

# 数字电路与数字系统实验

---

## 实验九 显示器

---

计算机科学与技术系

191220008 陈南瞳  
[924690736@qq.com](mailto:924690736@qq.com)

2020.11.13

# 一、实验目的

VGA 接口是 IBM 制定的一种视频数据的传输标准, 是电脑显示器最典型的接口。本实验的目的是学习 VGA 接口原理, 学习 VGA 接口控制器的设计方法。

## (一) 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 (横条或竖条均可以)。

## (二) 图片显示

利用上述控制器, 在显示器上显示一张静态图片。

我们建议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。

### 低比特颜色显示方案

显存分配大小为  $640 \times 512$  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, 我们给行 v\_addr 分配了 512 行的空间。这样, 可以不用对地址进行复杂的转换。此处只需要分配 327680 个连续的存储单元, 不需要考虑 h\_addr 大于 640 的情况。

assign 红、绿、蓝颜色的时候, 根据 12bit 显存数据中对应颜色的 4bit 值, 设置输出 8bit 数据的高 4 位, 低 4 位置零。

对显存用 .mif 文件初始化。可以自己用常用的脚本语言生成 .mif 文件, 我们也提供了一张  $640 \times 512$  的 12bit 图片的 my\_picture.mif 文件, 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。

需要注意的是, 显存占用空间较大, 实现时需要用时钟沿驱动的显存, 这样系统可以用 BLOCK RAM(M10K) 来实现。当资源不够时, Quartus 可能会无法综合, 耗费大量时间编译。

### 拓展要求:

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  $100 \times 100$  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。

## 二、实验原理（知识背景）

### 1、VGA 接口的外观和引脚功能

VGA（Video Graphics Array）接口，即视频图形阵列。VGA 接口最初是用于连接 CRT 显示器的接口，CRT 显示器因为设计制造上的原因，只能接受模拟信号输入，这就需要显卡能输出模拟信号。关于模拟信号和数字信号的区别，请参考（Analog Signal 及 Digital Signal）。VGA 接口就是显卡上输出模拟信号的接口，在传统的 CRT 显示器中，使用的都是 VGA 接口，现在仍有不少液晶显示器或投影仪还支持 VGA 口。VGA 接口是 15 针/孔的梯形插头，分成 3 排，每排 5 个，如图所示：

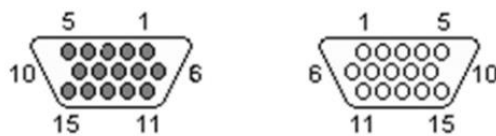


图 9-1: VGA 接口形状示意图

VGA 接口的接口信号主要有 5 个：R（Red）、G（Green）、B（Blue）、HS（Horizontal Synchronization）和 VS（Vertical Synchronization），即红、绿、蓝、水平同步和垂直同步（也称行同步和帧同步）。

### 2、VGA 的工作原理

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，也有更高的分辨率，如 1024×768、1280×1024、1920×1200 等。从人眼的视觉效果考虑，屏幕刷新的频率（每秒钟显示的帧数）应该大于 24，这样屏幕看起来才不会闪烁，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，重新开始从屏幕的左上端开始显示下一帧图像，如图所示。

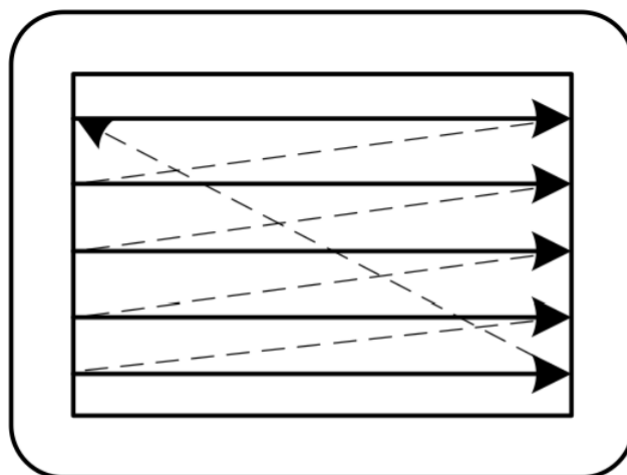


图 9-2: 显示器扫描示意图

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。VGA 行扫描、场扫描时序示意图如图所示:

由图可知, 在标准的  $640 \times 480$  的 VGA 上有效地显示一行信号需要  $96+48+640+16=800$  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。

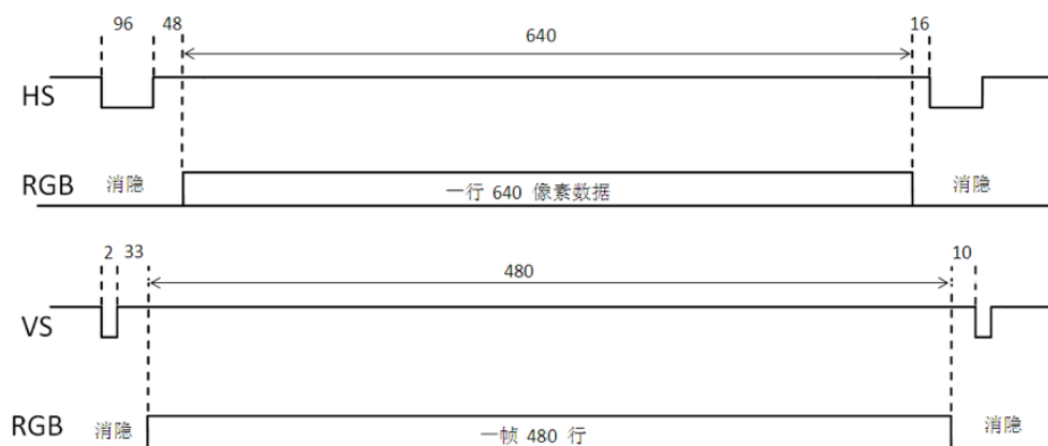


图 9-3: VGA 行扫描、场扫描时序示意图

在标准的  $640 \times 480$  的 VGA 上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  $640 \times 480$  的 VGA 上的一幅图像需要  $525 \times 800 = 420k$  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。

### 三、实验环境/器材等

#### 1) 软件环境:

Quartus (Quartus Prime 17.1) Lite Edition

#### 2) 硬件环境:

DE10-Standard 开发板

FPGA 部分:

Intel Cyclone V SE 5CSXFC6D6 F31C6N

- 110K 逻辑单元
- 5,761Kbit RAM

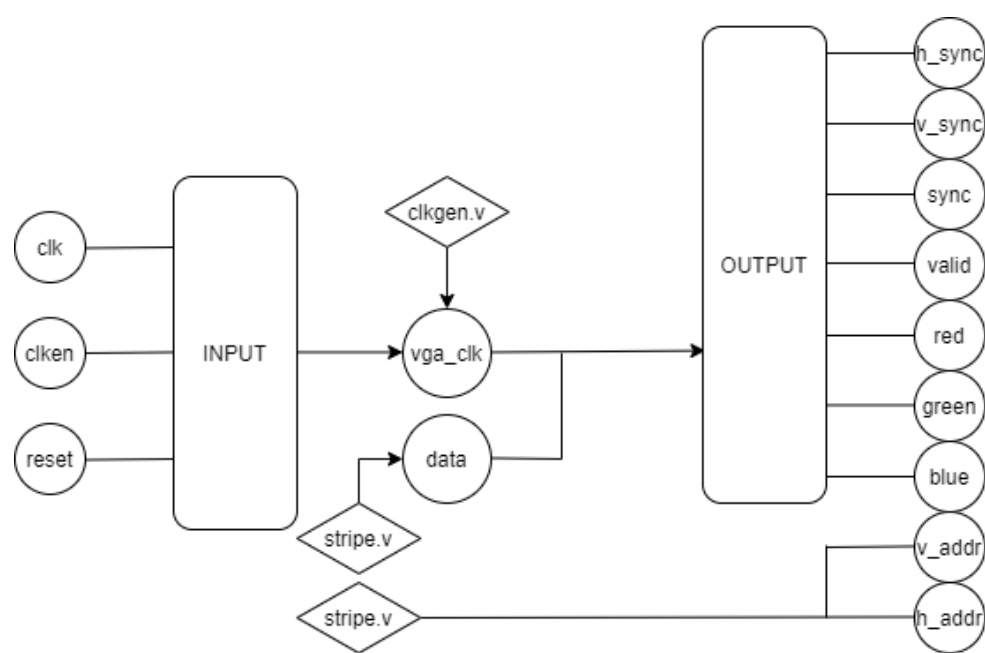
HPS 部分:

Dual-core ARM Cortex A9

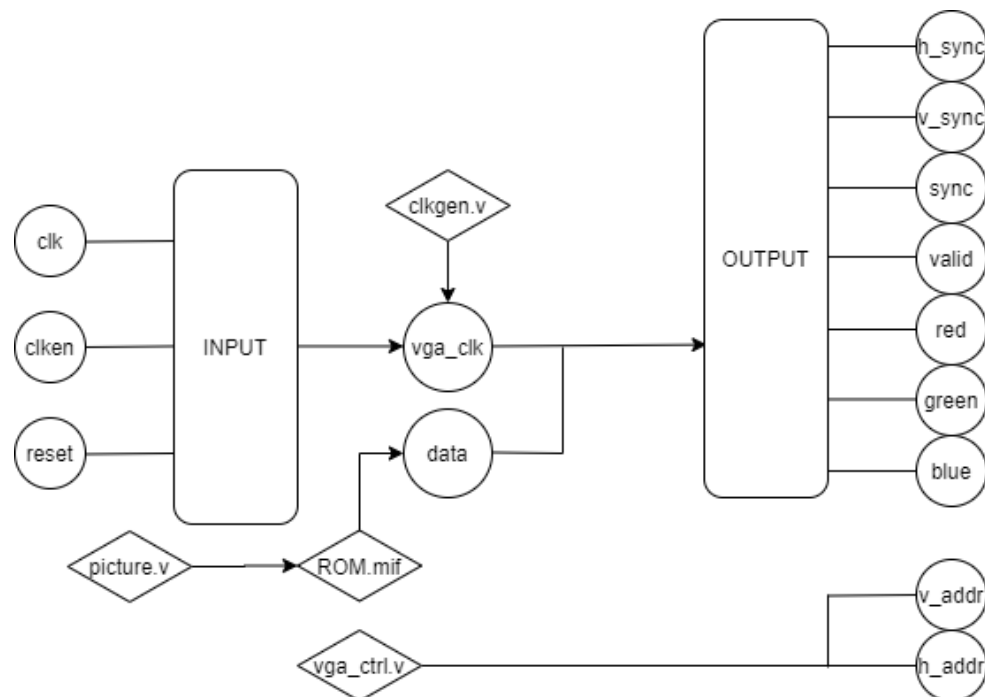
- 925MHz
- 1GB DDR

### 四、程序代码或流程图

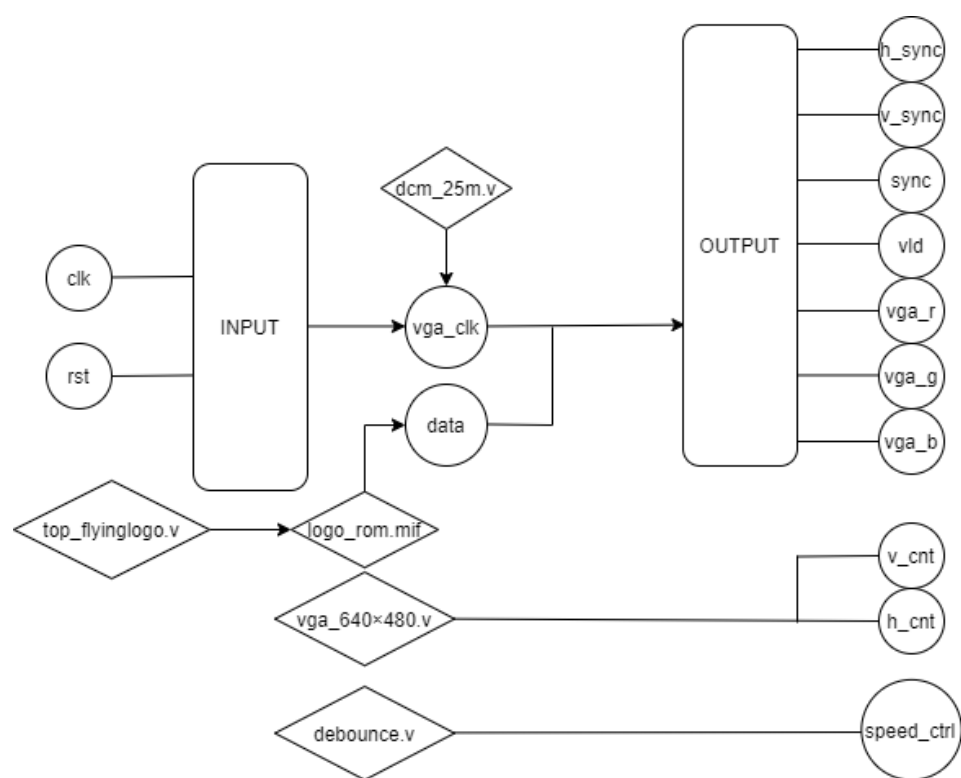
显示条纹：



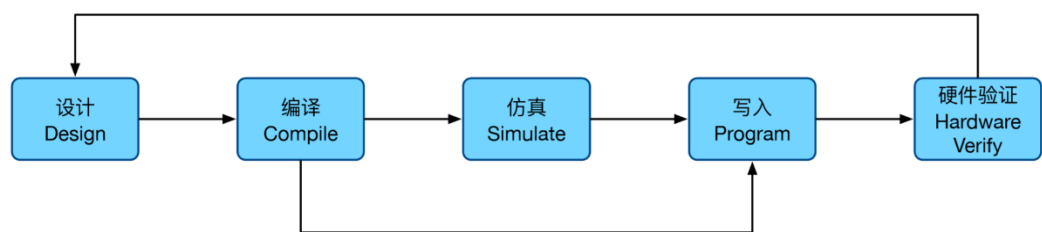
显示图片：



显示动图：



## 五、实验步骤/过程



设计：

```

assign vga_clk = temp_clk;

clkgen #(25000000) my_vgaclk(clk, reset, 1'b1, temp_clk);

vga_ctrl my_vgactrl(temp_clk, reset, data, h_addr, v_addr, h_sync, v_sync, valid,

always @(posedge vga_clk)
begin
    if (h_addr <= 106)
        data = 24'hAA0000;
    else if (h_addr <= 212)
        data = 24'h0BB000;
    else if (h_addr <= 318)
        data = 24'h00cc00;
    else if (h_addr <= 424)
        data = 24'h000DD0;
    else if (h_addr <= 530)
        data = 24'h0000EE;
    else
        data = 24'hF0000F;
end

```

```

assign vga_clk = temp_clk;

clkgen #(25000000) my_vgaclk(clk, reset, 1'b1, temp_clk);
vga_ctrl my_vgactrl(temp_clk, reset, data, h_addr, v_addr, h_sync, v_sync, valid,
ROM my_rom(rom_addr, temp_clk, temp_color);

always @(posedge clk)
begin
    data[3:0] = 4'b0000;
    data[7:4] = temp_color[3:0];
    data[11:8] = 4'b0000;
    data[15:12] = temp_color[7:4];
    data[19:16] = 4'b0000;
    data[23:20] = temp_color[11:8];
    rom_addr = {h_addr, v_addr[8:0]};
end

```

```

dcm_25m #(25000000) u0 (
    // clock in ports
    .clk_in(clk),          // input clk_in1
    // Status and control signals
    .rst(rst),
    // clock control signals
    .clken(1),
    // clock out ports
    .clk_out(pclk)        // output clk_out1
);

logo_rom u1 (
    .clock(pclk),          // input wire clka
    .address(rom_addr),    // input wire [13 : 0] addra
    .q(douta)              // output wire [11 : 0] douta
);

vga_640x480 u2 (
    .pclk(pclk),
    .reset(rst),
    .hsync(hsync),
    .vsync(vsync),
    .valid(valid),
    .h_cnt(h_cnt),
    .v_cnt(v_cnt)
);

```



测试：无

编译：

Quartus Prime Lite Edition - C:/Digital\_Experiment/EXP9/EXP9\_1/EXP9\_1 - EXP9\_1

File Edit View Project Assignments Processing Tools Window Help

EXP9\_1

Project Navigator Files

- Files
  - clkgen.v
  - vga\_ctrl.v
  - stripe.v

Table of Contents

- Flow Summary
  - Flow Settings
    - Flow Status: Successful - Sun Nov 08 09:05:18 2020
    - Quartus Prime Version: 17.1.0 Build 590 10/25/2017 SJ Lite Edition
    - Revision Name: EXP9\_1
    - Top-level Entity Name: EXP9\_1
    - Family: Cyclone V
    - Device: 5CSXFC6D6F31C6
    - Timing Models: Final
    - Logic utilization (in ALMs): 82 / 41,910 (< 1 %)
    - Total registers: 77
    - Total pins: 32 / 499 (6 %)
    - Total virtual pins: 0
    - Total block memory bits: 0 / 5,662,720 (0 %)
    - Total DSP Blocks: 0 / 112 (0 %)
    - Total HSSI RX PCSs: 0 / 9 (0 %)
    - Total HSSI PMA RX Deserializers: 0 / 9 (0 %)
    - Total HSSI TX PCSs: 0 / 9 (0 %)
    - Total HSSI PMA TX Serializers: 0 / 9 (0 %)
    - Total PLLs: 0 / 15 (0 %)
    - Total DLLs: 0 / 4 (0 %)
  - Flow Non-Default
  - Flow Elapsed Time
  - Flow OS Summary
  - Flow Log
  - Analysis & Synthesis
  - Fitter
  - Assembler
  - TimeQuest Timing Analysis
  - EDA Netlist Writer
  - Flow Messages
  - Flow Suppresses

Tasks

Compilation

Task

- Compile Design
- Analysis & Synthesis
- Fitter (Place & Route)
- Assembler (Generate programming file)
- TimeQuest Timing Analysis
- EDA Netlist Writer
- Edit Settings
- Program Device (Open Programmer)

Messages

System (6) Processing (128)

100% 00:01:15

Quartus Prime Lite Edition - C:/Digital\_Experiment/EXP9/EXP9\_2/EXP9\_2 - EXP9\_2

File Edit View Project Assignments Processing Tools Window Help

EXP9\_2

Project Navigator Files

- Files
  - clkgen.v
  - vga\_ctrl.v
  - picture.v
  - ROM.qip

Table of Contents

- Flow Summary
  - Flow Settings
    - Flow Status: Successful - Sun Nov 08 09:32:30 2020
    - Quartus Prime Version: 17.1.0 Build 590 10/25/2017 SJ Lite Edition
    - Revision Name: EXP9\_2
    - Top-level Entity Name: EXP9\_2
    - Family: Cyclone V
    - Device: 5CSXFC6D6F31C6
    - Timing Models: Final
    - Logic utilization (in ALMs): 247 / 41,910 (< 1 %)
    - Total registers: 103
    - Total pins: 32 / 499 (6 %)
    - Total virtual pins: 0
    - Total block memory bits: 3,932,160 / 5,662,720 (69 %)
    - Total DSP Blocks: 0 / 112 (0 %)
    - Total HSSI RX PCSs: 0 / 9 (0 %)
    - Total HSSI PMA RX Deserializers: 0 / 9 (0 %)
    - Total HSSI TX PCSs: 0 / 9 (0 %)
    - Total HSSI PMA TX Serializers: 0 / 9 (0 %)
    - Total PLLs: 0 / 15 (0 %)
    - Total DLLs: 0 / 4 (0 %)
  - Flow Non-Default
  - Flow Elapsed Time
  - Flow OS Summary
  - Flow Log
  - Analysis & Synthesis
  - Fitter
  - Assembler
  - TimeQuest Timing Analysis
  - EDA Netlist Writer
  - Flow Messages
  - Flow Suppresses

Tasks

Compilation

Task

- Compile Design
- Analysis & Synthesis
- Fitter (Place & Route)
- Assembler (Generate programming file)
- TimeQuest Timing Analysis
- EDA Netlist Writer
- Edit Settings
- Program Device (Open Programmer)

Messages

System (12) Processing (143)

100% 00:02:41



Pin Planner - C:/Digital\_Experiment/EXP9/EXP9\_2/EXP9\_2 - EXP9\_2

File Edit View Processing Tools Window Help

Report not available

Groups Report

Tasks

Early Pin Plan

Early Pin Plan

Run I/O As:

Top View - Wire Bond  
Cyclone V - 5CSXFC6D6F31C6

Pin Legend

Symbol Pin Typ

User I/O

User a...

Fitter a...

Unbon...

Reserv...

DEV\_OE

DIFF\_n

DIFF\_p

DIFF\_n

Filter: Pins: all

Node Name	Direction	Location	I/O Bank	REF Gro	er Locat	Standards	Reserved	ent Stre	lew Rat	arential	alog Se	'VCCT_C	I/O Pin	ted Ref	non Mo	Slew R	arential	Commo	Preser
clk	Input	PL...14	3B	B3B_NO	PL...14	2.5 V		12m...lt											
clken	Input	PIN_Y27	5B	B5B_NO	PIN_Y27	2.5 V		12m...lt											
green[7]	Output	PL...23	4A	B4A_NO	PL...23	2.5 V		12m...lt	1 (...lt)										
green[6]	Output	PL...23	4A	B4A_NO	PL...23	2.5 V		12m...lt	1 (...lt)										
green[5]	Output	PL...24	4A	B4A_NO	PL...24	2.5 V		12m...lt	1 (...lt)										
green[4]	Output	PL...24	4A	B4A_NO	PL...24	2.5 V		12m...lt	1 (...lt)										
green[3]	Output	PL...24	4A	B4A_NO	PL...24	2.5 V		12m...lt	1 (...lt)										
green[2]	Output	PL...25	4A	B4A_NO	PL...25	2.5 V		12m...lt	1 (...lt)										
green[1]	Output	PL...25	4A	B4A_NO	PL...25	2.5 V		12m...lt	1 (...lt)										
green[0]	Output	PL...26	4A	B4A_NO	PL...26	2.5 V		12m...lt	1 (...lt)										
h_sync	Output	PL...19	4A	B4A_NO	PL...19	2.5 V		12m...lt	1 (...lt)										
red[7]	Output	PL...26	4A	B4A_NO	PL...26	2.5 V		12m...lt	1 (...lt)										
red[6]	Output	PL...26	4A	B4A_NO	PL...26	2.5 V		12m...lt	1 (...lt)										
red[5]	Output	PL...26	4A	B4A_NO	PL...26	2.5 V		12m...lt	1 (...lt)										
red[4]	Output	PL...27	4A	B4A_NO	PL...27	2.5 V		12m...lt	1 (...lt)										
red[3]	Output	PL...27	4A	B4A_NO	PL...27	2.5 V		12m...lt	1 (...lt)										
red[2]	Output	PL...27	4A	B4A_NO	PL...27	2.5 V		12m...lt	1 (...lt)										
red[1]	Output	PL...28	4A	B4A_NO	PL...28	2.5 V		12m...lt	1 (...lt)										
red[0]	Output	PL...29	4A	B4A_NO	PL...29	2.5 V		12m...lt	1 (...lt)										
reset	Input	PL...30	5B	B5B_NO	PL...30	2.5 V		12m...lt											
sync	Output	PL...22	4A	B4A_NO	PL...22	2.5 V		12m...lt	1 (...lt)										
v_sync	Output	PL...18	4A	B4A_NO	PL...18	2.5 V		12m...lt	1 (...lt)										
valid	Output	PL...22	4A	B4A_NO	PL...22	2.5 V		12m...lt	1 (...lt)										
vga_clk	Output	PL...21	4A	B4A_NO	PL...21	2.5 V		12m...lt	1 (...lt)										
<<ne...de>>																			

0% 00:00:00

Pin Planner - C:/Digital\_Experiment/EXP9/EXP9\_3/EXP9\_3 - EXP9\_3

File Edit View Processing Tools Window Help

Report not available

Groups Report

Tasks

Early Pin Plan

Early Pin Plan

Run I/O As:

Top View - Wire Bond  
Cyclone V - 5CSXFC6D6F31C6

Pin Legend

Symbol Pin Typ

User I/O

User a...

Fitter a...

Unbon...

Reserv...

DEV\_OE

DIFF\_n

DIFF\_p

DIFF\_n

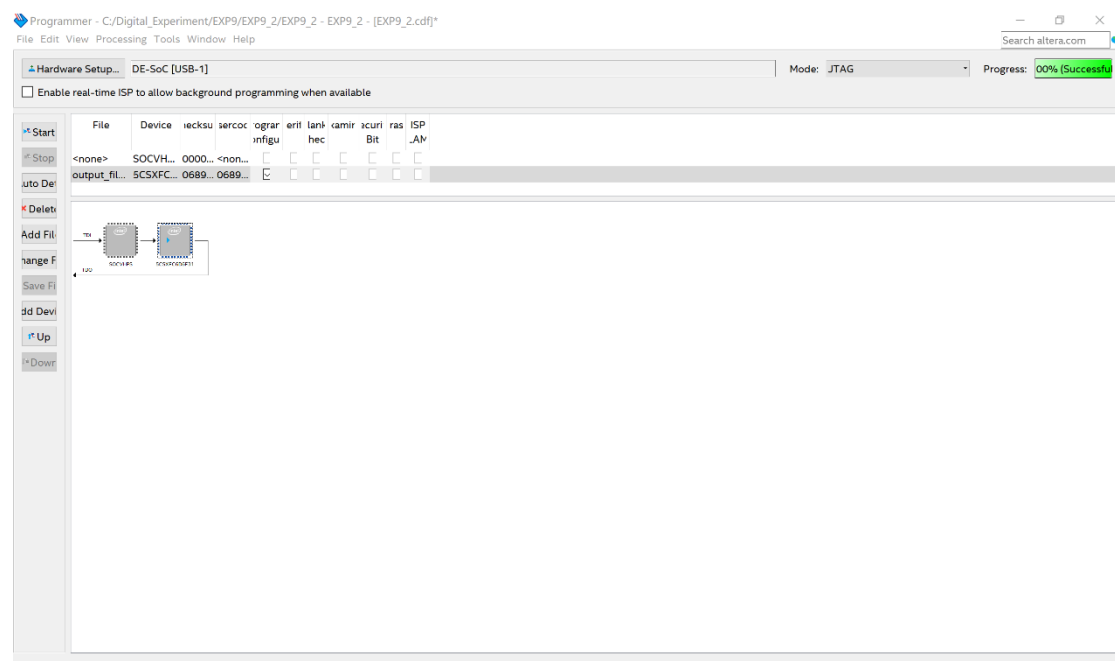
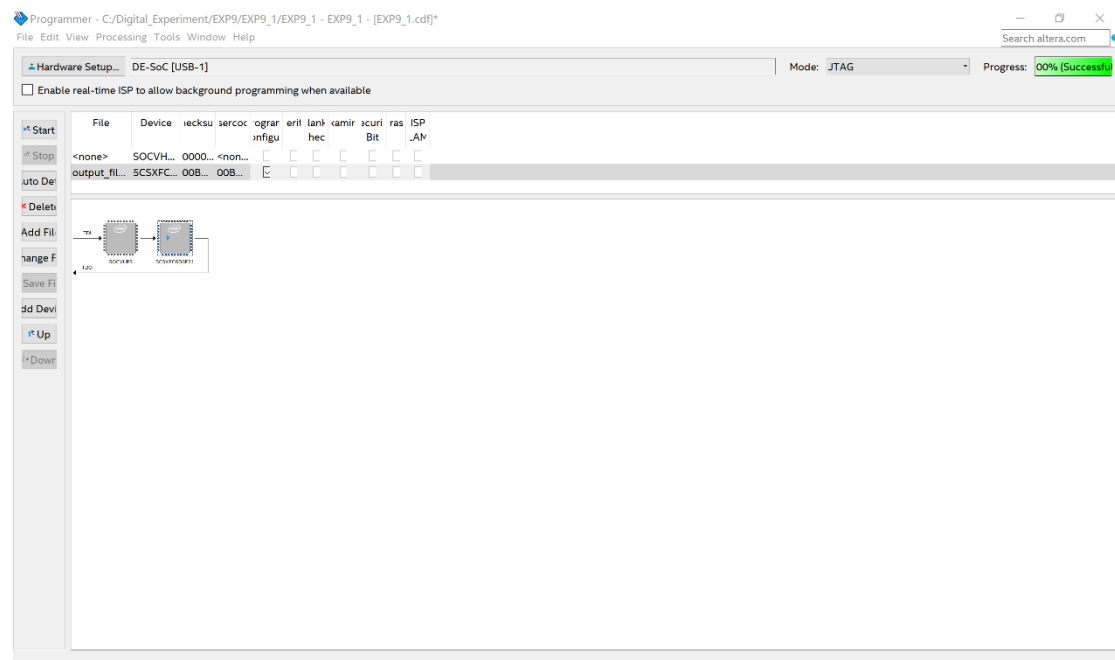
Filter: Pins: all

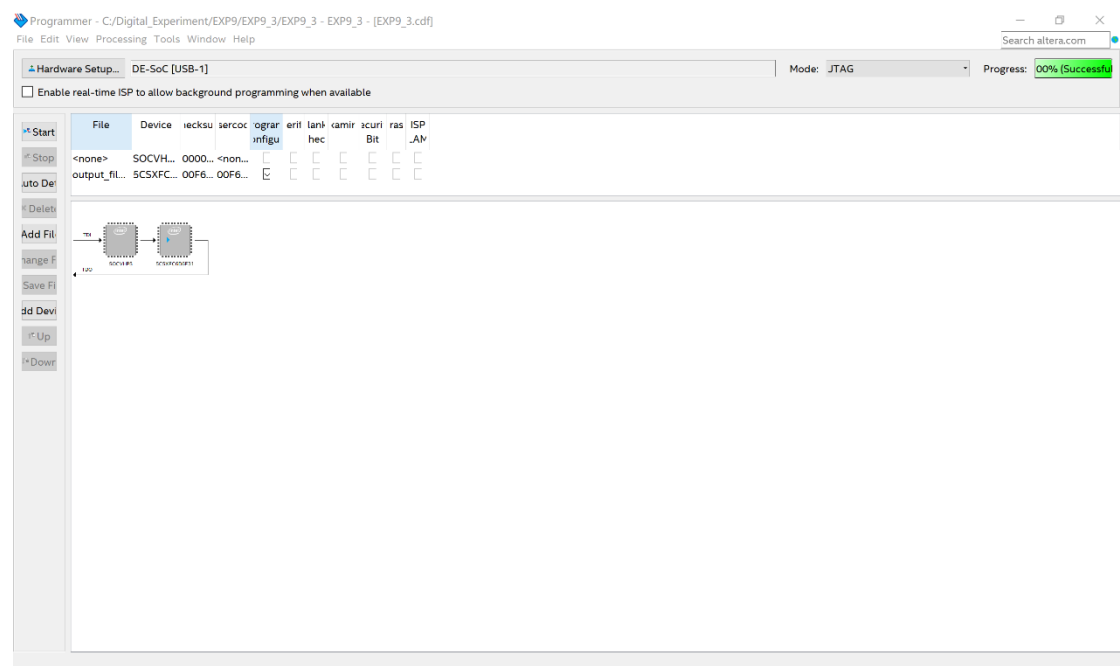
Node Name	Direction	Location	I/O Bank	REF Gro	er Locat	Standards	Reserved	ent Stre	lew Rat	arential	alog Se	'VCCT_C	I/O Pin	ted Ref	non Mo	Slew R	arential	Commo	Preser
clk	Input	PL...14	3B	B3B_NO	PL...14	2.5 V		12m...lt											
h_sync	Output	PL...19	4A	B4A_NO	PL...19	2.5 V		12m...lt	1 (...lt)										
rst	Input	PL...30	5B	B5B_NO	PL...30	2.5 V		12m...lt											
sync	Output	PL...22	4A	B4A_NO	PL...22	2.5 V		12m...lt	1 (...lt)										
v_sync	Output	PL...18	4A	B4A_NO	PL...18	2.5 V		12m...lt	1 (...lt)										
vga_b[7]	Output	PL...16	4A	B4A_NO	PL...16	2.5 V		12m...lt	1 (...lt)										
vga_b[6]	Output	PL...16	4A	B4A_NO	PL...16	2.5 V		12m...lt	1 (...lt)										
vga_b[5]	Output	PL...17	4A	B4A_NO	PL...17	2.5 V		12m...lt	1 (...lt)										
vga_b[4]	Output	PL...19	4A	B4A_NO	PL...19	2.5 V		12m...lt	1 (...lt)										
vga_b[3]	Output	PL...19	4A	B4A_NO	PL...19	2.5 V		12m...lt	1 (...lt)										
vga_b[2]	Output	PL...20	4A	B4A_NO	PL...20	2.5 V		12m...lt	1 (...lt)										
vga_b[1]	Output	PL...20	4A	B4A_NO	PL...20	2.5 V		12m...lt	1 (...lt)										
vga_b[0]	Output	PL...21	4A	B4A_NO	PL...21	2.5 V		12m...lt	1 (...lt)										
vga_clk	Output	PL...21	4A	B4A_NO	PL...21	2.5 V		12m...lt	1 (...lt)										
vga_g[7]	Output	PL...23	4A	B4A_NO	PL...23	2.5 V		12m...lt	1 (...lt)										
vga_g[6]	Output	PL...23	4A	B4A_NO	PL...23	2.5 V		12m...lt	1 (...lt)										
vga_g[5]	Output	PL...24	4A	B4A_NO	PL...24	2.5 V		12m...lt	1 (...lt)										
vga_g[4]	Output	PL...24	4A	B4A_NO	PL...24	2.5 V		12m...lt	1 (...lt)										
vga_g[3]	Output	PL...24	4A	B4A_NO	PL...24	2.5 V		12m...lt	1 (...lt)										
vga_g[2]	Output	PL...25	4A	B4A_NO	PL...25	2.5 V		12m...lt	1 (...lt)										
vga_g[1]	Output	PL...25	4A	B4A_NO	PL...25	2.5 V		12m...lt	1 (...lt)										
vga_g[0]	Output	PL...26	4A	B4A_NO	PL...26	2.5 V		12m...lt	1 (...lt)										
vga_r[7]	Output	PL...26	4A	B4A_NO	PL...26	2.5 V		12m...lt	1 (...lt)										
vga_r[6]	Output	PL...26	4A	B4A_NO	PL...26	2.5 V		12m...lt	1 (...lt)										
vga_r[5]	Output	PL...26	4A	B4A_NO	PL...26	2.5 V		12m...lt	1 (...lt)										
vga_r[4]	Output	PL...27	4A	B4A_NO	PL...27	2.5 V		12m...lt	1 (...lt)										

0% 00:00:00

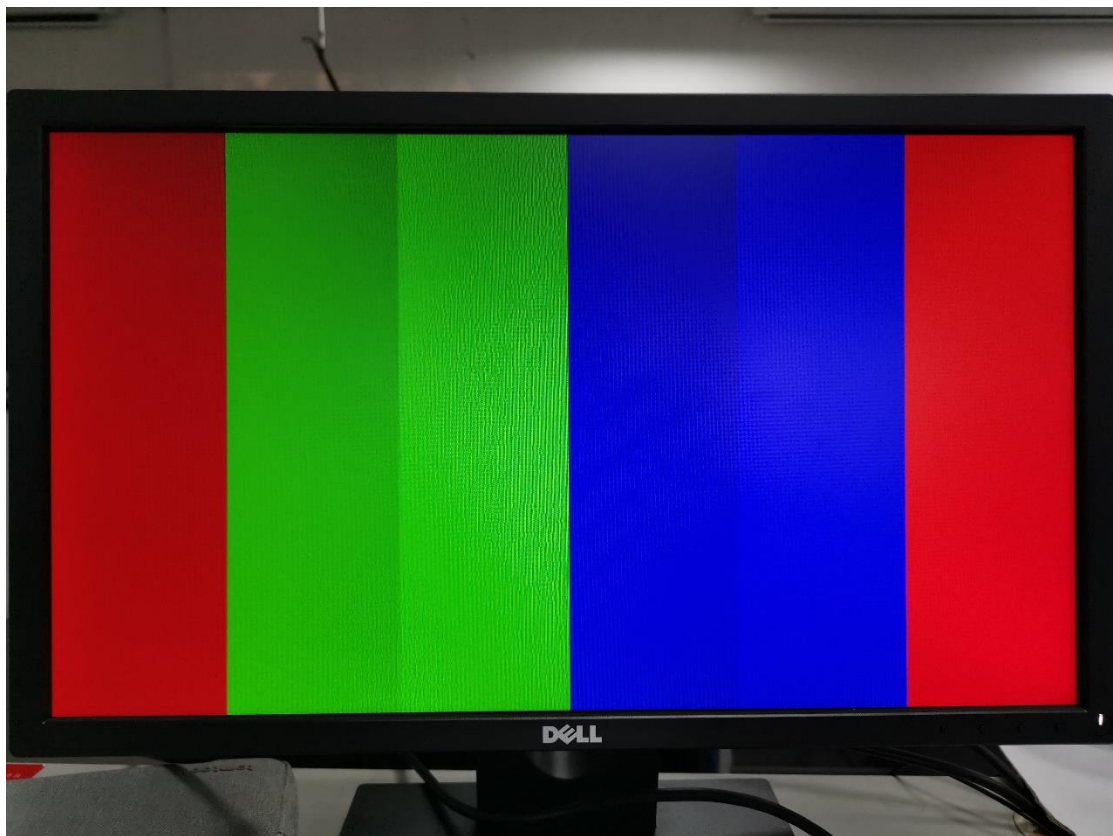
仿真：无

写入:





硬件验证：







## 六、测试方法

该实验无法使用 test bench 进行仿真验证，故决定通过显示器上不同的反馈来推断 bug 的可能位置。

## 七、实验结果

经过多次的调整后，实际结果与预期结果完全符合：

显示器上出现六条颜色不同的条纹、一个北大楼的图片、一个随机反复反弹的自定义图片。

## 八、实验中遇到的问题及解决办法

### 1、图片显示不完全或者颜色不对或者内容不正确

解决办法：

- ①图片显示不完全，则考虑长度和宽度是否设置正确，或 ROM 的大小设置错误；
- ②图片颜色不对，则考虑 ROM 字长设置错误，或 mif 文件字长设置错误；
- ③图片内容不正确，则考虑图片长度宽度设置错误或 ROM 大小设置错误。

### 2、动图的内容不正确或颜色有误

解决办法：最初生成 mif 文件的方法是用 Photoshop 生成 BMP 格式的图片，然后用 Bmp2Mif 软件将 BMP 图片转化为 mif 文件，但 Photoshop 只能生成字长至少为 16 位的图片，而要求是 12 位，若生成不同字长的图片，则会导致动图颜色有误，若原始图片不为正方形，则会导致动图内容不正确，故舍弃该方法。后来采用 Matlab 生成 mif 文件。

### 3、显示动图时有时候屏幕边缘可能会出现不同颜色的细边框

解决办法：在更改图片长度和宽度时，代码中与长宽度有关的计算也要相应更改。

## 九、实验得到的启示

- 1、要学会从实验结果的差异来推断实验过程中出现的问题，例如根据图片内容不正确，图片颜色有误，屏幕出现边框，无法显示图片等等现象来推断代码的 bug 出现位置。
- 2、要有阅读他人代码然后进行修改添加的能力，对于已有的框架，要充分理解后合理运用，不可盲目增添一些无用或重复的功能。

## 十、意见和建议

- 1、讲义中对于代码的解析感觉比较混乱，建议可以先用流程图的方式将代码过程分为不同模块，了解大致步骤后，再分模块进行讲解，讲解时可以采用序号的形式分步骤解析。
- 2、建议可以给出实验结果的效果图或视频，不然可能无法理解动图的触壁反弹的具体要求和效果。