

## dsPIC30F2010 Rev. A0 Silicon Errata

### dsPIC30F2010 (Rev. A0) Silicon Errata

The dsPIC30F2010 (Rev. A0) samples you have received conform to the specifications and functionality described in the following documents:

- DS70030 – dsPIC30F Programmer's Reference Manual
- DS70118 – dsPIC30F2010 Data Sheet
- DS70046 – dsPIC30F Family Reference Manual

The exceptions to the specifications in the documents listed above are described in this section.

dsPIC30F2010 Rev A0 silicon is identified by performing a "Reset and Connect" operation to the device using MPLAB® ICD 2 within the MPLAB IDE. The following text is visible under the MPLAB ICD 2 section in the Output window within MPLAB IDE:

```
MPLAB ICD 2 Ready
Connecting to MPLAB ICD 2
...Connected
Setting Vdd source to target
Target Device dsPIC30F2010 found, revision = 0x0
...Reading ICD Product ID
Running ICD Self Test
...Passed
MPLAB ICD 2 Ready
```

The errata described in this document will be fixed in future revisions of dsPIC30F2010 silicon.

### Silicon Errata Summary

The following list summarizes the errata described in further detail through the remainder of this document:

1. Run-Time Self Programming (RTSP) of Program Flash Memory  
RTSP operations need to be timed by the application software. Self-timed write operations are not supported.
2. Write/Erase of Data EEPROM  
Write/Erase operations performed on Data EEPROM need to be timed by the application software. Self-timed write operations are not supported.
3. PSV Operations Using SR  
In certain instructions, fetching one of the operands from Program Memory using Program Space Visibility (PSV) will corrupt specific bits in the Status Register, SR.

4. Early Termination of Nested-DO loops  
When using two DO loops in a nested fashion, terminating the inner-level DO loop by setting the EDT(CORCON<11>) bit will produce unexpected results.
5. Reset during Run-Time Self Programming (RTSP) of Program Flash Memory  
When a device reset occurs while an RTSP operation is in progress, code execution may lead into an Address Error trap.
6. Data EEPROM Speed  
Data EEPROM is operational at a device throughput of up to 25 MIPS.
7. Y-Space Data Dependency  
When an instruction that writes to a location in the address range of Y-data memory is immediately followed by a MAC-type DSP instruction that reads a location also resident in Y-data memory, the operations will not be performed as specified.
8. IPC2 SFR Write Sequence  
A specific write sequence for IPC2 (Interrupt Priority Control 2) SFR is required.
9. Interrupting a REPEAT Loop  
When a REPEAT loop is interrupted by two or more interrupts in a nested fashion an Address Error Trap may be caused.
10. 32-bit General Purpose Timers  
The 32-bit General Purpose Timers do not function as specified for prescaler ratios other than 1:1.
11. 10-bit A/D Converter – Sequential Sampling  
Sampling multiple channels sequentially using any conversion trigger other than the auto-convert feature requires SAMC bits to be non-zero.
12. Power-down Current, IPD  
The device exhibits IPD less than 0.1  $\mu$ A. However, certain workarounds are required to achieve IPD in this range.
13. Watchdog Timer  
The Watchdog Timer does not function as specified.

The following sections will describe the errata and work around to these errata, where they may apply.

## 1. Module: RTSP of Program Flash Memory

When performing Run Time Self Programming (RTSP) operations on Program Flash Memory or write operations on Data EEPROM, the device automatically times the erase/write operation. For this revision of silicon, this method of timing the erase/write operation is not supported.

Note that this erratum does not affect programming Flash Memory using a device programmer, such as MPLAB® ICD 2 or PRO MATE®.

### Work around

When updating Program Flash the programming cycle time must be controlled using an on-chip timer resource. Setting the TWRI(NVMCON<8>) bit to a logic '1' enables the Program Flash programming cycle time to be terminated by the next acknowledged interrupt source. Therefore, the user must ensure that a single timer is configured to generate a CPU recognized interrupt and terminate the programming cycle.

The timer cycle should be set for a value greater than 2 milliseconds but less than 5 milliseconds.

Example 1 demonstrates this work around for a programming operation. A similar work around may be applied for an erase operation.

### EXAMPLE 1:

```
;The following code example assumes that the
;Write-latches have been pre-loaded and
;Timer1 has been set up to interrupt at the
;end of the programming cycle.
CLR    MyFlag          ;Clear a flag
CLR    TMR1            ;Clear Timer1
BSET   T1CON, #TON     ;Turn Timer1 On
DISI   #8              ;
MOV    #0X4101, W0     ;Load NVMCON with
MOV    W0, NVMCON      ;bit8 set
MOV    #0X55, W0       ;Perform Unlock
MOV    W0, NVMKEY      ;sequence
MOV    #0XAA, W0
MOV    W0, NVMKEY
BSET   NVMCON, #WR     ;Set the WR bit
NOP    ;CPU stalls until
NOP    ;next interrupt
L1: BTSS MyFlag, #0     ;Optionally wait
BRA    L1              ;for flag set
      ;by Timer1 ISR
      BCLR   T1CON, #TON ;Turn off Timer1
      .....          ;Continue

__T1Interrupt:          ;Timer1 ISR
SETM   MyFlag          ;Set a flag
BCLR   IFS0, #T1IF     ;Clear T1IF and
RETFIE ;return from ISR
```

## 2. Module: Write/Erase of Data EEPROM

When performing write/erase operations on Data EEPROM, the device automatically times the write/erase operation. For this revision of silicon, this method of timing the erase/write operation is not supported.

Note that this erratum does not affect writing to Data EEPROM using a device programmer, such as MPLAB ICD 2 or PRO MATE.

### Work around

When updating Data EEPROM the write cycle time must be controlled using an on-chip timer resource. Setting the TWRI(NVMCON<8>) bit to a logic '1' enables the Data EEPROM write cycle time to be terminated by the next acknowledged interrupt source. Therefore, the user must ensure that a single timer is configured to generate a CPU recognized interrupt and terminate the write cycle.

The timer cycle should be set for a value greater than 2 milliseconds but less than 5 milliseconds.

Example 2 demonstrates this work around. A similar work around may be applied for an erase operation.

### EXAMPLE 2:

```
;The following code example assumes that the
;Write-latches have been pre-loaded and
;Timer1 has been set up to interrupt at the
;end of the write/erase cycle.
CLR    MyFlag          ;Clear a flag
CLR    TMR1            ;Clear Timer1
BSET   T1CON, #TON     ;Turn Timer1 On
DISI   #8              ;
MOV    #0X4105, W0     ;Load NVMCON with
MOV    W0, NVMCON      ;bit8 set
MOV    #0X55, W0       ;Perform Unlock
MOV    W0, NVMKEY      ;sequence
MOV    #0XAA, W0
MOV    W0, NVMKEY
BSET   NVMCON, #WR     ;Set the WR bit
NOP
NOP
L1: BTSS MyFlag, #0     ;Optionally, wait
BRA    L1              ;for flag set
      ;by Timer1 ISR
      BCLR   T1CON, #TON ;Turn off Timer1
      .....          ;Continue

__T1Interrupt:          ;Timer1 ISR
SETM   MyFlag          ;Set a flag
BCLR   IFS0, #T1IF     ;Clear T1IF and
RETFIE ;return from ISR
```

### 3. Module: PSV Operations Using SR

When one of the operands of instructions shown in Table 1 is fetched from Program Memory using Program Space Visibility (PSV), the Status Register, SR and/ or the results may be corrupted. These instructions are identified in Table 1. Example 3 demonstrates one scenario where this occurs.

**TABLE 1:**

Instruction <sup>2</sup>	Examples of Incorrect Operation	Data Corruption IN
ADDC	ADDC W0, [W1++], W2 ;See Note 1	SR<1:0> bits <sup>(3)</sup> , Result in W2
SUBB	SUBB.b W0, [++W1], W3 ;See Note 1	SR<1:0> bits <sup>(3)</sup> , Result in W3
CPB	CPB W0, [W1++], W4 ;See Note 1	SR<1:0> bits <sup>(3)</sup>
RLC	RLC [W1], W4 ;See Note 1	SR<1:0> bits <sup>(3)</sup> , Result in W4
RRC	RRC [W1], W2 ;See Note 1	SR<1:0> bits <sup>(3)</sup> , Result in W2
ADD (Accumulator-based)	ADD [W1++], A ;See Note 1	SR<1:0> bits <sup>(4)</sup>
LAC	LAC [W1], A ;See Note 1	SR<15:10> bits <sup>(4)</sup>

**Note 1:** The errata only affects these instructions when a PSV access is performed to fetch one of the source operands in the instruction. A PSV access is performed when the Effective Address of the source operand is greater than 0x8000 and the PSV (CORCON<2>) bit is set to '1'. In the examples shown, the data access from program memory is made via the W1 register.

**2:** Refer to the Programmer's Reference Manual for details on the dsPIC30F Instruction set.

**3:** SR<1:0> bits represent Sticky Zero and Carry status bits respectively.

**4:** SR<15:10> bits represent Accumulator overflow and saturation status bits

#### EXAMPLE 3:

```
.include "p30fxxxx.inc"
.....
MOV.B #0x00, W0 ;Load PSVPAG register
MOV.B WREG, PSVPAG
BSET CORCON, #PSV;Enable PSV
....
MOV #0x8200, W1;Set up W1 for
;indirect PSV access
;from 0x000200
ADD W3, [W1++], W5 ;This instruction
;works ok
ADDC W4, [W1++], W6;Carry flag and
;W6 gets
;corrupted here!
```

#### Work around

##### Work around 1: For Assembly Language Source Code

To work around the erratum in the MPLAB® ASM30 assembler, the application may perform a PSV access to move the source operand from Program memory to RAM or a W register prior to performing the operations listed in Table 1. The work around for Example 3 is demonstrated in Example 4.

#### EXAMPLE 4:

```
.include "p30fxxxx.inc"
.....
MOV.B #0x00, w0 ;Load PSVPAG register
MOV.B WREG, PSVPAG
BSET CORCON, #PSV;Enable PSV
....
MOV #0x8200, W1;Set up W1 for
;indirect PSV access
;from 0x000200
ADD W3, [W1++], W5;This instruction
;works ok
MOV [W1++], W2 ;Load W2 with data
;from program memory
ADDC W4, W2, W6 ;Carry flag and W4
;results are ok!
```

#### Work Around 2: For C Language Source Code

For applications using C language, MPLAB C30 versions 1.20.04 or higher provide the following command-line switch that implements a work around for the erratum.

```
-merrata=psv
```

Refer to the "readme.txt" file in the MPLAB C30 v1.20.04 toolsuite for further details.

## 4. Module: Early Termination of Nested DO Loops

When using two DO loops in a nested fashion, terminating the inner-level DO loop by setting the EDT(CORCON<11>) bit will produce unexpected results. Specifically, the device may continue executing code within the outer DO loop forever. This erratum does not affect the operation of the MPLAB C30 compiler.

### Work around

The application should save the DCOUNT SFR prior to entering the inner DO loop and restore it upon exiting the inner DO loop. This work around is shown in Example 5.

### EXAMPLE 5:

```
.include "p30fxxxx.inc"
.....
DO #CNT1, LOOP0      ;Outer loop start
....
PUSH    DCOUNT      ;Save DCOUNT
DO      #CNT2, LOOP1  ;Inner loop
....
                ;starts
BTSS    Flag, #0
BSET    CORCON, #EDT;Terminate inner
....
                ;DO-loop early
....
LOOP1: MOV    W1, W5    ;Inner loop ends
POP     DCOUNT        ;Restore DCOUNT
...
LOOP0: MOV    W5, W8    ;Outer loop ends

Note:  For details on the functionality of
       EDT bit, see section 2.9.2.4
       in the dsPIC30F Family Reference
       Manual.
```

## 5. Module: Reset During RTSP of Program Flash Memory

If a device reset occurs while an RTSP operation is in progress, code execution after the reset may lead to an Address Error Trap.

### Work around

The user should define an Address Error Trap service routine as shown in Example 6 in order to resume normal code execution.

### EXAMPLE 6:

```
__AddressError:
    bclr    RCON, #TRAPR    ;Clear the Trap
                        ;Reset Flag Bit
    bclr    INTCON1, #ADDRERR ;Clear the
                        ;Address Error
                        ;trap flag bit
    reset                        ;Software reset
```

## 6. Module: Data EEPROM – Speed

At device throughput greater than 25 MIPS, read operations performed on Data EEPROM may not function correctly.

### Work around

When reading data from Data EEPROM, the application should perform a clock-switch operation to lower the frequency of the system clock so that the throughput is less than 25 MIPS. This may be easily performed at any time via the Oscillator Postscaler bits, POST (OSCCON<7:6>), that allow the application to divide the system clock down by a factor of 4, 16 or 64.

## 7. Module: Y-Space Data Dependency

When an instruction that writes to a location in the address range of Y-data memory (addresses between 0x0900 and 0x09FF) is immediately followed by a MAC-type DSP instruction that reads a location also resident in Y-data memory, the two operations will not be executed as specified. This is demonstrated in Example 7.

### EXAMPLE 7:

```
MOV      #0x090A, W0      ;Load address > =
                        ;0x900 into W0
MOV      #0x09B0, W10     ;Load address >=
                        ;0x900 into W10
MOV      W2, [W0++]       ;Perform indirect
                        ;write via W0 to
                        ;address >= 0x900
MAC      W4*W5, A, [W10] +=2, W5 ;Perform
                        ;read operation
                        ;using Y-AGU

:Unexpected Results!
```

### Work around

#### Work around 1:

Insert a NOP between the two instructions as shown in Example 8.

### EXAMPLE 8:

```
MOV      #0x090A, W0      ;Load address > =
                        ;0x900 into W0
MOV      #0x09B0, W10     ;Load address >=
                        ;0x900 into W10
MOV      W2, [W0++]       ;Perform indirect
                        ;write via W0 to
                        ;address >= 0x900
NOP                        ;No operation
MAC      W4*W5, A, [W10] +=2, W5 ;Perform
                        ;read operation
                        ;using Y-AGU

:Correct Results!
```

## Work around 2:

If Work around #1 is not feasible due to application real-time constraints, the user may take precautions to ensure that a write operation performed on a location in Y-data memory is not immediately followed by a DSP MAC-type instruction that performs a read operation of a location in Y-data memory.

## 8. Module: Interrupt Controller

A specific write sequence for IPC2 (Interrupt Priority Control 2) SFR is required to prevent possible data corruption in the IEC2 (Interrupt Enable Control 2) SFR. Interrupts must be disabled during this IPC2 SFR write sequence.

### Work around

An example of this write sequence is shown in Example 9.

### EXAMPLE 9:

```
mov  #IPC2, w0      ;Point w0 to IPC2
mov  #0x4444, w1    ;Write data to go to IPC2
disi #2             ;Disable interrupts for
                    ;next two cycles
mov  w1, IPC2       ;Write the data to IPC2
mov  #IPC2, w0      ;Target w1 to keep IPC2
                    ;address on bus
```

When coding in C, the write sequence shown above can be implemented using inline assembly instructions. The equivalent write sequence using the C30 compiler is shown in Example 10.

### EXAMPLE 10:

```
asm volatile( "push.d  w0\n\t"
              "mov  #IPC2,w0\n\t"
              "mov  #0x4444,w1\n\t"
              "disi #2\n\t"
              "mov  w1, IPC2\n\t"
              "mov  #IPC2, w0\n\t"
              "pop.d  w0");
//Note: There are no commas between
//      the quoted strings in the code
//      segment above.
```

## 9. Module: Interrupting a REPEAT loop

When interrupt nesting is enabled (or NSTDIS (INTCON1<15>) bit is '0'), the following sequence of events will lead to an Address Error Trap:

1. REPEAT-loop is active
2. An interrupt is generated during the execution of the REPEAT-loop.
3. The CPU executes the Interrupt Service Routine (ISR) of the source causing the interrupt.
4. Within the ISR, when the CPU is executing the first instruction cycle of the 3-cycle RETFIE (Return-from-interrupt) instruction, a second interrupt is generated by a source with a higher interrupt priority.

### Work around

Processing of Interrupt Service Routines should be disabled while the RETFIE instruction is being executed. This may be accomplished in two different ways:

1. Place a DISI instruction immediately before the RETFIE instruction in all interrupt service routines of interrupt sources that may be interrupted by other higher priority interrupt sources (with priority levels 1 through 6). This is shown in Example 11 in the Timer1 ISR. In this example, a DISI instruction inhibits level 1 through level 6 interrupts for 2 instruction cycles, while the RETFIE instruction is executed.

### EXAMPLE 11:

```
T1Interrupt:      ;Timer1 ISR
— PUSH  W0        ;This line optional
.....
BCLR  IFS0, #T1IF
POP   W0          ;This line optional
DISI  #1
RETFIE             ;Another interrupt occurs
                  ;here and it is processed
                  ;correctly
```

2. Immediately prior to executing the RETFIE instruction, increase the CPU priority level by modifying the IPL<2:0> (SR<7:5>) bits to '111' as shown in Example 12. This will disable all interrupts between priority levels 1 through 7.

### EXAMPLE 12:

```
T1Interrupt:      ;Timer1 ISR
— PUSH  W0
.....
BCLR  IFS0, #T1IF
MOV.B #0xE0, W0
MOV.B WREG, SR
POP   W0
RETFIE             ;Another interrupt occurs
                  ;here and it is processed
                  ;correctly
```

## 10. Module: 32-bit General Purpose Timers

Pairs of 16-bit timers may be combined to form 32-bit timers. For example, Timer2 and Timer3 are combined into a single 32-bit timer. For this release of silicon, when a 32-bit timer is prescaled by ratios other than 1:1, unexpected results may occur.

### Work around

None. The application may only use the 1:1 prescaler for 32-bit timers.

## 11. Module: 10-bit A/D Converter – Sequential Sampling

Sampling multiple channels sequentially using any conversion trigger source other than the auto-convert feature requires SAMC bits to be non-zero. Thus, if the following conditions are all satisfied, the module may not operate as specified:

- Multiple S/H channels are sampled sequentially  
CHPS(ADCON2<9:8>) is not equal to '00' and  
SIMSAM(ADCON1<3>) = 0
- Auto-convert option is not chosen as the conversion trigger  
SSRC(ADCON1<7:5>) is not equal to '111'
- SAMC(ADCON3<12:8>) is equal to '00000'

### Work around

Set the value of the SAMC bits to anything other than '00000'. The module will now operate as specified.

## 12. Module: IPD – Sleep Current

The device exhibits IPD of approximately 100  $\mu$ A.

### Work around

If the application does not use the on-chip A/D converter, it is possible to reduce the IPD to values below 0.1  $\mu$ A. The following additional measures need to be taken in these circumstances:

1. In the application hardware, the VREF+/RB0 pin (Pin 2) on the dsPIC30F2010 should be connected to the circuit ground (GND).
2. In the application software, the code sequence shown in Example 13 should be executed to bring the device into the power-saving Sleep mode.

### EXAMPLE 13:

```
.include "p30f2010.inc"
.....
BCLR   ADCON1, #ADON   ;Required code
MOV     #0x2000, W0     ;sequence for
MOV     W0, ADCON2      ;low power-down
BCLR   PMD1, #ADCMD     ;current.
PWRSAV #SLEEP_MODE     ;Device enters
                               ;SLEEP mode here
```

## 13. Module: Watchdog Timer

The Watchdog Timer does not function as specified. If the CLRWDT instruction is not executed before the Watchdog Timer is half-expired or greater, the device will reset.

### Work around

The user must always issue the CLRWDT instruction before the Watchdog timer is half-expired. For instance, if the Watchdog time-out period is configured for 2 msec, the CLRWDT instruction must be executed faster than every 1 msec.

## APPENDIX A: REVISION HISTORY

### Revision A (1/2004)

Original version of the document.

### Revision B (2/2004)

Document status was updated from “Confidential” to “Advance Information” designation.

Clarifications/Corrections to dsPIC30F Datasheets were removed.

### Revision C (2/2004)

Changes were made to the C code example under Errata #5.

### Revision D (4/2004)

Errata #8, “Motor Control PWM: Configuration Fuse Bits” was removed.

Added Errata #3 and #9.

### Revision E (11/2004)

Added errata #3 and #4.

### Revision F (12/2004)

Added errata #13.

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.

---

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's products as critical components in life support systems is not authorized except with express written approval by Microchip. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

#### **Trademarks**

The Microchip name and logo, the Microchip logo, Accuron, dsPIC, KEELOQ, microID, MPLAB, PIC, PICmicro, PICSTART, PRO MATE, PowerSmart, rPIC, and SmartShunt are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.


AmpLab, FilterLab, MXDEV, MXLAB, PICMASTER, SEEVAL, SmartSensor and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, dsPICDEM, dsPICDEM.net, dsPICworks, ECAN, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, Migratable Memory, MPASM, MPLIB, MPLINK, MPSIM, PICkit, PICDEM, PICDEM.net, PICLAB, PICtail, PowerCal, PowerInfo, PowerMate, PowerTool, rLAB, rPICDEM, Select Mode, Smart Serial, SmartTel and Total Endurance 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.

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

© 2004, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

 Printed on recycled paper.

**QUALITY MANAGEMENT SYSTEM**  
**CERTIFIED BY DNV**  
**== ISO/TS 16949:2002 ==**

*Microchip received ISO/TS-16949:2002 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona and Mountain View, California in October 2003. The Company's quality system processes and procedures are for its PICmicro® 8-bit MCUs, 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

---

### DS80178EAMERICAS

#### Corporate Office

2355 West Chandler Blvd.  
Chandler, AZ 85224-6199  
Tel: 480-792-7200  
Fax: 480-792-7277  
Technical Support:  
<http://support.microchip.com>  
Web Address:  
[www.microchip.com](http://www.microchip.com)

#### Atlanta

Alpharetta, GA  
Tel: 770-640-0034  
Fax: 770-640-0307

#### Boston

Westford, MA  
Tel: 978-692-3848  
Fax: 978-692-3821

#### Chicago

Itasca, IL  
Tel: 630-285-0071  
Fax: 630-285-0075

#### Dallas

Addison, TX  
Tel: 972-818-7423  
Fax: 972-818-2924

#### Detroit

Farmington Hills, MI  
Tel: 248-538-2250  
Fax: 248-538-2260

#### Kokomo

Kokomo, IN  
Tel: 765-864-8360  
Fax: 765-864-8387

#### Los Angeles

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

#### San Jose

Mountain View, CA  
Tel: 650-215-1444  
Fax: 650-961-0286

#### Toronto

Mississauga, Ontario,  
Canada  
Tel: 905-673-0699  
Fax: 905-673-6509

### ASIA/PACIFIC

#### Australia - Sydney

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

#### China - Beijing

Tel: 86-10-8528-2100  
Fax: 86-10-8528-2104

#### China - Chengdu

Tel: 86-28-8676-6200  
Fax: 86-28-8676-6599

#### China - Fuzhou

Tel: 86-591-8750-3506  
Fax: 86-591-8750-3521

#### China - Hong Kong SAR

Tel: 852-2401-1200  
Fax: 852-2401-3431

#### 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: 86-755-8203-2660  
Fax: 86-755-8203-1760

#### China - Shunde

Tel: 86-757-2839-5507  
Fax: 86-757-2839-5571

#### China - Qingdao

Tel: 86-532-502-7355  
Fax: 86-532-502-7205

### ASIA/PACIFIC

#### India - Bangalore

Tel: 91-80-2229-0061  
Fax: 91-80-2229-0062

#### India - New Delhi

Tel: 91-11-5160-8632  
Fax: 91-11-5160-8632

#### Japan - Kanagawa

Tel: 81-45-471- 6166  
Fax: 81-45-471-6122

#### Korea - Seoul

Tel: 82-2-554-7200  
Fax: 82-2-558-5932 or  
82-2-558-5934

#### Singapore

Tel: 65-6334-8870  
Fax: 65-6334-8850

#### Taiwan - Kaohsiung

Tel: 886-7-536-4818  
Fax: 886-7-536-4803

#### Taiwan - Taipei

Tel: 886-2-2500-6610  
Fax: 886-2-2508-0102

#### Taiwan - Hsinchu

Tel: 886-3-572-9526  
Fax: 886-3-572-6459

### EUROPE

#### Austria - Weis

Tel: 43-7242-2244-399  
Fax: 43-7242-2244-393

#### Denmark - Ballerup

Tel: 45-4420-9895  
Fax: 45-4420-9910

#### France - Massy

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

#### Germany - Ismaning

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

#### Italy - Milan

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

#### Netherlands - Drunen

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

#### England - Berkshire

Tel: 44-118-921-5869  
Fax: 44-118-921-5820

09/27/04