

微算機系統

小組專案報告

實驗五：

N位元左/右移位萬用暫存器

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11月16日

一、實驗內容：

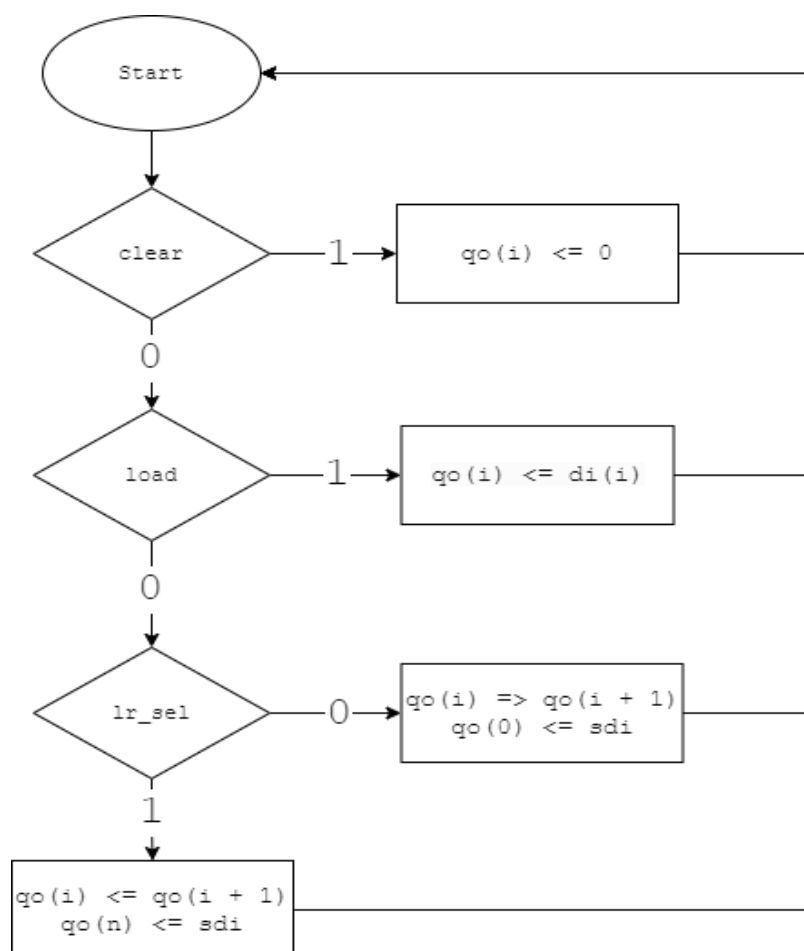
1. 設計出一個可輸出8位元並左/右移的移位暫存器，且必須使用GENERIC來完成此實驗。
2. 首先判斷clk是否正緣觸發，clk = “1” 觸發，反之不變。
3. 設定clear決定是否清空暫存器，clear = “1” 清除，反之不變。
4. 設定load決定是否平行輸入di，load = “1” 可平行載入，反之不變。
5. 設定lr_sel決定左/右位移，lr_sel = “1” 左移，lr_sel = “0” 右移。
6. 使用FOR LOOP產生左/右位移。
7. 將sdi串列輸入最左/右邊位元，sdi = “0” 補0，sdi = “1” 補1。
8. 測試時，將改變clk，並以指撥開關平行輸入di，並將設定lr_sel，並且串列輸入sdi，而8顆LED燈上必須可正確顯示出暫存器qo之結果。

Variable	Pin Location	Signal Name
di(0)	PIN_AB28	SW[0]
di(1)	PIN_AC28	SW[1]
di(2)	PIN_AC27	SW[2]
di(3)	PIN_AD27	SW[3]
di(4)	PIN_AB27	SW[4]
di(5)	PIN_AC26	SW[5]
di(6)	PIN_AD26	SW[6]
di(7)	PIN_AB26	SW[7]
load	PIN_AB25	SW[9]
lr_sel	PIN_AC24	SW[10]
clear	PIN_AB24	SW[11]
sdi	PIN_AB23	SW[12]
clk	PIN_M23	KEY[0]

qo(0)	PIN_G19	LED[0]
qo(1)	PIN_F19	LED[1]
qo(2)	PIN_E19	LED[2]
qo(3)	PIN_F21	LED[3]
qo(4)	PIN_F18	LED[4]
qo(5)	PIN_E18	LED[5]
qo(6)	PIN_J19	LED[6]
qo(7)	PIN_H19	LED[7]

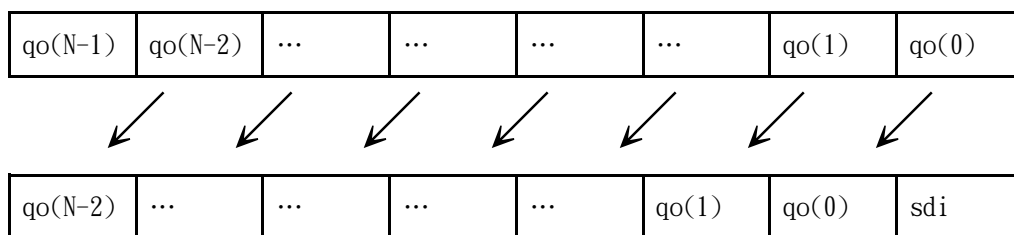
二、實驗過程及結果：

(一) 預期實驗結果的流程示意圖

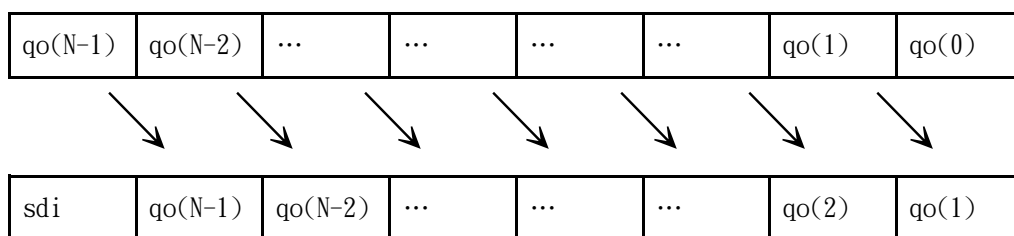


(二) 設計電路示意圖

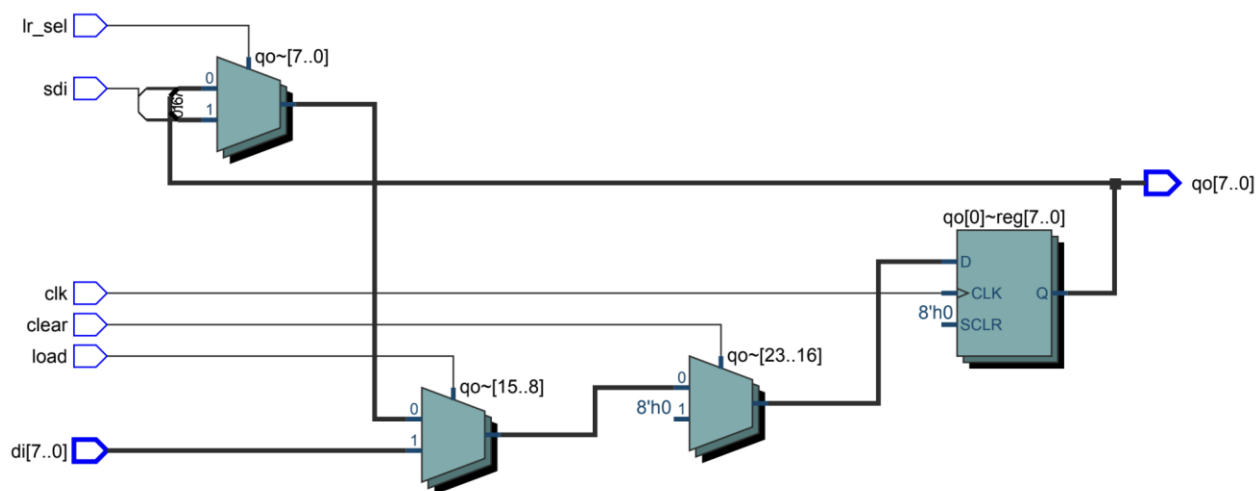
1. $lr_sel = 0$: 左移



2. $lr_sel = 1$: 右移

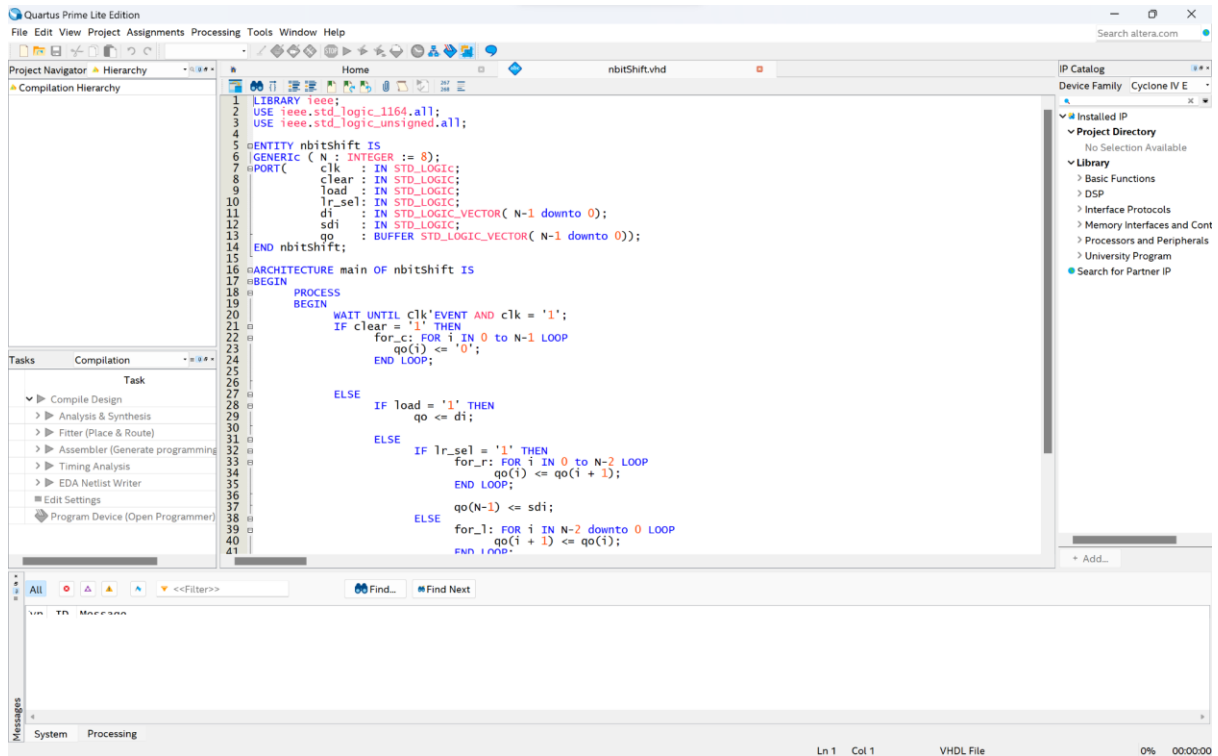


(三) 卡諾圖化簡後設計出的電路

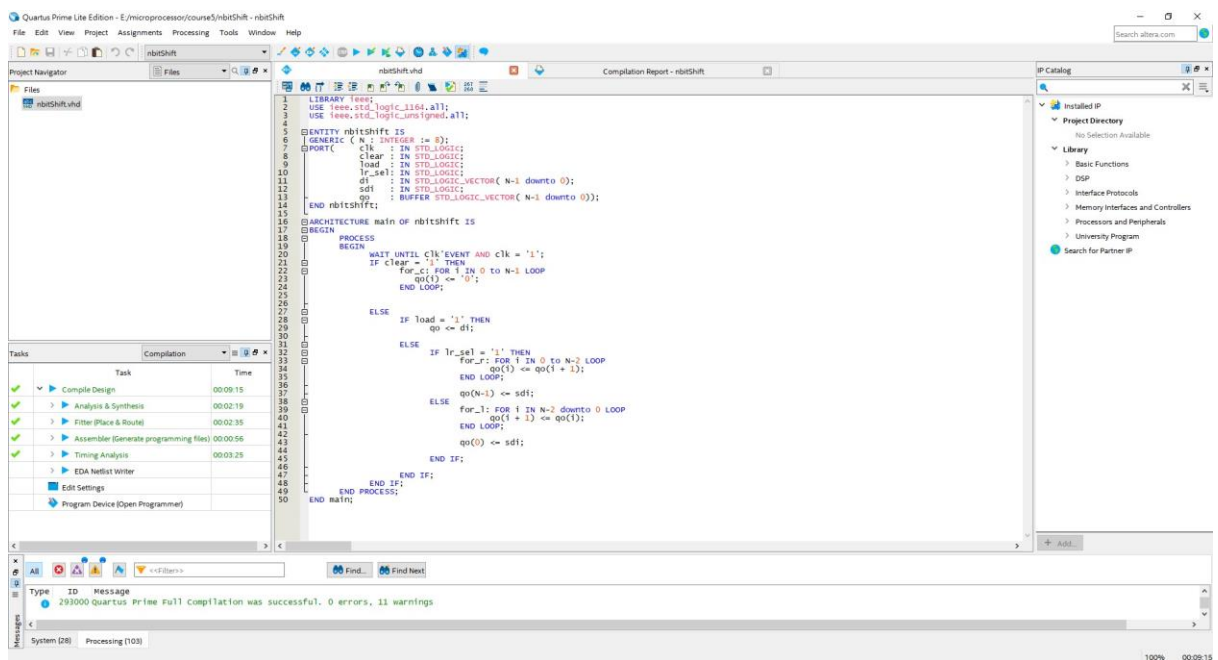


(四) 實驗過程

(1) 撰寫程式碼



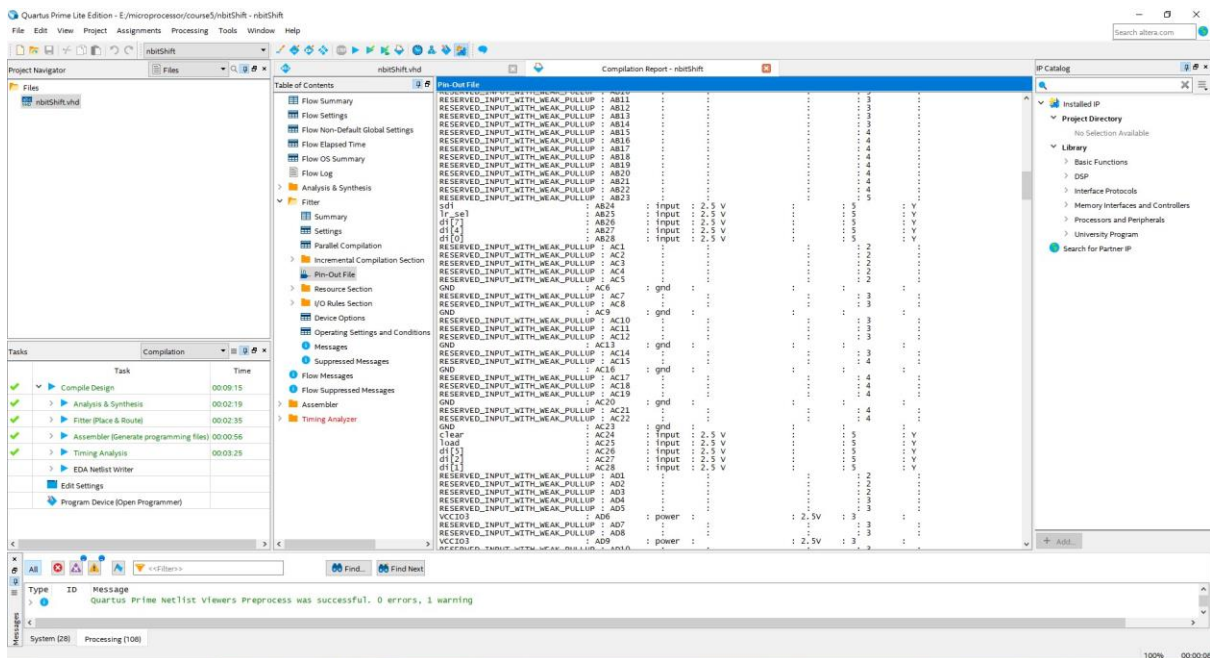
(2) 編譯成功



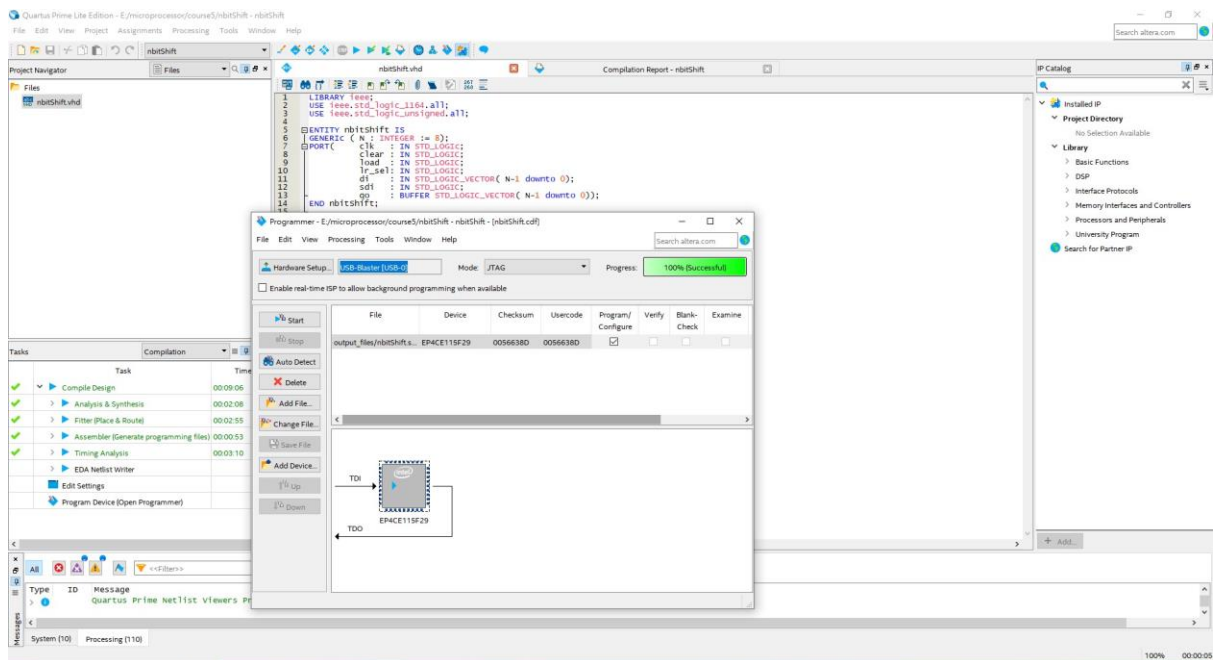
(3) 接腳位



(4) 確認接線於正確腳位



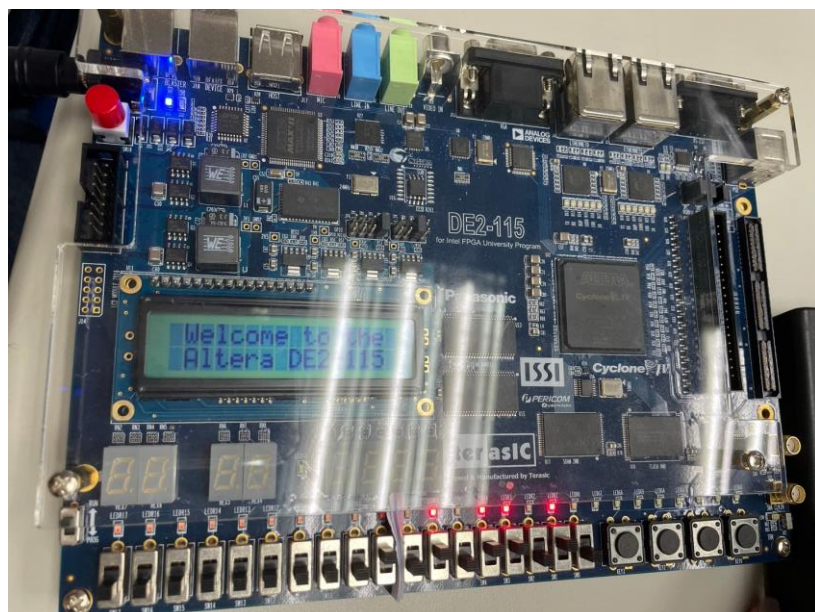
(5) 燒錄視窗設定及燒錄成功畫面



(五) 實驗結果

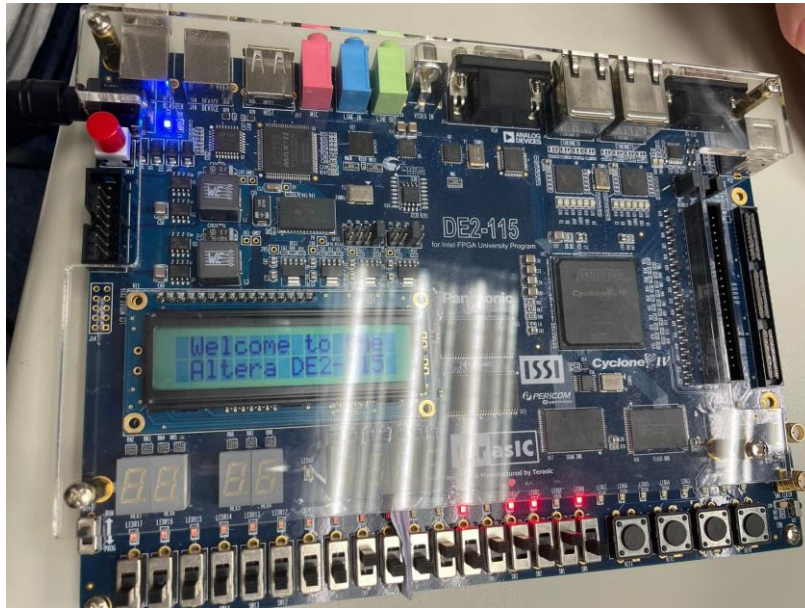
(1) di: 01011010 ; clear: 0 ; load: 1

輸出結果 01011010



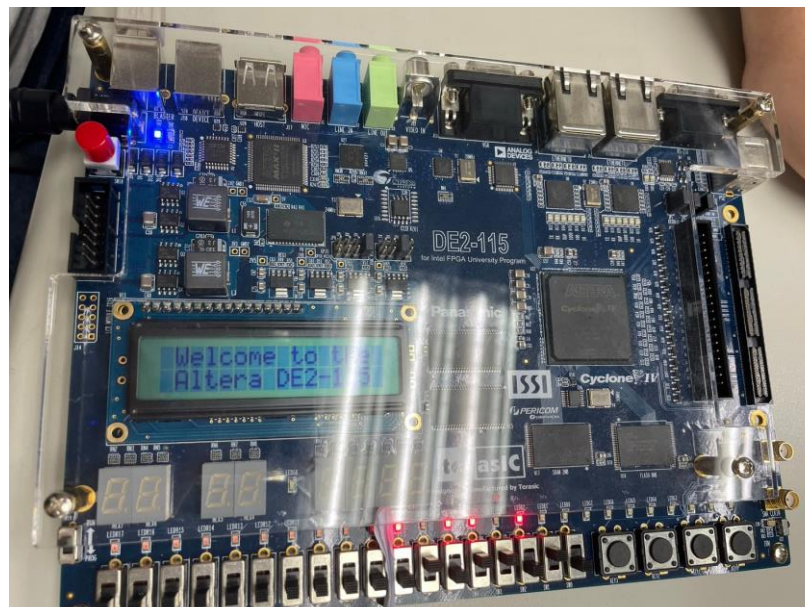
(2) qo: 01011010 ; load: 0 ; clear: 0 ; lr_sel: 1 ; sdi: 0

輸出結果 00101101



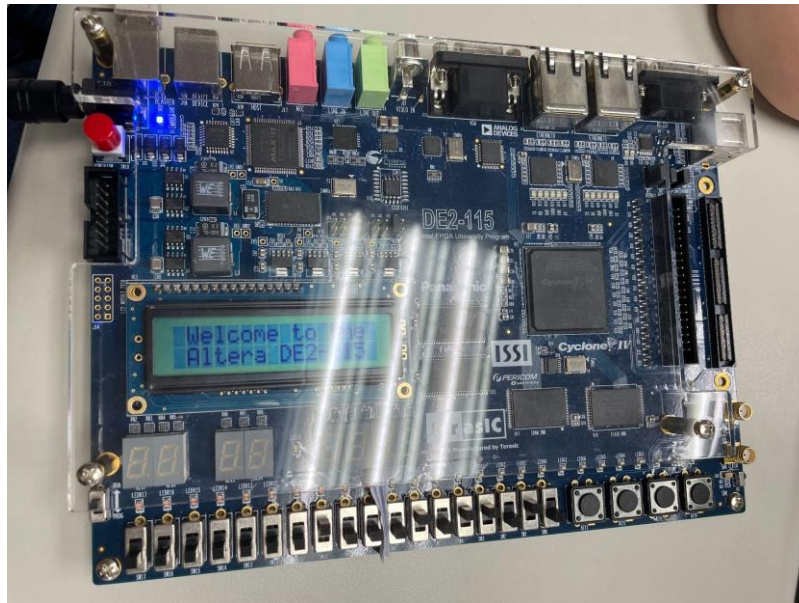
(3) qo: 01011010 ; load: 0 ; clear: 0 ; lr_sel: 0 ; sdi: 0

輸出結果 10110100



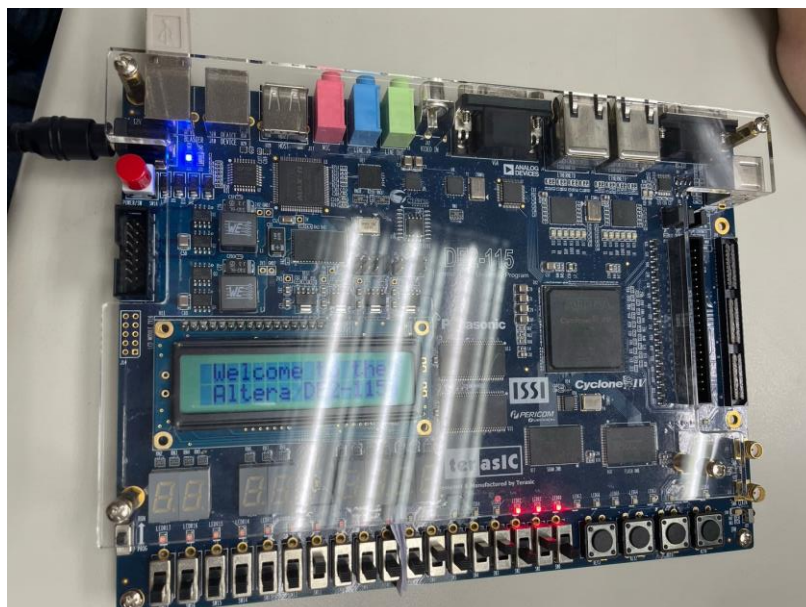
(4) clear: 1

輸出結果 00000000



(5) qo: 00000011 ; load: 0 ; clear: 0 ; lr_sel: 0 ; sdi: 1

輸出結果 00000111



● 實際操作影片連結；

<https://drive.google.com/file/d/1QXcdVnAiu27K8vsx1nYF-CPpHc8c6D7W/view?usp=sharing>

三、程式碼

nbitShift.vhd

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

ENTITY nbitShift IS
  GENERIC ( N : INTEGER := 8);
  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 nbitShift;

ARCHITECTURE main OF nbitShift IS
BEGIN
  PROCESS
  BEGIN
    WAIT UNTIL Clk'EVENT AND clk = '1';
    IF clear = '1' THEN
      for_c: FOR i IN 0 to N-1 LOOP
        qo(i) <= '0';
      END LOOP;
    ELSE
      IF load = '1' THEN
        qo <= di;
      END IF;
    END IF;
  END PROCESS
END main;
```

```

ELSE
    IF lr_sel = '1' THEN
        for_r: FOR i IN 0 to
            qo
        (i) <= qo(i + 1);
    END LOOP;
    qo(N-1) <= sdi;
ELSE
    for_l: FOR i IN N-2 d
        qo(i
    + 1) <= qo(i);
    END LOOP;
    qo(0) <= sdi;
END IF;
END IF;
END IF;
END PROCESS;
END main;

```