Lab2

107060017 廖凰汝

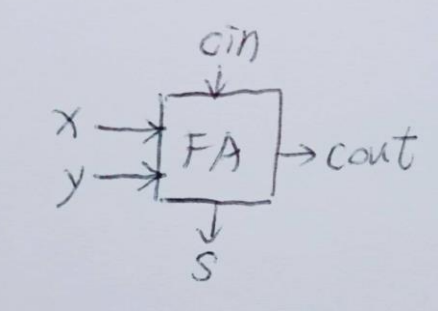
1.
Design Specification

✓ **Input / Output**

For a full adder(s+cout=x+y+cin):

Input: x, y, cin.
Output: s, cout.

✓ **Block Diagram**



Design Implementation

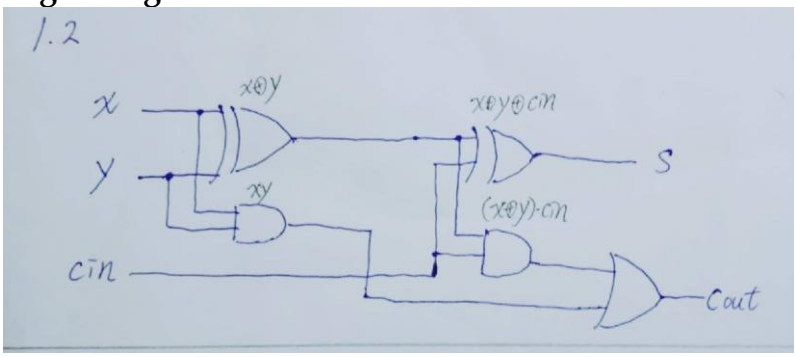
✓ **Function Descriptions**

此功能為輸入三個 1 bit 數字並用二進位加法器輸出 2bit 結果。
做法為上學期邏輯設計所學的 Full adder 的 logic functions,用 assign output 的方式得到 output.

✓ **Logic Functions**

s = x ^ y ^cin
cout = (x ^ y) & cin | x & y

✓ **Logic Diagram**



✓ **I/O pin**

I/O	x	y	cin	s	cout
LOC	V17	V16	W16	U16	E19

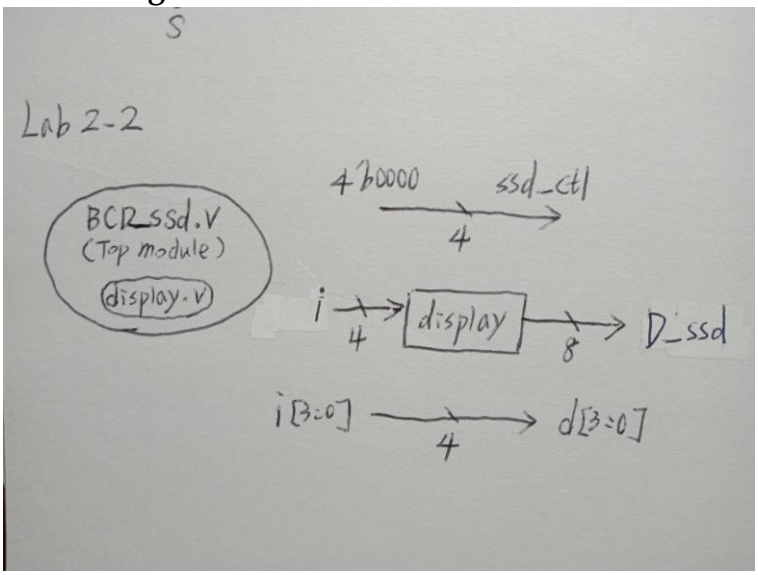
2.
Design Specification

✓ **Input / Output**

For BCD to 7-segment display decoder:

Input: i[3:0].
Output: D_ssd[7:0], d[3:0], ssd_ctl[3:0].

✓ **Block Diagram**



Design Implementation

✓ **Function Descriptions**

本功能為輸入一個 4 bit BCD 數字將他轉為 7-segment display 而把數字顯現在七段顯示器上面，四個顯示器顯示相同數字，4 個 LED 燈 monitor BCD number。在 display module 中，利用查表(如下表)方式將輸入的 BCD 4 bit 數字轉成 8bit 控制 7-segment display 的訊號。另外利用 default 來處理大於 9 的輸入。在 top module(BCD_ssd)中，把控制 which 7-segment to display 的 4bit 輸出全都等於 0 使四個都顯示相同數字。再來就是把控制 LED 燈的訊號直接相等於 BCD 輸入，意思就是哪個 switch 開啟，對應的 LED 燈就亮。

✓ **Logic Functions**

(top module)
d = i.
ssd_ctl = 4'b0000.
(display module)
利用 case 的查表方式將 i->D_ssd
*BCD->7-segment display list
0000->00000011
0001->10011111
0010->00100101
0011->00001101

0100->10011001
0101->01001001
0110->11000001
0111->00011011
1000->00000001
1001->00011001
Default->01110001

✓ **I/O pin**

I/O	i[3]	i[2]	i[1]	i[0]	d[3]	d[2]	d[1]	d[0]
LOC	W17	W16	V16	V17	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]
LOC	W7	W6	U8	V8	U5	V5	U7	V7
I/O	ssd_ctl[3]	ssd_ctl[2]	ssd_ctl[1]	ssd_ctl[0]				
LOC	U2	U4	V4	W4				

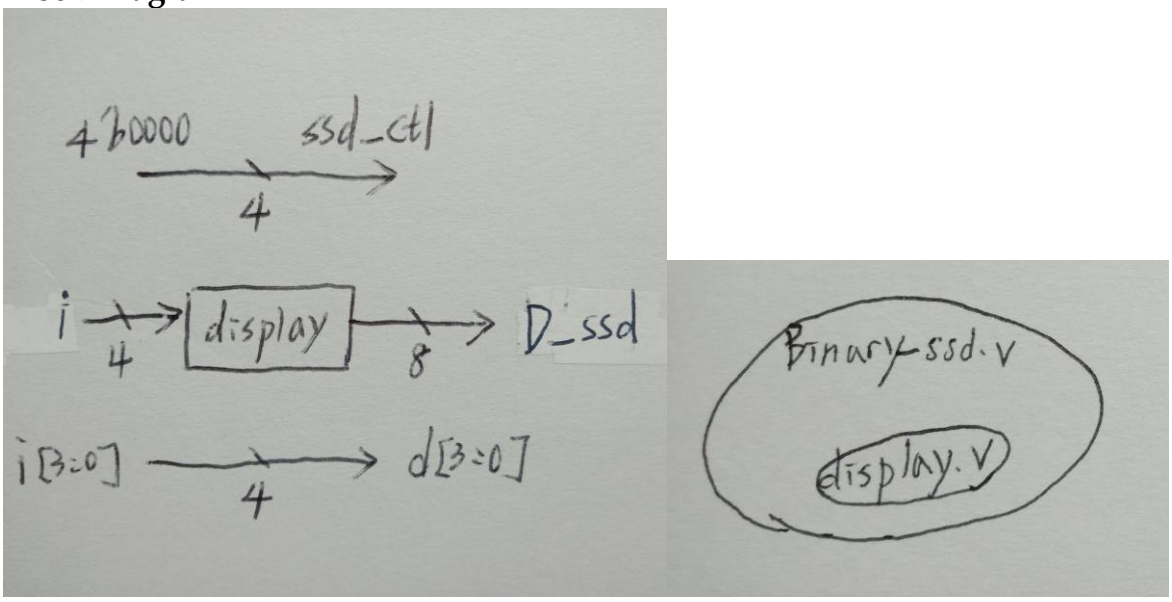
3. Design Specification

✓ **Input / Output**

For a binary to 7-segment display decoder:

Input: i[3:0].
Output: D_ssd[7:0], d[3:0],ssd_ctl[3:0].

✓ **Block Diagram**



Design Implementation

- ✓ **Function Descriptions**
  本功能與上一小題類似，只差在輸入從 BCD 數字改為 4 bit binary numbers。因此除了 display module 中的查表新增了 9-15 的輸入與對應輸出，其餘都沒有變。
- ✓ **Logic Functions**
  d = i.
  ssd_ctl = 4'b0000.
  利用 case 的查表方式將 i->D_ssd
  *binary->7-segment display list
  0000->00000011
  0001->10011111
  0010->00100101
  0011->00001101
  0100->10011001
  0101->01001001
  0110->11000001
  0111->00011011
  1000->00000001
  1001->00011001
  1010->00010001
  1011->11000001
  1100->01100011
  1101->10000101
  1110->01100001
  1111->01110001
Default->01110001
- ✓ **I/O pin**

I/O	i[3]	i[2]	i[1]	i[0]	d[3]	d[2]	d[1]	d[0]
LOC	W17	W16	V16	V17	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]
LOC	W7	W6	U8	V8	U5	V5	U7	V7
I/O	ssd_ctl[3]	ssd_ctl[2]	ssd_ctl[1]	ssd_ctl[0]				
LOC	U2	U4	V4	W4				

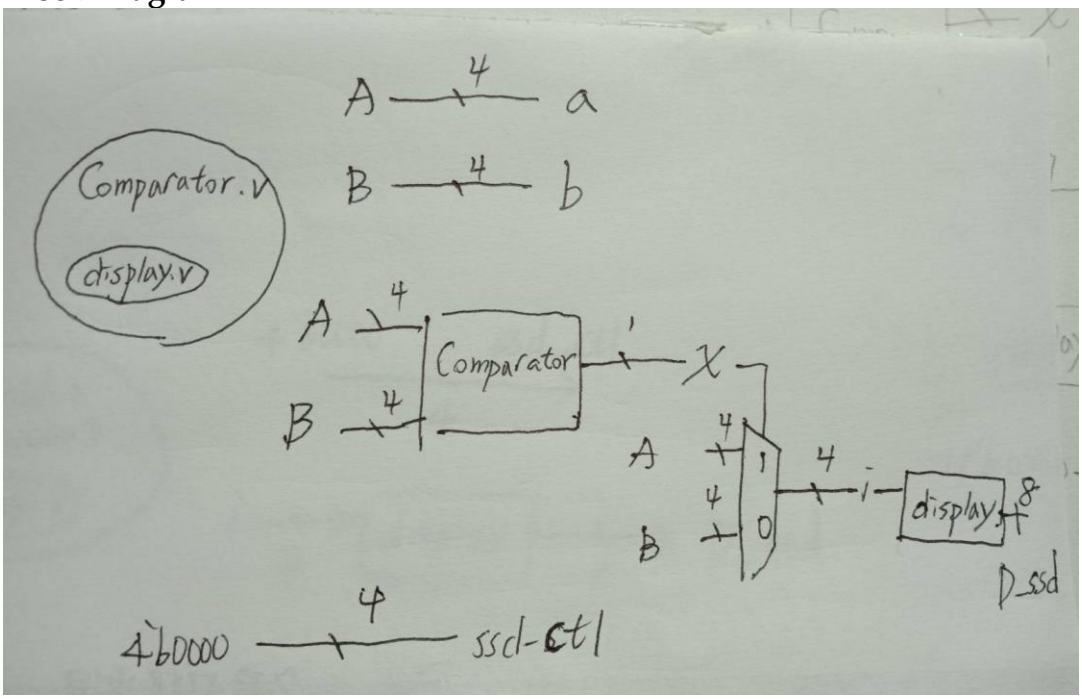
4.
Design Specification

- ✓ **Input / Output**

  For a 4-bit magnitude comparator:

  Input: A[3:0], B[3:0].
  Output: ssd_ctl [3:0], x, a[3:0], b[3:0], D_ssd [7:0].

✓ **Block Diagram**



Design Implementation

✓ **Function Descriptions**

此功能為輸入兩個 4bit binary numbers A, B，若 A>B 則輸出 1 使 LED 亮，若 A<=B 則輸出 0. 然後 binary numbers are displayed on 8 LEDs. 另外，題目沒有說但我多加的功能為，使 7 段顯示器顯示較大的那個數字，若兩個一樣的則顯示 B。

我的作法為使作一個 comparator(用 if else)來比較 A 是否>B，若是，則輸入 A 到 binary to 7-segment display module 並使 LED 的輸出為 1.若非，則輸入 B 到 display module 並使 LED 輸出為零. 至於 binary to 7-segment display module 的作法與上題相同。

對於 binary numbers are displayed on 8 LEDs 的功能，直接使 output = input.

✓ **Logic Functions**

a = A.
b = B.
ssd_ctl = 4'b0000.
if (A>B) x = 1, i=A.
else x = 0, i=B.
(display module)
利用 case 的查表方式將 i->D_ssd
*binary->7-segment display list
0000->00000011
0001->10011111
0010->00100101
0011->00001101
0100->10011001
0101->01001001
0110->11000001
0111->00011011

1000->00000001
1001->00011001
1010->00010001
1011->11000001
1100->01100011
1101->10000101
1110->01100001
1111->01110001
Default->01110001

✓ **I/O pin**

I/O	A[3]	A[2]	A[1]	A[0]	B[3]	B[2]	B[1]	B[0]
LOC	W13	W14	V15	W15	W17	W16	V16	V17
I/O	a[3]	a[2]	a[1]	a[0]	b[3]	b[2]	b[1]	b[0]
LOC	V14	U14	U15	W18	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]
LOC	W7	W6	U8	V8	U5	V5	U7	V7
I/O	ssd_ctl[3]	ssd_ctl[2]	ssd_ctl[1]	ssd_ctl[0]	x			
LOC	U2	U4	V4	W4	L1			

Discussion

第一次把 verilog code 實作到 FPGA 板上，滿興奮的。而這禮拜碰到比較新的觀念是七段顯示器的用法，要做一個 BCD or binary to 7-segment display 的 decoder 大概都是用查表的方式作。另外可以先 define 一些常數，若發現自己再算哪裡該亮哪裡不該時有寫錯，直接修正比較清楚。
這禮拜像助教 demo 時有學到，不要因為懶就直接按 generate bitstream，要先按 run synthesis 這樣若有錯誤訊息比較易讀。

Conclusion

這次學習到如何將 velilog code program 到 FPGA，以及七段顯示器的原理及用法。
作一個 BCD or binary to 7-segment display 的 decoder 可以用 case 來查表。
只是這次七段顯示器我們都還是使他同時顯示同個數字，並不會要用到特定頻率去控制視覺暫留的方法。

