数字电路与数字系统

实验报告

实　　验:　　VGA 接口控制器实现

姓　名:　　周心同

学　　号:　　201220069

目录

1.实验目的

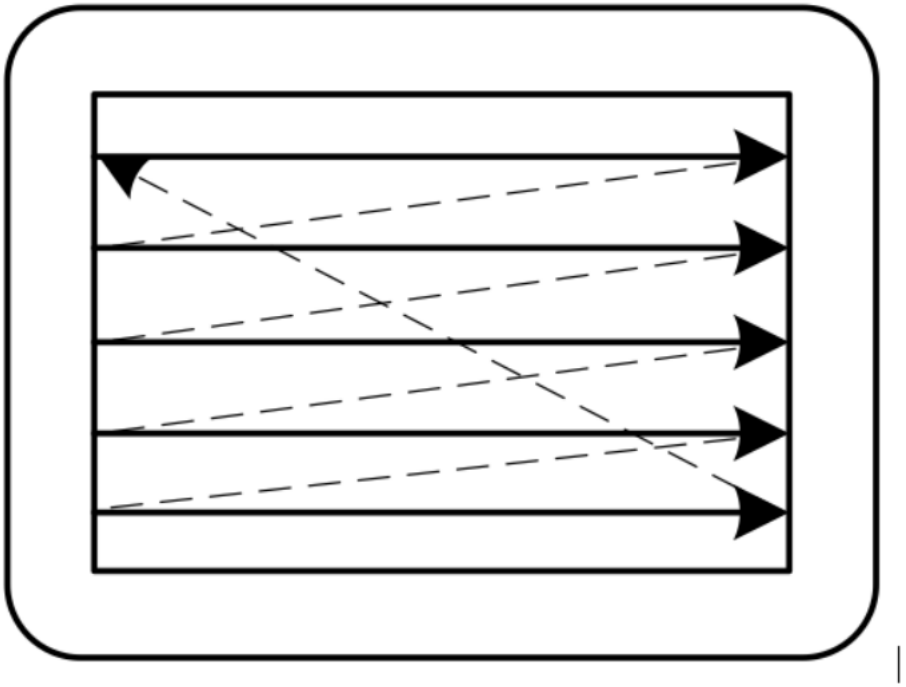
学习 VGA 接口原理

学习 VGA 接口控制器的设计方法

2.实验原理

2.1 VGA

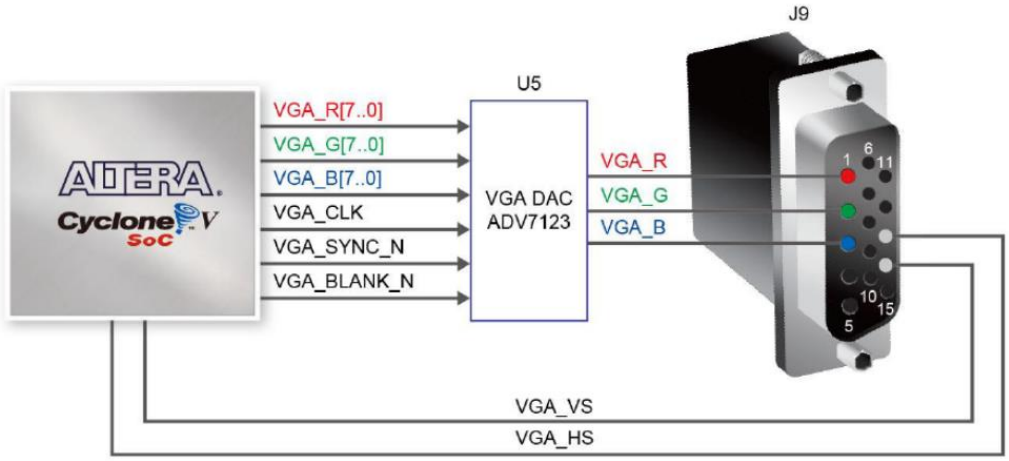
工作原理：VGA 每一帧图像的显示都是从屏幕的左上角开始一行一行进行的，行同步信号是一个负脉冲，行同步信号有效后，由 RGB 端送出当前行显示的各像素点的 RGB 电压值，当一帧显示结束后，由帧同步信号送出一个负脉冲，重新开始从屏幕\的左上端开始显示下一帧图像，如图所示。



RGB 端并不是所有时间都在传送像素信息，由于 CRT 的电子束从上一行 的行尾到下一行的行头需要时间，从屏幕的右下角回到左上角开始下一帧也需 要时间，这

时 RGB 送的电压值为 0（黑色），这些时间称为电子束的行消隐时 间和场消隐时间，行消隐时间以像素为单位，帧消隐时间以行为单位。

显示原理：DE10-Standard 开发板上使用了一块 VGA DAC ADV7123 芯片来实现 VGA 功能。该芯片完成 FPGA 数字信号到 VGA 模拟信号的转换，具体连接方式如 图 所示。



开发板和 ADV7123 芯片之间的接口引脚包括 3 组 8bit 的颜色信号 VGA_R[7:0], VGA_G[7:0], VGA_B[7:0]，行同步信号 VGA_HS，帧同步信号 VGA_VS，VGA 时钟信号 VGA_CLK，VGA 同步（低有效）VGA_SYNC_N, 和 VGA 消隐信号（低有效）VGA_BLANK_N。

2.2 通用时钟

通过传进去参数的不同，使得时钟的频率不同。

```
module clkgen(
    input clkin,
    input rst,
    input clken,
    output reg clkout
    );
    parameter clk_freq=1000;
    parameter countlimit=50000000/2/clk_freq; //自动计算计数次数
```

```
reg[31:0] clkcount;
always @ (posedge clkin)
 if(rst)
  begin
     clkcount=0;
     clkout=1'b0;
  end
 else
 begin
  if(clken)
     begin
          clkcount=clkcount+1;
          if(clkcount>=countlimit)
           begin
              clkcount=32'd0;
              clkout=~clkout;
           end
          else
              clkout=clkout;
      end
   else
     begin
          clkcount=clkcount;
          clkout=clkout;
     end
  end
endmodule
```

2.3 vga 驱动信号

代码过长，这里不贴，见于文件 vga_ctrl 模块，与 pdf 基本相同

3.实验环境

- 软件环境

Quartus 17.1 Lite

- 硬件环境

开发板：DE10 Standard

FPGA：Intel Cyclone V SE 5CSXFC6D6F31C6N

4.实验步骤和结果

4.1 显存模块

按照 pdf 要求，显存分配大小为 640 × 512 word, 每个 word 为 12bit。

由于只用读，所以用 ip 核生成，用.mif 文件初始化。

4.2 vga 驱动信号模块

基本于 pdf 相同，唯一不同的是按照 pdf 所述要求将赋值改为如下：

```
wire [3:0] zero=0;


// 设 置 输 出 的 颜 色 值
assign vga_r = {vga_data[11:8],zero};
assign vga_g = {vga_data[7:4],zero};
assign vga_b = {vga_data[3:0],zero};
```

4.3 顶层模块与接口

主要注意 pdf 要求以及接口对应。另外再加个分频器。其余没什么工作量。

```
//

clkgen #(25000000) my_vgaclk(CLOCK_50,SW[0],1'b1,VGA_CLK);

assign VGA_SYNC_N=0;


wire [9:0] h_addr;
wire [9:0] v_addr;
wire [18:0] addr={h_addr,v_addr[8:0]};
wire [11:0] vga_data;
rom ro(
    addr,
    VGA_CLK,
    vga_data);

vga_ctrl myvga(
VGA_CLK, //25MHz时钟
0, //置位
vga_data, // 上层模块提供的VGA颜色数据
h_addr, // 提供给上层模块的当前扫描像素点坐标
v_addr,
VGA_HS, // 行同步和列同步信号
VGA_VS,
VGA_BLANK_N, //消隐信号
VGA_R, // 红绿蓝颜色信号
VGA_G,
VGA_B
 );
```

4.3 拓展模块：jpg 转 mif

配置 python 环境，装 pillow 模块，运行如下代码:

img2mif.py - D:\NJU_ALL\Sophomore\Digital Logic and Principles of Computer Organization\2021\exp8_board\img2mif.py (3.10.0)

File Edit Format Run Options Window Help

```
from PIL import Image
height = 480
width = 640
ratio = width /height

def img2mif(filename):
    filepath = "./"+filename
    outpath="pic.mif"
    f=open(outpath, "w")
    f.write("--VGA Memory Map\n---Height: 480,Width: 640\n\nWIDTH=12;\nDEPTH=327680;\nADDRESS_RADIX=HEX;\nDATA_RADIX=HEX;\nCONTENT\nBEGIN\n")
    image=Image.open(filepath)
    image=image.crop((1,1,image.height*ratio,image.height))
    image=image.resize((width,height))
    image.save("new_"+filename)
    cnt = 0
    for c in range(0, image.width):
            for r in range(0,512):
                    cnt=cnt+1
                    if r < height:
                            R=image.getpixel((c,r))[0]
                            G=image.getpixel((c,r))[1]
                            B=image.getpixel((c,r))[2]
                    else:
                            R=15
                            G=15
                            B=15
                    f.write("{0:06x}: {1:x}{2:x}{3:x};\n".format(cnt-1, R>>4, G>>4, B>>4))
    f.write("END;")
    f.close()
```

传入参数即图片地址，即可生成 mif 文件。

5. 实验中遇到的问题及解决办法

问题：用 python 运行代码时出错。

解决办法：装 pillow 库。

6.实验启示和建议

这个实验难点反而在生成.mif 文件上了，vga 基础功能倒是比较简单。。。