

---

## Section 48. Memory Organization and Permissions

---

### HIGHLIGHTS

This section of the manual contains the following major topics:

48.1	Introduction .....	48-2
48.2	Control Registers .....	48-3
48.3	Memory Layout .....	48-13
48.4	The System Bus.....	48-16
48.5	System Bus Arbitration .....	48-18
48.6	Access Permissions.....	48-19
48.7	Effects of Reset.....	48-21
48.8	Operation in Power-Saving Modes .....	48-21
48.9	Debug Mode .....	48-21
48.10	Code Examples.....	48-22
48.11	Related Application Notes.....	48-25
48.12	Revision History.....	48-26

**Note:** This family reference manual section is meant to serve as a complement to device data sheets. Depending on the device variant, this manual section may not apply to all PIC32 devices.

Please consult the note at the beginning of the “**Memory Organization**” chapter in the current device data sheet to check whether this document supports the device you are using.

Device data sheets and family reference manual sections are available for download from the Microchip Worldwide Web site at: <http://www.microchip.com>

## 48.1 INTRODUCTION

The PIC32MZ family of microcontrollers provides 4 GB of unified virtual memory address space. All memory regions, including program memory, data memory, external memory, SFRs, and Configuration registers, reside in this address space at their respective unique addresses. PIC32MZ devices can execute from boot Flash, program Flash, data memory, or external memory. A highly configurable protection scheme can prevent access by user software or DMA to selected memory regions while the secure kernel or boot code retains full access.

The PIC32MZ System Bus, which is implemented as a multi-layer fabric (crossbar), allows concurrent transactions by connecting a multitude of initiators to a multitude of targets. Bus initiators consist of the CPU, general purpose DMA, and peripherals with dedicated DMA, while bus targets include program memory, data memory, and peripherals.

Key features of PIC32MZ memory organization include the following:

- 32-bit native data width
- Integrated Memory Management Unit (MMU) with fixed mapping allows for securely configurable memory and peripheral access control permissions
- Bus arbitration scheme is implemented using a least recently serviced (LRS) priority to provide a Quality of Service (QOS) for the CPU, general purpose DMA, and peripherals with dedicated DMA
- Dual Flash panels allow for live updates of program memory
- Separate dual boot Flash memory allows updates of boot code
- Dual RAM banks can be used to avoid bus arbitration when using DMA
- External serial and or parallel memory can be mapped into virtual memory space for data access or code execution using the Serial Quad Interface (SQI) or External Bus Interface (EBI)
- Cacheable and non-cacheable address regions

## 48.2 CONTROL REGISTERS

The PIC32 MMU has the following SFRs:

- **SBFLAG: System Bus Status Flag Register**

This bit encoded register indicates which, if any, Bus targets are reporting a permission group violation errors.

- **SBTxELOG1: System Bus Target 'x' Error Log Register 1 ('x' = 0-13)**

This register provides details regarding a permission group violation error, if one exists.

- **SBTxELOG2: System Bus Target 'x' Error Log Register 2 ('x' = 0-13)**

This register provides details regarding a permission group violation error, if one exists.

- **SBTxECON: System Bus Target 'x' Error Control Register ('x' = 0-13)**

This register provides control over permission group violation error reporting.

- **SBTxECLRS: System Bus Target 'x' Single Error Clear Register ('x' = 0-13)**

This register provides a mechanism for clearing reports of single permission group violation errors.

- **SBTxECLRM: System Bus Target 'x' Multiple Error Clear Register ('x' = 0-13)**

This register provides a mechanism for clearing reports of multiple permission group violation errors.

- **SBTxREGy: System Bus Target 'x' Region 'y' Register ('x' = 0-13; 'y' = 0-8)**

This register is used to configure the base address, priority and size for a target's memory region. A target has multiple memory regions.

- **SBTxRDy: System Bus Target 'x' Region 'y' Read Permissions Register ('x' = 0-13; 'y' = 0-8)**

This register is used to configure read permissions for each of the groups of a target's memory regions.

- **SBTxWRy: System Bus Target 'x' Region 'y' Write Permissions Register ('x' = 0-13; 'y' = 0-8)**

This register is used to configure write permissions for each of the groups of a target's memory regions.

Table 48-1 provides a brief summary of the related Memory Organization and Permissions registers. Corresponding registers appear after the summary, followed by a detailed description of each bit.

**Table 48-1: System Bus SFR Summary**

Name		Bit 31/15	Bit 30/14	Bit 29/13	Bit 28/12	Bit 27/11	Bit 26/10	Bit 25/9	Bit 24/8	Bit 23/7	Bit 22/6	Bit 21/5	Bit 20/4	Bit 19/3	Bit 118/2	Bit 17/1	Bit 16/0
SBFLAG <sup>(1)</sup>	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	15:0	—	—	T13PGV	T12PGV	T11PGV	T10PGV	T9PGV	T8PGV	T7PGV	T6PGV	T5PGV	T4PGV	T3PGV	T2PGV	T1PGV	T0PGV
SBTxELOG1 <sup>(1)</sup>	31:16	MULTI	—	—	—	CODE<3:0>				—	—	—	—	—	—	—	—
	15:0	INITID<7:0>								REGION<3:0>				—	CMD<2:0>		
SBTxELOG2 <sup>(1)</sup>	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	15:0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GROUP<1:0>	
SBTxECON <sup>(1)</sup>	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	15:0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ERRP
SBTxECLRS <sup>(1)</sup>	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	15:0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLEAR
SBTxECLRM <sup>(1)</sup>	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	15:0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLEAR
SBTxREGy <sup>(1,2)</sup>	31:16	BASE<21:6>															
	15:0	BASE<5:0>							PRI	—	SIZE<4:0>				—	—	—
SBTxRDy <sup>(1,2)</sup>	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	15:0	—	—	—	—	—	—	—	—	—	—	—	—	GROUP3	GROUP2	GROUP1	GROUP0
SBTxWRy <sup>(1,2)</sup>	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	15:0	—	—	—	—	—	—	—	—	—	—	—	—	GROUP3	GROUP2	GROUP1	GROUP0

**Legend:** — = unimplemented, read as '0'.

**Note 1:** Refer to the “**Memory Organization**” chapter in the specific device data sheet for the list of available targets and their descriptions.

**Note 2:** For some target regions, certain bits in this register are read-only with preset values. Refer to the “**Memory Organization**” chapter in the specific device data sheet for more information.

## Section 48. Memory Organization and Permissions

**Register 48-1: SBFLAG: System Bus Status Flag Register**

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	U-0	U-0	R-0	R-0	R-0	R-0	R-0	R-0
	—	—	T13PGV	T12PGV	T11PGV	T10PGV	T9PGV	T8PGV
7:0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
	T7PGV	T6PGV	T5PGV	T4PGV	T3PGV	T2PGV	T1PGV	T0PGV

**Legend:**

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

bit 31-14 **Unimplemented:** Read as '0'

bit 13-0 **TxPGV:** Target 'x' Permission Group Violation Status bits ('x' = 0-13)

Refer to the “**Memory Organization**” chapter in the specific device data sheet for the list of available targets and their descriptions.

1 = Target is reporting a permission group violation

0 = Target is not reporting a permission group violation

**Note:** All errors are cleared at the source (i.e., SBTxELOG1, SBTxELOG2, SBTxECLRS, or SBTxECLRM registers).

# PIC32 Family Reference Manual

**Register 48-2: SBTxELOG1: System Bus Target 'x' Error Log Register 1 ('x' = 0-13)**

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
	MULTI	—	—	—	CODE<3:0>			
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
	INITID<7:0> <sup>(1)</sup>							
7:0	R-0	R-0	R-0	R-0	U-0	R-x	R-x	R-x
	REGION<3:0>				—	CMD<2:0>		

**Legend:**

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

bit 31 **MULTI:** Multiple Permission Violations Status bit

1 = Multiple errors have been detected

0 = No multiple errors have been detected

bit 30-28 **Unimplemented:** Read as '0'

bit 27-24 **CODE<3:0>:** Error Code bits

Indicates the type of error reported.

1111 = Reserved

•  
•  
•

0100 = Reserved

0011 = Permission violation

0010 = Reserved

0001 = Reserved

0000 = No error

bit 23-16 **Unimplemented:** Read as '0'

**Note 1:** Selections vary by device. Refer to the “**Memory Organization**” chapter in the specific device data sheet to determine which selections are available.

**Note:** Refer to the “**Memory Organization**” chapter in the specific device data sheet for the list of available System Bus targets and their descriptions.

## Section 48. Memory Organization and Permissions

### Register 48-2: SBTxELOG1: System Bus Target 'x' Error Log Register 1 ('x' = 0-13) (Continued)

bit 15-8 **INITID<7:0>**: Initiator ID of Requestor bits

11111111 = Reserved

•  
•  
•

00001111 = Reserved

00001110 = Crypto Engine

00001101 = Flash Controller

00001100 = SQ11

00001011 = CAN2

00001010 = CAN1

00001001 = Ethernet Write

00001000 = Ethernet Read

00000111 = USB

00000110 = DMA Write (DMPRI (CFGCON<25>) = 1)

00000101 = DMA Write (DMPRI (CFGCON<25>) = 0)

00000100 = DMA Read (DMPRI (CFGCON<25>) = 1)

00000011 = DMA Read (DMPRI (CFGCON<25>) = 0)

00000010 = CPU (CPUPRI (CFGCON<24>) = 1)

00000001 = CPU (CPUPRI (CFGCON<25>) = 0)

00000000 = Reserved

bit 7-4 **REGION<3:0>**: Requested Region Number bits

1111 - 0000 = Target's region that reported a permission group violation (this number corresponds to the 'y' of the SBTxREGy region definition register name)

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

bit 2-0 **CMD<2:0>**: Transaction Command of the Requestor bits

111 = Reserved

110 = Reserved

101 = Write (a non-posted write)

100 = Reserved

011 = Read (a locked read caused by a Read-Modify-Write transaction)

010 = Read

001 = Write

000 = Idle

**Note 1:** Selections vary by device. Refer to the “Memory Organization” chapter in the specific device data sheet to determine which selections are available.

**Note:** Refer to the “Memory Organization” chapter in the specific device data sheet for the list of available System Bus targets and their descriptions.

# PIC32 Family Reference Manual

**Register 48-3: SBTxELOG2: System Bus Target 'x' Error Log Register 2 ('x' = 0-13)**

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
7:0	U-0	U-0	U-0	U-0	U-0	U-0	R-0	R-0
	—	—	—	—	—	—	GROUP<1:0>	

**Legend:**

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

bit 31-3 **Unimplemented:** Read as '0'

bit 1-0 **GROUP<1:0>:** Requested Permissions Group bits

11 = Group 3

10 = Group 2

01 = Group 1

00 = Group 0

**Note:** Refer to the “Memory Organization” chapter in the specific device data sheet for the list of available System Bus targets and their descriptions.

**Register 48-4: SBTxECON: System Bus Target 'x' Error Control Register ('x' = 0-13)**

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
	—	—	—	—	—	—	—	ERRP
15:8	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
7:0	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—

**Legend:**

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

bit 31-25 **Unimplemented:** Read as '0'

bit 24 **ERRP:** Error Control bit

1 = Report protection group violation errors

0 = Do not report protection group violation errors

bit 23-0 **Unimplemented:** Read as '0'

**Note:** Refer to the “Memory Organization” chapter in the specific device data sheet for the list of available System Bus targets and their descriptions.



## Section 48. Memory Organization and Permissions

**Register 48-5: SBTxECLRS: System Bus Target 'x' Single Error Clear Register ('x' = 0-13)**

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
7:0	U-0	U-0	U-0	U-0	U-0	U-0	U-0	R-0
	—	—	—	—	—	—	—	CLEAR

**Legend:**

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

bit 31-1 **Unimplemented:** Read as '0'

bit 0 **CLEAR:** Clear Single Error on Read bit

A single error as reported via SBTxELOG1 and SBTxELOG2 is cleared by a read of this register. If the error log has been cleared, a value of '1' will be returned in the CLEAR field.

**Note:** Refer to the “**Memory Organization**” chapter in the specific device data sheet for the list of available System Bus targets and their descriptions.

**Register 48-6: SBTxECLRM: System Bus Target 'x' Multiple Error Clear Register ('x' = 0-13)**

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
7:0	U-0	U-0	U-0	U-0	U-0	U-0	U-0	R-0
	—	—	—	—	—	—	—	CLEAR

**Legend:**

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

bit 31-1 **Unimplemented:** Read as '0'

bit 0 **CLEAR:** Clear Multiple Errors on Read bit

Multiple errors as reported via SBTxELOG1 and SBTxELOG2 is cleared by a read of this register. If the error log has been cleared, a value of '1' will be returned in the CLEAR field.

**Note:** Refer to the “**Memory Organization**” chapter in the specific device data sheet for the list of available System Bus targets and their descriptions.

# PIC32 Family Reference Manual

**Register 48-7: SBTxREGy: System Bus Target 'x' Region 'y' Register ('x' = 0-13; 'y' = 0-8)**

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	R/W0	R/W-0	R/W0	R/W-0	R/W0	R/W-0	R/W0	R/W-0
	BASE<21:14>							
23:16	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	BASE<13:6>							
15:8	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x	U-0
	BASE<5:0>						PRI	—
7:0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0
	SIZE<4:0>					—	—	—

**Legend:**

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 31-10 **BASE<21:0>**: Region Base Address bits

Bits 31:10 of the base address bits of this permission region as defined in physical memory space. These bits must define an address that is aligned to the size specified by the SIZE<4:0> bits.

bit 9 **PRI**: Region Priority Level bit

1 = Level 2

0 = Level 1

bit 8 **Unimplemented**: Read as '0'

bit 7-3 **SIZE<4:0>**: Region Size bits

Permissions for a region are only active if the SIZE is non-zero. Region size =  $2^{(SIZE - 1)} \times 1024$  (bytes)

11111 = Reserved

•

•

•

11000 = Reserved

10111 = 4,294,967,296

10110 = 2,147,483,648

10101 = 1,073,741,824

10100 = 536,870,912

10011 = 268,435,456

10010 = 134,217,728

10001 = 67,108,864

10000 = 33,554,432

01111 = 16,777,216

01110 = 8,388,608

01101 = 4,194,304

01100 = 2,097,152

01011 = 1,048,576

01010 = 524,288

01001 = 262,144

01000 = 131,072

00111 = 65,536

00110 = 32,768

00101 = 16,384

00100 = 8,192

00011 = 4,096

00010 = 2,048

00001 = 1,024

00000 = Region not present

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

**Note 1:** Refer to the “**Memory Organization**” chapter in the specific device data sheet for the list of available System Bus targets and their descriptions.

**2:** For some target regions, certain bits in this register are read-only with preset values. Refer to the “**Memory Organization**” chapter in the specific device data sheet for more information.

## Section 48. Memory Organization and Permissions

**Register 48-8: SBTxRDy: System Bus Target 'x' Region 'y' Read Permissions Register ('x' = 0-13; 'y' = 0-8)**

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
7:0	U-0	U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1
	—	—	—	—	GROUP3	GROUP2	GROUP1	GROUP0

**Legend:**

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

bit 31-4 **Unimplemented:** Read as '0'

bit 3 **GROUP3:** Group 3 Read Permissions bits

1 = Privilege Group 3 has read permission

0 = Privilege Group 3 does not have read permission

bit 2 **GROUP2:** Group 2 Read Permissions bits

1 = Privilege Group 2 has read permission

0 = Privilege Group 2 does not have read permission

bit 1 **GROUP1:** Group 1 Read Permissions bits

1 = Privilege Group 1 has read permission

0 = Privilege Group 1 does not have read permission

bit 0 **GROUP0:** Group 0 Read Permissions bits

1 = Privilege Group 0 has read permission

0 = Privilege Group 0 does not have read permission

**Note 1:** Refer to the “**Memory Organization**” chapter in the specific device data sheet for the list of available System Bus targets and their descriptions.

**2:** For some target regions, certain bits in this register are read-only with preset values. Refer to the “**Memory Organization**” chapter in the specific device data sheet for more information.

# PIC32 Family Reference Manual

**Register 48-9: SBTxWRy: System Bus Target ‘x’ Region ‘y’ Write Permissions Register (‘x’ = 0-13; ‘y’ = 0-8)**

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
7:0	U-0	U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1
	—	—	—	—	GROUP3	GROUP2	GROUP1	GROUP0

**Legend:**

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as ‘0’

-n = Value at POR

‘1’ = Bit is set

‘0’ = Bit is cleared

bit 31-4 **Unimplemented:** Read as ‘0’

bit 3 **GROUP3:** Group 3 Write Permissions bits

1 = Privilege Group 3 has write permission

0 = Privilege Group 3 does not have write permission

bit 2 **GROUP2:** Group 2 Write Permissions bits

1 = Privilege Group 2 has write permission

0 = Privilege Group 2 does not have write permission

bit 1 **GROUP1:** Group 1 Write Permissions bits

1 = Privilege Group 1 has write permission

0 = Privilege Group 1 does not have write permission

bit 0 **GROUP0:** Group 0 Write Permissions bits

1 = Privilege Group 0 has write permission

0 = Privilege Group 0 does not have write permission

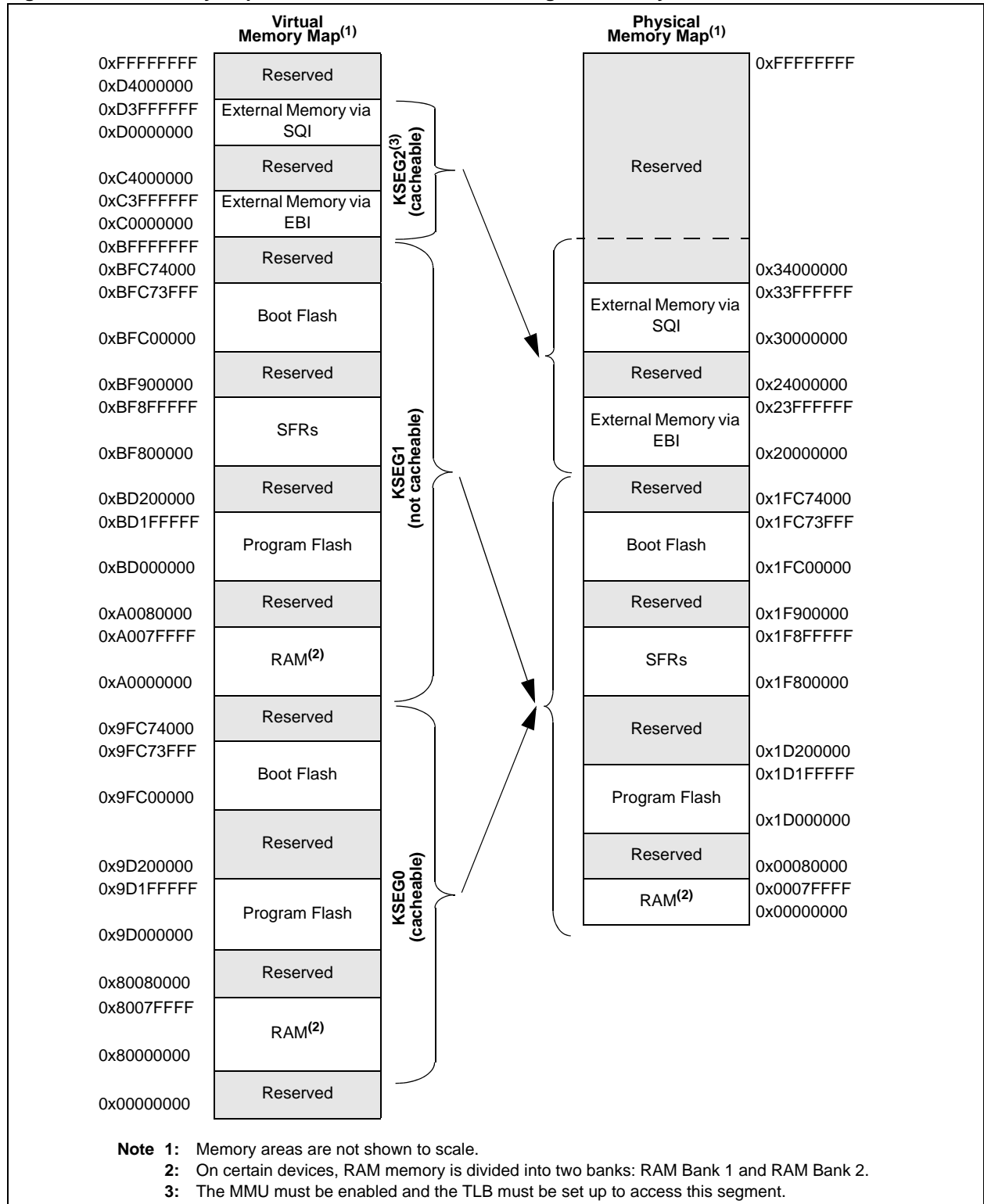
**Note 1:** Refer to the “**Memory Organization**” chapter in the specific device data sheet for the list of available System Bus targets and their descriptions.

**2:** For some target regions, certain bits in this register are read-only with preset values. Refer to the “**Memory Organization**” chapter in the specific device data sheet for more information.

## 48.3 MEMORY LAYOUT

Figure 48-1 shows an example of the memory map for a device with 2048 KB of program memory for a typical application running in Kernel mode. Physical addresses, shown on the right, are used by DMA, the Flash controller (when programming Flash memory), and when defining protected memory regions. Virtual addresses are used by the CPU for fetching and reading or writing data or peripheral SFRs.

**Figure 48-1: Memory Map for Devices with 2048 KB of Program Memory**



For a typical application running in Kernel mode, virtual memory is divided into three segments named KSEG0, KSEG1, and KSEG2. The segments, KSEG0 and KSEG1, both translate to physical address 0x0 and include all of program Flash and data memory; however, KSEG0 is cacheable and KSEG1 is not. This arrangement allows the CPU to access identical physical address space from the virtual segment of KSEG0 and KSEG1 so that the application can choose to execute any or all code as either cached or uncached by branching or calling the function in the cached or uncached region. The uncached region, KSEG1, provides virtual address space translation to the Special Function Registers for PIC32MZ family devices. KSEG2 maps to external memory, which is connected to the device through the SMI and EBI modules using the Translation Lookaside Buffer (TLB).

There are two additional virtual memory segments: a user segment, KUSEG, which occupies the lower 2 GB of virtual memory, and KSEG3, which occupies the upper most 512 MB block of memory. For information regarding the use of these segments and their configuration using the TLB, please refer to **Section 50. “CPU for Devices with microAptiv™ Core”** (DS60001192).

## 48.3.1 Virtual to Physical Address Calculation (and Vice-Versa)

To translate the virtual address in KSEG0 or KSEG1 to a physical address, perform a “Bitwise AND” operation of the virtual address with 0x1FFFFFFF:

- Physical Address = Virtual Address & 0x1FFFFFFF

For physical address to KSEG0 virtual address translation, perform a “Bitwise OR” operation of the physical address with 0x80000000:

- KSEG0 Virtual Address = Physical Address | 0x80000000

For physical address to KSEG1 virtual address translation, perform a “Bitwise OR” operation of the physical address with 0xA0000000:

- KSEG1 Virtual Address = Physical Address | 0xA0000000

To translate from KSEG0 to KSEG1 virtual address, perform a “Bitwise OR” operation of the KSEG0 virtual address with 0x20000000:

- KSEG1 Virtual Address = KSEG0 Virtual Address | 0x20000000 (since KSEG2 is mapped using the TLB, virtual to physical calculations depend on the specific configuration)

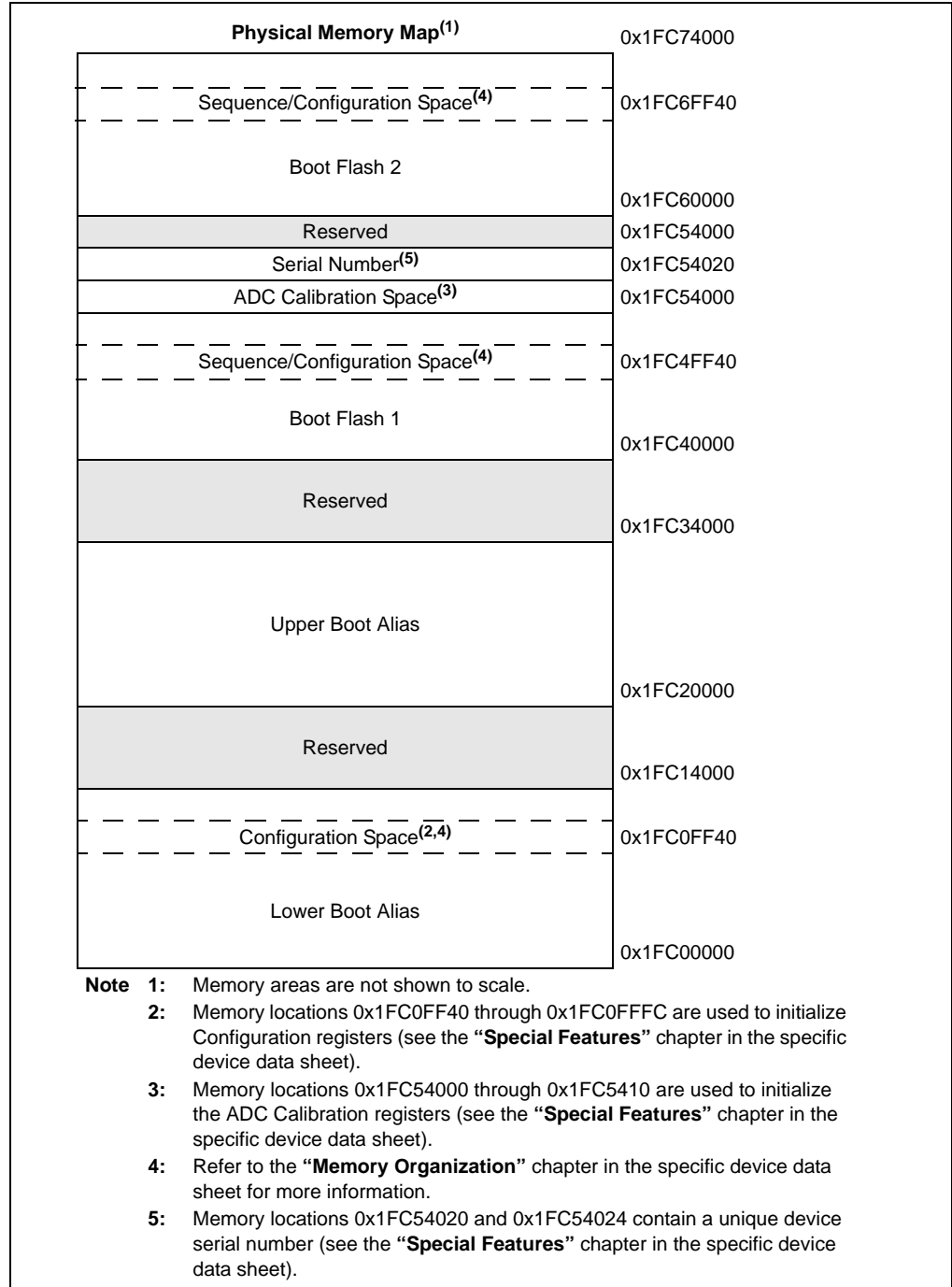
## 48.3.2 Boot Flash Memory

The Boot Flash Memory (BFM) region has special features which facilitate a Dual Boot implementation. Two identical Boot Flash regions are implemented where one is active (used at startup) and the other is available for field upgrades. Each BFM region exists in a fixed address space, either Boot Flash 1 or Boot Flash 2 and in either the Lower or Upper Boot Aliased address space as shown in [Figure 48-2](#).

At Power-on Reset (POR), the sequence words are read and the Boot Flash 1 or Boot Flash 2 region with the larger sequence number is mapped to the Lower Boot Aliased region while the smaller sequence number is mapped to the Upper Boot Aliased region. Configuration data is then read from the Lower Boot Aliased region and finally code execution begins at the reset vector located at the beginning of the Lower Boot Alias (0x1FC00000 in the example in [Figure 48-2](#)). See **Section 52. “Flash Memory with Support for Live Update”** (DS60001193) for more information.

The sequence number is stored in the lower 16 bits of the sequence word. The upper 16 bits are set to the complement of the lower sixteen bits. This allows the system to know that the sequence word has been programmed with a sequence number. In systems where Dual Boot is not used and it is desired to use both boot regions for the applications boot code, it is recommended that the Sequence Word be reserved and not used for code or data storage to insure proper aliasing at start-up.

**Figure 48-2: Boot and Alias Memory Map**



## 48.3.3 Dual Ram Banks

On certain devices, RAM Memory is divided into two banks, each with their own target bus interface. Two bus interfaces allow two different bus initiators to access RAM simultaneously without arbitration. This allows for a system design where RAM memory dedicated to CPU-only use (stack and general purpose data storage) is allocated in one RAM bank, while RAM memory dedicated to DMA use (data buffers) is allocated in another. High-speed peripherals can be storing or fetching data from RAM buffers during CPU RAM accesses without arbitration impacting their performance. Please see **Chapter 17. “Linking Programs”** in the “*MPLAB® XC32 C/C++ Compiler User’s Guide*” (DS50001686) documentation for specifying and configuring RAM usage for your project.

## 48.4 THE SYSTEM BUS

The PIC32MZ family of devices incorporates a System Bus, which is implemented as a multi-layer fabric that allows concurrent transactions by multiple initiators (bus masters) to multiple targets (bus slaves). There are no arbitration delays unless two initiators attempt access to the same target at the same time. Initiators include the CPU, general purpose DMA, and bus master peripherals with dedicated DMA access.

The following is a list of Initiators for a typical device:

- CPU
- General Purpose DMA
- Crypto Engine
- SQI
- High-Speed USB
- CAN
- Ethernet
- Flash Controller (for Run-Time Self-Programming (RTSP))

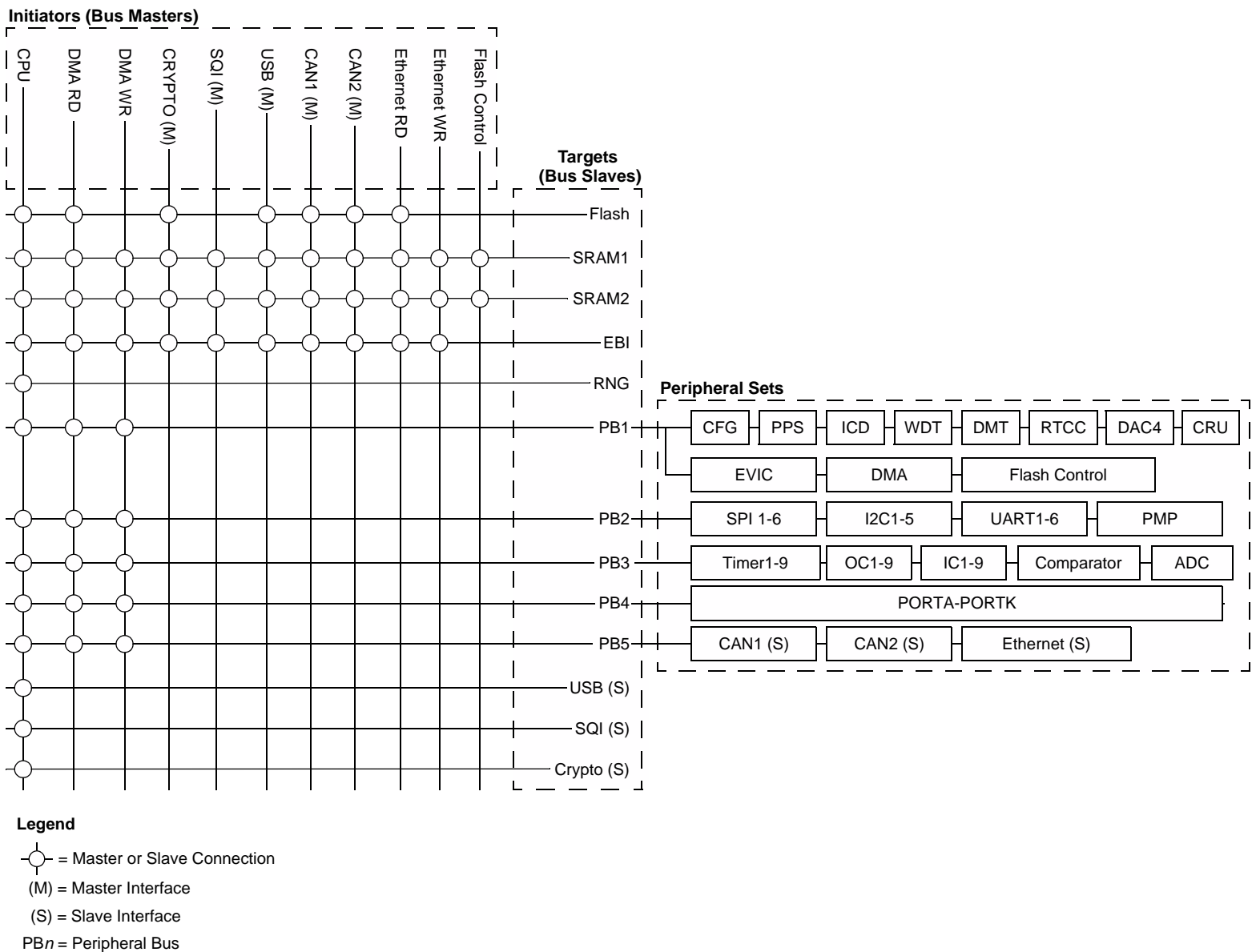
Targets include Flash and RAM memory as well as all peripherals of the device, such as the UART, SPI, I<sup>2</sup>C, etc. Some peripherals are grouped together as a single target to form a Peripheral Bus, which in addition to sharing a target interface, also shares a common peripheral bus clock source (PBCLK). See the “**Device Overview**” and “**Oscillator Configuration**” chapters of the specific device data sheet for more information.

All bus master peripherals with integrated DMA, in addition to being initiators, are also targets. This is necessary to access their Configuration SFRs. [Figure 48-3](#) provides a diagram of the PIC32MZ System Bus. All data buses are 32 bits wide. Initiators can only access targets as indicated in the diagram.

SRAM is the only target accessible by every initiator. In most applications, SRAM will be a frequent target for CPU operations, as well as DMA transfers between communication peripherals and data buffers. To avoid arbitration in these situations, many PIC32MZ implementations provide two SRAM banks. It is recommended that access to DMA serviced buffers should be through non-cached memory regions (KSEG1) or cache coherency issues will result.



Figure 48-3: System Bus Diagram



## 48.5 SYSTEM BUS ARBITRATION

Two initiators cannot access the same target at the same time. If this occurs, arbitration takes place, which prioritizes the access of the two or more initiators, granting access to one while holding off any others. Arbitration can be minimized or avoided in system design by partitioning SRAM allocation for the CPU and DMA initiators into the two SRAM banks. This also applies to devices where identical peripherals are available on different peripheral buses.

When arbitration is necessary, it is handled using a least recently serviced (LRS) priority to provide Quality of Service (QOS) for most initiators. The Flash controller initiator, which is a very low bandwidth initiator, always uses a fixed high priority to guarantee data for Flash row programming operations.

The CPU can be assigned a fixed high priority to SRAM access for interrupt processing using the CPUPRI (CFGCON<24>) Configuration bit. The general purpose DMA can be assigned a fixed high priority to SRAM access using the DMAPRI (CFGCON<25>) Configuration bit. Refer to the “**Special Features**” chapter of the specific device data sheet for information on the CFGCON register.

**Note:** DMA must always be disabled when changing the DMAPRI bit setting.

Improper use of fixed priority arbitration can have serious negative effects on other initiators. Fixed high priority settings are not recommended for an initiator that uses significant bandwidth. It is intended for low bandwidth applications that need low latency. Initiators set to fixed priority share arbitration with other initiators with fixed high priority. The dual RAM banks, however, can be utilized to allow for a DMA initiator to maintain low latency with a fixed high priority while still maintaining an adequate quality of service for a high bandwidth bus master DMA initiator. An example of this would be using DMA to implement low-cost controllerless (LCC) graphics (a low latency requirement) in a system with Ethernet or High-Speed USB. DMA buffers for graphics would be placed in one RAM bank while the Ethernet or USB buffers are placed in the other RAM bank.

Table 48-2 provides an example of a typical initiator list for a device with the initiator identifiers (ID) and quality of service priority options (QOS). The IDs are used to identify permission access violations, as discussed in 48.6 “Access Permissions”. Note that the initiators with selectable QOS have unique IDs for both the HIGH and LRS priority levels.

**Table 48-2: Initiator ID and QOS**

Name	ID	QOS
CPU	1	LRS
CPU	2	HIGH
DMA Read	3	LRS
DMA Read	4	HIGH
DMA Write	5	LRS
DMA Write	6	HIGH
USB	7	LRS
Ethernet Read	8	LRS
Ethernet Write	9	LRS
CAN1	10	LRS
CAN2	11	LRS
SQI1	12	LRS
Flash Controller	13	HIGH
Crypto Engine	14	LRS

## 48.6 ACCESS PERMISSIONS

The PIC32MZ family of devices provides configurable memory protection features that can be used to restrict access by bus initiators to bus targets or bus target regions. Access violations are logged and can optionally generate interrupts. These features are typically used by operating systems and boot loaders to prevent one task or application from interfering with the execution of another task or application and can also be used to help prevent an untrusted application from accessing protected memory regions for the purpose of intellectual property infringement.

Each bus initiator can be assigned to one of four permission groups. Each bus target contains one or more memory regions that can be configured to deny or allow access to each of the permission groups. See [Figure 48-3](#) for a diagram of typical bus targets and bus initiators.

Target permission violations can be configured to generate an interrupt allowing the operating system to intercept and handle these events. This is accomplished by setting the ERRP bit of a target's system bus error control register, SBTxECON, resulting in that target's bit being set in the SBFLAG register when a permission violation occurs. Setting of the target's bit in the SBFLAG register will result in a System Bus protection violation interrupt being generated, assuming that the interrupt is configured and enabled in the Interrupt Controller. Refer to the “**CPU Exceptions and Interrupt Controller**” chapter in the specific device data sheet for more information.

The service routine for this interrupt can query the SBFLAG of the System Bus to determine which target generated the violation. In addition, it can query the System Bus target registers, SBTxELOG1 and SBTxELOG2, for specific details about the violation including the target region where the error occurred. Finally, errors can be cleared using the System Bus target registers SBTxECLRS and SBTxECLRM.

If an initiator attempts to access a protected region, the write does not occur regardless of the interrupt enable or reporting status. Offending read instructions will return values of 0x0.

Permission groups for each initiator are assigned using the CFGPG register (see the “**Special Features**” chapter of the specific device data sheet for information on this register). On reset, all initiators are assigned to Permission Group 0.

Access permissions by initiators to the System Bus targets are configured using the SBTxREGy, SBTxRDy, and SBTxWRy registers. These three registers define a memory region and the read and write permissions for that region. Each target contains a set of these registers for the default memory space region, which is the size of the entire target address space. In addition, most targets have at least one definable (base address and size) permission memory region. The upper 22 bits of the base address are specified by the BASE<21:0> bits (SBTxREGy<31:10>) with the lower 10 bits fixed at 0. The base address must be aligned to the size specified by the SIZE<4:0> bits (SBTxREGy<7:3>). For example, a base address of 9216 (0x2400) would only allow for a size of 1024 (0x400), whereas a base address of 8192 (0x2000) would allow for sizes of 1024 (0x400), 2048 (0x800), 4096 (0x1000), and 8192 (0x2000). The memory space containing the CFGPG, SBTxREGy, SBTxRDy, and SBTxWRy registers is itself one of the target regions (Target 0 - System Bus) with definable permissions.

When an initiator attempts access to a region of memory that is defined in two target regions, the priority level determines which permission level is required. Level 0 is the lowest priority and Level 3 is the highest. Permissions for the highest level of the defined address space are always used. For any target, region 0 (the default space the size of the entire target) is always Level 0 while region 1 is always Level 3. If a target has regions beyond 0 and 1, the level for the region is defined in the read-only bit, LEVEL, in the SBTxREGy register. In these cases, a LEVEL bit value of '0' indicates a priority of 1 while a LEVEL bit value of '1' indicates priority 2. Never configure overlapping regions with the same priority level.

Configuration of permissions occurs in boot software at initialization where each initiator is assigned a permission group and the permission regions for all targets are configured. It is assumed that the boot software is trusted code. At reset, the CPU is set to Group 0 permissions. When the configurations are complete, the boot code will set the CPU privilege to the appropriate group (disabling access to the secure regions) prior to branching to the application.

**Note:** The boot code must reside in a page that has permissions of Group 0 and the Group being set when the setting of the CPUPG<1:0> (CFGPG<1:0>) Configuration bits is changed.

Permission Group 0 is intended to be the Secure permission group, configured to be the only group with permissions to the Target 0 System Bus region, which allows changing of the initiator permission groups and target permissions as well as any memory regions that the application wants to protect. The reason for this is that after the CPU permissions have been changed, blocking access to the secure target regions, the only run-time mechanism that can be used to change the CPU permission is the Non-Maskable Interrupt (NMI). When the NMI occurs, the CPU privilege is changed to Group 0 prior to vectoring to the Interrupt Service Routine (ISR). It is required that the NMI service routine will reside in memory where Group 0 privileges are allowed. After the NMI, the CPU has access to secure memory regions. Once the secure operations are complete, software must change the CPU privilege to the appropriate group blocking access to the secure regions in the same manner that it did at start-up in the boot code.

Permission Group 3 is a special case permission group for use when the processor is in Debug mode. Debug mode is the state of the processor when an external debugger has control over it so that memory or SFRs can be inspected or altered. The DBGPER<2:0> (DEVCFG0<14:12>) bits allow selection of permissions in Debug mode for Groups 0, 1, and 2. When the processor is in Debug mode and the CPU permission, defined by the CPU1PG<1:0> (CFGPG<1:0>) bits is set to one of the denied permission groups specified the DBGPER<2:0> bits, the transaction request is assigned Group 3 permissions. The System Bus target region permission registers, SBTxRDy and SBTxWRy, reset to permission settings that allow access for Groups 0 through 2 and deny access to Group 3. By using the DBGPER<2:0> bits, it is possible to prevent access by the debugger until protected boot code defines permissions specifying which memory regions are accessible by the debugger. These features require debug tools that are designed to support them. In addition, the EJTAGBEN (DEVCFG0<30>) bit should be set to '0' and code protection should be enabled.

## 48.6.1 Interrupts Vectors and Service Routines

The PIC32MZ family of devices allows placement of the ISRs and exception handlers anywhere in system memory. Care must be taken to insure that the read privileges for the target region that contains the ISR and exception handling code include any permissions that software may use for the CPU. The NMI interrupt is a special case, which always vectors to the fixed reset or NMI location that would typically be used for boot code.

## 48.6.2 Peripheral Permissions

Peripheral registers, when used in conjunction with interrupts, must also be assigned read and write permissions that allow access by any permission group that the CPU might be set to in the target application. If it is desired that access to peripherals be protected from access by the CPU, the DMA (either general purpose or DMA associated with a bus master) must be used. In this case, the DMA and the target peripheral can be assigned a permission group without regard to the permission group (or groups) used by the CPU.

## 48.6.3 Cache

Permissions are checked when the target memory is accessed by an initiator. If the target memory is SRAM or Flash and a cache is in use, the access to the target occurs when the instructions or data is moved between the target and the cache. The permissions of the CPU are not checked when the contents are accessed from cache. If the cache contains a high privilege permission group data and the CPU permission is changed to a lower privilege group, the cache must be flushed to prevent access by the lower privilege CPU setting. Flushing the cache should be done just prior to changing the CPU privilege setting.

## 48.6.4 Aliased and Bank Swapped Memory

On devices where memory is aliased and a fixed and aliased region of the same memory exists, both regions must be protected. It may also be necessary to protect the SWAP (NVMCON<7>) bit of the Flash Controller, only allowing access to the secure permission group.

## 48.7 EFFECTS OF RESET

### 48.7.1 On Reset

The contents of RAM are unchanged and the CPU is set to Permission Group 0. All target regions are set to allow access by all four permission groups. Programmable target regions and sizes are set to their reset values.

### 48.7.2 On Power-up or Brown-out Reset

The contents of RAM are undefined and the CPU is set to Permission Group 0. All target regions are set to allow access by all four permission groups. Programmable target regions and sizes are set to their reset values.

## 48.8 OPERATION IN POWER-SAVING MODES

### 48.8.1 Sleep Mode

When the device wakes from Sleep mode, the contents of RAM are unchanged and the CPU permission group is unchanged. Settings for target regions remain unchanged.

### 48.8.2 Idle Mode

When the device exits Idle mode, the CPU permission group is unchanged. Settings for target regions remain unchanged. Exiting Idle mode does not cause changes to RAM contents; however, DMA peripherals that can be enabled to run in Idle mode can modify RAM contents while the CPU is in Idle mode.

## 48.9 DEBUG MODE

To prevent access to protected regions by the debugger prior to boot code setting up the protection schemes, limit EJTAG functionality by setting the EJTAGBEN (DEVCFG0<30>) bit to '0', preventing the probe from being able to boot from EJTAG.

In addition, the Device Configuration Word, DEVCFG0, contains the DBGPER<2:0> (DEVCFG0<6:4>) bits, which are used in conjunction with the device debugging tools to control access during Debug mode. Refer to “**Special Features**” chapter in the specific device data sheets and the related development tools documentation for more information. In addition, refer to [48.6 “Access Permissions”](#) regarding use of Permission Group 3 when debugging.

**Note:** Permission groups do not prevent Debug code from setting up a trace into a protected region. If the trace is not disabled by clearing the TRCEN (DEVCFG0<5>) bit, EJTAG Serial Execution must be disabled if protection is desired.

## 48.10 CODE EXAMPLES

[Example 48-1](#) shows how to set up Flash memory permissions for a device that supports two application code spaces. Application 1, which includes the boot loading software, is trusted code. Application 2 is not trusted code and is prevented from accessing Application 1 Flash regions.

This device in this example includes dual boot banks with fixed and aliased regions. The application uses only the lower aliased boot region. The lower aliased boot Flash consists of five 16 KB memory pages for a total of 80 KB. Pages 0 through 3 have been selected as space for the boot code of Application 1 (64 KB). Page 0 must be used as it contains the reset and NMU vectors. Boot Flash page 4 (16 KB) is designated as dual privilege space needed for the Interrupt Vector Table (IVT) and for the code that changes the CPU privilege from Application 1 to Application 2. In both situations, this code must be able to execute while the CPU is in the permission group of either Application 1 or Application 2.

This device in this example includes 2 MB of program Flash memory, which is divided equally between the two applications with the upper half allocated to Application 2 and the lower half allocated to Application 1.

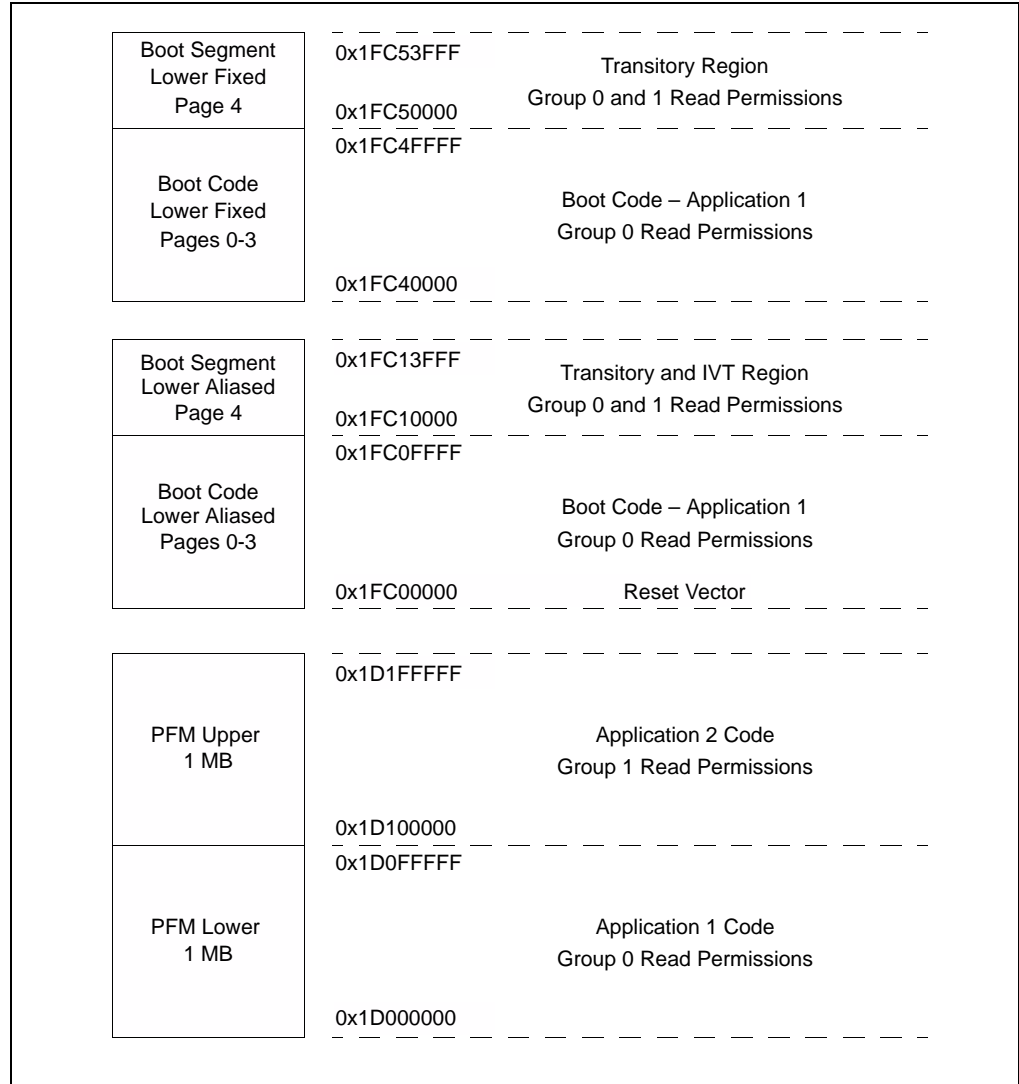
<b>Note:</b> Because of the dual boot, aliased and fixed regions, both regions must be protected with identical schemes.
--

At reset, the CPU is set to Group 0 permissions, which will be used as the Group Permission for Application 1. Application 1 initialization occurs, which includes setting target regions and permissions as defined in the example code. When Application 1 is ready to relinquish control to Application 2, it will change the CPU permission group from 0 to the permission group of Application 2, which is Permission Group 1 that used in this example. The CPU permission group change occurs in code executing from the transitory region, and then jumps to the start-up location in Application 2.

Application 1 can cooperatively relinquish control back to Application 2 by generating an NMI in software, or the Watchdog Timer (WDT) can be used to pre-empt execution of Application 2 and return control back to Application 1.

# Section 48. Memory Organization and Permissions

**Example 48-1: Flash Memory Permissions for Two Application Code Spaces**



To accomplish the permission requirements previously described, the regions must be defined and the permissions set for the IVT/transitory region and the Application 2 code space. The Flash default region that encompasses all of Flash will then be set to only allow only access by Group 0, which is the permission group for Application 1. Finally, the System Bus target permissions are defined allowing access only by Application 1.

The permissions for the IVT/transitory and the Application 2 code space will take precedence over the default permissions as they are defined in registers with higher priority levels. [Example 48-2](#) provides the code that can be used to set permissions for two application spaces.

**Note:** The code shown in [Example 48-2](#) only addresses the permission settings for the Flash memory. It does not cover the other aspects of permission control for DMA initiators or RAM, external memory, and peripheral targets.

## Example 48-2: Code to Set Permissions for Two Application Spaces

```
// Transitory Region, last page of Lower Aliased Boot Memory
// Must be configured for the aliased and fixed regions
SBT1REG3bits.BASE = 0x1FC10000 >> 10; // Lower aliased region, starting address
SBT1REG3bits.SIZE = 0x5; // Lower aliased size is one 16 KB page
SBT1REG4bits.BASE = 0x1FC50000 >> 10; // Lower fixed region, starting address
SBT1REG4bits.SIZE = 0x5; // Size is identical to alias

// Set permissions for both aliased and fixed regions
SBT1RD3 = 0x3; // Allow access by Group 0 and 1
                // permissions and block access to Group 2 and 3 permissions
SBT1RD4 = 0x3; // Duplicate for the fixed region

// Application 2 Region, upper half of PFM
SBT1REG7bits.BASE = 0x1D100000 >> 10; // PFM upper region, starting address
SBT1REG7bits.SIZE = 0xB; // Upper PFM, 1 MB size
SBT1RD7 = 0x2; // Allow access only for Group 1 permissions while blocking
                // access by groups 0, 2 and 3

// Set default region to allow access only for Group 0
SBT1RD0 = 0x1;

// Set the System Bus to allow access to only group 0
SBT0RD0 = SBT0WR0 = 0x01;
SBT0RD1 = SBT0WR1 = 0x01;
```



## Section 48. Memory Organization and Permissions

---

### 48.11 RELATED APPLICATION NOTES

This section lists application notes that are related to this section of the manual. These application notes may not be written specifically for the PIC32 device family, but the concepts are pertinent and could be used with modification and possible limitations. The current application notes related to the Memory Organization and Permissions are:

Title	Application Note #
No related application notes at this time.	N/A

<b>Note:</b> Please visit the Microchip Web site ( <a href="http://www.microchip.com">www.microchip.com</a> ) for additional application notes and code examples for the PIC32 family of devices.
---

## 48.12 REVISION HISTORY

### Revision A (November 2013)

This is the initial released version of the document.

---

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

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

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

---

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

#### **Trademarks**

The Microchip name and logo, the Microchip logo, dsPIC, FlashFlex, KEELQ, KEELQ logo, MPLAB, PIC, PICmicro, PICSTART, PIC<sup>32</sup> logo, rPIC, SST, SST Logo, SuperFlash and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

FilterLab, Hampshire, HI-TECH C, Linear Active Thermistor, MTP, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.


Analog-for-the-Digital Age, Application Maestro, BodyCom, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, HI-TIDE, In-Circuit Serial Programming, ICSP, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, mTouch, Omniscient Code Generation, PICC, PICC-18, PICDEM, PICDEM.net, PICkit, PICtail, REAL ICE, rLAB, Select Mode, SQI, Serial Quad I/O, Total Endurance, TSHARC, UniWinDriver, WiperLock, ZENA and Z-Scale are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

GestIC and ULPP are registered trademarks of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

© 2013, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

 Printed on recycled paper.

ISBN: 978-1-62077-637-7

**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, KEELQ® 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](http://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: 852-2401-1200  
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: 86-571-2819-3187  
Fax: 86-571-2819-3189

**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 - Druenen**  
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