

同济大学计算机系

数字逻辑课程综合实验报告



学 号 1750871

姓 名 葛超

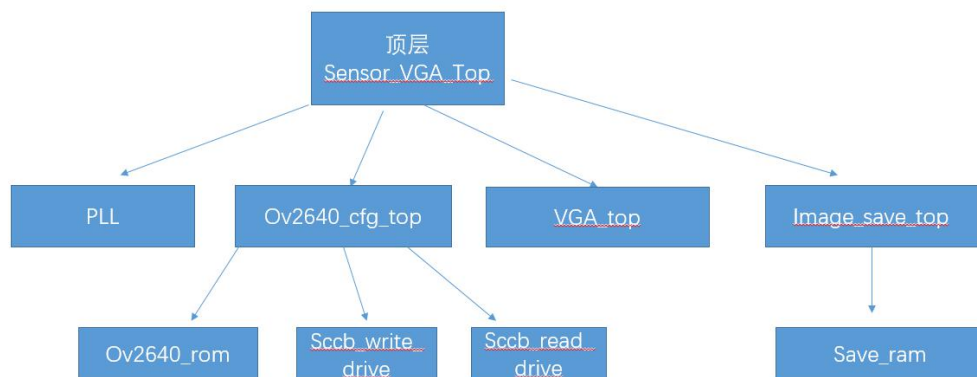
专 业 电子与信息工程

授课老师 郭玉臣

一、实验内容

Ov2640 摄像头

二、Ov2640 摄像头数字系统总框图



三、系统控制器设计

(1)含有状态机的模块内容

顶层状态机

```
localparam IDLE          = 3'b000,    //空闲
              S_DELAY     = 3'b001,    //等待
              S_INIT      = 3'b010,    //初始化
              INIT_SEND_WAIT = 3'b011, //初始化写入
              IIC_IDLE     = 3'b100,    //IIC 空闲
              IIC_WRITE    = 3'b101,    //IIC 写
              IIC_READ     = 3'b110,    //IIC 读
              IIC_SEND_WAIT = 3'b111;  //IIC 等待
```

```
localparam TIME_5ms      = 32'd500000;
```

写状态机

```
localparam I2C_CLK_DIV_Cnt = 10'd1000;
```

```
localparam IDLE = 3'b000,    //空闲
              START = 3'b001, //开始
              SEND  = 3'b010, //发送数据
              ACK   = 3'b011, //响应
              STOP  = 3'b100;
```

//因为 SCCB 内部需要 400K 时钟，所以需要分频。

```
reg [9:0] iic_clk_div;
```

读状态机

```
localparam IDLE = 3'b000,  
            START = 3'b001,  
            SEND = 3'b010,  
            ACK = 3'b011,  
            RS = 3'b100,  
            STOP = 3'b101;
```

(2) 逻辑表达式:

顶层模块:

```
assign Sensor_Driv_PWDN = 1'b0;  
assign rst_n = pll_locked;  
assign I2C_SDA = i2c_tri_en ? i2c_sda_out : 1'bz;  
assign i2c_sda_in = I2C_SDA;  
assign VGA_VS = vs_o;  
assign VGA_HS = hs_o;  
assign VGA_R = vga_r;  
assign VGA_G = vga_g;  
assign VGA_B = vga_b;  
ov2640_cfg  
assign scl = (state_sel == 3'b011) ? iic_read_scl : iic_write_scl;  
assign sda_out = (state_sel == 3'b011) ? iic_read_sda_out : iic_write_sda_out;  
assign tri_en = (state_sel == 3'b011) ? iic_read_tri_en : iic_write_tri_en;  
assign iic_write_sda_in = (state_sel == 3'b011) ? 1'b1 : sda_in;  
assign iic_read_sda_in = (state_sel == 3'b011) ? sda_in : 1'b1;  
assign conf_ae_flag = (iic_busy == 1'b0) && (iic_busy_d1 == 1'b1);  
VGA_top  
assign hs_o = (hcount < `HSYNC_A) ? 0 : 1;  
assign vs_o = (vcount < `VSYNC_A) ? 0 : 1;  
assign VGA_data_ena_r = ((hcount >= `HSYNC_A + `HSYNC_B && hcount < `HSYNC_A +  
`HSYNC_B + `HSYNC_C) && (vcount >= `VSYNC_A + `VSYNC_B && vcount < `VSYNC_A  
+ `VSYNC_B + `VSYNC_C)) ? 1:0;
```

四、子系统模块建模

(该部分要求对实验中的所有子系统模块进行描述, 给出各子系统的功能框图及接口信号定义, 并列出各模块建模的 verilog 代码)

(1) PLL



由 FPGA 自带的 100M 时钟，分频出所需要的时钟，包括 SCCB 传输协议所需的 100M 时钟，Sensor 所需要的 24M 时钟，640*480 分辨率 VGA 所需要的 31.5M 时钟，并且把锁相环分频结束后的 lock 信号作为初始化信号。

In_1:100M 时钟 out_1:31.5M out_2:100M out_3:24M

```

module clk_pll
(
  // Clock in ports
  input      clk_in1,
  // Clock out ports
  output     clk_out1,
  output     clk_out2,
  output     clk_out3,
  output     clk_out4,
  // Status and control signals
  input      reset,
  output     locked
);
  
```

(2)存储上层模块

复位信号，摄像头传入时钟，摄像头行场信号，摄像头传入的数据，vga 时钟，vga 读取地址，输出到 vga 的 12bit 数据。

```

module Image_save_Top (
  input  rst_n,
  input  sensor_clk,
  input  sensor_vs,
  input  sensor_hs,
  input  [7:0] sensor_data,

  input  vga_clk,
  input  [31:0] vga_read_addr,
  output [11:0] vga_read_data
);
  
```



(3)图像存储---save_ram

用来存储 Sensor 发送过来的数据，ov2640 发送回来的是 RGB565 格式的数据，由于每次传输回来时 8bit，所以要进行拼接，拼接出来的 16bit 数据 data 中的[15:12]对应 r，[10:7]对应 g，[4:1]对应 b。以 12bit 数据格式存储起来，每存储够 1 帧，vga 读取显示。由于 vga 分辨率是 640*480 所以需要空间大小为 12*640*480。所需地址由上层 save_top 传入。

```
module save_ram (
    input    clk_a,
    input    [31:0] addr_a,
    input    [15:0] data_a,

    input    clk_b,
    input    [31:0] addr_b,
    output   [11:0] data_b
);
    reg [11:0] Sensor_ram [0:307199];
    always @(posedge clk_a) begin
        Sensor_ram[addr_a] <= {data_a[15:12],data_a[10:7],data_a[4:1]};
    end
    reg [11:0] dout_r;
    always @(posedge clk_b) begin
        dout_r <= Sensor_ram[addr_b];
    end
    assign    data_b = dout_r;
endmodule
```

(4).VGA 显示

显示器扫描方式分为逐行扫描和隔行扫描：逐行扫描是扫描从屏幕左上角一点开始，从左像右逐点扫描，每扫描完一行，电子束回到屏幕的左边下一行的起始位置，在这期间，CRT 对电子束进行消隐，每行结束时，用行同步信号进行同步；当扫描完所有的行，形成一帧，用场同步信号进行场同步，

并使扫描回到屏幕左上方，同时进行场消隐,开始下一帧。隔行扫描是指电子束扫描时每隔一行扫一线，完成一屏后在返回来扫描剩下的线。

完成一行扫描的时间称为水平扫描时间,其倒数称为行频率;完成一帧(整屏)扫描的时间称为垂直扫描时间,其倒数称为场频率,即刷新一屏的频率,常见的有 60Hz, 75Hz 等等。标准的 VGA 显示的场频 60Hz, 行频 31.5KHz。

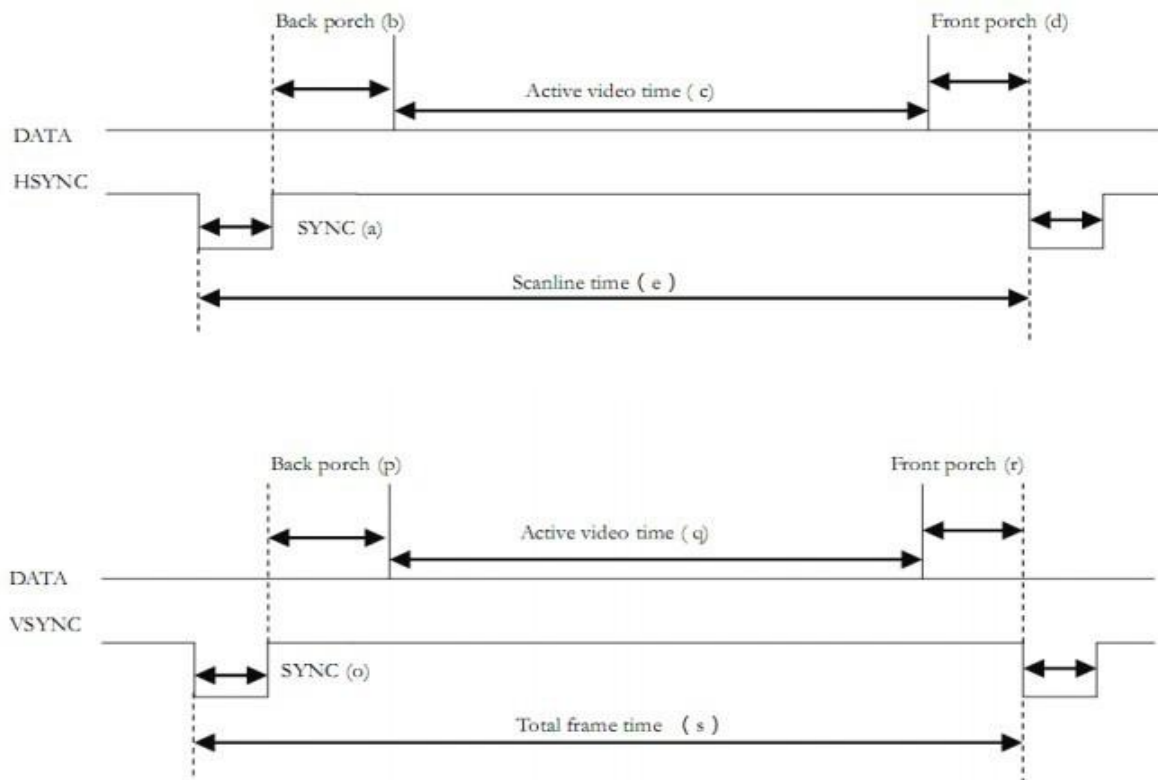
行场消隐信号:是针对老式显像管的成像扫描电路而言的。电子枪所发出的电子束从屏幕的左上角开始向右扫描，一行扫完需将电子束从右边移回到左边以便扫描第二行。在移动期间就必须有一个信号加到电路,使得电子束不能发出。不然这个回扫线会破坏屏幕图像的。这个阻止回扫线产生的信号就叫作消隐信号,场信号的消隐也是一个道理。

显示带宽:带宽指的显示器可以处理的频率范围。如果是 60Hz 刷新频率的 VGA, 其带宽达 $640 \times 480 \times 60 = 18.4\text{MHz}$, 70Hz 的刷新频率 1024x768 分辨率的 SVGA, 其带宽达 $1024 \times 768 \times 70 = 55.1\text{MHz}$ 。

时钟频率: 以 640x480@59.94Hz (60Hz) 为例，每场对应 525 个行周期 ($525 = 10 + 2 + 480 + 33$), 其中 480 为显示行。每场有场同步信号, 该脉冲宽度为 2 个行周期的负脉冲, 每显示行包括 800 点时钟, 其中 640 点为有效显示区, 每一行有一个行同步信号, 该脉冲宽度为 96 个点时钟。由此可知: 行频为 $525 \times 59.94 = 31469\text{Hz}$, 需要点时钟频率: $525 \times 800 \times 59.94$ 约 25MHz。

VGA 时序分析:





输入时钟，复位信号，12bit 数据，32 位地址，输出的行场信号，输出的 rgb 数据。

```

`define HSYNC_A    16'd40
`define HSYNC_B    16'd128
`define HSYNC_C    16'd640
`define HSYNC_D    16'd24
`define HSYNC_ACT   `HSYNC_C

//-- Vertical timing information
`define VSYNC_A    16'd2
`define VSYNC_B    16'd29
`define VSYNC_C    16'd480
`define VSYNC_D    16'd9
`define VSYNC_ACT   `VSYNC_C

```

```

module VGA_Top (

    input  clk,

    input  rst_n,

    input  [11:0] ram_din,

    output [31:0] ram_addr,

    output hs_o,

    output vs_o,

    output [4-1:0] vga_r,

    output [4-1:0] vga_g,

    output [4-1:0] vga_b

);

```

(5)SCCB 顶层

```

module ov2640_cfg_top #(
    // parameter DVC_ID = 8'H20
    //从机 id
    parameter id_w = 'd8,
    //寄存器地址 16bit
    parameter addr_w = 'd16,
    //写入寄存器数据 8bit
    parameter data_w = 'd8,
    parameter byte_num = 3'd5
)
(
    //--Sys Ctrl
    input clk,
    input rst_n,
    //从机 ID
    input [7:0] DVC_ID,
    //--I2C Bus
    output scl,
    output tri_en,
    //输入数据
    input sda_in,

```



```

//输出数据
output sda_out,
//--I2C wr & rd

//可写标志
input iic_write_en, //pulse
input [data_w-1:0] iic_write_data,
input [addr_w-1:0] iic_write_addr,
input iic_read_en, //pulse
input [addr_w-1:0] iic_read_addr,
output [data_w-1:0] iic_read_data,
//--I2C wr & rd ? ? ?
input [15:0] expo_add,
input [15:0] analog_gain,
input [15:0] digital_gain,
//摄像头 初始化完成标志
output wire sensor_ini_done,
//读操作完成
output wire iic_read_done_pulse,
output conf_ae_flag,
output reg sensor_rst
);
//状态机
localparam IDLE = 3'b000, //空闲
S_DELAY = 3'b001, //等待
S_INIT = 3'b010, //初始化
INIT_SEND_WAIT = 3'b011, //初始化写入
IIC_IDLE = 3'b100, //IIC 空闲
IIC_WRITE = 3'b101, //IIC 写
IIC_READ = 3'b110, //IIC 读
IIC_SEND_WAIT = 3'b111; //IIC 等待

localparam TIME_5ms = 32'd500000;

```

(6)SCCB 协议

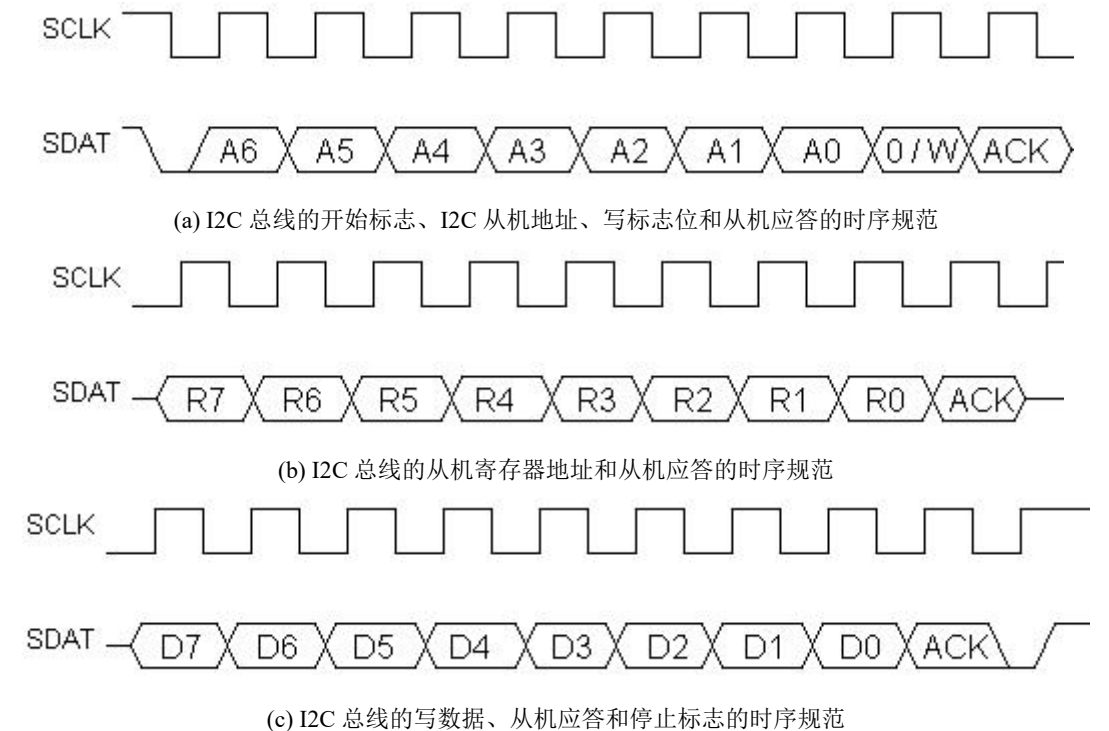
I2C 总线简介

I2C(Inter-Integrated Circuit Bus)总线，是由 Philip 公司制定的一种串行总线，属于双向两线制同步总线，广泛应用在 IC 之间的通信互联^[11]。总线规定空闲时，数据线 SDAT 和时钟线 SCLK 都处于高电平状态，因此在硬件设计上，SCLK 和 SDAT 必须加上一个上拉电阻，保持在高电平状态。在 I2C 总线协议中，开始标志和停止标志的要求如下：在 SCLK 高电平期间，SDAT 由高电平变为低电平，表示 I2C 总线协议的开始；在 SCLK 处于高电平期间，SDAT 由低电平变为高电平，表示 I2C 总线协议的停止^[12,13]。除了开始标志和停止标志，I2C 总线协议还

规定数据传输的要求，要求在数据传输的过程中，当 SCLK 处于高电平时 SDAT 上的数据必须不能改变，在 SCLK 为低电平时 SDAT 上的数据可改变。每传输 8 位数据，必须有应答信号进行响应，应答信号既可能由主机产生，也可能由从机产生，应答信号对本次数据传输进行响应，表示本次传输成功或失败，以便于后续数据的传输^[14,15]。下面通过一次完整的写操作过程说明，和一次完整的读操作过程说明，了解一下 I2C 总线的读写时序要求，并对 I2C 总线协议进行详细的介绍。

I2C 写操作

I2C 总线协议的写操作时序，I2C 总线进行一次写操作的完整时序图如图 4-3 所示，SCLK 表示时钟，SDAT 表示数据。



I2C 总线写操作的详细通信过程如下：

- 1) 空闲时：时钟信号 SCLK 和数据信号 SDAT 都处于高电平状态。
- 2) 主机发送开始标志：在时钟信号 SCLK 处于高电平时，主机控制数据信号 SDAT 由高电平变为低电平，产生一个下降沿，表示 I2C 总线的开始，标志着 I2C 总线通信的开始。I2C 总线协议采用主从模式通信，一个主机对应一个从机或多个，每个从机都有一个独立的地址，主机通过地址寻址的方式和从机建立联系。
- 3) 主机发送从机地址和写操作标志位：在产生开始标志后，需要先发送从机的地址进行寻址，从机地址是 7 位的，在发送从机地址的同时，最后 1 位是 I2C 总线的读写标志位，1 表示读操作，0 表示写操作。此时主机控制数据信号，在

连续 8 个 SCLK 低电平期间，依次输出从机的地址 A6~A0 和写操作标志信号，前 7 位表示从机地址，第 8 位为 0 表示主机要对从机进行写操作。

4) 从机应答：主机发送过从机地址和写操作标志位后，需要从机进行一个应答，才能进行后续的数据传输，此时主机释放数据总线，由于上拉电阻的存在，此时 SDAT 为高电平，然后等待从机的应答，如果从机能够正常工作，它将会发送低电平的应答信号 ACK，此时数据总线 SDAT 将会被拉低，然后在下一个 SCLK 的高电平时，主机对 SDAT 进行检测，SDAT 若为低电平，则表示从机应答成功，主机将会继续发送后续数据；SDAT 若为高电平，则表示从机应答失败或从机不存在，则返回空闲转态。

5) 主机发送寄存器地址：I2C 从机的内部有多个寄存器，先对寄存器进行寻址，然后再对寄存器进行读写操作。在从机正确应答后，此时主机控制数据信号，在连续 8 个 SCLK 低电平期间，依次输出从机寄存器的地址信号 R7~R0。

6) 从机对寄存器寻址应答：此时主机释放数据总线，由于上拉电阻的存在，此时 SDAT 为高电平，然后等待从机的应答，若从机内部寄存器存在这个地址，则对寄存器寻址进行正确的应答，输出低电平把 SDAT 拉低，否则不应答，SDAT 仍为高电平。然后在下一个 SCLK 的高电平时，主机对 SDAT 进行检测，SDAT 若为低电平，则表示从机对寄存器寻址应答成功，主机将会继续发送后续数据；SDAT 若为高电平，则表示从机对寄存器寻址应答失败，则返回空闲转态。

7) 主机发送数据：从机对寄存器寻址应答后，主机接着发送要写入从机寄存器里的数据，在从机正确应答后，此时主机控制数据信号，在连续 8 个 SCLK 低电平期间，依次输出数据信号 D7~D0，高位在前，然后在时钟的上升沿，把要写入的数据发送给从机，从机会自动接收，并存入之前应答过的寄存器里。主机发送完 8 位数据后，释放数据总线让 SDAT 拉高。

8) 从机数据应答：从机接收完 8 位数据后返回应答信号，输出低电平把 SDAT 拉低。然后在下一个 SCLK 的高电平时，主机对 SDAT 进行检测，SDAT 若为低电平，则表示从机接收数据成功，主机将会继续后续操作；SDAT 若为高电平，则表示从机接收数据失败。

9) 主机发送停止标志：接收到从机应答信号后，在时钟信号 SCLK 为高电平时，主机控制数据信号 SDAT 由低电平拉高为高电平，产生停止标志信号，表示 I2C 总线通信停止。一次写操作顺利完成，同时时钟总线 and 数据总线都回到空闲时的高电平状态，等待下一次操作命令。

这是一次完整的 I2C 总线写操作通信过程，本过程中地址位和数据位都是 8 位，在步骤 5 和步骤 7 都是传输 8 位的数据，步骤 5 传输的是地址，步骤 7 传输的是数据。还有地址和数据是 16 位的情况，那么步骤 5 连续传输两次，一次传

输地址的高 8 位，一次传输地址的低 8 位，即可完成 16 位地址的传输，同理步骤 7 也进行两次，即可完成 16 位数据的传输。此外还有地址是 16 位而数据是 8 位的情况和地址是 8 位而数据是 16 位的情况，则步骤 5 和步骤 7 进行类似的处理，即可实现不同地址位不同数据位的 I2C 总线的通信。

```
module iic_write_drive_ov2640 #(
    //上层传入
    parameter data_w = 'd40,
    parameter byte_num = 3'd5
)
(
    input clk,
    input rst_n,
    //写请求信号
    input iic_send,

    input [data_w-1:0] send_data,
    //???输出时钟
    output reg scl,
    output reg tri_en,
    input wire sda_in,
    //接收数据是否成功
    output reg sda_out,
    output reg iic_busy
);
```

状态机

```
localparam I2C_CLK_DIV_Cnt = 10'd1000;
localparam IDLE = 3'b000, //空闲
          START = 3'b001, //开始
          SEND = 3'b010, //发送数据
          ACK = 3'b011, //响应
          STOP = 3'b100;

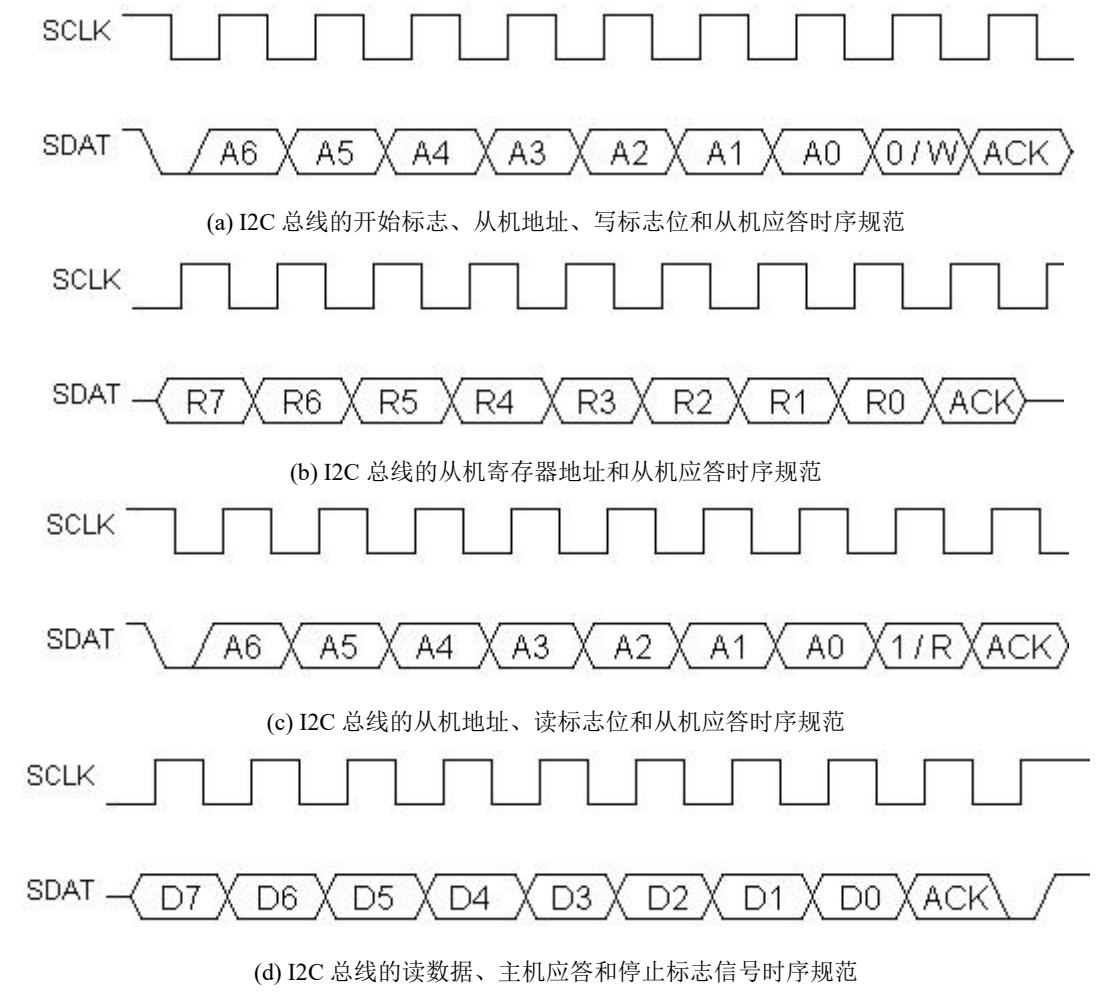
//因为 SCCB 内部需要 400K 时钟，所以需要分频。
reg [9:0] iic_clk_div;
```

由于需要判别每次传输的数据是地址还是数据所以还需要在内部定义小状

态进行判别。

I2C 读操作

I2C 通信协议的读操作时序，I2C 总线进行一次读操作的完整时序图如图 4-2 所示，SCLK 表示时钟，SDAT 表示数据。



I2C 总线读操作的详细通信过程如下：

- 1) 空闲时：时钟信号 SCLK 和数据信号 SDAT 都处于高电平。
- 2) 主机发送开始标志：在时钟信号 SCLK 处于高电平时，主机控制数据信号 SDAT 由高电平变为低电平，产生一个下降沿，表示 I2C 总线的开始，标志着 I2C 总线通信的开始。I2C 总线协议采用主从模式通信，一个主机对应一个从机或多个，每个从机都有一个独立的地址，主机通过地址寻址的方式和从机建立联系。
- 3) 主机发送从机地址和写信号：在产生开始标志后，需要先发送从机的地址进行寻址，从机地址是 7 位的，在发送从机地址的同时，最后 1 位是 I2C 总线的读写标志位，1 表示读操作，0 表示写操作。此时主机控制数据信号，在连续

8 个 SCLK 低电平期间，依次输出从机的地址 A6~A0 和写操作标志信号，前 7 位表示从机地址，第 8 位为 0 表示主机要对从机进行写操作。读操作开始前，需要先进行写，写入需要进行读操作的寄存器的地址。

4) 从机对从机寻址应答：主机发送过从机地址和写操作标志位后，需要从机进行一个应答，才能进行后续的数据传输，此时主机释放数据总线，由于上拉电阻的存在，此时 SDAT 为高电平，然后等待从机的应答，如果从机能够正常工作，它将会发送低电平的应答信号 ACK，此时数据总线 SDAT 将会被拉低，然后在下一个 SCLK 的高电平时，主机对 SDAT 进行检测，SDAT 若为低电平，则表示从机应答成功，主机将会继续发送后续数据；SDAT 若为高电平，则表示从机应答失败或从机不存在，则返回空闲转态。

5) 主机发送寄存器地址：I2C 从机的内部有多个寄存器，先对寄存器进行寻址，然后再对寄存器进行读写操作。在从机正确应答后，此时主机控制数据信号，在连续 8 个 SCLK 低电平期间，依次输出从机寄存器的地址信号 R7~R0。

6) 从机对寄存器寻址应答：此时主机释放数据总线，由于上拉电阻的存在，此时 SDAT 为高电平，然后等待从机的应答，若从机内部寄存器存在这个地址，则对寄存器寻址进行正确的应答，输出低电平把 SDAT 拉低；否则不应答，SDAT 仍为高电平。然后在下一个 SCLK 的高电平时，主机对 SDAT 进行检测，SDAT 若为低电平，则表示从机对寄存器寻址应答成功，主机将会继续发送后续数据；SDAT 若为高电平，则表示从机对寄存器寻址应答失败，则返回空闲转态。

7) 主机再次发送开始标志：从机对寄存器寻址正确应答后，主机必须要再次发出开始标志，在 SCLK 高电平时，SDAT 产生下降沿，然后对被寻址的寄存器发送读操作命令。

8) 主机再次发送从机地址和读命令：重复步骤 3 的过程，再次对从机进行地址寻址，但是第 8 位必须是 1 表示读命令，从机对读操作命令应答后，将输出被步骤 5 寻址的寄存器里的数据。

9) 从机再次对从机寻址应答：整个过程和步骤 4 类似。

10) 从机发送数据：步骤 8 发出从机地址和读操作命令，从机应答后将输出之前被寻址过的寄存器的数据 D7~D0，并在时钟信号 SCLK 的控制下将数据输出给主机，以便主机对数据进行采样接收，从机发送完 8 位数据后将释放数据线 SDAT，等待主机的应答，此时是主机发送应答信号。

11) 主机发送应答位：主机接收完 8 位数据后，控制数据线 SDAT 拉低，产生低电平应答信号 ACK，从机会在下一个时钟的上升沿检测数据线，判断主机是否应答，从而决定后续的操作。

12) 主机发送停止标志：本次过程值读取 8 位的数据，主机对接收数据应答

后，需要产生停止信号来终止本次读操作。在时钟信号 SCLK 为高电平时，主机控制数据信号 SDAT 由低电平变为高电平，表示 I2C 协议停止通信。一次完整的读操作顺利完成，同时时钟总线 and 数据总线都回到空闲时的高电平状态，等待下一次操作命令。

这是一次完整的 I2C 总线读操作通信过程，本过程中地址位和数据位都是 8 位，此外还有 16 位地址和 16 位数据的通信过程，区别就在发送地址的步骤和接收数据的步骤，发送地址时连续发送两次即可，接收数据时，当接收了 8 位数据并且主机应答后，不发送停止信号，从机就会继续从当前地址的下一个地址输出数据。

```
module iic_read_driver_ov2640 #(
    parameter    id_w = 'd8,
    parameter    addr_w = 'd16,
    parameter    data_w = 'd8,
    parameter    byte_num = 3'd5
)
(
    input clk,
    input rst_n,
    output reg scl,
    output reg tri_en,
    input  wire sda_in,
    output reg sda_out,
    input  iic_send,
    input  wire [id_w+addr_w+id_w-1:0] send_data,
    output wire [data_w-1:0] receive_data,
    output reg iic_read_com
);
localparam I2C_CLK_DIV_Cnt = 10'd1000;
localparam    IDLE    = 3'b000,
               START = 3'b001,
               SEND   = 3'b010,
               ACK     = 3'b011,
               RS      = 3'b100,
               STOP    = 3'b101;
```

```

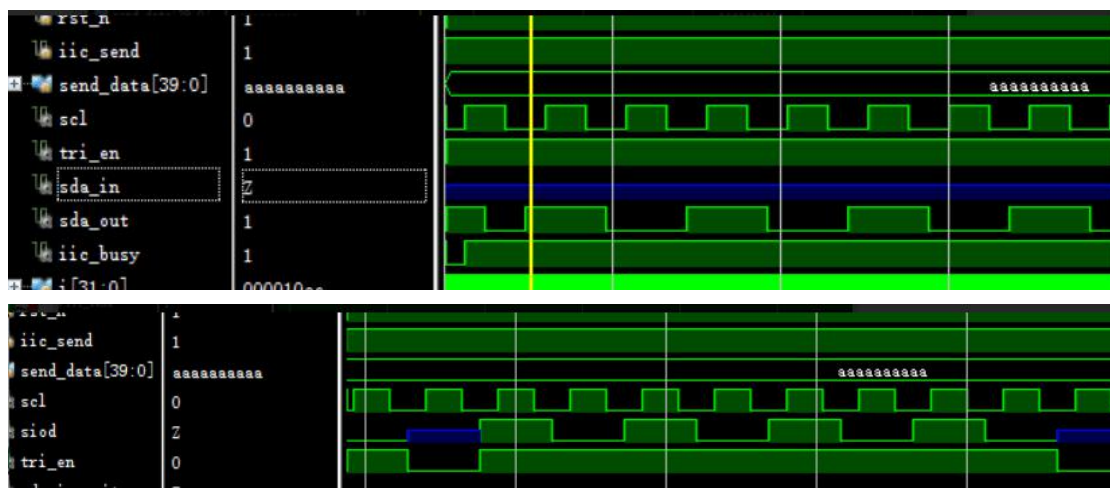
reg [2:0] byte_num_rs;
always @(*) begin
    if (addr_w == 'd16)
        byte_num_rs <= 3'd3;
    else if (addr_w == 'd8)
        byte_num_rs <= 3'd2;
end

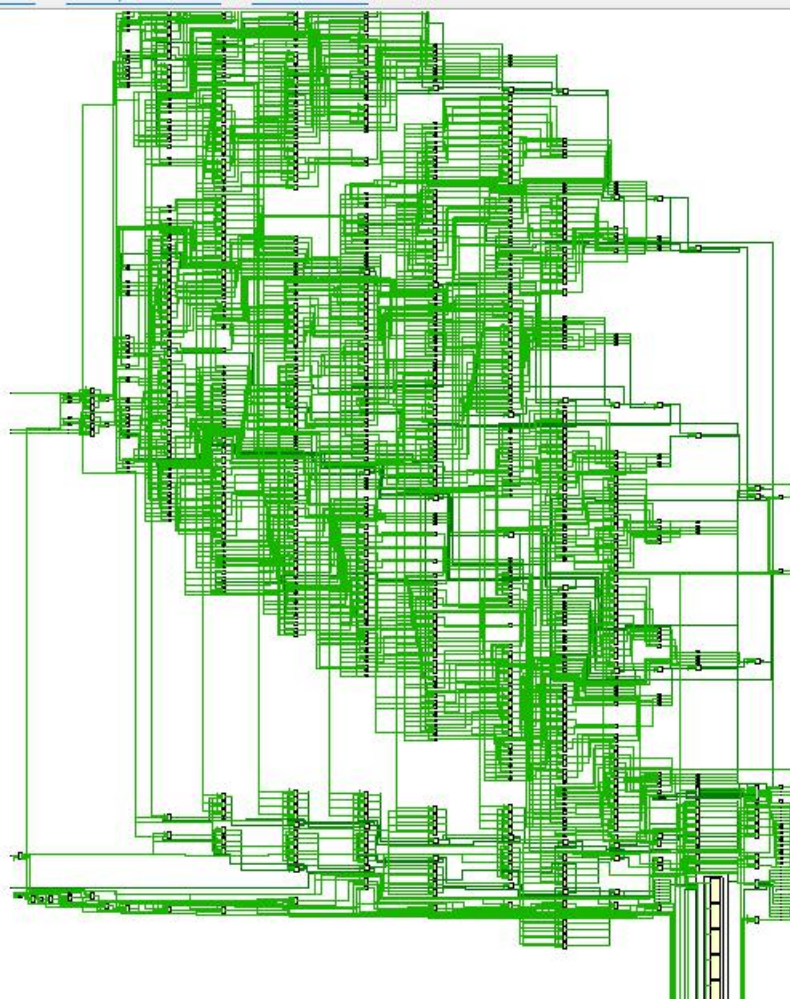
```

五、测试模块建模

无 TEST_BEACH 测试代码，主要都是直接从屏幕上得到反馈，进行修改。

六、实验结果







七、结论

八、心得体会及建议