数电实验EXP9

刘永鹏 191220070 693901492@qq.com 11.4

1. 实验内容

1. 在上述 VGA 控制器中，根据扫描的行或列数据，输出两种以上的不同颜 色条纹（横条或竖条均可以）
2. 利用上述控制器，在显示器上显示一张静态图片。 我们建议可以使用低比特的颜色显示的方式来绕过 RAM 不足的问题。当 然有兴趣的同学可以通过其他方式来实现高分辨率的图像显示。
3. 显示一张自定义的图片，自行完成图片格式到 mif 文件的转换。如有余力，可以显示一张在屏幕上按特定速度移动的图片。即图片本身大小远小于显 示器分辨率，例如 100 × 100 像素大小。图片随时钟按特定方向以随机速度（x 方向和 y 方向速度可不同）在屏幕内移动，当图片边界触及屏幕边界时按弹性 碰撞方式改变运动方向。最终效果类似弹球游戏，图片在屏幕内不停反弹。

2. 实验原理

1. VGA的工作原理：实验pdf上讲得蛮清楚，就不复制粘贴了。
2. 本次实验绝大部分代码都已经给出，因此自己实现的部分实际上比较简单，在此不再赘述。

3. 实验环境

本实验显然不适合仿真调试，因此直接上机实验。

4. 实验代码/实验截图

首先，对于顶层实体，要调用以下几个模块：

```
clkgen #(25000000) my_vgaclk(CLOCK_50,1'b0,1'b1,VGA_CLK);

vga_ctrl ctrl(.pclk(VGA_CLK),.vga_r(VGA_R),.vga_g(VGA_G),.vga_b(VGA_B),
    .valid(VGA_BLANK_N),.vsync(VGA_VS),.hsync(VGA_HS),.v_addr(v_addr),
    .h_addr(h_addr),.reset(reset),.vga_data(vga_data));

picdisplay
sd(.h_addr(h_addr),.v_addr(v_addr),.vga_data(vga_data),.clk(CLOCK_50),.en(1)
);

```

其中clkgen，vga_ctrl都已给出，理解代码直接调用即可。

而picdisplay处理显示内容。在条纹部分，它负责显示条纹，而在图片显示部分，它负责显示图片。这里采用**4bits color**来解决显存大小不够的问题。

在扩展功能：debounce部分中，调用情况有所不同：

```
top_flyinglogo tfl(.clk(CLOCK_50), .rst(0), .hsync(VGA_HS), .vsync(VGA_VS),
                        .vga_r(VGA_R[7:4]), .vga_g(VGA_G[7:4]),
.vga_b(VGA_B[7:4])
                        ,.pclk(VGA_CLK),.valid(VGA_BLANK_N));
```

这里直接调用**top_flyinglogo**，但对其进行一些修改：将pclk, valid作为输出，这样才能与开发板上的VGA芯片接口相连。其他的修改在下面给出。

接下来是picdisplay.v的部分：

```
always @ (h_addr or v_addr) //条纹
begin
    if(h_addr > 300 && h_addr < 330)
    begin
        vga_data = 24'h103050; // 某种蓝色
    end
    else vga_data = 24'hf0f0f0; //白色
end
```

```
pic mypic(addr, clk, data);//IPCatalog中生成的时序控制mif显存，并载入图片mif文件

always @ (h_addr or v_addr) //图片
begin
    if(en)
    begin
    addr = {h_addr, v_addr[8:0]};
    vga_data = {data[11:8], 4'b0000, data[7:4], 4'b0000, data[3:0], 4'b0000};
    end
end
```

下面是显示本人室友奇怪照片的实验结果。

对于flylogo部分，作如下修改后即可以直接使用：

```
   output            pclk;
   output            valid;//改为输出
......
   parameter [9:0] logo_length = 10'd100;
   parameter [9:0] logo_hight  = 10'd100;//修改logo大小为100*100
......
clkgen #(25000000) u0
         (
         // Clock in ports
          .clkin(clk),      // input clk_in1
          // Clock out ports
          .clkout(pclk),     // output clk_out1
          // Status and control signals
```

```
          .rst(rst),
             .clken(1'b1));

    pic u1 (
          .clock(pclk),    // input wire clka
          .address(rom_addr),  // input wire [13 : 0] addra
          .q(douta)  // output wire [11 : 0] douta
    );//这两段修改为自己的模块端口
```

其他部分不变，即可直接调用。实验结果在压缩包中有一段10s的视频。

5. 实验过程

代码编写+上机调试。

6. 实验结果

已验收，已完成所有扩展功能。

7. 问题与解决方案

1. 本次实验主要在于代码理解，自己编写的部分事实上并不太多。
2. 主要的困难来源于代码理解，在最后调用flyinglogo时由于忽略了VGA_CLK的输出而卡了一段时间，好在最终得以解决。
3. 使用matlab，利用给出的代码生成图片对应的mif文件。更改相应参数后也可以生成100*100的文件。