

IIC总线协议，AT24C08存储器

——阿拉丁神丢

IIC总线（Inter Integrated Circuit Bus）：是Philips公司推出的串行总线标准（为二线制）。总线上扩展的外围器件及外设接口通过总线寻址，是具备总线仲裁和高低速设备同步等功能的高性能多主机总线。

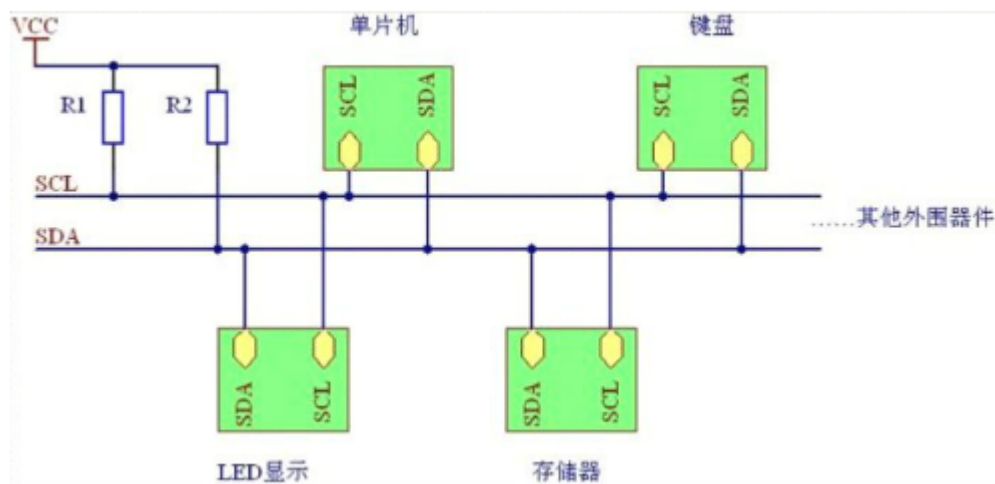
I2C总线工作原理

串行数据线SDA和串行时钟线SCL构成的，可发送和接收数据。

■所有挂载在I2C总线上的器件和接口电路都应具有I2C总线接口，且所有的SDA/SCL同名端相连。总线上所有器件要依靠SDA发送的地址信号寻址，不需要片选线。

■特点：组成系统结构简单，占用空间小，芯片管脚的数量少，无需片选信号，价格低。允许若干兼容器件共享总线，应用比较广泛。总线的长度可达7.6m，传送速度可达400kbps，标准速率为100kbps。支持多个组件。支持多主控器件（某时刻只能有一个主控器件）。I2C总线上所有设备的SDA, SCL引脚必须外接上拉电阻。

典型的I2C总线系统结构,如图所示



I2C总线器件的寻址方式

由于所有器件都通过SCL和SDA连接在I2C总线上，因此，主器件在进行数据传输前需要通过寻址，选择需要通信的从器件。I2C总线上所有外围器件都需要有唯一的7位地址，由器件地址和引脚地址两部分组成。

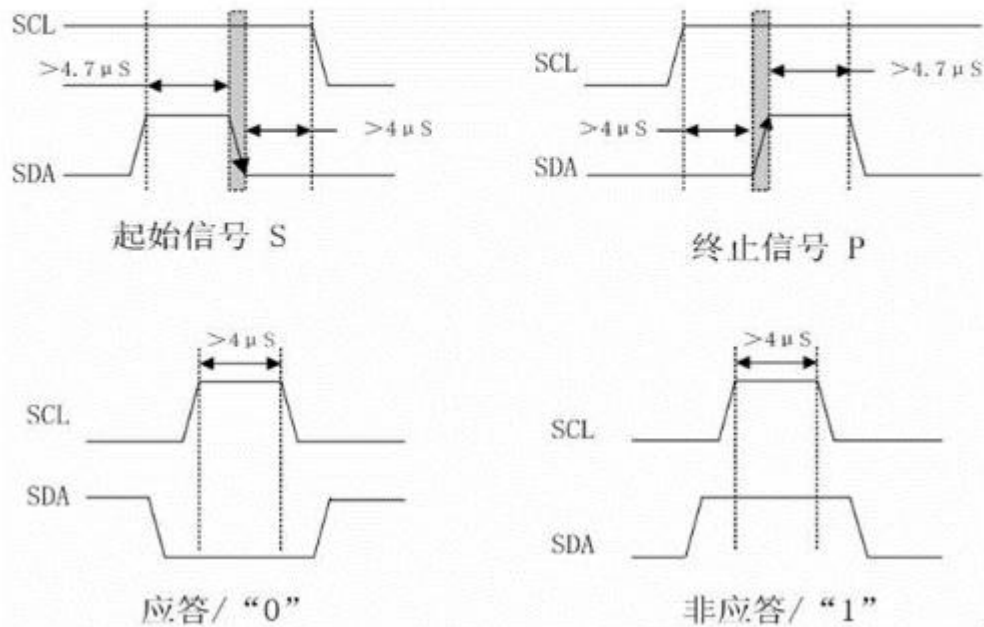
器件地址是I2C器件固有的地址编码，器件出厂时就已经给定，不可更改。

引脚地址是由I2C总线外围器件的地址引脚（A2，A1，A0）决定，根据其在电路中接电源正极、接地或悬空的不同，形成不同的地址代码。

I2C 总线

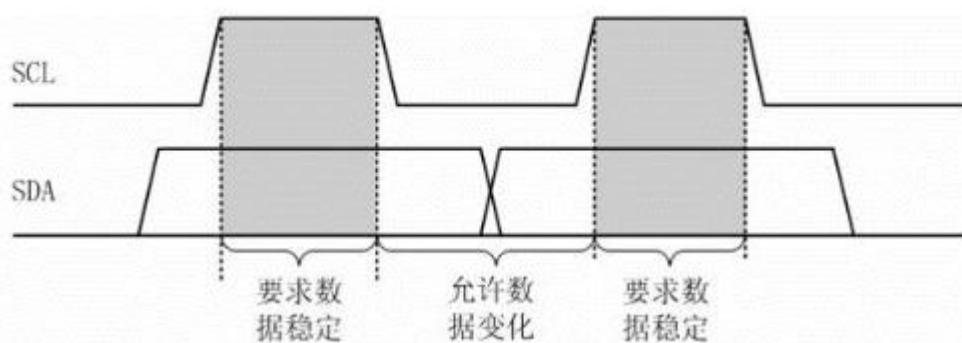
概述

- I2C总线采用两线制，由数据线SDA和时钟线SCL构成。I2C总线对数据通信时序进行了严格的定义。



I2C总线数据操作

在I2C总线上，数据是伴随着时钟脉冲，一位一位地传送的，数据位由**高到低**传送，每位数据占一个时钟脉冲。I2C总线上的在时钟线SCL高电平期间，数据线SDA的状态就表示要传送的数据，高电平为数据1，低电平为数据0。在数据传送时，SDA上数据的改变在时钟线为低电平时完成，而SCL为高电平时，SDA必须保持稳定，否则SDA上的变化会被当作起始或终止信号而致使数据传输停止。



I2C总线在传送数据过程中共有三种类型信号：开始信号、停止信号和应答信号。

开始信号：**SCL**保持高电平的状态下，**SDA**出现下降沿。出现开始信号以后，总线被认为“忙”。

停止信号：**SCL**保持高电平的状态下，**SDA**出现上升沿。停止信号过后，总线被认为“空闲”。

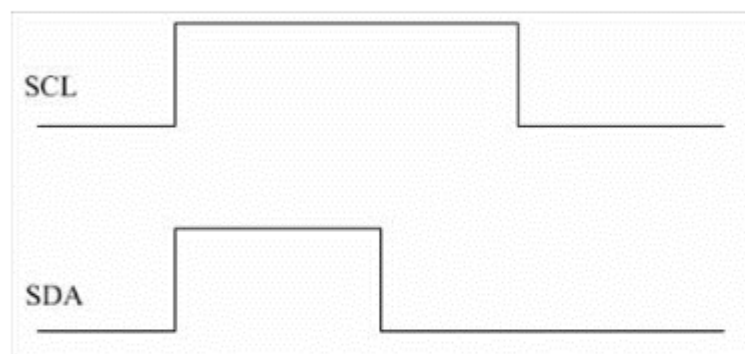
应答信号：接收数据的器件在接收到8位数据后，向发送数据的器件发出特定的低电平脉冲，表示已收到数据。

总线空闲：**SCL**和**SDA**都保持高电平。

总线忙：在数据传送开始以后，**SCL**为高电平的时候，**SDA**的数据必须保持稳定，只有当**SCL**为低电平的时候才允许**SDA**上的数据改变。

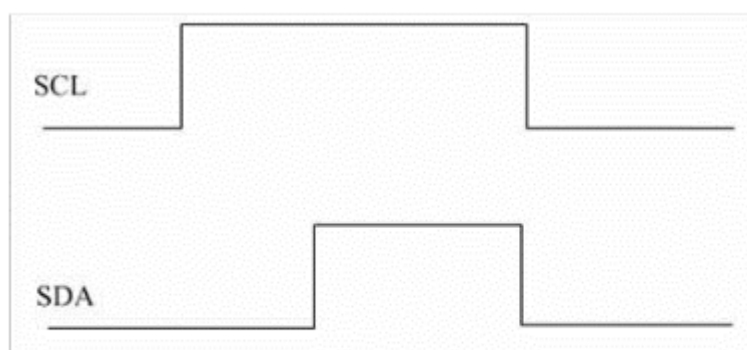
I2C协议起始信号时序

起始信号用于开始I2C总线通信。其中，起始信号是在时钟线**SCL**为高电平期间，数据**SDA**上高电平向低电平变化的下降沿信号。起始信号出现以后，才可以进行后续的I2C总线寻址或数据传输等



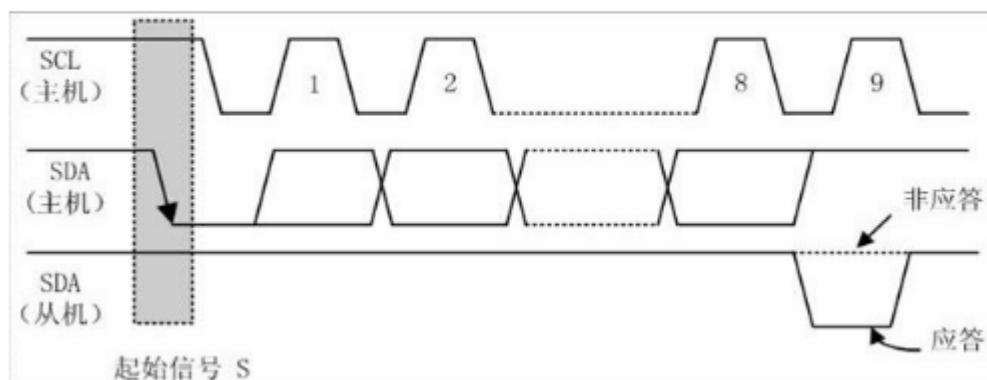
I2C协议终止信号时序

终止信号用于终止I2C总线通信。其中，终止信号是在时钟线**SCL**为高电平期间，数据线**SDA**上低电平到高电平变化的上升沿信号。终止信号一出现，所有I2C总线操作都结束，并释放总线控制权。终止信号的时序如下：



I2C协议应答信号时序

应答信号用于表明I2C总线数据传输的结束。I2C总线数据传输时，一个字节数据传输完毕后都必须由主器件产生应答信号。主器件在第9个时钟位上释放数据总线SDA，使其处于高电平状态，此时从器件输出低电平拉低数据总线SDA为应答信号。应答信号的时序，如图所示。



I2C总线的传送格式

I2C总线的传送格式为主从式，对系统中的某一器件来说有四种工作方式：主发送方式、从发送方式、主接收方式、从接收方式。只讲主发送从接收（单片机发送 24C08接收）主器件产生开始信号以后，发送的第一个字节为控制字节。前七位为从器件的地址片选信号。最低位为数据传输方向位（高电平表示读从器件，低电平表示写从器件），然后发送一个选择从器件片内地址的字节，来决定开始读写数据的起始地址。接着再发送数据字节，可以是单字节数据，也可以是一组数据，由主器件来决定。从器件每接收到一个字节以后，都要返回一个应答信号（ASK=0）。主器件在应答时钟周期高电平期间释放SDA线，转由从器件控制，从器件在这个时钟周期的高电平期间必须拉低SDA线，并使之成为稳定的低电平，作为有效的应答信号。

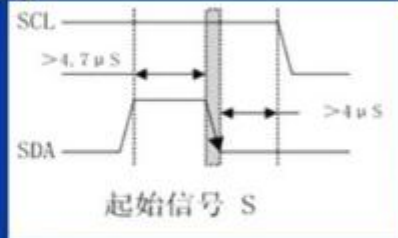
控制字节

I²C协议起始信号时序

起始信号用于开始I²C总线通信。其中，起始信号是在时钟线SCL为高电平期间，数据线SDA上高电平向低电平变化的下降沿信号。起始信号出现以后才可以进行后续的I²C总线寻址或数据传输等

```
*void start()// 起始位
{
    SDA = 1;    //SDA初始化为高电平“1”
    SCL = 1;    //开始数据传输时，要求SCL为高电平“1”
    _nop_();    //等待一个机器周期
    _nop_();    //等待一个机器周期
    _nop_();    //等待一个机器周期
    _nop_();    //等待一个机器周期
    _nop_();    //等待一个机器周期

    SDA = 0;    //SDA的下降沿被认为是起始信号
    _nop_();    //等待一个机器周期
    _nop_();    //等待一个机器周期
    _nop_();    //等待一个机器周期
    _nop_();    //等待一个机器周期
    _nop_();    //等待一个机器周期
    _nop_();    //等待一个机器周期
    SCL = 0;    //SCL为低电平时，SDA上数据才允许变化(即允许以后的数据传输)
}
```

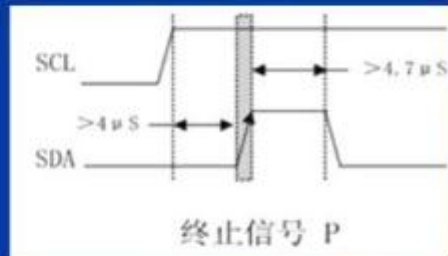


起始信号时序

I²C协议终止信号时序

- 终止信号用于终止I²C总线通信。其中，终止信号是在时钟线SCL为高电平期间，数据线SDA上低电平到高电平变化的上升沿信号。终止信号一出现，所有I²C总线操作都结束，并释放总线控制权。终止信号的时序，

```
*void stop()// 终止位
{
    SDA = 0;    //SDA初始化为低电平“0”
    SCL = 1;    //结束数据传输时，要求SCL为高电平“1”
    _nop_();    //等待一个机器周期
    _nop_();    //等待一个机器周期
    _nop_();    //等待一个机器周期
    _nop_();    //等待一个机器周期
    _nop_();    //等待一个机器周期
    SDA = 1;    //SDA的上升沿被认为是结束信号
    _nop_();    //等待一个机器周期
    _nop_();    //等待一个机器周期
    _nop_();    //等待一个机器周期
    _nop_();    //等待一个机器周期
    _nop_();    //等待一个机器周期
    _nop_();    //等待一个机器周期
    SDA=0;
    SCL=0;
}
```



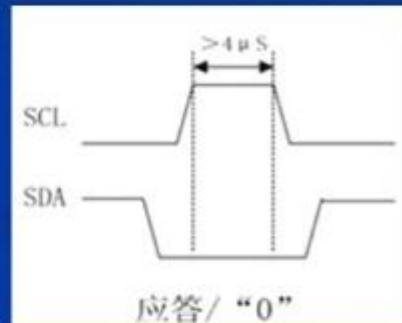
终止信号时序

I²C协议应答信号时序

- 应答信号用于表明I²C总线数据传输的结束。I²C总线数据传输时，一个字节数据传输完后都必须由主器件产生应答信号。主器件在第9个时钟位上释放数据总线SDA，使其处于高电平状态，此时从器件输出低电平拉低数据总线SDA为应答信号。应答信号的时序，如图所示。

```

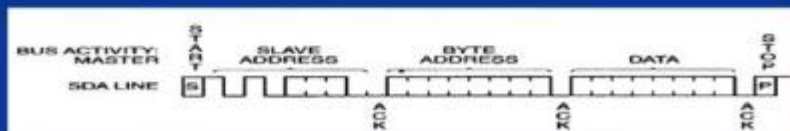
*bit Ask() //检测应答
*{
    bit ack_bit; //储存应答位
    SDA = 1; // 发送设备（主机）
    //应在时钟脉冲的高电平期间（SCL=1）释放SDA线。
    //以让SDA线转由接收设备（AT24Cxx）控制
    _nop_(); //等待一个机器周期
    _nop_(); //等待一个机器周期
    SCL = 1; //根据上述规定，SCL应为高电平
    _nop_(); //等待一个机器周期
    _nop_(); //等待一个机器周期
    _nop_(); //等待一个机器周期
    _nop_(); //等待一个机器周期
    _nop_(); //等待一个机器周期
    ack_bit = SDA; //接受设备（AT24Cxx）向SDA送低电平，
    //表示已经接收到一个字节
    //若送高电平，表示没有接收到，传送异常 结束发送
    SCL = 0; //SCL为低电平时，
    //SDA上数据才允许变化（即允许以后的数据传递）
    return ack_bit; // 返回AT24Cxx应答位
}
    
```



应答信号时序

写操作

- 写操作分为字节写和页面写两种操作。
- 在字节写模式下 主器件发送起始命令和从器件地址信息 R/#位置零 给从器件 在从器件产生应答信号后 主器件发送16的字节地址 主器件在收到从器件的另一个应答信号后 再发送数据到被寻址的存储单元 再次应答 并在主器件产生停止信号后
- 开始内部数据的擦写 在内部擦写过程中 从器件不再应答主器件的任何请求



```

*void WriteCurrent(unsigned char y)
*{
    start();
    unsigned char i;
    for(i = 0; i < 8; i++) // 循环写入8个位
    {
        SDA = (bit)(y << 80); //通过按位“与”运算将最高位数据送到S//因为传送时高位在前，低位在后
        _nop_(); //等待一个机器周期
        SCL = 1; //在SCL的上升沿将数据写入AT24Cxx
        _nop_(); //等待一个机器周期
        _nop_(); //等待一个机器周期
        SCL = 0; //将SCL重新置为低电平，以在SCL线形成传送数据所需的8个脉冲
        y <<= 1; //将y中的各二进制位向左移一位
    }
}
    
```