微算機系統

實驗二

組別： 14
班級、姓名與學號：四資二　洪晟毅　104590048
四資二　梁皓鈞　104360098

日期： 2016.10.18

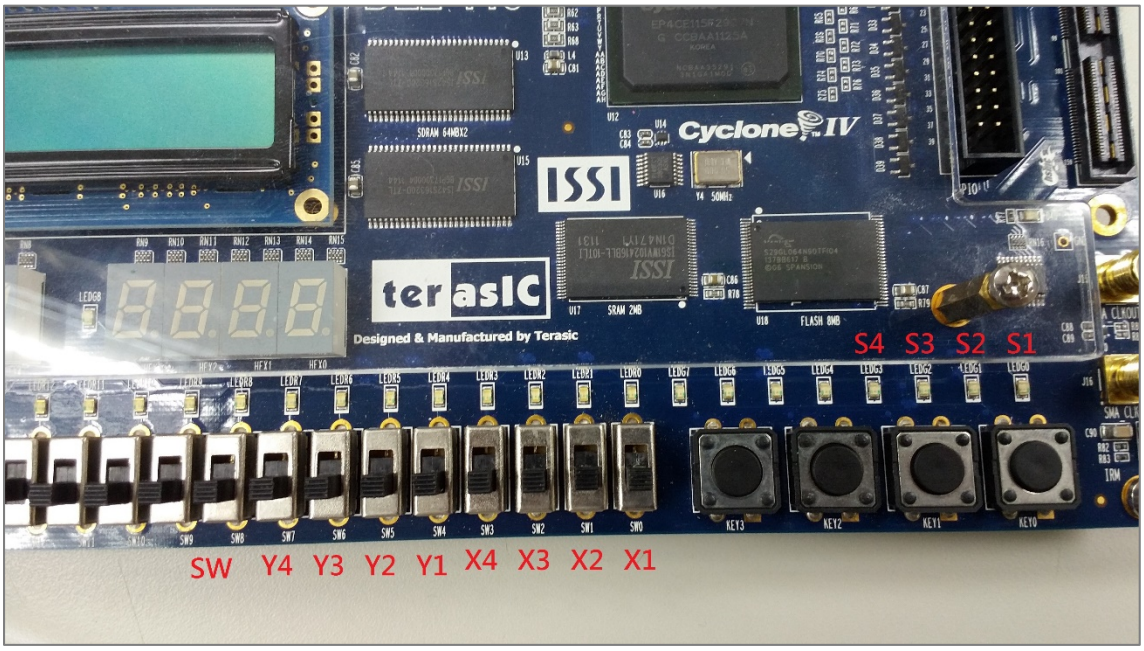
1. 實驗內容：

以1-bit 的全加器作為基礎，僅使用 VHDL 的邏輯運算子實作4-bits 的漣波進位加法器，加分題為在相同電路上實作加法器及減法器，並新增1個 Switch 開關，能隨時切換加法與減法功能。

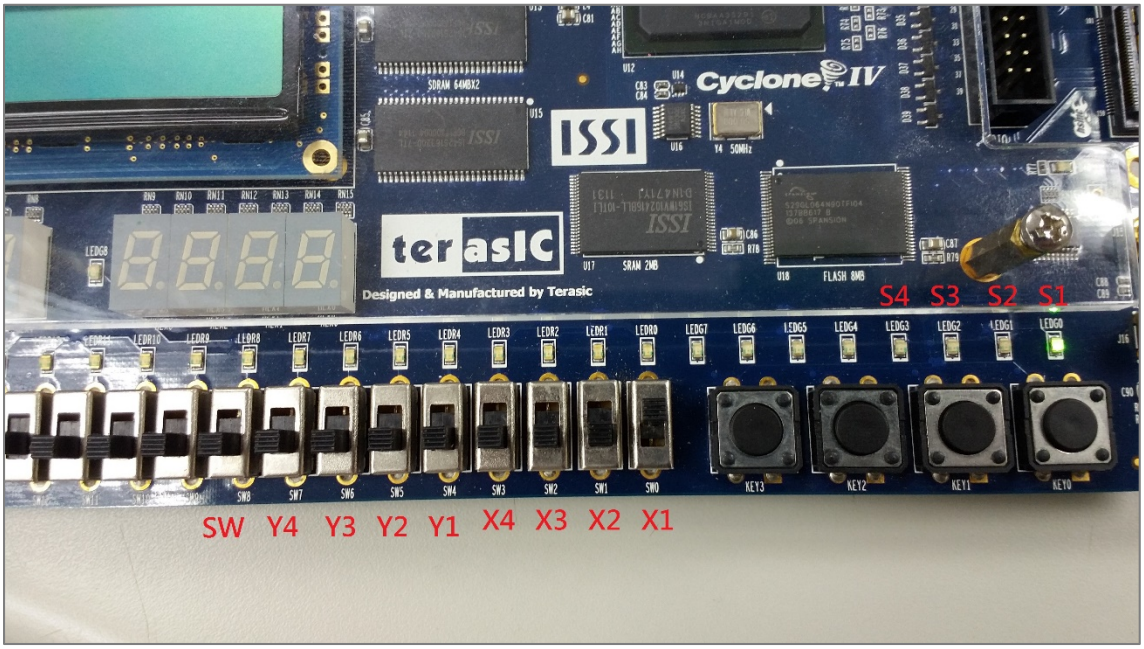
2. 實驗過程及結果：

最初由皓鈞很快就撰寫出加法器的 VHDL 程式碼，但因為皓鈞對減法器所使用的2的補數系統不熟悉，導致遲遲無法寫出加分題指定的減法器，晟毅對數字系統則相對熟悉許多，於是就由晟毅重新規劃一下程式的架構，讓皓鈞簡單修改原先加法器的程式碼，順利完成包含加分題要求的所有功能。

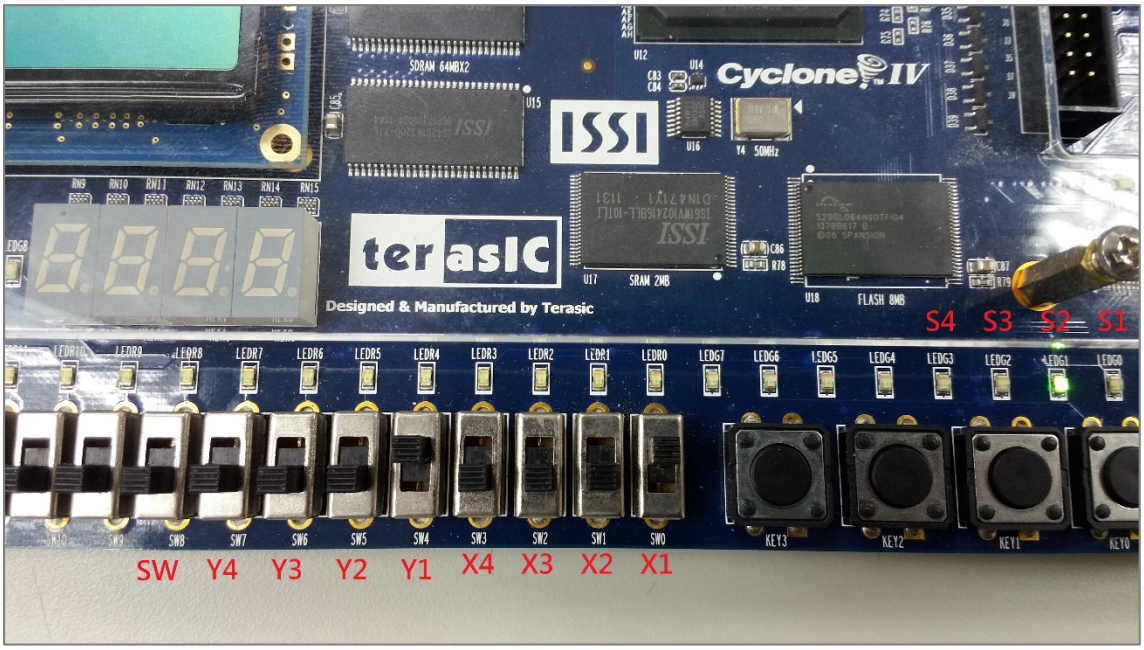
a. 最初 SW=0代表加法，輸入 X、Y 皆為0000_2，因此輸出 S 也為0000_2



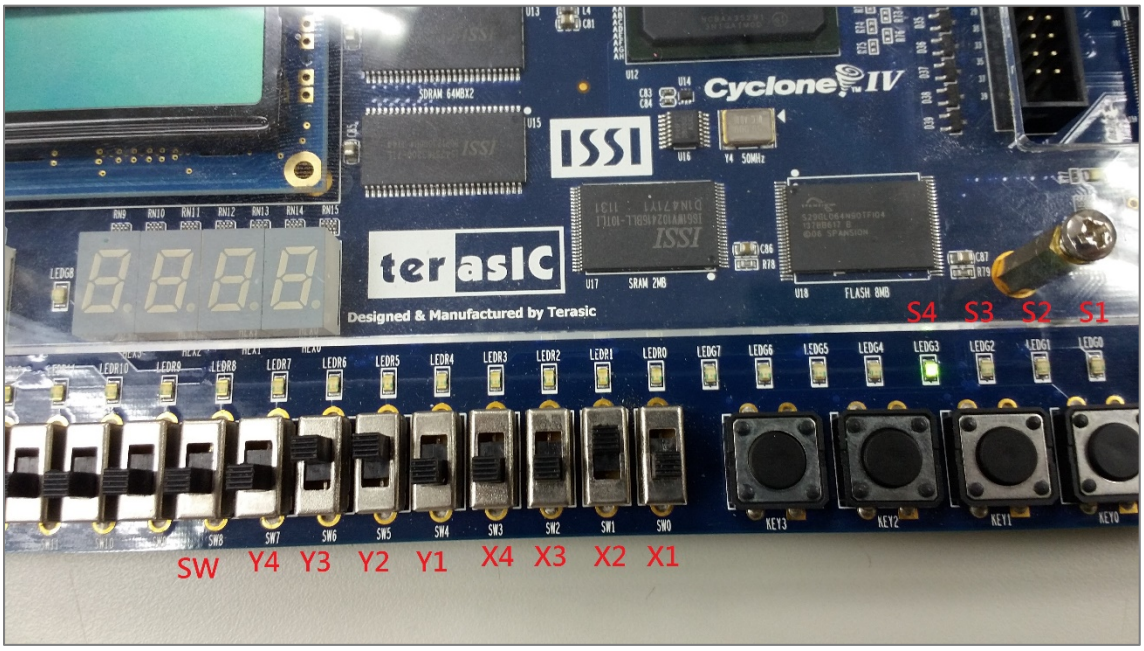
b. 輸入 X=0001_2，Y=0000_2，可以正確得到 S=0001_2



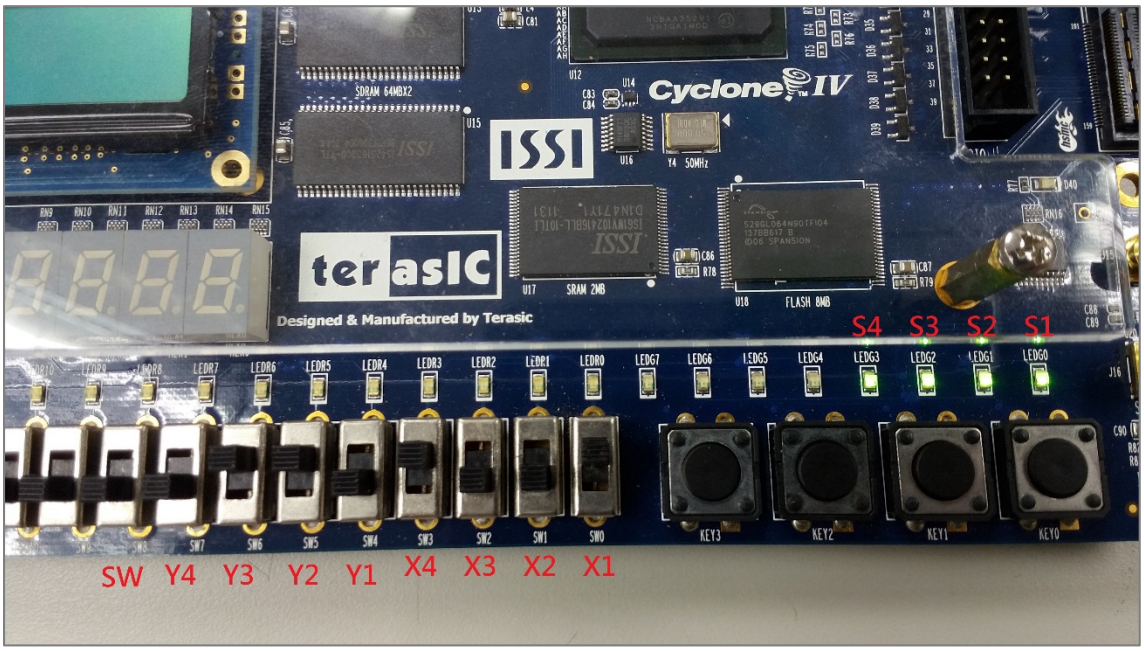
c. 再修改 Y=0001_2，就會得到 S=0010_2 進位後的結果。



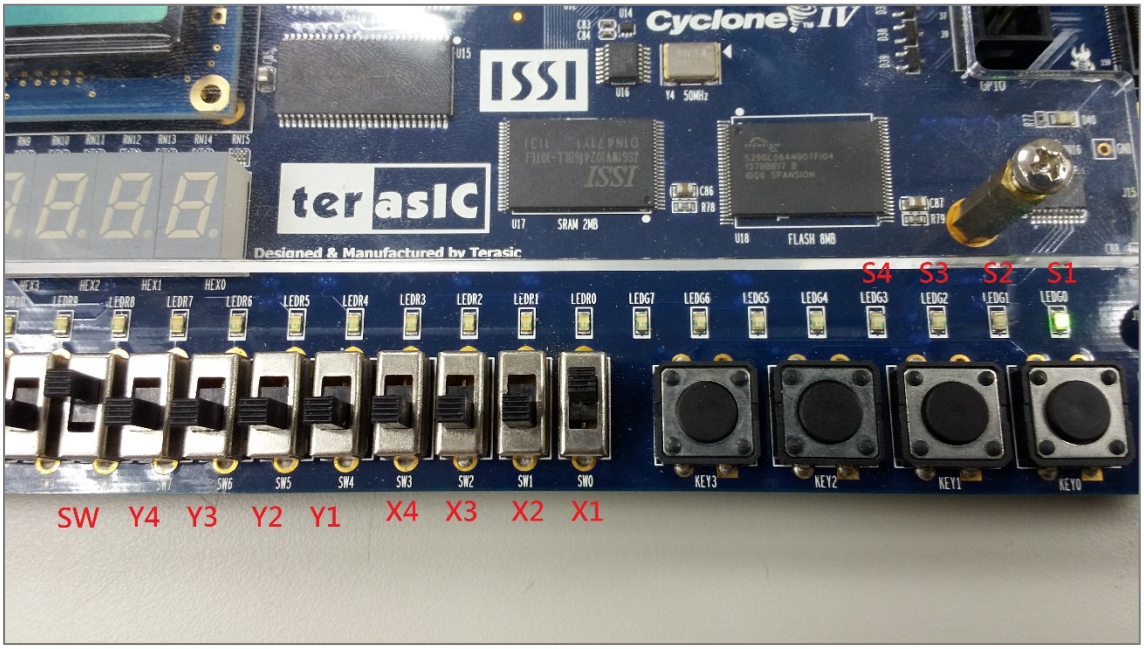
d. 測試一些特殊的數字，X=0010_2 (2)， Y=0110_2 (6)，會得到 S=1000_2 (8)



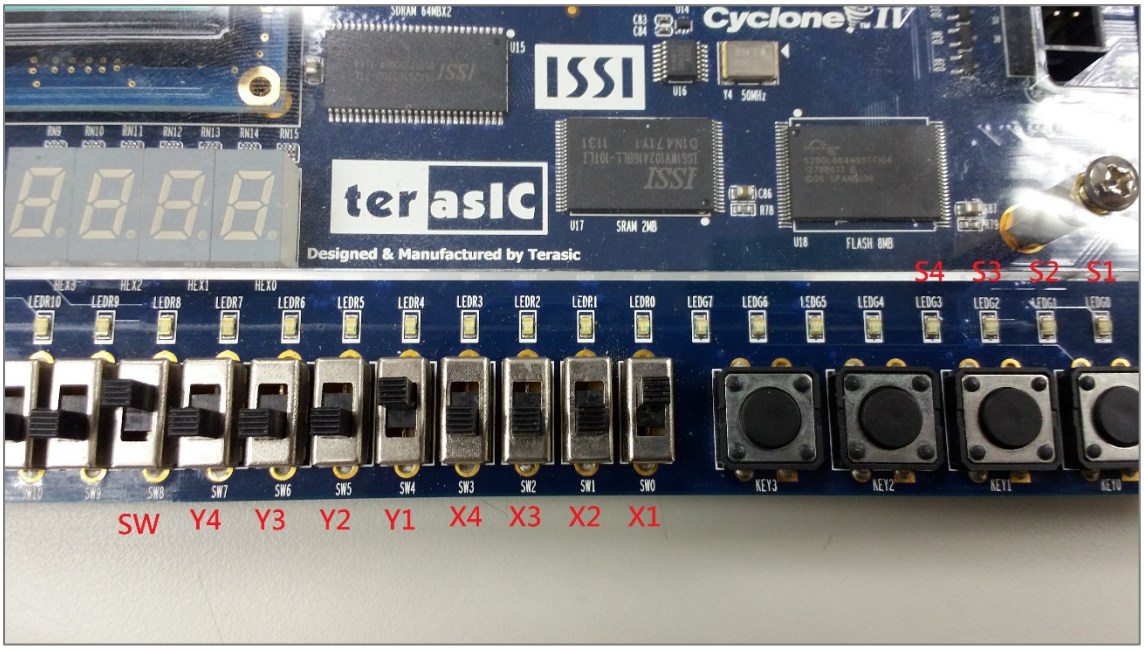
e. X=1001_2 (9)，Y=0110_2 (6)，S=1111_2 (15)，15為最大的數字。



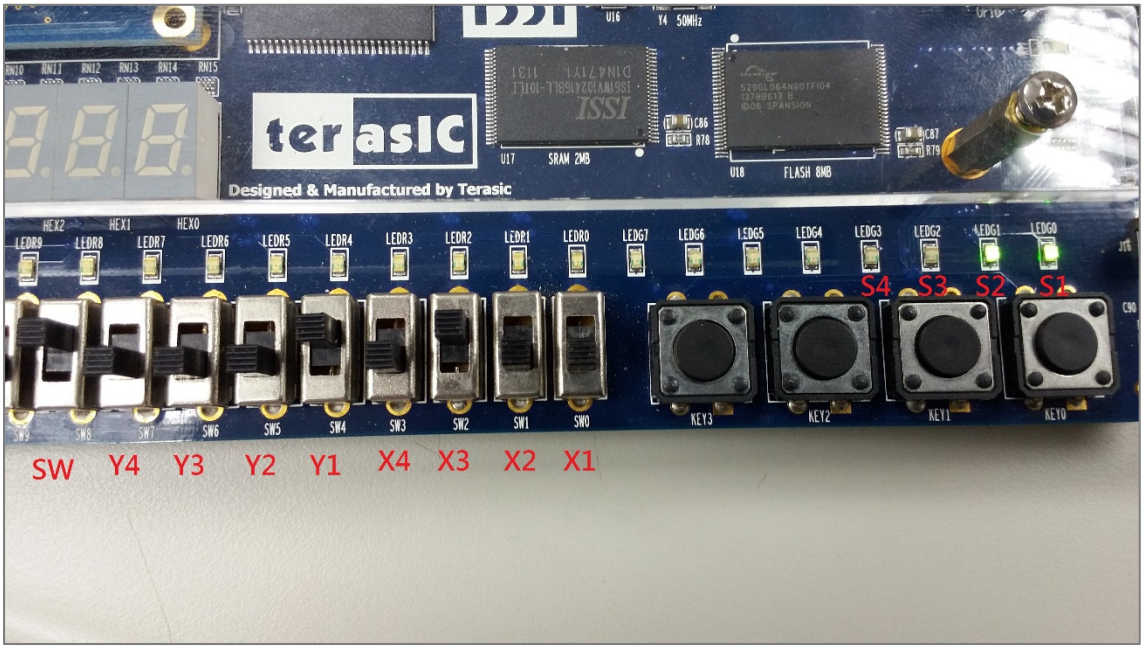
f. 測試減法功能，將 SW 設為1，僅輸入 X=0001_2 可以得到 S=0001_2。



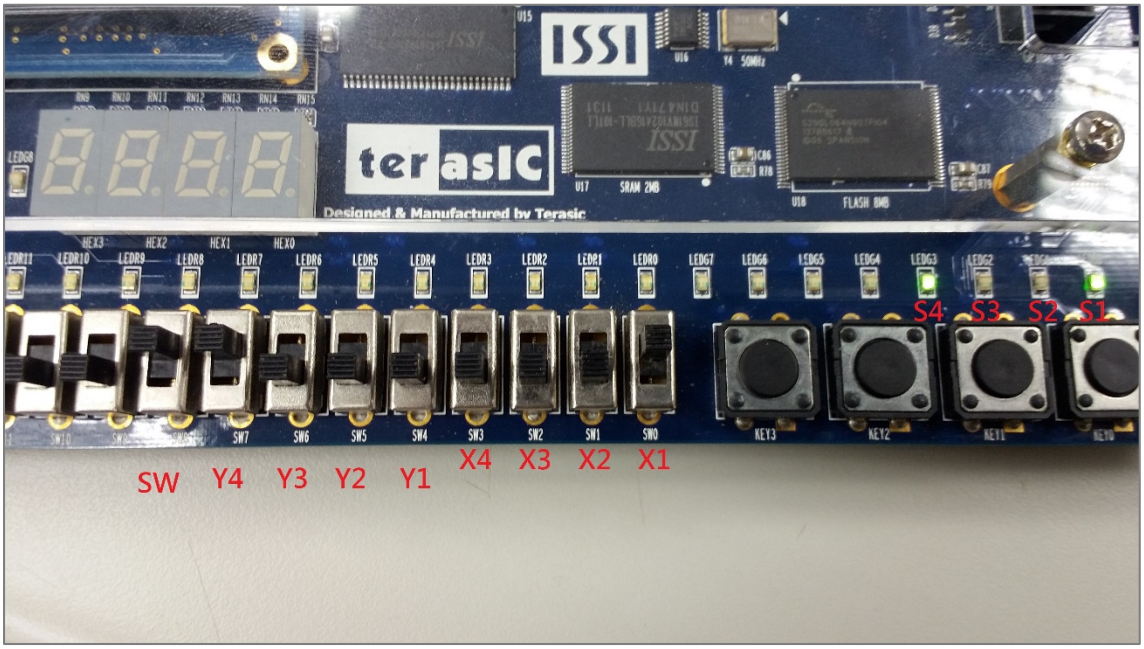
g. 再增加 Y=0001_2，相同的數字相減就會得到 S=0000_2。



h. $X=0100_2$ (4) 減掉 $Y=0001_2$ (1) 得到 $S=0011_2$ (3)。



i. $X=0001_2$ (1)，$Y=1000_2$ (8)，$S=11001_2$ (-7)，S 還有1個位元作為符號位元代表正負號，但此處並沒有顯示出來。



程式碼解釋：

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.all;

ENTITY Lab2 IS
    PORT(
        x0, x1, x2, x3 : IN STD_LOGIC;
        y0, y1, y2, y3 : IN STD_LOGIC;
        S0, S1, S2, S3 : OUT STD_LOGIC;
        SW : IN STD_LOGIC
    );
END Lab2;

ARCHITECTURE Lab2_content OF Lab2 IS
SIGNAL
    C0, C1, C2, C3 : STD_LOGIC;
BEGIN
    C0 <= SW;

    S0 <= x0 XOR (y0 XOR SW) XOR C0;
    C1 <= (x0 AND (y0 XOR SW)) OR (x0 AND C0) OR ((y0 XOR SW) AND C0);

    S1 <= x1 XOR (y1 XOR SW) XOR C1;
    C2 <= (x1 AND (y1 XOR SW)) OR (x1 AND C1) OR ((y1 XOR SW) AND C1);

    S2 <= x2 XOR (y2 XOR SW) XOR C2;
    C3 <= (x2 AND (y2 XOR SW)) OR (x2 AND C2) OR ((y2 XOR SW) AND C2);

    S3 <= x3 XOR (y3 XOR SW) XOR C3;

END Lab2_content;
```

紅色圈：定義了 ENTITY 外面接的接口名字，x 系列是第一串數字, y 系列是第二串數字。S 系列是 sum，會把 x 跟 y 相加或者相減的 sum 寫在 S 系列。而 SW 則是用來控制要加還是減。

橙色圈：定義一個 SIGNAL，C 系列是用來存放上一個 carry 進來的值。

黃色圈：把相加或者相減運算出來的邏輯。並且以 SW 作為要相加還是相減。

3. 實驗心得：

梁皓鈞(104360098)：

這一次的實驗內容，在基本實驗沒什麼問題，對於 Boolean Expression 以及邏輯運算上在加法器都了解原理，但是在加分題中的減法器我就不太熟悉了。雖然在大一時候有上過數位邏輯而且分數也不錯，可是始終因為我 Secondary Education 中沒有接觸過電子電路相關的課程，而大一數位邏輯因為教很快所以很多時候我都為了考試沒辦法要先背下來。因此對於數值的計算不是很熟悉。但幸好晟毅他耐心地教我，讓我回想起數位邏輯中二補數的運用。

透過這一次實驗讓我回憶起電路相關的知識以及二進制的計算方式，對我十分有用，畢竟因為之後的微算機實驗以及考試都需要用到相關知識，幸好在這些實驗跟考試前經過這次實驗作為複習。

這次實驗比較麻煩就是明明寫出來的程式是對的，可是因為 DE2板子好像有點因為使用太久而出現一點誤差。例如當 Switch 是 OFF 的情況下燈卻會亮起來，因此需要每一次都把所有 Switch 都用點力壓下去 OFF 才會回復到正常狀態。

此外，我認為自己比較不熟悉的是實作。即使我知道要把一個數字轉成負數只要弄成二補數就可以，同時也知道二補數就是把原本數字全部 bit 反向然後加1，但我的知識也只是在理論上明白，實際上要我使用 Logic 去實現我就做不到了。後來透過晟毅我知道了原來是使用到 XOR 達到反向效果然後再加一後就可以變成負數了。有關這一點我會之後再進行複習，希望可以更加熟悉這方面的計算。

晟毅：

皓鈞與我對於加法器的布林代數式、邏輯運算都沒有問題，我和他都成功獨立寫出加法器的 VHDL 程式碼。但當要進行加分題的需求時候，因皓鈞對於數位邏輯的數字系統不太熟悉，遲遲無法將減法器的功能實作出來，更無法將加減法功能順利整合，而我就將它原有的程式碼稍微重新規劃後，跟他解釋二進位補數系統的原理，他也逐漸想起關於這方面的細節，最後也順利一起將加分題的需求解決。

實作上比較麻煩的部分可能是必須聯想到 XOR 閘實現 NOT 閘的關鍵，藉由 A XOR 1即可得到 A 的反向的特性來實現出加減法切換的開關功能，讓 SW 輸入1時，使輸入的 y 全部位元進行反向，並且讓 MSB 的位元先增加一個 carry，即可獲得 y 的2補數，而使 x、y*相加即是 x-y 的結果。

4. 組員貢獻度及工作內容：

名字	負責項目內容	貢獻比例	貢獻總和
皓鈞	負責程式碼撰寫	15%	50%
	報告實驗內容的程式碼解釋	10%	
	報告實驗心得撰寫	25%	
晟毅	負責實驗結果驗證	10%	50%
	報告實驗內容、實驗過程撰寫	10%	
	報告實驗心得撰寫	25%	
	程式重新規劃	5%	
總計		100%	100%