**微算機系統**

實驗六

第16組

資工二 陸詠涵 108580451

資工二 陳佳均 108830035

日期： 2020.12.20

1. 實驗內容：

本次實驗主旨目標

設計出一個16bits除法器，並以clock可進行觸發的六個狀態之狀態機。並且學習透過TYPE設定State\_type來進行狀態的切換。最後再將remainder後面8bits和quotion顯示在燒錄版上。

<目標一>16bits除法器

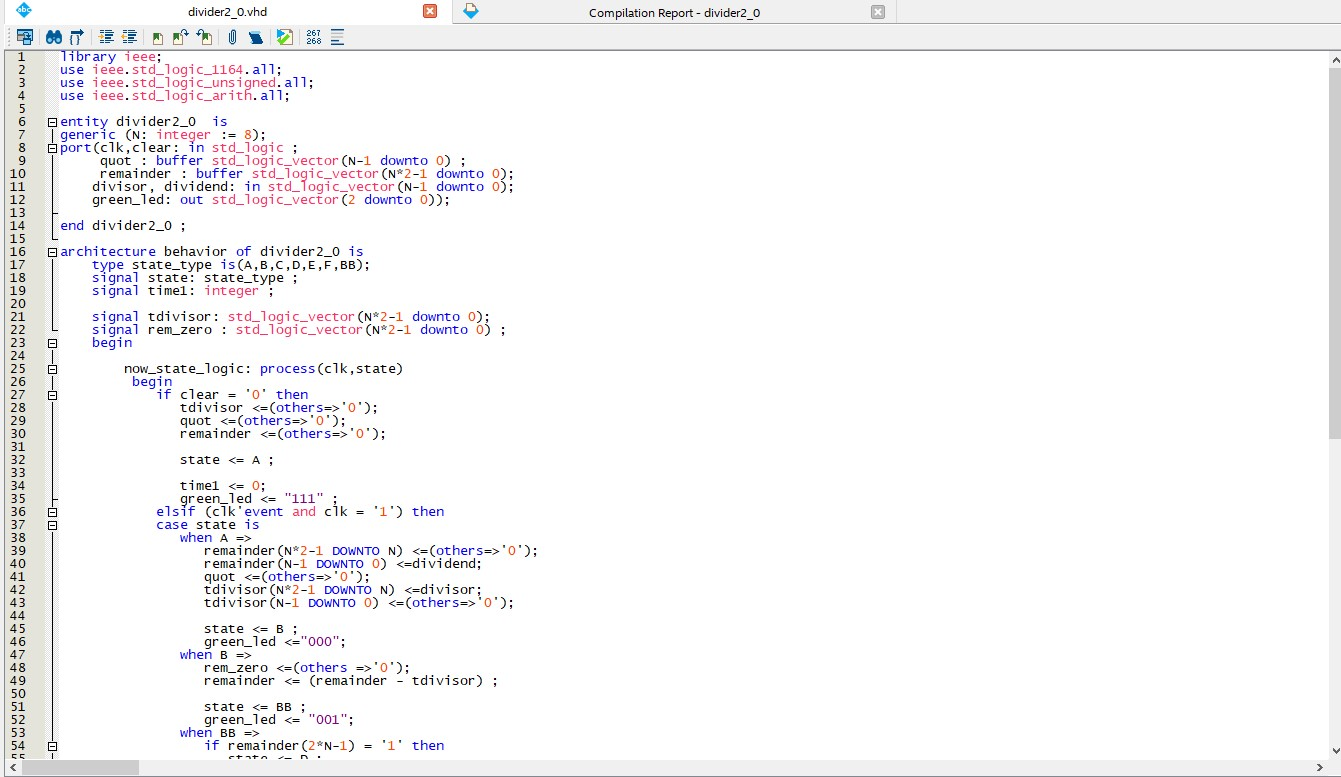
1. 實驗過程及結果：

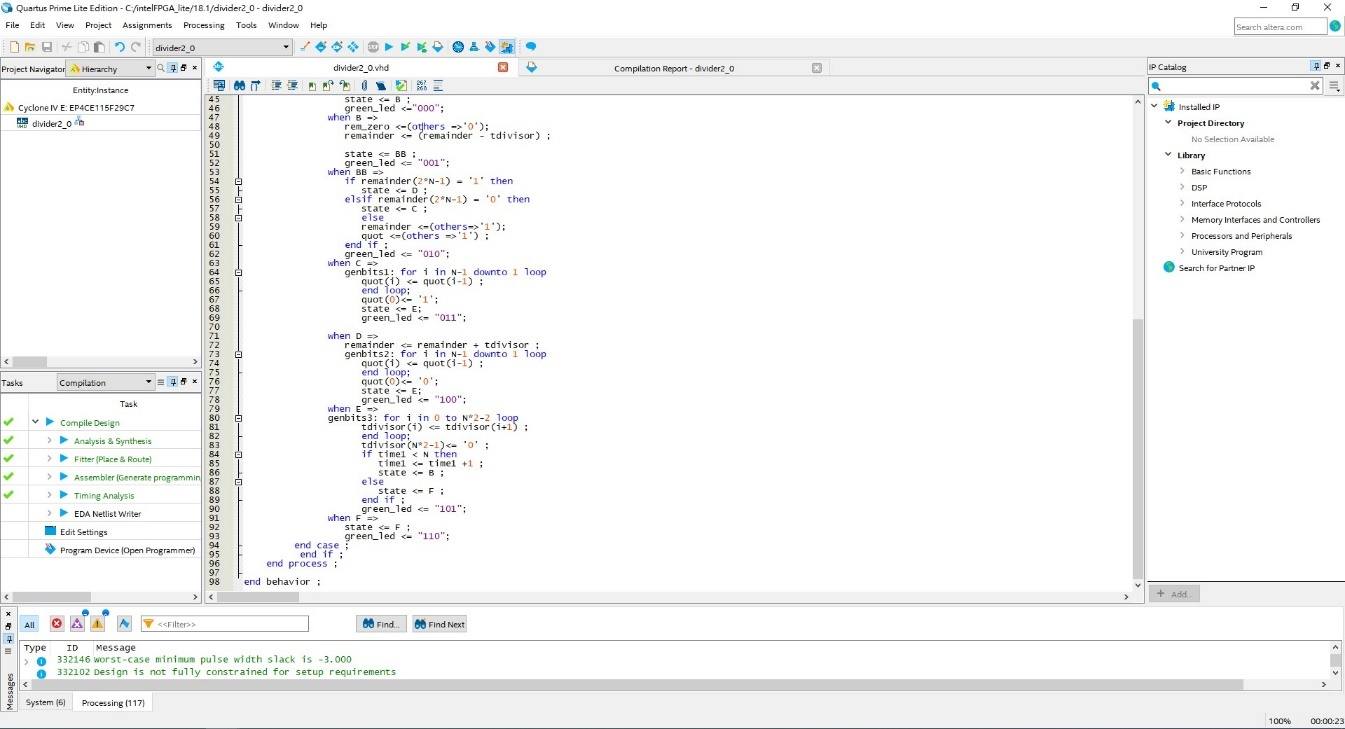
撰寫你如何完成本次作業的流程方法。

請附上結果圖片(放置於此部分即可)。

實驗步驟:

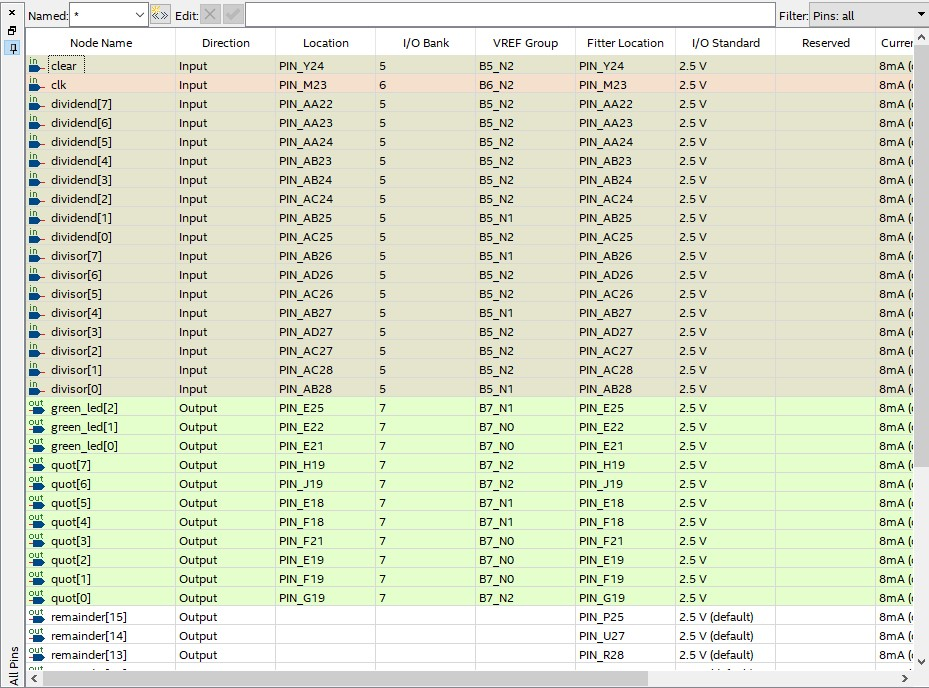
Step1:設計VHDL程式碼



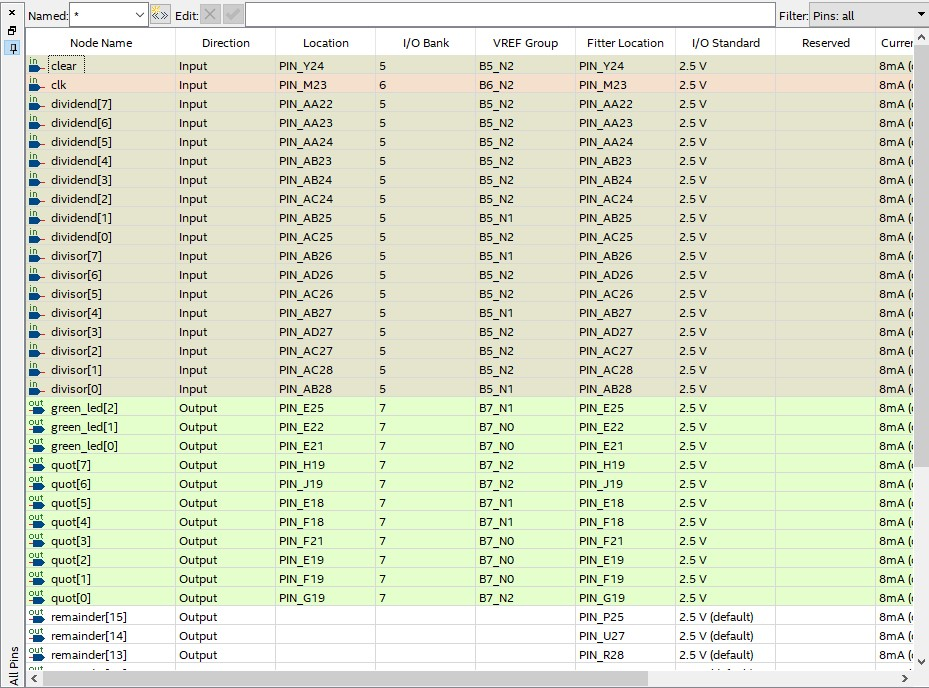


(picture 1) 16bits除法器主程式碼

Step2:接腳位

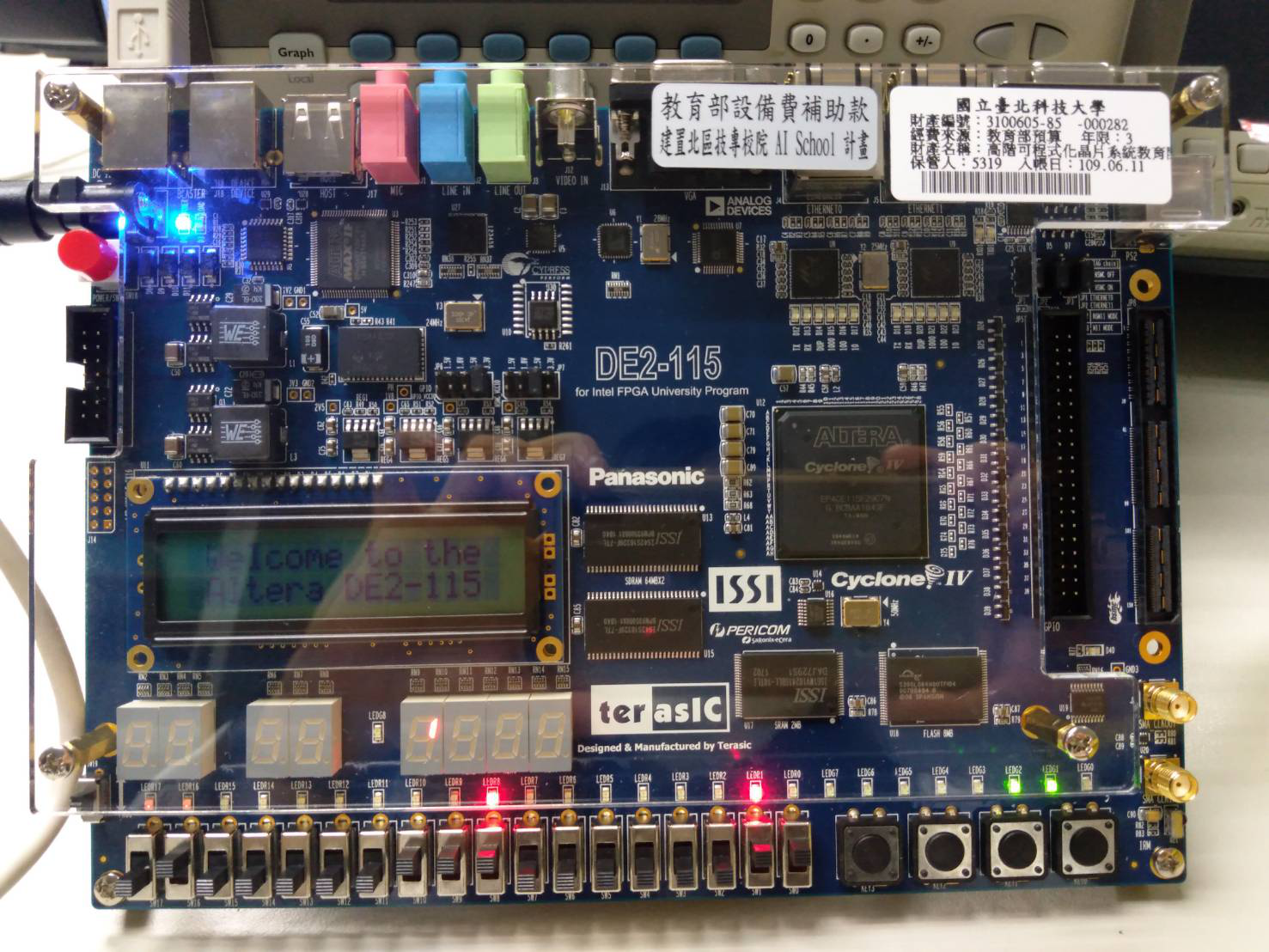


(picture 2) 16bits除法器邏輯電路圖



(picture 3) 16bits除法器所接的腳位

Step3:燒錄到模板中



(picture 4) 16bits除法器的結果

1. 程式碼（請調整成最小行高，行高0點）

|  |
| --- |
| 目標一 |
| library ieee;  use ieee.std\_logic\_1164.all;  use ieee.std\_logic\_unsigned.all;  use ieee.std\_logic\_arith.all;  entity divider2\_0 is  generic (N: integer := 8);  port(clk,clear: in std\_logic ;  quot : buffer std\_logic\_vector(N-1 downto 0) ;  remainder : buffer std\_logic\_vector(N\*2-1 downto 0);  divisor, dividend: in std\_logic\_vector(N-1 downto 0);  green\_led: out std\_logic\_vector(2 downto 0));    end divider2\_0 ;  architecture behavior of divider2\_0 is  type state\_type is(A,B,C,D,E,F,BB);  signal state: state\_type ;  signal time1: integer ;  signal tdivisor: std\_logic\_vector(N\*2-1 downto 0);  signal rem\_zero : std\_logic\_vector(N\*2-1 downto 0) ;  begin    now\_state\_logic: process(clk,state)  begin  if clear = '0' then  tdivisor <=(others=>'0');  quot <=(others=>'0');  remainder <=(others=>'0');    state <= A ;    time1 <= 0;  green\_led <= "111" ;  elsif (clk'event and clk = '1') then  case state is  when A =>  remainder(N\*2-1 DOWNTO N) <=(others=>'0');  remainder(N-1 DOWNTO 0) <=dividend;  quot <=(others=>'0');  tdivisor(N\*2-1 DOWNTO N) <=divisor;  tdivisor(N-1 DOWNTO 0) <=(others=>'0');    state <= B ;  green\_led <="000";  when B =>  rem\_zero <=(others =>'0');  remainder <= (remainder - tdivisor) ;    state <= BB ;  green\_led <= "001";  when BB =>  if remainder(2\*N-1) = '1' then  state <= D ;  elsif remainder(2\*N-1) = '0' then  state <= C ;  else  remainder <=(others=>'1');  quot <=(others =>'1') ;  end if ;  green\_led <= "010";  when C =>  genbits1: for i in N-1 downto 1 loop  quot(i) <= quot(i-1) ;  end loop;  quot(0)<= '1';  state <= E;  green\_led <= "011";    when D =>  remainder <= remainder + tdivisor ;  genbits2: for i in N-1 downto 1 loop  quot(i) <= quot(i-1) ;  end loop;  quot(0)<= '0';  state <= E;  green\_led <= "100";  when E =>  genbits3: for i in 0 to N\*2-2 loop  tdivisor(i) <= tdivisor(i+1) ;  end loop;  tdivisor(N\*2-1)<= '0' ;  if time1 < N then  time1 <= time1 +1 ;  state <= B ;  else  state <= F ;  end if ;  green\_led <= "101";  when F =>  state <= F ;  green\_led <= "110";  end case ;  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

因為上次的除法狀態器我們這組是用兩個process寫的，所以這次16bits除法器我們一開始也是用這個方式，一個單純寫狀態，一個寫要執行的指令。但是，除錯了很久，我們依然沒過，所以的邏輯我們問助教都沒問題，可能是程式碼排列順序，導致時間的誤差，沒有同時進行而錯誤，所以後來我們就改只有一個程序，結果就成功了，可是還是想知道為什麼寫兩個程序不會過。

資工二108830035 陳佳均

這次的實驗在我們搞懂除法原理之後做得都還挺順，速度也蠻快。原本預期可以提早結束，結果不知為何，除法器的順序一直出錯，一直跟我們自己人工跟著程式跑出來的結果不一樣。後來經過我們的再三除錯還是無法正確運作。最後，礙於時間的關係，我們還是放棄原本兩個process的版本，把他們合而為一，這才做出來。但還是想知道，究竟是哪裡出了問題。