.		
0000	A8E761	BSET NVMCON, #WR
0000	000000	NOP
0000	000000	NOP
—	—	Externally time 'P13a' ms (see **Section 13.0 "AC/DC Characteristics and Timing Requirements"**)
0000	000000	NOP
0000	000000	NOP
0000	A9E761	BCLR NVMCON, #WR
0000	000000	NOP
0000	000000	NOP
Step 13: Update the row address stored in NVMADR.		
0000	430307	ADD W6, W7, W6
0000	883B16	MOV W6, NVMADR
Step 14: Reset device internal PC.		
0000	040100	GOTO 0x100
0000	000000	NOP
Step 15: Repeat Steps 10-14 until all 24 rows of executive memory are erased.		
Step 16: Initialize NVMADR and NVMADRU to erase data memory and initialize W7 for row address updates.		
0000	2XXXX6	MOV #<lower 16-bits of starting Data EEPROM address>, W6
0000	883B16	MOV W6, NVMADR
0000	2007F6	MOV #0x7F, W6
0000	883B16	MOV W6, NVMADRU
0000	200207	MOV #0x20, W7
Step 17: Set NVMCON to erase 1 row of data memory.		
0000	24075A	MOV #0x4075, W10
0000	883B0A	MOV W10, NVMCON

11.10 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. To ensure efficient execution and facilitate verification on the programmer, four instruction words are read from the device at a time.

Table 11-10 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 and W6 registers. The upper byte of the starting source address is stored to TBLPAG, while the lower 16 bits of the source address are stored to W6.

To minimize the reading time, the packed instruction word format that was utilized for writing is also used for reading (see Figure 11-5). In Step 3, the write pointer W7 is initialized, and four instruction words are read from code memory and stored to working registers W0:W5. In Step 4, the four instruction words are clocked out of the device from the VISI register using the REGOUT command. In Step 5, the internal PC is reset to 0x100, as a precautionary measure, to prevent the PC from incrementing into unimplemented memory when large devices are being read. Lastly, in Step 6, Steps 3-5 are repeated until the desired amount of code memory is read.

TABLE 11-10: SERIAL INSTRUCTION EXECUTION FOR READING CODE MEMORY

Command (Binary)	Data (Hexadecimal)	Description
Step 1: Exit the Reset vector.		
0000	040100	GOTO 0x100
0000	040100	GOTO 0x100
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) and store the next four locations of code memory to W0:W5.		
0000	EB0380	CLR W7
0000	000000	NOP
0000	BA1B96	TBLRDL [W6], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	BADBB6	TBLRDH.B [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	BADBD6	TBLRDH.B [++W6], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	BA1BB6	TBLRDL [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	BA1B96	TBLRDL [W6], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	BADBB6	TBLRDH.B [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	BADBD6	TBLRDH.B [++W6], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	BA0BB6	TBLRDL [W6++], [W7]
0000	000000	NOP
0000	000000	NOP

TABLE 11-10: SERIAL INSTRUCTION EXECUTION FOR READING CODE MEMORY (CONTINUED)

Command (Binary)	Data (Hexadecimal)	Description
Step 4: Output W0:W5 using the VISI register and REGOUT command.		
0000	883C20	MOV W0, VISI
0000	000000	NOP
0001	<VISI>	Clock out contents of VISI register
0000	000000	NOP
0000	883C21	MOV W1, VISI
0000	000000	NOP
0001	<VISI>	Clock out contents of VISI register
0000	000000	NOP
0000	883C22	MOV W2, VISI
0000	000000	NOP
0001	<VISI>	Clock out contents of VISI register
0000	000000	NOP
0000	883C23	MOV W3, VISI
0000	000000	NOP
0001	<VISI>	Clock out contents of VISI register
0000	000000	NOP
0000	883C24	MOV W4, VISI
0000	000000	NOP
0001	<VISI>	Clock out contents of VISI register
0000	000000	NOP
0000	883C25	MOV W5, VISI
0000	000000	NOP
0001	<VISI>	Clock out contents of VISI register
0000	000000	NOP
Step 5: Reset the device internal PC.		
0000	040100	GOTO 0x100
0000	000000	NOP
Step 6: Repeat steps 3-5 until all desired code memory is read.		

11.11 Reading Configuration Memory

The procedure for reading configuration memory is similar to the procedure for reading code memory, except that 16-bit data words are read instead of 24-bit words. Since there are seven Configuration registers, they are read one register at a time.

Table 11-11 shows the ICSP programming details for reading all of the configuration memory. Note that the TBLPAG register is hard-coded to 0xF8 (the upper byte address of configuration memory), and the read pointer W6 is initialized to 0x0000.

TABLE 11-11: SERIAL INSTRUCTION EXECUTION FOR READING ALL CONFIGURATION MEMORY

Command (Binary)	Data (Hexadecimal)	Description
Step 1: Exit the Reset vector.		
0000	040100	GOTO 0x100
0000	040100	GOTO 0x100
0000	000000	NOP
Step 2: Initialize TBLPAG, and the read pointer (W6) and the write pointer (W7) for TBLRD instruction.		
0000	200F80	MOV #0xF8, W0
0000	880190	MOV W0, TBLPAG
0000	EB0300	CLR W6
0000	EB0380	CLR W7
0000	000000	NOP
Step 3: Read the Configuration register and write it to the VISI register (located at 0x784).		
0000	BA0BB6	TBLRDL [W6++], [W7]
0000	000000	NOP
0000	000000	NOP
0000	883C20	MOV W0, VISI
0000	000000	NOP
Step 4: Output the VISI register using the REGOUT command.		
0001	<VISI>	Clock out contents of VISI register
0000	000000	NOP
Step 5: Reset device internal PC.		
0000	040100	GOTO 0x100
0000	000000	NOP
Step 6: Repeat steps 3-5 six times to read all of configuration memory.		

TABLE 12-1: PROGRAMMING THE PROGRAMMING EXECUTIVE (CONTINUED)

Command (Binary)	Data (Hexadecimal)	Description
Step 8: 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 9: Repeat Steps 7-8 eight times to load the write latches for the 32 instructions.		
Step 10: Unlock the NVMCON for programming.		
0000	200558	MOV #0x55, W8
0000	883B38	MOV W8, NVMKEY
0000	200AA9	MOV #0xAA, W9
0000	883B39	MOV W9, NVMKEY
Step 11: Initiate the programming cycle.		
0000	A8E761	BSET NVMCON, #15
0000	000000	NOP
0000	000000	NOP
—	—	Externally time 'P12a' ms (see **Section 13.0 "AC/DC Characteristics and Timing Requirements"**)
0000	000000	NOP
0000	000000	NOP
0000	A9E761	BCLR NVMCON, #15
0000	000000	NOP
0000	000000	NOP
Step 12: Reset the device internal PC.		
0000	040100	GOTO 0x100
0000	000000	NOP
Step 13: Repeat Steps 7-12 until all 23 rows of executive memory are programmed.		

TABLE 12-2: READING EXECUTIVE MEMORY (CONTINUED)

Command (Binary)	Data (Hexadecimal)	Description
Step 4: Output W0:W5 using the VISI register and REGOUT command.		
0000	883C20	MOV W0, VISI
0000	000000	NOP
0001	—	Clock out contents of VISI register
0000	883C21	MOV W1, VISI
0000	000000	NOP
0001	—	Clock out contents of VISI register
0000	883C22	MOV W2, VISI
0000	000000	NOP
0001	—	Clock out contents of VISI register
0000	883C23	MOV W3, VISI
0000	000000	NOP
0001	—	Clock out contents of VISI register
0000	883C24	MOV W4, VISI
0000	000000	NOP
0001	—	Clock out contents of VISI register
0000	883C25	MOV W5, VISI
0000	000000	NOP
0001	—	Clock out contents of VISI register
Step 5: Reset the device internal PC.		
0000	040100	GOTO 0x100
0000	000000	NOP
Step 6: Repeat Steps 3-5 until all 736 instruction words of executive memory are read.		

TABLE 13-1: AC/DC CHARACTERISTICS (CONTINUED)

AC/DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise stated) Operating Temperature: 25° C is recommended			
Param. No.	**Sym**	**Characteristic**	**Min**	**Max**	**Units**	**Conditions**
P9b	TDLY5	Delay between PGD ↓by programming executive to PGD released by programming executive	15	—	µs	—
P10	TDLY6	Delay between PGD released by programming executive to first PGC ↑ of response	5	—	µs	—
P11	TDLY7	Delay between clocking out response words	10	—	µs	—
P12a	TPROG	Row Programming cycle time	1	4	ms	ICSP mode
P12b	TPROG	Row Programming cycle time	0.8	2.6	ms	Enhanced ICSP mode
P13a	TERA	Bulk/Row Erase cycle time	1	4	ms	ICSP mode
P13b	TERA	Bulk/Row Erase cycle time	0.8	2.6	ms	Enhanced ICSP mode

APPENDIX A: DEVICE-SPECIFIC INFORMATION

A.1 Checksum Computation

The checksum computation is described in **Section 6.8 "Checksum Computation"**. Table A-1 shows how this 16-bit computation can be made for each dsPIC30F device. Computations for read code protection are shown both enabled and disabled. The checksum values assume that the Configuration registers are also erased. However, when code protection is enabled, the value of the FGS register is assumed to be 0x5.

A.2 dsPIC30F5011 and dsPIC30F5013

A.2.1 ICSP PROGRAMMING

The dsPIC30F5011 and dsPIC30F5013 processors require that the FBS and FSS registers be programmed with 0x0000 before the device is chip erased. The steps to perform this action are shown in Table 11-4.

A.2.2 ENHANCED ICSP PROGRAMMING

The dsPIC30F5011 and dsPIC30F5013 processors require that the FBS and FSS registers be programmed with 0x0000 using the `PROGC` command before the `ERASEB` command is used to erase the chip.

TABLE A-1: CHECKSUM COMPUTATION

Device	Read Code Protection	Checksum Computation	Erased Value	Value with 0xAAAAAA at 0x0 and Last Code Address
dsPIC30F2010	Disabled	CFGB+SUM(0:001FFF)	0xD406	0xD208
	Enabled	CFGB	0x0404	0x0404
dsPIC30F2011	Disabled	CFGB+SUM(0:001FFF)	0xD406	0xD208
	Enabled	CFGB	0x0404	0x0404
dsPIC30F2012	Disabled	CFGB+SUM(0:001FFF)	0xD406	0xD208
	Enabled	CFGB	0x0404	0x0404
dsPIC30F3010	Disabled	CFGB+SUM(0:003FFF)	0xA406	0xA208
	Enabled	CFGB	0x0404	0x0404
dsPIC30F3011	Disabled	CFGB+SUM(0:003FFF)	0xA406	0xA208
	Enabled	CFGB	0x0404	0x0404
dsPIC30F3012	Disabled	CFGB+SUM(0:003FFF)	0xA406	0xA208
	Enabled	CFGB	0x0404	0x0404
dsPIC30F3013	Disabled	CFGB+SUM(0:003FFF)	0xA406	0xA208
	Enabled	CFGB	0x0404	0x0404
dsPIC30F3014	Disabled	CFGB+SUM(0:003FFF)	0xA406	0xA208
	Enabled	CFGB	0x0404	0x0404
dsPIC30F4011	Disabled	CFGB+SUM(0:007FFF)	0x4406	0x4208
	Enabled	CFGB	0x0404	0x0404
dsPIC30F4012	Disabled	CFGB+SUM(0:007FFF)	0x4406	0x4208
	Enabled	CFGB	0x0404	0x0404
dsPIC30F4013	Disabled	CFGB+SUM(0:007FFF)	0x4406	0x4208
	Enabled	CFGB	0x0404	0x0404
dsPIC30F5011	Disabled	CFGB+SUM(0:00AFFF)	0xFC06	0xFA08
	Enabled	CFGB	0x0404	0x0404
dsPIC30F5013	Disabled	CFGB+SUM(0:00AFFF)	0xFC06	0xFA08
	Enabled	CFGB	0x0404	0x0404
dsPIC30F5015	Disabled	CFGB+SUM(0:00AFFF)	0xFC06	0xFA08
	Enabled	CFGB	0x0404	0x0404

Item Description:

SUM(a:b) = Byte sum of locations a to b inclusive (all 3 bytes of code memory)

CFGB = **Configuration Block (masked)** = Byte sum of ((FOSC&0xC10F) + (FWDT&0x803F) + (FBORPOR&0x87B3) + (FBS&0x310F) + (FSS&0x330F) + (FGS&0x0007) + (FICD&0xC003))