`timescale 1ns / 1ps

module top(

input clk,

input[4:0] button,

input reset,

output [6:0] display,

output [5:0] AN,

output [4:0] led

);

wire clk1hz, clk1khz, gameover;

wire [4:0] bout;

wire [4:0] timecount;

wire [5:0] score\_count;

wire [3:0] num;

wire [9:0] bcdtime;

wire [9:0] bcdscore;

wire [4:0] ledin;

slow\_clk clk1 (.in\_clk(clk), .out\_clk(clk1hz));

fast\_clk clk2 (.in\_clk(clk), .out\_clk(clk1khz));

//debouncer deb(.bout(bout[0]), .clk(clk), .reset(reset) , .bin(button[0]));

//debouncer deb1(.bout(bout[1]), .clk(clk), .reset(reset) , .bin(button[1]));

//debouncer deb2(.bout(bout[2]), .clk(clk), .reset(reset) , .bin(button[2]));

//debouncer deb3(.bout(bout[3]), .clk(clk), .reset(reset) , .bin(button[3]));

//debouncer deb4(.bout(bout[4]), .clk(clk), .reset(reset) , .bin(button[4]));

random ran(.clk(clk1hz), .reset(reset), .gameover(gameover), .rand(led));

//always @ (posedge clk)begin

//if ((button == ledin) || gameover)begin

// led <= 0;

//end

//end

timer30 tim(.clk(clk1hz),.reset(reset),.count(timecount),.gameover(gameover));

score\_counter score(.clk(clk1hz), .reset(reset), .button(button),.gameover(gameover), .ledin(led), .score\_count(score\_count));

bcdconverter bcd1(.binary({3'b000, timecount}), .hundreds(bcdtime[9:8]),.tens(bcdtime[7:4]),.ones(bcdtime[3:0]));

bcdconverter bcd2(.binary({2'b00,score\_count}),.hundreds(bcdscore[9:8]),.tens(bcdscore[7:4]),.ones(bcdscore[3:0]));

display\_control dispctrl(.anode(AN),.num(num), .onesa(bcdtime[3:0]),.onesb(bcdscore[3:0]),.tensa(bcdtime[7:4]),.tensb(bcdscore[7:4]),.clk(clk1khz),.reset(reset));

seven\_segment seg(.sev(display),.num(num));

endmodule

`timescale 1ns / 1ps

module debouncer(

input clk,

input reset,

input bin,

output reg bout

);

reg [15:0] deb\_count;

reg output\_exist;

initial begin

bout = 0;

output\_exist = 0;

deb\_count = 16'b0000000000000000;

end

always @ (posedge clk) begin

bout <= 1'b0;

if(~reset) begin

output\_exist <= 1'b0;

deb\_count <= 16'b0000000000000000;

end

else if (bin && (~output\_exist)) begin

deb\_count <= deb\_count + 1'b1;

if(deb\_count == 16'b1111111111111111)begin

bout <= 1'b1;

output\_exist <= 1'b1;

deb\_count <= 16'b0000000000000000;

end

end

end

endmodule

`timescale 1ns / 1ps

module score\_counter(

input clk,

input[4:0] button,

input reset,

input gameover,

input[4:0] ledin,

output reg [5:0] score\_count=0

);

always @(button) begin

if(~reset)begin

score\_count <= 0;

end

//five =0 enable for score counter

if(~gameover) begin

if(button == ledin) begin

if(score\_count <= 6'b111111)

score\_count <= score\_count + 1;

end

end

else if (gameover)

score\_count = 0;

end

endmodule

`timescale 1ns / 1ps

module random(

input clk,

input reset,

input gameover,

output reg [4:0] rand = 5'b11111

);

reg [2:0] ran=3'b111;

reg [2:0] randh=3'b111;

wire tap = randh[2]^randh[1];

always@(posedge clk)

begin

if (gameover)

randh =3'b000;

else if(~reset)begin

rand = 5'b00000;

randh = 3'b111;

end

randh = {randh[1:0], tap};

if ((randh<6)&&(randh>0)&& (reset))

ran <= randh-1;

else if ((randh>=6)&& reset && (randh>7))

randh = randh+1;

else if ((randh>=6)&& reset && (randh==7))

randh= randh-1;

case(ran)

0: rand = 5'b00001;

1: rand = 5'b00010;

2: rand = 5'b00100;

3: rand = 5'b01000;

4: rand = 5'b10000;

endcase

end

endmodule

`timescale 1ns / 1ps

module timer30(

input clk,

input reset,

output reg [4:0] count, //30 second timer

output reg gameover = 0

);

reg[4:0] current\_count=5;

reg five = 1;

always @ (posedge clk) begin

if(~reset)begin

five = 1;

current\_count=5;

end

if(five && (current\_count==0))begin

current\_count<=30;

gameover = 0;

five =0;

end

else if(current\_count == 0)begin

current\_count<=0;

gameover = 1;

end

else if(current\_count >= 1)begin

current\_count<=current\_count -1;

end

else

current\_count <= current\_count;

count<=current\_count;

end

endmodule