module wending(clock,sw1,sw2,sw3,sw4,z1,z2,z3,r,d1,d0,l1,l2,l3,led0,led5,led10,led15);

input sw1,sw2,sw3,sw4,clock,z1,z2,z3,r;

output [7:0] d1,d0;

output l1,l2,l3,led0,led5,led10,led15;

reg [7:0] d1,d0;

reg [3:0] count;

wire clock;

//reg l1,l2,l3,led0,led5,led10,led15;

reg [32:0] counter;

reg [7:0] s;

reg [6:0] cnt;

reg counter1,counter2,counter3,counter4;

initial begin

cnt<=7'b0000000;

count<=4'b0000;

/\*l1=0;

l2=0;

l3=0;

led0=0;

led5=0;

led10=0;

led15=0;\*/

end

parameter

s0= 8'b00000000,

s5= 8'b00000001,

s10=8'b00001011,

s15=8'b00000010,

s20=8'b00000011,

s25=8'b00000100,

s30=8'b00000101,

s35=8'b00000110,

s40=8'b00000111,

s45=8'b00001000,

s50=8'b00001001,

s99=8'b11111111;

localparam

sec2 = 4'd8;

always @ (posedge clock)

begin

counter <= counter + 1;

end

always @ (posedge counter[23])

begin

//sw1 5cent

if(sw1 == 0)

begin

if(counter1==0)

begin

cnt <= cnt + 7'b0000101;

counter1 <= 1;

end

end

if(sw1 == 1)

counter1<=0;

//sw2 10 cent

if(sw2 == 0)

begin

if(counter2==0)begin

cnt <= cnt + 7'b0001010;

counter2 <= 1;

end

end

if(sw2 == 1)

counter2<=0;

//sw3 25 cent

if(sw3 == 0)

begin

if(counter3==0)begin

cnt <= cnt + 7'b0011001;

counter3 <= 1;

end

end

if(sw3 == 1)

counter3<=0;

//sw3 50 cent

if(sw4 == 0)

begin

if(counter4==0)begin

cnt <= cnt + 7'b0110010;

counter4 <= 1;

end

end

if(sw4 == 1)

counter4<=0;

//reset cnt

if(z1==1)

begin

count<=0;

cnt<=0;

end

if(z2==1)

begin

count<=0;

cnt<=0;

end

if(z3==1)

begin

count<=0;

cnt<=0;

end

if(r==0)

begin

count<=0;

cnt<=0;

end

//check cnt and assign state s

if(cnt==7'b0000000)

begin

s <=s0;

end

else if (cnt==7'b0000101)

begin

s <=s5;

end

else if (cnt==7'b0001010)

begin

s <=s10;

end

else if (cnt==7'b0001111)

begin

s <=s15;

end

else if (cnt==7'b0010100)

begin

s <=s20;

end

else if (cnt==7'b0011001)

begin

s <=s25;

end

else if (cnt==7'b0011110)

begin

s <=s30;

end

else if (cnt==7'b0100011)

begin

s <=s35;

end

else if (cnt==7'b0101000)

begin

s <=s40;

end

else if (cnt==7'b0101101)

begin

s <=s45;

end

else if (cnt==7'b0110010)

begin

s <=s50;

end

else if(cnt>=7'b0110111)

begin

s <=s99;

end

end

always @ (\*)

begin

case (s)

s0 : begin d1<=8'b11000000; d0<=8'b11000000;led0=0;led5=0; led10=0;led15=0;l1=0;l2=0;l3=0;end

s5: begin d1<=8'b11000000; d0<=8'b10010010;/\*led0=0;led5=0; led10=0;led15=0;l1=0;l2=0;l3=0;\*/ end

s10: begin d1<=8'b11111001; d0<=8'b11000000;/\*led0=0;led5=0; led10=0;led15=0;l1=0;l2=0;l3=0;\*/end

s15: begin d1<=8'b11111001; d0<=8'b10010010;/\*led0=0;led5=0; led10=0;led15=0;l1=0;l2=0;l3=0;\*/ end

s20: begin d1<=8'b10100100; d0<=8'b11000000;/\*led0=0;led5=0; led10=0;led15=0;l1=0;l2=0;l3=0;\*/ end

s25: begin d1<=8'b10100100; d0<=8'b10010010;/\*led0=0;led5=0; led10=0;led15=0;l1=0;l2=0;l3=0;\*/ end

s30: begin d1<=8'b10110000; d0<=8'b11000000;/\*led0=0;led5=0; led10=0;led15=0;l1=0;l2=0;l3=0;\*/ end

s35: begin

if(z1==1)

begin

d1<=8'b11000000;

d0<=8'b11000000;

led0=1;

l1=1;

end

else begin

d1<=8'b10110000;

d0<=8'b10010010;

end

end

s40: begin

if(z1==1)

begin

d1<=8'b11000000;

d0<=8'b10010010;

led5=1;

l1=1;

end

else begin

d1<=8'b10011001;

d0<=8'b11000000;

end

end

s45: begin

if(z1==1)

begin

d1<=8'b11111001;

d0<=8'b11000000;

led10=1;

l1=1;

end

else if(z2==1)

begin

d1<=8'b11000000;

d0<=8'b11000000;

led0=1;

l2=1;

end

else if(z3==1)

begin

d1<=8'b11000000;

d0<=8'b11000000;

led0=1;

l3=1;

end

else

begin

d1<=8'b10011001;

d0=8'b10010010;

end

end//state

s50: begin

if(z1==1)

begin

d1<=8'b11111001;

d0<=8'b10010010;

l1=1;

led15=1;

end

else if(z2==1)

begin

d1<=8'b11000000;

d0<=8'b11000000;

led5=1;

l2=1;

end

else if(z3==1)

begin

d1<=8'b11000000;

d0<=8'b10010010;

led5=1;

l3=1;

end

else

begin

d1<=8'b10010010;

d0<=8'b11000000;

end

end

s99 : //errror message

begin

d1<=8'b10000110;

d0<=8'b10001000;

end

endcase

end

endmodule