**微算機系統**

實驗二

組別： 108590451

班級、姓名與學號： 資工二 陸詠涵 108580451

日期： 2020.10.08

1. 實驗內容：

本次實驗主旨目標

將上課新教的package語法、之前實驗一(七段顯示器)和全加法器

結合，實際了解組合電路的內部運作，及最後燒錄的結果。

1. 目標一-8-bits 多位元加法器
2. 目標二-BCD加法器
3. 實驗過程及結果：

撰寫你如何完成本次作業的流程方法。

請附上結果圖片(放置於此部分即可)。

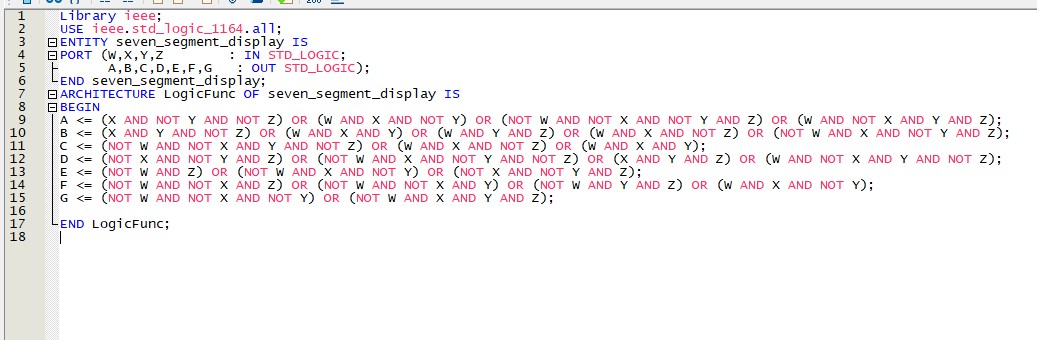
1. 目標一-8-bits 多位元加法器:

實驗步驟:

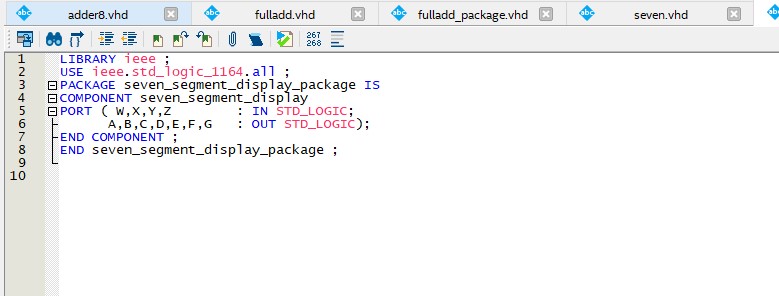
Step1:設計VHDL程式碼

利用上課所學的package 語法，分別寫出全加法器及七段顯示器(實驗一)的程式，再寫8-bits多元加法器的主程式來連接(利用USE work.fulladd\_package.all ;

和USE work.seven\_segment\_display\_package.all ;先將全加法器及七段顯示器呼叫主程式，再利用stage0: fulladd PORT MAP ( '0', X(0), Y(0), S(0), CC(1) ) ;來對應實際的inputs和outputs)。



(picture 1)七段顯示器的VHDL程式碼

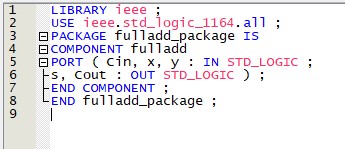


(picture 2)七段顯示器packag的VHDL程式碼

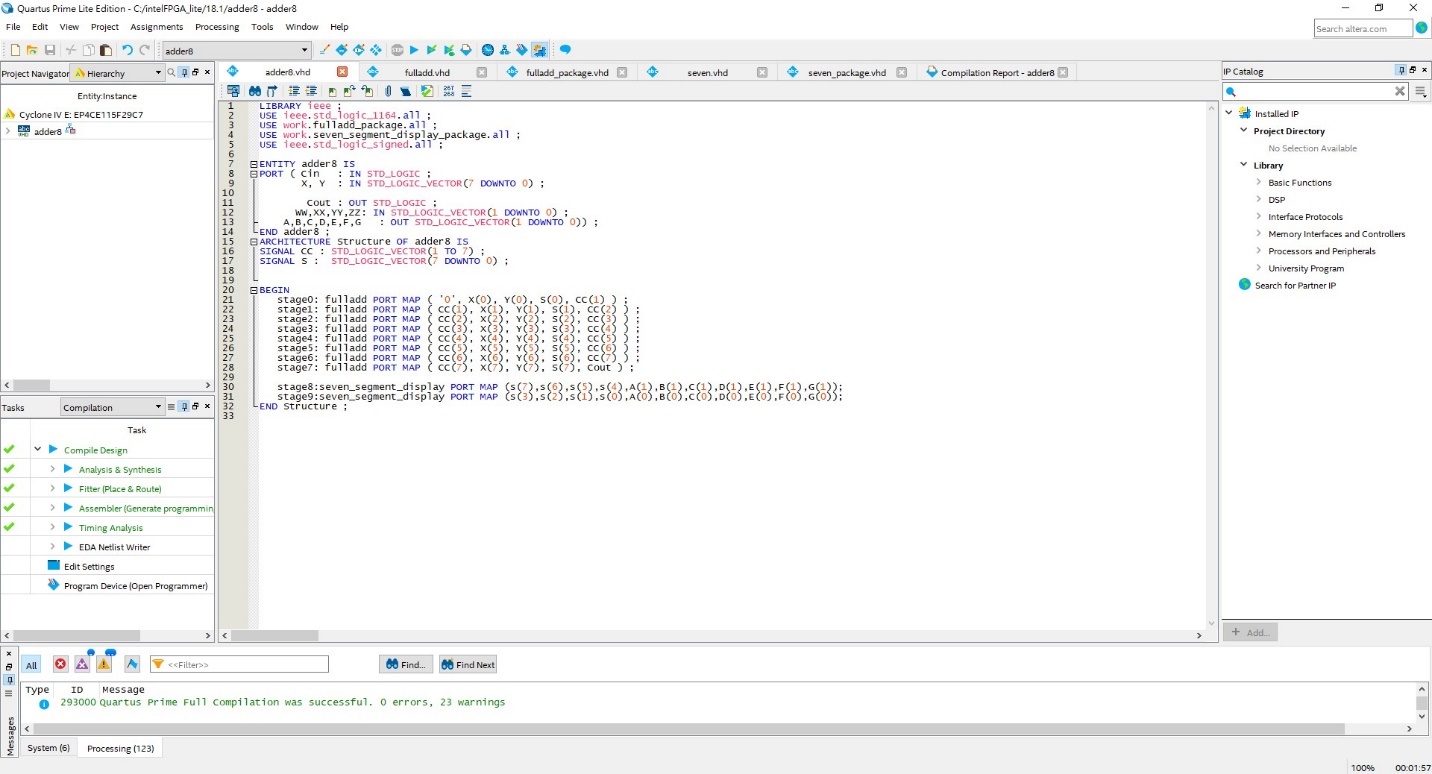
一張含有 文字 的圖片

自動產生的描述

(picture 3)全加法器的VHDL程式碼

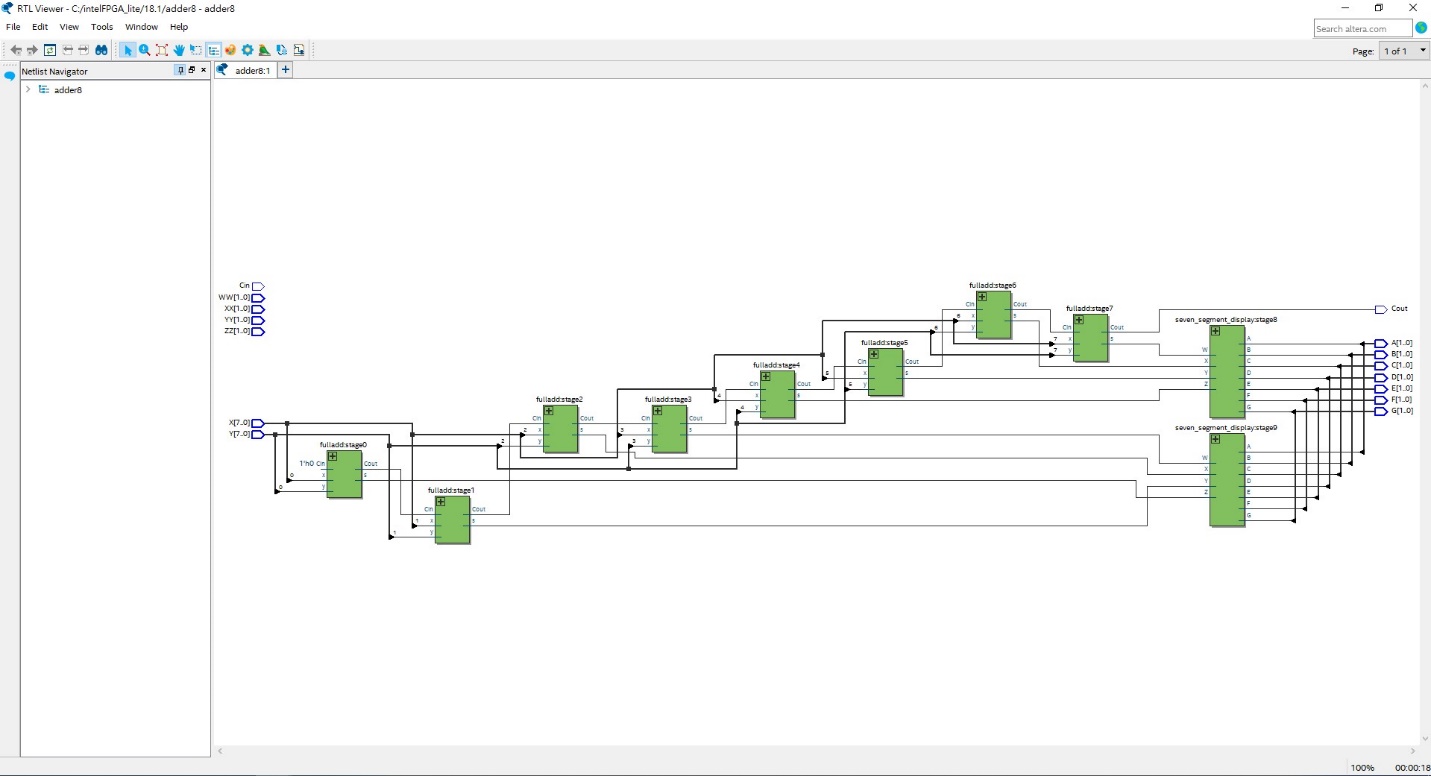


(picture 4)全加法器package的VHDL程式碼

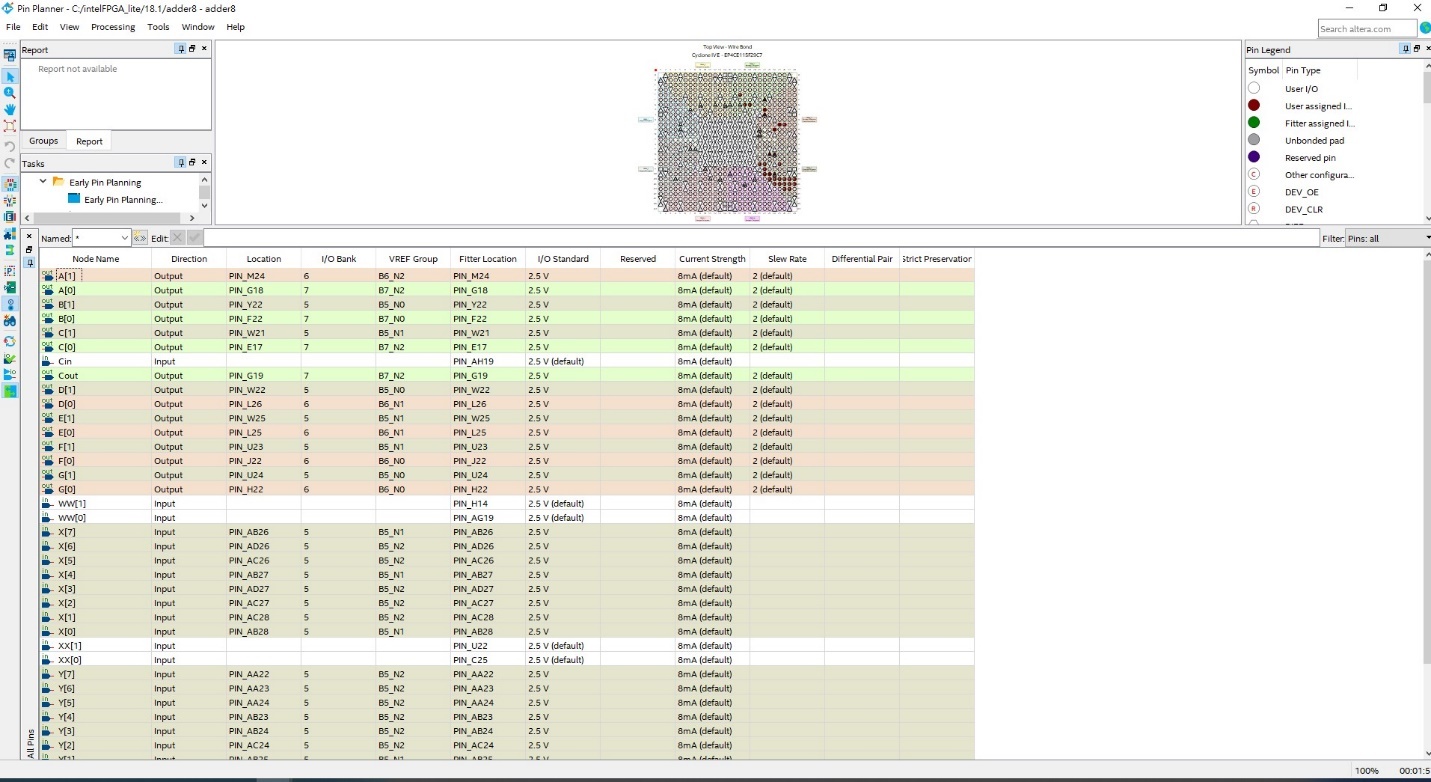


(picture 5)8-bits 多位元加法器的VHDL主程式碼

Step2:接腳位



(picture 6) 8-bits 多位元加法器邏輯電路圖



(picture 7) 8-bits 多位元加法器所接的腳位

Step3:燒錄到模板中

一張含有 電子用品, 電路 的圖片

自動產生的描述

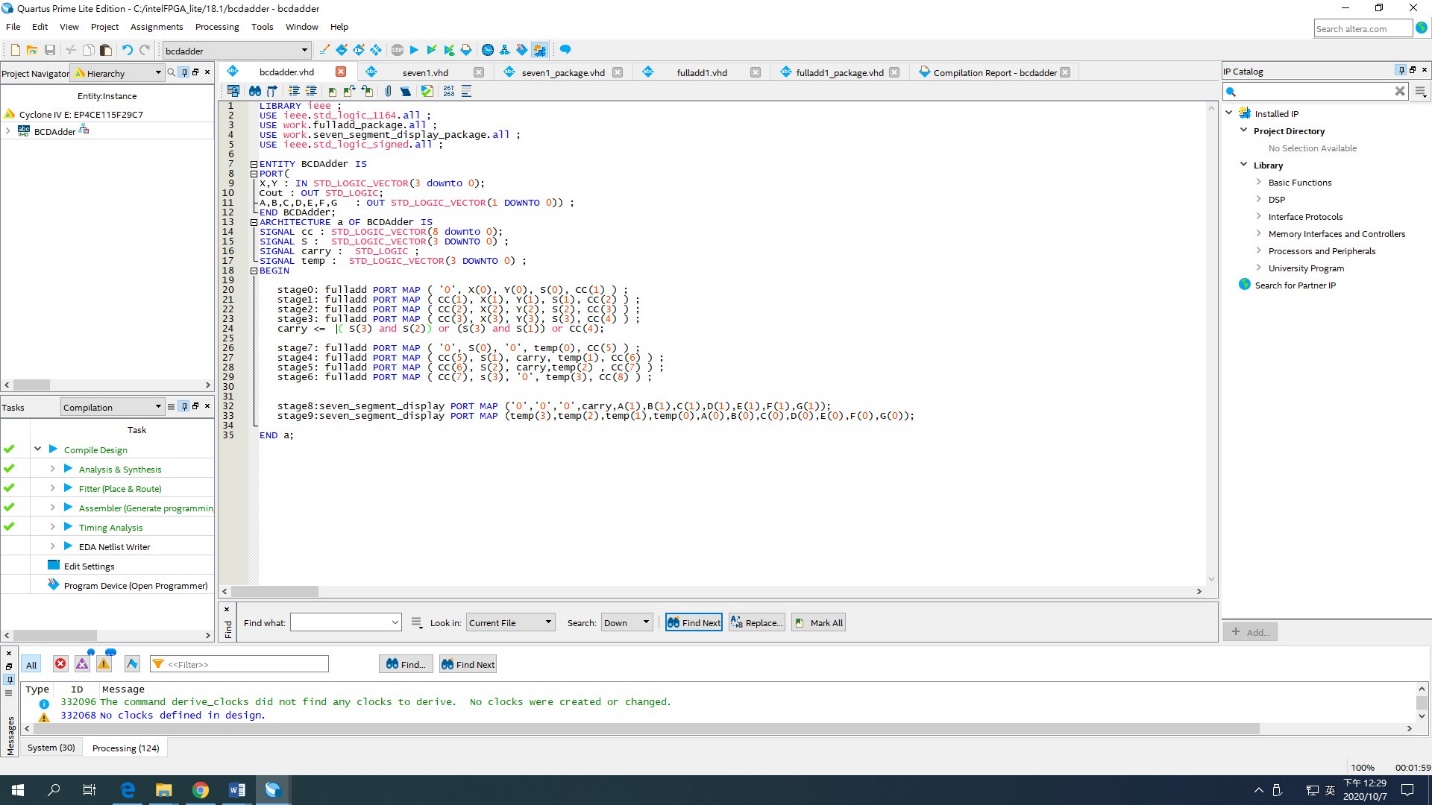
(picture 8) 8-bits 多位元加法器3+3所加的結果

1. 目標二-BCD加法器

實驗步驟:

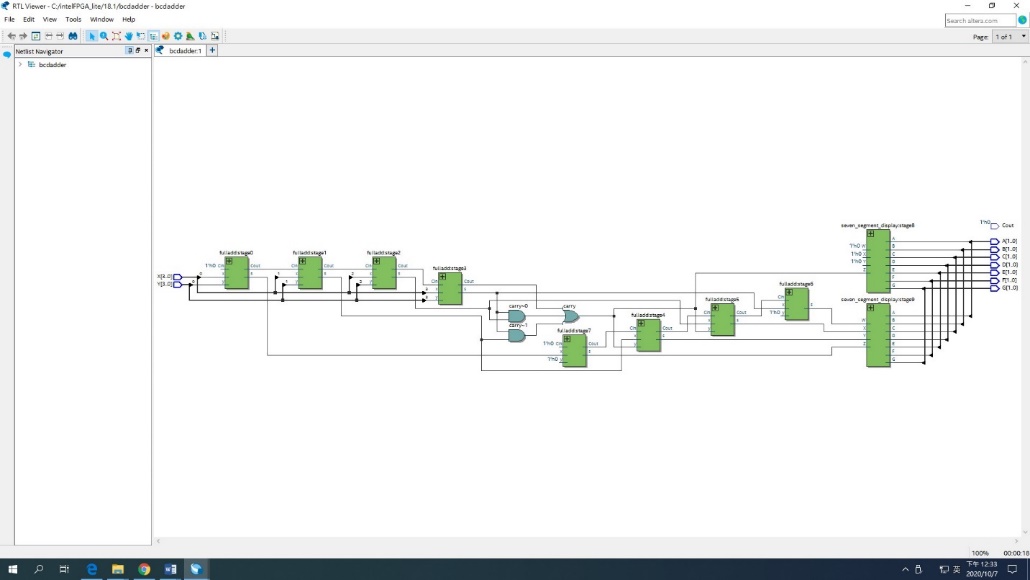
Step1:設計VHDL程式碼

將目標一的fulladder和七段顯示器的程式碼留著，再另外寫一個BCDadder的主程式碼(大部分和8-bits 多位元加法器的VHDL主程式碼，但因BCDadder是利用4-bits顯示0~9的是數字，再加法上會有進位的問題 ex:當num >10時，要進位”加110(base 2)”，所以多了carry <= ( S(3) and S(2) ) or ( S(3) and S(1) ) or CC(4) ; 來判斷進位)。

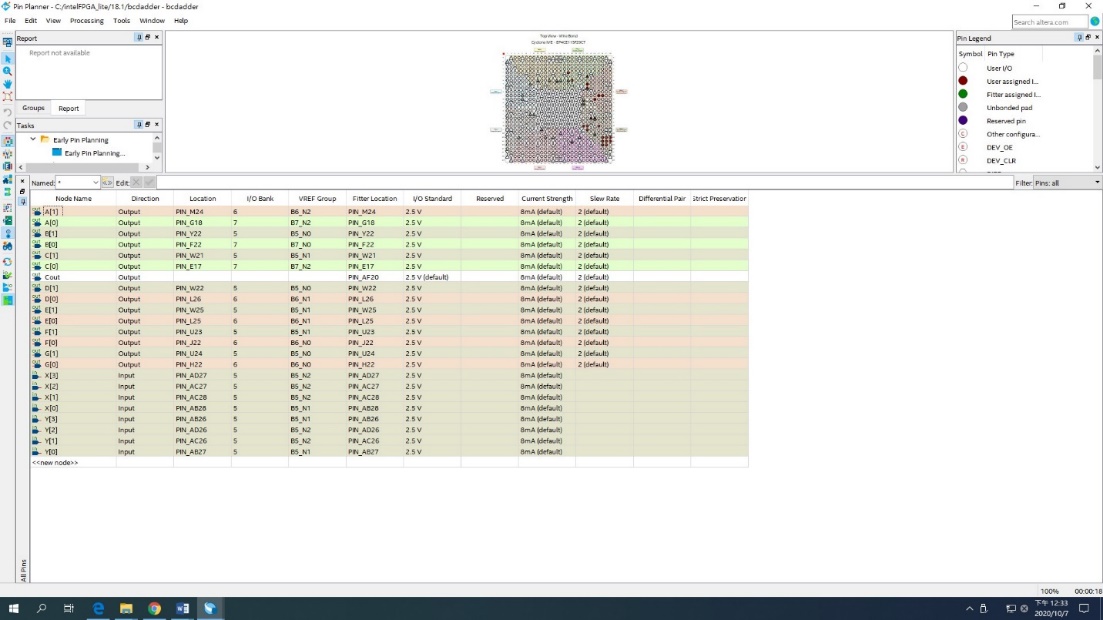


(picture 9)BCD加法器的VHDL主程式碼

Step2:接腳位



(picture 10) BCD加法器邏輯電路圖



(picture 11) BCD加法器所接的腳位

Step3:燒錄到模板中

一張含有 電路, 電子用品 的圖片

自動產生的描述

(picture 12) BCD加法9+9的燒錄結果

1. 程式碼（請調整成最小行高，行高0點）

|  |
| --- |
| 目標一 |
| (程式碼)  **實驗二目標一**  **fulladd**  LIBRARY ieee ;  USE ieee.std\_logic\_1164.all ;  ENTITY fulladd IS  PORT ( Cin, x, y : IN STD\_LOGIC ;  s, Cout : OUT STD\_LOGIC ) ;  END fulladd ;  ARCHITECTURE LogicFunc OF fulladd IS  BEGIN  s <= x XOR y XOR Cin ;  Cout <= (x AND y) OR (Cin AND x) OR (Cin AND y) ;  END LogicFunc ;  **fulladd\_package**  LIBRARY ieee ;  USE ieee.std\_logic\_1164.all ;  PACKAGE fulladd\_package IS  COMPONENT fulladd  PORT ( Cin, x, y : IN STD\_LOGIC ;  s, Cout : OUT STD\_LOGIC ) ;  END COMPONENT ;  END fulladd\_package ;  **seven\_segment\_display**  Library ieee;  USE ieee.std\_logic\_1164.all;  ENTITY seven\_segment\_display IS  PORT (W,X,Y,Z : IN STD\_LOGIC;  A,B,C,D,E,F,G : OUT STD\_LOGIC);  END seven\_segment\_display;  ARCHITECTURE LogicFunc OF seven\_segment\_display IS  BEGIN  A <= (X AND NOT Y AND NOT Z) OR (W AND X AND NOT Y) OR (NOT W AND NOT X AND NOT Y AND Z) OR (W AND NOT X AND Y AND Z);  B <= (X AND Y AND NOT Z) OR (W AND X AND Y) OR (W AND Y AND Z) OR (W AND X AND NOT Z) OR (NOT W AND X AND NOT Y AND Z);  C <= (NOT W AND NOT X AND Y AND NOT Z) OR (W AND X AND NOT Z) OR (W AND X AND Y);  D <= (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);  E <= (NOT W AND Z) OR (NOT W AND X AND NOT Y) OR (NOT X AND NOT Y AND Z);  F <= (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);  G <= (NOT W AND NOT X AND NOT Y) OR (NOT W AND X AND Y AND Z);  END LogicFunc;  **seven\_segment\_display\_package**  LIBRARY ieee ;  USE ieee.std\_logic\_1164.all ;  PACKAGE seven\_segment\_display\_package IS  COMPONENT seven\_segment\_display  PORT ( W,X,Y,Z : IN STD\_LOGIC;  A,B,C,D,E,F,G : OUT STD\_LOGIC);  END COMPONENT ;  END seven\_segment\_display\_package ;  **Adder8(8-bits)**  LIBRARY ieee ;  USE ieee.std\_logic\_1164.all ;  USE work.fulladd\_package.all ;  USE work.seven\_segment\_display\_package.all ;  USE ieee.std\_logic\_signed.all ;  ENTITY adder8 IS  PORT ( Cin : IN STD\_LOGIC ;  X, Y : IN STD\_LOGIC\_VECTOR(7 DOWNTO 0) ;    Cout : OUT STD\_LOGIC ;  WW,XX,YY,ZZ: IN STD\_LOGIC\_VECTOR(1 DOWNTO 0) ;  A,B,C,D,E,F,G : OUT STD\_LOGIC\_VECTOR(1 DOWNTO 0)) ;  END adder8 ;  ARCHITECTURE Structure OF adder8 IS  SIGNAL CC : STD\_LOGIC\_VECTOR(1 TO 7) ;  SIGNAL S : STD\_LOGIC\_VECTOR(7 DOWNTO 0) ;  BEGIN  stage0: fulladd PORT MAP ( '0', X(0), Y(0), S(0), CC(1) ) ;  stage1: fulladd PORT MAP ( CC(1), X(1), Y(1), S(1), CC(2) ) ;  stage2: fulladd PORT MAP ( CC(2), X(2), Y(2), S(2), CC(3) ) ;  stage3: fulladd PORT MAP ( CC(3), X(3), Y(3), S(3), CC(4) ) ;  stage4: fulladd PORT MAP ( CC(4), X(4), Y(4), S(4), CC(5) ) ;  stage5: fulladd PORT MAP ( CC(5), X(5), Y(5), S(5), CC(6) ) ;  stage6: fulladd PORT MAP ( CC(6), X(6), Y(6), S(6), CC(7) ) ;  stage7: fulladd PORT MAP ( CC(7), X(7), Y(7), S(7), Cout ) ;    stage8:seven\_segment\_display PORT MAP (s(7),s(6),s(5),s(4),A(1),B(1),C(1),D(1),E(1),F(1),G(1));  stage9:seven\_segment\_display PORT MAP (s(3),s(2),s(1),s(0),A(0),B(0),C(0),D(0),E(0),F(0),G(0));  END Structure ; |
| 目標二 |
| (程式碼)  **實驗二目標二**  **Fulladd1**  LIBRARY ieee ;  USE ieee.std\_logic\_1164.all ;  ENTITY fulladd IS  PORT ( Cin, x, y : IN STD\_LOGIC ;  s, Cout : OUT STD\_LOGIC ) ;  END fulladd ;  ARCHITECTURE LogicFunc OF fulladd IS  BEGIN  s <= x XOR y XOR Cin ;  Cout <= (x AND y) OR (Cin AND x) OR (Cin AND y) ;  END LogicFunc ;  **Fulladd1\_package**  LIBRARY ieee ;  USE ieee.std\_logic\_1164.all ;  PACKAGE fulladd\_package IS  COMPONENT fulladd  PORT ( Cin, x, y : IN STD\_LOGIC ;  s, Cout : OUT STD\_LOGIC ) ;  END COMPONENT ;  END fulladd\_package ;  **Seven1\_segment\_display**  Library ieee;  USE ieee.std\_logic\_1164.all;  ENTITY seven\_segment\_display IS  PORT (W,X,Y,Z : IN STD\_LOGIC;  A,B,C,D,E,F,G : OUT STD\_LOGIC);  END seven\_segment\_display;  ARCHITECTURE LogicFunc OF seven\_segment\_display IS  BEGIN  A <= (X AND NOT Y AND NOT Z) OR (W AND X AND NOT Y) OR (NOT W AND NOT X AND NOT Y AND Z) OR (W AND NOT X AND Y AND Z);  B <= (X AND Y AND NOT Z) OR (W AND X AND Y) OR (W AND Y AND Z) OR (W AND X AND NOT Z) OR (NOT W AND X AND NOT Y AND Z);  C <= (NOT W AND NOT X AND Y AND NOT Z) OR (W AND X AND NOT Z) OR (W AND X AND Y);  D <= (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);  E <= (NOT W AND Z) OR (NOT W AND X AND NOT Y) OR (NOT X AND NOT Y AND Z);  F <= (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);  G <= (NOT W AND NOT X AND NOT Y) OR (NOT W AND X AND Y AND Z);  END LogicFunc;  **Seven1\_segment\_display\_package**  LIBRARY ieee ;  USE ieee.std\_logic\_1164.all ;  PACKAGE seven\_segment\_display\_package IS  COMPONENT seven\_segment\_display  PORT ( W,X,Y,Z : IN STD\_LOGIC;  A,B,C,D,E,F,G : OUT STD\_LOGIC);  END COMPONENT ;  END seven\_segment\_display\_package ;  **BCDadder**  LIBRARY ieee ;  USE ieee.std\_logic\_1164.all ;  USE work.fulladd\_package.all ;  USE work.seven\_segment\_display\_package.all ;  USE ieee.std\_logic\_signed.all ;  ENTITY BCDadder IS  PORT ( Cin : IN STD\_LOGIC ;  X, Y : IN STD\_LOGIC\_VECTOR(7 DOWNTO 0) ;  Cout : OUT STD\_LOGIC ;  A,B,C,D,E,F,G : OUT STD\_LOGIC\_VECTOR(1 DOWNTO 0)) ;  END BCDadder ;  ARCHITECTURE a OF BCDadder IS  SIGNAL CC : STD\_LOGIC\_VECTOR(8 downto 0) ;  SIGNAL S : STD\_LOGIC\_VECTOR(3 DOWNTO 0) ;  SIGNAL carry : STD\_LOGIC ;  SIGNAL temp : STD\_LOGIC\_VECTOR(3 DOWNTO 0) ;  BEGIN  stage0: fulladd PORT MAP ( '0', X(0), Y(0), S(0), CC(1) ) ;  stage1: fulladd PORT MAP ( CC(1), X(1), Y(1), S(1), CC(2) ) ;  stage2: fulladd PORT MAP ( CC(2), X(2), Y(2), S(2), CC(3) ) ;  stage3: fulladd PORT MAP ( CC(3), X(3), Y(3), S(3), CC(4) ) ;  carry <= ( S(3) and S(2) ) or ( S(3) and S(1) ) or CC(4) ;  stage4: fulladd PORT MAP ('0', S(0), '0', temp(0), CC(5) ) ;  stage5: fulladd PORT MAP ( CC(5), S(1), carry , temp(1), CC(6) ) ;  stage6: fulladd PORT MAP ( CC(6), S(2), carry , temp(2), CC(7) ) ;  stage7: fulladd PORT MAP ( CC(7), S(3), '0', temp(3), CC(8) ) ;    stage8:seven\_segment\_display PORT MAP ('0','0','0',carry,A(1),B(1),C(1),D(1),E(1),F(1),G(1));  stage9:seven\_segment\_display PORT MAP (temp(3),temp(2),temp(1),temp(0),A(0),B(0),C(0),D(0),E(0),F(0),G(0));  END a; |

1. 實驗心得：

每個人的心得報告至少150字以上，有關於此實驗所遇到的難題，解決方法或是對於實驗過程的分析等。（每名組員都要寫）

這次實驗，我們組花了整整四節課的時間。前面第一、二節課，因為我們還不熟悉package的用法，所以只有分別把fulladder和七段顯示器的程式碼寫出來。之後，因為我們怕時間不夠，在期限內無法讓助教檢查，所以我們有利用回家的時間把fulladder、七段顯示器的package和8-bits 多位元加法器寫出來。所以在第三節課一上課十分鐘就把目標一搞定了。但是，BCDadder讓我和我的parnter卡觀很久，原因在於，我們兩個忘記BCD有進位的問題了。在找出問題點之後，我們嘗試了幾次之後，就成功了。