Section 23. CodeGuard™ Security

HIGHLIGHTS

This section of the manual contains the following major topics:

23.1 CODE PROTECTION OVERVIEW

The on-chip program Flash memory in a dsPIC33E/PIC24E device can be organized into two code space segments. Each of these segments has an implied security privilege level and system function.

- General Segment (GS) – This segment is designed for the end-user's system code.
- Auxiliary Segment (AS) – This segment is designed for the end-user's system code or EEPROM emulation.

Any operation of the system that potentially allows exposure of the code or data contents is restricted, based on the segment from which the operation originated, or the segment to which the operation targets.

Restricted operations include:

- Programming, Erase or Verify Operations
- Reads or Writes of Code Space
- Reads or Writes of Protected Data Space
- Code Flow Change into a Secure Segment from outside the segment
- Interrupt Vectors into a Secure Segment

Configuration bits are provided to enable access to the Secure Segments and their parameters. These bits allow configuration of both the sizes and restrictions of the program Flash memory, and RAM segments.

23.2 CONTROL REGISTERS

Several Configuration and Special Function Registers (SFRs) control the security functions. The key registers for supporting the code security features are:

- **FGS: General Segment Configuration Register**
- **FAS: Auxiliary Segment Configuration Register**

Register 23-1: FGS: General Segment Configuration Register

U-0	U-0	R/P	R/P	U-0	U-0	R/P	R/P
—	—	GSSK<1:0>		—	—	GSS	GWRP
bit 7							bit 0

Legend:	r = Reserved	P = Programmable bit	
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-6 **Unimplemented:** Read as '0'

bit 5-4 **GSSK<1:0>:** General Segment Key bits
These bits must be set to '00' if GWRP = 1 and GSS = 1.
These bits must be set to '11' for any other value of the GWRP and GSS bits.
Any mismatch between either the GWRP or GSS bits, and the GSSK bits (as described above), will result in code protection getting enabled. A Flash bulk erase will be required to unlock the device.

bit 3-2 **Unimplemented:** Read as '0'

bit 1 **GSS:** General Segment Program Flash Code Protection bit
1 = General Segment not protected
0 = High security; general program Flash segment is protected

bit 0 **GWRP:** General Segment Program Flash Write Protection bit
1 = General segment can be written
0 = General segment is write-protected

Register 23-2: FAS: Auxiliary Segment Configuration Register

U-0	U-0	R/P	R/P	U-0	U-0	R/P	R/P
—	—	APLK<1:0>		—	—	APL	AWRP
bit 7							bit 0

Legend:	r = Reserved	P = Programmable bit	
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-6 **Unimplemented:** Read as '0'

bit 5-4 **APLK<1:0>:** Auxiliary Segment Key bits
These bits must be set to '00' if AWRP = 1 and APL = 1.
These bits must be set to '11' for any other value of the AWRP and APL bits.
Any mismatch between either the AWRP or APL bits, and the APLK bits (as described above), will result in code protection getting enabled. A Flash bulk erase will be required to unlock the device.

bit 3-2 **Unimplemented:** Read as '0'

bit 1 **APL:** Auxiliary Segment Program Flash Code Protection bit
1 = Auxiliary Segment not protected
0 = High security; general program Flash segment is protected

bit 0 **AWRP:** Auxiliary Segment Program Flash Write Protection bit
1 = Auxiliary segment can be written
0 = Auxiliary segment is write-protected

23.3 DEFINITION OF SECURITY PRIVILEGES

It is important to understand the relative privilege levels of the two code protection segments. Operations can be described as being relative to higher or lower privilege segments.The lower privilege segment can only access code from the higher segment by issuing calls.

Rules governing access privileges are discussed in sections **23.4 "Rules Concerning Program Flow"** through **23.6.1 "Rules for Programming Devices in RTSP"**. Table 23-1 presents a summary overview of these rules during normal run-time operation.

Table 23-1: Privileged Operations Rules Summary

Target Segment		**General Segment**				**Auxiliary Segment**				**IVT**			
Protection Level		None		High		None		High		None		High	
Write-Protected		No	Yes	No	Yes	No	Yes	No	Yes	No	Yes	No	Yes
Requested Operation (Yes/No)													
PC Rollover into Target Segment		Yes	Yes	Yes	Yes	N/A	N/A	N/A	N/A	N/A **(Note 3)**			
Program Flow Change (PFC) from General Segment reset vector instruction to Target Segment **(Note 1)**		Yes	Yes	Yes	Yes	Yes	Yes	**Note 2**	**Note 2**	**Note 4**			
Program Flow Change (PFC) from Auxiliary Segment reset vector instruction to Target Segment **(Note 1)**		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	**Note 4**			
Vector Flow Change (VFC) to Target Segment **(Note 5)**		Yes	Yes	Yes	Yes	Yes	Yes	**Note 2**	**Note 2**	**Note 4**			
PFC from AS to Target Segment **(Note 1)**		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	**Note 4**			
PFC from GS to Target Segment **(Note 1)**		Yes	Yes	Yes	Yes	Yes	Yes	**Note 2**	**Note 2**	**Note 4**			
R/W of Target Segment RAM while executing from: **Note:** Stack assumed to be in GS RAM space, access needed.	AS	Yes	Yes	Yes	Yes	N/A	N/A	N/A	N/A	N/A			
	GS	Yes	Yes	Yes	Yes	N/A	N/A	N/A	N/A				
Table Read/PSV of Target Segment Program Flash while executing from: **(Note 7)**	AS	Yes	Yes	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	GS	Yes	Yes	Yes	Yes	Yes	Yes	No	No	Yes	Yes	Yes	Yes
Table Write of Target Segment (load write latches)		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Note 1: Program Flow Change (PFC) is defined as when the PC is loaded with a new value instead of the normal automatic increment. It includes `JUMP`, `CALL`, `RETURN`, `RETFIE`, `Computed Jump`, etc.

2: PFC is allowed only to the last 32 instruction locations of the segment.

3: Since execution is not permitted in the VS segment, this condition is not possible.

4: A PFC operation (i.e., branch, call, etc.) into the IVT segment is possible. But as soon as execution is attempted out of this segment an illegal address trap will result (unless pointed to Reset vector at address 0x000000).

5: Vector Flow Change (VFC) is defined as when the PC is loaded with a Interrupt or trap vector address.

6: Operation allowed if there is no higher security privilege-segment defined.

7: TBLRD or DS read will execute, but returns all '0's if not allowed.

Table 23-1: Privileged Operations Rules Summary (Continued)

Target Segment		General Segment				Auxiliary Segment				IVT			
Protection Level		None		High		None		High		None		High	
Write-Protected		No	Yes	No	Yes	No	Yes	No	Yes	No	Yes	No	Yes
Requested Operation (Yes/No)													
Program/Erase row of Target Segment program Flash while executing from:	AS	Yes	No	No	No	Yes	No	Yes	No	Yes	No	No	No
	GS	Yes	No	Yes	No	Yes	No	No	No	Yes	No	No	No
Erase All		Command not valid in RTSP mode											
Erase GS Segment/code-protect		Erase GS segment and GS code protection fuses, Erase VS											
Erase AS Segment/code-protect		Erase AS segment and AS code protection fuses											
Erase AS Segment		Erase AS segment only											
Erase GS Segment		Erase GS Segment Only											
Program configuration register		Yes											

Note 1: Program Flow Change (PFC) is defined as when the PC is loaded with a new value instead of the normal automatic increment. It includes `JUMP`, `CALL`, `RETURN`, `RETFIE`, `Computed Jump`, etc.

2: PFC is allowed only to the last 32 instruction locations of the segment.

3: Since execution is not permitted in the VS segment, this condition is not possible.

4: A PFC operation (i.e., branch, call, etc.) into the IVT segment is possible. But as soon as execution is attempted out of this segment an illegal address trap will result (unless pointed to Reset vector at address 0x000000).

5: Vector Flow Change (VFC) is defined as when the PC is loaded with a Interrupt or trap vector address.

6: Operation allowed if there is no higher security privilege-segment defined.

7: TBLRD or DS read will execute, but returns all '0's if not allowed.

23.4 RULES CONCERNING PROGRAM FLOW

Program flow refers to the execution sequence of program instructions in program memory. Normally, instructions are executed sequentially as the Program Counter (PC) increments.

Program Flow Change (PFC) occurs when the PC is reloaded as a result of Call, Jump, Computed Jump, Return, Return from Subroutine, or other form of branch instruction. A PFC allows the program flow to follow an alternate path. A normal PFC only allows the program to branch within the same segment. A Restricted PFC allows the program to branch to a special Segment Access Area of a higher security segment.

Vector Flow Change (VFC) occurs when the PC is reloaded with an Interrupt or Trap vector.

Jumping into secure code at unintended locations can expose code to algorithm detection. Therefore, PFC and VFC operations are restricted if they violate the privilege hierarchy.

PFCs within a segment are unrestricted. Generally, PFC and VFC changes from one segment to another segment are not restricted, except as follows:

To ensure the integrity of the operations of code within the Auxiliary Segment, the user must restrict program flow options to this segment. Program flow can be limited to only allow the segment access areas to be a branch target. The segment access areas are the last 32 instruction locations of the Auxiliary Segment code space.

The owners of the code within the Auxiliary Segment code space can ensure that the access area contains branches to specified sections of the application code, verified to not expose the algorithm.

If a PFC or VFC targets a restricted location, that operation will cause a security reset. The device will Reset and set the IOPUWR (RCON<14>) status bit, indicating an illegal operation.

In addition to this specific security reset, there are also program flow checks that are built into all devices.

If a program flow or vector flow change targets unimplemented program memory space, an address error trap occurs.

Code execution from the vector segment, other than the instruction at the Reset location, is not allowed. If attempted, it results in an address error trap.

23.5 RULES CONCERNING INTERRUPTS

23.5.1 Interrupts and Traps in Secure Modes

Interrupt handling is restricted for the following reasons:

- A Return from Interrupt is one way to corrupt intended program flow (by changing the return address in the stack).
- The secure code should have the opportunity to clear sensitive information before responding to an interrupt.

23.5.1.1 AUXILIARY SEGMENT INTERRUPT VECTOR

Note:	Interrupts occurring when code executes in the Auxiliary Segment causes the processor to vector to the special interrupt vector for that segment. Users may employ a special ISR within the protected segment to hide critical data, and then manually vector to the real ISR by reading the INTTREG SFR.

If an interrupt occurs while the program is running in the Auxiliary Segment, the processor obtains the interrupt vector from the special interrupt vector location at 0x7FFFEA.

Note:	Refer to the specific device data sheet for the actual interrupt vector address location.

23.5.1.2 INTERRUPT AND TRAP HANDLING SEQUENCE WHILE EXECUTING FROM THE AUXILIARY SEGMENT

The sequence for handling interrupts and traps is as follows:

1. Interrupt or trap occurs while code is executing in a Auxiliary Segment.
2. Return address is pushed on the Stack.
3. The interrupt vector of 0x7FFFEA is loaded into the PC instead of the usual interrupt vector.
4. Special ISR is executed at address 0x7FFFEA.
5. A determination is made if execution of the GS interrupt is allowed. If execution is not allowed, the following two sub-steps are performed; otherwise, if execution is allowed, step 6 through step 14 are performed:
   a) The INTTREG SFR is read to determine which interrupt to clear.
   b) Clear the interrupt and return from the special interrupt back to the Auxiliary Segment.
6. Actual return address is retrieved from Stack and saved.
7. Actual return address is replaced with new return address. For example, 0x7FFFFA.
8. INTTREG SFR is read to determine which interrupt vector to jump to.
9. Interrupt vector is read from the vector table and executes an indirect jump.
10. Users ISR begins execution.
11. User code executes.
12. Return from interrupt (back to location 0x7FFFFA).
13. Read actual return address from saved location.
14. Execute indirect jump to go back to Auxiliary Segment.

23.6 SECURITY FEATURES AND DEVICE OPERATIONAL MODE

Security functions are dependant on the operational mode of the device. Each device can operate in one of following modes:

- In Run-Time Self-Programming (RTSP) mode (normal device operation), the application code is running and the application code can invoke self programming.
- In the In-Circuit Serial Programming™ (ICSP™) mode, the programming mode provides native, low-level programming capability to erase, program and verify the chip. The device is under the command of a device programmer such as PRO MATE® 3 or MPLAB® ICD 2.

23.6.1 Rules for Programming Devices in RTSP

The device programs itself by using erase commands to first clear a portion of the code. It then writes the new code or data into the write latches, and finally uses a programming command to program the write latch contents into the Flash array. Erase or programming commands are specified by the device specific NVMCON SFR. The NVMOP bit field selects the particular function and the ERASE bit selects between programming and erase functions. The WR bit within the NVMCON register invokes programming operations. Consequently, to protect code integrity, the device restricts the operations that occur on setting the WR bit.

23.6.1.1 ERASING AND PROGRAMMING CODE ROWS OR PAGES

Depending on the implementation of the Flash array, the NVMOP bit specifies erasing or programming a page of the program Flash array. If segment write protection is enabled, no erase or programming operations occurs within that segment.

23.6.1.2 ERASING A SEGMENT AND CLEARING CODE PROTECTION

There are several variants of NVMOP commands that will erase an entire segment of program Flash and clear the code protection Configuration bits in one operation. This is the only way to release code protection on a segment. These commands can be executed when running from any segment. These commands will not erase any contents from the segment RAM.

23.6.2 Rules for Programming Devices Using ICSP

When the device is connected to a device programmer, the allowable operations are limited to erasing, programming and verifying the device code memory.

- The device programmer uses segment erase commands to erase the device and clear the code protection.
- Programming commands are ignored, if any level of code protection is selected. To program, the General and Auxiliary Segments must have no code protection.
- Devices with any level of code protection cannot be verified. Attempts to verify code-protected devices results in reading ‘0’s.

Once the device is programmed with the desired code, the Configuration bits are written to enable the code protection level. After this operation, the only way to change the device code is by the code itself, or by erasing and clearing the code protection once more.

23.7 TYPICAL PROCEDURES FOR BOOT LOADING A DEVICE

A typical scenario for boot loading a device using code protection of these devices, is a system upgraded in the field. Here, the device uses two segments, the Auxiliary Segment and the General Segment. The General Segment contains the application. The Auxiliary Segment contains a secure boot loader. Both segments have high security enabled.

At system Reset, the device vectors to the application in the General Segment.

As the system is operating in the field, a technician connects a reprogramming tool to the system. The application recognizes this connection and branches to a location within the Auxiliary Segment access area. This branch is highly secure and the attempt to modify this branch likely results in a device Reset.

The Auxiliary Segment contains code that allows encrypted communication with the tool. The encryption keys are safe within the contents of the Auxiliary Segment code because only the Auxiliary Segment can access them. If serialized programming is used when the boot loader is initially programmed into the system, the encryption key could be specific to a particular system, further enhancing the strength of the encrypted communication.

Once the boot loader verifies valid communication with the external programming tool, it can then erase the code within the General Segment and clear the General Segment code protection.

The boot loader then receives the encrypted code update from the tool, decrypts it and programs it into the general space.

As the boot loader is running, it is immune from the disruption from interrupts or traps as it can vector those to a secure location within the boot loader itself.

As the boot loader finishes, it can program the Configuration bits to reprotect the general space, make any necessary updates to the vectors and then return to the general application.

23.8 RELATED DOCUMENTS

This section lists documents related to this section of the manual. These documents may not be written specifically for the dsPIC33E/PIC24E Product Family, but the concepts are pertinent and could be used with modification and possible limitations. The current documents related to CodeGuard™ Security are:

Title	Document #
No applications notes at this time.	

Note: For additional Application Notes and code examples for the dsPIC33E/PIC24E device family, visit the Microchip web site (www.microchip.com).

23.9 REVISION HISTORY

Revision A (February 2011)

This is the initial released version of this document.

NOTES:

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

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

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

Trademarks

ISBN: 978-1-60932-388-2

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.

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://www.microchip.com/support
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

Cleveland
Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Farmington Hills, MI
Tel: 248-538-2250
Fax: 248-538-2260

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453

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 - Chongqing
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

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 - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

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: 886-3-6578-300
Fax: 886-3-6578-370

Taiwan - Kaohsiung
Tel: 886-7-213-7830
Fax: 886-7-330-9305

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

02/18/11