
PIC[®] MCU和dsPIC[®] DSC的B类安全软件库

作者: Veena Kudva、Adrian Aur 和
Anton Alkhimenok
Microchip Technology Inc.

简介

本应用笔记介绍了B类安全软件库程序，其作用是检测单通道MCU中是否发生故障。这些程序根据IEC 60730标准开发，能够支持B类认证过程。这些程序可直接集成到最终用户的应用程序中，以测试和验证控制器的关键功能，而不会影响最终用户的应用程序。

本应用笔记还介绍了B类安全软件库中的应用程序编程接口（Application Programming Interface, API）函数。

在启动或运行时，可以周期性调用B类安全软件程序来测试以下组件：

- CPU 寄存器
- CPU 程序计数器
- 不变存储器
- 可变存储器
- 时钟

此外，本应用笔记还概述了多种不属于B类安全软件库的技术，以测试外部通信、时序、I/O 外设、模拟I/O和模拟多路开关等组件。

注： 本文档中使用的术语“IEC 60730标准”是指“IEC 60730-1 第5版”版权所有 © 2013 IEC，瑞士日内瓦。www.iec.ch。

IEC 60730标准概述

注： 感谢国际电工委员会（International Electrotechnical Commission, IEC）允许我们复制国际标准IEC 60730-1 第5版（2013）内的信息。所有这类摘要的版权为瑞士日内瓦IEC组织所有。保留所有权利。有关IEC的更多信息，请访问www.iec.ch。IEC对作者引用的摘要及内容的编排和上下文不承担任何责任，IEC也不对文中其他任何内容及其准确性承担任何责任。

IEC 60730标准定义了确保家用电器中使用的受控设备安全运行的测试和诊断方法。IEC 60730标准的附录H将安全软件分为以下三类（见附录B：“IEC 60730-1 表H.1（第3版的H.11.12.7）”）：

- A类
- B类
- C类

B类安全软件库执行属于B类的重要测试和诊断方法。这些方法采用各种措施来检测与软件相关的故障和错误，并作出响应。

根据IEC 60730标准，采用B类函数进行控制应具有以下结构之一：

- 具有功能测试的单通道
在该结构中，功能测试在应用程序固件执行之前执行。
- 具有周期性自检功能的单通道
在该结构中，周期性测试被嵌入在固件中，且在固件处于执行模式时进行周期性自检。
- 不具备比较功能的双通道
在该结构中，可通过两种不同的方法来执行指定操作。

系统要求

要运行B类安全软件库，建议满足以下系统要求：

因为测试需要独立间隙监视，所以系统硬件必须至少有两个独立的时钟源（例如，内部振荡器、晶振和工频）。

B类安全软件库

16位B类安全软件库包含多个API，它们通过故障检测最大限度地提高应用程序的可靠性。这些API有助于应用程序符合IEC 60730标准。使用该库可以实现以下测试：

- CPU寄存器测试
- 程序计数器测试
- 不变存储器（闪存/EEPROM）测试
- 可变存储器测试
- 时钟测试

下面的部分将讨论每种测试及其实现细节。另外，每部分还列出了执行相应测试所需的API。

CPU寄存器测试

CPU寄存器测试用于实现IEC 60730标准定义的H.2.16.5功能测试。该测试用于检测CPU寄存器中是否存在卡住故障，即确保寄存器中的位不是一直为0或1，它是一种非破坏性测试。

该测试执行以下主要任务：

1. 在测试前保存待测试的CPU寄存器内容，并在测试后将其恢复。
2. 首先向待测试的寄存器依次写入二进制序列010101...和101010...（长度取决于架构），然后再读取寄存器内容进行验证。
3. 如果返回值与先前存入的值不匹配，则测试将返回错误代码。

API函数

下面的API函数用于实现CPU寄存器测试：

```
CLASSB_CPURegistersTest
```

程序计数器测试

程序计数器（Program Counter, PC）测试用于实现IEC 60730标准定义的H.2.16.5功能测试。PC中保存下一条将要执行的指令的地址。

该测试执行以下主要任务：

1. PC测试调用位于闪存中不同地址处的函数。
2. 这些函数复位错误标志。
3. 在应用程序代码的多个位置测试错误标志。
4. 如果错误标志清零，表明PC已跳转到正确的位置。

API函数

下面的API函数用于实现PC测试：

- CLASSB_CPUPCTest()
- CLASSB_CPUPCTestGetResult()

不变存储器（闪存/EEPROM）测试

不变存储器（闪存/EEPROM）测试用于实现 IEC 60730 标准定义的周期性修改校验和 H.2.19.3.1 功能测试。该测试用于检测不变存储器中是否存在单个位故障。不希望系统内的不变存储器（例如闪存和 EEPROM 存储器）中保存的数据在程序执行期间发生变化。闪存/EEPROM 不变存储器测试使用循环冗余校验（CRC-16）计算周期性校验和。下面给出了用于计算 CRC-16 的 CRC 多项式。

CRC-16 = 1 1000 0000 0000 0101 = 8005（十六进制）

图1给出了不变存储器测试的流程图。

CRC16 计算函数返回最终 CRC 值，该值用于执行以下操作：

- 1. 在系统启动时，如果 CRC 标志设置为 0x00，则计算参考 CRC 校验和。
- 2. 参考校验和存储在闪存或 EEPROM 存储器中，CRC 标志设置为 0xFF。
- 3. 如果 CRC 标志设置为 0xFF，则可以周期性调用 CRC16 计算函数。
- 4. 将第3步计算得到的校验和与参考校验和进行比较。
- 5. 如果两个值相匹配，用户应用程序会将状态位置 1，以指示不变存储器已通过测试，未发现错误。

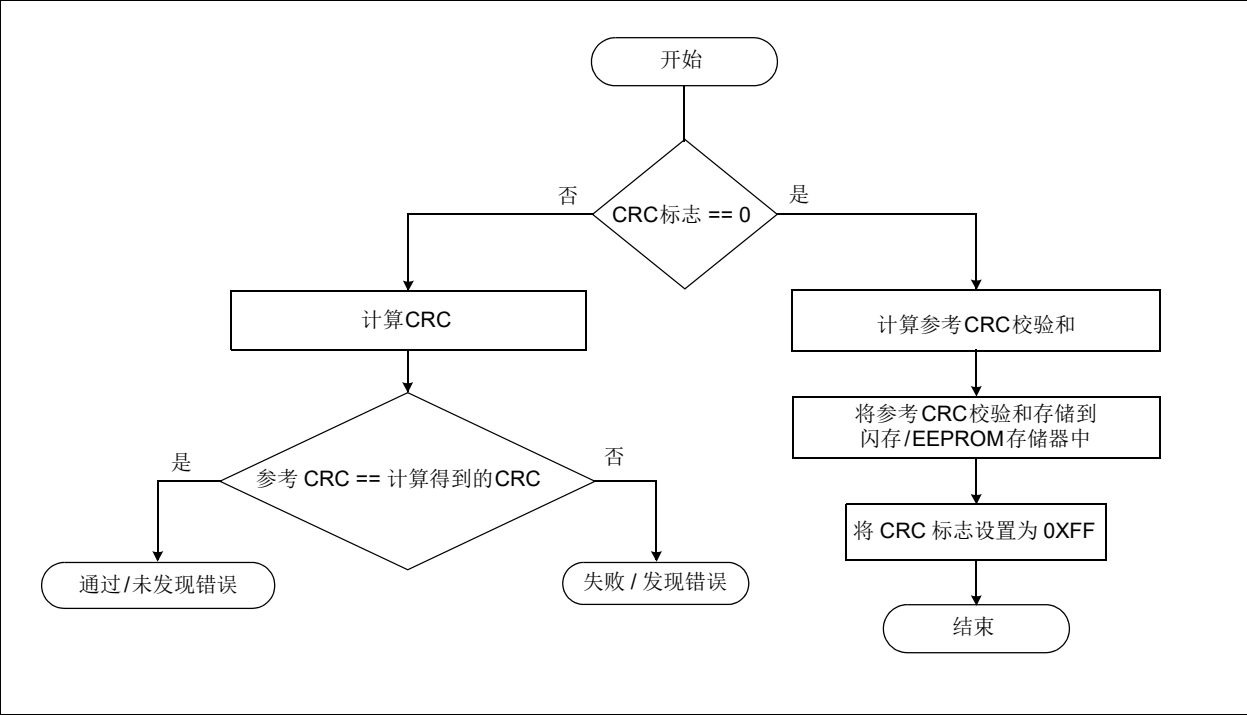
API 函数

下面的 API 函数用于实现不变存储器测试：

- CLASSB_CRCFlashTest
- CLASSB_CRCEEPROMTest

注： 16 位 EEPROM 测试仅适用于 dsPIC30F 器件。

图1： 不变存储器测试流程图



可变存储器测试

可变存储器测试用于实现 IEC 60730 标准定义的周期性静态存储器 H.2.19.6 功能测试。该测试用于检测可变存储器中是否存在单个位故障。可变存储器内的数据在程序执行期间可能会发生改变。RAM 存储器测试用于确定 RAM 存储器中的某位是否一直为 1 或 0。March 存储器测试和 Checkerboard 测试是两种广泛用于检查直流故障的静态存储器算法。

使用 B 类安全软件库可以实现以下测试：

- March 测试
 - March C/C- 测试
 - March B 测试

MARCH 测试

March 测试在存储器阵列中的每个存储单元上执行一组限定数量的操作。每个操作执行以下任务：

1. 向存储单元内写入 0（w0）。
2. 向存储单元内写入 1（w1）。
3. 从存储单元内读取预期值 0（r0）。
4. 从存储单元内读取预期值 1（r1）。

March 测试符号

图2 说明了 March 测试中使用的符号。

图2: MARCH 测试符号

↑↑: 地址序列按升序排列。

↓↓: 地址序列按降序排列。

↑↓: 地址序列按升序或降序排列。

r0: 表示读操作（从存储单元内读取 0）。

r1: 表示读操作（从存储单元内读取 1）。

w0: 表示写操作（向存储单元内写入 0）。

w1: 表示写操作（向存储单元内写入 1）。

MARCH C/C- 测试

March C/C- 测试用于检测可变存储器中是否存在以下类型的故障：

- 卡住故障
- 寻址故障
- 转换故障
- 耦合故障

March C/C- 测试的复杂度分别为 11n 和 10n，其中 n 表示存储器中的位数。该测试能够以破坏性模式或非破坏性模式运行。如果以非破坏性模式运行，则需要使用缓冲区空间来存储待测试和恢复的存储器内容。当系统启动时，如果需要，可以在初始化存储器之前运行该测试。

例1 给出了用于演示如何实现 March C 测试的伪代码。

API 函数

下面的 API 函数用于实现 March C/C- 测试：

CLASSB_RAMMarchCTest

图3 给出了一个 March C 算法。

图3: MARCH C 算法

MarchC

{

↑↓(w0); ↑(r0, w1); ↑↓(r1, w0);

↑↓(0); ↓↓(r0, w1); ↓↓(r1, w0); ↑(r0)

}

注: C- 可跳过该步骤。

例1: MARCH C测试的伪代码

```

/* 升序: 写入0 */
for(i=0;i<=(n-1);i++)
    x(i)=0;

/* 升序: 读为0, 写入1 */
for(i=0;i<=(n-1);i++)
{
    if (x(i)==0)
        x(i) =1;
    else
        return fail;
}

/* 升序: 读为1, 写入0 */
for(i=0;i<=(n-1);i++)
{
    if(x(i)==1)
        x(i)=0;
    else
        return fail; }

/* 仅限标准March    C*/
/* 升序: 读为0 */

if ( minus != 0){

    for(i=(n-1);i>=0;i--)
    {
        if(x(i)==0) {}
        else
            return fail ;
    }
}

/* 降序: 读为0, 写入1 */
for(i=(n-1);i>=0;i--)
{
    if(x(i)==0)
        x(i)=1;
    else
        return fail;
}

/* 降序: 读为1, 写入0 */
for(i=(n-1);i>=0;i--)
{
    if(x(i)==1)
        x(i)=0;
    else
        return fail;
}

/* 升序: 读为0 */
for(i=(n-1);i>=0;i--)
{
    if(x(i)==0) {}
    else
        return fail;
}
return pass;

```

MARCH B 测试

March B 是一种非冗余测试，用于检测以下类型的故障：

- 卡住故障
- 链接的幂等耦合故障
- 跳变耦合故障

该测试的复杂度为 $17n$ ，其中 n 表示存储器中的位数。该测试能够以破坏性模式或非破坏性模式运行。如果以非破坏性模式运行，则需要使用缓冲区空间来存储待测试和恢复的存储器内容。当系统启动时，如果需要，可以在初始化存储器之前运行该测试。

图4 给出了一个 March B 算法。

图4： MARCH B 算法

```
MarchB
{
   $\Downarrow(w0); \Uparrow(r0, w1, r1, w0, r0, w1); \Uparrow(r1, w0, w1);$ 
   $\Downarrow(r1, w0, w1, w0); \Downarrow(r0, w1, w0);$ 
}
```

例2 给出了用于演示如何实现 March B 测试的伪代码。

API 函数

下面的 API 函数用于实现 March B 测试：

`CLASSB_RAMMarchBTest`

例2: MARCH B测试的伪代码

```

/* 写入0 */
for(i=0;i<=(n-1);i++)
    x(i)=0;
/* 升序: 读为0, 写入1; 读为1, 写入0; 读为0, 写入1 */
for(i=0;i<=(n-1);i++)
{
    if(x(i)=0)
    {
        x(i)=1;
    }
    else
        return fail;
    if(x(i)=1)
    {
        x(i)=0;
    }
    else
        return fail;
    if(x(i)=0)
    {
        x(i)=1;
    }
    else
        return fail;
/* 升序: 读为1, 写入0; 写入1 */
for(i=0;i<=(n-1);i++)
{
    if(x(i)=1)
    {
        x(i)=0;
        x(i)=1;
    }
    else
        return fail;
/* 降序: 读为1, 写入0, 写入1, 写入0 */
for(i=(n-1);i>=0;i--)
{
    if(x(i)=1)
    {
        x(i)=0;
        x(i)=1;
        x(i)=0;
    }
    else
        return fail;
/* 降序: 读为0, 写入1, 写入0; */
for(i=(n-1);i>=0;i--)
{
    if(x(i)=0)
    {
        x(i)=1;
        x(i)=0;
    }
    else
        return fail;
}
return pass;

```

CHECKERBOARD RAM测试

Checkerboard RAM测试向一系列相邻存储单元写入棋盘格图样。该测试以4字节为单位（存储块）执行。这是一种非破坏性存储器测试。

该测试执行以下主要任务：

1. 将待测试的存储单元的内容保存到CPU寄存器中。
2. 向存储单元“N”写入二进制值（长度取决于架构）101010...，向存储单元“N+1”写入取反后的二进制值010101...，以此类推，直到写满整个存储块。
3. 读取当前存储块中所有存储单元的内容，并进行验证。如果值匹配，则函数继续执行；否则，函数停止执行，并返回错误。
4. 重复执行第2步和第3步，分别向“N”和“N+1”单元写入取反后的图样。
5. 测试完一个存储块后，立即开始测试下一个存储块，直到所有要求的存储区域均已测试。

API函数

下面的API函数用于实现Checkerboard RAM测试：

`CLASSB_RAMCheckerboardTest`

时钟测试

根据IEC 60730标准，只需测试时钟的谐波和次谐波。时钟测试用于实现IEC 60730标准定义的独立时隙监视H.2.18.10.4功能测试。该测试验证系统时钟的可靠性（即，系统时钟不应太快也不应太慢）。

时钟测试函数用于验证CPU时钟是否正常工作。

该测试执行以下主要任务：

1. 测试需要使用独立时钟源（或参考时钟源），可以是辅助振荡器或电源线。参考时钟应连接到Timer1等定时器。
2. 在测试期间统计每个参考时钟周期内的CPU时钟周期数。
3. 如果时钟周期数超出指定范围，该函数将返回错误代码。

API函数

下面的API函数用于实现时钟测试：

`CLASSB_ClockTest`

可变存储器及不变存储器的寻址及内部数据路径

对于PIC MCU和dsPIC DSC等单芯片单片机或数字信号控制器，周期性静态存储器测试用于测试可变存储器，而周期性校验和用于测试不变存储器。这些测试用于检测内部地址总线和内部数据路径中是否存在卡住故障。

错误地址寻址

具有外部存储器的单片机才需要进行该测试。

外部通信

IEC 60730 B类规范建议采取以下措施以确保组件之间的可靠通信：

传输冗余

传输冗余是一种故障/错误控制技术，用于防止输入和输出信息中出现同步和/或系统性错误。其实现方式是在发送器与接收器之间至少连续传输两次数据，然后进行比较。

协议测试

协议测试是一种故障/错误控制技术，通过与计算机组件之间来回传输数据的方式检测内部通信协议中的错误。

CRC 单字

CRC 多项式用于计算所发送报文的CRC 校验和。在发送端，将CRC 校验和添加到报文之后再发送出去。在接收端，接收器使用相同的CRC 多项式计算CRC 校验和，然后将计算得到的值与接收到的值进行比较。

时序

PIC MCU 和 dsPIC DSC 具有多个专用通信接口，例如 UART、I²C 和 SPI 模块。IEC 60730 B 类规范建议这些模块使用时隙监视，以确保在正确的时间点进行通信。

合理性检查

I/O 外设、模拟多路开关和 A/D 转换器的合理性检查执行如下：

I/O 外设

I/O 引脚的合理性检查可通过翻转 I/O 并检查引脚状态来执行。

模拟多路开关

要验证模拟多路开关的操作，应向所有通道施加一个已知的电压值。读取这些值，并将其与施加的电压进行比较以进行验证。

A/D 转换器

要测试 A/D 转换器的模拟功能，应向模拟输入施加一个已知的外部电压，然后将转换结果与施加的电压进行比较。

CLASSB_CPURegistersTest

说明

该函数执行CPU寄存器测试。该测试首先向CPU寄存器依次写入值0x5555和0xAAAA，然后再读取寄存器内容来进行验证。如果值不匹配，该函数将返回一个非零值。结果返回到W0寄存器中。因此，W0寄存器的内容不会保留。每次寄存器测试之前都会保存待测试的CPU寄存器内容，待测试完成后进行恢复。

原型

```
CLASSBRESULT CLASSB_CPURegistersTest();
```

参数

无

返回值

CLASSB_TEST_PASS（返回值 = 0）——测试成功完成

CLASSB_TEST_FAIL（返回值 != 0）——测试失败

源文件

classb_registers.s

表1：资源需求

参数	需求
闪存程序存储器	1054个字节
执行时间	240个指令周期
中断	禁止39个指令周期

CLASSB_CPUPCTest

说明

该函数执行程序计数器（PC）测试，即PC的功能测试。该测试调用位于闪存中不同地址处的函数。RAM中存储的错误标志的指针将传送到这些函数。这些函数会在返回值前清零错误标志。如果错误标志未清零，表示PC已损坏，函数未被调用。要检查PC错误标志，应使用另一个函数CLASSB_CPUPCTestGetResult()。测试函数的地址基于器件闪存大小。闪存大小应使用CLASSB_DEVICE_FLASH_SIZE_KB参数在classb_config.h文件中指定。

原型

```
void CLASSB_CPUPCTest();
```

参数

无

返回值

无

源文件

classb_pc.s

表2: 资源需求

参数	需求
闪存程序存储器	41个字节
执行时间	32条指令周期
中断	禁止32个指令周期

CLASSB_CPUPCTestGetResult

说明

该函数用于检查CLASSB_CPUPCTest() 函数执行的PC测试的结果，应在应用程序代码的多个位置进行调用，尤其是关键操作之前。

原型

```
CLASSBRESULT CLASSB_CPUPCTestGetResult();
```

参数

无

返回值

CLASSB_TEST_PASS (返回值 = 0) —— 测试成功完成

CLASSB_TEST_FAIL (返回值 != 0) —— 测试失败

源文件

classb.h

备注

无

表3: 资源需求

参数	需求
闪存程序存储器	N/A
执行时间	N/A
中断	允许

CLASSB_CRCFlashTest

说明

该函数执行不变存储器测试。它使用提供的 *crcSeed* 计算自 *startAddress* 起 *length* 个字节数据的 CRC，然后返回最终 CRC 值。

原型

```
uint16_t CLASSB_CRCFlashTest(startAddress, length, crcSeed)
```

参数

- *startAddress*——被测试存储器的第一个地址（必须为偶数）
- *length*——被测试存储器的字节长度（必须为偶数）
- *crcSeed*——CRC 校验和的初始值

返回值

crc_Result 保存 CRC 结果

源文件

classb_crc.c

备注

无

表 4: 资源需求

参数	需求
闪存	383 个字节
执行时间	160000 个指令周期/KB
中断	允许

AN1778

CLASSB_CRCEEPROMTest

说明

该函数计算EEPROM存储区的CRC校验和。变量`startAddress`和`length`必须为偶数。

原型

```
uint16_t CLASSB_CRCEEPROMTest (startAddress, length, crcSeed)
```

参数

- `startAddress`——被测试EEPROM存储器的第一个地址（必须为偶数，以字节为单位）
- `length`——被测试EEPROM存储器的字节长度（必须为偶数）
- `crcSeed`——CRC校验和的初始值

返回值

该函数返回标准16位CRC校验和

备注

无

源文件

`classb_crc.c`

表5: 资源需求

参数	需求
闪存	383个字节
执行时间	128000个指令周期/KB
中断	允许

CLASSB_RAMMarchCTest

说明

该函数对RAM存储器执行March C和March C-测试。变量`startAddress`和`length`必须为偶数。该测试能够以破坏性模式或非破坏性模式运行。当未指定用于保存存储器内容的缓冲区时，测试以破坏性模式运行，存储器内容不进行保存。要以非破坏性模式运行，必须定义缓冲区地址。被测试的RAM区域不得超出存储缓冲区区域。此外，当系统启动时，还可以在初始化存储器之前运行该测试。要在系统启动时运行该测试，必须将`classb_config.h`中的`CLASSB_MARCH_C_STARTUP`或`CLASSB_MARCH_C_MINUS_STARTUP`参数设置为非零值。测试期间禁止中断。

原型

```
CLASSBRESULT CLASSB_RAMMarchCTest (startAddress, length, bufferAddress, minus)
```

参数

- `startAddress`——被测试存储器的第一个地址（必须为偶数）
- `length`——被测试存储器的字节长度（必须为偶数）
- `bufferAddress`——用于保存被测试存储器内容的缓冲区的第一个地址（必须为偶数）；如果该参数为NULL，则被测试存储器将清空
- `minus`——如果参数为非零，则使用“minus”算法。

返回值

`CLASSB_TEST_PASS`（返回值 = 0）——测试成功完成

`CLASSB_TEST_FAIL`（返回值 != 0）——测试失败

备注

无

源文件

`classb_marchc.s`

表6： 资源需求

参数	需求
存储器	1364 个字节
执行时间	March C 115200 个指令周期/KB March C- 121600 个指令周期/KB
中断	测试期间禁止

CLASSB_RAMMarchBTest

说明

该函数对RAM存储器执行March B测试。变量startAddress和length必须为偶数。该测试能够以破坏性模式或非破坏性模式运行。当未指定bufferAddress时，测试以破坏性模式运行，存储器内容不进行保存。要以非破坏性模式运行，必须定义bufferAddress。被测试的RAM区域不得超出存储缓冲区区域。此外，当系统启动时，还可以在初始化存储器之前运行该测试。要在系统启动时运行该测试，必须将classb_config.h中的CLASSB_MARCH_B_STARTUP参数设置为非零值。测试期间禁止中断。

原型

```
CLASSBRESULT CLASSB_RAMMarchBTest(startAddress, length, bufferAddress);
```

参数

- startAddress——被测试存储器的第一个地址（必须为偶数）
- length——被测试存储器的字节长度（必须为偶数）
- bufferAddress——用于保存被测试存储器内容的缓冲区的第一个地址（必须为偶数）；如果该参数为NULL，则被测试存储器将清空

返回值

CLASSB_TEST_PASS（返回值 = 0）——测试成功完成
CLASSB_TEST_FAIL（返回值 != 0）——测试失败

备注

无

源文件

classb_marchb_.s

表7：资源需求

参数	需求
闪存	2003 个字节
执行时间	192000 个指令周期/KB
中断	测试期间禁止

CLASSB_RAMCheckerboardTest

说明

该函数对RAM存储器执行Checkerboard测试。测试的对象是从`startAddress`起的`length`个字节。可变长度必须为4的倍数。

原型

```
CLASSBRESULT CLASSB_RAMCheckerboardTest (startAddress, length)
```

参数

- `startAddress`——被测试存储器的第一个地址（必须为偶数）
- `length`——被测试存储器的字节长度（必须为4字节的倍数）。

返回值

CLASSB_TEST_PASS（返回值 = 0）——测试成功完成

CLASSB_TEST_FAIL（返回值 != 0）——测试失败

备注

无。

源文件

```
classb_checkerboard.s
```

表8： **资源需求**

参数	需求
闪存	190个字节
堆栈	8000个指令周期/KB
中断	禁止24个指令周期

CLASSB_ClockTest

说明

该函数执行时钟测试，用于验证CPU时钟源是否在可接受的频率容差范围内工作。参考时钟（例如，外部晶振或工频）需连接一个定时器。必须通过应用程序代码将该定时器初始化为针对参考时钟脉冲进行计数。周期寄存器必须设置为最大值。必须在classb_config.h中使用编译时间选项CLASSB_CLOCK_TEST_TIMER_ADDRESS指定定时器/计数器寄存器的地址。classb_config.h文件中的编译选项CLASSB_CLOCK_TEST_TIME_MS定义测试时间——当时钟介于1-32 MHz之间且参考时钟介于50 Hz-33 kHz时，测试时间约为20 ms。测试期间禁止中断。

原型

CLASSB_ClockTest (uint32_t clockFrequency, uint32_t referenceFrequency, uint16_t tolerance)

参数

- clockFrequency——时钟源的频率
- referenceFrequency——参考时钟（例如电源线或辅助振荡器）的频率
- tolerance——最大有效频率容差，可以是1（0.1%）至100（10%）之间的值

返回值

CLASSB_TEST_PASS（返回值 = 0）——测试成功完成

CLASSB_TEST_FAIL（返回值 != 0）——测试失败

备注

无。

源文件

classb_clock.s

classb_clock.c

表9：资源需求

参数	需求
闪存	633个字节
执行时间	160000-640000个指令周期
中断	测试期间禁止

总结

本应用笔记介绍了如何实现IEC 60730标准提出的各种诊断措施。这些措施确保了B类受控设备的安全运行。另外，本应用笔记也介绍了B类安全软件库中提供的各种API。这些API可直接集成到最终用户的应用程序中以测试和验证控制器的关键功能，并旨在通过故障检测最大限度地提高应用程序的可靠性。当在dsPIC DSC或PIC MCU上执行时，这些API有助于应用程序满足IEC 60730标准的要求。

Microchip已经开发了B类安全软件库来帮助您实现安全软件程序。如果需要更多的支持，请联系您的Microchip销售代表或应用工程师。

参考资料

- IEC 60730 标准 “*Automatic Electrical Controls for Household and Similar Use*”，IEC 60730-1 第5版，2013年
- “*Essentials of Electronic Testing for Digital, Memory, and Mixed-Signal VLSI Circuits*”，作者：Bushnell M. 和 Agarwal V.，New York: Springer，2000年第一版，2005年第2次印刷修订版
- “*Memory Testing*”，作者：Wu C.
- “*RAM Fault Models and Memory Testing*”，Wu C.
- Suk D.S. 和 Reddy S.M. “*A March Test for Functional Faults in Semiconductor Random-Access Memories*”，IEEE Trans. Computers, Vol. C-30, No. 12, 1981, pp. 982-985

附录A： 源代码

软件许可协议

Microchip Technology Incorporated（以下简称“本公司”）在此提供的软件只面向本公司客户，并只用于本公司生产的产品。

本软件为本公司和/或其供应商所有，并受到适用的版权法保护。保留所有权利。用户在使用时如果违反上述限制，可能会依法受到刑事制裁，并可能由于违背本许可证的条款和条件而承担民事责任。

本软件按“原样”提供，不附有任何形式的保证，无论是明示的、暗示的或法定的，包括（但不限于）有关适销性和特定用途的暗示保证。在任何情况下，因任何原因产生的任何特殊的、附带的或间接的损失，本公司概不承担任何责任。

本应用笔记中提及的所有软件均以单个 zip 文件的形式提供。可从如下 Microchip 公司网站下载此压缩文件：

www.microchip.com/classb

附录B: IEC 60730-1表H.1 (第3版的H.11.12.7)

下表的复制获得了国际电工委员会 (IEC) 的允许。IEC 60730-1 第5版“版权所有 © 2013 IEC, 瑞士日内瓦。www.iec.ch”。

表B-1: 表H.1 (第3版的H.11.12.7) ——可接受的故障/错误应对措施^a

组件 ^b	故障/错误	软件类		可接受措施示例 ^{c d e}	定义
		B	C		
1. CPU 1.1 寄存器	卡住	rq		功能测试, 或者 周期性自检, 具体实现方式为: - 静态存储器测试, 或 - 带单个位冗余的字保护	H.2.16.5 H.2.16.6 H.2.19.6 H.2.19.8.2
	直流故障		rq	冗余CPU比较, 具体实现方式为: - 互换比较 - 独立硬件比较器、 内部错误检测或 带比较功能的冗余存储器, 或者 周期性自检, 具体实现方式为: - walkpat存储器测试 - Abraham测试 - 透明 GALPAT 测试: 带多个位冗余的字保护 或 静态存储器测试和 带单个位冗余的字保护	H.2.18.15 H.2.18.3 H.2.18.9 H.2.19.5 H.2.19.7 H.2.19.1 H.2.19.2.1 H.2.19.8.1 H.2.19.6 H.2.19.8.2
1.2 指令解码和执行	解码和 执行错误		rq	冗余CPU比较, 具体实现方式为: - 互换比较 - 独立硬件比较器、 内部错误检测或 使用等效类测试的周期性自检	H.2.18.15 H.2.18.3 H.2.18.9 H.2.18.5
1.3 程序计数器	卡住	rq		功能测试、 周期性自检、 程序序列独立时隙监视 或者 程序序列逻辑监视 周期性自检和监视, 具体实现方式为: - 独立时隙和逻辑 监视 - 内部错误检测, 或者 冗余功能通道比较, 具体实现方式为: - 互换比较 - 独立硬件比较器	H.2.16.5 H.2.16.6 H.2.18.10.4 H.2.18.10.2 H.2.16.7 H.2.18.10.3 H.2.18.9 H.2.18.15 H.2.18.3
	直流故障		rq		

CPU: 中央处理单元

rq: 指示的软件类需要明确故障范围。

a 表H.1按照H.11.12至H.11.12.2.12 (包含H.11.12和H.11.12.2.12) 的要求应用。

b 为了评估故障/错误, 一些组件划分为多个子功能。

c 对于表中的每个子功能, 软件类C的措施将覆盖软件类B的故障/错误。

d 通常认为一些可接受措施提供的保障等级高于标准提出的要求。

e 每个子功能有多种措施, 其中一些是备用措施。

f 必要时由制造商划分为多个子功能。

表B-1: 表H.1 (第3版的H.11.12.7) ——可接受的故障/错误应对措施^a (续)

组件 ^b	故障/错误	软件类		可接受措施示例 ^{c d e}	定义
		B	C		
1.4 寻址	直流故障		rq	冗余CPU比较, 具体实现方式为: - 互换比较 - 独立硬件比较器; 内部错误检测; 使用地址线测试模式的 周期性自检; 全总线冗余或 多个位总线奇偶校验	H.2.18.15 H.2.18.3 H.2.18.9 H.2.16.7 H.2.18.22 H.2.18.1.1 H.2.18.1.2
1.5 数据路径指令解码	直流故障和执行		rq	冗余CPU比较, 具体实现方式为: 互换比较、 独立硬件比较器、 内部错误检测、 使用测试模式的周期性自检、 数据冗余或 多个位总线奇偶校验	H.2.18.15 H.2.18.3 H.2.18.9 H.2.16.7 H.2.18.2.1 H.2.18.1.2
2. 中断处理和执行	无中断或中断过于频繁	rq		功能测试; 或者 时隙监视	H.2.16.5 H.2.18.10.4
	无中断或 与不同源相关的 中断过于频繁		rq	冗余功能通道比较, 具体实现方式为: 互换比较、 独立硬件比较器或 独立时隙和逻辑监视	H.2.18.15 H.2.18.3 H.2.18.10.3
3. 时钟	频率错误 (对于石英 同步时钟: 仅限谐波/ 次谐波)	rq	rq	频率监视或者 时隙监视 频率监视、 时隙监视或者 冗余功能通道比较, 具体实现方式为: - 互换比较 - 独立硬件比较器	H.2.18.10.1 H.2.18.10.4 H.2.18.10.1 H.2.18.10.4 H.2.18.15 H.2.18.3
CPU: 中央处理单元 rq: 指示的软件类需要明确故障范围。 a 表H.1按照H.11.12至H.11.12.2.12 (包含H.11.12和H.11.12.2.12) 的要求应用。 b 为了评估故障/错误, 一些组件划分为多个子功能。 c 对于表中的每个子功能, 软件类C的措施将覆盖软件类B的故障/错误。 d 通常认为一些可接受措施提供的保障等级高于标准提出的要求。 e 每个子功能有多种措施, 其中一些是备用措施。 f 必要时由制造商划分为多个子功能。					

表B-1: 表H.1 (第3版的H.11.12.7) ——可接受的故障/错误应对措施^a (续)

组件 ^b	故障/错误	软件类		可接受措施示例 ^{c d e}	定义
		B	C		
4. 存储器 4.1 不变存储器	所有单个位故障 占有信息错误的99.6%	rq	rq	周期性 修改校验和 、 多个校验和 或者 带单个位冗余的字保护 冗余CPU比较, 具体实现方式为: - 互换比较 - 独立硬件比较器 或 带比较功能的冗余存储器 , 或者周期性循环冗余校验, 具体实现方式为: - 单字 - 双字或 带多个位冗余的字保护	H.2.19.3.1 H.2.19.3.2 H.2.19.8.2 H.2.18.15 H.2.18.3 H.2.19.5 H.2.19.4.1 H.2.19.4.2 H.2.19.8.1
4.2 可变存储器	直流故障 直流故障和动态交叉线路	rq	rq	周期性 静态存储器测试 , 或者 带单个位冗余的字保护 冗余CPU比较, 具体实现方式为: - 互换比较 - 独立硬件比较器 或 带比较功能的冗余存储器 , 或者周期性自检, 具体实现方式为: - walkpat存储器测试 - Abraham测试 - 透明 GALP AT测试 或 带多个位冗余的字保护	H.2.19.6 H.2.19.8.2 H.2.18.15 H.2.18.3 H.2.19.5 H.2.19.7 H.2.19.1 H.2.19.2.1 H.2.19.8.1
4.3 寻址 (与可变存储器和不变存储器相关)	卡住 直流故障	rq	rq	带单个位冗余的字保护 (包含地址), 或者 冗余CPU比较, 具体实现方式为: - 互换比较 、 - 独立硬件比较器 或 全总线冗余测试模式 , 或者 周期性循环冗余校验, 具体实现方式为: - 单字 - 双字或 带多个位冗余的字保护 (包含地址)	H.2.19.18.2 H.2.18.15 H.2.18.3 H.2.18.1.1 H.2.18.22 H.2.19.4.1 H.2.19.4.2 H.2.19.8.1
CPU: 中央处理单元 rq: 指示的软件类需要明确故障范围。 a 表H.1按照H.11.12至H.11.12.2.12 (包含H.11.12和H.11.12.2.12) 的要求应用。 b 为了评估故障/错误, 一些组件划分为多个子功能。 c 对于表中的每个子功能, 软件类C的措施将覆盖软件类B的故障/错误。 d 通常认为一些可接受措施提供的保障等级高于标准提出的要求。 e 每个子功能有多种措施, 其中一些是备用措施。 f 必要时由制造商划分为多个子功能。					

表B-1: 表H.1 (第3版的H.11.12.7) ——可接受的故障/错误应对措施^a (续)

组件 ^b	故障/错误	软件类		可接受措施示例 ^{c d e}	定义
		B	C		
5. 内部数据路径					
5.1 数据	卡住 直流故障	rq	rq	带单个位冗余的字保护 冗余CPU比较, 具体实现方式为: - 互换比较 - 独立硬件比较器、 带多个位冗余的字保护 (包含地址)、数据冗余、 测试模式或 协议测试	H.2.19.8.2 H.2.18.15 H.2.18.3 H.2.19.8.1 H.2.18.2.1 H.2.18.22 H.2.18.14
5.2 寻址	地址错误 地址错误和 多种寻址	rq	rq	带单个位冗余的字保护 (包含地址) 冗余CPU比较, 具体实现方式为: - 互换比较 - 独立硬件比较器、 带多个位冗余的字保护(包含地址)、 全总线冗余或 测试模式(包含地址)	H.2.19.8.2 H.2.18.15 H.2.18.3 H.2.19.8.1 H.2.18.1.1 H.2.18.22
6 外部通信	汉明距离3	rq		带多个位冗余的字保护、 CRC——单字、 传输冗余或 协议测试	H.2.19.8.1 H.2.19.4.1 H.2.18.2.2 H.2.18.14
6.1 数据	汉明距离4		rq	CRC——双字、 数据冗余或者冗余功能通道比较, 具体实现方式为: - 互换比较 - 独立硬件比较器	H.2.19.4.2 H.2.18.2.1 H.2.18.15 H.2.18.3
6.2 寻址	地址错误 地址错误和多种 寻址	rq	rq	带多个位冗余的字保护(包含地址)、 CRC——单字(包含地址)、 传输冗余或 协议测试 CRC——双字(包含地址)、 数据和地址全总线冗余或 冗余通信通道比较, 具体实现方式为: - 互换比较 - 独立硬件比较器	H.2.19.8.1 H.2.19.4.1 H.2.18.2.2 H.2.18.14 H.2.19.4.2 H.2.18.1.1 H.2.18.15 H.2.18.3
CPU: 中央处理单元 rq: 指示的软件类需要明确故障范围。 a 表H.1按照H.11.12至H.11.12.2.12(包含H.11.12和H.11.12.2.12)的要求应用。 b 为了评估故障/错误, 一些组件划分为多个子功能。 c 对于表中的每个子功能, 软件类C的措施将覆盖软件类B的故障/错误。 d 通常认为一些可接受措施提供的保障等级高于标准提出的要求。 e 每个子功能有多种措施, 其中一些是备用措施。 f 必要时由制造商划分为多个子功能。					

表B-1: 表H.1 (第3版的H.11.12.7) ——可接受的故障/错误应对措施^a (续)

组件 ^b	故障/错误	软件类		可接受措施示例 ^{c d e}	定义
		B	C		
6.3 时序	时间点错误 序列错误	rq rq	rq rq	时隙监视、 调度传输 时隙和逻辑监视或 冗余通信通道比较, 具体实现方式为: - 互换比较 - 独立硬件比较器 逻辑监视、 时隙监视或 调度传输 (与时间点错误时的选项相同)	H.2.18.10.4 H.2.18.18 H.2.18.10.3 H.2.18.15 H.2.18.3 H.2.18.10.2 H.2.18.10.4 H.2.18.18
7. 输入/输出外设 7.1 数字I/O	故障 条件 (第H.27条 中指定)	rq	rq	合理性检查 冗余CPU比较, 具体实现方式为: - 互换比较 - 独立硬件比较器、 输入比较、 多个并行输出、 输出验证、 测试模式或 代码安全	H.2.18.13 H.2.18.15 H.2.18.3 H.2.18.8 H.2.18.11 H.2.18.12 H.2.18.22 H.2.18.2
7.2 模拟I/O 7.2.1 A/D转换器和 D/A转换器	故障 条件 (第H.27条 中指定)	rq	rq	合理性检查 冗余CPU比较, 具体实现方式为: - 互换比较 - 独立硬件比较器、 输入比较、 多个并行输出、 输出验证或 测试模式	H.2.18.13 H.2.18.15 H.2.18.3 H.2.18.8 H.2.18.11 H.2.18.12 H.2.18.22
7.2.2 模拟多路开关	寻址 错误	rq	rq	合理性检查 冗余CPU比较, 具体实现方式为: - 互换比较 - 独立硬件比较器、 输入比较或 测试模式	H.2.18.13 H.2.18.15 H.2.18.3 H.2.18.8 H.2.18.22
8. 监视器件和比较器	任一输出超出静态 和动态功能规范		rq	测试监视、 冗余监视和比较或 错误识别方法	H.2.18.21 H.2.18.17 H.2.18.6

CPU: 中央处理单元

rq: 指示的软件类需要明确故障范围。

a 表H.1按照H.11.12至H.11.12.2.12 (包含H.11.12和H.11.12.2.12) 的要求应用。

b 为了评估故障/错误, 一些组件划分为多个子功能。

c 对于表中的每个子功能, 软件类C的措施将覆盖软件类B的故障/错误。

d 通常认为一些可接受措施提供的保障等级高于标准提出的要求。

e 每个子功能有多种措施, 其中一些是备用措施。

f 必要时由制造商划分为多个子功能。

表B-1: 表H.1 (第3版的H.11.12.7) ——可接受的故障/错误应对措施^a (续)

组件 ^b	故障/错误	软件类		可接受措施示例 ^{c d e}	定义
		B	C		
9. 定制芯片 ^f 例如, ASIC、 GAL和门阵列	任一输出超出 静态和 动态功能规范	rq	rq	周期性自检 周期性自检和监视、 具有比较功能的双通道（多种选择）或 错误识别方法	H.2.16.6 H.2.16.7 H.2.16.2 H.2.18.6
CPU: 中央处理单元 rq: 指示的软件类需要明确故障范围。 a 表H.1按照H.11.12至H.11.12.2.12（包含H.11.12和H.11.12.2.12）的要求应用。 b 为了评估故障/错误, 一些组件划分为多个子功能。 c 对于表中的每个子功能, 软件类C的措施将覆盖软件类B的故障/错误。 d 通常认为一些可接受措施提供的保障等级高于标准提出的要求。 e 每个子功能有多种措施, 其中一些是备用措施。 f 必要时由制造商划分为多个子功能。					

附录C： 版本历史

版本A（2014年8月）

本应用笔记的初始版本。

注:

请注意以下有关 Microchip 产品代码保护功能的要点：

- Microchip 的产品均达到 Microchip 数据手册中所述的技术规范。
- Microchip 确信：在正常使用且符合工作规范的情况下，Microchip 系列产品非常安全。
- Microchip 注重并积极保护其知识产权。严禁任何试图破坏 Microchip 产品代码保护功能的行为，这种行为可能会违反《数字千年版权法案》（Digital Millennium Copyright Act）。
- Microchip 或任何其他半导体厂商均无法保证其代码的安全性。代码保护并不意味着我们保证产品是“牢不可破”的。代码保护功能处于持续发展。Microchip 承诺将不断改进产品的代码保护功能。

提供本文档的中文版本仅为了便于理解。请勿忽视文档中包含的英文部分，因为其中提供了有关 Microchip 产品性能和使用情况的有用信息。Microchip Technology Inc. 及其分公司和相关公司、各级主管与员工及事务代理机构对译文中可能存在的任何差错不承担任何责任。建议参考 Microchip Technology Inc. 的英文原版文档。

本出版物及其提供的信息仅适用于 Microchip 产品，包括设计、测试以及将 Microchip 产品集成到您的应用中。以其他方式使用这些信息都将被视为违反条款。本出版物中的器件应用信息仅为您提供便利，将来可能会发生更新。如需额外的支持，请联系当地的 Microchip 销售办事处，或访问 <https://www.microchip.com/en-us/support/design-help/client-supportservices>。

Microchip “按原样”提供这些信息。Microchip 对这些信息不作任何明示或暗示、书面或口头、法定或其他形式的声明或担保，包括但不限于针对非侵权性、适销性和特定用途的适用性的暗示担保，或针对其使用情况、质量或性能的担保。

在任何情况下，对于因这些信息或使用这些信息而产生的任何间接的、特殊的、惩罚性的、偶然的或间接的损失、损害或任何类型的开销，Microchip 概不承担任何责任，即使 Microchip 已被告知可能发生损害或损害可以预见。在法律允许的最大范围内，对于因这些信息或使用这些信息而产生的所有索赔，Microchip 在任何情况下所承担的全部责任均不超出您为获得这些信息向 Microchip 直接支付的金额（如有）。如果将 Microchip 器件用于生命维持和 / 或生命安全应用，一切风险由买方自负。买方同意在由此引发任何一切损害、索赔、诉讼或费用时，会维护和保障 Microchip 免于承担法律责任。除非另外声明，在 Microchip 知识产权保护下，不得暗中以其他方式转让任何许可证。

有关 Microchip 质量管理体系的更多信息，请访问 www.microchip.com/quality。

商标

Microchip 的名称和徽标组合、Microchip 徽标、AdapteC、AnyRate、AVR、AVR 徽标、AVR Freaks、BesTime、BitCloud、CryptoMemory、CryptoRF、dsPIC、flexPWR、HELD0、IGLOO、JukeBlox、KeeLoq、Kleer、LANCheck、LinkMD、maXStylus、maXTouch、MedialB、megaAVR、Microsemi、Microsemi 徽标、MOST、MOST 徽标、MPLAB、OptoLyzer、PIC、picoPower、PICSTART、PIC32 徽标、PolarFire、Prochip Designer、QTouch、SAM-BA、SenGenuity、SpyNIC、SST、SST 徽标、SuperFlash、Symmetricom、SyncServer、Tachyon、TimeSource、tinyAVR、UNI/O、Vectron 及 XMEGA 均为 Microchip Technology Incorporated 在美国和其他国家或地区的注册商标。

AgileSwitch、APT、ClockWorks、The Embedded Control Solutions Company、EtherSynch、Flashtec、Hyper Speed Control、HyperLight Load、IntelliMOS、Libero、motorBench、mTouch、Powermite 3、Precision Edge、ProASIC、ProASIC Plus、ProASIC Plus 徽标、Quiet-Wire、SmartFusion、SyncWorld、Temux、TimeCesium、TimeHub、TimePictra、TimeProvider、TrueTime、WinPath 和 ZL 均为 Microchip Technology Incorporated 在美国的注册商标。

Adjacent Key Suppression、AKS、Analog-for-the-Digital Age、Any Capacitor、AnyIn、AnyOut、Augmented Switching、BlueSky、BodyCom、CodeGuard、CryptoAuthentication、CryptoAutomotive、CryptoCompanion、CryptoController、dsPICDEM、dsPICDEM.net、Dynamic Average Matching、DAM、ECAN、Espresso T1S、EtherGREEN、GridTime、IdealBridge、In-Circuit Serial Programming、ICSP、INICnet、Intelligent Paralleling、Inter-Chip Connectivity、JitterBlocker、Knob-on-Display、maxCrypto、maxView、memBrain、MindI、MiWi、MPASM、MPF、MPLAB Certified 徽标、MPLIB、MPLINK、MultiTRAK、NetDetach、NVM Express、NVMe、Omniscient Code Generation、PICDEM、PICDEM.net、PICKit、PICKtail、PowerSmart、PureSilicon、QMatrix、REAL ICE、Ripple Blocker、RTAX、RTG4、SAM-ICE、Serial Quad I/O、simpleMAP、SimpliPHY、SmartBuffer、SmartHLS、SMART-I.S.、storClad、SQL、SuperSwitcher、SuperSwitcher II、Switchtec、SynchroPHY、Total Endurance、TSHARC、USBCheck、VariSense、VectorBlox、VeriPHY、ViewSpan、WiperLock、XpressConnect 和 ZENA 均为 Microchip Technology Incorporated 在美国和其他国家或地区的商标。

SQTP 为 Microchip Technology Incorporated 在美国的服务标记。

AdapteC 徽标、Frequency on Demand、Silicon Storage Technology、Symmcom 和 Trusted Time 均为 Microchip Technology Inc. 在除美国外的国家或地区的注册商标。

GestIC 为 Microchip Technology Inc. 的子公司 Microchip Technology Germany II GmbH & Co. KG 在除美国外的国家或地区的注册商标。

在此提及的所有其他商标均为各持有公司所有。

© 2022, Microchip Technology Incorporated 及其子公司版权所有。
ISBN:

全球销售及服务网点

美洲

公司总部 **Corporate Office**
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 1-480-792-7200
Fax: 1-480-792-7277

技术支持:
<http://www.microchip.com/support>

网址: www.microchip.com

亚特兰大 Atlanta
Duluth, GA
Tel: 1-678-957-9614
Fax: 1-678-957-1455

奥斯汀 Austin, TX
Tel: 1-512-257-3370

波士顿 Boston
Westborough, MA
Tel: 1-774-760-0087
Fax: 1-774-760-0088

芝加哥 Chicago
Itasca, IL
Tel: 1-630-285-0071
Fax: 1-630-285-0075

达拉斯 Dallas
Addison, TX
Tel: 1-972-818-7423
Fax: 1-972-818-2924

底特律 Detroit
Novi, MI
Tel: 1-248-848-4000

休斯敦 Houston, TX
Tel: 1-281-894-5983

印第安纳波利斯 Indianapolis
Noblesville, IN
Tel: 1-317-773-8323
Fax: 1-317-773-5453
Tel: 1-317-536-2380

洛杉矶 Los Angeles
Mission Viejo, CA
Tel: 1-949-462-9523
Fax: 1-949-462-9608
Tel: 1-951-273-7800

罗利 Raleigh, NC
Tel: 1-919-844-7510

纽约 New York, NY
Tel: 1-631-435-6000

圣何塞 San Jose, CA
Tel: 1-408-735-9110
Tel: 1-408-436-4270

加拿大多伦多 Toronto
Tel: 1-905-695-1980
Fax: 1-905-695-2078

亚太地区

中国 - 北京
Tel: 86-10-8569-7000

中国 - 成都
Tel: 86-28-8665-5511

中国 - 重庆
Tel: 86-23-8980-9588

中国 - 东莞
Tel: 86-769-8702-9880

中国 - 广州
Tel: 86-20-8755-8029

中国 - 杭州
Tel: 86-571-8792-8115

中国 - 南京
Tel: 86-25-8473-2460

中国 - 青岛
Tel: 86-532-8502-7355

中国 - 上海
Tel: 86-21-3326-8000

中国 - 沈阳
Tel: 86-24-2334-2829

中国 - 深圳
Tel: 86-755-8864-2200

中国 - 苏州
Tel: 86-186-6233-1526

中国 - 武汉
Tel: 86-27-5980-5300

中国 - 西安
Tel: 86-29-8833-7252

中国 - 厦门
Tel: 86-592-238-8138

中国 - 香港特别行政区
Tel: 852-2943-5100

中国 - 珠海
Tel: 86-756-321-0040

台湾地区 - 高雄
Tel: 886-7-213-7830

台湾地区 - 台北
Tel: 886-2-2508-8600

台湾地区 - 新竹
Tel: 886-3-577-8366

亚太地区

澳大利亚 Australia - Sydney
Tel: 61-2-9868-6733

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

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

印度 India - Pune
Tel: 91-20-4121-0141

日本 Japan - Osaka
Tel: 81-6-6152-7160

日本 Japan - Tokyo
Tel: 81-3-6880-3770

韩国 Korea - Daegu
Tel: 82-53-744-4301

韩国 Korea - Seoul
Tel: 82-2-554-7200

马来西亚 Malaysia - Kuala Lumpur
Tel: 60-3-7651-7906

马来西亚 Malaysia - Penang
Tel: 60-4-227-8870

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

新加坡 Singapore
Tel: 65-6334-8870

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

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

欧洲

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

丹麦 Denmark - Copenhagen
Tel: 45-4485-5910
Fax: 45-4485-2829

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

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

德国 Germany - Garching
Tel: 49-8931-9700

德国 Germany - Haan
Tel: 49-2129-3766400

德国 Germany - Heilbronn
Tel: 49-7131-72400

德国 Germany - Karlsruhe
Tel: 49-721-625370

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

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

以色列 Israel - Ra'anana
Tel: 972-9-744-7705

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

意大利 Italy - Padova
Tel: 39-049-7625286

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

挪威 Norway - Trondheim
Tel: 47-7288-4388

波兰 Poland - Warsaw
Tel: 48-22-3325737

罗马尼亚 Romania - Bucharest
Tel: 40-21-407-87-50

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

瑞典 Sweden - Gothenburg
Tel: 46-31-704-60-40

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

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