TB3242

Configuring the DMA Peripheral

Introduction

Author: Keith Curtis, Microchip Technology Inc.

The Direct Memory Access (DMA) peripheral is a byte-wide transfer system that can copy data from Program Flash Memory, Data EEPROM, General Purpose Registers (GPR) and Special Function Registers (SFR) to General Purpose or Special Function Registers. The transfer is transparent to the CPU and can be configured to interleave its transfers with the processor's operations or suspend processor operations until the DMA transfer is complete.

DMA transfers can be triggered by software and a variety of CIP trigger signals including clocks, timers outputs, comparators, communication peripherals, as well as CLCs. DMA transfers can operate as one-shot transfers or be configured to run continuously until stopped by either software intervention or a hardware CIP-based abort trigger.

Configuring the DMA peripheral involves several steps that can be a little confusing for someone who is new to the peripheral. This technical brief will cover the general configuration process and will highlight many of the available options.

Important: Several device families have multiple DMA peripherals. To access the registers associated with a specific DMA peripheral will require setting the DMASELECT register with the appropriate values. The DMA registers will then be available in the common memory address block.

To configure a DMA peripheral, the following sections need to be configured:

- Source and destination memory address
- Source and destination message sizes
- DMA trigger events
- DMA abort events (optional)
- DMA automation
- DMA interrupts (optional)
- DMA priority and enabling of the overall DMA peripheral

There will also be a ***closing section*** that covers the variety of Status bits and registers available for monitoring the DMA progress.

The source and destination memory addresses include both the address and the type of memory; EEPROM, Flash Program Memory or SFR/GPR data memory.

Important: DMA can source data from EEPROM, Flash Program Memory and SFR/GPR data memory, but can only use GPR/SFR memory as a destination.

The source and destination sizes are the number of bytes in a message. Each byte is considered a transaction, a group of transactions that occur in response to a trigger are considered a message, and a group of messages are considered a DMA operation. The size variables refer to the number of transactions that constitute a message. For example;

- Transferring data from a serial port Rx register to a data buffer would have a source size of one and a destination size equal to the buffer size.
- Transferring data from a data buffer to a serial Tx register would have a source size equal to the buffer size and a destination size of one.
- Transferring data from a data buffer to a 16-bit PWM duty cycle register would have a source size equal to the buffer size and a destination size of two.

The DMA trigger event can either be software or a CIP-based hardware trigger, such as timer roll over, comparator output or a group of signals combined in a CLC combinational logic function. The DMA abort is a similar function with both software and CIP-based hardware options.

Note: This is optional in that DMA can also be configured to operate continuously once enabled or shut down at the end of the DMA operation.

The DMA automation is what happens after the transfer; the source/destination addresses can be post incremented, post decremented or fixed. The automation also refers to which counter rollover, if any, will terminate the DMA operation.

The DMA peripheral is also capable of generating up to four specialized interrupts; one at the rollover of the source counter, one at the rollover of the destination counter, one in the event of an abort and one for an overrun error event.

Note: Individual devices may have options specific to that device. This document can be used as a general guide, but for specific settings and options, refer to the device data sheet.

Table of Contents

1. Setting the Source

The DMA peripheral can source data from three different memory systems in the microcontroller; Data EEPROM, Program Flash Memory and SFR/GPR memory.

1.1 Configuring the DMA Source

To configure the DMA source, the first step is to select the type of source memory. This is accomplished by setting the SMR bits in the DMAnCON1 register. Table 1-1 lists the possible options and the associated bit combination.

Table 1-1. Source Memory Selection

SMR[1:0]	Source Memory Region
11	Data EEPROM
10	Data EEPROM
01	Program Flash Memory
00	SFR/GPR data space

Once the type of source memory is selected, the next step is to load the DMAnSSAL, DMAnSSAH and DMAnSSAU registers. These registers specify the address of the source data. Together, the three registers can hold up to a 22-bit address.

Note: For Data EEPROM and SFR/GPR memory, only the DMAnSSAL and DMAnSSAH registers are used and the contents of the DMAnSSAU register is ignored. However, it is considered good programming practice to program DMAnSSAU with the value 0x38 for Data EEPROM access (program memory prefix address for the EEPROM area in Program Flash Memory address space).

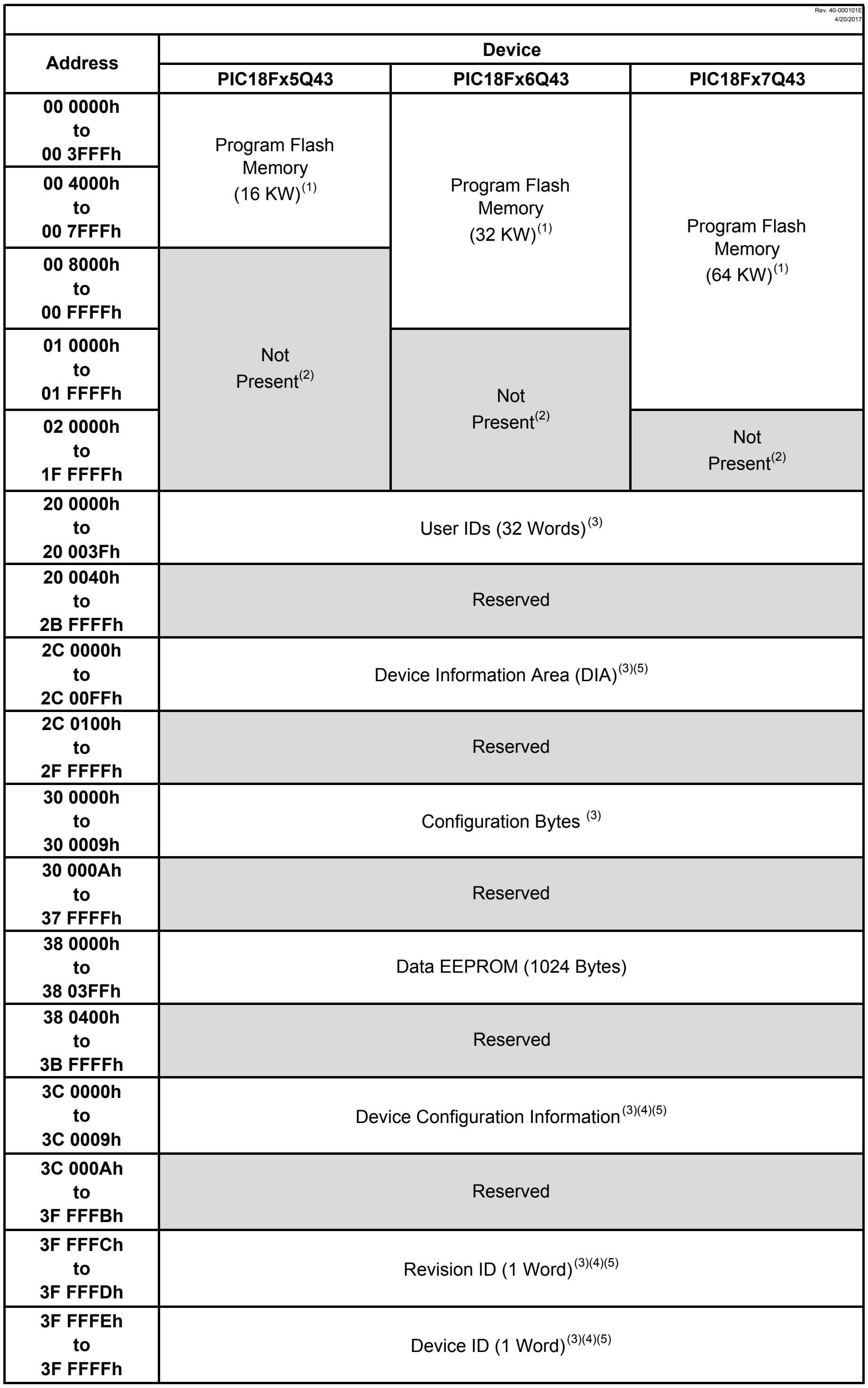
1.2 Loading the Source Registers

Loading the source registers can be accomplished by using the set address routines supplied by MCC. They can be loaded individually by the system software, or all three registers can be loaded together by equating the DMAxSSA register with a 16- or 24-bit constant or variable containing the desired address.

```
DMAnSSa = 0x2C0070; // Access Device Information Area
```

Note: If the SMR bits are set for access to PFM, data EEPROM addresses can be accessed by loading the DMAnSSAU register with the value 0x38. In fact, loading the DMAnSSAU with other 8-bit MSb values will allow access to other upper flash memory areas including the Device Information Area (DIA) and User ID words. See Figure 1-1.

Figure 1-1. Program and Data Memory Map



Rev. 40-000101E
4/20/2017

Address	Device		
	PIC18Fx5Q43	PIC18Fx6Q43	PIC18Fx7Q43
00 0000h to 00 3FFFh	Program Flash Memory (16 KW)[1]	Program Flash Memory (32 KW)[1]	Program Flash Memory (64 KW)[1]
00 4000h to 00 7FFFh			
00 8000h to 00 FFFFh	Not Present[2]		
01 0000h to 01 FFFFh		Not Present[2]	
02 0000h to 1F FFFFh			Not Present[2]
20 0000h to 20 003Fh	User IDs (32 Words)[3]		
20 0040h to 2B FFFFh	Reserved		
2C 0000h to 2C 00FFh	Device Information Area (DIA)[3][5]		
2C 0100h to 2F FFFFh	Reserved		
30 0000h to 30 0009h	Configuration Bytes [3]		
30 000Ah to 37 FFFFh	Reserved		
38 0000h to 38 03FFh	Data EEPROM (1024 Bytes)		
38 0400h to 3B FFFFh	Reserved		
3C 0000h to 3C 0009h	Device Configuration Information[3][4][5]		
3C 000Ah to 3F FFFBh	Reserved		
3F FFFCh to 3F FFFDh	Revision ID (1 Word)[3][4][5]		
3F FFFEh to 3F FFFFh	Device ID (1 Word)[3][4][5]		

Note 1: Storage Area Flash is implemented as the last 128 Words of User Flash, if enabled.

2: The addresses do not roll over. The region is read as ‘0’.

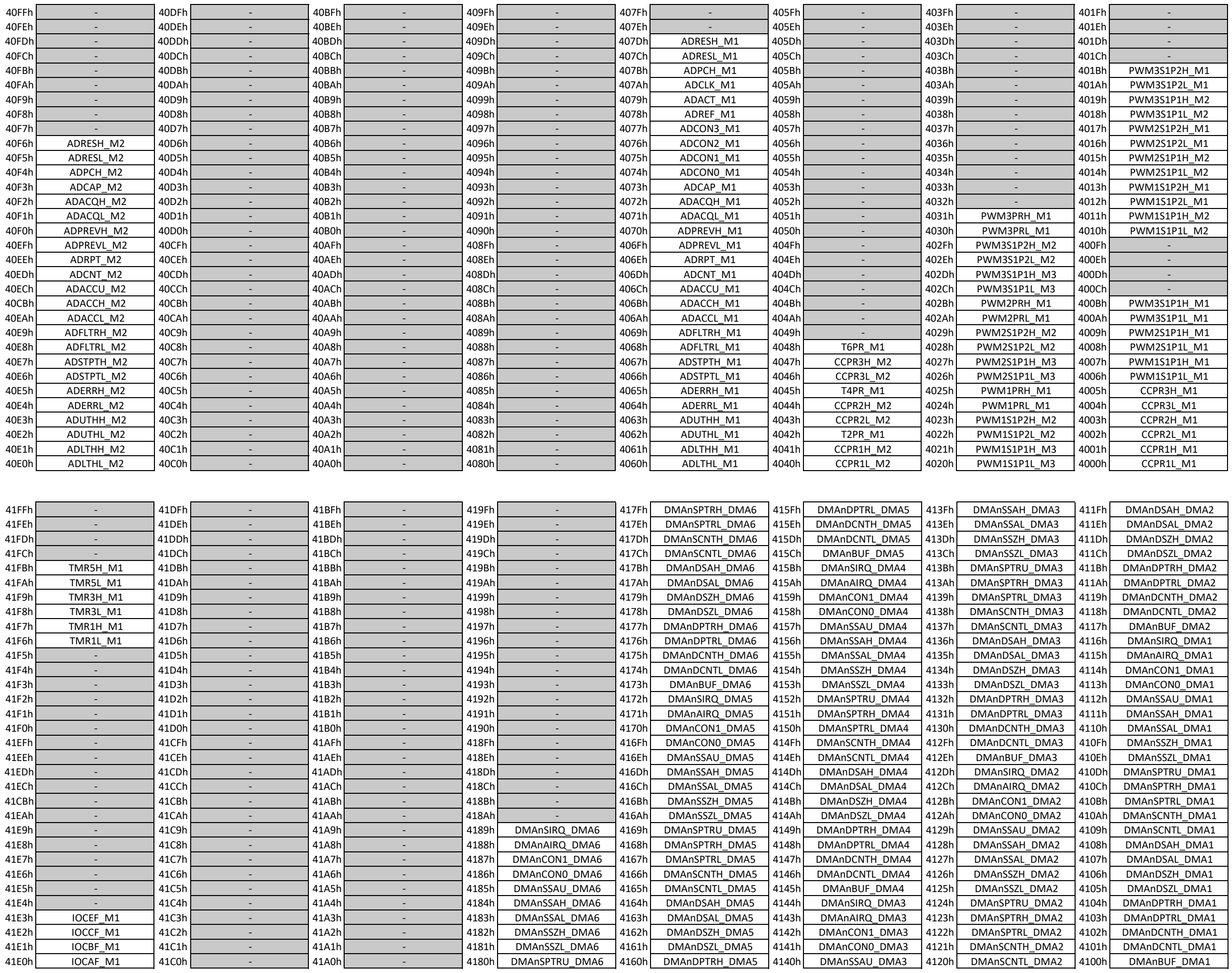
3: Not code-protected.

4: Hard-coded in silicon.

5: This region cannot be written by the user and it’s not affected by a Bulk Erase.

In addition to the standard set of SFR memory locations, an additional copy of select SFRs is available to the DMA system exclusively. This section of SFR memory contains shadow copies of multiple SFRs. For programmer's convenience, the registers have been placed adjacent to each other to facilitate DMA access. See Figure 1-2.

Figure 1-2. Special Function Register MAP (DMA Access Only)



40FFh	-	40DFh	-	40BFh	-	409Fh	-	407Fh	-	405Fh	-	403Fh	-	401Fh	-
40FEh	-	40DEh	-	40BEh	-	409Eh	-	407Eh	-	405Eh	-	403Eh	-	401Eh	-
40FDh	-	40DDh	-	40BDh	-	409Dh	-	407Dh	ADRESH_M1	405Dh	-	403Dh	-	401Dh	-
40FCh	-	40DCh	-	40BCh	-	409Ch	-	407Ch	ADRESL_M1	405Ch	-	403Ch	-	401Ch	-
40FBh	-	40DBh	-	40BBh	-	409Bh	-	407Bh	ADPCH_M1	405Bh	-	403Bh	-	401Bh	PWM3S1P2H_M1
40FAh	-	40DAh	-	40BAh	-	409Ah	-	407Ah	ADCLK_M1	405Ah	-	403Ah	-	401Ah	PWM3S1P2L_M1
40F9h	-	40D9h	-	40B9h	-	4099h	-	4079h	ADACT_M1	4059h	-	4039h	-	4019h	PWM3S1P1H_M2
40F8h	-	40D8h	-	40B8h	-	4098h	-	4078h	ADREF_M1	4058h	-	4038h	-	4018h	PWM3S1P1L_M2
40F7h	-	40D7h	-	40B7h	-	4097h	-	4077h	ADCON3_M1	4057h	-	4037h	-	4017h	PWM2S1P2H_M1
40F6h	ADRESH_M2	40D6h	-	40B6h	-	4096h	-	4076h	ADCON2_M1	4056h	-	4036h	-	4016h	PWM2S1P2L_M1
40F5h	ADRESL_M2	40D5h	-	40B5h	-	4095h	-	4075h	ADCON1_M1	4055h	-	4035h	-	4015h	PWM2S1P1H_M2
40F4h	ADPCH_M2	40D4h	-	40B4h	-	4094h	-	4074h	ADCON0_M1	4054h	-	4034h	-	4014h	PWM2S1P1L_M2
40F3h	ADCAP_M2	40D3h	-	40B3h	-	4093h	-	4073h	ADCAP_M1	4053h	-	4033h	-	4013h	PWM1S1P2H_M1
40F2h	ADACQH_M2	40D2h	-	40B2h	-	4092h	-	4072h	ADACQH_M1	4052h	-	4032h	-	4012h	PWM1S1P2L_M1
40F1h	ADACQL_M2	40D1h	-	40B1h	-	4091h	-	4071h	ADACQL_M1	4051h	-	4031h	PWM3PRH_M1	4011h	PWM1S1P1H_M2
40F0h	ADPREVH_M2	40D0h	-	40B0h	-	4090h	-	4070h	ADPREVH_M1	4050h	-	4030h	PWM3PRL_M1	4010h	PWM1S1P1L_M2
40EFh	ADPREVL_M2	40CFh	-	40AFh	-	408Fh	-	406Fh	ADPREVL_M1	404Fh	-	402Fh	PWM3S1P2H_M2	400Fh	-
40EEh	ADRPT_M2	40CEh	-	40AEh	-	408Eh	-	406Eh	ADRPT_M1	404Eh	-	402Eh	PWM3S1P2L_M2	400Eh	-
40EDh	ADCNT_M2	40CDh	-	40ADh	-	408Dh	-	406Dh	ADCNT_M1	404Dh	-	402Dh	PWM3S1P1H_M3	400Dh	-
40ECh	ADACCU_M2	40CCh	-	40ACh	-	408Ch	-	406Ch	ADACCU_M1	404Ch	-	402Ch	PWM3S1P1L_M3	400Ch	-
40CBh	ADACCH_M2	40CBh	-	40ABh	-	408Bh	-	406Bh	ADACCH_M1	404Bh	-	402Bh	PWM2PRH_M1	400Bh	PWM3S1P1H_M1
40EAh	ADACCL_M2	40CAh	-	40AAh	-	408Ah	-	406Ah	ADACCL_M1	404Ah	-	402Ah	PWM2PRL_M1	400Ah	PWM3S1P1L_M1
40E9h	ADFLTRH_M2	40C9h	-	40A9h	-	4089h	-	4069h	ADFLTRH_M1	4049h	-	4029h	PWM2S1P2H_M2	4009h	PWM2S1P1H_M1
40E8h	ADFLTRL_M2	40C8h	-	40A8h	-	4088h	-	4068h	ADFLTRL_M1	4048h	T6PR_M1	4028h	PWM2S1P2L_M2	4008h	PWM2S1P1L_M1
40E7h	ADSTPTH_M2	40C7h	-	40A7h	-	4087h	-	4067h	ADSTPTH_M1	4047h	CCPR3H_M2	4027h	PWM2S1P1H_M3	4007h	PWM1S1P1H_M1
40E6h	ADSTPTL_M2	40C6h	-	40A6h	-	4086h	-	4066h	ADSTPTL_M1	4046h	CCPR3L_M2	4026h	PWM2S1P1L_M3	4006h	PWM1S1P1L_M1
40E5h	ADERRH_M2	40C5h	-	40A5h	-	4085h	-	4065h	ADERRH_M1	4045h	T4PR_M1	4025h	PWM1PRH_M1	4005h	CCPR3H_M1
40E4h	ADERRL_M2	40C4h	-	40A4h	-	4084h	-	4064h	ADERRL_M1	4044h	CCPR2H_M2	4024h	PWM1PRL_M1	4004h	CCPR3L_M1
40E3h	ADUTHH_M2	40C3h	-	40A3h	-	4083h	-	4063h	ADUTHH_M1	4043h	CCPR2L_M2	4023h	PWM1S1P2H_M2	4003h	CCPR2H_M1
40E2h	ADUTHL_M2	40C2h	-	40A2h	-	4082h	-	4062h	ADUTHL_M1	4042h	T2PR_M1	4022h	PWM1S1P2L_M2	4002h	CCPR2L_M1
40E1h	ADLTHH_M2	40C1h	-	40A1h	-	4081h	-	4061h	ADLTHH_M1	4041h	CCPR1H_M2	4021h	PWM1S1P1H_M3	4001h	CCPR1H_M1
40E0h	ADLTHL_M2	40C0h	-	40A0h	-	4080h	-	4060h	ADLTHL_M1	4040h	CCPR1L_M2	4020h	PWM1S1P1L_M3	4000h	CCPR1L_M1

41FFh	-	41DFh	-	41BFh	-	419Fh	-	417Fh	DMAnSPTRH_DMA6	415Fh	DMAnDPTRL_DMA5	413Fh	DMAnSSAH_DMA3	411Fh	DMAnDSAH_DMA2
41FEh	-	41DEh	-	41BEh	-	419Eh	-	417Eh	DMAnSPTRL_DMA6	415Eh	DMAnDCNTH_DMA5	413Eh	DMAnSSAL_DMA3	411Eh	DMAnDSAL_DMA2
41FDh	-	41DDh	-	41BDh	-	419Dh	-	417Dh	DMAnSCNTH_DMA6	415Dh	DMAnDCNTL_DMA5	413Dh	DMAnSSZH_DMA3	411Dh	DMAnDSZH_DMA2
41FCh	-	41DCh	-	41BCh	-	419Ch	-	417Ch	DMAnSCNTL_DMA6	415Ch	DMAnBUF_DMA5	413Ch	DMAnSSZL_DMA3	411Ch	DMAnDSZL_DMA2
41FBh	TMR5H_M1	41DBh	-	41BBh	-	419Bh	-	417Bh	DMAnDSAH_DMA6	415Bh	DMAnSIRQ_DMA4	413Bh	DMAnSPTRU_DMA3	411Bh	DMAnDPTRH_DMA2
41FAh	TMR5L_M1	41DAh	-	41BAh	-	419Ah	-	417Ah	DMAnDSAL_DMA6	415Ah	DMAnAIRQ_DMA4	413Ah	DMAnSPTRH_DMA3	411Ah	DMAnDPTRL_DMA2
41F9h	TMR3H_M1	41D9h	-	41B9h	-	4199h	-	4179h	DMAnDSZH_DMA6	4159h	DMAnCON1_DMA4	4139h	DMAnSPTRL_DMA3	4119h	DMAnDCNTH_DMA2
41F8h	TMR3L_M1	41D8h	-	41B8h	-	4198h	-	4178h	DMAnDSZL_DMA6	4158h	DMAnCON0_DMA4	4138h	DMAnSCNTH_DMA3	4118h	DMAnDCNTL_DMA2
41F7h	TMR1H_M1	41D7h	-	41B7h	-	4197h	-	4177h	DMAnDPTRH_DMA6	4157h	DMAnSSAU_DMA4	4137h	DMAnSCNTL_DMA3	4117h	DMAnBUF_DMA2
41F6h	TMR1L_M1	41D6h	-	41B6h	-	4196h	-	4176h	DMAnDPTRL_DMA6	4156h	DMAnSSAH_DMA4	4136h	DMAnDSAH_DMA3	4116h	DMAnSIRQ_DMA1
41F5h	-	41D5h	-	41B5h	-	4195h	-	4175h	DMAnDCNTH_DMA6	4155h	DMAnSSAL_DMA4	4135h	DMAnDSAL_DMA3	4115h	DMAnAIRQ_DMA1
41F4h	-	41D4h	-	41B4h	-	4194h	-	4174h	DMAnDCNTL_DMA6	4154h	DMAnSSZH_DMA4	4134h	DMAnDSZH_DMA3	4114h	DMAnCON1_DMA1
41F3h	-	41D3h	-	41B3h	-	4193h	-	4173h	DMAnBUF_DMA6	4153h	DMAnSSZL_DMA4	4133h	DMAnDSZL_DMA3	4113h	DMAnCON0_DMA1
41F2h	-	41D2h	-	41B2h	-	4192h	-	4172h	DMAnSIRQ_DMA5	4152h	DMAnSPTRU_DMA4	4132h	DMAnDPTRH_DMA3	4112h	DMAnSSAU_DMA1
41F1h	-	41D1h	-	41B1h	-	4191h	-	4171h	DMAnAIRQ_DMA5	4151h	DMAnSPTRH_DMA4	4131h	DMAnDPTRL_DMA3	4111h	DMAnSSAH_DMA1
41F0h	-	41D0h	-	41B0h	-	4190h	-	4170h	DMAnCON1_DMA5	4150h	DMAnSPTRL_DMA4	4130h	DMAnDCNTH_DMA3	4110h	DMAnSSAL_DMA1
41EFh	-	41CFh	-	41AFh	-	418Fh	-	416Fh	DMAnCON0_DMA5	414Fh	DMAnSCNTH_DMA4	412Fh	DMAnDCNTL_DMA3	410Fh	DMAnSSZH_DMA1
41EEh	-	41CEh	-	41AEh	-	418Eh	-	416Eh	DMAnSSAU_DMA5	414Eh	DMAnSCNTL_DMA4	412Eh	DMAnBUF_DMA3	410Eh	DMAnSSZL_DMA1
41EDh	-	41CDh	-	41ADh	-	418Dh	-	416Dh	DMAnSSAH_DMA5	414Dh	DMAnDSAH_DMA4	412Dh	DMAnSIRQ_DMA2	410Dh	DMAnSPTRU_DMA1
41ECh	-	41CCh	-	41ACh	-	418Ch	-	416Ch	DMAnSSAL_DMA5	414Ch	DMAnDSAL_DMA4	412Ch	DMAnAIRQ_DMA2	410Ch	DMAnSPTRH_DMA1
41CBh	-	41CBh	-	41ABh	-	418Bh	-	416Bh	DMAnSSZH_DMA5	414Bh	DMAnDSZH_DMA4	412Bh	DMAnCON1_DMA2	410Bh	DMAnSPTRL_DMA1
41EAh	-	41CAh	-	41AAh	-	418Ah	-	416Ah	DMAnSSZL_DMA5	414Ah	DMAnDSZL_DMA4	412Ah	DMAnCON0_DMA2	410Ah	DMAnSCNTH_DMA1
41E9h	-	41C9h	-	41A9h	-	4189h	DMAnSIRQ_DMA6	4169h	DMAnSPTRU_DMA5	4149h	DMAnDPTRH_DMA4	4129h	DMAnSSAU_DMA2	4109h	DMAnSCNTL_DMA1
41E8h	-	41C8h	-	41A8h	-	4188h	DMAnAIRQ_DMA6	4168h	DMAnSPTRH_DMA5	4148h	DMAnDPTRL_DMA4	4128h	DMAnSSAH_DMA2	4108h	DMAnDSAH_DMA1
41E7h	-	41C7h	-	41A7h	-	4187h	DMAnCON1_DMA6	4167h	DMAnSPTRL_DMA5	4147h	DMAnDCNTH_DMA4	4127h	DMAnSSAL_DMA2	4107h	DMAnDSAL_DMA1
41E6h	-	41C6h	-	41A6h	-	4186h	DMAnCON0_DMA6	4166h	DMAnSCNTH_DMA5	4146h	DMAnDCNTL_DMA4	4126h	DMAnSSZH_DMA2	4106h	DMAnDSZH_DMA1
41E5h	-	41C5h	-	41A5h	-	4185h	DMAnSSAU_DMA6	4165h	DMAnSCNTL_DMA5	4145h	DMAnBUF_DMA4	4125h	DMAnSSZL_DMA2	4105h	DMAnDSZL_DMA1
41E4h	-	41C4h	-	41A4h	-	4184h	DMAnSSAH_DMA6	4164h	DMAnDSAH_DMA5	4144h	DMAnSIRQ_DMA3	4124h	DMAnSPTRU_DMA2	4104h	DMAnDPTRH_DMA1
41E3h	IOCEF_M1	41C3h	-	41A3h	-	4183h	DMAnSSAL_DMA6	4163h	DMAnDSAL_DMA5	4143h	DMAnAIRQ_DMA3	4123h	DMAnSPTRH_DMA2	4103h	DMAnDPTRL_DMA1
41E2h	IOCCF_M1	41C2h	-	41A2h	-	4182h	DMAnSSZH_DMA6	4162h	DMAnDSZH_DMA5	4142h	DMAnCON1_DMA3	4122h	DMAnSPTRL_DMA2	4102h	DMAnDCNTH_DMA1
41E1h	IOCBF_M1	41C1h	-	41A1h	-	4181h	DMAnSSZL_DMA6	4161h	DMAnDSZL_DMA5	4141h	DMAnCON0_DMA3	4121h	DMAnSCNTH_DMA2	4101h	DMAnDCNTL_DMA1
41E0h	IOCAF_M1	41C0h	-	41A0h	-	4180h	DMAnSPTRU_DMA6	4160h	DMAnDPTRH_DMA5	4140h	DMAnSSAU_DMA3	4120h	DMAnSCNTL_DMA2	4100h	DMAnBUF_DMA1

2. Setting the Destination

The DMA peripheral can store data into only one memory system in the microcontroller; SFR/GPR memory. As a result, there is not a corresponding set of destination memory select bits.

2.1 Configuring Destination Registers

To configure the DMA destination, the only step is to load the DMAnDSAL and DMAnDSAH registers. These registers specify the address of the data destination. Together, the two registers can hold up to a 16-bit address.

2.2 Loading Destination Registers

Loading the destination registers can be accomplished by using the set address routines supplied by MCC. They can be loaded individually by the system software, or both registers can be loaded together by equating the DMAnDSA register with a 16 constant or a variable containing the desired address.

```
DMAnSSA = &U4TXB; // destination USART 4 TX reg.
```

Note: The & prefix is used with the UxTXB label to specify the address of the SFR, rather than the SFR’s contents.

3. Setting the Size of Transfer

A DMA transaction is the transfer of a single byte. A DMA message consists of one or more transactions. A complete DMA process consists of one or more messages. The source and destination size registers determine the size of the DMA message and the number of DMA messages in the DMA process. The smaller of the source/destination sizes determines the size of the DMA message and the larger value, divided by the smaller size, determines the number of messages in the DMA process.

The source size is loaded into the DMAnSSZ registers. The destination is loaded into the DMAnDSZ registers. Both registers are 16-bit and located such that a 16-bit assignment will load both registers correctly, allowing either individual loading or equating the variable to a 16-bit value of variable.

Note: The size of the source and destination are typically different. For examples, see Table 3-1.

Once a transfer has been completed;

1. The counter registers are decremented
2. The address registers are updated (post increment, post decrement, fixed)
3. And the data transfers will continue until either or both of the DMA counters (DMAnSCNTL/H or DMAnDCNTL/H) roll over. When the counters roll over, they are reloaded with the value from the DMAnSSZL/H or DMAnDSZL/H registers and the event ends the current DMA message.

Note: If one of the size registers is programmed with 1, then only one transfer will occur for each trigger event. If the smallest size register is N, then N transactions will occur for each DMA trigger.

Table 3-1. Example Message Size

Operation	Example	SCNT	DCNT	Comments
Read from single SFR location to RAM	UART Receive Buffer	1	N	N equals the number of bytes desired in the destination buffer. N ≥ 1.
Write to single SFR location from RAM	UART Transmit Buffer	N	1	N equals the number of bytes desired in the source buffer. N ≥ 1.
Read from multiple SFR location	ADC Result registers	2	2*N	N equals the number of ADC results to be stored in memory. N ≥ 1.
Write to Multiple SFR registers	PWM Duty Cycle registers	2*N	2	N equals the number of PWM duty cycle values to be loaded from a memory table. N ≥ 1.

4. Selecting the Transfer Trigger

The next step in configuring the DMA peripheral is to select the transfer trigger event. The value loaded into the DMAnSIRQ register selects the hardware event that will trigger the DMA. Potential triggers can include CLCs, ADC conversions, communications peripherals, timers and even other DMA channels. Table 4-1 lists an example of all the possible DMA triggers for the PIC18F-Q43.

Once the DMAnSIRQ register is configured, the SIRQEN bit in the DMAnCON0 register must be set to enable the selected hardware trigger.

If a 0x00 is loaded into the DMAnSIRQ register, or the SIRQEN bit is cleared, then the only method for triggering a DMA transfer is to manually trigger a transfer by setting the DGO bit in the DMAnCON0 register.

Table 4-1. DMAnSIRQ Interrupt Sources

DMAnSIRQ	DMA Start/Abort Trigger source
0x0	-
0x1	HLVD (High/Low-Voltage Detect)
0x2	OSF (Oscillator Fail)
0x3	CSW (Clock Switching)
0x4	-
0x5	CLC1 (Configurable Logic Cell)
0x6	-
0x7	IOC (Interrupt-on-Change)
0x8	INT0
0x9	ZCD (Zero-Cross Detection)
0xA	AD (ADC Conversion Complete)
0xB	ACT (ADC Auto-Conversion Trigger)
0xC	CM1 (Comparator)
0xD	SMT1 (Signal Measurement Timer)
0xE	SMT1PRA
0xF	SMT1PWA
0x10	ADT
0x11 - 0x13	-
0x14	DMA1SCNT (Direct Memory Access)
0x15	DMA1DCNT
0x16	DMA1OR
0x17	DMA1A
0x18	SPI1RX (Serial Peripheral Interface)
0x19	SPI1TX
0x1A	SPI1
0x1B	TMR2
0x1C	TMR1

...........continued

DMAnSIRQ	DMA Start/Abort Trigger source
0x1D	TMR1G
0x1E	CCP1 (Capture/Compare/PWM)
0x1F	TMR0
0x20	U1RX
0x21	U1TX
0x22	U1E
0x23	U1
0x24 - 0x25	-
0x26	PWM1RINT
0x27	PWM1GINT
0x28	SPI2RX
0x29	SPI2TX
0x2A	SPI2
0x2B	-
0x2C	TMR3
0x2D	TMR3G
0x2E	PWM2RINT
0x2F	PWM2GINT
0x30	INT1
0x31	CLC2
0x32	CWG1 (Complementary Waveform Generator)
0x33	NCO1 (Numerically Controlled Oscillator)
0x34	DMA2SCNT
0x35	DMA2DCNT
0x36	DMA2OR
0x37	DMA2A
0x38	I2C1RX
0x39	I2C1TX
0x3A	I2C1
0x3B	I2C1E
0x3C	-
0x3D	CLC3
0x3E	PWM3RINT
0x3F	PWM3GINT
0x40	U2RX

...........continued

DMAnSIRQ	DMA Start/Abort Trigger source
0x41	U2TX
0x42	U2E
0x43	U2
0x44	TMR5
0x45	TMR5G
0x46	CCP2
0x47	SCAN
0x48	U3RX
0x49	U3TX
0x4A	U3E
0x4B	U3
0x4C	-
0x4D	CLC4
0x4E - 0x4F	-
0x50	INT2
0x51	CLC5
0x52	CWG2
0x53	NCO2
0x54	DMA3SCNT
0x55	DMA3DCNT
0x56	DMA3OR
0x57	DMA3A
0x58	CCP3
0x59	CLC6
0x5A	CWG3
0x5B	TMR4
0x5C	DMA4SCNT
0x5D	DMA4DCNT
0x5E	DMA4OR
0x5F	DMA4A
0x60	U4RX
0x61	U4TX
0x62	U4E
0x63	U4
0x64	DMA5SCNT

...........continued

DMAnSIRQ	DMA Start/Abort Trigger source
0x65	DMA5DCNT
0x66	DMA5OR
0x67	DMA5A
0x68	U5RX
0x69	U5TX
0x6A	U5E
0x6B	U5
0x6C	DMA6SCNT
0x6D	DMA6DCNT
0x6E	DMA6OR
0x6F	DMA6A
0x70	-
0x71	CLC7
0x72	CM2
0x73	NCO3
0x74 - 0x77	-
0x78	NVM
0x79	CLC8
0x7A	CRC
0x7B	TMR6
0x7C - 0x8F	-
0x90	PWM1 S1P1 (PWM1 Parameter 1 of Slice 1)
0x91	PWM1 S1P2 (PWM1 Parameter 2 of Slice 1)
0x92	PWM2 S1P1 (PWM2 Parameter 1 of Slice 1)
0x93	PWM2 S1P2 (PWM2 Parameter 2 of Slice 1)
0x94	PWM3 S1P1 (PWM3 Parameter 1 of Slice 1)
0x95	PWM3 S1P2 (PWM3 Parameter 2 of Slice 1)

5. Selecting an Abort Trigger

In addition to the trigger select, there is an abort select that can immediately terminate a DMA in response to a hardware trigger. DMAnAIRQ is configured with a selected hardware abort trigger and when the abort is detected, the DGO bit and the SIRQEN bits are both cleared. This happens immediately and any byte in transit is not stored to the designation. The options for an abort select signal is typically the same as the selections for a trigger signal. See Table 4-1 for abort event options.

Important: In addition to selecting an abort event, the AIRQEN bit must also be set to enable the abort event. If cleared, no trigger will abort a DMA message.

6. Configuring DMA Automation

Now that the DMA is mostly configured, options for the automation of the DMA must be selected. These include Post Increment/Decrement/Fixed and the termination event for the DMA transfers.

The next step in the configuration process is to set the Automation mode for both the source and destination. Both can be individually configured for post increment, post decrement or fixed operation. The source is configured using the SMODE bits and the destination is configured using the DMODE bits. Both sets of bits are in the DMAnCON1 register. See Tables 6-1 and 6-2 for selecting the mode.

Table 6-1. Source Operating Mode

SMODE1	SMODE0	Mode
1	1	Reserved
1	0	Post Decrement
0	1	Post Increment
0	0	Fixed (no Change)

Table 6-2. Destination Operating Mode

DMODE1	DMODE0	Mode
1	1	Reserved
1	0	Post Decrement
0	1	Post Increment
0	0	Fixed (no Change)

To select the source counter roll over to terminate the DMA process, the SSTP bit in the DMAnCON1 register is set. To select the destination counter roll over, the DSTP bit in the DMAnCON1 register must be set. If neither bit is set, then the DMA process can only be terminated by clearing the DGO bit in the DMAnCON0 register.

Note:

1. If a post decrement operating mode is chosen, then the associated Source/Destination Address register must be loaded with the last address of the variable space/peripheral registers, rather than the first.
2. If a fixed operating mode is chosen and the associated size register is set to 0x01, then a single transfer will terminate the DMA message. If a fixed operating mode is chosen and the associated size register is set to N (N > 0x01), then a N transfers will terminate the DMA message.

The second automation selection concerns which of the transaction counter roll over events will terminate the DMA operation. Both the source and destination have been configured for size. This selection determines which, if either, roll over will end the DMA process.

For example, if a USART is configured to receive and the DMA is configured to capture the received data and store it into a buffer, then the terminating roll over is typically the buffer’s counter. When that happens the DGO bit, the SIRQEN bit and the AIRWEN bit are all cleared. The hardware trigger enable SIRQEN is also cleared, preventing additional triggers by a hardware source.

7. Enabling Interrupts

The DMA peripheral can generate four interrupts; Source Count, Destination Count, Abort and Overrun. The source counter interrupt generates a system interrupt when the source counter reloads. The destination counter interrupt does the same thing when the destination counter reloads. Together, these two interrupts convey the completion of a DMA message and the full DMA process. The determination of which interrupt signifies which event is dependent upon which counter is reloaded from the larger size value.

The abort interrupt is generated whenever a hardware abort signal is received by the DMA peripheral. The overrun interrupt is generated whenever the DMA peripheral receives a second trigger event while still processing the first DMA.

These interrupts not only provide feedback to the controlling software, they also provide CIP signals that can be routed to other peripherals. An example would be using the DMA to reconfigure the ADC and then using the destination counter reload interrupt to trigger an ADC conversion.

8. Setting Transfer Priority and Enabling Transfers

PIC18F microcontrollers, which contain the DMA peripheral, also contain a System Arbiter function. The System Arbiter controls the relative priority of the DMA peripherals, the interrupts and normal code execution. It determines which function can suspend the other operations to take control of the address and data buses.

For example, the interrupts typically have the highest priority, followed by main code operation and then the DMA peripherals. This means that main code has control of the buses until an interrupt occurs. Then the interrupt has control, until the interrupt completes. Because the DMA has such a low priority, it can only take control of the buses when the main code is executing a branch or jump and the buses are idle. However, using the arbiter, it is possible to increase the priority of the DMA, allowing it to interrupt both the main code and the interrupt functions and take over control of the buses, until it completes its message.

In fact, using the arbiter, it is even possible to put the interrupt at a lower priority than the main code. However, as this would effectively disable the interrupt system, there are provisions within the arbiter that prevent this specific scenario.

Note: Because each device has a different number of DMA peripherals and different modes of interrupt operation, it is impractical to specify all the possible priority combinations using the System Arbiter in this document. Therefore, the reader is referred to the **System Arbiter** chapter in the desired device data sheet for the recommended configuration and operation of the System Arbiter.

To begin operation for a given DMA peripheral, the programmer will need to lock the peripheral using the PRLOCK sequence and set the EN bit in the appropriate DMAnCON0 register to enable DMA operation. If the transfer is to be initiated by hardware, the DMA peripheral will sit dormant until the hardware trigger is detected. If the DMA peripheral is configured for a software trigger, then set the DGO bit in the DMAnCON0 register to initiate the first DMA transfer. See Examples 8-1 and 8-2 for locking and unlocking sequences.

Example 8-1. Priority Lock Sequence

```
; Disable interrupts
BCF   INTCON0,GIE

; Bank to PRLOCK register
BANKSEL   PRLOCK
MOVLW     55h

; Required sequence, next 4 instructions
MOVWF   PRLOCK
MOVLW   AAh
MOVWF   PRLOCK
; Set PRLOCKED bit to grant memory access to peripherals
BSF   PRLOCK,0

; Enable Interrupts
BSF   INTCON0,GIE
```

Example 8-2. Priority Unlock Sequence

```
; Disable interrupts
 BCF   INTCON0,GIE

; Bank to PRLOCK register
BANKSEL   PRLOCK
MOVLW     55h

; Required sequence, next 4 instructions
MOVWF   PRLOCK
MOVLW   AAh
MOVWF   PRLOCK
; Clear PRLOCKED bit to allow changing priority settings
BCF   PRLOCK,0
```

```
; Enable Interrupts
BSF   INTCON0,GIE
```

Note:

1. The PRLOCK sequence is a requirement to start the DMA peripheral. This is an added layer of security to avoid priority mismatch errors.
2. Both hardware and software triggers can be used in the same DMA operation. This allows the trigger of DMA from hardware, while also initiating the first DMA message from software.

9. Monitor Transfer Status

The DMA peripheral has two Status bits, the DGO and XIP bits in the DMAnCON0 register. The DGO bit indicates that the DMA peripheral has been triggered and is currently in the process of transferring data. The XIP bit indicates that a transfer is occurring and the data being transferred is currently stored in the DMAxBUF register. This indicates that the memory read has occurred, but the memory write is pending.

In addition, the counter registers, DMAnSCNT and DMAnDCNT can be monitored to gauge the progress of the DMA operation. Each time the smaller counter register is reloaded, it will indicate the completion of a DMA message, and the reload the larger counter will indicate the completion of the DMA process. To gauge the overall progress of the DMA, the DMA address registers (DMAnSPTR and DMAnDPTR) can also be read to determine the last address read from and written to.

The Microchip Website

Microchip provides online support via our website at http://www.microchip.com/. This website is used to make files and information easily available to customers. Some of the content available includes:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQs), technical support requests, online discussion groups, Microchip design partner program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Product Change Notification Service

Microchip's product change notification service helps keep customers current on Microchip products. Subscribers will receive email notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, go to http://www.microchip.com/pcn and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Embedded Solutions Engineer (ESE)
- Technical Support

Customers should contact their distributor, representative or ESE for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in this document.

Technical support is available through the website at: http://www.microchip.com/support

Microchip Devices Code Protection Feature

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.

Legal Notice

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 unless otherwise stated.

Trademarks

ISBN: 978-1-5224-5602-5

Quality Management System

For information regarding Microchip's Quality Management Systems, please visit http://www.microchip.com/quality.

MICROCHIP

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:
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

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
Tel: 317-536-2380

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608
Tel: 951-273-7800

Raleigh, NC
Tel: 919-844-7510

New York, NY
Tel: 631-435-6000

San Jose, CA
Tel: 408-735-9110
Tel: 408-436-4270

Canada - Toronto
Tel: 905-695-1980
Fax: 905-695-2078

ASIA/PACIFIC

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

China - Beijing
Tel: 86-10-8569-7000

China - Chengdu
Tel: 86-28-8665-5511

China - Chongqing
Tel: 86-23-8980-9588

China - Dongguan
Tel: 86-769-8702-9880

China - Guangzhou
Tel: 86-20-8755-8029

China - Hangzhou
Tel: 86-571-8792-8115

China - Hong Kong SAR
Tel: 852-2943-5100

China - Nanjing
Tel: 86-25-8473-2460

China - Qingdao
Tel: 86-532-8502-7355

China - Shanghai
Tel: 86-21-3326-8000

China - Shenyang
Tel: 86-24-2334-2829

China - Shenzhen
Tel: 86-755-8864-2200

China - Suzhou
Tel: 86-186-6233-1526

China - Wuhan
Tel: 86-27-5980-5300

China - Xian
Tel: 86-29-8833-7252

China - Xiamen
Tel: 86-592-2388138

China - Zhuhai
Tel: 86-756-3210040

ASIA/PACIFIC

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

India - New Delhi
Tel: 91-11-4160-8631

India - Pune
Tel: 91-20-4121-0141

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

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

Korea - Daegu
Tel: 82-53-744-4301

Korea - Seoul
Tel: 82-2-554-7200

Malaysia - Kuala Lumpur
Tel: 60-3-7651-7906

Malaysia - Penang
Tel: 60-4-227-8870

Philippines - Manila
Tel: 63-2-634-9065

Singapore
Tel: 65-6334-8870

Taiwan - Hsin Chu
Tel: 886-3-577-8366

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

Taiwan - Taipei
Tel: 886-2-2508-8600

Thailand - Bangkok
Tel: 66-2-694-1351

Vietnam - Ho Chi Minh
Tel: 84-28-5448-2100

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

Finland - Espoo
Tel: 358-9-4520-820

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Garching
Tel: 49-8931-9700

Germany - Haan
Tel: 49-2129-3766400

Germany - Heilbronn
Tel: 49-7131-72400

Germany - Karlsruhe
Tel: 49-721-625370

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

Germany - Rosenheim
Tel: 49-8031-354-560

Israel - Ra’anana
Tel: 972-9-744-7705

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

Italy - Padova
Tel: 39-049-7625286

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

Norway - Trondheim
Tel: 47-72884388

Poland - Warsaw
Tel: 48-22-3325737

Romania - Bucharest
Tel: 40-21-407-87-50

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

Sweden - Gothenberg
Tel: 46-31-704-60-40

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

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