DSD 第 51 組 Final Project

107504507 吳葦誠

108504502 王珮榕

Project 名稱:卡比獸勇闖尖刺森林

1.Source Code

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company: IPEECS
// Engineer: WU WEICHENG & WANG PEIJUNG
//////////////////////////////////////////////////////////////////////////////////

module main(clk, neg_rst, hsync, vsync, vga_r, vga_g, vga_b, Mode, ps2_clock,
ps2_data, s2, s0, s3, LED, Enable, SevenSeg_Left, SevenSeg_Right);
    input                   clk;
    input                   neg_rst;
    input                   s3, s2, s0;
    input                   Mode;
    input                   ps2_clock;
    input                   ps2_data;

    output                  hsync,vsync;
    output [3:0]        vga_r, vga_g, vga_b;
    output [15:0]       LED;
    output [7:0]        Enable;
    output [7:0]        SevenSeg_Left, SevenSeg_Right;
    reg [15:0] LED;
    reg [ 7:0] Enable;
    reg [ 7:0] SevenSeg_Left, SevenSeg_Right;

    wire                  ctrlclk;
    wire                  valid;
    wire [9:0]          h_cnt,v_cnt;
    reg [11:0]          vga_data;
    wire [11:0]         snorlax_rom_dout;   //fix for 12 bits
    wire [11:0]         triangle1_1_rom_dout, triangle1_2_rom_dout,
triangle1_3_rom_dout,
    triangle1_4_rom_dout, triangle1_5_rom_dout, triangle1_6_rom_dout,
triangle1_7_rom_dout;
    wire [11:0]         triangle2_1_rom_dout, triangle2_2_rom_dout,
triangle2_3_rom_dout,
    triangle2_4_rom_dout, triangle2_5_rom_dout, triangle2_6_rom_dout,
triangle2_7_rom_dout, triangle2_8_rom_dout;
    wire [11:0]         fruit_rom_dout, cookie_rom_dout;

    reg [9:0]            cookie_rom_addr, fruit_rom_addr;
    reg [10:0]           snorlax_rom_addr;
    reg [11:0]           triangle1_1_rom_addr, triangle1_2_rom_addr,
triangle1_3_rom_addr
    , triangle1_4_rom_addr, triangle1_5_rom_addr, triangle1_6_rom_addr,
triangle1_7_rom_addr;
    reg [11:0]           triangle2_1_rom_addr, triangle2_2_rom_addr,
triangle2_3_rom_addr
    , triangle2_4_rom_addr, triangle2_5_rom_addr, triangle2_6_rom_addr,
triangle2_7_rom_addr, triangle2_8_rom_addr;

    wire                  snorlax_area;
    wire                  triangle1_1_area, triangle1_2_area, triangle1_3_area,
triangle1_4_area, triangle1_5_area, triangle1_6_area, triangle1_7_area;
    wire                  triangle2_1_area, triangle2_2_area, triangle2_3_area,
triangle2_4_area, triangle2_5_area, triangle2_6_area, triangle2_7_area,
triangle2_8_area;
    wire                  cookie_area, fruit_area;
    wire                  moving_stair_area, disappear_stair_area;
    wire                  rst;

    reg                   x_detect, y_detect, x_spike, y_spike;
    reg [9:0]           snorlax_x,snorlax_y, next_snorlax_x, next_snorlax_y;
    reg [9:0]           triangle1_1_x,triangle1_1_y, triangle1_2_x,triangle1_2_y,
triangle1_3_x,triangle1_3_y, triangle1_4_x,triangle1_4_y,
    triangle1_5_x,triangle1_5_y, triangle1_6_x,triangle1_6_y,
triangle1_7_x,triangle1_7_y;
```

```
    reg [9:0]          triangle2_1_x, triangle2_2_x, triangle2_3_x, triangle2_4_x,
triangle2_5_x, triangle2_6_x, triangle2_7_x, triangle2_8_x, triangle2_y;
    reg [9:0]          stair_1_x, stair_2_x, stair_3_x, stair_4_x, stair_5_x, stair_6_x,
stair_7_x, stair_8_x, stair_9_x, stair_10_x, stair_11_x,
                        stair_1_y, stair_2_y, stair_3_y, stair_4_y, stair_5_y,
stair_6_y, stair_7_y, stair_8_y, stair_9_y, stair_10_y, stair_11_y;
   reg [9:0]           cookie_x, cookie_y, fruit_x, fruit_y, moving_stair_x,
moving_stair_y, disappear_stair_x, disappear_stair_y;

    //reg [9:0] next_triangle1_1_x, next_triangle1_1_y, next_triangle1_2_x,
next_triangle1_2_y;

    parameter [9:0] logo_length=10'd40;
    parameter [9:0] logo_height=10'd30;
    parameter [9:0] triangle_length=10'd40;
    parameter [9:0] triangle_height=10'd60;

    reg [27:0] counter28;
    reg [9:0] Score;
    reg [3:0] Life_1, Life_2, Life_3, Life_4;
    reg [3:0]   Score_Hundred, Score_Ten, Score_Digits;
    reg [2:0] LED_counter, Die_counter;
    reg [2:0] final;

    reg cookie_eaten;
    reg fruit_eaten;

    reg [1:0] CS;
    reg [1:0] NS;
    reg [2:0] chance;
    reg on_Elevator, on_moving;
    reg drop;
    reg touch;
    reg on_disappear;
    reg [1:0] disappear_counter;
    reg disappear_la;

    parameter Stop = 2'd0, Movement = 2'd1, Falling = 2'd2, Die = 2'd3;
```

```
wire button_left, button_start, button_right;
wire [7:0] keyboard_input, s_out;

assign rst = !neg_rst;
assign {vga_r,vga_g,vga_b} = vga_data;

ps2 ps2_1(ps2_clock, ps2_data, neg_rst, keyboard_input); //detect keyboard
//keyboard(.clk(ps2_clock), .data(ps2_data), .led(s_out));

debounce_better_version d3(.pb_1(s3), .clk(clk), .pb_out(button_left)); //S3
debounce_better_version d2(.pb_1(s2), .clk(clk), .pb_out(button_start)); //S2
debounce_better_version d0(.pb_1(s0), .clk(clk), .pb_out(button_right)); //S0

dcm_25M u0(
        // Clock in ports
        .clk_in1(clk),          // input clk_in1
        // Clock out ports
        .clk_out1(ctrlclk),         // output clk_out1
        .reset(rst));

SyncGeneration u1 (
      .pclk(ctrlclk),
      .reset(rst),
      .hSync(hsync),
      .vSync(vsync),
      .dataValid(valid),
      .hDataCnt(h_cnt),
      .vDataCnt(v_cnt)
      );

snorlax_rom u2 (
        .clka(ctrlclk),       // input wire clka
        .addra(snorlax_rom_addr),    // input wire [9 : 0] addra
        .douta(snorlax_rom_dout)    // output wire [11 : 0] douta
      );

triangle1_rom t1_1 (
        .clka(ctrlclk),        // input wire clka
        .addra(triangle1_1_rom_addr),    // input wire [11 : 0] addra
        .douta(triangle1_1_rom_dout)    // output wire [11 : 0] douta
    );
triangle1_rom t1_2 (
        .clka(ctrlclk),        // input wire clka
        .addra(triangle1_2_rom_addr),    // input wire [11 : 0] addra
        .douta(triangle1_2_rom_dout)    // output wire [11 : 0] douta
    );
triangle1_rom t1_3 (
        .clka(ctrlclk),        // input wire clka
        .addra(triangle1_3_rom_addr),    // input wire [11 : 0] addra
        .douta(triangle1_3_rom_dout)    // output wire [11 : 0] douta
    );
triangle1_rom t1_4 (
        .clka(ctrlclk),        // input wire clka
        .addra(triangle1_4_rom_addr),    // input wire [11 : 0] addra
        .douta(triangle1_4_rom_dout)    // output wire [11 : 0] douta
    );
triangle1_rom t1_5 (
        .clka(ctrlclk),        // input wire clka
        .addra(triangle1_5_rom_addr),    // input wire [11 : 0] addra
        .douta(triangle1_5_rom_dout)    // output wire [11 : 0] douta
    );
triangle1_rom t1_6 (
        .clka(ctrlclk),        // input wire clka
        .addra(triangle1_6_rom_addr),    // input wire [11 : 0] addra
        .douta(triangle1_6_rom_dout)    // output wire [11 : 0] douta
    );
triangle1_rom t1_7 (
        .clka(ctrlclk),        // input wire clka
        .addra(triangle1_7_rom_addr),    // input wire [11 : 0] addra
        .douta(triangle1_7_rom_dout)    // output wire [11 : 0] douta
    );

triangle2_rom t2_1 (    //use for spikes
        .clka(ctrlclk),        // input wire clka
        .addra(triangle2_1_rom_addr),    // input wire [11 : 0] addra
        .douta(triangle2_1_rom_dout)    // output wire [11 : 0] douta
      );
triangle2_rom t2_2 (    //use for spikes
        .clka(ctrlclk),        // input wire clka
        .addra(triangle2_2_rom_addr),    // input wire [11 : 0] addra
        .douta(triangle2_2_rom_dout)    // output wire [11 : 0] douta
      );
triangle2_rom t2_3 (    //use for spikes
        .clka(ctrlclk),        // input wire clka
        .addra(triangle2_3_rom_addr),    // input wire [11 : 0] addra
        .douta(triangle2_3_rom_dout)    // output wire [11 : 0] douta
      );
triangle2_rom t2_4 (    //use for spikes
        .clka(ctrlclk),        // input wire clka
        .addra(triangle2_4_rom_addr),    // input wire [11 : 0] addra
        .douta(triangle2_4_rom_dout)    // output wire [11 : 0] douta
      );
triangle2_rom t2_5 (    //use for spikes
        .clka(ctrlclk),        // input wire clka
        .addra(triangle2_5_rom_addr),    // input wire [11 : 0] addra
        .douta(triangle2_5_rom_dout)    // output wire [11 : 0] douta
      );
triangle2_rom t2_6 (    //use for spikes
        .clka(ctrlclk),        // input wire clka
        .addra(triangle2_6_rom_addr),    // input wire [11 : 0] addra
        .douta(triangle2_6_rom_dout)    // output wire [11 : 0] douta
      );
triangle2_rom t2_7 (    //use for spikes
        .clka(ctrlclk),        // input wire clka
        .addra(triangle2_7_rom_addr),    // input wire [11 : 0] addra
        .douta(triangle2_7_rom_dout)    // output wire [11 : 0] douta
      );
triangle2_rom t2_8 (    //use for spikes
        .clka(ctrlclk),        // input wire clka
        .addra(triangle2_8_rom_addr),    // input wire [11 : 0] addra
        .douta(triangle2_8_rom_dout)    // output wire [11 : 0] douta
      );
cookie_rom c1 (
        .clka(ctrlclk),        // input wire clka
        .addra(cookie_rom_addr),    // input wire [11 : 0] addra
        .douta(cookie_rom_dout)    // output wire [11 : 0] douta
      );
fruit_rom f1 (
        .clka(ctrlclk),        // input wire clka
        .addra(fruit_rom_addr),    // input wire [11 : 0] addra
        .douta(fruit_rom_dout)    // output wire [11 : 0] douta
      );
assign snorlax_area = ((v_cnt >= snorlax_y) & (v_cnt <= snorlax_y + logo_height - 1) & (h_cnt >= snorlax_x) & (h_cnt <= snorlax_x + logo_length - 1)) ? 1'b1 : 1'b0;
assign triangle1_1_area = ((v_cnt >= triangle1_1_y) & (v_cnt <= triangle1_1_y + triangle_height - 1) & (h_cnt >= triangle1_1_x) & (h_cnt <= triangle1_1_x + triangle_length - 1)) ? 1'b1 : 1'b0;
assign triangle1_2_area = ((v_cnt >= triangle1_2_y) & (v_cnt <= triangle1_2_y + triangle_height - 1) & (h_cnt >= triangle1_2_x) & (h_cnt <= triangle1_2_x + triangle_length - 1)) ? 1'b1 : 1'b0;
assign triangle1_3_area = ((v_cnt >= triangle1_3_y) & (v_cnt <= triangle1_3_y + triangle_height - 1) & (h_cnt >= triangle1_3_x) & (h_cnt <= triangle1_3_x + triangle_length - 1)) ? 1'b1 : 1'b0;
assign triangle1_4_area = ((v_cnt >= triangle1_4_y) & (v_cnt <= triangle1_4_y + triangle_height - 1) & (h_cnt >= triangle1_4_x) & (h_cnt <= triangle1_4_x + triangle_length - 1)) ? 1'b1 : 1'b0;
assign triangle1_5_area = ((v_cnt >= triangle1_5_y) & (v_cnt <= triangle1_5_y + triangle_height - 1) & (h_cnt >= triangle1_5_x) & (h_cnt <= triangle1_5_x + triangle_length - 1)) ? 1'b1 : 1'b0;
assign triangle1_6_area = ((v_cnt >= triangle1_6_y) & (v_cnt <= triangle1_6_y + triangle_height - 1) & (h_cnt >= triangle1_6_x) & (h_cnt <= triangle1_6_x + triangle_length - 1)) ? 1'b1 : 1'b0;
assign triangle1_7_area = ((v_cnt >= triangle1_7_y) & (v_cnt <= triangle1_7_y + triangle_height - 1) & (h_cnt >= triangle1_7_x) & (h_cnt <= triangle1_7_x + triangle_length - 1)) ? 1'b1 : 1'b0;
assign triangle2_1_area = ((v_cnt >= triangle2_y) & (v_cnt <= triangle2_y + triangle_height - 1) & (h_cnt >= triangle2_1_x) & (h_cnt <= triangle2_1_x + triangle_length - 1)) ? 1'b1 : 1'b0;
assign triangle2_2_area = ((v_cnt >= triangle2_y) & (v_cnt <= triangle2_y + triangle_height - 1) & (h_cnt >= triangle2_2_x) & (h_cnt <= triangle2_2_x + triangle_length - 1)) ? 1'b1 : 1'b0;
```

```
	assign triangle2_3_area = ((v_cnt >= triangle2_y) & (v_cnt <= triangle2_y + triangle_height - 1) & (h_cnt >= triangle2_3_x) & (h_cnt <= triangle2_3_x + triangle_length - 1)) ? 1'b1 : 1'b0;
	assign triangle2_4_area = ((v_cnt >= triangle2_y) & (v_cnt <= triangle2_y + triangle_height - 1) & (h_cnt >= triangle2_4_x) & (h_cnt <= triangle2_4_x + triangle_length - 1)) ? 1'b1 : 1'b0;
	assign triangle2_5_area = ((v_cnt >= triangle2_y) & (v_cnt <= triangle2_y + triangle_height - 1) & (h_cnt >= triangle2_5_x) & (h_cnt <= triangle2_5_x + triangle_length - 1)) ? 1'b1 : 1'b0;
	assign triangle2_6_area = ((v_cnt >= triangle2_y) & (v_cnt <= triangle2_y + triangle_height - 1) & (h_cnt >= triangle2_6_x) & (h_cnt <= triangle2_6_x + triangle_length - 1)) ? 1'b1 : 1'b0;
	assign triangle2_7_area = ((v_cnt >= triangle2_y) & (v_cnt <= triangle2_y + triangle_height - 1) & (h_cnt >= triangle2_7_x) & (h_cnt <= triangle2_7_x + triangle_length - 1)) ? 1'b1 : 1'b0;
	assign triangle2_8_area = ((v_cnt >= triangle2_y) & (v_cnt <= triangle2_y + triangle_height - 1) & (h_cnt >= triangle2_8_x) & (h_cnt <= triangle2_8_x + triangle_length - 1)) ? 1'b1 : 1'b0;

	assign stair_1_area = ((v_cnt >= stair_1_y) & (v_cnt <= stair_1_y + 10'd10 - 1) & (h_cnt >= stair_1_x) & (h_cnt <= stair_1_x + 10'd120 - 1)) ? 1'b1 : 1'b0;
	assign stair_2_area = ((v_cnt >= stair_2_y) & (v_cnt <= stair_2_y + 10'd10 - 1) & (h_cnt >= stair_2_x) & (h_cnt <= stair_2_x + 10'd80 - 1)) ? 1'b1 : 1'b0;
	assign stair_3_area = ((v_cnt >= stair_3_y) & (v_cnt <= stair_3_y + 10'd10 - 1) & (h_cnt >= stair_3_x) & (h_cnt <= stair_3_x + 10'd80 - 1)) ? 1'b1 : 1'b0;
	assign stair_4_area = ((v_cnt >= stair_4_y) & (v_cnt <= stair_4_y + 10'd10 - 1) & (h_cnt >= stair_4_x) & (h_cnt <= stair_4_x + 10'd80 - 1)) ? 1'b1 : 1'b0;
	assign stair_5_area = ((v_cnt >= stair_5_y) & (v_cnt <= stair_5_y + 10'd10 - 1) & (h_cnt >= stair_5_x) & (h_cnt <= stair_5_x + 10'd80 - 1)) ? 1'b1 : 1'b0;
	assign stair_6_area = ((v_cnt >= stair_6_y) & (v_cnt <= stair_6_y + 10'd10 - 1) & (h_cnt >= stair_6_x) & (h_cnt <= stair_6_x + 10'd80 - 1)) ? 1'b1 : 1'b0;
	assign stair_7_area = ((v_cnt >= stair_7_y) & (v_cnt <= stair_7_y + 10'd10 - 1) & (h_cnt >= stair_7_x) & (h_cnt <= stair_7_x + 10'd120 - 1)) ? 1'b1 : 1'b0;
	assign stair_8_area = ((v_cnt >= stair_8_y) & (v_cnt <= stair_8_y + 10'd10 - 1) & (h_cnt >= stair_8_x) & (h_cnt <= stair_8_x + 10'd80 - 1)) ? 1'b1 : 1'b0;
	assign stair_9_area = ((v_cnt >= stair_9_y) & (v_cnt <= stair_9_y + 10'd10 - 1) & (h_cnt >= stair_9_x) & (h_cnt <= stair_9_x + 10'd80 - 1)) ? 1'b1 : 1'b0;
	assign stair_10_area = ((v_cnt >= stair_10_y) & (v_cnt <= stair_10_y + 10'd10 -
1) & (h_cnt >= stair_10_x) & (h_cnt <= stair_10_x + 10'd80 - 1)) ? 1'b1 : 1'b0;
    assign stair_11_area = ((v_cnt >= stair_11_y) & (v_cnt <= stair_11_y + 10'd10 -
1) & (h_cnt >= stair_11_x) & (h_cnt <= stair_11_x + 10'd40 - 1)) ? 1'b1 : 1'b0;
    assign cookie_area = ((v_cnt >= cookie_y) & (v_cnt <= cookie_y + logo_height -
1) & (h_cnt >= cookie_x) & (h_cnt <= cookie_x + logo_height - 1)) ? 1'b1 : 1'b0;
    assign fruit_area = ((v_cnt >= fruit_y) & (v_cnt <= fruit_y + logo_height - 1) &
(h_cnt >= fruit_x) & (h_cnt <= fruit_x + logo_height - 1)) ? 1'b1 : 1'b0;

    assign moving_stair_area = ((v_cnt >= moving_stair_y) & (v_cnt <=
moving_stair_y + 10'd10 - 1) & (h_cnt >= moving_stair_x) & (h_cnt <= moving_stair_x
+ 10'd120 - 1)) ? 1'b1 : 1'b0;
    assign disappear_stair_area = ((v_cnt >= disappear_stair_y) & (v_cnt <=
disappear_stair_y + 10'd10 - 1) & (h_cnt >= disappear_stair_x) & (h_cnt <=
disappear_stair_x + 10'd120 - 1)) ? 1'b1 : 1'b0;

    always @(posedge ctrlclk or posedge rst)
    begin: pic_display
        if (rst == 1'b1) begin
            snorlax_rom_addr <= 11'd0;
            triangle1_1_rom_addr <= 12'd0;
            triangle1_2_rom_addr <= 12'd0;
            triangle1_3_rom_addr <= 12'd0;
            triangle1_4_rom_addr <= 12'd0;
            triangle1_5_rom_addr <= 12'd0;
            triangle1_6_rom_addr <= 12'd0;
            triangle1_7_rom_addr <= 12'd0;
            triangle2_1_rom_addr <= 12'd0;
            triangle2_2_rom_addr <= 12'd0;
            triangle2_3_rom_addr <= 12'd0;
            triangle2_4_rom_addr <= 12'd0;
            triangle2_5_rom_addr <= 12'd0;
            triangle2_6_rom_addr <= 12'd0;
            triangle2_7_rom_addr <= 12'd0;
            triangle2_8_rom_addr <= 12'd0;
            vga_data <= 12'h000;
        end
        else begin
            if (valid == 1'b1) begin
```

```
if(stair_1_area == 1'b1) begin
        vga_data <= 12'h00F;
end
else if(stair_2_area == 1'b1) begin
        vga_data <= 12'h0F0;
end
else if(stair_3_area == 1'b1) begin
        vga_data <= 12'hF00;
end
else if(stair_4_area == 1'b1) begin
        vga_data <= 12'h0FF;
end
else if(stair_5_area == 1'b1) begin
        vga_data <= 12'hF0F;
end
else if(stair_6_area == 1'b1) begin
        vga_data <= 12'hFF0;
end
else if(stair_7_area == 1'b1) begin
        vga_data <= 12'h555;
end
else if(stair_8_area == 1'b1) begin
        vga_data <= 12'h123;
end
else if(stair_9_area == 1'b1) begin
        vga_data <= 12'h456;
end
else if(stair_10_area == 1'b1) begin
        vga_data <= 12'h789;
end
else if(stair_11_area == 1'b1) begin
        vga_data <= 12'habc;
end
else if (snorlax_area == 1'b1) begin
        snorlax_rom_addr <= snorlax_rom_addr + 11'd1;
        vga_data <= snorlax_rom_dout;
end
```

```
else if(triangle1_1_area == 1'b1) begin
        triangle1_1_rom_addr <= triangle1_1_rom_addr + 12'd1;
        vga_data <= triangle1_1_rom_dout;
end
else if(triangle1_2_area == 1'b1) begin
        triangle1_2_rom_addr <= triangle1_2_rom_addr + 12'd1;
        vga_data <= triangle1_2_rom_dout;
end
else if(triangle1_3_area == 1'b1) begin
        triangle1_3_rom_addr <= triangle1_3_rom_addr + 12'd1;
        vga_data <= triangle1_3_rom_dout;
end
else if(triangle1_4_area == 1'b1) begin
        triangle1_4_rom_addr <= triangle1_4_rom_addr + 12'd1;
        vga_data <= triangle1_4_rom_dout;
end
else if(triangle1_5_area == 1'b1) begin
        triangle1_5_rom_addr <= triangle1_5_rom_addr + 12'd1;
        vga_data <= triangle1_5_rom_dout;
end
else if(triangle1_6_area == 1'b1) begin
        triangle1_6_rom_addr <= triangle1_6_rom_addr + 12'd1;
        vga_data <= triangle1_6_rom_dout;
end
else if(triangle1_7_area == 1'b1) begin
        triangle1_7_rom_addr <= triangle1_7_rom_addr + 12'd1;
        vga_data <= triangle1_7_rom_dout;
end
else if(triangle2_1_area == 1'b1) begin
        triangle2_1_rom_addr <= triangle2_1_rom_addr + 12'd1;
        vga_data <= triangle2_1_rom_dout;
end
else if(triangle2_2_area == 1'b1) begin
        triangle2_2_rom_addr <= triangle2_2_rom_addr + 12'd1;
        vga_data <= triangle2_2_rom_dout;
end
else if(triangle2_3_area == 1'b1) begin
        triangle2_3_rom_addr <= triangle2_3_rom_addr + 12'd1;
        vga_data <= triangle2_3_rom_dout;
end
else if(triangle2_4_area == 1'b1) begin
        triangle2_4_rom_addr <= triangle2_4_rom_addr + 12'd1;
        vga_data <= triangle2_4_rom_dout;
end
else if(triangle2_5_area == 1'b1) begin
        triangle2_5_rom_addr <= triangle2_5_rom_addr + 12'd1;
        vga_data <= triangle2_5_rom_dout;
end
else if(triangle2_6_area == 1'b1) begin
        triangle2_6_rom_addr <= triangle2_6_rom_addr + 12'd1;
        vga_data <= triangle2_6_rom_dout;
end
else if(triangle2_7_area == 1'b1) begin
        triangle2_7_rom_addr <= triangle2_7_rom_addr + 12'd1;
        vga_data <= triangle2_7_rom_dout;
end
else if(triangle2_8_area == 1'b1) begin
        triangle2_8_rom_addr <= triangle2_8_rom_addr + 12'd1;
        vga_data <= triangle2_8_rom_dout;
end
else if(cookie_area == 1'b1 & !cookie_eaten & Mode) begin
        cookie_rom_addr <= cookie_rom_addr + 10'd1;
        vga_data <= cookie_rom_dout;
end
else if(fruit_area == 1'b1 & !fruit_eaten & Mode) begin
        fruit_rom_addr <= fruit_rom_addr + 10'd1;
        vga_data <= fruit_rom_dout;
end
else if(moving_stair_area & Mode) begin
        vga_data <= 12'h753;
        //vga_data <= 12'h888;
end
else if(disappear_stair_area & Mode & !disappear_la) begin
        if(disappear_counter == 2'b00)
                vga_data <= 12'h444;
        else if(disappear_counter == 2'b01)
```

```
                    vga_data <= 12'h888;
            end
            else
                vga_data = 12'h000;
        end
        else begin
            vga_data <= 12'h000;
            if (v_cnt == 0) begin
                snorlax_rom_addr<=11'd0;
                triangle1_1_rom_addr <= 12'd0;
                triangle1_2_rom_addr <= 12'd0;
                triangle1_3_rom_addr <= 12'd0;
                triangle1_4_rom_addr <= 12'd0;
                triangle1_5_rom_addr <= 12'd0;
                triangle1_6_rom_addr <= 12'd0;
                triangle1_7_rom_addr <= 12'd0;
                triangle2_1_rom_addr <= 12'd0;
                triangle2_2_rom_addr <= 12'd0;
                triangle2_3_rom_addr <= 12'd0;
                triangle2_4_rom_addr <= 12'd0;
                triangle2_5_rom_addr <= 12'd0;
                triangle2_6_rom_addr <= 12'd0;
                triangle2_7_rom_addr <= 12'd0;
                triangle2_8_rom_addr <= 12'd0;
                cookie_rom_addr <= 12'd0;
                fruit_rom_addr <= 12'd0;
            end
        end
    end
end

reg Victory;

//FSM start
always @(*) begin :COMB
    NS = CS;
    case(CS)
    Stop:begin
```

```
        if(button_start | keyboard_input == 8'h1B)
            NS = Movement;
        else
            NS = Stop;
    end
    Movement:begin
        if(on_Elevator == 0)
            NS = Falling;
        else if(chance == 0 || drop)
            NS = Die;
        else if(Victory)
            NS = Stop;
        else
            NS = Movement;

    end
    Falling:begin
        if(on_Elevator == 1)
            NS = Movement;
        else if(chance == 0 || drop)
            NS = Die;
        else if(Victory)
            NS = Stop;
        else
            NS = Falling;
    end
    Die:begin
        if(Die_counter == 3'b110)
            NS = Stop;
        else
            NS = Die;
    end
    endcase
end

always @(posedge clk or posedge rst) begin :SEQ
    if(rst) begin
        CS <= Stop;
        end
        else
            CS <= NS;
    end
    //FSM end

    always @(posedge clk or posedge rst)    //counter++ for timing
    begin
        if(rst) begin
            counter28 <= 28'b0;
        end
        else begin
            counter28 <= counter28 + 1'b1;
        end
    end

    always @(posedge counter28[24] or posedge rst) begin
        if(rst) begin
            Victory <= 1'b0;
        end
        else if(CS != Stop) begin
            if(Mode == 1'b0) begin
                if(Score == 10'd200)
                    Victory <= 1'b1;
            end
            else if(Mode == 1'b1) begin
                if(Score == 10'd999)
                    Victory <= 1'b1;
            end
        end
        else if(CS == Stop)
            if(Victory & LED_counter == 3'b111 & LED ==
16'b0010_0100_0010_0100)
                Victory <= 1'b0;
    end

    reg cookie_added;
```

```
always @(posedge counter28[25] or posedge rst) begin
    if(rst) begin
        Score <= 0;
        cookie_added <= 0;
    end
    else if(CS != Stop) begin
        if(Score > 10'd979) begin
            Score <= 10'd999;
        end
        else
            if(cookie_eaten & !cookie_added & Mode) begin
                Score <= Score + 10'd120;
                cookie_added <= 1'b1;
            end
            else if(cookie_y == 10'd851) begin
                cookie_added <= 1'b0;
            end
            else if(touch & counter28[27:26] == 2'b11)
                if(Score <= 50) begin
                    Score <= 10'd0;
                end
                else
                    Score <= Score - 10'd50;
            else if(counter28[27:26] == 2'b11)
                Score <= Score + 10'd20;
    end
end

always @(posedge clk)
begin
if(Score > 10'd99) begin
    Score_Hundred <= Score / 100;
    Score_Ten      <= (Score / 10) % 10;
    Score_Digits   <= Score % 10;
end
else if(Score > 10'd19) begin
    Score_Hundred <= 4'ha;
    Score_Ten      <= Score / 10;
```

```
            Score_Digits    <= Score % 10;
        end
        else begin
            Score_Hundred <= 4'ha;
            Score_Ten          <= 4'ha;
            Score_Digits    <= 4'ha;
        end
        case(counter28[20:19])
            2'b00:begin Enable <= 8'b00010001; SevenSeg_Left <=
SevenSet(Score_Digits);      SevenSeg_Right <= SevenSet(Life_4); end
            2'b01:begin Enable <= 8'b00100010; SevenSeg_Left <=
SevenSet(Score_Ten);            SevenSeg_Right <= SevenSet(Life_3); end
            2'b10:begin Enable <= 8'b01000100; SevenSeg_Left <=
SevenSet(Score_Hundred);    SevenSeg_Right <= SevenSet(Life_2); end
            2'b11:begin Enable <= 8'b10001000; SevenSeg_Left <= SevenSet(Mode);
SevenSeg_Right <= SevenSet(Life_1); end
        endcase
        end

        always @(*) begin
            case(chance)
            3'b000: begin Life_4 = 4'ha; Life_3 = 4'ha; Life_2 = 4'ha; Life_1 = 4'ha; end
            3'b001: begin Life_4 = 4'h0; Life_3 = 4'ha; Life_2 = 4'ha; Life_1 = 4'ha; end
            3'b010: begin Life_4 = 4'h0; Life_3 = 4'h0; Life_2 = 4'ha; Life_1 = 4'ha; end
            3'b011: begin Life_4 = 4'h0; Life_3 = 4'h0; Life_2 = 4'h0; Life_1 = 4'ha; end
            3'b100: begin Life_4 = 4'h0; Life_3 = 4'h0; Life_2 = 4'h0; Life_1 = 4'h0; end
            default: begin Life_4 = 4'h1; Life_3 = 4'h2; Life_2 = 4'h3; Life_1 = 4'h4; end
            endcase
        end
        //Seven Segment Show End
        reg first;
        always @(posedge counter28[24] or posedge rst) begin
            if(rst) begin
                LED_counter <= 0;
                Die_counter <= 0;
                LED = 16'b0000_0000_0000_0000;
                first <= 1'b1;
            end
        else if(CS == Die) begin
                Die_counter <= Die_counter + 1;
                case(Die_counter)
                3'b000: begin LED <= 16'b0000_0000_0000_0000; end
                3'b001: begin LED <= 16'b1111_1111_1111_1111; end
                3'b010: begin LED <= 16'b0000_0000_0000_0000; end
                3'b011: begin LED <= 16'b1111_1111_1111_1111; end
                3'b100: begin LED <= 16'b0000_0000_0000_0000; end
                3'b101: begin LED <= 16'b1111_1111_1111_1111; end
                3'b110: begin LED <= 16'b0000_0000_0000_0000; end
                endcase
        end
        else if(CS != Stop & chance > 0 & touch & counter28[27:26] == 3'b11) begin
                case(LED_counter[1])
                1'b0 : LED <= 16'b1111_1111_1111_1111;
                1'b1 : LED <= 16'b0000_0000_0000_0000;
                endcase
                LED_counter <= LED_counter + 1;
        end
        else if(CS == Stop & Victory == 1'b1) begin
                case(LED_counter)
                3'b000: begin LED <= 16'b1000_0001_1000_0001; end
                3'b001: begin LED <= 16'b0100_0010_0100_0010; end
                3'b010: begin LED <= 16'b0010_0100_0010_0100; end
                3'b011: begin LED <= 16'b0001_1000_0001_1000; end
                3'b100: begin LED <= 16'b1000_0001_1000_0001; end
                3'b101: begin LED <= 16'b0100_0010_0100_0010; end
                3'b110: begin LED <= 16'b0010_0100_0010_0100; end
                3'b111: begin LED <= 16'b0001_1000_0001_1000; end
                endcase
                LED_counter <= LED_counter + 1;
        end
        else begin
                LED_counter <= 3'b000;
                Die_counter <= 3'b000;
                LED <= 16'b0000_0000_0000_0000;
        end
end
```

```
    //stairs start

    always @(*) begin

    end

    always @(posedge counter28[25] or posedge rst) begin
        if(rst) begin
            stair_1_x <= 10'd1;     stair_2_x <= 10'd1;     stair_3_x <= 10'd121;   stair_4_x <= 10'd241;   stair_5_x <= 10'd41;    stair_6_x <= 10'd161;   stair_7_x <= 10'd161;
            stair_1_y <= 10'd111; stair_2_y <= 10'd231; stair_3_y <= 10'd351; stair_4_y <= 10'd471; stair_5_y <= 10'd591; stair_6_y <= 10'd651; stair_7_y <= 10'd831;
            stair_8_x <= 10'd161; stair_9_x <= 10'd81;   stair_10_x <= 10'd1; stair_11_x <= 10'd121;
            stair_8_y <= 10'd171; stair_9_y <= 10'd471; stair_10_y <= 10'd771; stair_11_y <= 10'd891;
            triangle1_1_x <= 10'd161; triangle1_2_x <= 10'd201; triangle1_3_x <= 10'd81;   triangle1_4_x <= 10'd121; triangle1_5_x <= 10'd1;     triangle1_6_x <= 10'd41;     triangle1_7_x <= 10'd121;
            triangle1_1_y <= 10'd121; triangle1_2_y <= 10'd121; triangle1_3_y <= 10'd421; triangle1_4_y <= 10'd421; triangle1_5_y <= 10'd721; triangle1_6_y <= 10'd721; triangle1_7_y <= 10'd841;
            triangle2_1_x <= 10'd1; triangle2_2_x <= 10'd41; triangle2_3_x <= 10'd81;   triangle2_4_x <= 10'd121; triangle2_5_x <= 10'd161;     triangle2_6_x <= 10'd201;     triangle2_7_x <= 10'd241; triangle2_8_x <= 10'd281;
            triangle2_y <= 10'd1;
            cookie_x <= 10'd86; cookie_y <= 10'd551; fruit_x <= 10'd286; fruit_y <= 10'd431;
            moving_stair_y <= 10'd531; disappear_stair_x <= 10'd201; disappear_stair_y <= 10'd411;
        end
        else if ((CS == Falling | CS == Movement) & counter28[27] & counter28[26] & Mode == 1'b0) begin
            if(stair_1_y == 10'd111) stair_1_y <= 10'd891; else stair_1_y <= stair_1_y - 10'd60;
            if(stair_2_y == 10'd111) stair_2_y <= 10'd891; else stair_2_y <=
stair_2_y - 10'd60;
            if(stair_3_y == 10'd111) stair_3_y <= 10'd891; else stair_3_y <=
stair_3_y - 10'd60;
            if(stair_4_y == 10'd111) stair_4_y <= 10'd891; else stair_4_y <=
stair_4_y - 10'd60;
            if(stair_5_y == 10'd111) stair_5_y <= 10'd891; else stair_5_y <=
stair_5_y - 10'd60;
            if(stair_6_y == 10'd111) stair_6_y <= 10'd891; else stair_6_y <=
stair_6_y - 10'd60;
            if(stair_7_y == 10'd111) stair_7_y <= 10'd891; else stair_7_y <=
stair_7_y - 10'd60;
            if(stair_8_y == 10'd111) stair_8_y <= 10'd891; else stair_8_y <=
stair_8_y - 10'd60;
            if(stair_9_y == 10'd111) stair_9_y <= 10'd891; else stair_9_y <=
stair_9_y - 10'd60;
            if(stair_10_y == 10'd111) stair_10_y <= 10'd891; else stair_10_y <=
stair_10_y - 10'd60;
            if(stair_11_y == 10'd111) stair_11_y <= 10'd891; else stair_11_y <=
stair_11_y - 10'd60;

            if(triangle1_1_y == 10'd61) triangle1_1_y <= 10'd841; else
triangle1_1_y <= triangle1_1_y - 10'd60;
            if(triangle1_2_y == 10'd61) triangle1_2_y <= 10'd841; else
triangle1_2_y <= triangle1_2_y - 10'd60;
            if(triangle1_3_y == 10'd61) triangle1_3_y <= 10'd841; else
triangle1_3_y <= triangle1_3_y - 10'd60;
            if(triangle1_4_y == 10'd61) triangle1_4_y <= 10'd841; else
triangle1_4_y <= triangle1_4_y - 10'd60;
            if(triangle1_5_y == 10'd61) triangle1_5_y <= 10'd841; else
triangle1_5_y <= triangle1_5_y - 10'd60;
            if(triangle1_6_y == 10'd61) triangle1_6_y <= 10'd841; else
triangle1_6_y <= triangle1_6_y - 10'd60;
            if(triangle1_7_y == 10'd61) triangle1_7_y <= 10'd841; else
triangle1_7_y <= triangle1_7_y - 10'd60;

            //if(moving_stair_y == 10'd111) moving_stair_y <= 10'd891; else
moving_stair_y <= moving_stair_y - 10'd60;
```

```
            //if(disappear_stair_y == 10'd111) disappear_stair_y <= 10'd891; else
disappear_stair_y <= disappear_stair_y - 10'd60;
        end
        else if ((CS == Falling|| CS == Movement) & counter28[26] & Mode == 1'b1)
begin
            if(stair_1_y == 10'd111) stair_1_y <= 10'd891; else stair_1_y <=
stair_1_y - 10'd60;
            if(stair_2_y == 10'd111) stair_2_y <= 10'd891; else stair_2_y <=
stair_2_y - 10'd60;
            if(stair_3_y == 10'd111) stair_3_y <= 10'd891; else stair_3_y <=
stair_3_y - 10'd60;
            if(stair_4_y == 10'd111) stair_4_y <= 10'd891; else stair_4_y <=
stair_4_y - 10'd60;
            if(stair_5_y == 10'd111) stair_5_y <= 10'd891; else stair_5_y <=
stair_5_y - 10'd60;
            if(stair_6_y == 10'd111) stair_6_y <= 10'd891; else stair_6_y <=
stair_6_y - 10'd60;
            if(stair_7_y == 10'd111) stair_7_y <= 10'd891; else stair_7_y <=
stair_7_y - 10'd60;
            if(stair_8_y == 10'd111) stair_8_y <= 10'd891; else stair_8_y <=
stair_8_y - 10'd60;
            if(stair_9_y == 10'd111) stair_9_y <= 10'd891; else stair_9_y <=
stair_9_y - 10'd60;
            if(stair_10_y == 10'd111) stair_10_y <= 10'd891; else stair_10_y <=
stair_10_y - 10'd60;
            if(stair_11_y == 10'd111) stair_11_y <= 10'd891; else stair_11_y <=
stair_11_y - 10'd60;

            if(triangle1_1_y == 10'd61) triangle1_1_y <= 10'd841; else
triangle1_1_y <= triangle1_1_y - 10'd60;
            if(triangle1_2_y == 10'd61) triangle1_2_y <= 10'd841; else
triangle1_2_y <= triangle1_2_y - 10'd60;
            if(triangle1_3_y == 10'd61) triangle1_3_y <= 10'd841; else
triangle1_3_y <= triangle1_3_y - 10'd60;
            if(triangle1_4_y == 10'd61) triangle1_4_y <= 10'd841; else
triangle1_4_y <= triangle1_4_y - 10'd60;
            if(triangle1_5_y == 10'd61) triangle1_5_y <= 10'd841; else
triangle1_5_y <= triangle1_5_y - 10'd60;
```

```
            if(triangle1_6_y == 10'd61) triangle1_6_y <= 10'd841; else
triangle1_6_y <= triangle1_6_y - 10'd60;
            if(triangle1_7_y == 10'd61) triangle1_7_y <= 10'd841; else
triangle1_7_y <= triangle1_7_y - 10'd60;

            if(fruit_y == 10'd71) fruit_y <= 10'd851; else fruit_y <= fruit_y -
10'd60;
            if(cookie_y == 10'd71) cookie_y <= 10'd851; else cookie_y <= cookie_y
- 10'd60;

            if(moving_stair_y == 10'd111) moving_stair_y <= 10'd891; else
moving_stair_y <= moving_stair_y - 10'd60;
            if(disappear_stair_y == 10'd111) disappear_stair_y <= 10'd891; else
disappear_stair_y <= disappear_stair_y - 10'd60;
        end


    end
    reg turn;
    always @(posedge clk or posedge rst) begin
        if(rst) begin
            moving_stair_x <= 10'd201;
            turn <= 1'b1;      //0 left ; 1 right
        end
        else if ((CS == Falling || CS == Movement) & Mode == 1'b1) begin
            if(moving_stair_x == 10'd201 & turn == 1'b1 & counter28[25:0] ==
26'b0_1111_1111_1111_1111_1111_1111) begin
                turn <= 1'b0;
                moving_stair_x <= 10'd201;
            end
            else if(moving_stair_x == 10'd1 & turn == 1'b0 & counter28[25:0] ==
26'b0_1111_1111_1111_1111_1111_1111) begin
                turn <= 1'b1;
                moving_stair_x <= 10'd1;
            end
            else if(turn == 1'b1 & counter28[25:0] ==
26'b0_1111_1111_1111_1111_1111_1111) begin
                moving_stair_x <= moving_stair_x + 10'd40;
            end
            else if(turn == 1'b0 & counter28[25:0] ==
26'b0_1111_1111_1111_1111_1111_1111) begin
                moving_stair_x <= moving_stair_x - 10'd40;
            end
        end
    end
    //stairs end

    //snorlax move start
    always @(posedge counter28[25] or posedge rst) begin
        if(rst) begin
            snorlax_y <= 10'd191;
        end
        else if (CS == Falling & !drop & Mode == 1'b0) begin
            if(snorlax_x == stair_1_x & snorlax_y + 10'd100 == stair_1_y &
counter28[27:26] == 2'b11) begin
                snorlax_y <= snorlax_y;
            end
            else if(snorlax_x == stair_1_x +10'd40 & snorlax_y + 10'd100 ==
stair_1_y & counter28[27:26] == 2'b11) begin
                snorlax_y <= snorlax_y;
            end
            else if(snorlax_x == stair_1_x +10'd80 & snorlax_y + 10'd100 ==
stair_1_y & counter28[27:26] == 2'b11) begin
                snorlax_y <= snorlax_y;
            end
            else if(snorlax_x == stair_2_x & snorlax_y + 10'd100 == stair_2_y &
counter28[27:26] == 2'b11) begin
                snorlax_y <= snorlax_y;
            end
            else if(snorlax_x == stair_2_x + 10'd40 & snorlax_y + 10'd100 ==
stair_2_y & counter28[27:26] == 2'b11) begin
                snorlax_y <= snorlax_y;
            end
            else if(snorlax_x == stair_3_x & snorlax_y + 10'd100 == stair_3_y &
counter28[27:26] == 2'b11) begin
                snorlax_y <= snorlax_y;
            end
            else if(snorlax_x == stair_3_x + 10'd40 & snorlax_y + 10'd100 ==
stair_3_y & counter28[27:26] == 2'b11) begin
                snorlax_y <= snorlax_y;
            end
            else if(snorlax_x == stair_4_x & snorlax_y + 10'd100 == stair_4_y &
counter28[27:26] == 2'b11) begin
                snorlax_y <= snorlax_y;
            end
            else if(snorlax_x == stair_4_x + 10'd40 & snorlax_y + 10'd100 ==
stair_4_y & counter28[27:26] == 2'b11) begin
                snorlax_y <= snorlax_y;
            end
            else if(snorlax_x == stair_5_x & snorlax_y + 10'd100 == stair_5_y &
counter28[27:26] == 2'b11) begin
                snorlax_y <= snorlax_y;
            end
            else if(snorlax_x == stair_5_x + 10'd40 & snorlax_y + 10'd100 ==
stair_5_y & counter28[27:26] == 2'b11) begin
                snorlax_y <= snorlax_y;
            end
            else if(snorlax_x == stair_6_x & snorlax_y + 10'd100 == stair_6_y &
counter28[27:26] == 2'b11) begin
                snorlax_y <= snorlax_y;
            end
            else if(snorlax_x == stair_6_x    + 10'd40 & snorlax_y + 10'd100 ==
stair_6_y & counter28[27:26] == 2'b11) begin
                snorlax_y <= snorlax_y;
            end
            else if(snorlax_x == stair_7_x & snorlax_y + 10'd100 == stair_7_y &
counter28[27:26] == 2'b11) begin
                snorlax_y <= snorlax_y;
            end
            else if(snorlax_x == stair_7_x + 10'd40 & snorlax_y + 10'd100 ==
stair_7_y & counter28[27:26] == 2'b11) begin
                snorlax_y <= snorlax_y;
            end
            else if(snorlax_x == stair_7_x + 10'd80 & snorlax_y + 10'd100 ==
stair_7_y & counter28[27:26] == 2'b11) begin
```

```
                snorlax_y <= snorlax_y;
            end
            else if(snorlax_x == stair_8_x & snorlax_y + 10'd100 == stair_8_y &
counter28[27:26] == 2'b11) begin
                snorlax_y <= snorlax_y;
            end
            else if(snorlax_x == stair_8_x + 10'd40 & snorlax_y + 10'd100 ==
stair_8_y & counter28[27:26] == 2'b11) begin
                snorlax_y <= snorlax_y;
            end
            else if(snorlax_x == stair_9_x & snorlax_y + 10'd100 == stair_9_y &
counter28[27:26] == 2'b11) begin
                snorlax_y <= snorlax_y;
            end
            else if(snorlax_x == stair_9_x + 10'd40 & snorlax_y + 10'd100 ==
stair_9_y & counter28[27:26] == 2'b11) begin
                snorlax_y <= snorlax_y;
            end
            else if(snorlax_x == stair_10_x & snorlax_y + 10'd100 == stair_10_y &
counter28[27:26] == 2'b11) begin
                snorlax_y <= snorlax_y;
            end
            else if(snorlax_x == stair_10_x + 10'd40 & snorlax_y + 10'd100 ==
stair_10_y & counter28[27:26] == 2'b11) begin
                snorlax_y <= snorlax_y;
            end
            else if(snorlax_x == stair_11_x & snorlax_y + 10'd100 == stair_11_y &
counter28[27:26] == 2'b11) begin
                snorlax_y <= snorlax_y;
            end
            else if(counter28[26] == 1'b1)
                snorlax_y <= snorlax_y + 10'd60;
        end
        else if (CS == Falling & !drop & Mode == 1'b1) begin
            if(snorlax_x == stair_1_x & snorlax_y + 10'd100 == stair_1_y &
counter28[26]) begin
                snorlax_y <= snorlax_y;
            end
            else if(snorlax_x == stair_1_x +10'd40 & snorlax_y + 10'd100 == stair_1_y & counter28[26]) begin
                snorlax_y <= snorlax_y;
            end
            else if(snorlax_x == stair_1_x +10'd80 & snorlax_y + 10'd100 == stair_1_y & counter28[26]) begin
                snorlax_y <= snorlax_y;
            end
            else if(snorlax_x == stair_2_x & snorlax_y + 10'd100 == stair_2_y & counter28[26]) begin
                snorlax_y <= snorlax_y;
            end
            else if(snorlax_x == stair_2_x + 10'd40 & snorlax_y + 10'd100 == stair_2_y & counter28[26]) begin
                snorlax_y <= snorlax_y;
            end
            else if(snorlax_x == stair_3_x & snorlax_y + 10'd100 == stair_3_y & counter28[26]) begin
                snorlax_y <= snorlax_y;
            end
            else if(snorlax_x == stair_3_x + 10'd40 & snorlax_y + 10'd100 == stair_3_y & counter28[26]) begin
                snorlax_y <= snorlax_y;
            end
            else if(snorlax_x == stair_4_x & snorlax_y + 10'd100 == stair_4_y & counter28[26]) begin
                snorlax_y <= snorlax_y;
            end
            else if(snorlax_x == stair_4_x + 10'd40 & snorlax_y + 10'd100 == stair_4_y & counter28[26]) begin
                snorlax_y <= snorlax_y;
            end
            else if(snorlax_x == stair_5_x & snorlax_y + 10'd100 == stair_5_y & counter28[26]) begin
                snorlax_y <= snorlax_y;
            end
            else if(snorlax_x == stair_5_x + 10'd40 & snorlax_y + 10'd100 == stair_5_y & counter28[26]) begin
                snorlax_y <= snorlax_y;
            end
            else if(snorlax_x == stair_6_x & snorlax_y + 10'd100 == stair_6_y &
    counter28[26]) begin
                snorlax_y <= snorlax_y;
            end
            else if(snorlax_x == stair_6_x    + 10'd40 & snorlax_y + 10'd100 ==
    stair_6_y & counter28[26]) begin
                snorlax_y <= snorlax_y;
            end
            else if(snorlax_x == stair_7_x & snorlax_y + 10'd100 == stair_7_y &
    counter28[26]) begin
                snorlax_y <= snorlax_y;
            end
            else if(snorlax_x == stair_7_x + 10'd40 & snorlax_y + 10'd100 ==
    stair_7_y & counter28[26]) begin
                snorlax_y <= snorlax_y;
            end
            else if(snorlax_x == stair_7_x + 10'd80 & snorlax_y + 10'd100 ==
    stair_7_y & counter28[26]) begin
                snorlax_y <= snorlax_y;
            end
            else if(snorlax_x == stair_8_x & snorlax_y + 10'd100 == stair_8_y &
    counter28[26]) begin
                snorlax_y <= snorlax_y;
            end
            else if(snorlax_x == stair_8_x + 10'd40 & snorlax_y + 10'd100 ==
    stair_8_y & counter28[26]) begin
                snorlax_y <= snorlax_y;
            end
            else if(snorlax_x == stair_9_x & snorlax_y + 10'd100 == stair_9_y &
    counter28[26]) begin
                snorlax_y <= snorlax_y;
            end
            else if(snorlax_x == stair_9_x + 10'd40 & snorlax_y + 10'd100 ==
    stair_9_y & counter28[26]) begin
                snorlax_y <= snorlax_y;
            end
```

```
            else if(snorlax_x == stair_10_x & snorlax_y + 10'd100 == stair_10_y &
counter28[26]) begin
                snorlax_y <= snorlax_y;
            end
            else if(snorlax_x == stair_10_x + 10'd40 & snorlax_y + 10'd100 ==
stair_10_y & counter28[26]) begin
                snorlax_y <= snorlax_y;
            end
            else if(snorlax_x == stair_11_x & snorlax_y + 10'd100 == stair_11_y &
counter28[26]) begin
                snorlax_y <= snorlax_y;
            end
            else if(snorlax_x == disappear_stair_x & snorlax_y + 10'd100 ==
disappear_stair_y & counter28[26]) begin
                snorlax_y <= snorlax_y;
            end
            else if(snorlax_x == disappear_stair_x + 10'd40 & snorlax_y + 10'd100
== disappear_stair_y & counter28[26]) begin
                snorlax_y <= snorlax_y;
            end
            else if(snorlax_x == disappear_stair_x + 10'd80 & snorlax_y + 10'd100
== disappear_stair_y & counter28[26]) begin
                snorlax_y <= snorlax_y;
            end
            else if(snorlax_x == moving_stair_x & snorlax_y + 10'd100 ==
moving_stair_y & counter28[26]) begin
                snorlax_y <= snorlax_y;
            end
            else if(snorlax_x == moving_stair_x + 10'd40 & snorlax_y + 10'd100 ==
moving_stair_y & counter28[26]) begin
                snorlax_y <= snorlax_y;
            end
            else if(snorlax_x == moving_stair_x + 10'd80 & snorlax_y + 10'd100 ==
moving_stair_y & counter28[26]) begin
                snorlax_y <= snorlax_y;
            end
            else
                snorlax_y <= snorlax_y + 10'd60;
```

```
        end
        else if (CS == Movement & counter28[27:26] == 2'b11 & !drop & Mode ==
1'b0) begin
            snorlax_y <= snorlax_y - 10'd60;
        end
        else if (CS == Movement & counter28[26] == 1'b1 & !drop & Mode == 1'b1)
begin
            snorlax_y <= snorlax_y - 10'd60;
        end
        else
            snorlax_y <= snorlax_y;
    end

    always @(*) begin
        if(snorlax_y == 10'd11 | snorlax_y == 10'd431) begin
            drop = 1'b1;
        end
        else
            drop = 1'b0;
    end

    reg [23:0] debounce;        //counter use for debounce
    reg push;      //state means bounce time
    wire push_mew;
    wire [7:0] keyboard;
    debounce u5(.sig_in(push), .clk(clk), .sig_out(push_new));

    always @(posedge clk or posedge rst) begin
        if(rst) begin
            next_snorlax_x <= 10'd1;
            push <= 1'b0;
            debounce <= 0;
        end
        else begin
            if(push == 1) begin
                debounce <= debounce +1;
                if(debounce == 24'b1111_1111_1111_1111_1111_1111) begin
                    push <= 0;
                    debounce <= 0;
                end
            end
            else if((keyboard_input == 8'h23 | button_right == 1'b1) & push_new
== 0 & snorlax_x <= 10'd241) begin
                push <= 1;
                next_snorlax_x <= next_snorlax_x + 10'd40;
            end
            else if((keyboard_input == 8'h1C | button_left == 1'b1) & push_new
== 0 & snorlax_x >= 10'd41) begin
                push <= 1;
                next_snorlax_x <= next_snorlax_x - 10'd40;
            end
            else if(on_moving & counter28[25:0] ==
26'b0_1111_1111_1111_1111_1111_1111) begin
                if(turn & moving_stair_x < 10'd200) begin    //0 left ; 1 right
                    next_snorlax_x <= next_snorlax_x + 10'd40;
                end
                else if(!turn& moving_stair_x > 10'd40) begin
                    next_snorlax_x <= next_snorlax_x - 10'd40;
                end
            end
            else begin
                next_snorlax_x <= next_snorlax_x;
            end
        end
    end

    always @(posedge ctrlclk or posedge rst) begin
        if(rst) begin
            snorlax_x <= 10'd1;
        end
        else if (CS == Movement || CS == Falling) begin
            snorlax_x <= next_snorlax_x;
        end
        else begin
            snorlax_x <= snorlax_x;
        end
    end

    always @(*) begin       //detect on elevator or not
        if(!drop) begin
            if(snorlax_x == stair_1_x & snorlax_y + 10'd40 == stair_1_y) begin
                on_Elevator = 1'b1; touch = 1'b0; on_moving = 1'b0;
on_disappear = 1'b0;
            end
            else if(snorlax_x == stair_1_x +10'd40 & snorlax_y + 10'd40 ==
stair_1_y) begin
                on_Elevator = 1'b1; touch = 1'b0; on_moving = 1'b0;
on_disappear = 1'b0;
            end
            else if(snorlax_x == stair_1_x +10'd80 & snorlax_y + 10'd40 ==
stair_1_y) begin
                on_Elevator = 1'b1; touch = 1'b0; on_moving = 1'b0;
on_disappear = 1'b0;
            end
            else if(snorlax_x == stair_2_x & snorlax_y + 10'd40 == stair_2_y) begin
                on_Elevator = 1'b1; touch = 1'b0; on_moving = 1'b0;
on_disappear = 1'b0;
            end
            else if(snorlax_x == stair_2_x + 10'd40 & snorlax_y + 10'd40 ==
stair_2_y) begin
                on_Elevator = 1'b1; touch = 1'b0; on_moving = 1'b0;
on_disappear = 1'b0;
            end
            else if(snorlax_x == stair_3_x & snorlax_y + 10'd40 == stair_3_y) begin
                on_Elevator = 1'b1; touch = 1'b0; on_moving = 1'b0;
on_disappear = 1'b0;
            end
            else if(snorlax_x == stair_3_x + 10'd40 & snorlax_y + 10'd40 ==
stair_3_y) begin
                on_Elevator = 1'b1; touch = 1'b0; on_moving = 1'b0;
on_disappear = 1'b0;
            end
            else if(snorlax_x == stair_4_x & snorlax_y + 10'd40 == stair_4_y) begin
                on_Elevator = 1'b1; touch = 1'b0; on_moving = 1'b0;
```

```
on_disappear = 1'b0;
        end
        else if(snorlax_x == stair_4_x + 10'd40 & snorlax_y + 10'd40 ==
stair_4_y) begin
           on_Elevator = 1'b1; touch = 1'b0; on_moving = 1'b0;
on_disappear = 1'b0;
        end
        else if(snorlax_x == stair_5_x & snorlax_y + 10'd40 == stair_5_y) begin
           on_Elevator = 1'b1; touch = 1'b0; on_moving = 1'b0;
on_disappear = 1'b0;
        end
        else if(snorlax_x == stair_5_x + 10'd40 & snorlax_y + 10'd40 ==
stair_5_y) begin
           on_Elevator = 1'b1; touch = 1'b0; on_moving = 1'b0;
on_disappear = 1'b0;
        end
        else if(snorlax_x == stair_6_x & snorlax_y + 10'd40 == stair_6_y) begin
           on_Elevator = 1'b1; touch = 1'b0; on_moving = 1'b0;
on_disappear = 1'b0;
        end
        else if(snorlax_x == stair_6_x    + 10'd40 & snorlax_y + 10'd40 ==
stair_6_y) begin
           on_Elevator = 1'b1; touch = 1'b0; on_moving = 1'b0;
on_disappear = 1'b0;
        end
        else if(snorlax_x == stair_7_x & snorlax_y + 10'd40 == stair_7_y) begin
           on_Elevator = 1'b1; touch = 1'b0; on_moving = 1'b0;
on_disappear = 1'b0;
        end
        else if(snorlax_x == stair_7_x + 10'd40 & snorlax_y + 10'd40 ==
stair_7_y) begin
           on_Elevator = 1'b1; touch = 1'b0; on_moving = 1'b0;
on_disappear = 1'b0;
        end
        else if(snorlax_x == stair_7_x + 10'd80 & snorlax_y + 10'd40 ==
stair_7_y) begin
           on_Elevator = 1'b1; touch = 1'b0; on_moving = 1'b0;
on_disappear = 1'b0;
```

```
            end
            else if(snorlax_x == stair_8_x & snorlax_y + 10'd40 == stair_8_y) begin
                on_Elevator = 1'b1; touch = 1'b1; on_moving = 1'b0;
on_disappear = 1'b0;
            end
            else if(snorlax_x == stair_8_x + 10'd40 & snorlax_y + 10'd40 ==
stair_8_y) begin
                on_Elevator = 1'b1; touch = 1'b1; on_moving = 1'b0;
on_disappear = 1'b0;
            end
            else if(snorlax_x == stair_9_x & snorlax_y + 10'd40 == stair_9_y) begin
                on_Elevator = 1'b1; touch = 1'b1; on_moving = 1'b0;
on_disappear = 1'b0;
            end
            else if(snorlax_x == stair_9_x + 10'd40 & snorlax_y + 10'd40 ==
stair_9_y) begin
                on_Elevator = 1'b1; touch = 1'b1; on_moving = 1'b0;
on_disappear = 1'b0;
            end
            else if(snorlax_x == stair_10_x & snorlax_y + 10'd40 == stair_10_y)
begin
                touch = 1'b1; on_Elevator = 1'b1; on_moving = 1'b0;
on_disappear = 1'b0;
            end
            else if(snorlax_x == stair_10_x + 10'd40 & snorlax_y + 10'd40 ==
stair_10_y)    begin
                touch = 1'b1; on_Elevator = 1'b1; on_moving = 1'b0;
on_disappear = 1'b0;
            end
            else if(snorlax_x == stair_11_x & snorlax_y + 10'd40 == stair_11_y)
begin
                on_Elevator = 1'b1; touch = 1'b1; on_moving = 1'b0;
on_disappear = 1'b0;
            end
            else if(snorlax_x == disappear_stair_x & snorlax_y + 10'd40 ==
disappear_stair_y & !disappear_la) begin
                on_Elevator = 1'b1; touch = 1'b0; on_moving = 1'b0;
on_disappear = 1'b1;
```

```
            end
            else if(snorlax_x == disappear_stair_x + 10'd40 & snorlax_y + 10'd40
== disappear_stair_y & !disappear_la) begin
                on_Elevator = 1'b1; touch = 1'b0; on_moving = 1'b0;
on_disappear = 1'b1;
            end
            else if(snorlax_x == disappear_stair_x + 10'd80 & snorlax_y + 10'd40
== disappear_stair_y & !disappear_la) begin
                on_Elevator = 1'b1; touch = 1'b0; on_moving = 1'b0;
on_disappear = 1'b1;
            end
            else if(snorlax_x == moving_stair_x & snorlax_y + 10'd40 ==
moving_stair_y) begin
                on_Elevator = 1'b1; touch = 1'b0; on_moving = 1'b1;
on_disappear = 1'b0;
            end
            else if(snorlax_x == moving_stair_x + 10'd40 & snorlax_y + 10'd40 ==
moving_stair_y) begin
                on_Elevator = 1'b1; touch = 1'b0; on_moving = 1'b1;
on_disappear = 1'b0;
            end
            else if(snorlax_x == moving_stair_x + 10'd80 & snorlax_y + 10'd40 ==
moving_stair_y) begin
                on_Elevator = 1'b1; touch = 1'b0; on_moving = 1'b1;
on_disappear = 1'b0;
            end
            else begin
                on_Elevator = 1'b0; touch = 1'b0; on_moving = 1'b0;
on_disappear = 1'b0;
            end
        end
    end



    always @(posedge counter28[26] or posedge rst) begin
        if(rst | disappear_stair_y == 10'd891) begin
            disappear_counter <= 2'b00;
            disappear_la <= 1'b0;
        end
        else begin
            if(disappear_counter == 2'b01 & disappear_la == 1'b0) begin
                disappear_la <= 1'b1;
            end
            else if(on_disappear) begin
                disappear_counter <= disappear_counter + 1'b1;
            end
        end
    end

    reg fruit_added;

    always @(posedge counter28[25] or posedge rst) begin
        if(rst) begin
            chance <= 3'b100;
            fruit_added <= 1'b0;
        end
        else if(fruit_y == 10'd851) begin
            fruit_added <= 1'b0;
        end
        else if(fruit_eaten & !fruit_added & Mode) begin
            if(chance == 3'b100)
                chance <= chance;
            else
                chance <= chance + 1'b1;
            fruit_added <= 1'b1;
        end
        else if(touch & CS != Stop & counter28[27:26] == 2'b11) begin
            if(chance == 3'b000)
                chance <= chance;
            else
                chance <= chance - 1'b1;
        end
    end

    reg cookie_touch, fruit_touch;
```

```
always @(*) begin
    if(snorlax_x == cookie_x - 10'd5 & snorlax_y == cookie_y & Mode) begin
        cookie_touch = 1'b1; fruit_touch = 1'b0;
    end
    else if(snorlax_x == fruit_x - 10'd5 & snorlax_y == fruit_y & Mode) begin
        fruit_touch = 1'b1; cookie_touch = 1'b0;
    end
    else begin
        cookie_touch = 1'b0;
        fruit_touch = 1'b0;
    end
end

always @(posedge clk or posedge rst) begin
    if(rst) begin
        if(Mode) begin
            cookie_eaten <= 1'b0;
            fruit_eaten <= 1'b0;
        end
        else if(!Mode) begin
            cookie_eaten <= 1'b1;
            fruit_eaten <= 1'b1;
        end
    end
    else begin
        if(cookie_y == 10'd851) begin
            cookie_eaten <= 1'b0;
        end
        else if(fruit_y == 10'd851) begin
            fruit_eaten <= 1'b0;
        end
        else if(cookie_touch) begin
            cookie_eaten <= 1'b1;
        end
        else if(fruit_touch) begin
            fruit_eaten <= 1'b1;
        end
```

```
                end
        end

        //snorlax move end

function [7:0] SevenSet;
input [3:0] digits;

begin
        case(digits)
        4'h0: SevenSet = 8'b00111111;
        4'h1: SevenSet = 8'b00000110;
        4'h2: SevenSet = 8'b01011011;
        4'h3: SevenSet = 8'b01001111;
        4'h4: SevenSet = 8'b01100110;
        4'h5: SevenSet = 8'b01101101;
        4'h6: SevenSet = 8'b01111101;
        4'h7: SevenSet = 8'b00100111;
        4'h8: SevenSet = 8'b01111111;
        4'h9: SevenSet = 8'b01101111;
        4'ha: SevenSet = 8'b00000000;
        default: SevenSet = 8'b1111_1111;

        endcase
end
endfunction

endmodule

module debounce_better_version(input pb_1,clk,output pb_out);
wire slow_clk_en;
wire Q1,Q2,Q2_bar,Q0;
clock_enable u1(clk,slow_clk_en);
my_dff_en d0(clk,slow_clk_en,pb_1,Q0);

my_dff_en d1(clk,slow_clk_en,Q0,Q1);
my_dff_en d2(clk,slow_clk_en,Q1,Q2);
assign Q2_bar = ~Q2;
assign pb_out = Q1 & Q2_bar;
endmodule
// Slow clock enable for debouncing button
module clock_enable(input Clk_100M,output slow_clk_en);
        reg [26:0]counter=0;
        always @(posedge Clk_100M)
        begin
              counter <= (counter>=249999)?0:counter+1;
        end
        assign slow_clk_en = (counter == 249999)?1'b1:1'b0;
endmodule
// D-flip-flop with clock enable signal for debouncing module
module my_dff_en(input DFF_CLOCK, clock_enable,D, output reg Q=0);
        always @ (posedge DFF_CLOCK) begin
    if(clock_enable==1)
                      Q <= D;
        end
endmodule

module ps2(
        input clk,
        input data,
        input reset,
        output reg [7:0] drink
        );

        reg [7:0] data_curr;
        reg [7:0] data_pre;
        reg [3:0] b;
        reg flag;
        reg start;
        reg start2;
        reg [1:0] counter;

        always @(negedge clk or negedge reset) begin
        if(!reset) begin
                b<=4'h1;
                flag<=1'b0;
                data_curr<=8'hf0;
                data_pre<=8'hf0;
                drink <= 0;
                start <= 0; //keyboard signal start
                start2 <= 0;
                counter <= 0;
        end
        else begin
                if(data == 0 && !start)begin
                    start <= 1;
                    b <= 2;
                end

                if(start2) begin
                    counter <= counter + 1'b1;
                    if(counter == 2'b11) begin
                        start2 <= 1'b0;
                        drink <= 8'hf0;
                    end
                end
                if(data_curr == 8'hf0) begin
                    drink <= data_pre;
                    start2 <= 1;
                end
                else if(flag)
                    data_pre <= data_curr;


                case(b)
                1:;
                2: data_curr[0] <= data;
                3: data_curr[1] <= data;
                4: data_curr[2] <= data;
                5: data_curr[3] <= data;
                6: data_curr[4] <= data;
                7: data_curr[5] <= data;
                8: data_curr[6] <= data;
                9: data_curr[7] <= data;
```

```
            10: flag <= 1'b1;
            11: flag <= 1'b0;
            endcase
            if(b<=10) begin
                if(start)
                    b <= b + 1;
            end
            else begin
                b <= 1;
                start <= 0;
            end
        end
        end

endmodule
```

2.分工

資電四 吳葦誠 107504507 50%

整體架構、FSM、鍵盤輸入、尖刺與樓梯移動、移動消失樓梯、血量操作

資電三 王珮榕 108504502 50%

COE 檔相關、按鈕輸入、卡比移動、果實餅乾、LED、七段顯示器、分數操作