**Lab02 (109061256 陳立萍)**

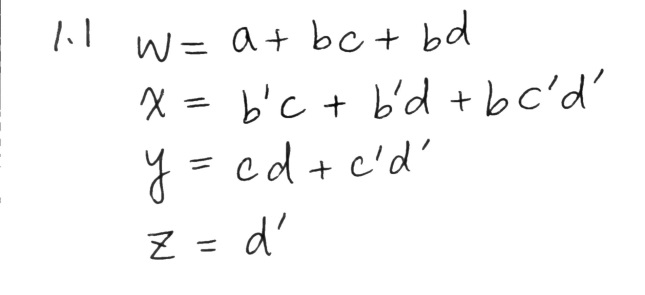
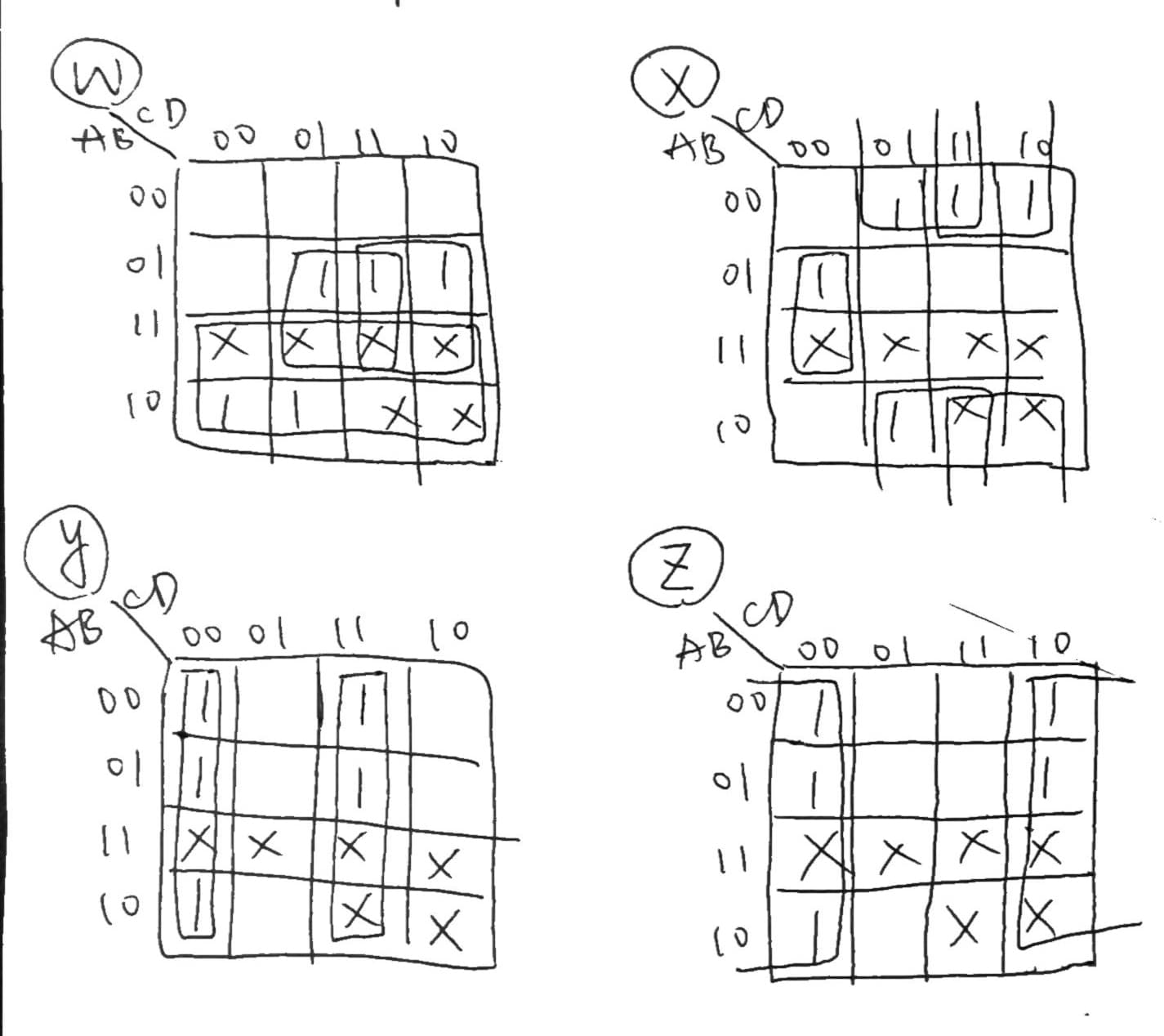
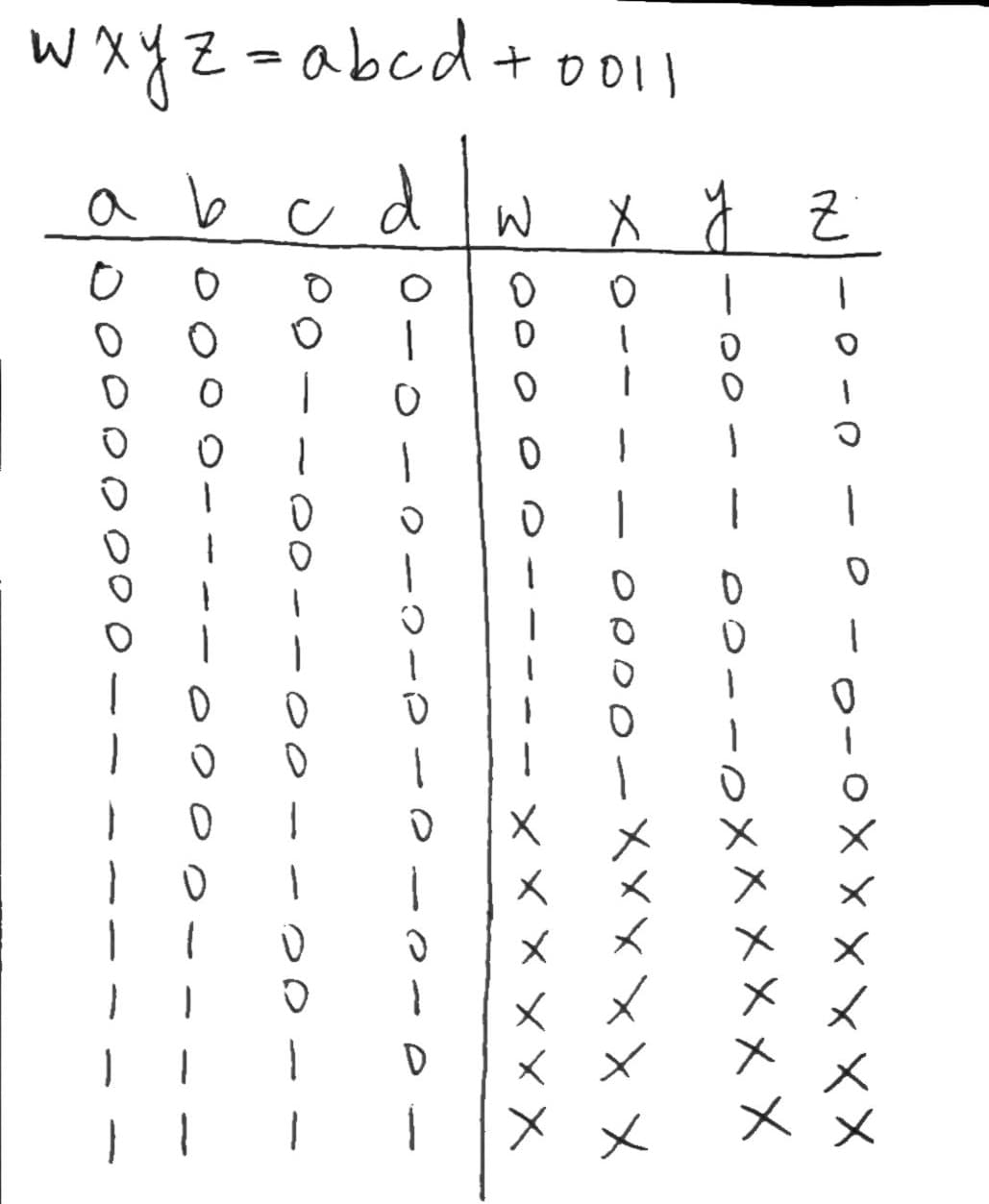
1. **Lab2\_1: a BCD-to-Excess-3 code converter**

**Design Specification**

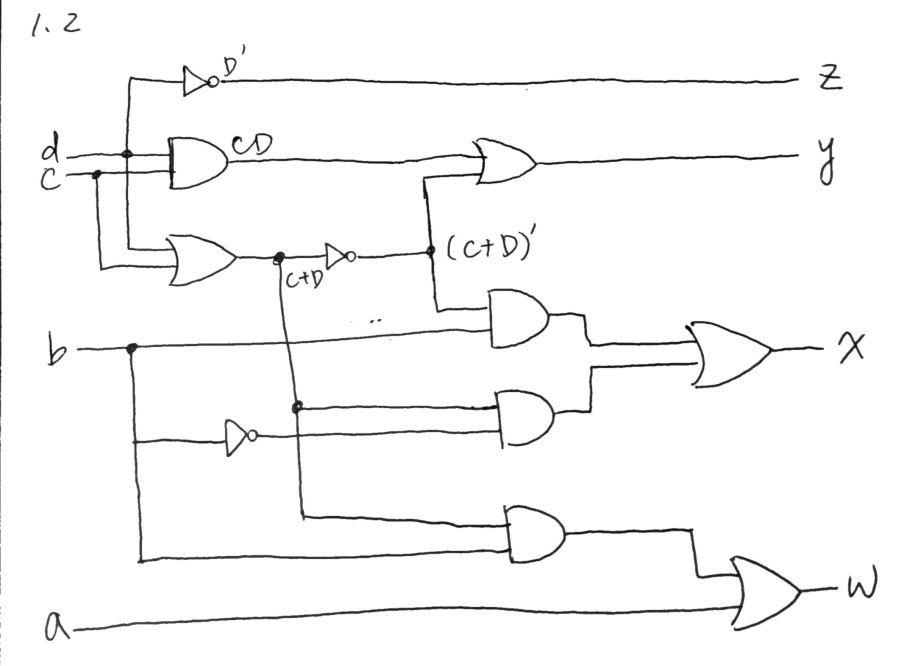
input: abcd(MSB to LSB)

output: wxyz(MSB to LSB)

abcd: 0000 ~ 1001



logic diagram



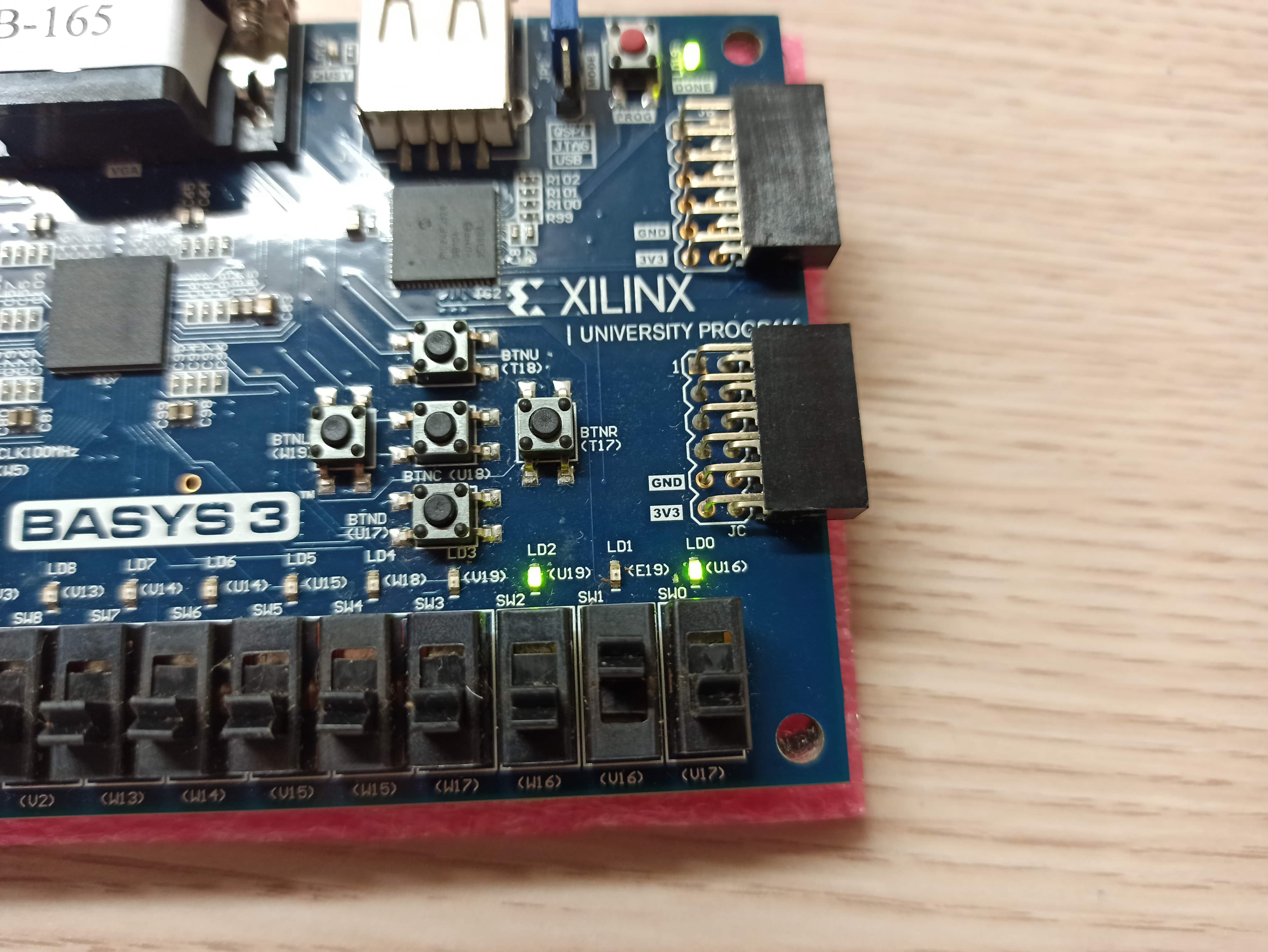
I/O pin assignment:

|  |  |  |  |
| --- | --- | --- | --- |
| a | b | c | d |
| W17 | W16 | V16 | V17 |

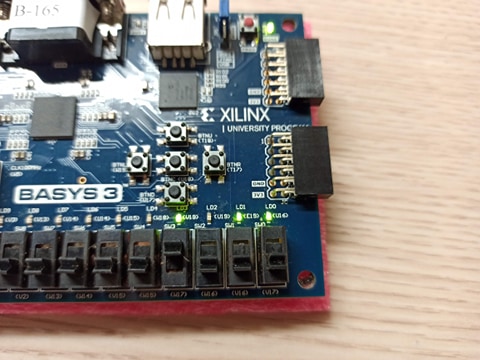
|  |  |  |  |
| --- | --- | --- | --- |
| W | x | y | Z |
| V19 | U19 | E19 | U16 |

**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。

1. **Lab2\_2:**

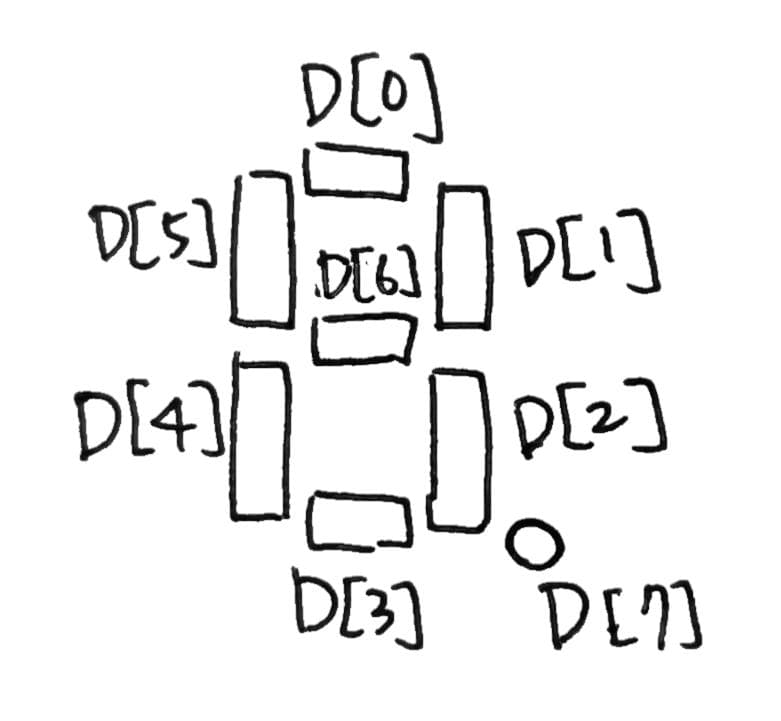
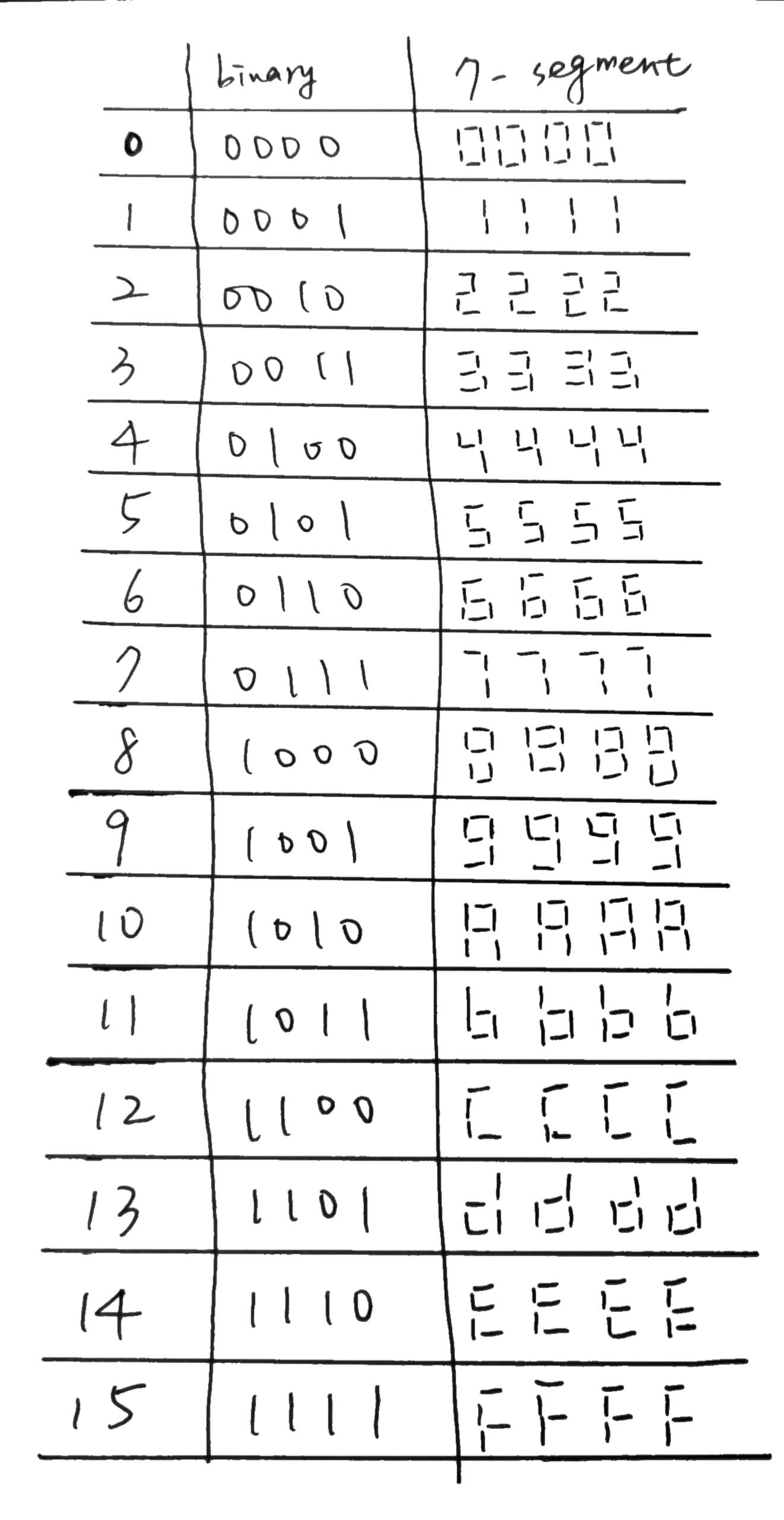
**Design Specification**

**Lab2\_display**

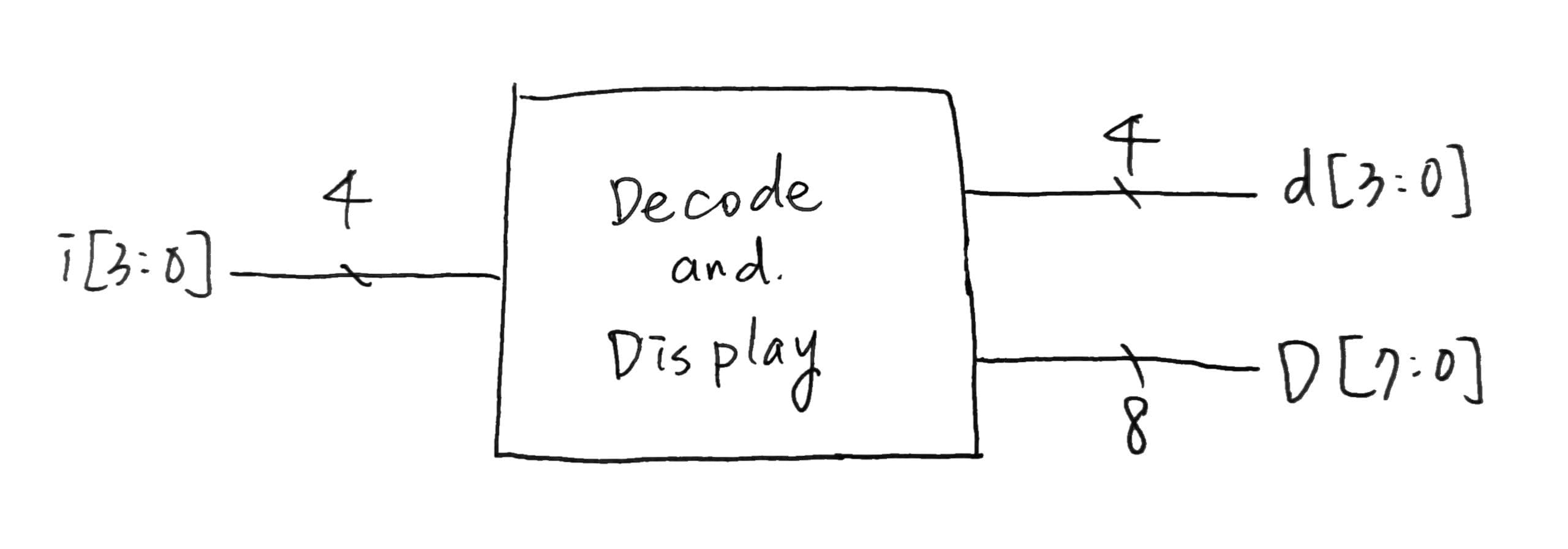
input: [3:0]i

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

reg [7:0] D



logic diagram



I/O pin assignment:

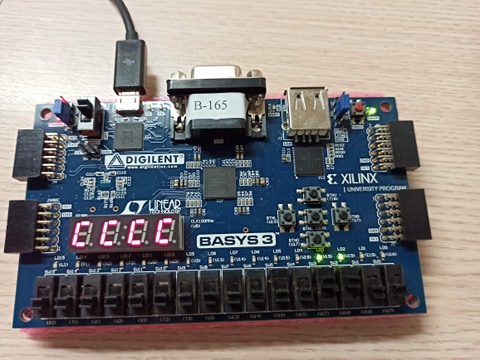
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| D[7] | D[6] | D[5] | D[4] | D[3] | D[2] | D[1] | D[0] |
| W7 | W6 | U8 | V8 | U5 | V5 | U7 | V7 |

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

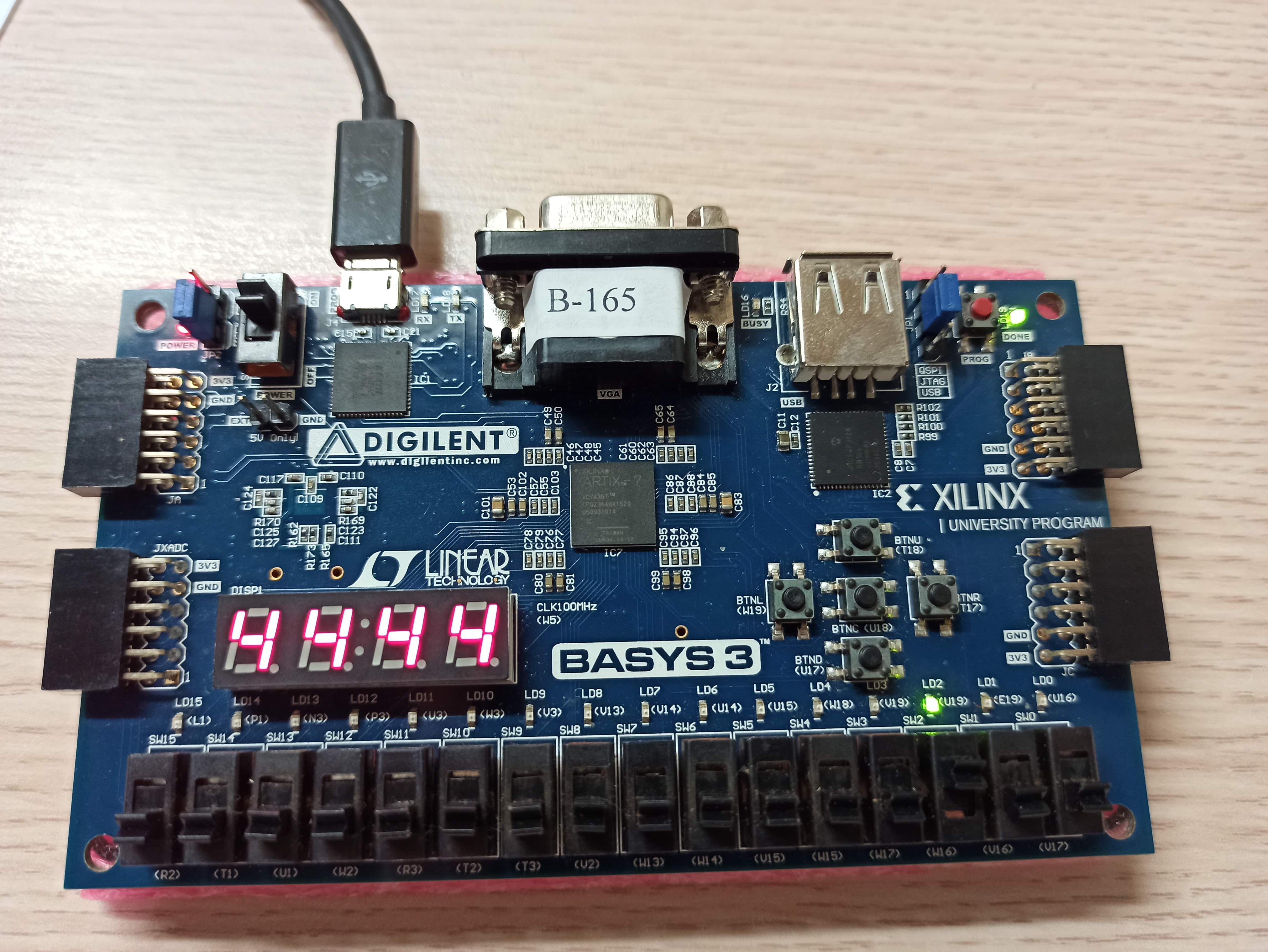
|  |  |  |  |
| --- | --- | --- | --- |
| i[3] | i[2] | i[1] | i[0] |
| W17 | W16 | V16 | V17 |

**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。

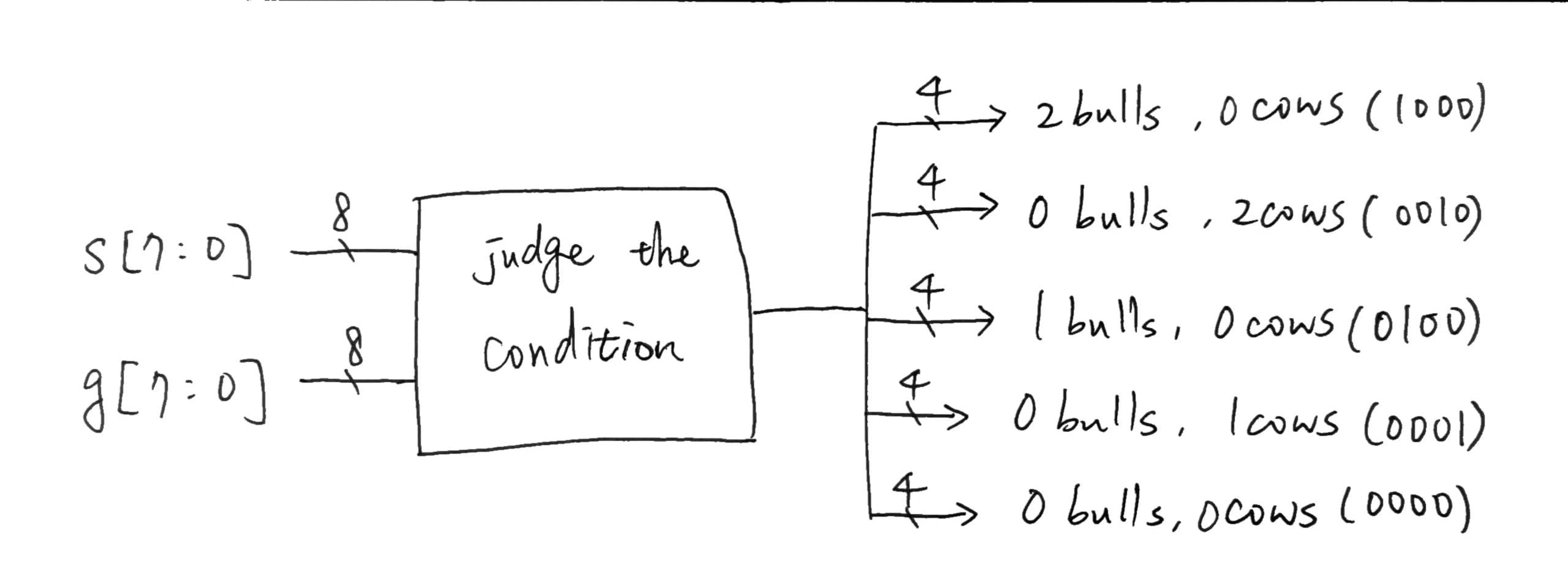
**Bonus. Lab2\_bonus:** Bulls and Cows game.

**Design Specification**

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

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

logic diagram



I/O pin assignment:

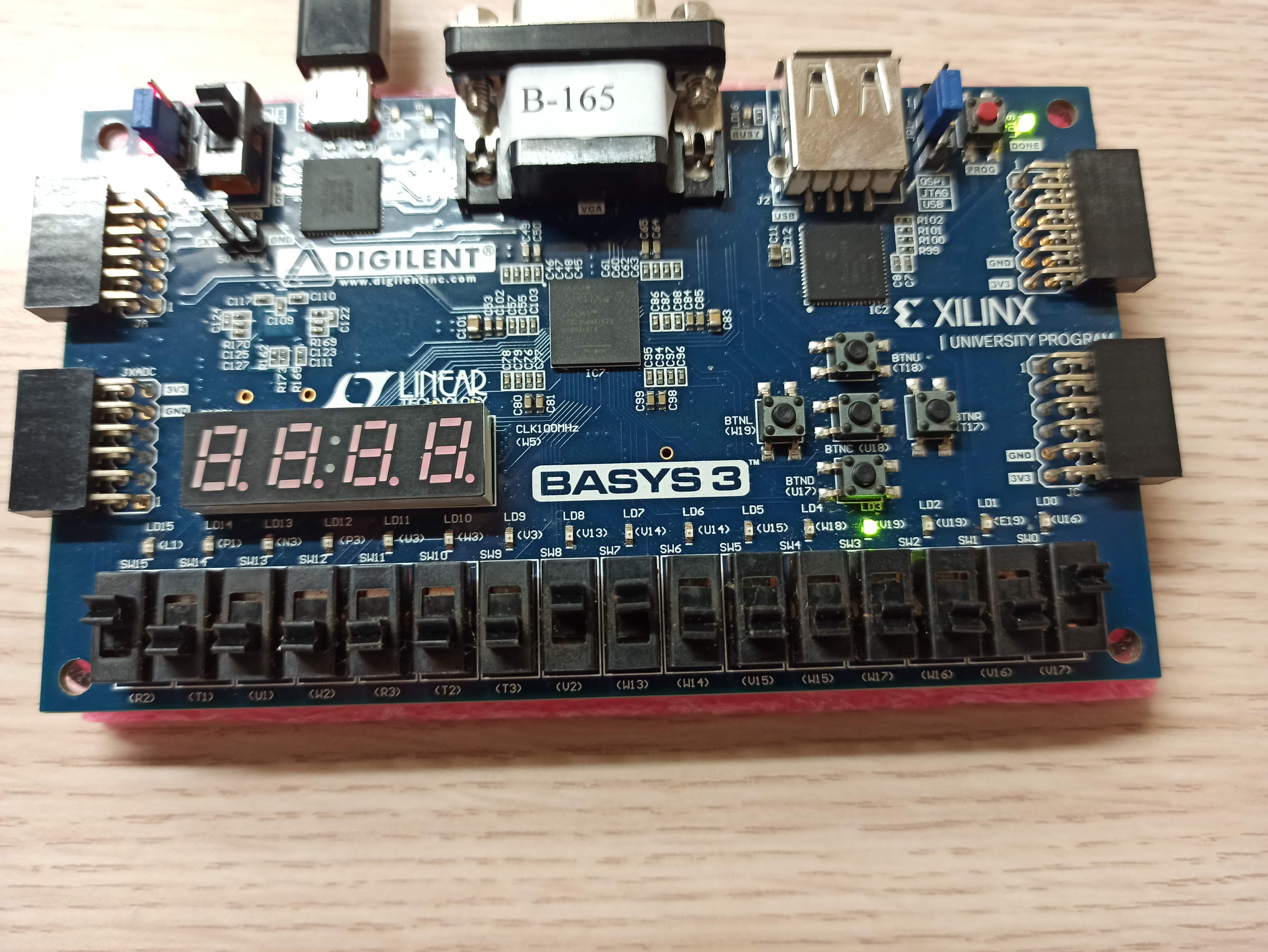
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| s[7] | s[6] | s[5] | s[4] | s[3] | s[2] | s[1] | s[0] |
| W13 | W14 | V15 | W15 | W17 | W16 | V16 | V17 |

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

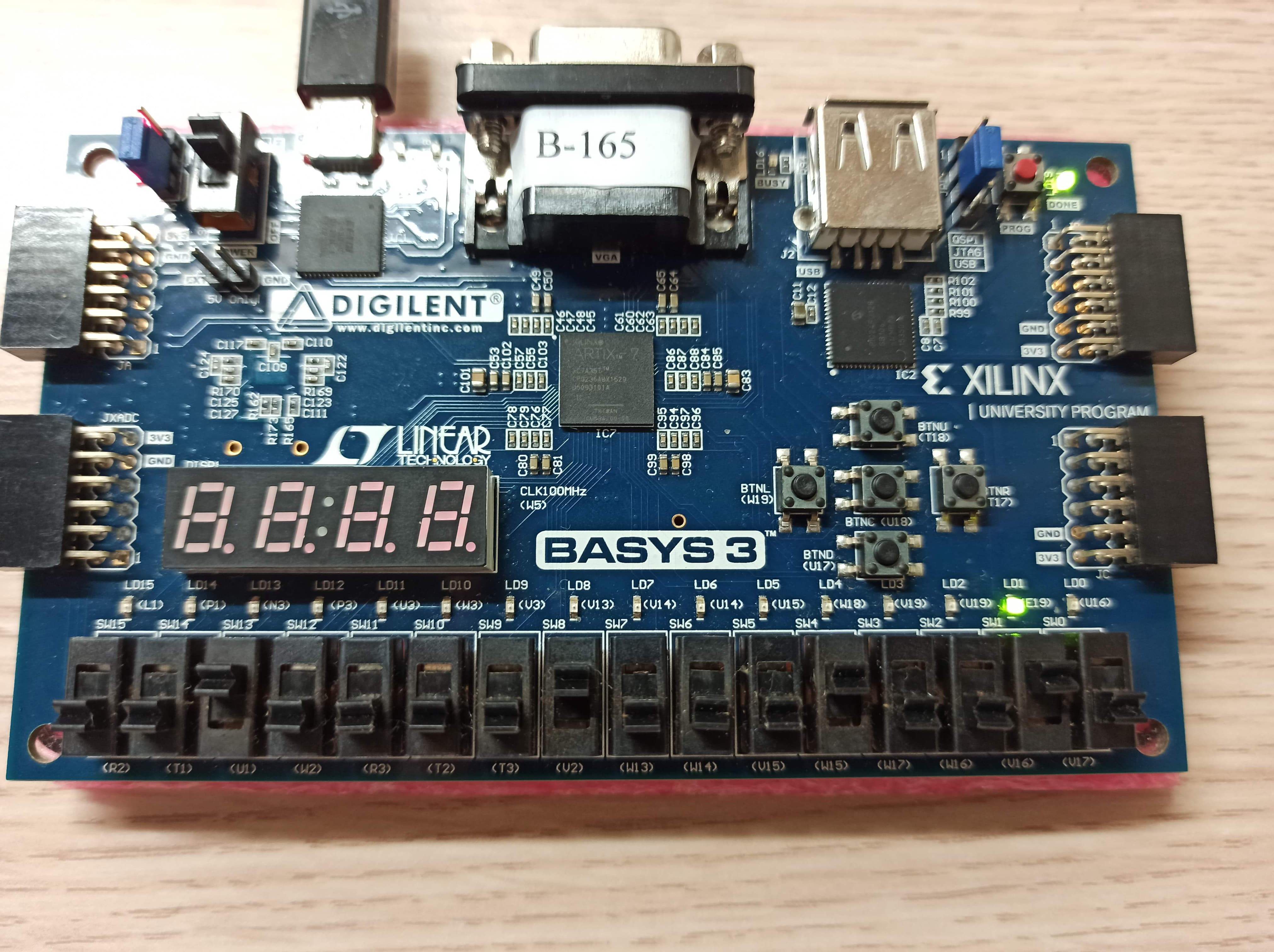
|  |  |  |  |
| --- | --- | --- | --- |
| bulls[1] | bulls[0] | cows[1] | cows[0] |
| V19 | U19 | E19 | U16 |

**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。