Breakout project

Interface Documentation

Project designed by.

Ehsan Emad Marvasti.

Amir Emad Marvasti.

1. VGA interface Documentation
2. PS2 interface Documentation
3. Audio interface Documentation

Part i. VGA interface:

For a better understanding of this part the code is shown below.the base idea of this module is repainting the screen when position or value of objects change.it begins to repaint when “startrepaint” changes to high value. In the fig below we show the screenshot of the game .

**40 PX**

**12:43**

**3**

5423

**120 PX**

module vga(input reset,input [2:0] level, input [7:0] min , input [7:0] sec, input [15:0] gamepoint,input [7:0] ballx,platex,input [6:0] bally,platey,input [5:0] ballsize,platesize,input clk,output reg [7:0] x,output reg [6:0] y,output reg plot,output reg [2:0] color);

reg startrepaint; // This reg indicates the beginning of repainting procedure

reg [7:0] painterlocx;// we have a reg called painterlocx and painterlocy which is a pointer that changes every clock and iterate the screen. These values iterate the screen when startrepaint is high.

reg [6:0] painterlocy;

reg repaintdone;//unimportant reg

reg [7:0] lastballx,lastplatex;// indicates last position of ball

reg [6:0] lastbally,lastplatey;// indicates last position of plate

reg [6:0] sevenseg [8:0];// in the screen we have 9 digit (2 digit for second , 2 digit for minute, 1 digit for level,4 digit gamepoint)we have used seven segment technique for showing digits.

**0**

2 1

3

6 4

5

All the numbers are the value of a segment for example 5 is 0111101 so if (sevenseg[0]==0111101) then first digit of minute is 5

function m (input [3:0] seseg,input [7:0] x,sx, input [6:0] y,sy); // shows if x is in sevenseg[seseg] area and the x and y pixel must be on

begin

m=0;

if(sevenseg[seseg][0]==1) // if this value ==1 then sevenseg[seseg][0] is on so if x and y are in the range of that seven segment’s segment the output of the function must change to 1

begin

if(sx<=x&x<=sx+5& y==sy)

m=1;

end

if(sevenseg[seseg][1]==1)

begin

if(sx+5==x&y<=sy+5& y>=sy)

m=1;

end

if(sevenseg[seseg][2]==1)

begin

if(sx==x&y<=sy+5& y>=sy)

m=1;

end

if(sevenseg[seseg][3]==1)

begin

if(sx<=x&x<sx+5& y==sy+5)

m=1;

end

if(sevenseg[seseg][4]==1)

begin

if(sx+5==x&y<=sy+10& y>=sy+5)

m=1;

end

if(sevenseg[seseg][5]==1)

begin

if(sx<=x&x<sx+5& y==sy+10)

m=1;

end

if(sevenseg[seseg][6]==1)

begin

if(sx==x&y<=sy+10& y>=sy+5)

m=1;

end

end

endfunction

always @(posedge clk)

begin

if(reset==1)// initializing the values when reset is high

begin

lastballx=0;

lastbally=0;

lastplatex=0;

lastplatey=0;

plot=0;

startrepaint=0;

repaintdone=1;

painterlocx=0;

painterlocy=0;

end

else

begin

case (min[7:4]) // set the corresponding sevensegment to case value

4'd0:sevenseg[0]=119;

4'd1:sevenseg[0]=18;

4'd2:sevenseg[0]=107;

4'd3:sevenseg[0]=59;

4'd4:sevenseg[0]=30;

4'd5:sevenseg[0]=61;

4'd6:sevenseg[0]=124;

4'd7:sevenseg[0]=19;

4'd8:sevenseg[0]=127;

4'd9:sevenseg[0]=63;

endcase

case (min[3:0])

4'd0:sevenseg[1]=119;

4'd1:sevenseg[1]=18;

4'd2:sevenseg[1]=107;

4'd3:sevenseg[1]=59;

4'd4:sevenseg[1]=30;

4'd5:sevenseg[1]=61;

4'd6:sevenseg[1]=124;

4'd7:sevenseg[1]=19;

4'd8:sevenseg[1]=127;

4'd9:sevenseg[1]=63;

endcase

case (sec[7:4])

4'd0:sevenseg[2]=119;

4'd1:sevenseg[2]=18;

4'd2:sevenseg[2]=107;

4'd3:sevenseg[2]=59;

4'd4:sevenseg[2]=30;

4'd5:sevenseg[2]=61;

4'd6:sevenseg[2]=124;

4'd7:sevenseg[2]=19;

4'd8:sevenseg[2]=127;

4'd9:sevenseg[2]=63;

endcase

case (sec[3:0])

4'd0:sevenseg[3]=119;

4'd1:sevenseg[3]=18;

4'd2:sevenseg[3]=107;

4'd3:sevenseg[3]=59;

4'd4:sevenseg[3]=30;

4'd5:sevenseg[3]=61;

4'd6:sevenseg[3]=124;

4'd7:sevenseg[3]=19;

4'd8:sevenseg[3]=127;

4'd9:sevenseg[3]=63;

endcase

case (level[2:0])

3'd0:sevenseg[4]=119;

3'd1:sevenseg[4]=18;

3'd2:sevenseg[4]=107;

3'd3:sevenseg[4]=59;

3'd4:sevenseg[4]=30;

3'd5:sevenseg[4]=61;

3'd6:sevenseg[4]=124;

3'd7:sevenseg[4]=19;

endcase

case (gamepoint[15:12])

4'd0:sevenseg[5]=119;

4'd1:sevenseg[5]=18;

4'd2:sevenseg[5]=107;

4'd3:sevenseg[5]=59;

4'd4:sevenseg[5]=30;

4'd5:sevenseg[5]=61;

4'd6:sevenseg[5]=124;

4'd7:sevenseg[5]=19;

4'd8:sevenseg[5]=127;

4'd9:sevenseg[5]=63;

endcase

case (gamepoint[11:8])

4'd0:sevenseg[6]=119;

4'd1:sevenseg[6]=18;

4'd2:sevenseg[6]=107;

4'd3:sevenseg[6]=59;

4'd4:sevenseg[6]=30;

4'd5:sevenseg[6]=61;

4'd6:sevenseg[6]=124;

4'd7:sevenseg[6]=19;

4'd8:sevenseg[6]=127;

4'd9:sevenseg[6]=63;

endcase

case (gamepoint[7:4])

4'd0:sevenseg[7]=119;

4'd1:sevenseg[7]=18;

4'd2:sevenseg[7]=107;

4'd3:sevenseg[7]=59;

4'd4:sevenseg[7]=30;

4'd5:sevenseg[7]=61;

4'd6:sevenseg[7]=124;

4'd7:sevenseg[7]=19;

4'd8:sevenseg[7]=127;

4'd9:sevenseg[7]=63;

endcase

case (gamepoint[3:0])

4'd0:sevenseg[8]=119;

4'd1:sevenseg[8]=18;

4'd2:sevenseg[8]=107;

4'd3:sevenseg[8]=59;

4'd4:sevenseg[8]=30;

4'd5:sevenseg[8]=61;

4'd6:sevenseg[8]=124;

4'd7:sevenseg[8]=19;

4'd8:sevenseg[8]=127;

4'd9:sevenseg[8]=63;

endcase

if(ballx!=lastballx || platex!=lastplatex || bally!=lastbally || platey!=lastplatey) // if positions change we set plot to 1 and and we start to repaint the screen

begin

plot=1;

startrepaint=1;

repaintdone=0;

painterlocx=0;painterlocy=0;

lastballx=ballx ; lastplatex=platex ; lastbally=lastbally ; lastplatey=platey; // we set last position to current position just because we don’t want this if to be called every clock pulse.

end

if(startrepaint==1)// when start repaint is set to one this if is true so the painting begins

begin

x=painterlocx;

y=painterlocy;

if(painterlocx>=ballx&painterlocx<=ballx+ballsize&painterlocy>=bally&painterlocy<=bally+ballsize)// if paintelocr is in the area of ball we set the color to zero

color=0;

else

begin

if(painterlocx>=platex&painterlocx<=platex+platesize&painterlocy>=105)// if painter is in the area of plate we set the color to 110

color=3'b110;

else

color=3'b111;

end

if(m(0,painterlocx,125, painterlocy,8)==1)// if m ==1 then the point’s color must be changed because as we said before this function indicates if the pointer is in the specific sevensegment and if that segment is on

color=3'b000;

if(m(1,painterlocx,132, painterlocy,8)==1)

color=3'b000;

if(m(2,painterlocx,143, painterlocy,8)==1)

color=3'b000;

if(m(3,painterlocx,150, painterlocy,8)==1)

color=3'b000;

if(m(4,painterlocx,135, painterlocy,50)==1)

color=3'b000;

if(m(5,painterlocx,125, painterlocy,80)==1)

color=3'b000;

if(m(6,painterlocx,132, painterlocy,80)==1)

color=3'b000;

if(m(7,painterlocx,143, painterlocy,80)==1)

color=3'b000;

if(m(8,painterlocx,150, painterlocy,80)==1)

color=3'b000;

if(painterlocx==121)// it’s a vertical line that seprates the sevensegment part from the game plain

color=3'b0;

if(painterlocx>=137&painterlocx<=139&painterlocy>=10&painterlocy<=12) // between the minute and second’s seven segment we paint a colon just for the traditional time illustration( 12 : 42)

color=3'b0;

if(painterlocx>=137&painterlocx<=139&painterlocy>=14&painterlocy<=16)

color =3'b0;

if(painterlocx==8'd160)// this part just makes sure that the painterloc register doesn’t cross the screen x and y limit and increase painterloc for the next clock;

begin

if(painterlocy==7'd120)

begin

startrepaint=0;

repaintdone=1;

plot=0;

painterlocy=0;

painterlocx=0;

end

else

begin

painterlocy=painterlocy+1;

painterlocx=0;

end

end

else

begin

painterlocx=painterlocx+1;

end

end

end

end

endmodule

Part ii. PS2 interface:

As you’ll see the code is like this

module ps2\_processor(input tx\_start , input [3:0] Data ,input clk, output reg right, left,reset);

always @(posedge clk)

begin

if(tx\_start==1)

begin

if(Data==6)

begin

right=1;

left=0;

reset=0;

end

if(Data==4)

begin

right=0;

left=1;

reset=0;

end

if(Data==1)

begin

reset=1;

right=0;

left=0;

end

end

else

begin

right=0;

reset=0;

left=0;

end

end

endmodule

analyzing of this code is very simple.

Part iii. Audio interface:

module Audio (input clk,input reset, input Hit\_wall , Hit\_ground ,Hit\_plate,Data\_request , output reg Data\_ready , output reg [3:0] sound\_code);

reg beginwall;

reg beginground;

reg beginplate;

reg [1:0] soundmark;//for playing 3 notes for hitting

reg [3:0] hitplate [2:0];// three sound\_code of hitplate

reg [3:0] hitground [2:0];

reg [3:0] hitwall [2:0];

always @(posedge clk)

begin

if(reset==1)

begin

beginwall=0;

beginground=0;

beginplate=0;

soundmark=0;

hitground[0]=4'b0111;// initializing sound of hitting ground

hitground[1]=4'b0101;

hitground[2]=4'b0001;

hitplate[0]=4'b0011;

hitplate[1]=4'b0110;

hitplate[2]=4'b0010;

hitwall[0]=4'b0111;

hitwall[1]=4'b0110;

hitwall[2]=4'b0101;

end

else

begin

if(soundmark==4)// soundmark ==4 stop playing notes

begin

beginground=0;

beginwall=0;

beginplate=0;

soundmark=0;

end

if(Hit\_ground==1)// if hit\_ground input ==1 then start playing hitting ground sound this sound contains three notes

begin

beginground=1;

beginwall=0;

end

if(Hit\_plate==1)

begin

beginplate=1;

beginwall=0;

end

if(Hit\_wall==1)

begin

beginwall=1;

beginplate=0;

end

if(Data\_request==1)

begin

if(beginground==1&&Data\_ready==0)

begin

sound\_code=hitground[soundmark];

soundmark=soundmark+1;

Data\_ready=1;

end

if(beginwall==1)

begin

sound\_code=hitwall[soundmark];

soundmark=soundmark+1;

Data\_ready=1;

end

if(beginplate==1)

begin

sound\_code=hitplate[soundmark];

soundmark=soundmark+1;

Data\_ready=1;

end

end

else

Data\_ready=0;

end

end

endmodule