

## Lab 2 Report

107091218 謝霖泳

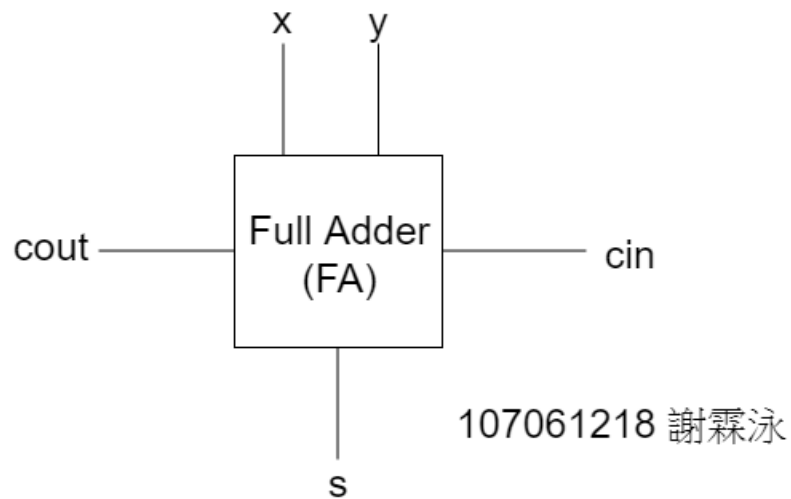
1.

### ➤ Design specification

For a full adder (FA):

Input: x, y, cin

Output: s, cout



### ➤ Design Implementation

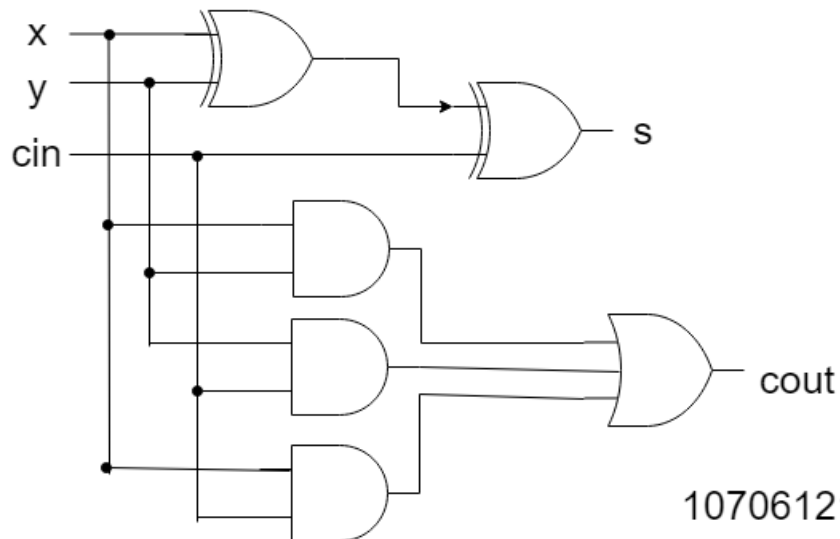
First, we can derive the truth table

x	y	cin	S	cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Then, by using K-map, we can derive the logic equations.

$$S = (x \oplus y) \oplus \text{cin} = x \oplus y \oplus \text{cin}$$

$$\text{cout} = x \& y + y \& \text{cin} + x \& \text{cin}$$



107061218 謝霖泳

i/o pin assignment 如同題目所要求

x	y	cin	S	cout
V17	V16	W16	U16	E19

### ➤ Discussion

這題題目因為是第一次 lab 題目的延伸，只是多了 i/o pin assignment 的部分，將自己的 design 實做在 FPGA 板上顯示出來，因此，並沒有遇到什麼困難。

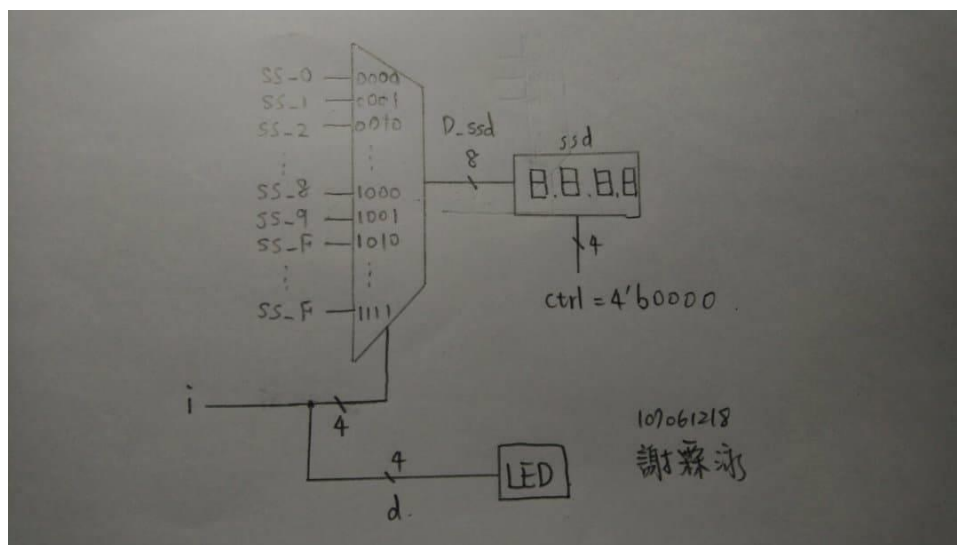
2.

### ➤ Design Specification

For a BCD to 7-segment display:

Input: [3:0] i

Output: [7:0] D\_ssd, [3:0] d, [3:0] ctrl



➤ Design Implementation

首先，由 FPGA 板下方最右邊的 4 個開關作為 4-bit  $i$  的輸入。SS\_3 的意思就是對應「3」這個數字顯示在七段顯示器上的結果，因為還有小數點，因此為 8-bit，依此類推。因為七段顯示器為 low active，因此，有亮的段值設為 0，暗的段則設為 1。以輸入的  $i$  作為 MUX 的 selection，去選擇對應的結果，並將之傳到七段顯示器。

此外，也另一個 4bit 的變數  $d$  作為 LED 燈的控制，哪個開關被打開，他上方的 LED 燈就要亮起來。因為 LED 與開關均為 high active，因此，其值與所輸入的  $i$  值理應相同，即  $d = i$ 。

另外，我還設計了一個 4bit 的 input 名為 ctrl，是用來控制 ssd 要亮哪個 digit，因為無特別要求，所以當然 4 個 digit 都亮，因此，ssd 的值我就直接給定為 4'b0000。

IO pin assignment

$i[3]$	$i[2]$	$i[1]$	$i[0]$	$d[3]$	$d[2]$	$d[1]$	$d[0]$
W17	W16	V16	V17	V19	U19	E19	U16

D_ssd[7]	D_ssd[6]	D_ssd[5]	D_ssd[4]	D_ssd[3]	D_ssd[2]	D_ssd[1]	D_ssd[0]
W7	W6	U8	V8	U5	V5	U7	V7

ctrl[3]	ctrl[2]	ctrl[1]	ctrl[0]
W4	V4	U4	U2

➤ Discussion

這題當時在設計時遇到一個自己沒有想透徹而導致的問題，就是 ssd decoder 的部分。因為念 7-seg dec. 的時候，我習慣從最上面那段開始念，以 2 為例，在 ssd 的顯示為亮亮暗亮亮暗亮暗，所以 decoder 值為 00100101，但在設定多 bit 時，由左而右為 MSB 到 LSB，所以當時我的七段 io pin 順序設反，導致無法正常顯示。

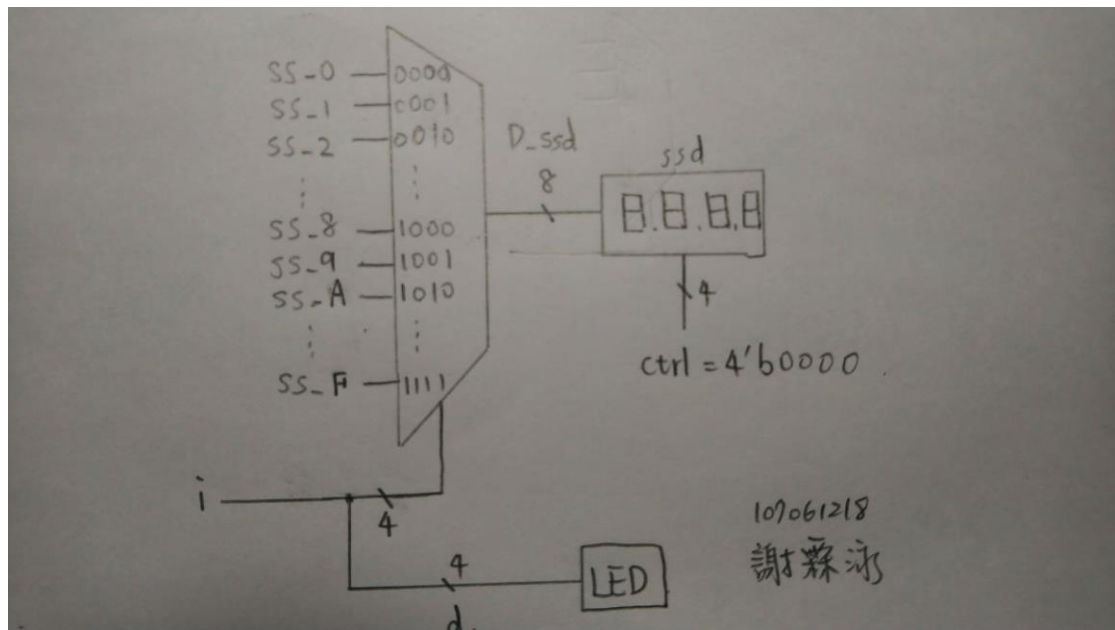
3.

➤ Design Specification

For a binary to 7-segment display:

Input: [3:0]  $i$

Output: [7:0] D\_ssd, [3:0]  $d$ , [3:0] ctrl



### ➤ Design Implementation

此題與上一小題的不同點僅在於 input 值在 10 以上的處理。上一小題若輸入的  $i$  超過 9，一律顯示為 F，這一題則是用 16 進位的方式顯示。換言之， $i=4'd10$  顯示 A， $i=4'd11$  顯示 B..... $i=4'd15$  則顯示 F。

首先，由 FPGA 板下方最右邊的 4 個開關作為 4-bit  $i$  的輸入。SS\_3 的意思就是對應「3」這個數字顯示在七段顯示器上的結果，因為還有小數點，因此為 8-bit，依此類推。因為七段顯示器為 low active，因此，有亮的段值設為 0，暗的段則設為 1。以輸入的  $i$  作為 MUX 的 selection，去選擇對應的結果，並將之傳到七段顯示器。

此外，也另一個 4bit 的變數  $d$  作為 LED 燈的控制，哪個開關被打開，他上方的 LED 燈就要亮起來。因為 LED 與開關均為 high active，因此，其值與所輸入的  $i$  值理應相同，即  $d=i$ 。

另外，我還設計了一個 4bit 的 input 名為 ctrl，是用來控制 ssd 要亮哪個 digit，因為無特別要求，所以當然 4 個 digit 都亮，因此，ssd 的值我就直接給定為  $4'b0000$ 。

IO pin assignment:

$i[3]$	$i[2]$	$i[1]$	$i[0]$	$d[3]$	$d[2]$	$d[1]$	$d[0]$
W17	W16	V16	V17	V19	U19	E19	U16

D_ssd[7]	D_ssd[6]	D_ssd[5]	D_ssd[4]	D_ssd[3]	D_ssd[2]	D_ssd[1]	D_ssd[0]
W7	W6	U8	V8	U5	V5	U7	V7

ctrl[3]	ctrl[2]	ctrl[1]	ctrl[0]
W4	V4	U4	U2

➤ Discussion

這一題與上一小題大致雷同，只要把上一小題的 decoder 增加 A~F 等顯示就可以了，並沒有遇到太大的問題。

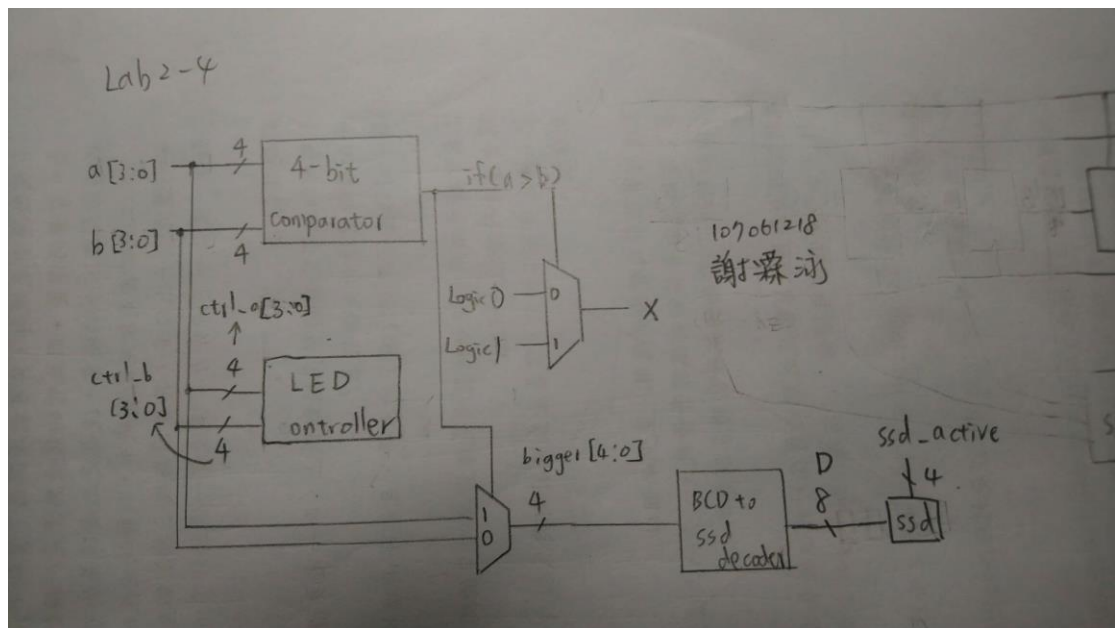
4.

➤ Design specification

For a 4-bit number comparator

Input: [3:0] a, [3:0] b

Output: x, [3:0] ctrl\_a, [3:0] ctrl\_b, [7:0] D, [3:0] ssd\_active



➤ Design Implementation

輸入兩個 4-bit 的數 a 和 b，進入 1 個 4-bit 的 comparator，其結果作為下方 MUX 要選擇 1 或 0 作為 output x 輸出的 selection。若  $a > b$ ，則  $x=1$ ，若  $x \leq b$ ，則  $x=0$ 。x 的輸出為 LED 燈 U14，若 x 為 1 則亮，反之則不亮。

再設兩 4-bit 變數 ctrl\_a 與 ctrl\_b，作為 LED 燈的控制。也就是把哪個開關撥上來，其上相對應的 LED 燈也要亮，所以  $ctrl\_a = a$  且  $ctrl\_b = b$ 。

此外，我還多設計一個功能，就是將較大那個數顯示到 ssd 上。那麼，要先選擇要顯示哪一個數，令一 4-bit 變數叫 bigger，用來記錄大數。其設計構想為利用 comparator 的結果作為另一 MUX 的 selection，若  $a > b$ ，則  $bigger = a$ ；若  $a \leq b$ ， $bigger = b$ 。

再來，將 4-bit 的 bigger 變數傳入 ssd decoder，得到 1 個 8-bit 的結果作為 output 傳到 7-segment display 將之顯示出來。ssd\_active 是用來控制 ssd 要亮哪個 digit，因為 4 個 digit 都要亮，因此，ssd 的值我就直接給定為 4'b0000。

Input 腳位

a[3]	a[2]	a[1]	a[0]	b[3]	b[2]	b[1]	b[0]
W17	W16	V16	V17	R2	R1	V1	W2
最右邊 4 個開關				最右邊 4 個開關			

LED 控制

ctrl_a[3]	ctrl_a[2]	ctrl_a[1]	ctrl_a[0]	ctrl_b[3]	ctrl_b[2]	ctrl_b[1]	ctrl_b[0]
V19	U19	E19	U16	L1	P1	N3	P3
最右邊 4 個 LED				最右邊 4 個 LED			

ssd decoder

D [7]	D [6]	D [5]	D [4]	D [3]	D [2]	D [1]	D [0]
W7	W6	U8	V8	U5	V5	U7	V7

ssd active 與 output

ssd_active [3]	ssd_active [2]	ssd_active [1]	ssd_active [0]	x
W4	V4	U4	U2	U14

## ➤ Discussion

這一題輸入了兩個數，藉由他們的大小來決定輸出是 1 還是 0，並將大數顯示到七段顯示器上。因為題目要求  $a < b$  時 output  $x = 1$ ，其他狀況均為零，所以，我也測試過板子上的結果。當我兩邊的數大小相同時，代表 x 的那個 LED 燈是不會亮的，與我的設計預設結果一致。

## 5. conclusion

這次實驗更讓我了解到 block diagram 的重要，也更讓我體會到一個 module、一個 module 的概念。這次最後一題顯示到 ssd 的部分應該可以呼叫上一小題的 module，但這次我的打法並沒有呼叫之前 module，而是又把

它全部打出來，之後會嘗試呼叫 module 的方式。