PIC24F Data Memory

HIGHLIGHTS

This section of the manual contains the following major topics:

1.0 INTRODUCTION

As Harvard architecture devices, PIC24F microcontrollers feature separate program and data memory spaces and buses. The PIC24F architecture also allows the direct access of program memory from the data space during code execution.

2.0 DATA MEMORY ORGANIZATION

2.1 Data Address Space

The PIC24F core has a 16-bit wide data memory space, addressable as a single linear range. The data space is accessed using two Address Generation Units (AGUs), one each for read and write operations. The data space memory map is shown in Figure 2-1.

All Effective Addresses (EAs) in the data memory space are 16 bits wide and point to bytes within the data space. This gives a data space address range of 64 Kbytes or 32K words. The lower 32 Kbytes of the data memory space (that is, when EA<15> = 0) are used for implemented memory addresses, while the upper half (EA<15> = 1) is reserved for the Program Space Visibility (PSV) area. For details on PSV, refer to **Section 5.0 "Interfacing Program and Data Memory Spaces"**.

Note 1:	Please refer to the specific device data sheet for actual implementation of data memory in a specific device.
2:	If an EA points to a location outside of the physically implemented data memory area in a device, an all zero word or byte will be returned.

2.2 Data Space Width

The data memory space is organized in byte-addressable, 16-bit wide blocks. Data is aligned in data memory and registers as 16-bit words, but all the data space EAs resolve to bytes. The Least Significant Bytes (LSBs) of each word have even addresses, while the Most Significant Bytes (MSBs) have odd addresses.

2.3 Near Data Memory

An 8-Kbyte address space, between 0000h and 1FFFh, is referred to as near data memory. Near data memory is directly addressable via a 13-bit absolute address field within all the file register instructions.

Near data memory is also addressable by all the Indirect Addressing modes, where the data memory address can be pointed to by any of the 16-bit Working registers. Data memory region beyond 1FFFh is addressable only by the Indirect Addressing modes.

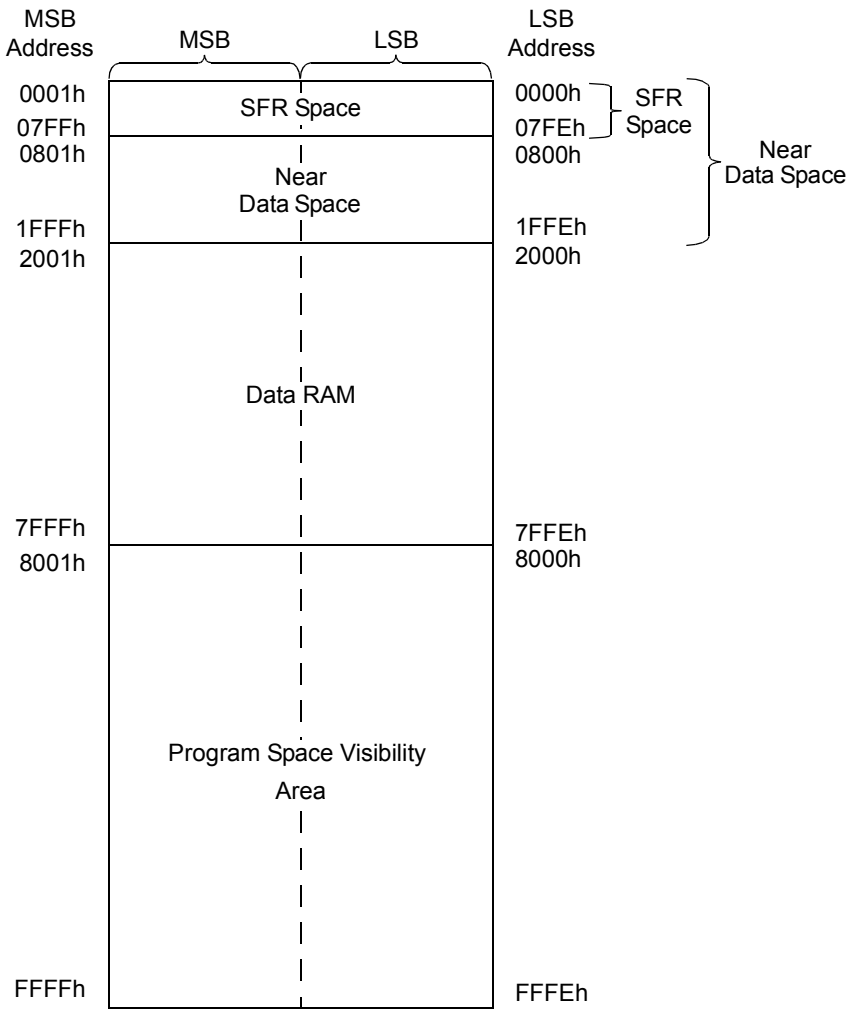
The memory regions included in the near data region will depend on the amount of data memory implemented for each PIC24F family device variant. At a minimum, the near data region will include all of the SFRs. Refer to Figure 2-1 for more details.

2.4 SFR Space

The first 2 Kbytes of the Near Data Space, from 0000h to 07FFh, are primarily occupied with Special Function Registers (SFRs). These are used by the PIC24F core and peripheral modules for controlling the operation of the device.

SFRs are distributed among the modules that they control and are generally grouped together by module. In devices with fewer peripheral modules, much of the SFR space contains unused addresses; these are read as '0'.

Figure 2-1: Data Space Memory Map for PIC24F Devices



Note 1: Data memory areas are not shown to scale.

2: Near data memory can be accessed directly via file register instructions that encode a 13-bit address into the opcode. At a minimum, the near data memory region overlaps all of the SFR space.

3: The entire data memory can be accessed indirectly via W registers, or directly, using the `MOV` instruction.

4: The upper half of the data memory can be mapped into a segment of program memory space for Program Space Visibility (PSV).

5: The actual size of the data memory implementation may differ. Refer to the specific device data sheet for more information.

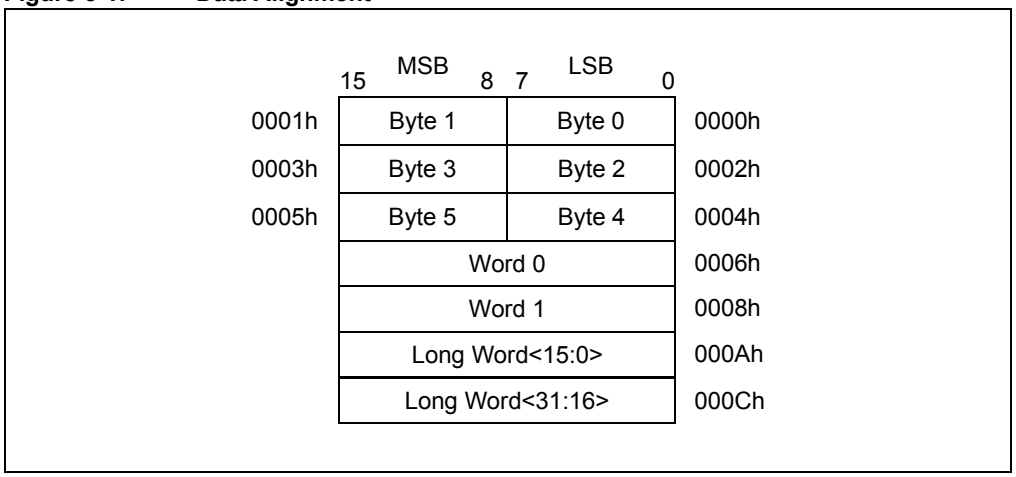
3.0 DATA ALIGNMENT

To maintain backward compatibility with PIC® MCU devices and to improve the data space memory usage efficiency, the PIC24F instruction set supports both word and byte operations. As a consequence of byte accessibility, all Effective Address (EA) calculations are internally scaled to step through word-aligned memory.

The LSb of a 16-bit data address is ignored during word operations. Word data is aligned in the little-endian format with the Least Significant Byte (LSB) at the even address (LSb = 0) and the Most Significant Byte (MSB) at the odd address (LSb = 1).

For byte operations, the LSb of the data address is used to select the byte that is accessed. Figure 3-1 shows the data alignment for word and byte operations.

Figure 3-1: Data Alignment



Data byte reads will read the complete word, which contains the byte, using the LSb of any EA to determine which byte to select. The selected byte is placed onto the LSB of the data path. That is, data memory and registers are organized as two parallel, byte-wide entities with shared (word) address decode, but separate write lines. Data byte writes only write to the corresponding side of the array or register which matches the byte address.

All Effective Address calculations are automatically adjusted depending on whether a byte or a word access is performed. For example, an address will be incremented by 2 for a word operation that post-increments the Address Pointer. Similarly, the address will be incremented by 1 for a byte operation that post-increments the Address Pointer.

All word accesses must be aligned to an even address. Misaligned word data fetches are not supported, so care must be taken when mixing byte and word operations, or translating from 8-bit MCU code. If a misaligned read or write is attempted, an address error trap will be generated. If the error occurred on a read, the instruction underway is completed; if it occurred on a write, the instruction will be executed, but the write will not occur. In either case, a trap is then executed, allowing the system and/or user to examine the machine state prior to execution of the address Fault. For additional details regarding the interrupts, refer to the *"dsPIC33/PIC24 Family Reference Manual",* **"Interrupts"** (DS70000600).

All byte loads into any W register are loaded into the Least Significant Byte. The Most Significant Byte is not modified.

A Sign-Extend (SE) instruction is provided to allow users to translate 8-bit signed data to 16-bit signed values. Alternatively, for 8-bit unsigned data, users can clear the MSB of any W register by executing a Zero-Extend (ZE) instruction on the appropriate address.

Although most instructions are capable of operating on word or byte data sizes, it should be noted that some instructions operate only on words.

4.0 SOFTWARE STACK

For stack operations, some portion of the data memory of the PIC24F devices needs to be allocated as stack. For additional details on software stack, refer to **"CPU"** (DS39703) in the *"dsPIC33/PIC24 Family Reference Manual"*.

5.0 INTERFACING PROGRAM AND DATA MEMORY SPACES

The PIC24F architecture uses a 24-bit wide program space and 16-bit wide data space. The architecture is also a modified Harvard scheme, meaning that data can also be present in the program space. To use this data successfully, it must be accessed in a way that preserves the alignment of information in both spaces.

Aside from normal execution, the PIC24F architecture provides two methods by which program space can be accessed during operation:

- Using table instructions to access individual bytes or words anywhere in the program space
- Remapping a portion of the program space into the data space (Program Space Visibility)

Table instructions allow an application to read or write to small areas of the program memory. This makes the method ideal for accessing data tables that need to be updated from time to time. It also allows access to all bytes of the program word. The remapping method allows an application to access a large block of data on a read-only basis, which is ideal for look-ups from a large table of static data. It can only access the least significant word of the program memory. For additional details regarding the program and data memory interface, please refer to **"Program Memory"** (DS39715) in the *"dsPIC33/PIC24 Family Reference Manual"*.

All PIC24 devices are able to map any page in the implemented program memory space into the data space, allowing the contents of the page to be directly read. This feature is known as Program Space Visibility (PSV).

Certain PIC24F devices expand PSV with read/write access to an extended range of virtual memory pages, which map to external memory spaces and certain data-intensive peripherals. This expansion is known as Extended Data Space (EDS). EDS is explained in detail in **"Data Memory with Extended Data Space (EDS)"** (DS39733) in the *"dsPIC33/PIC24 Family Reference Manual"*.

5.1 PSV Configuration

Program Space Visibility is enabled by setting the PSV bit (CORCON<2>). A description of the CORCON register can be found in **"CPU"** (DS39703) in the *"dsPIC33/PIC24 Family Reference Manual"*.

When PSV is enabled, each data space address in the upper half of the data memory map (data memory with higher addresses) will map directly into a program address, as shown in Figure 5-1. The PSV window allows access to the lower 16 bits of the 24-bit program word. The upper 8 bits of the program memory data should be programmed to force an illegal instruction, or a `NOP`, to maintain machine robustness. Note that table instructions provide the only method of reading the upper 8 bits of each program memory word.

Figure 5-2 illustrates how the PSV address is generated. The 15 LSbs of the PSV address are provided by the W register that contains the Effective Address. The MSb of the W register is not used to form the address. Instead, the MSb specifies whether to perform a PSV access from program space or a normal access from data memory space. If a W register Effective Address of 8000h or greater is used, the data access will occur from program memory space when PSV is enabled. All accesses will occur from data memory when the W register Effective Address is less than 8000h.

The remaining address bits are provided by the PSVPAG register (PSVPAG<7:0>), as shown in Figure 5-2. The PSVPAG bits are concatenated with the 15 LSbs of the W register, holding the Effective Address to form a 23-bit program memory address. PSV can only be used to access values in program memory space. Table instructions must be used to access values in the user configuration space.

The LSb of the W register value is used as a byte select bit, which allows instructions using PSV to operate in Byte or Word mode.

Figure 5-1: Program Space Visibility Operation

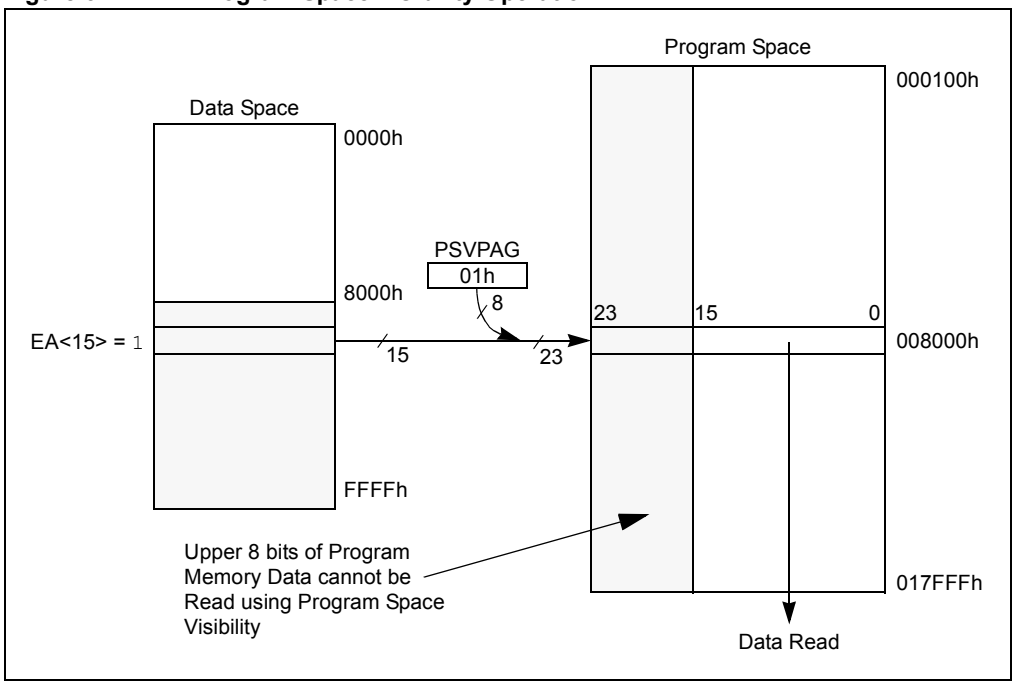
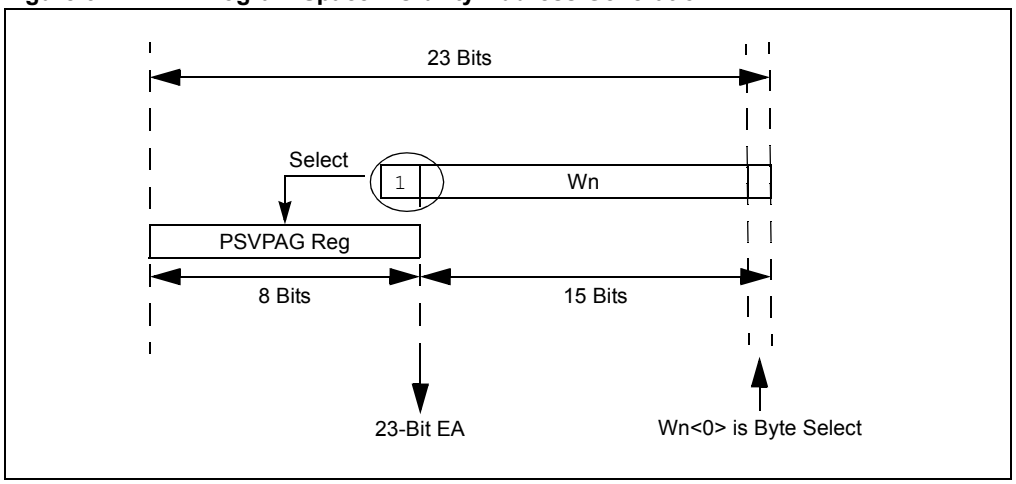


Figure 5-2: Program Space Visibility Address Generation



5.2 PSV Timing

Instructions that use PSV will require two extra instruction cycles to complete execution, except all `MOV` instructions (including the `MOV.D` instruction) that require only one extra cycle to complete execution.

The additional instruction cycles are used to fetch the PSV data on the program memory bus.

5.2.1 USING PSV IN A `REPEAT` LOOP

Instructions that use PSV within a `REPEAT` loop eliminate the extra instruction cycle(s) required for the data access from program memory, hence incurring no overhead in execution time. However, the following iterations of the `REPEAT` loop will incur an overhead of two instruction cycles to complete execution:

- The first iteration.
- The last iteration.
- Instruction execution prior to exiting the loop due to an interrupt.
- Instruction execution upon re-entering the loop after an interrupt is serviced.

5.2.2 PSV AND INSTRUCTION STALLS

For more information about instruction stalls using PSV, refer to **"CPU"** (DS39703) in the *"dsPIC33/PIC24 Family Reference Manual"*.

5.3 Using PSV with Microchip Development Tools

When developing an application in higher level languages, such as C, it is necessary to indicate to the compiler that read-only variable data for PSV access is to be located in the program memory space. To do this, the data needs to be properly defined for the compiler. This can be done several ways:

- Assign the attribute: `space(psv)`
- Assign the attribute: `space(auto_psv)`
- Use the `const` qualifier

When the PSV attribute is used, the appropriate value for the PSVPAG register must be assigned before access. The value for PSVPAG should be first saved so it can be restored after use.

The `auto_psv` attribute is the compiler-managed PSV section. Sections greater than 32 Kbytes are allowed and automatically managed. By default, the compiler places all `const` qualified variables into the `auto_psv` space. When `auto_psv` is used, the compiler will save and/or restore the PSVPAG register dynamically, as needed. The tool chain will arrange for PSVPAG to be correctly initialized in the compiler run-time start-up code.

For detailed information on these attributes, refer to **Section 10.4 "Variables in Program Space"** in the "*MPLAB® XC16 C Compiler User's Guide*" (DS50002071).

6.0 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 dsPIC33/PIC24 device families, but the concepts are pertinent and could be used with modification and possible limitations. The current application notes related to the PIC24F Data Memory are:

Title	Application Note #

No related application notes at this time.

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

7.0 REVISION HISTORY

Revision A (January 2007)

This is the initial released revision of this document.

Revision B (February 2015)

Updates the document format, and removes the previously assigned master section number as part of the realignment of PIC24 technical documentation. The document reference number format is also updated.

Updates the title to “PIC24F Data Memory” for clarity.

Expands **Section 5.0 “Interfacing Program and Data Memory Spaces”** by adding the discussion of Program Space Visibility in **Section 5.1 “PSV Configuration”** and **Section 5.2 “PSV Timing”**. The material in these sections is relocated in its entirety from the **“Program Memory”** (DS39715) section in the *“dsPIC33/PIC24 Family Reference Manual”* and does not reflect new technical content. Also adds **Section 5.3 “Using PSV with Microchip Development Tools”** as original new material.

Removes Table 3-1 (“Implemented Regions of SFR Data Space”) as it contains obsoleted information.

Other minor typographic corrections throughout the document.

NOTES:

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

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

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

Trademarks

ISBN: 978-1-63277-073-8

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 - Dongguan
Tel: 86-769-8702-9880

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

ASIA/PACIFIC

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

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

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: 886-7-213-7828

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

01/27/15