Binary Blitz

Digital Systems Laboratory Verilog GAMES Final Project

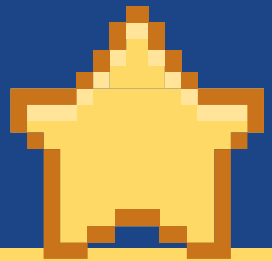
Binary Blitz

Digital Systems Laboratory Verilog GAMES Final Project

411712l3H 張慎修、411750333H 吳炳煌

TABLE OF CONTENTS

01
Motivation



Motivation

Learn Binary Unsigned Number

Enhance Logical Thinking

Improve Mathematical Skills

Increase Reaction Speed

02
Schedule

MAY/JUNE CALENDAR

21	22	23	24	25	26	27
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24

Design Thinking

Thinking the game content, function and design for someone

Practice

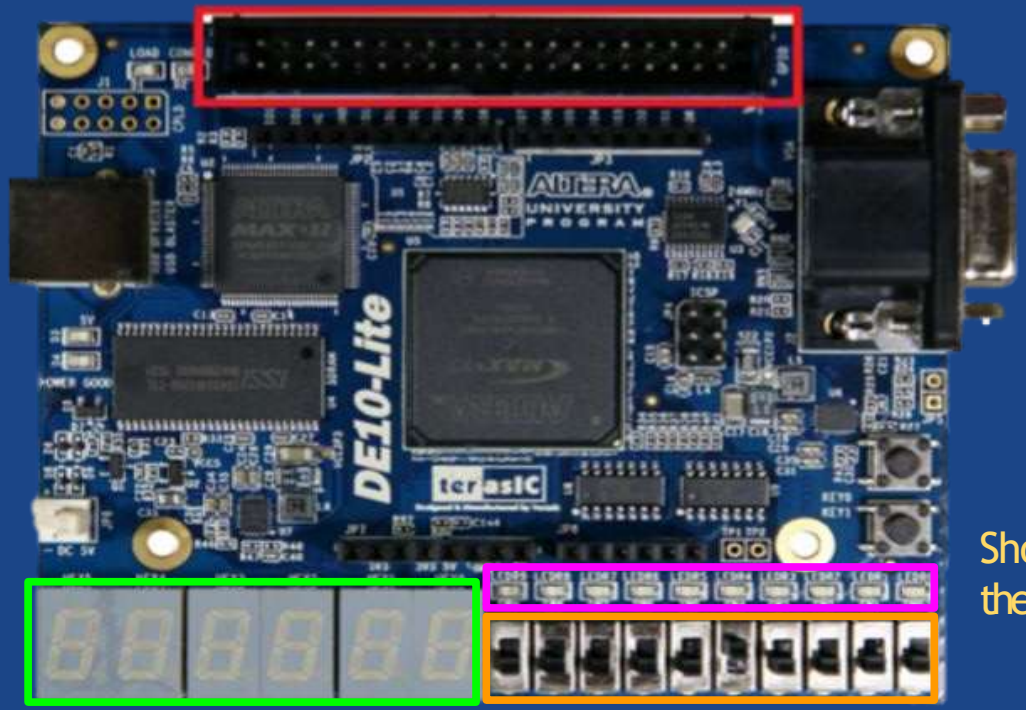
Write programs ,Debug ,combine with buzzer and the concept teaching by teacher

Record/PPT Final presentation


03
Functions/
Design Details

Functions/Design Details

Connect the FPGA and buzzer using GND and PIN_W10



Show the question on segment 0,1

Show the user answer on segment 2,3

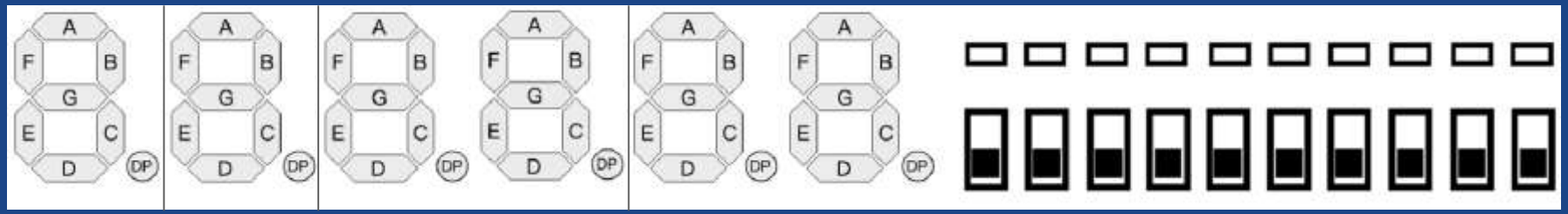
Show the countdown timer second on segment 5

Show the pili led to increase the excitement

Input the switch to calculate the binary unsigned number

Functions/Design Details

Pili led display



Countdown Timer

Display Switch input

Display Lsfr Question

Input Answer Switch

Functions/Design Details

```
module timer(led,clk, rst, pb, a, seg5, seg_a, seg_b ,speaker,Q,out0,out1);
    input clk, rst, pb;
    input [3:0] a;
    output reg [6:0] seg5;
    output reg [7:0] seg_a, seg_b;
    output reg speaker;
    parameter clkdivider = 10000000/349/2;
    output reg [9:0] led;

    output reg [4:1] Q;
    output reg [7:0] out0;
    output reg [7:0] out1;

    reg [6:0]sec;
    reg [4:0] state, nstate;
    reg [23:0] tone;

    reg clk1,clk2;
    reg [29:0] count;//pili
    reg [22:0] counter;//timer
    reg [15:0] counter1;//buzz
```

Functions/Design Details

```
always@(posedge clk)begin
    tone = tone + 1;
end


always @(posedge pb or negedge rst) begin
if (!rst) begin
    Q = 4'b1000;
  end
  else begin
        Q = {Q[3:1], Q[4]^Q[3]};
        if (Q == 4'b1110) begin
          Q = 4'b1100;
        end
        else if (Q == 4'b1100) begin
          Q = 4'b1000;
        end
  end
end
```

Functions/Design Details

```
//delay clock
always@(posedge clk or negedge rst or negedge pb)begin
    if(~rst)begin
        counter = 0;
        counter1 = 0;
        clk1 = 0;
    end
    else if(~pb)begin
        counter = 0;
        count = 0;
        clk1 = 0;
    end
    else if(counter==1000000)begin//timer
        counter =0;
        clk1 = ~clk1;
    end
    else if(count==250000)begin//pili
        count = 0;
        clk2 = ~clk2;
    end
    else if(counter1 == 0)begin
        if (sec == 4'b0000)begin
            if(a != Q)begin//wrong
                if(tone[22])
                    counter1 = clkdivider*1.5;
                else
                    counter1 = clkdivider;
            end
            else begin//correct
                if(tone[22])
                    counter1 = clkdivider;
                else
                    counter1 = clkdivider/3;
            end
        end
    end
    else
        counter = counter - 1;
        counter1 = counter1 - 1;
        count = count + 1;
end
```

Functions/Design Details

```
always @(posedge clk2 or negedge rst )begin
    if(~rst) state = 0;
    else state = nstate;
end

always@(posedge clk1 or negedge rst or negedge pb )begin
    if(~rst | ~pb)begin
        sec = 4'b0011;
    end

    else
        sec = sec - 1;
end

always @(posedge clk)begin
    if(counter1 == 0)
        speaker = ~speaker;
end

always @(state)begin
    if(state == 5'b10000)
        nstate = 5'b00001;
    else
        nstate = state + 1;
```

Functions/Design Details

```
always@(state)begin//pili
    case(state)
        5'b00000: led = 10'b1000000001;
        5'b00001: led = 10'b0100000010;
        5'b00010: led = 10'b0010000100;
        5'b00011: led = 10'b0001001000;
        5'b00100: led = 10'b0000110000;
        5'b00101: led = 10'b0001001000;
        5'b00110: led = 10'b0010000100;
        5'b00111: led = 10'b0100000010;
        5'b01000: led = 10'b1000000001;
        5'b01001: led = 10'b0100000010;
        5'b01010: led = 10'b0010000100;
        5'b01011: led = 10'b0001001000;
        5'b01100: led = 10'b0000110000;
        5'b01101: led = 10'b0001001000;
        5'b01110: led = 10'b0010000100;
        5'b01111: led = 10'b0100000010;
        5'b10000: led = 10'b1000000001;
        5'b10001: led = 10'b1000000001;
        5'b10010: led = 10'b1000000001;
       default: led = 10'b0000000000;
    endcase
end

always@(sec)begin//timer
    case(sec)
         4'b0000:  seg5 = 7'b1000000;//0
         4'b0001:  seg5 = 7'b1111001;//1
         4'b0010:  seg5 = 7'b0100100;//2
         4'b0011:  seg5 = 7'b0110000;//3
         default:  seg5 = 7'b1111111;//all blind
    endcase
end
```

Functions/Design Details

```
    always@(a)begin//switch
        case(a)
            4'b0000: {seg_b, seg_a} = {8'b11111111, 8'b11000000}; // 0
            4'b0001: {seg_b, seg_a} = {8'b11111111, 8'b11111001}; // 1
            4'b0010: {seg_b, seg_a} = {8'b11111111, 8'b10100100}; // 2
            4'b0011: {seg_b, seg_a} = {8'b11111111, 8'b10110000}; // 3
            4'b0100: {seg_b, seg_a} = {8'b11111111, 8'b10011001}; // 4
            4'b0101: {seg_b, seg_a} = {8'b11111111, 8'b10010010}; // 5
            4'b0110: {seg_b, seg_a} = {8'b11111111, 8'b10000010}; // 6
            4'b0111: {seg_b, seg_a} = {8'b11111111, 8'b11111000}; // 7
            4'b1000: {seg_b, seg_a} = {8'b11111111, 8'b10000000}; // 8
            4'b1001: {seg_b, seg_a} = {8'b11111111, 8'b10010000}; // 9
            4'b1010: {seg_b, seg_a} = {8'b11111001, 8'b11000000}; // 10
            4'b1011: {seg_b, seg_a} = {8'b11111001, 8'b11111001}; // 11
            4'b1100: {seg_b, seg_a} = {8'b11111001, 8'b10100100}; // 12
            4'b1101: {seg_b, seg_a} = {8'b11111001, 8'b10110000}; // 13
            4'b1110: {seg_b, seg_a} = {8'b11111001, 8'b10011001}; // 14
            4'b1111: {seg_b, seg_a} = {8'b11111001, 8'b10010010}; // 15
        endcase
    end
endmodule
```

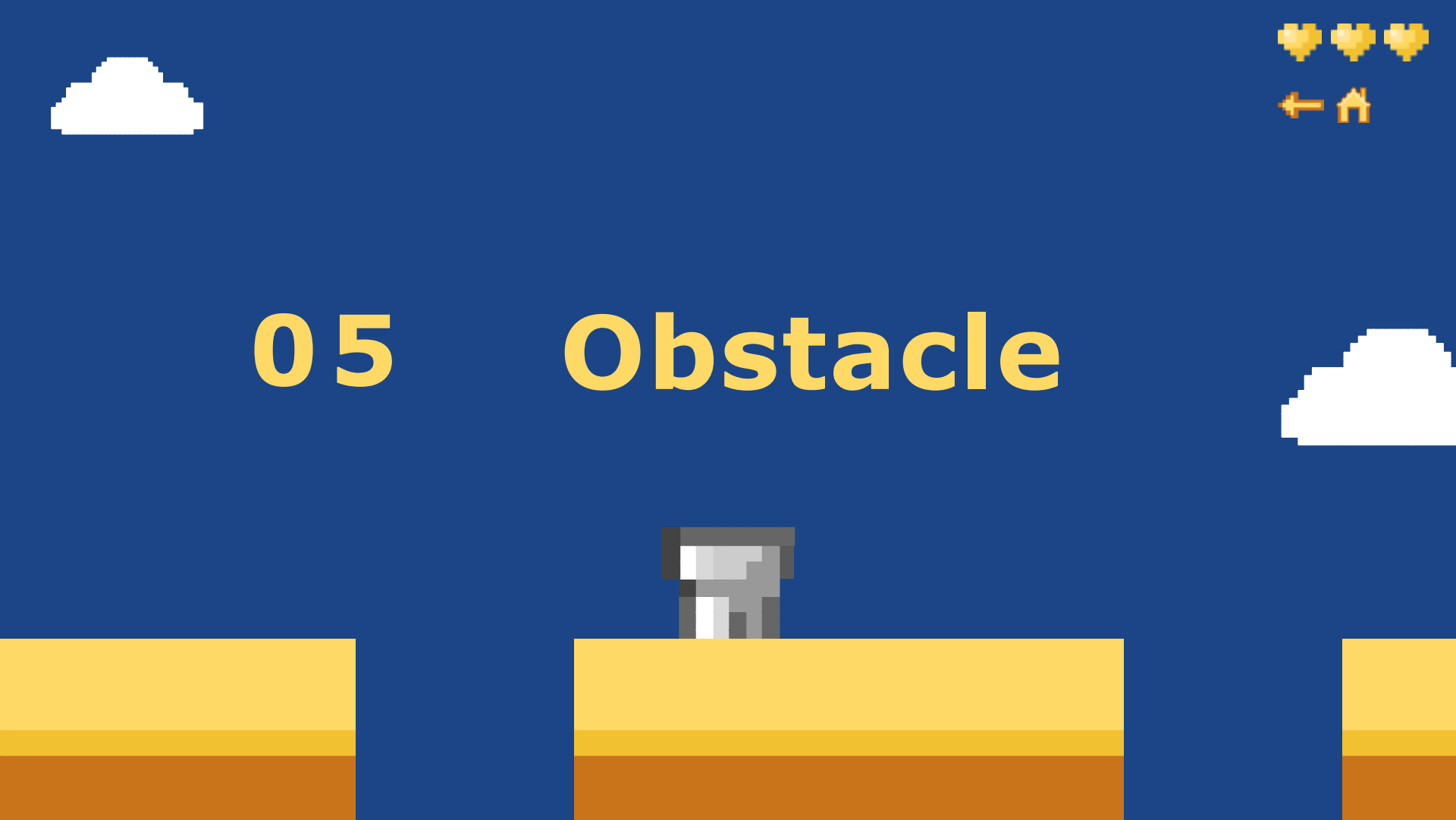
```
always @(Q) begin
  case (Q)
    4'b0001: {out1, out0} = {8'b11111111, 8'b11111001}; // 01
    4'b0010: {out1, out0} = {8'b11111111, 8'b10100100}; // 02
    4'b0011: {out1, out0} = {8'b11111111, 8'b10110000}; // 03
    4'b0100: {out1, out0} = {8'b11111111, 8'b10011001}; // 04
    4'b0101: {out1, out0} = {8'b11111111, 8'b10010010}; // 05
    4'b0110: {out1, out0} = {8'b11111111, 8'b10000010}; // 06
    4'b0111: {out1, out0} = {8'b11111111, 8'b11111000}; // 07
    4'b1000: {out1, out0} = {8'b11111111, 8'b10000000}; // 08
    4'b1001: {out1, out0} = {8'b11111111, 8'b10010000}; // 09
    4'b1010: {out1, out0} = {8'b11111001, 8'b11000000}; // 10
    4'b1011: {out1, out0} = {8'b11111001, 8'b11111001}; // 11
    4'b1100: {out1, out0} = {8'b11111001, 8'b10100100}; // 12
    4'b1101: {out1, out0} = {8'b11111001, 8'b10110000}; // 13
    4'b1110: {out1, out0} = {8'b11111001, 8'b10011001}; // 14
    4'b1111: {out1, out0} = {8'b11111001, 8'b10010010}; // 15
    default: {out1, out0} = {8'b11111111, 8'b11111111};
  endcase
end
```


04
Display

Display


Binary Blitz
Display
Digital Systems Laboratory Verilog GAMES Final Project
41171213H 張愼修、41175033H 吳炳煌

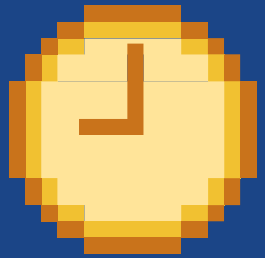
05 Obstacle



Obstacle

Determine win/lose

When we done every part of game, we have no idea to combine them to become a game ,we got idea in that the LSFR and the input switch will make the segment change , so we deal with it.

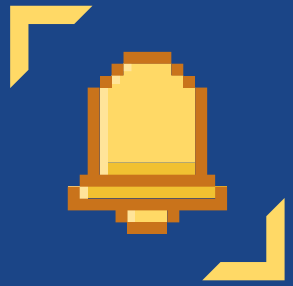


Divide different clock

We have countdown timer ,pili led and buzzer ,we want the three part have different frequent ,we learn it has clk, clk1 in the class, so we try to add clk2 to complete the game.

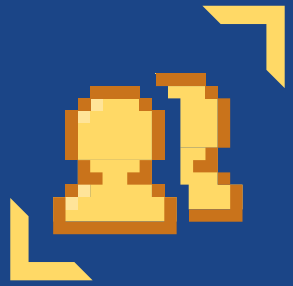

06
Future
Prospects

Future Prospects



Buzzer

Add the player can adjust volume , variable tones and make rhythm and beats let the game more fun and meaning of education.



More Player

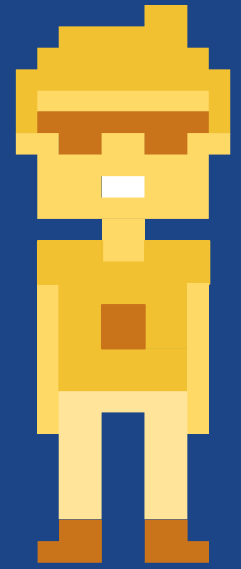
Add more players mode with score tracking for intense binary bables, increase the game’s fun and challenge.

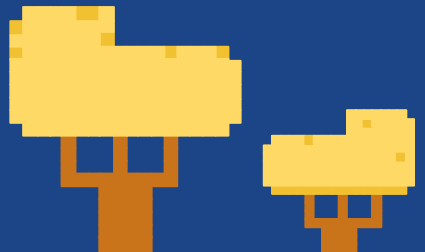
07
Feedback

FIRST ANNOUNCEMENT

楊同學：**Its a good idea that mini games can be fun and educating.**

吳同學：**It was very creative and also fun! I look forward to future development.**






THANK
YOU