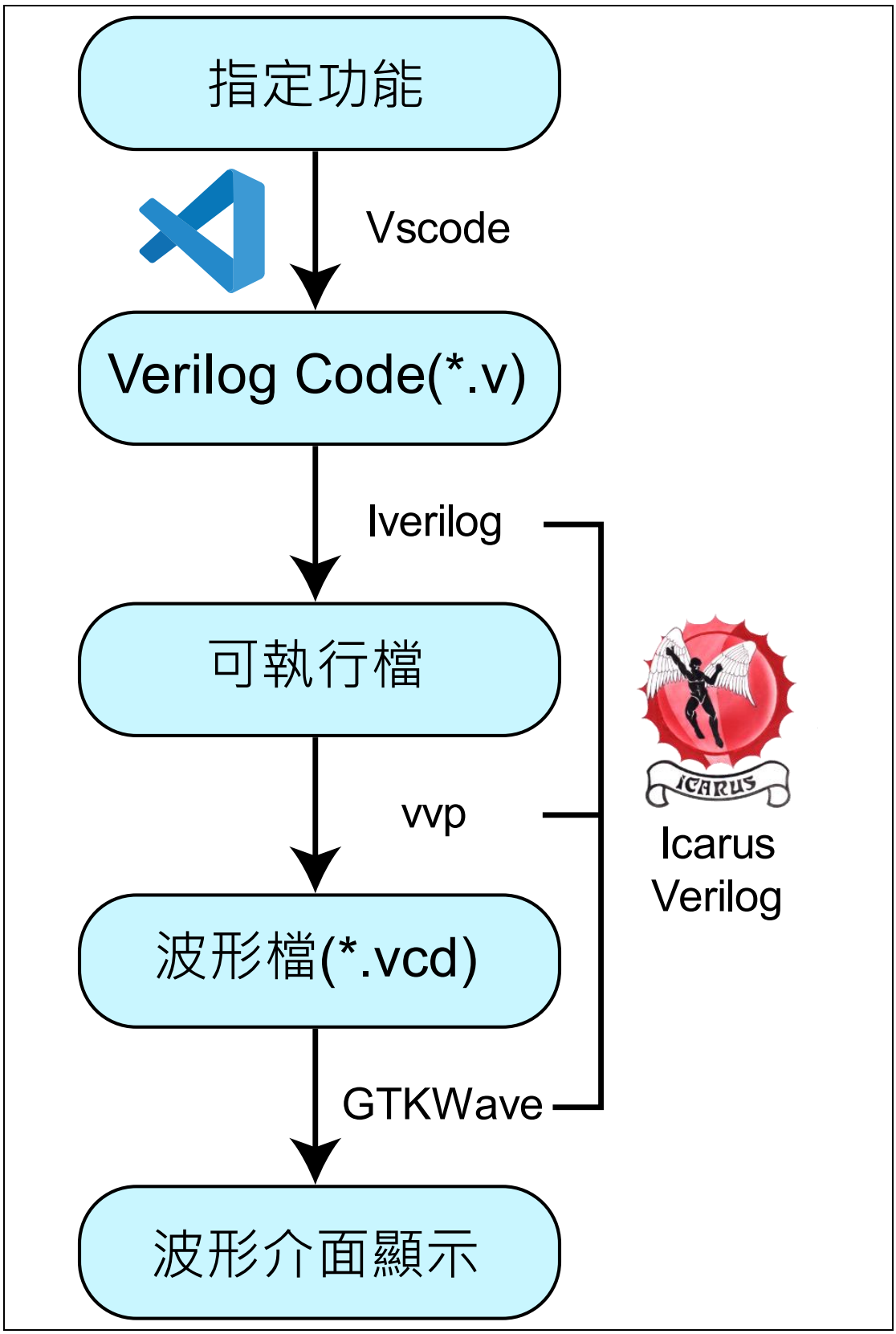
邏輯系統實驗

Lab 11

2021/05/20(四)

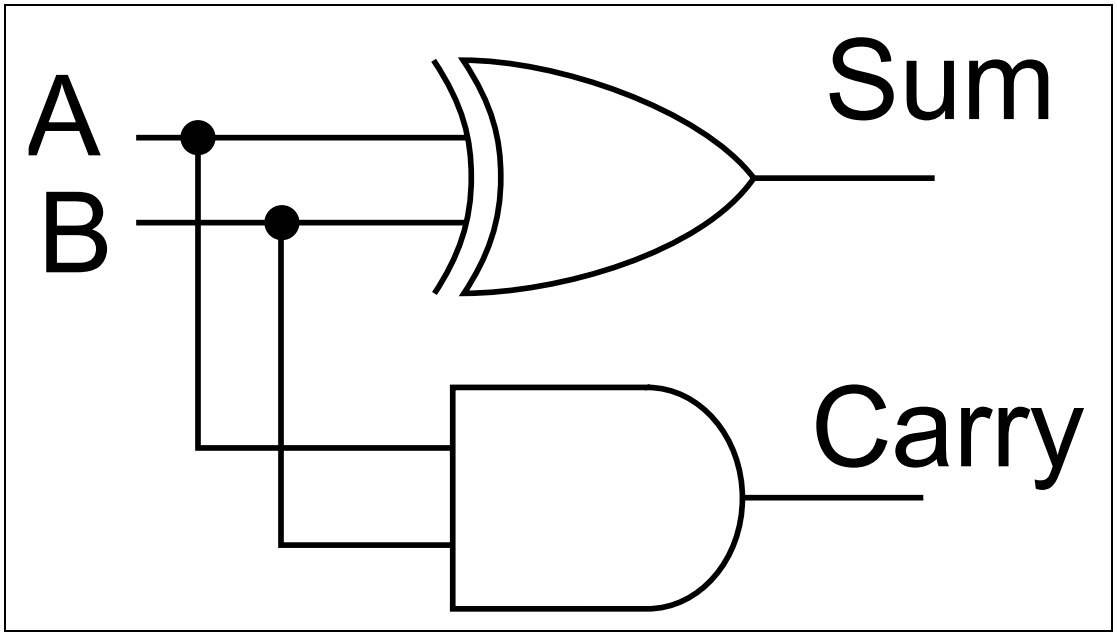
第 1 組	
組員姓名	學號
陳旭祺	E24099059
張振杰	E24085034
何啟造	E34085337

- **實驗流程圖**


指定功能
Vscode
Verilog Code(*.v)
Iverilog
可執行檔
vvp
波形檔(*.vcd)
GTKWave
波形介面顯示
Icarus
Verilog

- 實作題(一):

1. Schematic View



2. Verilog Code

```
module halfadder (A,B,Carry,Sum);
    input A,B;
    output Carry, Sum;

    assign Carry = A & B;
    assign Sum = A ^ B;
endmodule
```

3. Testbench for Verilog Code

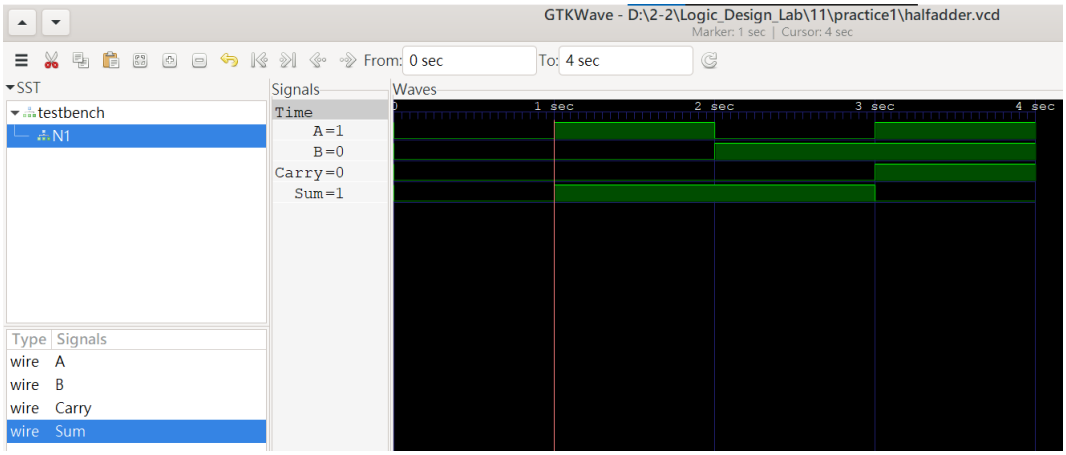
```
module testbench ();
    reg A, B;
    wire carry, sum;
    halfadder N1(A,B, carry, sum);

    initial begin
        //--將 halfaddr 的訊號寫到.vcd 檔--
            $dumpfile("halfadder.vcd");
            $dumpvars(0, N1);
```

```
        //----------------------------
            A = 0; B = 0;
        #1  A = 1; B = 0;
        #1  A = 0; B = 1;
        #1  A = 1; B = 1;
        #1  $finish;
    end
endmodule
```

4. Simulation Result

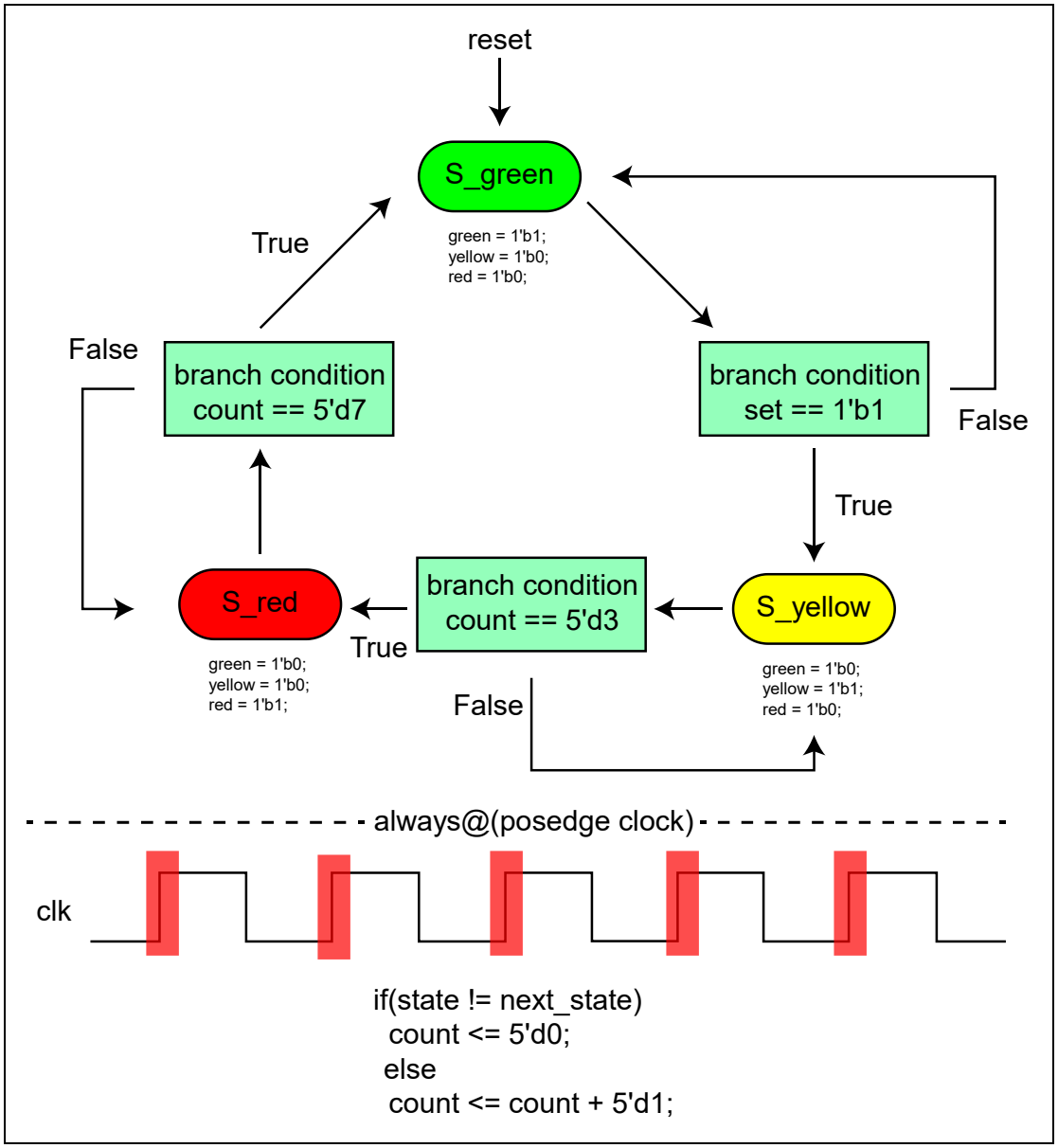
Explanation of Waveform



同 Lab4-1 Half Adder

- **實作題(二):**

1. Schematic View



2. Verilog Code

```
module lab11_1(clock,reset,set,green,yellow,red);

    input clock;
    input reset;
    input set;

    output green;
    output yellow;
    output red;

    reg [1:0]   state,next_state;
    reg         green,yellow,red;
    reg [4:0]   count;

    always @(posedge clock) begin
        state <= next_state;
    end

    always @(state or reset or set or count) begin
        if(reset == 1'b1)
            next_state = 2'd0;
        else begin
            case(state)
                2'd0:
                    if(set == 1'b1)
                        next_state = 2'd1;
                    else
                        next_state = 2'd0;
                2'd1:
                    if(count==5'd3)
                        next_state = 2'd2;
                    else
                        next_state = 2'd1;
                2'd2:
                    if(count==5'd7)
                        next_state = 2'd0;
```

```
                else
                    next_state = 2'd2;
            default:next_state = 2'd0;
        endcase
    end
end

always @(state) begin
    case(state)
        2'd0: begin
            green = 1'b1;
            yellow = 1'b0;
            red = 1'b0;
        end
        2'd1: begin
            green = 1'b0;
            yellow = 1'b1;
            red = 1'b0;
        end
        2'd2: begin
            green = 1'b0;
            yellow = 1'b0;
            red = 1'b1;
        end
        default: begin
            green = 1'b0;
            yellow = 1'b0;
            red = 1'b0;
        end
    endcase
end

always @(posedge clock) begin
    if(state != next_state)
        count <= 5'd0;
    else
        count <= count + 5'd1;
```

```
    end

endmodule
```

3. Testbench for Verilog Code

```
`timescale 100ms / 1ms
`define CYC 2.5
module testbench ();

    reg clock;
    reg reset;
    reg set;
    wire green,yellow,red;
    integer counter = 0;

    always #(`CYC/2) clock = ~clock;
    initial clock = 1'b1;

    lab11_1 lb11_1(clock,reset,set,green,yellow,red);

    always @(posedge clock) begin
        counter <= counter + 1;
    end

    initial begin
        $dumpfile("trafficlight.vcd");
        $dumpvars(0,testbench);
        reset = 1 ;
        set = 0;
        #(`CYC) reset = 0;
        #(`CYC) begin
            set = 1;
            //counter = -1;
        end
        #(`CYC) set = 0;
        #(17*`CYC) set = 1;
        #(3*`CYC) set = 0;
```

```
        #(5*`CYC) reset = 1;
        #(5*`CYC) $finish;
    end

    always@(posedge clock) begin
        if(green == 1'b1)
            $display("(%d/4)s:GREEN",counter);
        else if(yellow == 1'b1)
            $display("(%d/4)s:YELLOW",counter);
        else if(red == 1'b1)
            $display("(%d/4)s:RED",counter);
        else
            $display("(%d/4)s:NO LIGHT",counter);
    end

endmodule
```

4. Simulation Result

Terminal

```
PS D:\2-2\Logic_Design_Lab\11\2> vvp lab11
VCD info: dumpfile trafficlight.vcd opened for output.
(          0/4)s:NO LIGHT
(          1/4)s:GREEN
(          2/4)s:YELLOW
(          3/4)s:YELLOW
(          4/4)s:YELLOW
(          5/4)s:YELLOW
(          6/4)s:RED
(          7/4)s:RED
(          8/4)s:RED
(          9/4)s:RED
(         10/4)s:RED
(         11/4)s:RED
(         12/4)s:RED
(         13/4)s:RED
(         14/4)s:GREEN
(         15/4)s:GREEN
(         16/4)s:GREEN
(         17/4)s:GREEN
(         18/4)s:GREEN
(         19/4)s:GREEN
(         20/4)s:YELLOW
(         21/4)s:YELLOW
(         22/4)s:YELLOW
(         23/4)s:YELLOW
(         24/4)s:RED
(         25/4)s:RED
(         26/4)s:RED
(         27/4)s:RED
(         28/4)s:GREEN
(         29/4)s:GREEN
(         30/4)s:GREEN
(         31/4)s:GREEN
testbench.v:34: $finish called at 8250 (1ms)
(         32/4)s:GREEN
```

Testbench 最後的程式為

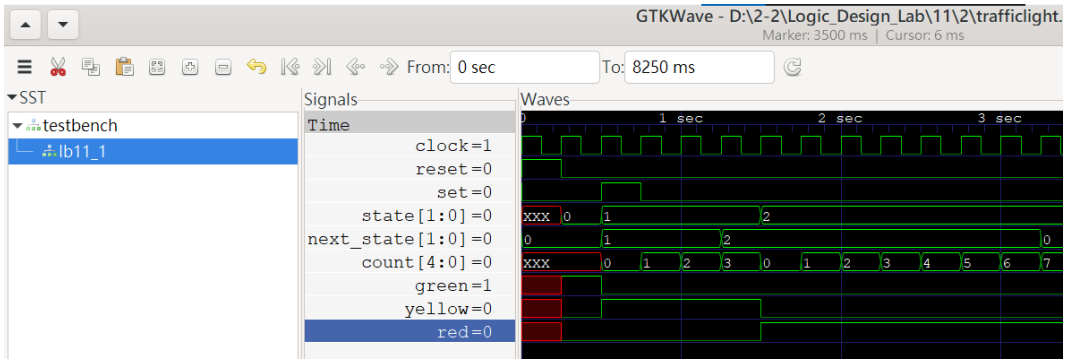
```
if(green == 1'b1)
            $display("(%d/4)s:GREEN",counter);
```

而 counter 定義為 integer 且正緣觸發 counter <= counter + 1;

當$green == 1'b1$ 則 display ”GREEN”文字在終端機上，方便我們去 trace 模擬結果

Explanation of Waveform

功能介紹	
啟動按鈕(set)	按下按鈕時紅綠燈會從綠燈轉為黃燈，紅燈，再轉回綠燈
初始化按鈕(reset)	將紅綠燈初始到綠燈的狀態

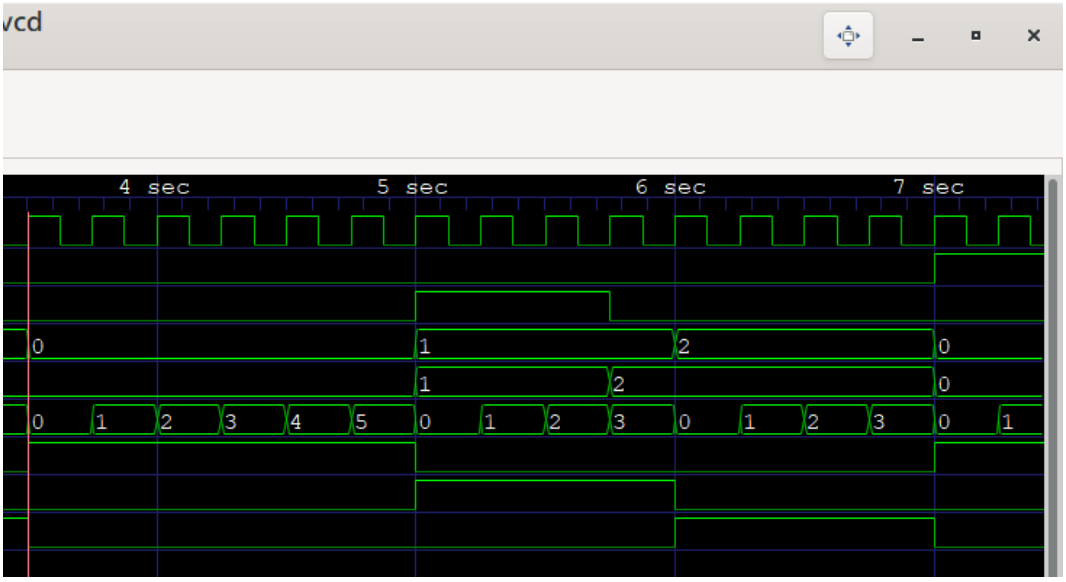


$Reset$ == 1，初始化 state 為 0，此時綠燈。

接下來Set == 1，state 由 0(綠燈)轉 1(黃燈)，

數到$count$ == $5'd3$時 state 再轉為 2(紅燈)，

再數到$count$ == $5'd7$時 state 再轉為 0(綠燈)，也就是原本的狀態



接下來一樣Set == 1，state 由 0(綠燈)轉 1(黃燈)，

數到$count$ == $5'd3$時 state 再轉為 2(紅燈)，

但當 count 只數到 3 時$Reset$ == 1，因此會直接重置為 state 為 0(綠燈)

● 心得

1. 組員一 陳旭祺

由於疫情加劇，實作課改為在家完成，因此不能用 FPGA 去做電路合成擬真，改用一些免費開源的 EDA tools 去跑模擬。實驗一沿用 Lab4-1 Half Adder 的程式去測試實驗環境的安裝，主要學習 **Icarus Verilog(iverilog+GTKWave)**這個仿真工具，以下為該軟體介紹: 全平台轻量开源 verilog 仿真工具 iverilog+GTKWave 使用教程、Icarus Verilog User Guide、Icarus Verilog 官方網站，Icarus Verilog 編譯器主要包含 3 個工具:

iverilog	編譯 Verilog 和 VHDL 文件，進行語法檢查，生成可執行文件
vvp	根據可執行文件，生成仿真波形文件
gtkwave	打開仿真波形文件，圖形化顯示波形

基本的下指令參數類似 gcc 編譯器

$-o$	指定生成文件的名稱
$-y$	指定包含文件夾
$-l$	`include 語句包含了頭文件路徑

值得注意的是 testbench 文件中有幾行 iverilog 編譯器專用的語句，如果不加的話後面不能生成 vcd 文件

```
initial begin
        $dumpfile("trafficlight.vcd"); //生成的 vcd 文件名稱
        $dumpvars(0,testbench);        //testbench 模組名稱
end
```

2. 組員二 張振杰

這次實驗主要是用 verilog 模擬出紅綠燈的表現，我對於講義上的某些 code 的功能還是語法不了解， 就一小塊一小塊的 code 來看的話會比較清楚，但在整體的 code 之間的銜接對我來說有點混亂，在腦中要自己想象出來說實話有點困難，因為概念沒有到很清楚。這次實驗我也才知道原來 Verilog 可以在 VSCode 裡面先寫好，然後在 run 的時候可以以 Verilog 的視窗顯示結果，我查了一下原來是因為可以多種環境

3. 組員三 何啟造

這是進入線上課程後的第一個實驗。它是讓我們先安裝好編寫 Verilog 的環境然後再做出“紅綠燈”的效果。看了很久後來理解就覺得還好。