Lab 02

一.Lab02-1:Full Adder

1. Design Specification

   Input : x, y, cin

   Output : s, cout

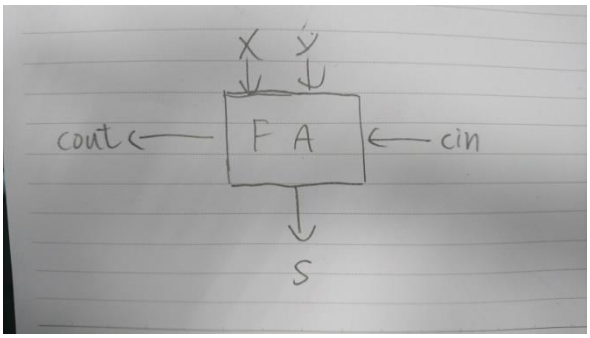

圖 1 Lab02-1 區塊圖

2. Design Implementation
   a. Logic function :
      (1) s = x⊕y⊕cin
      (2) cout = xy + cin(x⊕y)
   b. Logic diagram :

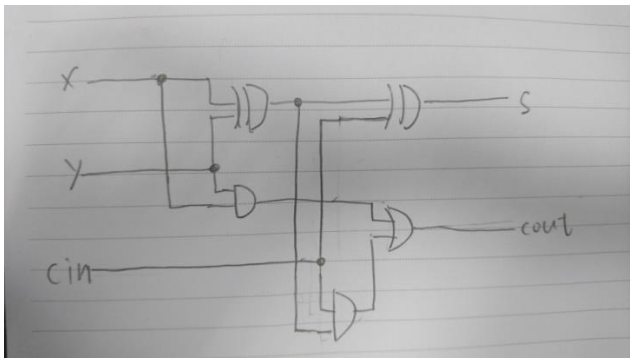

圖 2 Lab02-1 邏輯圖

   c. Pin assignment :
      (1) Input :
         A. x = V17
         B. y = V16
         C. cin = W16
      (2) Output :
         A. s = U16
         B. cout = E19

3. Discussion

(1) 思考過程

code 參照 Lab01-1，再按照題目將 I/O 接上板子上指定的位置。

(2) 過程中的 bug

一開始直接在 xdc 檔裡面打 I/O 和板子如何連接，常常會少打一兩個字母或是其他的錯誤，跑 Synthesis 時就會失敗。之後在程式的 Window 裡找到 I/O Port 的選項，可以直接叫出一個視窗，可以直接選擇 I/O 要接哪個位置，不用在一個一個打。

二. Lab02-2: Seven-Segment Display Decoder

1. Design Specification

Input : [3:0] i 4-bit BCD number，和 4 個 Switches 連接。

Output : [3:0]d 這 4-bit 是和 LED 連接，來表示上面的 4-bit 的各 bit 是 0 或 1，0 的話不亮，1 則亮。

[7:0] D_ssd 和板子上的七段顯示器連接，隨著[3:0] i 不同而改變。

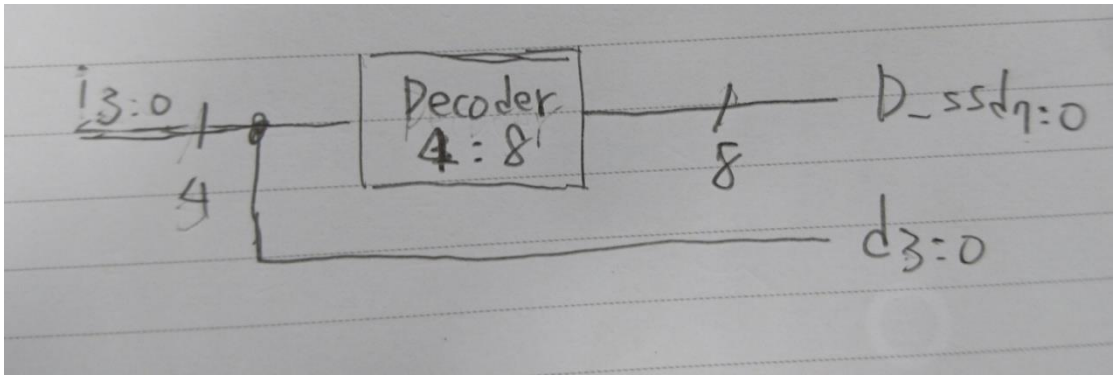

圖 3 Lab02-2 的區塊圖

2. Design Implementation

a. Logic function :

藉由寫出 truth table 和畫 K-map 來得到 Logic function

(1) D_ssd :

$D\_ssd[7] = i_3'i_2'i_1'i_0 + i_3'i_2i_1'i_0'$

$D\_ssd[6] = i_3i_2 + i_3i_1 + i_3i_0 + i_2i_1'i_0 + i_2i_1i_0'$

$D\_ssd[5] = i_3i_2 + i_3i_0 + i_2'i_1i_0'$

$D\_ssd[4] = i_3i_2 + i_3i_1 + i_2'i_1'i_0 + i_2i_1'i_0' + i_2i_1i_0$

$D\_ssd[3] = i_3i_2i_1' + i_3'i_0$

$D\_ssd[2] = i_3'i_2'i_0 + i_3'i_2'i_1 + i_3'i_1i_0$

$D\_ssd[1] = i_3'i_2'i_1' + i_3'i_2i_1i_0'$

$D\_ssd[0] = 1$

(2) d :

d[3] = i_3

d[2] = i_2

d[1] = i_1

d[0] = i_0

b. Pin assignment :

(1) Input :

A. i :

i[3] = W17

i[1] = W16

i[2] = V17

i[0] = V16

(2) Output :

A. D_ssd :

D_ssd[7] = W7

D_ssd[7] = W6

D_ssd[7] = U8

D_ssd[7] = V8

D_ssd[7] = U5

D_ssd[7] = V5

D_ssd[7] = U7

D_ssd[7] = V7

B. d :

d[3] = V19

d[2] = U19

d[1] = E19

d[0] = U16

3. Discussion

(1) 思考過程 ：

Output 裡的 d 是顯示 Input 的 i，因此只需要寫 d = i 就可以。而 D_ssd 是七段顯示器，雖然是利用 Decoder 的原理，但並不是上學期所教的 One-hot 的簡單 Decoder，所以 funtion 更複雜，透過 Truth table 和 K-map 需要耗時許久。不過可以透過 Verilog 裡 case 的語法，將不同的 Input 直接對應到不同的 Ouput，不必透過 AND/OR Gates 來達成，降低複雜度。

(2) 過程中的 bug :

一開始把 D_ssd 和板子連接時，沒有注意到順序，本來應該從 D_ssd[7] 到 D_ssd[0]，卻輸入成從 D_ssd[0]到 D_ssd[7]，結果顯示得亂七八糟，之後重頭看了下 code 就發現問題，很快就解決了。

三.Conclusion

終於可以不用再看波形圖了，過去只能看波形圖感覺很沒有實感也很空虛，不過這次看到自己打的 code 讓板子運作就很有成就感，而且還學到如何使用 case，可以減少許多打 code 的時間。

四.References

1. 大一上的邏輯設計講義
   讓我知道 Decoder 的原理。
2. 實驗講義
   如何將 Verilog code 和板子連接。