

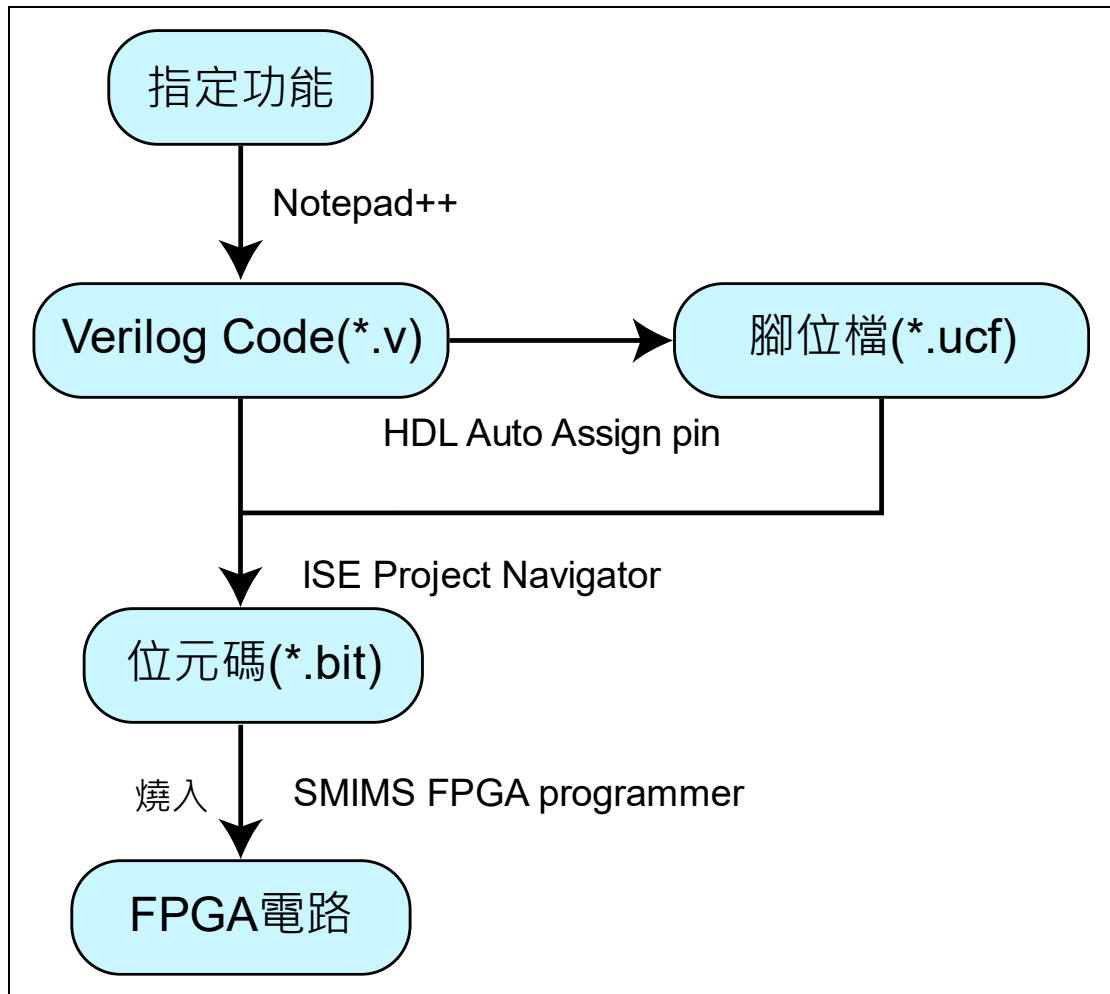
邏輯系統實驗

Lab 6

2021/04/15(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



● 實作題(一): 指撥開關與 LED 之使用





1. Verilog Code

```
module FPGA_practice(led0,led1,led2,led3,in0,in1);  
    input in0,in1;  
    output led0,led1,led2,led3;  
  
    assign led0 = {in0,in1} == 2'b11 ? 1'b0 : 1'b1;  
    assign led1 = {in1,in0} == 2'b10 ? 1'b0 : 1'b1;  
    assign led2 = {in1,in0} == 2'b01 ? 1'b0 : 1'b1;  
    assign led3 = {in1,in0} == 2'b00 ? 1'b0 : 1'b1;  
endmodule
```

2. UFC file to specify pin

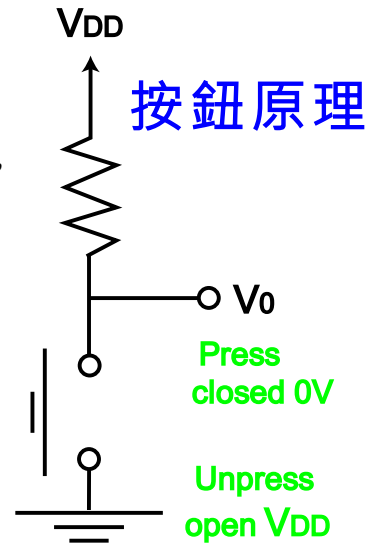
```
#PACE: Start of Constraints generated by PACE
#PACE: Start of PACE I/O Pin Assignments
NET "in0" LOC = "E1" | IOSTANDARD = LVCMOS33 ;
NET "in1" LOC = "E2" | IOSTANDARD = LVCMOS33 ;
NET "led0" LOC = "A3" | IOSTANDARD = LVCMOS33 ;
NET "led1" LOC = "B3" | IOSTANDARD = LVCMOS33 ;
NET "led2" LOC = "A2" | IOSTANDARD = LVCMOS33 ;
NET "led3" LOC = "B2" | IOSTANDARD = LVCMOS33 ;
#PACE: Start of PACE Area Constraints
#PACE: Start of PACE Prohibit Constraints
#PACE: End of Constraints generated by PACE
```

3. FPGA result

$(in0, in1) = (0,0)$	$(in0, in1) = (0,1)$
	
$(in0, in1) = (1,0)$	$(in0, in1) = (1,1)$
	

功能介紹&結果解釋

按鈕開關原理如右圖所示，按下為 0、放開為 1，
因此以 $(in0, in1) = (0,0)$ 為例符合判斷式
 $\{in1, in0\} == 2'b00$ 因此 $led3 = 1'b0$ ，
但別於一般習慣，在這裡我認為燈泡是 low-active，
也就是說 0 的時候燈泡亮、1 的時候燈泡暗，如此
假設才符合實驗結果。



● 實作題(二): General I/O

1. Verilog Code

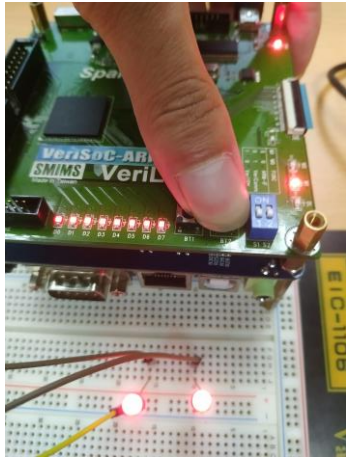
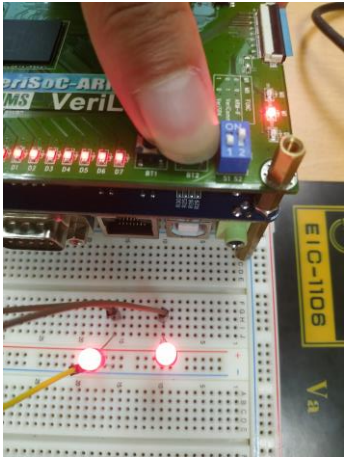
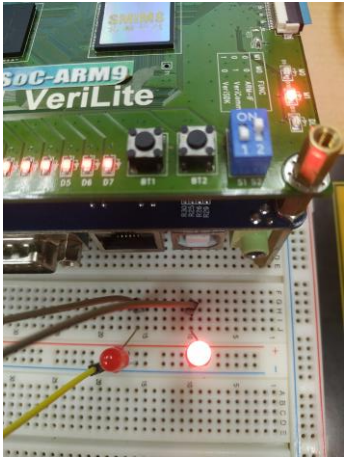
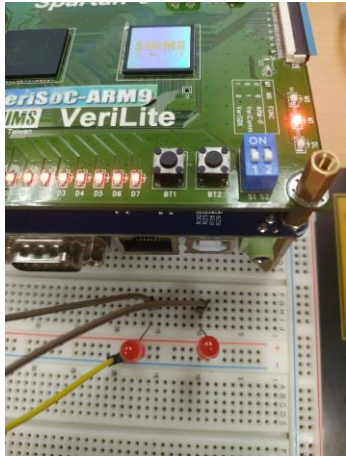
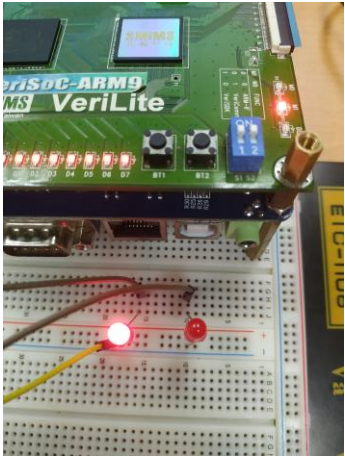
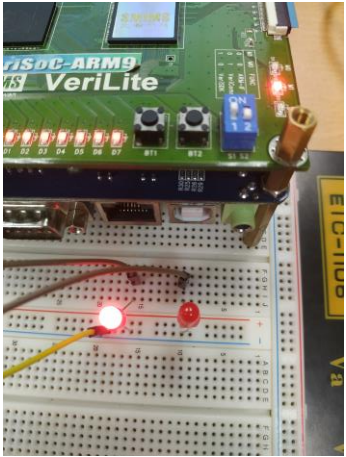
```
module FPGA_practice2(led0,led1,in0,in1,in2);
    input in0,in1,in2;
    output led0,led1;

    assign led0 = ~in2 | (in2 & in1 & ~in0);
    assign led1 = ~in2 | (in2 & ~in1);
endmodule
```

2. UFC file to specify pin

```
#PACE: Start of Constraints generated by PACE
#PACE: Start of PACE I/O Pin Assignments
NET "in0" LOC = "E2" | IOSTANDARD = LVCMOS33 ;
NET "in1" LOC = "E1" | IOSTANDARD = LVCMOS33 ;
NET "in2" LOC = "D3" | IOSTANDARD = LVCMOS33 ;
NET "led0" LOC = "D5" | IOSTANDARD = LVCMOS33 ;
NET "led1" LOC = "B5" | IOSTANDARD = LVCMOS33 ;
#PACE: Start of PACE Area Constraints
#PACE: Start of PACE Prohibit Constraints
#PACE: End of Constraints generated by PACE
```

3. FPGA result

$(in2, in1, in1) = (0, 1, 1)$ 	$(in2, in1, in0) = (0, 1, 0)$ 	$(in2, in1, in0) = (1, 1, 0)$ 
$(in2, in1, in0) = (1, 1, 1)$ 	$(in2, in1, in0) = (1, 0, 0)$ 	$(in2, in1, in0) = (1, 0, 1)$ 
<p>功能介紹&結果解釋</p> <p>由 Verilog Code 去觀察，我認為此電路有點像 Priority 選擇邏輯電路，而優先順序為 $in1 > in0$ 而 $in2$ 比較像 enable 的控制訊號，當 $in2 = 0$，電路 disable，無論 $in1$、$in0$ 為多少，兩個輸出皆 1，因此燈泡亮；反之當 $in2 = 1$，電路 enable，而在 $in1 = 0$，$led0 = 0$ (右邊燈泡暗)、$led1 = 1$ (左邊燈泡亮)；最後只有在 $(in2, in1) = (1, 1)$ 時，$in0$ 改變，讓 output 改變。</p>		

● 實作題(三): 七段顯示器

1. Verilog Code

```
module SecenSeg(dout, din, clk);
    input [3:0]din;
    input clk;
    output [6:0]dout;
    reg [6:0]dout;

    always@(din)
    begin
        case (din)
            4'b1111:dout = 7'b1000_000;
            4'b1110:dout = 7'b1111_001;
            4'b1101:dout = 7'b0100_100;
            4'b1100:dout = 7'b0110_000;

            4'b1011:dout = 7'b0011_001;
            4'b1010:dout = 7'b0010_010;
            4'b1001:dout = 7'b0000_010;
            4'b1000:dout = 7'b1111_000;

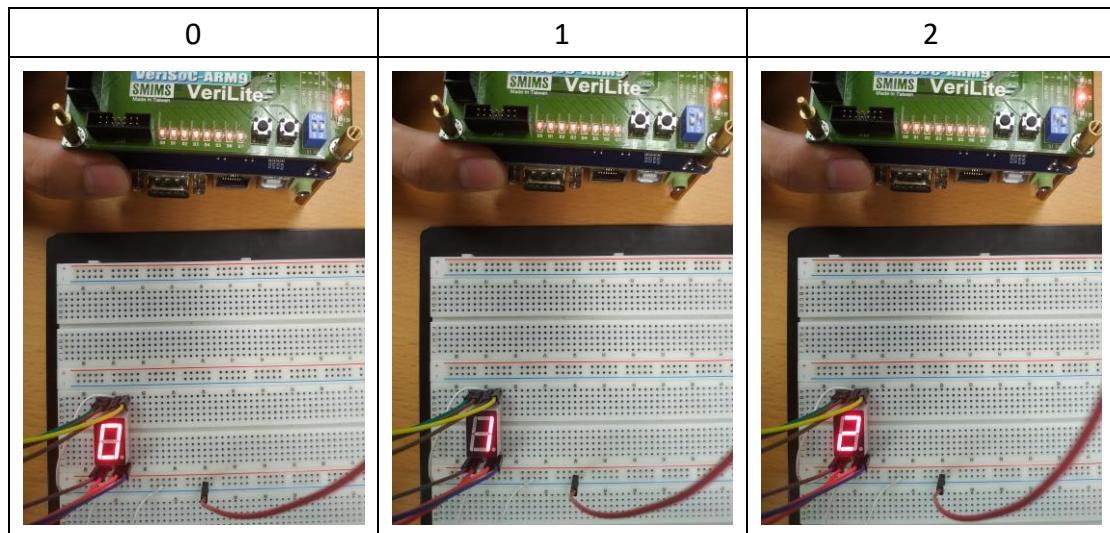
            4'b0111:dout = 7'b0000_000;
            4'b0110:dout = 7'b0010_000;
            4'b0101:dout = 7'b0001_000;
            4'b0100:dout = 7'b0000_011;

            4'b0011:dout = 7'b1000_110;
            4'b0010:dout = 7'b0100_001;
            4'b0001:dout = 7'b0000_110;
            4'b0000:dout = 7'b0001_110;
            default:dout = 7'b1111_111;
        endcase
    end
endmodule
```

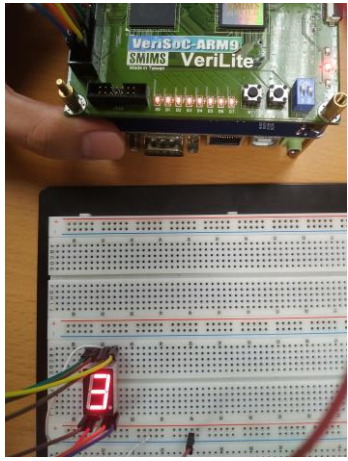
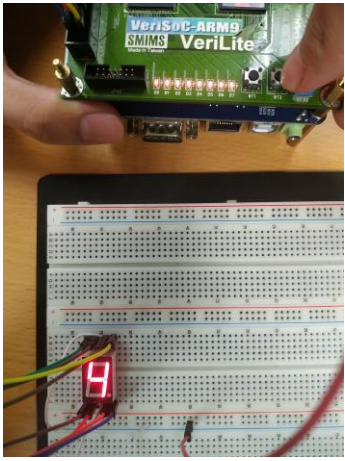
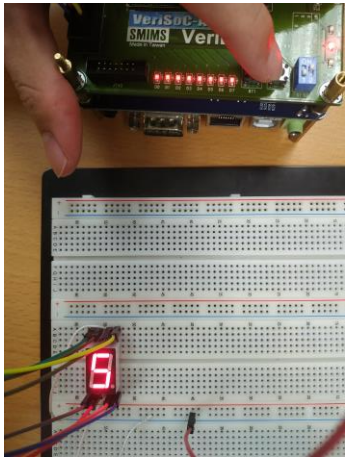
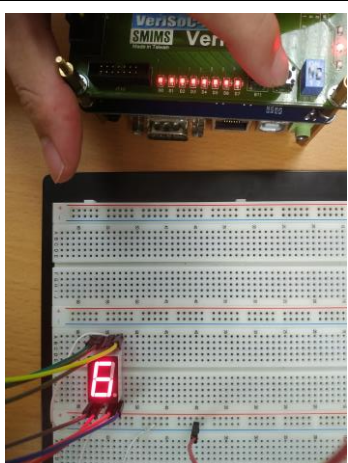
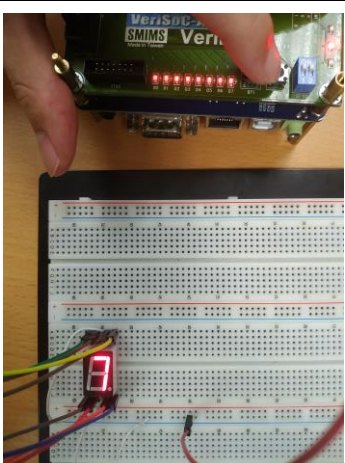
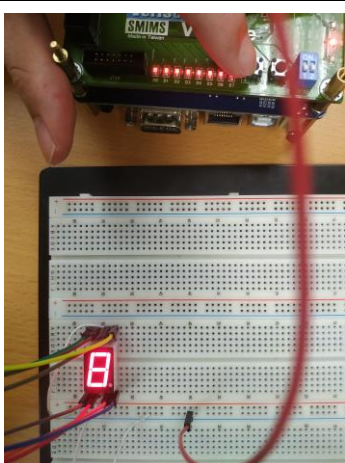
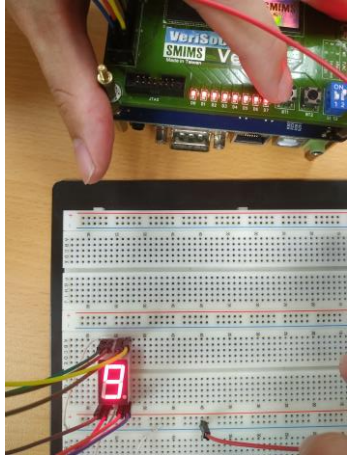
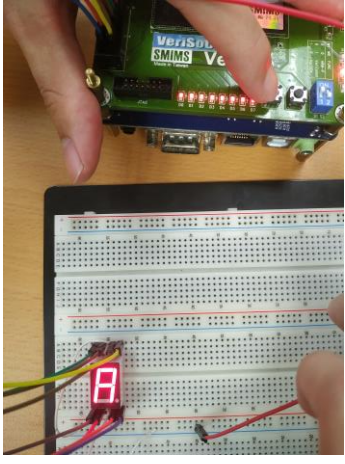
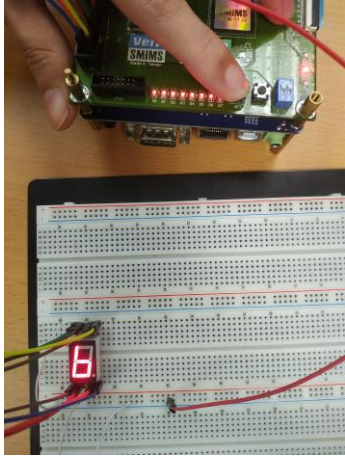

2. UFC file to specify pin

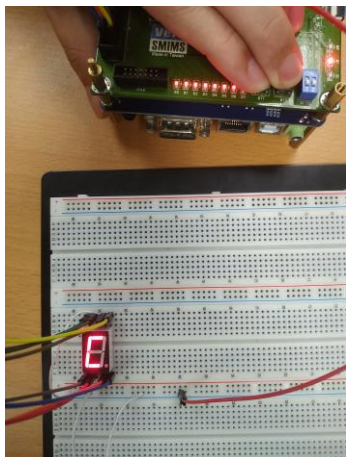
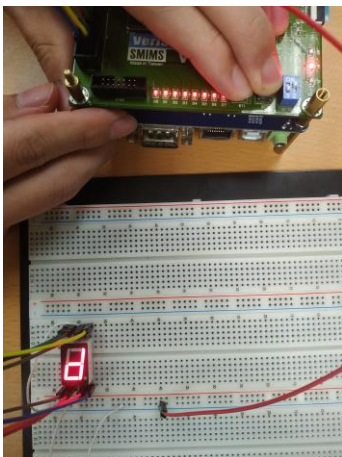
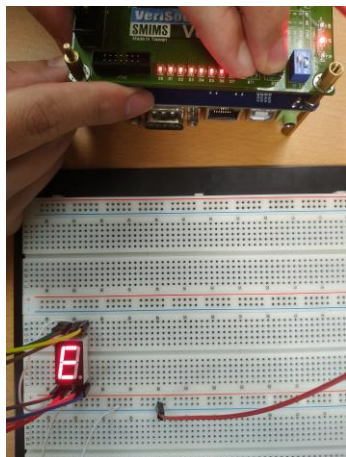
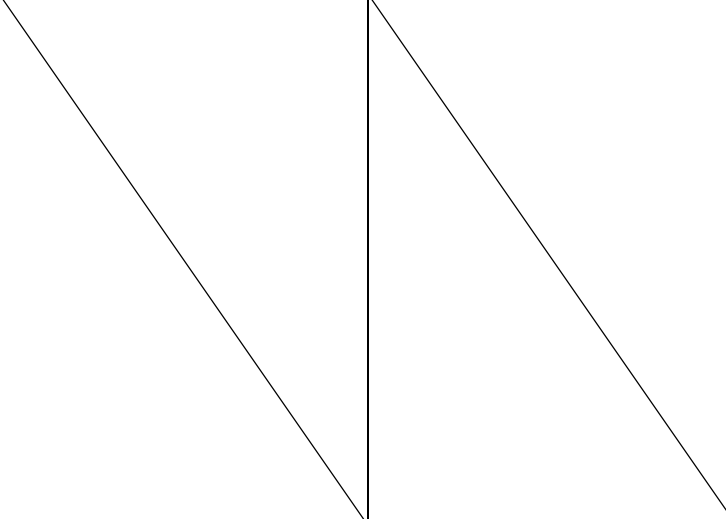
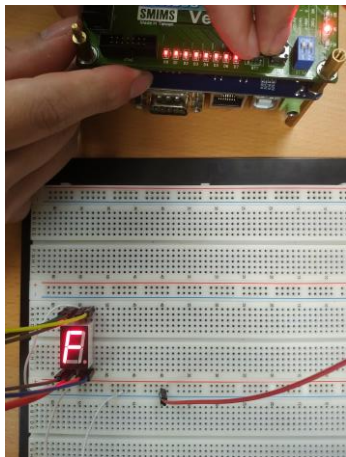
```
#PACE: Start of Constraints generated by PACE
#PACE: Start of PACE I/O Pin Assignments
NET "din[0]" LOC = "E2" | IOSTANDARD = LVCMOS33 ;
NET "din[1]" LOC = "E1" | IOSTANDARD = LVCMOS33 ;
NET "din[2]" LOC = "D3" | IOSTANDARD = LVCMOS33 ;
NET "din[3]" LOC = "D1" | IOSTANDARD = LVCMOS33 ;
NET "clk" LOC = "F2" | IOSTANDARD = LVCMOS33 ;
NET "dout[0]" LOC = "C5" | IOSTANDARD = LVCMOS33 ;
NET "dout[1]" LOC = "D5" | IOSTANDARD = LVCMOS33 ;
NET "dout[2]" LOC = "A5" | IOSTANDARD = LVCMOS33 ;
NET "dout[3]" LOC = "C6" | IOSTANDARD = LVCMOS33 ;
NET "dout[4]" LOC = "E8" | IOSTANDARD = LVCMOS33 ;
NET "dout[5]" LOC = "E6" | IOSTANDARD = LVCMOS33 ;
NET "dout[6]" LOC = "E7" | IOSTANDARD = LVCMOS33 ;
#PACE: Start of PACE Area Constraints
#PACE: Start of PACE Prohibit Constraints
#PACE: End of Constraints generated by PACE
NET "clk" TNM_NET = "clk";
```

3. FPGA result



邏輯系統實驗 Lab6 第 1 組

<p>3</p> 	<p>4</p> 	<p>5</p> 
<p>6</p> 	<p>7</p> 	<p>8</p> 
<p>9</p> 	<p>10(A)</p> 	<p>11(B)</p> 

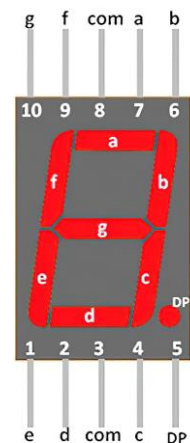
12(C)	13(D)	14(E)
		
15(F)		
		

功能介紹&結果解釋

由 Verilog Code 觀察，雖然有 `input clk` 但 `always@()` 由 `din` 驅動而非 `clk`，因此我認為還是 combinational 電路，輸入有四個 bit 也就是四個按鈕，因此可以輸入 0 – 15。由右圖七段顯示器可知除了兩端 COM 接到工作電壓 V_{CC} (J3.11)，而共需要 7 個輸出訊號，也就是 Verilog Code 的 `output reg[6:0]dout`。

General I/O J3			
GND	J3.12	VCC5V	J3.11
E8	J3.10	E7	J3.9
C6	J3.8	E6	J3.7
D6	J3.6	F7	J3.5
A5	J3.4	C5	J3.3
B5	J3.2	D5	J3.1

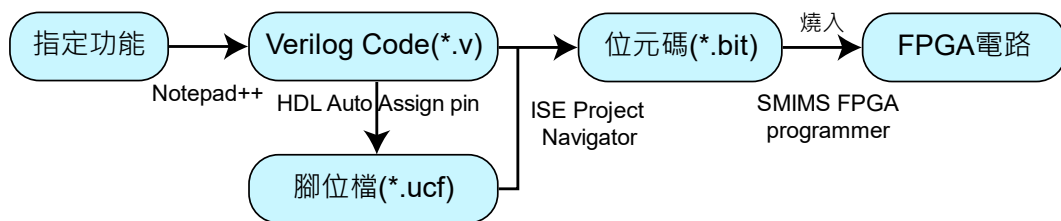
(功能 腳位代號) (功能 腳位代號)



● 心得

1. 組員一 陳旭祺

FPGA(Field-Programmable Gate Array)是現場可編程門陣列的英文縮寫。它是一種靈活的可以根據設計改變內部電路結構的晶片。FPGA 不僅僅應用於數字電路的實現，它廣泛地應用於 ASIC 原型驗證，嵌入式系統，數位訊號處理等等方面。目前 FPGA 的生產商最大的兩家是 AMD 的 Xilinx 和 Intel 的 Altera，我們使用的就是前者。因此我們不需要跑繁瑣的下晶片流程，加上 FPGA 可重複設計，容錯率、彈性較高，Verilog Code 打錯、燒錄錯，我們還可以重複以下實驗流程圖



2. 組員二 張振杰

這次實驗的一開始我對於實驗中要把腳位檔燒錄 FPGA 的實際過程非常感興趣，而且這次實驗所用到的板子總共有 3 層，其餘兩層這次實驗沒用到。FPGA 板子由於能夠重複燒錄且不會有任何潛在問題，容錯率高，所以燒錄後如與期望結果不符就可以推斷程式碼內部出現了差錯。這次實驗也讓我看到了軟硬體的操作，獲益良多。

3. 組員三 何啟造

這次的實驗是用現場可程式邏輯閘陣列(FPGA)來進行。我們可以用 Verilog 所寫好的電路設計燒錄到 FPGA 上進行設計，所以使用上非常的方便。而這次實驗我們用 Verilog 寫好解碼器和編碼器後，燒錄到 FPGA 並進行測試，也嘗試將 FPGA 輸出的結果和實體電路中的七段顯示器接起來以顯示結果。總體來說算是非常有趣的實驗。