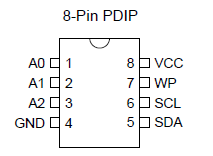
# 基于FPGA的I2C读写EEPROM

I2C在芯片的配置中应用还是很多的，比如摄像头、VGA转HDMI转换芯片，之前博主分享过一篇I2C协议的基础学习， ，这篇就使用Verilog来实现EEPROM的读写，进行一个简单的I2C实战应用。

## EEPROM

我使用的这个芯片是AT24C32，它手册上还有一种AT24C64，其实操作都是一样的，只是内存大小不同，AT24C32是32k（4096x8）AT24C64是64k（9=8192x8），



SCL设置为频率200Khz

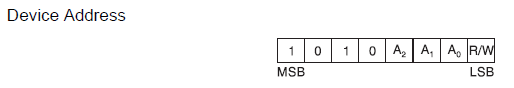
SCL clk posedge data输入EEPROM

SCL clk negedge data输出EEPROM

SDA 双向Pin

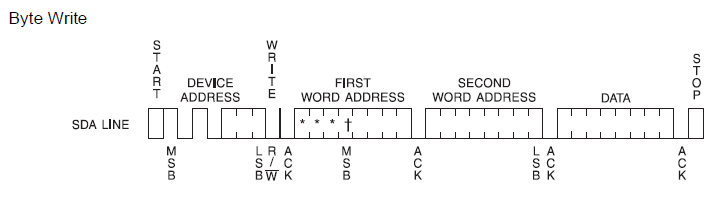
A2，A1，A0 Device Addr default all 0，只操作一片可悬空引脚。

WP 接地正常读写，WP接Vcc写操作被禁止

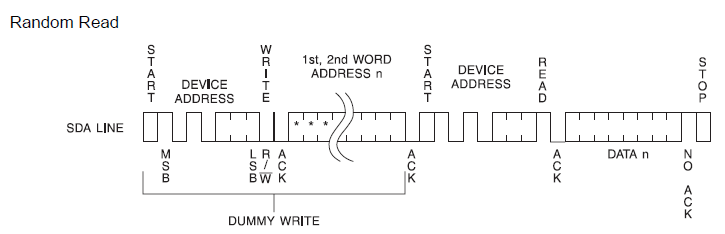


字节寻址地址，是由12（AT24C32）或13bit（AT24C64）的地址组成，需要操作16位字地址高3或4位忽略即可。

Device Address 8’hA0写器件地址，8’hA1读器件地址

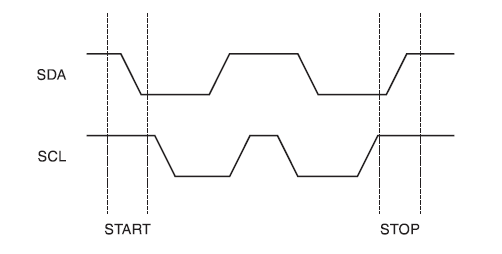


写字节操作



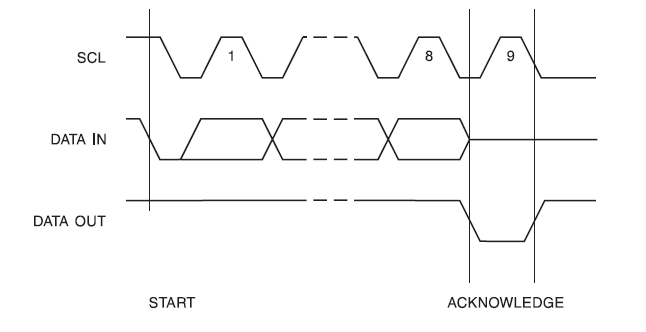
随机读字节操作

我这个芯片是双字节数据地址，所以在写数据地址时要写两次，先是高字节后是低字节。



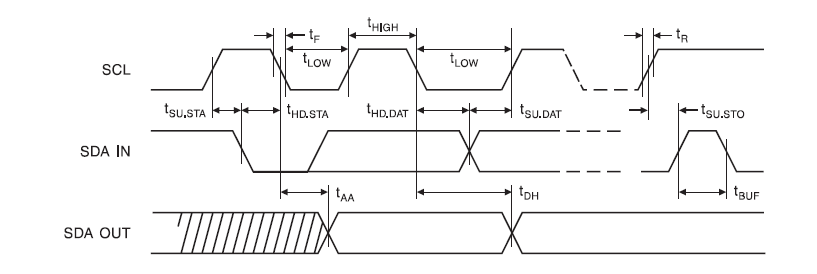
开始结束标志

这个I2C总线的时序是一致的。



EEPROM应答

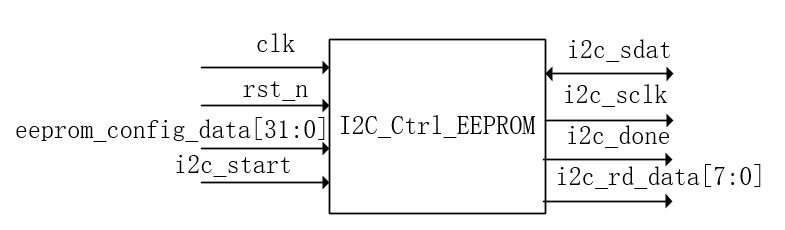
输出应答scl的第九个周期给出，低电平应答。如果主机没有收到应答，需要重新配置。



数据传输时序

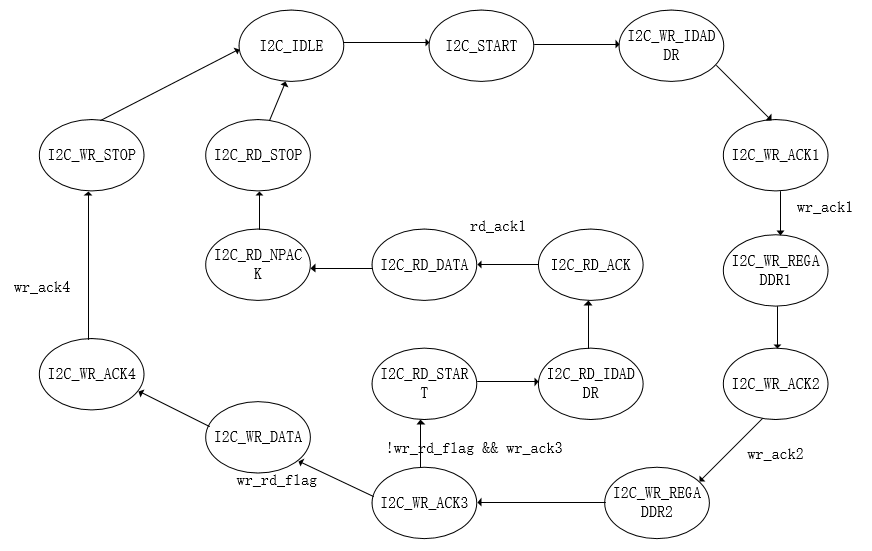
sda数据线在scl时钟的下降沿中间变化，可以避免产生误触开始结束标志。

## I2C Design



i2c\_start为高电平有效，传输完成后会产生一个i2c\_done结束标志，表示操作完成。

## I2C状态转移图



## I2C写操作

（1）产生start位

（2）传送器件地址ID\_Address，器件地址的最后一位为数据的传输方向位，R/W，低电平0表示主机往从机写数据（W），1表示主机从从机读数据（R）。这里按照手册给出的操作图，应该是W即低电平。ACK应答，应答是从机发送给主机的应答，这里不用管。

（3）传送写入器件寄存器地址，即数据要写入的位置。同样ACK应答不用管。

（4）传送要写入的数据。ACK应答不用管。

（5）产生stop信号。

## I2C读操作

（1）产生start信号

（2）传送器件地址（写ID\_Address），这里按照手册给出的操作图，最低位是W即低电平。ACK。

（3）传送字地址（写REG\_Address），ACK。

（4）再次产生start信号

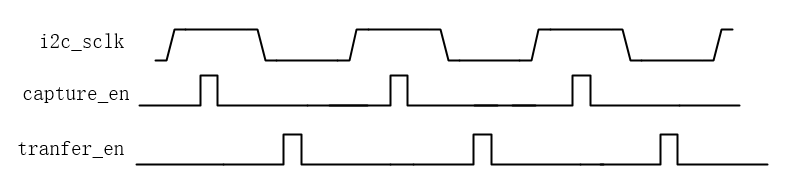
（5）再传送一次器件地址，这里根据手册最低位是读R高电平，ACK。

（6）读取一个字节的数据，读数据最后结束前无应答ACK信号。

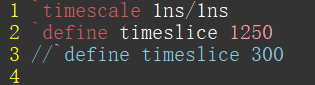
（7）产生stop信号。

读写操作的写器件地址和写数据地址操作是一样的，状态转移图中读写操作中这两部分复用了，根据读写标志来判断。

其他部分没啥好说的根据时序图写就行了，需要注意的一点是我们应该在sclk的高电平的中间采样数据，在sclk低电平的中间改变数据，当sclk为高电平的时候，sda为出现下降沿为start位， sda出现上升沿为stop位，所以在sclk为高电平的时候sda应该保持稳定不能随意乱动。这就又回到了数据传输有效的条件，只有在sclk为低电平期间，才允许数据变化，在高电平期间，不允许数据变化，否则就会出现起始位或结束位。



EEPROM有个仿真模型，在夏雨闻老师的书里面就有，这个模型默认是200khz的sclk驱动，仿真的时候可以将时间参数改小，我这里也分享出来。



仿真模型代码点击阅读原文可以查看。

根据仿真模型仿真的话基本不会有什么问题，需要注意的是操作的完成标志。从仿真上看到输入读写都没问题，但是stop标志没产生好，仿真看到读写操作没问题，但实际还是不行的，需要严格按照EEPROM的手册操作时序进行，差一点就不行。

I2C的代码我分享出来，我最后使用拨码开关作为读写使能，数码管显示读出来的输出，最后实现了对指定存储地址读写数据。

I2C设计代码点击阅读原文可以查看。