

微算機系統

小組專案報告

實驗二：

多位元加法器 & 多位元減法器設計

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：112.10.05

一、實驗內容：

（一）目標一

1. 以1-bit之全加器為基礎，將上述之8-bit多位元加法器邏輯函數以 Package 與 component 語法包裝，之後以實驗板上兩顆七段顯示器及LED燈來呈現結果。
2. 測試時，將以指撥開關輸入，而兩顆七段顯示器上必須可正確顯示出運算的結果（十六進位），結果皆為正整數，範圍為0~FF（需偵測OverFlow，OverFlow時亮LED燈）。
3. 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。

（二）目標二

1. 以1-bit之全加器為基礎，將上述之8-bit多位元減法器邏輯函數以 Package 與 component 語法包裝，之後以實驗板上兩顆七段顯示器及LED燈來呈現結果。
2. 測試時，將以指撥開關輸入，而兩顆七段顯示器上必須可正確顯示出運算的結果（十六進位），結果皆為正整數，
3. 範圍為0~FF（需偵測OverFlow，OverFlow時亮LED燈）。
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
A	1	0	1	0	0	0	0	1	0	0	0
B	1	0	1	1	1	1	0	0	0	0	0
C	1	1	0	0	1	1	1	0	0	1	0
D	1	1	0	1	1	0	0	0	0	1	0
E	1	1	1	0	0	1	1	0	0	0	0
F	1	1	1	1	0	1	1	1	0	0	0

(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~FF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. 減法器

減法器以全加器為基礎，將y取一的補數(y加上not)，且初始Cin為1
(以達到2的補數之效果)

(1)若c[7]為1，則有end-carry，計算結果為正數，因此取c[7]，判別為無overflow(0~FF)

(2)若c[7]為0，則無end-carry，計算結果為負數，因此取c[7]，判別為有overflow(不在0~FF範圍內)

3. 七段顯示器

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

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

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

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

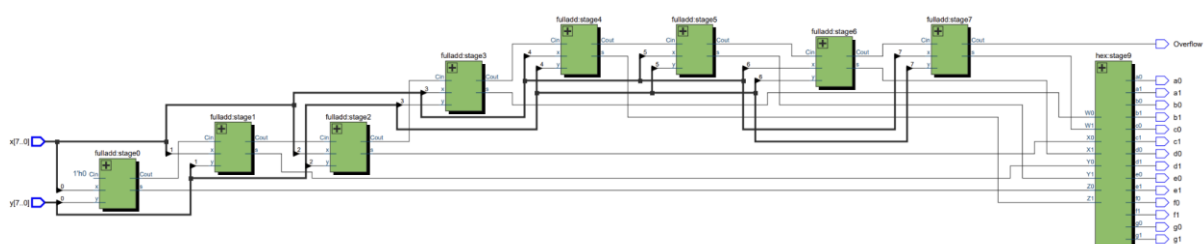
$$e = \underline{\underline{W}} \underline{\underline{Z}} + \underline{W} \underline{X} \underline{\underline{Y}} + \underline{\underline{X}} \underline{\underline{Y}} \underline{\underline{Z}}$$

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

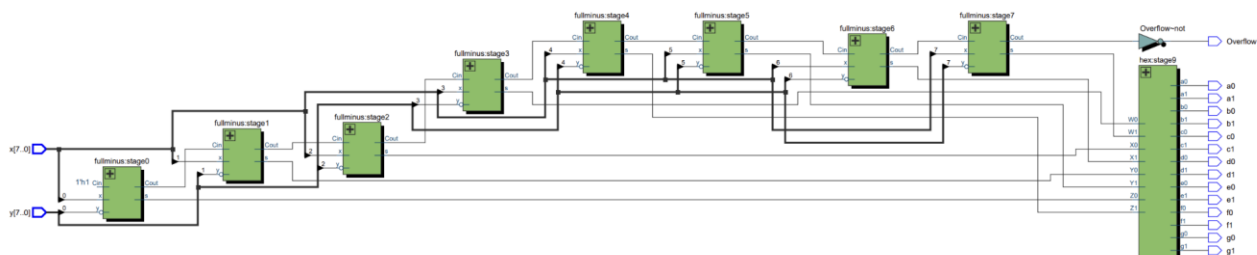
$$g = \underline{W}\underline{X}\underline{Y} + \underline{W}\underline{X}\underline{Y}\underline{Z}$$

(三) 布林代數化簡後設計出的電路

1. 目標一



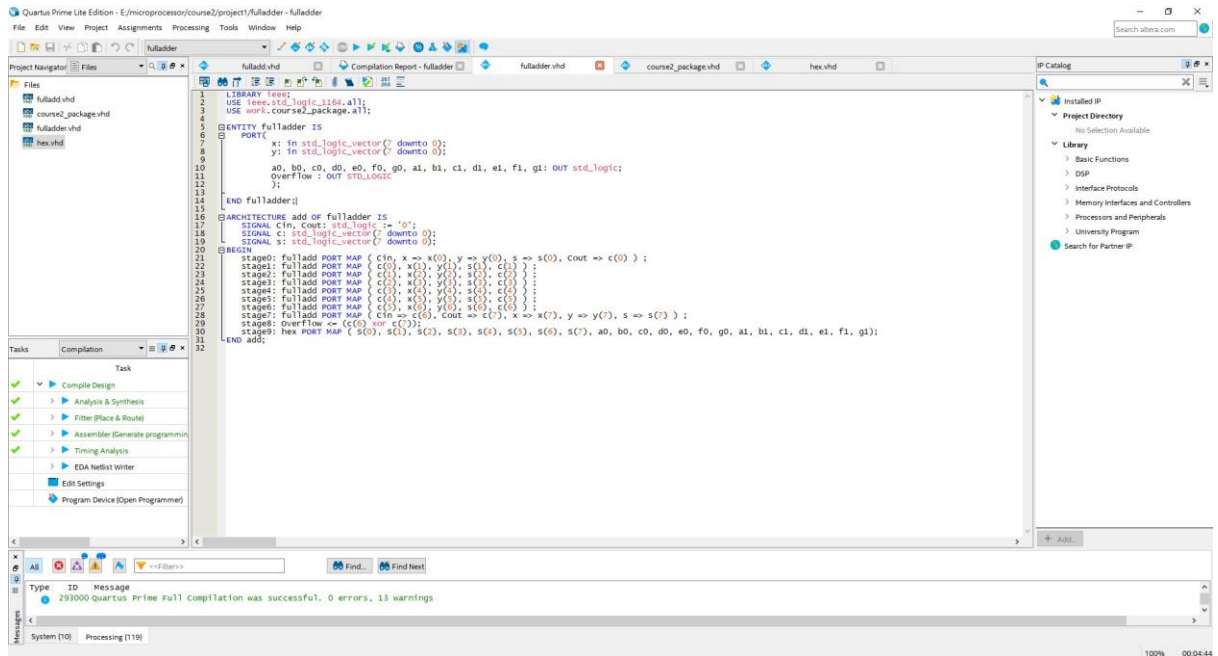
2. 目標二



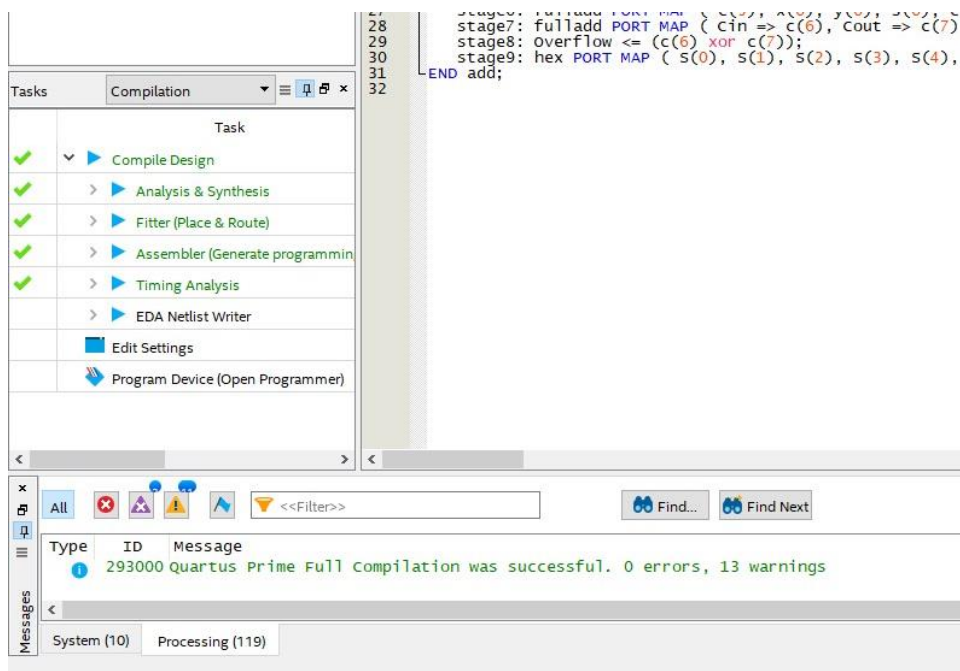
(四) 實驗過程

1. 目標1

(1) 撰寫程式碼



(2) 編譯成功



(3) 接腳位

Pin Planner - E:\microprocessor\course2\project\fulladder - fulladder

Report not available

Groups Report

Tasks

Early Pin Planning

Early Pin Planning

Pin Legend

Symbol Pin Type

User I/O

User assigned L

Filter assigned L

Unbonded pad

Reserved pin

Other configura...

DEV_OE

DEV_CLB

Filter: Pins: all

Node Name	Direction	Location	I/O Bank	VREF Group	Filter Location	I/O Standard	Reserved	Current Strength	Slew Rate	Differential Pair	Strict Preservation
a0	Output	PN_G18	7	B7_N2	PN_G18	2.5 V		8mA (default)	2 (default)		
a1	Output	PN_M24	6	B6_N2	PN_M24	2.5 V		8mA (default)	2 (default)		
b0	Output	PN_F22	7	B7_N0	PN_F22	2.5 V		8mA (default)	2 (default)		
b1	Output	PN_V22	5	B5_N0	PN_V22	2.5 V		8mA (default)	2 (default)		
c0	Output	PN_E17	7	B7_N2	PN_E17	2.5 V		8mA (default)	2 (default)		
c1	Output	PN_W21	5	B5_N1	PN_W21	2.5 V		8mA (default)	2 (default)		
d0	Output	PN_L26	6	B6_N1	PN_L26	2.5 V		8mA (default)	2 (default)		
d1	Output	PN_V22	5	B5_N0	PN_V22	2.5 V		8mA (default)	2 (default)		
e0	Output	PN_L26	6	B6_N1	PN_L26	2.5 V		8mA (default)	2 (default)		
e1	Output	PN_W25	5	B5_N1	PN_W25	2.5 V		8mA (default)	2 (default)		
f0	Output	PN_J22	6	B6_N0	PN_J22	2.5 V		8mA (default)	2 (default)		
f1	Output	PN_U23	5	B5_N1	PN_U23	2.5 V		8mA (default)	2 (default)		
g0	Output	PN_N22	6	B6_N0	PN_N22	2.5 V		8mA (default)	2 (default)		
g1	Output	PN_U24	5	B5_N0	PN_U24	2.5 V		8mA (default)	2 (default)		
Overflow	Output	PN_G19	7	B7_N2	PN_G19	2.5 V		8mA (default)	2 (default)		
x[7]	Input	PN_AB26	5	B5_N1	PN_AB26	2.5 V		8mA (default)			
x[6]	Input	PN_AD26	5	B5_N1	PN_AD26	2.5 V		8mA (default)			
x[5]	Input	PN_AC26	5	B5_N2	PN_AC26	2.5 V		8mA (default)			
x[4]	Input	PN_AB27	5	B5_N1	PN_AB27	2.5 V		8mA (default)			
x[3]	Input	PN_AD27	5	B5_N2	PN_AD27	2.5 V		8mA (default)			
x[2]	Input	PN_AC27	5	B5_N2	PN_AC27	2.5 V		8mA (default)			
x[1]	Input	PN_AC28	5	B5_N2	PN_AC28	2.5 V		8mA (default)			
x[0]	Input	PN_AB28	5	B5_N1	PN_AB28	2.5 V		8mA (default)			
y[7]	Input	PN_AA22	5	B5_N2	PN_AA22	2.5 V		8mA (default)			
y[6]	Input	PN_AA23	5	B5_N2	PN_AA23	2.5 V		8mA (default)			
y[5]	Input	PN_AA24	5	B5_N2	PN_AA24	2.5 V		8mA (default)			
y[4]	Input	PN_AB23	5	B5_N2	PN_AB23	2.5 V		8mA (default)			
y[3]	Input	PN_AB24	5	B5_N2	PN_AB24	2.5 V		8mA (default)			
y[2]	Input	PN_AC24	5	B5_N2	PN_AC24	2.5 V		8mA (default)			
y[1]	Input	PN_AB25	5	B5_N1	PN_AB25	2.5 V		8mA (default)			
y[0]	Input	PN_AC25	5	B5_N2	PN_AC25	2.5 V		8mA (default)			

All Pins

0%

00:00:00

(4) 確認接線於正確腳位

Quartus Prime Lite Edition - E:\microprocessor\course2\project\fulladder - fulladder

File Edit View Project Assignments Processing Tools Window Help

Project Navigator

Files

fulladder.vhd

course2_package.vhd

fulladder.vhd

hex.vhd

Table of Contents

Flow Summary

Flow Settings

Flow Non-Default Global Settings

Flow Elapsed Time

Flow OS Summary

Flow Log

Analysis & Synthesis

Filter

Summary

Settings

Parallel Compilation

Incremental Compilation Section

Pin-Out File

Resource Section

I/O Rules Section

Device Options

Operating Settings and Conditions

Messages

Suppressed Messages

Flow Messages

Flow Suppressed Messages

Assembler

Timing Analyzer

Compilation Report - fulladder

Pin	Signal	Direction	Strength	Speed	Power
AB17	RESERVED_INPUT_WITH_WEAK_PULLUP	Input	2.5 V	4	
AB18	RESERVED_INPUT_WITH_WEAK_PULLUP	Input	2.5 V	4	
AB19	RESERVED_INPUT_WITH_WEAK_PULLUP	Input	2.5 V	4	
AB20	RESERVED_INPUT_WITH_WEAK_PULLUP	Input	2.5 V	4	
AB21	RESERVED_INPUT_WITH_WEAK_PULLUP	Input	2.5 V	4	
AB22	RESERVED_INPUT_WITH_WEAK_PULLUP	Input	2.5 V	4	
AB23	RESERVED_INPUT_WITH_WEAK_PULLUP	Input	2.5 V	4	
AB24	RESERVED_INPUT_WITH_WEAK_PULLUP	Input	2.5 V	4	
AB25	RESERVED_INPUT_WITH_WEAK_PULLUP	Input	2.5 V	4	
AB26	RESERVED_INPUT_WITH_WEAK_PULLUP	Input	2.5 V	4	
AB27	RESERVED_INPUT_WITH_WEAK_PULLUP	Input	2.5 V	4	
AB28	RESERVED_INPUT_WITH_WEAK_PULLUP	Input	2.5 V	4	
AC1	RESERVED_INPUT_WITH_WEAK_PULLUP	Input	2.5 V	2	
AC2	RESERVED_INPUT_WITH_WEAK_PULLUP	Input	2.5 V	2	
AC3	RESERVED_INPUT_WITH_WEAK_PULLUP	Input	2.5 V	2	
AC4	RESERVED_INPUT_WITH_WEAK_PULLUP	Input	2.5 V	2	
AC5	RESERVED_INPUT_WITH_WEAK_PULLUP	Input	2.5 V	2	
AC6	RESERVED_INPUT_WITH_WEAK_PULLUP	Input	2.5 V	2	
AC7	RESERVED_INPUT_WITH_WEAK_PULLUP	Input	2.5 V	2	
AC8	RESERVED_INPUT_WITH_WEAK_PULLUP	Input	2.5 V	2	
AC9	RESERVED_INPUT_WITH_WEAK_PULLUP	Input	2.5 V	2	
AC10	RESERVED_INPUT_WITH_WEAK_PULLUP	Input	2.5 V	2	
AC11	RESERVED_INPUT_WITH_WEAK_PULLUP	Input	2.5 V	2	
AC12	RESERVED_INPUT_WITH_WEAK_PULLUP	Input	2.5 V	2	
AC13	RESERVED_INPUT_WITH_WEAK_PULLUP	Input	2.5 V	2	
AC14	RESERVED_INPUT_WITH_WEAK_PULLUP	Input	2.5 V	2	
AC15	RESERVED_INPUT_WITH_WEAK_PULLUP	Input	2.5 V	2	
AC16	RESERVED_INPUT_WITH_WEAK_PULLUP	Input	2.5 V	2	
AC17	RESERVED_INPUT_WITH_WEAK_PULLUP	Input	2.5 V	2	
AC18	RESERVED_INPUT_WITH_WEAK_PULLUP	Input	2.5 V	2	
AC19	RESERVED_INPUT_WITH_WEAK_PULLUP	Input	2.5 V	2	
AC20	RESERVED_INPUT_WITH_WEAK_PULLUP	Input	2.5 V	2	
AC21	RESERVED_INPUT_WITH_WEAK_PULLUP	Input	2.5 V	2	
AC22	RESERVED_INPUT_WITH_WEAK_PULLUP	Input	2.5 V	2	
AD1	RESERVED_INPUT_WITH_WEAK_PULLUP	Input	2.5 V	2	
AD2	RESERVED_INPUT_WITH_WEAK_PULLUP	Input	2.5 V	2	
AD3	RESERVED_INPUT_WITH_WEAK_PULLUP	Input	2.5 V	2	
AD4	RESERVED_INPUT_WITH_WEAK_PULLUP	Input	2.5 V	2	
AD5	RESERVED_INPUT_WITH_WEAK_PULLUP	Input	2.5 V	2	
AD6	RESERVED_INPUT_WITH_WEAK_PULLUP	Input	2.5 V	2	
AD7	RESERVED_INPUT_WITH_WEAK_PULLUP	Input	2.5 V	2	
AD8	RESERVED_INPUT_WITH_WEAK_PULLUP	Input	2.5 V	2	
AD9	RESERVED_INPUT_WITH_WEAK_PULLUP	Input	2.5 V	2	
AD10	RESERVED_INPUT_WITH_WEAK_PULLUP	Input	2.5 V	2	
AD11	RESERVED_INPUT_WITH_WEAK_PULLUP	Input	2.5 V	2	

Messages

332240 No Minimum Pulse width paths to report

332202 Design is not fully constrained for setup requirements

332202 Design is not fully constrained for hold requirements

Quartus Prime Timing analyzer was successful. 0 errors, 5 warnings

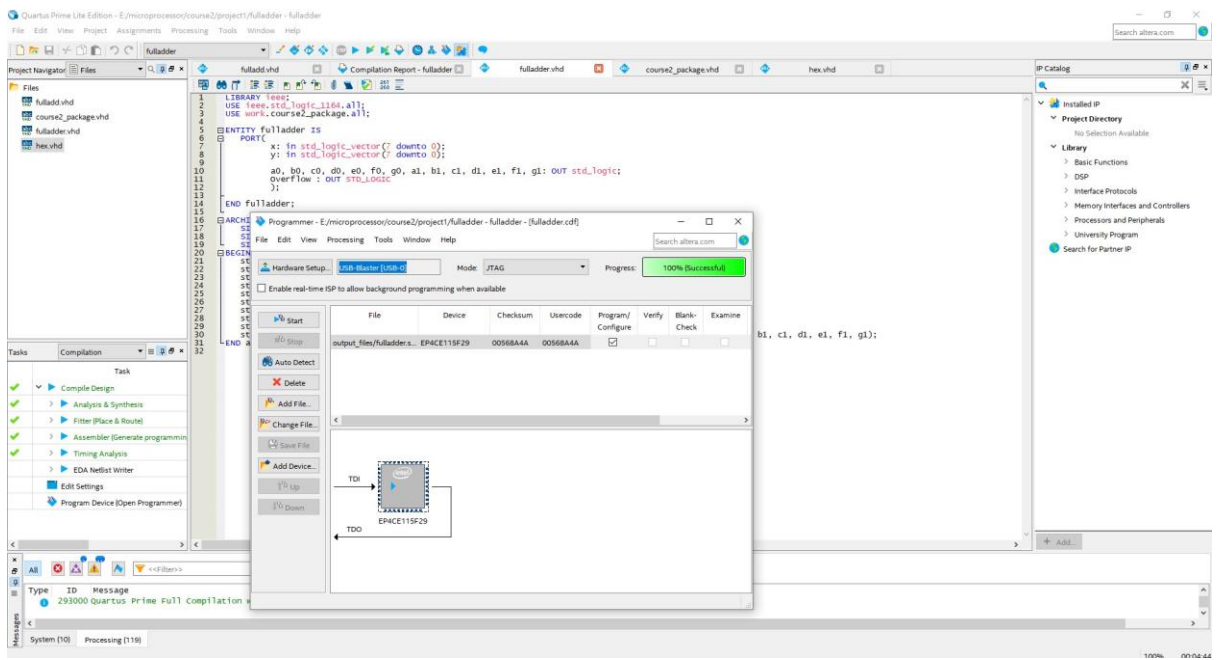
293000 Quartus Prime Full Compilation was successful. 0 errors, 13 warnings

System Processing (118)

100%

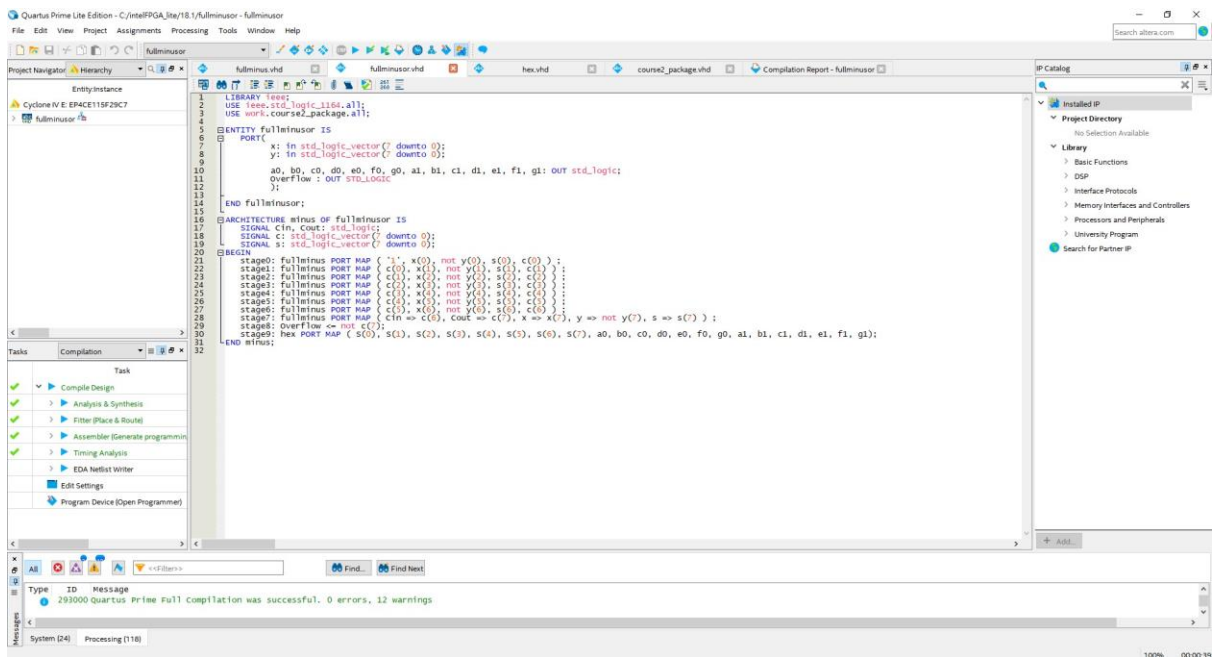
00:02:33

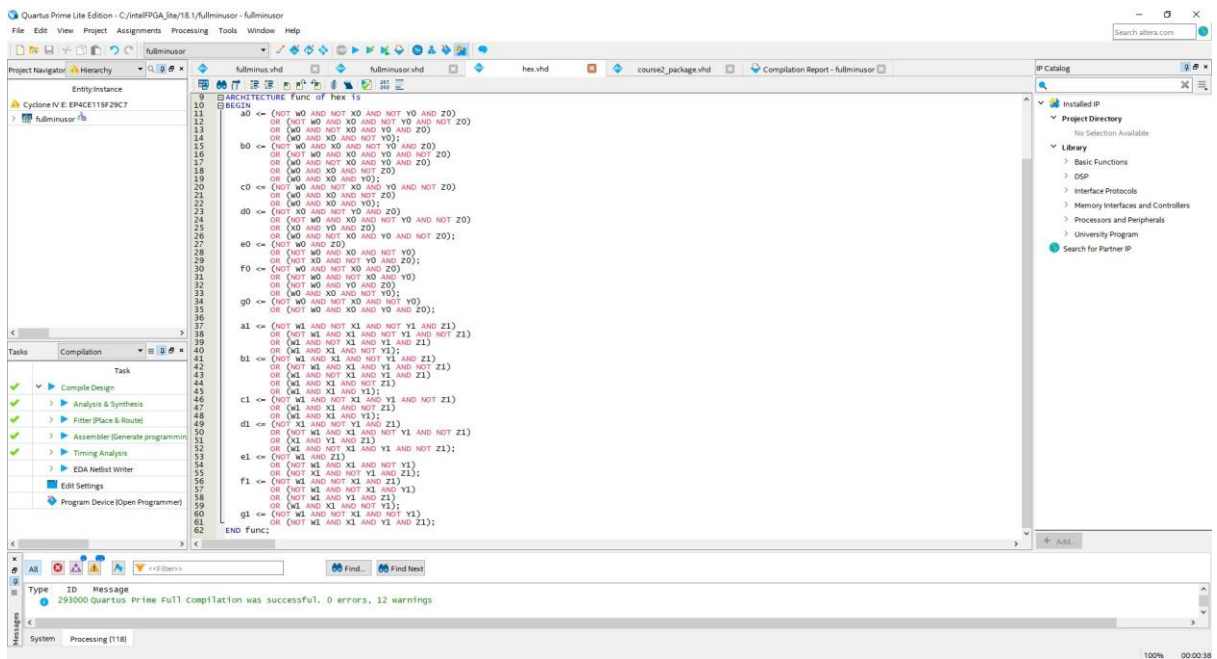
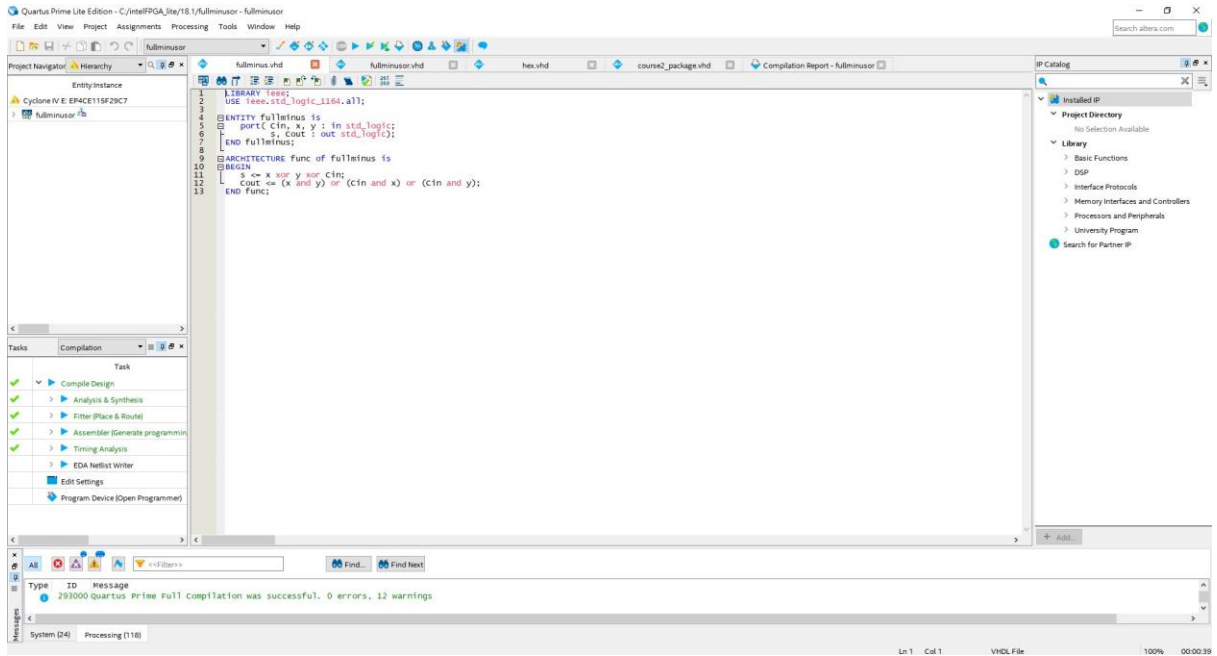
(5) 燒錄視窗設定及燒錄成功畫面

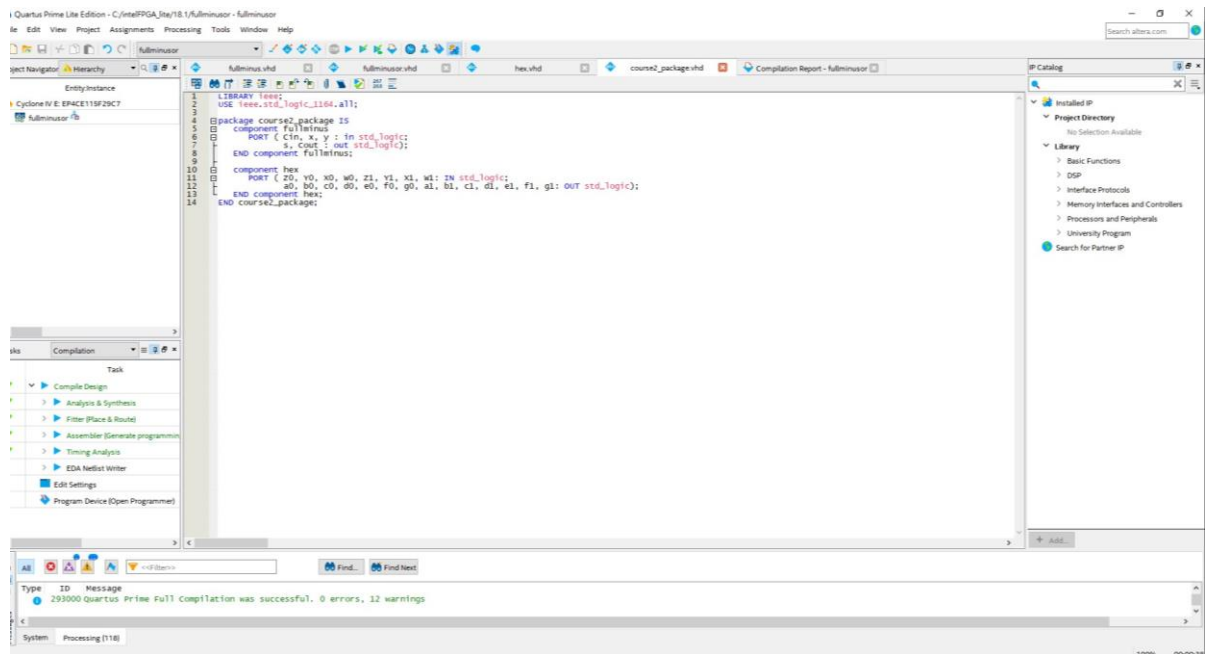


2. 目標2

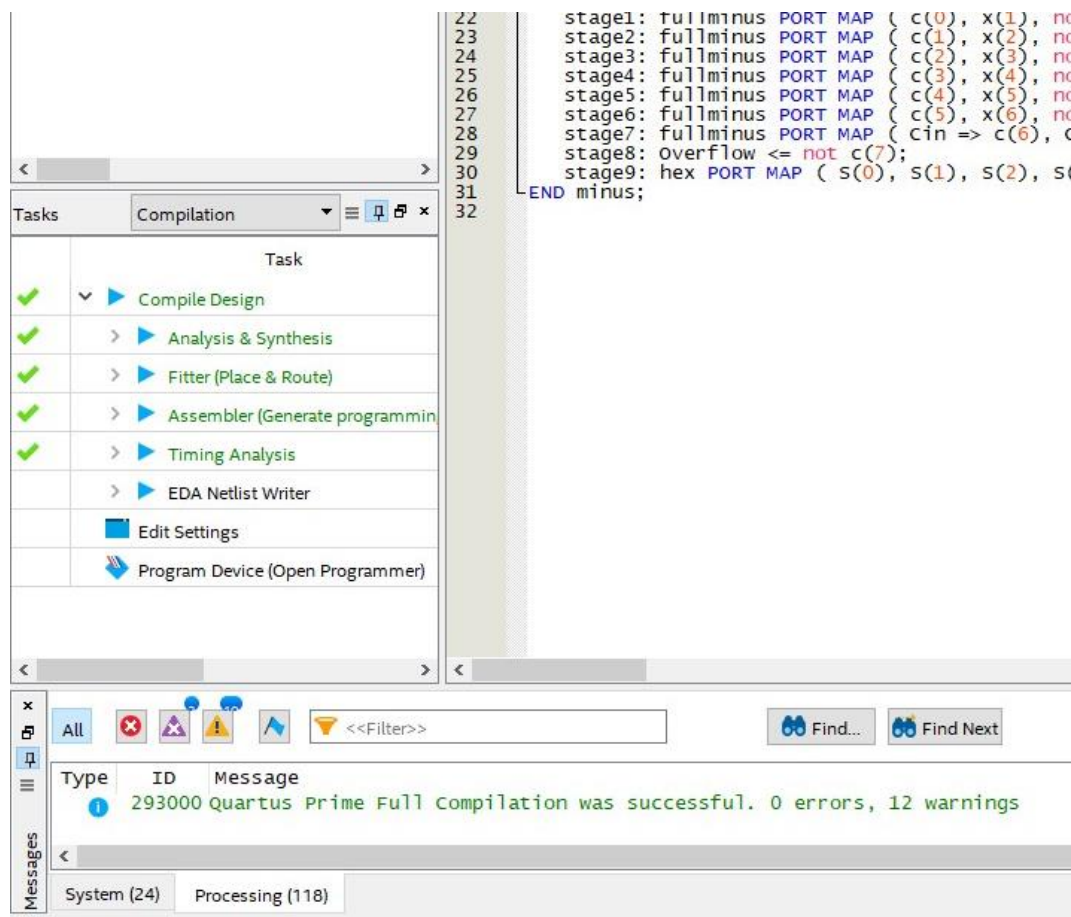
(1) 撰寫程式碼



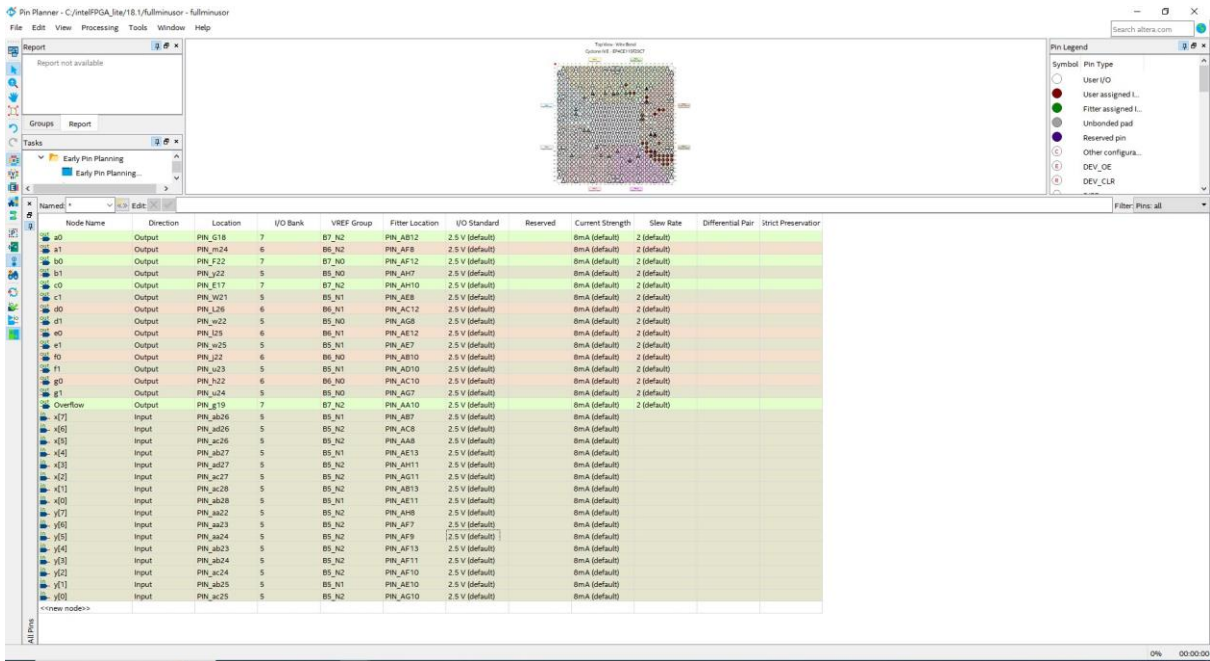




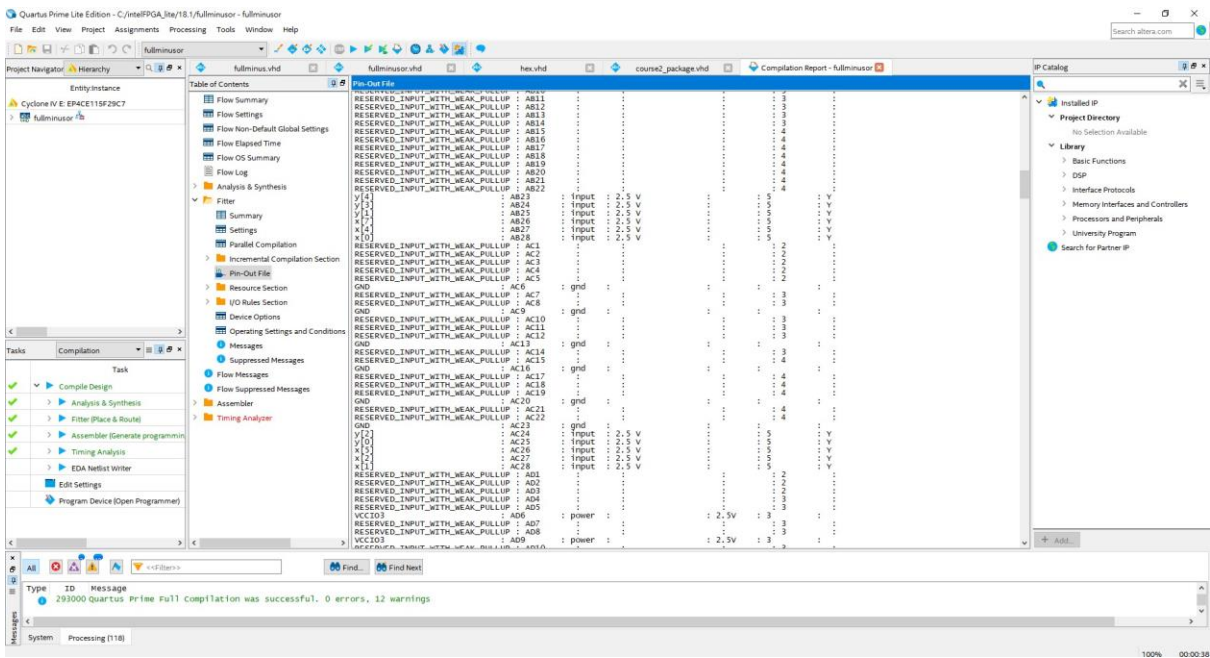
(2) 編譯成功



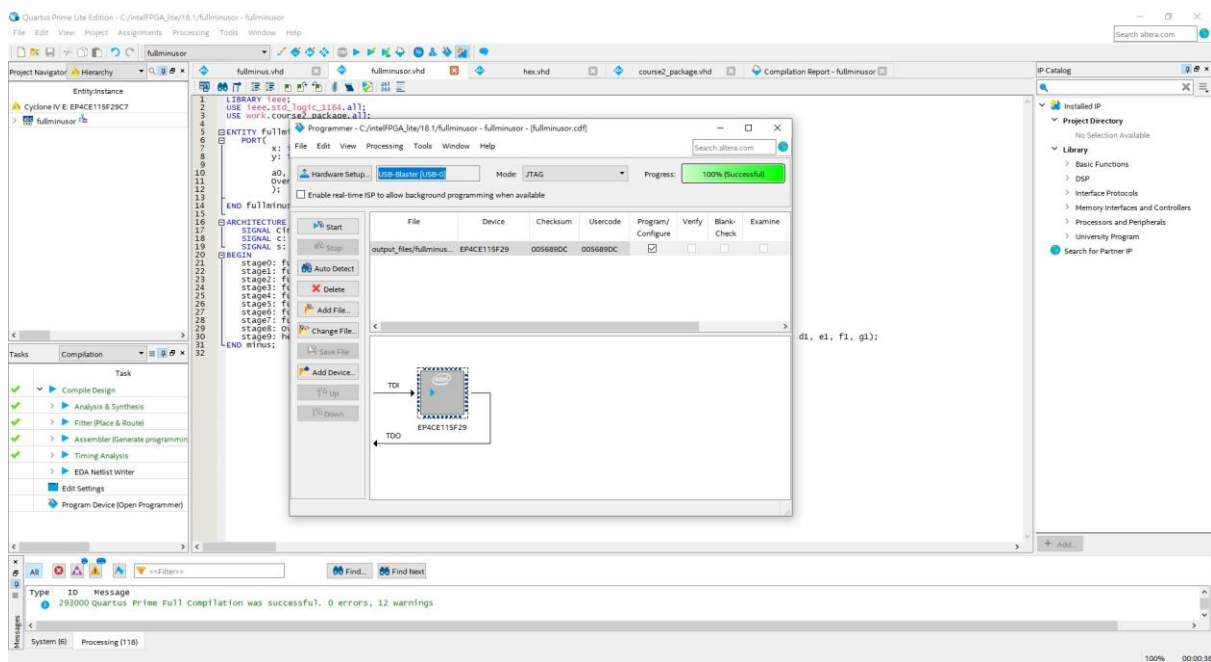
(3) 接腳位



(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. 目標一

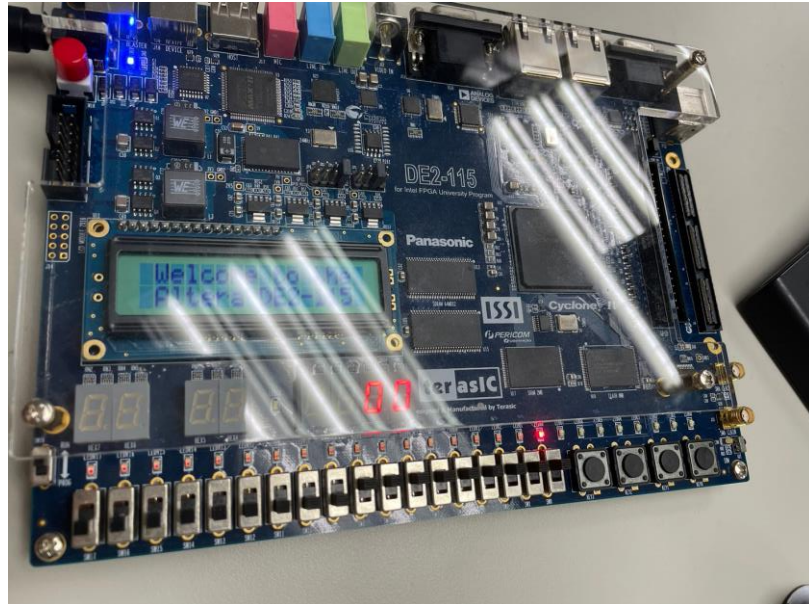
(1) 輸入 $8 + 9$

輸出 11



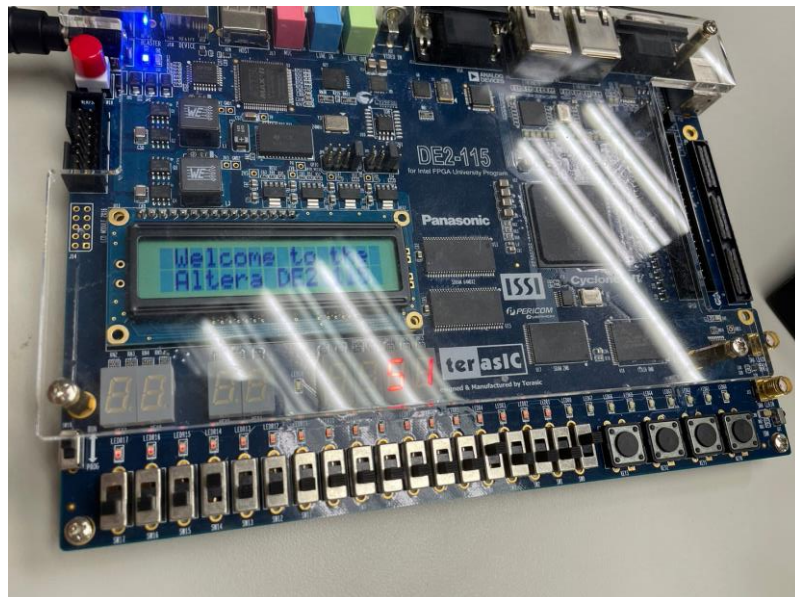
(2) 輸入 $255 + 1$

輸出 overflow



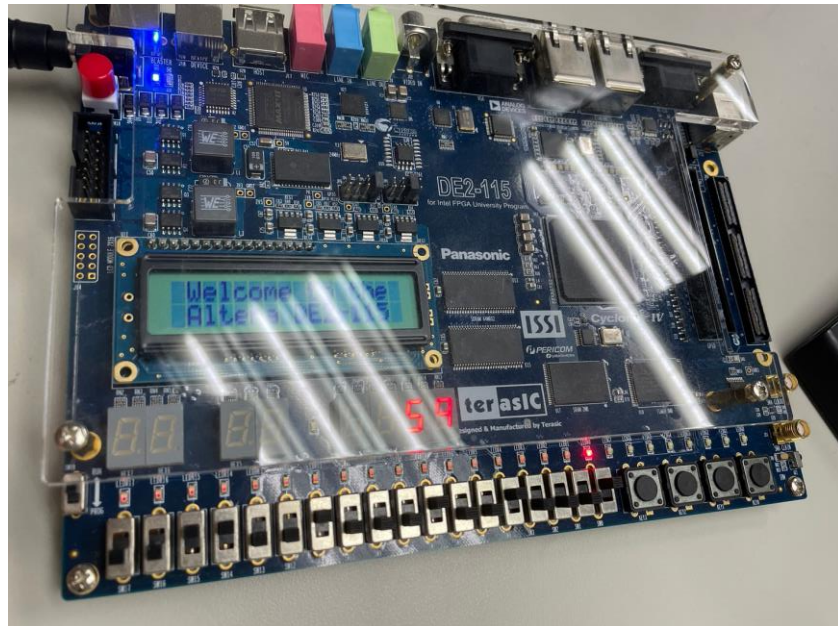
(3) 輸入 $17 + 64$

輸出 51



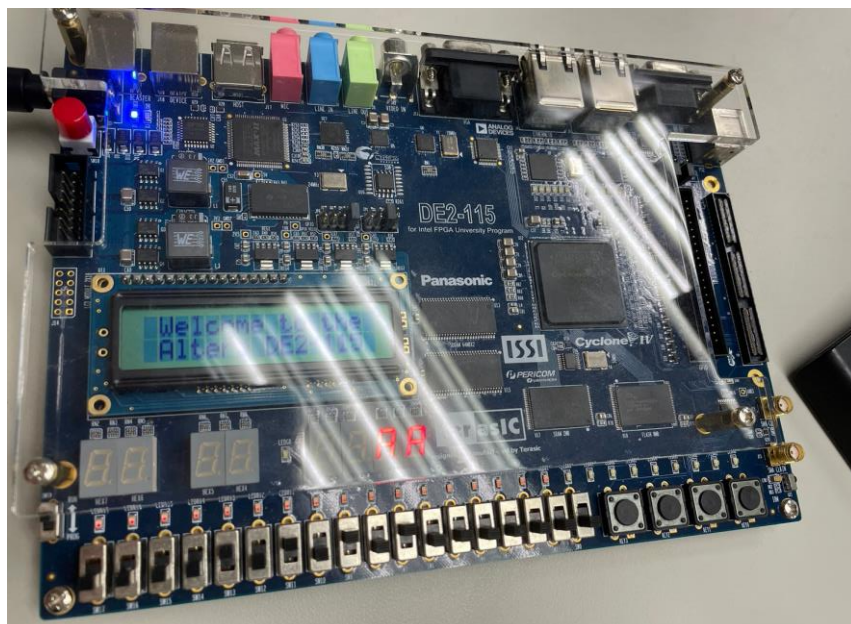
(4) 輸入 $144 + 200$

輸出 overflow



(5) 輸入 $128 + 42$

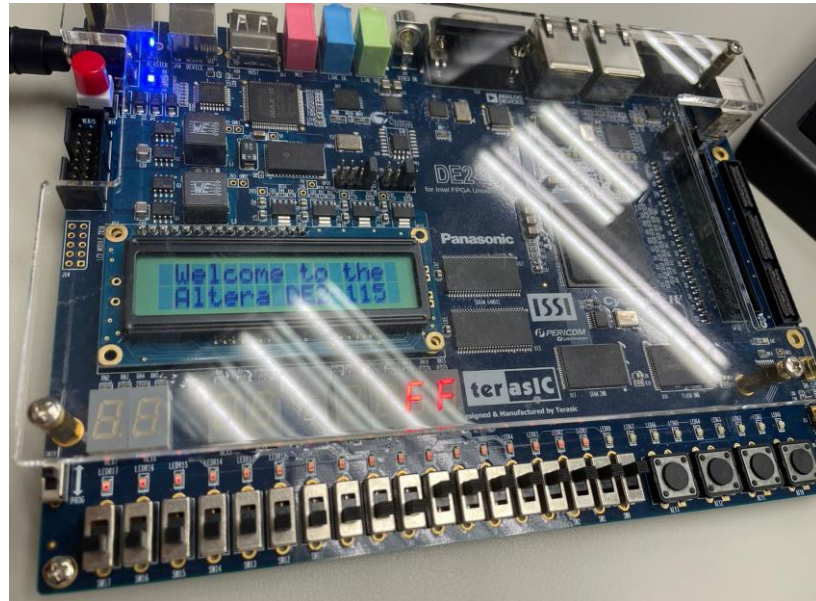
輸出 AA



2. 目標二

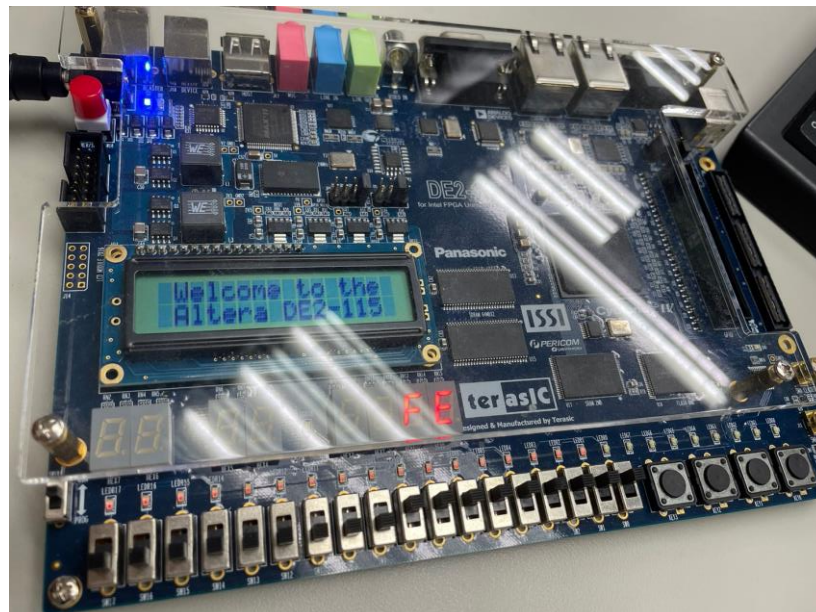
(1) 輸入 255 - 0

輸出 FF



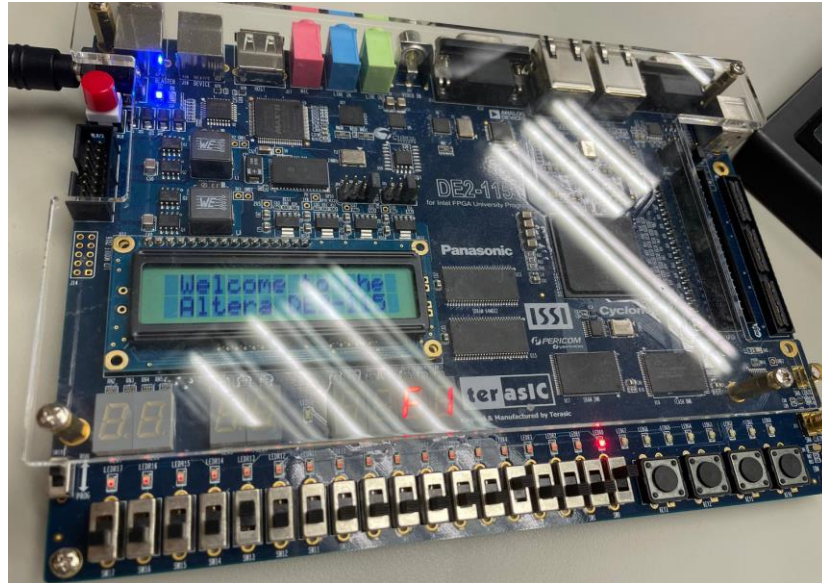
(2) 輸入 255 - 1

輸出 FE



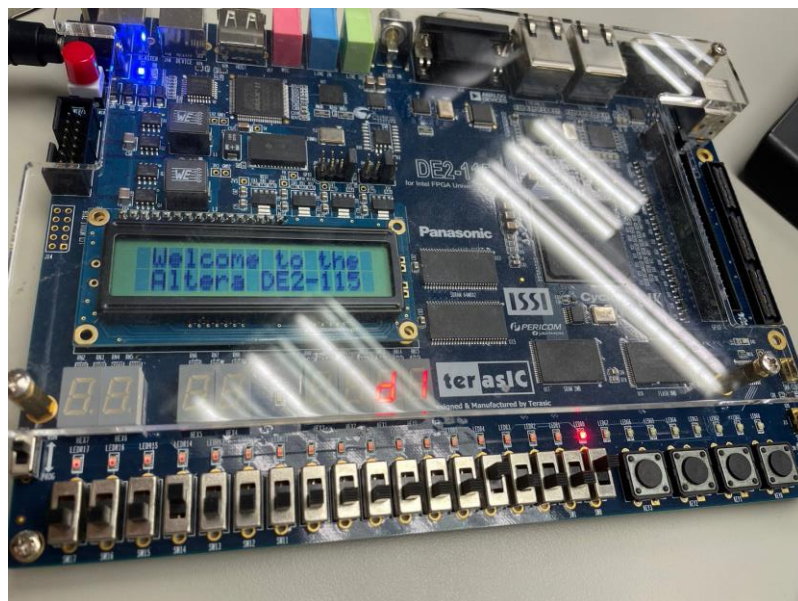
(3) 輸入 17 - 32

輸出 overflow



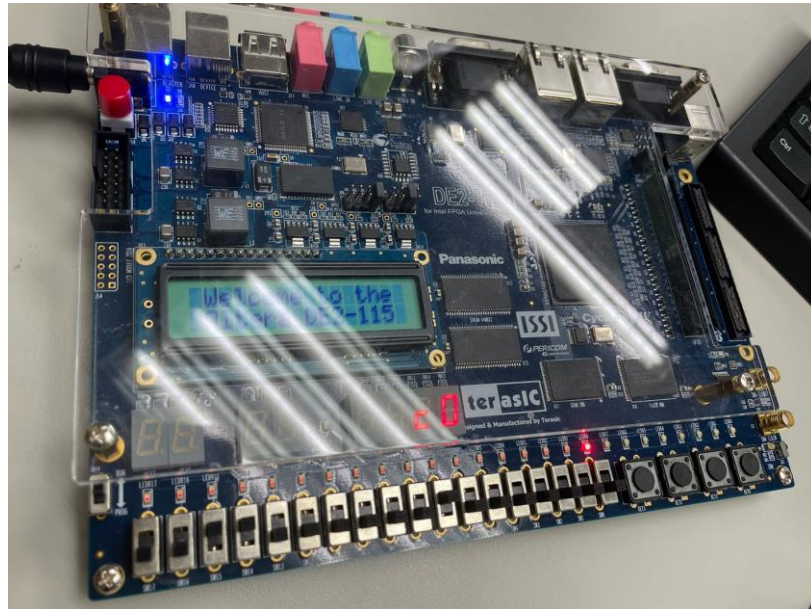
(4) 輸入 17 - 64

輸出 overflow



(5) 輸入 128 - 192

輸出 overflow



● 實際操作影片連結；

加法器

https://drive.google.com/file/d/1T8RgH_05tyBPENGBHwcOKQyrLGd_a-Wg/view?usp=drive_link

減法器

https://drive.google.com/file/d/1Y-oFlrYsdJcpWcAUaWbTgeR0V2pJpC-e/view?usp=drive_link

三、程式碼

(一) 目標一

fulladder.vhd

```
LIBRARY ieee;

USE ieee.std_logic_1164.all;

USE work.course2_package.all;

ENTITY fulladder IS

    PORT(

        x: in std_logic_vector(7 downto 0);

        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 fulladder;
```

```
ARCHITECTURE add OF fulladder IS
```

```
    SIGNAL Cin, Cout: std_logic := '0';
```

```
    SIGNAL c: std_logic_vector(7 downto 0);
```

```
    SIGNAL s: std_logic_vector(7 downto 0);
```

```
BEGIN
```

```
    stage0: fulladd PORT MAP ( Cin, x => x(0), y => y(0), s => s(0), Cout =  
> c(0) ) ;
```

```
    stage1: fulladd PORT MAP ( c(0), x(1), y(1), s(1), c(1) ) ;
```

```
    stage2: fulladd PORT MAP ( c(1), x(2), y(2), s(2), c(2) ) ;
```

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

```
    stage4: fulladd PORT MAP ( c(3), x(4), y(4), s(4), c(4) ) ;
```

```
    stage5: fulladd PORT MAP ( c(4), x(5), y(5), s(5), c(5) ) ;
```

```
    stage6: fulladd PORT MAP ( c(5), x(6), y(6), s(6), c(6) ) ;
```

```
    stage7: fulladd PORT MAP ( Cin => c(6), Cout => c(7), x => x(7), y =>  
y(7), s => s(7) ) ;
```

```
    stage8: Overflow <= c(7);
```

```
    stage9: hex 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 add;
```

package. vhd

```
LIBRARY ieee;

USE ieee.std_logic_1164.all;

package course2_package IS

    component fulladd

        PORT ( Cin, x, y : in std_logic;

              s, Cout : out std_logic);

    END component fulladd;

    component hex

        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, g
1: OUT std_logic);

        END component hex;

END course2_package;
```

fulladd. vhd

```
LIBRARY ieee;

USE ieee.std_logic_1164.all;

ENTITY fulladd is

    port( Cin, x, y : in std_logic;

          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;
```

hex. vhd


```
LIBRARY ieee;
```

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

```
ENTITY hex 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 s  
td_logic);
```

```
END hex;
```

```
ARCHITECTURE func of hex 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;
```

(二) 目標二

fullminusor.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;
```

package.vhd

```

LIBRARY ieee;

USE ieee.std_logic_1164.all;

package course2_package IS

    component fullminus

        PORT ( Cin, x, y : in std_logic;

              s, Cout : out std_logic);

    END component fullminus;

    component hex

        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, g
1: OUT std_logic);

        END component hex;

END course2_package;

```

fullminus.vhd

```
LIBRARY ieee;

USE ieee.std_logic_1164.all;

ENTITY fullminus is

    port( Cin, x, y : in std_logic;

          Cout : out std_logic);

END fullminus;

ARCHITECTURE func of fullminus is

BEGIN

    s <= x xor y xor Cin;

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

END func;
```

hex. vhd

```
LIBRARY ieee;
```

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

```
ENTITY hex 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 s  
td_logic);
```

```
END hex;
```

```
ARCHITECTURE func of hex 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;
```