CodeGuard™ Intermediate Security

HIGHLIGHTS

This section of the manual contains the following major topics:

1.0 Introduction 2
2.0 Control Registers 2
3.0 Code Segment Organization 3
4.0 Security Privileges and Rules 8
5.0 Dual Boot Security 13
6.0 Design Tips 17
7.0 Related Documents 18
8.0 Revision History 19

1.0 INTRODUCTION

CodeGuard™ Intermediate Security provides protection for the intellectual property stored in Flash program memory. The features can be programmed to allow for a wide range of system security configurations, including multiple code authors on a single device and secure field updates. These features also provide additional security enhancements in devices that incorporate dual boot program memory.

Depending on the type of device, Flash program memory can be organized into multiple (up to four) code space segments. Each of these segments has an implied security privilege level and system function. Any operation of the system that has the potential to allow discovery of code or data contents is restricted, based on the segment from which the operation originated or the segment to which the operation targets. These include:

- Programming, erasing or verifying operations
- Reads and writes of code space
- Program Flow Changes into a secured segment from outside of that segment
- Interrupt vectors into a secured segment

CodeGuard Intermediate Security features apply only to the program memory space. Data memory is not restricted and may be freely accessed from any Code Segment.

2.0 CONTROL REGISTERS

The features of program code security are controlled entirely at device start-up by the device Configuration bits. The locations of these bits are a function of the device family. For most dsPIC33 and PIC24 devices, the Configuration bits are located in the FSEC and FBSLIM Flash Configuration registers. For detailed information on a particular device family, refer to the specific device data sheet.

The relevant Configuration bits discussed in this chapter are:

- CSS<2:0> (Configuration Segment Security Configuration)
- CWRP (Configuration Segment Write-Protect)
- BSEN (Boot Segment Enable)
- BSS<1:0> (Boot Segment Security Configuration)
- BWRP (Boot Segment Write-Protect)
- GSS<1:0> (General Segment Security Configuration)
- GWRP (General Segment Write-Protect)
- AIVTDIS (Disable Alternate IVT)
- BSLIM<12:0> (Boot Segment Limit Value)

For devices with dual boot program memory, the BTMOD<1:0> bits (generally found in the FBOOT Configuration register) also modify the behavior of CodeGuard security features, depending on the Boot mode selected.

3.0 CODE SEGMENT ORGANIZATION

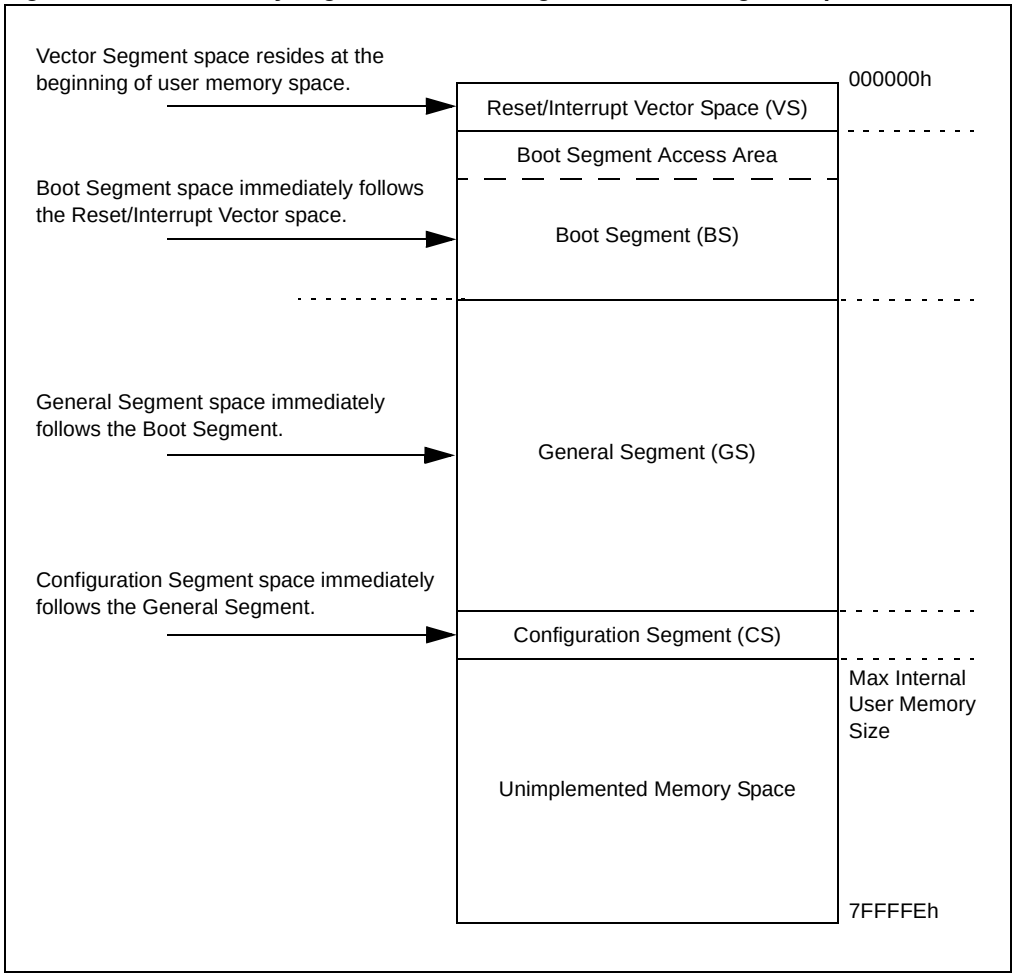
Flash program memory is divided into several segments, each having their own Code Protection (CP) and Write Protection (WRP) settings. Optionally, a Boot Segment (BS) can also be defined and partitioned from the General Segment (GS). The multiple segment approach allows restriction between segments for all types of access and operations. When operating in Dual Boot mode, the Code Segments also have restrictions between the Active and Inactive partitions.

3.1 Code Protection Bits

The individual Code Protection features for each segment are controlled by Flash Configuration bits located at the end of user program memory. These include Code Protection (CP) bits that define the security level (e.g., GSS and CSS) and Write Protection (WRP) bits, which block all write operations to a particular segment. Like all other Flash Configuration bits, the Configuration bits are set (= 1) by default and programmed by clearing (= 0) the individual bits.

Unlike other Flash Configuration bits, CP bits can only be programmed. Attempting to erase a CP bit (from '0' to '1') is not allowed. To erase a CP bit, a Chip Erase, Inactive Partition Erase or Page Erase that targets the CS page (when permitted by the existing CP values) must be used to erase all CP bits and delete Code Protection.

Figure 3-1: Memory Organization from Segment and Privilege Perspective



3.2 Boot Segment (BS)

The Boot Segment (BS) provides a highly secure code space for bootloader code, or other intellectual property that needs to be protected from other code executing on the same device or an external interface. The BS has a higher security privilege compared to the other segments and also has access to the other segments.

3.2.1 ALLOCATION OF THE BS

The BS is created by programming the BSEN Configuration bit (= 0) and defining a greater-than-zero page size with the BSLIMx Configuration bits (FBSLIM<12:0>). The value of the BSLIMx bits is stored as an inverted page address. This is done to prevent shrinking the size of the BS by clearing any of the bits, which would effectively place boot code in the lower security GS.

The BSLIMx bits are program only, like the CP bits, and are also “write-once” bits. If the value loaded from the Flash during the Reset sequence is not erased (all ‘1’s), then programming of the FBSLIM is prohibited; an attempt to do so will fail and have no effect.

3.2.2 SELECTING THE SECURITY LEVEL

The security level of the BS is set using the BSEN and BSS<1:0> Configuration bits, and is used to restrict access to the BS from the other segments. Two security options (standard and high), as well as no protection, are supported. See **Section 4.1 “Rules Concerning Program Flow”** for more information.

Figure 3-2: Boot Segment Allocation

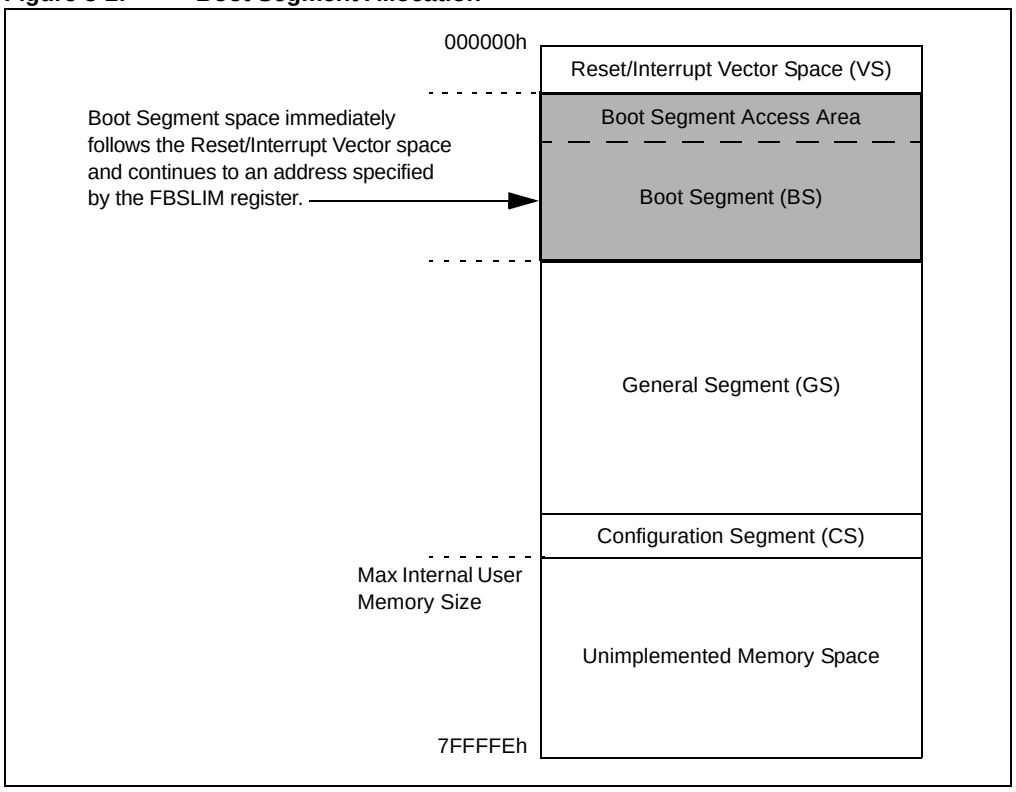


Table 3-1: Boot Segment Configurations

BSEN	BSS<1:0>	Security Level
1	xx	No Boot Segment Defined
0	11	No Security (other than optional write-protect)
0	10	Standard Security
0	0x	High Security

3.3 General Segment (GS)

The General Segment (GS) has a lower level of security privilege than the BS. Typically, the GS contains the majority of the application code. The GS begins at the page boundary after the VS or at the page boundary after the BS if the BS is implemented.

3.3.1 SECURITY LEVEL OF THE GS

Depending on the device, there are up to three levels of security to choose from for the General Segment. Configuration bits, GSS<1:0>, determine the level of protection for this segment (see Table 3-2). Two security options (standard and high), as well as no protection, are supported. See **Section 4.1 "Rules Concerning Program Flow"** for more information.

3.3.2 WRITE PROTECTION OF THE GS

The General Segment can be write-protected by programming the GWRP Configuration bit, similar to write-protecting the Boot Segment. Write Protection is disabled when the bit is unprogrammed (= 1). Programming the bit enables Write Protection for the General Segment.

Figure 3-3: General Segment Allocation

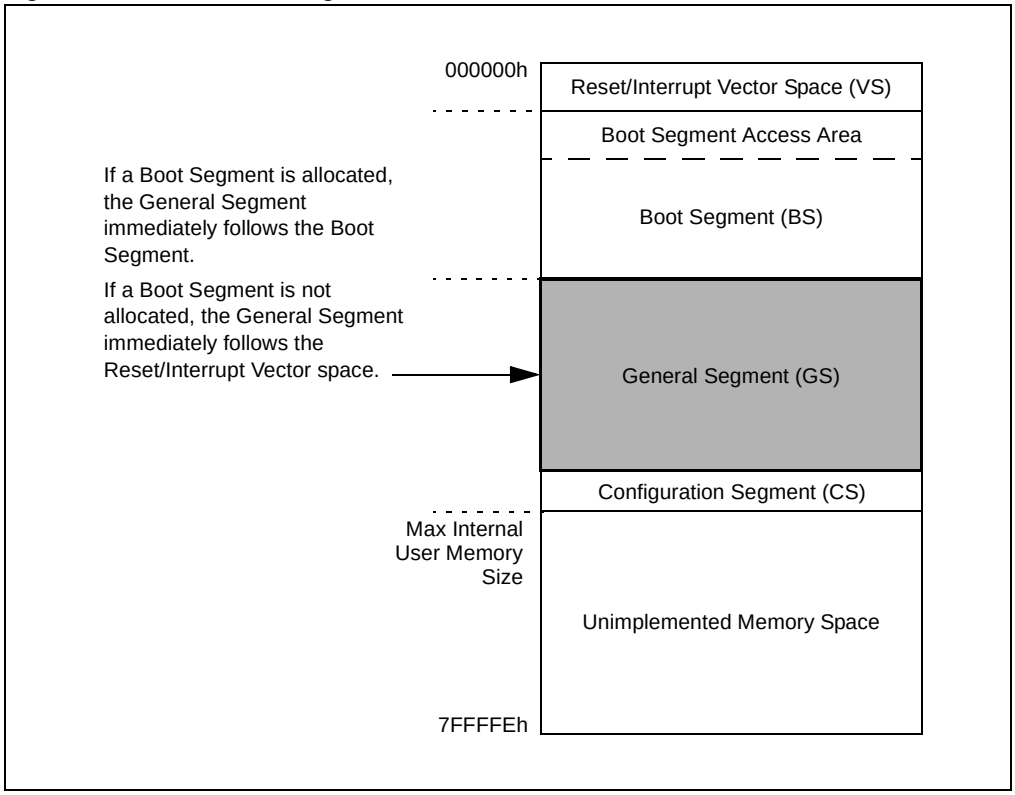


Table 3-2: General Segment Configurations

GSS<1:0>	Security Level
11	No Security (other than optional write-protect)
10	Standard Security
0x	High Security

3.4 Configuration Segment (CS)

The Configuration Segment (CS) is located in the last page of implemented program memory. The CS holds all configuration data in the Flash program memory, which is automatically read and loaded into the device Configuration registers during the Reset sequence. The CS does not contain independently executable code, so it has no special privilege level as compared to BS or GS. However, it does implement security and Write Protection that are independent from GS or BS.

3.4.1 SECURITY LEVEL OF THE CS

The security level is set to one of four levels with the CSS<2:0> Configuration bits (see Table 3-3). The CS features an additional level of security (Enhanced) to provide more flexibility in controlling Page Erase operations, independently of other program memory accesses, because the CS contains data that is critical to the security and Write Protection of the device. See **Section 4.1 "Rules Concerning Program Flow"** for more information.

3.4.2 WRITE PROTECTION OF THE CS

The Configuration Segment can be write-protected by programming the CWRP Configuration bit, similar to write-protecting the Boot Segment. Write Protection is disabled when the bit is unprogrammed (= 1). Programming the bit enables Write Protection for the Configuration Segment.

Figure 3-4: Configuration Segment Allocation

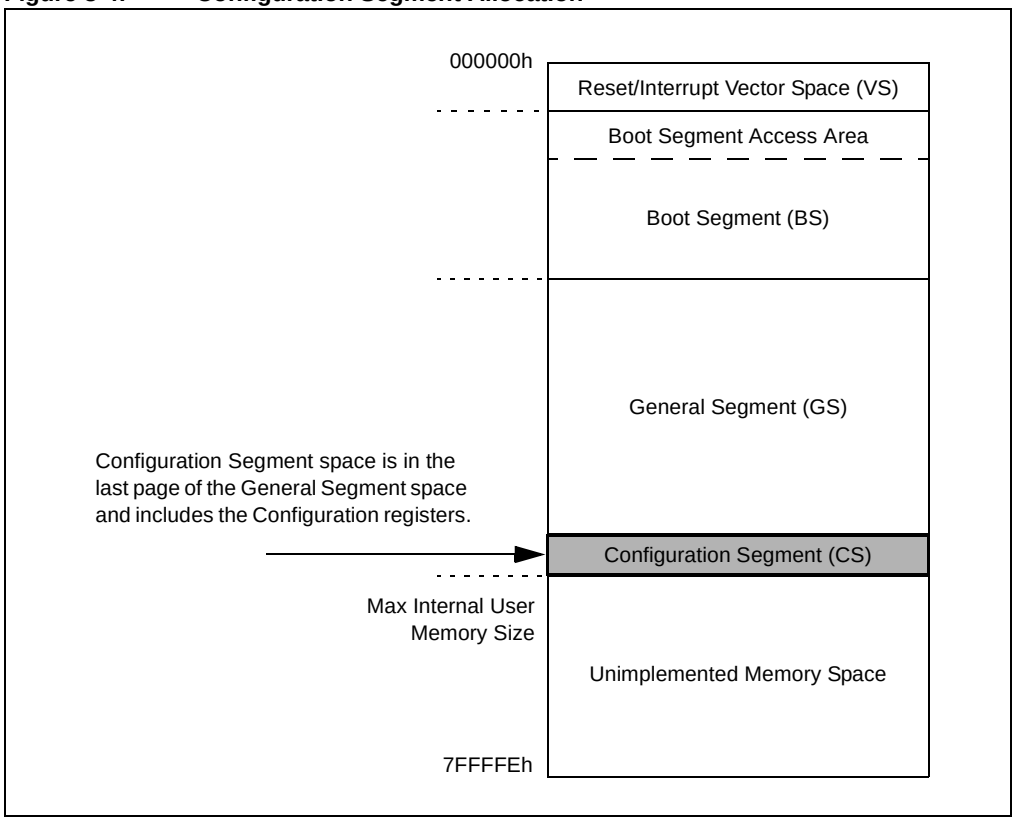


Table 3-3: Configuration Segment Configurations

CSS<2:0>	Security Level
111	No Security (other than optional write-protect)
110	Standard Security
10x	Enhanced Security
0xx	High Security

3.5 Vector Segment (VS)

The Vector Segment (VS) contains Reset, Trap and Interrupt Service Routine (ISR) vectors. For PIC24 devices, this is the first 256 instruction words of Flash memory; for dsPIC33 devices, this is the first 512 words. If a BS is defined, an optional Alternate Interrupt Vector Table (AIVT) may be used. Like the CS, the VS does not independently execute code, so it has no special privilege level in comparison to other segments.

Protection of the VS depends upon the state of the BS or GS security settings. If a Boot Segment is defined, the VS will assume the security level and Write Protection level of the BS. If no BS is defined, then the VS will assume the security level and Write Protection level of the GS.

Note:	The VS may be modified when programmed for high security. This allows programming the IVT and AIVT during a field update. See **Section 4.0 "Security Privileges and Rules"** for access conditions.

3.5.1 ALTERNATE INTERRUPT VECTOR TABLE (AIVT)

Devices with CodeGuard Intermediate Security have the option to implement a second IVT, or Alternate IVT, inside the BS. The AIVT is enabled by programming the AIVTDIS Configuration bit. To support the AIVT, the BS must be configured for a minimum size of at least two pages: one for the IVT and BS, and one for the AIVT. The AIVT will be located in the last page of the BS defined by the BSLIMx Configuration bits. Once the AIVT is enabled, the user may choose to direct exceptions to vector from the IVT or AIVT with the AIVTEN control bit (INTCON2<8>).

The AIVT inherits the BS security settings. If BS Code Protection is set to high security, all interrupts that occur while executing within the BS will vector to a single secure vector location within the BS, at the address: [BS Base Address + 40h]. This feature provides the BS the opportunity to protect the BS context and return address prior to allowing a GS ISR to execute. See **Section 4.2 "Rules Concerning Interrupts"** for details.

Note:	The Reset vector is not duplicated within the AIVT, so Resets always vector to 000000h.

3.5.2 AIVT CONSIDERATIONS FOR DUAL BOOT MODES

Both the IVT and AIVT may be read from the Active partition. This may present a lack of a security issue when updating code in the Inactive partition. To address this concern, the Inactive partition's AIVT can be disabled with the AIVTDIS control bit, effectively applying BS security to the code to block access from Code Segments in the Active partition. Code in the secure bootloader can then enable the AIVT before it is mapped to the Active partition.

4.0 SECURITY PRIVILEGES AND RULES

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 privileged segments. The lower privileged segment can only access code from the higher segment by issuing calls. Rules governing access privileges are discussed in the following sections. Table 4-1 through Table 4-4 present a summary overview of these rules during normal run-time operation.

4.1 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 a branch instruction, allowing the program flow to follow an alternate path. These instructions include Call, Jump, Computed Jump, Return or Return from Subroutine. A normal PFC only allows the program to branch within the same segment. A Restricted PFC (RPFC) 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 as the result of an interrupt request or hardware exception trap. These are primarily interrupt or trap vectors.

Jumping into secure code at unintended locations can expose the code to algorithm detection. Therefore, PFC and VFC operations are restricted if they violate the privilege hierarchy. PFCs within a segment are unrestricted. PFCs and VFCs from one segment to another are also not restricted, except when Boot Segment security is set to high. In that case, PFCs and VFCs between segments have the following restrictions:

- To ensure the integrity of the operations of code within the BS, the user must restrict program flow options to this segment by setting the security level to high
- Program flow can be limited to only allow the secure vector areas to be a branch target
- The secure vector access areas are the first 32 instruction locations of the BS

Figure 4-1 illustrates normal and restricted program flow.

The owners of the code inside of the BS 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 PFC or VFC 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 it is attempted, an address error trap results.

4.1.1 FLOW CHANGES INTO THE INACTIVE PARTITION

In Dual Boot mode, an attempted PFC to an Inactive partition address space is regarded as a flow change into an illegal address. This is because execution from Inactive partition address space is not possible and an illegal address trap will result.

Table 4-1: VS (Active Partition) Access Rules

Boot Segment		Undefined (GSS Security)						Defined (BSS Security)					
Segment Security Level		None		Standard		High		None		Standard		High	
Write Protection		N	Y	N	Y	N	Y	N	Y	N	Y	N	Y
Requested Operation													
Read of VS from	BS	N/A						Yes					
	GS	Yes						Yes					
Program/Page Erase of VS from	BS	N/A						Y	(1)	Y	(1)	Y	(1)
	GS	Y	N	Y	N	Y	N	Y	No				

Note 1: Operations from IVT are not permitted; operations from AIVT are permitted.

Table 4-2: BS and GS (Active Partition) Access Rules

Segment Security Level		None		Standard		High	
Write Protection		No	Yes	No	Yes	No	Yes
Requested Operation:							
Read of BS from	BS	Yes				Yes	
	GS	Yes		No			
Program/Page Erase of BS from	BS	Yes	No	Yes	No	Yes	No
	GS	Yes	No				
Read of GS from	BS	Yes				No	
	GS	Yes					
Program/Page Erase of GS from	BS	Yes(1)	No	Yes(1)	No		
	GS				No	Yes	No

Note 1: Page Erase of the last page of GS is defined by the security level set by CSS<2:0>.

Table 4-3: CS Access Rules

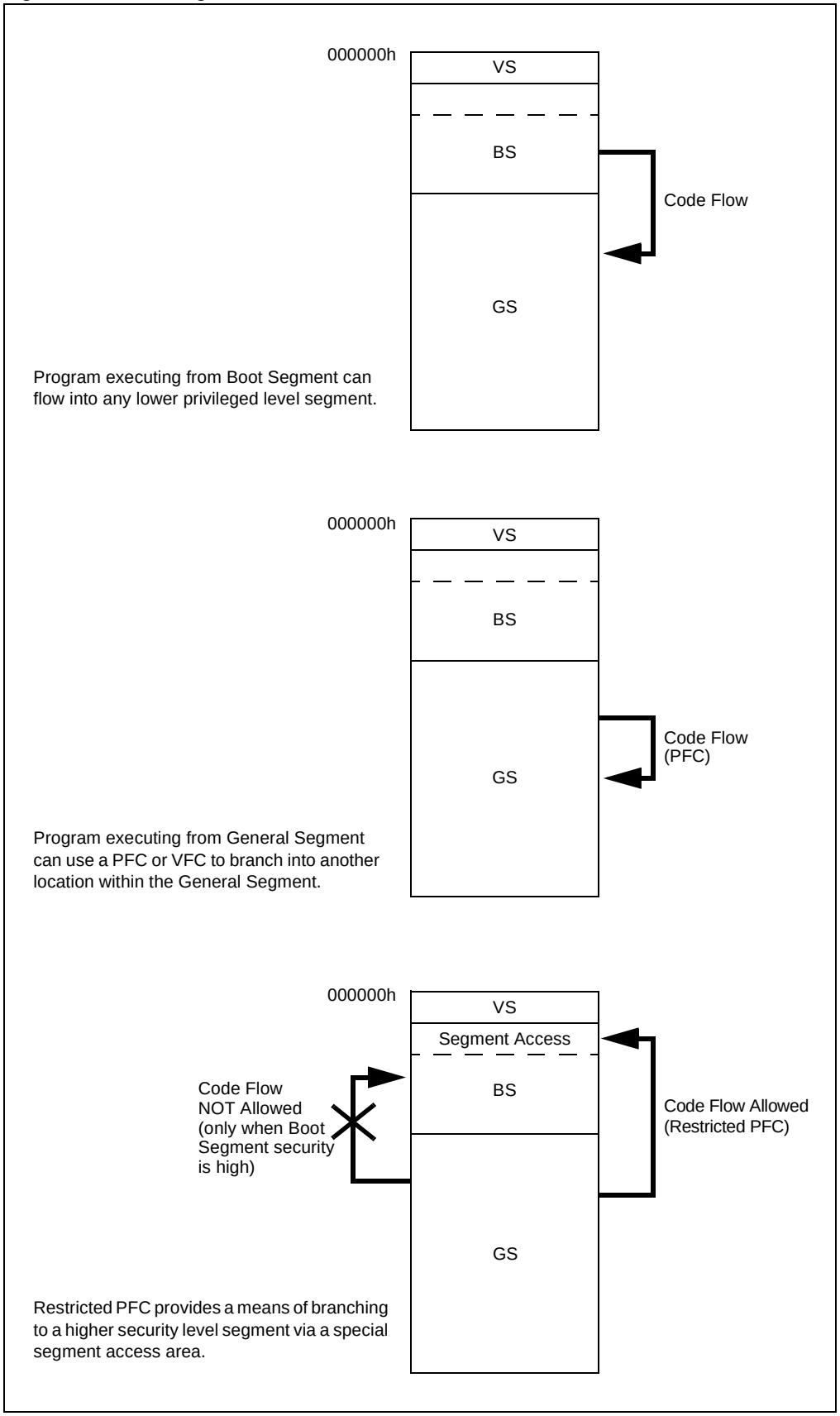
Active CS Security Level		None		Standard		Enhanced		High	
Write Protection		No	Yes	No	Yes	No	Yes	No	Yes
Requested Operation:									
Read of CS from	BS	Yes							
	GS	Yes							
Program of CS from	BS	Yes	No	Yes	No	Yes	No	Yes	No
	GS				No				
Page Erase of CS from	BS	Yes	No	Yes	No	Yes	No		
	GS		No						

Table 4-4: Partition Erase Rules

Requested Operation	From Segment	
	BS	GS
Chip Erase (Active Partition)(1)	No	No

Note 1: Chip Erase erases all user space.

Figure 4-1: Program Flow Rules



4.2 Rules Concerning Interrupts

Interrupt handling is restricted for the following reasons:

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

When BS Code Protection is high, all interrupts that occur while executing within the BS vector can be contained in a *single* secure vector location at address: [BS Start Address + 40h]. This feature provides the BS the opportunity to protect the BS context and the return address prior to allowing a GS ISR to execute. If the BS is not enabled, the IVT contains the system vectors that may target any address within GS.

4.2.1 SECURE INTERRUPT HANDLING SEQUENCE

The objective of a secure handling sequence is to delete any secure information in the W registers or data memory (to be restored later), prior to servicing a GS exception that occurs while executing from a BS configured for high security.

When an interrupt or hardware trap occurs while BS code is executing with high security, the recommended sequence of events is as follows:

1. Push the return address onto the stack.
2. Load the contents of the location, [BS Start Address + 40h], in the PC instead of the usual interrupt vector; this points to a special ISR for the BS.
3. The special BS ISR executes the following:
   a) Any secure information in the W registers or data memory is deleted.
   b) The actual return address from the stack is retrieved and saved in data memory (and encrypted if necessary).
   c) The actual return address is replaced with a new return address, located between the start of the BS and the BS Start Address + 03Eh (i.e., the first 32 instruction locations).
   d) The INTTREG is read to determine which interrupt vector to jump to.
   e) The interrupt vector is read from the Interrupt Vector Table and an indirect jump is executed.
4. Execute the application's ISR in the GS, execute user code and return from interrupt. This returns the application to the "New Return Address", which will be (or jump to) the recovery routine.
5. Read the actual return address from data memory.

4.3 Rules for Flash Access

4.3.1 RULES FOR FLASH READS

TBLRD and instructions that address program memory through the PSV addressing may be restricted. An unauthorized read of a protected program memory location will read as all '0's. The GS cannot read the BS unless the BS is configured for no security (BSS<1:0> = 11). The BS can read the GS unless the GS is configured for high security (BSS<1:0> = 0x). The CS and VS are always readable from the BS and GS, regardless of the security levels of these segments.

In devices that support the Dual Boot mode, the above rules apply between the Active and Inactive partition address spaces.

4.3.2 RULES FOR RUN-TIME SELF-PROGRAMMING (RTSP)

Run-Time Self-Programming (RTSP) is performed by first erasing a portion of Flash and then writing the new data to the write latches. The security features prevent the actual write operation based on the segment rules. If segment Write Protection is enabled, the write is blocked.

In devices that support Dual Boot mode, Write Protection is ignored for target segments that reside within the Inactive partition. The exception is the Protected Dual Boot mode, which is discussed in **Section 5.2.2 "Protected Dual Boot Mode"**. Privileged Dual Boot mode allows an Inactive partition to be erased; however, the security features will force the inactive BSLIMx bits to that of the active BSLIMx bits.

4.3.3 ERASING A SEGMENT AND CLEARING CODE PROTECTION

The Configuration Segment contains all device Code Protection control bits and resides in user space, immediately following the GS. The only way to release Code Protection of a segment is to erase the CS subject to the restrictions outlined above. It should be noted that the CS will be considerably smaller than a Flash page, so the GS is permitted to exist all the way up to the start of the CS in order to maximize available Flash space. Consequently, users should keep in mind that a CS Page Erase will:

- Assume the CS security level, which may be higher than that of the GS.
- Erase any GS code within the same page.

Therefore, CS updates (made by the BS) would also require the GS within the CS page to be rewritten.

4.3.4 RULES FOR IN-CIRCUIT SERIAL PROGRAMMING™ (ICSP™)

When the device is connected to a device programmer, the allowable operations are limited to erasing, programming and verifying the device code, and data Flash memory. The device programmer will use Chip, Partition or Page Erase commands to erase the device and clear the Code Protection. ICSP programming may only proceed on an unprotected General Segment, which is not write-protected. Attempts to verify code-protected segments within the device will return '0's. Once the device is programmed with the desired code and Boot mode, 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 ICSP erasure and clearing the Code Protection once more.

5.0 DUAL BOOT SECURITY

Note:	Dual boot operation is not present on all dsPIC33 or PIC24 devices. Please refer to the **"Memory Organization"** chapter in the specific device data sheet.

Additional security features are available for applications that use Dual Boot modes. In addition to the segment-to-segment privileges in the Active partition, there are also restrictions placed on operation from code executing in the Active partition into the Inactive partition. There are also two special Dual Boot modes that further enhance Code Protection. Code cannot be executed from any segment in the Inactive partition, including any form of erase or program operation.

5.1 Dual Boot Overview

Note:	For specific details regarding dual boot operation, refer to the **"Flash Program Memory"** section of the specific device data sheet.

When the device is in one of the Dual Boot modes, the memory can be programmed with two independent applications, each in its own partition (also referred to as Partition 1 and Partition 2). By definition, Partition 1 is the partition that is executed when the Boot Sequence Numbers of both partitions are the same value; this is generally the code partition which is programmed first into the device.

At start-up, the code in Partition 1 is mapped to the Active partition and its code is executed. The Active and Inactive partitions can be swapped, either during run time or initiated by changing the Boot Sequence Numbers and executing a device Reset.

Each code partition (i.e., Partitions 1 and 2) has its own independent Code Protection settings. This includes the Boot Segment size, security level and Write Protection that resides in each partition's VS. Write Protection is only applied to code when it is located in the Active partition and is ignored for code mapped to the Inactive partition (where it cannot be executed). This permits a write-protected segment within the Active partition to program or erase a segment in the Inactive partition, even if it is configured to be write-protected when moved into the Active partition.

Note:	If a partition swap is initiated using the `BOOTSWP` instruction, all configurations, including the BSLIMx bits and CP values, will not be reconfigured based on the newly Active partition's configuration. A device Reset is needed to reassign CP configuration data to the newly Active partition if it is different. Alternatively, they can be programmed identically to prevent security gaps when soft swapping.

Table 5-1: VS (Inactive Partition) Access Rules

Boot Segment Status		Undefined (GSS Security)			Defined (BSS Security)		
Segment Security Level		None	Standard	High	None	Standard	High
Requested Operation from Active Segment:							
Read of VS from	BS	N/A			Yes		
	GS	Yes			Yes		
Program/Page Erase of VS from	BS	N/A			Yes		
	GS	Yes			Yes	No	

Table 5-2: BS and GS (Inactive Partition) Access Rules

Segment Security Level		None	Standard	High
Requested Operation from Active Segment:				
Read of BS from	BS	Yes		
	GS	Yes	No	
Program/Page Erase of BS from	BS	Yes		
	GS	Yes	No	
Chip Erase from	BS	No		
	GS	No		
Inactive Partition Erase from	BS	Yes		
	GS	Yes	No	
Read of GS from	BS	Yes		No
	GS	Yes		
Program/Page Erase of GS from	BS	Yes		No
	GS	Yes		

Table 5-3: CS (Inactive Partition) Access Rules

Inactive CS Security Level		None	Standard	Enhanced	High
Requested Operation from Active Segment:					
Read of CS from	BS	Yes			
	GS	Yes			
Program of CS from	BS	Yes			
	GS	Yes		No	
Page Erase of CS from	BS	Yes			No
	GS	Yes	No		

5.2 Security Modes for Dual Boot

In Dual Boot mode, there are three security modes available based on the BTMODE setting:

- Dual Boot mode
- Protected Dual Boot mode
- Privileged Dual Boot mode

5.2.1 DUAL BOOT MODE

When the device operates in Dual Boot mode, the only security restrictions that are applied are those defined by the CP bits. Write Protection for code in the Inactive partition is always ignored and can always be programmed at any time. As with the Active partition, the Inactive partition VS inherits the GS privileges when no BS is defined, and inherits BS privileges when a BS is defined. Table 5-1 through Table 5-3 show the interaction from the Active partition operation on the Inactive partition's given CP settings.

5.2.2 PROTECTED DUAL BOOT MODE

Protected Dual Boot mode adds the additional capability for a "factory default" image in Partition 1 to become permanently erase/write-protected. When in Protected Dual Boot mode, Partition 1 is always write-protected when it is mapped to the Inactive partition, irrespective of its security settings. When it is mapped to the Active partition, the security of the code in Partition 1 is defined by the configuration of the write and CP bits.

5.2.2.1 Protected Dual Boot Mode Example

In the following example of Protected Dual Boot mode, the factory default code image in Partition 1 is mapped to the Inactive partition. Partition 2 contains the active code image to be executed. Factory default code should contain any code required to validate the non-factory code, as well as any procedures required for its recovery.

To achieve this configuration:

1. Configure the device for Protected Dual Boot mode.
2. Configure Partition 1 to enable a BS and with a Boot Sequence Number of FFFh.
3. Program the factory default code image.
4. Enable Write Protection for all Code Segments on Partition 1, unless it is necessary for the factory code to self-modify when it is mapped to the Active partition.
5. Program the desired application code image into (Inactive) Partition 2. Configure the partition to include a BS if the application code needs to self-modify.
6. Promote Partition 2 to the Active partition by programming the Partition 2 Boot Sequence value to be less than FFFh.

When updating the active code in the field, erase the Active partition first to reset the Boot Sequence Number, such that the factory default will be used if an error occurs.

Note:	In Protected Dual Boot mode, enabling Write Protection for Partition 1 fully protects the code from all writes and erases, except for a full Chip Erase.

5.2.3 PRIVILEGED DUAL BOOT MODE

Privileged Dual Boot mode adds additional security protection to allow for protection of intellectual property when multiple parties have software within the device by enforcing the boot size limit. Privileged Dual Boot mode is not available in all dual boot devices; refer to the specific device data sheet for more information.

When in Privileged Dual Boot mode, an Inactive partition erase operation forces the boot size limit to be automatically copied from the Active partition's Configuration Word to the Inactive partition's boot size limit. This prevents malicious code from being able to alter the inactive boot size, effectively placing Boot Segment code in the General Segment space where it may be accessed.

When operating in Privileged Dual Boot mode, it is recommended that the BS owner creates a BS of the same size in both the Active and Inactive partition space, even if nothing is to be programmed into the Inactive partition BS. This prevents the GS owner from writing malicious code into the Inactive partition GS via a lower sequence number and creating an Inactive partition BS to encompass it. The GS will have write access to the CS at this stage and the Trojan can become active at the next Reset, and could then read and dump the contents of the BS owner's code.

5.2.3.1 Privileged Dual Boot Mode Example

In the following example, application code from two separate parties is programmed into a single device that can be updated in the field without code stalling. For this example, Party 1 is the author of a proprietary algorithm and the bootloader code; Party 2 is the author of the main application code that should not have access to Party 1's proprietary algorithm. Ideally, the bootloader should use an encrypted communication scheme for field updated equipment with the encryption code secured in the BS.

In this scenario, Party 1 does the following:

1. Party 1 configures the device for Privileged Dual Boot mode with equally sized boot spaces defined in both partitions. BS Code Protection bits are configured as high; CS code is configured as standard, allowing the GS to write to, but not to erase (lower), security.
2. The device is programmed with the bootloader and proprietary algorithm into the BS; after this, Write Protection is enabled for the BS. The CS can now only be programmed from the BS and is effectively secured from Party 2 code in the GS.
3. The partially programmed device is shipped to Party 2.

Party 2 then programs the main application code into the GS, and sets the GS and CS security as high. Now the GS cannot be erased or programmed by the BS.

When field updates are required, the application code recognizes and authenticates the update request, then vectors to the bootloader in the BS. Code updates can be programmed into the Inactive partition's BS from the Active partition's BS.

Note:	In the event that the updated application requires a resized BS, the new partition size (defined by the BSLIMx bits) and the CP bits must be programmed before the application is programmed in case of a device Reset.

If the update is destined for the GS, store the data in RAM, then have the BS flag the GS that an update is required. A jump to a predefined location in the GS is done so that the GS can update itself.

The Inactive partition can be made Active and a software Reset is then executed.

6.0 DESIGN TIPS

Question 1: ***Can I bootload a device with basic Code Protection?***

Answer: Remember that devices with basic Code Protection only have one segment: the General Segment. Because there is only one segment, it is not possible to erase the segment and clear Code Protection without also erasing any bootloader that might be resident within the General Segment.

This limits the options for booting, but does not prevent it. The bootloader needs to erase and reprogram Flash in “less than segment” partitions, and the loader cannot select Write Protection for the General Segment. It is also not possible to protect the loaded code from compromises caused by the bootloader itself.

Question 2: ***Can the system load part of the code now and the rest of the code later?***

Answer: As long as neither Write Protection nor high security is selected for the segment, “incremental” loads are possible. Incremental loads are still possible in high-security segments as long as the loader resides within that segment. However, once the segment is write-protected, it cannot be changed until the entire segment is erased and Code Protection is cleared by a segment erase command.

You can choose to locate a jump table for interrupt vectors in an unprotected segment and update the jump table with changing interrupt vectors. This allows Boot Segment Write Protection.

7.0 RELATED DOCUMENTS

This section lists documents that are related to this section of the manual. These documents may not be written specifically for the dsPIC33 or PIC24 product families, but the concepts are pertinent and could be used with modification and possible limitations.

The current documents related to CodeGuard™ Intermediate Security are:

Title	Document #
CodeGuard™ Security: Protecting Intellectual Property in Collaborative System Designs	DS70179

Note: Please visit the Microchip web site (www.microchip.com) for additional Application Notes and code examples for the dsPIC33 and PIC24 families of devices.

8.0 REVISION HISTORY

Revision A (May 2014)

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

Printed on recycled paper.

ISBN: 978-1-63276-208-5

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
═ ISO/TS 16949 ═

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

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

Austin, TX
Tel: 512-257-3370

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
Novi, MI
Tel: 248-848-4000

Houston, TX
Tel: 281-894-5983

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

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

New York, NY
Tel: 631-435-6000

San Jose, CA
Tel: 408-735-9110

Canada - Toronto
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: 852-2943-5100
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8569-7000
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 - Hangzhou
Tel: 86-571-8792-8115
Fax: 86-571-8792-8116

China - Hong Kong SAR
Tel: 852-2943-5100
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: 86-755-8864-2200
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: 91-20-3019-1500

Japan - Osaka
Tel: 81-6-6152-7160
Fax: 81-6-6152-9310

Japan - Tokyo
Tel: 81-3-6880- 3770
Fax: 81-3-6880-3771

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: 886-3-5778-366
Fax: 886-3-5770-955

Taiwan - Kaohsiung
Tel: 886-7-213-7830

Taiwan - Taipei
Tel: 886-2-2508-8600
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 - Dusseldorf
Tel: 49-2129-3766400

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Germany - Pforzheim
Tel: 49-7231-424750

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Italy - Venice
Tel: 39-049-7625286

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Poland - Warsaw
Tel: 48-22-3325737

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

Sweden - Stockholm
Tel: 46-8-5090-4654

UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820

03/25/14