微算機系統

小組專案報告

實驗三：

BCD加法器電路設計

組別： 20

班級、姓名與學號：

醫工三 葉芸茜 B812110004

醫工三 湯青秀 B812110011

日期 :112年10月18日

一、實驗內容：

1. 以1bit全加器為基礎，將上述之 BCD 加法器邏輯函數以 Package 與 component 語法包裝，並以實驗板上面兩顆七段顯示器驗證 BCD 加 法電路是否正確。
2. 測試時，將以指撥開關輸入要相加的數字，而兩顆七段顯示器上必 須可正確顯示出 BCD 十進位加法運算之結果。
3. 在實驗二中，只須使用A0~A7與B0~B7，完成BCD十進位之正整數 相加，亦結果皆為正整數，範圍為 0~99。
4. 禁用IF和SWITCH語法，請以加法器(邏輯閘)方式實現，使用其他方式則酌以扣分。

Variable	Pin Location	Signal Name
X0	PIN_AB28	SW[0]
X1	PIN_AC28	SW[1]
X2	PIN_AC27	SW[2]
X3	PIN_AD27	SW[3]
X4	PIN_AB27	SW[4]
X5	PIN_AC26	SW[5]
X6	PIN_AD26	SW[6]
X7	PIN_AB26	SW[7]
Y0	PIN_AC25	SW[8]
Y1	PIN_AB25	SW[9]
Y2	PIN_AC24	SW[10]
Y3	PIN_AB24	SW[11]
Y4	PIN_AB23	SW[8]
Y5	PIN_AA24	SW[9]

Y6	PIN_AA23	SW[10]
Y7	PIN_AA22	SW[11]
a0	PIN_G18	HEX0[0]
b0	PIN_F22	HEX0[1]
c0	PIN_E17	HEX0[2]
d0	PIN_L26	HEX0[3]
e0	PIN_L25	HEX0[4]
f0	PIN_J22	HEX0[5]
g0	PIN_H22	HEX0[6]
a1	PIN_M24	HEX1[0]
b1	PIN_Y22	HEX1[1]
c1	PIN_W21	HEX1[2]
d1	PIN_W22	HEX1[3]
e1	PIN_W25	HEX1[4]
f1	PIN_U23	HEX1[5]
g1	PIN_U24	HEX1[6]
Overflow	PIN_G19	LEDR[0]

二、實驗過程及結果：

（一） 預期實驗結果的真值表

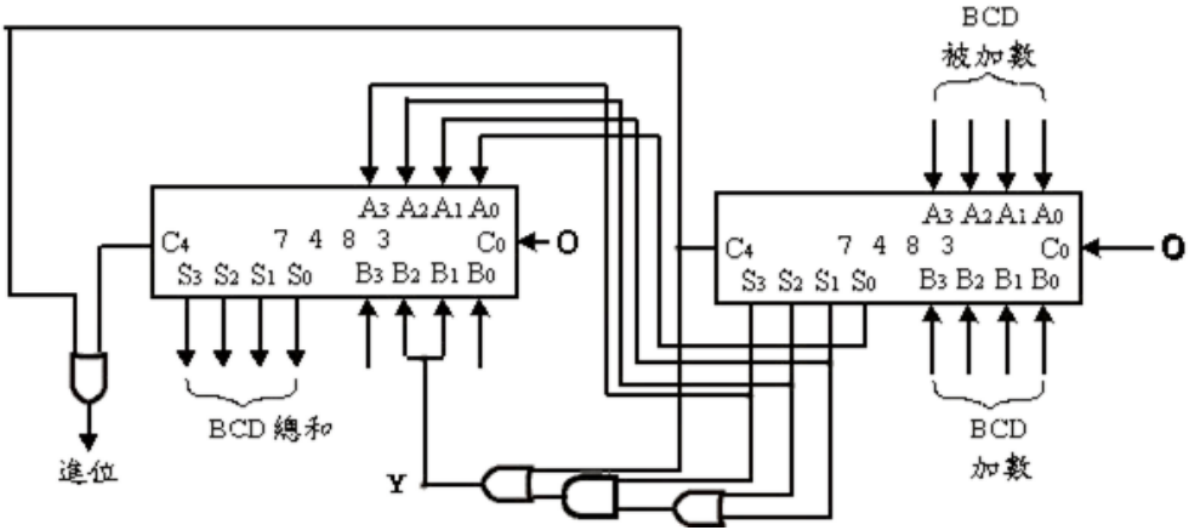
1. 1-bit 全加器

Input			Output	
Cin	x	y	s	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

2. 七段顯示器

數字	W	X	Y	Z	a	b	c	d	e	f	g
0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	1	1	0	0	1	1	1	1
2	0	0	1	0	0	0	1	0	0	1	0
3	0	0	1	1	0	0	0	0	1	1	0
4	0	1	0	0	1	0	0	1	1	0	0
5	0	1	0	1	0	1	0	0	1	0	0
6	0	1	1	0	0	1	0	0	0	0	0
7	0	1	1	1	0	0	0	1	1	1	1
8	1	0	0	0	0	0	0	0	0	0	0
9	1	0	0	1	0	0	0	1	1	0	0

3. 4-bit BCD判斷進位



(1) 判斷數字進位情況

- 數字大於9且小於16
- Cout(4)等於1 → 有進位

(2) 若符合(1)的任意條件，則計算結果加6(+0110)

數字	S3	S2	S1	S0	F
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	0
9	1	0	0	1	0
10	1	0	1	0	1
11	1	0	1	1	1
12	1	1	0	0	1
13	1	1	0	1	1
14	1	1	1	0	1
15	1	1	1	1	1

(S0、S1、S2、S3分別表示程式碼中的t(0)、t(1)、t(2)、t(3))

（二） 根據上方真值表輸出布林代數並化簡

1. 加法器

初始Cin為0

若c[7]為1，則有end-carry，判別為overflow(不在0~99範圍內)

$$s_i = x_i \oplus y_i \oplus c_i$$

$$c_{i+1} = x_i y_i + x_i c_i + y_i c_i$$

2. 七段顯示器

a = $\underline{W}\,\underline{X}\,\underline{Y}\,Z + \underline{W}\,X\,\underline{Y}\,\underline{Z} + W\,\underline{X}\,Y\,Z + W\,X\,\underline{Y}$

b = $\underline{W}\,X\,\underline{Y}\,Z + \underline{W}\,X\,Y\,\underline{Z} + W\,\underline{X}\,Y\,Z + W\,X\,\underline{Z} + W\,X\,Y$

c = $\underline{W}\,\underline{X}\,Y\,\underline{Z} + W\,X\,\underline{Z} + W\,X\,Y$

d = $\underline{X}\,\underline{Y}\,Z + \underline{W}\,X\,\underline{Y}\,\underline{Z} + X\,Y\,Z + W\,\underline{X}\,Y\,\underline{Z}$

e = $\underline{W}\,Z + \underline{W}\,X\,\underline{Y} + \underline{X}\,\underline{Y}\,Z$

f = $\underline{W}\,\underline{X}\,Z + \underline{W}\,\underline{X}\,Y + \underline{W}\,Y\,Z + W\,X\,\underline{Y}$

g = $\underline{W}\,\underline{X}\,\underline{Y} + \underline{W}\,X\,Y\,Z$

3. 4-bit BCD加法器

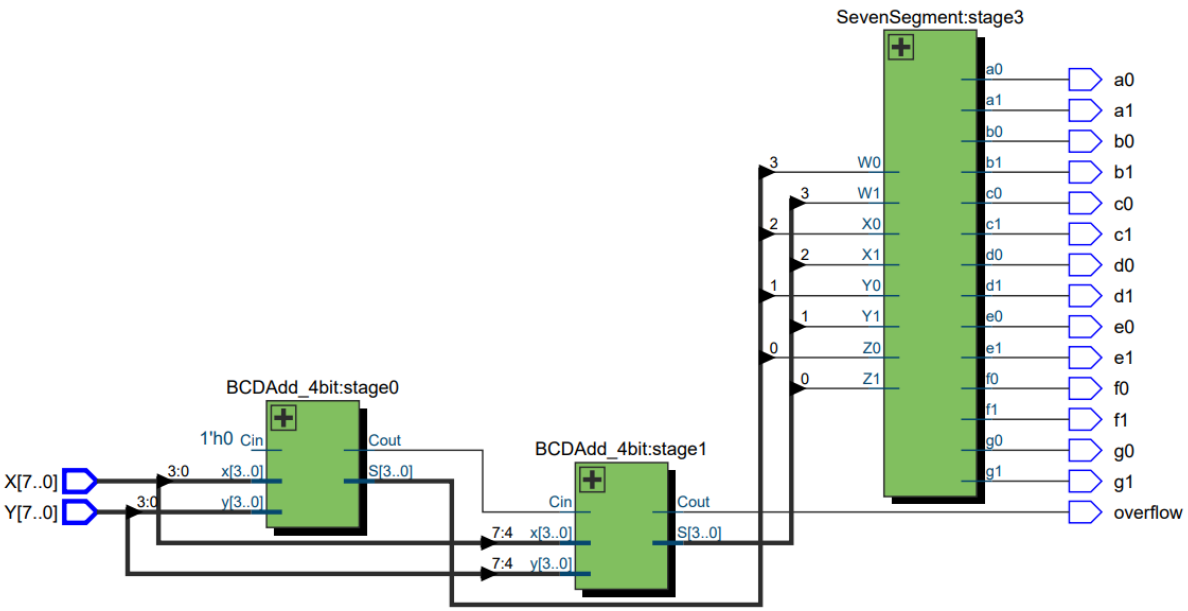
$$F = S_3\underline{S_2}\underline{S_1}\underline{S_0} + S_3\underline{S_2}S_1S_0 + S_3S_2\underline{S_1}\ \underline{S_0} + S_3\underline{S_2}S_1\underline{S_0} + S_3S_2S_1\underline{S_0} + S_3S_2S_1S_0$$

$$= S_3S_2 + S_3\underline{S_2}S_1$$

$$= S_3(S_2 + S_1)$$

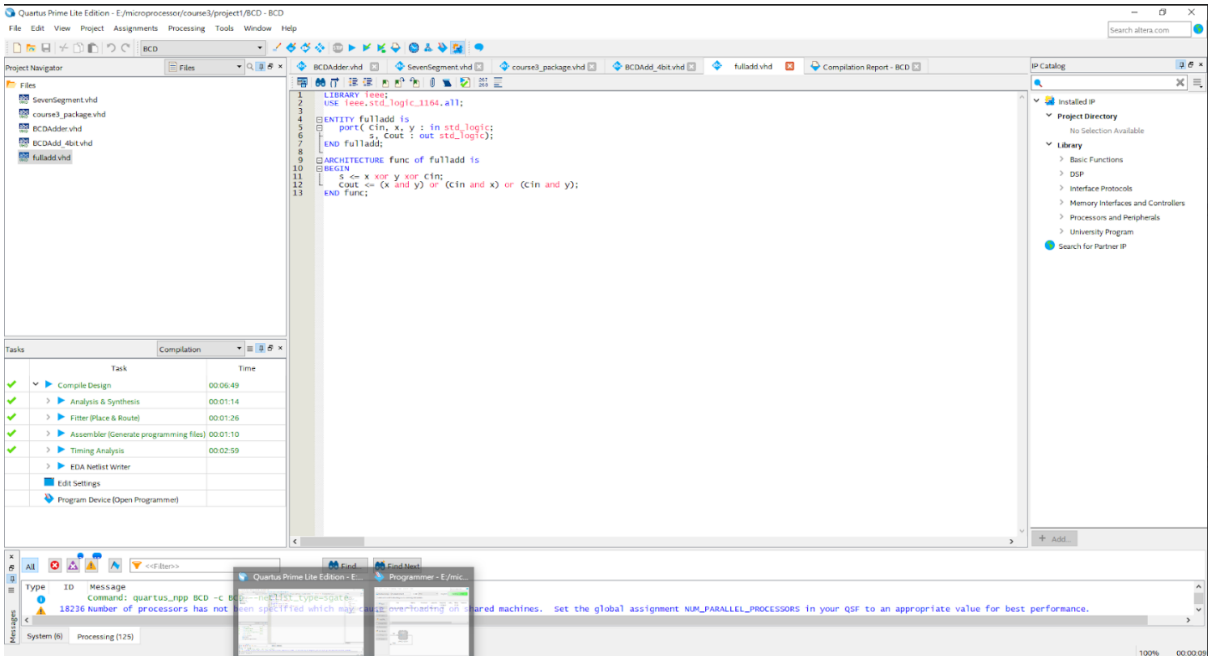
$$Y = F + Cout(4)$$

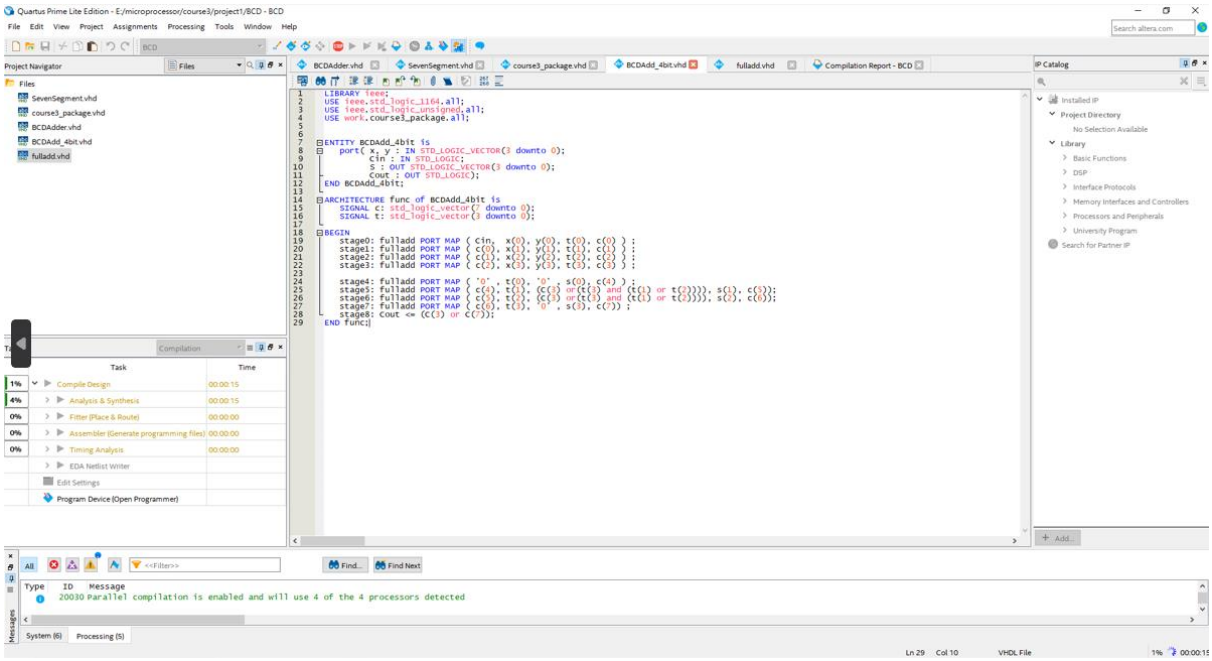
（三） 布林代數化簡後設計出的電路

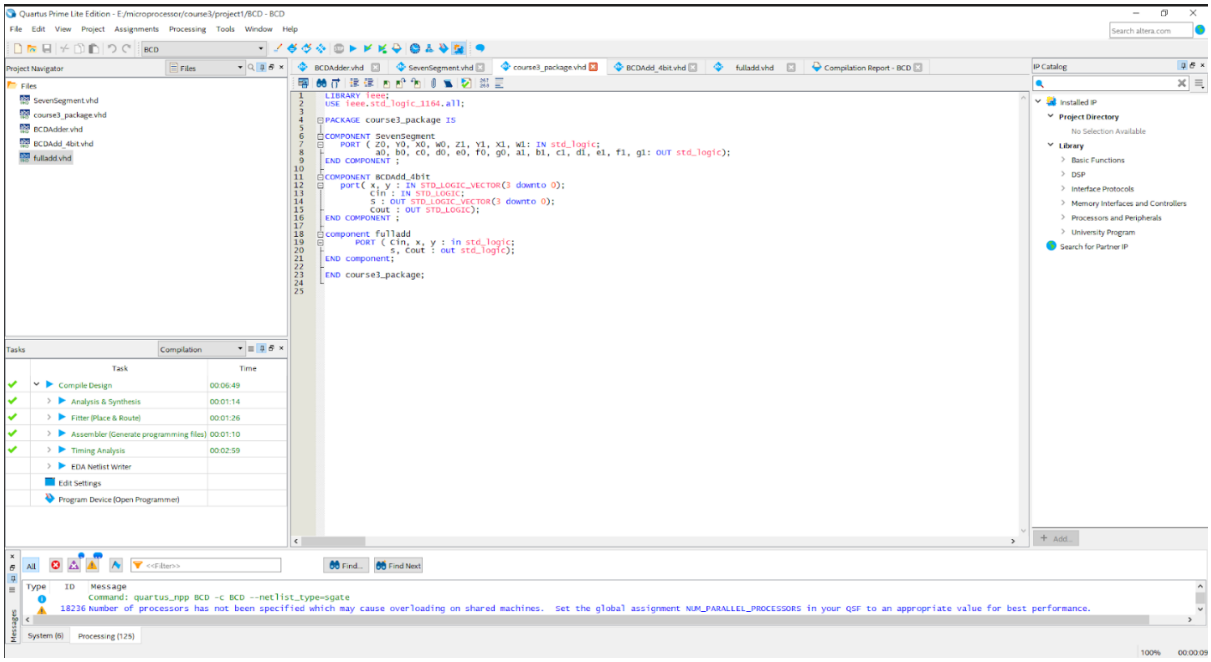


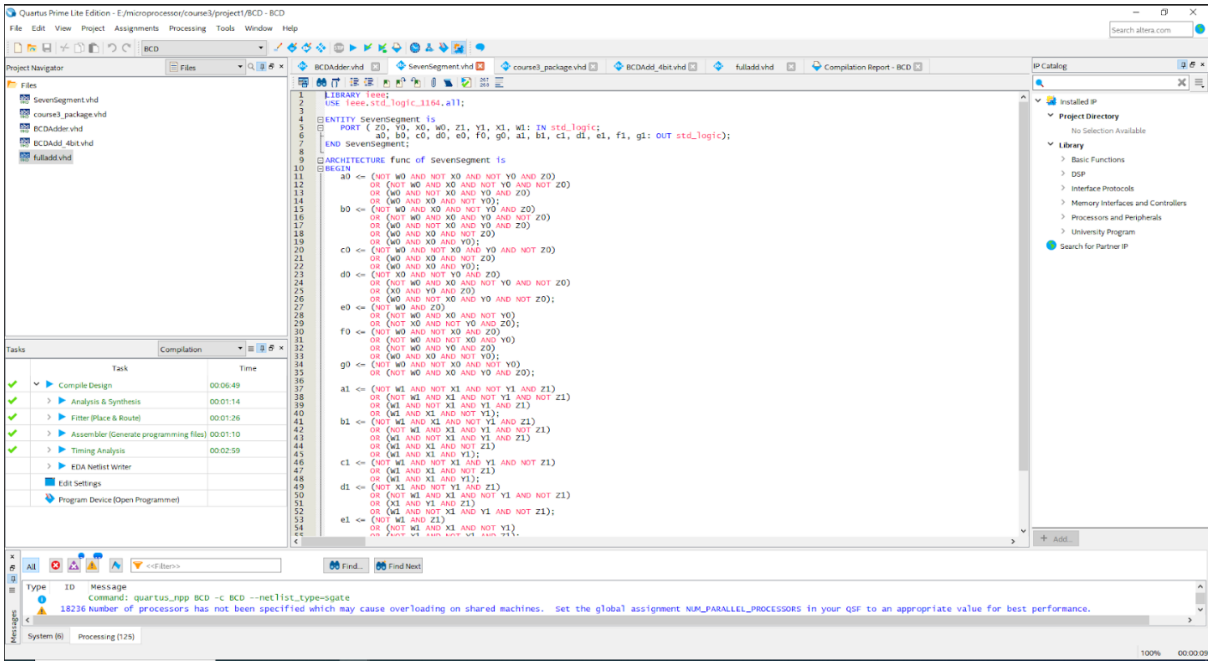
（四） 實驗過程

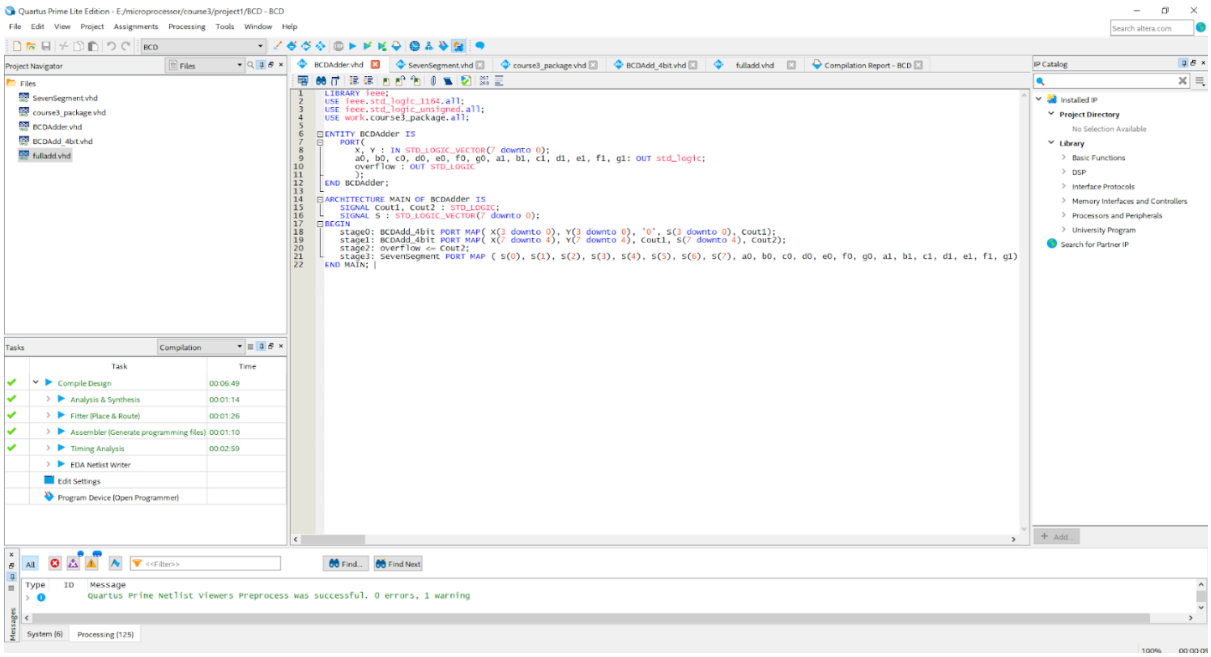
(1) 撰寫程式碼



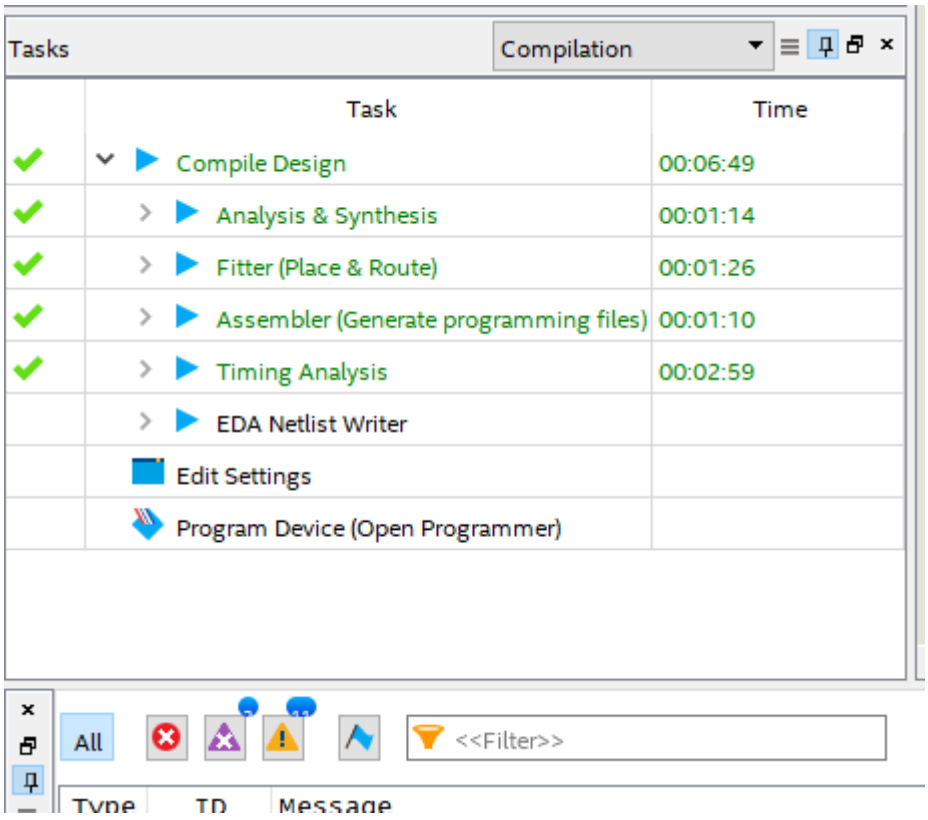



Quartus Prime Lite Edition - E:/microprocessor/course3/project1/BCD - BCD
File Edit View Project Assignments Processing Tools Window Help
Project Navigator
Files
SevenSegment.vhd
course3_package.vhd
BCDAdder.vhd
BCDAdd_4bit.vhd
fulladd.vhd
BCDAdder.vhd
SevenSegment.vhd
course3_package.vhd
BCDAdd_4bit.vhd
fulladd.vhd
Compilation Report - BCD
IP Catalog
Installed IP
Project Directory
No Selection Available
Library
Basic Functions
DSP
Interface Protocols
Memory Interfaces and Controllers
Processors and Peripherals
University Program
Search for Partner IP
Tasks
Compilation
Task
Time
Compile Design 00:06:49
Analysis & Synthesis 00:01:14
Fitter (Place & Route) 00:01:26
Assembler (Generate programming files) 00:01:10
Timing Analysis 00:02:59
EDA Netlist Writer
Edit Settings
Program Device (Open Programmer)
Find...
Find Next
Type ID Message
Command: quartus_npp BCD -c BCD --netlist_type=sgate
18236 Number of processors has not been specified which may cause overloading on shared machines. Set the global assignment NUM_PARALLEL_PROCESSORS in your QSF to an appropriate value for best performance.
System (6)
Processing (125)
100%
00:00:09

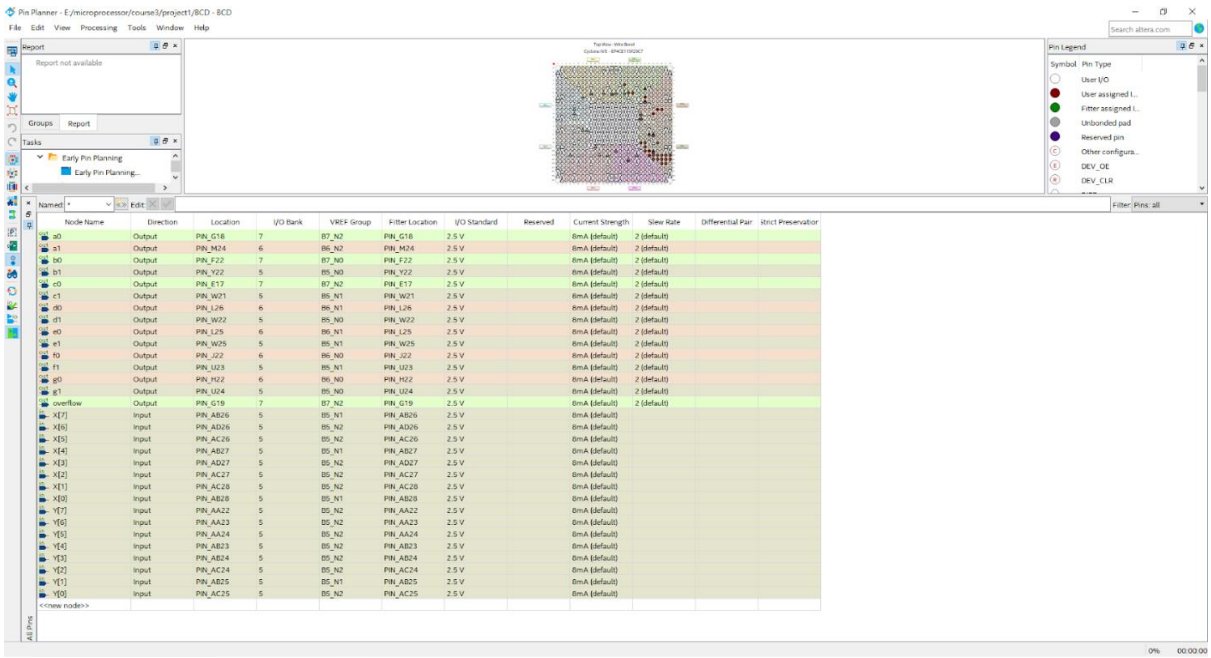

Quartus Prime Lite Edition - E:/microprocessor/course3/project1/BCD - BCD
File Edit View Project Assignments Processing Tools Window Help
Project Navigator
Files
SevenSegment.vhd
course3_package.vhd
BCDAdder.vhd
BCDAdd_4bit.vhd
fulladd.vhd
BCDAdder.vhd
SevenSegment.vhd
course3_package.vhd
BCDAdd_4bit.vhd
fulladd.vhd
Compilation Report - BCD
IP Catalog
Installed IP
Project Directory
No Selection Available
Library
Basic Functions
DSP
Interface Protocols
Memory Interfaces and Controllers
Processors and Peripherals
University Program
Search for Partner IP
Tasks
Compilation
Task
Time
Compile Design 00:06:49
Analysis & Synthesis 00:01:14
Fitter (Place & Route) 00:01:26
Assembler (Generate programming files) 00:01:10
Timing Analysis 00:02:59
EDA Netlist Writer
Edit Settings
Program Device (Open Programmer)
Find...
Find Next
Type ID Message
Command: quartus_npp BCD -c BCD --netlist_type=sgate
18236 Number of processors has not been specified which may cause overloading on shared machines. Set the global assignment NUM_PARALLEL_PROCESSORS in your QSF to an appropriate value for best performance.
System (6)
Processing (125)
100%
00:00:09


Quartus Prime Lite Edition - E:/microprocessor/course3/project1/BCD - BCD
File Edit View Project Assignments Processing Tools Window Help
Project Navigator
Files
SevenSegment.vhd
course3_package.vhd
BCDAdder.vhd
BCDAdd_4bit.vhd
fulladd.vhd
BCDAdder.vhd
SevenSegment.vhd
course3_package.vhd
BCDAdd_4bit.vhd
fulladd.vhd
Compilation Report - BCD
IP Catalog
Installed IP
Project Directory
No Selection Available
Library
Basic Functions
DSP
Interface Protocols
Memory Interfaces and Controllers
Processors and Peripherals
University Program
Search for Partner IP
LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_unsigned.all;
USE work.course3_package.all;
ENTITY BCDAdder IS
PORT(
X, Y : IN STD_LOGIC_VECTOR(7 downto 0);
a0, b0, c0, d0, e0, f0, g0, a1, b1, c1, d1, e1, f1, g1: OUT std_logic;
overflow : OUT STD_LOGIC
);
END BCDAdder;
ARCHITECTURE MAIN OF BCDAdder IS
SIGNAL Cout1, Cout2 : STD_LOGIC;
SIGNAL S : STD_LOGIC_VECTOR(7 downto 0);
BEGIN
stage0: BCDAdd_4bit PORT MAP(X(3 downto 0), Y(3 downto 0), '0', S(3 downto 0), Cout1);
stage1: BCDAdd_4bit PORT MAP(X(7 downto 4), Y(7 downto 4), Cout1, S(7 downto 4), Cout2);
stage2: overflow <= Cout2;
stage3: SevenSegment PORT MAP (S(0), S(1), S(2), S(3), S(4), S(5), S(6), S(7), a0, b0, c0, d0, e0, f0, g0, a1, b1, c1, d1, e1, f1, g1);
END MAIN;
Tasks
Compilation
Task
Time
Compile Design 00:06:49
Analysis & Synthesis 00:01:14
Fitter (Place & Route) 00:01:26
Assembler (Generate programming files) 00:01:10
Timing Analysis 00:02:59
EDA Netlist Writer
Edit Settings
Program Device (Open Programmer)
Find...
Find Next
Type ID Message
Quartus Prime Netlist Viewers Preprocess was successful. 0 errors, 1 warning
System (6)
Processing (125)

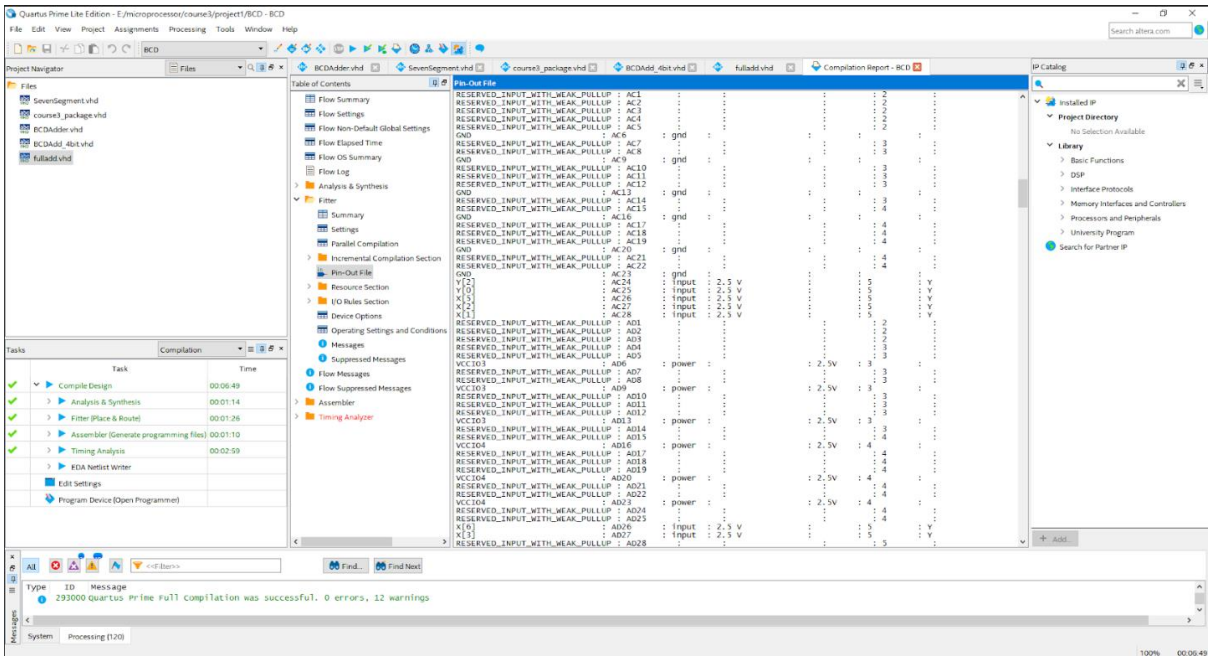
（2） 編譯成功


Tasks
Compilation
Task
Time
Compile Design 00:06:49
Analysis & Synthesis 00:01:14
Fitter (Place & Route) 00:01:26
Assembler (Generate programming files) 00:01:10
Timing Analysis 00:02:59
EDA Netlist Writer
Edit Settings
Program Device (Open Programmer)
All
<<Filter>>
Type
ID
Message

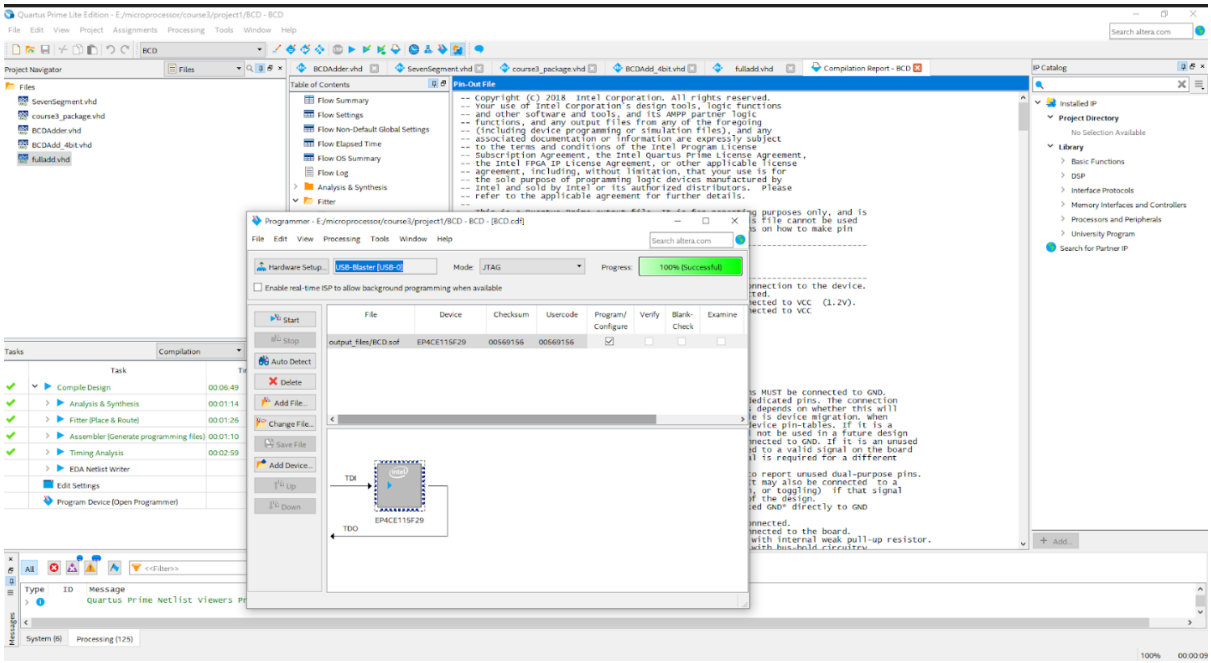
（3） 接腳位



(4)　確認接線於正確腳位



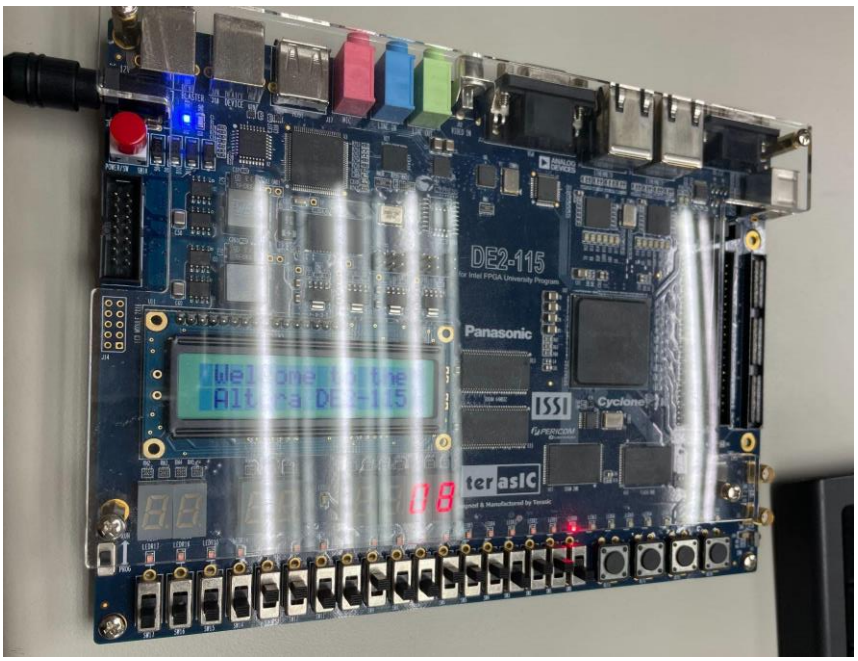
(5)　燒錄視窗設定及燒錄成功畫面



（五） 實驗結果

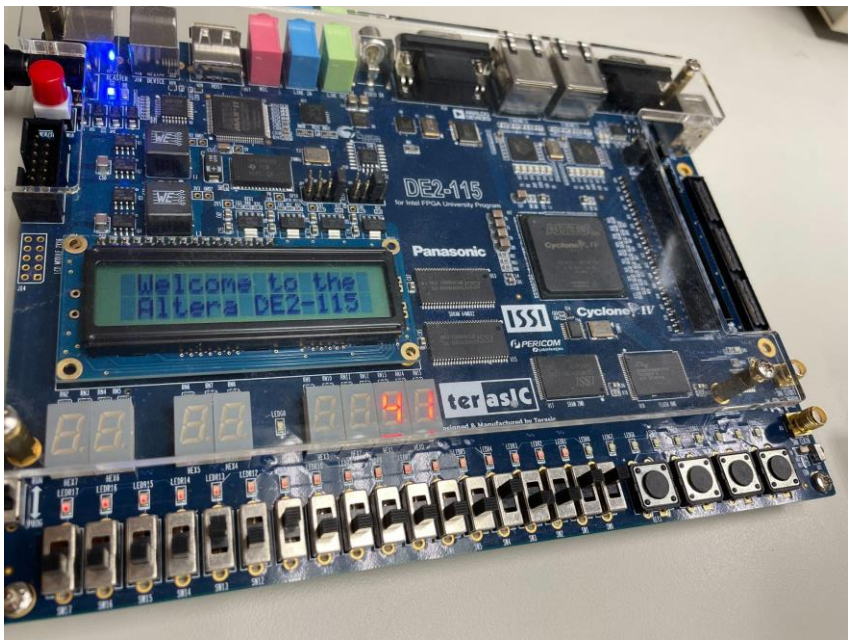
(1) 輸入 76 + 32

輸出 (1)08 → overflow



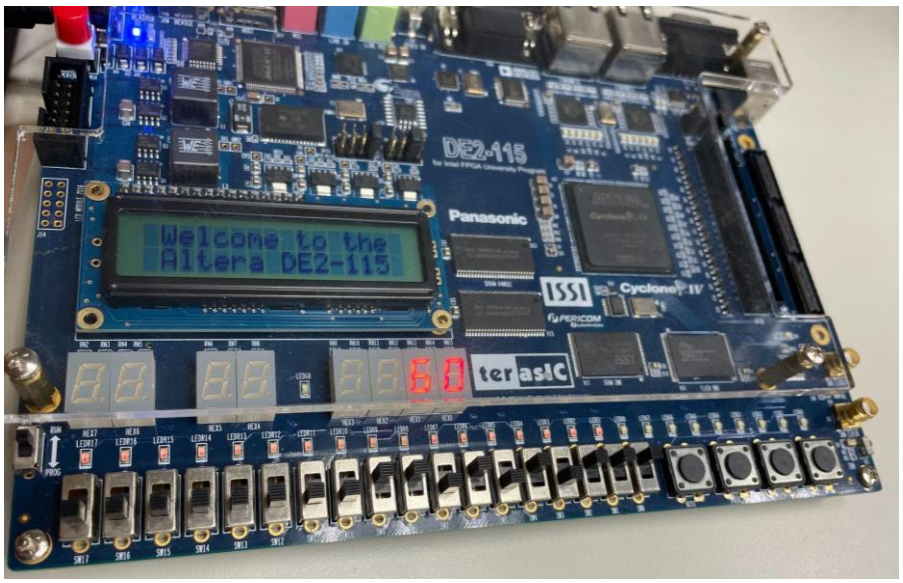
(2) 輸入 13 + 28

輸出 41



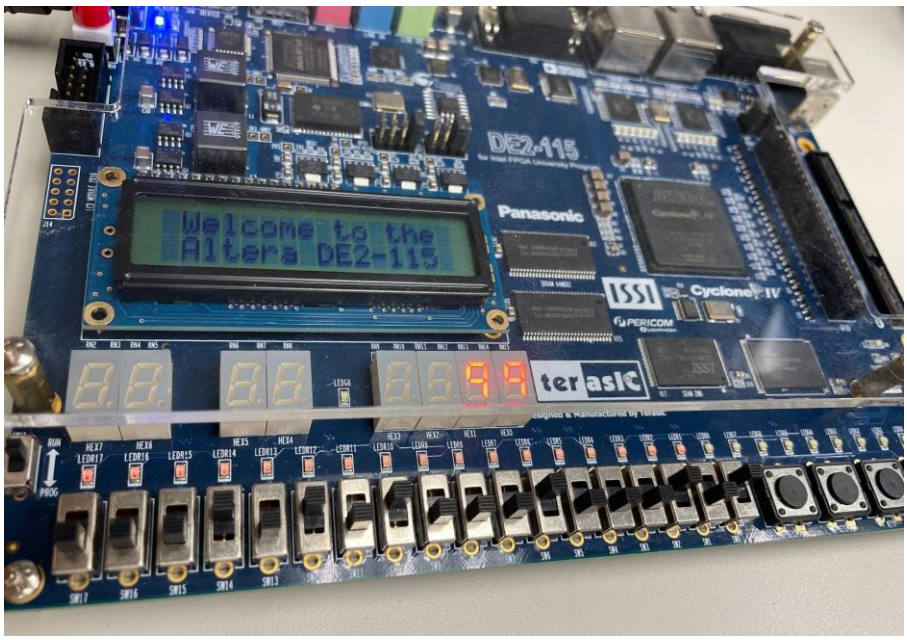
(3) 輸入 57 + 3

輸出 60



(4) 輸入 45 + 54

輸出 99



● 實際操作影片連結；https://drive.google.com/file/d/1JB8yiY5RkvYAkK6qLHuWNZrUOOiL97e5/view?usp=share_link

三、程式碼

BCDAdder.vhd

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_unsigned.all;
USE work.course3_package.all;


ENTITY BCDAdder IS
	PORT(
		X, Y : IN STD_LOGIC_VECTOR(7 downto 0);
		a0, b0, c0, d0, e0, f0, g0, a1, b1, c1, d1, e1, f1, g1: OUT std_logic;
		overflow : OUT STD_LOGIC
		);
END BCDAdder;


ARCHITECTURE MAIN OF BCDAdder IS
	SIGNAL Cout1, Cout2 : STD_LOGIC;
	SIGNAL S : STD_LOGIC_VECTOR(7 downto 0);
BEGIN
```

```
        stage0: BCDAdd_4bit PORT MAP( X(3 downto 0), Y(3 downto 0), '0', S(3 downto
0), Cout1);

        stage1: BCDAdd_4bit PORT MAP( X(7 downto 4), Y(7 downto 4), Cout1, S(7 dow
nto 4), Cout2);

        stage2: overflow <= Cout2;

        stage3: SevenSegment PORT MAP ( S(0), S(1), S(2), S(3), S(4), S(5), S(6), S(7), a0,
b0, c0, d0, e0, f0, g0, a1, b1, c1, d1, e1, f1, g1);

END MAIN;
```

BCDAdd_4bit.vhd

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_unsigned.all;
USE work.course3_package.all;



ENTITY BCDAdd_4bit is
        port( x, y : IN STD_LOGIC_VECTOR(3 downto 0);
                        Cin : IN STD_LOGIC;
                        S : OUT STD_LOGIC_VECTOR(3 downto 0);
                        Cout : OUT STD_LOGIC);
END BCDAdd_4bit;


ARCHITECTURE func of BCDAdd_4bit is
        SIGNAL c: std_logic_vector(7 downto 0);
        SIGNAL t: std_logic_vector(3 downto 0);


BEGIN
        stage0: fulladd PORT MAP ( Cin,  x(0), y(0), t(0), c(0) ) ;
        stage1: fulladd PORT MAP ( c(0), x(1), y(1), t(1), c(1) ) ;
        stage2: fulladd PORT MAP ( c(1), x(2), y(2), t(2), c(2) ) ;
```

```
        stage3: fulladd PORT MAP ( c(2), x(3), y(3), t(3), c(3) ) ;



        stage4: fulladd PORT MAP ( '0' , t(0), '0' , s(0), c(4) ) ;

        stage5: fulladd PORT MAP ( c(4), t(1), (c(3) or(t(3) and (t(1) or t(2)))), s(1), c(5));

        stage6: fulladd PORT MAP ( c(5), t(2), (c(3) or(t(3) and (t(1) or t(2)))), s(2), c(6));

        stage7: fulladd PORT MAP ( c(6), t(3), '0' , s(3), c(7)) ;

        stage8: Cout <= (c(3) or c(7));
END func;
```

fulladd.vhd

```
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 func of fulladd is

BEGIN

        s <= x xor y xor Cin;

        Cout <= (x and y) or (Cin and x) or (Cin and y);

END func;
```

SevenSegment.vhd

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY SevenSegment is
        PORT ( Z0, Y0, X0, W0, Z1, Y1, X1, W1: IN std_logic;
                        a0, b0, c0, d0, e0, f0, g0, a1, b1, c1, d1, e1, f1, g1: OUT std_logi
c);
END SevenSegment;

ARCHITECTURE func of SevenSegment is
BEGIN
        a0 <= (NOT W0 AND NOT X0 AND NOT Y0 AND Z0)
                        OR (NOT W0 AND X0 AND NOT Y0 AND NOT Z0)
                        OR (W0 AND NOT X0 AND Y0 AND Z0)
                        OR (W0 AND X0 AND NOT Y0);
        b0 <= (NOT W0 AND X0 AND NOT Y0 AND Z0)
                        OR (NOT W0 AND X0 AND Y0 AND NOT Z0)
                        OR (W0 AND NOT X0 AND Y0 AND Z0)
                        OR (W0 AND X0 AND NOT Z0)
                        OR (W0 AND X0 AND Y0);
        c0 <= (NOT W0 AND NOT X0 AND Y0 AND NOT Z0)
```

```
                OR (W0 AND X0 AND NOT Z0)
                OR (W0 AND X0 AND Y0);
d0 <= (NOT X0 AND NOT Y0 AND Z0)
                OR (NOT W0 AND X0 AND NOT Y0 AND NOT Z0)
                OR (X0 AND Y0 AND Z0)
                OR (W0 AND NOT X0 AND Y0 AND NOT Z0);
e0 <= (NOT W0 AND Z0)
                OR (NOT W0 AND X0 AND NOT Y0)
                OR (NOT X0 AND NOT Y0 AND Z0);
f0 <= (NOT W0 AND NOT X0 AND Z0)
                OR (NOT W0 AND NOT X0 AND Y0)
                OR (NOT W0 AND Y0 AND Z0)
                OR (W0 AND X0 AND NOT Y0);
g0 <= (NOT W0 AND NOT X0 AND NOT Y0)
                OR (NOT W0 AND X0 AND Y0 AND Z0);


a1 <= (NOT W1 AND NOT X1 AND NOT Y1 AND Z1)
                OR (NOT W1 AND X1 AND NOT Y1 AND NOT Z1)
                OR (W1 AND NOT X1 AND Y1 AND Z1)
                OR (W1 AND X1 AND NOT Y1);
b1 <= (NOT W1 AND X1 AND NOT Y1 AND Z1)
```

```
                    OR (NOT W1 AND X1 AND Y1 AND NOT Z1)
                    OR (W1 AND NOT X1 AND Y1 AND Z1)
                    OR (W1 AND X1 AND NOT Z1)
                    OR (W1 AND X1 AND Y1);
    c1 <= (NOT W1 AND NOT X1 AND Y1 AND NOT Z1)
                    OR (W1 AND X1 AND NOT Z1)
                    OR (W1 AND X1 AND Y1);
    d1 <= (NOT X1 AND NOT Y1 AND Z1)
                    OR (NOT W1 AND X1 AND NOT Y1 AND NOT Z1)
                    OR (X1 AND Y1 AND Z1)
                    OR (W1 AND NOT X1 AND Y1 AND NOT Z1);
    e1 <= (NOT W1 AND Z1)
                    OR (NOT W1 AND X1 AND NOT Y1)
                    OR (NOT X1 AND NOT Y1 AND Z1);
    f1 <= (NOT W1 AND NOT X1 AND Z1)
                    OR (NOT W1 AND NOT X1 AND Y1)
                    OR (NOT W1 AND Y1 AND Z1)
                    OR (W1 AND X1 AND NOT Y1);
    g1 <= (NOT W1 AND NOT X1 AND NOT Y1)
                    OR (NOT W1 AND X1 AND Y1 AND Z1);
END func;
```

package. vhd

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;

PACKAGE course3_package IS

COMPONENT SevenSegment
        PORT ( Z0, Y0, X0, W0, Z1, Y1, X1, W1: IN std_logic;
                        a0, b0, c0, d0, e0, f0, g0, a1, b1, c1, d1, e1, f1, g1: OUT std_logi
c);
END COMPONENT ;

COMPONENT BCDAdd_4bit
        port( x, y : IN STD_LOGIC_VECTOR(3 downto 0);
                        Cin : IN STD_LOGIC;
                        S : OUT STD_LOGIC_VECTOR(3 downto 0);
                        Cout : OUT STD_LOGIC);
END COMPONENT ;

component fulladd
                PORT ( Cin, x, y : in std_logic;
                                s, Cout : out std_logic);
```

```
END component;


END course3_package;
```