

---

---

## I<sup>2</sup>C 从模式

---

---

### 简介

作者：Christopher Best, Microchip Technology Inc.

集成电路间总线（常称为 I<sup>2</sup>C）是一种同步双线双向串行通信总线。I<sup>2</sup>C 模块可用于与其他 I<sup>2</sup>C 兼容的 EEPROM、显示驱动器、传感器或其他单片机器件进行通信。本技术简介将简要介绍从模式下独立 I<sup>2</sup>C 模块的特性和功能。不要将独立 I<sup>2</sup>C 模块与传统的主同步串行端口（Master Synchronous Serial Port, MSSP）混淆，后者包含 I<sup>2</sup>C 和串行外设接口（Serial Peripheral Interface, SPI）功能。

### I<sup>2</sup>C 模块的模式和特性

I<sup>2</sup>C 模块具有以下工作模式和特性：

- 主模式
- 包含字节无应答的从模式
- 多主器件模式
- 专用的接收和发送缓冲区
- 最多四个专用从地址寄存器<sup>(1)</sup>
- 7 位和 10 位寻址（含掩码）
- 广播呼叫寻址
- 中断
- 总线冲突检测
- 基于可编程源的总线超时检测
- SDA 保持时间选择
- 可编程总线空闲时间选择
- I<sup>2</sup>C、SMBus 2.0 和 SMBus 3.0 输入阈值选择
- 支持直接存储器访问（Direct Memory Access, DMA）<sup>(2)</sup>

### 注：

1. 只有在 7 位寻址模式下才支持四个专用从寄存器。在 10 位寻址模式下，仅支持两个专用地址寄存器。
2. 并非所有器件都提供直接存储器访问（DMA）。要确定 DMA 是否可用，请参见器件数据手册。

## 目录

简介.....	1
1. I <sup>2</sup> C 规范.....	4
2. I <sup>2</sup> C 模块概述.....	6
2.1. 专用发送/接收缓冲区.....	6
2.2. 地址缓冲区和寄存器.....	7
2.3. 启动条件.....	9
2.4. 重复启动条件.....	9
2.5. 应答 ( $\overline{\text{ACK}}$ ) /无应答 (NACK) 序列.....	10
2.6. 停止条件.....	12
2.7. SDA 和 SCL 引脚.....	12
2.8. 总线超时.....	13
2.9. 数据字节计数.....	14
2.10. 时钟延长.....	15
3. 中断.....	17
4. I <sup>2</sup> C 从模式操作.....	19
5. 从模式配置和操作.....	20
5.1. 初始化.....	20
6. 从模式发送.....	23
6.1. 从模式发送 (7 位寻址) .....	23
6.2. 从模式发送 (10 位寻址) .....	23
7. 从模式接收.....	26
7.1. 从模式接收 (7 位寻址) .....	26
7.2. 从模式接收 (10 位寻址) .....	27
8. 外部上拉电阻选择.....	29
9. 结论.....	30
Microchip 网站.....	31
变更通知客户服务.....	31
客户支持.....	31
Microchip 器件代码保护功能.....	31
法律声明.....	32
商标.....	32

质量管理体系..... 33

全球销售及服务网点..... 34

## 1. I<sup>2</sup>C 规范

I<sup>2</sup>C 规范由 Phillips Semiconductors 制定，用于在连接到双线总线的器件之间进行通信。Phillips 意识到消费类电子、工业电子和电信设计之间存在许多相似之处。不同的设计通常包含类似的元件，例如模数转换器（Analog-to-Digital Converter, ADC）、LCD 或 EEPROM，因而 Phillips 决定创建通信总线，使连接到总线的器件可以互传数据，从而简化系统设计并最大程度提高硬件效率。这样，设计人员便可使用来自多个制造商的器件，或者在多个设计中使用同一个器件。该规范还通过构建方案解决了接口问题，此方案规定任何 I<sup>2</sup>C 器件都可以与其他 I<sup>2</sup>C 器件通信，不需要更改器件的硬件或固件，现已作为行业标准。

I<sup>2</sup>C 规范将总线定义为双线双向通信方案。一条线路承载串行数据（SDA），一条线路承载串行时钟（SCL）。每个 I<sup>2</sup>C 器件都具有惟一地址，长度为 7 位或 10 位。I<sup>2</sup>C 器件可用作总线主器件和/或总线从器件，具体取决于器件和应用。该规范定义了数据传输速率，如下所述：

- 标准模式——传输速率最高 100 kbps
- 快速模式——传输速率最高 400 kbps
- 增强型快速模式——传输速率最高 1 Mbps
- 高速模式——传输速率最高 3.4 Mbps

Microchip 的 I<sup>2</sup>C 模块实现了支持标准模式、快速模式和增强型快速模式的主从硬件。本技术简介通篇引用 I<sup>2</sup>C 规范，以便读者理解 I<sup>2</sup>C 模块和 I<sup>2</sup>C 规范。

### I<sup>2</sup>C 协议术语

为正确理解规范中使用的语言，下面列出了规范中常用且本文通篇提及的术语：

表 1-1. I<sup>2</sup>C 总线术语

术语	说明
发送器	将数据移出到总线上的器件
接收器	从总线上移入数据的器件
主器件	启动数据传输、产生时钟信号和终止发送的器件
从器件	主器件寻址到的器件
多主器件	有多个器件可以启动数据传输的总线
仲裁	确保一次只有一个主器件控制总线的过程
同步	用于将总线上两个或更多器件的时钟进行同步的过程
空闲	SDA 和 SCK 线都处于逻辑高电平状态；总线上无活动
工作	每当有一个或多个主器件在控制总线时
地址从器件	已接收到匹配地址并且正在由主器件提供时钟的从器件
匹配地址	从器件接收到的与存储在 I <sup>2</sup> CxADR 寄存器之一中的值相匹配的地址字节
写请求	主器件发送 $R/\overline{W}$ 位清零的地址字节；主器件要向从器件写入数据
读请求	主器件发送 $R/\overline{W}$ 位置 1 的地址字节；主器件要从从器件接收数据
时钟延长	器件将时钟线保持为低电平以暂停通信
总线冲突	SDA 上的预期数据为逻辑高电平但采样到的数据为逻辑低电平

---

---

..... (续)

术语	说明
总线超时	总线上的器件保持对总线的控制的时间超过指定时间

## 2. I<sup>2</sup>C 模块概述

I<sup>2</sup>C 模块使用双线总线在单片机和其他 I<sup>2</sup>C 兼容器件之间提供同步串行接口。串行时钟（SCL）和串行数据（SDA）这两个信号连接是双向漏极开路线，它们均需要通过上拉电阻连接到电源电压。将线拉至地被视为逻辑 0，而将线悬空被视为逻辑 1。



**重要：** 逻辑 0（低电平）和逻辑 1（高电平）的电压值不固定，取决于总线电源电压。

根据 I<sup>2</sup>C 规范，逻辑输入低电平的电压值最高为  $V_{DD}$  的 30% ( $V_{IL} \leq 0.3 V_{DD}$ )，逻辑输入高电平的电压值为  $V_{DD}$  的 70% 至 100% ( $V_{IH} \geq 0.7 V_{DD}$ )。一些传统器件可能使用先前定义的固定电压  $V_{IL} = 1.5V$ ， $V_{IH} = 3.0V$ ，但所有新 I<sup>2</sup>C 兼容器件均需使用 30/70% 规范值。

所有 I<sup>2</sup>C 通信都使用 8 位数据字和 1 位应答序列来执行。总线上的所有事务均由主器件启动和终止。根据数据的传输方向，I<sup>2</sup>C 模块执行以下四种主要操作：

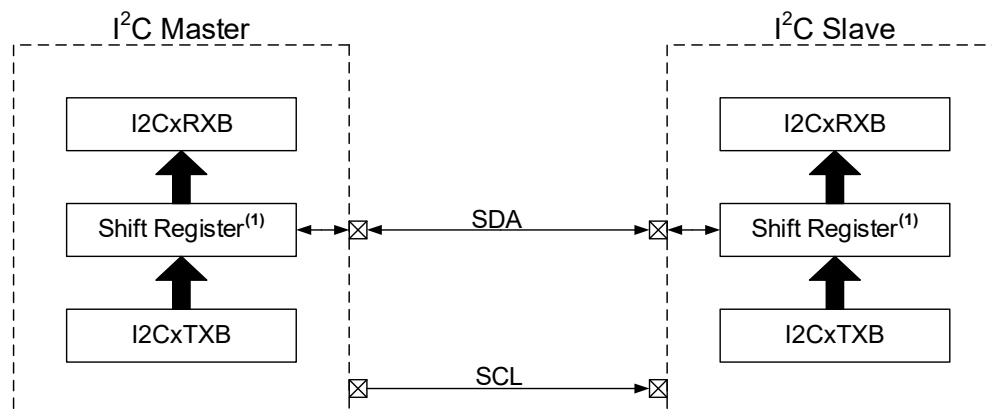
- 主发送——主器件向从器件发送数据
- 主接收——主器件从从器件接收数据
- 从发送——从器件向主器件发送数据
- 从接收——从器件从主器件接收数据

I<sup>2</sup>C 接口支持多主器件总线，即，一条总线上可存在多个主器件。主器件可通过在总线上发送惟一地址来选择从器件。当地址与从器件地址匹配时，从器件以应答序列（**ACK**）响应，主器件和该从器件之间可以开始进行通信。连接到总线的所有其他器件必须忽略对其不适用的事务。

### 2.1 专用发送/接收缓冲区

I<sup>2</sup>C 模块有两个专用数据缓冲区，一个用于发送（I2CxTXB），另一个用于接收（I2CxRXB）（见图 2-1）。

图 2-1. I<sup>2</sup>C 发送（I2CxTXB）和接收（I2CxRXB）缓冲区



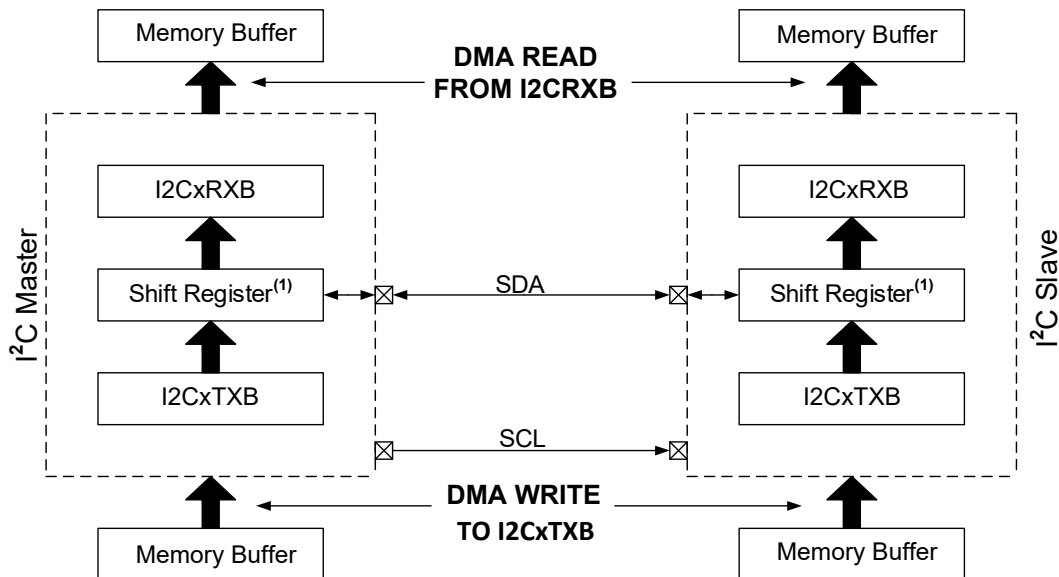
注:

1. 用户无法访问移位寄存器。

发送缓冲区 I2CxTXB 从软件或直接存储器访问 (DMA) 模块装入数据 (见图 2-2)。发送开始时, 装入 I2CxTXB 的数据被移入发送移位寄存器, 然后移出到总线上。当发送缓冲区空状态 (TXBE) 位置 1 (TXBE = 1, 表示缓冲区为空) 时, 可装入 I2CxTXB。当缓冲区为空且 I2CxCNT 寄存器不等于 0 时, I2C 发送中断标志 (I2CxTXIF) 位置 1, 如果 I2C 发送中断允许 (I2CxTXIE) 位置 1, 则通用 I2C 中断标志 (I2CxIF) 也将置 1。将新的数据字节装入 I2CxTXB 会清零 I2CxTXIF 标志位。如果用户软件尝试在 I2CxTXB 已满时向其中装入数据, 则发送写错误状态 (TXWE) 位将置 1, 会生成 NACK, 并忽略新数据。如果 TXWE 位置 1, 则软件必须先清零该位, 然后再尝试重新装入缓冲区。

接收缓冲区 I2CxRXB 保存从接收移位寄存器移入的一个数据字节。用户软件或 DMA 可通过 I2CxRXB 寄存器读取该字节 (见图 2-2)。接收到新字节后, 接收缓冲区满状态 (RXBF) 位和 I2C 接收中断标志 (I2CxRXIF) 位将置 1, 如果 I2C 接收中断允许 (I2CxRXIE) 位置 1, 则通用 I2CxIF 也将置 1。读取 I2CxRXB 会清零 RXBF 和 I2CxRXIF。如果在缓冲区为空 (RXBF = 0) 时进行读取, 则接收读错误状态 (RXRE) 位将置 1, 并且模块会生成 NACK。用户软件必须清零 RXRE 位才能恢复正常工作。

图 2-2. 使用 DMA 的 I<sup>2</sup>C 发送/接收缓冲区



注:

1. 用户无法访问移位寄存器。

通过将 I2CxSTAT1 寄存器的清除缓冲区 (CLRBF) 位置 1, 可以清除发送和接收缓冲区, 同时会清零 I2CxTXIF 和 I2CxRXIF 中断标志。

## 2.2 地址缓冲区和寄存器

I<sup>2</sup>C 模块有两个地址缓冲区寄存器, 即 I2CxADB0 和 I2CxADB1, 可用作从模式下的接收缓冲区 (见表 2-2)。它与 MSSP 模块的不同之处在于, MSSP 模块仅使用 SSPBUF 来接收或发送地址 (或数据)。地

址缓冲区通过地址缓冲区禁止 (Address Buffer Disable, ABD) 位使能。ABD 清零时, 将使能地址缓冲区; ABD 置 1 时, 将禁止地址缓冲区。

当 ABD 位清零并处于 7 位寻址模式时, I2CxADB0 将装入匹配的接收从地址和  $\overline{R/W}$  位, I2CxADB1 未使用。在 10 位寻址模式下, I2CxADB0 将装入匹配接收地址的低 8 位, I2CxADB1 将装入匹配地址的高 8 位, 其中包括  $\overline{R/W}$  位。

当 ABD 位置 1 并处于 7 位寻址模式时, I2CxRXB 将装入匹配的接收从地址和  $\overline{R/W}$  位, I2CxADB0/1 均不使用。在 10 位寻址模式下, I2CxRXB 将装入高位和低位匹配地址字节。在这种情况下, 用户软件必须先读取 I2CxRXB, 然后接收移位寄存器才会向 I2CxRXB 装入匹配的低地址字节。在 10 位寻址模式下, I2CxADB0/1 也不使用。

表 2-1. 从模式的地址缓冲区方向

模式	MODE[2:0]	I2CADB0	I2CADB1
从模式 (7 位)	000	RX	未使用
	001		
从模式 (10 位)	010	RX	RX
	011		

I<sup>2</sup>C 模块有四个额外的从模式地址寄存器, 即 I2CxADR0/1/2/3。这些寄存器最多可容纳 4 个 7 位从地址或 2 个 10 位从地址。I2CxADR0/1/2/3 寄存器也可用于形成屏蔽寄存器 (见表 2-2)。

在 7 位寻址模式下, 所有四个寄存器都可用于存储单独的从地址。发送中接收的第一个字节 (地址字节) 单独与 I2CxADR 寄存器中的各个值进行比较。接收地址字节的 LSb ( $\overline{R/W}$  位) 不用于确定匹配。如果需要地址掩码, 则 I2CxADR1 保存用于对装入 I2CxADR0 中的地址进行屏蔽的值, I2CxADR3 保存用于对 I2CxADR2 进行屏蔽的值。

表 2-2. 从模式地址寄存器

模式	MODE[2:0]	I2CADR0	I2CADR1	I2CADR2	I2CADR3
7 位	000	7 位地址	7 位地址	7 位地址	7 位地址
7 位 (带掩码)	001	7 位地址	I2CxADR0 的 7 位掩码	7 位地址	I2CxADR2 的 7 位掩码
10 位	010	低地址字节	高地址字节	低地址字节	高地址字节
10 位 (带掩码)	011	低地址字节	高地址字节	低地址掩码	高地址掩码

在 10 位寻址模式下, I2CxADR0 和 I2CxADR1 组合用来保存 10 位从地址, I2CxADR2 和 I2CxADR3 组合用来保存另一个 10 位从地址。I2CxADR0 和 I2CxADR2 保存 10 位地址的低 8 位, 而 I2CxADR1 和 I2CxADR3 保存 10 位地址的高 2 位、 $\overline{R/W}$  位和 5 位 11110 代码 (分配给地址高字节的高 5 位)。如果需要地址掩码, 则 I2CxADR0 和 I2CxADR1 组合用来形成 10 位从地址, I2CxADR2 和 I2CxADR3 组合用来形成 10 位地址掩码。地址掩码可用于忽略装入高地址字节的最高有效位中的 5 位地址代码。



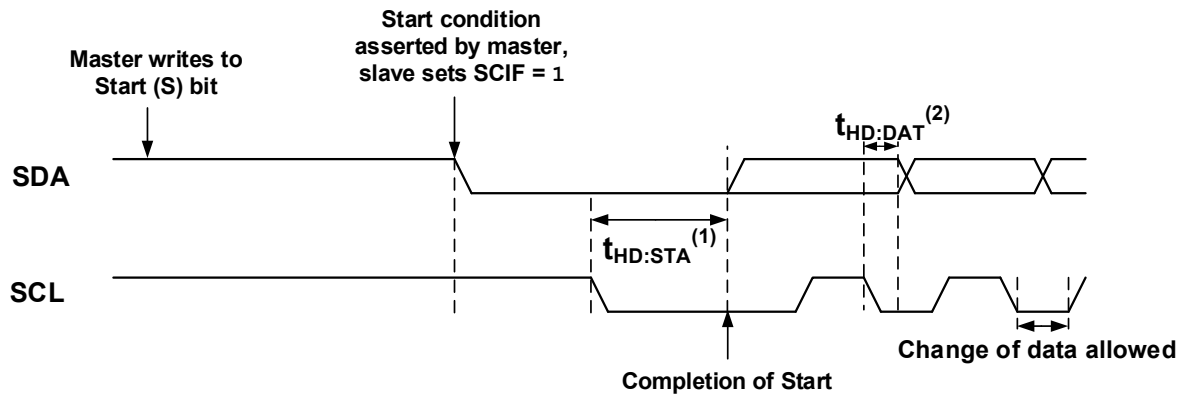
**重要:** 11110 代码由 I<sup>2</sup>C 规范指定, 但 Microchip 未严格执行。用户必须确保将正确的位值装入 10 位高地址字节中。如果主器件已将该特定代码包含在其要发送的地址中, 则从器件也必须将这些位包含在从地址中。



## 2.3 启动条件

I<sup>2</sup>C 规范将启动条件定义为在 SCL 线为空闲状态时，SDA 线从空闲状态（逻辑高电平）到有效状态（逻辑低电平）的转换（见 图 2-3）。启动条件始终由主器件发出，指示发送开始。

图 2-3. 启动条件



注:

1. 有关启动条件保持时间参数的信息，请参见器件数据手册。
2. SDA 保持时间通过 SDAHT[1:0]位配置。

根据 I<sup>2</sup>C 规范，出现启动条件时不能发生总线冲突。模块硬件使用总线空闲（BFRE）位来指示总线的状态。总线空闲时间选择（BFRET[1:0]）位定义在 BFRE 位置为有效之前主硬件必须检测到的总线空闲的 I<sup>2</sup>C 时钟周期数。BFRE 位置 1（BFRE = 1）时，视为总线处于空闲状态，主器件可发出启动条件。如果总线上有多个主器件（多主器件模式），并且它们尝试同时发出启动条件，则在通信的寻址阶段将发生总线冲突。

主器件发出启动条件时，从硬件会将启动条件中断标志（SCIF）置 1，如果启动条件中断允许（SCIE）位置 1，则通用 I2C 中断标志（I2CxIF）位也会置 1。从软件必须清零 SCIF 才能清零 I2CxIF 和继续正常工作。

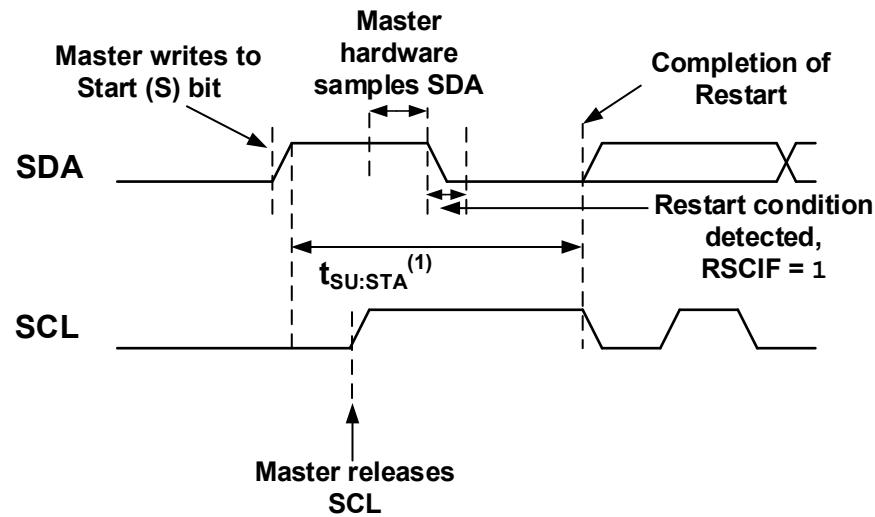
## 2.4 重复启动条件

重复启动条件与启动条件相同。如果主器件要在完成当前数据传输之后保持对总线的控制，主器件可发出重复启动条件（而非停止条件）。重复启动条件对从器件产生的影响与启动条件相同，即复位所有从器件逻辑并准备接收地址。重复启动条件始终由主器件发出。

当重复启动使能（RSEN）位置 1，I2CxCNT 为 0，且主硬件或用户软件将启动位置 1 时，会发生重复启动条件。

当启动位置 1 时，主硬件会释放 SDA（SDA 悬空为高电平）并持续  $T_{SCL}/2$ 。然后，硬件释放 SCL 并持续  $T_{SCL}/2$ ，同时采样 SDA。如果采样到 SDA 为低电平（而 SCL 为高电平），则表明已发生总线冲突，此时会将总线冲突检测中断标志（BCLIF）位置 1 并将主硬件置于空闲状态。如果采样到 SDA 为高电平（而 SCL 也为高电平），则主硬件会发出启动条件。在总线上检测到重复启动条件后，从硬件会将重复启动条件中断标志（RSCIF）位置 1。更多详细信息，请参见 图 2-4。

图 2-4. 重复启动条件



注:

1. 有关重复启动条件建立时间的信息, 请参见器件数据手册。

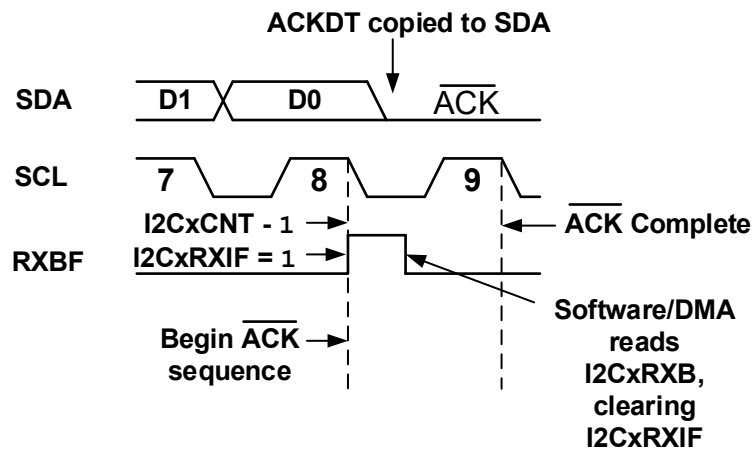
## 2.5 应答 ( $\overline{\text{ACK}}$ ) / 无应答 ( $\text{NACK}$ ) 序列

I<sup>2</sup>C 规范将应答序列 ( $\overline{\text{ACK}}$ ) 定义为成功传输任何字节时, 第 9 个 SCL 脉冲期间 SDA 线的逻辑低电平状态。在此期间, 发送器必须将 SDA 线的控制权让与接收器。然后, 接收器必须将 SDA 线拉为低电平, 并使其在第 9 个 SCL 脉冲的高电平期间保持低电平。

当接收器成功接收到匹配的地址字节或有效数据字节时, 它将在第 9 个 SCL 脉冲期间将 SDA 线拉为低电平, 以通知发送器其已成功接收到信息并准备好接收下一个字节。

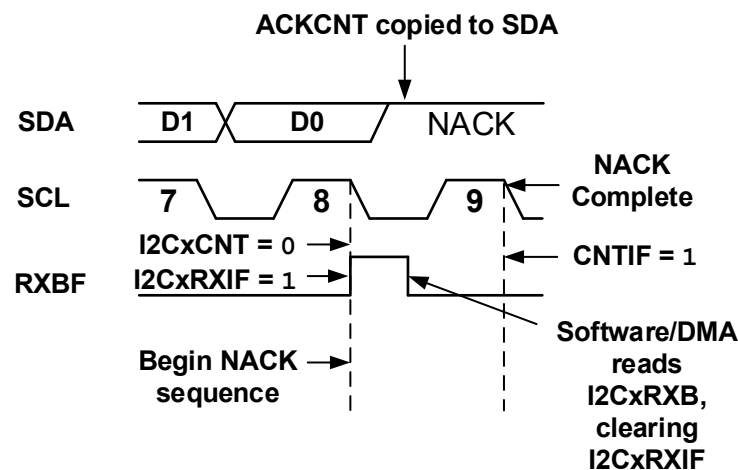
在接收到地址/数据字节后, 硬件会自动使能应答序列。在 SCL 的第 8 个下降沿, 应答数据 (ACKDT) 位或应答计数结束 (ACKCNT) 位的内容将复制到 SDA 输出。I2CxCNT 不为 0 时, ACKDT 位的值将复制到 SDA。I2CxCNT 为 0 时, ACKCNT 位的值将复制到 SDA。在大多数应用中, ACKDT 的值应为 0, 表示  $\overline{\text{ACK}}$  (见图 2-5)。

图 2-5. 从器件  $\overline{\text{ACK}}$  ( $\text{I2CxCNT} \neq 0$ )



如果 SDA 线在第 9 个 SCL 脉冲期间保持逻辑高电平，则将定义为无应答 (NACK) 序列 (见图 2-6)。

图 2-6. 从器件 NACK ( $\text{I2CxCNT} = 0$ )



当发生以下任何条件时，都会生成 NACK:

- 用于发送地址的总线上不存在从器件
- 接收器忙，未准备好进行通信
- 接收器获取其无法理解的数据或命令
- 接收器无法再接收任何数据
- 主接收器已接收到所请求的数据并准备终止发送
- 发生了 I<sup>2</sup>C 错误条件
- I2CxCNT 寄存器已达到值 0，ACKCNT 置 1 ( $\text{ACKNT} = 1$ )。

然后，主器件可决定是生成停止条件以终止传输，还是发出重复启动条件以保持对总线的控制并开始新的传输。

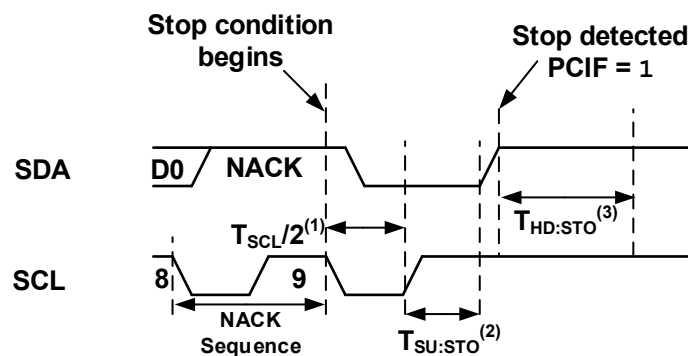
## 2.6 停止条件

I<sup>2</sup>C 规范将停止条件定义为在 SCL 线为空闲状态时，SDA 线从有效状态到空闲状态的转换。事务完成并且主器件准备好释放对总线的控制时，或者发生总线超时时，主器件将发出停止条件。如果在 SCL 线为高电平时，SDA 线先转换为低电平再转换为高电平，则会忽略停止条件，接收器将检测到启动/重复启动条件（见图 2-7）。



**重要：** 必须至少出现一个 SCL 低电平周期，停止条件才有效。

图 2-7. 停止条件



注：

1. 必须至少出现一个 SCL 低电平周期，停止条件才有效。
2. 有关停止条件建立时间的信息，请参见器件数据手册。
3. 有关停止条件保持时间的信息，请参见器件数据手册。

在所发送/接收 I<sup>2</sup>C 数据包的最后一个字节的  $\overline{\text{ACK}}/\text{NACK}$  序列之后，硬件会将 SCL 线拉为低电平并持续  $T_{\text{SCL}}/2$ ，然后释放 SCL。硬件将对 SCL 进行采样以确保逻辑高电平。然后释放 SDA，当 SCL 为高电平时，如果 SDA 从低电平转换为高电平，则会导致停止条件中断标志（PCIF）位置 1。

## 2.7 SDA 和 SCL 引脚

I<sup>2</sup>C 模块使用串行数据（SDA）和串行时钟（SCL）引脚来控制 I<sup>2</sup>C 总线。与之前版本的 MSSP 不同，必须通过设置相应端口的漏极开路控制寄存器（ODCONx）中的相应位，将 SCL 和 SDA 引脚配置为漏极开路操作。与之前版本的 MSSP 的另一点不同之处是，端口的方向控制寄存器（TRISx）必须通过清零相应的 TRIS 位来将 SDA 和 SCL 引脚配置为输出。最后，可以使用 I<sup>2</sup>C 焊盘控制（RxyI2C）寄存器配置每个引脚的压摆率控制、内部上拉电阻选择和输入阈值。



**重要：** 之前的 MSSP 模块建议使用外部上拉电阻（而非内部弱上拉电阻）。但是，现在可以使用内部弱上拉电阻，具体取决于总线发送频率和电容。可以在 RxyI2C 寄存器中配置内部上拉电阻。

SDA 和 SCL 引脚通常分配给两个 I/O 端口引脚，必须使用外设引脚选择（Peripheral Pin Select, PPS）模块使能。PPS 模块有两个专用 I<sup>2</sup>C 输入寄存器：I2CxSCLPPS（定义 SCL 输入引脚）和 I2CxDATPPS（定义 SDA 输入引脚）。SDA 和 SCL 输出也通过 PPS 模块定义。输出使用 RxyPPS 寄存器来定义引脚将输出的信号。



**重要：** 必须定义 SDA 和 SCL 输入和输出，并且必须将输入和输出分配给同一引脚。例如，如果 SDA 引脚分配给引脚 RC4，则 I2CxDATPPS 和 RC4PPS 寄存器必须映射到引脚 RC4。如果输入和输出信号未映射到同一引脚，或者其中一个信号根本未映射，则不会进行通信。

PPS 模块还允许使用备用引脚代替默认引脚位置。如果需要备用引脚位置，只需将新位置装入相应 PPS 寄存器即可。



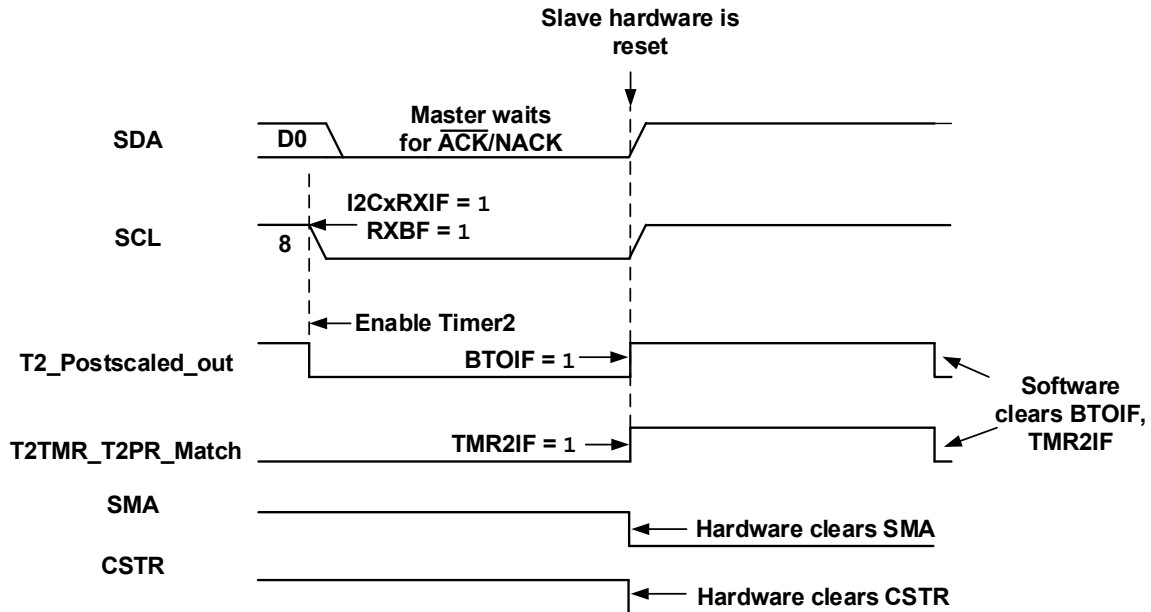
**重要：** 某些器件允许将数字外设重新定位到任何引脚，而另一些器件仅允许将数字外设移至两个 I/O 端口内的引脚。有关更多详细信息，请参见器件数据手册的 PPS 章节。此外，如果所需 I<sup>2</sup>C 引脚位置从默认引脚移至其他引脚，新位置可能未配置为 I<sup>2</sup>C 电平，需要将漏极开路（ODCONx）、压摆率（SLRCONx）和输入阈值控制（INLVLx）寄存器配置为 I<sup>2</sup>C 电平。

## 2.8 总线超时

SMBus 和 PMBus 协议需要总线看门狗来防止停止的器件无限期挂起总线。I<sup>2</sup>C 模块提供总线超时功能，可用于在其中一个总线器件响应时间过长时复位模块。I2CxBTO（I<sup>2</sup>C 总线超时）寄存器用于为模块选择超时源。当超时源到期时，I2CxBTO 寄存器会通知模块硬件并复位模块。

如果模块配置为从模式，并且从模式处于工作状态（从模式工作（SMA）位 = 1）时发生总线超时事件，则 SMA 和从器件时钟延长（CSTR）位清零，模块复位，总线超时中断标志（BTOIF）位置 1（见图 2-8）。如果 BTOIF 位置 1 时，总线超时中断允许（BTOIE）位置 1，则通用 I2C 错误中断标志（I2CxEIF）也会置 1。

图 2-8. 从接收总线超时事件示例



## 2.9 数据字节计数

数据字节计数表示一个完整 I<sup>2</sup>C 数据包中的字节的数量。I2CxCNT 寄存器用于指定完整事务的长度（以字节为单位）。每次由模块发送或接收一个数据字节时，装入 I2CxCNT 中的值都将递减。

字节传输导致 I2CxCNT 寄存器递减至 0 时，I2CxPIR 寄存器的字节计数中断标志（CNTIF）位将置 1，如果字节计数中断允许（CNTIE）位也置 1，则通用 I2C 中断标志（I2CxIF）位将置 1。I2CxIF 是只读位，只能通过清零 I2CxPIR 寄存器中的所有已允许中断标志位来清零。

可随时读取 I2CxCNT 寄存器，但建议执行双次阅读以确保有效读取。

可以写入 I2CxCNT 寄存器，但需要谨慎写入，以防寄存器损坏。如果在接收期间的第 8 个 SCL 下降沿或在发送期间的第 9 个 SCL 下降沿写入 I2CxCNT 寄存器，寄存器值可能会损坏。在从模式下，可在从器件延长时钟（CSTR = 1）时的任意时间内，或在收到停止条件后安全写入 I2CxCNT。如果 I<sup>2</sup>C 数据包长度超过 255 个字节，I2CxCNT 值可以在报文传输中进行更新，以防计数达到 0；但是，必须遵循上面列出的预防措施。

I2CxCON2 寄存器的自动装入 I2C 计数寄存器使能（ACNT）位置 1 时，I2CxCNT 值可自动装入。当 ACNT 置 1 时，地址字节后面的数据字节将装入 I2CxCNT，并且应答数据（ACKDT）位的值用于  $\overline{\text{ACK}}$  响应。

在从器件读取或主器件写入模式下且 I2CxCNT 值不为 0 时，ACKDT 位的值用于  $\overline{\text{ACK}}$  响应。当 I2CCNT = 0 时，应答计数结束（ACKCNT）位的值用于  $\overline{\text{ACK}}$  响应。

## 2.10 时钟延长

当从器件将 SCL 线保持为低电平以暂停总线通信时，会发生时钟延长现象。从器件可以延长时钟，以便可以有更多时间来处理数据或准备响应主器件。时钟延长可通过清零 I2CxCON1 寄存器的时钟延长禁止（Clock Stretching Disable, CSD）位使能，并且仅适用于多主器件模式和从模式。

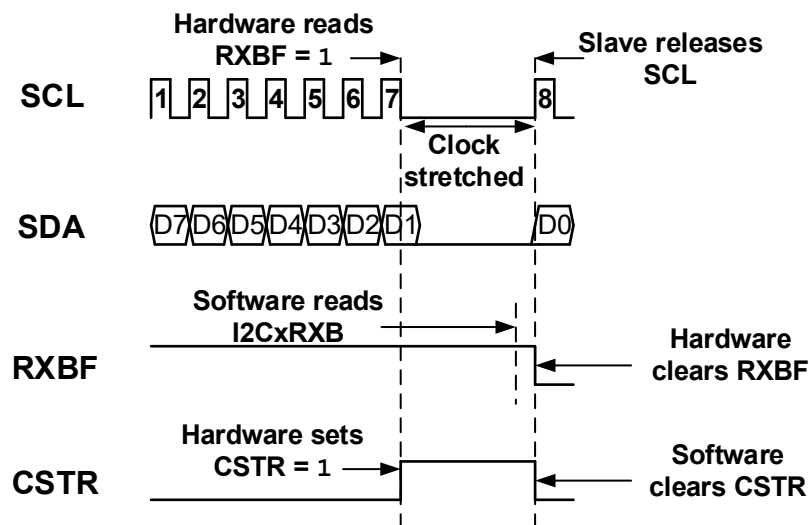
使能时钟延长时，从器件时钟延长（CSTR）位可用于确定时钟当前是否被延长，并且可用于在从器件准备好继续通信时释放 SCL 线。该位不能由软件置 1，但可在处理完字节后由软件清零。

### 用于缓冲区操作的时钟延长

使能后，将在缓冲区读/写操作期间强制进行时钟延长。这使得从器件有时间向 I2CxTXB 装入发送数据，或者从 I2CxRXB 读取数据以清除缓冲区。

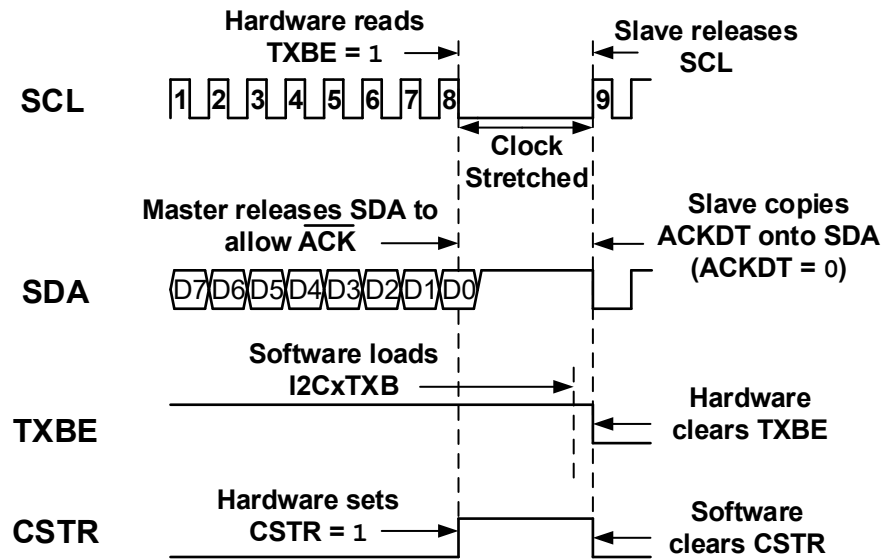
在从接收模式下，时钟延长有助于防止接收数据上溢。如果接收移位寄存器在 I2CxRXB 中仍有数据（RXBF = 1）时接收到新字节的前 7 位，则从硬件会自动延长时钟并将 CSTR 置 1。从器件读取该字节后，从软件必须将 CSTR 位清零以释放时钟并继续进行通信（见图 2-9）。

图 2-9. 接收缓冲区时钟延长



在从发送模式下，时钟延长有助于防止发送下溢。发送缓冲区空状态（TXBE）位置 1 且 I2xCNT 寄存器不等于 0 时，从硬件会延长时钟并在第 8 个 SCL 下降沿将 CSTR 置 1。这使得从软件有时间向 I2CxTXB 装入新数据。装入 I2CxTXB 后，从软件通过清零 CSTR 来释放时钟（见图 2-10）。

图 2-10. 发送缓冲区时钟延长



#### 用于其他从器件操作的时钟延长

除了上述时钟延长功能外，该模块还提供三个中断和保持使能功能。使能时钟延长（ $CSD = 0$ ）时，中断和保持使能功能提供中断并延长时钟，以便有时间进行地址识别、数据处理或  $\overline{ACK/NACK}$  响应。

发生传入地址匹配时，地址中断和保持使能功能将产生中断并延长 SCL 信号。该功能通过 I2CxPIE 寄存器的地址中断和保持使能（ADRIE）位使能。使能后，从器件时钟延长（CSTR）位和地址中断标志

（ADRIF）位由硬件置 1，SCL 线在所接收匹配地址的第 8 个 SCL 下降沿后延长。从器件完成地址处理后，软件决定将  $\overline{ACK}$  还是 NACK 发送回主器件。从软件必须清零 ADRIF 和 CSTR 位才能继续进行通信。

数据写入中断和保持使能功能会提供中断并在接收的数据字节后延长 SCL 输入。通过将 I2CxPIE 寄存器的数据写入中断和保持使能（WRIE）位置 1 来使能该功能。使能后，硬件会将 CSTR 和数据写入中断标志（WRIF）位置 1，并在 SCL 的第 8 个下降沿后延长 SCL 线。完成后，从软件必须清零 CSTR 和 WRIF 位才能恢复通信。

应答状态时间中断和保持使能功能会产生中断并在发送的应答阶段之后延长 SCL 信号。通过将应答状态时间中断和保持使能（ACKTIE）位置 1 来使能该功能。使能后，模块硬件会将 CSTR 和应答状态时间中断标志（ACKTIF）位置 1，并在 SCL 的第 9 个下降沿后延长时钟。此功能可针对所有地址、读取或写入事务使能时钟延长。完成后，从软件必须清零 CSTR 和 ACKTIF 位才能继续进行通信。



### 3. 中断

独立 I<sup>2</sup>C 模块包含额外的中断功能，旨在协助实现通信功能。除 MSSP 模块的启动/重复启动条件（SCIF）、停止条件（PCIF）、总线冲突（BCLIF）以及发送、接收和应答（SSPIF）中断外，独立 I<sup>2</sup>C 模块还增加了地址匹配（ADRIF）、发送缓冲区空（TXBE）、接收缓冲区满（RXBF）、总线超时（BTOIF）、数据字节计数（CNTIF）、应答状态时间（ACKTIF）和无应答检测（NACKIF）。

独立 I<sup>2</sup>C 模块包含一个新寄存器，即 I2C 中断标志寄存器（I2CxPIR），该寄存器可处理多个 I<sup>2</sup>C 相关中断。此外，当 I2CxPIR 中的任何标志位置 1 且 I2CxPIE 寄存器中的相关中断允许位置 1 时，通用 I2C 中断标志（I2CxIF）也会置 1。如果匹配的中断允许位置 1，则在中断标志位置 1 时，会产生中断。如果相应的中断允许位清零，则在发生中断条件时中断标志仍会置 1，但不会触发中断。



**重要：** 通用 I2CxIF 位是只读位，仅当 I2CxPIR 寄存器中的所有位均清零时才由硬件清零。

I2CxPIR 包含以下中断标志位：

- CNTIF：字节计数中断标志
- ACKTIF：应答状态时间中断标志
- WRIF：数据写入中断标志
- ADRIF：地址中断标志
- PCIF：停止条件中断标志
- RSCIF：重复启动条件中断标志
- SCIF：启动条件中断标志

I2xCNT 寄存器值达到 0 时，CNTIF 置 1（CNTIF = 1），表示已发送或接收数据帧中的所有字节。

I2xCNT = 0 时，CNTIF 在 SCL 的第 9 个下降沿之后置 1。

每当检测到  $\overline{\text{ACK}}$  时，如果器件在任意 I<sup>2</sup>C 从模式或 I<sup>2</sup>C 多主器件模式下作为从器件寻址，则接收到任何字节时，ACKTIF 会在 SCL 的第 9 个下降沿之后置 1（ACKTIF = 1）。

当模块接收到数据字节时，WRIF 在 SCL 的第 8 个下降沿之后置 1（WRIF = 1）。该位仅在 I<sup>2</sup>C 从模式或 I<sup>2</sup>C 多主器件模式下有效。接收到数据字节后，WRIF 置 1，接收缓冲区满（RXBF）状态位和 I<sup>2</sup>C 接收中断标志（I2CxRXIF）位也如此，如果数据写入中断和保持使能（WRIF）位置 1，则通用 I2CxIF 位也会置 1。WRIF 位是读/写位，必须由用户软件清零；而 RXBF、I2CxRXIF 和 I2CxIF 是只读位，通过读取 I2CxRXB 清零。

模块接收到匹配的 7 位地址字节或匹配的 10 位地址的高字节或低字节后，ADRIF 在 SCL 的第 8 个下降沿置 1。该位仅在从模式或多主器件模式下有效。接收到匹配的地址字节后，ADRIF 位置 1；如果地址中断和保持使能（ADRIF）位置 1，则通用 I2CxIF 位置 1。

在总线上检测到停止条件时，PCIF 会置 1。

RSCIF 在检测到重复启动条件时置 1。

SCIF 在检测到启动条件时置 1。

除 I2CxPIR 寄存器外，独立模块还包含 I2C 错误寄存器（I2CxERR）。I2CxERR 寄存器包含三个中断标志位，用于检测总线错误。这些位是读/写位，必须由用户软件清零。I2CxERR 寄存器还包含这三个功能的使能位。

---

I2CxERR 寄存器包含以下中断标志位：

- BTOIF：总线超时中断标志
- BCLIF：总线冲突检测中断标志
- NACKIF：NACK 检测中断标志

发生总线超时事件时，BTOIF 置 1。总线超时时间帧由 I2C 总线超时 (I2CxBTO) 寄存器控制。如果发生总线超时事件且模块配置为从模式并处于工作状态 (SMA = 1)，则 BTOIF 置 1，SMA 和 CSTR 位清零，模块复位。BTOIF 置 1 且总线超时中断允许 (BTOIE) 位置 1 时，通用 I2C 错误中断标志 (I2CxEIF) 位也会置 1。



**重要：** I2CxEIF 位是只读位，当 I2CxERR 寄存器中的所有错误中断标志位清零时，由硬件清零。

检测到总线冲突时，BCLIF 会置 1。采样到 SDA 输入为低电平，而 SDA 和 SCL 输出均为高电平时，会发生总线冲突。发生总线冲突事件时，BCLIF 置 1，如果总线冲突检测中断允许 (BCLIE) 位置 1，则通用 I2CxEIF 位也会置 1。

当主器件或从器件处于工作状态 (SMA = 1 || MMA = 1)，并且在总线上检测到 NACK 时，NACKIF 置 1。当 SDA 线释放为高电平时，将在第 9 个 SCL 脉冲出现 NACK 响应。当模块处于主模式时，如果主器件完成从从器件接收数据的过程，或者未接收到字节，则会发出 NACK。在从模式下，从器件未接收到匹配的地址或未接收到最后一个数据字节时，会发出 NACK。如果以下任意一位置 1，则还可自动发送 NACK，这会将 NACKIF 置 1，并且如果 NACK 检测中断允许 (NACKIE) 位置 1，则通用 I2CxEIF 也会置 1：

- TXWE：发送写错误状态位
- RXRE：接收读错误状态位
- TXU：发送下溢状态位
- RXO：接收上溢状态位

## 4. I<sup>2</sup>C 从模式操作

要开始任何 I<sup>2</sup>C 通信，主硬件会进行检查以确保总线处于空闲状态，即 SCL 和 SDA 线均悬空为高电平。主硬件监视总线空闲（BFRE）位的置 1 情况，置 1 表示总线空闲。主器件随后发送启动条件，后跟要通信的从器件的地址。从地址可以是 7 位或 10 位，具体取决于应用设计。

在 7 位寻址模式下，最低有效位（Least Significant bit, LSb）用作读/写（R/W）位，在 10 位寻址模式下，地址高字节的 LSb 被视为 R/W 位。当 R/W 位置 1 时，表示主器件要从从器件读取数据。如果 R/W 位清零，表示主器件要向从器件写入数据。如果总线上存在寻址的从器件，从器件必须以应答（ACK）序列响应。

然后，主器件继续从从器件接收数据和/或将数据写入从器件。数据的发送始终先从最高有效位（Most Significant bit, MSb）开始。当主器件要暂停进一步发送时，它会发送停止条件，向从器件发出通信将被终止的信号；或发送重复启动条件，向总线发出当前主器件希望保持对总线的控制以与相同从器件或其他从器件通信的信号。

从模式通过配置 I2CxCON0 寄存器的 MODE[2:0]位来进行选择。有四种从模式配置和两种多主器件模式：

- 带 7 位地址的 I<sup>2</sup>C 从模式
- 带 7 位地址（含掩码）的 I<sup>2</sup>C 从模式
- 带两个 10 位地址寄存器的 I<sup>2</sup>C 从模式
- 带一个 10 位地址（含掩码）的 I<sup>2</sup>C 从模式
- I<sup>2</sup>C 多主器件模式——带 7 位地址的主模式和带两个 7 位地址（含掩码）的从模式
- I<sup>2</sup>C 多主器件模式——带 7 位地址的主模式和带四个 7 位地址的从模式。

主器件将生成 SCL 脉冲以及启动、重复启动和停止条件。发送始终以启动条件开始，并能够以停止条件或重复启动条件结束。当主器件完成所有事务并准备释放总线时，会生成停止条件。如果主器件希望停止与一个从器件的通信，但希望保持对总线的控制以寻址另一从器件，则会发出重复启动条件。只有当 I2CxSTAT0 寄存器的总线空闲（BFRE）位置 1 时，对总线的控制才能置为有效。

从器件将等待硬件在总线上检测到启动条件。检测到启动条件后，从器件会等待将传入的地址信息装入接收移位寄存器中。然后将地址与装入 I2CxADR 寄存器中的地址进行比较。如果检测到地址匹配，则从硬件会将地址传输到 I2CxADB0/1 寄存器或 I2CxRXB 寄存器，具体取决于 ABD 位的状态。如果地址不匹配，则从器件无响应。

当 ABD 位清零时，地址缓冲区寄存器 I2CxADB0 和 I2CxADB1 有效，用于保存传入的匹配地址。

在 7 位寻址模式下，将传入地址与存储在 I2CxADR 寄存器中的地址进行比较。如果地址匹配，匹配的地址将装入 I2CxADB0 寄存器中。在 7 位寻址模式下，I2CxADB1 未使用。在 10 位寻址模式下，第一个传入的地址字节是 10 位地址的高字节，它将与 I2CxADR1 和 I2CxADR3 寄存器中的值进行比较，匹配的高地址字节将装入 I2CxADB1。如果未检测到匹配，则模块将变为空闲状态。第二个传入的字节是 10 位地址的低字节，它将与 I2CxADR0 和 I2CxADR2 寄存器中的值进行比较，匹配的低地址字节将装入 I2CxADB0。

ABD 位置 1 时，地址缓冲区无效。在这种情况下，匹配地址将装入 I2CxRXB 中。从软件必须读取 I2CxRXB 才能继续通信。

如果地址匹配，从器件会继续从主器件接收数据或将数据发送到主器件。

## 5. 从模式配置和操作

下列步骤可用于将 I<sup>2</sup>C 模块配置为从模式操作。

### 5.1 初始化

要开始 I<sup>2</sup>C 从模式通信，必须在初始化期间正确配置以下寄存器（见下面的代码示例）。

#### I<sup>2</sup>C 初始化示例

```
void I2C1_Initialize(void)
{
    I2C1ADR0 = 0x30;           //向地址寄存器装入从地址
    I2C1ADR1 = 0x80;
    I2C1ADR2 = 0x50;
    I2C1ADR3 = 0xC0;
    I2C1CON0 = 0x00;          // CSTR 使能时钟; MODE 四个 7 位地址;
    I2C1CON1 = 0x80;          //使能 CSD 时钟延长;
                                // ACKDT 应答; ACKCNT 无应答;
    I2C1CON2 = 0x00;          //使能 ABD; 禁止 ACNT;
                                // SDAHT 300 ns 保持时间;
    I2C1CNT = 0x00;           //清零计数寄存器
    PIR2bits.I2C1RXIF=0;      //清零所有 I2C 中断标志
    PIR3bits.I2C1TXIF=0;
    PIR3bits.I2C1EIF=0;
    I2C1ERRbits.NACKIF=0;
    PIR3bits.I2C1IF=0;
    I2C1PIRbits.PCIF=0;
    I2C1PIRbits.ADRIF=0;
    PIE2bits.I2C1RXIE=1;      //允许 I2C RX 中断
    PIE3bits.I2C1TXIE=1;      //允许 I2C TX 中断
    PIE3bits.I2C1EIE=1;       //允许 I2C 错误中断
    I2C1ERRbits.NACKIE=1;     //允许 I2C 错误中断以生成 NACK
    PIE3bits.I2C1IE=1;        //允许 I2C 中断
    I2C1PIEbits.PCIE=1;       //允许 I2C 中断以满足停止条件
    I2C1PIEbits.ADRIE=1;     //允许 I2C 中断以满足 I2C 地址匹配条件
                                //
    I2C1PIR = 0;              //清零所有错误标志
    I2C1ERR = 0;
    I2C1CON0bits.EN = 1;      //允许 I2C 模块
}

void PIN_MANAGER_Initialize(void)
{
    LATC = 0x00;
    TRISC = 0xE7;             // RC3 和 RC4 输出
    ANSELC = 0xE7;           //禁止模拟输入缓冲区
    WPUC = 0x00;              //转而使用 RxyI2C 值
    ODCONC = 0x18;           // RC3 和 RC4 漏极开路
    RC3PPS = 0x21;           // RC3->I2C1:SCL1;
    RC4PPS = 0x22;           // RC4->I2C1:SDA1;
    I2C1SCLPPS = 0x13;       // RC3->I2C1:SCL1;
    I2C1SDAPPS = 0x14;       // RC4->I2C1:SDA1;
}
```

**I2CxCON0:** I2CxCON0 寄存器包含模块使能 (EN) 位和模式选择 (MODE[2:0]) 位。MODE[2:0]位用于选择通信模式，EN 位用于使能从状态机硬件。EN 位置 1 (模块使能) 时，不应更改 MODE[2:0]位设置。

**I2CxCON1:** I2CxCON1 寄存器包含应答计数结束 (ACKCNT) 位、应答数据 (ACKDT) 位和时钟延长禁止 (CSD) 位。

ACKCNT 位反映 I2CxCNT 寄存器达到 0 (指示数据包结束) 后发送的值。ACKCNT 清零时，模块将发出 ACK; 置 1 时，模块将发出 NACK。ACKCNT 可在运行时修改，但只能在 I2CxCNT 达到 0 之前且发出应

答序列之前进行更改。如果 I2CxERR 或 I2CxSTAT 寄存器中存在错误，从硬件将自动改写此位设置并生成 NACK。

ACKDT 位反映接收到匹配地址后，或者接收到字节后（在 I2xCx\_CNT 不为 0 时）发送的值。ACKDT 清零时，会发出  $\overline{ACK}$ ；ACKDT 置 1 时，会发出 NACK。ACKDT 位值可在运行时修改，但只能在发出应答序列之前进行更改。如果 I2CxERR 或 I2CxSTAT 寄存器中存在错误，从硬件将自动改写此位设置并生成 NACK。

CSD 位可使能/禁止从器件延长 SCL 信号的能力。

**I2CxCON2:** I2CxCON2 寄存器包含自动装入 I2C 计数寄存器使能 (ACNT) 位、广播呼叫地址使能 (GCEN) 位、地址缓冲区禁止 (ABD) 位和 SDA 保持时间选择 (SDAHT[1:0]) 位。

ACNT 位可使能/禁止 I2Cx\_CNT 寄存器的自动装入。从器件不清楚数据包的大小时，或者主器件需要更改要发送的数据包的大小时，I2Cx\_CNT 的自动装入十分有用。

ACNT 置 1 时，匹配地址后的第一个字节用作装入 I2Cx\_CNT 寄存器的值。例如，如果主器件要将三个数据字节发送到从器件，则地址后面的字节的值将为 3，并且将在发送期间装入主器件的 I2Cx\_CNT 寄存器中。

从器件接收到该字节时，会将该字节装入从器件的 I2Cx\_CNT 寄存器。当然，前提是假设主器件和从器件都提供有 I2Cx\_CNT 寄存器功能，并且两个器件的 ACNT 均置 1。

ABD 位可使能/禁止专用地址缓冲区寄存器。在从模式下，从器件接收的匹配地址可装入 I2Cx\_ADB0/1 寄存器。ABD = 1 时，将忽略 I2Cx\_ADB0/1 寄存器，匹配的从地址将由硬件装入 I2Cx\_RXB 接收缓冲区。用户软件必须读取 I2Cx\_RXB 才能继续通信。

ABD = 0 时，匹配的接收地址数据存储于 I2Cx\_ADB0/1 中。

SDAHT[1:0]位用于配置 SCL 下降沿后 SDA 线保持有效的时间。应根据总线电容配置 SDAHT[1:0]位；电容较大的总线可能需要较长的保持时间来确保有效数据。

**I2Cx\_BTO:** I2Cx\_BTO 寄存器选择用于总线超时功能的时序源。当前超时源为 CLC 或定时器，这些模块也必须在初始化期间配置。配置超时源时，应使器件不会将总线暂停过长时间，且不会干扰及时的数据处理或时钟延长。

**I2Cx\_ERR:** I2Cx\_ERR 寄存器包含总线超时中断允许 (BTOIE) 位、总线冲突检测中断允许 (BCLIE) 位和 NACK 检测中断允许 (NACKIE) 位。如果应用不需要这些中断，则无需显式初始化该寄存器。

**I2Cx\_CNT:** I2Cx\_CNT 寄存器装有 I<sup>2</sup>C 数据包中包含的数据字节数。I2Cx\_CNT 可在初始化或运行期间直接装入，但建议仅在模块空闲或时钟延长期间写入该寄存器。在任何其他时间写入均可能会损坏寄存器。当 I2Cx\_CON2 的 ACNT 位置 1 时，I2Cx\_CNT 也可以在运行时自动装入。在这种情况下，地址字节后面的第一个字节由模块硬件装入 I2Cx\_CNT。I2Cx\_CNT 值应仅包含数据包中的数据字节数，而不包含任何地址字节。

**I2Cx\_PIE:** I2Cx\_PIE 寄存器包含几个 I<sup>2</sup>C 特定的中断允许位。只有在需要以下一个或多个中断时才需要初始化：

- 数据字节计数中断
- 应答中断和保持使能
- 数据写入中断和保持使能
- 地址中断和保持使能
- 停止条件中断
- 重复启动条件中断
- 启动条件中断

**RxyI2C:** RxyI2C 寄存器控制 I<sup>2</sup>C 特定的 I/O 焊盘。大多数 PIC<sup>®</sup> 器件都有一对或两对 I/O 引脚专用于 I<sup>2</sup>C 模块。RxyI2C 寄存器用于配置引脚压摆率、输入阈值和内部上拉配置。

SLEW 位控制压摆率。SLEW 置 1 时，将使能 I<sup>2</sup>C 特定的压摆率，这将改写标准引脚压摆率限制，并忽略与该引脚相关的 SLRCONx 位。SLEW 清零时，模块使用标准焊盘压摆率，该压摆率通过与引脚相关的 SLRCONx 位使能/禁止。较低的总线速度可能不需要任何压摆率限制，而较高速度的总线可能需要压摆率限制。

TH 位控制 I<sup>2</sup>C 输入阈值。这些位可配置为 SMBus 3.0、SMBus 2.0、I<sup>2</sup>C 特定或标准 I/O 输入阈值，以满足特定的协议要求。选择 SMBus 3.0、SMBus 2.0 或 I<sup>2</sup>C 特定阈值时，将忽略与引脚相关的 INLVLx 位。如果选择标准 I/O 阈值，则可将与引脚相关的 INLVLx 位配置为 ST 或 TTL 逻辑电平。

PU 位用于选择内部上拉驱动强度。可将 PU 位配置为增大上拉电流驱动，使内部上拉电阻足够用，从而无需外部上拉电阻。如果要在应用中使用外部上拉电阻，可将 PU 位配置为标准弱上拉，可通过与引脚相关的 WPUx 位使能/禁止。

**TRISx 寄存器:** TRISx 寄存器为端口引脚提供 I/O 方向支持。使用 I<sup>2</sup>C 模块时，与 SDA 和 SCL 引脚相关的 TRISx 位必须初始化为清零状态 (TRISxy = 0)。之前的所有 I<sup>2</sup>C 模块设计均需将 TRISx 位置 1。在运行期间，方向控制由模块硬件处理。

**ODCONx:** I<sup>2</sup>C 模块使用漏极开路电路配置。与 SCL 和 SDA 引脚相关的 ODCONx 位必须配置为漏极开路 (ODCONxy = 1)。

**I2C PPS 寄存器:** 外设引脚选择 (PPS) 功能允许数字信号从其默认引脚位置移至另一位置。要使能数字外设的输入和/或输出信号，必须配置相应的 PPS 寄存器。使用 I<sup>2</sup>C 模块时，出于 I<sup>2</sup>C 总线的双向性考虑，必须配置输入 PPS 和输出 PPS 寄存器。每个 I<sup>2</sup>C 信号的输入和输出 PPS 寄存器都必须连接到同一引脚。换言之，如果 I2CxSCLPPS 输入寄存器映射到引脚 RC3，则 RC3PPS 寄存器也必须映射到引脚 RC3。

输入配置由 I2CxSCLPPS 和 I2CxSDAPPS 寄存器处理。必须将这些寄存器映射到所需引脚，以使能引脚输入驱动器。输出配置由 RxyPPS 寄存器处理。寄存器名称中的 “xy” 是实际端口和引脚编号的占位符。例如，如果 SDA 线映射到端口引脚 RC4，则正确的寄存器名称为 RC4PPS。PPS 输出寄存器也必须映射到所需引脚，以使能引脚输出驱动器。

PPS 功能允许 I<sup>2</sup>C 引脚从默认位置移到其他位置，但需要采取额外的步骤。默认 I<sup>2</sup>C 引脚使用 RxyI2C 寄存器来定义压摆率、上拉配置和输入阈值。如果未使用默认引脚位置，则还必须配置其他寄存器，如 INLVLx、WPUx 和 SLRCONx。

## 6. 从模式发送

### 6.1 从模式发送（7 位寻址）

以下部分介绍了将 I<sup>2</sup>C 用于从模式发送时的事件序列。

1. 主器件发出启动条件。检测到启动后，从硬件将启动条件中断标志（SCIF）置 1。
2. 主器件发送  $\overline{R/W}$  置 1 的 7 位从地址。
3. 接收到的地址将与 I2CxADR 寄存器中的值进行比较。  
如果从器件配置为 7 位寻址模式（无掩码），则接收到的地址将与 I2CxADR0/1/2/3 寄存器中的每一个单独进行比较。在含掩码的 7 位寻址模式下，接收到的地址将使用 I2CxADR1 和 I2CxADR3 的值进行掩码，然后与 I2CxADR0 和 I2CxADR2 的值进行比较。  
如果发生匹配，则从模式工作（SMA）位置 1， $\overline{R/W}$  位信息（匹配地址的 bit 0）将传送到 I2CxSTAT0 寄存器的读信息（R）位，I2CxSTAT0 寄存器的数据（D）位将清零，地址中断标志（ADRIF）位置 1。如果 ABD 清零，匹配地址将复制到 I2CxADB0 寄存器。如果 ABD 置 1，匹配地址将复制到接收缓冲区 I2CxRXB，这会将接收缓冲区满（RXBF）和 I<sup>2</sup>C 接收中断标志（I2CxRXIF）位置 1。I2CRXIF 是只读位，必须通过将清除缓冲区（CLRBF）位置 1 或读取 I2CxRXB 来清零。  
若无地址匹配，则模块进入空闲状态。
4. 当 ADRIF 置 1 时，如果地址中断和保持使能（ADRIE）位置 1 且时钟延长禁止（CSD）位清零，则从硬件会将从器件时钟延长（CSTR）位和通用 I2CxIF 标志位置 1。这样，从器件便有时间读取 I2CxADB0 或 I2CxRXB，并且可基于接收到的地址有选择地发送  $\overline{ACK/NACK}$ 。从器件处理完地址后，软件会清零 CSTR 和 ADRIF，这将释放时钟并清零 I2CxIF 标志位。
5. 如果 TXBE = 1，I2CxCNT 具有非零值，并且 I2CxTXIF = 1，从硬件会将 CSTR 置 1，并等待软件装入 I2CxTXB。必须装入 I2CxTXB 才能清零 I2CxTXIF 并释放 SCL。将字节传输到移位寄存器后，I2CxCNT 递减。
6. 主器件发送第 9 个时钟脉冲，从硬件将 ACKDT 位的值传送到 SDA 线上。如果存在未决错误，例如接收上溢，则从硬件会自动生成 NACK 序列，NACKIF 会置 1，模块将进入空闲状态。从硬件检查 I2CxCNT。如果 I2CxCNT = 0，字节计数中断标志（CNTIF）将置 1。
7. 如果从器件发出了  $\overline{ACK}$  并且 I2CxCNT 非零，主器件会发送 8 个时钟脉冲，从器件将开始移出数据字节，先移出最高有效位（MSb）。如果 I2CxCNT = 0，则 CNTIF 置 1。
8. 主器件从从器件接收数据字节，然后发送  $\overline{ACK/NACK}$  序列。从硬件将  $\overline{ACK}$  值复制到应答状态（ACKSTAT）位，并将 ACKTIF 置 1。如果应答时间中断和保持使能（ACKTIE）位置 1，则从硬件会将 CSTR 和 I2CxIF 位置 1。从器件就绪后，软件会清零 CSTR 和 ACKTIF 位，这将释放 SCL 并清零 I2CxIF。
9. 继续执行步骤 13-16，直到主器件接收到所有请求的数据。接收到所有数据后，主器件会发送 NACK 条件，然后发送停止条件或重复启动条件。从硬件会将 NACKIF 和 PCIF 置 1，并清零 SMA。

### 6.2 从模式发送（10 位寻址）

以下部分介绍了将 I<sup>2</sup>C 用于从模式发送时的事件序列。

1. 主器件发出启动条件。检测到启动后，从硬件将启动条件中断标志（SCIF）置 1。
2. 主器件发送  $\overline{R/W}$  清零的 10 位高地址字节。

3. 接收到的地址将与 I2CxADR 寄存器中的值进行比较。  
如果从器件配置为 10 位寻址模式（无掩码），则接收到的高地址字节将与 I2CxADR1 和 I2CxADR3 寄存器中的值进行比较。在含掩码的 10 位寻址模式下，接收到的地址将使用 I2CxADR3 的值进行掩码，然后与 I2CxADR1 的值进行比较。如果发生匹配，则  $R/\overline{W}$  位信息（匹配地址的 bit 0）将传送到 I2CxSTAT0 寄存器的读信息（R）位，I2CxSTAT0 寄存器的数据（D）位将清零，地址中断标志（ADRIF）位置 1。如果 ABD 清零，匹配地址将复制到 I2CxADB1 寄存器。如果 ABD 置 1，匹配地址将复制到接收缓冲区 I2CxRXB，这会将接收缓冲区满（RXBF）和 I2C 接收中断标志（I2CxRXIF）位置 1。I2CxRXIF 是只读位，必须通过将清除缓冲区（CLRBF）位置 1 或读取 I2CxRXB 来清零。若无地址匹配，则模块进入空闲状态。
4. 当 ADRIF 置 1 时，如果地址中断和保持使能（ADRIE）位置 1 且时钟延长禁止（CSD）位清零，则从硬件会将器件时钟延长（CSTR）位和通用 I2CxIF 标志位置 1。这样，从器件便有时间读取 I2CxADB1 或 I2CxRXB，并且可基于接收到的地址有选择地发送  $\overline{ACK}/NACK$ 。从器件处理完地址后，软件会清零 CSTR 和 ADRIF，这将释放时钟并清零 I2CxIF 位。
5. 主器件发送第 9 个时钟脉冲，从硬件将 ACKDT 位的值传送到 SDA 线上。如果存在未决错误，例如接收上溢，则从硬件会自动生成 NACK 序列，模块将进入空闲状态。
6. 主器件发送低地址字节。  
如果从器件配置为 10 位寻址模式（无掩码），则接收到的低地址字节将与 I2CxADR0 和 I2CxADR2 寄存器中的值进行比较。在含掩码的 10 位寻址模式下，接收到的地址将使用 I2CxADR2 的值进行掩码，然后与 I2CxADR0 的值进行比较。如果发生匹配，则从模式工作（SMA）位置 1， $R/\overline{W}$  位信息将传送到 I2CxSTAT0 寄存器的 R 位，I2CxSTAT0 寄存器的 D 位清零，ADRIF 位置 1。如果 ABD 清零，匹配地址将复制到 I2CxADB0 寄存器。如果 ABD 置 1，匹配地址将复制到接收缓冲区 I2CxRXB，这会将 RXBF 和 I2CxRXIF 位置 1。I2CxRXIF 是只读位，必须通过将 CLRBF 位置 1 或读取 I2CxRXB 来清零。
7. 当 ADRIF 置 1 时，如果 ADRIE 位置 1 且 CSD 位清零，则从硬件会将 CSTR 位和通用 I2CxIF 位置 1。这样，从器件便有时间读取 I2CxADB0 或 I2CxRXB，并且可基于接收到的地址有选择地发送  $\overline{ACK}/NACK$ 。从器件处理完地址后，软件会清零 CSTR 和 ADRIF，这将释放时钟并清零 I2CxIF 位。
8. 主器件发送第 9 个时钟脉冲，从硬件将 ACKDT 位的值传送到 SDA 线上。如果存在未决错误，例如接收上溢，则从硬件会自动生成 NACK 条件，模块将进入空闲状态。
9. 在时钟的第 9 个下降沿之后，应答状态时间中断和保持（ACKTIF）位置 1。如果应答时间中断和保持使能（ACKTIE）位置 1，则从硬件会将 CSTR 和 I2CxIF 位置 1。从器件就绪后，软件会清零 CSTR 和 ACKTIF 位，这将释放 SCL 并清零 I2CxIF。
10. 主器件发出重复启动条件，从器件检测到重复启动后，硬件会将重复启动条件中断标志（RSCIF）置 1，如果重复启动条件中断允许（RSCIE）位置 1，则 I2CxIF 也会置 1。
11. 然后，主器件再次发送高地址字节，但  $R/\overline{W}$  位置 1。如果 SMA = 1 且高地址匹配，则  $R/\overline{W}$  位将复制到 I2CxSTAT0 的 R 位，I2CxSTAT0 的 D 位将清零。如果 ABD 清零，匹配地址将存储在 I2CxADB1 中。如果 ABD 置 1，匹配地址将复制到 I2CxRXB，这会将 RXBF 和 I2CxRXIF 位置 1。I2CxRXIF 是只读位，必须通过将 CLRBF 位置 1 或读取 I2CxRXB 来清零。
12. 当 ADRIF 置 1 时，如果 ADRIE 位置 1 且 CSD 位清零，则从硬件会将 CSTR 位和通用 I2CxIF 位置 1。这样，从器件便有时间读取 I2CxADB1 或 I2CxRXB，并且可基于接收到的地址有选择地发送  $\overline{ACK}/NACK$ 。从器件处理完地址后，软件会清零 CSTR 和 ADRIF，这将释放时钟并清零 I2CxIF 位。
13. 如果 TXBE = 1，I2xCNT 具有非零值，并且 I2CXTXIF = 1，从硬件会将 CSTR 置 1，并等待软件装入 I2CXTXB。必须装入 I2CXTXB 才能清零 I2CXTXIF 并释放 SCL。将字节传输到移位寄存器后，I2xCNT 递减。
14. 主器件发送第 9 个时钟脉冲，从硬件将 ACKDT 的值复制到 SDA。



15. 如果从器件发出了 **NACK**，则从硬件会将 **NACK** 检测中断标志 (**NACKIF**) 置 1 并进入空闲状态。如果从器件发出了  $\overline{\text{ACK}}$  并且 **I2CxCNT** 非零，主器件会发送 8 个时钟脉冲，从器件将开始移出数据字节，先移出最高有效位 (**MSb**)。如果 **I2CxCNT** = 0，则 **CNTIF** 置 1。
16. 主器件从从器件接收数据字节，然后发送  $\overline{\text{ACK/NACK}}$  序列。从硬件将  $\overline{\text{ACK}}$  值复制到应答状态 (**ACKSTAT**) 位，并将 **ACKTIF** 置 1。如果 **ACKTIE** 位置 1，则从硬件会将 **CSTR** 和 **I2CxIF** 位置 1。从器件就绪后，软件会清零 **CSTR** 和 **ACKTIF** 位，这将释放 **SCL** 并清零 **I2CxIF**。
17. 继续执行步骤 13-16，直到主器件接收到所有请求的数据。接收到所有数据后，主器件会发送 **NACK** 条件，然后发送停止条件或重复启动条件。从硬件会将 **NACKIF** 和 **PCIF** 置 1，并清零 **SMA**。

## 7. 从模式接收

### 7.1 从模式接收（7 位寻址）

以下部分介绍了将 I<sup>2</sup>C 用于从模式接收时的事件序列。

1. 主器件发出启动条件。检测到启动后，从硬件将启动条件中断标志（SCIF）置 1。
2. 主器件发送  $\overline{R/W}$  清零的 7 位从地址。
3. 接收到的地址将与 I2CxADR 寄存器中的值进行比较。  
如果从器件配置为 7 位寻址模式（无掩码），则接收到的地址将与 I2CxADR0/1/2/3 寄存器中的每一个单独进行比较。在含掩码的 7 位寻址模式下，接收到的地址将使用 I2CxADR1 和 I2CxADR3 的值进行掩码，然后与 I2CxADR0 和 I2CxADR2 的值进行比较。  
  
如果发生匹配，则从模式工作（SMA）位置 1， $\overline{R/W}$  位信息（匹配地址的 bit 0）将传送到 I2CxSTAT0 寄存器的读信息（R）位，I2CxSTAT0 寄存器的数据（D）位将清零，地址中断标志（ADRIF）位置 1。如果 ABD 清零，匹配地址将复制到 I2CxADB0 寄存器。如果 ABD 置 1，匹配地址将复制到接收缓冲区 I2CxRXB，这会将接收缓冲区满（RXBF）和 I2C 接收中断标志（I2CRXIF）位置 1。I2CRXIF 是只读位，必须通过将清除缓冲区（CLRBF）位置 1 或读取 I2CxRXB 来清零。  
  
如果未发生地址匹配，则模块保持空闲状态。
4. 当 ADRIF 置 1 时，如果地址中断和保持使能（ADRIE）位置 1 且时钟延长禁止（CSD）位清零，则从硬件会将从器件时钟延长（CSTR）位和通用 I2CxIF 标志位置 1。这样，从器件便有时间读取 I2CxADB0 或 I2CxRXB，并且可基于接收到的地址有选择地发送  $\overline{ACK/NACK}$ 。从器件处理完地址后，软件会清零 CSTR 和 ADRIF，这将释放时钟并清零 I2CxIF 位。
5. 主器件发送第 9 个时钟脉冲，从硬件将 ACKDT 位的值传送到 SDA 线上。如果存在未决错误，例如接收上溢，则从硬件会自动生成 NACK 条件，NACKIF 会置 1，模块将进入空闲状态。从硬件检查 I2CxCNT。如果 I2CCNT = 0，字节计数中断标志（CNTIF）将置 1。
6. 在第 9 个时钟脉冲的下降沿，应答时间中断标志（ACKTIF）位置 1，如果 CSD = 0，从硬件会将 CSTR 置 1。这样，从器件便有时间从 I2CxADB0 或 I2CxRXB 读取地址。完成后，从软件会清零 CSTR 以释放时钟，并清零 ACKTIF 以继续通信。
7. 如果从器件发送 NACK，则主硬件会生成停止条件。检测到停止条件时，停止条件中断标志（PCIF）置 1，从器件进入空闲状态。  
如果从器件发出了  $\overline{ACK}$ ，主器件会发送 8 位数据字节的前 7 位。
8. 如果在移位寄存器接收到新字节的前 7 位时前一个数据仍处于 I2CxRXB 寄存器中（RXBF = 1 且 I2CRXIF = 1），则 CSTR 位置 1，时钟将在 SCL 的第 7 个下降沿之后延长。这样，从软件便可读取 I2CxRXB，从而清零 RXBF 和 I2CxRXIF 位，并防止接收缓冲区上溢。RXBF 位清零后，软件会清零 CSTR，从而释放 SCL。
9. 主硬件将当前数据字节的第 8 位发送到从器件的接收移位寄存器，然后从硬件将完整字节传输到 I2CxRXB，并将 I2CxRXIF、数据写入中断和保持标志（WRIF）、数据（D）和 RXBF 位置 1。I2CxCNT 将递减 1。如果使能了数据写入中断和保持（WRIE = 1），则硬件会将 CSTR 置 1，以便从软件有时间读取 I2CxRXB，并在清零 CSTR 之前决定 ACKDT 位的状态。
10. 主器件发送第 9 个时钟脉冲。如果存在未决错误，例如接收上溢，则从硬件会自动生成 NACK 序列，NACKIF 会置 1，模块将进入空闲状态。

如果 I2CxCNT 不为 0，硬件会将 ACKDT 位的值作为  $\overline{\text{ACK}}$  值发送给主器件。由用户负责适当配置 ACKDT 位。在大多数情况下，应清零 ACKDT 位，以便主器件接收到  $\overline{\text{ACK}}$ （第 9 个 SCL 脉冲期间 SDA 上的逻辑低电平）。

如果 I2CxCNT = 0，硬件会将应答计数结束（ACKCNT）位的值作为  $\overline{\text{ACK}}$  值发送给从器件。由用户负责适当定义 ACKCNT 位。在大多数情况下，此位置置 1，指示 NACK 条件。当主硬件在总线上检测到 NACK 时，主硬件还将生成停止条件。如果 ACKCNT 位清零，将发出  $\overline{\text{ACK}}$ ，主硬件不会自动生成停止条件。

11. 在第 9 个时钟脉冲的下降沿，ACKTIF 位置 1，如果 ACKTIE 也置 1，则通用 I2CxIF 位置 1；如果 CSD 清零，则从硬件会将 CSTR 置 1。这样，从器件便有时间从 I2CxRXB 读取数据。完成后，从器件会清零 CSTR 以释放时钟，并清零 ACKTIF 以继续通信。
12. 转到步骤 7 继续执行步骤，直到 I2CxCNT = 0，或主器件发出停止条件。

## 7.2 从模式接收（10 位寻址）

1. 主器件发出启动条件。检测到启动后，从硬件将启动条件中断标志（SCIF）置 1。
2. 主器件发送  $R/\overline{W}$  清零的 10 位高地址字节。
3. 接收到的地址将与 I2CxADR 寄存器中的值进行比较。  
如果从器件配置为 10 位寻址模式（无掩码），则接收到的地址字节将与 I2CADR1 和 I2CADR3 寄存器中的值进行比较。在含掩码的 10 位寻址模式下，接收到的地址将使用 I2CxADR3 的值进行掩码，然后与 I2CxADR1 的值进行比较。如果发生匹配，则  $R/\overline{W}$  位信息（匹配地址的 bit 0）将传送到 I2CxSTAT0 寄存器的读信息（R）位，I2CxSTAT0 寄存器的数据（D）位将清零，地址中断标志（ADRIF）位置 1。如果 ABD 清零，匹配地址将复制到 I2CxADB1 寄存器。如果 ABD 置 1，匹配地址将复制到接收缓冲区 I2CxRXB，这会将接收缓冲区满（RXBF）和 I2C 接收中断标志（I2CxRXIF）位置 1。I2CxRXIF 是只读位，必须通过将清除缓冲区（CLRBF）位置 1 或读取 I2CxRXB 来清零。  
如果无地址匹配，则模块进入空闲状态。
4. 当 ADRIF 置 1 时，如果地址中断和保持使能（ADRIE）位置 1 且时钟延长禁止（CSD）位清零，则从硬件会将从器件时钟延长（CSTR）位和通用 I2CxIF 标志位置 1。这样，从器件便有时间读取 I2CxADB1 或 I2CxRXB，并且可基于接收到的地址有选择地发送  $\overline{\text{ACK}}/\text{NACK}$ 。从器件处理完地址后，软件会清零 CSTR 和 ADRIF，这将释放时钟并清零 I2CxIF 位。
5. 主器件发送第 9 个时钟脉冲，从硬件将 ACKDT 位的值传送到 SDA 线上。如果存在未决错误，例如接收上溢，则从硬件会自动生成 NACK 条件，模块将进入空闲状态。
6. 主器件发送低地址字节。  
如果从器件配置为 10 位寻址模式（无掩码），则接收到的低地址字节将与 I2CxADR0 和 I2CxADR2 寄存器中的值进行比较。在含掩码的 10 位寻址模式下，接收到的地址将使用 I2CxADR1 的值进行掩码，然后与 I2CxADR0 的值进行比较。如果发生匹配，则从模式工作（SMA）位置 1， $R/\overline{W}$  位信息将传送到 I2CxSTAT0 寄存器的 R 位，I2CxSTAT0 寄存器的 D 位清零，ADRIF 位置 1。如果 ABD 清零，匹配地址将复制到 I2CxADB0 寄存器。如果 ABD 置 1，匹配地址将复制到接收缓冲区 I2CxRXB，这会将 RXBF 和 I2CxRXIF 位置 1。I2CxRXIF 是只读位，必须通过将清除缓冲区（CLRBF）位置 1 或读取 I2CxRXB 来清零。
7. 当 ADRIF 置 1 时，如果 ADRIE 位置 1 且 CSD 位清零，则从硬件会将 CSTR 位和通用 I2CxIF 标志位置 1。这样，从器件便有时间读取 I2CxADB0 或 I2CxRXB，并且可基于接收到的地址有选择地发送  $\overline{\text{ACK}}/\text{NACK}$ 。从器件处理完地址后，软件会清零 CSTR 和 ADRIF，这将释放时钟并清零 I2CxIF 位。

8. 主器件发送第 9 个时钟脉冲，从硬件将 ACKDT 位的值传送到 SDA 线上。如果存在未决错误，例如接收上溢，则从硬件会自动生成 NACK 条件，模块将进入空闲状态。
9. 在时钟的第 9 个下降沿之后，应答状态时间中断和保持 (ACKTIF) 位置 1。如果应答时间中断和保持使能 (ACKTIE) 位置 1，则从硬件会将 CSTR 和 I2CxIF 位置 1。从器件就绪后，软件会清零 CSTR 和 ACKTIF 位，这将释放 SCL 并清零 I2CxIF。
10. 如果从器件发送 NACK，则主硬件会生成停止条件。检测到停止条件时，停止条件中断标志 (PCIF) 置 1，从器件进入空闲状态。  
如果从器件发出了 ACK，主硬件会发送 8 位数据字节的前 7 位。
11. 如果在移位寄存器接收到新字节的前 7 位时前一个数据仍处于 I2CxRXB 寄存器中 (RXBF = 1 且 I2CxRXIF = 1)，则 CSTR 位置 1，时钟将在 SCL 的第 7 个下降沿之后延长。这样，从软件便可读取 I2CxRXB，从而清零 RXBF 和 I2CxRXIF 位，并防止接收缓冲区上溢。RXBF 位清零后，软件会通过清零 CSTR 来释放 SCL。
12. 主硬件将当前数据字节的第 8 位发送到从器件的接收移位寄存器，然后从硬件将完整字节传输到 I2CxRXB，并将 I2CxRXIF、数据写入中断标志 (WRIF)、数据 (D) 和 RXBF 位置 1。I2CxCNT 将递减 1。如果使能了数据写入中断和保持 (WRIE = 1)，则硬件会将 CSTR 置 1，以便从软件有时间读取 I2CxRXB，并在清零 CSTR 之前决定 ACKDT 位的状态。
13. 主器件发送第 9 个时钟脉冲。如果存在未决错误，例如接收上溢，则从硬件会自动生成 NACK 条件，NACKIF 会置 1，模块将进入空闲状态。  
如果 I2CxCNT 不为 0，硬件会将 ACKDT 位的值作为  $\overline{\text{ACK}}$  值发送给主器件。由用户负责适当配置 ACKDT 位。在大多数情况下，应清零 ACKDT 位，以便主器件接收到  $\overline{\text{ACK}}$  (第 9 个 SCL 脉冲期间 SDA 上的逻辑低电平)。  
  
如果 I2CxCNT 为 0，硬件会将应答计数结束 (ACKCNT) 位的值作为  $\overline{\text{ACK}}$  值发送给从器件。由用户负责适当定义 ACKCNT 位。在大多数情况下，此位位置 1，指示 NACK 条件。当主硬件在总线上检测到 NACK 时，主硬件还将生成停止条件。如果 ACKCNT 位清零，将发出  $\overline{\text{ACK}}$ ，主硬件不会自动生成停止条件。
14. 在第 9 个时钟脉冲的下降沿，ACKTIF 位置 1，如果 ACKTIE 置 1，则通用 I2CxIF 位也置 1。如果 CSD = 0，从硬件会将 CSTR 置 1。这样，从器件便有时间从 I2CxRXB 读取数据。完成后，从软件会清零 CSTR 以释放时钟，并清零 ACKTIF 以继续通信。
15. 转到步骤 10 继续执行步骤，直到 I2CCNT = 0，或主器件发出停止条件。

## 8. 外部上拉电阻选择

I<sup>2</sup>C 规范提出了两种确定合适上拉电阻大小的方法。

第一种方法是，计算最大上拉电阻大小，此值为总线电容和上升时间的函数（见公式 8-1）。总线电容是总线/走线、总线连接点和总线引脚的总电容，在计算总线总电容时必须考虑所有这些因素。上升时间是信号从  $V_{IL(MAX)}$  ( $0.3 \cdot V_{DD}$ ) 跳变到  $V_{IH(MIN)}$  ( $0.7 \cdot V_{DD}$ ) 的时间段。上升时间值通常在器件的数据手册中提供。

应测量总线电容以获得最准确的上拉值，但也可使用估算值或 I<sup>2</sup>C 规范所定义的最大允许电容值。指定最大允许总线电容可防止上升时间缩短并允许工作在额定频率下。总线可在电容值超过上限的情况下工作，但此时频率较低。

### 公式 8-1. 最大上拉电阻大小

$$R_{p(max)} = \frac{t_{rise}}{0.8473 \cdot C_{bus}}$$

$R_{p(max)}$  = 最大上拉值

$t_{rise}$  = 最大上升时间

$C_{bus}$  = 总线总电容

另一种方法是，计算最小上拉电阻大小，此值为  $V_{DD}$  的函数（见公式 8-2）。电源电压限制了最小电阻值，因为对于标准模式（100 kHz）或快速模式（400 kHz），规定的最小灌电流为 3 mA；对于增强型快速模式（1 MHz），规定的最小灌电流为 20 mA。

### 公式 8-2. 最小上拉电阻大小

$$R_{p(min)} = \frac{V_{DD} - V_{OL(max)}}{I_{OL}}$$

$R_{p(min)}$  = 最小上拉值

$V_{DD}$  = 电源电压

$V_{OL(max)}$  = 最大输出低电压

$I_{OL}$  = 最小灌电流

## 9. 结论

本技术简介介绍了从模式配置下的独立 I<sup>2</sup>C 模块。有关更多信息，请访问 [www.microchip.com](http://www.microchip.com)。有关代码示例，请访问 [www.microchip.com/mplab/mplab-xpress](http://www.microchip.com/mplab/mplab-xpress)。

---

## Microchip 网站

---

Microchip 网站 <http://www.microchip.com/> 为客户提供在线支持。客户可通过该网站方便地获取文件和信息。只要使用常用的互联网浏览器即可访问，网站提供以下信息：

- **产品支持**——数据手册和勘误表、应用笔记和示例程序、设计资源、用户指南以及硬件支持文档、最新的软件版本以及归档软件
- **一般技术支持**——常见问题（FAQ）、技术支持请求、在线讨论组以及 Microchip 顾问计划成员名单
- **Microchip 业务**——产品选型和订购指南、最新 Microchip 新闻稿、研讨会和活动安排表、Microchip 销售办事处、代理商以及工厂代表列表

---

## 变更通知客户服务

---

Microchip 的变更通知客户服务有助于客户了解 Microchip 产品的最新信息。注册客户可在他们感兴趣的某个产品系列或开发工具发生变更、更新、发布新版本或勘误表时，收到电子邮件通知。

欲注册，请登录 Microchip 网站 <http://www.microchip.com/>。在“支持”（Support）下，点击“变更通知客户”（Customer Change Notification）服务后按照注册说明完成注册。

---

## 客户支持

---

Microchip 产品的用户可通过以下渠道获得帮助：

- 代理商或代表
- 当地销售办事处
- 应用工程师（FAE）
- 技术支持

客户应联系其代理商、代表或应用工程师（FAE）寻求支持。当地销售办事处也可为客户提供帮助。本文档后附有销售办事处的联系方式。

也可通过以下网站获得技术支持：<http://www.microchip.com/support>

---

## Microchip 器件代码保护功能

---

请注意以下有关 Microchip 器件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指标。
- Microchip 确信：在正常使用的情况下，Microchip 系列产品是当今市场上同类产品中最安全的产品之一。
- 目前，仍存在着恶意、甚至是非法破坏代码保护功能的行为。就我们所知，所有这些行为都不是以 Microchip 数据手册中规定的操作规范来使用 Microchip 产品的。这样做的人极可能侵犯了知识产权。
- Microchip 愿意与关心代码完整性的客户合作。
- 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中。Microchip 承诺将不断改进产品的代码保护功能。任何试图破坏 Microchip 代码保护功能的行为均可视为违反了《数字器件千年版权法案（Digital Millennium Copyright Act）》。如

果这种行为导致他人在未经授权的情况下，能访问您的软件或其他受版权保护的成果，您有权依据该法案提起诉讼，从而制止这种行为。

## 法律声明

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。

本出版物中所述的器件应用信息及其他类似内容仅为您提供便利，它们可能由更新之信息所替代。确保应用符合技术规范，是您自身应负的责任。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，包括但不限于针对其使用情况、质量、性能、适销性或特定用途的适用性的声明或担保。Microchip 对因这些信息及使用这些信息而引起的后果不承担任何责任。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伤害、索赔、诉讼或费用时，会维护和保障 Microchip 免于承担法律责任，并加以赔偿。除非另外声明，否则在 Microchip 知识产权保护下，不得暗或以其他方式转让任何许可证。

## 商标

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

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

Adjacent Key Suppression、AKS、Analog-for-the-Digital Age、Any Capacitor、AnyIn、AnyOut、BlueSky、BodyCom、CodeGuard、CryptoAuthentication、CryptoAutomotive、CryptoCompanion、CryptoController、dsPICDEM、dsPICDEM.net、Dynamic Average Matching、DAM、ECAN、EtherGREEN、In-Circuit Serial Programming、ICSP、INICnet、Inter-Chip Connectivity、JitterBlocker、KleerNet、KleerNet 徽标、memBrain、Mindi、MiWi、MPASM、MPF、MPLAB Certified 徽标、MPLIB、MPLINK、MultiTRAK、NetDetach、Omniscient Code Generation、PICDEM、PICDEM.net、PICkit、PICtail、PowerSmart、PureSilicon、QMatrix、REAL ICE、Ripple Blocker、SAM-ICE、Serial Quad I/O、SMART-I.S.、SQI、SuperSwitcher、SuperSwitcher II、Total Endurance、TSHARC、USBCheck、VariSense、ViewSpan、WiperLock、Wireless DNA 和 ZENA 均为 Microchip Technology Inc.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 和 Symmcom 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。

© 2019, Microchip Technology Incorporated 版权所有。

ISBN: 978-1-5224-4741-2

## 质量管理体系

---

有关 Microchip 质量管理体系的更多信息, 请访问 [www.microchip.com/quality](http://www.microchip.com/quality)。

## 全球销售及服务中心

美洲	亚太地区	亚太地区	欧洲
<b>公司总部</b> 2355 West Chandler Blvd. 钱德勒, 亚利桑那州 85224-6199 电话: 480-792-7200 传真: 480-792-7277 技术支持: <a href="http://www.microchip.com/support">http://www.microchip.com/support</a> 网址: <a href="http://www.microchip.com">www.microchip.com</a>	<b>澳大利亚 - 悉尼</b> 电话: 61-2-9868-6733 <b>中国 - 北京</b> 电话: 86-10-8569-7000 <b>中国 - 成都</b> 电话: 86-28-8665-5511 <b>中国 - 重庆</b> 电话: 86-23-8980-9588 <b>中国 - 东莞</b> 电话: 86-769-8702-9880 <b>中国 - 广州</b> 电话: 86-20-8755-8029 <b>中国 - 杭州</b> 电话: 86-571-8792-8115 <b>中国 - 香港特别行政区</b> 电话: 852-2943-5100 <b>中国 - 南京</b> 电话: 86-25-8473-2460 <b>中国 - 青岛</b> 电话: 86-532-8502-7355 <b>中国 - 上海</b> 电话: 86-21-3326-8000 <b>中国 - 沈阳</b> 电话: 86-24-2334-2829 <b>中国 - 深圳</b> 电话: 86-755-8864-2200 <b>中国 - 苏州</b> 电话: 86-186-6233-1526 <b>中国 - 武汉</b> 电话: 86-27-5980-5300 <b>中国 - 西安</b> 电话: 86-29-8833-7252 <b>中国 - 厦门</b> 电话: 86-592-2388138 <b>中国 - 珠海</b> 电话: 86-756-3210040	<b>印度 - 班加罗尔</b> 电话: 91-80-3090-4444 <b>印度 - 新德里</b> 电话: 91-11-4160-8631 <b>印度 - 浦那</b> 电话: 91-20-4121-0141 <b>日本 - 大阪</b> 电话: 81-6-6152-7160 <b>日本 - 东京</b> 电话: 81-3-6880-3770 <b>韩国 - 大邱</b> 电话: 82-53-744-4301 <b>韩国 - 首尔</b> 电话: 82-2-554-7200 <b>马来西亚 - 吉隆坡</b> 电话: 60-3-7651-7906 <b>马来西亚 - 槟榔屿</b> 电话: 60-4-227-8870 <b>菲律宾 - 马尼拉</b> 电话: 63-2-634-9065 <b>新加坡</b> 电话: 65-6334-8870 <b>台湾地区 - 新竹</b> 电话: 886-3-577-8366 <b>台湾地区 - 高雄</b> 电话: 886-7-213-7830 <b>台湾地区 - 台北</b> 电话: 886-2-2508-8600 <b>泰国 - 曼谷</b> 电话: 66-2-694-1351 <b>越南 - 胡志明市</b> 电话: 84-28-5448-2100	<b>奥地利 - 韦尔斯</b> 电话: 43-7242-2244-39 传真: 43-7242-2244-393 <b>丹麦 - 哥本哈根</b> 电话: 45-4450-2828 传真: 45-4485-2829 <b>芬兰 - 埃斯波</b> 电话: 358-9-4520-820 <b>法国 - 巴黎</b> 电话: 33-1-69-53-63-20 传真: 33-1-69-30-90-79 <b>德国 - 加兴</b> 电话: 49-8931-9700 <b>德国 - 哈恩</b> 电话: 49-2129-3766400 <b>德国 - 海尔布隆</b> 电话: 49-7131-72400 <b>德国 - 卡尔斯鲁厄</b> 电话: 49-721-625370 <b>德国 - 慕尼黑</b> 电话: 49-89-627-144-0 传真: 49-89-627-144-44 <b>德国 - 罗森海姆</b> 电话: 49-8031-354-560 <b>以色列 - 若那那市</b> 电话: 972-9-744-7705 <b>意大利 - 米兰</b> 电话: 39-0331-742611 传真: 39-0331-466781 <b>意大利 - 帕多瓦</b> 电话: 39-049-7625286 <b>荷兰 - 德卢内市</b> 电话: 31-416-690399 传真: 31-416-690340 <b>挪威 - 特隆赫姆</b> 电话: 47-72884388 <b>波兰 - 华沙</b> 电话: 48-22-3325737 <b>罗马尼亚 - 布加勒斯特</b> 电话: 40-21-407-87-50 <b>西班牙 - 马德里</b> 电话: 34-91-708-08-90 传真: 34-91-708-08-91 <b>瑞典 - 哥德堡</b> 电话: 46-31-704-60-40 <b>瑞典 - 斯德哥尔摩</b> 电话: 46-8-5090-4654 <b>英国 - 沃金厄姆</b> 电话: 44-118-921-5800 传真: 44-118-921-5820
<b>亚特兰大</b> 德卢斯, 佐治亚州 电话: 678-957-9614 传真: 678-957-1455 <b>奥斯汀, 德克萨斯州</b> 电话: 512-257-3370 <b>波士顿</b> 韦斯特伯鲁, 马萨诸塞州 电话: 774-760-0087 传真: 774-760-0088 <b>芝加哥</b> 艾塔斯卡, 伊利诺伊州 电话: 630-285-0071 传真: 630-285-0075 <b>达拉斯</b> 阿迪森, 德克萨斯州 电话: 972-818-7423 传真: 972-818-2924 <b>底特律</b> 诺维, 密歇根州 电话: 248-848-4000 <b>休斯顿, 德克萨斯州</b> 电话: 281-894-5983 <b>印第安纳波利斯</b> 诺布尔斯维尔, 印第安纳州 电话: 317-773-8323 传真: 317-773-5453 电话: 317-536-2380 <b>洛杉矶</b> 米镇维荷, 加利福尼亚州 电话: 949-462-9523 传真: 949-462-9608 电话: 951-273-7800 <b>罗利, 北卡罗来纳州</b> 电话: 919-844-7510 <b>纽约, 纽约州</b> 电话: 631-435-6000 <b>圣何塞, 加利福尼亚州</b> 电话: 408-735-9110 电话: 408-436-4270 <b>加拿大 - 多伦多</b> 电话: 905-695-1980 传真: 905-695-2078			