微算機系統

小組專案報告

實驗八：

簡易CPU實作

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12月22日

一、實驗內容：

1. 須先設計出Simple Processor，其指令執行之時序之控制機制使用FSM狀態機（Mealy機）實現
2. Simple Processor的基本功能需有：LOAD、MOVE、AND、ADD 、SUB和Slt等六個功能
3. 總共有四個暫存器(R0,R1,R2,R3)，使用RS和RT分別指定要使用的暫存器(七段顯示器須同步更新)
4. 使用七段顯示器顯示Rs、Rt及BUS的值，左邊兩組顯示Rt及Rs兩個暫存器，最右邊一組則持續顯示Bus的數值
5. BUS為當下Data輸入的值
6. 七段顯示器顯示為十六進制，兩顆七段顯示器為一組，範圍為 0~FF(超過FF顯示末兩位)
7. 當執行指令時，暫存器的內容被改變，則七段顯示器須同步更新
8. 當指撥開關接設定好後，須使用按鈕開關將其輸入，在這裡，指令和暫存器(或Data)的輸入是使用指令時序的方式，且須用狀態機來實現指令和暫存器(或Data)的時序
9. 將資料透過Bus輸入暫存器
10. 依照instruction code來判斷，輸入指定暫存器與執行指令，並顯示更新暫存器與Bus的數值

Variable	Pin Location	Signal Name
clk	PIN_M23	KEY[0]
output[0]	PIN_F19	LED[1]
output[1]	PIN_E19	LED[2]
output[2]	PIN_F21	LED[3]
data(0)	PIN_AB28	SW[0]
data(1)	PIN_AC28	SW[1]
data(2)	PIN_AC27	SW[2]

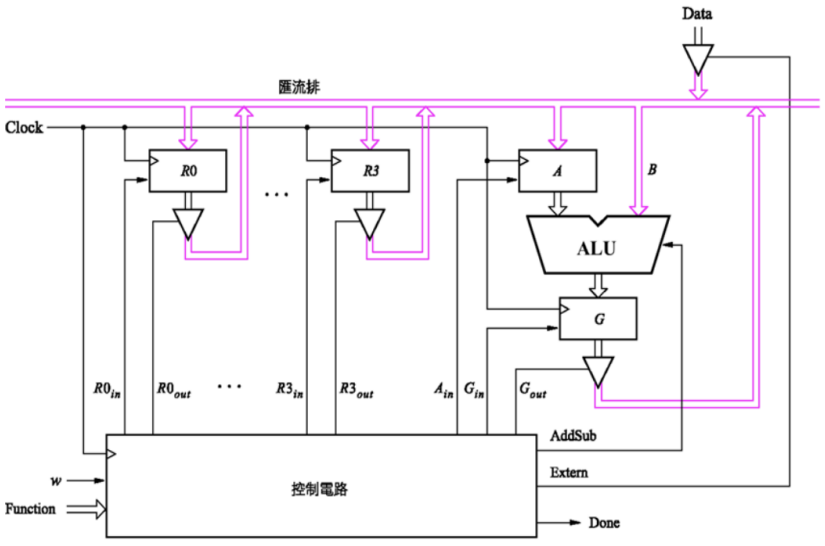
data(3)	PIN_AD27	SW[3]
data(4)	PIN_AB27	SW[4]
data(5)	PIN_AC26	SW[5]
data(6)	PIN_AD26	SW[6]
data(7)	PIN_AB26	SW[7]
opcode(0)	PIN_AC25	SW[8]
opcode(1)	PIN_AB25	SW[9]
opcode(2)	PIN_AC24	SW[10]
opcode(3)	PIN_AB24	SW[11]
rs(0)	PIN_AB23	SW[12]
rs(1)	PIN_AA24	SW[13]
rt(0)	PIN_AA23	SW[14]
rt(1)	PIN_AA22	SW[15]
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]
a2	PIN_AA25	HEX2[0]
b2	PIN_AA26	HEX2[1]
c2	PIN_Y25	HEX2[2]
d2	PIN_W26	HEX2[3]
e2	PIN_Y26	HEX2[4]
f2	PIN_W27	HEX2[5]
g2	PIN_W28	HEX2[6]
a3	PIN_V21	HEX3[0]
b3	PIN_U21	HEX3[1]
c3	PIN_AB20	HEX3[2]
d3	PIN_AA21	HEX3[3]
e3	PIN_AD24	HEX3[4]
f3	PIN_AF23	HEX3[5]
g3	PIN_Y19	HEX3[6]
a4	PIN_V21	HEX4[0]
b4	PIN_U21	HEX4[1]
c4	PIN_AB20	HEX4[2]
d4	PIN_AA21	HEX4[3]
e4	PIN_AD24	HEX4[4]
f4	PIN_AF23	HEX4[5]
g4	PIN_Y19	HEX4[6]
a5	PIN_V21	HEX5[0]
b5	PIN_U21	HEX5[1]
c5	PIN_AB20	HEX5[2]
d5	PIN_AA21	HEX5[3]

e5	PIN_AD24	HEX5[4]
f5	PIN_AF23	HEX5[5]
g5	PIN_Y19	HEX5[6]

二、實驗過程及結果：

（一） 預期實驗結果的流程示意圖



Instruction	Instruction code			usage	Result
	opcode	Rs	Rt		
Load	0000	2bit	2bit	Load Rs	Rs←Data
Move	0001			Move Rs, Rt,	Rs←Rt
Add	0010			Add Rs, Rt	Rs←Rs + Rt
And	0011			And Rs, Rt	Rs←Rs & Rt
Sub(A-B)	0101			Sub Rs, Rt	Rs←Rs – Rt
Sub(B-A)	1001			Sub Rs, Rt	Rs← Rt – Rs
Slt	0100			Slt Rs, Rt	if (Rs<Rt) Rs=1;else Rs=0
Div(加分題)	1000			Div Rs, Rt	Rs←Rs / Rt

（二） 設計電路簡介

s_temp select => 根據rs值將對應的暫存器值放入s_temp

t select => 根據rt值將對應的暫存器值放入t

process 1: 根據opcode做對應指令，將s_temp和t做運算後放入s

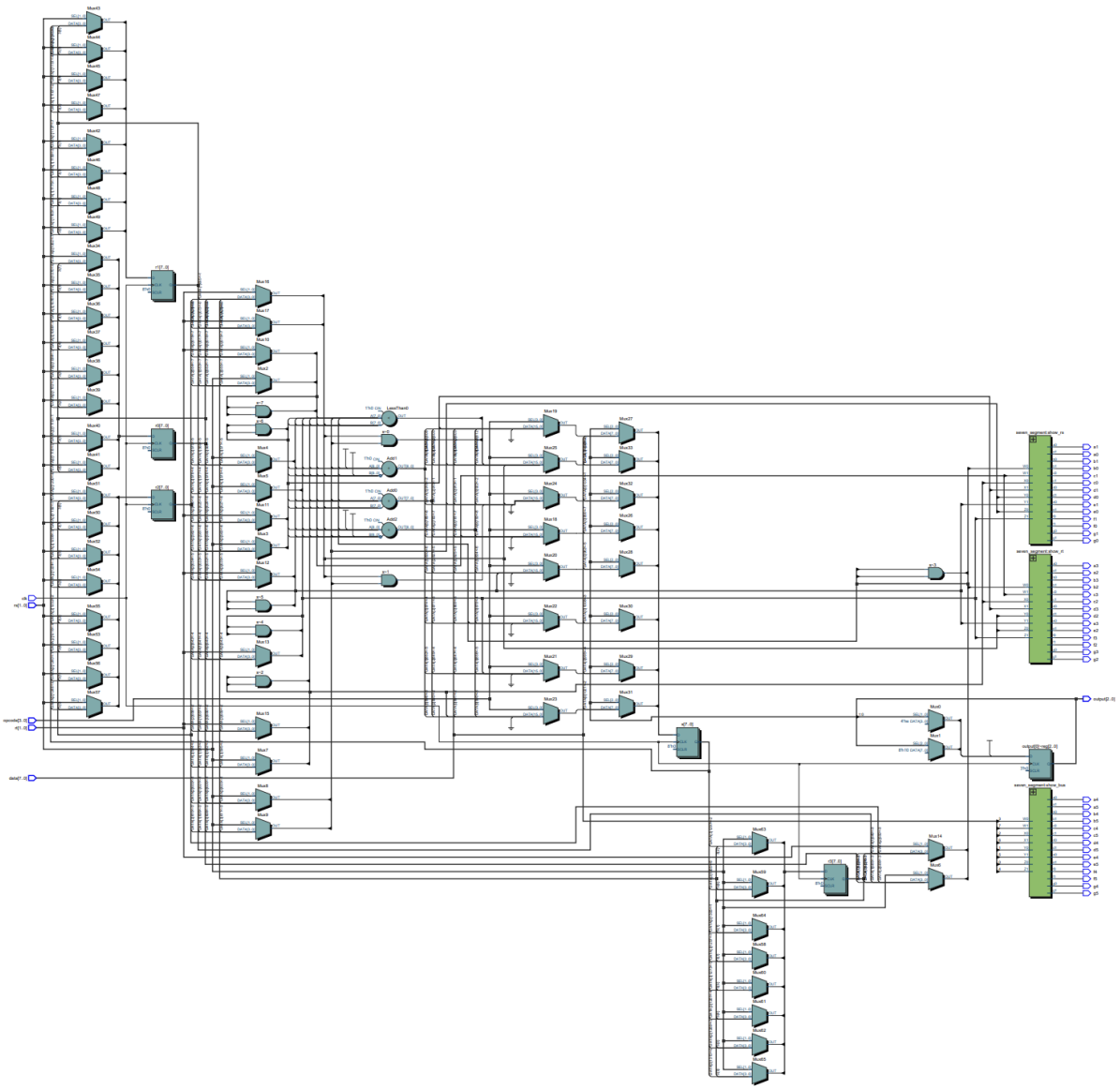
process 2: 根據rs值將計算結果s放回對應暫存器

show1: rs值 => s_temp

show2: rt值 => t

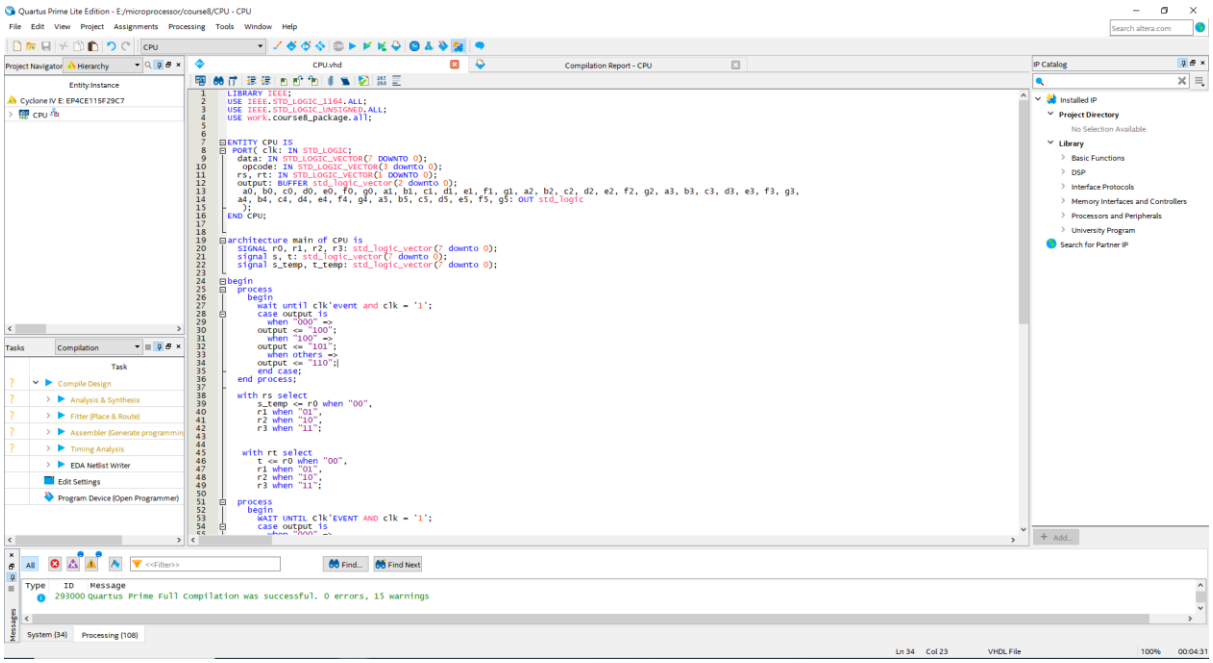
show3: bus => data

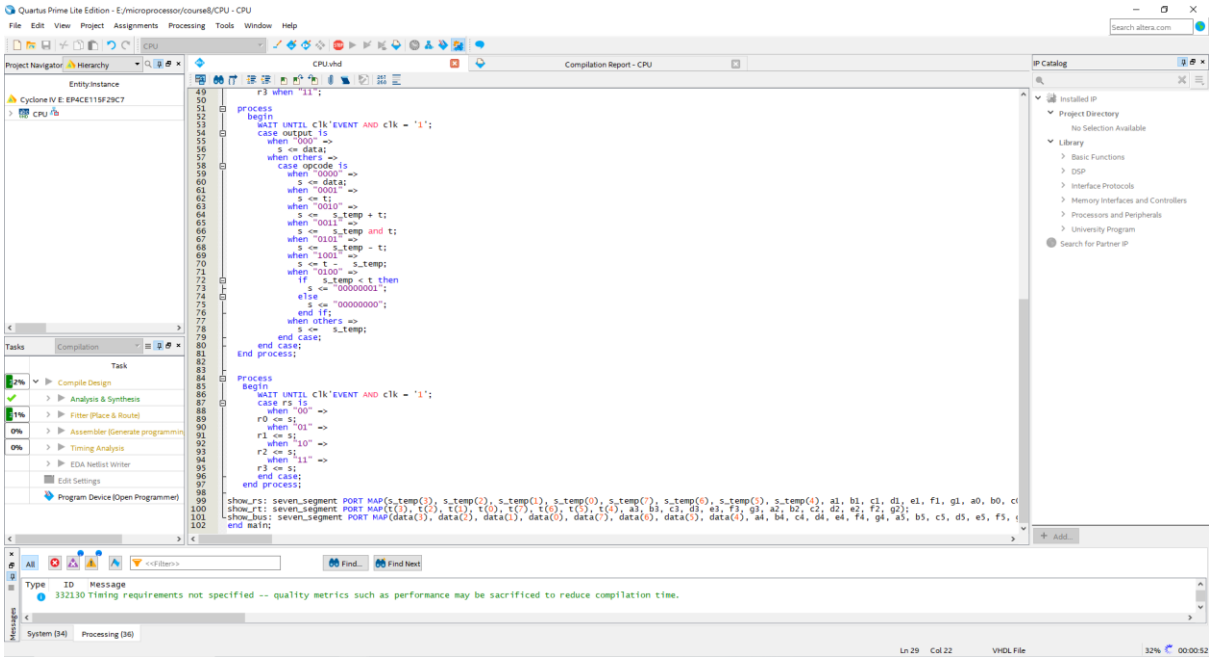
（三） 設計出的電路

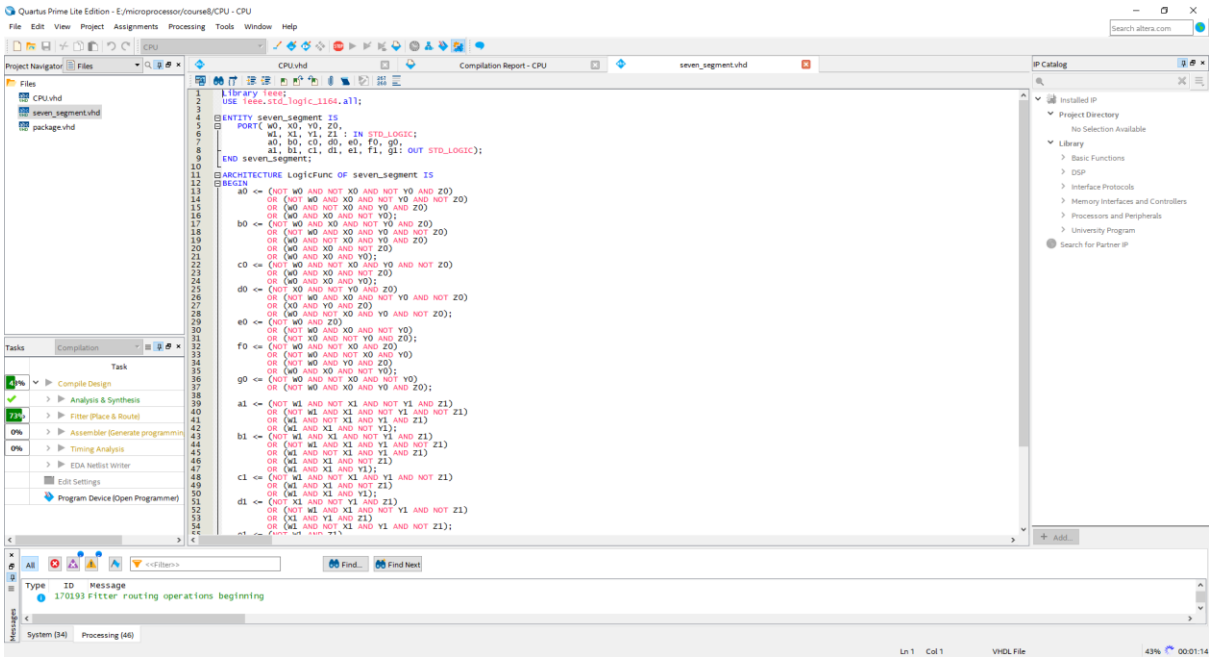


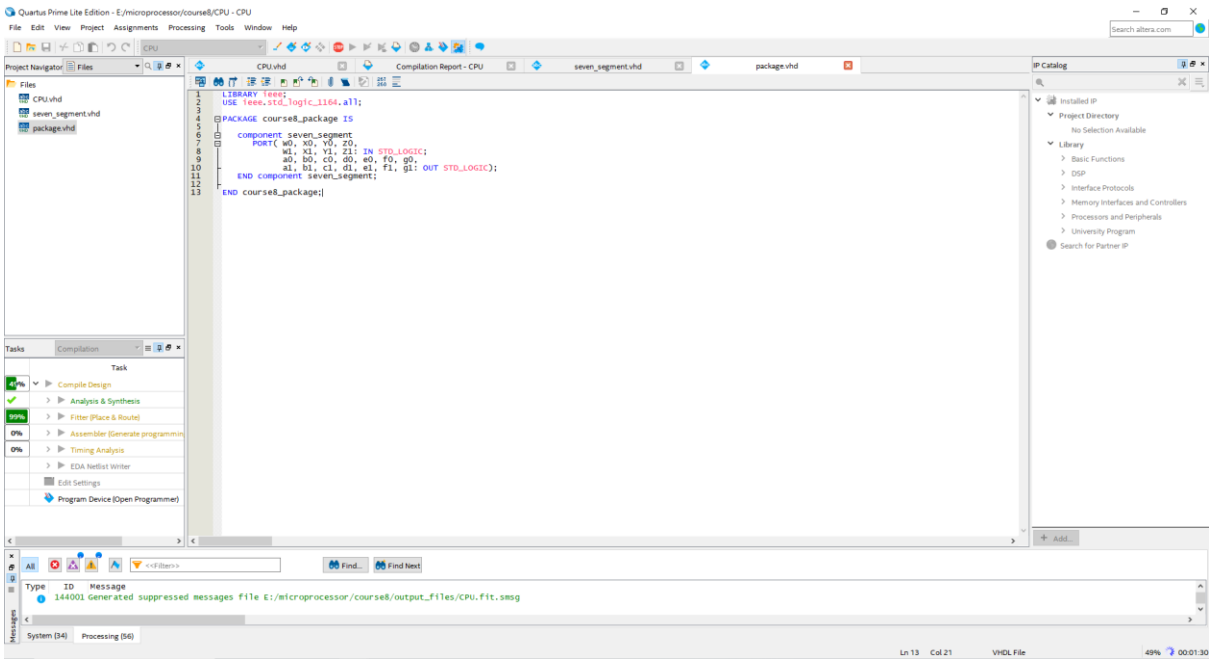
（四） 實驗過程

(1)　撰寫程式碼

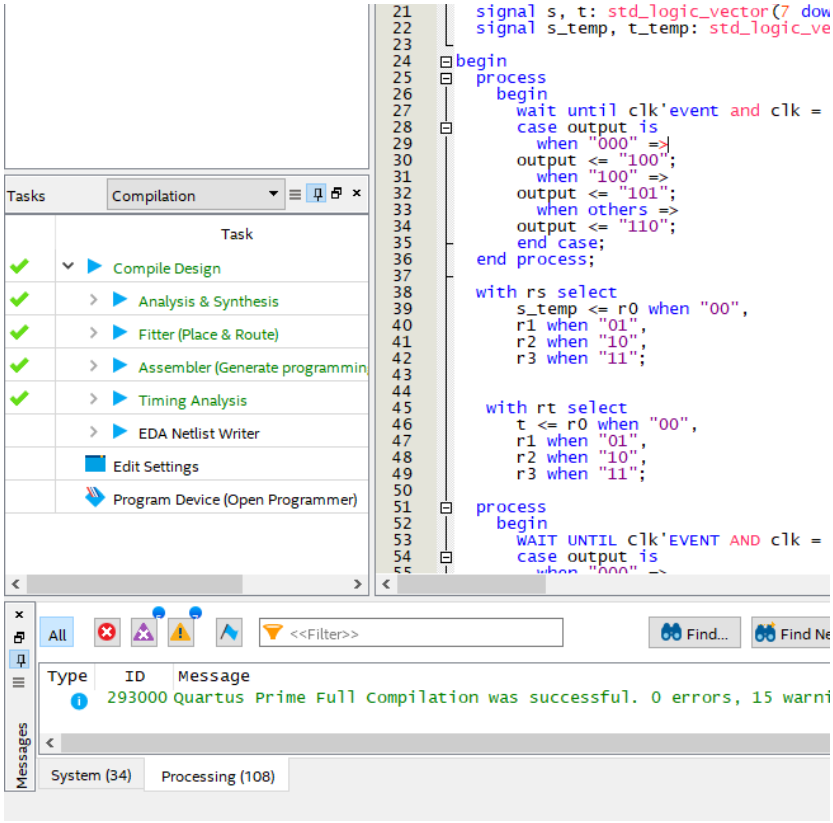




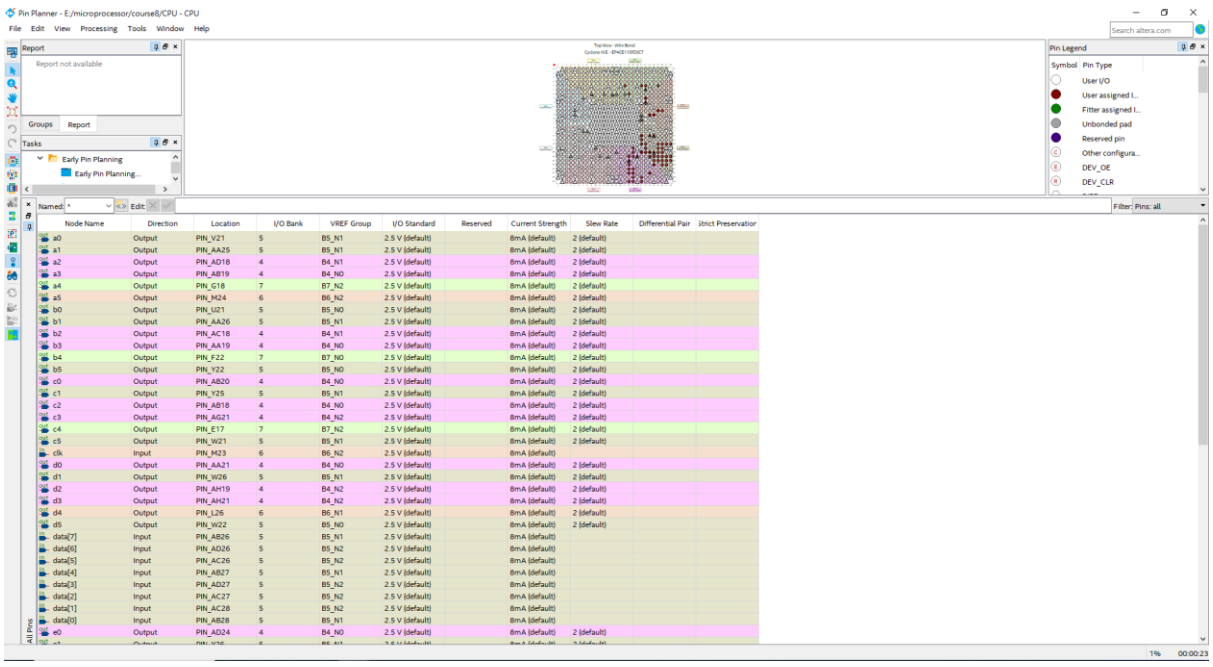

Quartus Prime Lite Edition - E:/microprocessor/course8/CPU - CPU
CPU.vhd
Compilation Report - CPU
seven_segment.vhd
Type ID Message
170193 Fitter routing operations beginning

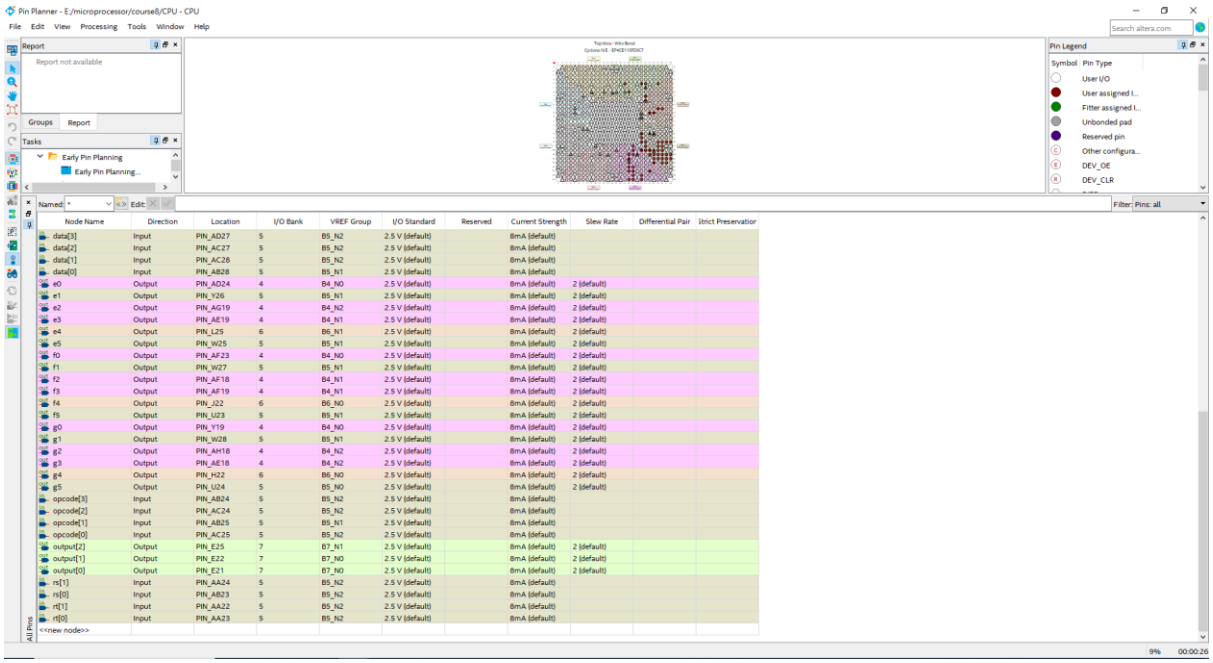

Quartus Prime Lite Edition - E:/microprocessor/course8/CPU - CPU
CPU.vhd
Compilation Report - CPU
seven_segment.vhd
package.vhd
Type ID Message
144001 Generated suppressed messages file E:/microprocessor/course8/output_files/CPU.fit.smsg

（2） 編譯成功

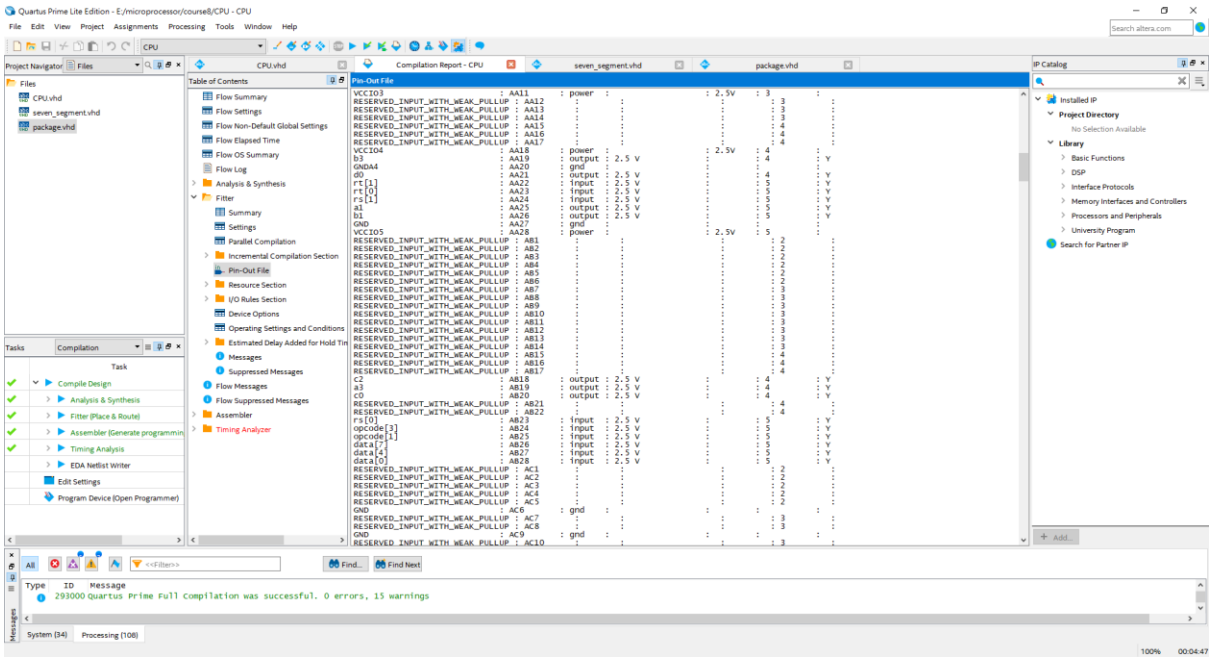

Tasks
Compilation
Task
Compile Design
Analysis & Synthesis
Fitter (Place & Route)
Assembler (Generate programmin
Timing Analysis
EDA Netlist Writer
Edit Settings
Program Device (Open Programmer)
signal s, t: std_logic_vector(7 dow
signal s_temp, t_temp: std_logic_ve
begin
process
begin
wait until clk'event and clk =
case output is
when "000" =>
output <= "100";
when "100" =>
output <= "101";
when others =>
output <= "110";
end case;
end process;
with rs select
s_temp <= r0 when "00",
r1 when "01",
r2 when "10",
r3 when "11";
with rt select
t <= r0 when "00",
r1 when "01",
r2 when "10",
r3 when "11";
process
begin
WAIT UNTIL clk'EVENT AND clk =
case output is
All
<<Filter>>
Find...
Find Ne
Type ID Message
293000 Quartus Prime Full Compilation was successful. 0 errors, 15 warni
Messages
System (34)
Processing (108)

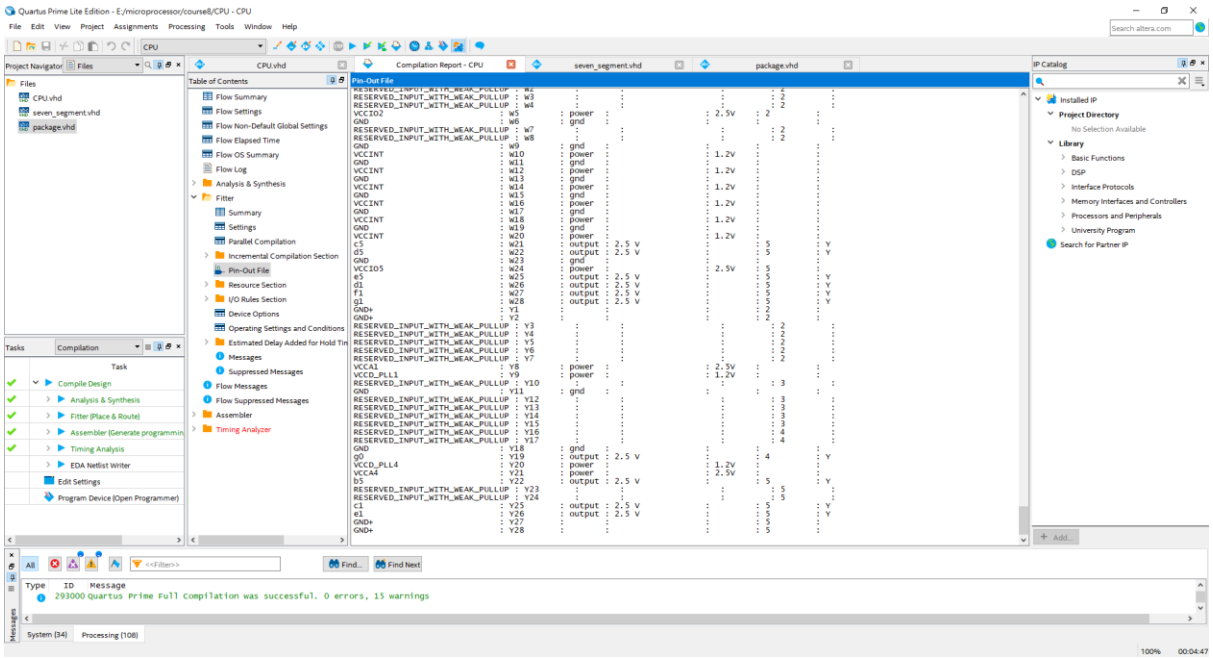
（3） 接腳位



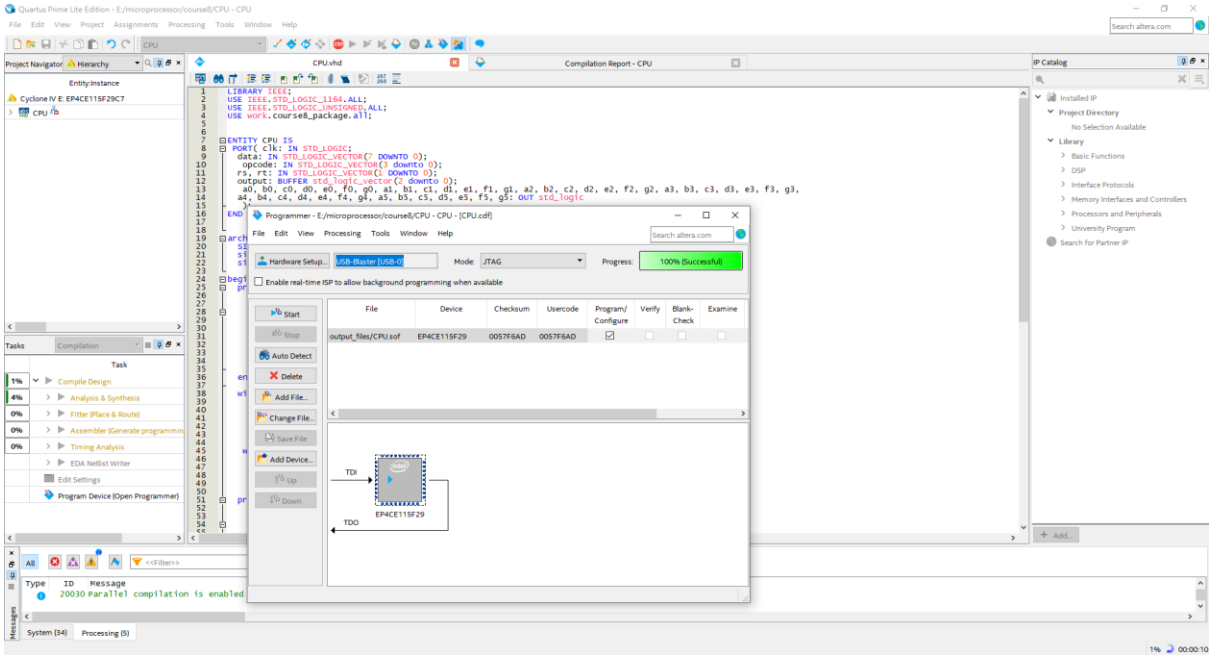

Pin Planner - E:/microprocessor/course8/CPU - CPU
File Edit View Processing Tools Window Help
Report
Report not available
Groups Report
Tasks
Early Pin Planning
Early Pin Planning...
Pin Legend
Symbol Pin Type
User I/O
User assigned I...
Fitter assigned I...
Unbonded pad
Reserved pin
Other configura...
DEV_OE
DEV_CLR
Named: * Edit:
Filter: Pins: all
Node Name Direction Location I/O Bank VREF Group I/O Standard Reserved Current Strength Slew Rate Differential Pair Strict Preservation
data[3] Input PIN_AD27 5 B5_N2 2.5 V (default) 8mA (default)
data[2] Input PIN_AC27 5 B5_N2 2.5 V (default) 8mA (default)
data[1] Input PIN_AC28 5 B5_N2 2.5 V (default) 8mA (default)
data[0] Input PIN_AB28 5 B5_N1 2.5 V (default) 8mA (default)
e0 Output PIN_AD24 4 B4_N0 2.5 V (default) 8mA (default) 2 (default)
e1 Output PIN_Y26 5 B5_N1 2.5 V (default) 8mA (default) 2 (default)
e2 Output PIN_AG19 4 B4_N2 2.5 V (default) 8mA (default) 2 (default)
e3 Output PIN_AE19 4 B4_N1 2.5 V (default) 8mA (default) 2 (default)
e4 Output PIN_L25 6 B6_N1 2.5 V (default) 8mA (default) 2 (default)
e5 Output PIN_W25 5 B5_N1 2.5 V (default) 8mA (default) 2 (default)
f0 Output PIN_AF23 4 B4_N0 2.5 V (default) 8mA (default) 2 (default)
f1 Output PIN_W27 5 B5_N1 2.5 V (default) 8mA (default) 2 (default)
f2 Output PIN_AF18 4 B4_N1 2.5 V (default) 8mA (default) 2 (default)
f3 Output PIN_AF19 4 B4_N1 2.5 V (default) 8mA (default) 2 (default)
f4 Output PIN_J22 6 B6_N0 2.5 V (default) 8mA (default) 2 (default)
f5 Output PIN_U23 5 B5_N1 2.5 V (default) 8mA (default) 2 (default)
g0 Output PIN_Y19 4 B4_N0 2.5 V (default) 8mA (default) 2 (default)
g1 Output PIN_W28 5 B5_N1 2.5 V (default) 8mA (default) 2 (default)
g2 Output PIN_AH18 4 B4_N2 2.5 V (default) 8mA (default) 2 (default)
g3 Output PIN_AE18 4 B4_N2 2.5 V (default) 8mA (default) 2 (default)
g4 Output PIN_H22 6 B6_N0 2.5 V (default) 8mA (default) 2 (default)
g5 Output PIN_U24 5 B5_N0 2.5 V (default) 8mA (default) 2 (default)
opcode[3] Input PIN_AB24 5 B5_N2 2.5 V (default) 8mA (default)
opcode[2] Input PIN_AC24 5 B5_N2 2.5 V (default) 8mA (default)
opcode[1] Input PIN_AB25 5 B5_N1 2.5 V (default) 8mA (default)
opcode[0] Input PIN_AC25 5 B5_N2 2.5 V (default) 8mA (default)
output[2] Output PIN_E25 7 B7_N1 2.5 V (default) 8mA (default) 2 (default)
output[1] Output PIN_E22 7 B7_N0 2.5 V (default) 8mA (default) 2 (default)
output[0] Output PIN_E21 7 B7_N0 2.5 V (default) 8mA (default) 2 (default)
rs[1] Input PIN_AA24 5 B5_N2 2.5 V (default) 8mA (default)
rs[0] Input PIN_AB23 5 B5_N2 2.5 V (default) 8mA (default)
rt[1] Input PIN_AA22 5 B5_N2 2.5 V (default) 8mA (default)
rt[0] Input PIN_AA23 5 B5_N2 2.5 V (default) 8mA (default)
<<new node>>
All Pins
9% 00:00:26

（4） 確認接線於正確腳位


Quartus Prime Lite Edition - E:/microprocessor/course8/CPU - CPU
File Edit View Project Assignments Processing Tools Window Help
Project Navigator Files
Files
CPU.vhd
seven_segment.vhd
package.vhd
Tasks Compilation
Task
Compile Design
Analysis & Synthesis
Fitter (Place & Route)
Assembler (Generate programmin
Timing Analysis
EDA Netlist Writer
Edit Settings
Program Device (Open Programmer)
CPU.vhd
Table of Contents
Flow Summary
Flow Settings
Flow Non-Default Global Settings
Flow Elapsed Time
Flow OS Summary
Flow Log
Analysis & Synthesis
Fitter
Summary
Settings
Parallel Compilation
Incremental Compilation Section
Pin-Out File
Resource Section
I/O Rules Section
Device Options
Operating Settings and Conditions
Estimated Delay Added for Hold Tim
Messages
Suppressed Messages
Flow Messages
Flow Suppressed Messages
Assembler
Timing Analyzer
Compilation Report - CPU
seven_segment.vhd
package.vhd
Pin-Out File
VCCIO3 : AA11 : power : : 2.5V : 3 :
RESERVED_INPUT_WITH_WEAK_PULLUP : AA12 : : : : 3 :
RESERVED_INPUT_WITH_WEAK_PULLUP : AA13 : : : : 3 :
RESERVED_INPUT_WITH_WEAK_PULLUP : AA14 : : : : 3 :
RESERVED_INPUT_WITH_WEAK_PULLUP : AA15 : : : : 4 :
RESERVED_INPUT_WITH_WEAK_PULLUP : AA16 : : : : 4 :
RESERVED_INPUT_WITH_WEAK_PULLUP : AA17 : : : : 4 :
VCCIO4 : AA18 : power : : 2.5V : 4 :
b3 : AA19 : output : 2.5 V : : 4 : Y
GNDA4 : AA20 : gnd : : : : :
d0 : AA21 : output : 2.5 V : : 4 : Y
rt[1] : AA22 : input : 2.5 V : : 5 : Y
rt[0] : AA23 : input : 2.5 V : : 5 : Y
rs[1] : AA24 : input : 2.5 V : : 5 : Y
a1 : AA25 : output : 2.5 V : : 5 : Y
b1 : AA26 : output : 2.5 V : : 5 : Y
GND : AA27 : gnd : : : : :
VCCIO5 : AA28 : power : : 2.5V : 5 :
RESERVED_INPUT_WITH_WEAK_PULLUP : AB1 : : : : 2 :
RESERVED_INPUT_WITH_WEAK_PULLUP : AB2 : : : : 2 :
RESERVED_INPUT_WITH_WEAK_PULLUP : AB3 : : : : 2 :
RESERVED_INPUT_WITH_WEAK_PULLUP : AB4 : : : : 2 :
RESERVED_INPUT_WITH_WEAK_PULLUP : AB5 : : : : 2 :
RESERVED_INPUT_WITH_WEAK_PULLUP : AB6 : : : : 2 :
RESERVED_INPUT_WITH_WEAK_PULLUP : AB7 : : : : 3 :
RESERVED_INPUT_WITH_WEAK_PULLUP : AB8 : : : : 3 :
RESERVED_INPUT_WITH_WEAK_PULLUP : AB9 : : : : 3 :
RESERVED_INPUT_WITH_WEAK_PULLUP : AB10 : : : : 3 :
RESERVED_INPUT_WITH_WEAK_PULLUP : AB11 : : : : 3 :
RESERVED_INPUT_WITH_WEAK_PULLUP : AB12 : : : : 3 :
RESERVED_INPUT_WITH_WEAK_PULLUP : AB13 : : : : 3 :
RESERVED_INPUT_WITH_WEAK_PULLUP : AB14 : : : : 3 :
RESERVED_INPUT_WITH_WEAK_PULLUP : AB15 : : : : 4 :
RESERVED_INPUT_WITH_WEAK_PULLUP : AB16 : : : : 4 :
RESERVED_INPUT_WITH_WEAK_PULLUP : AB17 : : : : 4 :
c2 : AB18 : output : 2.5 V : : 4 : Y
a3 : AB19 : output : 2.5 V : : 4 : Y
c0 : AB20 : output : 2.5 V : : 4 : Y
RESERVED_INPUT_WITH_WEAK_PULLUP : AB21 : : : : 4 :
RESERVED_INPUT_WITH_WEAK_PULLUP : AB22 : : : : 4 :
rs[0] : AB23 : input : 2.5 V : : 5 : Y
opcode[3] : AB24 : input : 2.5 V : : 5 : Y
opcode[1] : AB25 : input : 2.5 V : : 5 : Y
data[7] : AB26 : input : 2.5 V : : 5 : Y
data[4] : AB27 : input : 2.5 V : : 5 : Y
data[0] : AB28 : input : 2.5 V : : 5 : Y
RESERVED_INPUT_WITH_WEAK_PULLUP : AC1 : : : : 2 :
RESERVED_INPUT_WITH_WEAK_PULLUP : AC2 : : : : 2 :
RESERVED_INPUT_WITH_WEAK_PULLUP : AC3 : : : : 2 :
RESERVED_INPUT_WITH_WEAK_PULLUP : AC4 : : : : 2 :
RESERVED_INPUT_WITH_WEAK_PULLUP : AC5 : : : : 2 :
GND : AC6 : gnd : : : :
RESERVED_INPUT_WITH_WEAK_PULLUP : AC7 : : : : 3 :
RESERVED_INPUT_WITH_WEAK_PULLUP : AC8 : : : : 3 :
GND : AC9 : gnd : : : :
RESERVED_INPUT_WITH_WEAK_PULLUP : AC10 : : : : 3 :
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
Add...
Find... Find Next
Type ID Message
293000 Quartus Prime Full Compilation was successful. 0 errors, 15 warnings
System (34) Processing (108)
100% 00:04:47



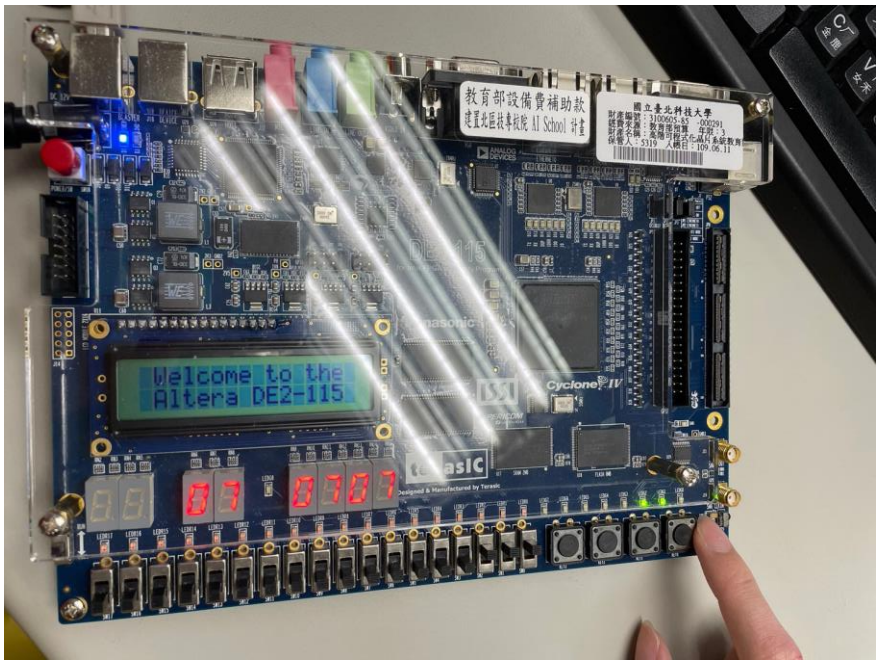
(5)　燒錄視窗設定及燒錄成功畫面



（五） 實驗結果

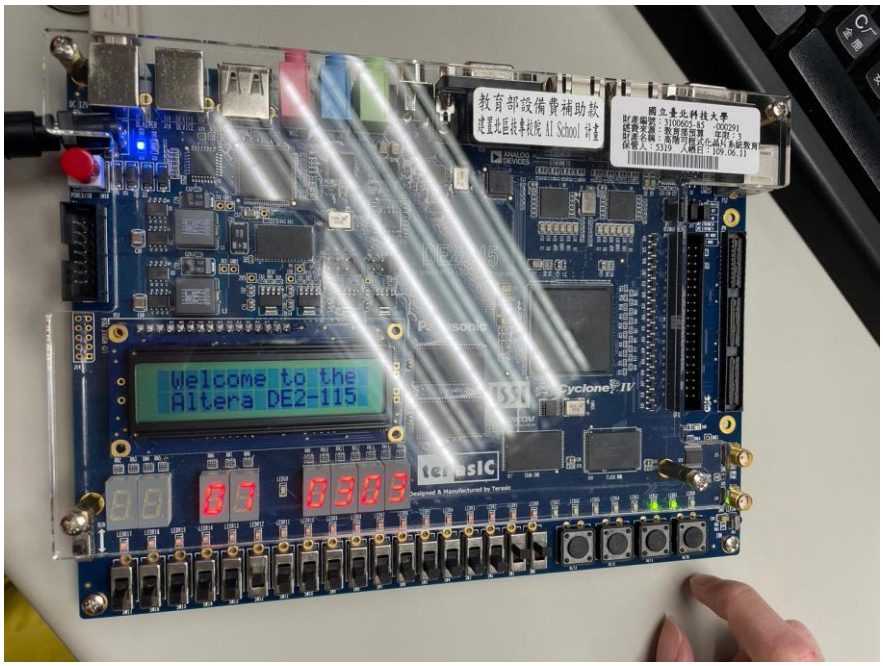
(1) opcode：0000(load)、rs：00、rt：00、data:07

輸出結果 bus：07、rs：07、rt：07（r0 = 07）



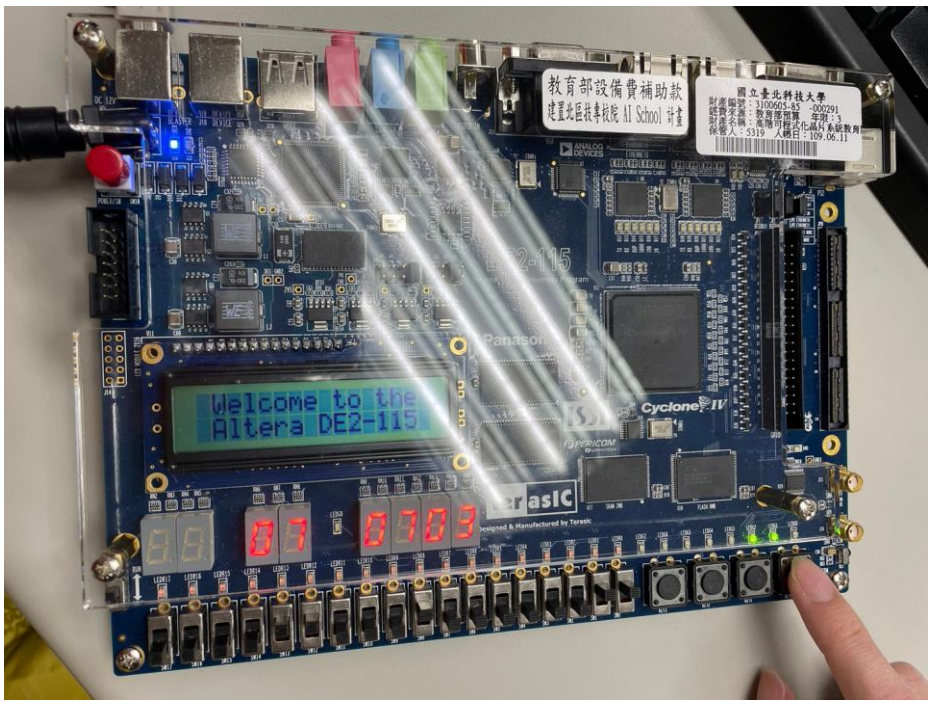
(2) opcode：0000(load)、rs：01、rt：00、data:03

輸出結果 bus：03、rs：03、rt：07（r0 = 07、r1 = 03）



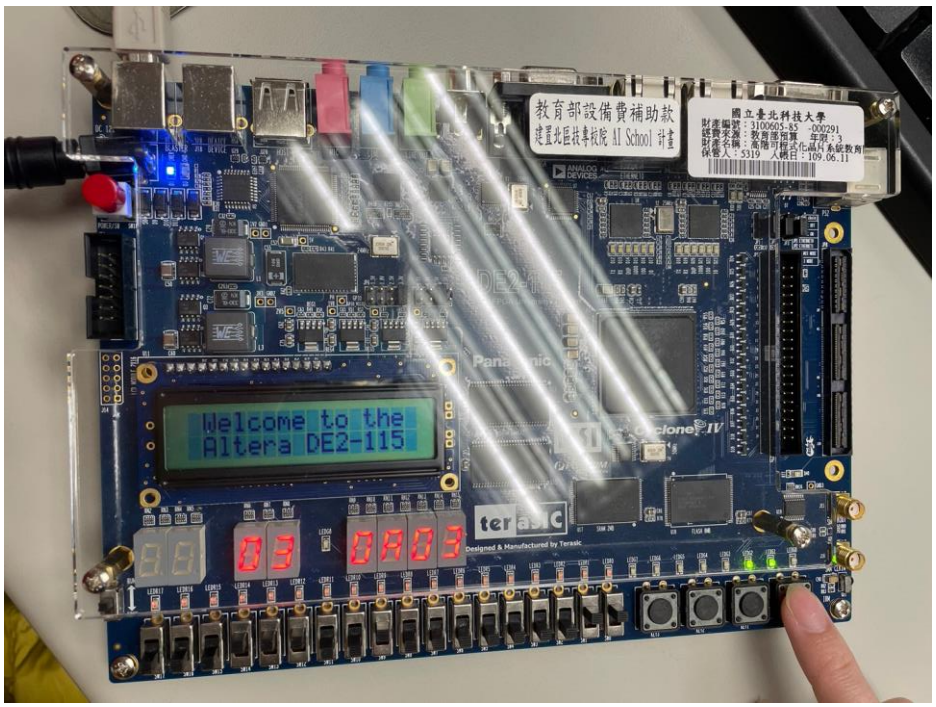
(3) opcode：0001(move)、rs：11、rt：00

輸出結果 rs：07、rt：07（r0 = 07、r1 = 03、r3 = 07）



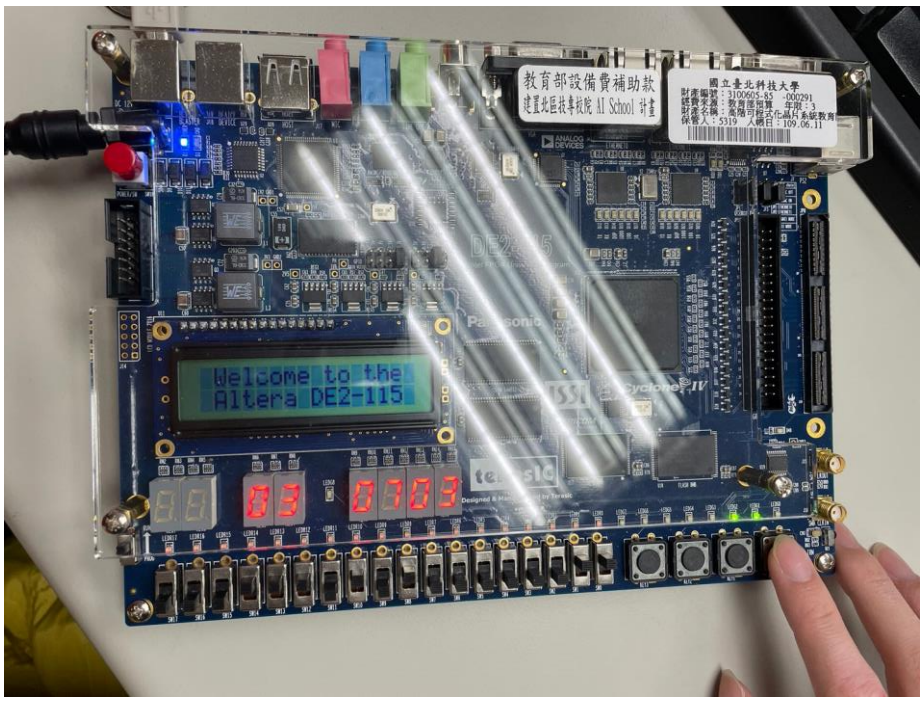
(4) opcode：0010(add)、rs：11、rt：01

輸出結果 rs：0A、rt：07（r0 = 07、r1 = 03、r3 = 0A）



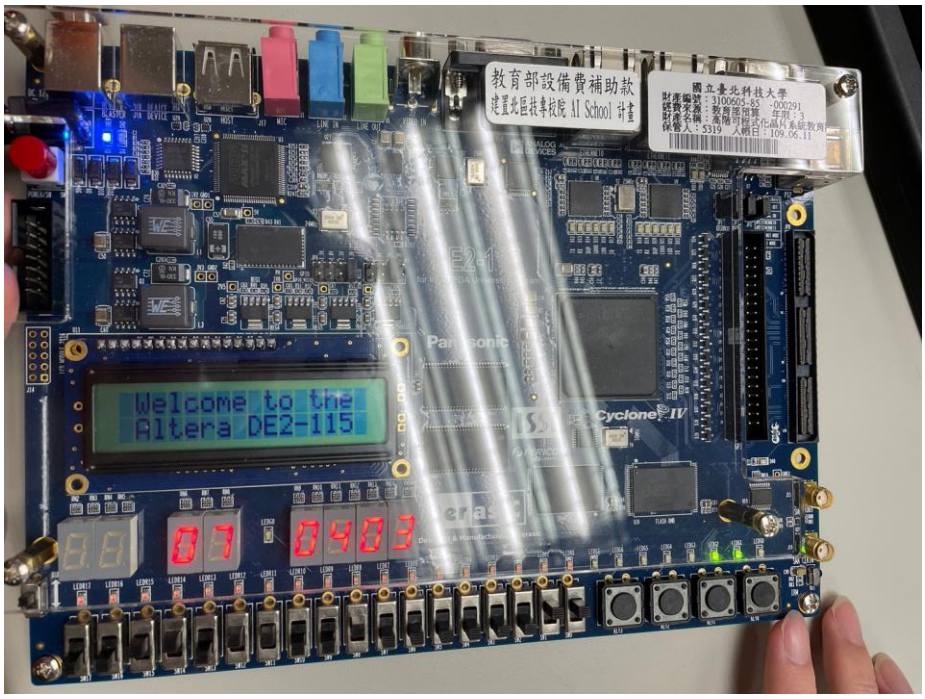
(5) opcode：0101(sub s-t)、rs：11、rt：01

輸出結果 rs：07、rt：03（r0 = 07、r1 = 03、r3 = 07）



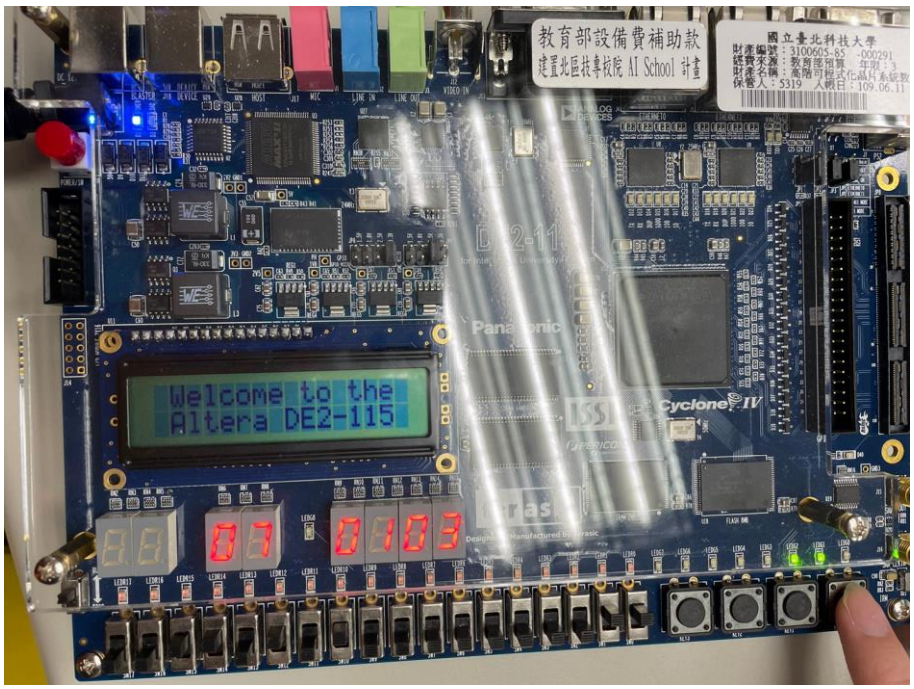
(6) opcode：1001(sub t-s)、rs：01、rt：11

輸出結果 rs：04、rt：07（r0 = 07、r1 = 04、r3 = 07）



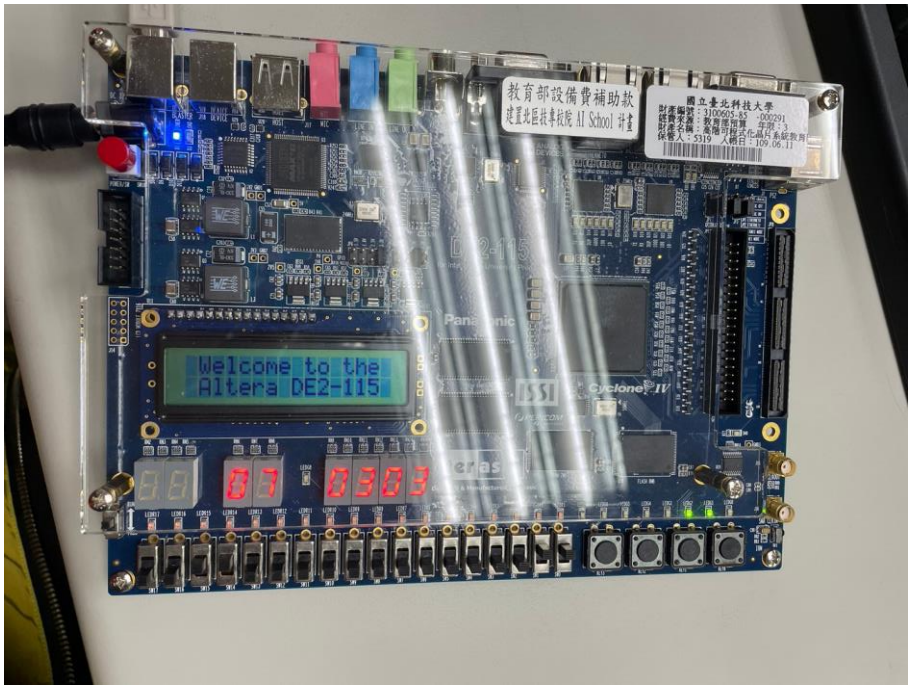
(7) opcode：0100(slt)、rs：01、rt：11

輸出結果 rs：01、rt：07（r0 = 07、r1 = 01、r3 = 07）



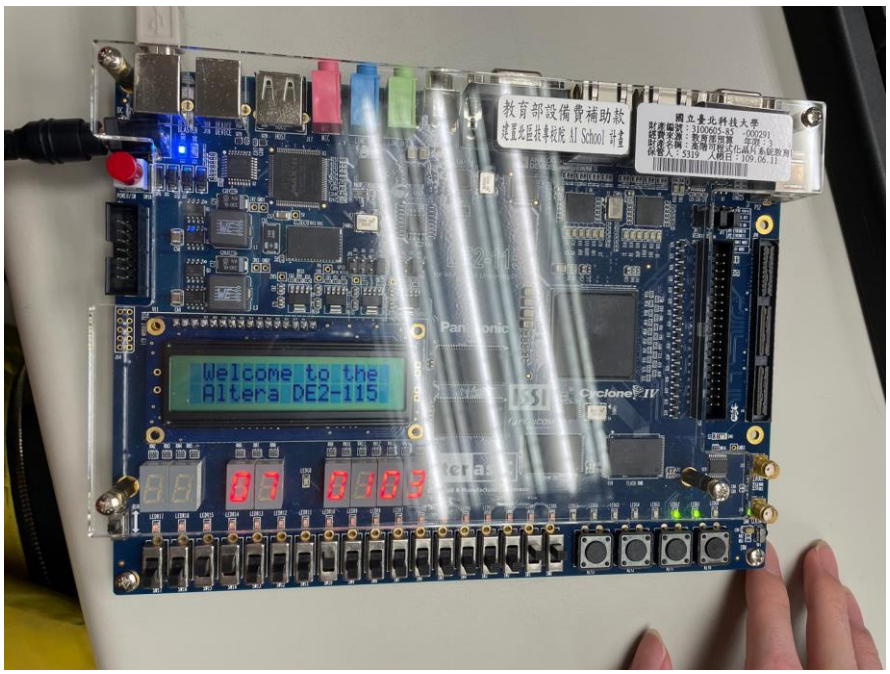
(8) opcode：0000(load)、rs：00、rt：11

輸出結果 rs：03、rt：07（r0 = 03、r1 = 01、r3 = 07）



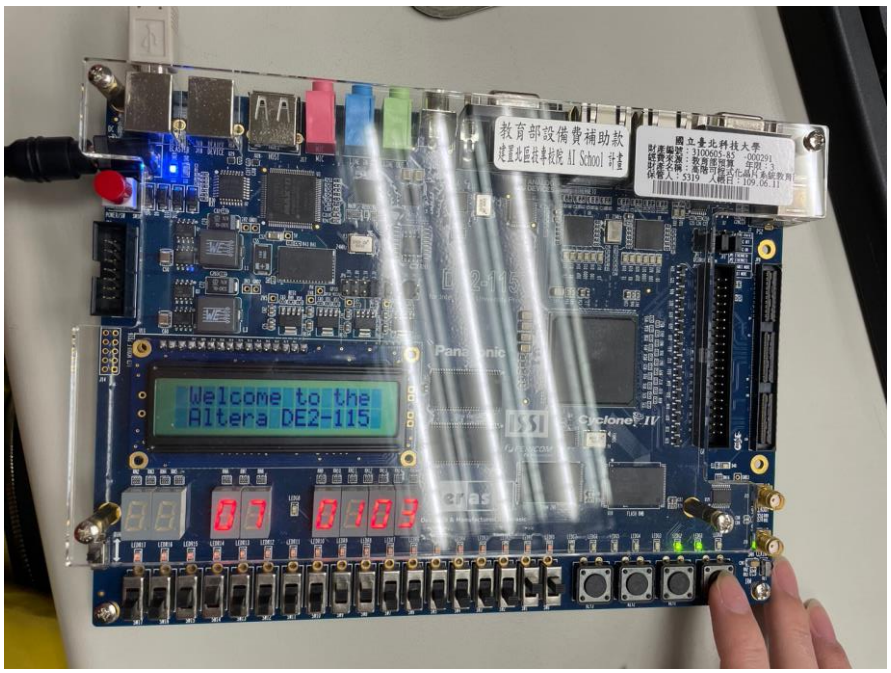
(9) opcode：0100(slt)、rs：00、rt：11

輸出結果 rs：01、rt：07（r0 = 01、r1 = 01、r3 = 07）



(10)opcode：0011(and)、rs：00、rt：11

輸出結果 rs：01、rt：07（r0 = 01、r1 = 01、r3 = 07）



● 實際操作影片連結；

https://drive.google.com/file/d/1nYdxZdXWYK0BNUyodIGAcp5CUUJ8KrlI/view?usp=sharing

三、程式碼

CPU. vhd

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
USE work.course8_package.all;


ENTITY CPU IS
 PORT( clk: IN STD_LOGIC;
  data: IN STD_LOGIC_VECTOR(7 DOWNTO 0);
   opcode: IN STD_LOGIC_VECTOR(3 downto 0);
  rs, rt: IN STD_LOGIC_VECTOR(1 DOWNTO 0);
  output: BUFFER std_logic_vector(2 downto 0);
   a0, b0, c0, d0, e0, f0, g0, a1, b1, c1, d1, e1, f1, g1, a2, b2, c2, d2, e2, f2, g2, a3, b3, c3, d3, e3, f
3, g3,
  a4, b4, c4, d4, e4, f4, g4, a5, b5, c5, d5, e5, f5, g5: OUT std_logic
   );
```

```
END CPU;



architecture main of CPU is

  SIGNAL r0, r1, r2, r3: std_logic_vector(7 downto 0);

  signal s, t: std_logic_vector(7 downto 0);

  signal s_temp, t_temp: std_logic_vector(7 downto 0);



begin

  process

    begin

      wait until clk'event and clk = '1';

      case output is

        when "000" =>

      output <= "100";

        when "100" =>

      output <= "101";

        when others =>

      output <= "110";

      end case;
```

```
end process;


with rs select
                s_temp <= r0 when "00",
                r1 when "01",
                r2 when "10",
                r3 when "11";


        with rt select
                t <= r0 when "00",
                r1 when "01",
                r2 when "10",
                r3 when "11";


process
  begin
    WAIT UNTIL Clk'EVENT AND clk = '1';
    case output is
      when "000" =>
```

```
  s <= data;
when others =>
  case opcode is
    when "0000" =>
      s <= data;
    when "0001" =>
      s <= t;
    when "0010" =>
      s <=   s_temp + t;
    when "0011" =>
      s <=   s_temp and t;
    when "0101" =>
      s <=   s_temp - t;
    when "1001" =>
      s <= t -   s_temp;
    when "0100" =>
      if   s_temp < t then
        s <= "00000001";
      else
```

```
            s <= "00000000";
          end if;
        when others =>
          s <=  s_temp;
      end case;
  end case;
End process;

Process
        Begin
  WAIT UNTIL Clk'EVENT AND clk = '1';
  case rs is
    when "00" =>
  r0 <= s;
    when "01" =>
  r1 <= s;
    when "10" =>
  r2 <= s;
    when "11" =>
```

```
    r3 <= s;

    end case;

        end process;



show_rs: seven_segment PORT MAP(s_temp(3), s_temp(2), s_temp(1), s_temp(0), s_temp(7), s_t
emp(6), s_temp(5), s_temp(4), a1, b1, c1, d1, e1, f1, g1, a0, b0, c0, d0, e0, f0, g0);

show_rt: seven_segment PORT MAP(t(3), t(2), t(1), t(0), t(7), t(6), t(5), t(4), a3, b3, c3, d3, e3, f3,
g3, a2, b2, c2, d2, e2, f2, g2);

show_bus: seven_segment PORT MAP(data(3), data(2), data(1), data(0), data(7), data(6), data(5),
data(4), a4, b4, c4, d4, e4, f4, g4, a5, b5, c5, d5, e5, f5, g5);

end main;
```

package. vhd

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

PACKAGE course8_package IS
	component seven_segment
		PORT( W0, X0, Y0, Z0,
			W1, X1, Y1, Z1: IN STD_LOGIC;
			a0, b0, c0, d0, e0, f0, g0,
			a1, b1, c1, d1, e1, f1, g1: OUT STD_LOGIC);
	END component seven_segment;
END course8_package;
```

seven_segment.vhd

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

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

ARCHITECTURE LogicFunc OF seven_segment 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 LogicFunc;
```