**微算機系統**

實驗五

組別： 14

班級、姓名與學號：四資二　洪晟毅　104590048

　　　　　　　　　四資二　梁皓鈞　104360098

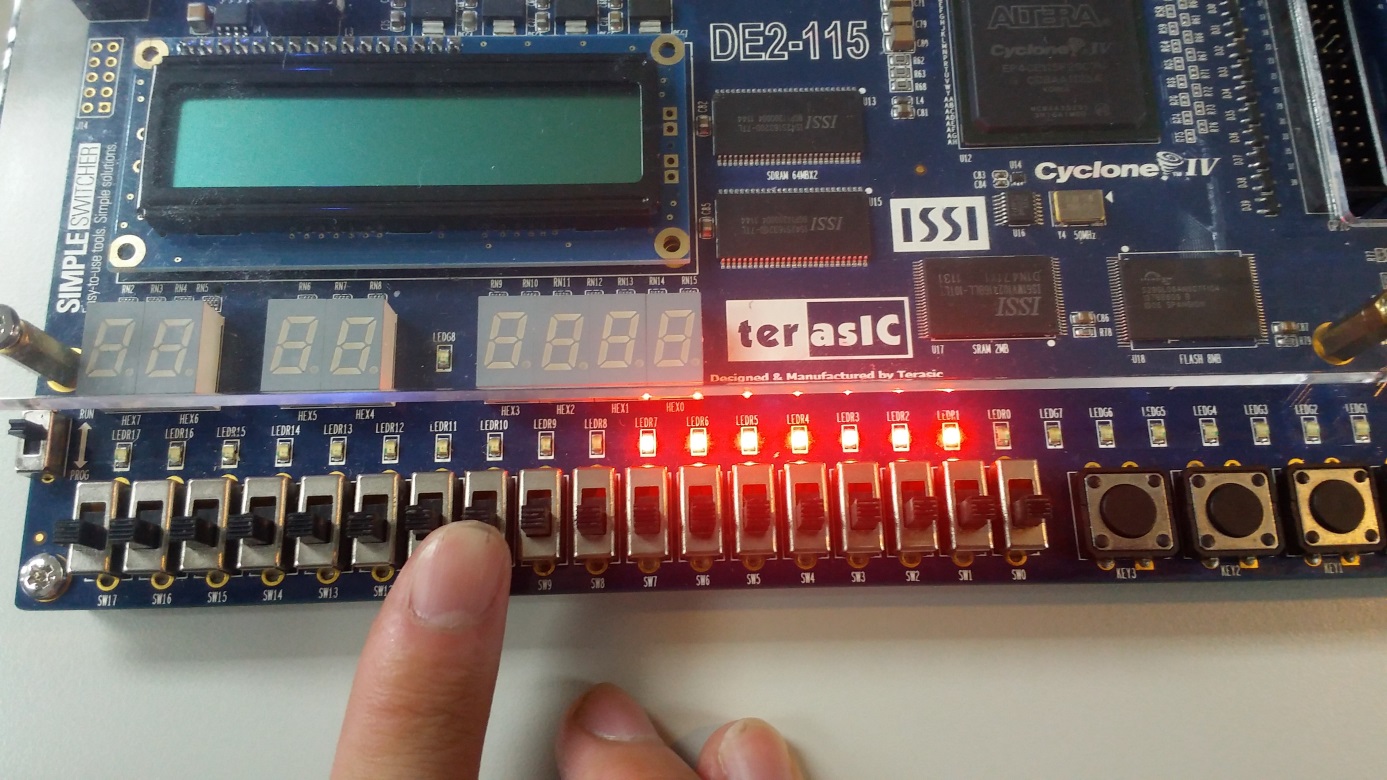
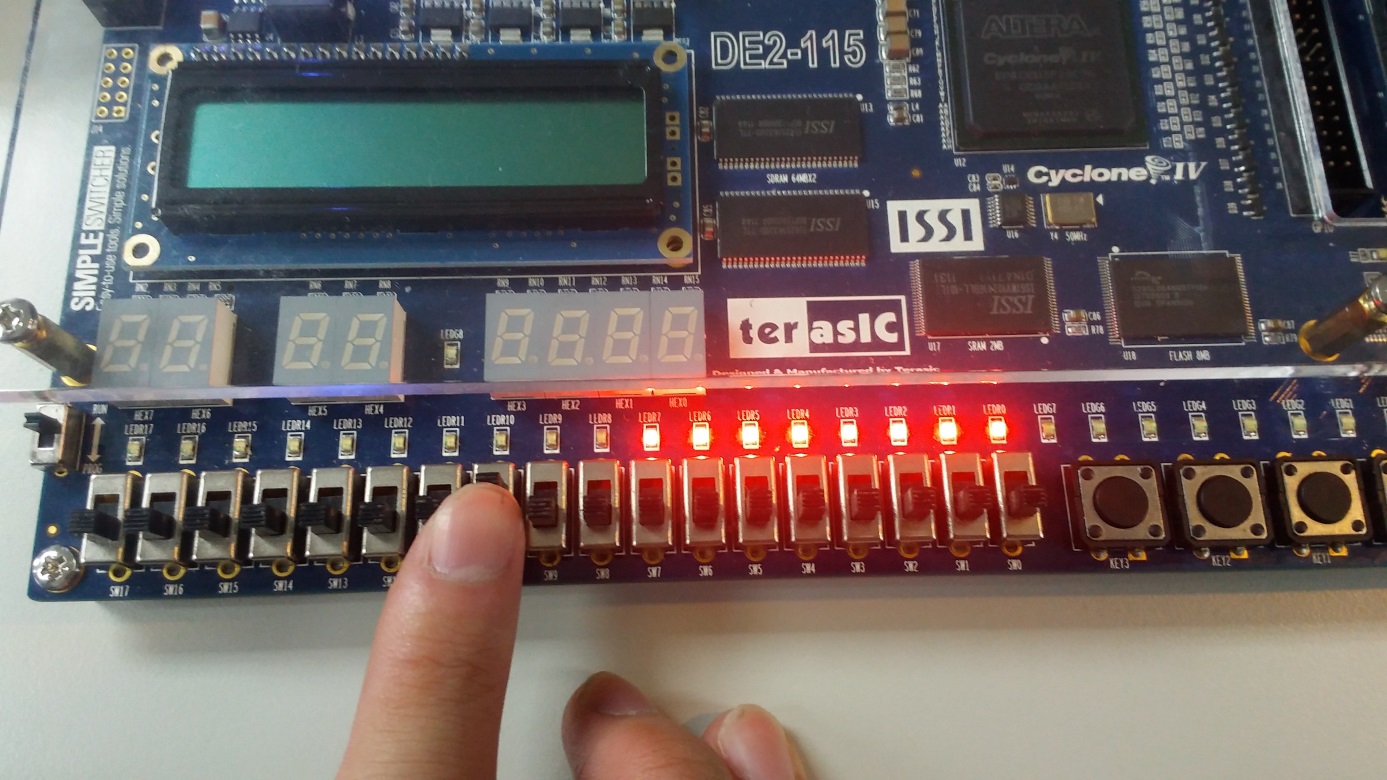
日期： 2016.12.19

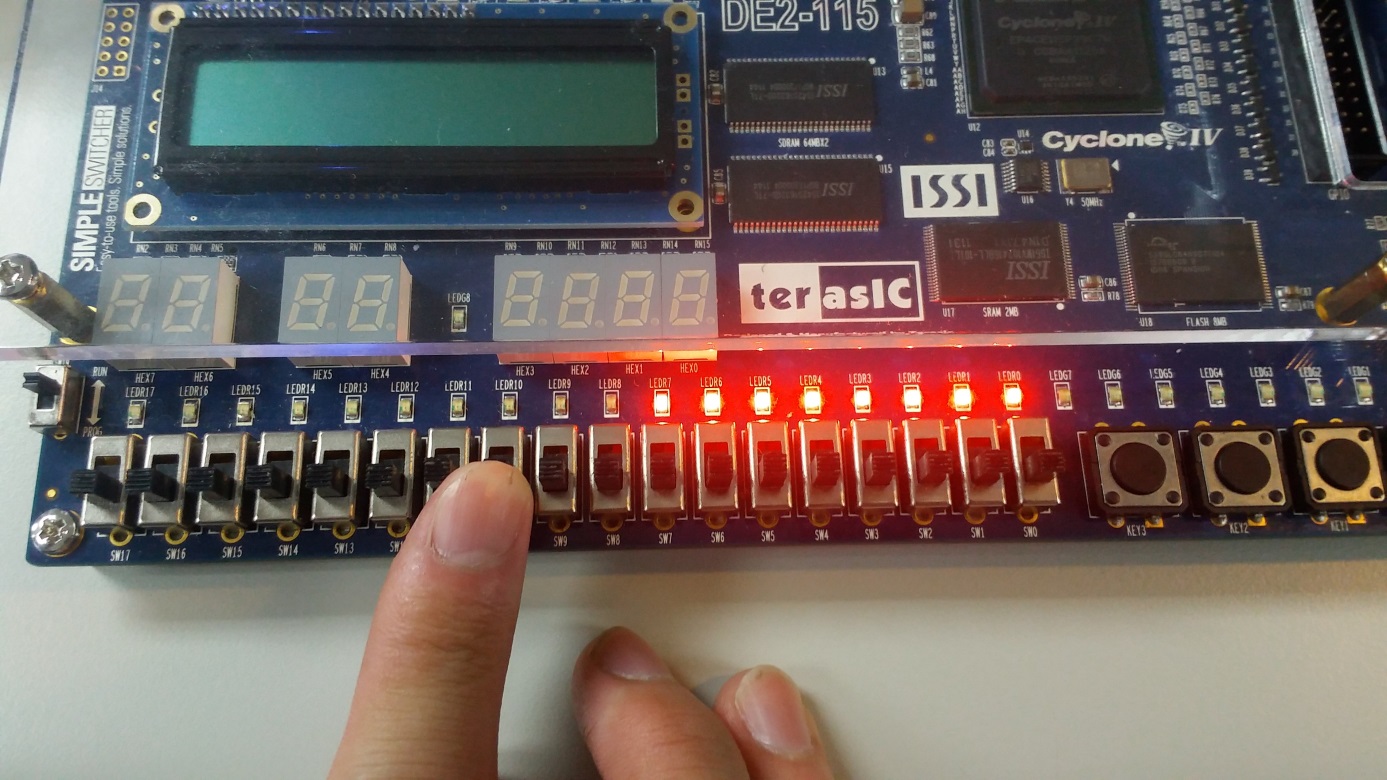
1. 實驗內容：

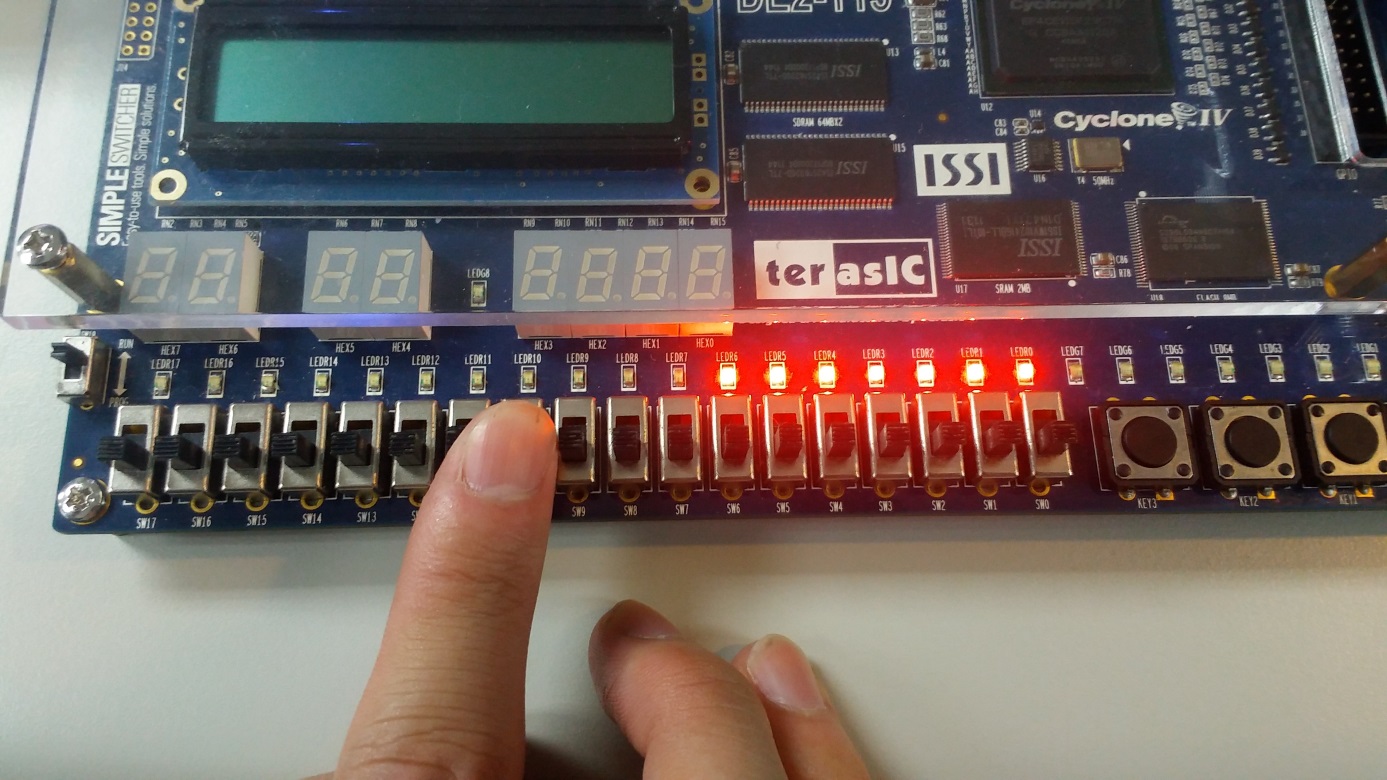
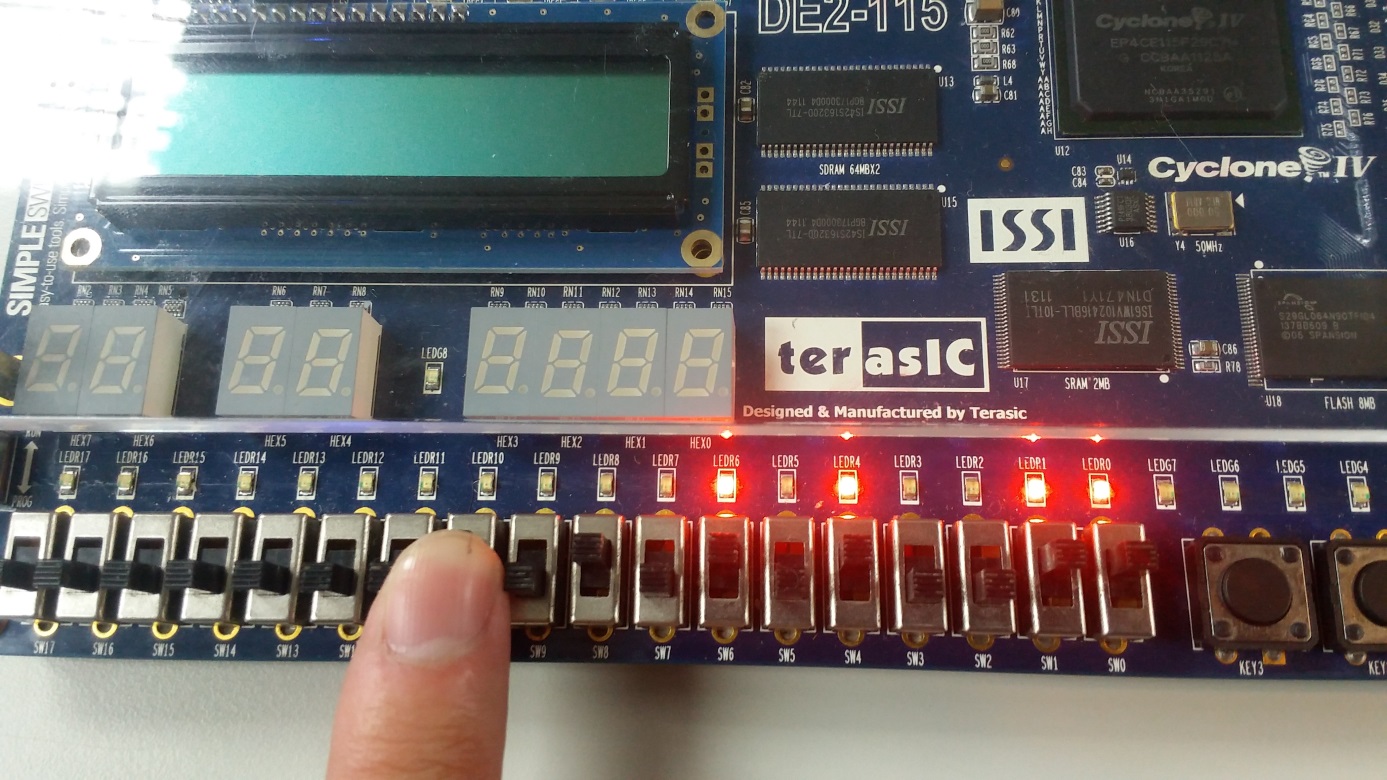
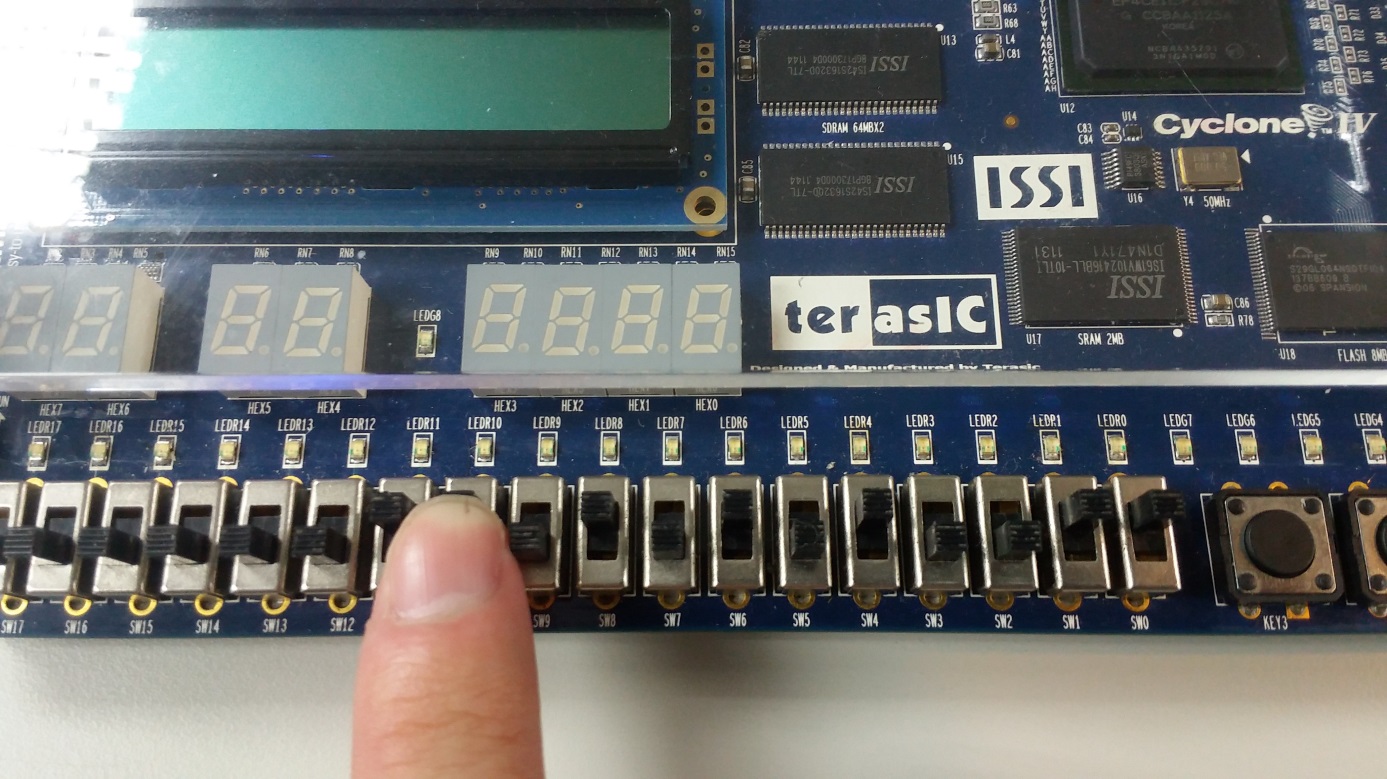
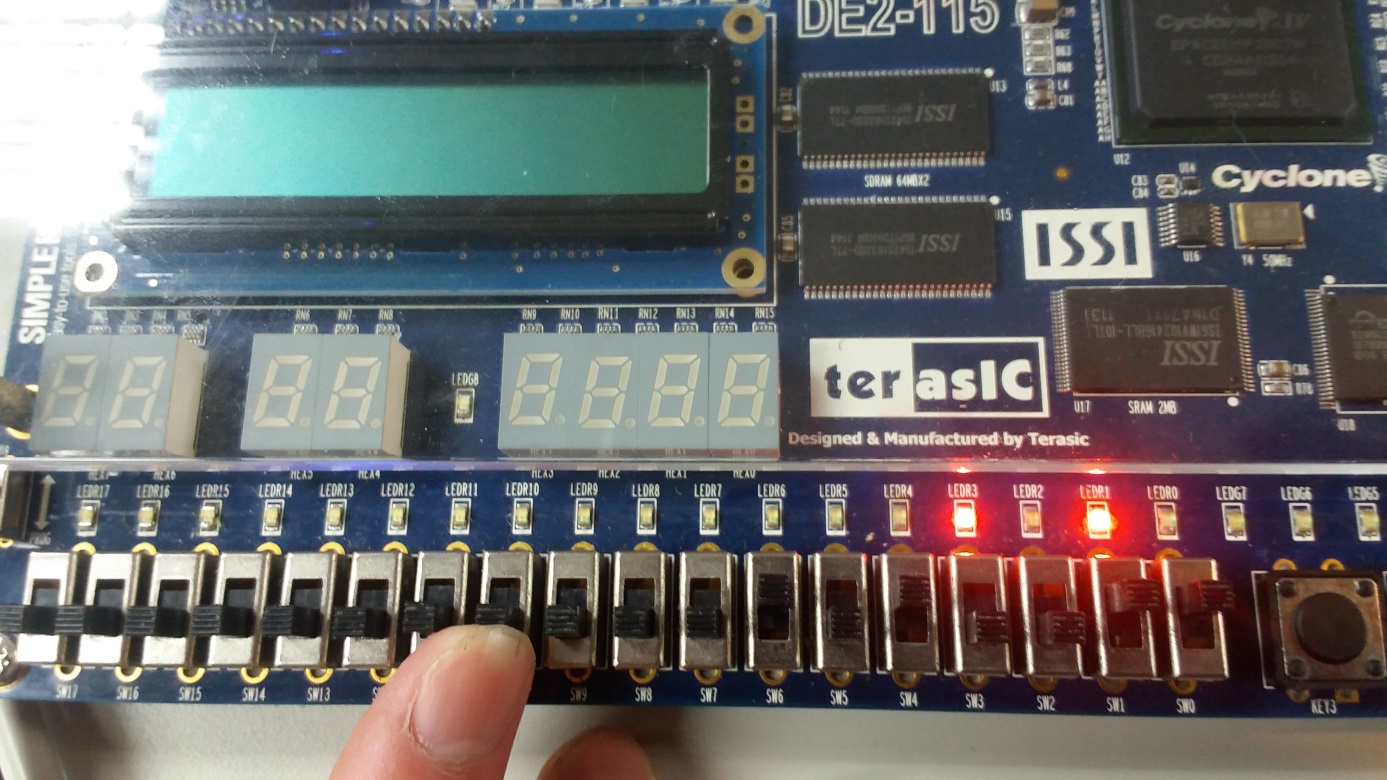
實驗內容分成兩個部分，一個是具有左右移功能的萬用暫存器，二是一個跑馬燈效果的強生計數器，其中兩者皆以switch輸入1作為clock的正緣觸發。

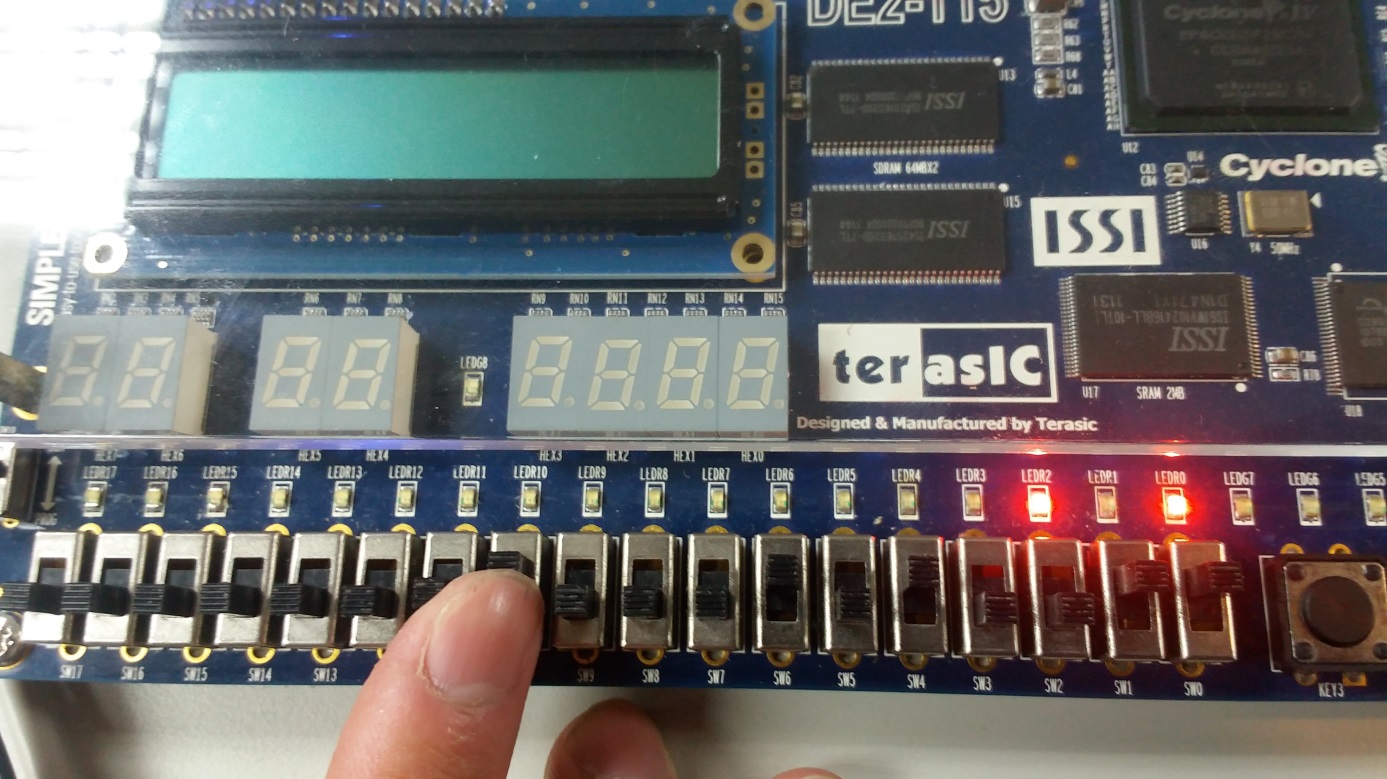
1. 實驗過程及結果：

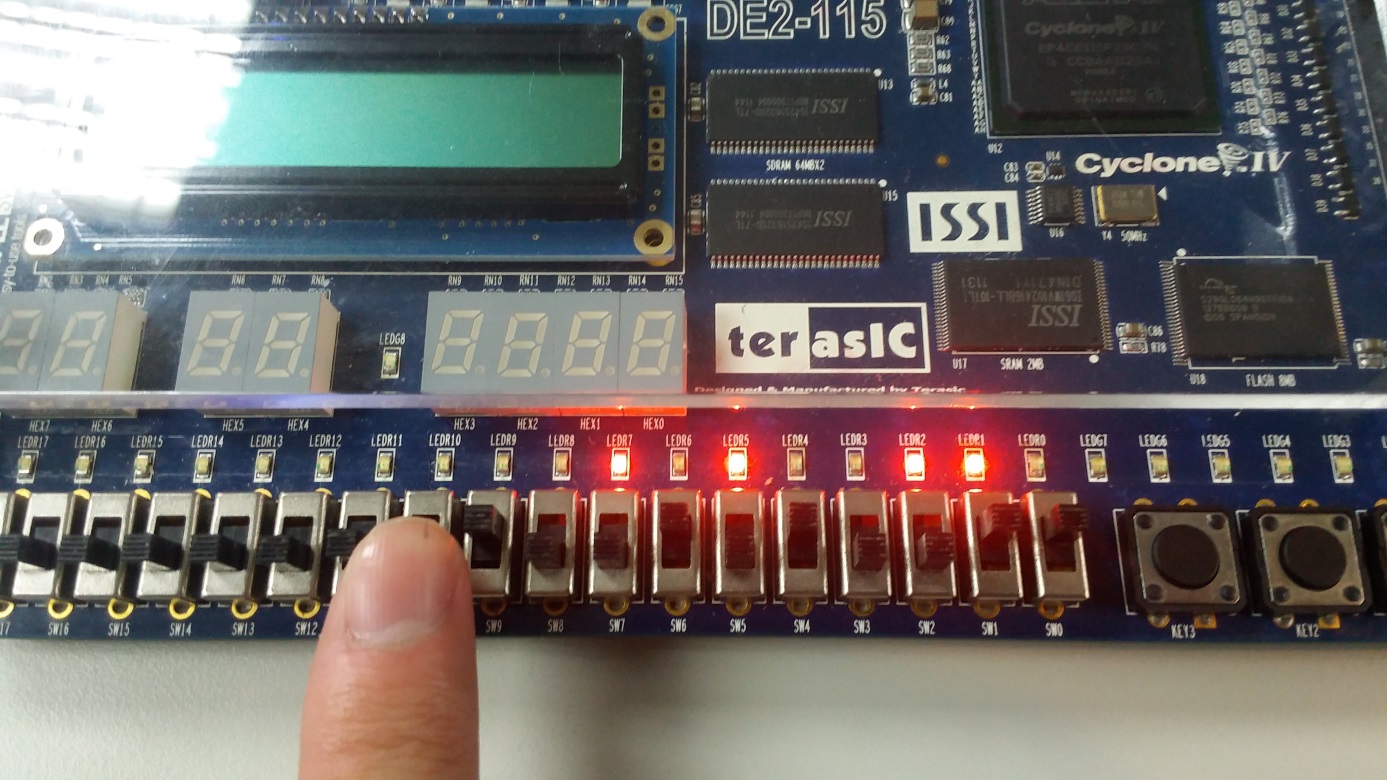
實驗記錄：

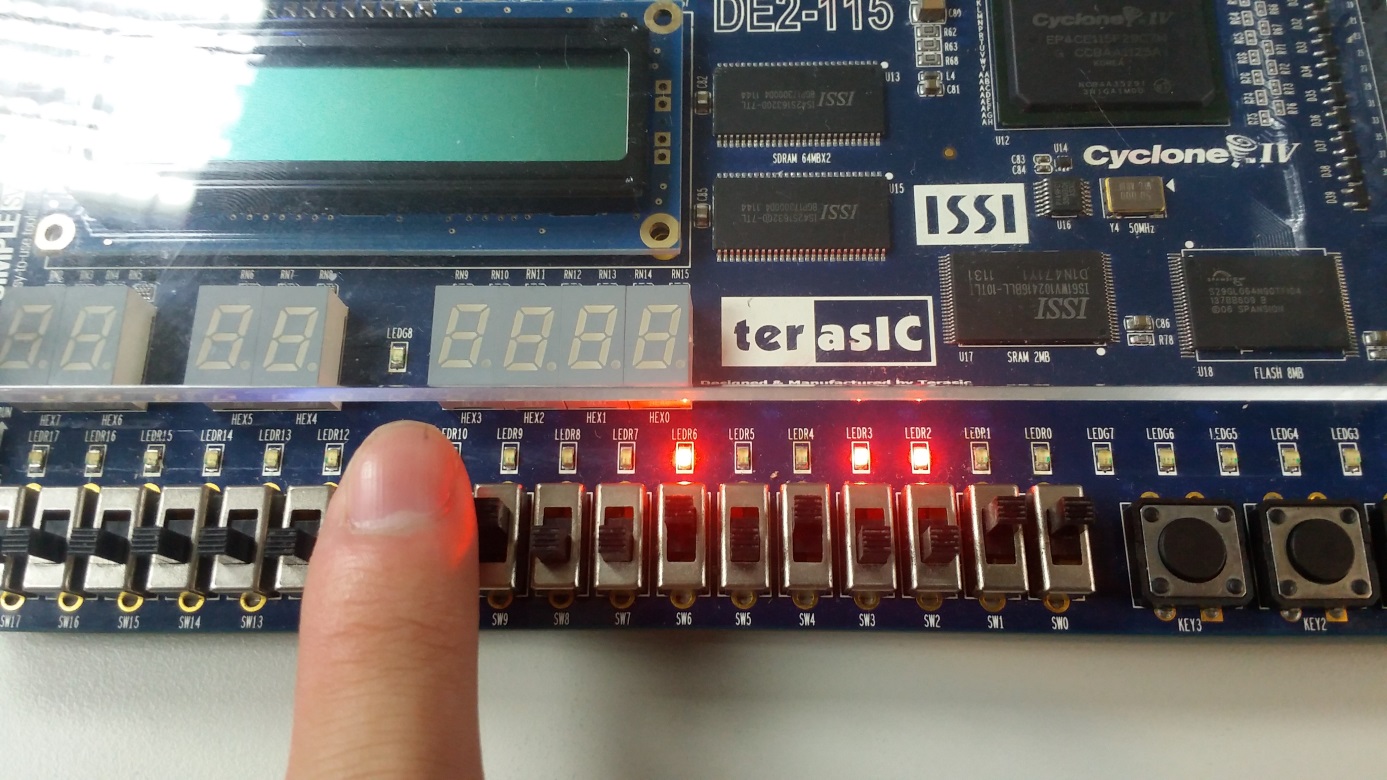
* 1. 強生計數器(Johnson Counter):
     1. 假設狀態開始從111111102
     2. 以switch模擬clock，觸發兩次正緣後，會發現輸出依序變為111111112及011111112。

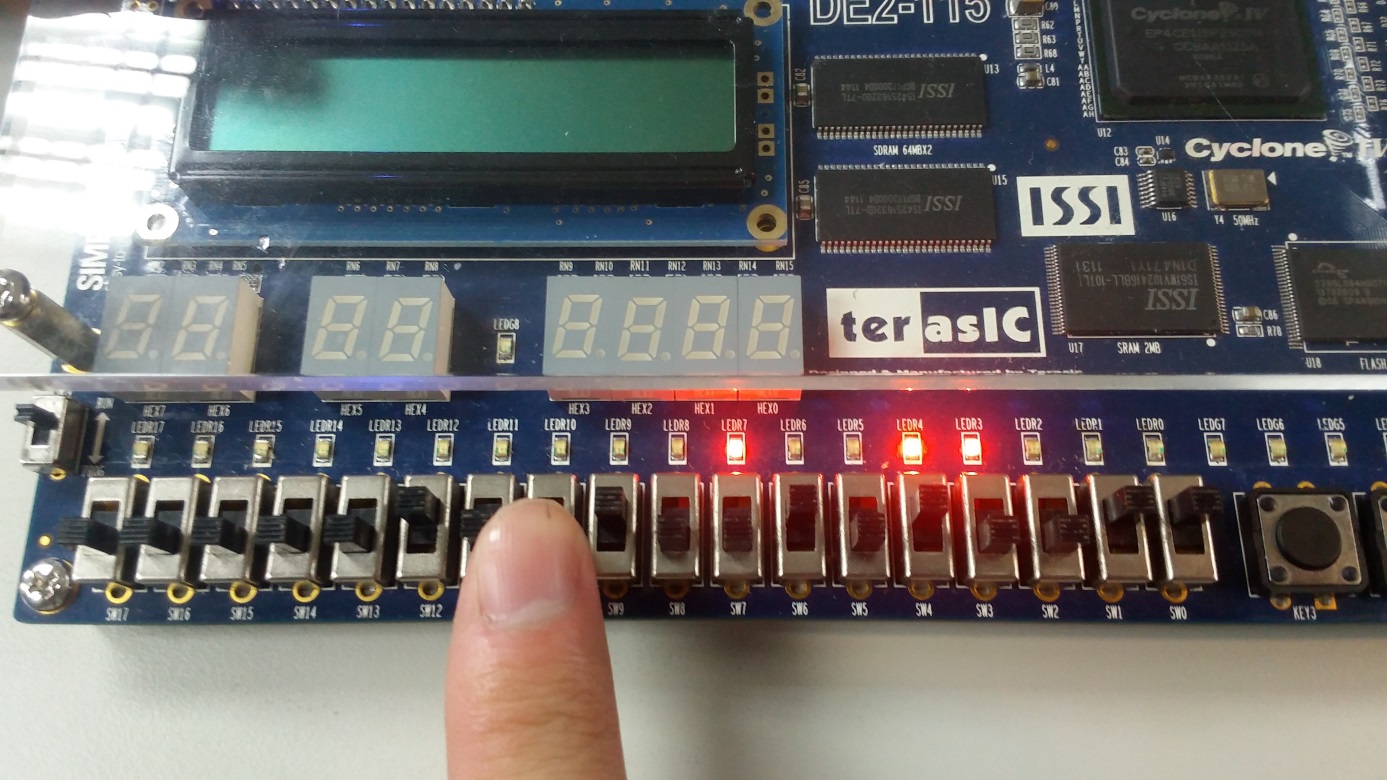


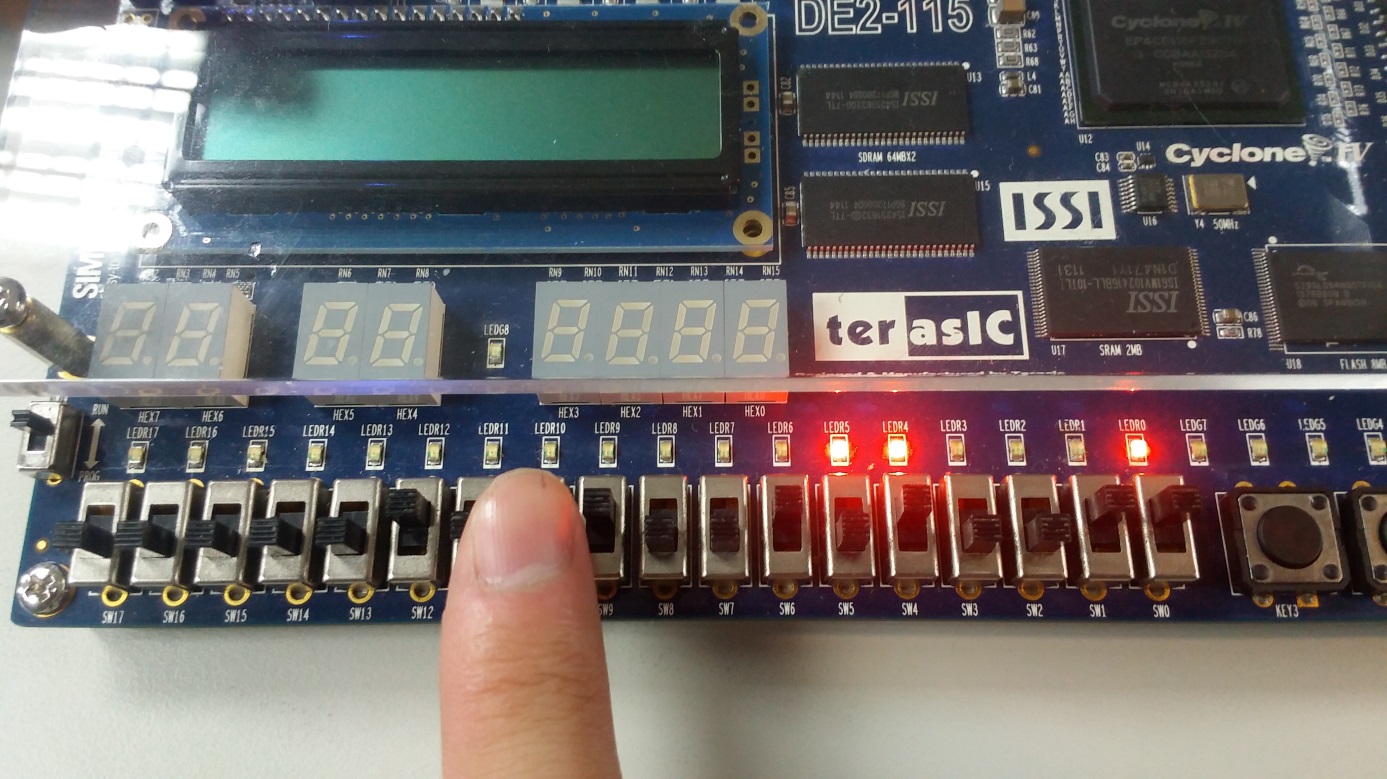
* 1. 8位元左/右位移萬用暫存器
     1. 此為設定load的效果，以switch模擬clock，1作為正緣觸發。
     2. 模擬觸發clear (SW[11])
     3. 模擬觸發右移 (SW[9]=0)

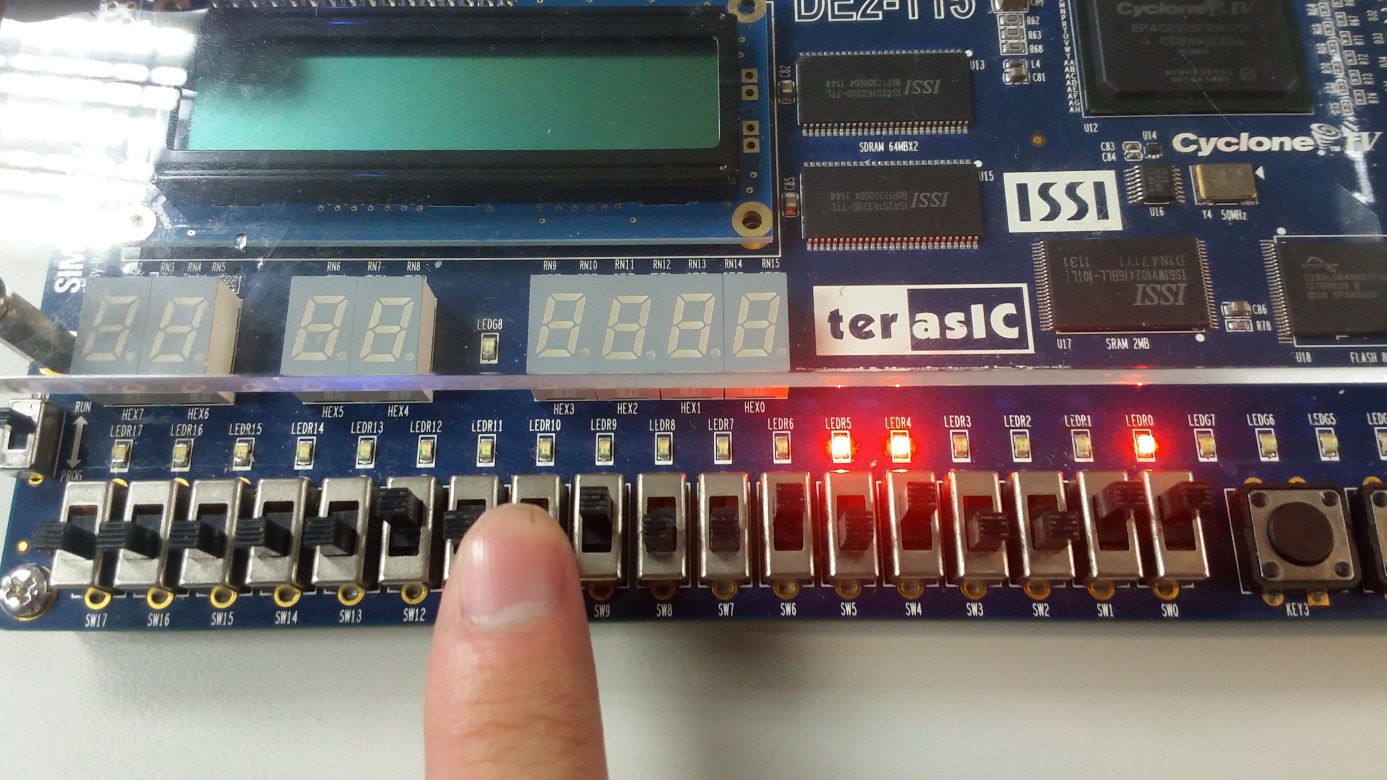


* + 1. 模擬觸發左移 (SW[9]=1)



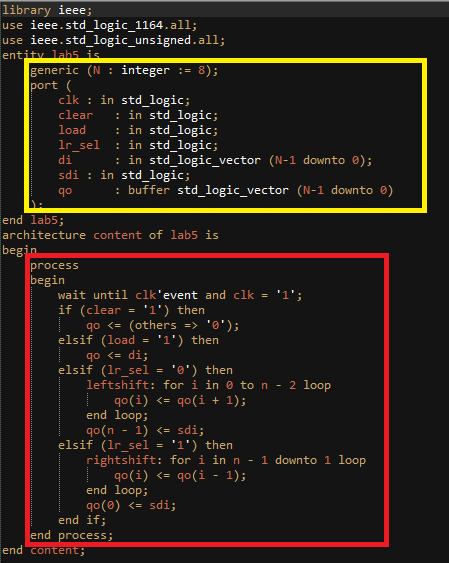
* + 1. 模擬觸發左移 (SW[9]=1)及sdi輸入1





程式碼解釋 :

8位元左/右位移萬用暫存器 :



黃色框 : 宣告外部接線  
紅色框 : 所有行為是等待clk’event and clk = ‘1’ 時才會運行, 也就是說只有當clk 有變動而且是由0 變1 時才會做以下動作。

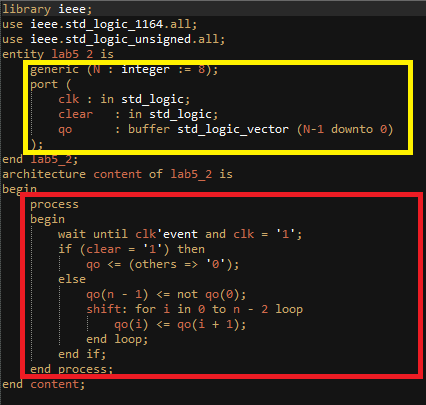
如果clear = ‘1’ 那麼就全部qo 都變成0

Else 如果load = ‘1’, 那麼全部di 平行地載入到qo

Else 如果lrsel = ‘0’ , qo 左移

Else 如果lrsel = ‘1’ , qo 右移

強生計數器 :



黃色框 : 宣告外部接線

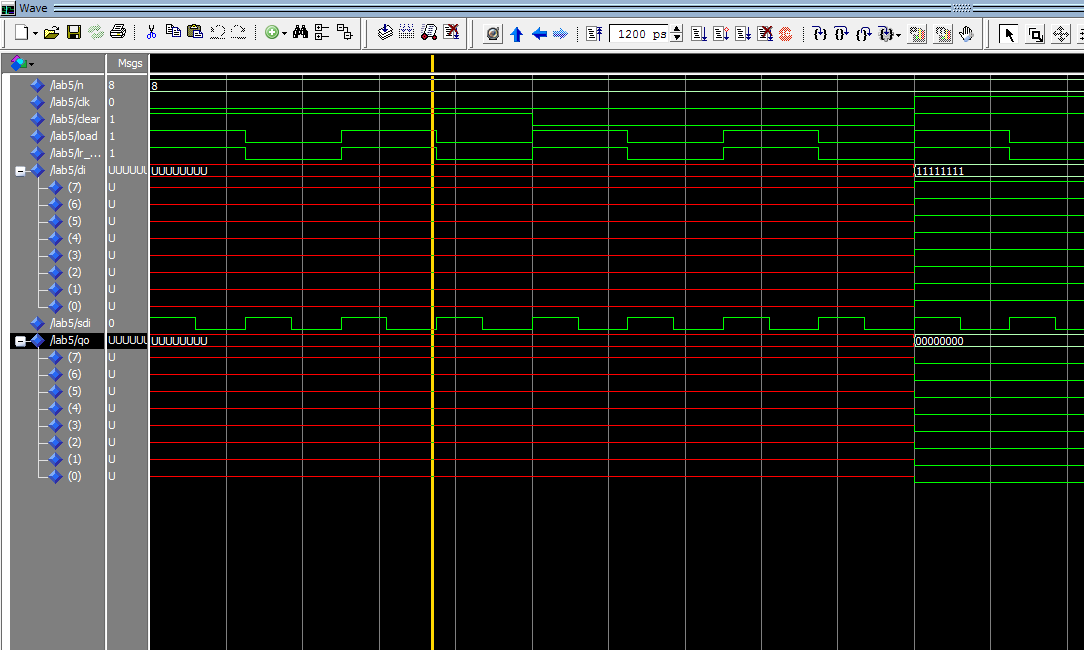
紅色框 : 所有行為會等 clk’event and clk = ‘1’ 時才會進行, 也就是說只有clk 由0 變 1時 才會動作。

如果clear = ‘1’ , 所有qo 都變成0

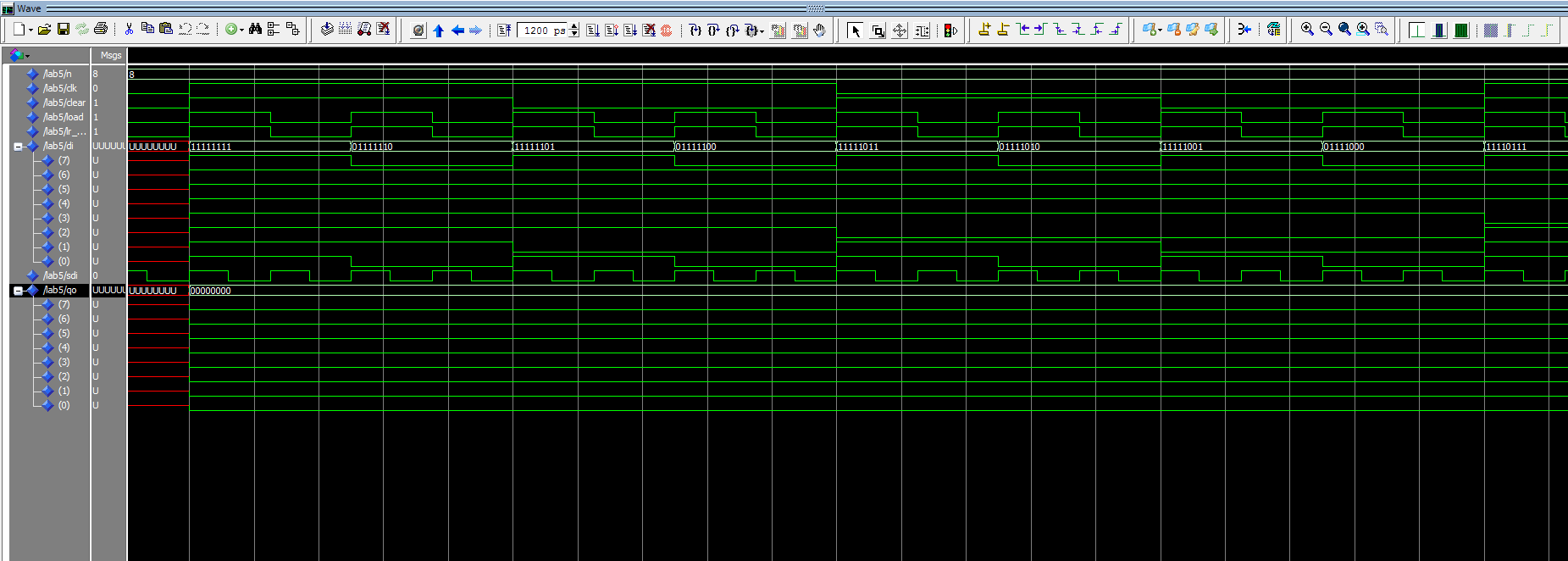
Else 最左邊的位元會接成NOT qo(0), 也就是qo(0) 的反向, 然後所有位元向右移  
由於是在Process 內所以不會出現因為接線全部都接成一樣, 沒有全部同步的問題。

Modelsim 模擬 :

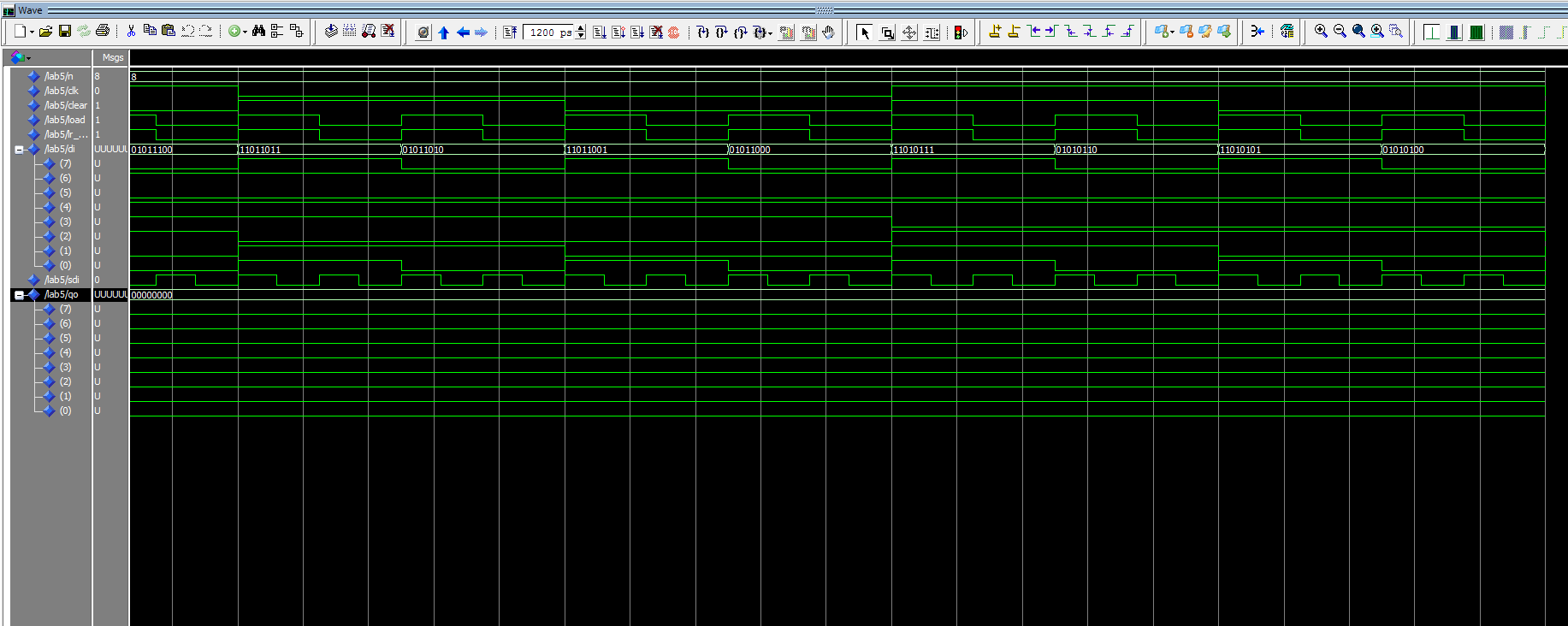
8位元左/右位移萬用暫存器 :



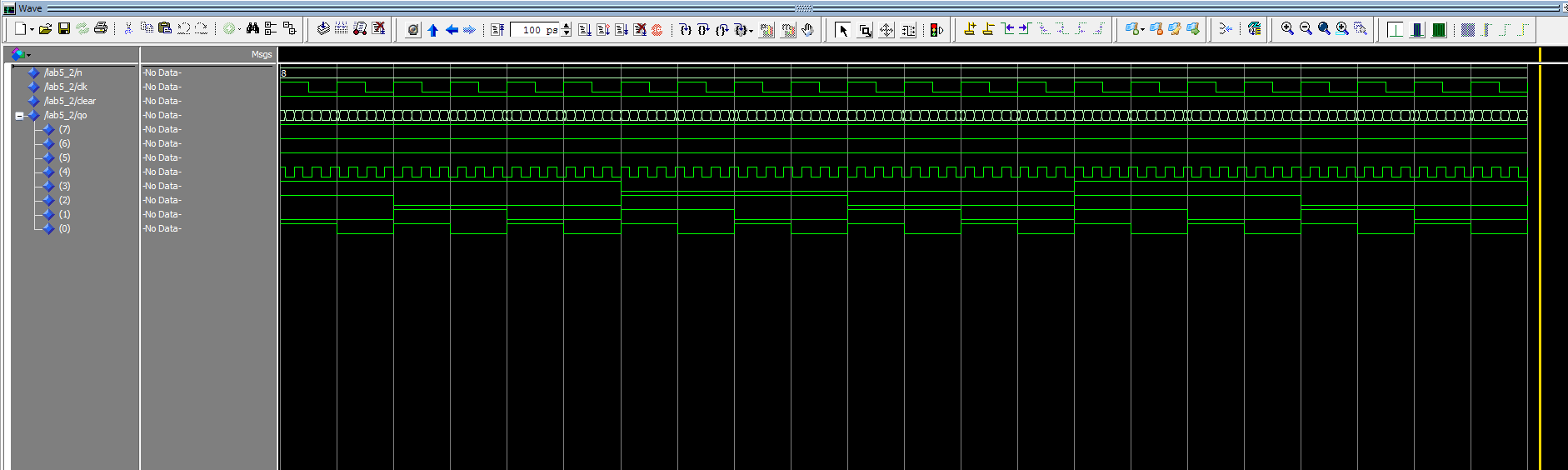
圖中qo 以及di 線為紅色, 是因為di 沒有任何動靜, 因此沒有load 進去也沒有顯示。



在紅色區域後di 開始有動靜, 因此開始不是之前的紅色狀態。



強生計數器 :



強生計數器的signal 比較少

主要是clk 以及vector 的變化。

Clk 設定為50 , 可以看到很明顯每一次變化都是在clk 由0變1時

而q(n – 1) , 即q(7) 的確每一次都是上一次clk’event 時q(0) 的反向

正確無誤

1. 實驗心得：

梁皓鈞(104360098)：

本次實驗難度不高，基本上只要有基本的flipflop 概念就不難做出來。而且flipflop 是需要到數位邏輯的概念，然而VHDL不是。VHDL是希望可以使用程式的邏輯去寫出數位的邏輯。因此要表達出clock 由0變1 時所有東西啟動這個概念，只要用 clk’event and clk = ‘1’ 這一句就可以解決。

而基本上比較需要理解的是同步跟非同步的問題。例如左右移時，是左邊拿去右邊的概念，因此如果是在同步，當一接線時就會出現全部變成一樣了。因此一定要用PROCESS 進行一步一步的接線存起來才不會有問題。

所以只要明白不能同步，需要一步一步接起來才行這個VHDL的概念，基本上整個實驗不難。只要花點時間，或者在紙上草稿思考一下就可以成功了。

這一次實驗也喚起了我數位邏輯的記憶，本身我對flipflop 的概念就不太熟悉，這一次實驗之後更加熟悉了。

對於自己VHDL的程式能力愈來愈好，感到非常高興。

洪晟毅(104590048)：

本次的實驗依照範例進行撰寫，其實難度不高，我認為最大的困難度在於一開始對題目、暫存器實作的概念理解，但只要順利撐過，其實後面的過程基本上是一帆風順。

最初我有些無法理解教授所提供的VHDL實作暫存器的方式，沒有額外儲存的電路，用簡單的IF進行輸入輸出的刷新，對於「當IF不成立就不刷新輸出，會保留上一次輸入的結果」這個部分我實在無法理解，因此我趁著實驗一開始的空餘時間，去找教授進行提問，根據教授說法及我理解的概念來看，應該是跟實驗版實作的原理有關係，在實驗版實作的部份上，似乎是相當於每個SIGNAL都有配置一個暫存器，所以如果當電路觸發VHDL撰寫的IF部分時，不成立就不會有輸出，因此不會有新的來源，輸出就會取得暫存器所記錄上次的輸入，才不會產生輸出不明的情況。

1. 組員貢獻度及工作內容：

|  |  |  |  |
| --- | --- | --- | --- |
| **名字** | **負責項目內容** | **貢獻比例** | **貢獻總和** |
| 皓鈞 | 負責主要程式碼撰寫 | 30% | 50% |
| 報告實驗內容的程式碼解釋 | 10% |
| 報告實驗心得撰寫 | 10% |
| 晟毅 | 負責實驗結果驗證 | 10% | 50% |
| 報告實驗內容、實驗過程撰寫 | 30% |
| 報告實驗心得撰寫 | 10% |
| 總計 | | 100% | 100% |