TB3164

Direct Memory Access on 8-bit PIC® Microcontrollers

Introduction

Authors: Mark Pallones and Ashish Makthal, Microchip Technology Inc.

In data-oriented applications, transferring data between peripherals or different memory regions without CPU intervention provides significant improvement in terms of latency and throughput.

The Direct Memory Access (DMA) controller peripheral in Microchip's 8-bit microcontroller can provide this improvement, allowing the CPU to spend time on other tasks rather than waiting for register flags or handling interrupts related to data movement. This technical brief provides an overview of the DMA controller's features. Code examples are also provided to show how to setup the module in different modes of operation. Figure 1 shows the functional block diagram for the DMA module.

Figure 1. DMA FUNCTIONAL BLOCK DIAGRAM

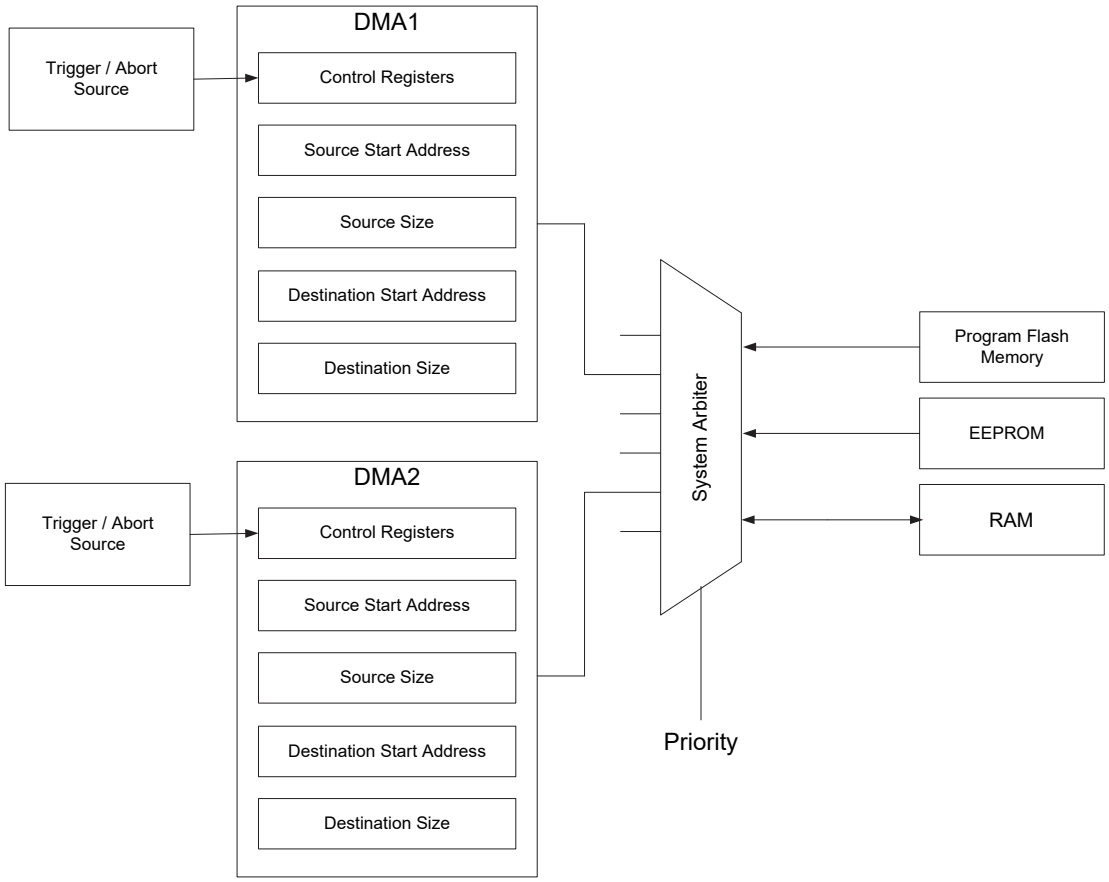

Table of Contents

1. System Operation

As shown in Figure 1, each DMA controller can be independently configured to move data between single or multiple addresses. The DMA controllers use the same instruction bus and data bus as the CPU for transferring data between memories. Depending on the priority set in the system arbiter, the DMA controller can move data either by utilizing unused CPU cycles or stalling the CPU. By default, the CPU has priority over DMA. In this case, the DMA steals unused cycles from the CPU to perform the read/write operations. Because of this concurrent operation between the CPU and DMA, the bandwidth of handling data is increased and the DMA module can operate in the background.

2. DMA Data Movement

2.1 Transaction

A DMA transaction refers to a byte of data movement from source address to destination address. It occurs as a two-step process: a read from the source address and storing the value in DMAxBUF register, then followed by writing the contents of the DMAxBUF register to the destination address. The timing of these read/write operations is dependent on the priority settings of the system arbiter.

2.2 Message

A DMA message consists of one or more transactions. The size of the DMA message is determined by the value programmed in the Source Message Size (DMAxSSZ) and Destination Message Size (DMAxDSZ) registers. The source and destination sizes can be different, but must be a multiple of each other for correct operation. The number of bytes transferred by the DMA module is determined by the largest of the two sizes.

For example, if the DMAxSSZ is 2 and DMAxDSZ is 6, then each message will consist of two transactions, and the complete DMA process will consist of three messages, which means a total of six bytes will be transferred. Table 2-1 shows examples of message size configuration values based on various DMA operations.

Table 2-1. EXAMPLE OF MESSAGE SIZE CONFIGURATION

DMA Operation	DMAxSSZ	DMAxDSZ	Comment
Reading the values from UART RX register and storing the values in a memory buffer	1	N	N equals the number of bytes to be stored in the destination buffer
Reading ADC results registers and storing the values in a memory buffer	2	2*N	N equals the number of results to be stored in the memory buffer
Loading PWM duty cycle values from a memory table	2*N	2	N equals the number of duty cycle values to be loaded from the memory table
Transmitting data bytes from a memory table through UART	N	1	N equals the number of bytes to be transmitted from the memory table

3. DMA Addressing

3.1 Memory Access

The DMA controller has read/write access of the data space (RAM) and read-only access of the Flash memory and EEPROM.

The start addresses for the source and destination can be configured using the Source Start Address (DMAxSSA) and Destination Start Address (DMAxDSA) registers, respectively. The Source Memory Region selection bits (SMR) in the DMAxCON1 register can be used to choose which memory region is being addressed by the DMAxSSA register. There are no bits to set the destination memory region since the DMA controller can only write to RAM.

3.2 Addressing modes

When a DMA transaction is in progress, the Source Pointer register (DMAxSPTR) and Destination Pointer register (DMAxDPTR) point to the locations the DMA module is currently addressing. Using the SMODE and DMODE bits in the DMAxCON1 register, the source and destination address pointers can be incremented, decremented or remain unchanged after every byte is transferred. This provides flexibility in structuring the data array.

4. Starting a DMA Transaction

4.1 Software Start

When the DMA module is enabled and all setup is complete, the DMA transfer can be initiated by setting the DGO bit in the DMAxCON0 register.

Based on the system arbiter priority settings, when a CPU cycle is granted, the DMA module will read a byte of data from the source address and transfer it to the DMAxBUF register. The XIP bit in the DMAxCON0 register is also set, indicating that a transfer is in progress. When the DMA module gets another CPU cycle grant, the data is moved to the destination address and the XIP bit is cleared.

4.2 Hardware Triggered Start

DMA transfers can also be started using triggers from available hardware trigger sources. The trigger can be configured using the Start Interrupt Request Source register (DMAxSIRQ). The list of available hardware triggers will vary based on the peripheral set offered by the device. Please refer to the device data sheet for the complete list of triggers.

For example, a DMA configured to move data out of a UART can use its own Transmit Interrupt flag (UxTXIF) as a trigger source. To initiate transfers based on the UART interrupts, the DMAxSIRQ register needs to be configured for the UART transmit interrupt and the Start-of-Transfer Interrupt Request bit (SIRQEN) of the DMAxCON0 register should be enabled. Setting the SIRQEN bit will arm the DMA module and will start the transfer, when the first interrupt trigger is received. This will also set the DGO bit to indicate that the transfer has been initiated. Every time the transmit buffer is empty, the DMA will be triggered and a new byte will be loaded into the UART transmit buffer. The SIRQEN bit can be automatically cleared using the DSTP and SSTP bits in the DMAxCON0 register. These bits decide if SIRQEN is cleared/not cleared when the destination/source counters reload.

4.3 Counter Reload

Once the DMA transfer is initiated, the corresponding value initialized in the DMAxSSZ and DMAxDSZ registers is loaded into the Source Count register (DMAxSCNT) and Destination Count register (DMAxDCNT). After every transaction, the DMAxSCNT and DMAxDCNT registers are decremented, thus indicating the number of bytes that are left in the current DMA message.

When the DMAxSCNT and DMAxDCNT registers equal to "1", they are reloaded with the value from the DMAxSSZ and DMAxDSZ registers, respectively. They are not dependent upon each other. If the source and destination sizes are different, they will reload at different times. The DMA operation can be stopped when either of these counters is reloaded, using the Source Counter Reload Stop bit (SSTP) and the Destination Counter Reload Stop bit (DSTP).

For example: Consider a case of transmitting 10 bytes of data from a buffer in general purpose RAM to the UART transmit buffer. In this case, the source size DMAxSSZ is 10 and the destination size DMAxDSZ is 1. If the DGO bit is set in software to start the DMA transfer, the first byte is moved to the UART transmit buffer and then the destination counter (DMAxDCNT) runs out. This will trigger a reload of the destination counter (DMAxDCNT), but the source counter (DMAxSCNT) will stay at 9. If the hardware trigger is not used, the user will need to set the DGO bit for every byte transfer. If SSTP = `1` (clear the trigger and DGO bit when source counter reloads) and the UART TX buffer empty interrupt is used as a

trigger to the DMA and DSTP = 0; the DMA will transfer all 10 bytes and then stop. Refer to Example-2 for more details regarding the setup of this operation.

5. Stopping a DMA Transaction

5.1 Normal Completion

The number of transactions in a DMA message is based on the setting of the Source and Destination Size registers and SSTP/DSTP bits. The DGO bit is cleared when the message transfer is complete. Refer to table 5-1 below for details.

Table 5-1. DMA OPERATION

Configuration	SSTP:DSTP = 00	SSTP:DSTP = 01	SSTP:DSTP = 10	SSTP:DSTP = 11
Source Size = Destination Size	DGO bit is never cleared. DMA operates until user firmware clears DGO bit or Abort trigger is received.	DGO bit is cleared when DSZ or SSZ transactions are completed and DCNT or SCNT are reloaded respectively		
Source Size != Destination Size		DGO bit is cleared when DSZ transactions are completed and DCNT is reloaded. Set DGO bit again for next DSZ transactions.	DGO bit is cleared when SSZ transactions are completed and SCNT is reloaded. Set DGO bit again for next SSZ transactions.	DGO bit is cleared when DSZ or SSZ transactions are completed and DCNT or SCNT are reloaded respectively.

5.2 Soft Stop/Pause

If the user firmware clears the DGO bit, the transfer will halt and the DMA will remain in the current configuration. If the DMA module was in between a byte read and write cycle, the data will be retained in the DMAxBUF register and will not be written to the destination address. The DMA module can be resumed by a Software/Hardware triggered start. The transfer will resume from where it was paused. The user does not need to reconfigure the DMA module before resuming the transfer.

5.3 Hardware Triggered Stop

The DMA message transfer can also be stopped using hardware trigger sources. These abort trigger sources can be selected through the Abort Interrupt Request Source register (DMAxAIRQ). Setting the Abort Transfer Interrupt Request Enable bit (AIRQEN) of the DMAxCON0 register enables the selected abort source trigger. Once the trigger is received, the DMA will perform a soft-stop by automatically clearing the DGO bit. The DMA will also clear the SIRQEN and AIRQEN bits. The DMA state information does not change in the event of an abort.

5.4 Hard Stop

The DMA message transfer can also be stopped by clearing the EN bit of the DMAxCON0 register. The DMA module returns to its default configuration. This is referred to as a hard-stop, as the DMA transfer cannot resume without reconfiguration.

6. Interrupts

Source and Destination Count Interrupt:

The Source Count Interrupt Flag (DMAxSCNTIF) and Destination Count Interrupt Flag (DMAxDCNTIF) are set when the corresponding count registers (DMAxSCNT and DMAxDCNT) are reloaded. This signifies that the message transfer is complete.

Abort Interrupt:

The Abort Interrupt Flag bit (DMAxAIF) is set when an abort trigger is received and the AIRQEN bit is set.

Overrun Interrupt:

The Overrun Interrupt Flag bit (DMAxORIF) is set when a new hardware trigger is received before the previous transaction is completed. This overrun condition does not affect the DMA operation, but is used to indicate that the DMA module may not be able to keep up with the DMA requests.

7. Code Examples

Code examples for configuring the DMA controller can be found in the examples referenced below. The examples show how to configure the DMA module to work with software/hardware triggers.

Example-1 shows the setup of the DMA module to transfer 512 bytes of data from the EEPROM to user RAM. This can be an application where a look-up table is stored in the EEPROM and can be made available in the RAM for faster access at run time.

Example-2 shows the setup of the DMA module to transfer 20 bytes of data between the user RAM and the UART transmit buffer. In this example, the UART transmit interrupt is used as a trigger for DMA operation.

Example-3 shows the setup of the DMA module to transfer data from the flash memory to PWM duty cycle registers, but an abort trigger is set up for an event on an I/O pin.

Notice: These examples are written for the PIC18F47K42. Minor edits may be required to make them compatible for other devices that have the DMA module.

Tip: Refer to the examples on the MPLAB Xpress Code Examples webpage for a working application code.

7.1 Example-1

This example shows the code snippet for the setup of the DMA module to transfer data from the EEPROM to a buffer in the user RAM space. The transfer is initiated by setting the DGO bit and has no abort triggers. In this example, 512 bytes of data that has been preloaded into the EEPROM at program time is moved into a RAM buffer called “DMA_READ_BUF” for faster access. User firmware needs to set the DGO bit in the DMA1CON0 register to initiate the DMA message. Once completed, the DMA1DCNTIF bit is set and since the interrupt for destination counter reload is enabled, the program will jump to the ISR.

```
void DMA1_Initialize(void)
  {
    DMA1SSA = 0x000000;       //set source start address
    DMA1DSA = &DMA_READ_BUF;  //set destination start address
    DMA1CON1 = 0x7A;          //DMODE = 01 | DSTP = 1 | SMR = 11 | SMODE = 01 | SSTP = 0
    DMA1SSZ = 0x0200;         //set source size = 512 bytes
    DMA1DSZ = 0x0200;         //set destination size = 512 bytes
    DMA1SIRQ = 0x00;          //set DMA Transfer Trigger Source
    DMA1AIRQ = 0x00;          //set DMA Transfer abort Source

    PIR2bits.DMA1DCNTIF = 0;  //clear Destination Count Interrupt Flag bit
    PIR2bits.DMA1SCNTIF = 0;  //clear Source Count Interrupt Flag bit
    PIR2bits.DMA1AIF = 0;     //clear abort Interrupt Flag bit
    PIR2bits.DMA1ORIF = 0;    //clear overrun Interrupt Flag bit

    PIE2bits.DMA1DCNTIE = 1;  //enable Destination Count 0 Interrupt
    PIE2bits.DMA1SCNTIE = 0;  //disable Source Count Interrupt
    PIE2bits.DMA1AIE = 0;     //disable abort Interrupt
    PIE2bits.DMA1ORIE = 0;    //disable overrun Interrupt

    asm("BCF INTCON0,7");     //disable Global Interrupts

    asm ("BANKSEL PRLOCK");   //
    asm ("MOVLW 0x55");       //
    asm ("MOVWF PRLOCK");     // Arbiter Priority lock
    asm ("MOVLW 0xAA");       // sequence
    asm ("MOVWF PRLOCK");     //
    asm ("BSF PRLOCK, 0");    //

    asm("BSF INTCON0,7");     // enable Global Interrupts

    DMA1CON0 = 0x80;          //EN = 1 | SIRQEN = 0 | DGO = 0 |xx| AIRQEN = 0 |x| XIP = 0
  }
```

7.2 Example-2

This example shows the code snippet for the setup of the DMA module to transfer data from the user RAM to the UART Transmit buffer. The transfer is triggered by the UART Transmit Interrupt. The data is stored in an array called “TX_DATA” and everytime a byte is sent out from the UART Transmit buffer, the DMA is triggered to load a new byte. User firmware **does not** need to set the DGO bit in the DMA1CON0 register. Setting the SSTP bit will clear the SIRQEN bit once all the data has been transferred. User firmware can set the SIRQEN bit to arm the DMA module again.

```
void DMA1_Initialize(void)
 {
     DMA1SSA = &TX_DATA;         //set source start address
     DMA1DSA = &U1TXB;           //set destination start address
     DMA1CON1 = 0x03;            //DMODE = 00 | DSTP = 0 | SMR = 01 | SMODE = 01 | SSTP = 1
     DMA1SSZ = 0x0017;           //set source size = 23 bytes
     DMA1DSZ = 0x0001;           //set destination size = 1 byte
     DMA1SIRQ = 0x1C;            //set DMA Transfer Trigger Source = U1TX
     DMA1AIRQ = 0x00;            //set DMA Transfer abort Source
```

```
    PIR2bits.DMA1DCNTIF = 0;    //clear Destination Count Interrupt Flag bit
    PIR2bits.DMA1SCNTIF = 0;    //clear Source Count Interrupt Flag bit
    PIR2bits.DMA1AIF = 0;       //clear abort Interrupt Flag bit
    PIR2bits.DMA1ORIF = 0;      //clear overrun Interrupt Flag bit
    PIE2bits.DMA1DCNTIE = 0;    //disable Destination Count Interrupt
    PIE2bits.DMA1SCNTIE = 0;    //disable Source Count Interrupt
    PIE2bits.DMA1AIE = 0;       //disable abort Interrupt
    PIE2bits.DMA1ORIE = 0;      //disable overrun Interrupt

    asm("BCF INTCON0,7");       //disable Global Interrupts

    asm ("BANKSEL PRLOCK");     //
    asm ("MOVLW 0x55");         //
    asm ("MOVWF PRLOCK");       //Arbiter Priority lock
    asm ("MOVLW 0xAA");         //sequence
    asm ("MOVWF PRLOCK");       //
    asm ("BSF PRLOCK, 0");      //

    asm("BSF INTCON0,7");       //enable Global Interrupts
    DMA1CON0 = 0x80;            //EN = 1 | SIRQEN = 0 | DGO = 0 |xx| AIRQEN = 0 |x| XIP = 0
 }
```

User firmware can set the SIRQEN bit whenever the DMA transaction needs to start. This can be done with the following line of code.

```
DMA1CON0bits.SIRQEN = 1;
```

7.3 Example-3

This example shows the code snippet for the setup of the DMA module to transfer data from a look-up table stored in the Program Flash Memory into the PWM duty cycle register. Each byte transfer is triggered by the roll-over of Timer0. The code also sets up an abort trigger when an Interrupt-on-Change (IOC) is detected. The data is stored in an array called “PWM_DATA” and each time the Timer0 rolls over a byte is transferred from the array into the PWM duty cycle register. User firmware **does not** need to set the DGO bit in the DMA1CON0 register as the timer overflow will automatically set it. Since the SSTP and DSTP bits are cleared, the DMA module will continue to operate until an abort trigger is detected. When an abort trigger is detected, the DGO, AIRQEN and SIRQEN bits are cleared. User firmware has to set the SIRQEN and AIRQEN to re-enable the DMA module.

```
void DMA1_Initialize(void)
 {
    DMA1SSA = &PWM_DATA;          //set source start address
    DMA1DSA = &PWM5DCH;           //set destination start address
    DMA1CON1 = 0x0A;              //DMODE = 00 | DSTP = 0 | SMR = 01 | SMODE = 01 | SSTP = 1
    DMA1SSZ = 0x00AC;             //set source size = 172 bytes
    DMA1DSZ = 0x0001;             //set destination size
    DMA1SIRQ = 0x1F;              //set DMA Transfer Trigger Source = TMR0
    DMA1AIRQ = 0x07;              //set DMA Transfer abort Source = IOC

    PIR2bits.DMA1DCNTIF = 0;      //clear Destination Count Interrupt Flag bit
    PIR2bits.DMA1SCNTIF = 0;      //clear Source Count Interrupt Flag bit
    PIR2bits.DMA1AIF = 0;         //clear abort Interrupt Flag bit
    PIR2bits.DMA1ORIF = 0;        //clear overrun Interrupt Flag bit

    PIE2bits.DMA1DCNTIE = 0;      //disable Destination Count 0 Interrupt
    PIE2bits.DMA1SCNTIE = 0;      //disable Source Count Interrupt
    PIE2bits.DMA1AIE = 1;         //enable abort Interrupt
    PIE2bits.DMA1ORIE = 0;        // isable overrun Interrupt

    asm("BCF INTCON0,7");         //disable Global Interrupts

    asm ("BANKSEL PRLOCK");       //
    asm ("MOVLW 0x55");           //
    asm ("MOVWF PRLOCK");         //Arbiter Priority lock
    asm ("MOVLW 0xAA");           //sequence
```

```
    asm ("MOVWF PRLOCK");          //
    asm ("BSF PRLOCK, 0");         //

    asm("BSF INTCON0,7");         //enable Global Interrupts

    DMA1CON0 = 0x80;               //EN = 1 | SIRQEN = 0 | DGO = 0 |xx| AIRQEN = 0 |x| XIP = 0
}
```

User firmware can set the SIRQEN bit whenever the DMA transaction needs to start. This can be done by the following line of code.

```
DMA1CON0bits.SIRQEN = 1;
```

The following lines of code need to be used to restart the DMA module and it will continue to run until an abort trigger is received.

```
DMA1CON0bits.SIRQEN = 1;
DMA1CON0bits.AIRQEN = 1;
```

8. Conclusion

This technical brief provides a brief overview of the DMA module and described the basic modes of operation. The sample code snippets should serve as a walk through on getting started in building an application that uses the DMA module. For more information about this module, refer to the device data sheet.

The Microchip Web Site

Microchip provides online support via our web site at http://www.microchip.com/. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **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 (FAQ), technical support requests, online discussion groups, Microchip consultant 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

Customer Change Notification Service

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

To register, access the Microchip web site at http://www.microchip.com/. Under "Support", click on "Customer Change Notification" and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or Field Application Engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site 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

Trademarks

ISBN: 978-1-5224-3539-6

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	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
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 **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	**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	**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	**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: 49-7131-67-3636 **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