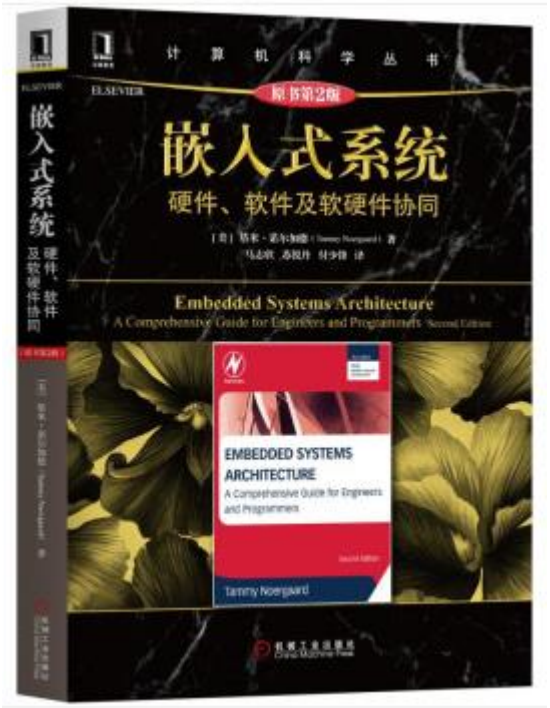
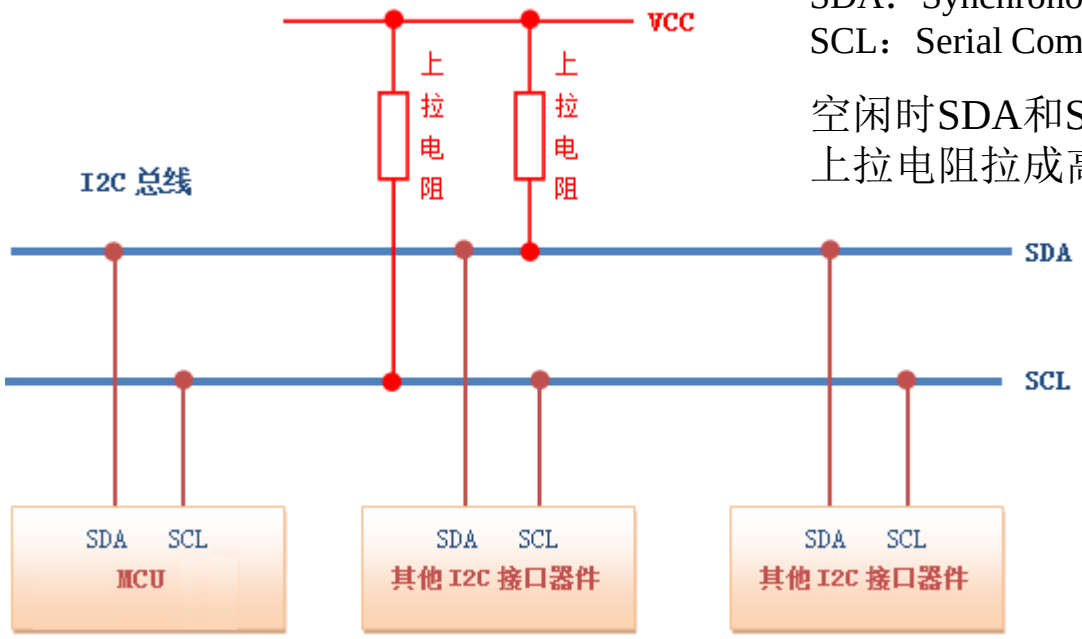
第八章 总线IIC



isszym 2019.12.11

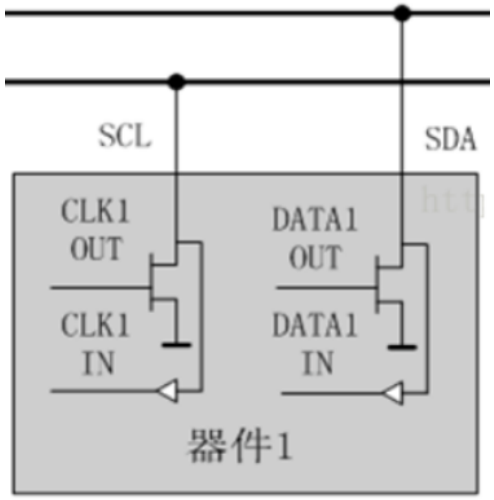
概述

- IIC(Inter-Integrated Circuit)总线是一种两线式的串行总线，可用于连接多个设备，并采用主从模式的半双工通信方式，任意时刻只能有一个主设备发送数据，其他设备只可以接收数据。
- IIC总线传输距离短，速度小，标准速度(最高速度100kHZ)，快速（最高400kHZ），常用于连接低速外设。



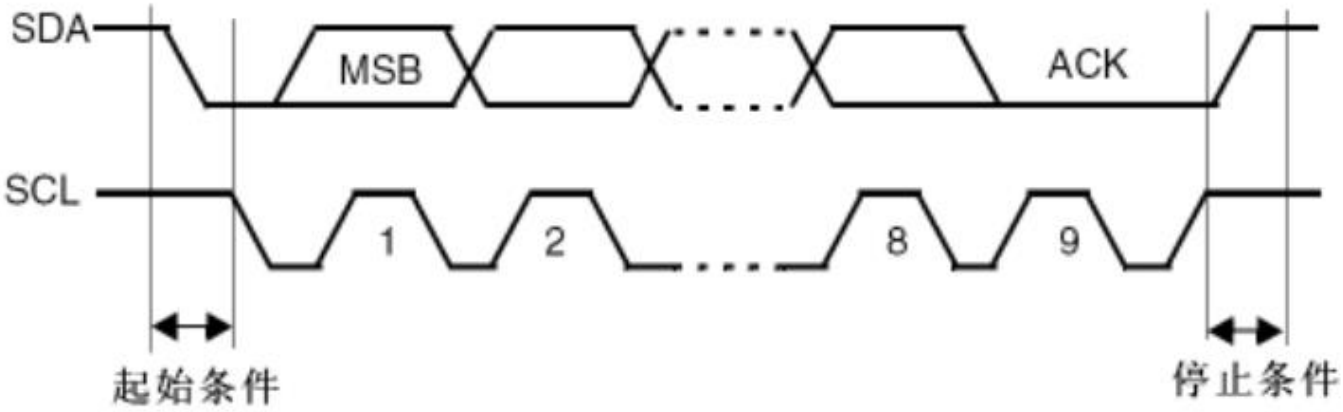
SDA：Synchronous Data Adapter
SCL：Serial Communication Loop

空闲时SDA和SCL被
上拉电阻拉成高电平

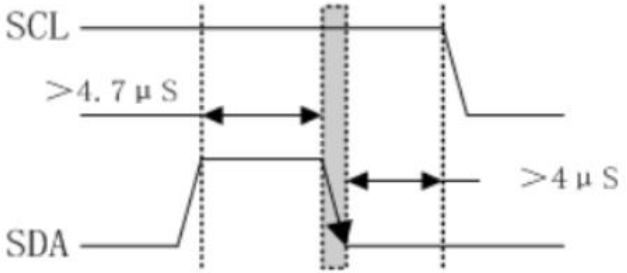


IIC 总线时序图

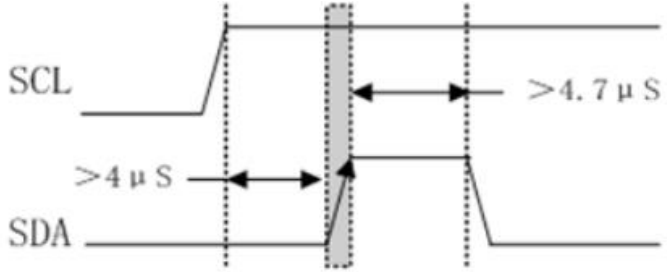
- 帧格式：1位起始位，8位数据，1位的ACK，1位停止位



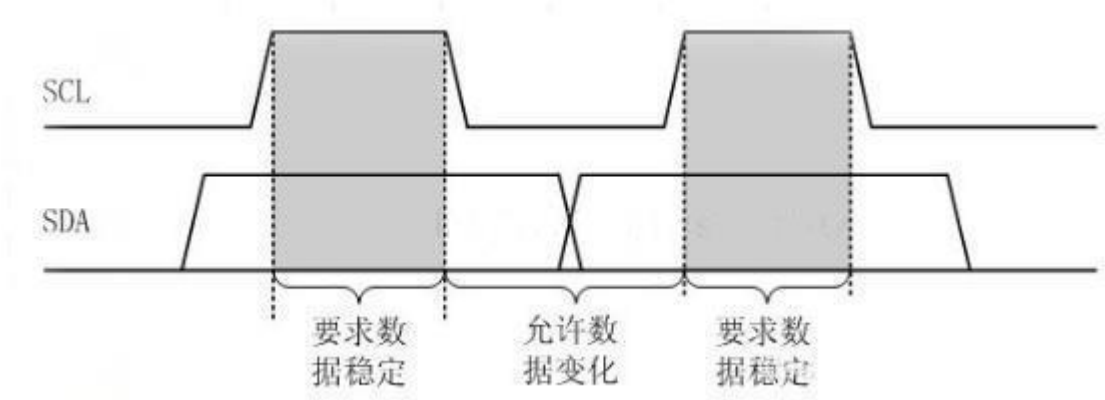
- **开始信号**：SCL保持高电平，SDA由高电平变为低电平后，至少保持4.7us后，SCL变为低电平。



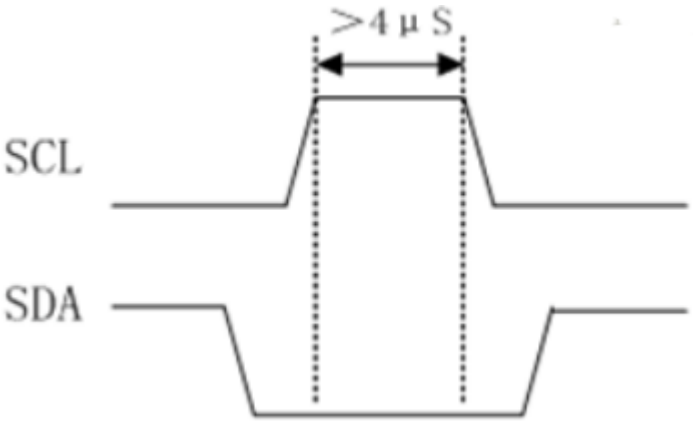
- 停止信号：SCL保持高电平，SDA由低电平变为高电平，并保持至少4.7us后才可以变为低电平（下一次传送的开始）。



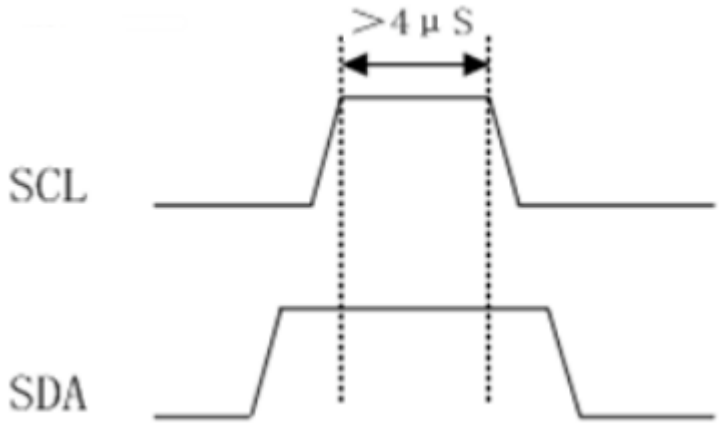
- **传输数据：**当SCL为高电平时，SDA必须保持稳定状态，不允许有电平跳变，只有在SCL为低电平时， SDA的状态才允许变化。



- **应答：**应答(ACK)出现在每一次主机完成8个数据位传输后紧跟着的时钟周期，低电平表示应答，高电平表示非应答(NAK)。

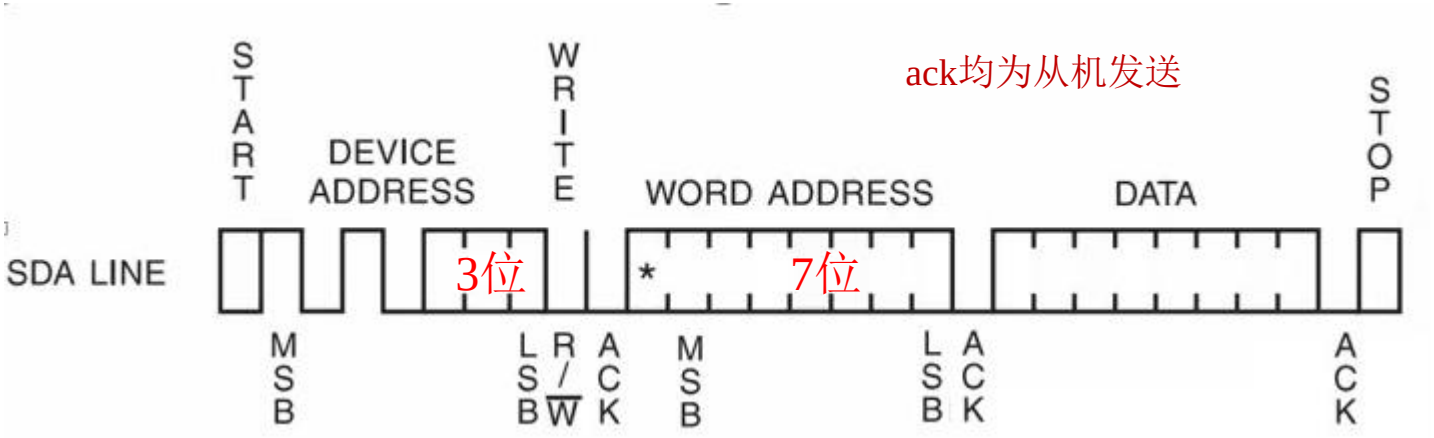


ACK



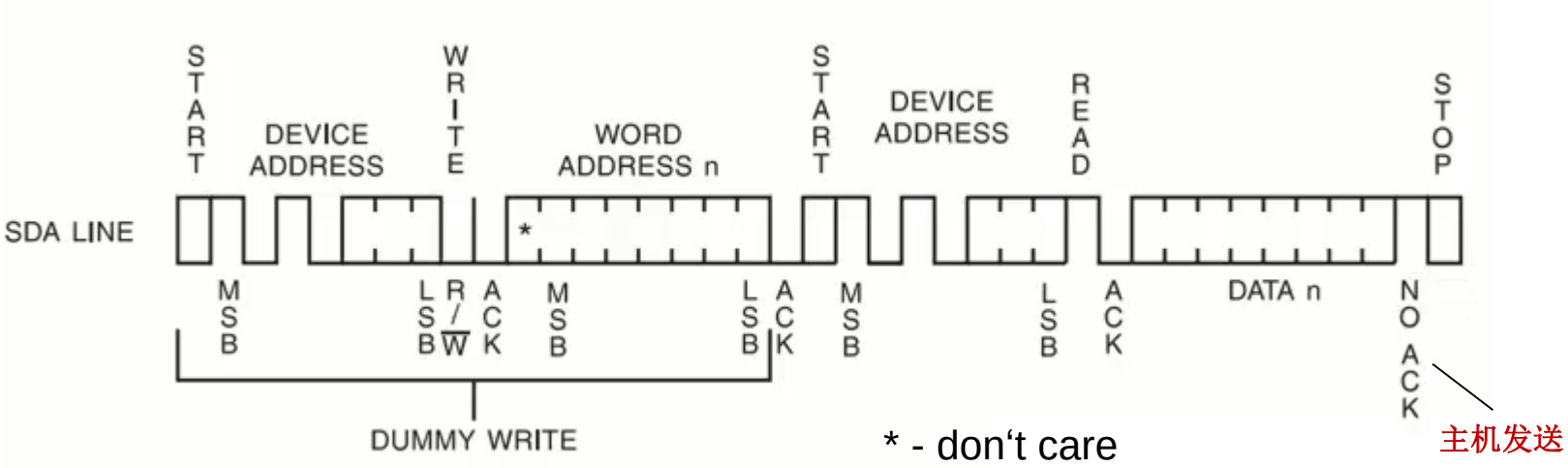
NAK

- **主机向从机写数据(24C02-SRAM)：**



1. 主机首先产生START信号，然后发送一个从机地址，这个地址共有7位，紧接着的第8位是数据方向位(R/W)，0表示主机发送数据(写)，1表示主机接收数据(读)。
2. 主机发送地址时，总线上的每个从机都将这7位地址码与自己的地址进行比较，若相同，则认为自己正在被主机寻址，根据R/T位将自己确定为发送器和接收器。
3. 这时候主机等待从机的应答信号(ACK)。
4. 当主机收到应答信号时，发送要访问从机的地址，例如，存储器地址， 继续等待从机的应答信号。
5. 当主机收到应答信号时，发送N个字节的数据，每发送一个字节都要等待从机的应答信号。
6. 主机产生停止信号，结束传送过程。

- **主机要从从机读数据(24C02-SRAM)**



1. 主机首先产生START信号，紧跟着发送一个从机地址，第8位为0，表明这帧是向从机写数据。
2. 主机等待从机的应答信号(ACK)，收到应答信号后发送要访问的地址，继续等待从机的应答信号。
3. 当主机收到应答信号后，主机要改变通信模式(主机将由发送变为接收，从机将由接收变为发送)所以主机重新发送一个开始start信号，然后紧跟着发送一个从机地址，注意此时该地址的第8位为1，表明将主机设置成接收模式开始读取数据。
4. 这时候主机等待从机的应答信号，当主机收到应答信号时，就可以接收1个字节的数据，当接收完成后，主机发送非应答信号，表示不再接收数据。
5. 主机进而产生停止信号，结束传送过程。

X = don't care; 1 = HIGH; 0 = LOW.

Slave address	R/W bit	Description
0000 000	0	general call address[1]
0000 000	1	START byte[2]
0000 001	X	CBUS address[3]
0000 010	X	reserved for different bus format[4]
0000 011	X	reserved for future purposes
0000 1XX	X	Hs-mode master code
1111 1XX	X	reserved for future purposes
1111 0XX	X	10-bit slave addressing

STM32F103C8T6的I2C主要特点

- 并行总线/I2C总线协议转换器
- 多主机功能：该模块既可做主设备也可做从设备
- I2C主设备功能
  - 产生时钟
  - 产生起始和停止信号
- I2C从设备功能
  - 可编程的I2C地址检测
  - 可响应2个从地址的双地址能力
  - 停止位检测
- 产生和检测7位/10位地址和广播呼叫
- 支持不同的通讯速度
  - 标准速度(高达100 kHz)
  - 快速(高达400 kHz)
- 状态标志：
  - 发送器/接收器模式标志
  - 字节发送结束标志
  - I2C总线忙标志

- 错误标志
  - 主模式时的仲裁丢失
  - 地址/数据传输后的应答(ACK)错误
  - 检测到错位的起始或停止条件
  - 禁止拉长时钟功能时的上溢或下溢
- 2个中断向量
  - 1个中断用于地址/数据通讯成功
  - 1个中断用于错误
- 可选的拉长时钟功能
- 具单字节缓冲器的DMA
- 可配置的PEC(信息包错误检测)的产生或校验：
  - 发送模式中PEC值可以作为最后一个字节传输
  - 用于最后一个接收字节的PEC错误校验
- 兼容SMBus 2.0
  - 25 ms时钟低超时延时
  - 10 ms主设备累积时钟低扩展时间
  - 25 ms从设备累积时钟低扩展时间
  - 带ACK控制的硬件PEC产生/校验
  - 支持地址分辨协议(ARP)
- 兼容SMBus

I2C功能描述

I2C模块接收和发送数据，并将数据从串行转换成并行，或并行转换成串行。可以开启或禁止中断。接口通过数据引脚(SDA)和时钟引脚(SCL)连接到I2C总线。允许连接到标准(高达100kHz)或快速(高达400kHz)的I2C总线。

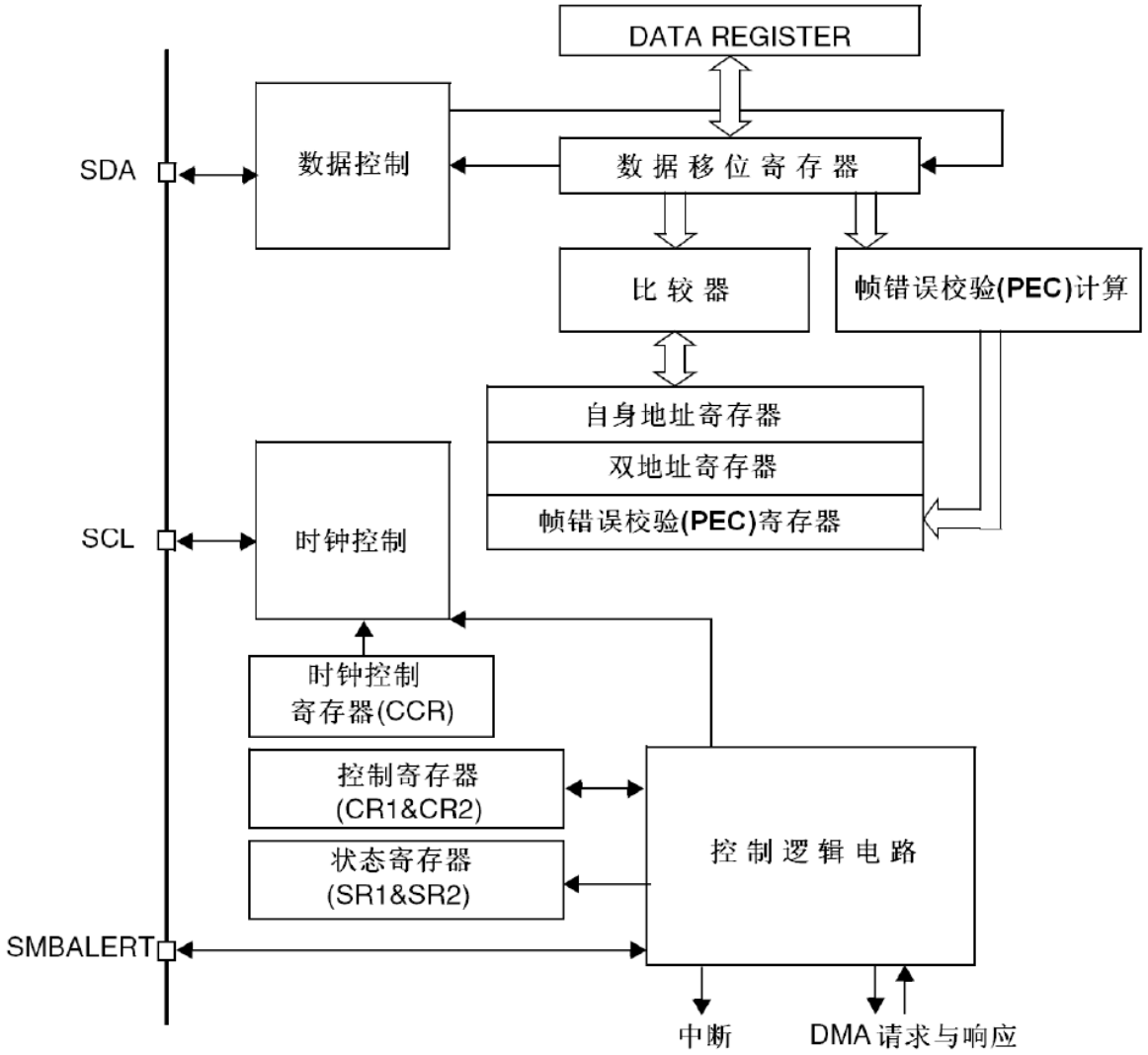
接口可以下述4种模式中的一种运行：

- 从发送器模式
- 从接收器模式
- 主发送器模式
- 主接收器模式

该模块默认地工作于从模式。接口在生成起始条件后自动地从从模式切换到主模式；当仲裁丢失或产生停止信号时，则从主模式切换到从模式。允许多主机功能。

软件可以开启或禁止应答(ACK)，并可以设置I²C接口的地址(7位、10位地址或广播呼叫地址)。

I²C接口的功能框图示于下图。



I2C主模式

在主模式时，I2C接口启动数据传输并产生时钟信号。串行数据传输总是以起始条件开始并以停止条件结束。当通过START位在总线上产生了起始条件，设备就进入了主模式。

以下是**主模式所要求的操作顺序：**

- 在I2C_CR2寄存器中设定该模块的输入时钟以产生正确的时序
- 配置时钟控制寄存器
- 配置上升时间寄存器
- 编程I2C_CR1寄存器启动外设
- 置I2C_CR1寄存器中的START位为1，产生起始条件

I2C模块的输入时钟频率必须至少是:

- 标准模式下为：2MHz
- 快速模式下为：4MHz

图245　主发送器传送序列图

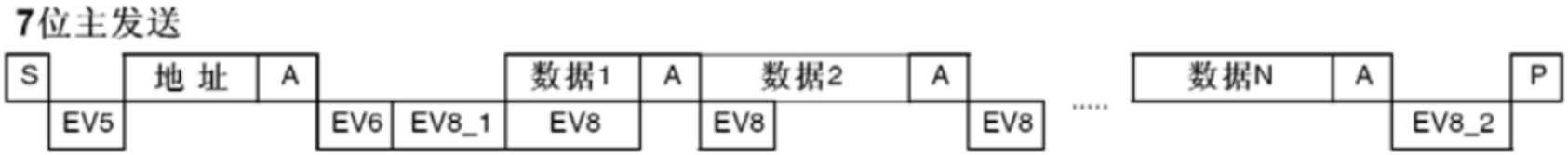
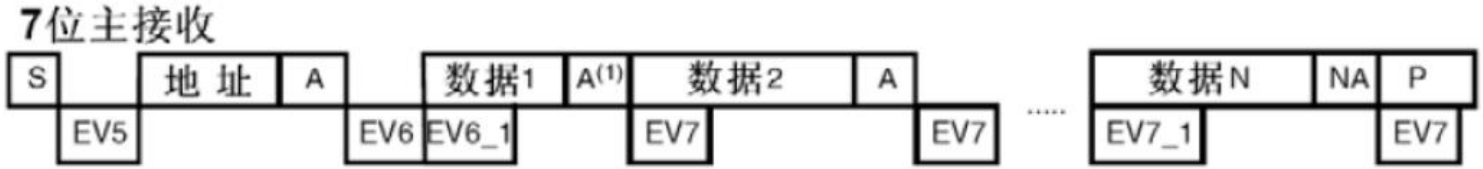


图246　主接收器传送序列图



说明：S=Start(起始条件)，Sr=重复的起始条件，P=Stop(停止条件)，A=响应，NA=非响应，EVx=事件(ITEVFEN=1时产生中断)。

- EV5：SB=1，读SR1然后将地址写入DR寄存器将清除该事件。
- EV6：ADDR=1，读SR1然后读SR2将清除该事件。
- EV6_1：没有对应的事件标志，只适于接收1个字节的情况。恰好在EV6之后(即清除了ADDR之后)，要清除响应和停止条件的产生位。
- EV7：RxNE=1，读DR寄存器清除该事件。
- EV7_1：RxNE=1，读DR寄存器清除该事件。设置ACK=0和STOP请求。
- EV8：TxE=1，移位寄存器非空，数据寄存器空，写入DR寄存器将清除该事件。
- EV8_1：TxE=1，移位寄存器空，数据寄存器空，写DR寄存器。
- EV8_2：TxE=1，BTF=1，请求设置停止位。TxE和BTF位由硬件在产生停止条件时清除。
- EV9：ADDR10=1，读SR1然后写入DR寄存器将清除该事件。

注：　1 – EV5、EV6、EV8_1和EV8_2事件拉长SCL低的时间，直到对应的软件序列结束。
　　2 – EV8的软件序列必须在当前字节传输结束之前完成。

起始条件

- 当BUSY=0时，设置START=1，I2C接口将产生一个开始条件并切换至主模式(M/SL位置位)。
- 注： 在主模式下，设置START位将在当前字节传输完后由硬件产生一个重开始条件。
- 一旦发出开始条件:
  - ✓ SB位被硬件置位，如果设置了ITEVFEN位，则会产生一个中断。
- 然后主设备等待读SR1寄存器，紧跟着将从地址写入DR寄存器(见图245和图246的EV5)。

从地址的发送

- 从地址通过内部移位寄存器被送到SDA线上。
- 在7位地址模式时，只需送出一个地址字节。一旦该地址字节被送出，ADDR位被硬件置位，如果设置了ITEVFEN位，则产生一个中断。
- 随后主设备等待一次读SR1寄存器，跟着读SR2寄存器(见图245和图246)。根据送出从地址的最低位，主设备决定进入发送器模式还是进入接收器模式。

- 在7位地址模式时，
  - ✓ 要进入发送器模式，主设备发送从地址时置最低位为’0’。
  - ✓ 要进入接收器模式，主设备发送从地址时置最低位为’1’。

主发送器

- 在发送了地址和清除了ADDR位后, 主设备通过内部移位寄存器将字节从DR寄存器发送到SDA线上。
- 主设备等待，直到TxE被清除，(见图245的EV8)。
- 当收到应答脉冲时:
  - ✓ TxE位被硬件置位，如果设置了INEVFEN和ITBUFEN位，则产生一个中断。
- 如果TxE被置位并且在上一次数据发送结束之前没有写新的数据字节到DR寄存器，则BTF被硬件置位，在清除BTF之前I2C接口将保持SCL为低电平；读出I2C_SR1之后再写入I2C_DR寄存器将清除BTF位。

主接收器

- 在发送地址和清除ADDR之后，I2C接口进入主接收器模式。在此模式下，I2C接口从SDA线接收数据字节，并通过内部移位寄存器送至DR寄存器。
- 在每个字节后，I2C接口依次执行以下操作：
  - ✓ 如果ACK位被置位，发出一个应答脉冲。
  - ✓ 硬件设置RxNE=1，如果设置了INEVFEN和ITBUFEN位，则会产生一个中断(见图246的EV7)。
- 如果RxNE位被置位，并且在接收新数据结束前，DR寄存器中的数据没有被读走，硬件将设置BTF=1，在清除BTF之前I2C接口将保持SCL为低电平；读出I2C_SR1之后再读出I2C_DR寄存器将清除BTF位。

关闭通信（主发送器）

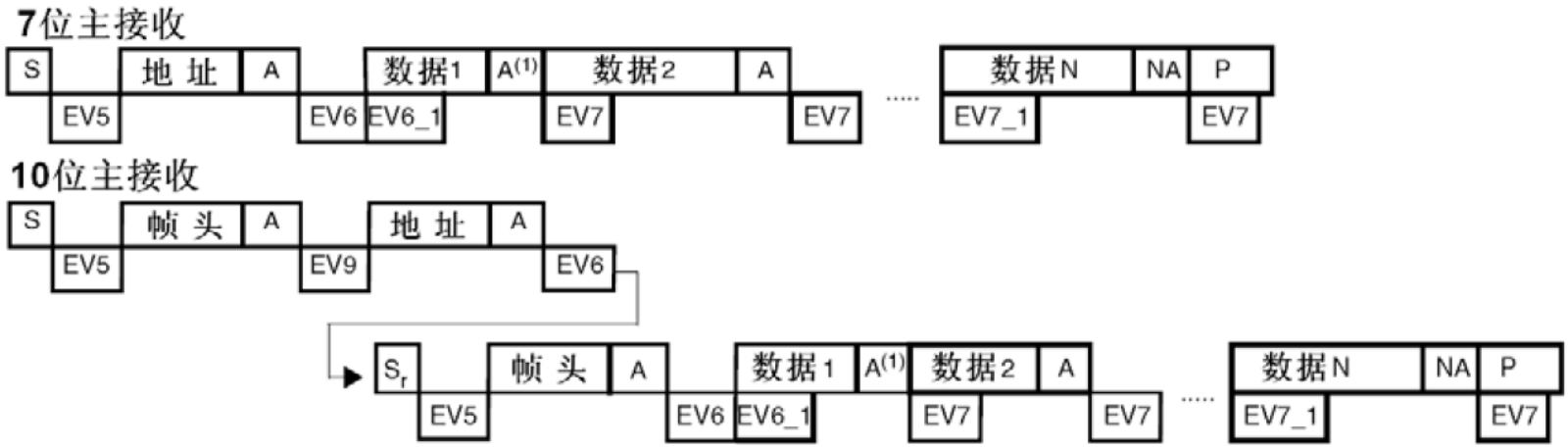
- 在DR寄存器中写入最后一个字节后，通过设置STOP位产生一个停止条件(见图245的EV8_2)，然后I2C接口将自动回到从模式(M/S位清除)。

注： 当TxE或BTF位置位时，停止条件应安排在出现EV8_2事件时。

关闭通信（主接收器）

- 主设备在从从设备接收到最后一个字节后发送一个NACK。接收到NACK后，从设备释放对SCL和SDA线的控制；主设备就可以发送一个停止/重起始条件。
  - ✓ 为了在收到最后一个字节后产生一个NACK脉冲，在读倒数第二个数据字节之后(在倒数第二个RxNE事件之后)必须清除ACK位。
  - ✓ 为了产生一个停止/重起始条件，软件必须在读倒数第二个数据字节之后(在倒数第二个RxNE事件之后)设置STOP/START位。
  - ✓ 只接收一个字节时，刚好在EV6之后(EV6_1时，清除ADDR之后)要关闭应答和停止条件的产生位。
- 在产生了停止条件后，I2C接口自动回到从模式(M/SL位被清除)。

图246　主接收器传送序列图



说明：S=Start(起始条件)，S_r=重复的起始条件，P=Stop(停止条件)，A=响应，NA=非响应，
EVx=事件(ITEVFEN=1时产生中断)

EV5：SB=1，读SR1然后将地址写入DR寄存器将清除该事件。
EV6：ADDR=1，读SR1然后读SR2将清除该事件。在10位主接收模式下，该事件后应设置CR2的START=1。
EV6_1：没有对应的事件标志，只适于接收1个字节的情况。恰好在EV6之后(即清除了ADDR之后)，要清除响应和停止条件的产生位。
EV7：RxNE=1，读DR寄存器清除该事件。
EV7_1：RxNE=1，读DR寄存器清除该事件。设置ACK=0和STOP请求。
EV9：ADDR10=1，读SR1然后写入DR寄存器将清除该事件。

1．如果收到一个单独的字节，则是NA。
2．EV5、EV6和EV9事件拉长SCL低电平，直到对应的软件序列结束。
3．EV7的软件序列必须在当前字节传输结束前完成。
4．EV6_1或EV7_1的软件序列 必须在当前传输字节的ACK脉冲之前完成。

错误条件

以下条件可能造成通讯失败。

总线错误(BERR)

在一个地址或数据字节传输期间，当I2C接口检测到一个外部的停止或起始条件则产生总线错误。此时：

- BERR位被置位为'1'；如果设置了ITERREN位，则产生一个中断；
- 在从模式情况下，数据被丢弃，硬件释放总线：
  - 如果是错误的开始条件，从设备认为是一个重启动，并等待地址或停止条件。
  - 如果是错误的停止条件，从设备按正常的停止条件操作，同时硬件释放总线。
- 在主模式情况下，硬件不释放总线，同时不影响当前的传输状态。此时由软件决定是否要中止当前的传输。

应答错误(AF)

当接口检测到一个无应答位时，产生应答错误。此时：

- AF位被置位，如果设置了ITERREN位，则产生一个中断；
- 当发送器接收到一个NACK时，必须复位通讯：
  - 如果是处于从模式，硬件释放总线。
  - 如果是处于主模式，软件必须生成一个停止条件。

仲裁丢失(ARLO)

当I2C接口检测到仲裁丢失时产生仲裁丢失错误，此时：

- ARLO位被硬件置位，如果设置了ITERREN位，则产生一个中断；
- I2C接口自动回到从模式(M/SL位被清除)。当I2C接口丢失了仲裁，则它无法在同一个传输中响应它的从地址，但它可以在赢得总线的主设备发送重起始条件之后响应；
- 硬件释放总线。

过载/欠载错误(OVR)

在从模式下，如果禁止时钟延长，I2C接口正在接收数据时，当它已经接收到一个字节(RxNE=1)，但在DR寄存器中前一个字节数据还没有被读出，则发生过载错误。此时：

- 最后接收的数据被丢弃；
- 在过载错误时，软件应清除RxNE位，发送器应该重新发送最后一次发送的字节。

在从模式下，如果禁止时钟延长，I2C接口正在发送数据时，在下一个字节的时钟到达之前，新的数据还未写入DR寄存器(TxE=1)，则发生欠载错误。此时：

- 在DR寄存器中的前一个字节将被重复发出；
- 用户应该确定在发生欠载错时，接收端应丢弃重复接收到的数据。发送端应按I2C总线标准在规定的时间更新DR寄存器。

在发送第一个字节时，必须在清除ADDR之后并且第一个SCL上升沿之前写入DR寄存器；如果不能做到这点，则接收方应该丢弃第一个数据。

SDA/SCL线控制

- 如果允许时钟延长：
  - 发送器模式：如果TxE=1且BTF=1：I2C接口在传输前保持时钟线为低，以等待软件读取SR1，然后把数据写进数据寄存器(缓冲器和移位寄存器都是空的)。
  - 接收器模式：如果RxNE=1且BTF=1：I2C接口在接收到数据字节后保持时钟线为低，以等待软件读SR1，然后读数据寄存器DR(缓冲器和移位寄存器都是满的)。
- 如果在从模式中禁止时钟延长：
  - 如果RxNE=1，在接收到下个字节前DR还没有被读出，则发生过载错。接收到的最后一个字节丢失。
  - 如果TxE=1，在必须发送下个字节之前却没有新数据写进DR，则发生欠载错。相同的字节将被重复发出。
  - 不控制重复写冲突。

DMA请求

- 概述

  DMA请求(当被使能时)仅用于数据传输。发送时数据寄存器变空或接收时数据寄存器变满，则产生DMA请求。DMA请求必须在当前字节传输结束之前被响应。当为相应DMA通道设置的数据传输量已经完成时，DMA控制器发送传输结束信号ETO到I2C接口，并且在中断允许时产生一个传输完成中断：

  - ✓ 主发送器：在EOT中断服务程序中，需禁止DMA请求，然后在等到BTF事件后设置停止条件。
  - ✓ 主接收器：当要接收的数据数目大于或等于2时，DMA控制器发送一个硬件信号EOT_1，它对应DMA传输(字节数-1)。如果在I2C_CR2寄存器中设置了LAST位，硬件在发送完EOT_1后的下一个字节，将自动发送NACK。在中断允许的情况下，用户可以在DMA传输完成的中断服务程序中产生一个停止条件。

利用DMA发送

- 通过设置I2C_CR2寄存器中的DMAEN位可以激活DMA模式。只要TxE位被置位，数据将由DMA从预置的存储区装载进I2C_DR寄存器。为I2C分配一个DMA通道，须执行以下步骤(x是通道号)：

1. 在DMA_CPARx寄存器中设置I2C_DR寄存器地址。数据将在每个TxE事件后从存储器传送至这个地址。
2. 在DMA_CMARx寄存器中设置存储器地址。数据在每个TxE事件后从这个存储区传送至I2C_DR。
3. 在DMA_CNDTRx寄存器中设置所需的传输字节数。在每个TxE事件后，此值将被递减。
4. 利用DMA_CCRx寄存器中的PL[0:1]位配置通道优先级。
5. 设置DMA_CCRx寄存器中的DIR位，并根据应用要求可以配置在整个传输完成一半或全部完成时发出中断请求。
6. 通过设置DMA_CCTx寄存器上的EN位激活通道。

- 当DMA控制器中设置的数据传输数目已经完成时，DMA控制器给I2C接口发送一个传输结束的 EOT/ EOT_1信号。在中断允许的情况下，将产生一个DMA中断。

注： 如果使用*DMA*进行发送时，不要设置*I2C_CR2*寄存器的*ITBUFEN*位。

利用DMA接收

- 通过设置I2C_CR2寄存器中的DMAEN位可以激活DMA接收模式。每次接收到数据字节时，将由DMA把I2C_DR寄存器的数据传送到设置的存储区(参考DMA说明)。设置DMA通道进行I2C接收，须执行以下步骤(x是通道号)：

  1. 在DMA_CPARx寄存器中设置I2C_DR寄存器的地址。数据将在每次RxNE事件后从此地址传送到存储区。
  2. 在DMA_CMARx寄存器中设置存储区地址。数据将在每次RxNE事件后从I2C_DR寄存器传送到此存储区。
  3. 在DMA_CNDTRx寄存器中设置所需的传输字节数。在每个RxNE事件后，此值将被递减。
  4. 用DMA_CCRx寄存器中的PL[0:1]配置通道优先级。
  5. 清除DMA_CCRx寄存器中的DIR位，根据应用要求可以设置在数据传输完成一半或全部完成时发出中断请求。
  6. 设置DMA_CCRx寄存器中的EN位激活该通道。

- 当DMA控制器中设置的数据传输数目已经完成时，DMA控制器给I2C接口发送一个传输结束的EOT/ EOT_1信号。在中断允许的情况下，将产生一个DMA中断。

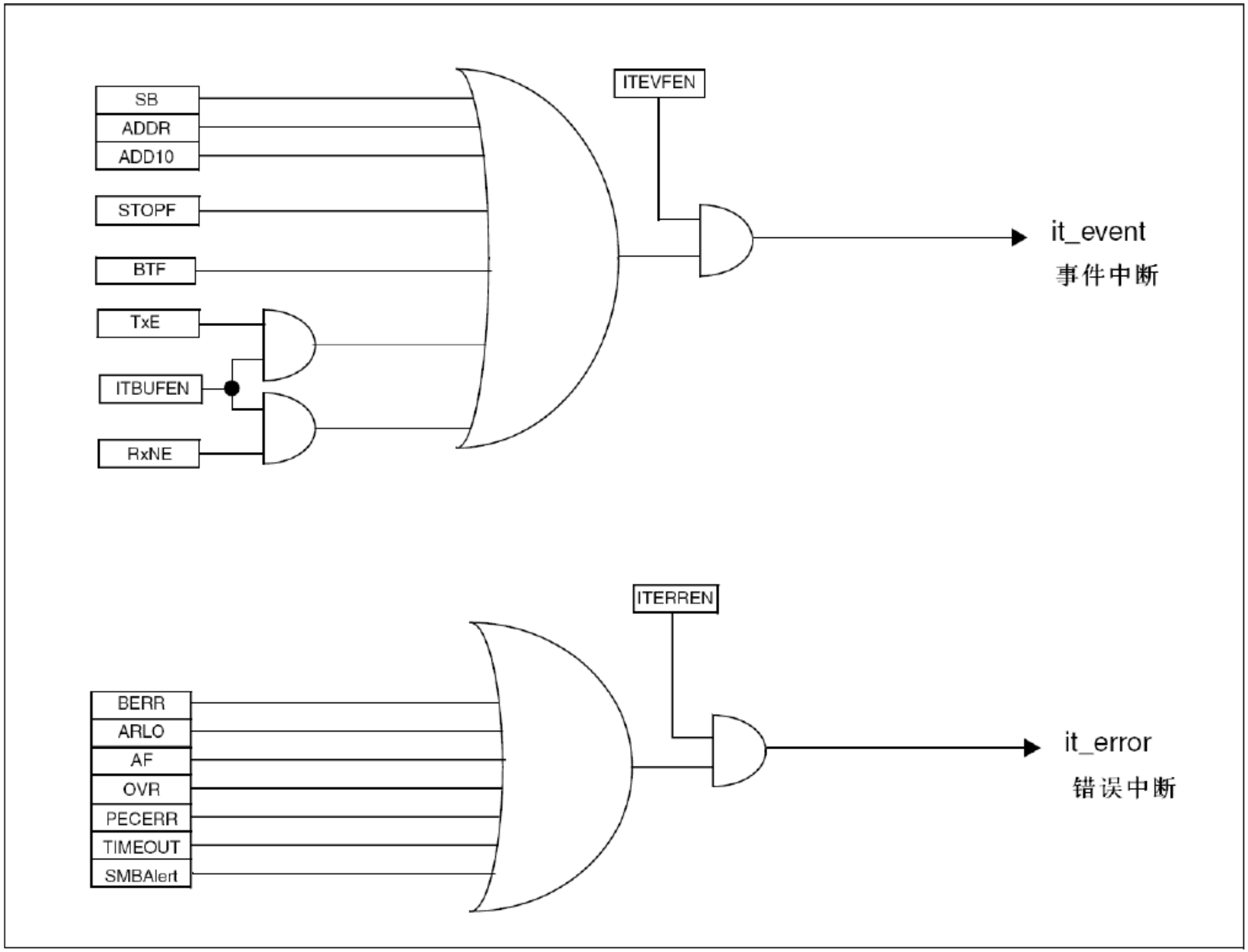
注： 如果使用*DMA*进行接收时，不要设置*I2C_CR2*寄存器的*ITBUFEN*位。

I2C中断请求

中断事件	事件标志	开启控制位
起始位已发送(主)	SB	ITEVFEN
地址已发送(主) 或 地址匹配(从)	ADDR	
10位头段已发送(主)	ADD10	
已收到停止(从)	STOPF	
数据字节传输完成	BTF	
接收缓冲区非空	RxNE	ITEVFEN 和 ITBUFEN
发送缓冲区空	TxE	
总线错误	BERR	ITERREN
仲裁丢失(主)	ARLO	
响应失败	AF	
过载/欠载	OVR	
PEC错误	PECERR	
超时/Tlow错误	TIMEOUT	
SMBus提醒	SMBALERT	

注： 1. SB、ADDR、ADD10、STOPF、BTF、RxNE和TxE通过逻辑或汇到同一个中断通道中。
2. BERR、ARLO、AF、OVR、PECERR、TIMEOUT和SMBALERT通过逻辑或汇到同一个中断通道中。

图247　I^2C中断映射图



IIC寄存器描述

- 控制寄存器1(I2C_CR1)

地址偏移：0x00 复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWRST	保留	ALERT	PEC	POS	ACK	STOP	START	NO STRETCH	ENGC	ENPEC	ENARP	SMB TYPE	保留	SMBUS	PE
rw	res	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	res	rw	rw

位15	**SWRST**：软件复位 (Software reset) 当被置位时，I^2C处于复位状态。在复位该位前确信I^2C的引脚被释放，总线是空的。 0：I^2C模块不处于复位状态； 1: I^2C模块处于复位状态。 **注**：该位可以用于BUSY位为'1'，在总线上又没有检测到停止条件时。
位14	保留位，硬件强制为0
位13	**ALERT**：SMBus提醒 (SMBus alert) 软件可以设置或清除该位；当PE=0时，由硬件清除。 0：释放SMBAlert引脚使其变高。提醒响应地址头紧跟在NACK信号后面； 1：驱动SMBAlert引脚使其变低。提醒响应地址头紧跟在ACK信号后面。

位12	**PEC**：数据包出错检测 (Packet error checking) 软件可以设置或清除该位；当传送PEC后，或起始或停止条件时，或当PE=0时硬件将其清除。 0：无PEC传输； 1：PEC传输(在发送或接收模式)。 注：仲裁丢失时，PEC的计算失效。
位11	**POS**: 应答/PEC位置(用于数据接收) (Acknowledge/PEC Position (for data reception)) 软件可以设置或清除该位，或当PE=0时，由硬件清除。 0：ACK位控制当前移位寄存器内正在接收的字节的(N)ACK。 PEC位表明当前移位寄存器内的字节是PEC； 1: ACK位控制在移位寄存器里接收的下一个字节的(N)ACK。PEC位表明在移位寄存器里接收的下一个字节是PEC。 注：POS位只能用在2字节的接收配置中，必须在接收数据之前配置。 为了NACK第2个字节，必须在清除ADDR为之后清除ACK位。 为了检测第2个字节的PEC，必须在配置了POS位之后，拉伸ADDR事件时设置PEC位。
位10	**ACK**：应答使能 (Acknowledge enable) 软件可以设置或清除该位，或当PE=0时，由硬件清除。 0：无应答返回； 1：在接收到一个字节后返回一个应答(匹配的地址或数据)。

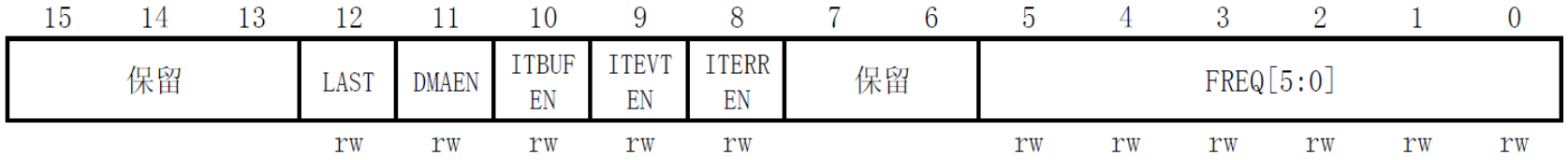
位9	**STOP**：停止条件产生 (Stop generation) 软件可以设置或清除该位；或当检测到停止条件时，由硬件清除；当检测到超时错误时，硬件将其置位。 在主模式下： 0：无停止条件产生； 1：在当前字节传输或在当前起始条件发出后产生停止条件。 在从模式下： 0：无停止条件产生； 1：在当前字节传输或释放SCL和SDA线。 注：当设置了STOP、START或PEC位，在硬件清除这个位之前，软件不要执行任何对I2C_CR1的写操作；否则有可能会第2次设置STOP、START或PEC位。
位8	**START**：起始条件产生 (Start generation) 软件可以设置或清除该位，或当起始条件发出后或PE=0时，由硬件清除。 在主模式下： 0: 无起始条件产生； 1：重复产生起始条件。 在从模式下： 0：无起始条件产生； 1：当总线空闲时，产生起始条件。

位7	**NOSTRETCH**：禁止时钟延长(从模式) (Clock stretching disable (Slave mode)) 该位用于当ADDR或BTF标志被置位，在从模式下禁止时钟延长，直到它被软件复位。 0：允许时钟延长； 1：禁止时钟延长。
位6	**ENGC**：广播呼叫使能 (General call enable) 0：禁止广播呼叫。以非应答响应地址00h； 1：允许广播呼叫. 以应答响应地址00h。
位5	**ENPEC**：PEC使能 (PEC enable) 0：禁止PEC计算； 1：开启PEC计算。
位4	**ENARP**：ARP使能 (ARP enable) 0：禁止ARP； 1：使能ARP。 如果SMBTYPE=0，使用SMBus设备的默认地址。 如果SMBTYPE=1，使用SMBus的主地址。
位3	**SMBTYPE**：SMBus类型 (SMBus type) 0：SMBus设备； 1：SMBus主机。
位2	保留位，硬件强制为0。

位1	**SMBUS**：SMBus模式 (SMBus mode) 0：I2C模式； 1：SMBus模式。
位0	**PE**：I^2C模块使能 (Peripheral enable) 0：禁用I^2C模块； 1：启用I^2C模块：根据SMBus位的设置，相应的I/O口需配置为复用功能。 **注**：如果清除该位时通讯正在进行，在当前通讯结束后，I^2C模块被禁用并返回空闲状态。 由于在通讯结束后发生PE＝0，所有的位被清除。 在主模式下，通讯结束之前，绝不能清除该位。

控制寄存器2(I2C_CR2)

地址偏移：0x04 复位值：0x0000



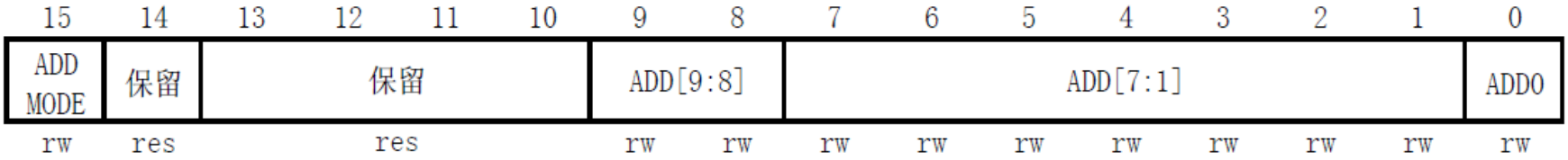
位15:13	保留位，硬件强制为0
位12	**LAST**: DMA最后一次传输 (DMA last transfer) 0：下一次DMA的EOT不是最后的传输； 1：下一次DMA的EOT是最后的传输。 **注**：该位在主接收模式使用， 使得在最后一次接收数据时可以产生一个NACK。
位11	**DMAEN**：DMA请求使能 (DMA requests enable) 0：禁止DMA请求； 1：当TxE=1或RxNE =1时，允许DMA请求。
位10	**ITBUFEN**：缓冲器中断使能 (Buffer interrupt enable) 0：当TxE=1或RxNE=1时，不产生任何中断； 1：当TxE=1或RxNE=1时，产生事件中断(不管DMAEN是何种状态)。

位9	**ITEVTEN**：事件中断使能 (Event interrupt enable) 0：禁止事件中断； 1：允许事件中断。 在下列条件下，将产生该中断： – SB = 1 (主模式)； – ADDR = 1 (主/从模式)； – ADD10= 1 (主模式)； – STOPF = 1 (从模式)； – BTF = 1，但是没有TxE或RxNE事件； – 如果ITBUFEN = 1，TxE事件为1； – 如果ITBUFEN = 1，RxNE事件为1。
位8	**ITERREN**：出错中断使能 (Error interrupt enable) 0：禁止出错中断； 1：允许出错中断。 在下列条件下，将产生该中断： – BERR = 1； – ARLO = 1； – AF = 1； – OVR = 1； – PECERR = 1； – TIMEOUT = 1； – SMBAlert = 1。

位7:6	保留位，硬件强制为0。
位5:0	**FREQ[5:0]**：I^2C模块时钟频率 (Peripheral clock frequency) 必须设置正确的输入时钟频率以产生正确的时序，允许的范围在2～36MHz之间： 000000：禁用 000001：禁用 000010：2MHz ... 100100：36MHz 大于100100：禁用。

自身地址寄存器1(I2C_OAR1)

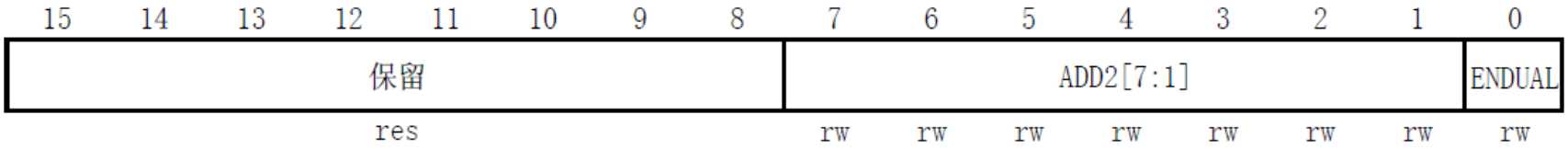
复位地址偏移：0x08 复位值：0x0000



位15	**ADDMODE**：寻址模式(从模式) (Addressing mode (slave mode)) 0：7位从地址(不响应10位地址)； 1：10位从地址(不响应7位地址)。
位14	必须始终由软件保持为'1'。
位13:10	保留位，硬件强制为0。
位9:8	**ADD[9:8]**：接口地址 (Interface address) 7位地址模式时不用关心。 10位地址模式时为地址的9~8位。
位7:1	**ADD[7:1]**：接口地址 (Interface address) 地址的7~1位。
位0	**ADD0**：接口地址 (Interface address) 7位地址模式时不用关心。 10位地址模式时为地址第0位。

自身地址寄存器2(I2C_OAR2)

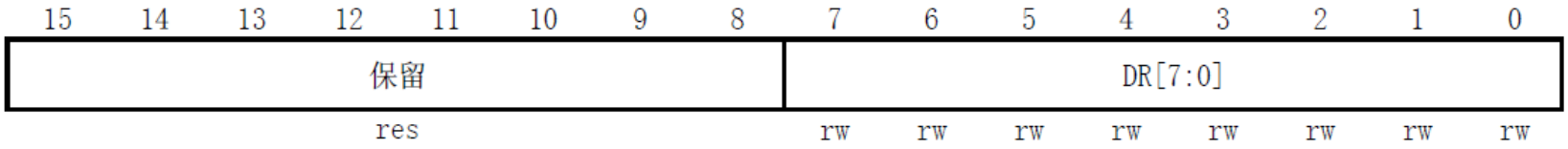
地址偏移：0x0C　复位值：0x0000



位15:8	保留位，硬件强制为0
位7:1	**ADD2[7:1]**：接口地址 (Interface address) 在双地址模式下地址的7~1位。
位0	**ENDUAL**：双地址模式使能位 (Dual addressing mode enable) 0：在7位地址模式下，只有OAR1被识别； 1：在7位地址模式下，OAR1和OAR2都被识别。

数据寄存器(I2C_DR)

地址偏移：0x10　复位值：0x0000



位15:8	保留位，硬件强制为0
位7:0	**DR[7:0]**：8位数据寄存器 (8-bit data register) 用于存放接收到的数据或放置用于发送到总线的数据 发送器模式：当写一个字节至DR寄存器时，自动启动数据传输。一旦传输开始(TxE=1)，如果能及时把下一个需传输的数据写入DR寄存器，I^2C模块将保持连续的数据流。 接收器模式：接收到的字节被拷贝到DR寄存器(RxNE=1)。在接收到下一个字节(RxNE=1)之前读出数据寄存器，即可实现连续的数据传送。 注：在从模式下，地址不会被拷贝进数据寄存器DR； 注：硬件不管理写冲突(如果TxE=0，仍能写入数据寄存器)； 注：如果在处理ACK脉冲时发生ARLO事件，接收到的字节不会被拷贝到数据寄存器里，因此不能读到它。

状态寄存器1(I2C_SR1)

地址偏移：0x14 复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMB ALERT	TIME OUT	保留	PEC ERR	OVR	AF	ARLO	BERR	TxE	RxNE	保留	STOPF	ADD10	BTF	ADDR	SB
rc w0	rc w0	res	rc w0	rc w0	rc w0	rc w0	rc w0	r	r	res	r	r	r	r	r

位15	**SMBALERT**：SMBus提醒 (SMBus alert) 在SMBus主机模式下： 0：无SMBus提醒； 1：在引脚上产生SMBAlert提醒事件。 在SMBus从机模式下： 0：没有SMBAlert响应地址头序列； 1：收到SMBAlert响应地址头序列至SMBAlert变低。 – 该位由软件写'0'清除，或在PE=0时由硬件清除。
位14	**TIMEOUT**：超时或Tlow错误 (Timeout or Tlow error) 0：无超时错误； 1：SCL处于低已达到25ms(超时)；或者主机低电平累积时钟扩展时间超过10ms(Tlow:mext)；或从设备低电平累积时钟扩展时间超过25ms(Tlow:sext)。 – 当在从模式下设置该位：从设备复位通讯，硬件释放总线。 – 当在主模式下设置该位：硬件发出停止条件。 – 该位由软件写'0'清除，或在PE=0时由硬件清除。
位13	保留位，硬件强制为0。

位12	**PECERR**：在接收时发生PEC错误 (PEC Error in reception) 0: 无PEC错误：接收到PEC后接收器返回ACK(如果ACK=1)； 1: 有PEC错误：接收到PEC后接收器返回NACK(不管ACK是什么值)。 – 该位由软件写'0'清除，或在PE=0时由硬件清除。
位11	**OVR**：过载/欠载 (Overrun/Underrun) 0: 无过载/欠载； 1: 出现过载/欠载。 – 当NOSTRETCH=1时，在从模式下该位被硬件置位，同时： – 在接收模式中当收到一个新的字节时(包括ACK应答脉冲)，数据寄存器里的内容还未被读出，则新接收的字节将丢失。 – 在发送模式中当要发送一个新的字节时，却没有新的数据写入数据寄存器，同样的字节将被发送两次。 – 该位由软件写'0'清除，或在PE=0时由硬件清除。。 **注**：如果数据寄存器的写操作发生时间非常接近SCL的上升沿，发送的数据是不确定的，并发生保持时间错误。
位10	**AF**：应答失败 (Acknowledge failure) 0：没有应答失败； 1：应答失败。 – 当没有返回应答时，硬件将置该位为'1'。 – 该位由软件写'0'清除，或在PE=0时由硬件清除。

位9	**ARLO**：仲裁丢失(主模式) (Arbitration lost (master mode)) 0：没有检测到仲裁丢失； 1：检测到仲裁丢失。 当接口失去对总线的控制给另一个主机时，硬件将将置该位为'1'。 – 该位由软件写'0'清除，或在PE=0时由硬件清除。 在ARLO事件之后，I^2C接口自动切换回从模式(M/SL=0)。 **注**：在SMBUS模式下，在从模式下对数据的仲裁仅仅发生在数据阶段，或应答传输区间(不包括地址的应答)。
位8	**BERR**：总线出错 (Bus error) 0：无起始或停止条件出错； 1：起始或停止条件出错。 – 当接口检测到错误的起始或停止条件，硬件将该位置'1'。 – 该位由软件写'0'清除，或在PE=0时由硬件清除。
位7	**TxE**：数据寄存器为空(发送时) (Data register empty (transmitters)) 0：数据寄存器非空； 1：数据寄存器空。 – 在发送数据时，数据寄存器为空时该位被置'1'，在发送地址阶段不设置该位。 – 软件写数据到DR寄存器可清除该位；或在发生一个起始或停止条件后，或当PE=0时由硬件自动清除。 如果收到一个NACK，或下一个要发送的字节是PEC(PEC=1)，该位不被置位。 注：在写入第1个要发送的数据后，或设置了BTF时写入数据，都不能清除TxE位，这是因为数据寄存器仍然为空。

位6	**RxNE**：数据寄存器非空(接收时) (Data register not empty (receivers)) 0：数据寄存器为空； 1：数据寄存器非空。 – 在接收时，当数据寄存器不为空，该位被置'1'。在接收地址阶段，该位不被置位。 – 软件对数据寄存器的读写操作清除该位，或当PE=0时由硬件清除。 在发生ARLO事件时，RxNE不被置位。 注：当设置了BTF时，读取数据不能清除RxNE位，因为数据寄存器仍然为满。
位5	保留位，硬件强制为0
位4	**STOPF**：停止条件检测位(从模式) (Stop detection (slave mode)) 0：没有检测到停止条件； 1：检测到停止条件。 – 在一个应答之后(如果ACK=1)，当从设备在总线上检测到停止条件时，硬件将该位置'1'。 – 软件读取SR1寄存器后，对CR1寄存器的写操作将清除该位，或当PE=0时，硬件清除该位。 **注**：在收到NACK后，STOPF位不被置位。
位3	**ADD10**：10位头序列已发送(主模式) (10-bit header sent (Master mode)) 0：没有ADD10事件发生； 1：主设备已经将第一个地址字节发送出去。 – 在10位地址模式下，当主设备已经将第一个字节发送出去时，硬件将该位置'1'。 – 软件读取SR1寄存器后，对CR1寄存器的写操作将清除该位，或当PE=0时，硬件清除该位。 **注**：收到一个NACK后，ADD10位不被置位。

位2	**BTF**：字节发送结束 (Byte transfer finished) 0：字节发送未完成； 1：字节发送结束。 当NOSTRETCH=0时，在下列情况下硬件将该位置'1'： – 在接收时，当收到一个新字节(包括ACK脉冲)且数据寄存器还未被读取(RxNE=1)。 – 在发送时，当一个新数据将被发送且数据寄存器还未被写入新的数据(TxE=1)。 – 在软件读取SR1寄存器后，对数据寄存器的读或写操作将清除该位；或在传输中发送一个起始或停止条件后，或当PE=0时，由硬件清除该位。 **注**：在收到一个NACK后，BTF位不会被置位。 如果下一个要传输的字节是PEC(I2C_SR2寄存器中TRA为'1'，同时I2C_CR1寄存器中PEC为'1')，BTF位不会被置位。
位1	**ADDR**：地址已被发送(主模式)/地址匹配(从模式) (Address sent (master mode)/matched (slave mode)) 在软件读取SR1寄存器后，对SR2寄存器的读操作将清除该位，或当PE=0时，由硬件清除该位。 **地址匹配(从模式)** 0：地址不匹配或没有收到地址； 1：收到的地址匹配。 – 当收到的从地址与OAR寄存器中的内容相匹配、或发生广播呼叫、或SMBus设备默认地址或SMBus主机识别出SMBus提醒时，硬件就将该位置'1'(当对应的设置被使能时)。 **地址已被发送(主模式)** 0：地址发送没有结束； 1：地址发送结束。 – 10位地址模式时，当收到地址的第二个字节的ACK后该位被置'1'。 – 7位地址模式时，当收到地址的ACK后该位被置'1'。 **注**：在收到NACK后，ADDR位不会被置位。
位0	**SB**：起始位(主模式) (Start bit (Master mode)) 0：未发送起始条件； 1：起始条件已发送。 – 当发送出起始条件时该位被置'1'。 – 软件读取SR1寄存器后，写数据寄存器的操作将清除该位，或当PE=0时，硬件清除该位。

状态寄存器2 (I2C_SR2)

地址偏移：0x18　复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PEC[7:0]								DUALF	SMB HOST	SMB DEFAULT	GEN CALL	保留	TRA	BUSY	MSL
r	r	r	r	r	r	r	r	r	r	r	r	res	r	r	r

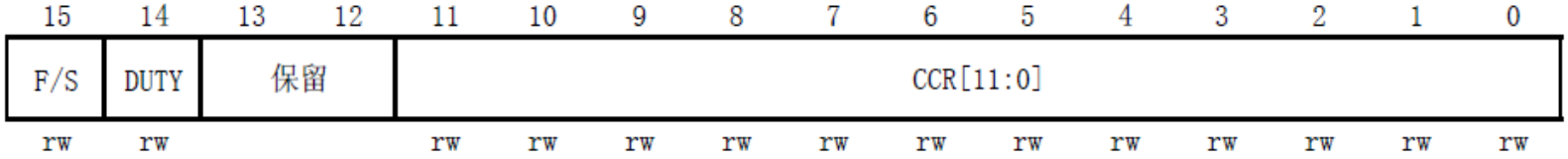
位15:8	**PEC[7:0]**：数据包出错检测 (Packet error checking register) 当ENPEC=1时，PEC[7:0]存放内部的PEC的值。
位7	**DUALF**：双标志(从模式) (Dual flag (Slave mode)) 0：接收到的地址与OAR1内的内容相匹配； 1：接收到的地址与OAR2内的内容相匹配。 – 在产生一个停止条件或一个重复的起始条件时，或PE=0时，硬件将该位清除。
位6	**SMBHOST**: SMBus主机头系列(从模式) (SMBus host header (Slave mode)) 0：未收到SMBus主机的地址； 1：当SMBTYPE=1且ENARP=1时，收到SMBus主机地址。 – 在产生一个停止条件或一个重复的起始条件时，或PE=0时，硬件将该位清除。

位5	**SMBDEFAULT**：SMBus设备默认地址(从模式) (SMBus device default address (Slave mode)) 0：未收到SMBus设备的默认地址； 1：当ENARP=1时，收到SMBus设备的默认地址。 – 在产生一个停止条件或一个重复的起始条件时，或PE=0时，硬件将该位清除。
位4	**GENCALL**：广播呼叫地址(从模式) (General call address (Slave mode)) 0：未收到广播呼叫地址； 1：当ENGC=1时，收到广播呼叫的地址。 – 在产生一个停止条件或一个重复的起始条件时，或PE=0时，硬件将该位清除。
位3	保留位，硬件强制为0
位2	**TRA**：发送/接收 (Transmitter/receiver) 0：接收到数据； 1：数据已发送； 在整个地址传输阶段的结尾，该位根据地址字节的R/W位来设定。 在检测到停止条件(STOPF=1)、重复的起始条件或总线仲裁丢失(ARLO=1)后，或当PE=0时，硬件将其清除。
位1	**BUSY**：总线忙 (Bus busy) 0：在总线上无数据通讯； 1：在总线上正在进行数据通讯。 – 在检测到SDA或SCl为低电平时，硬件将该位置'1'； – 当检测到一个停止条件时，硬件将该位清除。 该位指示当前正在进行的总线通讯，当接口被禁用(PE=0)时该信息仍然被更新。
位0	**MSL**：主从模式 (Master/slave) 0：从模式； 1：主模式。 – 当接口处于主模式(SB=1)时，硬件将该位置位； – 当总线上检测到一个停止条件、仲裁丢失(ARLO=1时)、或当PE=0时，硬件清除该位。

时钟控制寄存器(I2C_CCR)

地址偏移：0x1C 复位值：0x0000

注： *1.* 要求F_{PCLK1}应当是*10 MHz*的整数倍，这样可以正确地产生*400KHz*的快速时钟。
*2. CCR*寄存器只有在关闭*I2C*时*(PE=0)*才能设置

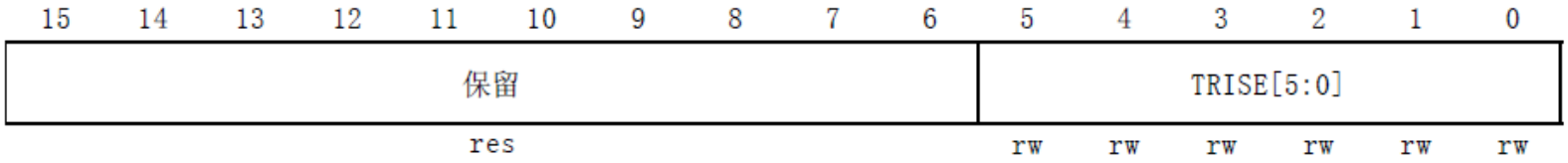


位15	**F/S**：I^2C主模式选项 (I^2C master mode selection) 0：标准模式的I^2C； 1：快速模式的I^2C。
位14	**DUTY**：快速模式时的占空比 (Fast mode duty cycle) 0：快速模式下：T_{low}/T_{high} = 2； 1: 快速模式下：T_{low}/T_{high} = 16/9(见CCR)。
位13:12	保留位，硬件强制为0。

位11:0	**CCR[11:0]**：快速/标准模式下的时钟控制分频系数(主模式) (Clock control register in Fast/Standard mode (Master mode)) 该分频系数用于设置主模式下的SCL时钟。 在I^2C标准模式或SMBus模式下： T_{high} = CCR ×T_{PCLK1} T_{low} = CCR ×T_{PCLK1} 在I^2C快速模式下： 如果DUTY = 0： T_{high} = CCR ×T_{PCLK1} T_{low} = 2 × CCR × T_{PCLK1} 如果DUTY = 1：(速度达到400kHz) T_{high} = 9 × CCR ×T_{PCLK1} T_{low} = 16 × CCR ×T_{PCLK1} 例如：在标准模式下，产生100kHz的SCL的频率： 如果FREQR = 08，T_{PCLK1} = 125ns，则CCR必须写入0x28(40×125ns = 5000 ns)。 **注**：1. 允许设定的最小值为0x04，在快速DUTY模式下允许的最小值为0x01； 2. $T_{high}=t_{r(SCL)}+t_{w(SCLH)}$，详见数据手册中对这些参数的定义； 3. $T_{low}=t_{f(SCL)}+t_{w(SCLL)}$，详见数据手册中对这些参数的定义； 4. 这些延时没有过滤器； 5. 只有在关闭I^2C时(PE = 0)才能设置CCR寄存器； 6. f_{CK}应当是10MHz的整数倍，这样可以正确产生400kHz的快速时钟。

TRISE寄存器(I2C_TRISE)

地址偏移：0x20 复位值：0x0002



位15:6	保留位，硬件强制为0
位5:0	**TRISE[5:0]**：在快速/标准模式下的最大上升时间(主模式) (Maximum rise time in Fast/Standard mode (Master mode)) 这些位必须设置为I^2C总线规范里给出的最大的SCL上升时间，增长步幅为1。 例如：标准模式中最大允许SCL上升时间为1000ns。如果在I2C_CR2寄存器中FREQ[5:0]中的值等于0x08且T_{PCLK1}=125ns，故TRISE[5:0]中必须写入09h(1000ns/125 ns = 8+1)。 滤波器的值也可以加到TRISE[5:0]内。 如果结果不是一个整数，则将整数部分写入TRISE[5:0]以确保t_{HIGH}参数。 注：只有当I^2C被禁用(PE=0)时，才能设置TRISE[5:0]。

I2C寄存器描述

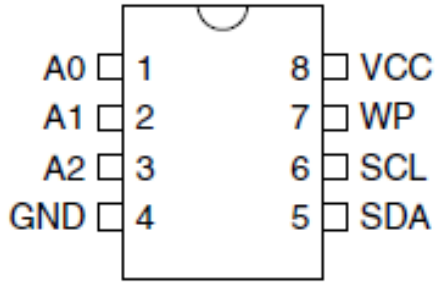
表174 I²C寄存器地址映象和复位值

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	I2C_CR1	保留																SWRST	保留	ALERT	PEC	POS	ACK	STOP	START	NOSTRETCH	ENGC	ENPEC	ENARP	SMBTYPE	保留	SMBUS	PE
	复位值																	0		0	0	0	0	0	0	0	0	0	0	0		0	0
0x04	I2C_CR2	保留																			LAST	DMAEN	ITBUFEN	ITEVTEN	ITERREN	保留	FREQ[5:0]						
	复位值																				0	0	0	0	0		0	0	0	0	0	0	
0x08	I2C_OAR1	保留																ADDMODE	保留	保留				ADD [9:8]		ADD[7:1]							ADD0
	复位值																	0	1					0	0	0	0	0	0	0	0		0
0x0C	I2C_OAR2	保留																								ADD2[7:1]							ENDUAL
	复位值																									0	0	0	0	0	0	0	0
0x10	I2C_DR	保留																								DR[7:0]							
	复位值																									0	0	0	0	0	0	0	0
0x14	I2C_SR1	保留																SMBALERT	TIMEOUT	保留	PECERR	OVR	AF	ARLO	BERR	TxE	RxNE	保留	STOPF	ADD10	BTF	ADDR	SB
	复位值																	0	0		0	0	0	0	0	0	0		0	0	0	0	0
0x18	I2C_SR2	保留																PEC[7:0]								DUALF	SMBHOST	SMBDEFAU	GENCALL	保留	TRA	BUSY	MSL
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0
0x1C	I2C_CCR	保留																F/S	DUTY	保留	CCR[11:0]												
	复位值																	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	I2C_TRISE	保留																										TRISE[5:0]					
	复位值																											0	0	0	0	1	0

关于寄存器起始地址，参见表1。

附录1、AT24C02

- 24C02是一个2K Bit的串行EEPROM存储器（掉电不丢失），内部含有256个字节。在24C02里面有一个8字节的页写缓冲器。



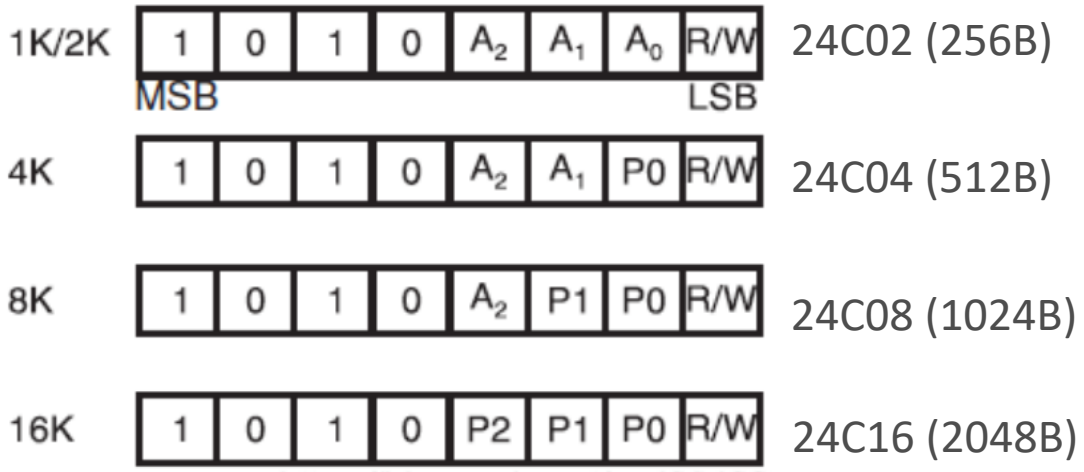
引脚说明：

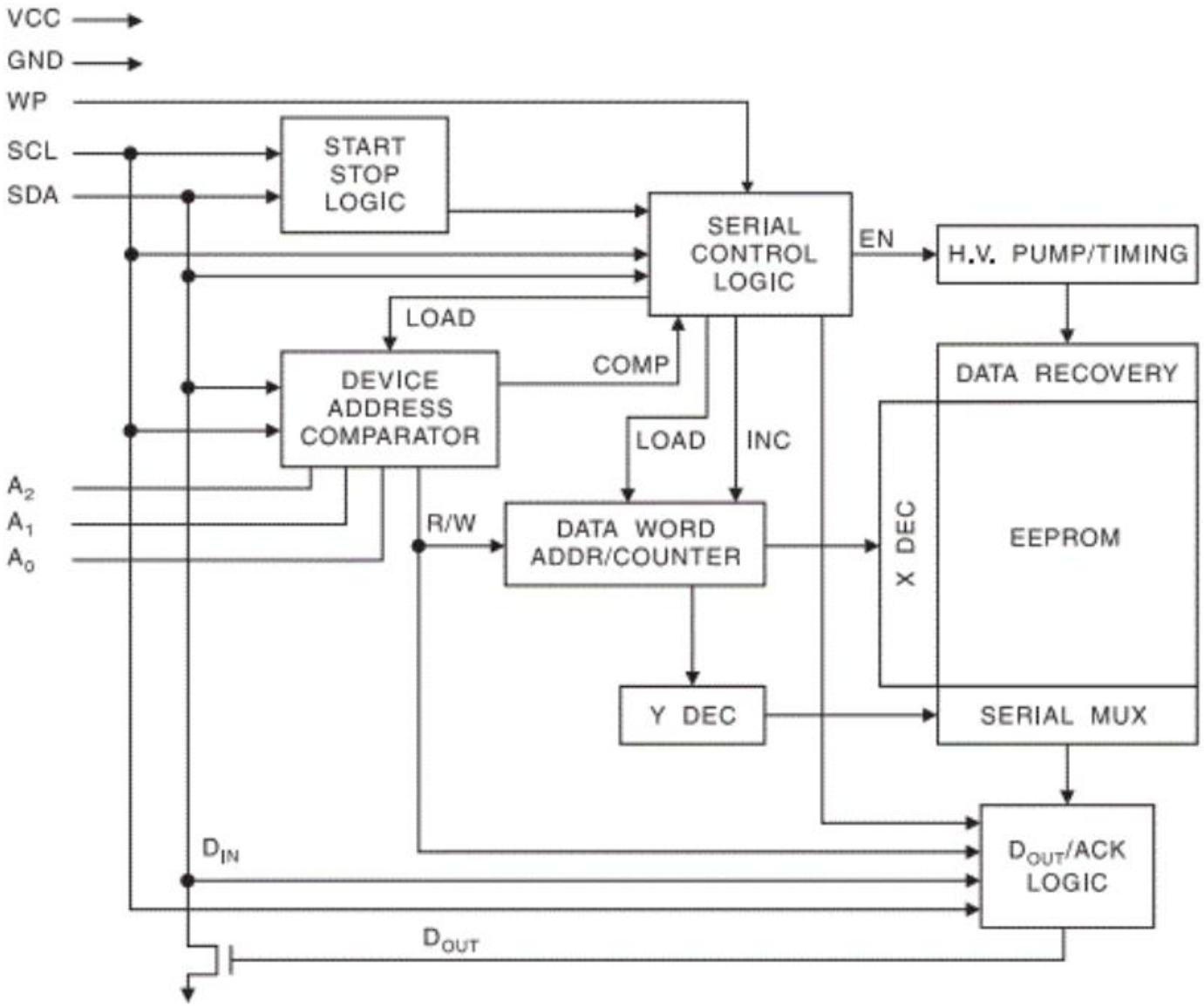
A2~A0 用于选择从设备地址

P0~P2 悬空或接Vss

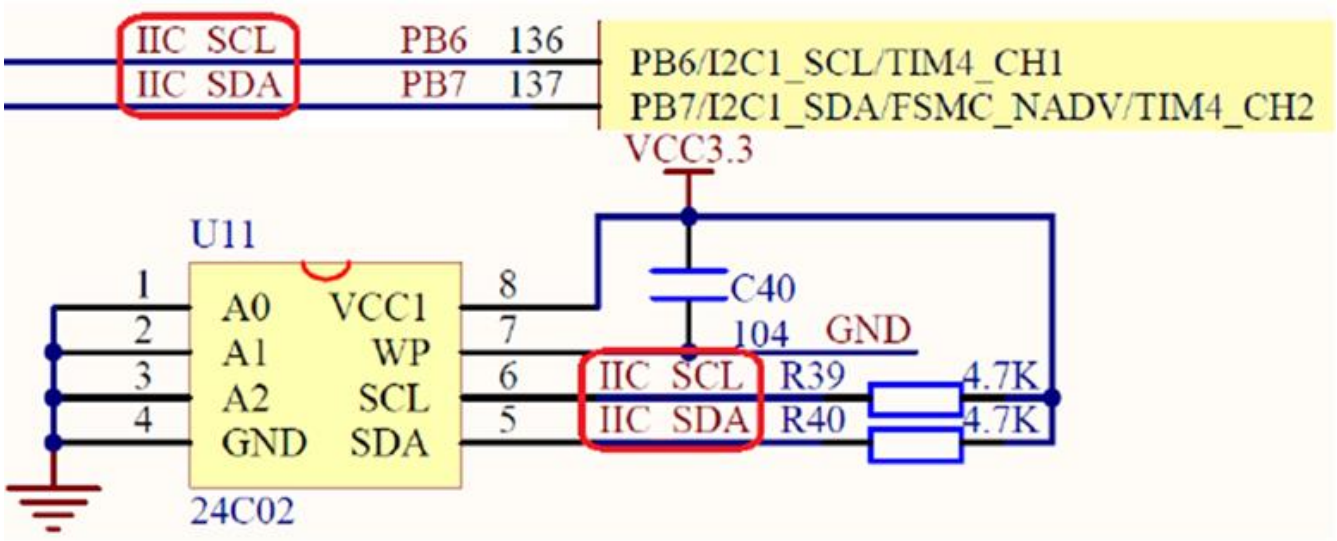
WP 写保护，高电平有效

device address



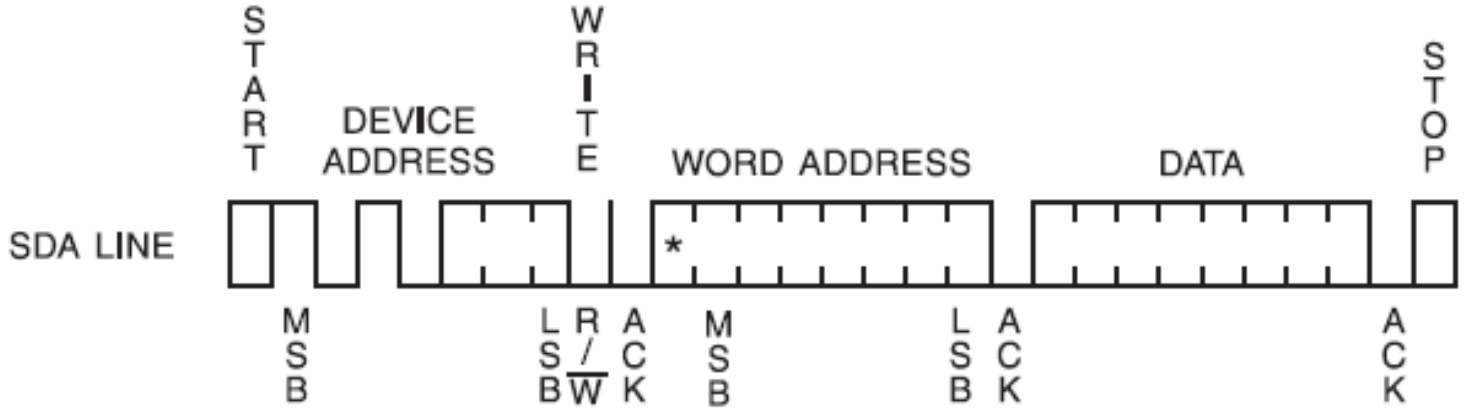

VCC
GND
WP
SCL
SDA
START
STOP
LOGIC
SERIAL
CONTROL
LOGIC
EN
H.V. PUMP/TIMING
LOAD
DEVICE
ADDRESS
COMPARATOR
COMP
DATA RECOVERY
LOAD
INC
A2
A1
A0
R/W
DATA WORD
ADDR/COUNTER
X DEC
EEPROM
Y DEC
SERIAL MUX
DIN
DOUT/ACK
LOGIC
DOUT

- STM32与24C02连接图



U11的设备地址为10100*000(A2A1A0)*

- **24C02字节写时序**

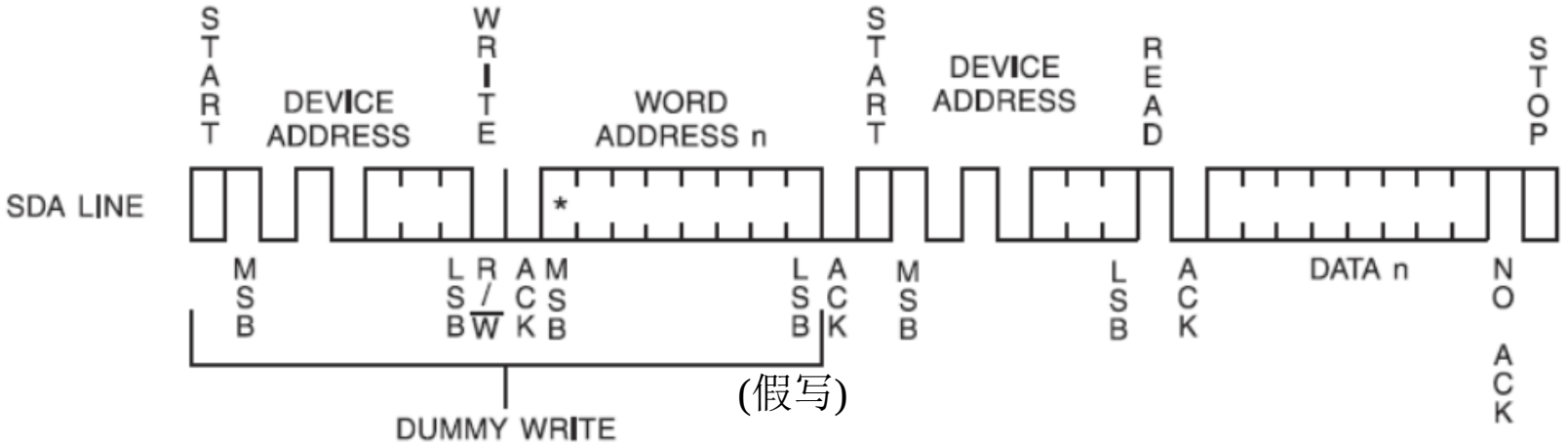


对24C02芯片进行写字节操作的时候，步骤如下：

1. 主设备发送一个开始信号(START)；
2. 接着发送7位从设备地址(DEVICE ADDRESS)和1个写操作位，即0xA0；
3. 等待应答信号(ACK)；
4. 发送数据的存储地址。24C02的地址从0x00~0xFF；
5. 然后发送要存储数据的第一字节、第二字节、…；每接收一个字节，24C02都会发出一个应答位报告该字节写入成功，如果没有回应答位，说明写入不成功。
6. 发送结束信号（STOP）停止总线。

注意：每成功写入一个字节，E2PROM的存储地址就会自动加1，当加到0xFF后，再写一个字节，地址就会溢出又变成0x00。写数据的时候需要注意，E2PROM是先写到缓冲区，然后再“搬运到”到掉电非易失区。 AT24C02的这个过程至少需要5ms！所以，在写多个字节时，每写一个字节之后，必须延时5ms才可以下一个字节。

- **24C02读随机地址的数据的时序**



1. 主设备先发送一个开始信号。
2. 发送7位器件地址和1位写操作位，即0xA0。
3. 发送要读取内存的地址(WORD ADDRESS) 。
4. 重新发送开始信号。
5. 发送7位设备地址和1位读操作位，即0xA1。
6. E2PROM会自动向主设备发送数据，每读入一个字节，主设备会回应一个应答信号ACK(0)，E2PROM会继续传输下一个地址的数据，主设备再发应答信号，...。
7. 如果主设备不想读了，就发送一个“非应答位NAK(1)”。
8. 最后发送结束信号。

- **24C02读当前地址数据的时序**