

Lab2 Report

2-1. A one bit full adder

●Design Specification

1. Input: x, y, cin
2. Output: $s, cout$
3. $s + cout = x + y + cin$



XOR		z
x	y	z
0	0	0
0	1	1
1	0	1
1	1	0

●Design Implementation

1. Logic function:

s 是 sum; $cout$ 則是表示進位與否。

利用 \oplus (XOR) 的特性，要有奇數個 "1"，結果才會是 1，來表示 s 。

利用相乘的特性，要 input 都是 1，結果才會是 1，來表示 $cout$ 。

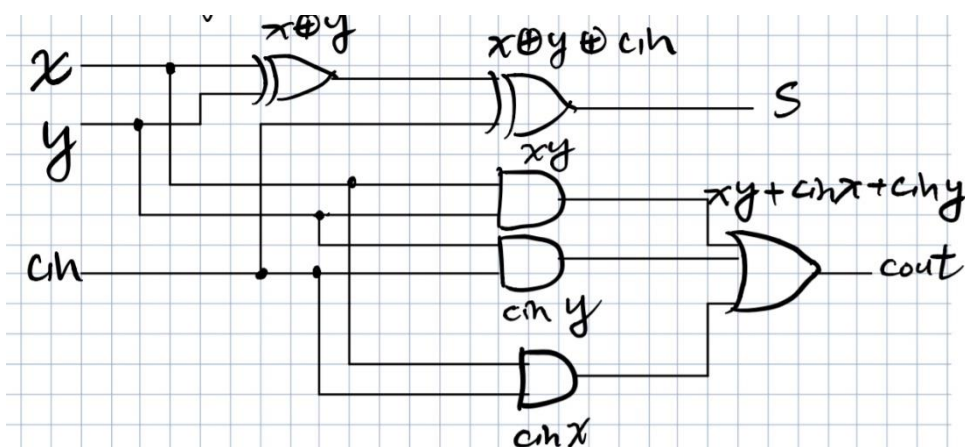
2. Logic equations:

$$S = x \oplus y \oplus cin$$

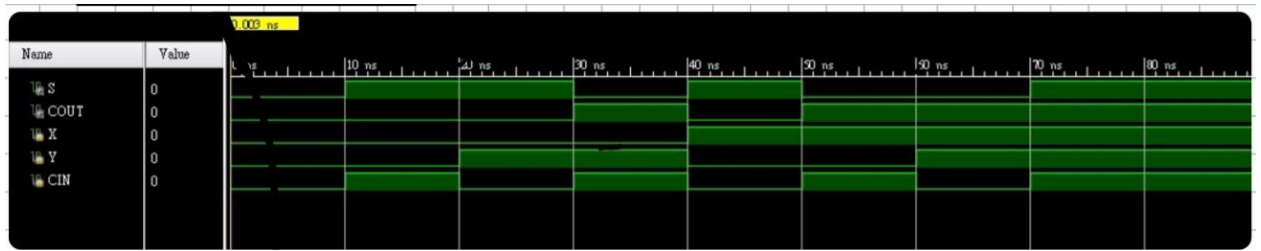
$$cout = xy + xcin + ycin$$

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

3. Logic diagram:



●Stimulation



●Reference

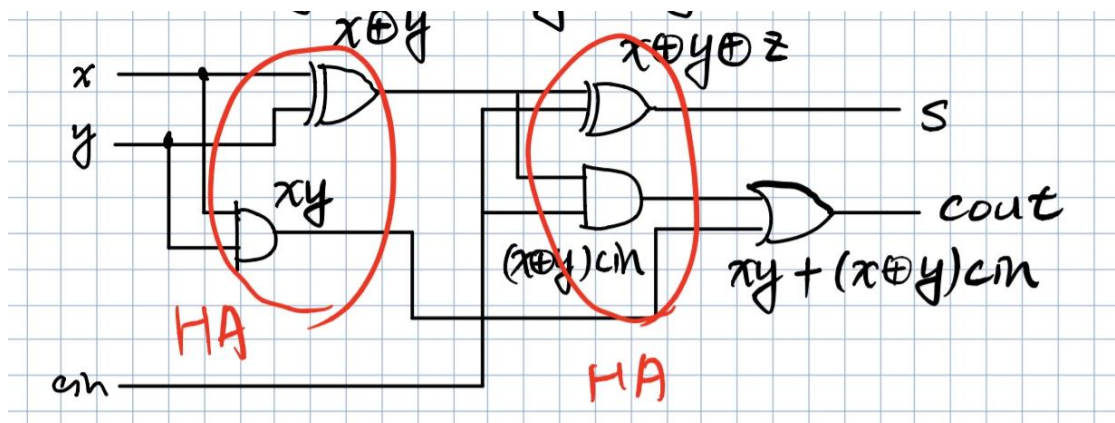
參照講義

另外還可以用 half adder 來表示 full adder

1. Logic equations:

$$\begin{aligned} \text{Cout} &= xy + x\text{in} + y\text{cin} \\ &= xy + \text{cin}(x + y) \\ &= xy + \text{cin}(xy + xy' + x'y + xy) \\ &= xy + xy\text{cin} + xy'\text{cin} + x'y\text{cin} \\ &= xy + \text{cin}(xy' + x'y) \end{aligned}$$

2. Logic diagram:



●Discussion

1. 測試出來的結果跟原本預期的一樣。
2. 因為要打這個結報而意外學會很多打其他符號的方式，像是 XOR 的符號 \oplus ，可以利用快捷鍵先打 2295 再同時按 Alt+x 即可。
3. 基本上跟 lab1_1 一樣，只要打上全加器的方程式再分別把輸出跟出入連接到板子上的位置設定即可。

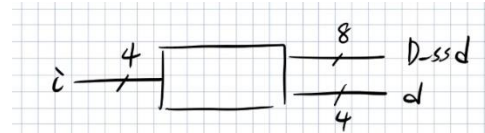
2-2. BCD to 7-segment display decoder

●Design Specification

1. **Input:** $i[3:0]$;

2. **Output:** $D_ssd[7:0], d[3:0], a[3:0]$;

(i 是 binary number, D_ssd 是 7-segment display, d 是為了同時控制 4 個七段顯示器、 a 是 output 為了用 LED 監控 input)



●Design Implementation

8'b00000011 represent 0

8'b10011111 represent 1

8'b00100101 represent 2

8'b00001101 represent 3

8'b10011001 represent 4

8'b01001001 represent 5

8'b01000001 represent 6

8'b00011111 represent 7

8'b00000001 represent 8

8'b00001001 represent 9

8'b01110001 represent F

當輸入的 $i[3:0]$ 轉為十進位後，分別產生下列輸出(0~9 以外輸出 F)

0: 8'b00000011

1: 8'b10011111

2: 8'b00100101

3: 8'b00001101

4: 8'b10011001

5: 8'b01001001

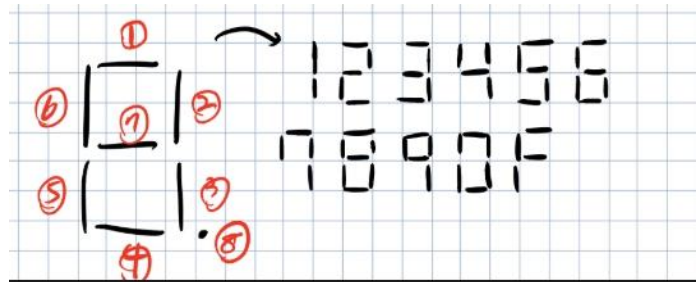
6: 8'b01000001

7: 8'b00011111

8: 8'b00000001

9: 8'b00001001

Default: 8'b01110001



↓以下是對應 FPGA 板的接腳

I/O	D_ss d[7]	D_ss d[6]	D_ss d[5]	D_ss d[4]	D_ss d[3]	D_ss d[2]	D_ss d[1]	D_ss d[0]
Pin	W7	W6	U8	V8	U5	V5	U7	V7

I/O	i[3]	i[2]	i[1]	i[0]	d[3]	d[2]	d[1]	d[0]
Pin	W4	V4	U4	U2	W17	W16	V16	V17

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

●Stimulation

此題已在演習課 demo 過 FPGA 板。

●Discussion

1. 經過這個實驗學會了如何使七段顯示器顯示出我們所想要的樣子，不過目前只能四個都顯示一樣的，如果要寫是不同數字的話，要用短暫時間的 delay(人眼的視覺暫留)來表示。
2. 因為不可以將同一個 output，同時接到兩個 FPGA 的 LED 跟七段顯示器上面，所以除了原本的 D_ssd 之外還需要另外設一個 d 來控制 LED 以監控 input。
3. 學會 case 的用法，也是一種可以表示 if、else 的方式，不過要記得設定 default 值。

2-3. Binary to 7-segment display decoder

●Design Specification

1. **Input:** i[3:0];

2. **Output:** D_ssd[7:0], d[3:0], a[3:0];

(i 是 binary number, D_ssd 是 7-segment display, d 是為了同時控制 4 個七段顯示器、a 是 output 為了用 LED 監控 input)

●Design Implementation

設計大略與 2-2 相同

不同的地方在於為 16 進位

8'b00000101 represent A

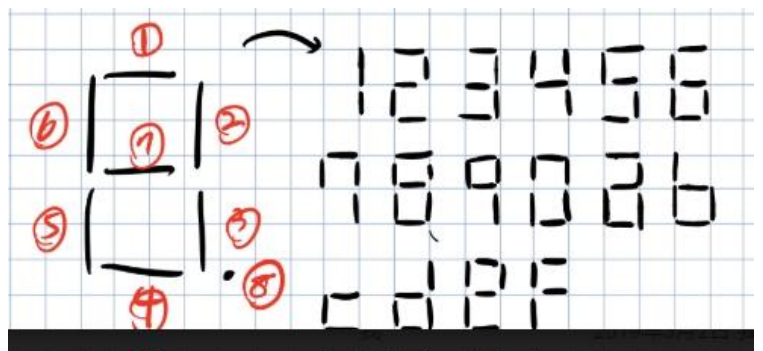
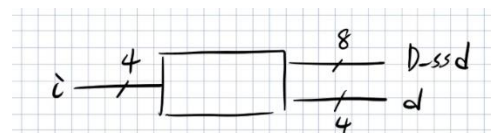
8'b11000001 represent B

8'b11100101 represent C

8'b10000101 represent D

8'b00100001 represent E

8'b01110001 represent F



另外 case 中 default 的值改為 `segs= 8'b00000000;`

↓以下是對應 FPGA 板的接腳

I/O	i[3]	i[2]	i[1]	i[0]	d[3]	d[2]	d[1]	d[0]
Pin	W4	V4	U4	U2	W4	V4	U4	U2

I/O	a[3]	a[2]	a[1]	a[0]
Pin	V19	U19	E19	U16

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

●Stimulation

此題已在演習課 demo 過 FPGA 板。

●Discussion

1. 這個實驗跟 2-2 實驗幾乎一樣，唯一學到的就是怎麼用自己的想像力用七段顯示器寫出 A~F。

2-4. Four Bits Comparator

●Design Specification

1. Input: [3:0]A、[3:0]B;

2. Output:[3:0]M、[3:0]N、X;

(M 用來連接 Input A, N 則用來連接 Input B, 來顯示 LED 燈；Output 則用來連接另外

一個 LED 燈以表示 A、B 的大小，若 A 比 B 大則亮燈，反之則不亮)

3. If(a<=b), x=0

If(a>b), x=1

●Design Implementation

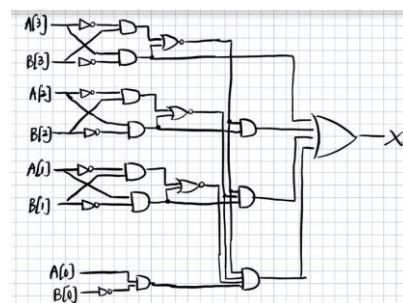
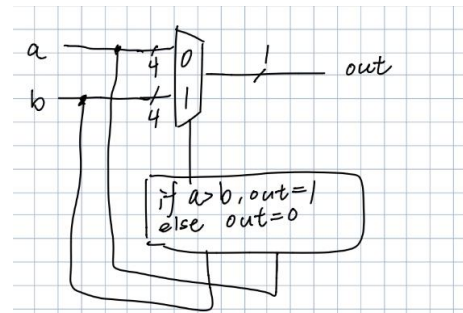
1. logic function

If(A<=B)

X=0

else

X=1



↑2. Logic diagram

M=A

N=B

↓以下是對應 FPGA 板的接腳

I/O	A[3]	A[2]	A[1]	A[0]	B[3]	B[2]	B[1]	B[0]
Pin	W17	W16	V16	V17	W13	W14	V15	W15
I/O	M[3]	M[2]	M[1]	M[0]	N[3]	N[2]	N[1]	N[0]
Pin	V19	U19	E19	U16	V14	U14	U15	W18
I/O	X							
Pin	L1							

●Stimulation

此題已在演習課 demo 過 FPGA 板。

●Discussion

1. 我對這題的問題跟 pre-lab3 一樣，就是在 verilog 上的程式碼中只要簡單打說如果 A 大於 B 的敘述就可以操作了，但是在之前學的邏輯設計中需要利用 \oplus (XOR) 分別來比較每個位數的大小，所以不太清楚在結報上要如何表示，但為了確保正確，我這幾次的報告兩種都有附上，會在之後上課詢問助教或教授解惑。
2. 其他就是更熟悉對於 FPGA 板子的操作了，已經可以不用看之前的講義來逐步模仿，就可以把自己寫的程式碼在板子上成功模擬了。

●Conclusion

這次的實驗讓我第一次了解到邏輯設計實驗的威力，大約整整花了兩天才搞清楚加上弄完全部相關的作業，手拙的我常常會在奇怪的地方把變數打成大寫或是打錯，但是自己都看不出來，這些小小的錯誤就大大造成我的困擾，常常幾個小時就這樣悄悄溜走了。不過最後能夠順利地在板子上面模擬真的還蠻酷炫的，有種終於可以把自己所學運用出來的感覺(雖然大部分還是從講義或是網路上面來的 QAQ)，不過拿著這個板子感覺就很潮，有種好像是電資院或是二專有修相關的人才會拿的到的!我會更加努力學習 VERILOG 的!