

微算機系統

實驗一

組別： 14

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

日期： 2016.10.03

1. 實驗內容：

此次實驗要利用 VHDL 撰寫1個7段顯示器的解碼器，將0000₍₂₎ 規格的二進制數字轉換為0~9、A~F 顯示在7段顯示器上。

進制轉換表：

10進制	0	1	2	3	4	5	6	7
2進制	0000	0001	0010	0011	0100	0101	0110	0111
16進制	0	1	2	3	4	5	6	7

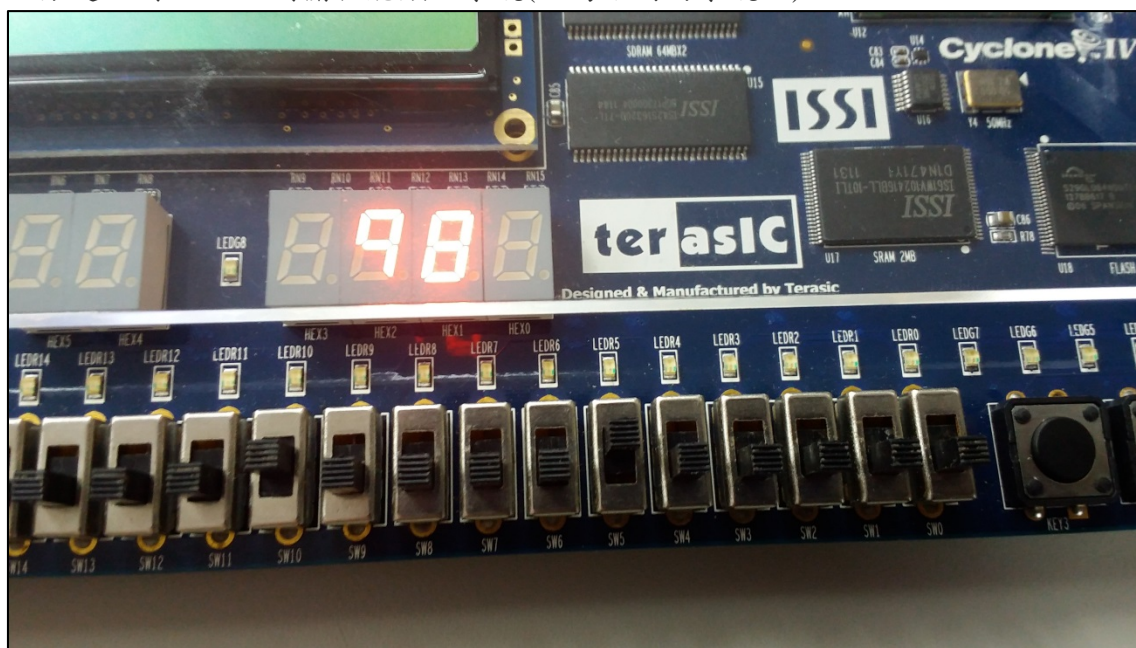
10進制	8	9	10	11	12	13	14	15
2進制	1000	1001	1010	1011	1100	1101	1110	1111
16進制	8	9	A	B	C	D	E	F

2. 實驗過程及結果：

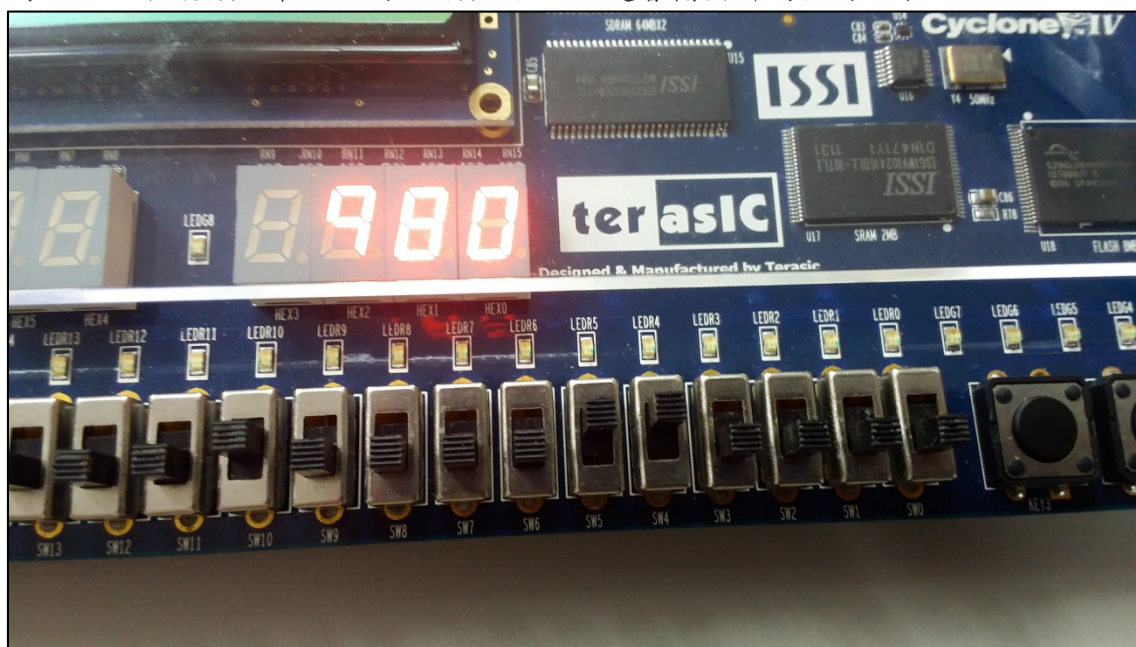
此次實驗要求的是使用 VHDL 中最簡單的邏輯運算子來完成所有功能，所以我們得先列出7段顯示器解碼器的真值表，但若是直接依照真值表下去撰寫，會得出很長一大串的程式碼，因此我們還得事先利用卡諾圖將7段顯示器中各個輸出的布林代數式作化簡，最後才用最簡的式子去撰寫程式碼。

實驗結果：

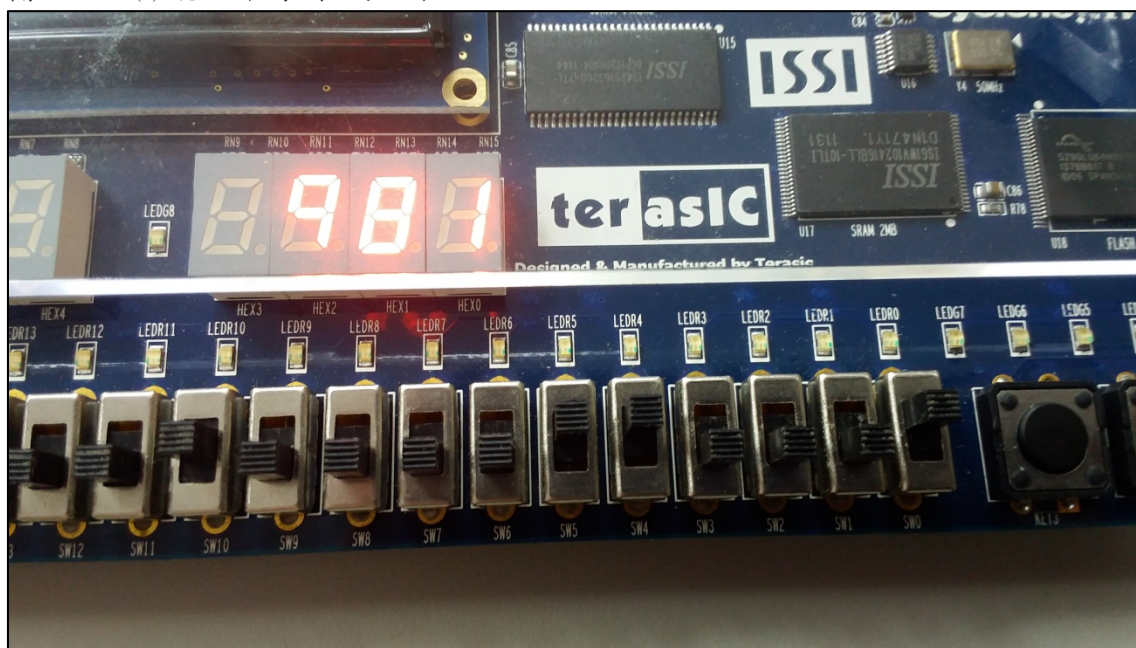
- a. 加分題，開啟 SW5開關就能顯示學號(此為皓鈞的學號98)。



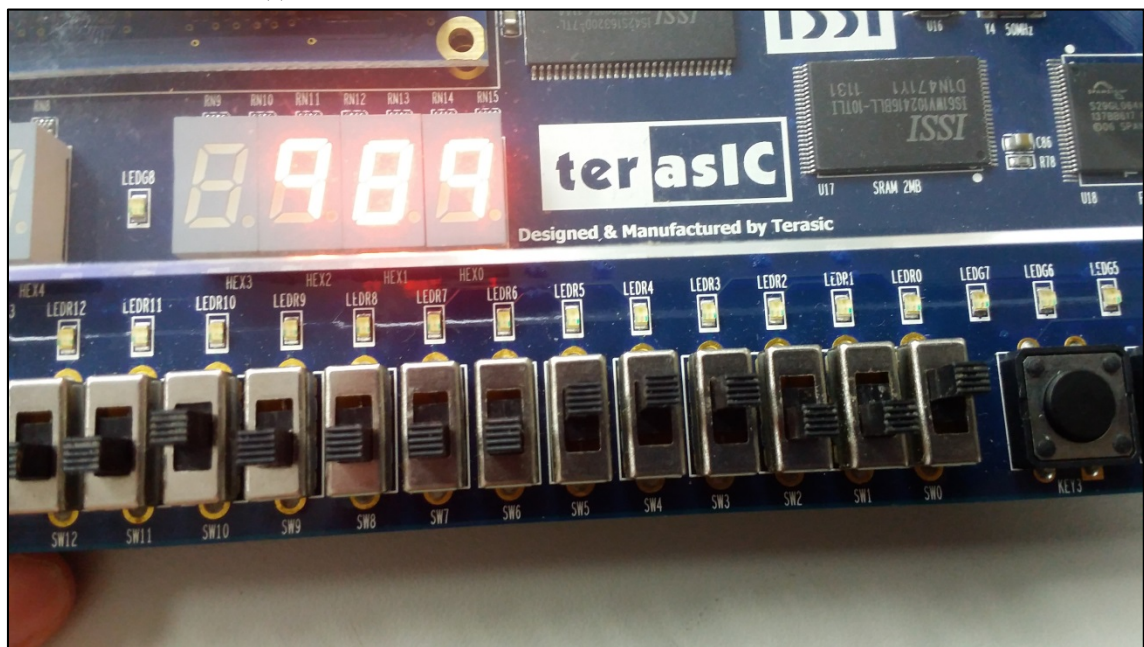
- b. 開啟 SW4就能顯示最右方的7段顯示器，此處會輸出解碼器的結果。



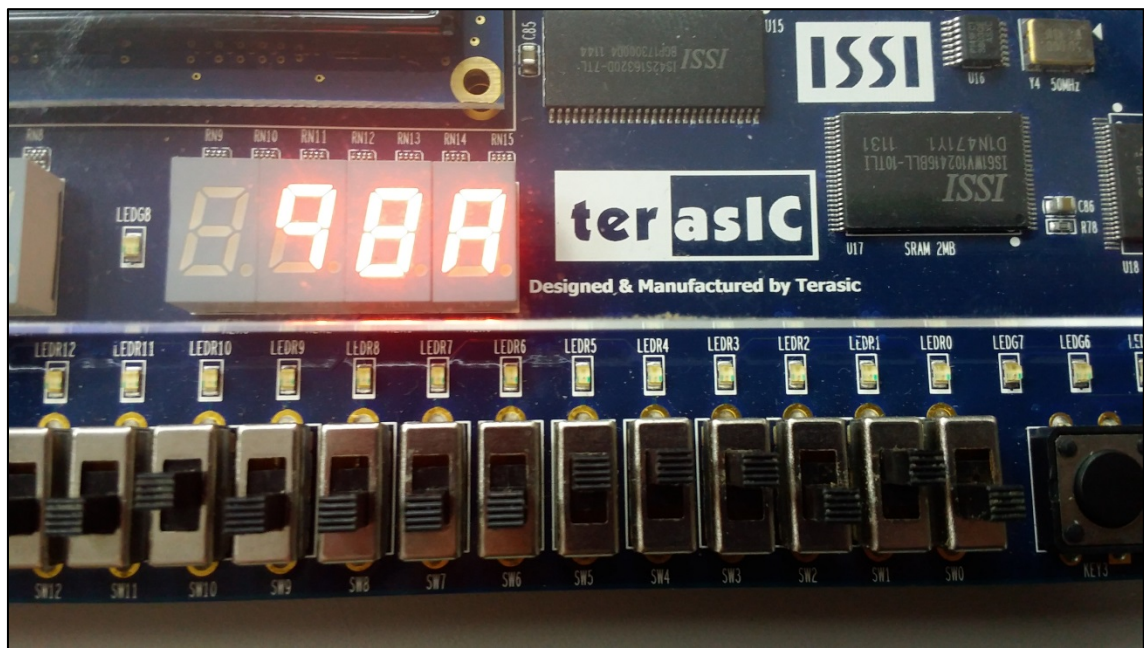
- c. 輸入0001₍₂₎ 能正確得到1的結果。



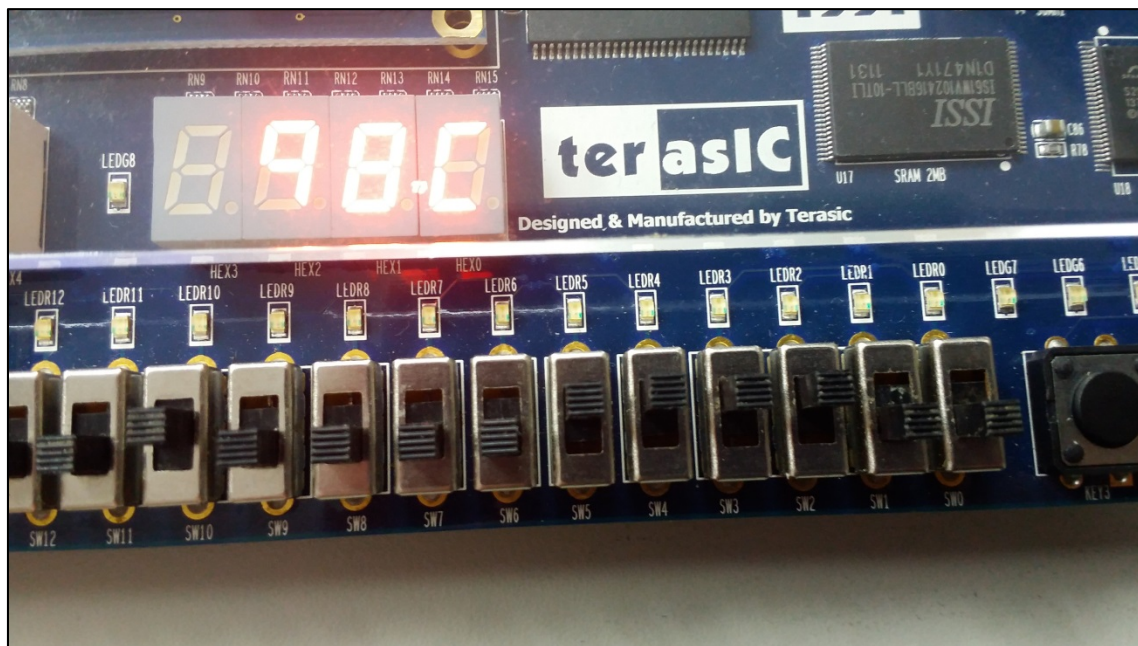
- d. 依序增加來到 $1001_{(2)}$ 會顯示9的結果。



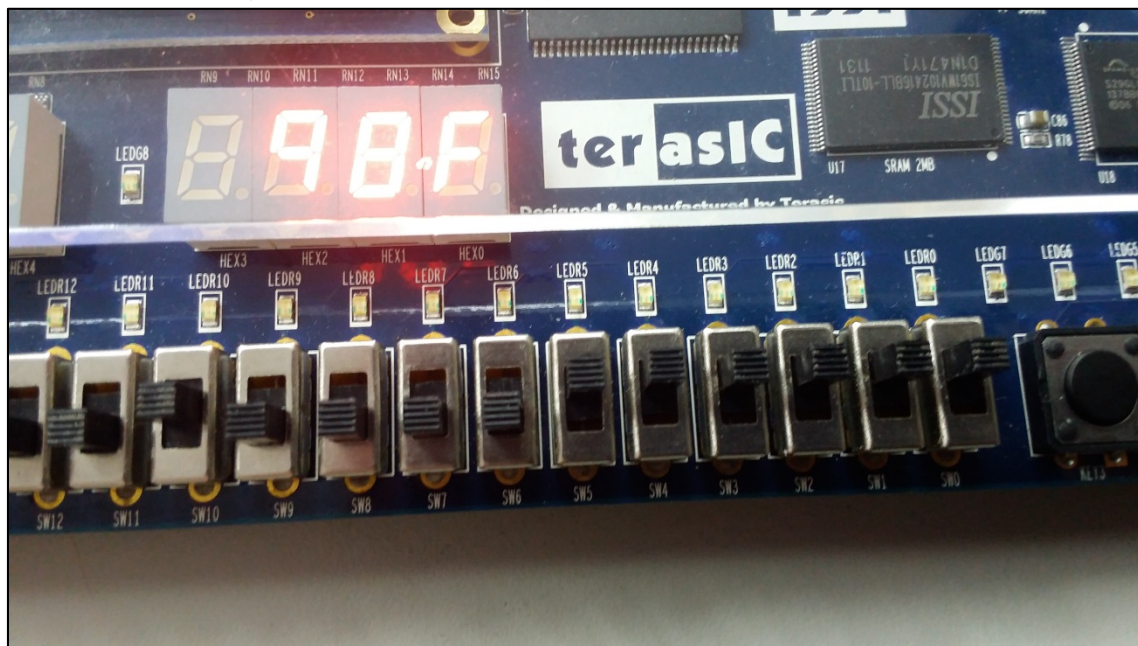
- e. 此時繼續增加1，會得到 $1010_{(2)} = A$ 的結果，A代表著10。



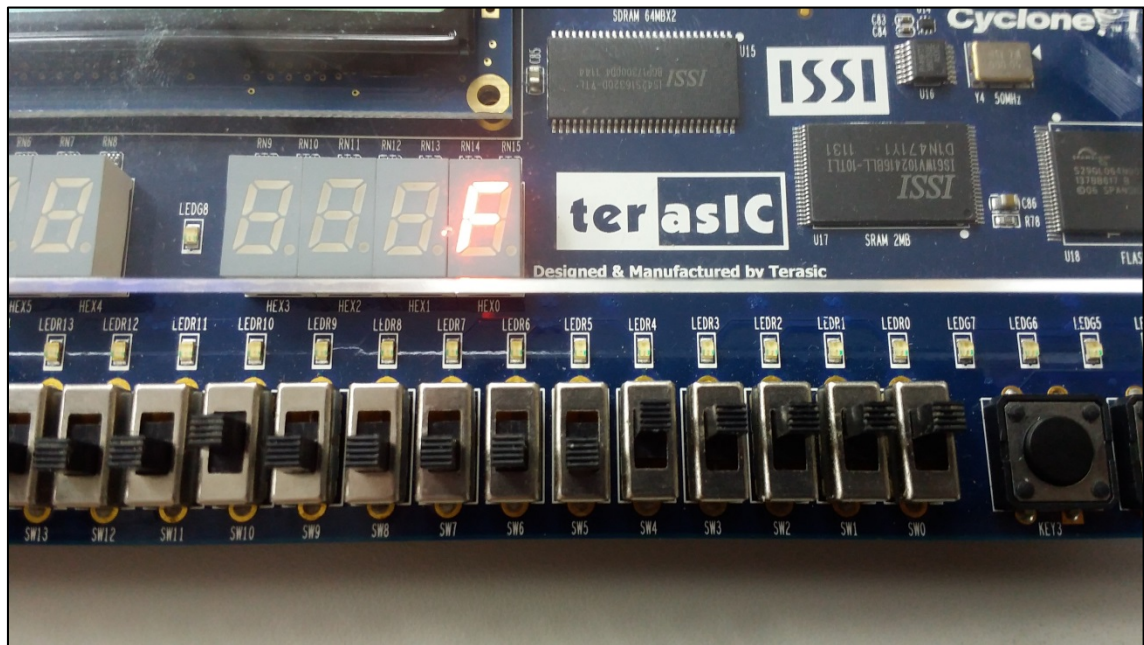
- f. 我們發現實驗題目中提供參考的真值表，在 $1100_{(2)} = C$ 的部分有錯誤，顯示出的是小寫 c，我們基於習慣修改成顯示大寫 C。



- g. 最後將所有開關轉為 On，就可以得到學號及 $1111_{(2)} = F$ 。



- h. 而關閉 SW5取消顯示學號也能正常顯示出 F。



程式碼解釋：

第一次寫的程式碼：

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
ENTITY vhd12_1 IS
    PORT (
        z, y, x, w, sw4, sw5 : IN STD_LOGIC;
        hex0, hex1, hex2 : OUT STD_LOGIC_VECTOR (0 to 6)
    );
END vhd12_1;
ARCHITECTURE content OF vhd12_1 IS
    SIGNAL inputs : STD_LOGIC_VECTOR (4 downto 0);
BEGIN
    inputs <= sw4 & w & x & y & z;
    WITH inputs SELECT
        hex0 <= "0000001" WHEN "10000",
                "1001111" WHEN "10001",
                "0010010" WHEN "10010",
                "0000110" WHEN "10011",
                "1001100" WHEN "10100",
                "0100100" WHEN "10101",
                "1100000" WHEN "10110",
                "0001111" WHEN "10111",
                "0000000" WHEN "11000",
                "0001100" WHEN "11001",
                "0001000" WHEN "11010",
                "1100000" WHEN "11011",
                "1110010" WHEN "11100",
                "1000010" WHEN "11101",
                "0110000" WHEN "11110",
                "0111000" WHEN "11111",
                "1111111" WHEN others;

    PROCESS (sw5)
    BEGIN
        IF (sw5 = '1') THEN
            hex1 <= "0000000";
            hex2 <= "0001100";
        ELSE
            hex1 <= "1111111";
            hex2 <= "1111111";
        END IF;
    END PROCESS;
END content;
```

橙色：定義 zyxw 為輸入，以及 sw4 sw5 為加分題中控制開關

定義 hex0 hex1 hex2 為輸出的 Seven Segment

黃色：用 With Select 語法，把 zyxw 以及 sw4 串接成為 Input Signal 進行 Condition，根據不同的 Signal 情況輸出不同的訊號到 Seven Segment，顯示由 0 到 F

紅色：把 sw5 放到 buffer，成為 Sensitive Variable，使用 IF 去去區分，當 sw5 開了後，輸出 8 與 9 的二進制到 hex 1 與 2，顯示出“98”，當 sw5 關了之後便全部熄掉

使用 Boolean Expression 後的程式碼：

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

ENTITY vhd12_1 IS
PORT (
    W, X, Y, Z, sw4, sw5 : IN STD_LOGIC;
    HEX, HEX1, HEX2: OUT STD_LOGIC_VECTOR(0 TO 6)
);
END vhd12_1;

ARCHITECTURE content OF vhd12_1 IS
BEGIN
    HEX(0) <= (((NOT W) AND (NOT X) AND (NOT Y) AND Z) OR
              ((NOT W) AND X AND (NOT Y) AND (NOT Z)) OR
              ((NOT W) AND X AND Y AND (NOT Z)) OR
              (W AND (NOT X) AND Y AND Z) OR
              (W AND X AND (NOT Y) AND (NOT Z)) OR
              (W AND X AND (NOT Y) AND Z)) AND sw4);
    HEX(1) <= (((NOT W) AND X AND (NOT Y) AND Z) OR
              ((NOT W) AND X AND Y AND (NOT Z)) OR
              (W AND (NOT X) AND Y AND Z) OR
              (W AND X AND (NOT Y) AND (NOT Z)) OR
              (W AND X AND Y AND (NOT Z)) OR
              (W AND X AND Y AND Z)) AND sw4);
    HEX(2) <= (((NOT W) AND (NOT X) AND Y AND (NOT Z)) OR
              (W AND X AND (NOT Z)) OR
              (W AND X AND Y)) AND sw4);
    HEX(3) <= (((NOT X) AND (NOT Y) AND Z) OR
              ((NOT W) AND X AND (NOT Y) AND (NOT Z)) OR
              (X AND Y AND Z) OR
              (W AND (NOT X) AND Y AND (NOT Z))) AND sw4);
    HEX(4) <= (((NOT W) AND Z) OR
              ((NOT W) AND X AND (NOT Y)) OR
              ((NOT X) AND (NOT Y) AND Z)) AND sw4);
    HEX(5) <= (((NOT W) AND (NOT X) AND Z) OR
              ((NOT W) AND (NOT X) AND Y) OR
              ((NOT W) AND Y AND Z) OR
              (W AND X AND (NOT Y))) AND sw4);
    HEX(6) <= (((NOT W) AND (NOT X) AND (NOT Y)) OR
              ((NOT W) AND X AND Y AND Z)) AND sw4);

    HEX1(0) <= sw5;
    HEX1(1) <= sw5;
    HEX1(2) <= sw5;
    HEX1(3) <= sw5;
    HEX1(4) <= sw5;
    HEX1(5) <= sw5;
    HEX1(6) <= sw5;

    HEX2(0) <= sw5;
    HEX2(1) <= sw5;
    HEX2(2) <= sw5;
    HEX2(5) <= sw5;
    HEX2(6) <= sw5;
END content;
```

橙色：定義 zyxw 為輸入，以及 sw4 sw5 為加分題中控制開關

定義 hex0 hex1 hex2 為輸出的 Seven Segment

黃色：使用 Logic Gate 的 Boolean Expression 去做，由真值表化簡後輸入
在最後 AND sw4，用途在於控制他只有在 sw4 = 1 時才會亮

紅色：直接針對輸出 '8' 在 hex1 以及 '9' 在 hex2 進行輸入

當 sw5 為 1 時，直接特定的 segment 會亮

3. 實驗心得：

梁皓鈞 (104360098):

在上星期的最後幾天看到這個實驗內容已經登上了北科i 學園。由於我因為轉系，課業量比較重因此我提早把實驗內容的文件下載下來，並且先在家做好了。VHDL 在我大一的時候已經學習過一點點，因此也相比之下比較快上手。在這個實驗我一開始是使用了With Select 以及 Process IF 去處理，所以比較快處理好。但後來才知道原來老師需要我們用數位邏輯去處理，因此後面我們也把程式碼改成了Boolean Expression 。

這次實驗難度不高，因為也只是單純顯示一些數字而已，簡單來說就只是左一個Seven Segment 然後另外兩個單純顯示，另外再找兩個Enable 去當作開關的作用而已。反而令我比較覺得困難的是從真值表找出Boolean Expression 。因為我不是工科出身的，而在香港的教育制度中也沒有技職體系。在完全學術的教育制度中完全沒有接觸過任何工科的東西，所以對於怎樣從真值表歸納出邏輯我還是不太熟練，在這一點我後來也有在家嘗試過，後來終於成功了。

基本上在VHDL 程式上的基本運作我應該是沒問題，只是題目是中文所以我看得比較慢而已，但整體不影響。暫時比較困擾的是我不清楚寫出來的程式(.vhd) 以及另外 (qsf) 及 (qpf) 檔案是否能在助教的環境下編譯出來。因為我在自己的Laptop 安裝的是 Quartus II 9.1 ，而編寫環境也是在9.1 版本上編寫的，因此不知道助教的環境是否能運行出我的專案。但可以確認的是我成功在我的Laptop 上透過9.1 版本的Quartus II 把程式燒錄進去DE2 板子，並且是成功運行通過助教測試的，所以估計是沒有問題，只是擔心在助教的環境不知道能不能順利編譯到，有關這一點之後會再請教助教。

我之所以會用9.1 是因為DE2板子不給我們帶回家燒錄測試，而9.1版本有Vector Wave Form 可以有所有輸入情況的波形圖顯示，因此可以在沒有板子燒錄的情況下，透過波形圖進行手動人手比對來debug。這樣便可以在上課前提早準備好沒有任何錯誤的實驗程式。

總括而言，這次實驗難度不高，屬於喚醒記憶性質，可以比較輕鬆掌握 到，而對我來說這一次實驗我重新溫習了怎樣去化簡真值表。

洪晟毅 (104590048):

一開始撰寫程式碼的是皓鈞，完整的程式碼從頭至尾都是他親自撰寫的，雖然功能順利完成，但我清楚了解這樣對我是沒有幫助的，所以我也試著從頭自己編寫，不去看他所寫出來的程式碼，遇到不理解的語法才會去請教皓鈞。我本來考慮使用很多IF判斷式來撰寫，但是皓鈞建議我一個WITH SELECT的語法，稍微經過研究之後，很順利的運用此語法完成所有的功能要求。

我們是最早完成的組別，在教授說明題目要求前就已事先預習並完成，因此當要驗收時才知道原來僅能使用最單純的邏輯運算子，所以我們只好再從頭開始撰寫，而且必須用最原始的方式來完成，幸好單純的邏輯開對我們而言完全不是問題，只是要化簡的量較多，總共有7個输出的布林代數式要處理，而撰寫程式碼的過程也挺枯燥的，基本上就是按照化簡的式子將AND、OR、NOT輸入至程式碼中。

雖然是非常機械式的步驟，但此段過程也不是毫無幫助，這讓我體會到當我們沒有別人所提供的工具，我們要從零創造東西時所體會到的困難，這沒有充足的堅持及專注力是非常不容易辦到的，也因此我們應該善用、珍惜前人所留下的成果。

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	15%	
	報告實驗心得撰寫	25%	
總計		100%	100%