

实验名称：实验九 VGA 接口控制器实现

姓名：张涵之

学号：191220154

班级：周一 5-6

邮箱：[191220154@smail.nju.edu.cn](mailto:191220154@smail.nju.edu.cn)

实验时间：2020/11/14

## 9.3 实验内容

### 9.3.1 显示不同颜色条纹

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。

实验目的：输出不同颜色的条纹。

实验原理：根据扫描的行数据，通过调整红绿蓝的比例显示不同的颜色，形成条纹。

程序代码或流程图：

clkgen 生成特定频率的时钟—— \

vga\_ctrl 生成各类驱动信号——> 主模块判断当前扫描行设置颜色

```
//=====
//  REG/WIRE declarations
//=====

wire [9:0] h_addr;
wire [9:0] v_addr;
reg [23:0] vga_data = 24'hfffffff;

clkgen #(25000000) my_vgaclk(CLOCK_50, SW[0], 1'b1, VGA_CLK);
vga_ctrl my_vga(VGA_CLK, SW[0], vga_data, h_addr, v_addr,
               VGA_HS, VGA_VS, VGA_BLANK_N, VGA_R, VGA_G, VGA_B);

//=====
//  structural coding
//=====

always @(v_addr) begin
    if (v_addr <= 69) vga_data <= 24'hff0000;
    else if (v_addr <= 137) vga_data <= 24'hffaa00;
    else if (v_addr <= 206) vga_data <= 24'hffff00;
    else if (v_addr <= 274) vga_data <= 24'h00ff00;
    else if (v_addr <= 343) vga_data <= 24'h00ffff;
    else if (v_addr <= 411) vga_data <= 24'h0000ff;
    else if (v_addr <= 480) vga_data <= 24'hff00ff;
    else vga_data <= 24'hffffff;
end

endmodule
```

实验步骤/过程：

已经提供了 clkgen 和 vga\_ctrl 的参考代码，故只需要在顶层模块中将它们综合。

阅读参考代码理解含义，用 System Builder 创建工作项目，填入参数。

根据控制模块输出的行参数判断扫描位置，给上层模块提供的颜色数据不同的赋值。

测试方法：连接显示器，在显示器观察输出的条纹。

实验结果：通过观察，显示器上输出的颜色条纹符合预期。

实验中遇到的问题及解决办法：不知道 System Builder 生成的输入输出端口和 vlg\_ctrl 模块的参数如何对应。解决方法：反复阅读代码，理解原理，从而得出需要的参数。

实验得到的启示：无。

意见和建议：无。

### 9.3.2 图片显示

利用上述控制器，在显示器上显示一张静态图片。

实验目的：在显示器上显示一张静态图片。

实验原理：用 MATLAB 生成静态图片的 mif 文件，读入寄存器，根据扫描的位置坐标读出该位置的颜色数据。使用低比特的颜色显示的方式来绕过 RAM 不足的问题。

程序代码或流程图：

clkgen 生成特定频率的时钟—— \

vga\_ctrl 生成各类驱动信号——> 主模块判断当前扫描行设置颜色

picture 读入 mif 格式图片数据 /

```
//=====
//  REG/WIRE declarations
//=====

wire [9:0]h_addr;
wire [9:0]v_addr;
reg [23:0]vga_data = 24'hffffff;
reg [18:0]addr = 19'h0;
wire [11:0]out_data;

clkgen #(25000000) my_vgaclk(CLOCK_50,SW[0],1'b1,VGA_CLK);
picture_pic(addr,CLOCK_50,out_data);
vga_ctrl my_vga(VGA_CLK,SW[0],vga_data,h_addr,v_addr,
    VGA_HS,VGA_VS,VGA_BLANK_N,VGA_R,VGA_G,VGA_B);

//=====
//  Structural coding
//=====

assign VGA_SYNC_N = 1'b0;

always @(v_addr or h_addr) begin
    addr = v_addr + (h_addr - 1) * 512 - 1;
    vga_data[3:0] <= 4'b0;
    vga_data[7:4] <= out_data[3:0];
    vga_data[11:8] <= 4'b0;
    vga_data[15:12] <= out_data[7:4];
    vga_data[19:16] <= 4'b0;
    vga_data[23:20] <= out_data[11:8];
end

endmodule
```

//对显存用.mif文件初始化，其中每像素按RGB各4比特，地址按列排列，开头是第一列像素512个点，其中超过480行的像素置为白色。然后顺序排列640列像素

//则根据行列数据得到在mif文件中的地址为 列坐标 + (行坐标 - 1) \* 列数 - 1

//红绿蓝颜色：根据12bit显存中的4bit值输出8bit数据的高4位，低4位置零

实验步骤/过程：

已经提供了clkgen和vga\_ctrl的参考代码，还需要存储模块读入mif格式的图片数据。

显存分配大小为640×512 word，每个word为12bit。用h\_addr的全部10位和v\_addr的低9位合成19位地址来索引显存。为方便寻址给v\_addr分配512行的空间。可以不用对地址进行转换，分配327680个连续的存储单元，不需要考虑h\_addr大于640的情况。

使用ROM:1-PORT实现mif文件的读入和存储模块。

How wide should the 'q' output bus be?	<input type="text" value="12"/> bits
How many 12-bit words of memory?	<input type="text" value="327680"/> words

根据控制模块输出的行参数判断扫描位置，进行相应的坐标转换得到内存地址，从存储模块读出图片信息，给上层模块提供的颜色数据高四位不同的赋值，低四位置零。

测试方法：连接显示器，在显示器观察输出的图片。

实验结果：通过观察，显示器上输出的图片效果符合预期。

实验中遇到的问题及解决办法：无。

实验得到的启示：无。

意见和建议：无。

