

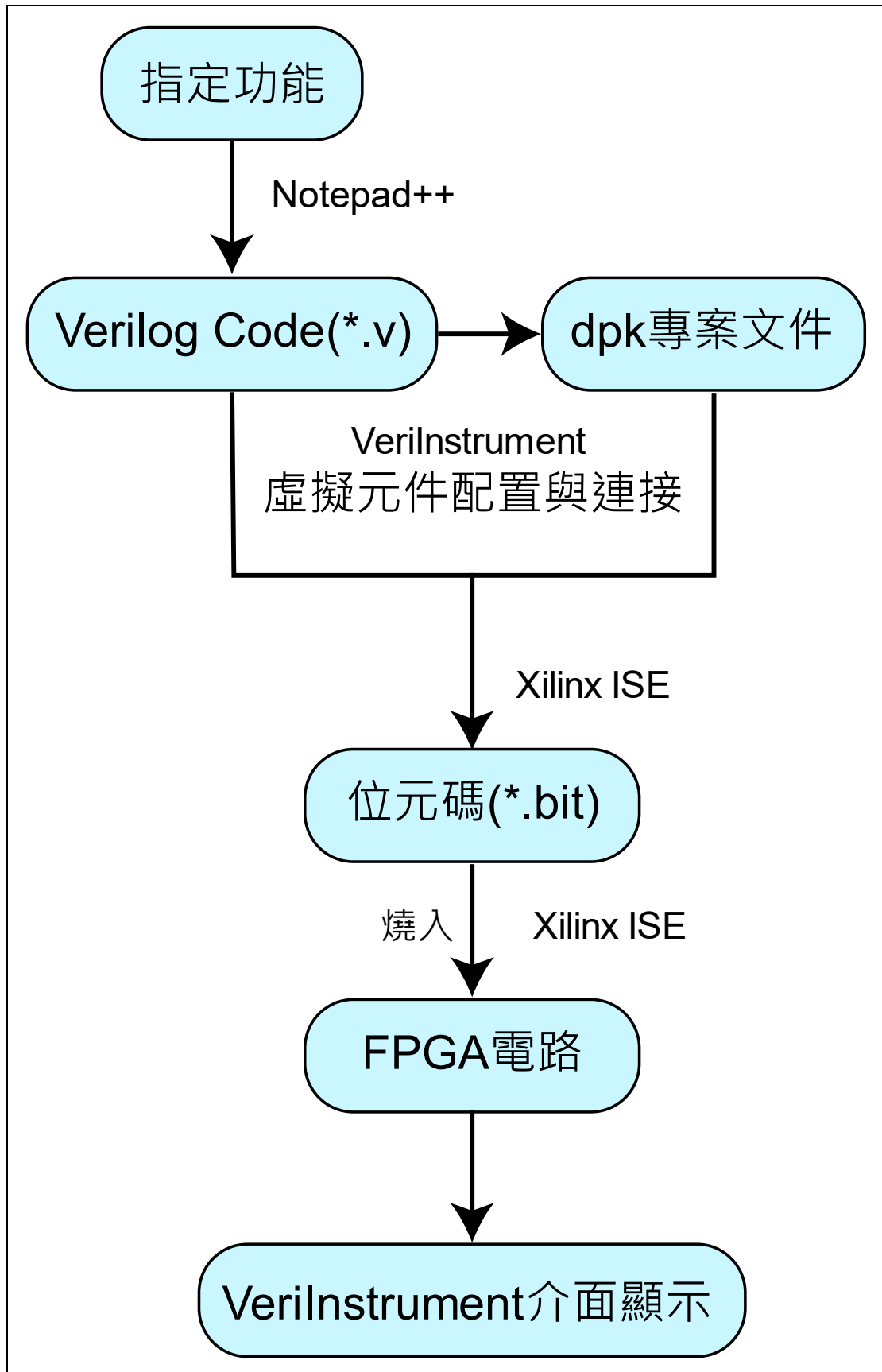
邏輯系統實驗

Lab 8

2021/04/29(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(同上次實驗 Lab7)



● 實作題(一): 四位元加法器

1. Verilog Code

adder4bits.v
<pre> module adder4bits(A, B, Carry, Sum); input [3:0] A, B; output Carry; output [3:0] Sum; wire [4:0] result; assign result = A + B; assign Carry = result[4]; assign Sum = result[3:0]; endmodule </pre>
SevenSeg.v
<pre> module SevenSeg(din,dout); input [3:0] din; output reg[6:0] dout; always@(din) begin case(din) 4'b0000:dout = 7'b0111111; 4'b0001:dout = 7'b0000110; 4'b0010:dout = 7'b1011011; 4'b0011:dout = 7'b1001111; 4'b0100:dout = 7'b1100110; 4'b0101:dout = 7'b1101101; 4'b0110:dout = 7'b1111101; 4'b0111:dout = 7'b0000111; 4'b1000:dout = 7'b1111111; 4'b1001:dout = 7'b1101111; 4'b1010:dout = 7'b1110111; 4'b1011:dout = 7'b1111100; 4'b1100:dout = 7'b0111001; 4'b1101:dout = 7'b1011110; 4'b1110:dout = 7'b1111001; 4'b1111:dout = 7'b1110001; endcase end endmodule </pre>

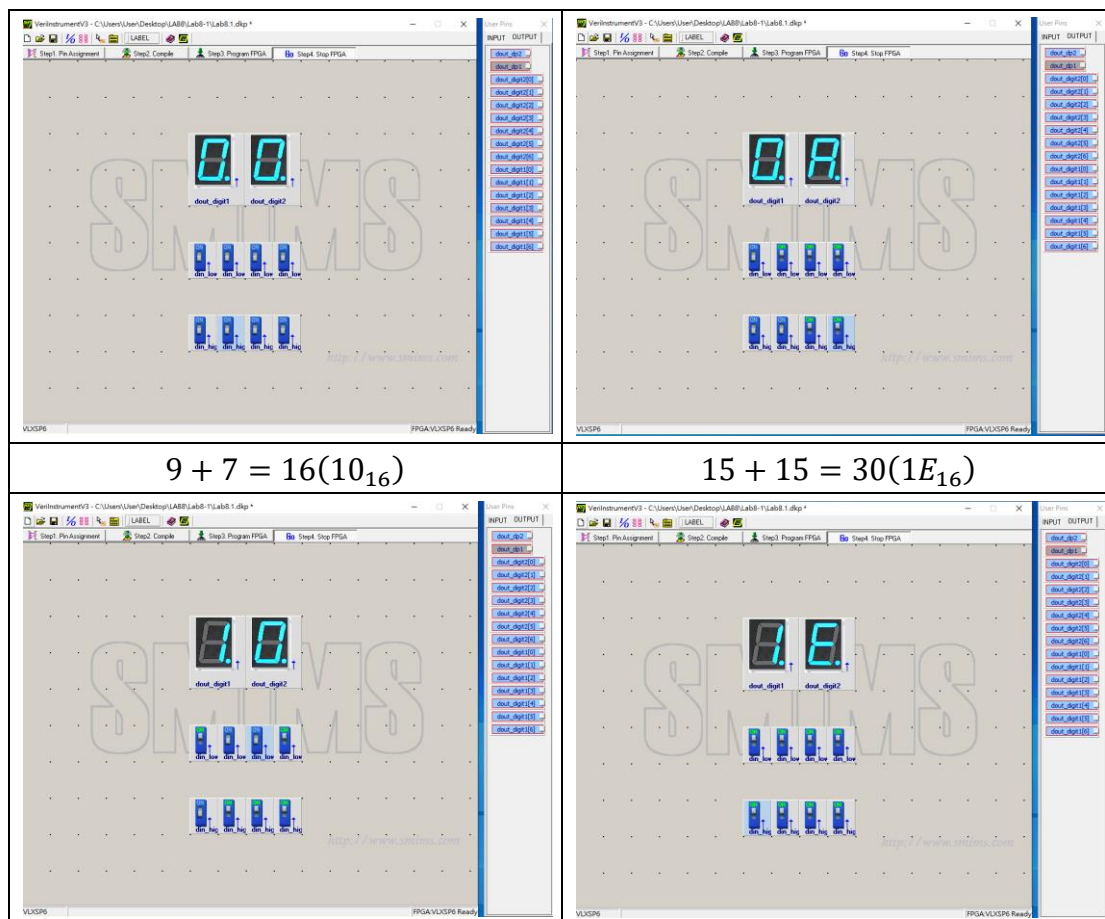
邏輯系統實驗 Lab8 第 1 組

<pre> default:dout = 7'b0000000; endcase end endmodule </pre>
Lab8_1.v
<pre> module Lab8_1 (din_high,din_low,dout_digit1,dout_digit2,dout_dp1,dout_dp2,clk); input clk; input [3:0] din_high; input [3:0] din_low; output [6:0] dout_digit1; output [6:0] dout_digit2; output dout_dp1; output dout_dp2; wire [7:0] value; adder4bits u1(.A(din_high), .B(din_low), .Carry(value[4]), .Sum(value[3:0])); SevenSeg u_digit1(value[7:4],dout_digit1); SevenSeg u_digit2(value[3:0],dout_digit2); assign value[7:5] = 3'b000; assign dout_dp1 = 1'b1; assign dout_dp2 = 1'b1; endmodule </pre>

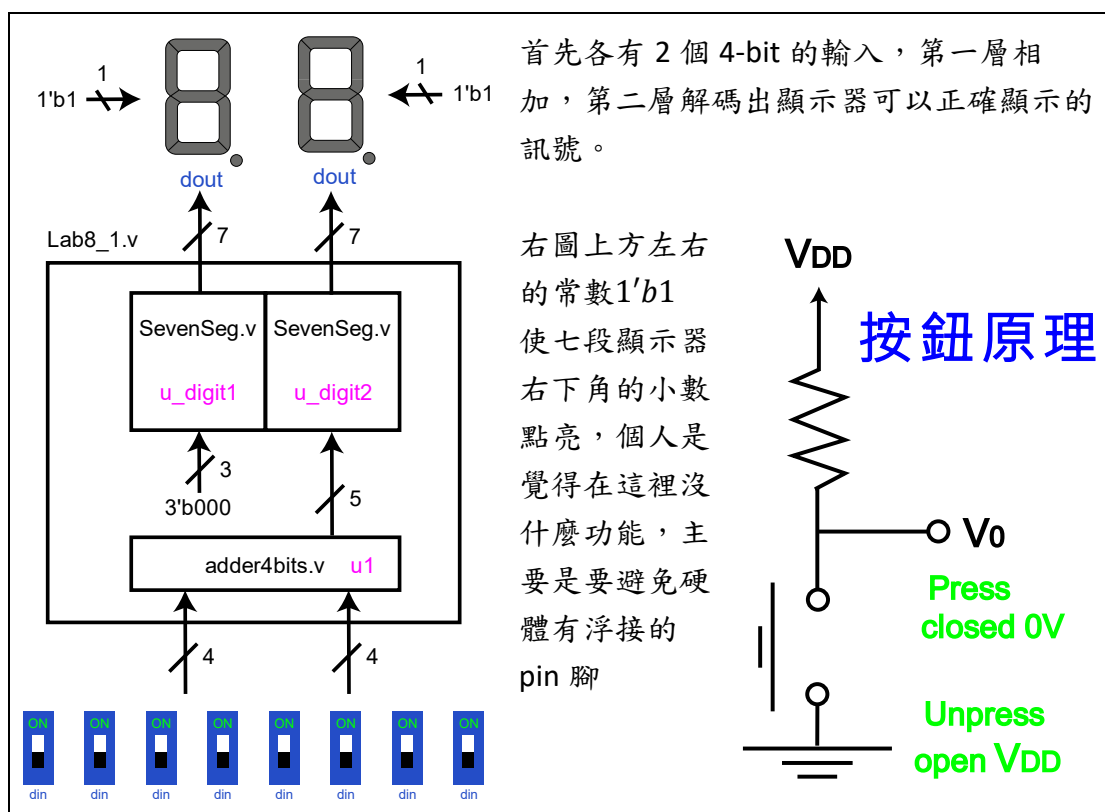
2. FPGA result

0 + 0 = 0	7 + 3 = 10
-----------	------------

邏輯系統實驗 Lab8 第 1 組



3. 硬體架構與功能解釋



● 實作題(二):多工器

1. Verilog Code

```
`include "SevenSeg.v" "adder4bits.v"

module Lab8_2
(sel,din_high,din_low,dout_digit1,dout_digit2,dout_dp1,dout_dp2,clk);
    input sel,clk;
    input [3:0] din_high;
    input [3:0] din_low;

    output [6:0] dout_digit1;
    output [6:0] dout_digit2;
    output dout_dp1;
    output dout_dp2;

    wire [7:0] value_in;
    wire [7:0] value_sum;
    wire [7:0] result;

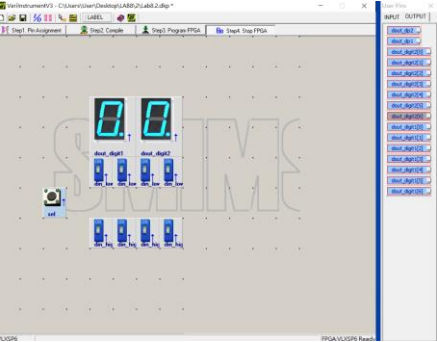
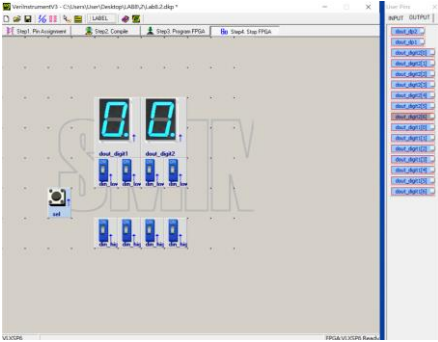
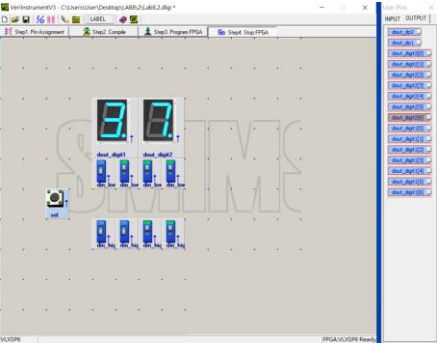
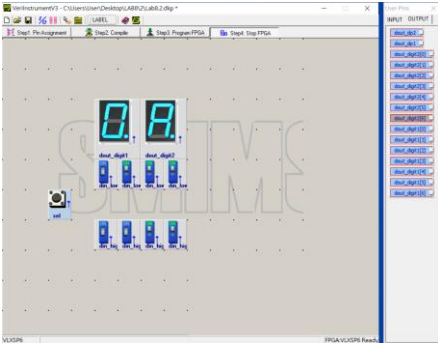
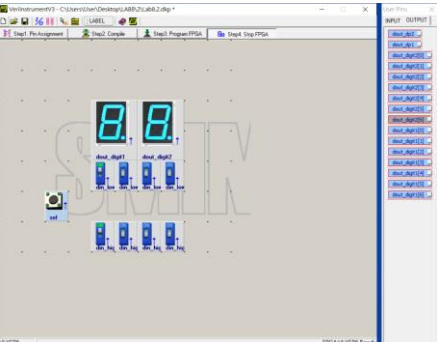
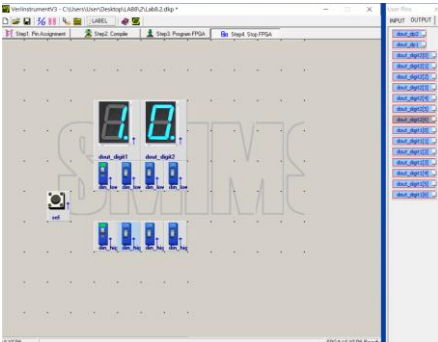
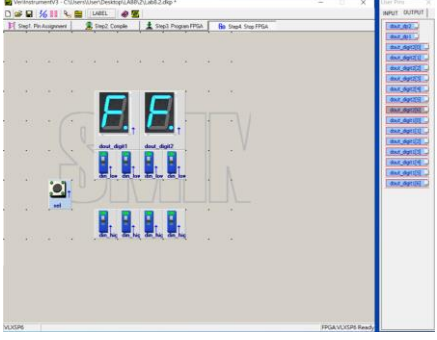
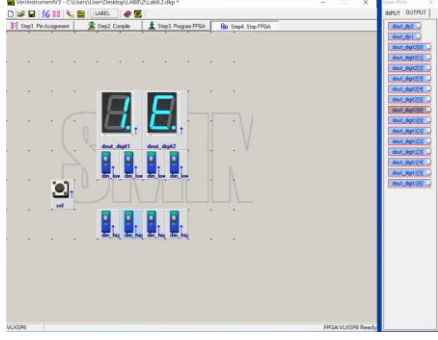
    adder4bits u1(
        .A      (din_high),
        .B      (din_low),
        .Carry   (value_sum[4]),
        .Sum     (value_sum[3:0])
    );

    assign value_sum[7:5] = 3'b000;
    assign value_in = {din_high,din_low};
    assign result = (sel == 1'b0)? value_sum : value_in;

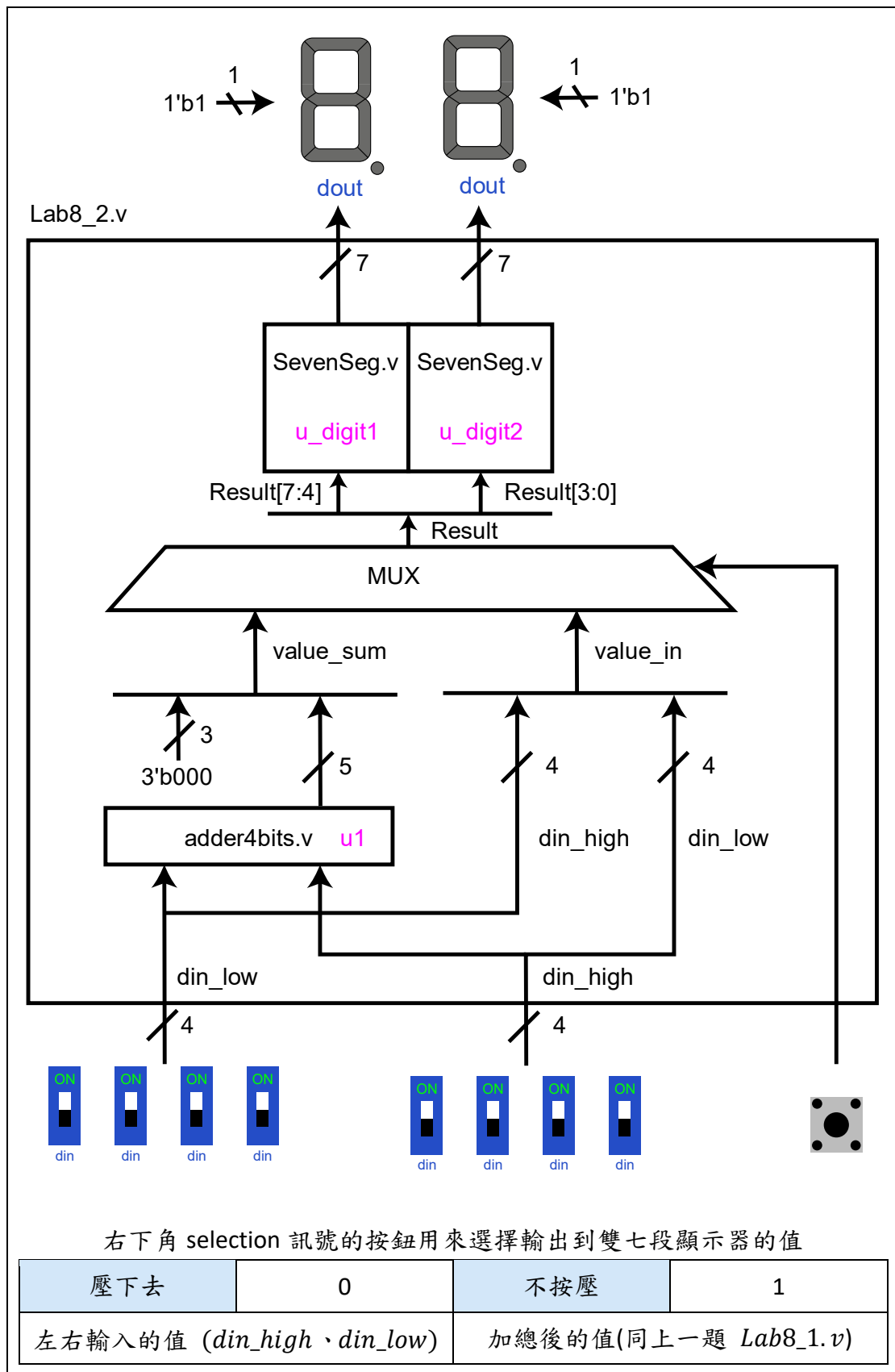
    SevenSeg u_digit1(result[7:4],dout_digit1);
    SevenSeg u_digit2(result[3:0],dout_digit2);

    assign dout_dp1 = 1'b1;
    assign dout_dp2 = 1'b1;
endmodule
```

2. FPGA result

Input($sel = 0$)	Output($sel = 1$)
$0 + 0$ 	0 
$3 + 7$ 	$10(A_{16})$ 
$8 + 8$ 	$16(10_{16})$ 
$15 + 15(F_{16} + F_{16})$ 	$30(1E_{16})$ 

3. 硬體架構與功能解釋



● 實作題(三):Case 語法應用

1. Verilog Code

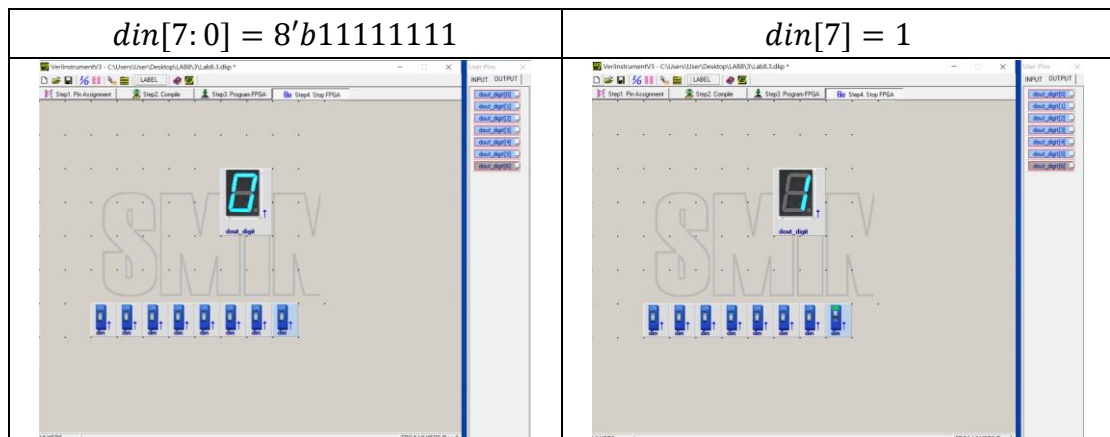
```
`include "SevenSeg.v"

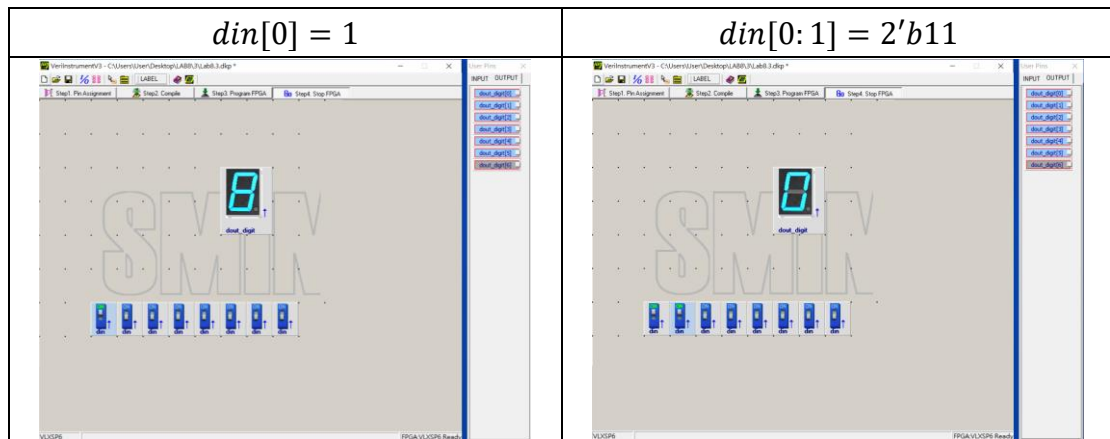
module Lab8_3 (din,dout_digit,clk);
    input      clk;
    input  [7:0] din;
    output [6:0] dout_digit;
    reg  [3:0] value;

    always@(din) begin
        case (din)
            8'b10000000: value = 4'b0001;
            8'b01000000: value = 4'b0010;
            8'b00100000: value = 4'b0011;
            8'b00010000: value = 4'b0100;
            8'b00001000: value = 4'b0101;
            8'b00000100: value = 4'b0110;
            8'b00000010: value = 4'b0111;
            8'b00000001: value = 4'b1000;
            default: value = 4'b0000;
        endcase
    end

    SevenSeg u_digit(value,dout_digit);
endmodule
```

2. FPGA result





3. 硬體架構與功能解釋

用Case去列一個個條件，在七段顯示器顯示Switch 1 – 8中的哪個 Bit 被設為 1，如果同時有兩個 Bit 以上的Switch被設為 1，則七段顯示器顯示 0

● 實作題(四):比較器

1. Verilog Code

```
`include "SevenSeg.v"

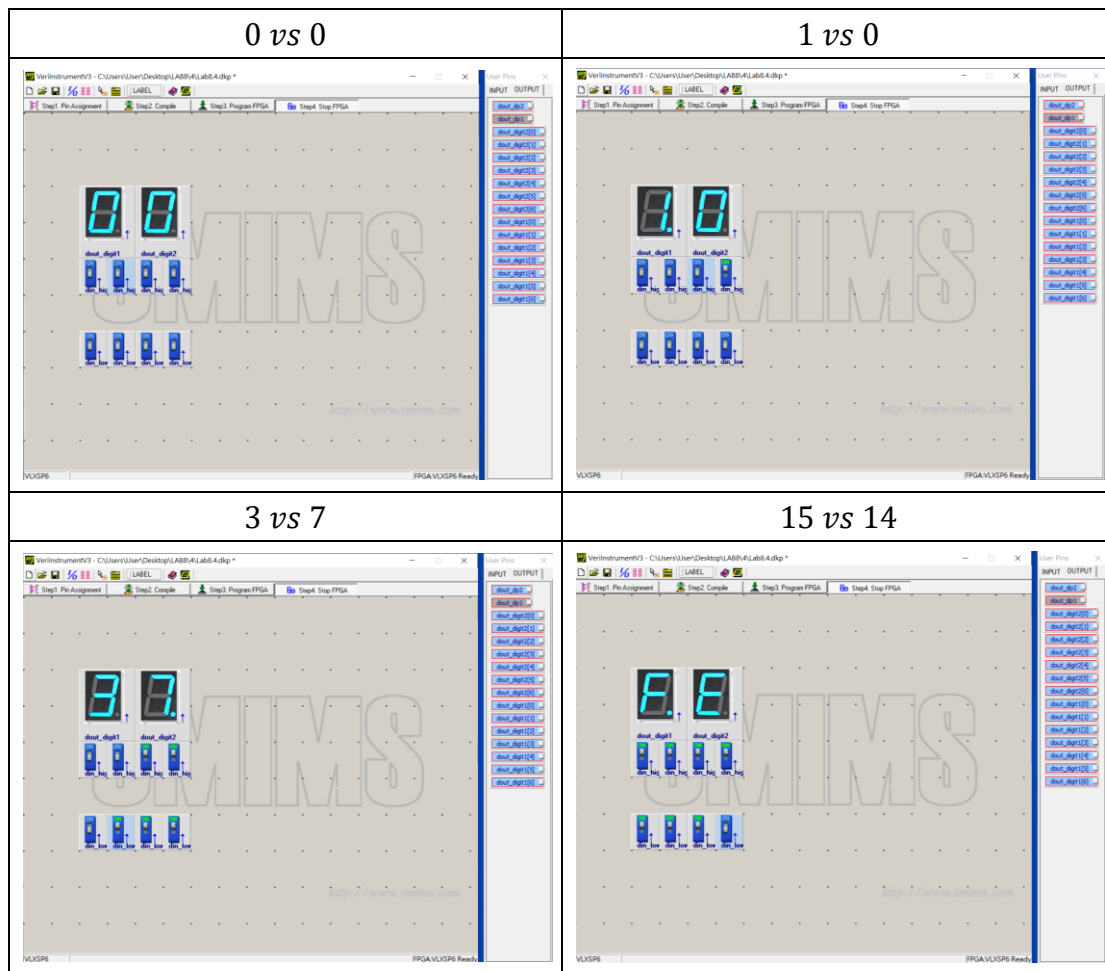
module Lab8_4
(din_high,din_low,dout_digit1,dout_digit2,dout_dp1,dout_dp2,clk);
    input      clk;
    input [3:0] din_high;
    input [3:0] din_low;

    output [6:0] dout_digit1;
    output [6:0] dout_digit2;
    output      dout_dp1;
    output      dout_dp2;

    SevenSeg u_digit1(din_high,dout_digit1);
    SevenSeg u_digit2(din_low ,dout_digit2);

    assign dout_dp1 = (din_high > din_low)? 1'b1 : 1'b0;
    assign dout_dp2 = (din_high < din_low)? 1'b1 : 1'b0;
endmodule
```

2. FPGA result



3. 硬體架構與功能解釋

2 個輸入輸出至雙七段顯示器	
一邊值大	兩值相同
那邊小數點會亮	兩邊都不亮

● 挑戰題(一):排序顯示器

1. Verilog Code

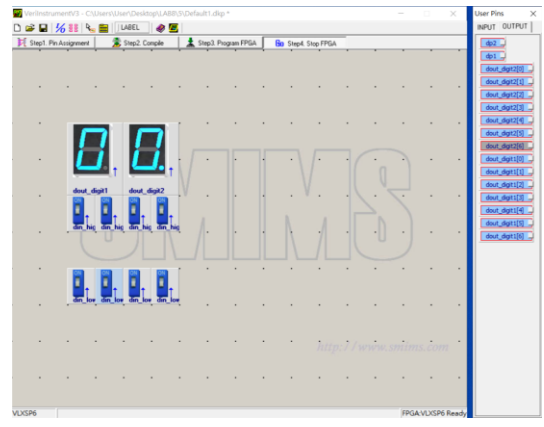
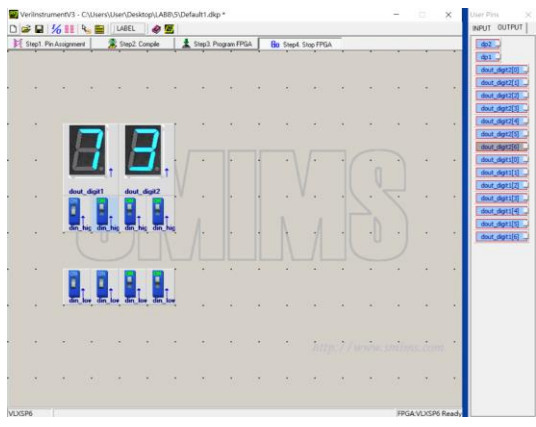
```
`include "SevenSeg.v" "adder4bits.v"

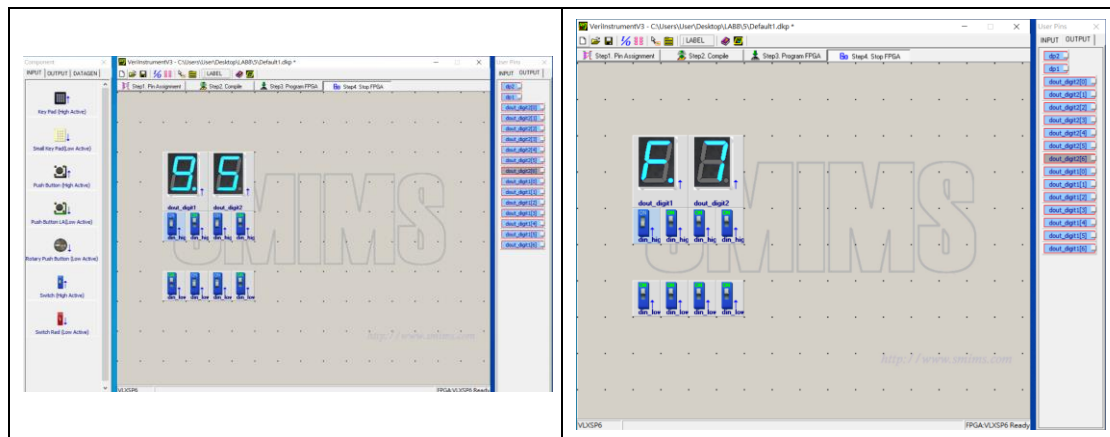
module Lab8_5
(din_high,din_low,dout_digit1,dout_digit2,dp1,dp2,clk);
    input        clk;
    input  [3:0]  din_high;
    input  [3:0]  din_low;
    output        dp1,dp2;
    output [6:0]  dout_digit1;
    output [6:0]  dout_digit2;
    wire  [6:0]  low, high;

    SevenSeg u_digit1(high,dout_digit1);
    SevenSeg u_digit2(low ,dout_digit2);

    assign low = (din_high > din_low)? din_low : din_high;
    assign high= (din_high > din_low)? din_high : din_low;
    assign dp1 = (din_high < din_low)? 1'b1 : 1'b0;
    assign dp2 = (din_high == din_low)? 1'b1 : 1'b0;
endmodule
```

2. FPGA result

0 vs 0	7 vs 3
	
5 vs 9	7 vs 15



3. 硬體架構與功能解釋

輸入的兩個輸入輸出至雙七顯示器， 而digit1顯示較大的值、digit2顯示較小的值	
dp1	dp2
輸入的兩個值是否有交換	兩個值是否相同

● 心得

1. 組員一 陳旭祺

本次實驗同上次是用 FPGA 加上 VeriInstrument 的實驗流程，一共有五個實作功能的部分，四位元加法器(2 個 4-bit 輸入相加)、多工器(選擇顯示相加結果或個別輸入)、Case 語法應用(1-bit 輸入顯示解碼位置、2-bit 以上輸入為 0)、比較器(2 個 4-bit 輸入比大小)、排序顯示器(2 個 4-bit 輸入按照數值大小排序)

2. 組員二 張振杰

這次實驗與上次實驗操作基本相同，這次實驗主要是讓我們了解 verilog 裡的指令使用。我也第一次了解到 switch 指令如果在大於 2 種的 case 都發生時所輸出的值應該是 0，因為正常來說在 coding 裡的 switch 是哪一個 case 有發生就把那個 case 給顯示出來或回傳值。這次實驗也讓我眼前為之一亮。

3. 組員三 何啟造

這次的實驗還是繼續之前的 verilog 實驗，而這次的難度也比較高，是要實作出四位元加法器、多工器、並用這些東西實作出一個排序顯示器。也算是讓我更加了解了 always、case、sel 這些類似於 C++的功能。