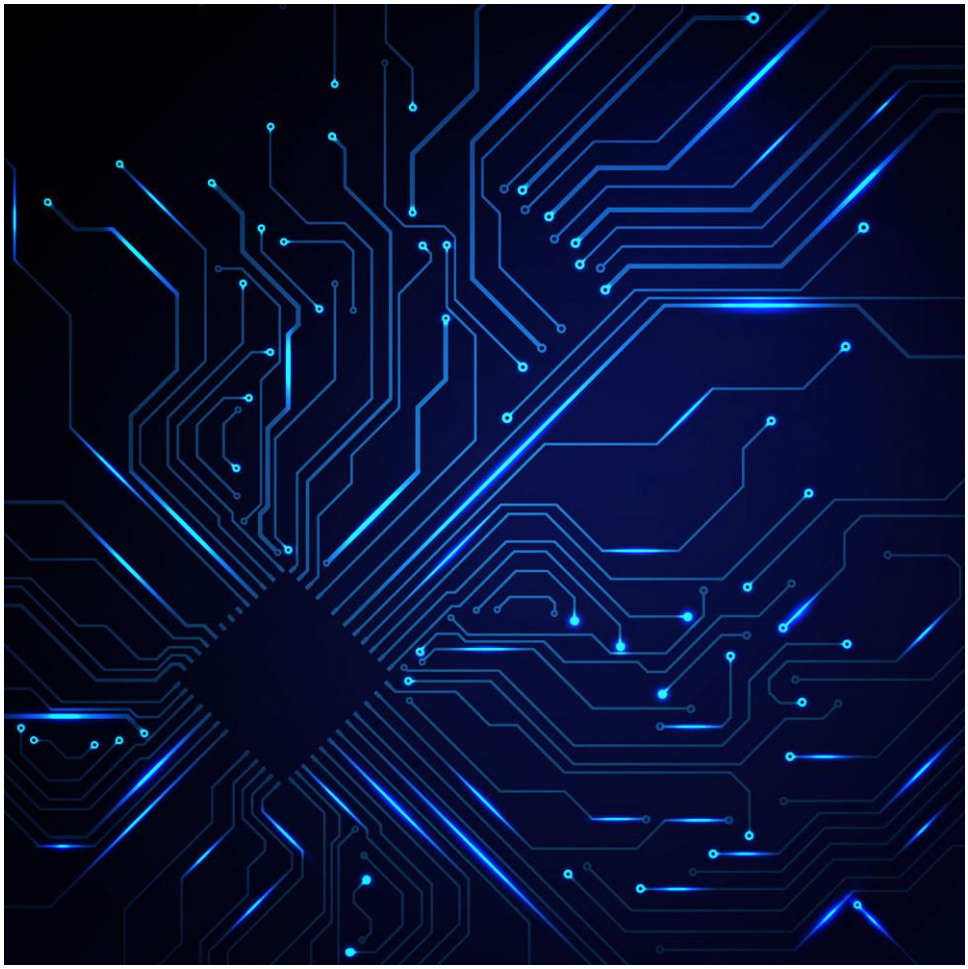


2023/02/22

實驗一

序向邏輯練習

姓名：吳秉宸　　學號：00957202

班級：資工 4A

E-mail：evan20010126@gmail.com

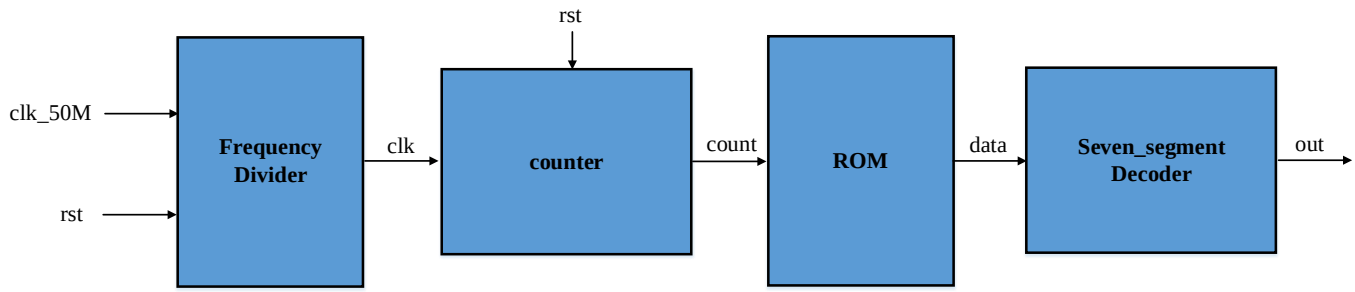
注意

1. 繳交時一律轉 PDF 檔
2. 繳交期限為
   隔週三上午 9 點
3. 一人繳交一份
4. 檔名：學號_HW?.pdf
   檔名請按照作業檔名格式進行填寫
   未依照格式不予批改

一、 跳學號

● 實驗說明：

1. 將學號逐一輸出，並接上七段顯示器，使用 DESIM 模擬並截圖模擬結果。



- 系統架構程式碼、測試資料程式碼與程式碼說明

  截圖請善用 win+shift+S

  - student_id.sv

```
module student_id(
    input clk_50M,
    input reset, //為0時重置系統
    input reset_div,
    output logic[6:0] out
    );

    logic clk; //除頻後的clk
    logic[2:0] count;
    logic[3:0] num;

    //FSM


    //除頻器(1/2)
    frequency_divider div1(
        .clk_after(clk),
        .clk(clk_50M),
        .reset_fd(reset_div)
        );
    /*
     @(posedge clk_50M or negedge reset) begin
        if(!reset)
            clk <= 0;
        else
            clk <= ~clk;
    end
    */

    //counter
    always_ff @(posedge clk or negedge reset) begin
        if(!reset)
            count <= 0;
        else if(count==3'h7)
            count<=0;
        else begin
            count <= count+1;
        end
    end

    //ROM
    ROM Rom1(.Rom_data_out(num), .Rom_addr_in(count));

    //7-seg
    seven_segment Seg1(.in(num), .out(out));

endmodule
```

本次實驗流程為，先利用除頻器除頻到需要的頻率，再利用 counter 計數，取出 ROM 內不同的學號，再利用 seven_decoder 將學號轉換為七段顯示器的編碼。

➢ frequency_divider.sv

```
module frequency_divider(
    output logic clk_after,
    input clk,
    input reset_fd
    );
    parameter n = 24;

    logic [n-1:0] count;

    always_ff @(posedge clk)
    begin
        if(!reset_fd)
            count <= 0;
        else
            count <= count + 1;
    end

    assign clk_after = count[16];
endmodule
```

此處將頻率除頻

➢ rom.sv

```
module ROM(
    output [3:0] Rom_data_out,
    input [2:0] Rom_addr_in
);
//---------

    logic [3:0] data;
    always_comb
        begin
            case (Rom_addr_in)
                3'h0:   data = 4'h0;
                3'h1:   data = 4'h0;
                3'h2:   data = 4'h9;
                3'h3:   data = 4'h5;
                3'h4:   data = 4'h7;
                3'h5:   data = 4'h2;
                3'h6:   data = 4'h0;
                3'h7:   data = 4'h2;
                default:data = 4'hX;
            endcase
```

```
        end
     assign Rom_data_out = data;

endmodule
```

此處填寫對應的學號。

➢ seven_segment.sv

```
module seven_segment(
    input [3:0] in,
    output logic [6:0] out
);

    always_comb
    begin
        case(in)
        4'h0: out = 7'hc0;
        4'h1: out = 7'hf9;
        4'h2: out = 7'ha4;
        4'h3: out = 7'hb0;
        4'h4: out = 7'h99;
        4'h5: out = 7'h92;
        4'h6: out = 7'h82;
        4'h7: out = 7'hf8;
        4'h8: out = 7'h80;
        4'h9: out = 7'h90;
        4'hA: out = 7'h88;
        4'hB: out = 7'h83;
        4'hC: out = 7'hc6;
        4'hD: out = 7'ha1;
        4'hE: out = 7'h86;
        4'hF: out = 7'h8e;
        endcase
    end
endmodule
```

將學號轉換為七段顯示器的編碼。

- Complie.do

```
#vlib work



# ----------------------------------------------------------------
vlog ../tb/test_student_id.sv
vlog ../../design/student_id.sv
vlog ../../design/seven_segment.sv
vlog ../../design/ROM.sv
vlog ../../design/frequency_divider.sv
# vlog ../../design/DE0_CV.v









# ----------------------------------------------------------------
```

此處編譯了所需要的所有檔案。

- Wave.do

```
onerror {resume}
quietly WaveActivateNextPane {} 0

add wave -noupdate -divider {TOP LEVEL INPUTS}

#add wave -noupdate -format Logic /testbench/clk
#add wave -noupdate -format Logic /testbench/reset

add wave -noupdate -format Logic /test_student_id/clk
add wave -noupdate -format Logic /test_student_id/reset
add wave -noupdate -format Literal -radix Hexadecimal /test_student_id/seven_seg

add wave -noupdate -divider {cpu}
```

此處添加在 ModeSim 想查看的訊號。

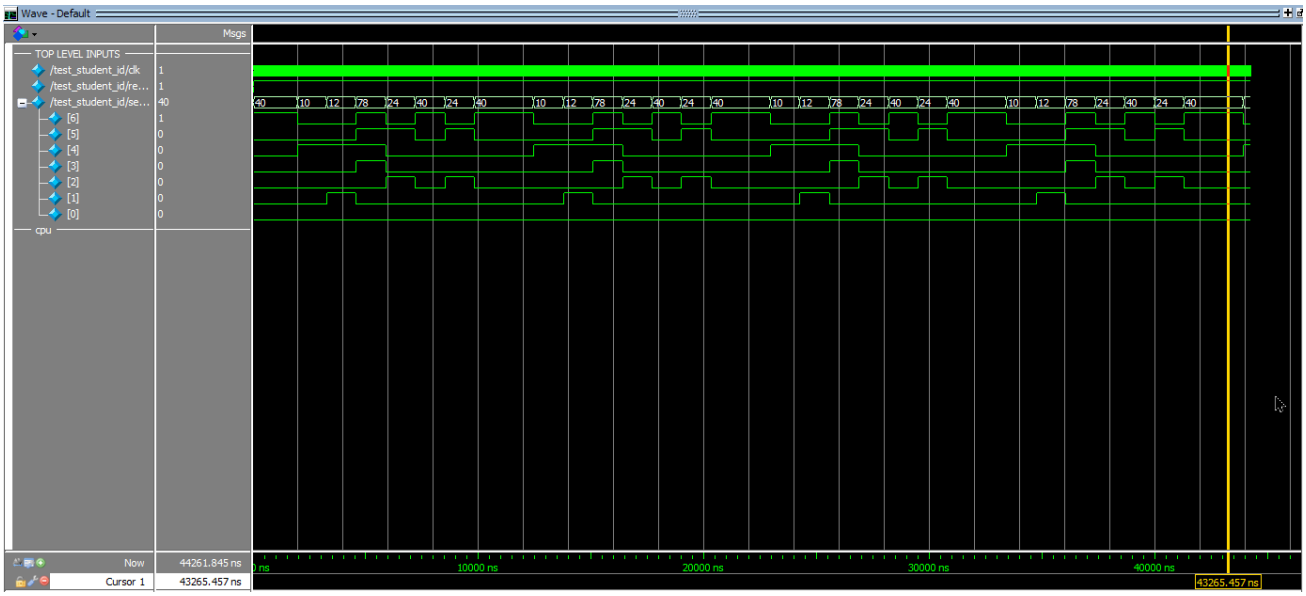
➢ Testbench.sv(test_student_id.sv)

```
module test_student_id;

      logic clk, reset, reset_div;
      logic [6:0] seven_seg;

      student_id id1(
            .clk_50M(clk),
            .reset(reset),
            .reset_div(reset_div),
            .out(seven_seg)
      );

      always #5 clk = ~clk;

      initial begin
            clk = 0; reset = 0; reset_div = 0;
            #10   reset_div = 1; reset = 1;
            #1000       $stop;
      end

endmodule
```

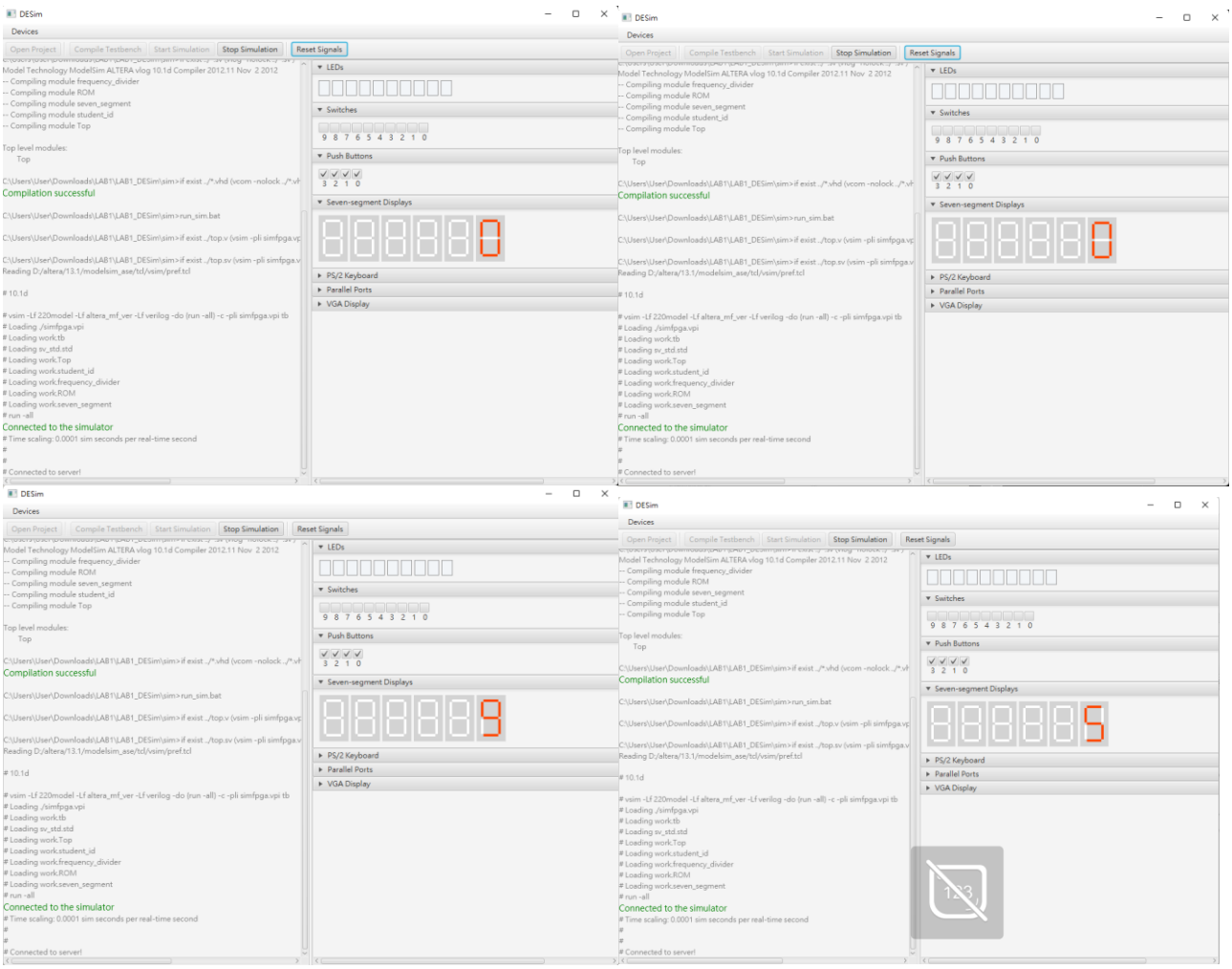
Testbench 先將初值 initial 成 clk = 0, reset = 0, reset_div = 0

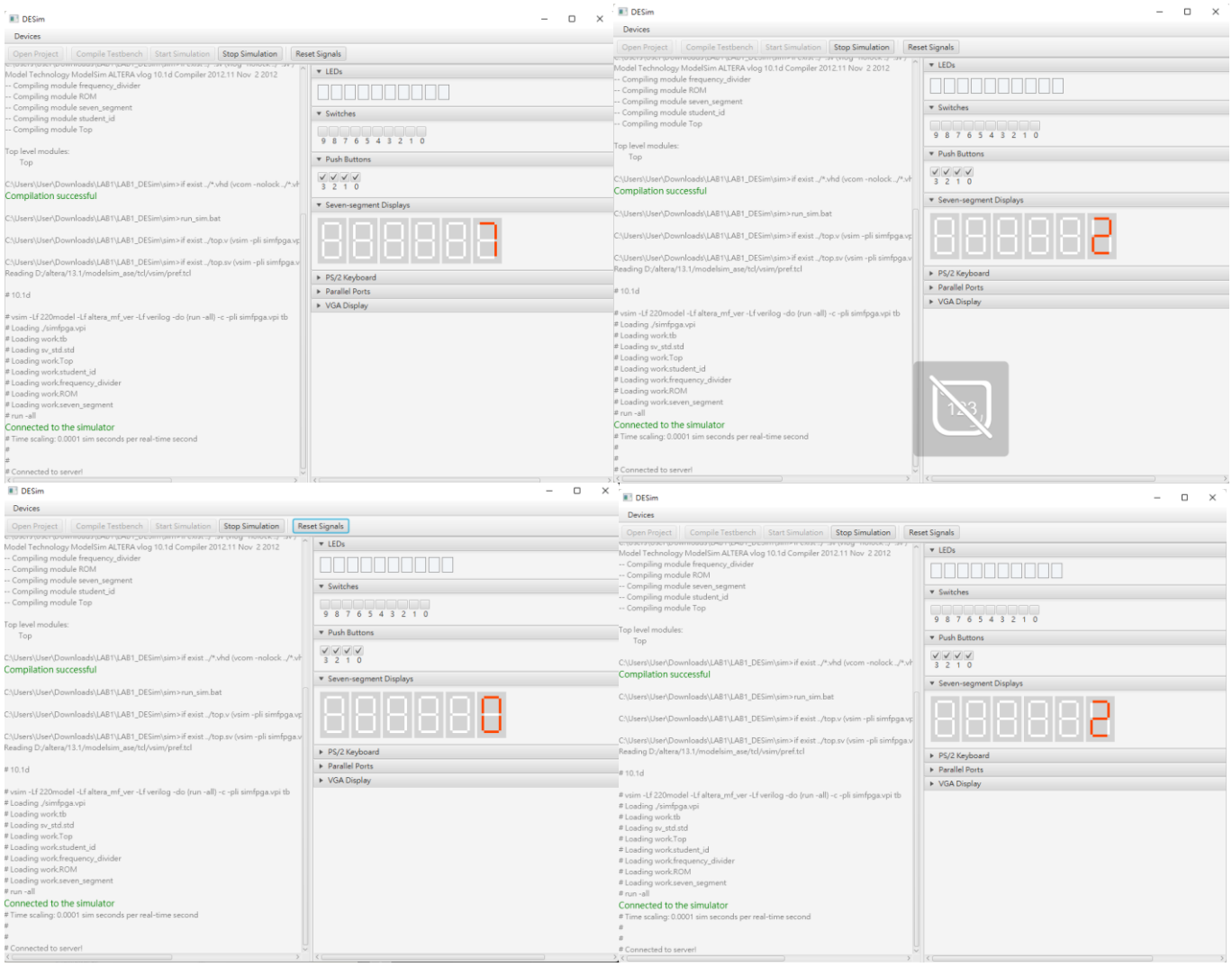
經過 10 個單位時間後將 reset_div = 1, reset =1 達到 reset 的效果，且每 5 個單位時間轉換一次 clk。

模擬結果與結果說明：



從模擬結果可以看出，學號對應的七段顯示器編碼正確。





從 DeSim 可以看到模擬結果為正確的學號。

● 結論與心得：

這次是本堂課的第一次實驗，上課時有講解基礎的 Verilog 語法，與上課專案使用方法，且最令我覺得特別的是，接觸到了 DeSim 這套軟體，可以方便在家查看燒錄結果。但在教室裡嘗試利用 DeSim 時，出現不知道的 Bug，原本有將 Rom 裡面的值全部改成自己的學號去跑，但畫面中只顯示 0，ModelSim 波型也顯示正常，到家依照同樣的步驟做一次才成功。