



2025/09/18

實驗二

組合邏輯練習

姓名：林承羿

學號：01257027

班級：資工 3A

E-mail：IanLin6225@gmail.com

※ 注意

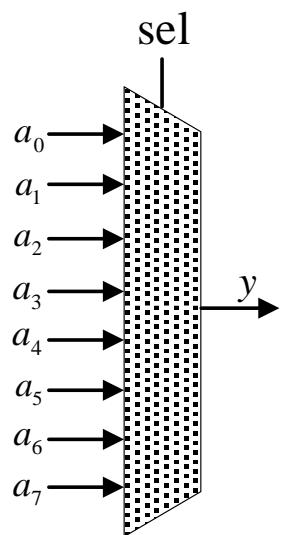
1. 繳交時一律轉 PDF 檔
2. 繳交期限為下周上課前
3. 一人繳交一份
4. 檔名請按照作業檔名格式進行填寫，未依照格式不予批改
5. 檔名範例：學號_姓名_HW2

1、多工器

● 實驗說明：

1. 實作多工器。
2. 輸入資料 $a_n [2:0]$ ： $a_0=0$ 、 $a_1=1$ 、 $a_2=2$ 、 $a_3=3$ 、 $a_4=4$ 、 $a_5=5$ 、 $a_6=6$ 、 $a_7=7$
3. 輸出： $y [2:0]$
4. Testbench 內容為 sel 由 0 到 7

●系統硬體架構方塊圖（接線圖）：



●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**

```
mux.sv > mux
1 module mux (
2     input logic [2:0] a,
3     input logic [2:0] sel,
4     output logic [2:0] y
5 );
6     always_comb begin
7         case (sel)
8             3'd0: y = a;
9             3'd1: y = a;
10            3'd2: y = a;
11            3'd3: y = a;
12            3'd4: y = a;
13            3'd5: y = a;
14            3'd6: y = a;
15            3'd7: y = a;
16        endcase
17    end
18 endmodule
```

由上圖可看到不管 sel 為何，y 總是為 a，主要是 tb 內容有關。在 tb 中，a 為一維陣列，但多工器中有 8 個輸入(直覺為 a₀~a₇ 共 8 個陣列)。由於輸入侷限在一個一維陣列，思考測資時 sel=0，a=0，又題意描述輸入的是 a_n，故猜想此時 n=0，如果邏輯成立，在 tb 中同一列的 a 即為 sel 選到的 a 要輸出給 y 之值。因此每項 sel 所要做的都是 y=a。

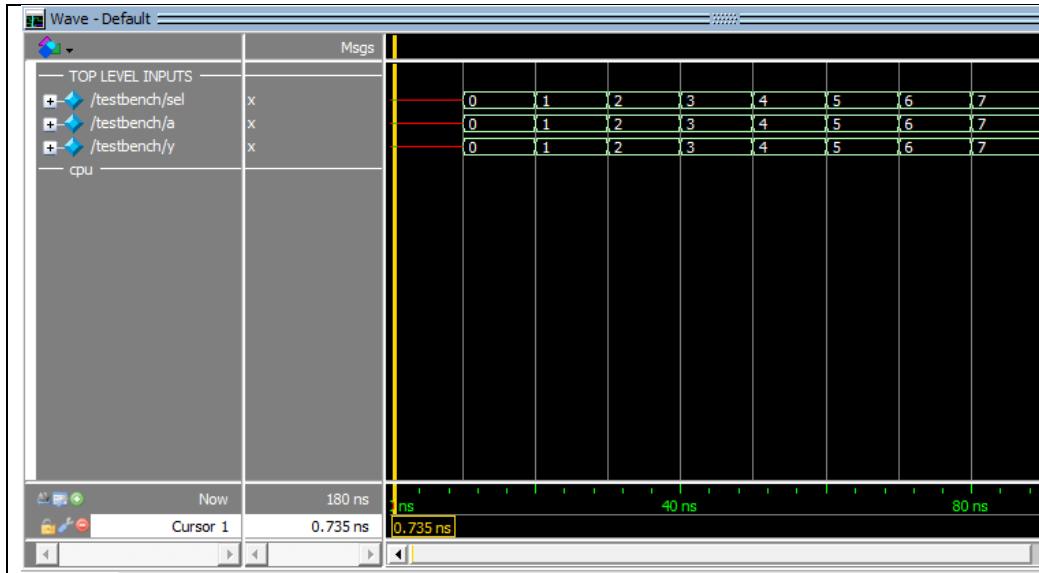
```

mux.sv > mux
1 module mux (
2     input logic [2:0] a,
3     input logic [2:0] sel,
4     output logic [2:0] y
5 );
6     assign y = a;
7 endmodule

```

上圖為本人對結果作的猜想，由於每個 case 都做相同的事情，在此題下，sel 就變得不是很重要，波型圖也如猜想一樣跑出相同的結果。

● 模擬結果與結果說明：



從上圖可以看到由 sel 選擇線決定接收 a 值並由 y 輸出。首先第一筆為 sel=0，y 應當為 a0，而此時的 tb 令 a=0 (依照說明，輸入測資為 an，a0=0、a1=1...)。可知結果正確。

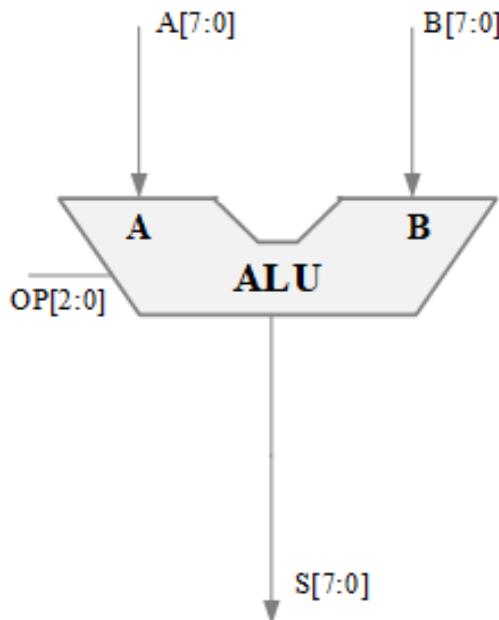
2、 算術邏輯單元(ALU)

● 實驗說明：

1. 實作 ALU，跑模擬波形。
2. $S = 5 + 0xb$
3. $S = 0xd - 7$
4. A[7:0]、B[7:0]、op 分別輸入資料(16 進位)：5, 0xb, 0。
5. A[7:0]、B[7:0]、op 分別輸入資料(16 進位)：0xd, 7, 1。
6. 輸入：A[7:0], B[7:0], op
7. 輸出：S[7:0]

OP	ALU 運算	註解
0	$S = A + B$	加法
1	$S = A - B$	減法

● 系統硬體架構方塊圖（接線圖）：



●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**

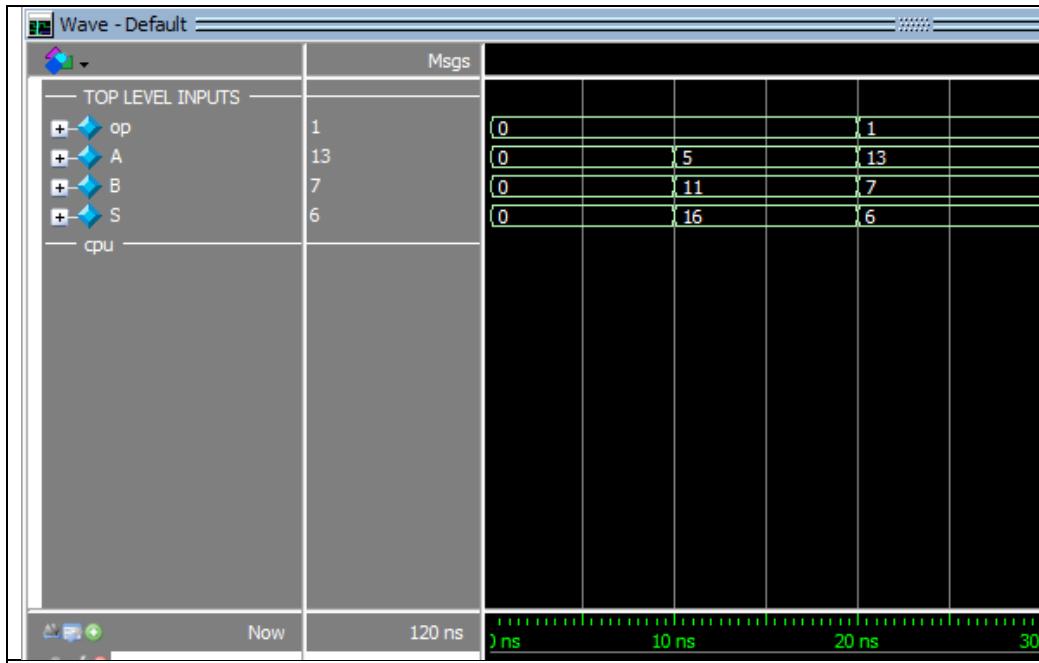
```
❶ alu.sv > ...
1  module alu (
2      input logic [7:0] A, B,
3      input logic [2:0] op,
4      output logic [7:0] S
5  );
6      always_comb begin
7          if (op == 3'd0) begin
8              S = A + B;
9          end
10         else begin
11             S = A - B;
12         end
13     end
14 endmodule
```

依照圖片命名如 header 所示，此題為 op 當 select 線之多工器(組合邏輯)，以上是 if else 寫法。

```
❶ alu.sv > alu
1  module alu (
2      input logic [7:0] A, B,
3      input logic [2:0] op,
4      output logic [7:0] S
5  );
6      always_comb begin
7          unique case (op)
8              3'd0: S = A + B;
9              3'd1: S = A - B;
10             endcase
11         end
12     endmodule
```

依照圖片命名如 header 所示，此題為 op 當 select 線之多工器(組合邏輯)，以上是 unique case 寫法，也可用 case 寫，但需要 default。

●模擬結果與結果說明：



從上圖可看到 op 為 0 時，執行加法 ($5 + 11 = 16$)，op 為 1 時，執行減法 ($13 - 7 = 6$)

●結論與心得：

透過此次實驗，我快速回憶起了如何撰寫 system verilog，並透過 testbench 測試程式是否正確運作，也更了解各種.do 檔案中每段指令所代表的意義，具體如修改查看的變數以顯示 modelSim 中的波型、修改 compile 的檔案。

另外，再換一個角度撰寫 system verilog 後，更能體會老師所講述的話語，心中要先有電路大概的形式才轉換成描述語言，而非一味的撰寫而不知道電路為何。