**微算機系統**

實驗四

組別： 108590451

班級、姓名與學號： 資工二 陸詠涵 108580451

資工二 陳佳均 108830035

日期： 2020.11.07

1. 實驗內容：

本次實驗主旨目標

利用上課所學的暫存器基本概念實際套用在VHDL程式邏輯上並燒錄在模板觀察實際結果。

<目標一>8bits暫存器

<目標二>16bits暫存器

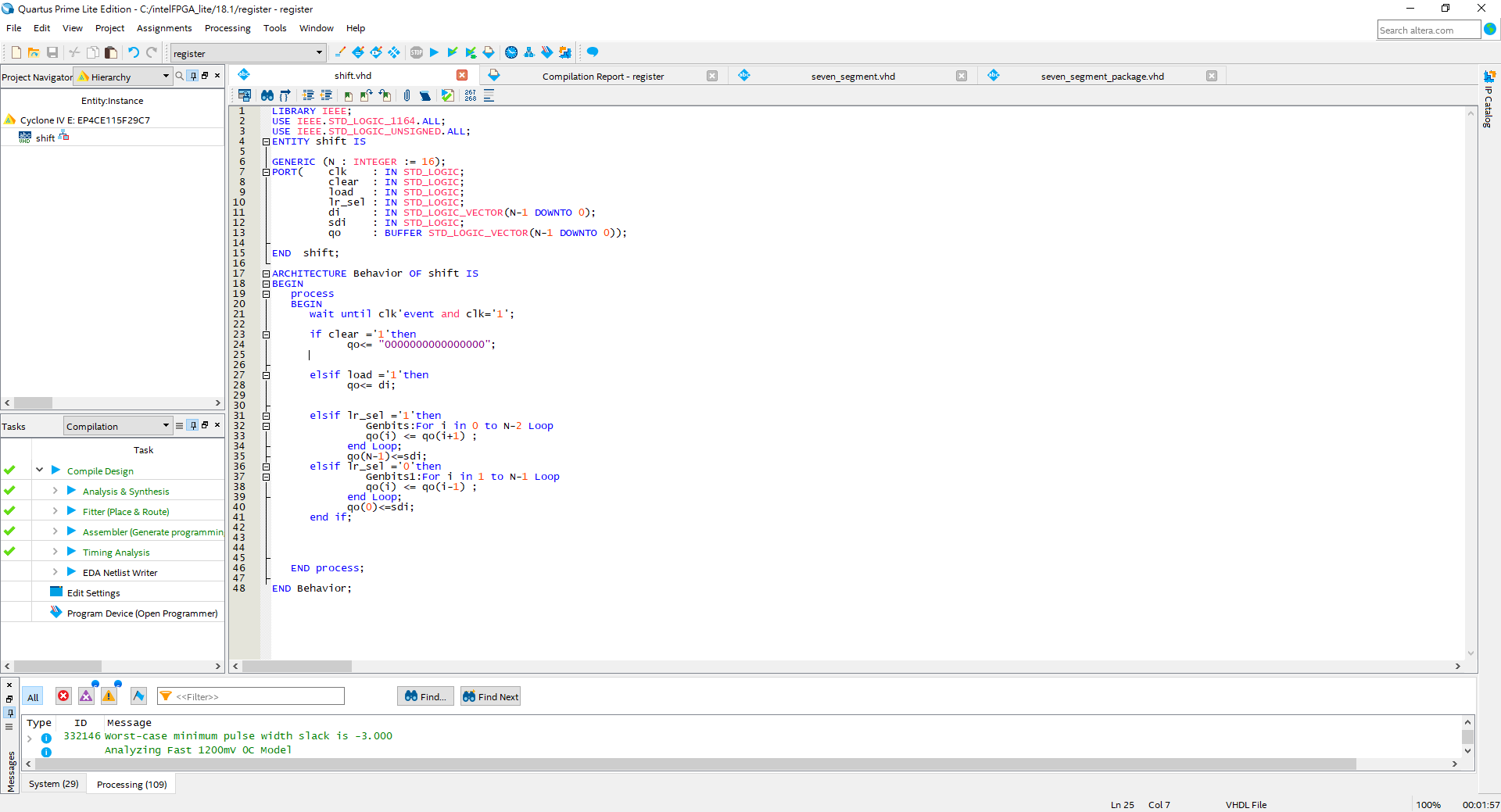
1. 實驗過程及結果：

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)。

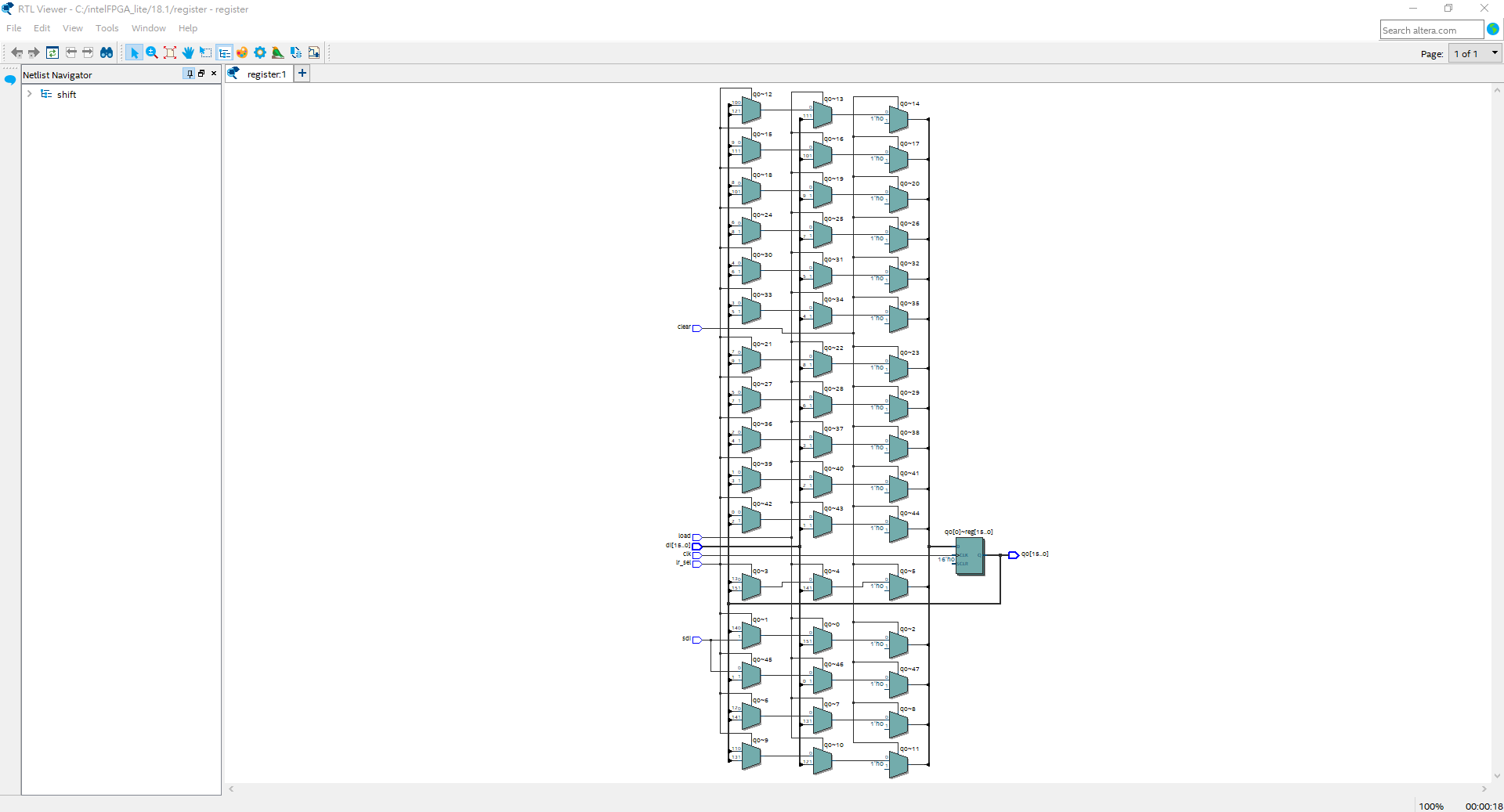
實驗步驟:

Step1:設計VHDL程式碼

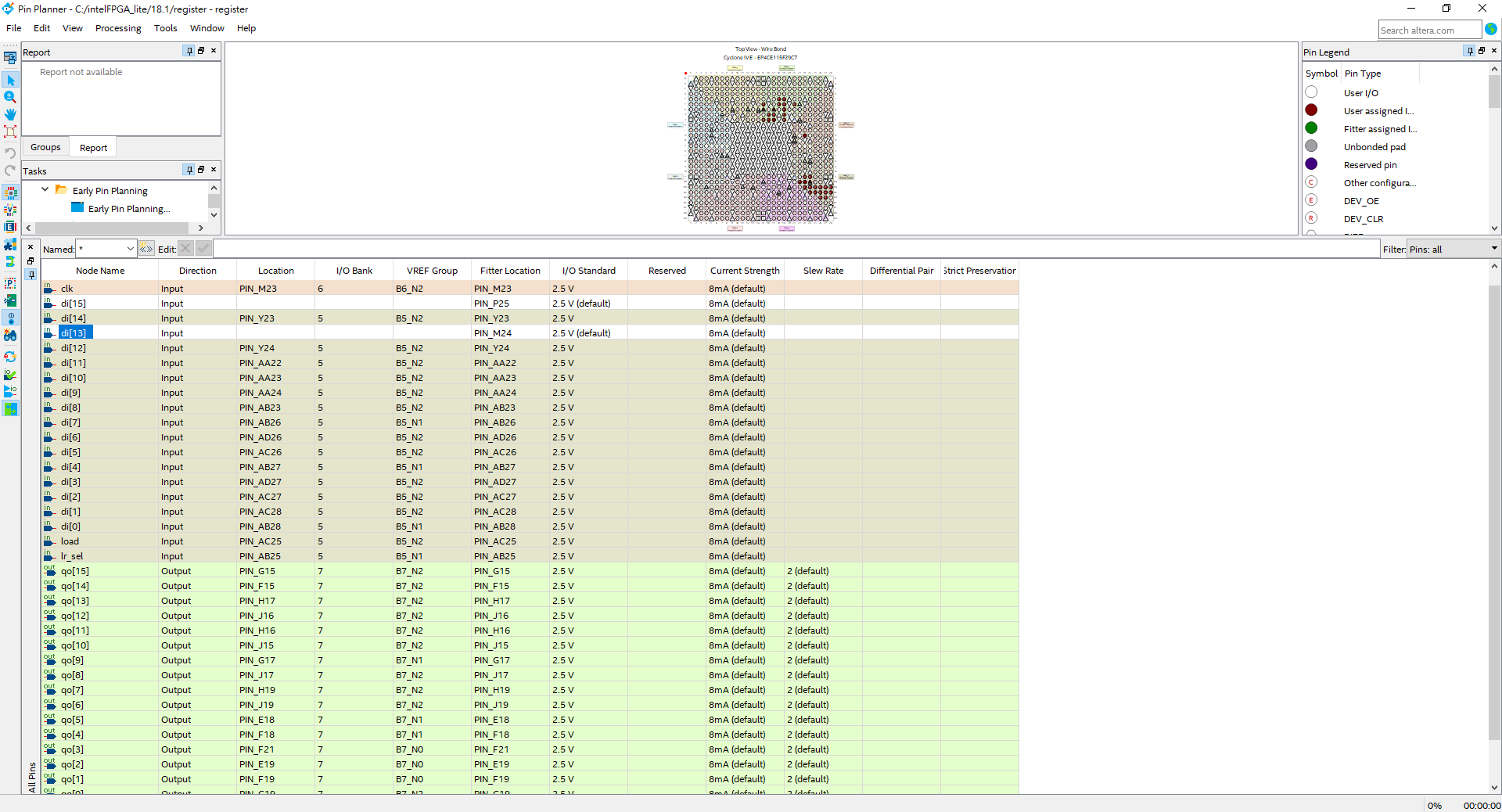


(picture 1) 16bits暫存器主程式碼

Step2:接腳位

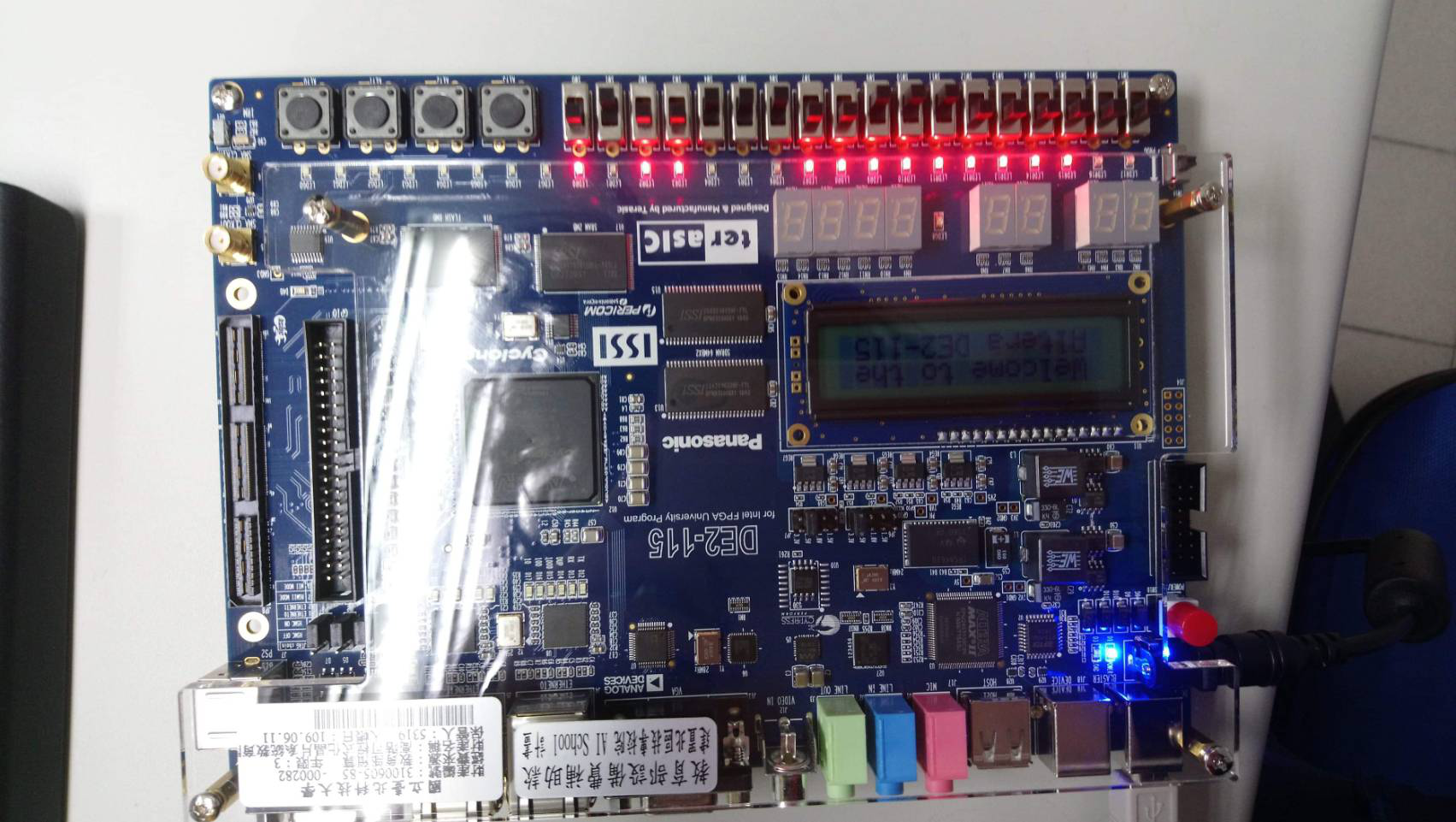


(picture 2) 16bits暫存器邏輯電路圖



(picture 3) 16bits暫存器所接的腳位

Step3:燒錄到模板中



(picture 4) 16bits暫存器的結果

1. 程式碼（請調整成最小行高，行高0點）

|  |
| --- |
| 目標一 |
| (程式碼)  **8bits暫存器**  library ieee;  use ieee.std\_logic\_1164.all;  --use work.seven\_segment\_display\_package.all;  entity SHIFT is  generic ( n: integer:=8);  port(  CLK,C0 :in std\_logic;  MD :in std\_logic\_vector(2 downto 0);  D :in std\_logic\_vector(n-1 downto 0);  QB :buffer std\_logic\_vector(n-1 downto 0);  CN :out std\_logic);  end entity SHIFT;  architecture BHV of SHIFT is  signal REG : std\_logic\_vector(n-1 downto 0);  signal CY : std\_logic;  begin  process(CLK,MD,C0)  begin  --if c0='1' then QB(7 downto 0)<= (others => '0');  --end if ;  if CLK'EVENT AND CLK='1' then  case MD is  when "000" =>REG(7 downto 0)=>(others => '0') ;  when "001" =>REG(0)<=C0;  for REG(i) in 7 downto 1 loop  REG(i) <= REG(i-1) ;  CY<=REG(7);  end loop;  --REG(7 downto 1)<=REG(6 downto 0); CY<=REG(7);  when "010" =>REG(7)<=C0;  REG(6 downto 0)<=REG(7 downto 1); CY<=REG(7);  when "011" =>REG(0)<=REG(7);  REG(7 downto 1)<=REG(6 downto 0); CY<=REG(7);  when "100" =>REG(7)<=REG(0);  REG(6 downto 0)<=REG(7 downto 1); CY<=REG(7);  when "101" =>REG(7 downto 0)<= D(7 downto 0);  when others =>REG<=REG; CY<=CY;  end case;  end if;  end process;  QB(7 downto 0)<=REG(7 downto 0); CN<=CY;    --output: seven\_segment\_display port map('0', result(6), result(5), CN,QB(7),QB(7 downto 0), QB(7 downto 0),QB(7 downto 0), QB(7 downto 0),QB(7 downto 0), QB(7 downto 0)) ;  end BHV; |
| 目標二 |
| (程式碼)  **16bits暫存器**  LIBRARY IEEE;  USE IEEE.STD\_LOGIC\_1164.ALL;  USE IEEE.STD\_LOGIC\_UNSIGNED.ALL;  ENTITY shift IS  GENERIC (N : INTEGER := 16);  PORT( clk : IN STD\_LOGIC;  clear : IN STD\_LOGIC;  load : IN STD\_LOGIC;  lr\_sel : IN STD\_LOGIC;  di : IN STD\_LOGIC\_VECTOR(N-1 DOWNTO 0);  sdi : IN STD\_LOGIC;  qo : BUFFER STD\_LOGIC\_VECTOR(N-1 DOWNTO 0));  END shift;  ARCHITECTURE Behavior OF shift IS  BEGIN  process  BEGIN  wait until clk'event and clk='1';    if clear ='1'then  qo<= "0000000000000000";    elsif load ='1'then  qo<= di;    elsif lr\_sel ='1'then  Genbits:For i in 0 to N-2 Loop  qo(i) <= qo(i+1) ;  end Loop;  qo(N-1)<=sdi;  elsif lr\_sel ='0'then  Genbits1:For i in 1 to N-1 Loop  qo(i) <= qo(i-1) ;  end Loop;  qo(0)<=sdi;  end if;      END process;  END Behavior; |

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）

資工二108590451 陸詠涵

這次老師要求用PROCESS內的FOR LOOP來設計程式碼，所以整體來看，程式碼變得簡潔有力。而且加上有GENERIC的加入，要將8BITS變成16BITS的暫存器只要改一些小地方，馬上就可以完成了。這次困難點應該是在接腳位的部分，一開始我們在設定模板那邊設錯了，導致找不到腳位，雖然最後有改過來，但不知道為什麼我們這台電腦不能直接打腳位，所以我們只能一個一個選，因此耗費比較多時間在上面。

資工二108830035 陳佳均

這次前面都做得還不錯，但是在腳位都接完硬體也燒入完之後才發現我們的clear功能跟我們原本預想的撥法不太一樣。結果就在那邊卡了很久，後來才發現原來是順序的問題，而且還是貴人相助。再來就是8位元轉16位元的部分，那時候我以為是要直接輸入16位元，還因為輸入的腳位不夠在頭大。結果，最後竟然只需要8位元的輸入，是輸出才要16個。