I²C/SMBus® 控制器

特性

- I²C/SMBus 2.0主机控制器
- I²C向后兼容模式
- 冲突检测硬件（多主器件支持）
- 从器件支持：
- 支持多个从器件地址：
  - 可编程从器件地址寄存器，用于两个通用从器件地址
  - 用于主机通知命令的SMBus主机地址（0001 000）
  - SMBus器件默认地址（1100 001）
  - 广播呼叫地址（0000 000）
- 数据包错误校验（PEC）支持选项
- 可选I²C/SMBus事务速度（最高1 MHz）
- ARP支持
- 支持7位SMBus寻址
- 主机通知命令（收到的报文）支持
- 输入噪声滤波器
- 支持主/从时钟延长
- 支持SMBus超时
- 位拆裂模式
- 支持I²C和SMBus时序
- 为MCTP公平协议提供硬件支持（见参考文档[5]）
- 最多16个可选总线端口
- 为网络层协议提供硬件支持
  - 支持几乎所有标准SMBus事务
  - 支持其他使用I²C的协议（例如，PMBus；见参考文档[6]）
  - 与外部DMA结合使用，支持在存储器和物理层之间通过硬件传输最多255个字节的数据
  - 无轮询并且每个网络层事务所需的中断不超过1个或2个

© 2019 Microchip Technology Inc.

致 客 户

我们旨在提供最佳文档供客户正确使用 Microchip 产品。为此，我们将不断改进出版物的内容和质量，使之更好地满足您的需求。出版物的质量将随新文档及更新版本的推出而得到提升。

如果您对本出版物有任何问题和建议，请通过电子邮件联系我公司 TRC 经理，电子邮件地址为 **CTRC@microchip.com**。我们期待您的反馈。

最新数据手册

欲获得本数据手册的最新版本，请访问我公司网站：

http://www.microchip.com

查看数据手册中任意一页下边角处的文献编号即可确定其版本。文献编号中紧跟数字串后的字母是版本号，例如：DS30000000A_CN 是文档的 A 版本。

勘误表

现有器件可能带有一份勘误表，描述了实际运行与数据手册中记载内容之间存在的细微差异以及建议的变通方法。一旦我们了解到器件 / 文档存在某些差异时，就会发布勘误表。勘误表上将注明其所适用的硅片版本和文件版本。

欲了解某一器件是否存在勘误表，请通过以下方式之一查询：

- Microchip 网站 **http://www.microchip.com**
- 当地 Microchip 销售办事处 （见最后一页）

在联络销售办事处时，请说明您所使用的器件型号、硅片版本和数据手册版本 （包括文献编号）。

客户通知系统

欲及时获知 Microchip 产品的最新信息，请到我公司网站 **http://www.microchip.com** 上注册。

© 2019 Microchip Technology Inc.

目录

1.0 约定 4
2.0 概述 5
3.0 硬件接口 7
4.0 电源、时钟和复位 8
5.0 中断接口 9
6.0 功能说明 10
7.0 寄存器接口 26
8.0 SMBus接口时序 47
附录A：数据手册版本历史 48

1.0 约定

第一个表定义了文档中使用的通用术语。第二个表定义了文档中使用的寄存器位访问类型符号。这些访问类型受CSL支持。

术语	定义
模块	用于识别或描述器件中实现的逻辑或IP模块。
保留	下表中定义的保留寄存器和位是只读值，读取时返回0。对这些保留寄存器的写操作将不起作用。
Microchip保留	Microchip保留存储单元不得修改为默认值以外的值。更改Microchip寄存器可能会导致不良结果。
b	数字后面的字母“b”表示二进制数。
h	数字后面的字母“h”表示十六进制数。

寄存器位类型符号	寄存器位说明
R	**读：**可以对具有该属性的寄存器或位进行只读操作。写入该位将不起作用。
R/W	**写：**可以对具有该属性的寄存器或位进行写入和回读操作。
R/WC	**写：**可以对具有该属性的寄存器或位进行写入和回读操作。写1可将值清零。写0将不起作用。

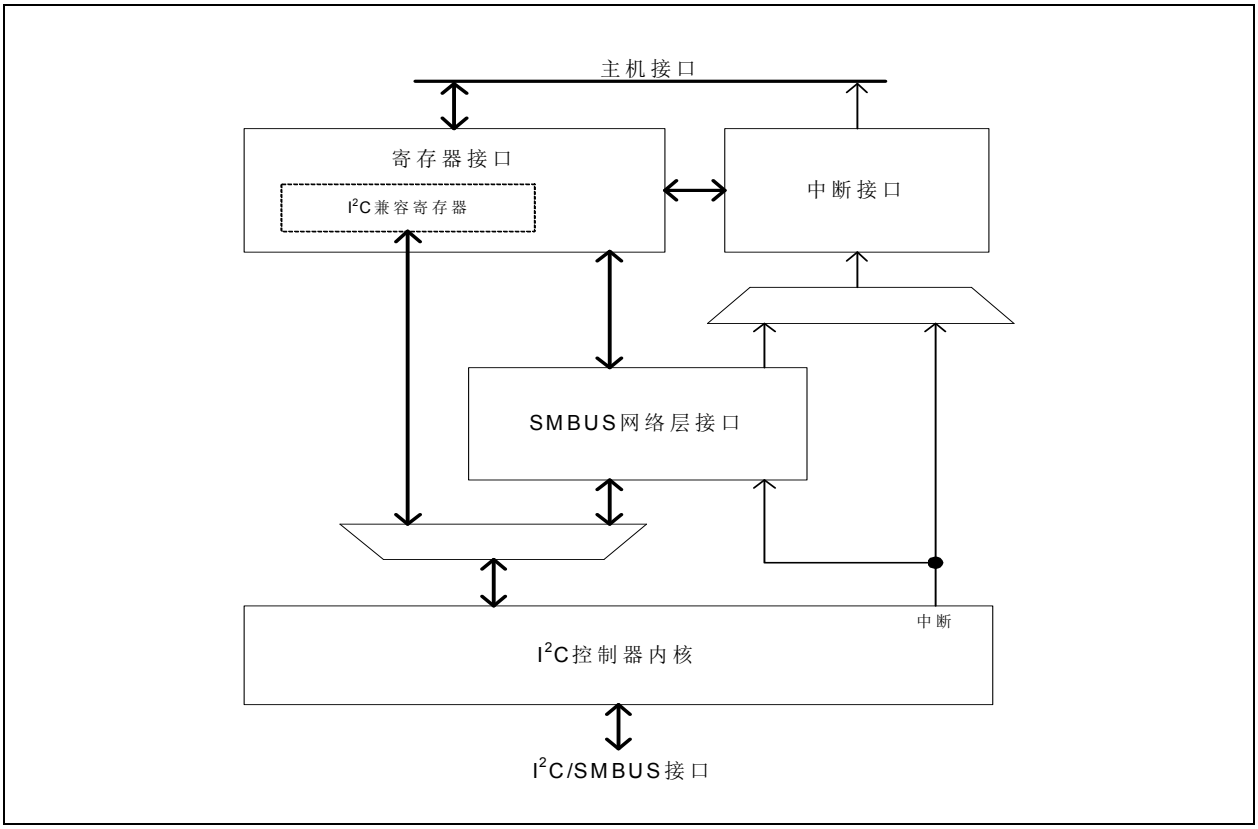
© 2019 Microchip Technology Inc.

2.0 概述

I²C总线协议和SMBus协议应用于系统内部通信的许多方面。I²C/SMBus控制器接口可以处理标准SMBus 2.0协议以及I²C接口。I²C/SMB控制器在以下两个级别上实现：低级I²C内核和使用I²C内核实现低级数据链路和物理层操作的高级网络层。

图2-1给出了I²C/SMBus控制器的框图。

图2-1： I²C/SMB控制器框图



I²C/SMB控制器也可以配置为在硬件中执行参考文档[1]中所述SMBus规范的物理层、数据链路层和网络层功能。网络层SMBus协议（例如，读取字节、写入字节和读取过程调用等）不能完全得到处理，并且在大多数情况下，必须在低级软件中进行部分处理。网络级协议的完整硬件实现本身存在限制，因为SMBus网络层协议不断更新版本（例如，PMBus规定了一种不属于标准SMBus 2.0规范的多帧写入协议）。

I²C/SMB控制器加快了硬件中的数据移动速度，但仍然依赖软件来获取网络层协议的详细信息。嵌入式控制器与外部DMA控制器配合使用，能够以最少的软件干预来处理所有当前网络层协议，甚至可以处理大多数将来的网络层协议。

2.1 参考文档

1. System Management Bus (SMBus) Specification，版本2.0，SBS Implementors Forum，2000年8月3日（http://www.sbs-forum.org/）。
2. The I²C-Bus Specification，版本2.1，Philips Semiconductors，2000年1月。
3. PCF8584 I²C-Bus Controller，产品规范，Philips Semiconductors，1997年10月21日。

4. Management Component Transport Protocol (MCTP) Base Specification，版本1.0，DMTF DSP0236，2007年6月20日。
5. Management Component Transport Protocol SMBus/I²C Transport Binding Specification，版本1.0，DMTF DSP0237，2007年6月20日。
6. PMBus™ Power System Management Protocol，版本1.1，System Management Interface Forum，2007年2月5日。

© 2019 Microchip Technology Inc.

3.0 硬件接口

I²C/SMB控制器的硬件接口包括2线SDA和SCL总线对。该接口还具有“总线端口选择”功能，可将2线总线对分配给16个可用的总线端口中的任何一个。

I²C/SMB总线直接由控制器操作。此外，软件还可以通过位拆裂接口直接控制两条总线，如第7.1.15节“位拆裂控制寄存器”所述。

3.1 SCL

SCL表示在参考文档[2]中定义并在整个规范中引用的I²C接口中的串行时钟线。根据参考文档[2]中的定义，SCL为通过外部电流源或上拉电阻连接到正电源电压的双向信号。

3.2 SDA

SDAT表示在参考文档[2]中定义并在整个规范中引用的I²C接口中的串行数据线。根据参考文档[2]中的定义，SDAT为通过外部电流源或上拉电阻连接到正电源电压的双向信号。

3.3 总线端口选择

I²C/SMB控制器可使用配置寄存器中的PORT_SEL位将2线SDAT和SCLK总线对分配给16个可用的总线端口中的任何一个。

4.0 电源、时钟和复位

本章定义了模块的电源、时钟和复位参数。

4.1 电源域

名称	说明
EC_PWR	I²C/SMB需要一个电源电压。

4.2 时钟输入

名称	说明
CORE_CLOCK	这是包含I²C/SMB的器件的主系统时钟
I2C_BAUD_CLOCK	波特率时钟为I²C/SMB控制器事务状态机和物理I²C总线接口提供时钟驱动。表7-2“建议的编程值”和表7-3“总线时钟寄存器与频率”列出了基于当前波特率时钟的时序选择。波特率时钟可能是以下频率之一： • 10 MHz • 16 MHz • 20 MHz
I2C_BUS_CLOCK	这是外部总线上的时钟，在SCL引脚上承载。当I²C/SMB用作从器件时，外部主器件控制该时钟的频率。当I²C/SMB用作主器件时，该时钟的频率通过总线时钟寄存器定义。

4.3 复位

名称	说明
RESET_EC	这是用于复位器件中大多数逻辑的复位信号。
RESET_I2C	该信号用于将该模块中的所有寄存器和逻辑复位为其默认状态。I2C_RESET在以下情况下置为有效： • MAIN_RESET置为有效 • I²C/SMB检测到需要复位模块的错误条件 • 配置寄存器中的RESET位置为有效

© 2019 Microchip Technology Inc.

5.0 中断接口

本章定义了从该模块生成的中断源。

表5-1： EC中断

源	说明
I2C_INT	至中断聚合器的中断请求。只要以下四个状态位中的任何一个在其相关中断允许位为高电平时置为有效，该中断就会置为有效： • PIN（低电平有效），通过ENI使能 • IDLE，通过ENIDI使能 • MDONE，通过ENMI使能 • SDONE，通过ENSI使能
I2C_INT_WAKE	至中断聚合器唤醒接口的唤醒请求。如果START_DETECT_INT_EN位为1时START_BIT_DETECTION状态位也为1，则该中断将置为有效。请参见以下部分。

5.1 I²C唤醒

只要使能，I²C/SMB就会异步采样SCL和SDA输入，即使其SLEEP_EN输入置为有效也是如此。如果在总线上检测到启动位且已使能I2C_INT_WAKE事件，则会生成唤醒事件。唤醒功能允许芯片进入低功耗模式而不会丢失I²C/SMBus命令。时钟信号（SCL）将在启动位后立即延长低电平时间，直到CORE_CLOCK达到其正常工作频率。时钟延长时间取决于休眠恢复时间，休眠恢复时间可能取决于芯片的实现方式以及检测到启动位时的芯片状态。

6.0 功能说明

I²C/SMB控制器包含一个内核接口（如第6.1节“I²C控制器内核”所述）和一个使用内核接口访问物理总线的网络层接口（如第6.2节“SMBus网络层接口”所述）。

6.1 I²C控制器内核

I²C/SMB实现了参考文档[2]和参考文档[3]中定义的物理层和数据链路层支持。核心寄存器组为外部总线提供一次一个字节的接口。

6.1.1 PIN操作

非待处理中断位（PIN）是用于同步串行通信的状态标志，只要I²C/SMB控制器需要处理，就会置为有效（0）。通常，在轮询的应用程序中读取PIN位以确定事务何时完成（见第6.1.6节“事务基元”）。

当用作发送器时，每次写入数据寄存器，PIN就会置为无效（1）。当用作接收器时，每次读取数据寄存器，PIN位就会置为无效。

在总线上发送或接收一个字节（九个时钟脉冲，包括应答）后，PIN位将自动置为有效（0）（有效），表示已发送/接收一个完整字节。当PIN位随后设置为逻辑1（无效）时，状态寄存器中的所有状态位（包括BER位）都将复位为0，即使已发生总线超时也是如此。在仲裁失败后的字节应答（ACK）后，PIN也会置为有效。PIN在出现总线错误（BER）条件后立即置为有效。

当控制寄存器中的ENI位置为有效（1）后，将允许硬件PIN中断。在这种情况下，每次PIN置为有效（0）时，PIN位就会触发外部中断（另见第5.0节“中断接口”）。

6.1.1.1 时钟延长

当用作从发送器或从接收器时，如果PIN置为有效（0），I²C/SMB控制器将通过使SCL信号保持低电平来暂停总线发送，直到PIN位置为无效为止。在从数据寄存器中读取当前字节（用作从接收器时）或向数据寄存器中写入下一个字节（用作从发送器时）之前，时钟延长可防止进一步发送或接收数据。

6.1.2 从器件地址解码

配置为从器件时，I²C/SMB最多可响应五个从器件地址：

表6-1： 从器件地址

从器件地址	配置寄存器		状态寄存器		
	GC_DIS	DSA	AAS	LRB/AD0	SAD
广播呼叫地址（0000_000b）	0	x	1	1	0
自身地址1	x	x	1	0	0
自身地址2	x	x	1	0	0
SMBus主机地址（0001_000b）	x	1	1	0	1
SMBus设备默认地址（1100_001b）	x	1	1	1	1

6.1.2.1 报警响应地址解码

单从器件可以通过SMBALERT#边带信号（超出本规范的范围）向主机发送信号，告知需要服务。主机将处理中断，并通过报警响应地址（Alert Response Address，ARA）（0001 100）同时访问所有SMBALERT#器件。如上文所述，可选择使用ARA的报警功能，并且该功能适用于单从器件。本规范假设I²C/SMB控制器不需要在硬件中对该地址进行解码即可支持报警功能，因为它可以配置为主器件。如果I²C/SMB控制器配置为从器件并将SMBALERT#置为有效（例如，使用专用于此目的的GPIO输出），软件可以选择使用自身地址之一来解码ARA。

© 2019 Microchip Technology Inc.

6.1.3 启动字节支持

需要通过I²C/SMB控制器来提供I²C总线规范（参考文档[2]）中定义的启动字节支持，表6-2“I2C/SMB控制器寻址行为汇总”对此进行了汇总。启动字节支持包括对第6.1.3.1节“启动字节地址”中所述的启动字节地址的惟一从器件响应。

表6-2： I²C/SMB控制器寻址行为汇总

M/S（注1）	地址	R/W位（注2）	ACK/NACK	说明（注3）
M	任意	0	ACK	主器件在从器件地址后发送DATA、STOP或SR。
			NACK	主器件在从器件地址后发送STOP或SR。
		1	ACK	主器件在从器件地址后接收DATA，发送STOP或SR。
		1	NACK	主器件在从器件地址后发送STOP或SR。
S	0000 000b	1	ACK	ACK是对参考文档[2]中定义的启动字节地址的非法从器件响应。
			NACK	对启动字节地址的正确从器件响应
	>0000 000b	1	ACK	从器件在从器件地址后向主器件发送数据。
			NACK	从器件不可用。
	任意	0	ACK	从器件在从器件地址后从主器件接收数据。
			NACK	从器件不可用。

注 1： M = 主器件（发送ADDR和R/W位），S = 从器件（发送ACK/NACK作为对从器件地址的响应）。
2： 0 = 写入（主发送器），1 = 读取（主接收器）。
3： DATA = 下一个数据字节，STOP = 停止条件，SR = 重复启动条件。

6.1.3.1 启动字节地址

启动字节地址在参考文档[2]的表2中定义为R/nW位置1的从器件地址0000 000b。I²C/SMB控制器从接口将忽略启动字节地址（即，对地址做出NACK响应，丢弃数据且不将PIN位置为有效）以便为重复启动条件做好准备。

I²C/SMB控制器配置为主器件时依赖软件来判断对启动字节地址的从器件响应。

6.1.4 硬件PEC支持

I²C/SMB控制器包括硬件PEC支持，有助于降低SMBus 2.0数据包错误校验的软件开销。针对PEC支持的用户接口元素包括一个8位PEC寄存器和一个PEC使能位。

当PEC使能位PCEN置为有效时，PEC硬件将根据SMBus 2.0规范（参考文档[1]）定义的发送和接收串行数据流计算PEC字节。只要状态寄存器中的PIN位置为有效，就可以通过软件从PEC寄存器中读取有效PEC数据。当PCEN未置为有效时，软件可通过写入PEC寄存器将PEC字节初始化为任意值。每当主器件状态机发出停止位或从器件状态机完成处理时，PEC就会清零。如果主器件中出现仲裁失败的情况且从器件状态机空闲，则PEC也会清零（即，如果主器件对外部主器件的仲裁失败，则PEC复位；但如果外部主器件以从器件为目标，则PEC不复位）。

当主器件状态机工作时，如果READ_PEC为1，则当状态机完成接收操作时，PEC寄存器的内容将复制到主器件接收缓冲寄存器。当READ_PEC为1时，主器件状态机会将最终PEC字节从从器件复制到主器件存储器接收缓冲区中，随后还会复制PEC寄存器的内容。软件可以通过检查存储器缓冲区中的最后一个字节（PEC寄存器的副本）是否为0来验证是否已正确接收传入数据流。

当从器件状态机工作时，如果状态机正在接收数据，PEC寄存器的内容将在外部主器件发出停止条件或重复启动条件后复制到从器件接收缓冲寄存器中。因此，从器件状态机复制到从器件存储器接收缓冲区的内容始终比通过I²C总线接收的内容多一个字节（PEC寄存器的内容）。软件可以确定SMBus上使用了哪种协议，从而确定通过I²C总线接收的最后一个字节是否为PEC字节。如果接收到的最后一个字节是PEC字节，则在接收到的数据得到正确验证后，存储器缓冲区中的最后一个字节（PEC寄存器的副本）将为0。

当主器件状态机和从器件状态机被禁止时，由软件负责提供PEC支持，包括：使用PEC寄存器初始化PEC字节、在适当的情况下使能PEC以及在需要时将PEC寄存器的值插入数据寄存器或将该值与其进行比较。

6.1.4.1 PEC示例

下表给出了一个32字节报文的PEC计算示例，其中每个字节的报文数据等于十六进制的字节数。

表6-3： CRC-8示例

字节编号	报文（十六进制）	CRC（十六进制）
	0	0
1	1	7
2	2	1B
3	3	48
4	4	E3
5	5	BC
6	6	2F
7	7	D8
8	8	3E
9	9	85
10	A	A4
11	B	44
12	C	FF
13	D	D0
14	E	14
15	F	41
16	10	B0
17	11	6E
18	12	73
19	13	27
20	14	99
21	15	AD
22	16	28
23	17	BD
24	18	72
25	19	16
26	1A	24
27	1B	BD
28	1C	6E
29	1D	5E
30	1E	C7

© 2019 Microchip Technology Inc.

表6-3： CRC-8示例（续）

字节编号	报文（十六进制）	CRC（十六进制）
31	1F	6
32	20	F2
	F2	0

6.1.5 初始化

复位后，需要通过软件来实现初始化过程（如表6-4所示），以便能够正确配置I²C/SMB来成功执行I²C兼容事务。如果由于外部复位信号或由于配置寄存器中的RESET位置为有效而发生复位，则应遵循此顺序。该过程同时适用于I²C内核层和网络层。

表6-4： 复位后的初始化顺序

	操作	说明
1.	向控制寄存器写入80h。	将PIN位置为有效；关闭串行接口（ESO = 0）并禁止中断（ENI）。
2.	向自身地址寄存器写入一个有效的从器件地址。	初始化从器件地址
3.	向总线时钟寄存器写入XXh。	配置所需的总线时钟。表7-3“总线时钟寄存器与频率”列出了建议的值。
4.	向控制寄存器写入C1h。	PIN保持有效；开启串行接口（ESO = 1）；配置肯定应答（ACK = 1）；SDA和SCL均为高电平（注意：NBB将由于复位而置为无效（1））。
5.	更新其他寄存器。	所有其他寄存器（配置寄存器和空闲调节寄存器等）均设置为适当的值
6.	延时。（见注1）	等待的时间等于仅同步多主器件系统中的NBB位所需的最长I²C报文时间（见第6.1.7节“I²C多主器件支持”）。
注 1： 为了正确同步，软件等待的周期必须等于另一个主器件可能占用总线的最长时间。如果该周期未知，或外部器件占用总线的时间超过指定周期，就无法正确初始化I²C。		

6.1.6 事务基元

所有I²C和SMBus兼容型协议都可以使用I²C/SMB的事务基元实现。这些基元包括主发送器、主接收器、后跟重复启动条件的主发送器以及成为主接收器和从接收器/发送器事务描述。

后面的事务描述不包括用于检测总线争用的步骤（如第6.1.7节“I²C多主器件支持”所述），必须将这些步骤集成到用于多主器件环境的所有软件中。有关初始化的描述，请参见第6.1.5节“初始化”。

在所有情况下，下述基元均假设配置寄存器中的ENAB位置为有效且同一寄存器中的RESET位未置为有效。

6.1.6.1 主器件完成

为了在主器件发出停止命令后重新使能从接收器，需要按照表6-5所述的步骤操作：

表6-5： 重新使能从接收器

	操作	说明
1.	读取状态寄存器。	轮询NBB位，直到该位置为无效（1），即等待总线空闲
2.	向控制寄存器写入C1h。	PIN保持有效；开启串行接口（ESO = 1）；配置肯定应答（ACK = 1）；SDA和SCL均为高电平。

6.1.6.2 主发送器

假设I²C/SMB控制器已初始化（如第6.1.5节“初始化”所述）且ESO位置为有效（1）。

表6-6： 主发送器模式

	操作	说明
1.	读取状态寄存器。	轮询NBB位，直到该位置为无效（1）。
2.	向数据寄存器写入D0（R/nW位）= 0的从器件地址。	设置主发送器模式。
3.	向控制寄存器写入C5h。	PIN = ESO = 1，STA = 1，STO = 0，ACK = 1。 I²C/SMB控制器生成“启动”条件，并随时钟输出用于从器件应答的地址和时钟脉冲。写入数据寄存器的下一个字节将立即通过I²C总线传输。也就是说，如果从器件地址的R/nW位 = 0，则I²C/SMB控制器将保持主发送器模式。
4.	读取状态寄存器。	轮询PIN位，直到该位置为有效（0），即字节发送完成。如果LRB/AD0 = 0（即接收到从器件ACK响应）且需要发送更多字节，则转到步骤5；其他情况下，如果不需要发送更多字节或从器件对字节做出NACK（否定应答）响应，则转到步骤6。
5.	向数据寄存器写入下一个字节。	发送下一个字节。转到步骤4。
6.	向控制寄存器写入C3h。	PIN = ESO = 1，STA = 0，STO = 1，ACK = 1。 I²C/SMB控制器将生成“停止”条件并进入从接收器/发送器模式。如果需要重复“启动”条件而非“停止”条件，不要执行该步骤，并转到第15页上的第6.1.6.4节“后跟重复启动条件的主发送器以及成为主接收器” 中的表6-8“重复启动”。

© 2019 Microchip Technology Inc.

6.1.6.3 主接收器

假设I²C/SMB控制器已初始化（如第6.1.5节“初始化”所述）且ESO位置为有效（1）。

表6-7： 主接收器模式

	操作	说明
1.	向数据寄存器写入D0（R/nW位）= 1的从器件地址。	设置主接收器模式。
2.	读取状态寄存器。	轮询NBB位，直到该位置为无效（1）。
3.	向控制寄存器写入C5h。	PIN = ESO = 1，STA = 1，STO = 0，ACK = 1。 I²C/SMB控制器生成“启动”条件，并随时钟输出用于从器件应答的地址和时钟脉冲。
4.	读取状态寄存器。	轮询PIN位，直到该位置为有效（0），即字节接收完成。如果LRB/AD0 = 0（即接收到从器件ACK响应）且需要接收两个以上字节，则转到步骤5；其他情况下，如果要接收最后一个字节，则转到步骤6，如果从器件对地址做出NACK（否定应答）响应，则转到步骤9。
5.	读取数据寄存器。	检索接收到的数据字节。转到步骤4。请注意，读取的第一个字节是从器件地址的“假读”值。后续读取值是从从器件接收的数据。
6.	向控制寄存器写入40h。	ESO = 1。将ACK位置为无效（0），以准备通过否定应答结束传输。
7.	读取数据寄存器。	检索接收到的倒数第二个字节。接收最后一个字节并对其做出NACK（否定应答）响应。
8.	读取状态寄存器。	轮询PIN位，直到该位置为有效（0），即最后一个字节接收完成。
9.	向控制寄存器写入C3h。	PIN = ESO = 1，STA = 0，STO = 1，ACK = 1。 I²C/SMB控制器将生成“停止”条件并进入从接收器/发送器模式。
10.	读取数据寄存器。	检索最终接收的字节（即得到NACK响应的从器件地址）。

6.1.6.4 后跟重复启动条件的主发送器以及成为主接收器

表6-8： 重复启动

	操作	说明
1.	执行主发送器模式，不包括最终“停止”条件。	请参见第6.1.6.2节“主发送器”。不要执行表6-6中的步骤6。
2.	向控制寄存器写入45h。	PIN = 0，ESO = 1，STA = 1，STO = 0，ACK = 1。I²C/SMB控制器将仅生成重复“启动条件”，即数据寄存器的当前内容不随时钟输出到I²C总线。发送到数据寄存器的下一个字节应为“从器件地址”+“读取”位（R/nW = 1）。
3.	向数据寄存器写入D0（R/nW位）= 1的从器件地址。	设置主接收器模式。
4.	执行主接收器模式，不包括“启动”生成。	请参见第6.1.6.3节“主接收器”。从表6-7中的步骤4开始执行

6.1.6.5 从接收器/发送器

假设I²C/SMB控制器已初始化（如第6.1.5节“初始化”所述）且ESO位置为有效（1）。

表6-9： 从接收器/发送器

	操作	说明
1.	读取状态寄存器。	轮询作为从器件寻址（AAS）位，直到该位置为有效（1）。
2.	读取状态寄存器。	轮询PIN位，直到该位置为有效（0）；即地址接收完成。
3.	读取数据寄存器。	检索接收到的地址字节以确定R/nW位（D0）是0还是1，从而区分是从接收器（R/nW = 0，转到步骤1）还是从发送器模式（R/nW = 1，转到步骤1）。 请注意，如果R/nW = 1，则不会通过读取数据寄存器复位PIN位（1），因为从器件是发送器。
从发送器模式		
1.	读取状态寄存器。	针对从器件发送的数据和主器件ACK轮询PIN位，直到该位置为有效（0）。请注意，首次通过此循环时，PIN位将保持有效（0），因为从器件是发送器。如果LRB/AD0 = 1（主器件否定应答），则转到步骤3。
2.	写入数据寄存器。	在总线上发送数据字节。转到步骤1。
3.	写入数据寄存器。	执行“假写”以将PIN位置为无效（1）。请注意，假写数据在NACK 后不会在总线上发送，并且时钟不会保持低电平。I²C/SMB控制器成为从接收器。 **结束从发送器模式。**
从接收器模式		
1.	读取状态寄存器。	针对主器件发送的数据轮询PIN位，直到该位置为有效（0）。如果检测到“停止”条件（STS = 1），则转到4。如果这不是从器件将接收的倒数第二个字节，转到步骤3。
2.	写入控制寄存器。	将控制寄存器中的ACK位更改为0，以在最后一个字节后发送NACK。
3.	读取数据寄存器。	检索接收到的数据字节。转到步骤1。
4.	读取数据寄存器。	假读数据寄存器或写入控制寄存器中的PIN位以将状态寄存器中的PIN位置为无效。 **结束从接收器模式。**

6.1.7 I²C多主器件支持

I²C/SMB控制器包括多主器件支持。仲裁依赖于所有I²C接口到该总线的“线与”连接。如果两个或两个以上主器件试图同时将信息发送到总线上，则第一个生成1的主器件（其他主器件生成0）的仲裁将失败。

多主器件支持由网络层硬件自动处理。对于仅使用I²C内核层基元的操作，多主器件支持包括第6.1.7.1节“硬件支持”中定义的硬件，并且很大程度上取决于下面访问总线、检测总线争用和仲裁失败几节所述的软件。

© 2019 Microchip Technology Inc.

6.1.7.1 硬件支持

I²C多主器件支持的硬件包括状态寄存器中的LAB和NBB位。NBB位供主器件用于访问总线。LAB位用于检测总线争用。有关这些位的定义，请参见第7.1.2节“状态寄存器”。

6.1.7.2 访问总线

通常，在状态寄存器中的NBB位未置为有效（1）前，多主器件环境下的I²C主器件不得尝试访问总线。惟一例外是首次使能时可能存在以下情况：I²C/SMB控制器可能在另一个外部I²C主器件已开始事务且未检测到总线繁忙条件后不久上电。为了避免这种情况下发生总线争用，应在每个主器件的初始化序列中引入一段等于最长I²C总线事务时间的延时。

软件负责识别总线异常，例如NBB位未同步故障或总线卡住故障（例如，使用位拆裂控制寄存器），这些故障可能会阻止或禁止报文事务。

6.1.7.3 总线访问边界条件

当内部主器件确认NBB无效并尝试开始事务时，存在总线访问边界条件。请注意，此时总线处于空闲状态，并且没有系统复位或唤醒等导致的总线同步问题。在此尝试期间（即紧接在事务启动之前），外部主器件会开始事务。

在这种情况下，典型的系统级响应将要求标准总线仲裁解决冲突，而失败的主器件正常退出，否则会检测到错误启动条件并触发总线错误（BER），这将使两个主器件的事务均无效。

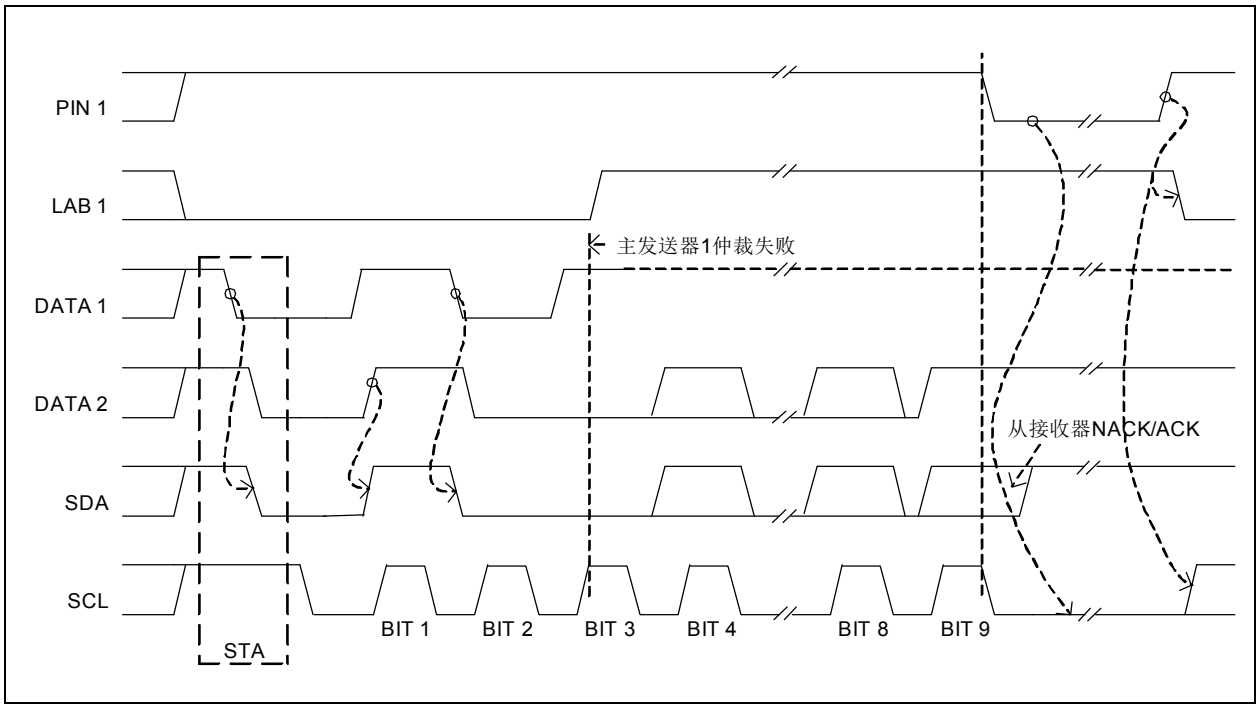
在SMBus控制器内核中，如果内部主器件在NBB有效时将控制寄存器中的STA位置为有效并且主控制器尚未参与事务（即不是重复启动条件），则LAB位将置为有效并且内部主器件总线驱动器将被禁止，以免影响正在进行的事务。由于避免了报文损坏，因此不会发生总线错误。

6.1.7.4 检测总线争用

报文事务开始后，主器件获知另一个主器件尚未同时开始事务的惟一方式是使用状态寄存器中的LAB位。仲裁在SDA线上以如下方式进行：发送高电平的主器件（另一个主器件正在发送低电平）将关闭其数据输出级，因为总线上的电平与其自身的电平不对应（见下文的图6-1）。I²C主器件驱动器负责在传输后续数据之前检测仲裁失败并终止事务。

请注意，在I²C/SMB控制器中，在数据传输协议的数据和应答阶段以及启动和重复启动条件期间，总线争用检测以主器件和从器件的形式在硬件中发生。停止期间不检查总线争用检测。

图6-1： 检测两个主器件之间的争用



6.1.7.5 仲裁失败

如果检测总线争用时发生仲裁失败，则I²C控制器成为从接收器，并使时钟线（SCL）无限期保持低电平。此时，如果主器件在仲裁中败给自身从地址，则读取主器件发送命令对应的数据寄存器或写入主器件接收命令对应的数据寄存器将清零PIN位。

仲裁失败后的操作由软件负责。例如，如果仲裁在数据传输之前失败（例如，由于从器件地址冲突），则仲裁失败的主器件通常会在一段时间后重试该事务。如果两个主器件同时从同一个从器件读取数据，并且主器件1所需的数据少于主器件2，则主器件1将在其最后一个字节应答（ACK）期间发生仲裁失败，并且会放弃已接收到所有必要数据的事务（即不生成停止条件），而主器件2则会继续完成事务。

如果启动条件期间内部主器件的仲裁失败，并且内部从器件获得地址匹配，则LAB和AAS位将在PIN有效时置为有效。这表示内部主器件在仲裁中败给寻址内部从器件的外部主器件，因此应放弃内部主器件事务。

有关仲裁和时钟同步的更多详细信息，请参见参考文档[1]。

6.1.8 输入滤波

I²C/SMB的SCLK和SDAT输入引脚上均包含一个模拟滤波器。此滤波器设计为抑制小于50 ns的脉冲，并始终使大于140 ns的脉冲通过。

默认情况下，滤波器处于禁止状态。要使能滤波器，将第39页上的配置寄存器中的FEN位置1。

© 2019 Microchip Technology Inc.

6.2 SMBus网络层接口

I²C/SMB包括两个可以自动执行主器件事务和从器件事务的网络层状态机。网络层状态机无需通过软件逐字节控制，只需向服务软件发送一个或两个中断即可完成大多数I²C或SMBus传输。主从器件状态机彼此独立，可以并行运行。各状态机通过其自身寄存器配置，主器件状态机通过主器件命令寄存器配置，从器件状态机通过从器件命令寄存器配置。主器件和从器件均使用DMA通道在片上存储器和外部总线之间移动数据。

对于主器件（内部或外部）声明I²C总线并且先发送从器件地址再发送一系列数据字节的写事务，不需要翻转I²C总线。用于主器件传输或从器件传输的网络层控制器将自动完成传输，并且可通过中断向软件发送传输已完成的信号。读传输需要翻转I²C总线。对于这些传输，主器件和从器件网络层状态机都可以暂时停止传输，在延长总线时钟的同时中断EC，直到软件重新配置DMA控制器来处理事务的读取阶段为止。

可以将网络层状态机配置为多个不同的I²C事务类型，每种事务类型都可以选择性包含PEC计算。以下是一些示例。

6.2.1 主器件示例

表6-10： 主器件SMB写字节示例

	操作	说明
1.	软件配置3个字节的存储器缓冲区： • bit 0为0的目标从器件地址 • 命令字节 • 数据字节	将从地址字节的bit 0设置为0，以便传输写事务。
2.	配置主器件DMA通道： • 源地址为步骤1中的缓冲区 • 目标地址为主器件发送缓冲寄存器 • 长度为3	
3.	允许I²C/SMB中断： • 将配置寄存器中的ENMI置1 • 允许中断聚合器中的I²C/SMB中断	在完成主器件写事务后，该位将允许对EC的中断。
4.	按以下内容设置主器件命令寄存器： • WRITECOUNT = 3 • START0 = 1 • STOP = 0 • MPROCEED = 1 • MRUN = 1	主器件命令寄存器中的所有其他位域均为0。 此命令寄存器配置在传输开始时（在从器件地址之前）将启动位置为有效，在传输结束时将停止位置为有效。 启动任何主器件传输之前，都需要将MRUN和MPROCEED设置为1。
5.	软件等待中断	如果没有错误，则写字节事务正常终止。

表6-11： 带PEC的主器件SMB写入字示例

	操作	说明
1.	软件配置4个字节的存储器缓冲区： • bit 0为0的目标从器件地址 • 命令字节 • 2个数据字节	将从地址字节的bit 0设置为0，以便传输写事务。
2.	配置主器件DMA通道： • 源地址为步骤1中的缓冲区 • 目标地址为主器件发送缓冲寄存器 • 长度为4	
3.	允许I²C/SMB中断： • 将配置寄存器中的ENMI置1 • 允许中断聚合器中的I²C/SMB中断	在完成主器件写事务后，该位将允许对EC的中断。
4.	按以下内容设置主器件命令寄存器： • WRITECOUNT = 4 • PEC_TERM = 1 • START0 = 1 • STOP = 0 • MPROCEED = 1 • MRUN = 1	主器件命令寄存器中的所有其他位域均为0。 此命令寄存器配置在传输开始时（在从器件地址之前）将启动位置为有效，在传输结束时将停止位置为有效。 启动任何主器件传输之前，都需要将MRUN和MPROCEED设置为1。
5.	软件等待中断	如果没有错误，则写入字事务正常终止。

表6-12： 主器件写块示例

	操作	说明
1.	软件配置4个字节的存储器缓冲区： • bit 0为0的目标从器件地址 • 命令字节 • 2个数据字节	将从地址字节的bit 0设置为0，以便传输写事务。
2.	配置主器件DMA通道： • 源地址为步骤1中的缓冲区 • 目标地址为主器件发送缓冲寄存器 • 长度为4	
3.	允许I²C/SMB中断： • 将配置寄存器中的ENMI置1 • 允许中断聚合器中的I²C/SMB中断	在完成主器件写事务后，该位将允许对EC的中断。

© 2019 Microchip Technology Inc.

表6-12： 主器件写块示例（续）

	操作	说明
4.	按以下内容设置主器件命令寄存器： • WRITECOUNT = 4 • PEC_TERM = 1 • START0 = 1 • STOP = 0 • MPROCEED = 1 • MRUN = 1	主器件命令寄存器中的所有其他位域均为0。 此命令寄存器配置在传输开始时（在从器件地址之前）将启动位置为有效，在传输结束时将停止位置为有效。 启动任何主器件传输之前，都需要将MRUN和MPROCEED设置为1。
5.	软件等待中断	如果没有错误，则写块事务正常终止。

表6-13： 带PEC的主器件读块示例

	操作	说明
1.	软件分配两个缓冲区： •一个3字节缓冲区 •一个*n*+2字节缓冲区	3字节缓冲区用于读块操作的写阶段。 缓冲区大小*n*是允许的最大块。额外的2个字节是从从器件接收的PEC字节和通过控制器计算出的PEC字节。
2.	软件使用读块命令的写部分填充3字节缓冲区 • 字节0是从器件地址，bit 0 = 0（用于写操作） • 字节1是命令字节 • 字节2是同一从器件地址，bit 0 = 1（用于读操作）	
3.	配置主器件DMA通道： • 源地址为步骤1中的缓冲区 • 目标地址为主器件发送缓冲寄存器 • 长度为3	
4.	允许I²C/SMB中断： • 将配置寄存器中的ENMI置1 • 允许中断聚合器中的I²C/SMB中断	当命令的写阶段完成时，将发生I²C/SMB中断。
5.	按以下内容设置主器件命令寄存器： • WRITECOUNT = 3 • READCOUNT = 1 • READ_PEC = 1 • READM = 1 • START0 = 1 • STARTN = 1 • STOP = 0 • MPROCEED = 1 • MRUN = 1	主器件命令寄存器中的所有其他位域均为0。 从器件将改写READCOUNT位域。将该值设为1就可以启动操作的读阶段。 此命令寄存器配置在传输开始时将启动位（在从器件地址之前）、重复启动位（在第二个从器件地址之前）置为有效，在传输结束时将停止位置为有效。 启动任何主器件传输之前，都需要将MRUN和MPROCEED设置为1。
6.	软件等待中断	如果没有错误，则在发生中断时，命令的写阶段将正常完成。

表6-13： 带PEC的主器件读块示例（续）

	操作	说明
7.	软件检查接收缓冲区的最后2个字节	如果最后2个字节相同，则PEC计算正确
8.	收到中断后，软件将重新配置主器件DMA通道： • 源地址为主器件接收缓冲寄存器 • 目标地址为步骤1中的*n*+2字节缓冲区 • 长度为*n* + 2	主器件状态机已暂停，但并非停止工作，而是等待软件重新配置DMA通道。总线时钟会延长，直到软件通知状态机继续运行。
9.	允许I²C/SMB中断： • 将配置寄存器中的ENMI置1 • 允许中断聚合器中的I²C/SMB中断	当命令的读阶段完成时，将发生I²C/SMB中断。
10.	将主器件命令寄存器中的MPROCEED位置1	命令寄存器中的所有其他位保持不变。将MPROCEED置1可使主器件状态机完成事务。
11.	软件等待中断	如果没有错误，则表示命令的读阶段已正常完成。

6.2.2 从器件示例

表6-14： 从器件配置

	操作	说明
1.	在自身地址寄存器中设置所需的从器件地址	
2.	软件在存储器中分配一个*n*字节的缓冲区，其大小足以容纳事务的最长写阶段。	如果为从器件事务使能了PEC，则缓冲区大小应包括两个字节，分别为接收的PEC字节和计算的PEC字节。
3.	配置从器件DMA通道： • 源地址为从器件接收缓冲寄存器 • 目标地址为步骤2中分配的缓冲区。 • 长度为*n*	如果使能了PEC，则DMA长度为*n*+2。
4.	允许I²C/SMB中断： • 将配置寄存器中的ENSI置1 • 允许中断聚合器中的I²C/SMB中断	
5.	按以下内容设置从器件命令寄存器： • SLAVE_READCOUNT = *n* • SPROCEED = 1 • SRUN = 1	如果需要PEC，还需将SLAVE_PEC置1。
6.	软件等待中断	如果没有错误，则在发生中断时，命令的写阶段将正常完成。
从器件状态机在发生中断时终止（写事务，SRUN = 0）		
7.	软件处理从主器件接收的命令	

© 2019 Microchip Technology Inc.

表6-14： 从器件配置（续）

	操作	说明
8.	转到步骤2。	重新使能从器件状态机以进行主器件的下一个事务。
从器件状态机在发生中断时工作（读事务，SRUN = 1）		
7.	软件处理从主器件接收的命令	
8.	软件在存储器中分配一个*m*字节的缓冲区，其大小足以容纳事务的读阶段。	软件准备要发送到主器件的数据
9.	配置从器件DMA通道： • 源地址为步骤8中分配的缓冲区。 • 目标地址为从器件发送缓冲寄存器。 • 长度为*m*	
10.	允许I²C/SMB中断： • 将配置寄存器中的ENSI置1 • 允许中断聚合器中的I²C/SMB中断	
11.	按以下内容设置从器件命令寄存器： • SLAVE_WRITECOUNT = *m* • SPROCEED = 1 • SRUN = 1	如果需要计算PEC，则将SLAVE_PEC置1。
12.	软件等待中断	如果没有错误，则在发生中断时，命令的读阶段将正常完成。
13.	转到步骤2。	重新使能从器件状态机以进行主器件的下一个事务。

6.2.3 错误条件

如果状态寄存器中的LAB或BER位在主器件状态机处理事务时变为1，则状态机会将MRUN清零、将完成寄存器中的MDONE位置1并停止运行。如果允许，将生成一个中断。I²C控制器内核将复位。

如果主器件状态机由于主器件的总线仲裁失败而终止，则除了状态寄存器中的LAB位之外，完成寄存器中的LAB位也会置1。软件应刷新主器件发送缓冲寄存器和主器件接收缓冲寄存器（使用FLUSH_MXBUF和FLUSH_MRBUF）并重新配置主器件事务。

6.3 DMA接口

主器件网络层状态机和从器件网络层状态机都具有与外部DMA控制器的接口。分配给主器件的DMA通道可用于在存储器与主器件发送缓冲寄存器和主器件接收缓冲寄存器之间传输数据，而分配给从器件的通道可用于在存储器与从器件发送缓冲寄存器和从器件接收缓冲寄存器之间传输数据。

DMA通道和控制器互锁，因此分配的DMA通道将在I²C/SMB终止时终止传输，I²C/SMB将在DMA通道由于总线错误或数据溢出而停止时终止事务。

6.4 MCTP公平协议支持

此器件支持MCTP公平协议。公平协议通过配置寄存器中的FAIR位使能。参考文档[5]中的图6-2“SMBus和I2C的公平仲裁时序测量值”和表6-15“100 KHz/400 KHz SMBus/I2C的公平仲裁时序值”介绍了MCTP公平协议。表中包含100 kHz SMBus和400 kHz I²C总线的时序：

图6-2： SMBUS和I²C的公平仲裁时序测量值

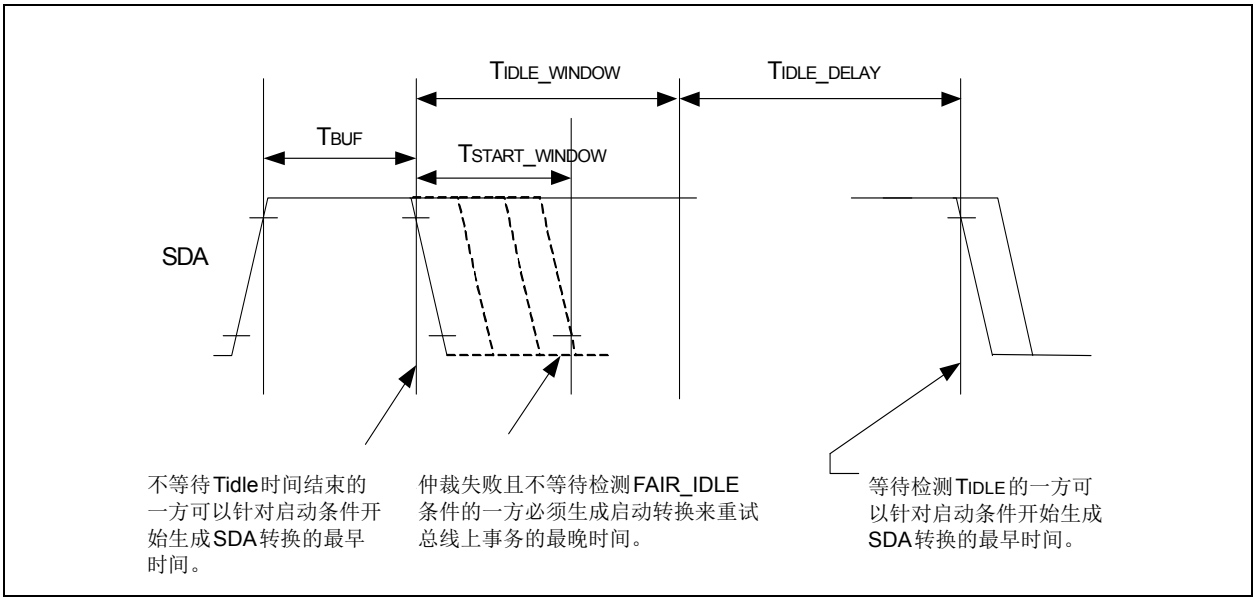


表6-15： 100 KHz/400 KHz SMBUS/I²C的公平仲裁时序值

符号	最小值	最大值	单位	注
T_{BUF}	4.7/1.3	-	µs	符合SMBus 100 kHz/I²C 400 kHz规范
T_{START_WINDOW}	-	20/4	µs	在这段时间内，如果器件在仲裁失败后尝试重新获得总线访问权，则不等待检测FAIR_IDLE条件的器件必须生成启动条件。
T_{IDLE_WINDOW}	30/5	60/20	µs	在这段时间内，等待检测FAIR_IDLE条件的器件不得检测总线繁忙条件。如果在这段间隔内未检测到总线繁忙，则存在FAIR_IDLE条件。
T_{IDLE_DELAY}	31/16	-	µs	检测FAIR_IDLE条件的器件必须等待此延时，然后才能尝试生成启动条件。该延时可适应总线上不同器件实现的TIDLE_WINDOW间隔之间的差异，此外还有额外的时间来适应正在生成启动条件的器件与正在监视启动条件的器件之间的总线偏移。这样便可确保检测到TIDLE_WINDOW的一方不会在正在检测FAIR_IDLE的其他器件完成其TIDLE窗口的检查之前生成启动条件。否则，即使发生了FAIR_IDLE条件，其他器件也无法检测到。因此，TIDLE_DELAY必须大于TIDLE_WINDOW最大值和最小值之差。

使能公平协议后，只要主器件状态机赢得SMBus仲裁，内部状态（Fair_Wait）就会置1。如果Fair_Wait不为真，则在总线空闲时间达到超时调节寄存器中的BUS_IDLE_MIN值确定的周期后，状态寄存器中的nBB位将置为无效。该值应设置为T_{BUF}周期。如果Fair_Wait为真，则NBB将在BUS_IDLE_MIN确定的周期加上FAIR_BUS_IDLE_MIN + FAIR_IDLE_DELAY的额外周期后置为无效。如果总线在BUS_IDLE_MIN + FAIR_BUS_IDLE_MIN周期到期后保持空闲，则Fair_Wait状态将清零，但之后总线空闲的周期必须达到FAIR_IDLE_DELAY才能将NBB置为无效。

© 2019 Microchip Technology Inc.

6.5 总线超时

总线超时允许I²C/SMB实现SMBus规范（参考文档[1]）的表1和第3.1.1.2节中定义的硬件“超时”。超时检测通过配置寄存器中的TCEN位使能。使能位拆裂（位拆裂控制寄存器中的BBEN位置为有效）时，不得使能超时。下表对超时进行了总结：

表6-16： 总线超时

超时	使能	状态	调节参数	参考文档[1]中的符号
总线空闲（SDA = 1）	BIDEN	CHDH	CLOCK_HIGH_TIME_OUT	$T_{HIGH\ (MAX)}$
数据保持低电平	BIDEN	CHDL	CLOCK_HIGH_TIME_OUT	$T_{HIGH\ (MAX)}$
从器件累积超时	SCEN	SCTO	SLAVE_CUM_TIME_OUT	$T_{LOW:SEXT}$
主器件累积超时	MCEN	MCTO	MASTER_CUM_TIME_OUT	$T_{LOW:MEXT}$
时钟低电平超时	DTEN	DTO	SLAVE_CUM_TIME_OUT	$T_{TIMEOUT}$

对于表6-16中列出的任何超时，如果使能位和状态位均为1，则完成寄存器中的TIMERR位置1。当发生超时时，如果控制器作为主器件或从器件主动参与事务，则完成寄存器中的BER也置1。

通常，总线超时表明存在严重问题。当发生超时时，如果控制器参与事务，则必须复位控制器。

6.5.1 器件超时

控制器用作事务中的主器件时，可以检测到参考文档[1]中定义的时钟低电平超时（也称为器件超时）。TCEN和DTEN都必须为1才能使能检测。超时周期由SLAVE_CUM_TIME_OUT值定义。使能器件超时并且发生超时时（即，包括主器件在内的任何器件均使SCL在$T_{TIMEOUT,\ MIN}$内保持低电平），主器件将使SCL在MASTER_CUM_TIME_OUT定义的额外周期内保持低电平，以确保总线上的所有器件都能检测到错误。

6.5.2 总线繁忙

控制器始终可保证停止位和后续启动位之间的空闲时间最短，与TCEN位的状态无关。空闲时间由超时调节寄存器的BUS_IDLE_MIN位域设置，对应于参考文档[1]和参考文档[2]中的时序参数T_{BUF}。在停止位后经过空闲时间后，状态寄存器中的NBB位置为无效。对于由作为主器件的控制器发起的事务以及由外部主器件发起的事务，NBB位置为无效。另外，控制器用作主器件时，在总线上出现停止位之后至少需要经过空闲周期，才会将启动位置为有效。

上电复位或软复位后的NBB同步所需的超时不包括在硬件实现的超时中，由固件负责。

7.0 寄存器接口

表7-1： I²C/SMBUS控制器寄存器汇总

偏移量（十六进制）	寄存器名称
00h	控制寄存器
	状态寄存器
04h	自身地址寄存器
08h	数据寄存器
0Ch	主器件命令寄存器
10h	从器件命令寄存器
14h	PEC寄存器
18h	重复启动保持时间寄存器
1Ch	Microchip保留
20h	完成寄存器
24h	空闲调节寄存器
28h	配置寄存器
2Ch	总线时钟寄存器
30h	模块ID寄存器
34h	版本寄存器
38h	位拆裂控制寄存器
3Ch	Microchip保留寄存器
40h	数据时序寄存器
44h	超时调节寄存器
48h	从器件发送缓冲寄存器
4Ch	从器件接收缓冲寄存器
50h	主器件发送缓冲寄存器
54h	主器件接收缓冲寄存器
58h	Microchip保留
5Ch	Microchip保留
60h	唤醒状态寄存器
64h	唤醒使能寄存器
68h	Microchip保留寄存器

不得修改标有“Microchip保留”字样的寄存器。

表7-2： 建议的编程值

20 MHz波特时钟：				
寄存器	默认值	100K的值	400K的值	1M的值（注 1：）
总线时钟寄存器	00_00_65_65h	00_00_65_65h	00_00_13_20h	00_00_09_0Ah
数据时序寄存器	0E_54_62_0Ah	0E_54_62_0Ah	0E_15_15_08h	06_08_08_01h
重复启动保持时间寄存器	00_00_00_56h	00_00_00_56h	00_00_00_0Eh	00_00_00_09h
空闲调节寄存器	02_80_02_6Ch	02_80_02_6Ch	00_14_00_64h	00_14_00_64h

注 1： MCTP被定义为在100 kHz和400 kHz下工作，但尚未定义为支持I²C增强型快速模式（见参考文档[5]）。因此，该值对于1 MHz下的I²C增强型快速模式可能没有意义。

© 2019 Microchip Technology Inc.

表7-2： 建议的编程值（续）

超时调节寄存器	30_C6_F5_FBh	30_C6_F5_FBh	0D_C6_F5_FBh	06_C6_F5_FBh
Microchip保留寄存器	7h			
16 MHz波特时钟：				
寄存器	**默认值**	**100k的值**	**400k的值**	**1M的值（注1）**
总线时钟寄存器	00_00_4F_4Fh	00_00_4F_4Fh	00_00_0F_17h	00_00_05_09h
数据时序寄存器	0C_4D_50_06h	0C_4D_50_06h	04_0A_0A_06h	04_06_06_01h
重复启动保持时间寄存器	00_00_00_4Dh	00_00_00_4Dh	00_00_00_0Ah	00_00_00_06h
空闲调节寄存器	01_FC_01_EDh	01_FC_01_EDh	01_00_00_50h	01_00_00_50h (见注1)
超时调节寄存器	4B_9C_C2_C7h	4B_9C_C2_C7h	15_9C_C2_C7h	08_9C_C2_C7h
Microchip保留寄存器	5h			
10 MHz波特时钟：				
寄存器	**默认值**	**100k的值**	**400k的值**	
总线时钟寄存器	00_00_31_31h	00_00_31_31h	00_00_0A_0Dh	N/A
数据时序寄存器	07_2A_31_04h	07_2A_31_04h	03_08_08_04h	N/A
重复启动保持时间寄存器	00_00_00_2Bh	00_00_00_2Bh	00_00_00_07h	N/A
空闲调节寄存器	01_40_01_36h	01_40_01_36h	00_A0_00_32h	N/A
超时调节寄存器	30_C6_F5_FBh	30_C6_F5_FBh	0D_C6_F5_FBh	N/A
Microchip保留寄存器	2h			

注 1： MCTP被定义为在100 kHz和400 kHz下工作，但尚未定义为支持I²C增强型快速模式（见参考文档[5]）。因此，该值对于1 MHz下的I²C增强型快速模式可能没有意义。

表7-3： 总线时钟寄存器与频率

总线频率（kHz）	总线时钟寄存器设置		
	10 MHz波特率时钟	**16 MHz波特率时钟**	**20 MHz波特率时钟**
1000	N/A	0509h	090Ah
400	0A0Dh	0F17h	1320h
333	0A12h	0F1Fh	1626h
100	3131h（默认值）	4F4Fh（默认值）	6565h（默认值）
80	3D3Eh	6363h	7C7Dh
40	7C7Ch	C7C7h	F9F9h

7.1 寄存器说明

7.1.1 控制寄存器

如果使用网络层功能（使用主器件命令寄存器和从器件命令寄存器），则无需访问该寄存器。

偏移量	00h			
Bit	**说明**	**类型**	**默认值**	**复位事件**
31:8	保留	R	-	-
7	PIN 非待处理中断位用作软件复位功能。向PIN位写入逻辑1会将所有状态位置为无效，但HBB位不受PIN位的影响。PIN位是自清零位。向该位写入逻辑0将不起作用。	W	N/A	RESET_I2C
6	ESO 使能串行输出位用于使能和禁止串行数据输出（SDA）。当ESO置为有效（1）时，SDA将使能。当ESO未置为有效（0）时，SDA将禁止。ESO位不影响对控制器中寄存器的访问。 当I²C/SMB控制器主动参与总线事务时，不得将ESO位置为无效。	W	N/A	RESET_I2C
5:4	保留	R	-	-
3	ENI 使能中断位按第5.0节“中断接口”所述控制中断接口。	W	N/A	RESET_I2C
2	STA 发送启动条件。 STA和STO位控制I²C启动条件的生成、从器件地址和R/nW位（位于数据寄存器中）的发送、重复启动条件的生成以及停止条件的生成，如表7-4中所述。	W	N/A	RESET_I2C
1	STO 发送停止条件。 STA和STO位控制I²C启动条件的生成、从器件地址和R/nW位（位于数据寄存器中）的发送、重复启动条件的生成以及停止条件的生成，如表7-4中所述。	W	N/A	RESET_I2C
0	ACK 应答。ACK通常必须置为有效（1）。这将导致控制器在每个字节后自动发送一个应答（在第9个时钟脉冲期间发生）。当控制器在主接收器模式下工作并且不需要从发送器发送更多数据时，ACK位必须置为无效（0）。这将导致I²C总线上出现否定应答，从而停止从器件的进一步发送。	W	N/A	RESET_I2C

© 2019 Microchip Technology Inc.

表7-4： 串行总线控制指令表

STA	STO	模式	功能	操作
1	0	SLV/REC	启动	发送启动条件+地址， 数据寄存器bit 0（R/nW）= 0时保持MST/TRM； 数据寄存器bit 0（R/nW）= 1时进入MST/REC。
		MST/TRM	重复启动	与SLV/REC相同
0	1	MST/REC； MST/TRM	停止读； 停止写	发送停止条件，进入SLV/REC模式。 在主接收模式下，接收的最后一个字节必须以NACK终止。
1	1	保留		保留
0	0	ANY	NOP	空操作

表7-4中未提及的所有其他STA和STO模式组合均为NOP。

7.1.2 状态寄存器

只有传统I²C操作才需要使用该寄存器。如果使用网络层功能，则无需访问该寄存器。

偏移量	00h			
Bit	**说明**	**类型**	**默认值**	**复位事件**
31:8	保留	R	-	-
7	PIN 第6.1.1节“PIN操作”介绍了PIN（待处理中断）的操作。	R	1h	RESET_I2C
6	SAD 如第6.1.2节“从器件地址解码”所述，SAD（SMBus地址解码）置为有效。	R	0h	RESET_I2C
5	STS 当检测到外部生成的停止条件时，该位置为有效（1）。该位仅在从接收器模式下使用。	R	0h	-
4	BER BER（总线错误）置为有效时，表示已检测到错误的启动/停止条件或已检测到总线超时。如果该位置为有效，则NBB（1）和PIN（0）置为无效。BER置为有效后，必须先复位控制器，然后才能采取进一步措施。	R	0h	-

偏移量	00h			
Bit	**说明**	**类型**	**默认值**	**复位事件**
3	LRB/AD0 LBR/AB0（“接收的最后一位”或“地址0”）仅在PIN位置为有效（0）时才有效。 当AAS位未置为有效（0）时（即控制器未作为从器件寻址），该位将保留通过总线接收的最后一位的值。通常，这将是从器件应答的值；因此，可通过测试该位来完成从器件应答校验。 当AAS位置为有效（1）时（即控制器已作为从器件寻址），如果接收到的从器件地址为“广播呼叫”（00h）地址，该位将设置为逻辑1；如果接收到的从器件地址与自身地址寄存器中编程的值匹配，则该位将设置为逻辑0。	R	0h	RESET_I2C
2	AAS AAS（作为从器件寻址）仅在PIN置为有效（0）时才有效。用作从器件时，当通过总线传入的地址与自身地址寄存器中的值（移位1位）匹配时或者接收到“广播呼叫”地址（00h）时（“广播呼叫”通过LRB/AD0位指示），该位置1	R	0h	RESET_I2C
1	LAB 在多主器件模式中，如果在仲裁中败给总线上的另一个主器件（如第6.1.7节“I²C多主器件支持”所述），则LAB（仲裁失败）置1。	R	0h	RESET_I2C
0	NBB 总线繁忙位指示总线何时处于使用中。零表示总线正忙，无法访问。该位由启动条件置为有效（0），在停止条件后置为无效（1）。请参见第6.5.2节“总线繁忙”。	R	1h	RESET_I2C

© 2019 Microchip Technology Inc.

7.1.3 自身地址寄存器

凭借该寄存器，I²C/SMB能够响应两个不同的从器件地址。请参见第6.1.2节“从器件地址解码”。

> **注：** 如果从器件的地址与**自身地址寄存器**中编程的值相同，内部主器件不应尝试对它件执行事务，这将被视为非法操作。如果发生这种情况，控制器必须复位。

偏移量	04h			
Bit	**说明**	**类型**	**默认值**	**复位事件**
31:15	保留	R	-	-
14:8	OWN_ADDRESS_2 该位域用于配置控制器作为从器件寻址时响应的两个地址之一。自身地址位域将偏移一位，因此如果将该位域编程为值55h，值Ahh会被识别为从器件地址。 当该位域与传入的从器件地址匹配时，状态寄存器中的AAS位将置1。 地址“0000000b”为广播呼叫地址。Own_Address 0h将与广播呼叫地址匹配，通过将配置寄存器中的GC_DIS置1来禁止该匹配时除外。	R/W	0h	RESET_I2C
7	保留	R	-	-
6:0	OWN_ADDRESS_1 该位域用于配置控制器作为从器件寻址时响应的两个地址之一。数据寄存器和自身地址位域将偏移一位，因此如果将该位域编程为值55h，值Ahh会被识别为从器件地址。 当该位域与传入的从器件地址匹配时，状态寄存器中的AAS位将置1。 地址“0000000b”为广播呼叫地址。Own_Address 0h将与广播呼叫地址匹配，通过将配置寄存器中的GC_DIS置1来禁止该匹配时除外。	R/W	0h	RESET_I2C

7.1.4 数据寄存器

偏移量	08h			
Bit	**说明**	**类型**	**默认值**	**复位事件**
31:8	保留	R	-	-
7:0	DATA 该寄存器用于保存移出到I²C端口或从该端口移入的数据。	R/W	0h	RESET_I2C

7.1.5 主器件命令寄存器

偏移量	0Ch			
Bit	**说明**	**类型**	**默认值**	**复位事件**
31:8	保留	R	-	-
31:24	READ_COUNT 该位域用于计数从I²C端口读入主器件接收缓冲寄存器的字节数，其值必须大于0，主器件状态机才能启动读阶段。每向主器件接收缓冲寄存器读入一个字节，该位域就递减1。如果READM位置1，则可以用读入的第一个字节改写该位域。 读取计数是块读取的硬限制。超出该值的任何已接收数据都将被忽略。 READ_COUNT 不包含 PEC 字节，因此从从器件读取的字节数为 READ_COUNT + READ_PEC。由于状态机在从从器件复制PEC字节后将PEC寄存器的内容复制到存储器中，因此复制到存储器中的字节数为READ_COUNT + 2XREAD_PEC。	R/W	0h	RESET_I2C
23:16	WRITE_COUNT 该位域用于计数从主器件发送缓冲寄存器发送到I²C/SMBus端口的字节数。主器件发送缓冲寄存器每向端口写入一个字节，该位域就递减1。	R/W	0h	RESET_I2C
15:14	保留	R	-	-
13	READ_PEC 如果该位为0，当READ_COUNT达到0时，将停止读取I²C/SMBus端口。如果该位为1，当READ_COUNT 为0 后，将继续再读取一个字节。这允许控制器在已读取一个M字节的块后再读取从外部器件返回的PEC。读取PEC字节后，该位域将清零。	R/W	0h	RESET_I2C
12	READM 如果该位为1，则当READ_COUNT为1时，READ_COUNT位域将替换为从I²C/SMBus端口读取的字节。READ_COUNT更新后，该位将清零。	R/W	0h	RESET_I2C
11	PEC_TERM 如果该位为1，则当WRITE_COUNT为0时，将发送PEC寄存器的副本。发送PEC寄存器数据后，PEC寄存器和该位都将清零。	R/W	0h	RESET_I2C
10	停止 如果此位为1，则在事务完成后发送停止位。	R/W	0h	RESET_I2C
9	STARTN 如果该位为1，则在发送WRITE_COUNT数据的最后一个字节之前发送一个启动位。	R/W	0h	RESET_I2C
8	START0 如果该位为1，则在发送WRITE_COUNT数据的第一个字节之前发送一个启动位。	R/W	0h	RESET_I2C
7:2	保留	R	-	-

© 2019 Microchip Technology Inc.

偏移量	0Ch			
Bit	**说明**	**类型**	**默认值**	**复位事件**
1	MPROCEED 为了启动主器件I²C/SMB事务，该位和该寄存器中的MRUN位都必须置1。当主器件状态机停止时，由于事务已完成，或者由于事务已达到需要软件协助的程度，该位将由硬件清零。 如果状态机将该位清零，但使MRUN保持置1（例如，在读取传输中的重复启动条件后，需要软件重新配置主器件DMA通道），则软件将通过向该位写入1来重新启动事务。要完成事务，MRUN必须保持置1。 软件不得向该位写入0。	R/W	0h	RESET_I2C
0	MRUN 为了启动主器件I²C/SMB事务，该位和该寄存器中的MPROCEED位都必须置1。当主器件事务成功完成或由于错误条件而完成时，硬件会将该位清零。 软件不得向该位写入0。	R/W	0h	RESET_I2C

7.1.6 从器件命令寄存器

从器件命令寄存器配置控制器作为从器件时的响应方式。

偏移量	10h			
Bit	**说明**	**类型**	**默认值**	**复位事件**
31:24	保留	R	-	-
23:16	SLAVE_READCOUNT 每次将一个字节从DATA寄存器复制到从器件接收缓冲寄存器时，该位域就递减1。当该位域递减到0时，状态机将不应答任何额外接收的数据。 如果该位域中的值比字节数小1并且外部主器件正在进行发送，则重复启动地址有可能得不到应答。SLAVE_READ_COUNT须保持足够大的值，以便满足外部主器件的最大预期传输要求。外部主器件发出停止位后，计数错误不会导致从器件在随后的启动中不应答自身的从器件地址。	R/W	0h	RESET_I2C

偏移量	10h			
Bit	**说明**	**类型**	**默认值**	**复位事件**
15:8	SLAVE_WRITECOUNT 该位域设置为软件要发送到主器件的字节数。如果该位域在外部主器件请求数据且从器件发送缓冲寄存器为空时大于0，则状态机会延长I²C/SMBus时钟，直至缓冲区不为空。如果该位域在主器件请求数据且缓冲寄存器已满时大于0，则相应寄存器的内容将复制到DATA 寄存器中且该位域将递减 1。如果在主器件请求数据时，SLAVE_WRITE_COUNT和SLAVE_PEC均为0，则从器件会将从器件发送缓冲寄存器的内容重新发送到主器件中。如果发生这种情况，完成寄存器中的SPROT位将置1。	R/W	0h	RESET_I2C
7:3	保留	R	-	-
2	SLAVE_PEC 如果在主器件请求数据时，SLAVE_WRITE_COUNT为0且该位为1，则PEC寄存器的内容将复制到DATA寄存器中。PEC寄存器的内容发送到主器件后，寄存器将清零，并且该位将设置为0。	R/W	0h	RESET_I2C
1	SPROCEED 为了启动从器件I²C/SMB事务，该位和该寄存器中的SRUN位都必须置1。当主器件状态机停止时，由于事务已完成，或者由于事务已达到需要软件协助的程度，该位将由硬件清零。 如果状态机将该位清零，但使SRUN保持置1（例如，在读或写传输中的重复启动条件后，需要软件重新配置从器件DMA通道），则软件将通过向该位写入1来重新启动事务。要完成事务，SRUN必须保持置1。 软件不得向该位写入0。	R/W	0h	RESET_I2C
0	SRUN 为了启动从器件I²C/SMB事务，该位和该寄存器中的SPROCEED位都必须置1。当从器件事务成功完成或由于错误条件而完成时，硬件会将该位清零。 软件不得向该位写入0。	R/W	0h	RESET_I2C

7.1.7 PEC寄存器

偏移量	14h			
Bit	**说明**	**类型**	**默认值**	**复位事件**
31:8	保留	R	-	-
7:0	PEC SMBus数据包错误校验。有关详细信息，请参见第6.1.4节“硬件PEC支持”。	R/W	0h	RESET_I2C

© 2019 Microchip Technology Inc.

7.1.8 重复启动保持时间寄存器

偏移量	18h			
Bit	**说明**	**类型**	**默认值**	**复位事件**
31:8	保留	R	-	-
7:0	RPT_START_HOLD_TIME 这是重复启动位的I²C规范中时序要求$t_{Hd:Sta}$的值。此位域用于保存时钟，直到满足重复启动位的保持时间为止。 有关不同时钟速率下该位域的建议值，请参见表7-2“建议的编程值”。	R/W	请参见表7-2	RESET_I2C

7.1.9 完成寄存器

偏移量	20h			
Bit	**说明**	**类型**	**默认值**	**复位事件**
31	SDONE 从器件完成。 1 = 从器件事务已完成 0 = 从器件事务未完成	R/WC	0h	RESET_I2C
30	MDONE 1 = 主器件事务已完成 0 = 主器件事务未完成	R/WC	0h	RESET_I2C
29	IDLE 满足以下条件时，该位置1： • 总线变为空闲状态（在状态寄存器中的NBB的上升沿） • 允许空闲中断后，总线处于空闲状态 • 使能控制器后，总线处于空闲状态（配置寄存器中的ENAB位置1）	R/WC	0h	RESET_I2C
28:26	保留	R	-	-
25	MTR 主器件发送/接收。该位用于报告主器件状态机将MDONE置为有效时所处的阶段。 1 = 主器件刚刚完成事务的发送阶段 0 = 主器件刚刚完成事务的接收阶段	R	0h	RESET_I2C
24	MNAKX 主器件在发送时接收到NACK 1 = 当主器件通过I²C/SMBus端口发送数据时，外部从器件向主器件发送一个NACK信号 0 = 主器件发送期间未接收到NACK信号	R/WC	0h	RESET_I2C
23:22	保留	R	-	-

偏移量	20h			
Bit	**说明**	**类型**	**默认值**	**复位事件**
21	REPEAT_WRITE 1 = 从器件状态机暂停，因为它检测到重复启动写操作（包含从器件地址的字节的bit[0]为0） 0 = 未检测到重复写操作	R/WC	0h	RESET_I2C
20	REPEAT_READ 1 = 从器件状态机暂停，因为它检测到重复启动读操作（包含从器件地址的字节的bit[0]为1） 0 = 未检测到重复读操作	R/WC	0h	RESET_I2C
19	SPROT 从器件协议写计数错误 1 = 从器件命令寄存器中的WRITE_COUNT位域递减计数至0后外部主器件发送NACK信号，或者在计数器达到0前从器件收到NACK信号 0 = 未发生写计数错误	R/WC	0h	RESET_I2C
18	保留	R	-	-
17	STR 从器件发送/接收 1 = 从器件刚刚完成事务的接收阶段 0 = 从器件刚刚完成事务的发送阶段	R	0h	RESET_I2C
16	SNAKR 从器件在接收过程中发送NACK信号 1 = 当从器件从I²C/SMBus端口接收数据时，从器件状态机向外部发送主器件发送NACK信号 0 = 未向主器件发送NACK信号	R/WC	0h	RESET_I2C
15	保留	R	-	-
14	LAB 仲裁失败状态。如果状态寄存器中的LAB位在主器件或从器件事务进行过程中置1，则该位将置1。	R/WC	0h	RESET_I2C
13	BER 总线错误状态。如果状态寄存器中的BER位在主器件或从器件事务进行过程中置1，则该位将置1。	R/WC	0h	RESET_I2C
12	CHDH 时钟高电平数据高电平超时状态。这是总线空闲超时状态。请参见第6.5节“总线超时”。 1 = 发生了超时 0 = 未发生超时	R/WC	0h	RESET_I2C

© 2019 Microchip Technology Inc.

偏移量	20h			
Bit	**说明**	**类型**	**默认值**	**复位事件**
11	CHDL 时钟高电平数据低电平超时状态。如果在SDA保持低电平时，SCL保持高电平，则该状态将置为有效。请参见第6.5节“总线超时”。 1 = 发生了超时 0 = 未发生超时	R/WC	0h	RESET_I2C
10	SCTO 从器件累积超时状态。请参见第6.5节“总线超时”。 1 = 发生了超时 0 = 未发生超时	R/WC	0h	RESET_I2C
9	MCTO 主器件累积超时状态。请参见第6.5节“总线超时”。 1 = 发生了超时 0 = 未发生超时	R/WC	0h	RESET_I2C
8	DTO 器件超时状态。请参见第6.5节“总线超时”。 1 = 发生了超时 0 = 未发生超时	R/WC	0h	RESET_I2C
7	保留	R	-	-
6	TIMERR 检测到超时错误。如果任何超时状态位（CHDH、CHDL、MCTO和DTO）均置1并且通过其各自的超时使能位使能，则该位为1。	R	0h	RESET_I2C
5	BIDEN 该位用于使能总线空闲超时校验，如第6.5节“总线超时”所述。 1 = 使能超时校验 0 = 禁止超时校验	R/W	0h	RESET_I2C
4	SCEN 该位用于使能从器件累积超时校验，如第6.5节“总线超时”所述。 1 = 使能超时校验 0 = 禁止超时校验	R/W	0h	RESET_I2C
3	MCEN 该位用于使能主器件累积超时校验，如第6.5节“总线超时”所述。 1 = 使能超时校验 0 = 禁止超时校验	R/W	0h	RESET_I2C

偏移量	20h			
Bit	**说明**	**类型**	**默认值**	**复位事件**
2	DTEN 该位用于使能器件超时校验，如第6.5节“总线超时”所述。 1 = 使能超时校验 0 = 禁止超时校验	R/W	0h	RESET_I2C
1:0	保留	R	-	-

7.1.10 空闲调节寄存器

偏移量	24h			
Bit	**说明**	**类型**	**默认值**	**复位事件**
31:28	保留	R	-	-
27:16	FAIR_IDLE_DELAY 该位域用于建立MCTP T_{IDLE_DELAY}周期。使能MCTP公平协议时，该位域将导致状态机检测到一个空闲周期（FAIR BUS IDLE MIN）后，主器件事务的启动时间被额外延迟一个周期（T_{IDLE_DELAY}）。该位域用于定义编程延时所需的波特率时钟周期数。 有关不同时钟速率下该位域的建议值，请参见表7-2“建议的编程值”。	R/W	请参见表7-2	RESET_I2C
15:12	保留	R	-	-
11:0	FAIR_BUS_IDLE_MIN 总线空闲周期应编程为表6-15“100 KHz/400 KHz SMBus/I2C的公平仲裁时序值”中定义的T_{IDLE_WINDOW}时间。该位域用于定义满足公平协议所需的波特率时钟周期数。 有关不同时钟速率下该位域的建议值，请参见表7-2“建议的编程值”。	R/W	请参见表7-2	RESET_I2C

© 2019 Microchip Technology Inc.

7.1.11 配置寄存器

偏移量	28h			
Bit	**说明**	**类型**	**默认值**	**复位事件**
31	ENSI 允许从器件完成中断。 1 = 允许从器件完成中断 0 = 禁止从器件完成中断	R/W	0h	RESET_I2C
30	ENMI 允许主器件完成中断。 1 = 允许主器件完成中断 0 = 禁止主器件完成中断	R/W	0h	RESET_I2C
29	ENIDI 允许空闲中断。如果主器件命令寄存器中的MRUN或从器件命令寄存器中的SRUN置1，则不得使用该中断。 1 = 允许空闲中断 0 = 禁止空闲中断	R/W	0h	RESET_I2C
28	ENABLE_AAS 作为从器件中断寻址的中断允许。如果状态寄存器中的AAS位置为有效，则该中断置为有效。 1 = 允许AAS中断 0 = 禁止AAS中断	R/W	0h	RESET_I2C
27:20	保留	R	-	-
19	FLUSH_MRBUF 向该位写入1将清零主器件接收缓冲寄存器并将其标记为空。该位是自清零位。写入0不起作用。	W	0h	RESET_I2C
18	FLUSH_MXBUF 向该位写入1将清零主器件发送缓冲寄存器并将其标记为空。该位是自清零位。写入0不起作用。	W	0h	RESET_I2C
17	FLUSH_SRBUF 向该位写入1将清零从器件接收缓冲寄存器并将其标记为空。该位是自清零位。写入0不起作用。	W	0h	RESET_I2C
18	FLUSH_SXBUF 向该位写入1将清零从器件发送缓冲寄存器并将其标记为空。该位是自清零位。写入0不起作用。	W	0h	RESET_I2C
15	保留	R	-	-
14	GC_DIS 广播呼叫禁止 1 = 使能作为从器件时的广播呼叫地址（0h）响应。 0 = 禁止作为从器件时的广播呼叫地址（0h）响应。	R/W	0h	RESET_I2C
13	Microchip保留 必须始终写为0。	R/W	0h	RESET_I2C

偏移量	28h			
Bit	**说明**	**类型**	**默认值**	**复位事件**
12	FAIR 1 = 使能MCTP公平协议 0 = 禁止MCTP公平协议	R/W	0h	RESET_I2C
11	DSA 解码SMBus地址。 1 = 从控制器将匹配自身地址寄存器中的从器件地址、广播呼叫地址（如果通过GC_DIS使能）和两个SMBus定义的地址，即SMBus主机地址（0001_000b）和SMBus设备默认地址（110_001b） 0 = 从控制器将仅匹配自身地址寄存器中的从器件地址和广播呼叫地址（如果通过GC_DIS使能）	R/W	0h	RESET_I2C
10	ENAB I²C/SMBus控制器使能。 1 = 使能控制器 0 = 禁止控制器且控制器处于最低功耗状态	R/W	0h	RESET_I2C
9	RESET 当RESET置为有效（1）时，RESET位自身以外的所有逻辑和寄存器都将初始化为上电默认状态。RESET不是自清零位，必须在至少一个波特率时钟周期的时间内保持有效。 RESET置为有效时进行的寄存器读操作将返回默认寄存器值。 RESET位在主系统复位时自行复位，而不是基于模块复位信号RESET_I2C进行复位。	R/W	0h	RESET_EC
8	FEN 输入滤波使能。请参见第6.1.8节“输入滤波”。根据I²C规范，如果不支持外部滤波，则需要进行输入滤波。 1 = 使能输入滤波 0 = 禁止输入滤波	R/W	0h	RESET_I2C
7	PCEN PEC使能位。仅当状态寄存器中的PIN位置为无效或I²C/SMB控制器不参与事务时，才能更改PCEN。 1 = 使能硬件PEC支持 0 = 禁止硬件PEC支持	R/W	0h	RESET_I2C
6	Microchip保留 必须始终写为0。	R/W	0h	RESET_I2C

© 2019 Microchip Technology Inc.

偏移量	28h			
Bit	**说明**	**类型**	**默认值**	**复位事件**
5	SLOW_CLOCK 1 = 总线时钟寄存器的基本周期乘以4，因此频率除以4。它不会影响其他时序计算（例如数据时序寄存器或超时调节寄存器中的计算） 0 = 总线时钟寄存器的基本周期为控制器的波特率时钟周期	R/W	0h	RESET_I2C
4	TCEN 时序校验使能。 1 = 使能总线超时（如第6.5节“总线超时”所述） 0 = 禁止总线超时	R/W	0h	RESET_I2C
3:0	PORT_SEL 该位域在16个可用的总线端口中选择一个用于当前工作的I²C/SMBus端口。未选择的总线端口保持非活动状态。有关每个I²C/SMBus控制器的端口配置，请参见具体器件的数据手册。	R/W	0h	RESET_I2C

注： 在将该寄存器中的ENAB位设置为1b之前，应将该寄存器中的所有位域设置为所需的值。否则可能会导致信号接口出现故障，进而造成状态机行为不正确。

7.1.12 总线时钟寄存器

总线时钟寄存器用于确定控制器作为主器件时的I²C/SMBus总线时钟。总线时钟周期由以下公式确定：

公式7-1： $\mathbf{BUS_CLOCK_{PERIOD} = ((LOW_PERIOD+1) + (HIGH_PERIOD+1)) \times I2C_BAUD_CLOCK_{PERIOD}}$

表7-3“总线时钟寄存器与频率”列出了几种理想总线频率下总线时钟寄存器的相应设置示例。

仅当控制寄存器中的ESO位未置为有效时，才能更改总线时钟寄存器，即不得在事务进行过程中更改寄存器。为了遵循I²C、SMBus和MCTP规范，可能需要在更改总线速度时调整表7-2“建议的编程值”中列出的寄存器。

如果配置寄存器中的SLOW_CLOCK位置1，则$I2C_BAUD_CLOCK_{PERIOD}$项将乘以4，表7-3“总线时钟寄存器与频率”中所示的频率将除以4。

偏移量	2Ch			
Bit	**说明**	**类型**	**默认值**	**复位事件**
31:16	保留	R	-	-
15:8	HIGH_PERIOD 该位域用于定义I²C/SMBus总线时钟的高电平阶段所包含的I2C_BAUD_CLOCK周期数。时钟周期数比该位域的值大1。 当HIGH_PERIOD和LOW_PERIOD均为00h时，总线时钟被禁止，主器件事务将不会发生。	R/W	请参见表7-2	RESET_I2C
7:0	LOW_PERIOD 该位域用于定义I²C/SMBus总线时钟的低电平阶段所包含的I2C_BAUD_CLOCK周期数。时钟周期数比该位域的值大1	R/W	请参见表7-2	RESET_I2C

7.1.13 模块ID寄存器

偏移量	30h			
Bit	**说明**	**类型**	**默认值**	**复位事件**
31:8	保留	R	-	-
7:0	ID 模块ID	R	11h	RESET_I2C

7.1.14 版本寄存器

偏移量	34h			
Bit	**说明**	**类型**	**默认值**	**复位事件**
31:8	保留	R	-	-
7:0	REVISION 模块版本号	R	最新版本	RESET_I2C

7.1.15 位拆裂控制寄存器

偏移量	38h			
Bit	**说明**	**类型**	**默认值**	**复位事件**
31:7	保留	R	-	-
6	BBDATI 位拆裂数据输入。该位始终返回SDA引脚的状态。	R	1h	RESET_I2C
5	BBCLKI 位拆裂时钟输入。该位始终返回SCL引脚的状态。	R	1h	RESET_I2C

© 2019 Microchip Technology Inc.

偏移量	38h			
Bit	**说明**	**类型**	**默认值**	**复位事件**
4	BBDAT 位拆裂模式数据状态。 1 = SDA引脚处于三态 0 = 如果BBEN = 1且DADIR = 1，则SDA引脚驱动为低电平。如果BBEN = 1或DADIR = 1，则SDA引脚处于三态	R/W	0h	RESET_I2C
3	BBCLK 位拆裂模式时钟状态。 1 = SCL引脚处于三态 0 = 如果BBEN = 1且DADIR = 1，则SCL引脚驱动为低电平。如果BBEN = 1或DADIR = 1，则SCL引脚处于三态	R/W	0h	RESET_I2C
2	DADIR 位拆裂模式数据方向。 1 = 如果BBEN = 1，则SDA引脚为输出 0 = 如果BBEN = 1，则SDA引脚为输入	R/W	0h	RESET_I2C
1	CLDIR 位拆裂模式时钟方向。 1 = 如果BBEN = 1，则SCL引脚为输出 0 = 如果BBEN = 1，则SCL引脚为输入	R/W	0h	RESET_I2C
0	BBEN 位拆裂模式使能。 1 = 使能位拆裂模式 0 = 禁止位拆裂模式	R/W	0h	RESET_I2C

7.1.16 MICROCHIP保留寄存器

偏移量	3Ch			
Bit	**说明**	**类型**	**默认值**	**复位事件**
31:8	保留	R	-	-
7:0	MICROCHIP_RSVD 不得写入该寄存器，否则可能发生不良后果。	R/W	请参见表7-2	RESET_I2C

7.1.17 数据时序寄存器

表7-2“建议的编程值”列出了数据时序寄存器的建议值。

偏移量	40h			
Bit	**说明**	**类型**	**默认值**	**复位事件**
31:24	FIRST_START_HOLD 该位域用于确定第一个启动位传输期间SDA驱动为低电平后的SCL保持时间。它是初始启动位的I²C规范的参数$T_{HD:STA}$。重复启动保持时间由重复启动保持时间寄存器确定。	R/W	请参见表7-2	RESET_I2C
23:16	STOP_SETUP 该位域用于确定重复启动条件下自SCL上升沿起的SDA建立时间。它是I²C规范中的参数$T_{SU:STOP}$。	R/W	请参见表7-2	RESET_I2C
16:8	RESTART_SETUP 该位域用于确定重复启动条件下自SCL上升沿起的SDA建立时间。它是I²C规范中的参数$T_{SU:STA}$。	R/W	请参见表7-2	RESET_I2C
7:0	DATA_HOLD 该位域用于确定SCL驱动为低电平后的SDA保持时间。它是I²C规范中的参数$T_{HD:DAT}$。	R/W	请参见表7-2	RESET_I2C

7.1.18 超时调节寄存器

第6.5节“总线超时”介绍了超时和总线繁忙计算。表7-2“建议的编程值”列出了超时调节寄存器的建议值。

超时MASTER_CUM_TIME_OUT、SLAVE_CUM_TIME_OUT和CLOCK_HIGH_TIME_OUT仅在SMBus规范中定义，未在I²C规范中定义。由于SMBus仅被定义为在100 KHz下工作，因此这些超时仅由100 KHz或以下总线速度的规范定义。

Baud_Clock_Period为I2C_BAUD_CLOCK的周期时间。

偏移量	44h			
Bit	**说明**	**类型**	**默认值**	**复位事件**
31:24	BUS_IDLE_MIN 该位域用于确定最短总线空闲时间（也定义为停止条件和启动条件之间的时间）。它是I²C规范中的参数T_{BUF}，在此定义为 BUS_IDLE_MIN × Baud_Clock_Period	R/W	请参见表7-2	RESET_I2C
23:16	MASTER_CUM_TIME_OUT 该位域用于确定主器件累积超时持续时间，即SMBus规范中的参数$T_{LOW:MEXT}$，在此定义为 MASTER_CUM_TIME_OUT × Baud_Clock_Period × 512 仅当控制器用作主器件时才会检测到该超时。检测到该超时时，不会放弃活动事务；软件可以选择这样执行。	R/W	请参见表7-2	RESET_I2C

© 2019 Microchip Technology Inc.

偏移量	44h			
Bit	**说明**	**类型**	**默认值**	**复位事件**
16:8	SLAVE_CUM_TIME_OUT 该位域用于确定从器件累积超时持续时间，即SMBus规范中的参数$T_{LOW:SEXT}$，在此定义为 SLAVE_CUM_TIME_OUT × Baud_Clock_Period × 1024 如果在控制器用作从器件时检测到该超时，则会放弃事务并释放总线。如果在控制器用作主器件时检测到该超时，则不会放弃事务，以配合无法实现累积超时的从器件。	R/W	请参见表7-2	RESET_I2C
7:0	CLOCK_HIGH_TIME_OUT 时钟高电平超时周期，即SMBus规范中的参数T_{HIGH}，在此定义为 CLOCK_HIGH_TIME_OUT × Baud_Clock_Period × 2	R/W	请参见表7-2	RESET_I2C

7.1.19 从器件发送缓冲寄存器

偏移量	48h			
Bit	**说明**	**类型**	**默认值**	**复位事件**
31:8	保留	R	-	-
7:0	SLAVE_TRANSMIT_BUFFER	R/W	0h	RESET_I2C

7.1.20 从器件接收缓冲寄存器

偏移量	4Ch			
Bit	**说明**	**类型**	**默认值**	**复位事件**
31:8	保留	R	-	-
7:0	SLAVE_RECEIVE_BUFFER	R/W	0h	RESET_I2C

7.1.21 主器件发送缓冲寄存器

如果主器件事务由于仲裁失败而终止，则软件应使主器件发送缓冲区为空（使用配置寄存器中的FLUSH_MXBUF）。

偏移量	50h			
Bit	**说明**	**类型**	**默认值**	**复位事件**
31:8	保留	R	-	-
7:0	MASTER_TRANSMIT_BUFFER	R/W	0h	RESET_I2C

7.1.22 主器件接收缓冲寄存器

偏移量	54h			
Bit	**说明**	**类型**	**默认值**	**复位事件**
31:8	保留	R	-	-
7:0	MASTER_RECEIVE_BUFFER	R/W	0h	RESET_I2C

7.1.23 唤醒状态寄存器

偏移量	60h			
Bit	**说明**	**类型**	**默认值**	**复位事件**
31:1	保留	R	-	-
0	START_BIT_DETECTION 如果在使能控制器时检测到启动位，则该位置1。 写入1时，该位清零。写0不起作用。	R/WC	0h	RESET_I2C

7.1.24 唤醒使能寄存器

偏移量	64h			
Bit	**说明**	**类型**	**默认值**	**复位事件**
31:1	保留	R	-	-
0	START_DETECT_INT_EN 允许启动位检测中断。启动位检测中断可唤醒。 1 = 允许启动位检测中断 0 = 禁止启动位检测中断	R/W	0h	RESET_I2C

© 2019 Microchip Technology Inc.

8.0 SMBUS接口时序

图8-1： I²C/SMBUS时序

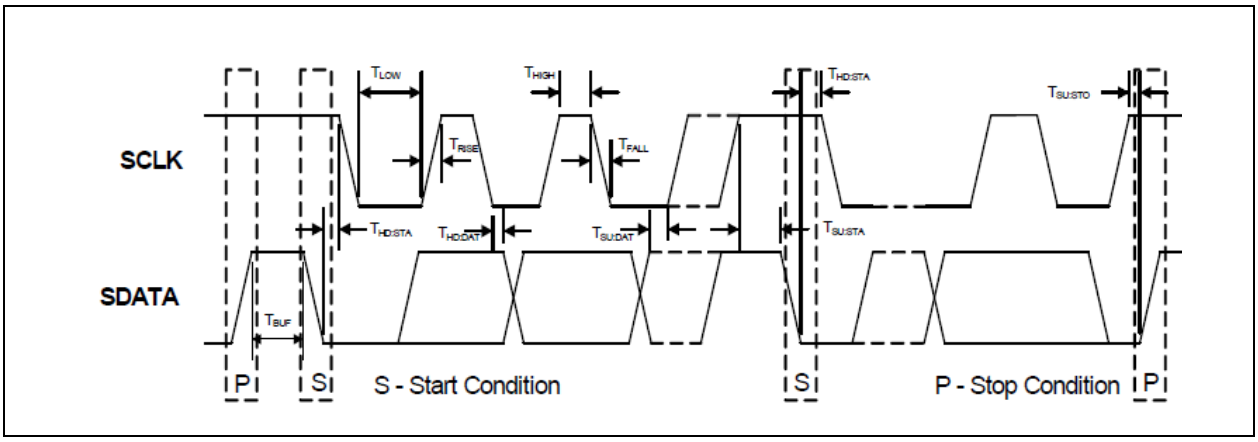


表8-1： I²C/SMBUS时序参数

符号	参数	标准模式		快速模式		单位
		最小值	最大值	最小值	最大值	
T_{BUF}	总线空闲时间	4.7		1.3		µs
$T_{SU;STA}$	启动条件建立时间	4.7		0.6		µs
$T_{HD;STA}$	启动条件保持时间	4.0		0.6		µs
T_{LOW}	SCL低电平时间	4.7		1.3		µs
T_{HIGH}	SCL高电平时间	4.0		0.6		µs
T_{RISE}	SCL和SDA上升时间		1.0		0.3	µs
T_{FALL}	SCL和SDA下降时间		0.3		0.3	µs
$T_{SU;DAT}$	数据建立时间	0.25		0.1		µs
$T_{HD;DAT}$	数据保持时间	0		0		µs
$T_{SU;STO}$	停止条件建立时间	4.0		0.6		µs

© 2019 Microchip Technology Inc.

附录A： 数据手册版本历史

表A-1： 版本历史

版本	节/图/条目	修正
DS00002379A（02-24-17）	文档初始版本	

© 2019 Microchip Technology Inc.

MICROCHIP 网站

Microchip 网站 （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。我们的网站提供以下内容：

- **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** —— 常见问题解答 （FAQ）、技术支持请求、在线讨论组以及 Microchip 设计伙伴计划成员名单
- **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，请访问 www.microchip.com/pcn，然后按照注册说明进行操作。

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 （ESE）
- 技术支持

客户应联系其代理商、代表或 ESE 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 http://microchip.com/support 获得网上技术支持。

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 愿与那些注重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。

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，在 Microchip 知识产权保护下，不得暗中或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、 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 Incorporated 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、 Precision Edge、ProASIC、ProASIC Plus、ProASIC Plus 徽标、 Quiet-Wire、 SmartFusion、 SyncWorld、 Temux、 TimeCesium、 TimeHub、TimePictra、TimeProvider、Vite、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、 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 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 和 Symmcom 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。

© 2019, Microchip Technology Incorporated 版权所有。

ISBN：978-1-5224-5258-4

© 2019 Microchip Technology Inc.

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: 45-4450-2828
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 - Gothenberg
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

© 2019 Microchip Technology Inc.