实验九 VGA接口控制器实现

实验报告

181860085 汤昊

数字电路与数字系统二班

邮箱：1174639585@qq.com

2019.11.1

一．实验目的

学习vga接口的工作原理，学习vga接口控制器的设计方法。

二．实验原理

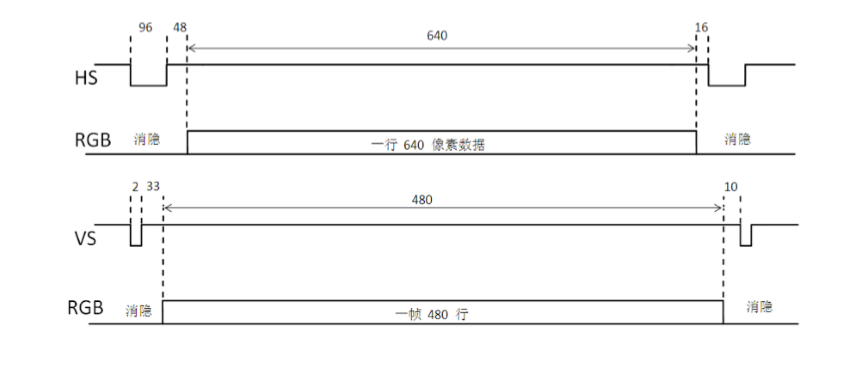
（1）vga图像显示

图像的显⽰是以像素（点）为单位，显⽰器的分辨率是指屏幕每⾏有多少 个像素及每帧有多少⾏，标准的VGA分辨率是640×480

VGA 显⽰器⼀般的刷新频率是60HZ。 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，重新开始从屏的左上端开始显⽰下⼀帧图像。

RGB端并不是所有时间都在传送像素信息，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。

（2）vga扫描时序



有效显⽰⼀⾏信号需要 96+48+640+16=800个像素点的时间，其中⾏同步负脉冲宽度为96个像素点时间，⾏消隐后沿需要48个像素点时间，然后每⾏显⽰640个像素点，最后⾏消 隐前沿需要16个像素点的时间。所以⼀⾏中显⽰像素的时间为640个像素点时间，⼀⾏消隐时间为160个像素点时间。

有效显⽰⼀帧图像需要2+33+480+10=525⾏ 时间，其中场同步负脉冲宽度为2个⾏显⽰时间，场消隐后沿需要33个⾏显⽰ 时间，然后每场显⽰480⾏，场消隐前沿需要10个⾏显⽰时间，⼀帧显⽰时间 为525⾏显⽰时间，⼀帧消隐时间为45⾏显⽰时间。

因此，在 640×480 的VGA上的⼀幅图像需要 525×800 =420k个像素点的 时间。⽽每秒扫描60帧共需要约25M个像素点的时间。

（3）vga接口信号

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

三．实验环境及器材

开发软件：quartus prime 17.1

开发器材：DE-standard 开发板

vga接口的显示器

四．接口控制器

（1）模块划分

vga信号生成

地址及信号

分开的rgb颜色数据

颜色数据

及时钟信号

25MHz时钟

顶层控制模块

时钟信号

显存地址

及时钟信号

颜色数据

显存

（2）25MHz时钟——clkgen模块

一个可以根据调用时参数生成不同频率时钟的生成器。

原理：使用了parameter量clk\_freq，而生成器产生一次输出的计数次数与此参数有关，countlimit=50000000/2/clk\_freq。

在本实验中需要25MHz的时钟，故传参25000000。

（3）vga信号产生——vga\_ctrl模块

根据扫描时序产生以下vga信号：

x\_cnt，y\_cnt：根据时钟信号计数

h\_sync，v\_sync：x\_cnt>96产生行同步信号，y\_cnt>2产生列同步信号

h\_addr，v\_addr：行和列地址，在x\_cnt>144和y\_cnt>35时才是有效像素坐标

h\_valid，v\_valid：x\_cnt超过784产生行消隐信号，y\_cnt超过515产生列消隐信号

valid：行列都消隐时消隐

vga\_r/g/b：RGB颜色数据

（4）显存——memory模块

实验采用低比特颜色显示方案，将v\_addr和h\_addr合成显存地址，在时钟沿获得输出（这样quartus可以使用BLOCKRAM(M10K)综合）。

addr[8:0]=v\_addr[8:0];

addr[18:9]=h\_addr;

可以这样赋值的原因是显存的空间和照片mif的大小都是640\*512，这样addr=h\_addr\*512+v\_addr，故低九位直接对应v\_addr低 九位，高十位直接对应h\_addr。

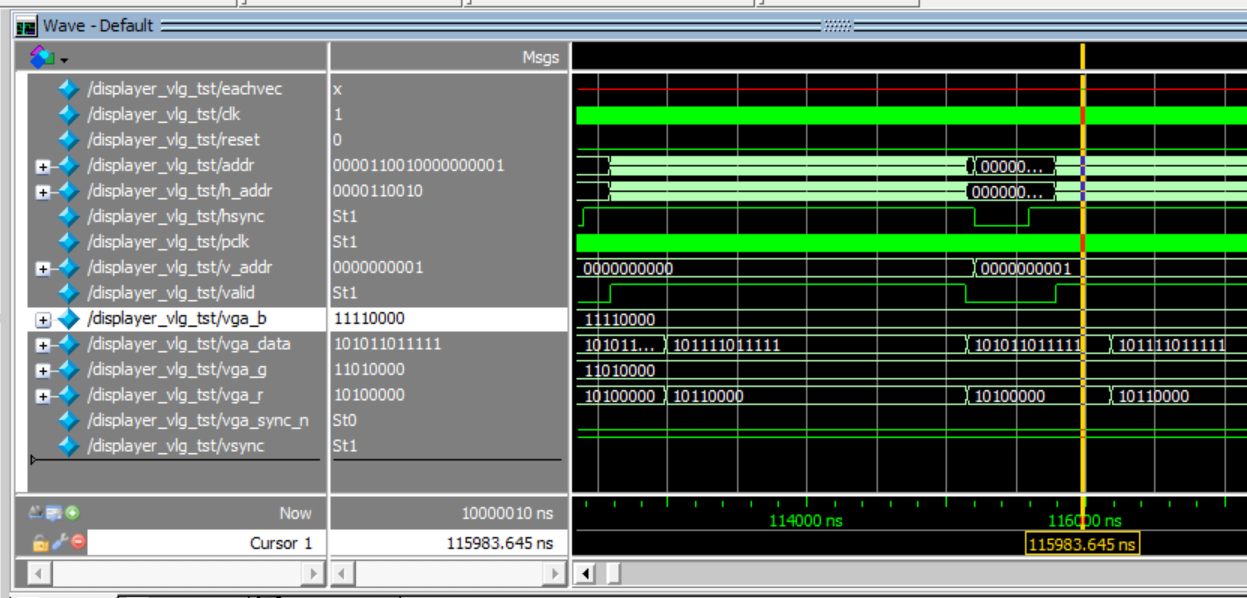
（5）顶层控制模块——displayer模块

* 实例化时钟，将25MHz时钟信号送给vga\_ctrl和memory，在不需要显存如显示条纹时，需要在时钟上升沿对颜色数据处理。
* 实例化vga\_ctrl，将各种信号引出，根据行列信号合成显存地址
* 实例化memory，得到显存地址的数据送给vga\_ctrl分成RGB数据输出

五．显示不同颜色条纹

（1）设计思路：在时钟上升沿对行地址进行判断，若小于某值则为一种颜色，不小于为另一种颜色。设置的值为100

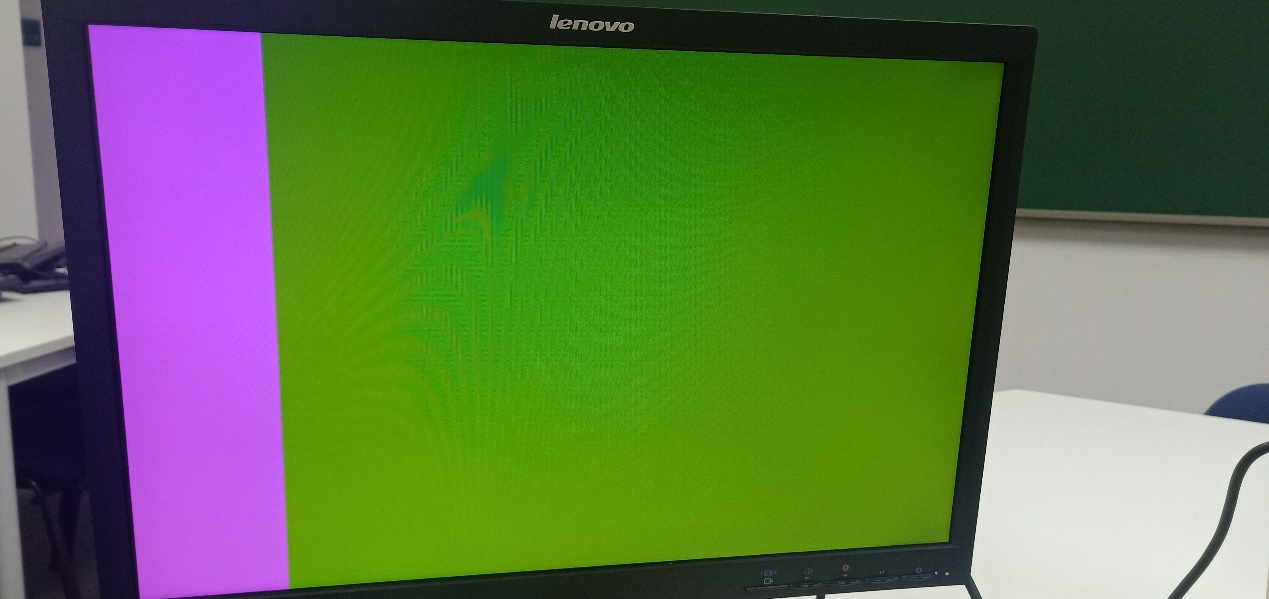
（2）仿真模拟结果



设置h\_addr<100时，数据为12’hADF，h\_addr>=100时数据为12‘hBDF，仿真结果中可以看到在光标所放的位置附近有一次101011011111变成101111011111

（3）实际效果

由于12‘hADF，12‘hBDF对应颜色较淡，最终换成了对比较强的紫色和绿色，数据分别为12’hB3F和12‘h480，效果如下图所示。



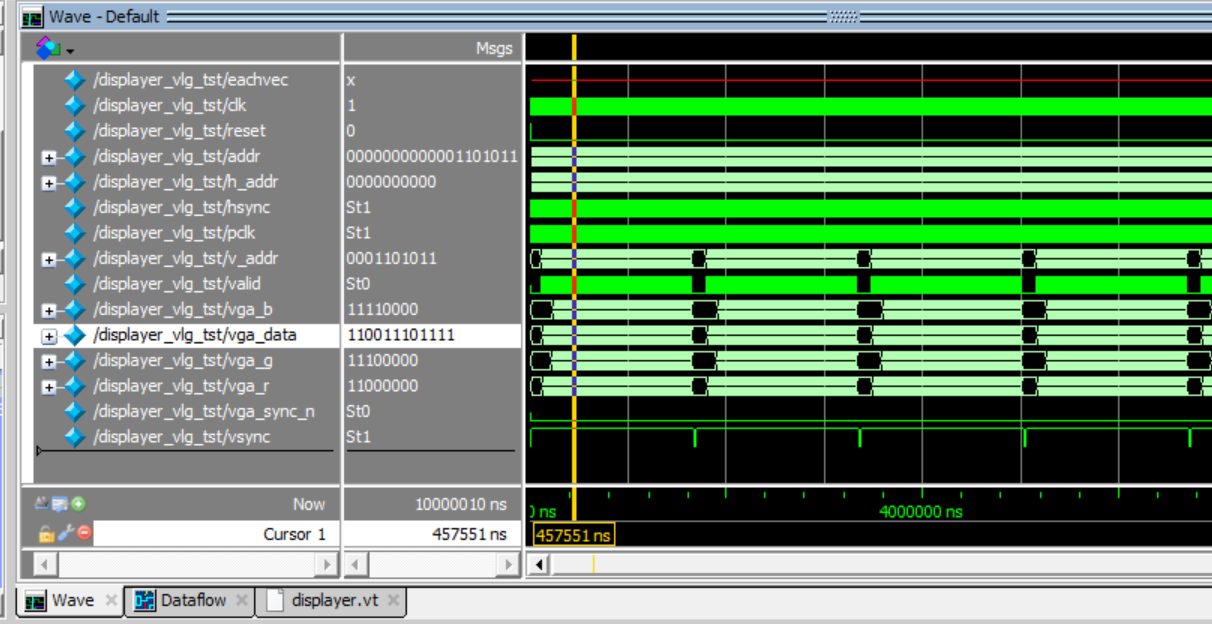
六．显示静态图片

1.显示给出图片

（1）设计思路

原有基础上在顶层模块中引入显存，用给出的mif文件初始化。显存rom使用IP核创建。

（2）仿真模拟



可以看到随着v\_addr，h\_addr变化，addr，vga\_data，vga\_r/g/b都在不停变化

（3）显示结果

图片格式为640\*480



2.显示自定义图片

（1）设计思路

原有基础上使用matlab脚本生成自定义图片的mif文件，初始化显存。

初始化方式为(\* ram\_init\_file = "my\_picture.mif" \*) reg [11:0] mem[327679:0];

但使用此语句则无法使用仿真模拟测试。

（2）显示结果

使用的图片格式为640\*400



七．动态显示图片

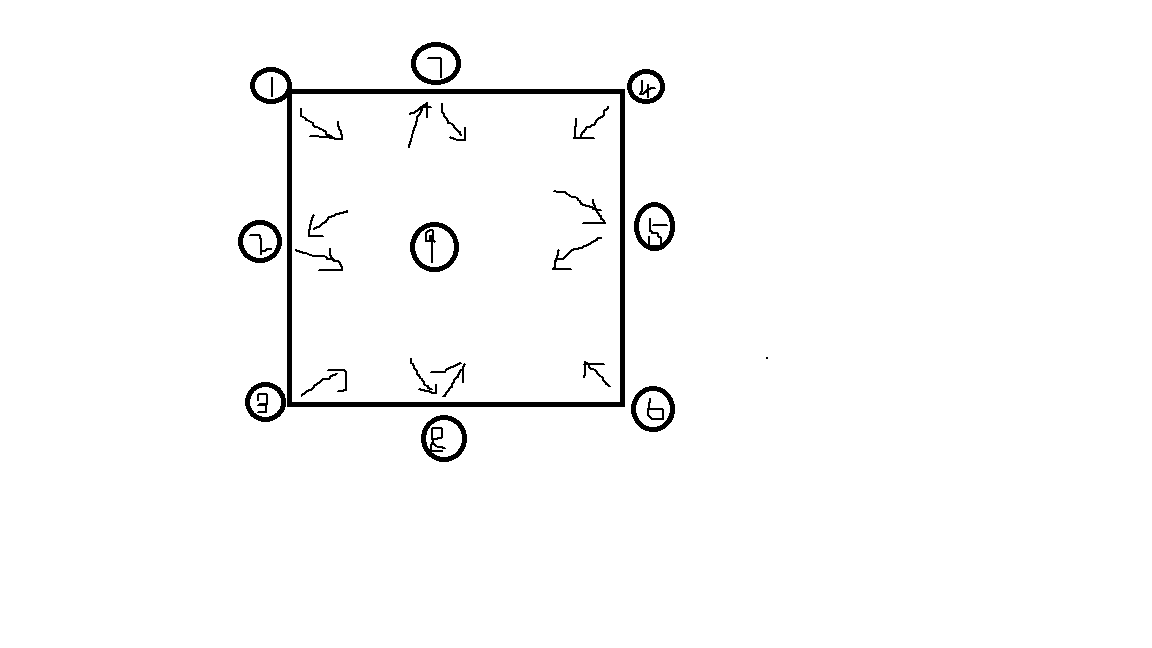
1.确定图片显示区域

代码中使用logo\_x和logo\_y确定图片的起始位置（左上角），再根据logo的高度和宽度确定当前的行列地址是否在图片区域内，在则rom\_addr加一，取出对应颜色显示，不在则rom\_addr保持原来的位置，颜色显示为黑色

2.实现图片的移动和反弹

flag\_add\_sub代表移动方式，logo\_x和logo\_y会根据这个标志进行相应的加减坐标，如当标志为00时，在时钟沿两个坐标都要加一，标志位11时反之。

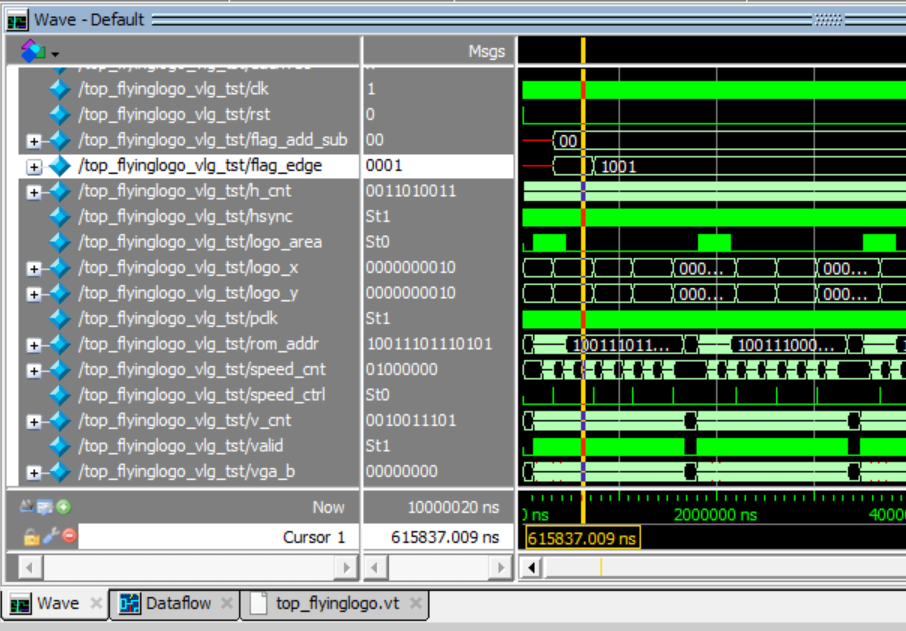
|  |  |
| --- | --- |
| 00 | 右下 |
| 01 | 右上 |
| 10 | 左下 |
| 11 | 左上 |



根据logo\_x和logo\_y目前所在的位置分成了如图所示9种情况，flag\_edge记录了是那种情况

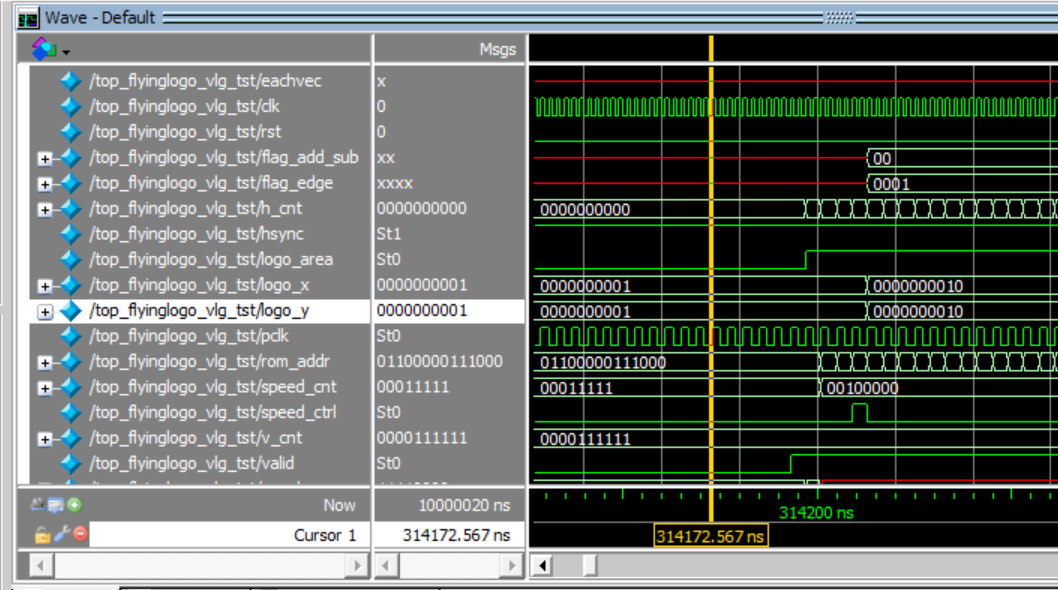
1. 图片在左上角，应该右下移动
2. 图片在左边缘，应该向相反方向移动，实现反弹效果
3. 图片在左下角，应该向右上移动
4. 图片在右上角，应该向左下移动
5. 图片在右边缘，应该向相反方向移动，实现反弹效果
6. 图片在右下角，应该向左上角移动
7. 图片在上边缘，应该向相反方向移动，实现反弹效果
8. 图片在下边缘，应该向相反方向移动，实现反弹效果
9. 图片未触及边缘，移动方向不变

可以看到下图仿真模拟中logo\_x,y从1到2时（右下移动），flag\_edge的状态从0001变为1001，从第一个状态到第九个状态



3.速度方向的控制信号

speed\_cnt初始为0，在v\_cnt[5]==0h\_cnt==1时加一，即当行数在32-64时（不包括64）且列数为1时计数，而speed\_ctrl有效的条件是 speed\_cnt[5]先0，再连续两个为1才会输出speed\_ctrl为1，即在speed\_cnt[5]变为32时有效一次，故行数为63时speed\_ctrl有效（如下图仿真模拟所示），判断当前的状态，得出移动的方向



八．实验反思和收获

* 使用初始化方式为(\* ram\_init\_file = “\*.mif" \*) 时仿真无法执行，故体现不出颜色数据的变化
* 在动态显示模块中显存地址的寻址方式为直接得到显存地址，即在上升沿及logo\_area有效时加一，即行优先，但这样与生成mif文件数据的格式冲突（代码中颜色数据的排列是列优先），实际测试中显示的图像是原来的转置，在修改生成mif文件的顺序后正常。