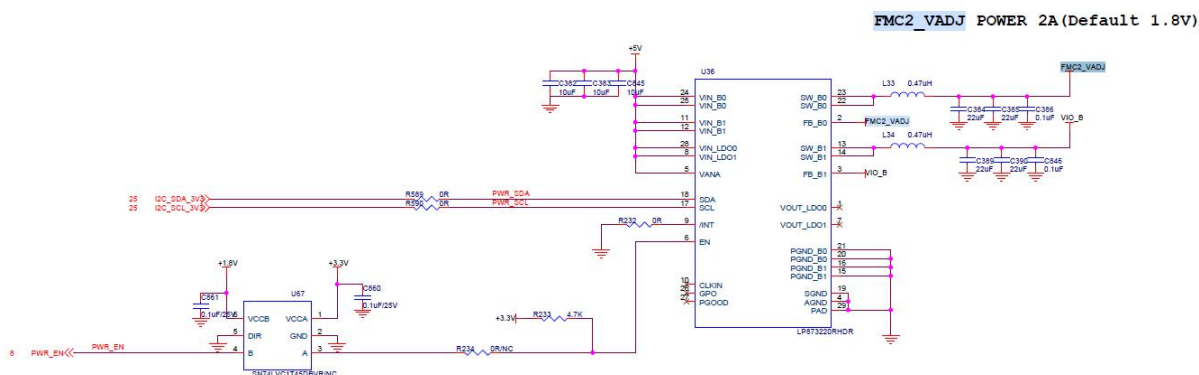


可调电源实验

1. 实验简介

AXKU040上有一片调节io电压的大小的芯片，通过调节内部寄存器的值可以更改输出的电压。开发板上 bank 64 65 默认配置的1.8v LP873220在原理图



2. 实验原理

这里我们对I2C协议做一个简单的了解后续章节我们再仔细讲解：

I2C物理连接

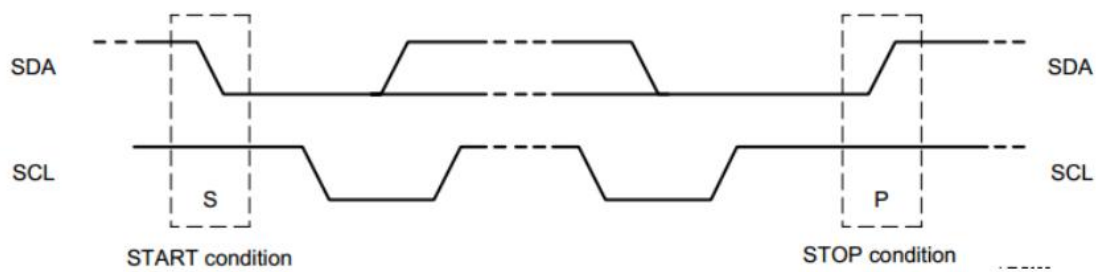
分别由SDA(串行数据线和SCL(串行时钟线及上拉电阻组成。通信原理是通过对SCL和SDA线高低电平时序的控制，来产生I2C总线协议所需要的信号进行数据的传递。在总线空闲状态时，这两根线一般被上面所接的上拉电阻拉高，保持着高电平。

起始条件S和停止条件P

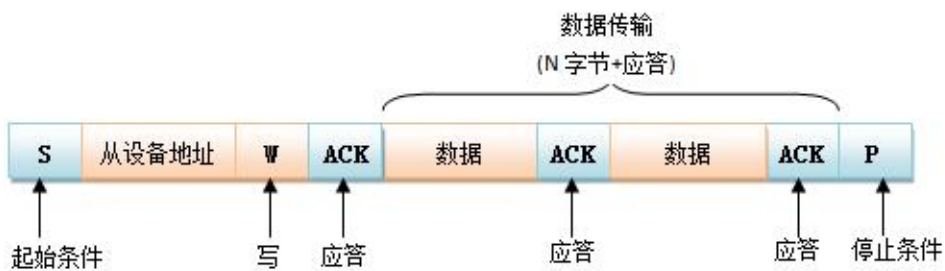
起始条件S：当SCL高电平时，SDA由高电平向低电平转换；

停止条件P：当SCL高电平时，SDA由低电平向高电平转换。

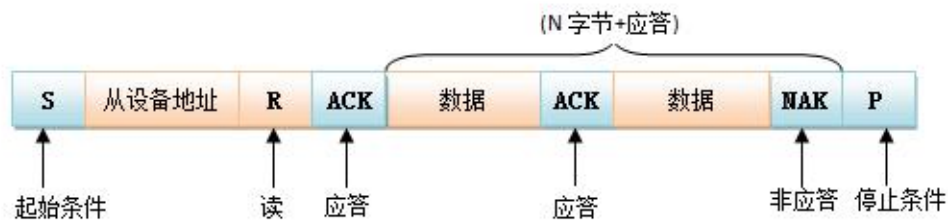
起始和停止条件一般由主机产生。总线在起始条件后处于busy的状态，在停止条件的某段时间后，总线才再次处于空闲状态。如图为起始和停止条件的信号产生时序图。



下图i2c读写的数据的过程



- 主设备从从设备中读数据。数据传输格式如下：



数据传输方式:

①在上面的图中我们可以知道，读写数据前要知道LP873220挂载到总线上的设备地址0xc2i2c总线上数据是按字节发送，字节发送顺序高位到低串行发送。这个时候大家可能会有疑惑再我们i2c总线上只有一跟SDA信号线进行数据传输，主机再读写从设备数据的时候怎么区分?在本程序前面7位固定设备地址位最后一位 0代表主机向从机发送数据 1代表主机读取从机数据。

②在发送完设备地址或者一字节数据，发送完设备地址。从设备需要在发送完最后一位时间的时候发送给主机一个应答信号（ack低电平）表示回应主机地址正确，接下来就可以发送数据每发完一字节数据都要回复ack表示接收到字节数据。

③在发送过程中i2c每发送一位数据都有一个对应的脉冲相对应（或同步控制），即在SCL串行时钟的配合下，在SDA逐位串行的发送每一位数据。在SCL的高电平期间 SDA电平必须保持稳定 在SCL低电平期间可以发送变化 高电平数据为1 低电平数据为0

3. 程序设计

程序设计非常简单一共分为三个模块 其中 reg_config.v产生配置lp873220的时钟和发送给从机的数据输出对应电压值，可以通过芯片数据手册看到下图对应电压的数值

```
//register of lp873220
always@(reg_index)
begin
    case(reg_index)
        0:reg_data<=16'h06fc; //BUCK0_VSET(addr=0x06), b2=1.8V, d4 =2.5V, fc=3.3V.
        1:reg_data<=16'h07fc; //BUCK1_VSET(addr=0x07), b2=1.8V, d4 =2.5V, fc=3.3V
        default:reg_data<=16'hffff;
    endcase
end
```

i2c_com就是我们的底层控制SDA总线发送数据给lp873220部分代码展示:

```
case(cyc_count)
0:begin ack1<=1;ack2<=1;tr_end<=0;sclk<=1;reg_sdat<=1;end
1:reg_sdat<=0; //开始传输
2:sclk<=0;
3:reg_sdat<=i2c_data[23];
4:reg_sdat<=i2c_data[22];
5:reg_sdat<=i2c_data[21];
6:reg_sdat<=i2c_data[20];
7:reg_sdat<=i2c_data[19];
8:reg_sdat<=i2c_data[18];
9:reg_sdat<=i2c_data[17];
10:reg_sdat<=i2c_data[16];
11:reg_sdat<=1; //应答信号
12:begin reg_sdat<=i2c_data[15];ack1<=i2c_sdat;end
13:reg_sdat<=i2c_data[14];
14:reg_sdat<=i2c_data[13];
15:reg_sdat<=i2c_data[12];
16:reg_sdat<=i2c_data[11];
17:reg_sdat<=i2c_data[10];
18:reg_sdat<=i2c_data[9];
19:reg_sdat<=i2c_data[8];
20:reg_sdat<=1; //应答信号
21:begin reg_sdat<=i2c_data[7];ack2<=i2c_sdat;end
22:reg_sdat<=i2c_data[6];
23:reg_sdat<=i2c_data[5];
24:reg_sdat<=i2c_data[4];
25:reg_sdat<=i2c_data[3];
26:reg_sdat<=i2c_data[2];
27:reg_sdat<=i2c_data[1];
28:reg_sdat<=i2c_data[0];
29:reg_sdat<=1; //应答信号
30:begin ack3<=i2c_sdat;sclk<=0;reg_sdat<=0;end
31:sclk<=1;
32:begin reg_sdat<=1;tr_end<=1;end //IIC传输结束
endcase
```

4. 实验结果

烧写好程序之后找到开发板上的u36编号 用万能表测我们2号引脚测得改变后的电压值

