**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

設計出一個基於除法器狀態，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。

<目標一>除法狀態器

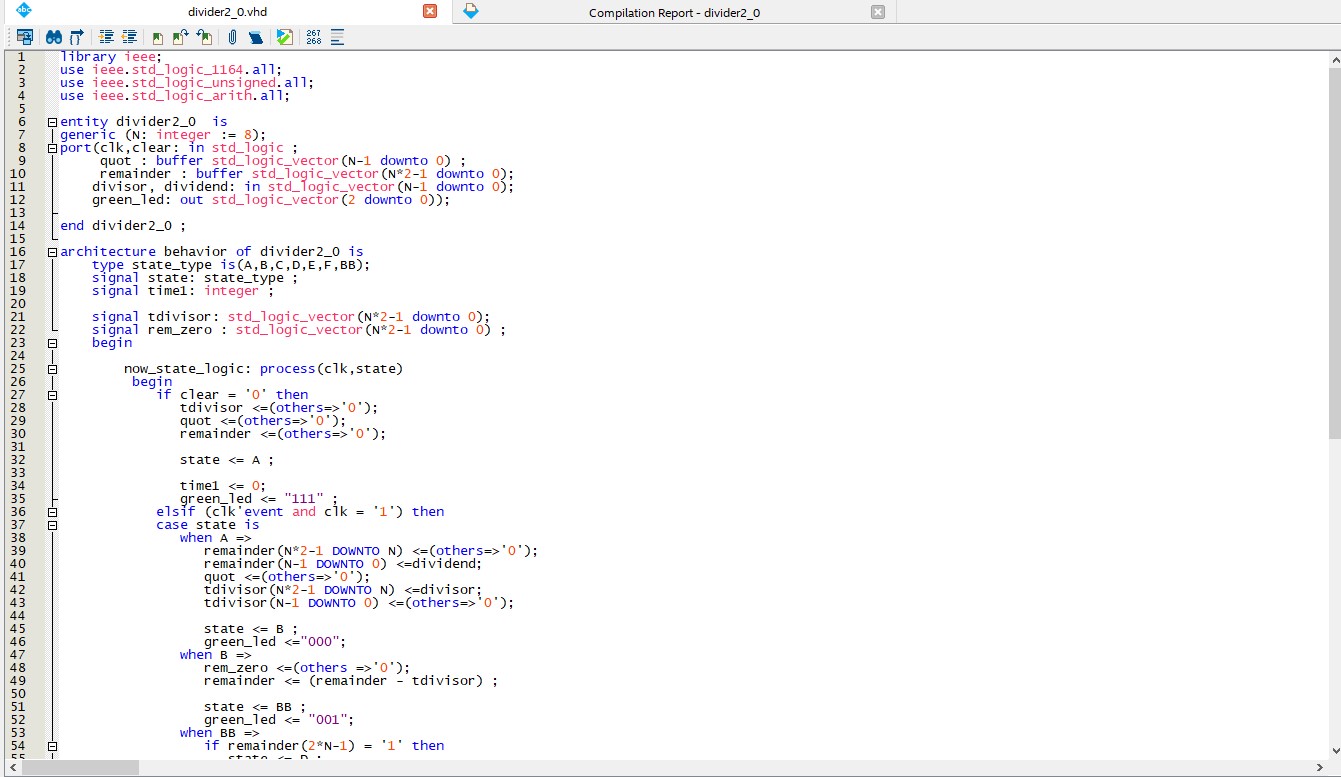
1. 實驗過程及結果：

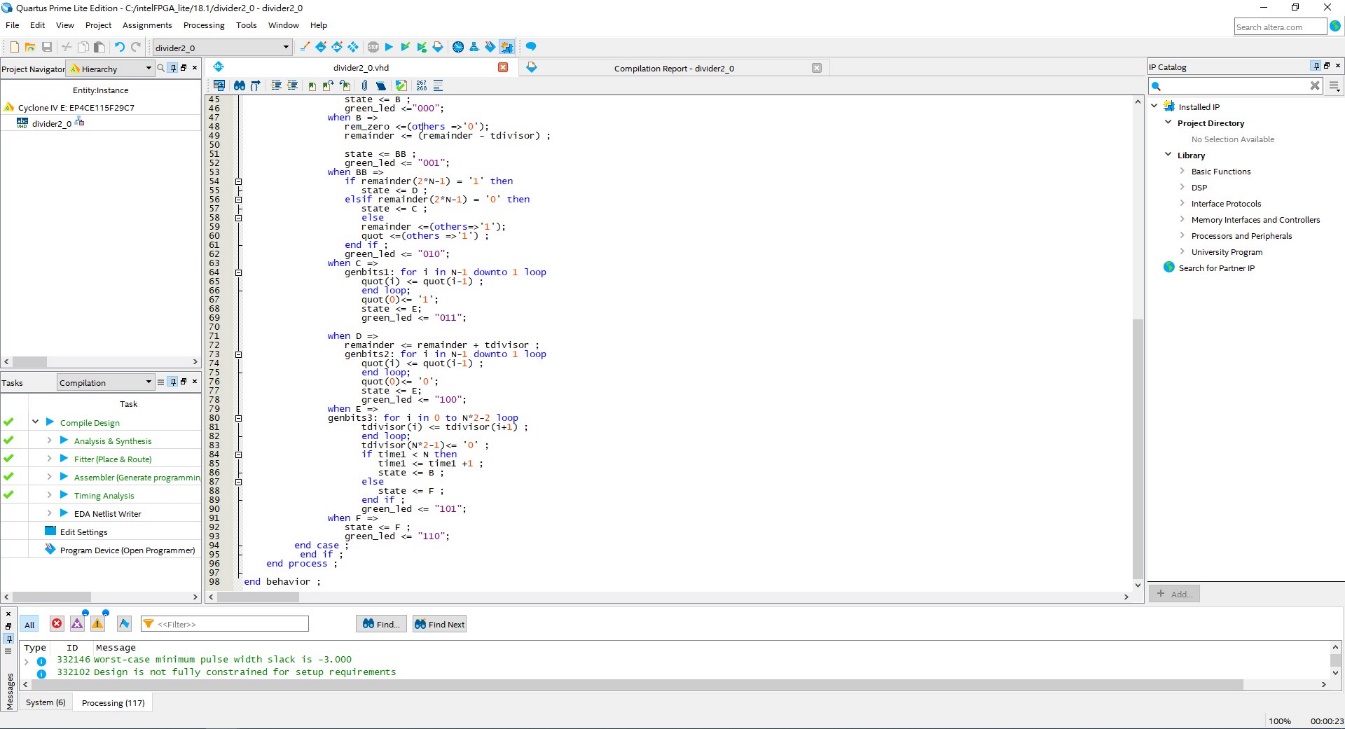
撰寫你如何完成本次作業的流程方法。

請附上結果圖片(放置於此部分即可)。

實驗步驟:

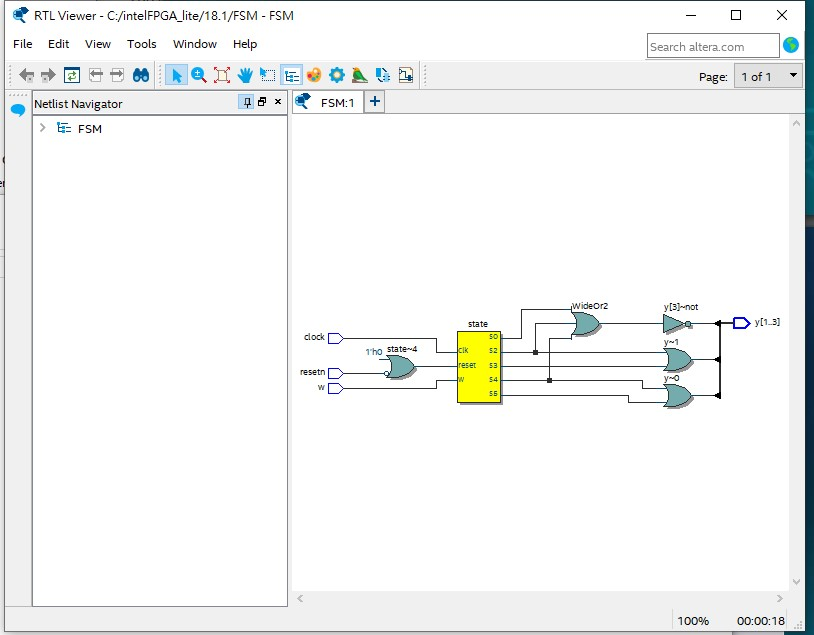
Step1:設計VHDL程式碼





(picture 1) 除法狀態器主程式碼

Step2:接腳位



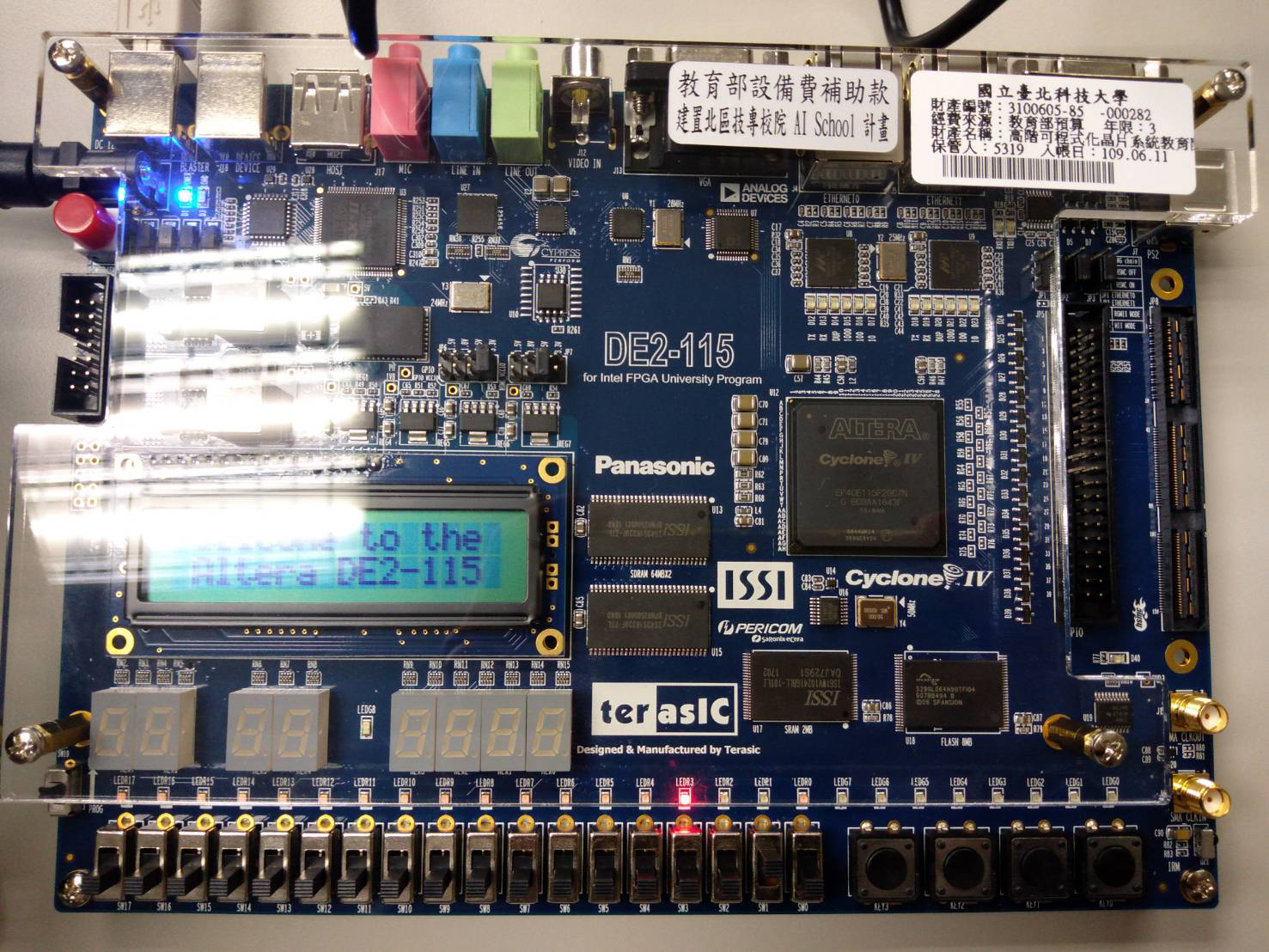
(picture 2) 除法狀態器邏輯電路圖

一張含有 桌 的圖片

自動產生的描述

(picture 3) 除法狀態器所接的腳位

Step3:燒錄到模板中



(picture 4) 除法狀態器的結果

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;  entity FSM is  port(clock, w, resetn : in std\_logic ;  y : out std\_logic\_vector(1 to 3));  end FSM ;    architecture behavior of FSM is  type state\_type is (s0,s1,s2,s3,s4,s5);  signal state : state\_type ;  begin  next\_state\_logic: process(resetn ,clock)  begin  if resetn = '0' then  state <= s0 ;  elsif (clock'event and clocl = '1') then  case state is  when s0 =>  if w = '1' then  state <= s1 ;  else  state <= s0 ;  end if ;  when s1 =>  if w = '1' then  state <= s3 ;  else  state <= s2 ;  end if ;  when s2 =>  state <= s4 ;  when s3 =>  state <= s4 ;  when s4 =>  if w = '0' then  state <= s1 ;  else  state <= s5 ;  end if ;  when s5 =>  state <= s1 ;  end case ;  end if ;  end process;  output\_logic : process(state, w)  begin  case state is  when s0 =>  y <= "000" ;  when s1 =>  y <= "001" ;  when s2 =>  y <= "010" ;  when s3 =>  y <= "011" ;  when s4 =>  y <= "100" ;  when s5 =>  y <= "101" ;  end case ;  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

這次的實驗我們這組是在考完期中考後做的，所以我們在設計程式時感覺有變快很多，越來越了解語法該如何使用，這次實驗的程式明顯比之前少很多，但多了新的語法type的使用，所以要特別注意。然後這次我們用了兩個process，一個是下一個狀態的設定，一個是輸出結果的。但是，我們這個程式碼比較像暴力解法，可能之後我們會再討論看看能不能設計更精簡的程式碼。

資工二108830035 陳佳均

這次實驗雖然不是很複雜，但因為之前我們的重點都比較放在準備段考，所以稍微忽略了這次的實驗。因此我們這次的實驗是採取暴力破解法，也就是把所有的狀態都列出來，然後直接告訴程式應該要走哪裡，而不是交由程式自己判斷。因此，之後我們還會再找時間研究type的真正用法。原因是因為我們這種寫法，我個人是覺得完全不用用到type啦，就是一種硬要用的感覺。