**计算机组成原理 实验报告**

姓名： 杨佳熹 学号：PB17000050 实验日期：2019-4-18

**一、实验题目：**

Lab4 存储器与显示控制器

**二、实验目的：**

设计一基于存储器，可以通过按键来改变其内容，并且通过vga接口在屏幕上显示。

**三、实验平台：**

Vivado

**四、实验过程：**

本次实验主要用了两个模块，一个是vga扫描模块另一个时pcu模块，也是接引脚的模块。

PCU模块：

module PCU(rst,clk,rgb,dir,draw,vr,vg,vb,hs,vs);

input rst;

input clk;

input [11:0] rgb;

input [3:0] dir;

input draw;

output [3:0]vr,vg,vb;

output hs,vs;

reg [5:0] t;

reg [15:0] paddr;

reg [11:0] pdata;

wire [11:0]dpo;

wire [15:0] x,y;

wire we;

wire clk\_50m;

wire rst\_n;

wire [15:0]dpra;

wire en;

reg flag;

reg clk\_1p;

reg [31:0]cnt;

wire [10:0] px,py;

reg state;

reg flag;

clk\_wiz\_0 clk\_wiz\_0(

.clk\_in1 (clk),

.clk\_out1 (clk\_50m),

.reset (rst),

.locked (rst\_n)

);

dist\_mem\_gen\_1 dist\_mem\_gen\_1(

.a (paddr),

.d (rgb),

.dpra (dpra),

.clk (clk\_50m),

.we (draw),

.dpo (dpo)

);

VGA dcu (clk\_50m,rst,hs,vs,x,y);

assign dpra=256\*y+x;

assign {vr,vg,vb}= ((dpra<=paddr+3)&&(dpra>=paddr-3)||(dpra==paddr+256\*3)||(dpra==paddr+256\*2)||(dpra==paddr+256\*1)||(dpra==paddr-256\*3)||(dpra==paddr-256\*2)||(dpra==paddr-256\*1))? 12'b111111111111:dpo;

always @ (posedge clk\_50m or posedge rst)

begin

if(rst)

begin

cnt=0;

clk\_1p=0;

end

else if(cnt==500000)

begin

clk\_1p=~clk\_1p;

cnt=0;

end

else

begin

cnt=cnt+1;

end

end

always @ (posedge clk\_1p)

begin

if(dir==4'b0000)

begin

t<=0;

end

else if(t<10)

t<=t+1;

end

always @ (t)

begin

if(t==10)

state=1;

else

begin

state=0;

end

end

always @ (posedge clk\_1p or posedge rst)

begin

if(rst)

begin

paddr<=256\*128+128;

flag=1;

end

else if(dir==4'b0000)

flag=1;

else if(state)

begin

case(dir)

4'b0001:paddr<=paddr-1;

4'b0010:paddr<=paddr-256;

4'b0100:paddr<=paddr+1;

4'b1000:paddr<=paddr+256;

4'b0011:paddr<=paddr-257;

4'b0110:paddr<=paddr-255;

4'b1100:paddr<=paddr+257;

4'b1001:paddr<=paddr+255;

endcase

end

else if(flag&&!state)

begin

case(dir)

4'b0001:paddr<=paddr-1;

4'b0010:paddr<=paddr-256;

4'b0100:paddr<=paddr+1;

4'b1000:paddr<=paddr+256;

4'b0011:paddr<=paddr-257;

4'b0110:paddr<=paddr-255;

4'b1100:paddr<=paddr+257;

4'b1001:paddr<=paddr+255;

endcase

flag=0;

end

end

endmodule

声明部分含有控制按键、时钟、复位、行有效、列有效、颜色，并且对将对应的接口赋值；之后是对ip核进行调用，ip核采用dist\_mem\_gen,由与coe文件generate比较慢，所以跳过coe这一步，运用defaultcoe进行赋值。这一部分对时间频率进行了更改，一个是用于vga的50mhz和用于写的10hz。根据实验要求，写的实现需要先加入一定的延迟，所以代码中写入两个状态，分别为按键有效小于一秒和大于一秒；状态一只执行一次写，状态二执行连续写。接下来是写入方向的实现，在10hz的时钟上升沿进入always部分，并写一个多路选择判断上左下右以及斜向进而改变存储器读取地址。

VGA模块：

module VGA (

input clk, rst,

output hs, vs,

output [15:0] x, y

);

wire en;

parameter HD = 800, HF = 56, HS = 120, HB = 64;

parameter VD = 600, VF = 37, VS = 6, VB = 23;

reg [15:0] hc, vc;

assign x = en ? hc : 0;

assign y = en ? vc : 0;

assign en = (hc < 712) && (vc < 457) && (hc > 456) && (vc > 201);

assign hs = (hc >= HS) ;

assign vs = (vc >= VS );

// assign en = (hc < HD) && (vc < VD);

// assign hs = ~((hc >= HD + HF) && (hc < HD + HF + HS));

// assign vs = ~((vc >= VD + VF) && (vc < VD + VF + VS));

always @ (posedge clk or posedge rst)

if(rst) hc <= 15'd0;

else if(hc == 15'd1039) hc <= 15'd0;

else hc <= hc+1'b1;

always @ (posedge clk or posedge rst)

if(rst) vc <= 15'd0;

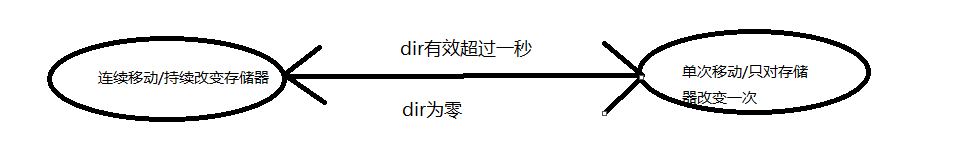
else if(vc == 15'd665) vc <= 15'd0;

else if(hc == 15'd1039) vc <= vc+1'b1;

endmodule

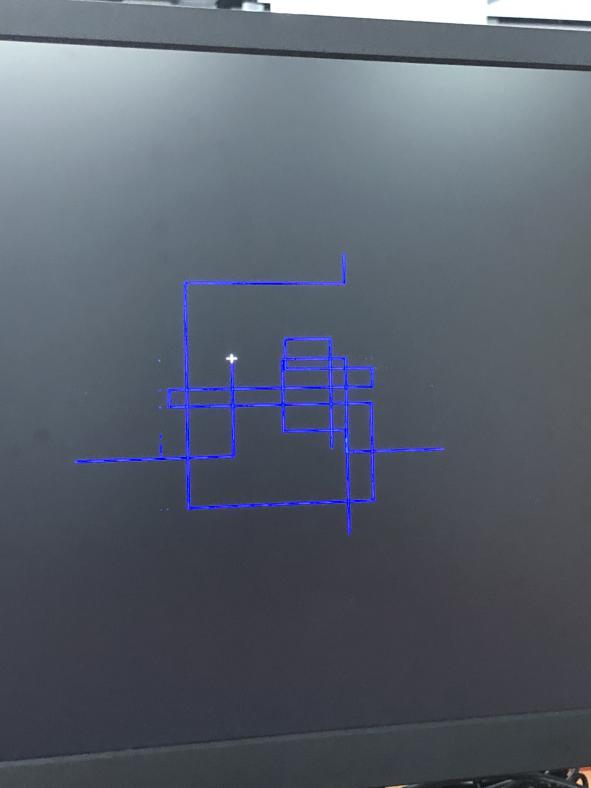
VGA的原理是扫描整个屏幕，并且输出是扫描点的12位颜色。所以这个模块主要是对屏幕的扫描，对照参数表，调好分辨率和有效区域，并且返回当时扫描的位置，参数的赋值主要参考老师所给ppt。

附加内容的实现：这次的附加内容是连续移动，斜向移动和将画笔位置做成十字。连续移动主要是通过切换两个状态来判断是只走一个像素点还是dir有效超过一秒且连续走，状态切换如下图。



斜向移动较为容易，就是除了上下左右再加入斜向的选择通路，并对存储器写入。十字的实现需要基于vga显示原理，一般的是将vga返回的坐标所对应的数据直接输出产生颜色，而这里是判断显示坐标师傅在画笔的上下左右三个像素以内，若在则显示十字颜色，我这里设置为白色。

**五、实验结果：**



这里接入机房的显示器验证下载结果，只需把显示屏的vga线接入到板子上的vga接口即可。

**六、心得体会：**

这次试验较前几次实验难度有提高，特别是vga显示部分，需要先把显示原理弄明白，包括行有效和列有效的取值，和分辨率的调整。其次是由于这次实验用了ip核，含有65536位地址，所以生成时间很长，这在调试时非常痛苦，需要等很久才能看到结果。总体来说实验过程较为顺利，只要把原理搞懂实现不会太难。