邏輯系統實驗 實驗五

組別：7

成員：章子嚴、張軒、魏晉成

學號：E24065018、E24066129、E24066226

內容

實驗內容：

1. 基礎題(一)：4 bit ALU

```
1  module ALU_4bit(in1, in2, sel, result);
2     input[3:0] in1, in2;
3     input[2:0] sel;
4     output reg[3:0] result;
5
6     always@(in1 or in2 or sel)
7     begin
8     case(sel)
9     3'b000:result=in1+in2;
10    3'b001:result=in1-in2;
11    3'b010:result=in1&in2;
12    3'b011:result=in1|in2;
13    3'b100:result=in1^in2;
14    3'b101:result=in1>>in2;
15    3'b110:result=in1<<in2;
16    3'b111:result=in1>in2;
17    endcase
18    end
19 endmodule
```

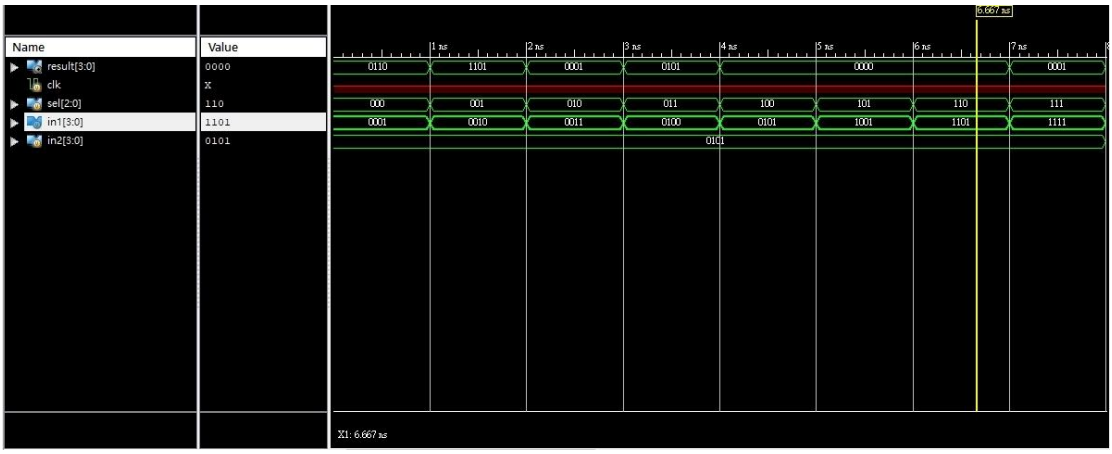
(圖一) 4 bit ALU 的 verilog code

```
1  module system(clk, sel, in1, in2, result);
2     input clk;
3     input[2:0] sel;
4     input[3:0] in1, in2;
5     output[3:0] result;
6
7     ALU_4bit ALU(in1, in2, sel, result);
8
9  endmodule
```

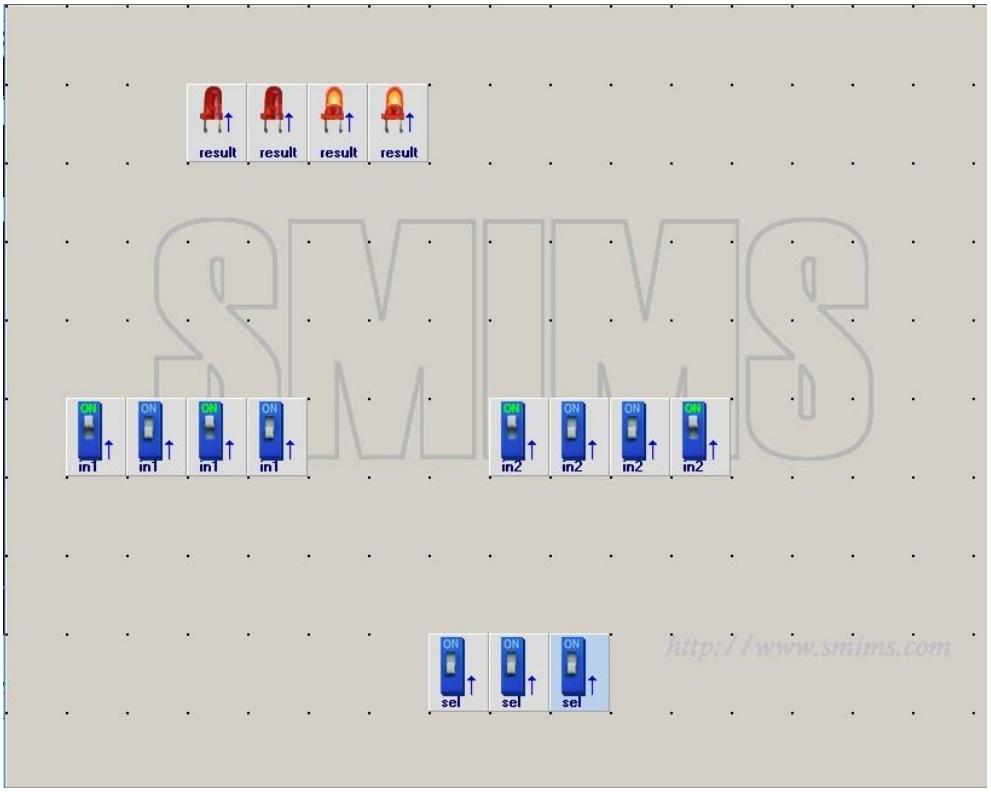
(圖二) 將 4 bit ALU 封裝起來的系統之 verilog code

```
44
45      initial begin
46              sel=3'b000; in1=4'b0001; in2=4'b0101;
47            #1 sel=3'b001; in1=4'b0010; in2=4'b0101;
48            #1 sel=3'b010; in1=4'b0011; in2=4'b0101;
49            #1 sel=3'b011; in1=4'b0100; in2=4'b0101;
50            #1 sel=3'b100; in1=4'b0101; in2=4'b0101;
51            #1 sel=3'b101; in1=4'b1001; in2=4'b0101;
52            #1 sel=3'b110; in1=4'b1101; in2=4'b0101;
53            #1 sel=3'b111; in1=4'b1111; in2=4'b0101;
54            #1 $finish;
55      end
56
57      initial begin
58            $monitor($time, "sel=%d, in1=%d, in2=%d, result=%d", sel, in1, in2, result);
59      end
60
61  endmodule
62
63
```

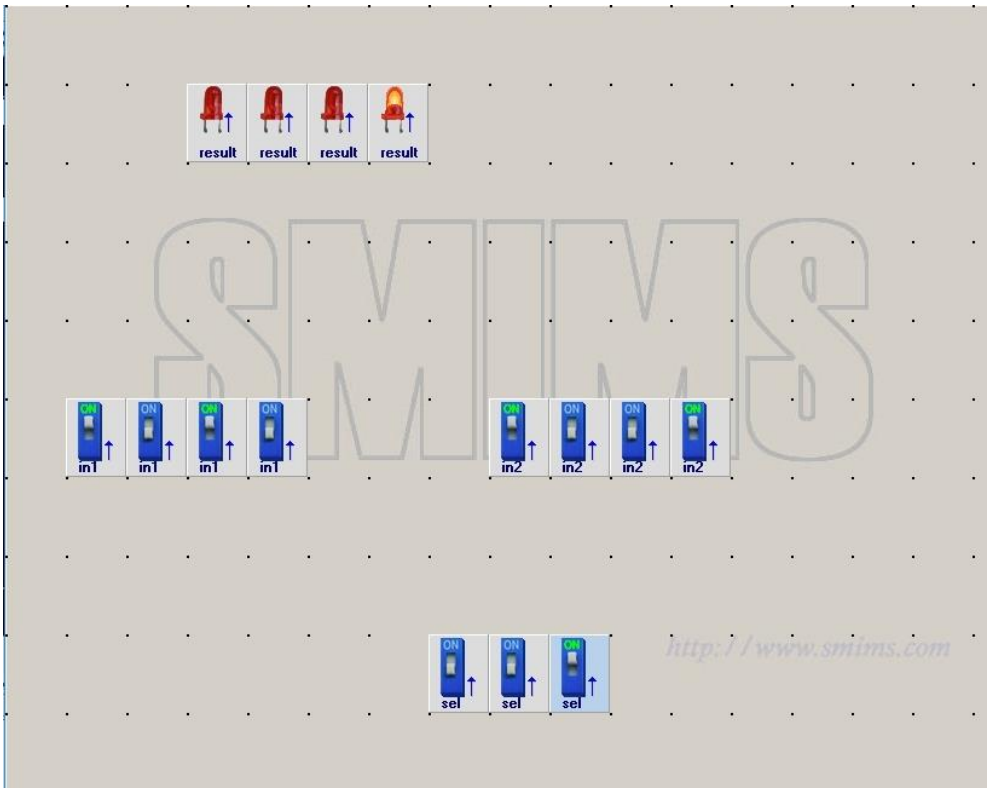
(圖三)系統之 test bench



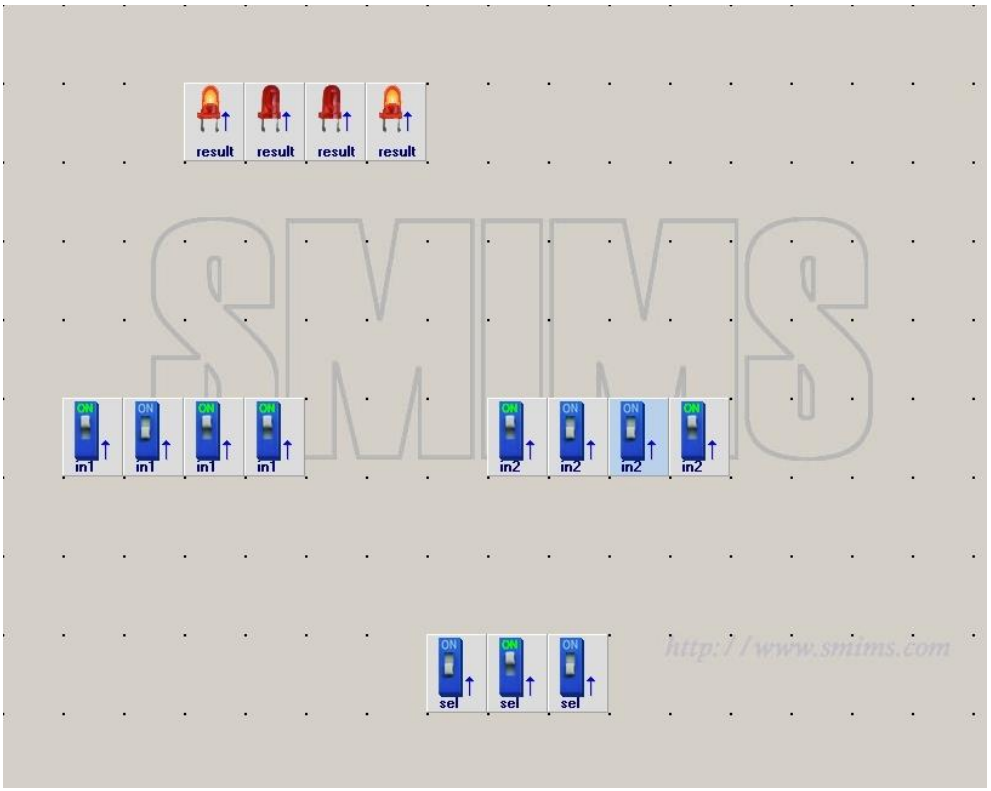
(圖四)在 FPGA 上驗證所得到的波形圖



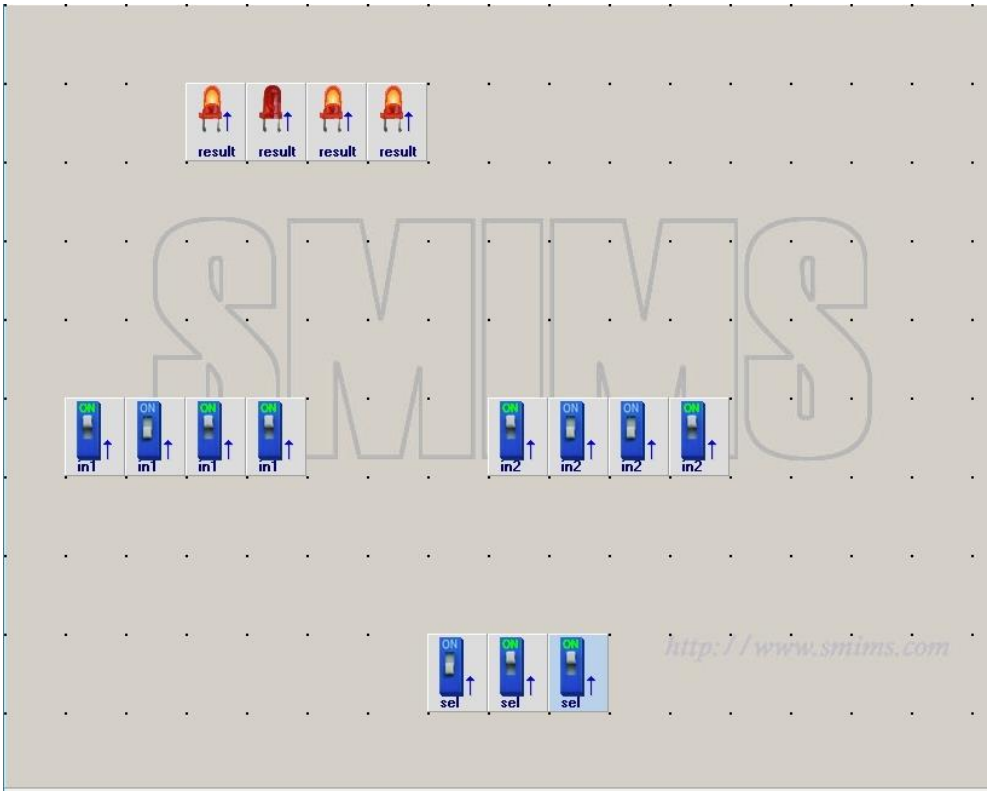
(圖五)利用 VeriInstrucment 進行驗證-1



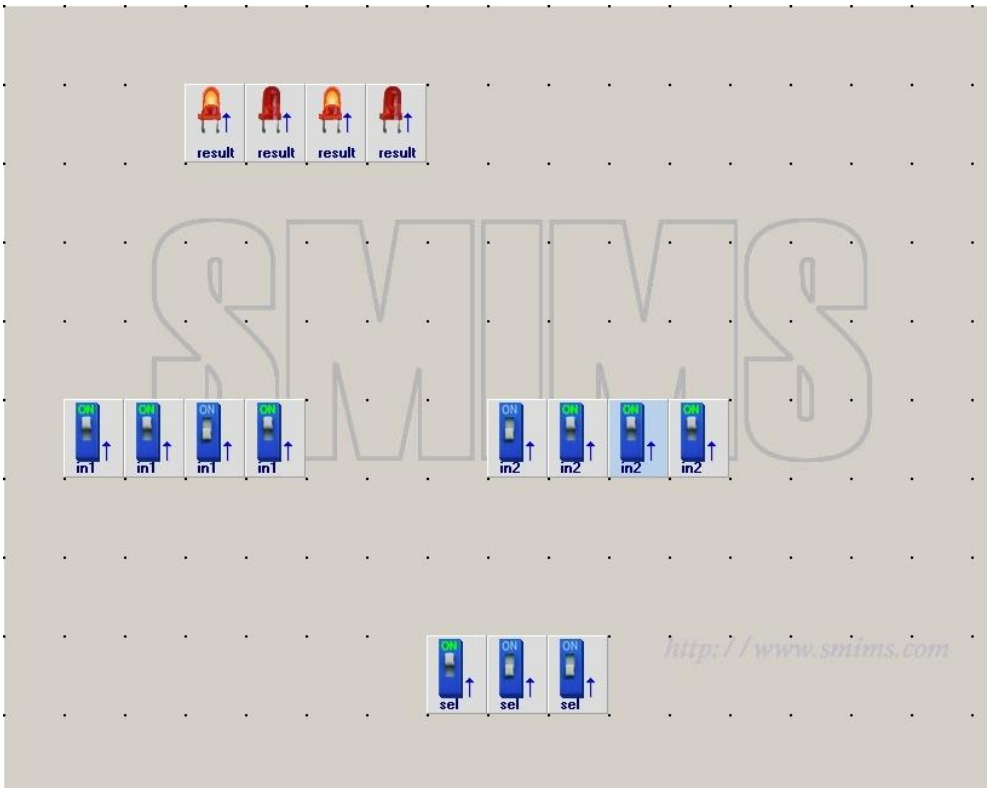
(圖六)利用 VeriInstrucment 進行驗證-2



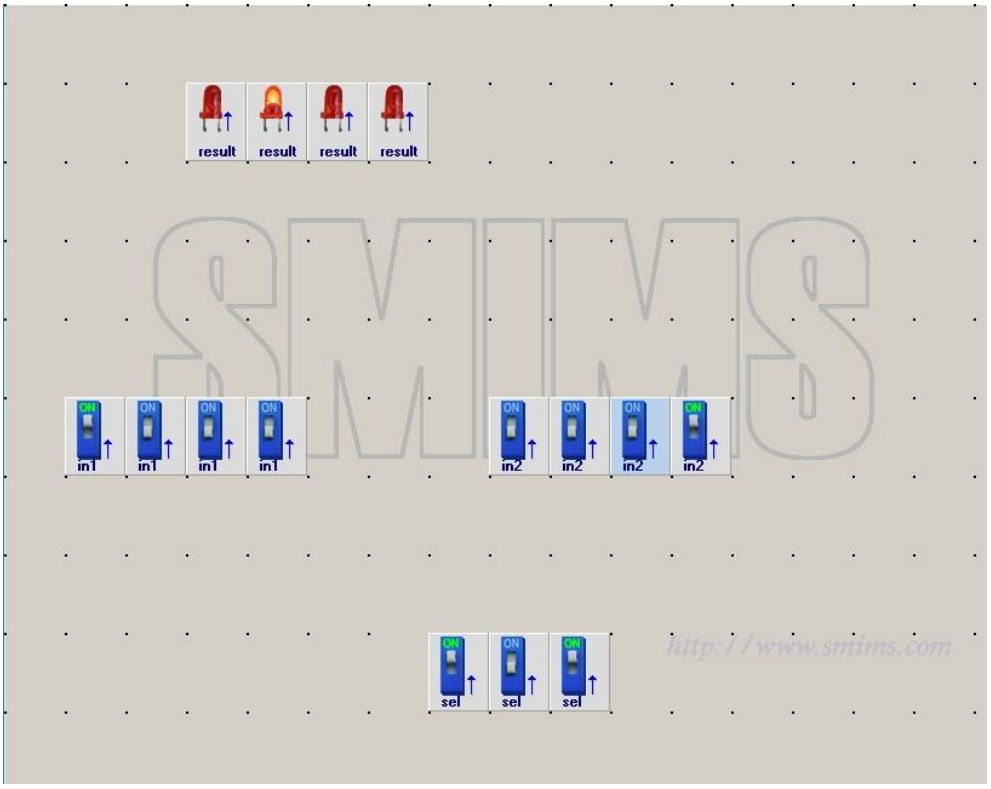
(圖七)利用 VeriInstrucment 進行驗證-3



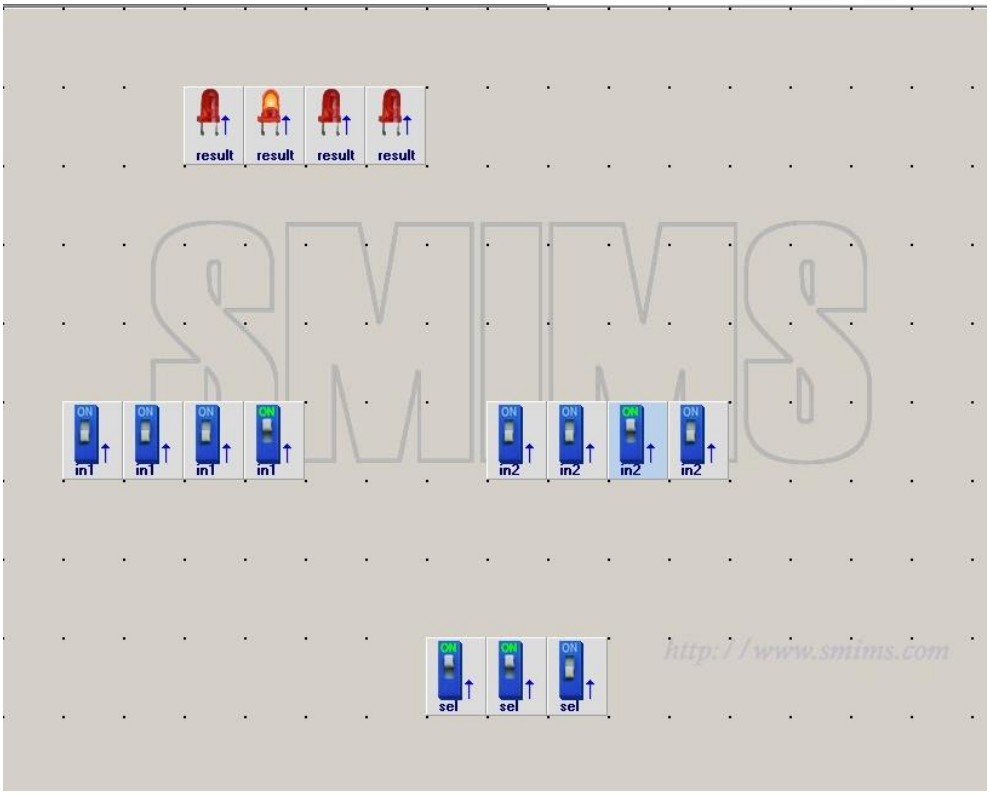
(圖八)利用 VeriInstrucment 進行驗證-4



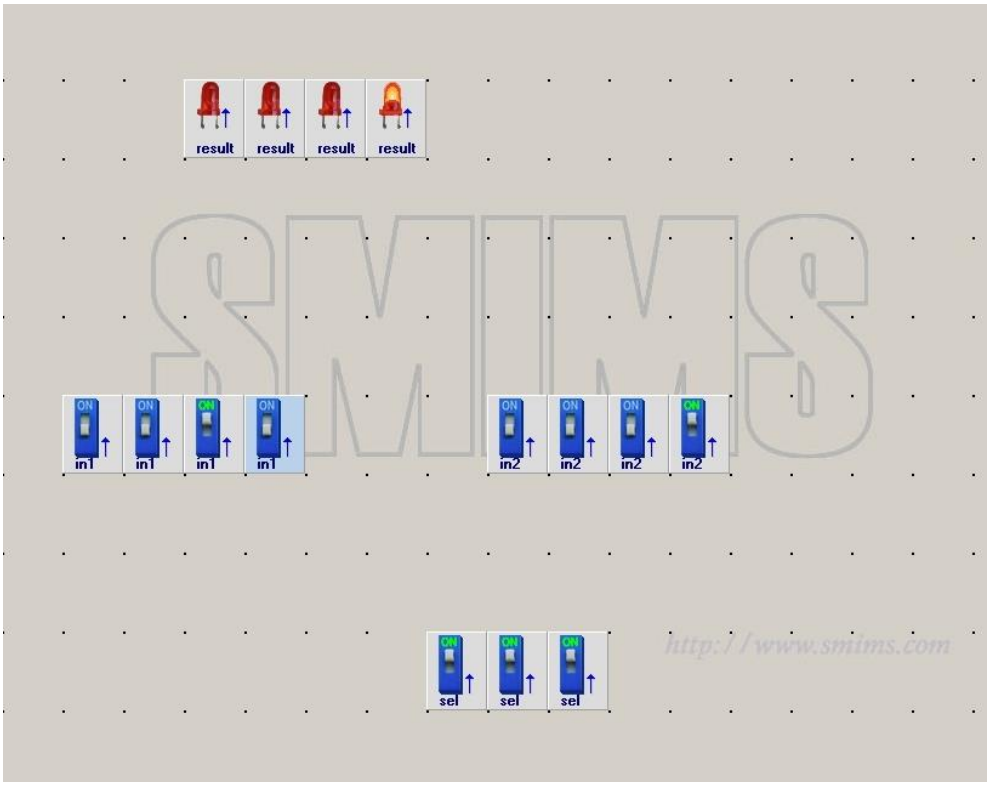
(圖九)利用 VeriInstrucment 進行驗證-5



(圖十)利用 VeriInstrucment 進行驗證-6



(圖十一)利用 VeriInstrucment 進行驗證-7



(圖十二)利用 VeriInstrucment 進行驗證-2

2. 基礎題(二)：4 bit 加法器與七段顯示器：

```
module fourBitBCD(in, out);
    input [3:0]in;

    output[6:0]out;
    reg[6:0]out;

    always@(in)begin
        case(in)
            4'b0000:out=7'b0111111;
            4'b0001:out=7'b0000110;
            4'b0010:out=7'b1011011;
            4'b0011:out=7'b1001111;
            4'b0100:out=7'b1100110;
            4'b0101:out=7'b1101101;
            4'b0110:out=7'b1111101;
            4'b0111:out=7'b0000111;
            4'b1000:out=7'b1111111;
            4'b1001:out=7'b1101111;
            4'b1010:out=7'b1110111;
            4'b1011:out=7'b1111100;
            4'b1100:out=7'b0111001;
            4'b1101:out=7'b1011110;
            4'b1110:out=7'b1110001;
            4'b1111:out=7'b0000000;
        endcase
    end


endmodule
```

(圖十三)4bit 轉七段顯示器模組的 verilog code

```
module fourBitFullAdder(inA, inB, out);
    input [3:0]inA, inB;

    output[3:0]out;
    reg [3:0]out;

    always@(*)begin
        out=inA+inB;
    end

endmodule
```

(圖十四)4bit 全加器模組的 verilog code

```
module system(clk, inA, inB, out);
    input clk;
    input[3:0]inA, inB;

    output[6:0]out;

    wire [3:0]adderOut;

    fourBitFullAdder FA0(.inA(inA), .inB(inB), .out(adderOut));

    fourBitBCD BCD0(.in(adderOut), .out(out));

endmodule
```

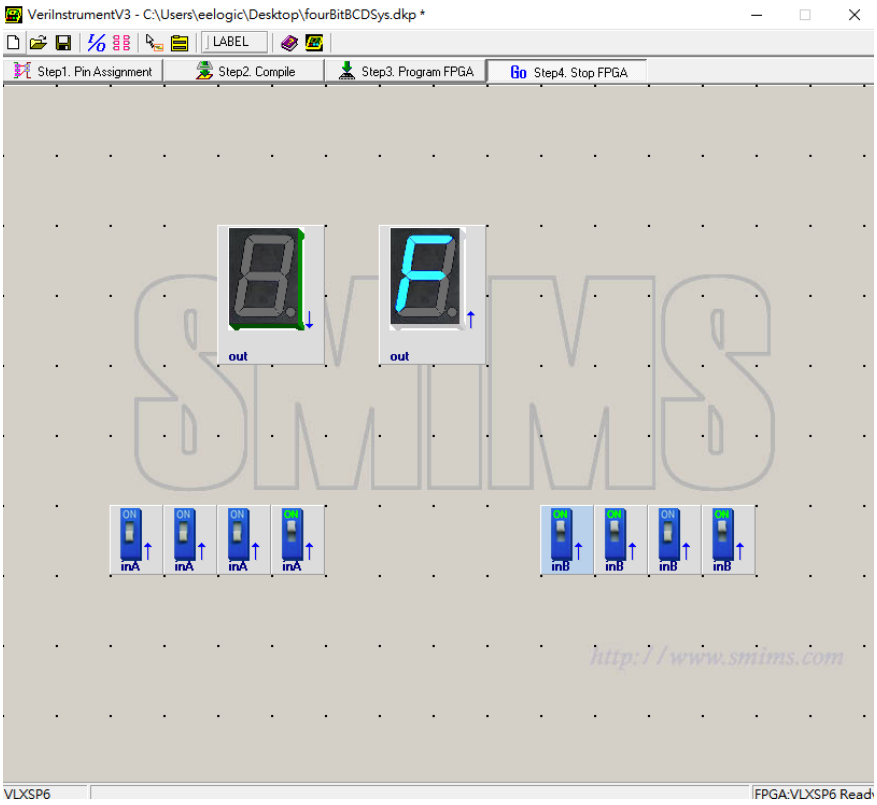
(圖十五)將全部組合而成的模組之 verilog code

```
module test;
    reg[3:0]inA, inB;

    wire[6:0]out;

    wire [3:0]adderOut;

    integer i, j;

    system SYS0(.inA(inA), .inB(inB), .out(out));

    initial begin
        for(i=0; i<8; i=i+1)begin
            for(j=0; j<8; j=j+1)begin
                inA=i; inB=j;#1;
            end
        end
    end

    initial begin
        $monitor("%d, %d, %b", inA, inB, out);
    end

endmodule
```

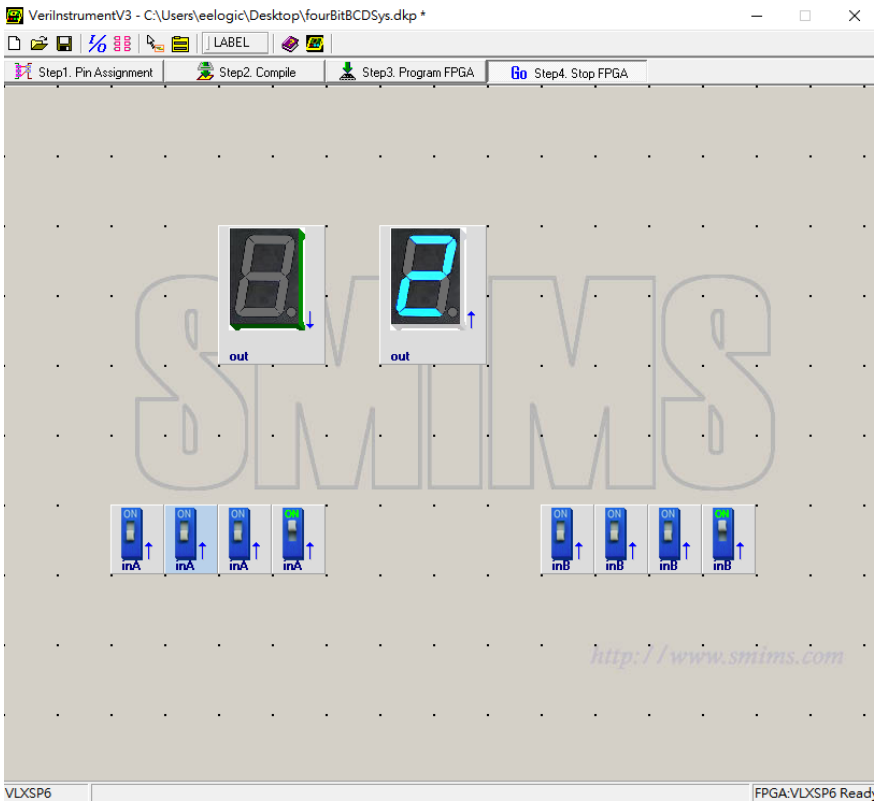
(圖十六) test bench

(圖十七)test bench 驗證所得的波形圖(節錄)

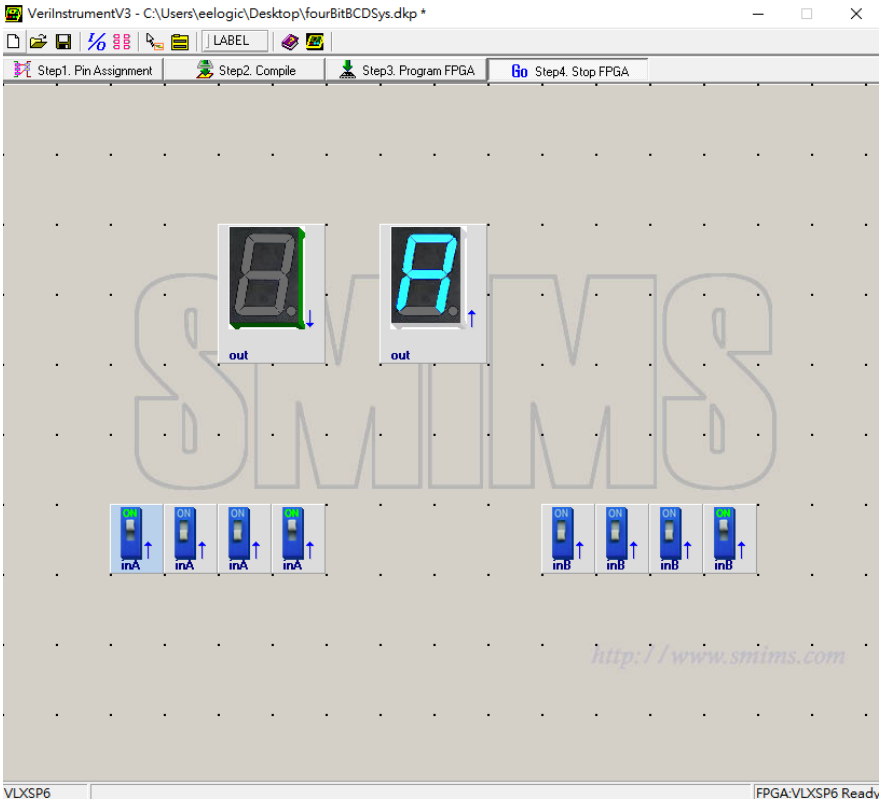
(圖十八)test bench 驗證所得的波形圖(節錄)



(圖十九)利用 VeriInstrument 驗證的結果-1



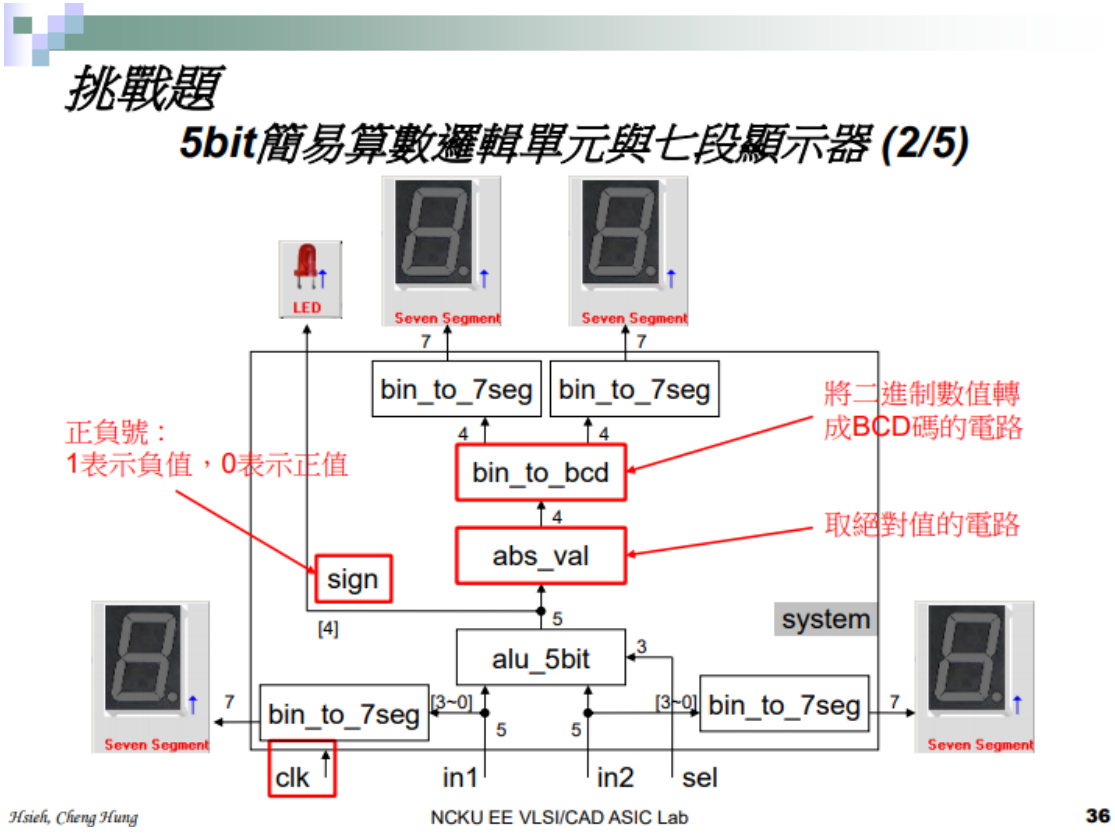
(圖二十)利用 VeriInstrument 驗證的結果-2



(圖二十一)利用 VeriInstrument 驗證的結果-3 找不到目錄項目。

3. 挑戰題(一)：5bit 簡易算數邏輯單元與七段顯示器：

利用如下(圖二十二)之結構組成這個系統，因此會有(圖二十三)的絕對值模組”absVal.v”、(圖二十四)的二進碼二進數轉二進碼十進數模組”binToBCD.v”、(圖二十五)的五位元簡易算術邏輯單元”fiveBitALU.v”、(圖二十六)的四位元二進碼十進數轉七段顯示器模組”fourBitBCD.v”、(圖二十七)的整合系統”system.v”。



(圖二十二)系統結構圖

```
1  module absVal(in, out);
2      input [4:0]in;
3
4      output[3:0]out;
5      reg[3:0]out;
6
7      always@(*)begin
8          if(in[4]==1'b1)out=(~in[3:0])+1;
9          else out=in[3:0];
10     end
11 endmodule
```

(圖二十三)絕對值模組 absVal

```
1  module binToBCD(in, outA, outB);
2      input [3:0]in;
3
4      output[3:0]outA, outB;
5      reg[3:0]outA, outB;
6
7      always@(*)begin
8          if(in>4'b1001)begin
9              outA=1;
10             outB=in-'d10;
11         end
12         else begin
13             outA=0;
14             outB=in;
15         end
16     end
17
18 endmodule
```

(圖二十四)二進碼二進數轉二進碼十進數模組 binToBCD

```
module fiveBitALU(inA, inB, sel, out);
    input [4:0]inA, inB;
    input [2:0]sel;

    output[4:0]out;
    reg   [4:0]out;

    parameter   ADD=3'b000,
                SUB=3'b001,
                AND=3'b010,
                OR =3'b011,
                XOR=3'b100,
                SHR=3'b101,
                SHL=3'b110,
                CMP=3'b111;

    always@(*)begin
        case(sel)
            ADD:out=inA+inB;
            SUB:out=inA-inB;
            AND:out=inA&inB;
            OR :out=inA|inB;
            XOR:out=inA^inB;
            SHR:out=inA>>inB;
            SHL:out=inA<<inB;
            CMP:out=inA>inB;
            default:out=5'b0;
        endcase
    end
endmodule
```

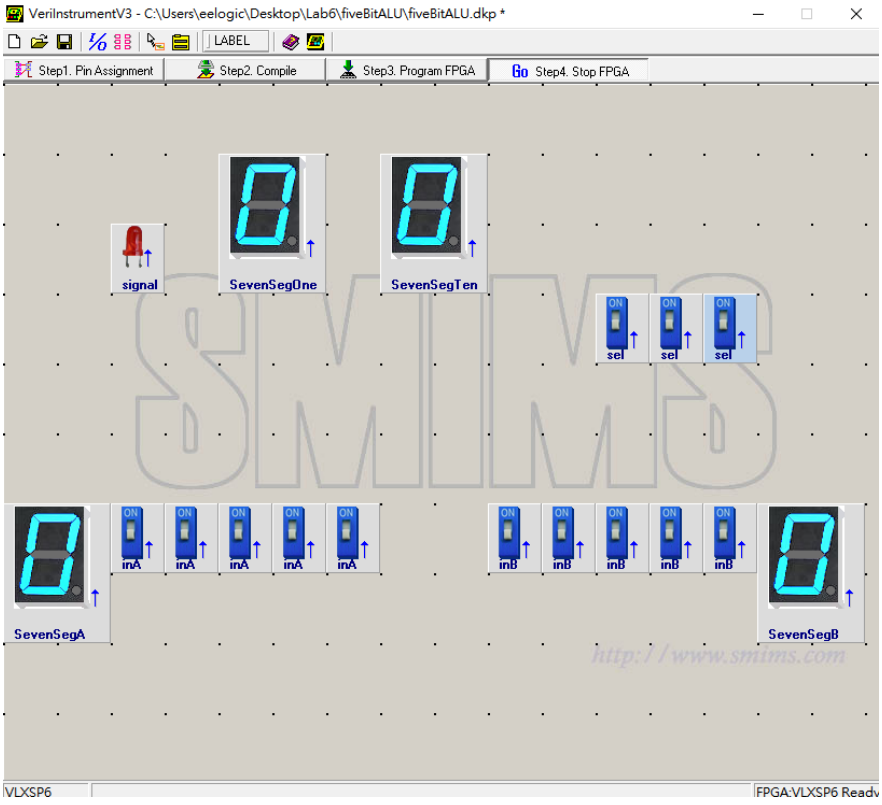
(圖二十五)五位元簡易算術邏輯單元 fiveBitALU.v

```
module fourBitBCD(in, out);
    input [3:0]in;

    output[6:0]out;
    reg[6:0]out;

    always@(*)begin
        case(in)
            4'b0000:out=7'b0111111;
            4'b0001:out=7'b0000110;
            4'b0010:out=7'b1011011;
            4'b0011:out=7'b1001111;
            4'b0100:out=7'b1100110;
            4'b0101:out=7'b1101101;
            4'b0110:out=7'b1111101;
            4'b0111:out=7'b0000111;
            4'b1000:out=7'b1111111;
            4'b1001:out=7'b1101111;
            4'b1010:out=7'b1110111;
            4'b1011:out=7'b1111100;
            4'b1100:out=7'b0111001;
            4'b1101:out=7'b1011110;
            4'b1110:out=7'b1110001;
            4'b1111:out=7'b0000000;
            default:out=7'b0000000;
        endcase
    end
endmodule
```

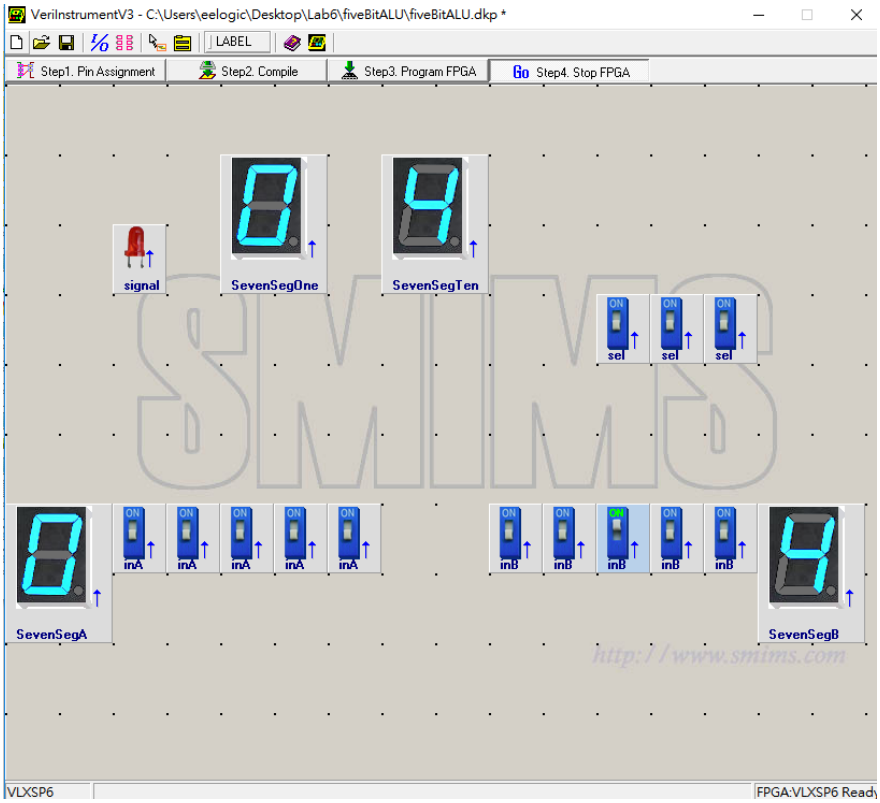
(圖二十六)四位元二進碼十進數轉七段顯示器模組 fourBitBCD.v

```
module system(clk, inA, inB, sel, SevenSegA, SevenSegB, SevenSegOne, SevenSegTen, signal);

    input clk;
    input[4:0] inA, inB;
    input[2:0] sel;

    output[6:0]SevenSegA, SevenSegB, SevenSegOne, SevenSegTen;
    output signal;

    wire [4:0]ALUOut0;
    wire [3:0]absValOut0, BCDOutA, BCDOutB;

    fourBitBCD SS0(.in(inA[3:0]), .out(SevenSegA));
    fourBitBCD SS1(.in(inB[3:0]), .out(SevenSegB));

    fiveBitALU ALU0(.inA(inA), .inB(inB), .sel(sel), .out(ALUOut0));

    absVal ABS0(.in(ALUOut0), .out(absValOut0));

    assign signal=ALUOut0[4];

    binToBCD BTB0(.in(absValOut0), .outA(BCDOutA), .outB(BCDOutB));

    fourBitBCD SS2(.in(BCDOutA), .out(SevenSegOne));
    fourBitBCD SS3(.in(BCDOutB), .out(SevenSegTen));

endmodule
```

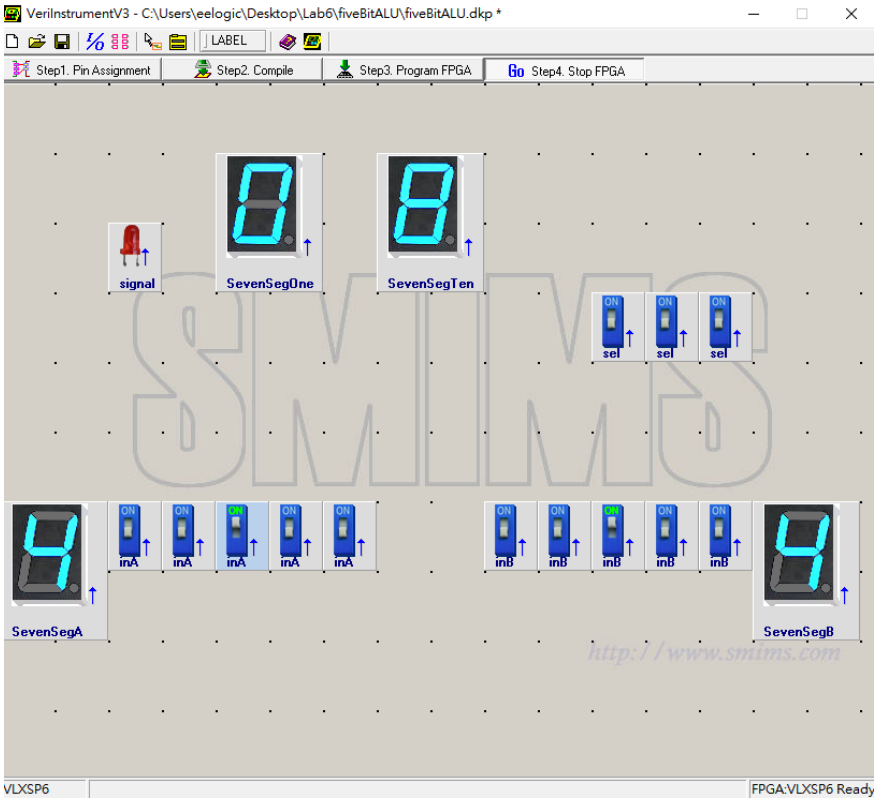
(圖二十七)整合系統 system.v



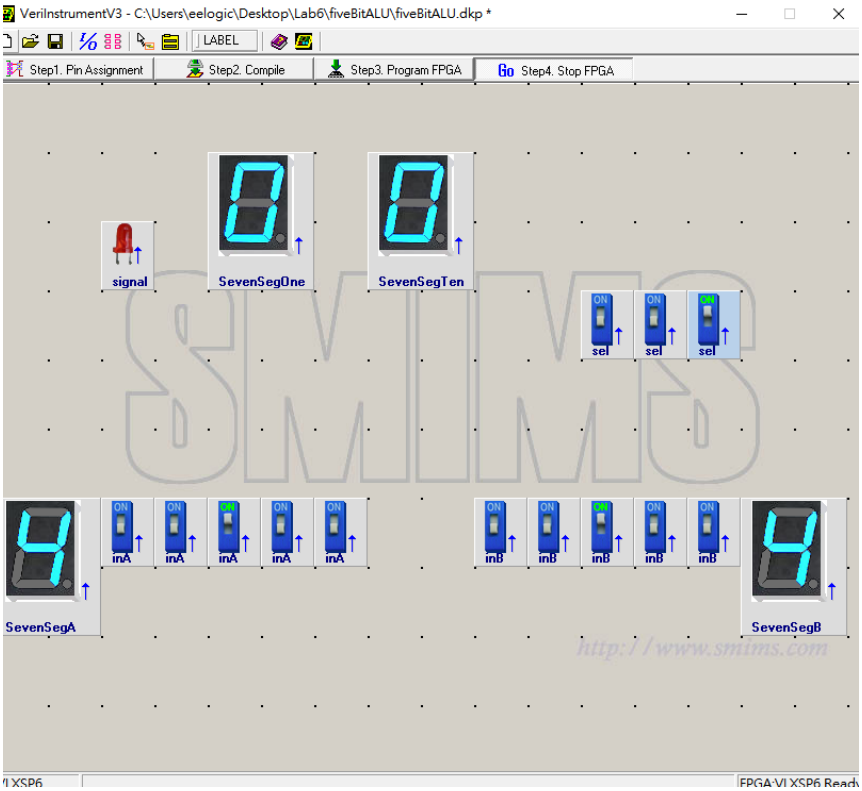
(圖二十八)使用 FPGA 驗證的結果-1



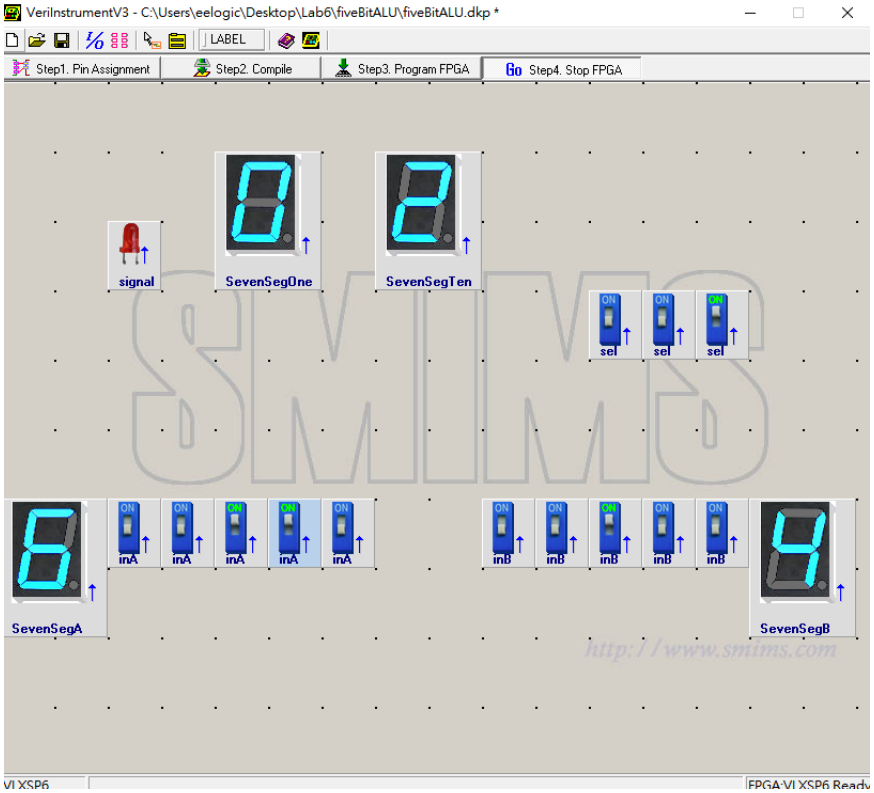
(圖二十九)使用 FPGA 驗證的結果-2



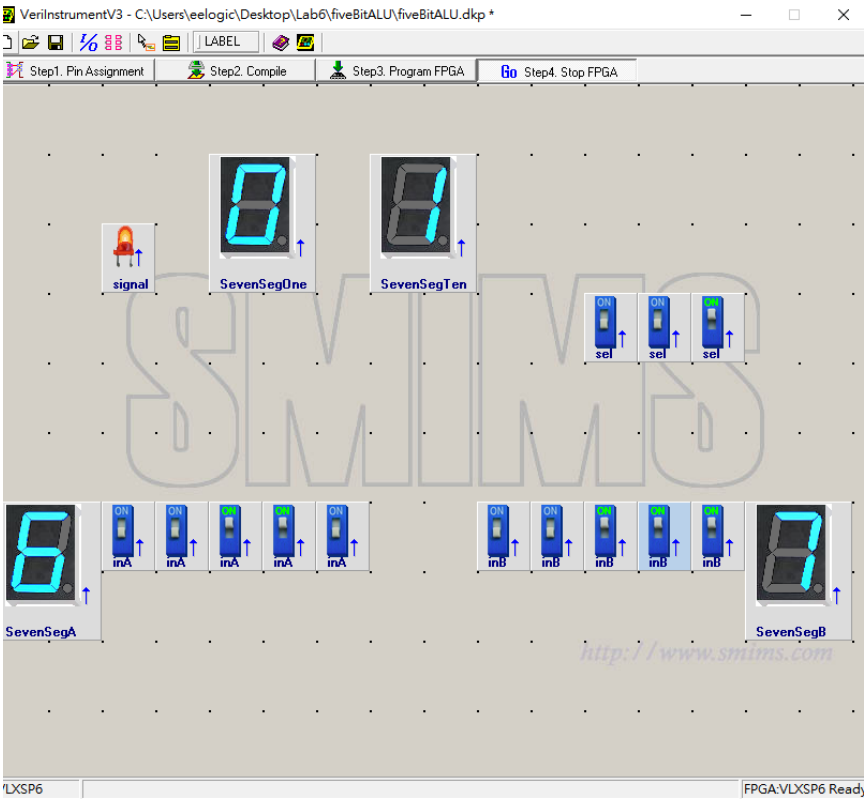
(圖三十)使用 FPGA 驗證的結果-3



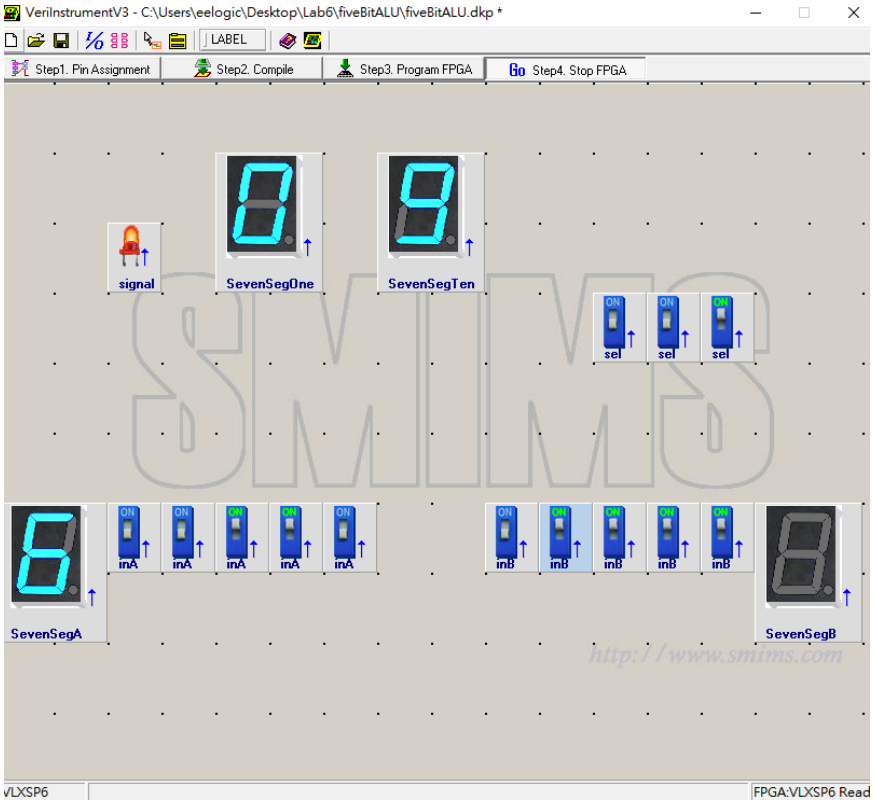
(圖三十一)使用 FPGA 驗證的結果-4



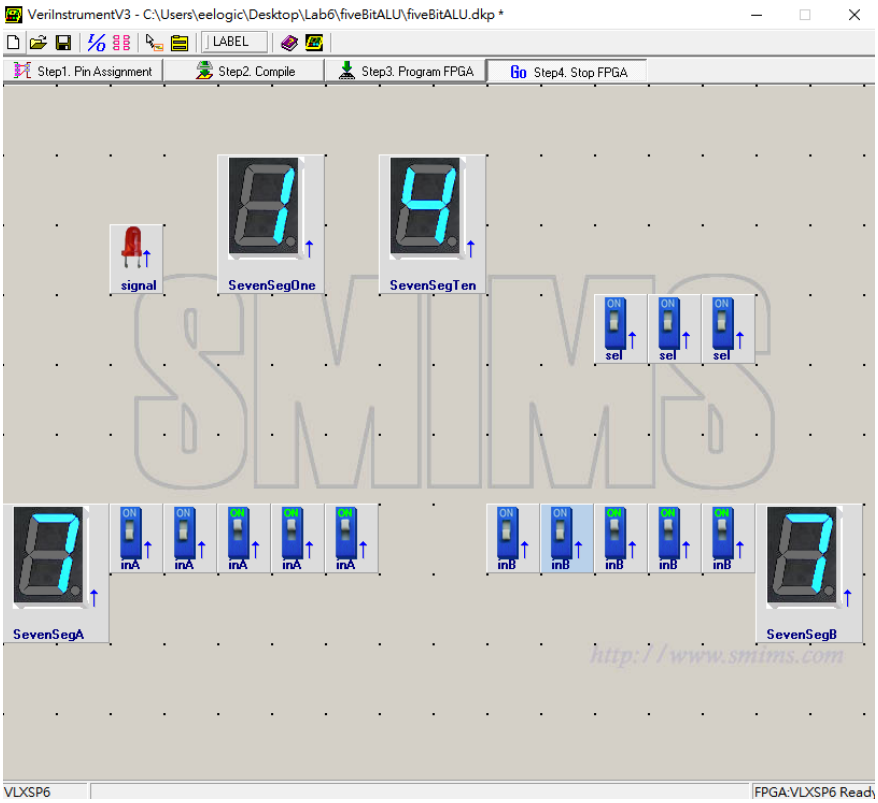
(圖三十二)使用 FPGA 驗證的結果-5



(圖三十三)使用 FPGA 驗證的結果-6



(圖三十四)使用 FPGA 驗證的結果-7



(圖三十五)使用 FPGA 驗證的結果-8

心得：

1. 張軒：有了 veriInstrument 的協助，檢查結果變得簡單多了(但是弄出 veriInstrument 的過程很麻煩...)，比如七段顯示器，如果用 testbench 檢查的話，每一個 case 都會生出一組二位數，根本不知道它是七段顯示器裡的什麼數字呀，所以有了虛擬的七段顯示器就方便多啦!總之用起來就是比 testbench 方便直觀(而且比較炫炮)。

2. 章子嚴：我覺得用 veriInstrument 檢查簡單多了，只不多步驟有點麻煩，如果用 testbench 檢查就比較複雜再用 veriInstrument 后，我還可以在 veriInstrument 的虛擬元件調整輸入輸出，原來虛擬元件要跟著次序排才可以跑，要不然就不能跑

3. 魏晉成：使用 VeriInstrument 的過程中，雖然視覺化，但程式完成後後的步驟繁多，相對麻煩；若是使用 test bench 以及波形圖進行驗證，反而可以使用 for 迴圈做所有可能的遍歷，但如果遇到如 7 段顯示器的非直覺化視覺元件，則比較不方便；因此需要在兩者之間做權衡，各取其長處，才能發揮到最大的功效。