Lab2-1 Emulate lab1.1 (a full adder) to 4 LED outputs

1. Design Specification

功能: 將三個數入的一二進位數相加後輸出。

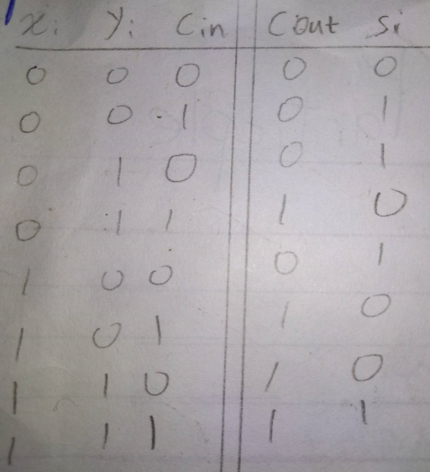
輸入: x, y, cin (用來輸入二進位數)

輸出: s, cout (輸出結果，s為第一位數，cout為第二位數)

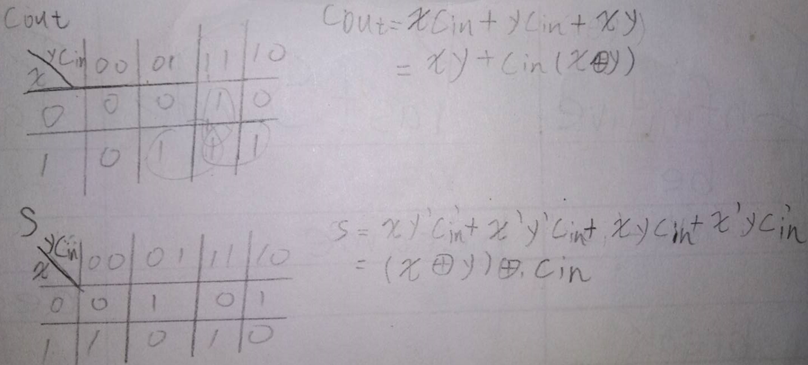
邏輯函式: s = (x ^ y) ^ cin ; cout = ( x + y ) | (cin \* ( x + y))

2. Design Implementation

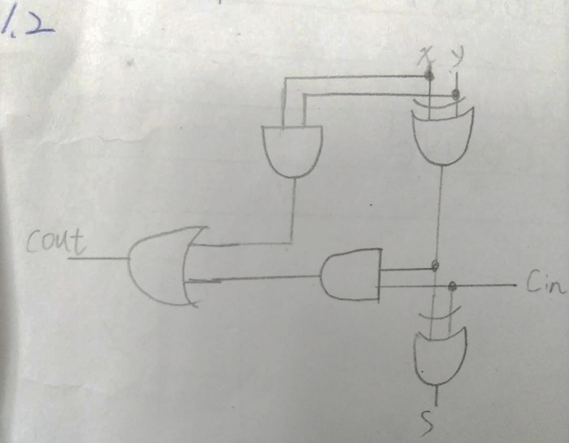
在設計過程中，我使用最基本的方法，也就是先寫出真值表(如下圖)



然後再使用K-map將算式列出來進行簡化。



最後再將其化成logic diagram，因為輸出的形式和LED要求的一樣，所以不用另行處理。



3. I/O assignment

輸入: x 接至 V17 腳位(右下第一個開關)

y 接至 V16 腳位(右下第二個開關)

cin 接至 W16 腳位(右下第三個開關)

輸出: s 接至 U16 腳位(右下第一個開關上LED)

cout 接至 E19 腳位(右下第二個開關LED)

Lab2-2 Derive a BCD to 7-segment display decoder, and also use four LEDs to monitor the 4-bit BCD number.

1. Design Specification

功能:輸入一位BCD，而將相對應的十進位數顯示於七段顯示器及LED上。

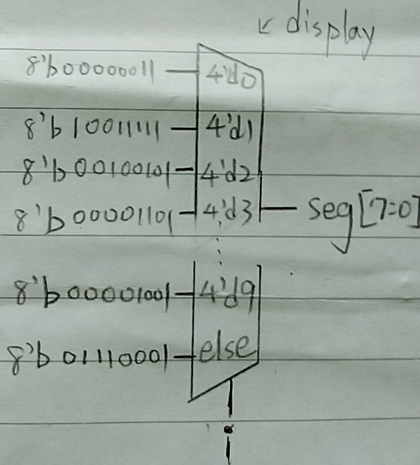
輸入: i[3:0] (輸入一位BCD)

輸出: D\_ssd[7:0] (給予七段顯示器訊號)

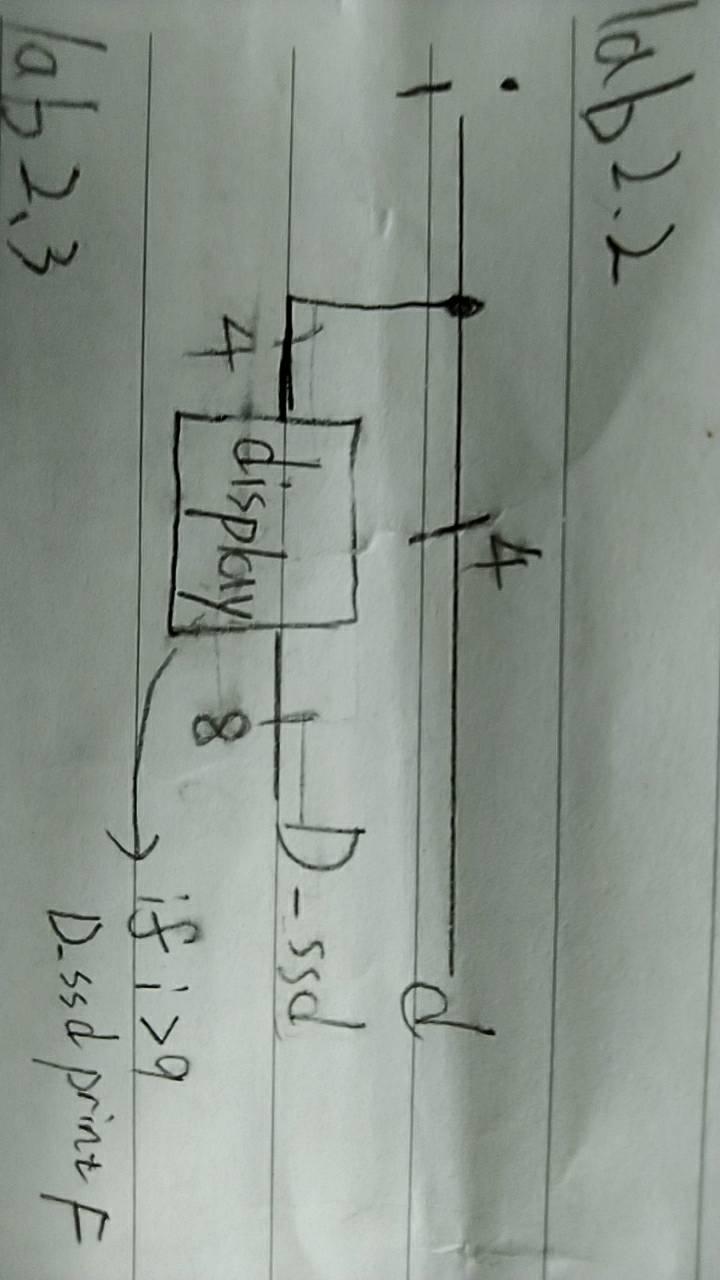
d[3:0] (給予LED訊號)

2. Design Implementation

首先控制七段顯示器的每個數字顯示有八條輸出，而每一個輸入的BCD都要選擇不同的訊號線才可顯示。所以在這個時候，就需要一個多工器，輸入BCD來選擇對應的輸出。而大於九的部分就輸出F的樣子。



而輸入的二進位數形式由於和LED要求的一樣，所以不用另行處理，再設一個net輸出變數就好。以下為block diagram。



3. I/O assignment

輸入: i[3] 接至 W17 腳位(右下第四個開關)

i[2] 接至 W16 腳位(右下第三個開關)

i[1] 接至 V16 腳位(右下第二個開關)

i[0] 接至 V17 腳位(右下第一個開關)

輸出: D\_ssd[7] 接至 W7 腳位(於顯示器上)

D\_ssd[6] 接至 W6 腳位(於顯示器上)

D\_ssd[5] 接至 U8 腳位(於顯示器上)

D\_ssd[4] 接至 V8 腳位(於顯示器上)

D\_ssd[3] 接至 U5 腳位(於顯示器上)

D\_ssd[2] 接至 V5 腳位(於顯示器上)

D\_ssd[1] 接至 U7 腳位(於顯示器上)

D\_ssd[0] 接至 V7 腳位(於顯示器上)

d[3] 接至 V19 腳位(右下第四個開關LED)

d[2] 接至 U19 腳位(右下第三個開關LED)

d[1] 接至 E19 腳位(右下第二個開關LED)

d[0] 接至 U16 腳位(右下第一個開關LED)

Lab2-3 Derive a binary to 7-segment display decoder, and also use four LEDs to monitor the 4-bit binary number.

1. Design Specification

功能:輸入binary，並將相對應的十六進位數顯示於七段顯示器及LED上。

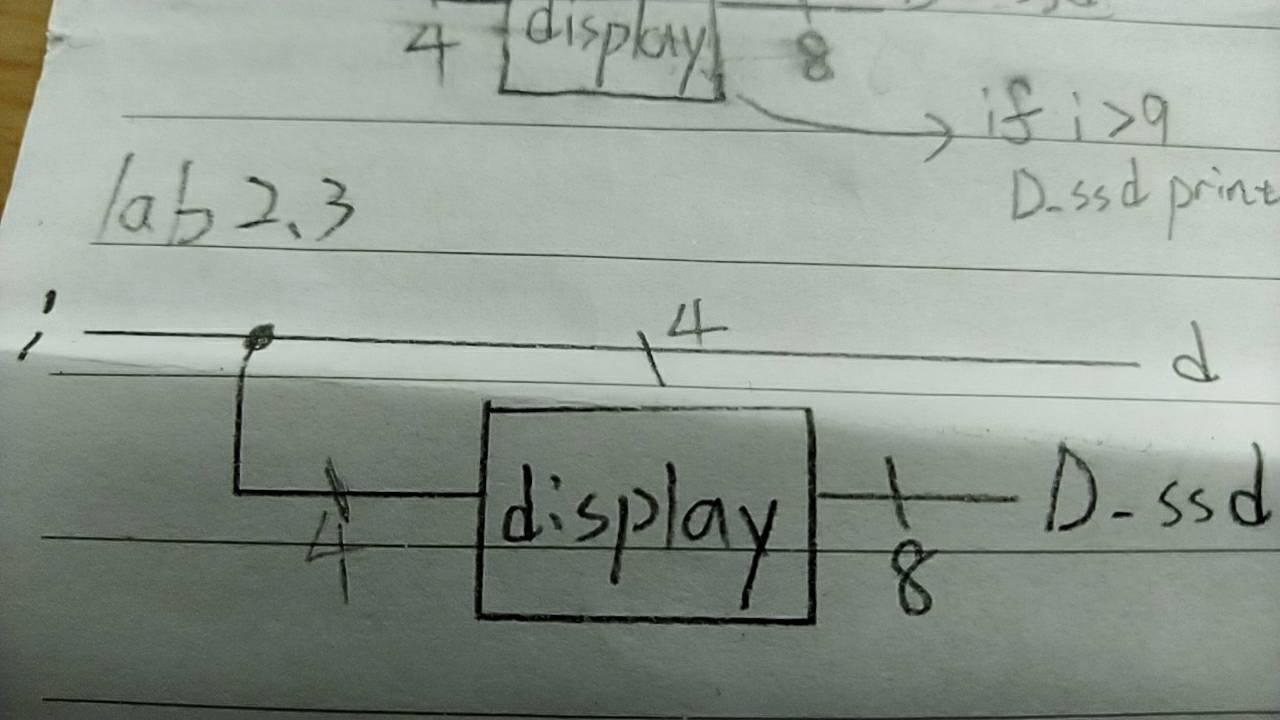
輸入: i[3:0] (輸入binary)

輸出: D\_ssd[7:0] (給予七段顯示器訊號)

d[3:0] (給予LED訊號)

2. Design Implementation

主要的結構都和前一題差不多，唯一的不同是這一題輸入大於九的時候會出現A,b,C,d,E,F，其中b和d為了辨識方便，會輸出小寫。而不是只出現F。



3. I/O assignment

同題Lab 3.2

Lab2-4 Design a combinational circuit that compares two 4-bit unsigned numbers A and B to see whether A is greater than B. The circuit has one output X such that X = 0 if A ≤ B and X = 1 if A > B. (also need to monitor

A and B on LED)

1. Design Specification

功能:輸入兩個數字A及B，比大小，如A ≤ B則輸出0，反之則輸出1

輸入: A[3:0], B[3:0] (輸入四位二進位數字)

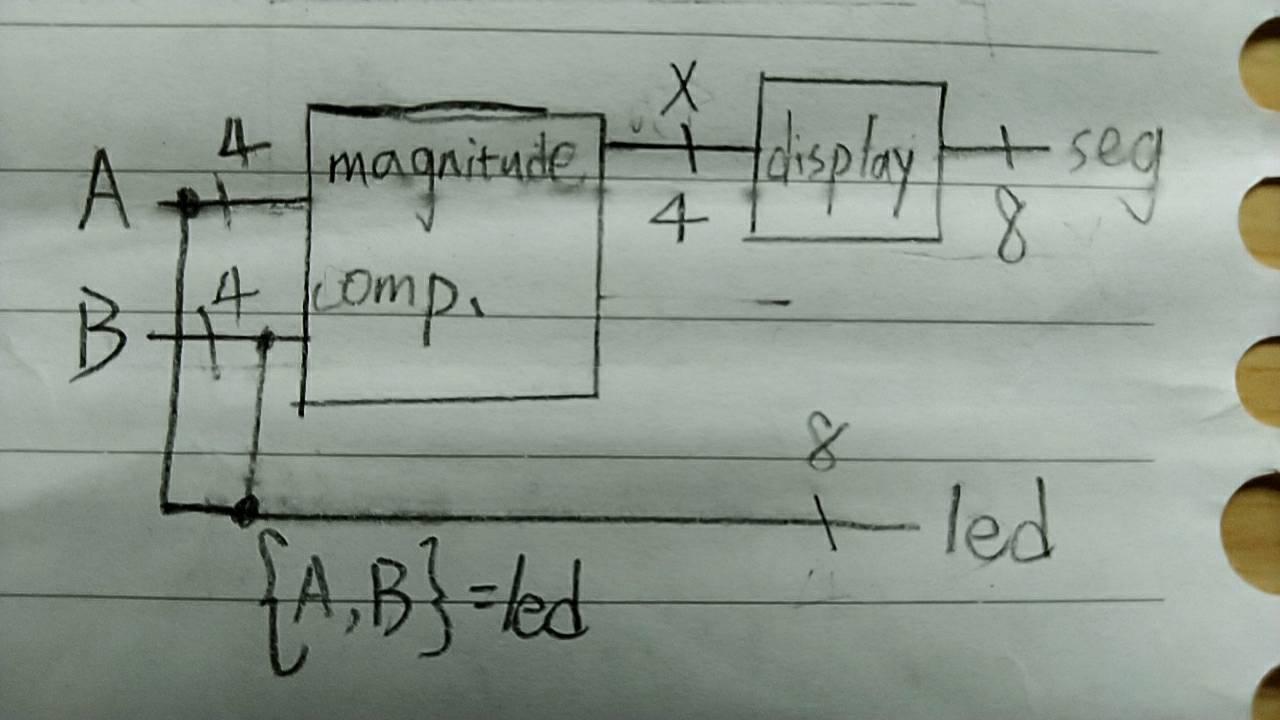
輸出: seg[7:0] (給予七段顯示器訊號)

led[7:0] (給予LED訊號)

2. Design Implementation

第一步，要將兩個數字做比較，比較的部分就由magnitude comparator block所代表。輸入兩個四位元數字A及B，比大小，如A ≤ B則輸出0，反之則輸出1。然後將其所得，傳到display block中將其由七段顯示器顯示出來。

至於LED部分則設net點[7:0]led，將其統整後輸出。



3. I/O assignment

輸入: A[3] 接至 W13 腳位(右下第八個開關)

A[2] 接至 W14 腳位(右下第七個開關)

A[1] 接至 V15 腳位(右下第六個開關)

A[0] 接至 W15 腳位(右下第五個開關)

B[3] 接至 W17 腳位(右下第四個開關)

B[2] 接至 W16 腳位(右下第三個開關)

B[1] 接至 V16 腳位(右下第二個開關)

B[0] 接至 V17 腳位(右下第一個開關)

輸出: seg[7] 接至 W7 腳位(於顯示器上)

seg[6] 接至 W6 腳位(於顯示器上)

seg[5] 接至 U8 腳位(於顯示器上)

seg[4] 接至 V8 腳位(於顯示器上)

seg[3] 接至 U5 腳位(於顯示器上)

seg[2] 接至 V5 腳位(於顯示器上)

seg[1] 接至 U7 腳位(於顯示器上)

seg[0] 接至 V7 腳位(於顯示器上)

led[7] 接至 V14 腳位(右下第八個開關LED)

led[6] 接至 U14 腳位(右下第七個開關LED)

led[5] 接至 U15 腳位(右下第六個開關LED)

led[4] 接至 W18 腳位(右下第五個開關LED)

led[3] 接至 V19 腳位(右下第四個開關LED)

led[2] 接至 U19 腳位(右下第三個開關LED)

led[1] 接至 E19 腳位(右下第二個開關LED)

led[0] 接至 U16 腳位(右下第一個開關LED)

Conclusion :

很高興終於可以將上學期的所學展現，這是半年來第一次可以看到自己的邏設實驗終於清清楚楚的在晶片上顯示，雖然只是最基本的輸入顯示出來，但也終於自己開始感覺像個硬體工程師了。