Lab02 (109061256 陳立萍)

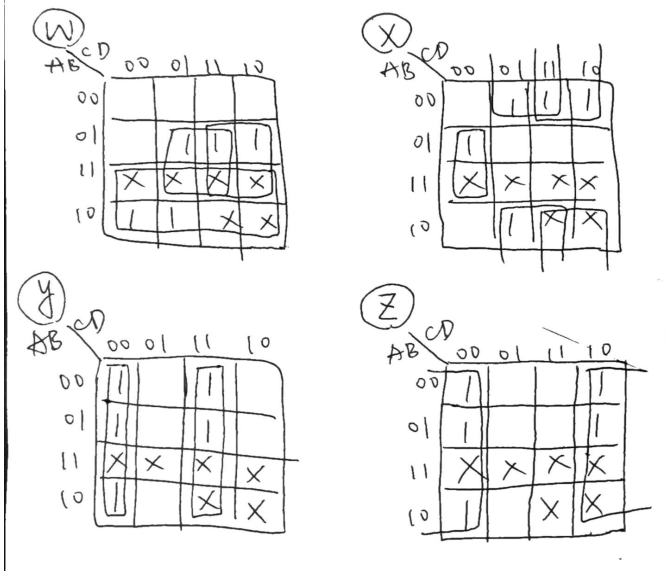
1. Lab2_1: a BCD-to-Excess-3 code converter

Design Specification

input: abcd(MSB to LSB)
output: wxyz(MSB to LSB)
abcd: 0000 ~ 1001

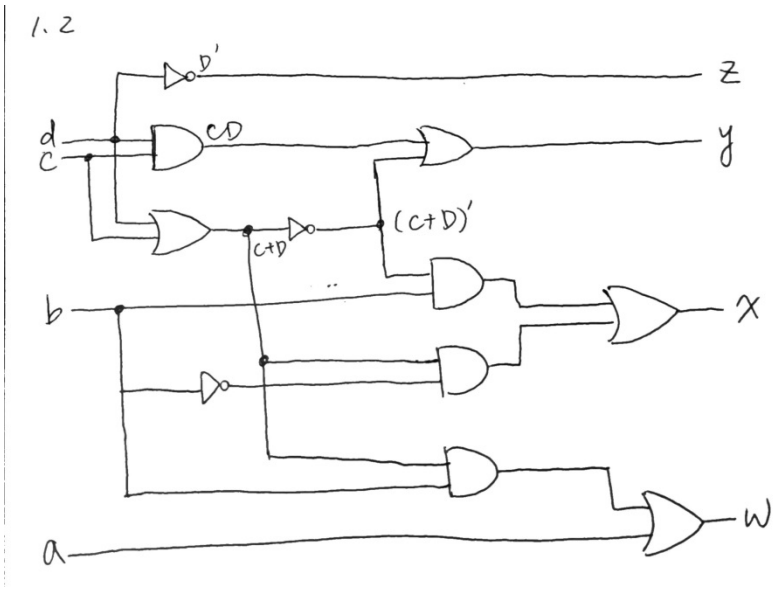
$wxyz = abcd + 0011$

a	b	c	d	w	x	y	z
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0	X	X	X	X
1	0	1	1	X	X	X	X
1	1	0	0	X	X	X	X
1	1	0	1	X	X	X	X
1	1	1	0	X	X	X	X
1	1	1	1	X	X	X	X



$w = a + bc + bd$
$x = b'c + b'd + bc'd'$
$y = cd + c'd'$
$z = d'$

logic diagram



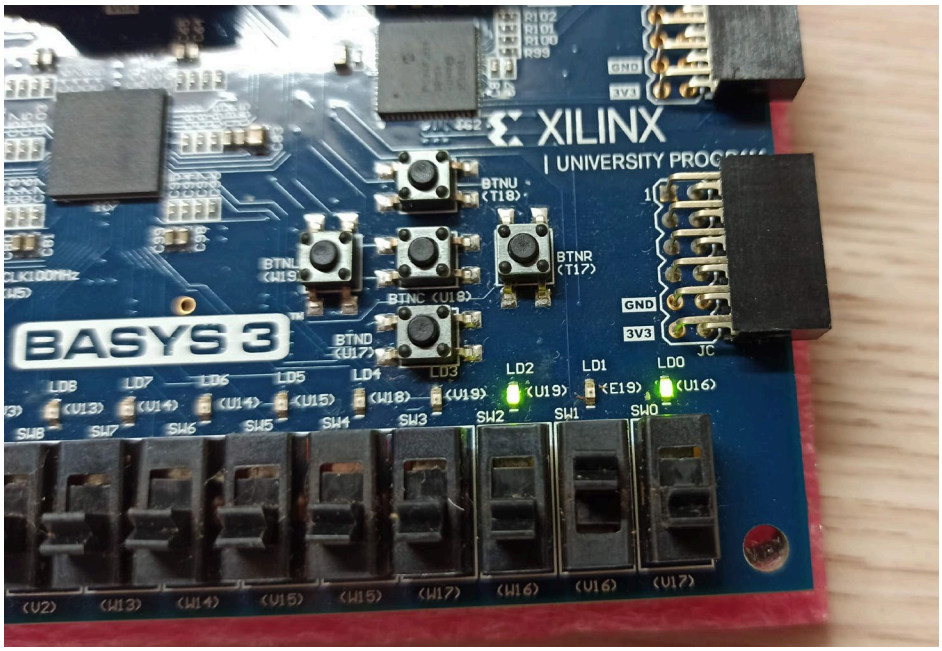
I/O pin assignment:

a	b	c	d
W17	W16	V16	V17

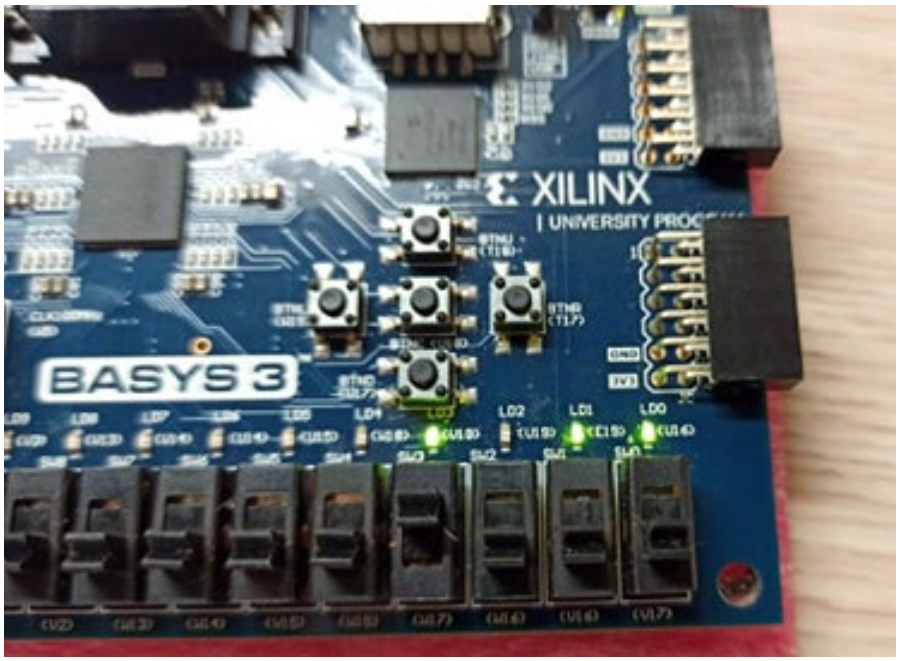
W	x	y	Z
V19	U19	E19	U16

Design Implementation

abcd =0010, wxyz =0101



abcd =1000, wxyz =1011



設計方法：

BCD 轉換成 Excess-3 的式子如上 1.1 ，wxyz = abcd + 0011，接著寫出對應的 truth table，再用 k-map 化簡 output (wxyz)，得到 wxyz 各自的 Boolean function.
此即是如同 lab1_1，只是加了接上 pin 腳的動作，根據 lab2_1 題目要求的方式一個一個接上 pin 腳，就可以在板子上得到對應的結果。

2. Lab2_2:

Design Specification

Lab2_display

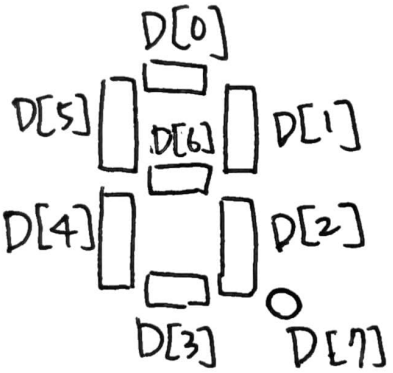
input: [3:0]i

output: [7:0]D, [3:0]d

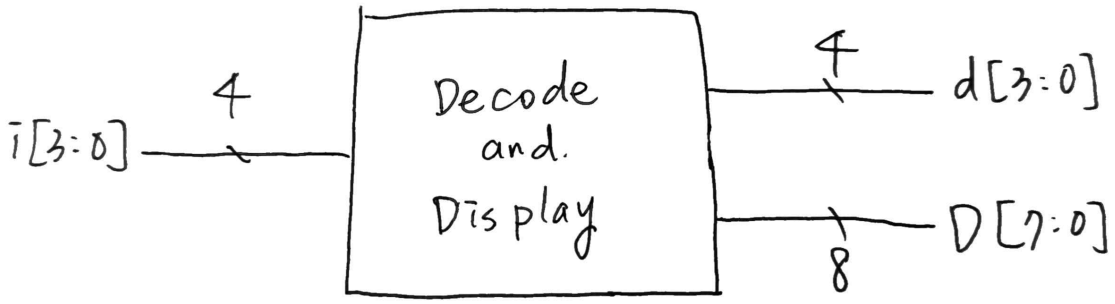
reg [7:0] D

---

	binary	7-segment
0	0000	0 0 0 0
1	0001	1 1 1 1
2	0010	2 2 2 2
3	0011	3 3 3 3
4	0100	4 4 4 4
5	0101	5 5 5 5
6	0110	6 6 6 6
7	0111	7 7 7 7
8	1000	8 8 8 8
9	1001	9 9 9 9
10	1010	A A A A
11	1011	b b b b
12	1100	C C C C
13	1101	d d d d
14	1110	E E E E
15	1111	F F F F



logic diagram



I/O pin assignment:

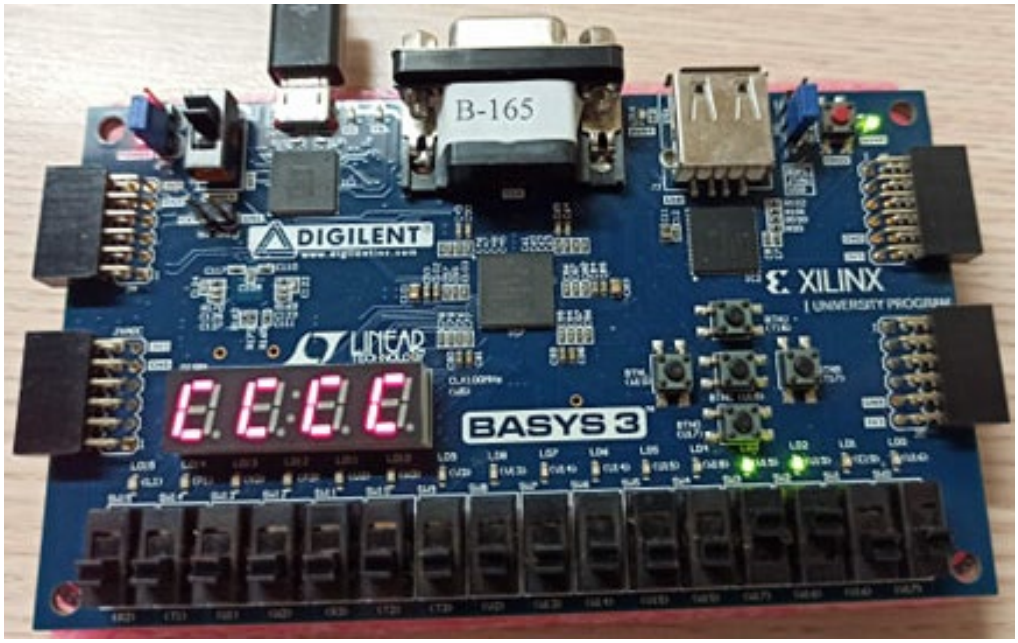
D[7]	D[6]	D[5]	D[4]	D[3]	D[2]	D[1]	D[0]
W7	W6	U8	V8	U5	V5	U7	V7

d[3]	d[2]	d[1]	d[0]
V19	U19	E19	U16

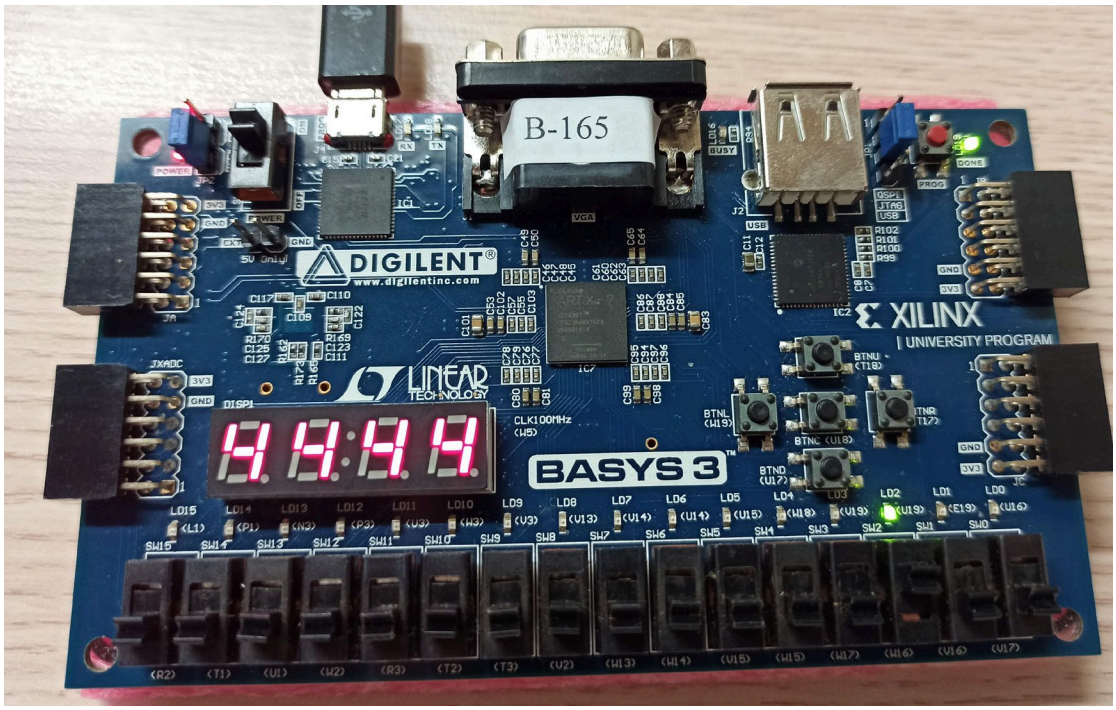
i[3]	i[2]	i[1]	i[0]
W17	W16	V16	V17

Design Implementation

[3:0]i = 1100 = C



[3:0]i = 0100 = 4



設計方法：

先定義每一個數字的七段顯示器顯示方式，1 是代表燈亮，0 代表燈不亮，而 D[7]至 D[0]由上圖可知各自的定義位置，再依照題目設定將 input 設定為 i[3:0]，接著判斷輸入的 i 等於多少，再用 define 過的值輸出應對的結果，另外 d[3:0]用來控制四個開關上方的 LED 燈要不要量，如果 i[x]=1 則 LED 要亮，若 i[x]=0 則燈不亮。

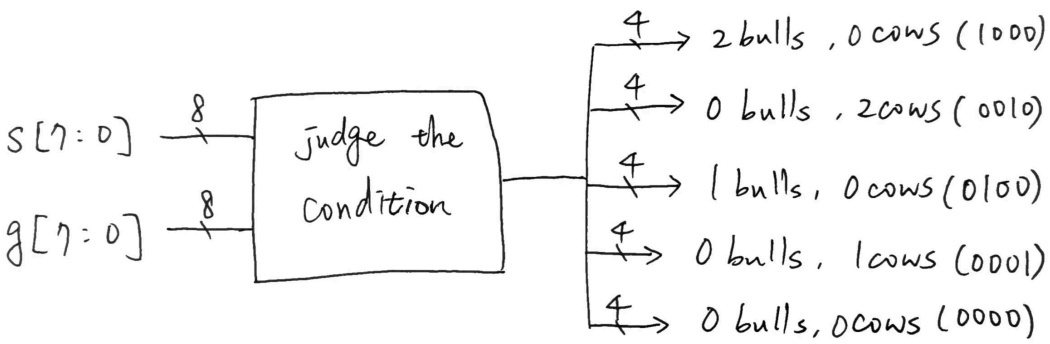
Bonus. Lab2_bonus: Bulls and Cows game.

Design Specification

input [7:0] s, [7:0] g

output [1:0] bulls, [1:0] cows

logic diagram



I/O pin assignment:

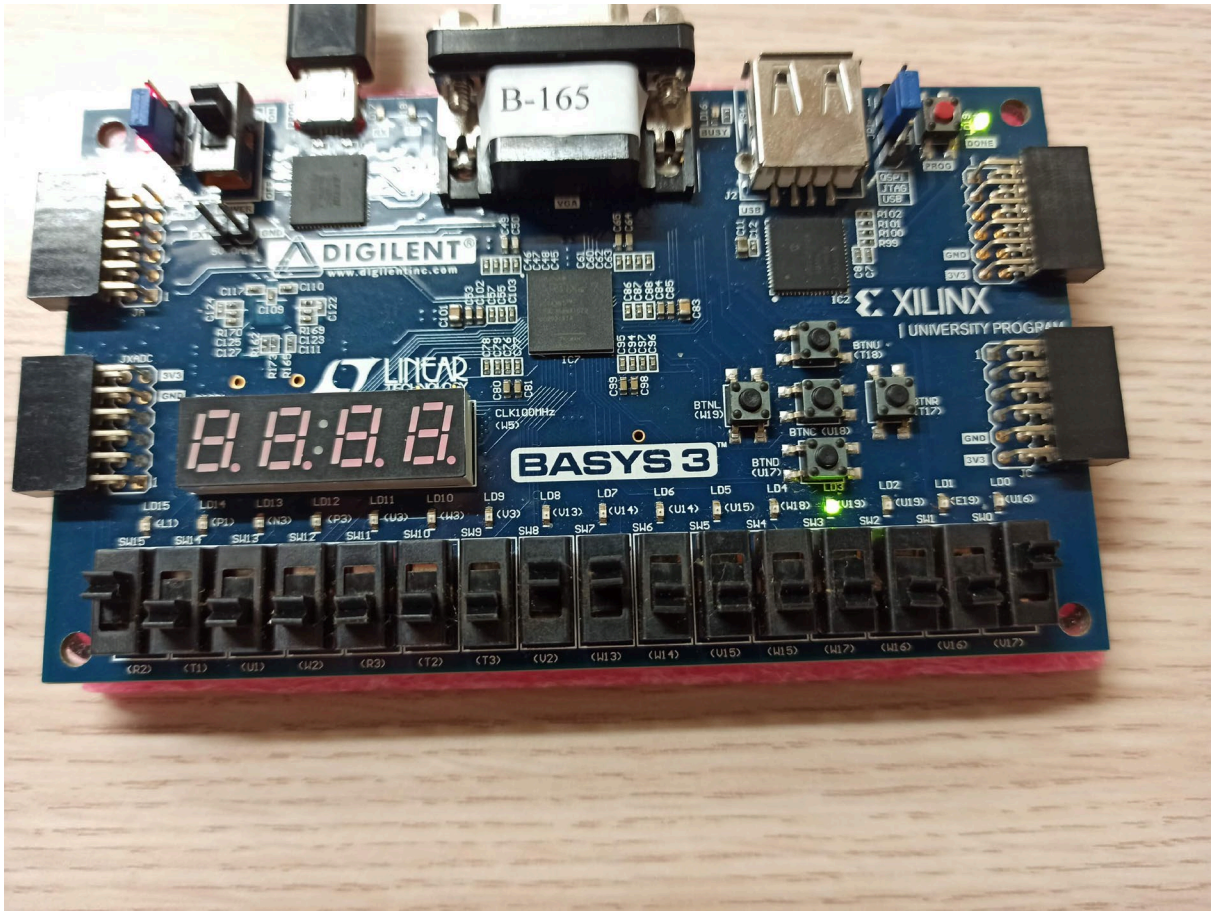
s[7]	s[6]	s[5]	s[4]	s[3]	s[2]	s[1]	s[0]
W13	W14	V15	W15	W17	W16	V16	V17

g[7]	g[6]	g[5]	g[4]	g[3]	g[2]	g[1]	g[0]
R2	T1	U1	W2	R3	T2	T3	V2

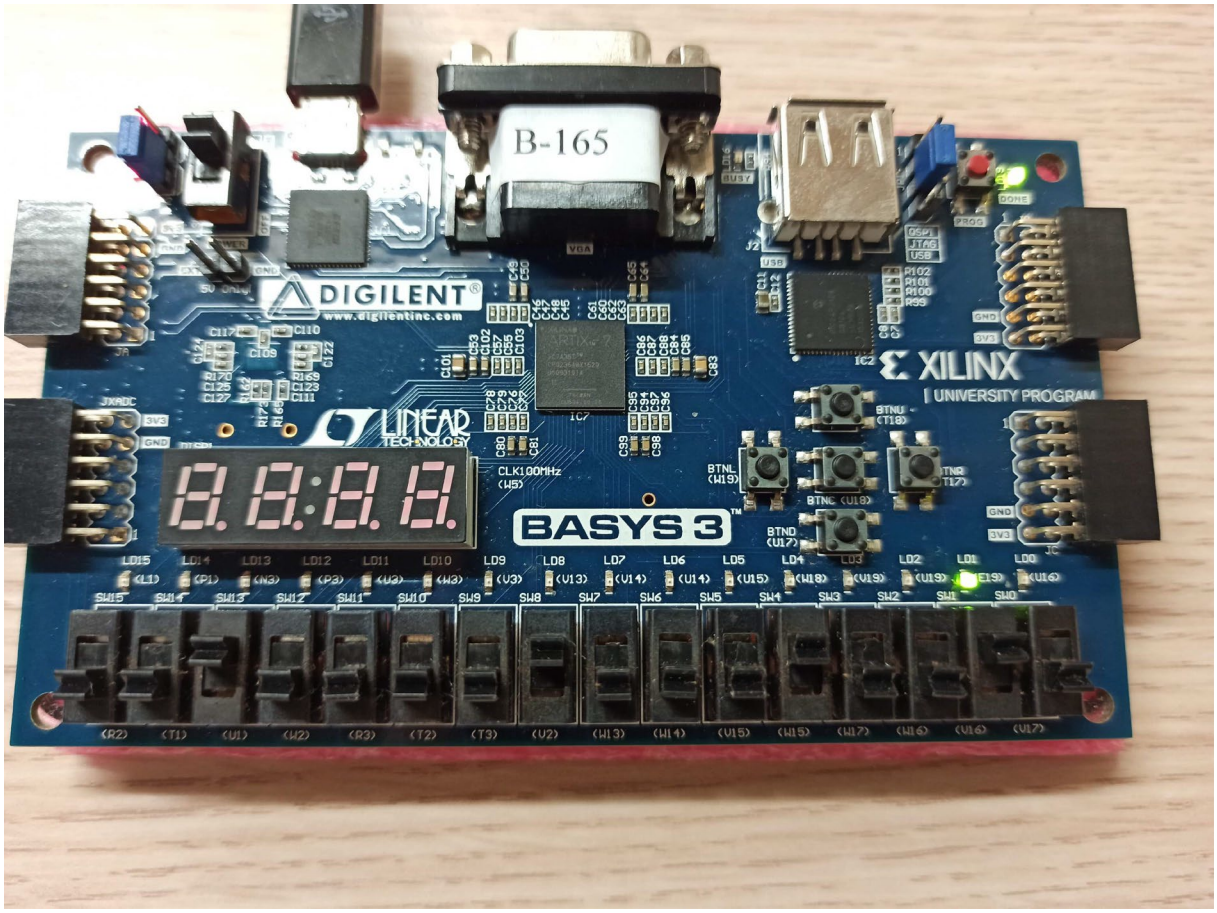
bulls[1]	bulls[0]	cows[1]	cows[0]
V19	U19	E19	U16

Design Implementation

[7:0]s=10000001, [7:0]g=10000001, bulls=10, cows=01



[7:0]s=00100001, [7:0]g=00010010, bulls=00, cows=10



設計方法：

這裡設定 s 為 secret number,而 g 為 guess number，各為 8 個 bits。在這裡使用 if, else if, else 的語法來判斷猜中的 bulls 數目和 cows 數目，並用 V19, U19, E19, U16 四顆燈泡表示 bulls 和 cows 的數目，其中 V19, U19 燈泡表示 bulls 數目， E19, U16 燈泡表示 cows 數目, 1 是亮 0 是不亮，舉例來說，若 U19 燈泡亮，V19 燈泡不亮即表示 2 進位的 01 也就是十進位的 1，也就是代表 1 個 bulls，同理若是 U19 燈泡不亮 V19 燈泡亮亦即表示 2 進位的 10 也就是 10 進位的 2。

Discussion

Lab2_1

這題非常簡單，用 lab1_的程式碼就可以執行，再按照題目的要求即可在 fpga 版上呈現亮燈，可以看到執行的結果是否正確結果顯示與 truth table 相符。

Lab2_2

這題其實有很多做法，但是剛好看到 define 語法，便使用看看 define 語法，用此種方法雖然不是最簡短的，但是最容易了解，且如果燈亮處定義錯誤，也很方便更改。這題一開始 define 每一個七段顯示器就都正確，因此執行的結果皆正確。

Lab2_bonus

這題原本看似非常難，因為題目的敘述複雜，很難理解，後來理解後，我就想用 if, else if 的語法來作判斷，雖然使得版面非常冗長，但卻是最容易理解的方法，直接判斷並輸出。另外這題在做完的時候一直顯示 not got the license 一直無法執行，搞了將近 1 小時候才想到，原來是把資料夾放在有中文的目錄下，因此才會無法執行，修改為英文檔名後，馬上就執行成功了。

Conclusion

這次的 lab 花了好多時間在寫，因為第一個 lab 和上學期學到的就只需要用 assign，而這次的 lab 卻需要許多不同的語法和想法才能成功做出來，另外這是第一次把結果用到板子上，可以看到實際的結果，感覺學到了很多，尤其是做 lab 的過程與思考。

References

教授授課頭影片：語法運用，符號運用。