**Lab2 Report**



**Ⅰ. Lab2\_1 (BCD to excess 3 code converter)**

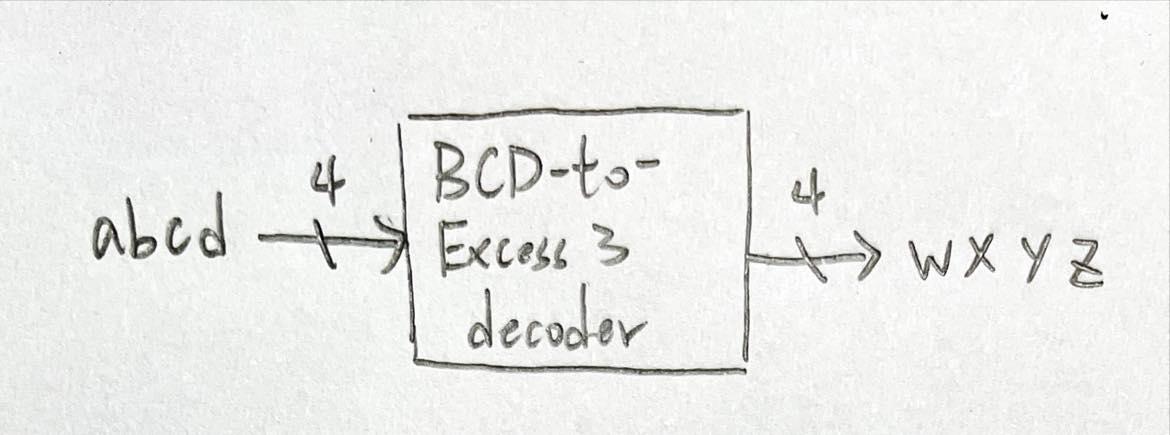
**Design Specification**

IO:

Input: a,b,c,d.

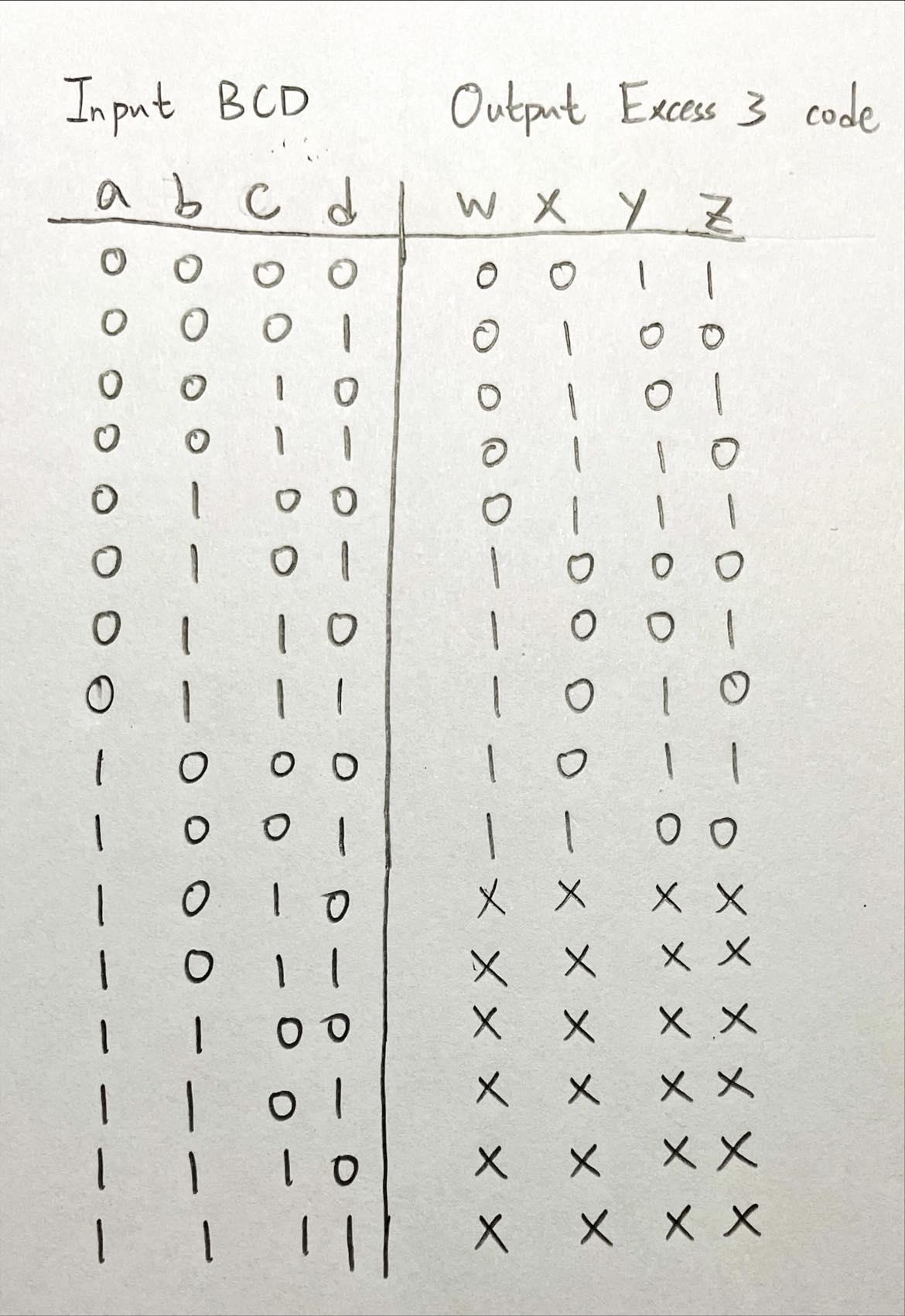
Output: w,x,y,z.

Block diagram:

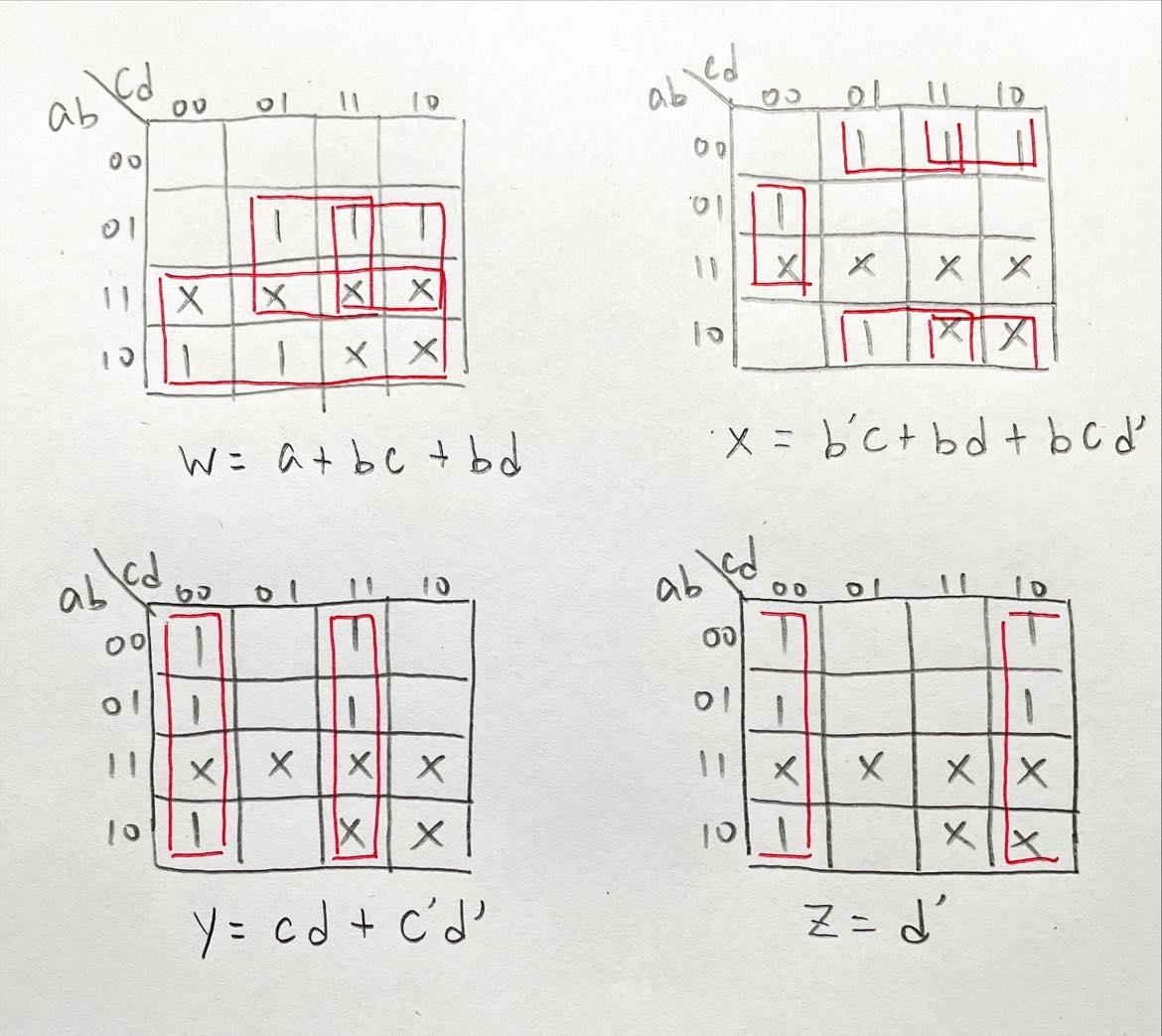


**Design Implementation**

First, draw the truth table of converter as follows:



Then, use K-map to derive logic function:



Boolean function:

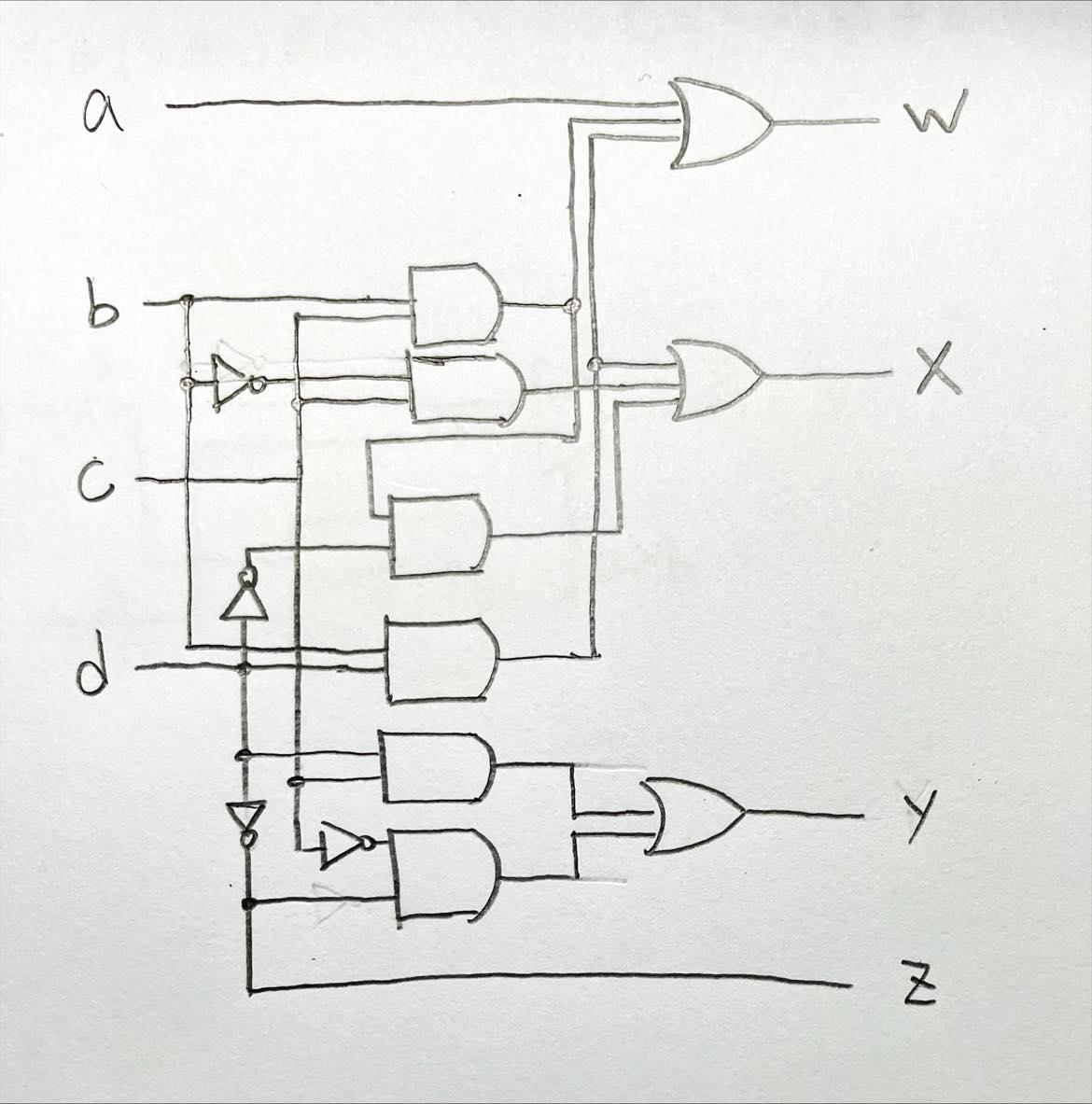
w = a+bc+bd

x = b’c+bd+bcd’

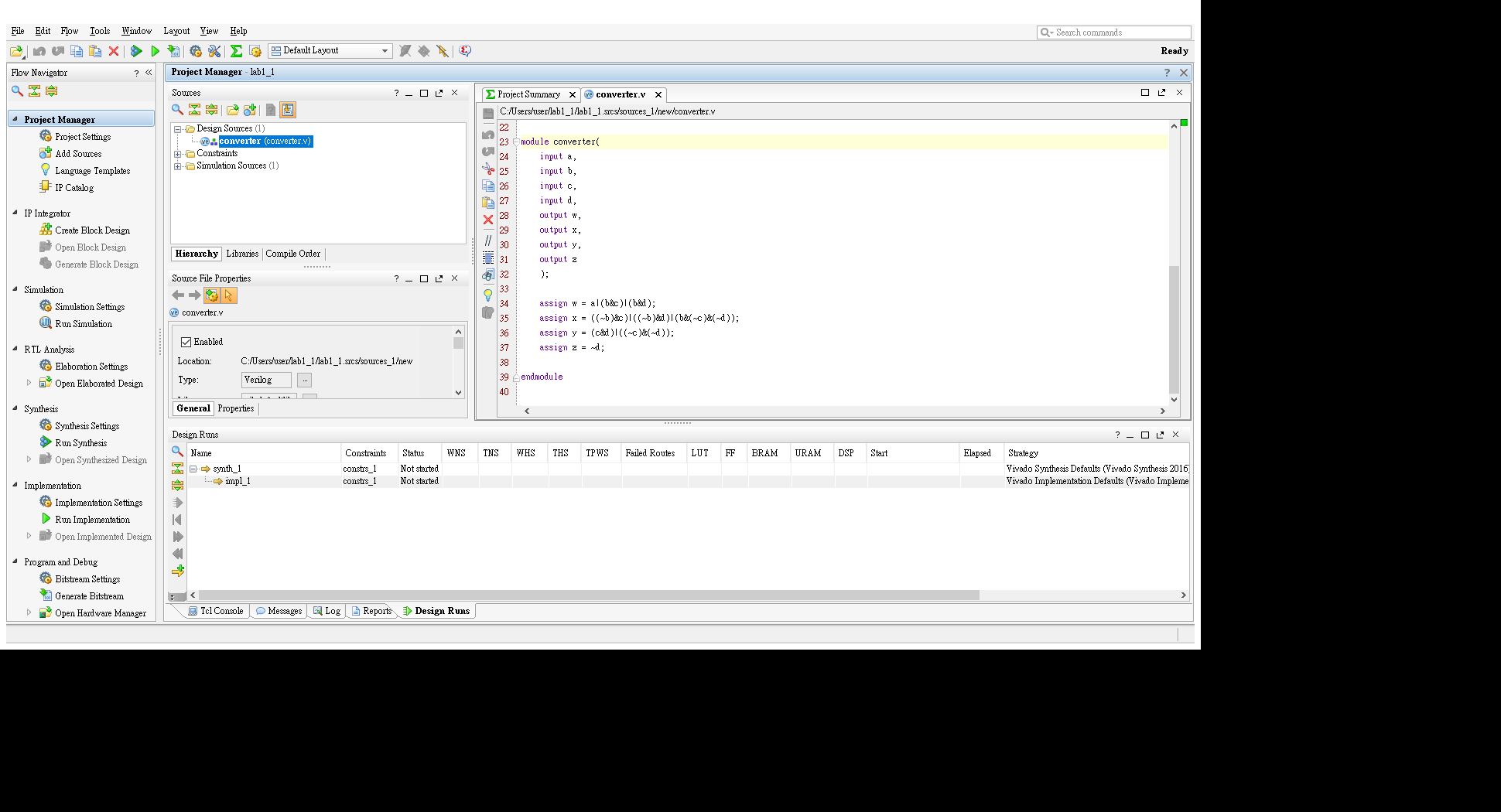
y = cd+c’d’

z = d’

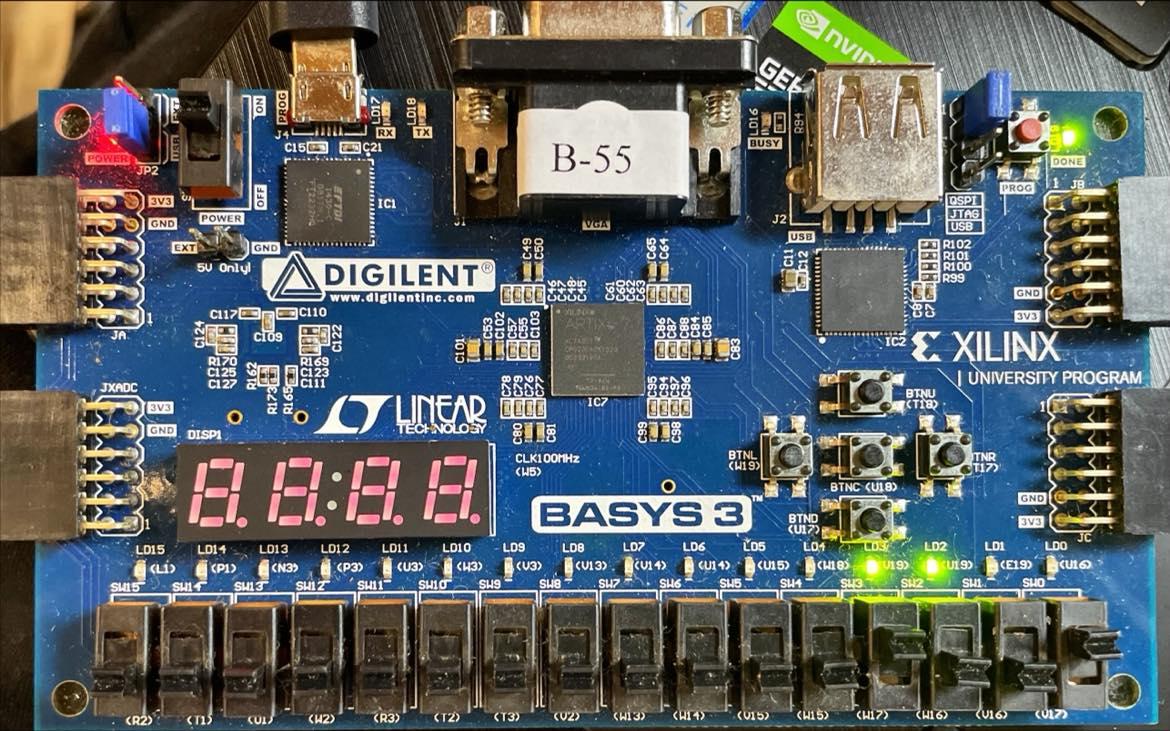
Logic diagram:



Use logic function to construct Verilog RTL code for the converter:



Finally, use parameters from the problem to construct .xdc file



As above picture, 1001 in BCD equals to 1100 in Excess-3-code.

**Ⅱ. Lab2\_2 (7-segment display decoder)**

**Design Specification**

IO:

Input: i[3:0].

Output: D[7:0], d[3:0].

**Design Implementation**

Because of readability, I divide the project into two parts (anode and cathode)

Cathode：

First, I define 8-bit binary number to represent cathode of 0~F.

0：00000011

1：10011111

2：00100101

3：00001101

4：10011001

5：01001001

6：01000001

7：00011111

8：00000001

9：00001001

A：00010001

b：11000001

C：01100011

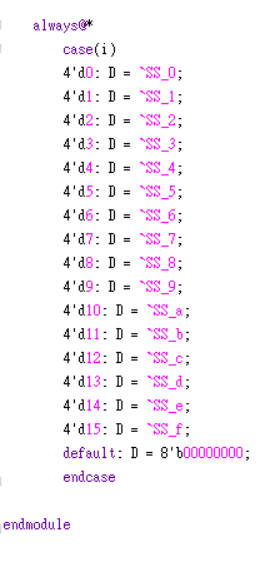
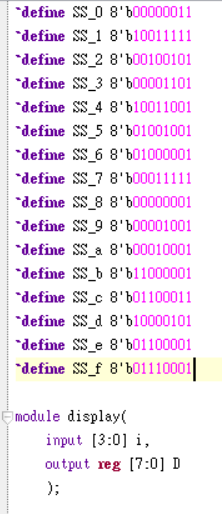
d：10000101

E：01100001

F：01110001

Then, input [3:0] i decide which number or letter should be displayed.

Finally, construct Verilog RTL code for cathode:

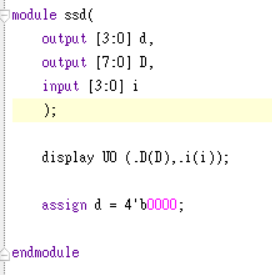


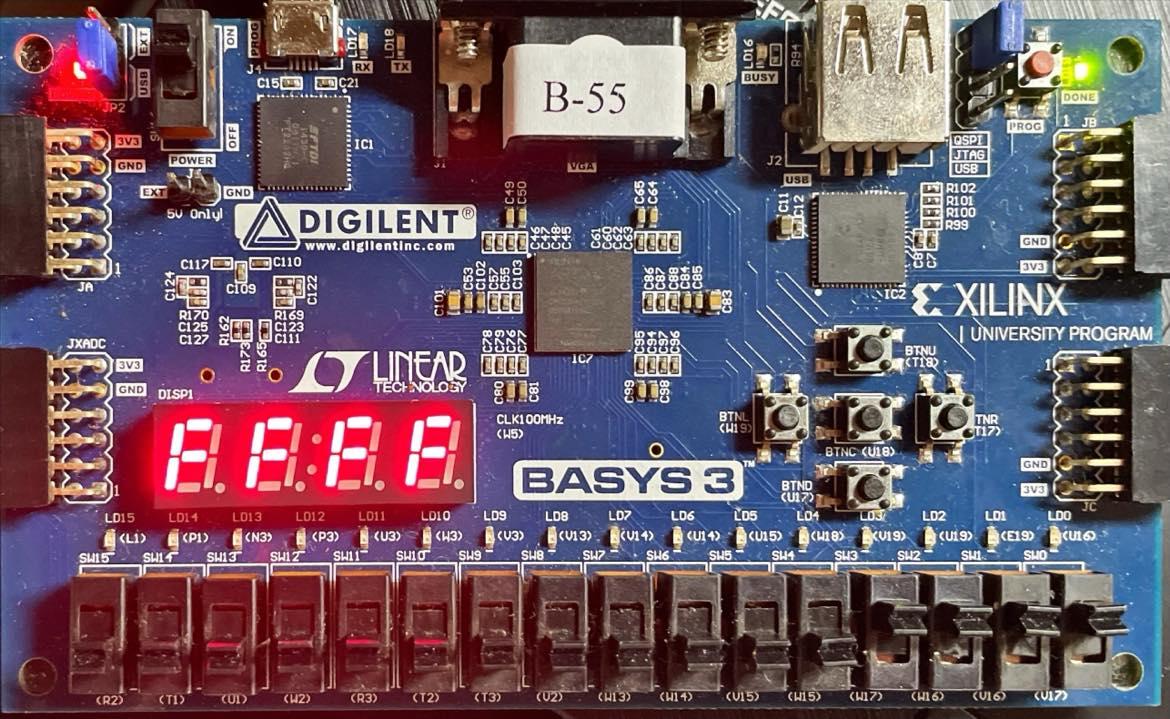
Anode：

Because all 7-segment displays need to be effected, anode should be set 0000.

And I set this module as top module.

Construct Verilog RTL code for anode:





As above picture, binary number 1111 equals to hexadecimal number F.

**Ⅲ. Lab2\_bonus(Bulls and Cows)**

**Design Specification**

IO:

Input: [3:0] A1 (for first secret number), [3:0] A2 (for second secret number)

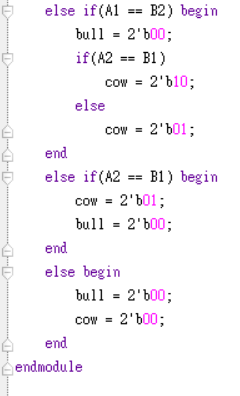
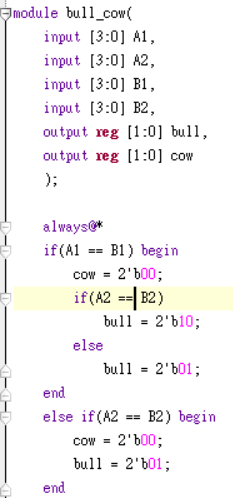
[3:0] B1 (for first guess number), [3:0] B2 (for second guess number).

Output: [1:0] bull, [1:0] cow

**Design Implementation**

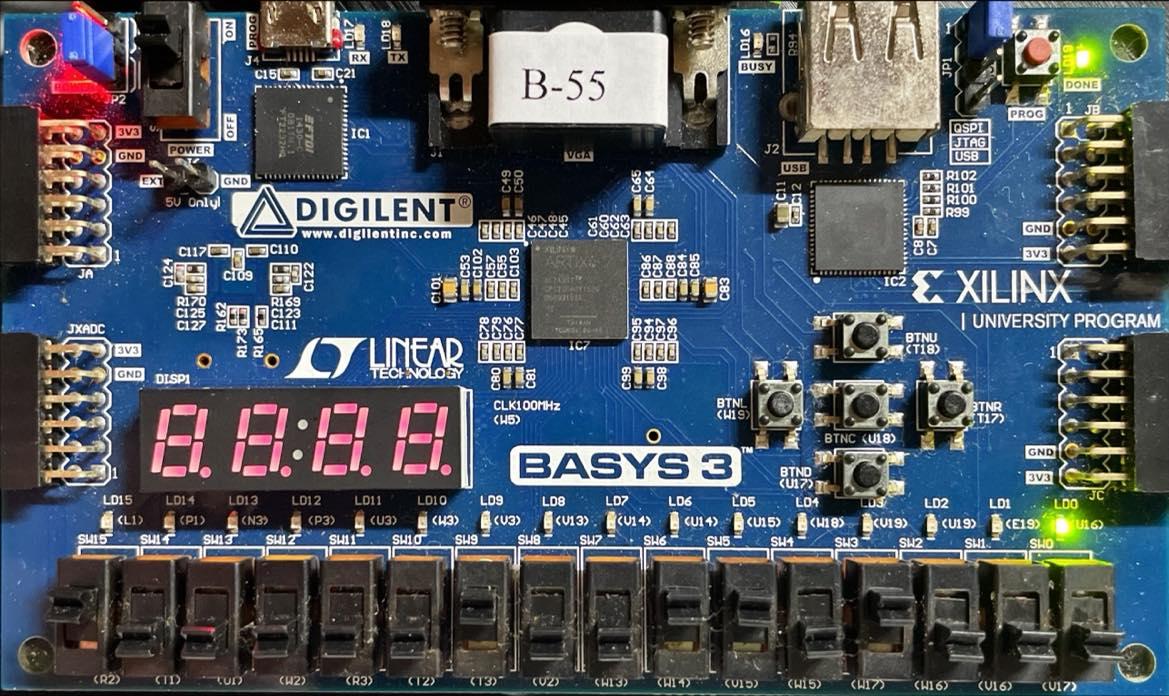
Because the rule of problem (two numbers are not the same), having a bull represents no cow or having a cow represents no bull.

I use if…else (MUX) to compare if two numbers are the same as follows:



Finally, set I/O Pins according to the problem

(16 DIP switches as the BCD inputs, 4 LEDs to show the number of bulls and cows)



As above picture, secret numbers are 82, guess numbers are 68, so answer is a cow.

**Discussion**

**Lab2\_1:**

本次實驗與lab1\_1是相同的題目，差別只在於需要在FPGA板上呈現結果，所以本題的目的應該是讓我們熟悉設定I/O Pins和燒進FPGA板的流程。

**Lab2\_2:**

本實驗是要把二進制的數字轉乘十六進制，並實現在七段顯示器上，在設計的過程就是不斷的賦值，且為了可讀性，我把陰極和陽極分開來寫，而我也在本次實驗中看懂了一些error message(例如忘記設定3.3V)，雖然這格實驗不難，但我也從中學到了一些知識。

**Lab2\_bonus:**

本實驗是個應用題，也就是小時候常玩的幾A幾B的小遊戲，做起來是不太難，主要還是用if...else去判斷，唯一要注意的就是在always裡的output要記得加reg的資料型態，因為我常常因為忘了加而出錯。

**Conclusion**

這次實驗與以前最不同的地方就是接上了電路板，雖然跑電路板需要消耗一段時間，但是相比之下卻變得不像波形圖一樣那麼枯燥乏味，也有了實際操作的真實感。這次的題目並不複雜，是熟悉FPGA的良好機會，也終於看到上學期的理論基礎慢慢的實體化，做起來相當有成就感。