

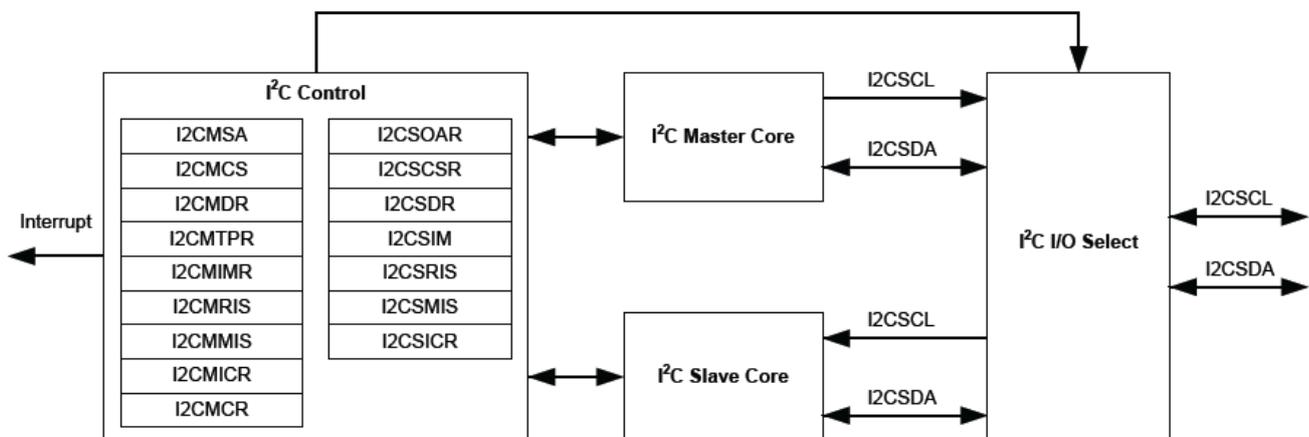
15 内部集成电路(I²C)接口

内部集成电路(I²C)采用两根线的设计(一根串行数据线 SDA 和一根串行时钟线 SCL)来提供双向的数据传输。并连接到外部的 I²C 设备如串行存储器(RAM 和 ROM)、网络设备、LCD、音频发生器等。I²C 总线也被用在产品的开发和生产过程中作为系统测试和诊断。LM3S9B96 具有两个 I²C 模块来与其他的 I²C 设备进行交互(包括发送数据和接收数据)。

Stellaris® LM3S9B96 包含的两个 I²C 模块具有如下特性:

- 在 I²C 总线上的设备可以被设计为主机或从机
 - 在主机或从机模式下都支持发送和接受数据
 - 支持它们作为主机和从机的同步操作
- 四种 I²C 的模式
 - 主机传送
 - 主机接收
 - 从机传送
 - 从机接收
- 两种传输速度: 标准(100Kbps)和快速(400Kbps)
- 主机和从机产生中断
 - 主机因为传送或接收数据结束(或者是因为错误而取消)产生中断
 - 从机在主机向其发送数据或发出请求时, 或检测到 START 或 STOP 信号时产生中断。
- 主机带有仲裁和时钟同步功能, 支持多主机以及 7 位寻址模式

15.1 结构图

图 15.1 I²C 模块结构图

15.2 引脚信号描述

表 15.1 和表 15.2 列出了 I²C 接口的外部引脚，并且逐一描述了它们的功能。I²C 接口的引脚是一些 GPIO 引脚的备用功能，并且复位时默认为 GPIO 功能。但是 I2C0SCL 和 I2C0SDA 引脚除外，它俩默认的位 I²C 功能。在下表中“引脚复用/分配”一列中列出了 I²C 可以使用的 GPIO 引脚。GPIO 备用功能选择(GPIOAFSEL)寄存器中的 AFSEL 位必须置位才能使用 I²C 功能。后面括号中的数字是要使用 I²C 功能 GPIO 控制寄存器(GPIOCTL)相应的 PMCn 位域里的编码值。注意：I2C 引脚必须设置为开漏，即将 GPIO 开漏选择寄存器(GPIOODR)里相应的位置位。配置 GPIO 更多的信息请见“通用输入/输出端口”一章。

表 15.1 I²C 的引脚(100LQFP)

引脚名称	编号	引脚复用/分配	引脚类型	缓冲类型 ^a	描述
I2C0SCL	72	PB2(1)	I/O	OD	I ² C 模块 0 时钟线
I2C0SDA	65	PB3(1)	I/O	OD	I ² C 模块 0 数据线
I2C1SCL	14	PJ0(11)	I/O	OD	I ² C 模块 1 时钟线
	19	PG0(3)			
	26	PA0(8)			
	34	PA6(1)			
I2C1SDA	18	PJ1(11)	I/O	OD	I ² C 模块 1 数据线
	27	PG1(3)			
	35	PA1(8)			
	87	PA7(1)			

a、标有 TTL 的表明该引脚兼容 TTL 电平

表 15.2 I²C 的引脚(108BGA)

引脚名称	编号	引脚复用/分配	引脚类型	缓冲类型 ^a	描述
I2C0SCL	A11	PB2(1)	I/O	OD	I ² C 模块 0 时钟线
I2C0SDA	E11	PB3(1)	I/O	OD	I ² C 模块 0 数据线

表 15.2 I²C 的引脚(108BGA)(续)

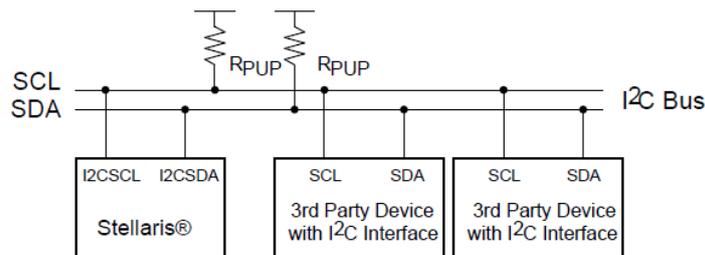
引脚名称	编号	引脚复用/分配	引脚类型	缓冲类型 ^a	描述
I2C1SCL	F3 K1 L3 L6	PJ0(11) PG0(3) PA0(8) PA6(1)	I/O	OD	I ² C 模块 1 时钟线
I2C1SDA	K2 M3 M6 B6	PG1(3) PA1(8) PA7(1) PJ1(11)	I/O	OD	I ² C 模块 1 数据线

a、标有 TTL 的表明该引脚兼容 TTL 电平

15.3 功能描述

每一个I²C模块由主机和从机两个模块组成，这两个功能均可作为独立的外设来实现。对于正确的操作，SDA和SCL管脚必须连接到双向的开漏引脚。典型的I²C配置如 [图 15.2](#)所示。

I²C 的时序图详细请见 1322 页。

图 15.2 I²C 总线配置

15.3.1 I²C 总线功能概述

I²C 总线只有两个跟线：SDA 和 SCL，在 Stellaris®控制器上被命名为 I²CSDA 和 I²CSCL。SDA 是双向的串行数据线，SCL 是双向的串行时钟线。当两根线都处于高电平的时候，总线处于空闲状态。

在I²C总线上的每一次传输包含 9 位，包含 8 个数据位和一个应答位。每次传输的字节数(从一个有效的起始条件开始到一个有效的结束条件为止的时间，详细在“起始和停止条件”中描述，[见 791 页](#))是有限制的。但是每个字节后面都要跟一个应答位，并且必须先传输数据的MSB位。当接收器不能完整接收另一个字节时，它可以保持时钟线SCL为低电平，并迫使发送器进入等待状态。当接收器释放了时钟线SCL的时候，数据传输得以继续进行。

15.3.1.1 起始和停止条件

I²C总线协议定义了两种状态来启动和停止传输：开始和结束。当SCL为高电平时，SDA线由高到低的跳变被定义为开始信号；当SCL为高电平的时候，SDA线由低到高的跳变被定义

为结束信号。总线在起始条件之后被视为忙状态，在停止条件之后被视为空闲(free)状态。见 图 15.3。

图 15.3 开始和结束条件



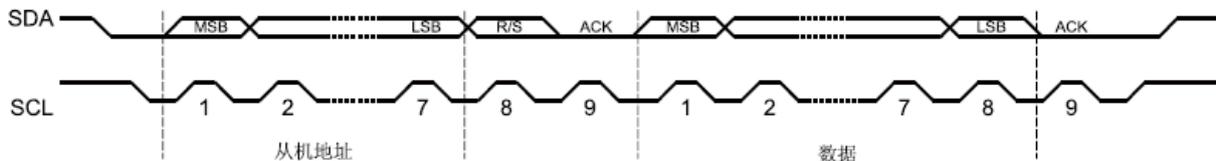
STOP 位决定了是结束数据传输周期还是重新开始一个开始条件。要生成一个单一传输周期，要在 I²C 主从机地址寄存器(I2CMSA)写入所需的地址，将 R/S 位清零，并且还要控制寄存器写入 ACK=X(0 或 1)，STOP=1，START=1，RUN=1 来执行操作和停止。当操作结束(或因错误而中止)，中断引脚变得活跃，此时可以从 I2CMDR 寄存器读取数据。当 I²C 模块处于主机接收模式的时候，通常将 ACK 位置位，这样，在每个字节后 I²C 总线控制器都会自动传输一个应答信号。当主机不再需要从机传输数据的时候，该位必须清零。

在处于从机模式的时候，I2CSRIS 寄存器里的两位指示了检测 I²C 总线上的开始和停止条件，而 I2CSMIS 寄存器里的两位则允许开始和停止条件向控制器提交中断(如果中断被使能的话)。

15.3.1.2 带有 7 位地址的数据格式

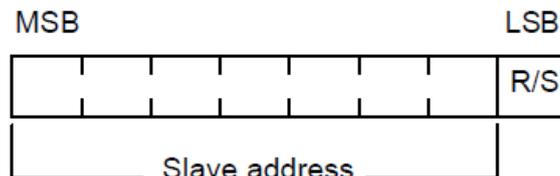
数据的传输格式如 图 15.4 所示，开始条件之后，接着传输从机地址。地址共有 7 位，紧接着的第 8 位是数据传输方向为(I2CMSA寄存器的R/S位)。如果R/S位是 0，表明它是一个传输操作(发送)，如果 R/S 位为 1 则表明它是一个需要数据(接收)。数据传输总是由主机生成一个停止条件终止的，然而，主机可以在没有产生停止信号的时候，通过再产生一个开始信号和总线上另一个设备的地址，来与另一个设备通信。因此，在一次传输过程中可能会存在各种不同组合的接收/发送格式。

图 15.4 带有 7 位地址的完整的数据传输



第一个字节的前七位构成了从机的地址(见 图 15.5)，第八位决定数据传输的方向。R/S 位的值为 0 意味着主机将会传输(发送)数据给选定的从机，如果该位的值为 1 则表明主机将从从机那接收数据。

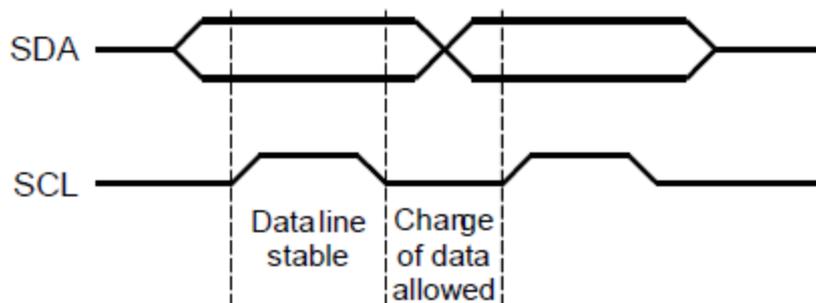
图 15.5 第一个字节的 R/S 位



15.3.1.3 数据有效性

在时钟的高电平期间，SDA线上的数据必须稳定。数据线上的数据只有在时钟线为低电平的时候才能改变。见 [图 15.6](#)。

图 15.6 在 I2C 总线的位传输过程中的数据有效性



15.3.1.4 应答

总线上所有传输都要带有应答时钟周期，该时钟周期由主机产生。发送器(可以是主机或从机)在应答周期过程中释放SDA线，即SDA为高电平。为了响应传输，接收器必须在应答时钟周期过程中拉低SDA。接收器在应答周期过程中发出的数据必须符合“数据有效性”中说明的数据有效性的要求([见 793 页](#))。

当从机不能响应从机地址时，从机必须将 SDA 线保持在高电平状态，使得主机可产生停止条件来中止当前的传输。如果主机在传输过程中用作接收器，那么它有责任应答从机发出的每次传输。由于主机控制着传输中的字节数，因此它通过在最后一个数据字节上不产生应答来向从机发送器指示数据的结束。然后从机发送器必须释放 SDA 线，以便主机可以产生停止条件或重复起始条件。

15.3.1.5 仲裁

只有在总线空闲时，主机才可以启动传输。在起始条件的最少保持时间内，两个或两个以上的主机都有可能产生起始条件。在这些情况下，当 SCL 为高电平时仲裁机制在 SDA 线上产生。在仲裁过程中，第一个竞争的主机在 SDA 上设置 ‘1’ (高电平)，而另一个主机发送 ‘0’ (低电平)，则发送 ‘1’ 的这个主机将关闭其数据输出阶段并退出直至总线再次空闲。

仲裁可以在多个位上发生。仲裁的第一个阶段是比较地址位，如果两个主机都试图寻址相同的器件，则仲裁继续比较数据位。

15.3.2 可用的速度模式

I²C 总线可以运行在普通模式(100kbps)和快速模式(400kbps)。所选的模式必须和总线上的其他 I²C 设备相匹配。选择速度模式是由 I²C 主定时器周期寄存器(I2CMTPR)里的值决定的，该值决定了 SCL 的速率是标准的 100kbps 还是快速的 400kbps。

I²C 时钟速率由这四个值决定的 CLK_PRD, TIMER_PRD, SCL_LP 和 SCL_HP:

CLK_PRD 系统时钟周期
 SCL_LP SCL 时钟的低电平阶段(固定为 6)
 SCL_HP SCL 时钟的高电平阶段(固定为 4)
 TIMER_PRD I2CMTPR 寄存器里设定的值

I²C 时钟周期按如下计算:

$$\text{SCL_PERIOD} = 2 \times (1 + \text{TIMER_PRD}) \times (\text{SCL_LP} + \text{SCL_HP}) \times \text{CLK_PRD}$$

例如:

$$\text{CLK_PRD} = 50\text{ns}$$

$$\text{TIMER_PRD} = 2$$

$$\text{SCL_LP} = 6$$

$$\text{SCL_HP} = 4$$

产生的 SCL 频率是:

$$1/\text{SCL_PERIOD} = 333 \text{ Khz}$$

[表 15.3](#)给出了在不同系统时钟速率下, 产生标准和快速SCL频率所需的定时器周期的实例

表 15.3 I²C 主定时器不同速率模式的实例

系统时钟	定时器周期	标准模式	定时器周期	快速模式
4MHz	0x01	100kbps	-	-
6 MHz	0x02	100kbps	-	-
12.5 MHz	0x06	89kbps	0x01	312kbps
16.7 MHz	0x08	93kbps	0x02	278kbps
20 MHz	0x09	100kbps	0x02	333kbps
25 MHz	0x0C	96.2kbps	0x03	312kbps
33 MHz	0x10	97.1kbps	0x04	330kbps
40 MHz	0x13	100kbps	0x04	400kbps
50 MHz	0x18	100kbps	0x06	357kbps
80 MHz	0x27	100kbps	0x09	400kbps

15.3.3 中断

I2C 在发生下列事件时可以产生中断:

- 主机传输完毕
- 主机仲裁丢失
- 主机发送错误
- 从机接收完成
- 从机请求传输
- 总线检测到结束条件
- 总线检测到开始条件

I²C 主机和 I²C 从机具有独立的 interrupt 信号。然而两种模式下都能因为多种情况产生 interrupt, 但只有一个 interrupt 信号进入 interrupt 控制器。

15.3.3.1 I²C 主机中断

I²C 主机模式在传输完毕(包括发送和接收)会产生中断, 当仲裁丢失或传输过程中发生错误也都会产生中断。要使能 I²C 主机的中断, 软件必须将 I²C 主机中断屏蔽寄存器(I2CMIMR)里的 IM 位置位。当中断发生时, 软件必须检测 I²C 主机控制/状态寄存器(I2CMCS)里的 ERROR 和 ARBLST 位, 来验证在过去的传输过程中没有发生错误, 并确保仲裁没有丢失。如果最后的应答信号不是由从机发出, 则可断定有错误发生。如果没有检测到错误的发生, 并且主机没有丢失仲裁, 应用成员就可以处理传输。可以向主机中断清除寄存器(I2CMICR)的 IC 位写入 1 来清除中断。

如果应用程序不需要中断, 那么原始的中断状态总是可以从 I²C 主机原始中断状态寄存器(I2CMRIS)看到。

15.3.3.2 I²C 从机中断

从机模式可以在已接收完数据或是需要从主机接收数据的时候产生中断。通过将 I²C 从机中断屏蔽寄存器(I2CSIMR)的 DATAIM 位置位来使能从机中断。软件通过检查 I²C 从机控制/状态寄存器(I2CSCSR)的 RREQ 和 TREQ 位来决定模块是否应该从 I²C 数据寄存器(I2CSDR)写入(发送)或读取(接收)数据。如果从机模块处于接收状态, 并且已经从发送器接收到了第一个字节, 则 FBR 和 RREQ 位一起置位。通过向 I²C 中断清除寄存器(I2CSICR)的 DATAIC 位置位来清除中断。

另外, 从机模式时, 在检测到开始和结束信号的时候也可以产生中断。要使能这些中断, 需要将 I²C 中断屏蔽寄存器(I2CSIMR)的 STARTIM 和 STOPIM 位置位, 并且要向 I²C 中断清除寄存器(I2CSICR)的 STOPIC 和 STARTIC 位写 1 来清除中断中断标志。

如果应用程序不需要中断, 那么原始的中断状态总是可以从 I²C 从机原始中断状态寄存器(I2CSRIS)看到。

15.3.4 回送操作

该 I²C 模块可放置到内部回送模式以用于诊断或调试工作。这通过置位 I²C 主机配置寄存器(I2CMCR)的 LPBK 位来完成。在回送模式中, 主机和从机模块的 SDA 和 SCL 信号结合在一起。

15.3.5 命令序列流程图

本节将详细介绍作在主机和从机模式下执行各种 I²C 传输类型的步骤。

15.3.5.1 I²C 主机命令序列

下列图中展示了作为 I²C 主机可用的命令序列

图 15.7 主机单次发送

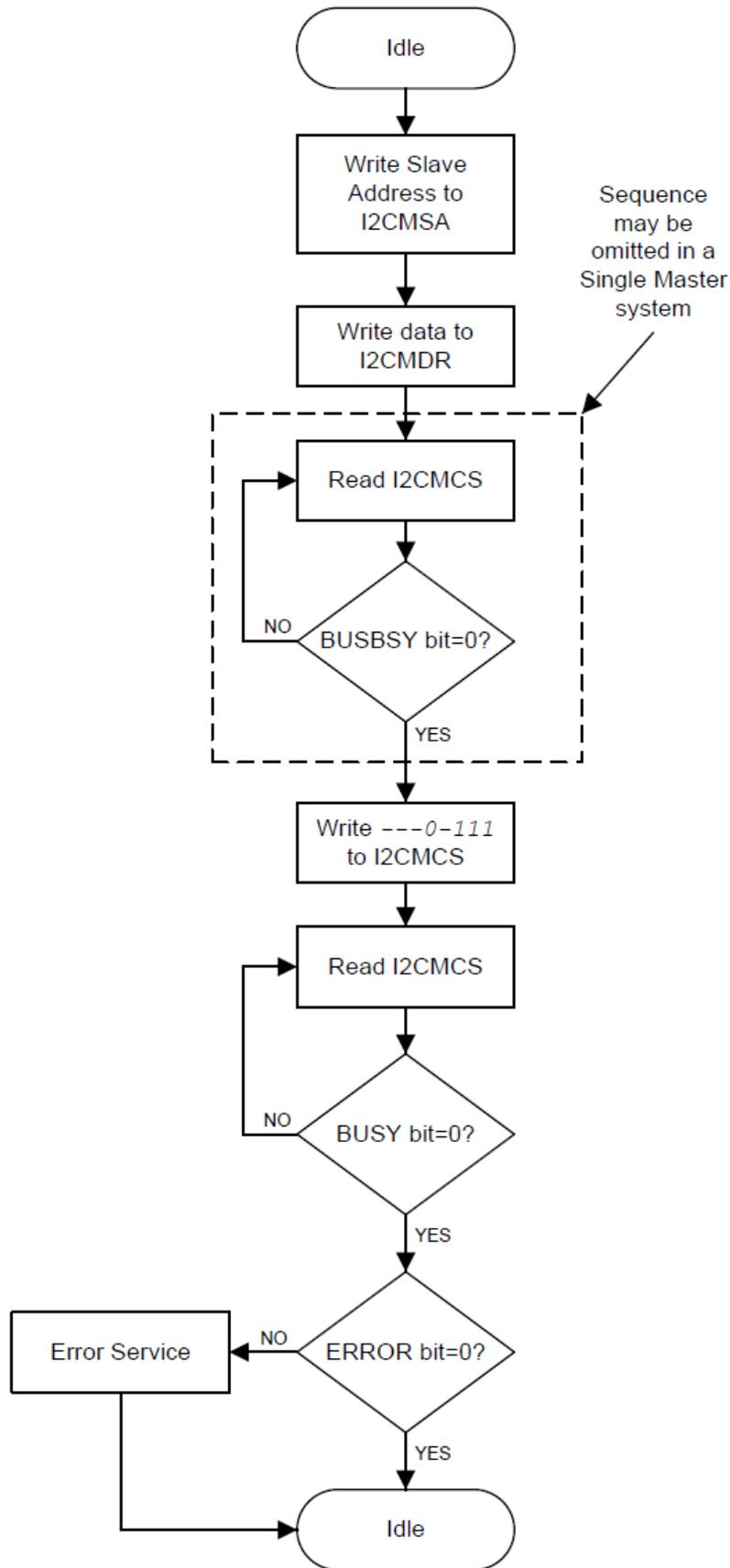


图 15.8 主机单次接收

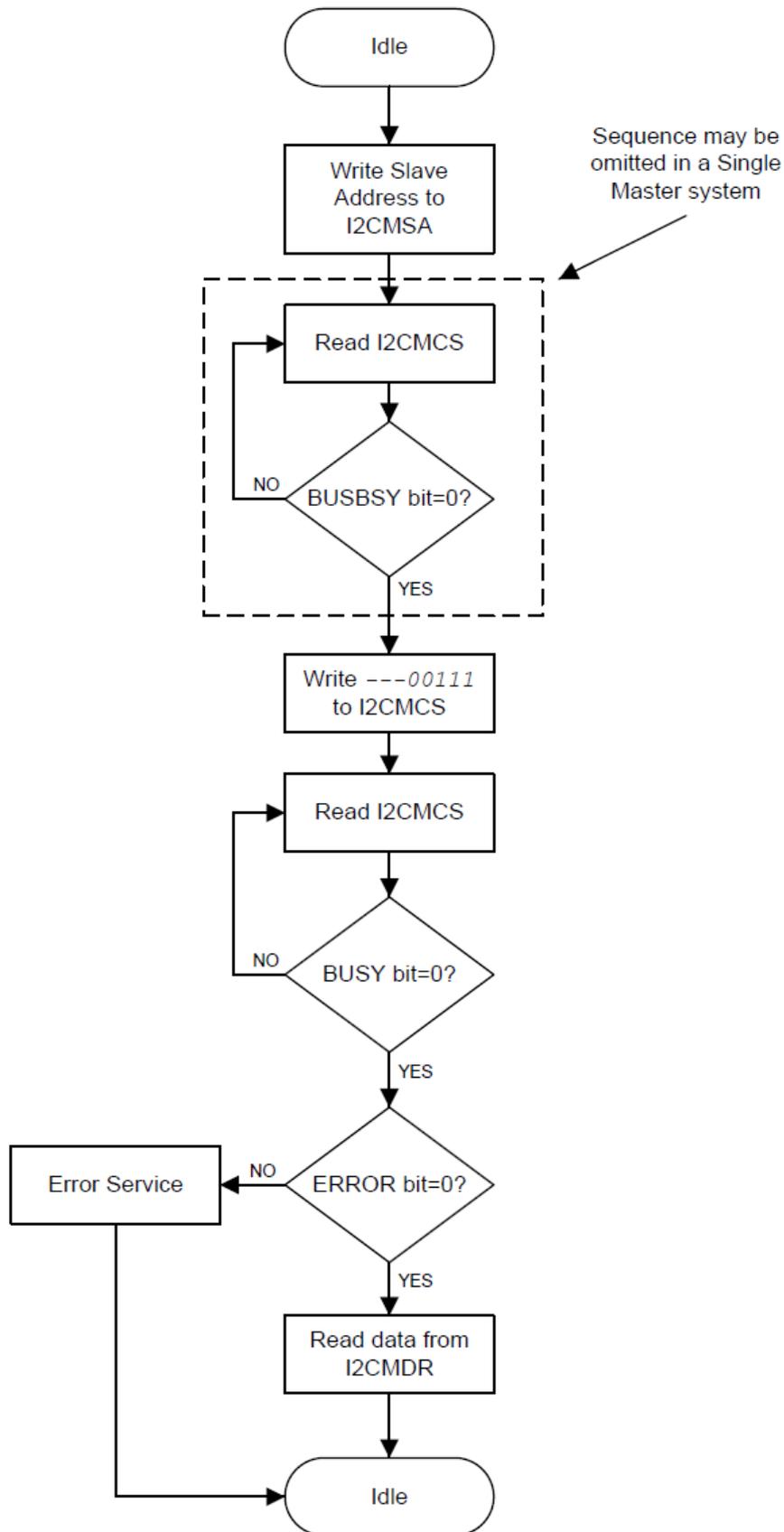


图 15.9 主机重复启动传输

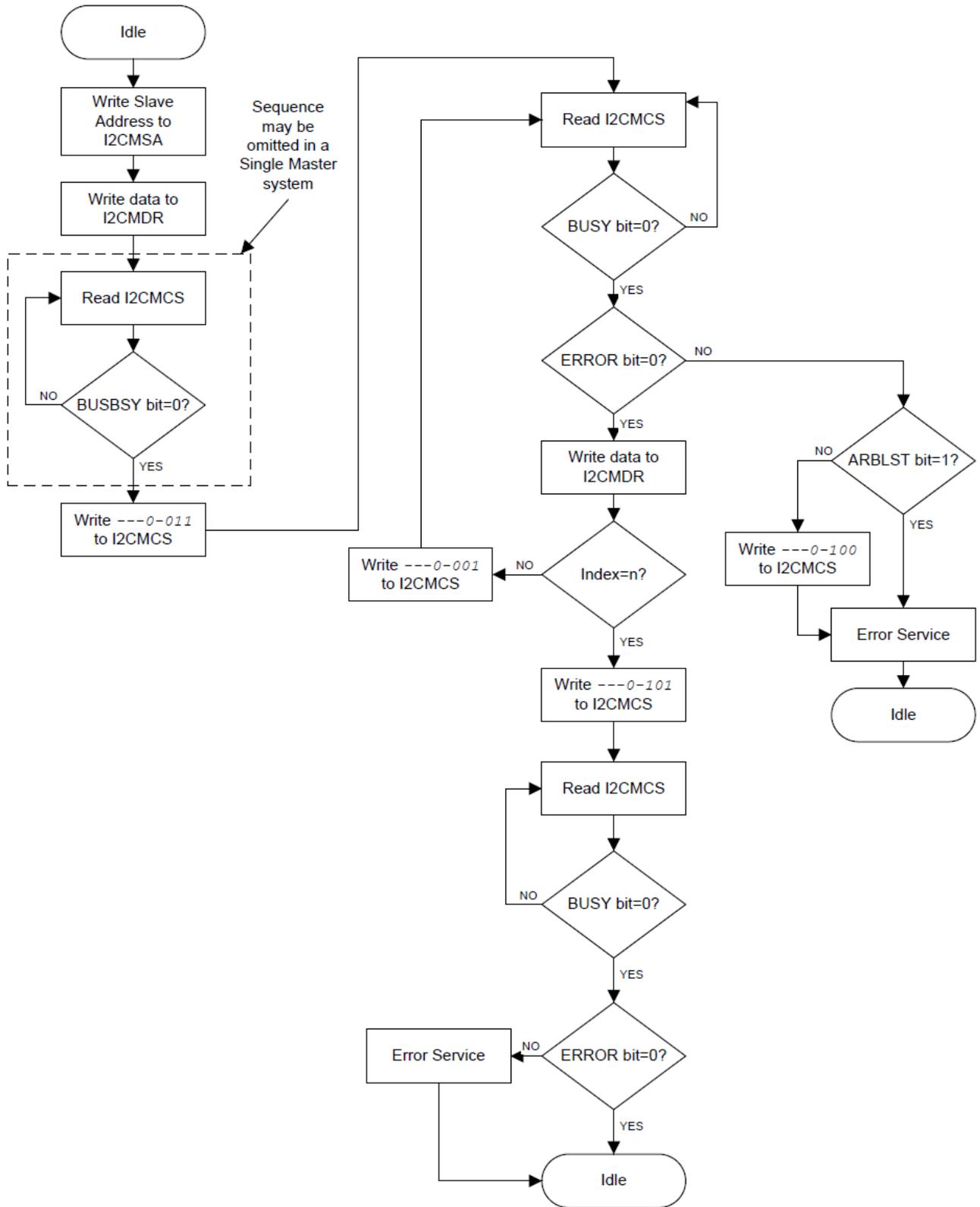


图 15.10 主机重复启动接收

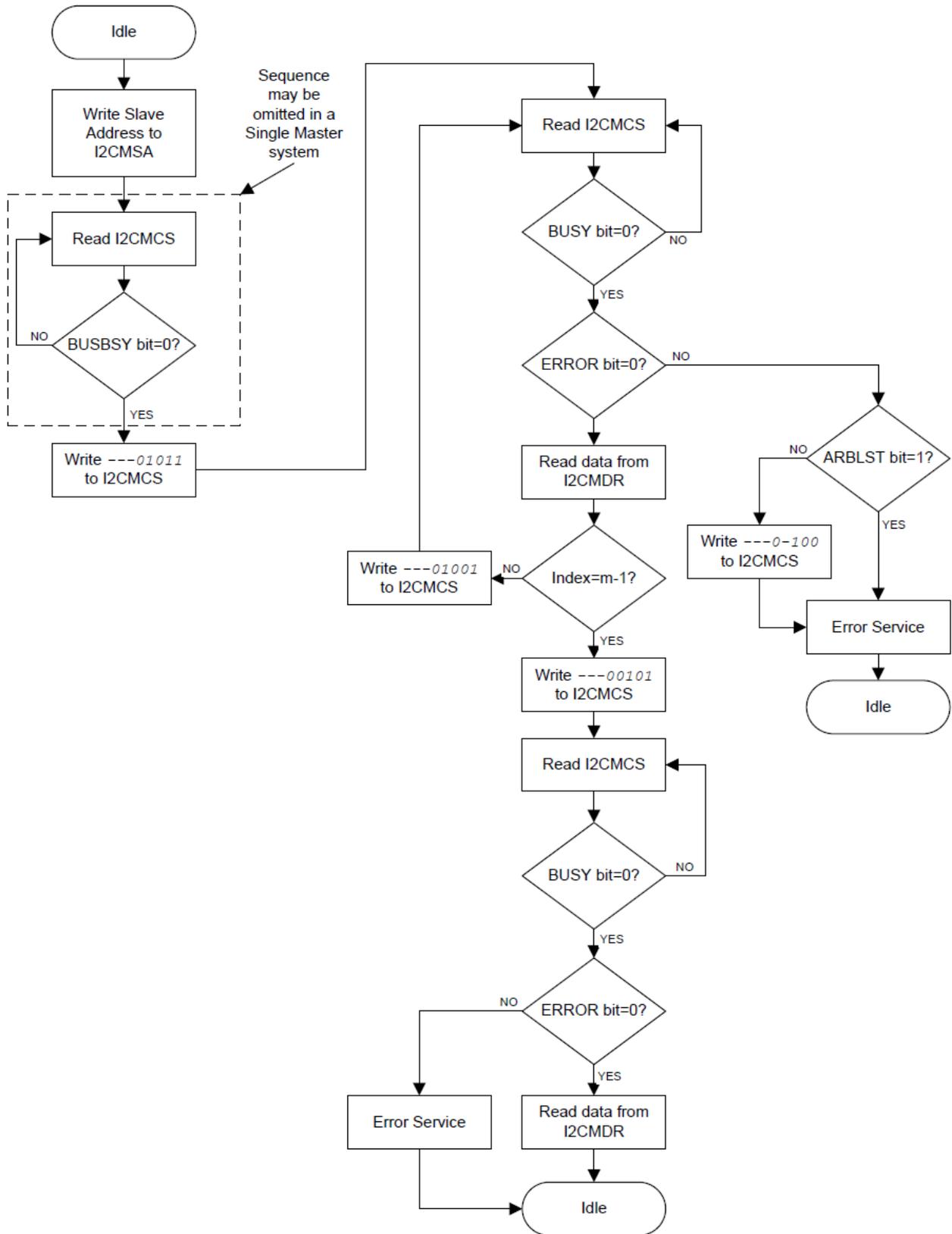


图 15.11 主机在重复启动传输后重复启动接收

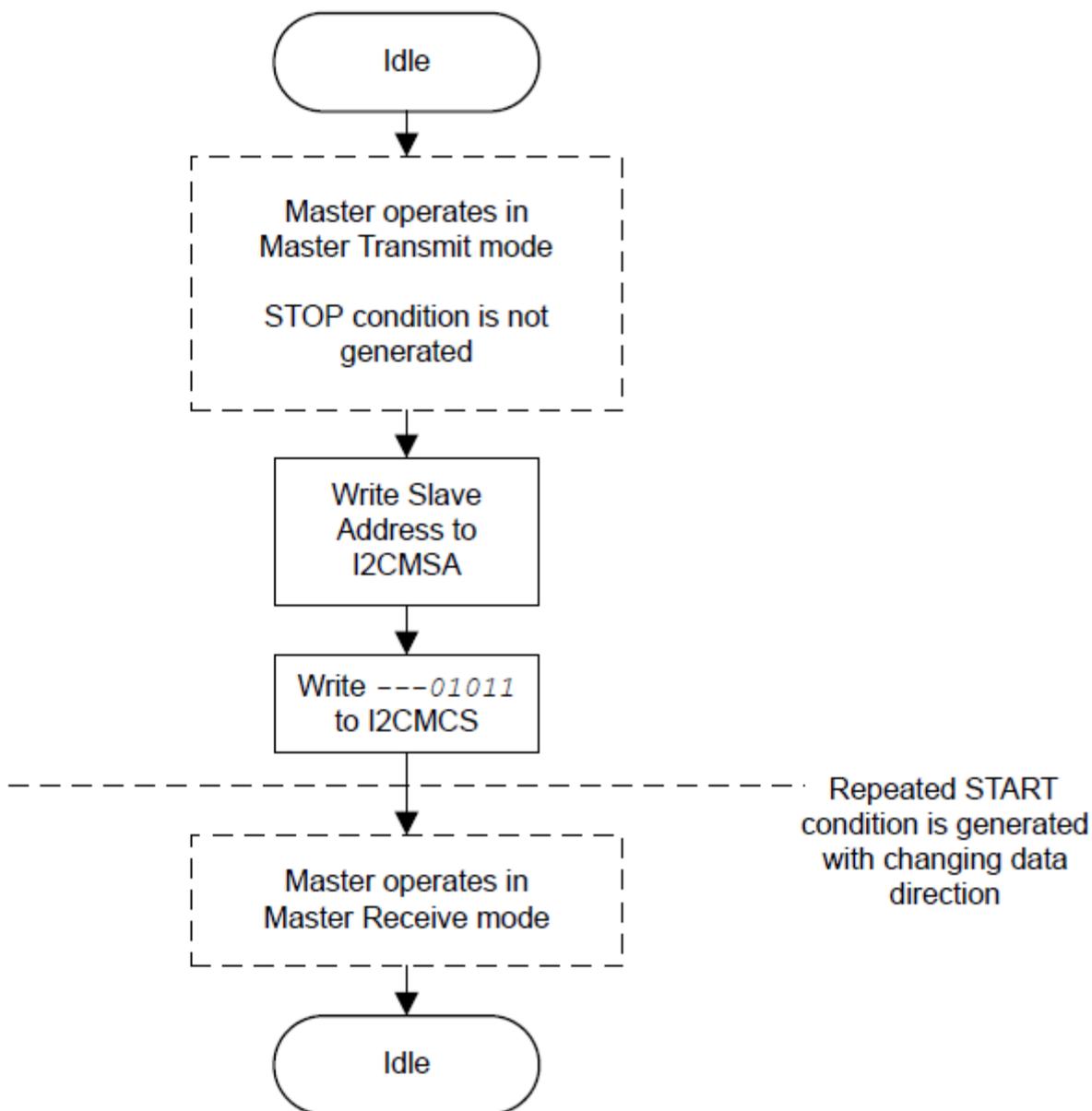
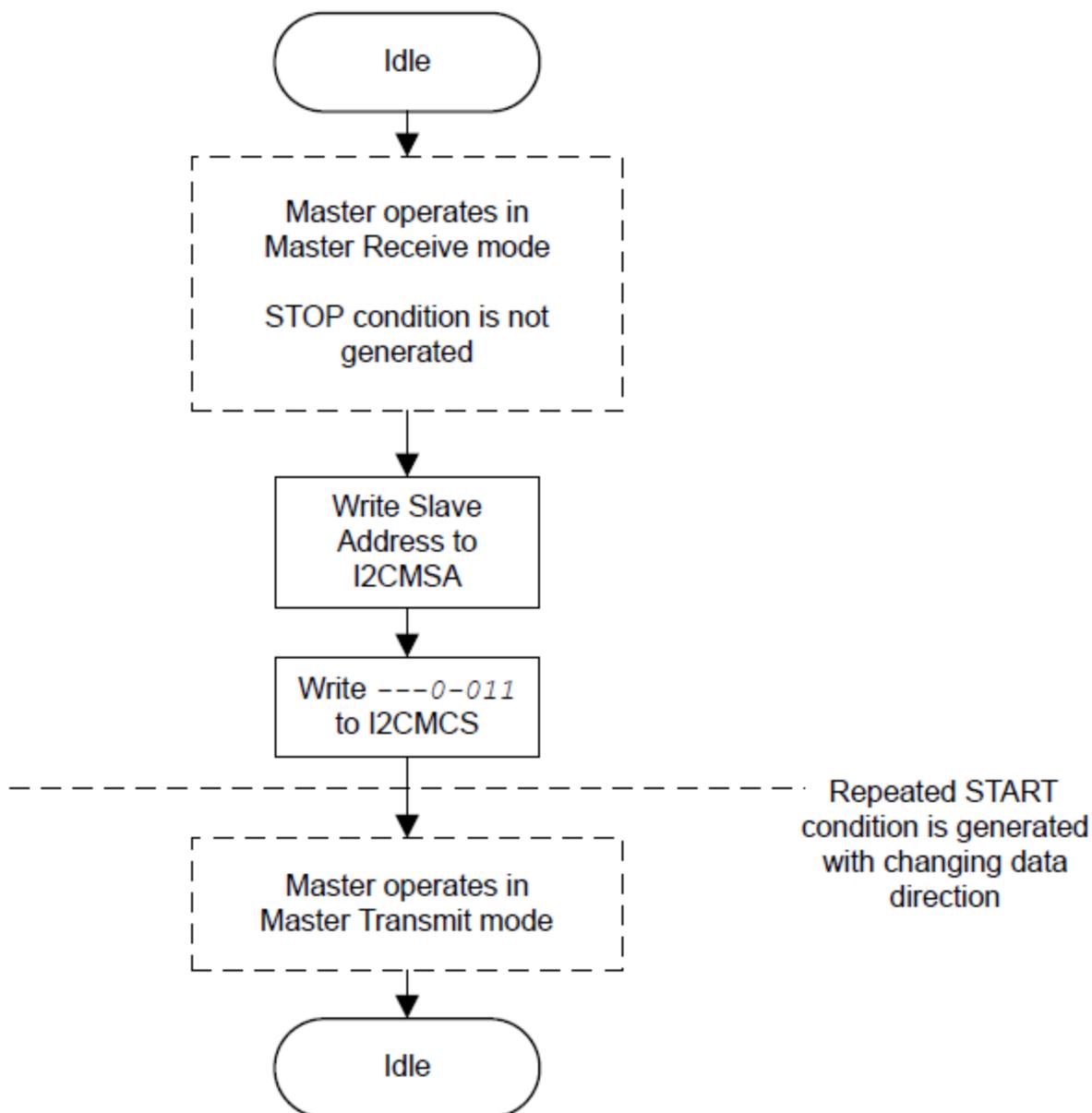


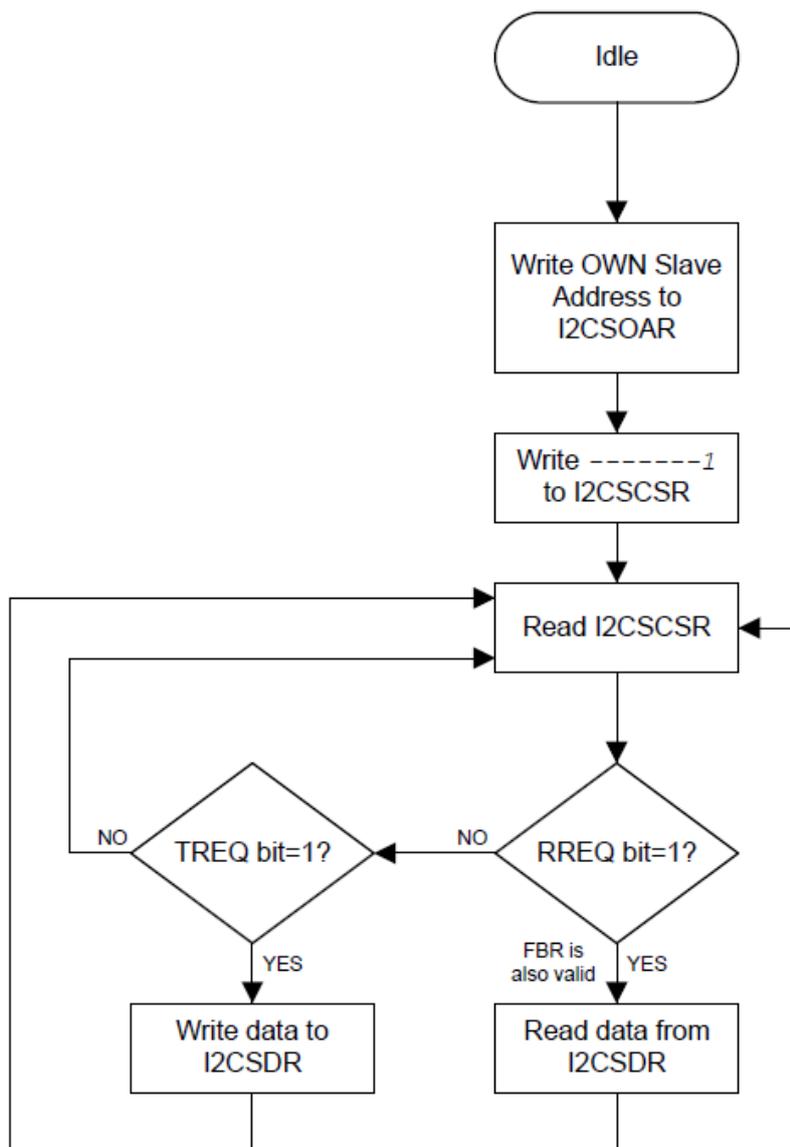
图 15.12 主机重复启动接收后重复启动发送



15.3.5.2 I²C 从机命令序列

下图中介绍了 I²C 从机可用的命令序列

图 15.13 从机命令序列



15.4 初始化和配置

下面的例子展示了如何配置 I²C 模块作为主机发送一个字节。这里假设系统时钟为 20MHz。

1. 在系统控制模块里向 RCGC1 寄存器写入 0x0000.1000 来使能 I²C 模块的时钟(见 281 页)。
2. 在系统控制模块里向 RCGC2(见 293 页)寄存器写入适当的值来启用相应的 GPIO 模块的时钟。欲知要使能哪个 GPIO 模块请参考表 24.5。
3. 在 GPIO 模块里，用 GPIOAFSEL 寄存器使能相关引脚的备用功能。欲知要使能哪个引脚，请参考表 24.4。
4. 使能 I²C 引脚的开漏。见 434 页。
5. 配置 GPIOPCTL 寄存器的 PMCn 位域来使能引脚的 I²C 功能。详情请见 447 页和表 24.5
6. 向 I2CMCR 寄存器写入 0x0000.0010 来初始化 I²C 主机。

7. 向 I2CMTPR 寄存器写入正确的值将 SCL 速率设置为 100kbps。写入 I2CMTPR 寄存器的值代表了在一个 SCL 周期内系统时钟的个数。TPR 的值由以下方程决定：

$$TPR = (\text{System Clock} / (2 * (\text{SCL_LP} + \text{SCL_HP}) * \text{SCL_CLK})) - 1;$$

$$TPR = (20\text{MHz} / (2 * (6+4) * 100000)) - 1;$$

$$TPR = 9$$
 向 I2CMTPR 寄存器写入 0x0000.0009
8. 给主机指定从机地址，向 I2CMSA 寄存器写入 0x0000.0076，下一个操作时发送。这里是设置从机地址为 0x3B。
9. 向 I2CMDR 寄存器写入需要传送的数据设置数据寄存器中准备发送的数据(字节)。
10. 通过向 I2CMCS 寄存器写入 0x0000.0007 的值来启动从主机到从机一个字节的数据发送 (STOP, START, RUN)。
11. 等待，直到传输结束，通过查询 I2CMCS 寄存器的 BUSBSY 位直至它已被清零。

15.5 寄存器映射

表 15.4 列出了 I²C 的寄存器。所有的寄存器都是以 I²C 的基地址，以十六进制的偏移量递增的顺序排列的。I²C 寄存器的基地址为：

- I²C 主机 0:0x4002.0000
- I²C 从机 0:0x4002.0800
- I²C 主机 1:0x4002.1000
- I²C 从机 1:0x4002.1800

注意：在使用下面这些寄存器之前，应该向使能 I2C 模块的时钟。

表 15.4 I²C 寄存器映射

偏移量	名称	类型	复位值	描述	见页面
I²C 主机					
0x000	I2CMSA	R/W	0x0000.0000	I ² C 主从机地址寄存器	805
0x004	I2CMCS	R/W	0x0000.0000	I ² C 主机控制/状态寄存器	806
0x008	I2CMDR	R/W	0x0000.0000	I ² C 主机数据寄存器	810
0x00C	I2CMTPR	R/W	0x0000.0001	I ² C 主机定时器周期寄存器	811
0x010	I2CMIMR	R/W	0x0000.0000	I ² C 主机中断屏蔽寄存器	812
0x014	I2CMRIS	RO	0x0000.0000	I ² C 主机原始中断状态寄存器	813
0x018	I2CMMIS	RO	0x0000.0000	I ² C 主机屏蔽后的中断状态寄存器	814
0x01C	I2CMICR	WO	0x0000.0000	I ² C 主机中断清除寄存器	815
0x020	I2CMCR	R/W	0x0000.0000	I ² C 主机配置寄存器	816

表 15.4 I2C 寄存器映射(续)

偏移量	名称	类型	复位值	描述	见页面
I2C 从机					
0x000	I2CSOAR	R/W	0x0000.0000	I ² C 从机地址寄存器	817
0x004	I2CSCSR	R/W	0x0000.0000	I ² C 从机控制/状态寄存器	818
0x008	I2CSDR	R/W	0x0000.0000	I ² C 从机数据寄存器	820
0x00C	I2CSIMR	R/W	0x0000.0001	I ² C 从机中断屏蔽寄存器	821
0x010	I2CSRIS	R/W	0x0000.0000	I ² C 从机原始中断状态寄存器	822
0x014	I2CSMIS	RO	0x0000.0000	I2C 从机屏蔽后的中断状态寄存器	823
0x018	I2CSICR	RO	0x0000.0000	I ² C 从机中断清除寄存器	824

15.6 寄存器描述(I²C 主机)

本章的剩余部分列出了 I²C 所有的寄存器,并逐一详细描述了它们。它们的排列顺序是以 I²C 的基地址以十六进制的偏移量按递增的顺序排列的。

寄存器 1: I²C 主从机地址寄存器(I2CMSA), 偏移量 0x000

该寄存器包含 8 位: 7 位地址(A6:A0)和 1 位接收/发送位。接收/发送位决定了下一个操作是接收(高)还是发送(低)。

I²C 主从机地址寄存器

I2C 0 基地址: 0x4002.0000

I2C 1 基地址: 0x4002.1000

偏移量: 0x000

类型: R/W, 复位值 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO								
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								SA							R/S
类型	RO	R/W														
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/位域	名称	类型	复位值	描述
31:8	保留	RO	0x0000 00	软件不得依赖保留位的值。为保障软件与将来产品的兼容性,对寄存器进行“读-修改-写”操作时,不得变更保留位的值。
7:1	SA	R/W	0x00	I ² C 从机地址 A6:A0 的值指定了从机的地址
0	R/S	R/W	0	接收/发送 接收/发送位决定了下一个操作时接收(高)还是发送(低) 取值 描述 0 发送 1 接收

寄存器 2: I²C 主机控制/状态寄存器(I2CMCS), 偏移量 0x004

该寄存器在读的时候可以访问 7 位, 在写的时候可以访问 4 位。

在读的时候, 状态寄存器包含 7 位, 这 7 位反映了 I2C 总线控制器的状态。

在写的时候, 控制寄存器包含 4 位: RUN, START, STOP 和 ACK 位。START 位用来产生开始或重复开始信号。

STOP 位决定周期是在数据周期结束时停还是继续在重复开始的情况下运行。要生成一个单一的传输周期, I²C 主从机地址寄存里写入需要的地址, R/S 位清零, 控制寄存器写入下列值: ACK=X(0 或 1)、STOP=1、START=1、RUN=1, 来执行操作或停止。当操作结束(或因错误而中止)时, 中断引脚变得活跃, 并且可以从 I2CMDR 寄存器读取数据。当 I2C 模块运行在主机接收模式的时候, 通常将 ACK 位置位, 这样, 在每个字节后 I2C 总线控制器都会自动传输一个应答信号。当主机不再需要从机传输数据的时候, 该位必须清零。

只读的状态寄存器**I²C 主机控制/状态寄存器**

I2C 0 基地址: 0x4002.0000

I2C 1 基地址: 0x4002.1000

偏移量: 0x004

类型: RO, 复位值: 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO								
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								BUSBSY	IDLE	ARBLST	DATAACK	ADRACK	ERROR	BUSY	
类型	RO	RO	RO	RO	RO	RO	RO	RO								
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/位域	名称	类型	复位值	描述
31:7	保留	RO	0x0000 00	软件不得依赖保留位的值。为保障软件与将来产品的兼容性, 对寄存器进行“读-修改-写”操作时, 不得变更保留位的值。
6	BUSBSY	RO	0	总线忙 取值 描述 0 I ² C 总线空闲 1 I ² C 总线忙碌
5	IDEL	RO	0	I2C 空闲 取值 描述 0 I ² C 总线没有空闲 1 I ² C 总线空闲

4	ARBLST	RO	0	I2C 仲裁丢失
				取值 描述
				0 I2C 总线赢得仲裁
				1 I2C 总线仲裁丢失
3	DATAACK	RO	0	应答数据
				取值 描述
				0 发送数据时应答
				1 发送数据时不应答
2	ADRACK	RO	0	应答地址
				取值 描述
				0 传输地址时应答
				1 传输地址时不应答
1	ERROR	RO	0	错误
				取值 描述
				0 在最后的操作中没有检测到错误
				1 在最后的操作中已经发生了错误
0	BUSY	RO	0	I2C 忙碌
				取值 描述
				0 控制器空闲
				1 控制器忙碌
				当 BUSY 置位时，其他的状态位是无效的

只写的控制寄存器

I2C 主机控制/状态寄存器

I2C 0 基地址: 0x4002.0000

I2C 1 基地址: 0x4002.1000

偏移量: 0x004

类型: WO, 复位值: 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	WO	WO	WO	WO												
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留												ACK	STOP	START	RUN
类型	WO	WO	WO	WO												
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/位域	名称	类型	复位值	描述
31:4	保留	WO	0x0000 000	软件不得依赖保留位的值。为保障软件与将来产品的兼容性,对寄存器进行“读-修改-写”操作时,不得变更保留位的值。
3	ACK	WO	0	数据应答允许 取值 描述 0 主机在收到数据的时候不自动应答 1 主机在收到数据的时候自动应答, 见表 15.5
2	STOP	WO	0	生成停止信号 取值 描述 0 控制器不生成停止信号 1 控制器生成停止信号, 见表 15.5
1	START	WO	0	生成开始信号 取值 描述 0 控制器不生成开始信号 1 控制器生成开始信号, 见表 15.5
0	RUN	WO	0	I2C 主机允许 取值 描述 0 主机禁止 1 允许主机发送或接收数据, 见表 15.5

表 15.5 写入 I2CMCS 的[3:0]位解码

当前状态	I2CMSA[0]	I2CMCS[3:0]				描述
	R/S	ACK	STOP	START	RUN	
空闲	0	X ^a	0	1	1	START 条件之后跟随 SEND(主机进入主机发送状态)
	0	X	1	1	1	START 条件之后跟随 SEND 和 STOP 条件(主机保持在空闲状态)
	1	0	0	1	1	START 条件之后跟随带有非应答的接收操作(主机进入主机接收状态)。
	1	0	1	1	1	START 条件之后跟随 RECEIVE 和 STOP 条件(主机保持在空闲状态)。
	1	1	0	1	1	START 条件之后跟随 RECEIVE(主机进入主机接收状态)。
	1	1	1	1	1	非法
其余未列出的情况都不执行任何操作						不执行任何操作

表 15.5 写入 I2CMCS 的[3:0]位解码(续)

当前状态	I2CMSA[0]		I2CMCS[3:0]			描述
	R/S	ACK	STOP	START	RUN	
主机发送	X	X	0	0	1	发送操作(主机仍然在发送状态)
	X	X	1	0	0	停止条件(主机进入空闲状态)
	X	X	1	0	1	发送之后跟着是停止条件(主机进入空闲状态)
	0	X	0	1	1	重复 START 条件之后跟随 SEND(主机仍然在发送状态)
	0	X	1	1	1	重复的 START 条件之后跟随 SEND 和 STOP 条件(主机进入空闲状态)。
	1	0	0	1	1	重复 START 条件之后跟随带有非应答的接收操作(主机进入主机接收状态)。
	1	0	1	1	1	重复 START 条件之后跟随 SEND 和 STOP 条件(主机进入空闲状态)。
	1	1	0	1	1	重复 START 条件之后跟随 RECEIVE(主机进入主机接收状态)。
	1	1	1	1	1	非法
	其余未列出的情况都不执行任何操作					不执行任何操作
主机接收	X	0	0	0	1	带有非应答的接收操作(主机保持在接收状态)。
	X	X	1	0	0	STOP 条件 (主机进入空闲状态)。 ^b
	X	0	1	0	1	RECEIVE 接收之后跟随 STOP 条件(主机进入空闲状态)。
	X	1	0	0	1	接收操作(主机保持在主机接收状态)。
	X	1	1	0	1	非法
	1	0	0	1	1	重复的 START 条件之后跟随带有非应答的接收操作(主机保持在主机接收状态)。
	1	0	1	1	1	重复的 START 条件之后跟随 RECEIVE 和 STOP 条件(主机进入空闲状态)。
	1	1	0	1	1	重复的 START 条件之后跟随 RECEIVE(主机保持在主机接收状态)
	0	X	0	1	1	重复的 START 条件之后跟随 SEND(主机进入主机发送状态)。
	0	X	1	1	1	重复的 START 条件之后跟随 SEND 和 STOP 条件(主机进入空闲状态)。
其余未列出的情况都不执行任何操作					不执行任何操作	

a、表中 X 的表示改为可以为 0 也可以为 1

b、在主机接收模式，停止条件在主机执行数据非应答或从机执行地址非应答后产生。

寄存器 3: I²C 主机数据寄存器(I2CMDR), 偏移量 0x008

重要: 读取该寄存器的时候要小心, 因为读的时候可能会改变它的状态。

在主机发送数据时, 该寄存器包含要发送的数据; 在主机接收数据时, 该寄存器包含接收到的数据。

I2C 主机数据寄存器

I2C 0 基地址: 0x4002.0000

I2C 1 基地址: 0x4002.1000

偏移量: 0x008

类型: R/W, 复位值: 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO								
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								DATA							
类型	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W							
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/位域	名称	类型	复位值	描述
31:8	保留	RO	0x0000 00	软件不得依赖保留位的值。为保障软件与将来产品的兼容性, 对寄存器进行“读-修改-写”操作时, 不得变更保留位的值。
7:0	DATA	R/W	0x00	要发送的数据 在发送期间要发送的数据

寄存器 4: I²C 主机定时周期寄存器(I2CMTPR), 偏移量 0x00C

该寄存器指定了 SCL 的时钟周期

警告: 在访问该寄存器的时候不要设置第 7 位, 否则将会发生不可预知的行为。

I²C 主机定时周期寄存器

I2C 0 基地址: 0x4002.0000

I2C 1 基地址: 0x4002.1000

偏移量: 0x00C

类型: R/W, 复位值: 0x0000.0001

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO								
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								TPR							
类型	RO	R/W														
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/位域	名称	类型	复位值	描述
31:8	保留	RO	0x0000 00	软件不得依赖保留位的值。为保障软件与将来产品的兼容性, 对寄存器进行“读-修改-写”操作时, 不得变更保留位的值。
7:0	TPR	R/W	0x00	<p>SCL 时钟周期</p> <p>该区域指定了 SCL 的时钟周期</p> $SCL_PRD = 2 \times (1 + TPR) \times (SCL_LP + SCL_HP) \times CLK_PRD$ <p>在此</p> <p>SCL_PRD 是 SCL 的线性周期(I²C 时钟)</p> <p>TPR 是定时器周期寄存器的值(范围由 1 到 127)</p> <p>SCL_LP 是 SCL 低电平周期(固定为 6)</p> <p>SCL_HP 是 SCL 高电平周期(固定为 4)</p>

寄存器 5: I²C 主机中断屏蔽寄存器(I2CMIMR), 偏移量 0x010

该寄存器控制是否将原始中断提交到控制器中断。

I2C 主机中断屏蔽寄存器

I2C 0 基地址: 0x4002.0000

I2C 1 基地址: 0x4002.1000

偏移量: 0x010

类型: R/W, 复位值: 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															
类型	RO	R/W														
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/位域	名称	类型	复位值	描述
31:1	保留	RO	0x0000 000	软件不得依赖保留位的值。为保障软件与将来产品的兼容性,对寄存器进行“读-修改-写”操作时,不得变更保留位的值。
0	IM	R/W	0	中断屏蔽 取值 描述 0 RIS 中断已经发生,但是没有发送到中断控制器 1 当I2CMRIS 寄存器里的RIS位置位时主机的中断被送到中断控制器

寄存器 6: I²C 原始中断状态寄存器(I2CMRIS), 偏移量 0x014

该寄存器指示了是否有中断发生。

I²C 原始中断状态寄存器

I2C 0 基地址: 0x4002.0000

I2C 1 基地址: 0x4002.1000

偏移量: 0x014

类型: RO, 复位值: 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															
类型	RO															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/位域	名称	类型	复位值	描述
31:1	保留	RO	0x0000 000	软件不得依赖保留位的值。为保障软件与将来产品的兼容性,对寄存器进行“读-修改-写”操作时,不得变更保留位的值。
0	RIS	RO	0	原始中断状态 取值 描述 0 没有发生中断 1 主机已经有中断发生 通过向 I2CMICR 寄存器的 IC 位写 1 来清除该位。

寄存器 7: I²C 主机屏蔽后的中断状态寄存器(I2CMMIS), 偏移量 0x018

该寄存器指示了中断是否被提交给了中断控制器

I2C 主机屏蔽后的中断状态寄存器

I2C 0 基地址: 0x4002.0000

I2C 1 基地址: 0x4002.1000

偏移量: 0x018

类型: RO, 复位值: 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															
类型	RO															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/位域	名称	类型	复位值	描述
31:1	保留	RO	0x0000 000	软件不得依赖保留位的值。为保障软件与将来产品的兼容性,对寄存器进行“读-修改-写”操作时,不得变更保留位的值。
0	MIS	RO	0	原始中断状态 取值 描述 0 没有发生中断或是被屏蔽 1 未屏蔽的主机中断已经发生有待处理 通过向 I2CMICR 寄存器的 IC 位写 1 来清除该位。

寄存器 8: I²C 主机中断清除寄存器(I2CMICR), 偏移量 0x01C

该寄存器清除原始的中断。

I²C 主机中断清除寄存器

I2C 0 基地址: 0x4002.0000

I2C 1 基地址: 0x4002.1000

偏移量: 0x01C

类型: WO, 复位值: 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															IC
类型	RO	WO														
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/位域	名称	类型	复位值	描述
31:1	保留	RO	0x0000 000	软件不得依赖保留位的值。为保障软件与将来产品的兼容性,对寄存器进行“读-修改-写”操作时,不得变更保留位的值。
0	RIS	WO	0	中断清除 向该位写入 1 清除 I2CMRIS 的 RIS 位和 I2CMMIS 的 MIS 位 读取该寄存器的值没有任何意义

寄存器 9: I²C 主机配置寄存器(I2CMCR), 偏移量 0x020

该寄存器对 I²C 的模式(主机或从机)进行配置, 并为测试模式回送设置接口。

I²C 主机配置寄存器

I2C 0 基地址: 0x4002.0000

I2C 1 基地址: 0x4002.1000

偏移量: 0x020

类型: R/W, 复位值: 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO												
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留											SFE	MFE	保留		LPBK	
类型	RO	R/W	R/W	RO	RO	RO	R/W										
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/位域	名称	类型	复位值	描述
31:6	保留	RO	0x0000 00	软件不得依赖保留位的值。为保障软件与将来产品的兼容性, 对寄存器进行“读-修改-写”操作时, 不得变更保留位的值。
5	SFE	R/W	0	I2C 从机功能使能 取值 描述 0 从机模式禁止使用。 1 使用从机模式
4	MFE	R/W	0	I2C 主机功能使能 取值 描述 0 主机模式禁止使用 1 使用主机模式
3:1	保留	RO	0x0	软件不得依赖保留位的值。为保障软件与将来产品的兼容性, 对寄存器进行“读-修改-写”操作时, 不得变更保留位的值。
0	LPBK	R/W	0	I2C 回送功能 取值 描述 0 正常使用 1 配置为回送的测试模式

15.7 寄存器描述(I²C 从机)

本章的剩余部分列出了 I²C 所有的寄存器, 并逐一详细描述了它们。它们的排列顺序是以 I²C 的基地址以十六进制偏移量按递增的顺序排列的。同见“寄存器描述(I²C 主机)”一节

寄存器 10: I²C 从机自身地址寄存器(I2COAR), 偏移量 0x000

该寄存器包括 7 个地址位, 以识别 I²C 总线上的 Stellaris® I2C 设备。

I2C 从机自身地址寄存器

I2C Slave 0 base: 0x4002.0800

I2C Slave 1 base: 0x4002.1800

偏移量: 0x000

类型: R/W, 复位值: 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO								
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								OAR							
类型	RO	R/W														
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/位域	名称	类型	复位值	描述
31:8	保留	RO	0x0000 00	软件不得依赖保留位的值。为保障软件与将来产品的兼容性, 对寄存器进行“读-修改-写”操作时, 不得变更保留位的值。
7:0	TPR	RO	0x00	I2C 自身地址 改寄存器指定了从机自身的地址(A6:A0 位)

寄存器 11: I²C 从机控制/状态寄存器(I2CSCSR), 偏移量 0x004

该寄存器在写的时候访问一个控制位, 在读的时候访问三个状态位

只读的状态寄存器包含三个位: FBR 位, RREQ 位和 TREQ 位。接收到的第一字节(FBR)位仅在 Stellaris®器件检测到其自身从机地址并接收到第一个来自 I2C 主机的数据字节之后才会置位。接收请求(RREQ)位表示 Stellaris® I2C 设备已接收到来自 I2C 主机的数据字节。从 I2C 从机数据(I2CSDR)寄存器中读取一个数据字节来清零 RREQ 位。发送请求 (TREQ)位表示 Stellaris® I2C 设备作为从机发送器被寻址。写一个数据字节到 I2C 从机数据(I2CSDR)寄存器来清零 TREQ 位。

只写控制寄存器包括一个位: DA 位。DA 位使能和禁能 Stellaris® I2C 从机操。

只读状态寄存器**I2C 从机控制/状态寄存器**

I2C Slave 0 base: 0x4002.0800

I2C Slave 1 base: 0x4002.1800

偏移量: 0x004

类型: RO, 复位值: 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO													
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留													FBR	TREQ	RREQ
类型	RO	RO	RO													
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/位域	名称	类型	复位值	描述
31:3	保留	RO	0x0000 000	软件不得依赖保留位的值。为保障软件与将来产品的兼容性, 对寄存器进行“读-修改-写”操作时, 不得变更保留位的值。
2	FBR	RO	0	接收到了第一个字节 取值 描述 0 还没有接收到第一个字节 1 接收到了紧跟随自身地址的第一个字节 仅当 RREQ 位置位时该位才有效。并且从 I2CSDR 寄存器读完数据后该位自动清零。 注意: 该位不能用于从机发送操作。

位/位域	名称	类型	复位值	描述
1	TREQ	RO	0	发送请求 取值 描述 0 没有发送请求 1 该设备作为从机发送器被寻址，并使用扩展时钟将主机延时，直至数据已被写入 I2CSDR 寄存器。
0	RREA	RO	0	接收请求 取值 描述 0 没有数据要接收 1 该设备含有来自 I2C 主机的未处理的接收数据，并使用扩展时钟来延时主机直至数据已从 I2CSDR 寄存器中读取

只写控制寄存器

I2C 从机控制/状态寄存器

I2C Slave 0 base: 0x4002.0800

I2C Slave 1 base: 0x4002.1800

偏移量: 0x004

类型: WO, 复位值: 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO															
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															
类型	RO	WO														
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/位域	名称	类型	复位值	描述
31:1	保留	RO	0x0000 000	软件不得依赖保留位的值。为保障软件与将来产品的兼容性，对寄存器进行“读-修改-写”操作时，不得变更保留位的值。
0	DA	WO	0	设备有效 取值 描述 0 禁止 I2C 从机的操作 1 使能 I2C 从机的操作

寄存器 12: I²C 从机数据寄存器(I2CSDR), 偏移量 0x008

重要: 读取该寄存器的时候要小心, 执行读操作可能会改变位的状态

该寄存器含有在从机发送状态中准备发送的数据, 以及在从机接收状中接收到的数据。

I²C 从机数据寄存器

I2C Slave 0 base: 0x4002.0800

I2C Slave 1 base: 0x4002.1800

偏移量: 0x008

类型: R/W, 复位值: 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO								
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留								DATA							
类型	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W							
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/位域	名称	类型	复位值	描述
31:8	保留	RO	0x0000 00	软件不得依赖保留位的值。为保障软件与将来产品的兼容性, 对寄存器进行“读-修改-写”操作时, 不得变更保留位的值。
7:0	DATA	R/W	0x00	传送的数据 该字段包含了在从机接收或发送操作中传输的数据。

寄存器 13: I²C 中断屏蔽寄存器(I2CSIMR), 偏移量 0x00C

该寄存器控制是否将原始中断提交到控制器中断。

I²C 中断屏蔽寄存器

I2C Slave 0 base: 0x4002.0800

I2C Slave 1 base: 0x4002.1800

偏移量: 0x00C

类型: R/W, 复位值: 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO													
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留													STOPI	START	DATA
类型	RO	RO	R/W													
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/位域	名称	类型	复位值	描述
31:3	保留	RO	0x0000 000	软件不得依赖保留位的值。为保障软件与将来产品的兼容性,对寄存器进行“读-修改-写”操作时,不得变更保留位的值。
2	STOPI	RO	0	停止条件中断屏蔽 取值 描述 0 STOPRIS 中断被抑制,不发送到中断控制器 1 当 I2CSRIS 寄存器中的 STOPRIS 位置位时,停止条件中断被送到中断控制器
1	START	RO	0	开始条件中断屏蔽 取值 描述 0 TARTRIS 中断被抑制,不发送到中断控制器 1 当 I2CSRIS 寄存器中的 STARTRIS 位置位时,开始条件中断被送到中断控制器
0	DATA	R/W	0	数据中断屏蔽 取值 描述 0 DATARIS 中断被抑制,不发送到中断控制器 1 当 I2CSRIS 寄存器中的 DATARIS 位置位时,数据接收或需求数据中断被送到中断控制器

寄存器 14: I²C 从机原始中断状态寄存器(I2CSRIS), 偏移量 0x010

该寄存器表示是否有中断正等待处理

I²C 从机原始中断状态寄存器

I2C Slave 0 base: 0x4002.0800

I2C Slave 1 base: 0x4002.1800

偏移量: 0x010

类型: RO, 复位值: 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO													
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留													STOPRIS	STARTRIS	DATARIS
类型	RO	RO	RO													
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/位域	名称	类型	复位值	描述
31:3	保留	RO	0x0000 000	软件不得依赖保留位的值。为保障软件与将来产品的兼容性,对寄存器进行“读-修改-写”操作时,不得变更保留位的值。
2	STOPRIS	RO	0	停止条件原始中断状态 取值 描述 0 STOPRIS 中断被抑制,不发送到中断控制器 1 当 I2CSRIS 寄存器中的 STOPRIS 位置位时,停止条件中断被送到中断控制器 在 I2CSICR 寄存器的 STOPIC 位写入 1 来清除该位
1	STARTRIS	RO	0	开始条件原始中断状态 取值 描述 0 TARTRIS 中断被抑制,不发送到中断控制器 1 当 I2CSRIS 寄存器中的 STARTRIS 位置位时,开始条件中断被送到中断控制器 在 I2CSICR 寄存器的 STARTIC 位写入 1 来清除该位
0	DATARIS	RO	0	数据原始中断状态 取值 描述 0 没有中断 1 当 I2CSRIS 寄存器中的 DATARIS 位置位时,数据接收或需求数据中断被送到中断控制器 在 I2CSICR 寄存器的 DATAIC 位写入 1 来清除该位

寄存器 15: I²C 从机屏蔽后的中断状态寄存器(I2CSMIS), 偏移量 0x014

该寄存器表示是否发出中断信号。

I²C 从机屏蔽后的中断状态寄存器

I2C Slave 0 base: 0x4002.0800

I2C Slave 1 base: 0x4002.1800

偏移量: 0x014

类型: RO, 复位值: 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO													
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留													STOPMIS	STARTMIS	DATAMIS
类型	RO	R/W	R/W	RO												
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/位域	名称	类型	复位值	描述
31:3	保留	RO	0x0000 000	软件不得依赖保留位的值。为保障软件与将来产品的兼容性,对寄存器进行“读-修改-写”操作时,不得变更保留位的值。
2	STOPMIS	R/W	0	停止条件屏蔽后的中断状态 取值 描述 0 没有中断发生或是被屏蔽 1 有一个没有屏蔽的停止条件中断等待处理 在 I2CSICR 寄存器的 STOPIC 位写入 1 来清除该位
1	STARTMIS	R/W	0	开始条件屏蔽后的中断状态 取值 描述 0 没有中断发生或是被屏蔽 1 有一个没有屏蔽的开始条件中断等待处理 在 I2CSICR 寄存器的 STARTIC 位写入 1 来清除该位
0	DATAMIS	RO	0	数据屏蔽后的中断状态 取值 描述 0 没有中断发生或是被屏蔽 1 有一个未屏蔽的数据接收或需求中断等待处理 在 I2CSICR 寄存器的 DATAIC 位写入 1 来清除该位

寄存器 16: I²C 从机中断清除寄存器(I2CSICR), 偏移量 0x018

该寄存器清除原始的中断。读取该寄存器返回没有意义的数。

I²C 从机中断清除寄存器

I2C Slave 0 base: 0x4002.0800

I2C Slave 1 base: 0x4002.1800

偏移量: 0x018

类型: WO, 复位值: 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO														
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留													STOPIC	STARTIC	DATAIC	
类型	RO	WO	WO	WO													
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/位域	名称	类型	复位值	描述
31:3	保留	RO	0x0000 000	软件不得依赖保留位的值。为保障软件与将来产品的兼容性,对寄存器进行“读-修改-写”操作时,不得变更保留位的值。
2	STOPIC	WO	0	<p>停止条件中断清除</p> <p>向该位写入 1 清除 I2CSRIS 寄存器的 STOPRIS 位和 I2CSMIS 寄存器的 STOPMIS 位。</p> <p>读取该位返回没有意义的数</p>
1	STARTIC	WO	0	<p>开始条件中断清除</p> <p>向该位写入 1 清除 I2CSRIS 寄存器的 STOPRIS 位和 I2CSMIS 寄存器的 STOPMIS 位。</p> <p>读取该位返回没有意义的数</p>
0	DATAIC	WO	0	<p>数据中断清除</p> <p>向该位写入 1 清除 I2CSRIS 寄存器的 STOPRIS 位和 I2CSMIS 寄存器的 STOPMIS 位。</p> <p>读取该位返回没有意义的数</p>

北京锐鑫同创公司相关信息

技术支持

如果您对文档有所疑问，您可以在办公时间（星期一至星期五上午 8:30~11:50；下午 1:30~5:30）拨打技术支持电话或 E-mail 联系。

北京锐鑫同创是 TI 第三方合作伙伴，专注于 TI Stellaris M3 产品的市场推广、方案设计和技术服务，同时提供开发板、仿真器、编程器等开发工具，公司以“把握市场脉搏，专注技术创新，提供诚信服务，实现共赢发展！”为核心价值理念，为客户提供实时、高效的技术和服务。

电话：010-82418301

传真：010-82418302

Email: support@realsense.com.cn

网站: www.realsense.com.cn

技术论坛: www.hellom3.cn