

電機一乙 數位邏輯實習

第九週實習作業報告

組員：02 葉峻呈 15 邱宇柔

I.實驗目的與原理

A.作業 1

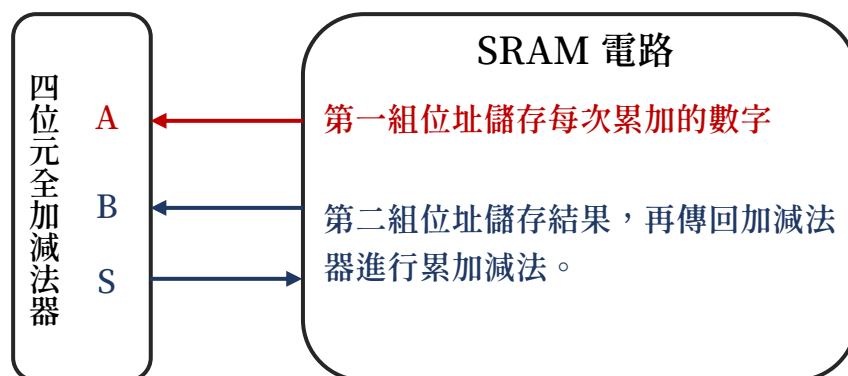
設計一四位元雙向移位暫存器，每個位元均由一個正反器儲存，設計方式不限，我們選擇拉圖來作為本次的設計方式。其狀態表如下。

Inputs			Next State			
LSh (Left)	RSh (Right)	Ld (Load)	Q_3^+	Q_2^+	Q_1^+	Q_0^+
0	0	0	Q_3	Q_2	Q_1	Q_0
0	0	1	D_3	D_2	D_1	D_0
0	1	X	SI	Q_3	Q_2	Q_1
1	0	X	Q_2	Q_1	Q_0	SI
1	1	X	Q_3	Q_2	Q_1	Q_0

圖一：四位元雙向移位暫存器狀態表

B.作業 2

設計一靜態隨機存取記憶體 SRAM 並連接先前設計的加減法器，做成累加減法器。其中 SRAM 電路必須可以儲存四組數字，每組 4 位元，並且只要保持通電，裡面儲存的資訊就可以恆常保持，反之，當電力供應停止時，其內儲存的資料會消失，這與在斷電後還能儲存資料的 ROM 或快閃記憶體是不同的。電路示意如圖二。



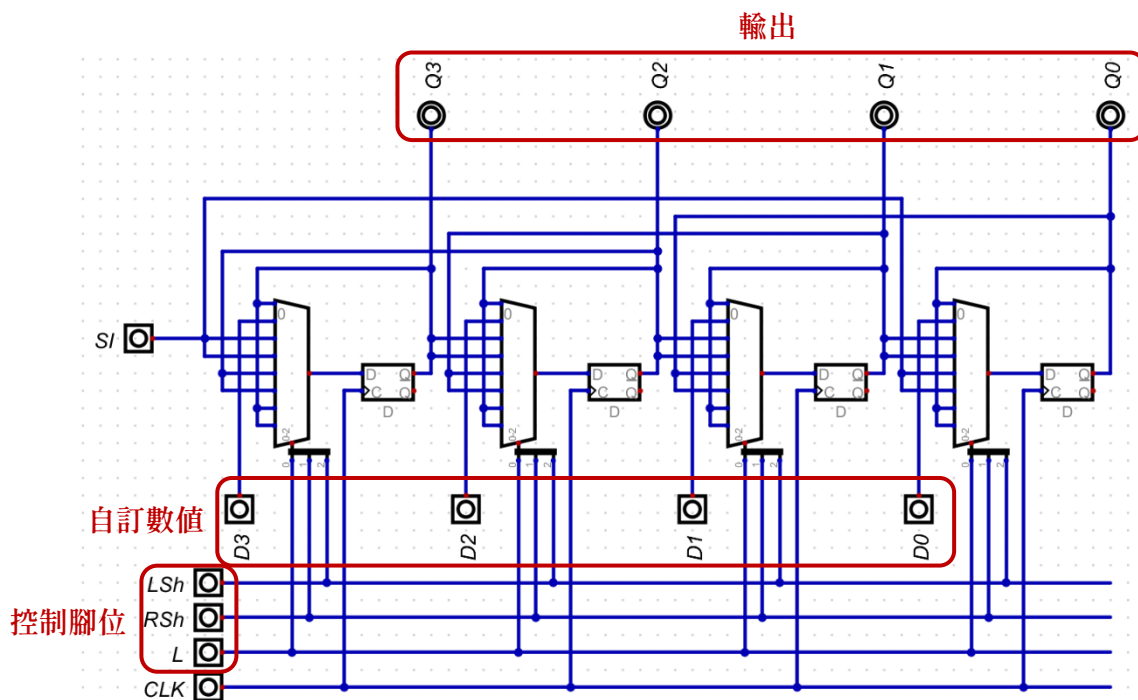
圖二：累加減法器電路示意圖

II. 實驗過程

A. 作業 1

參考圖一狀態表，當 LSh 為 1、RSh 為 0 時，資料左移，輸出 Q 由前級取代；為 01 時，資料右移，輸出 Q 由次級取代；為 00 時，若 Ld 為 1，存入自訂數值 Data，Ld 為 0 則維持原態；為 11 時無功能，維持原態。因在執行左移及右移時，最高位和最低位會出現空缺，所以必須提供 SI 來指定應填補進去的值。

爲了完成上述功能，我們以輸入 LSh、RSh、Ld 作為選擇腳位，透過多工器來控制要存入 D 型正反器的數值，電路如下圖三。



圖三：雙向移位暫存器電路圖

B. 作業 2

在設計 SRAM 的 VHDL 時，我們需要設定一個可以儲存四組 4bit 的記憶體，並分別給予四個不同的位置。接著按照各個腳位給予的訊號，來決定要對特定位置的數值進行輸出還是寫入。其 VHDL 如下圖，D 為輸入數值，共 4bit；Address 為位置選擇，共 2bit；Chip 為電源；WE、OE 則用來控制該將數值存入還是輸出。

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

```
entity my_SRAM is
```

```
    port ( D: in std_logic_vector(3 downto 0);  
          Address: in std_logic_vector(1 downto 0);  
          ChipSelect, WE, OE: in std_logic;  
          DO: out std_logic_vector(3 downto 0));
```

腳位設定

```
end my_SRAM;
```

```
architecture SRAM_simple of my_SRAM is
```

```
    type memory is array (0 to 3) of std_logic_vector(3 downto 0);
```

```
    signal SRAM_internal: memory := ("0000", "0000", "0000", "0000");
```

記憶體

```
    signal Address_int: integer;
```

```
    signal Address_temp: std_logic_vector(1 downto 0);
```

```
begin
```

```
    Address_int <= to_integer(unsigned(Address));
```

位置選擇

```
    DO <= "ZZZZ" when ChipSelect = '0'
```

```
    else "ZZZZ" when OE = '0'
```

```
    else SRAM_internal(Address_int) when OE = '1';
```

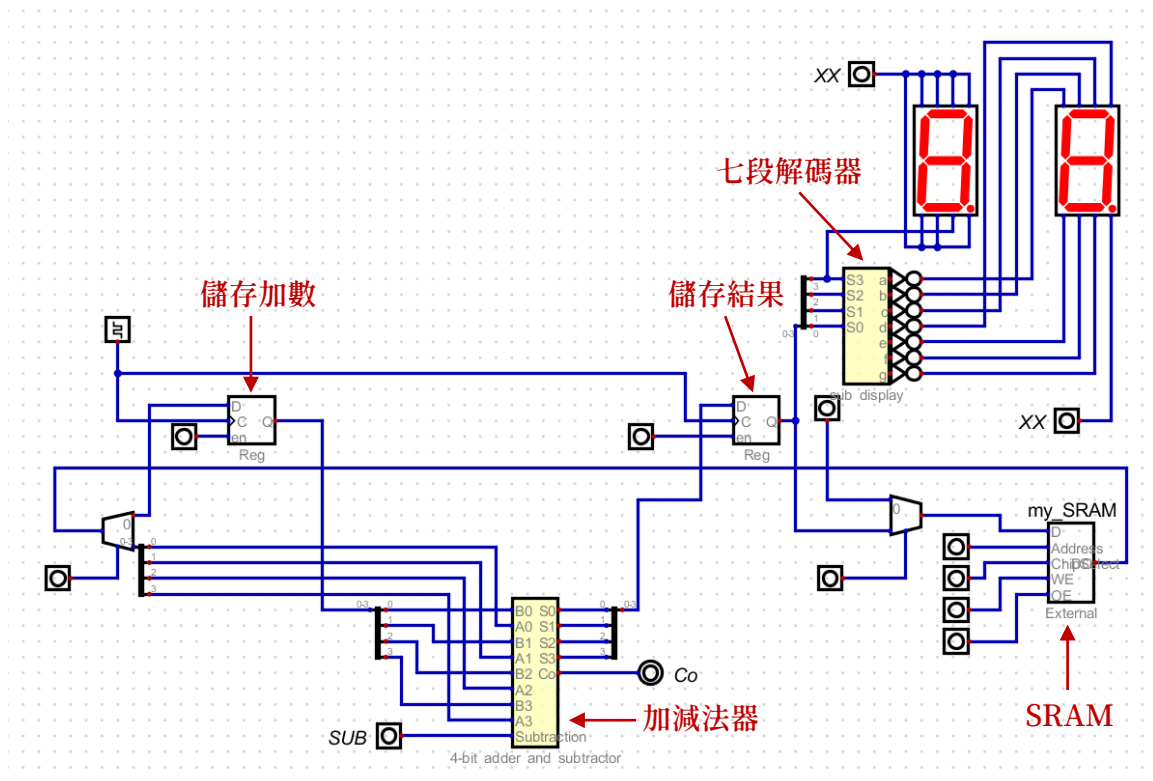
輸出

```
    SRAM_internal(Address_int) <= D when WE = '1';
```

寫入

```
end SRAM_simple;
```

完成 SRAM 後，參考投影片，配合暫存器、多工器、解多工器及加減法等實作出累加減法器之電路，並將輸出接至七段顯示器，方便觀察。



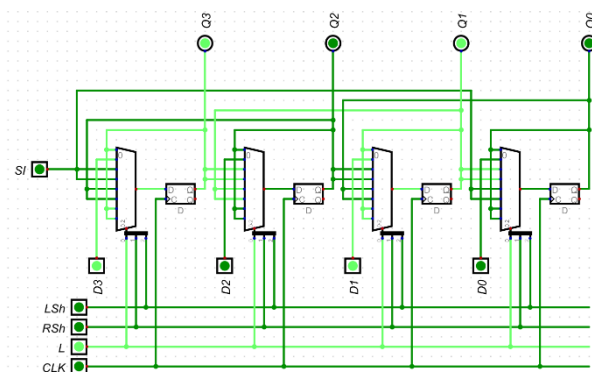
圖四：累加減法器電路圖

III. 模擬驗證

A. 作業 1

利用 Digital 進行模擬，驗證電路執行結果與預期相符。測試結果如下。

1. 寫入：



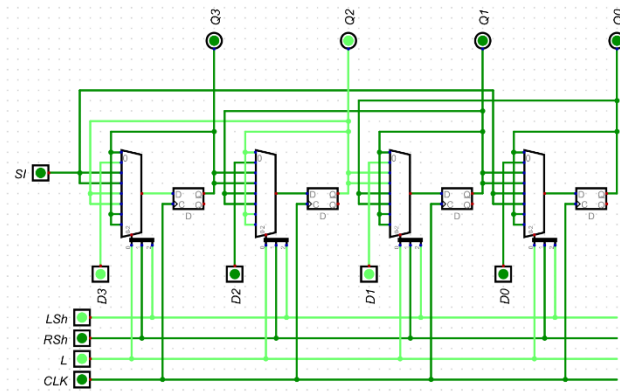
圖五：模擬 1-a

當 L 為 1，由 CLK 觸發後

D = 1010 存入正反器。此時輸出

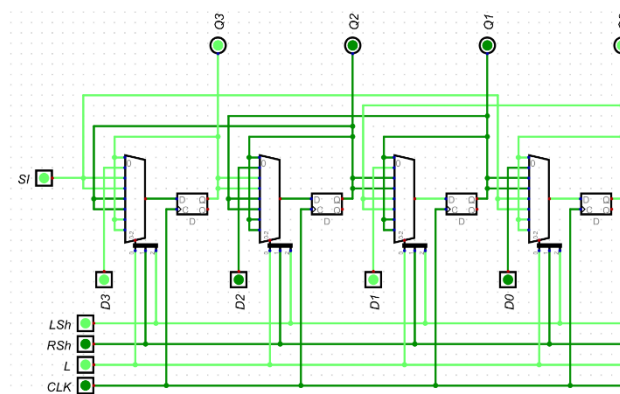
Q 為 1010。

2. 左移



圖六：模擬 1-b

當 LSh 為 1，CLK 觸發後左移 1 位元，Q0 由 SI = 0 遞補，此時輸出 Q 為 0100。

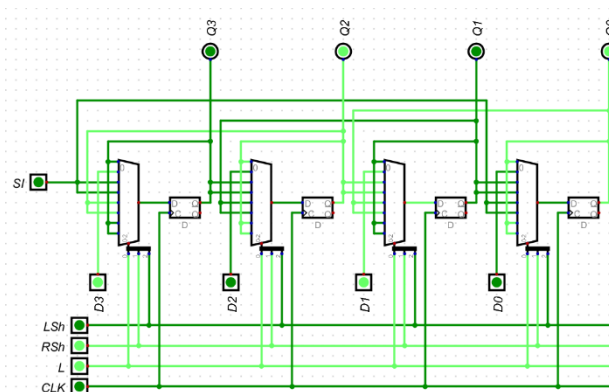


圖七：模擬 1-c

LSh 維持為 1，SI 改為 1，由 CLK 觸發後再次左移，此時輸出 Q 為 1001。

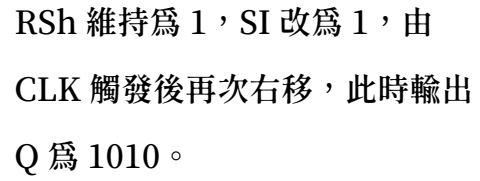
2. 右移:

同左移，先將 1010 存入(如圖五)後再進行後續操作。



圖八：模擬 1-d

當 RSh 為 1，CLK 觸發後右移 1 位元，Q3 由 SI = 0 遞補，此時輸出 Q 為 0101。



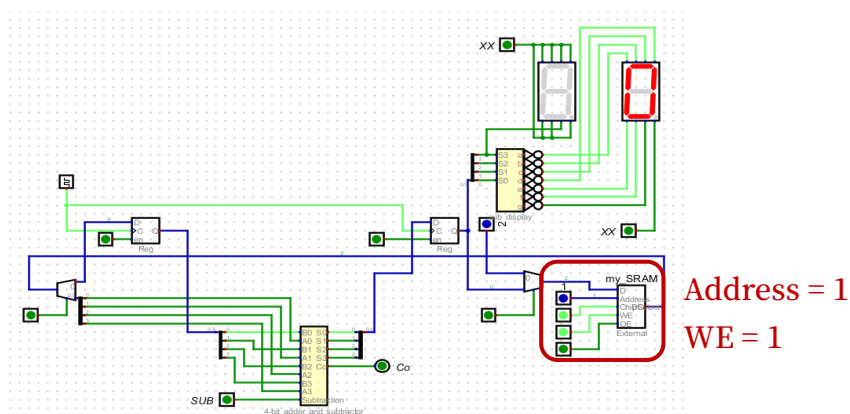
由以上操作我們可以知道，LSh 及 RSh 的優先權大於 L，因此當 LSh、RSh 其中一方為 1 時，無論 L 腳位的輸入為何都不影響左右移的動作。

利用 Digital 測試，驗證電路無誤。

-

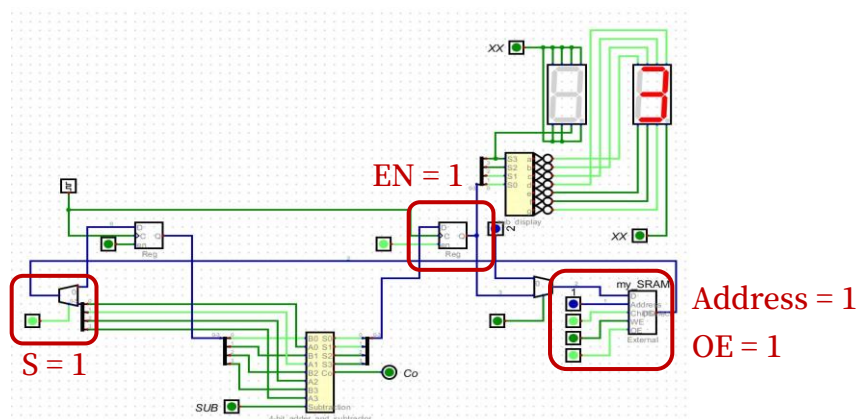
圖十一：模擬 2-b

3. 將被加數 2 存入 SRAM 的位置 1。



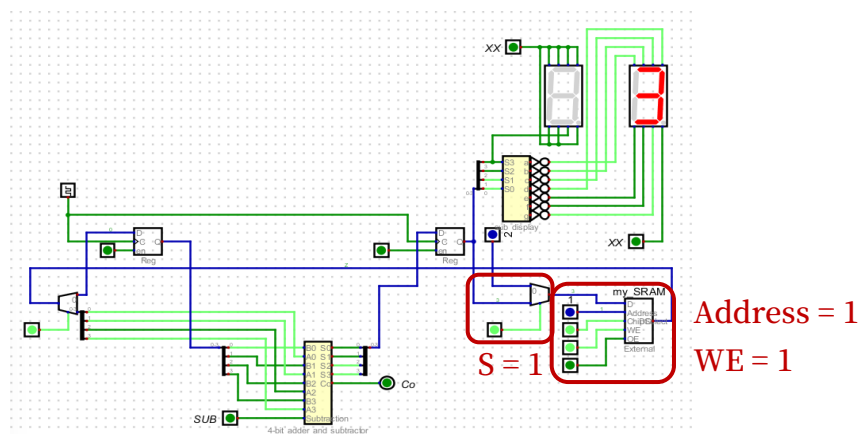
圖十二：模擬 2-c

4. 輸出 SRAM 的位置 1，再按下解多工器的選擇腳位使其為 1，讓加數及被加數可以在經過加減法器後存入第二顆暫存器，七段顯示器變為 $2+1=3$



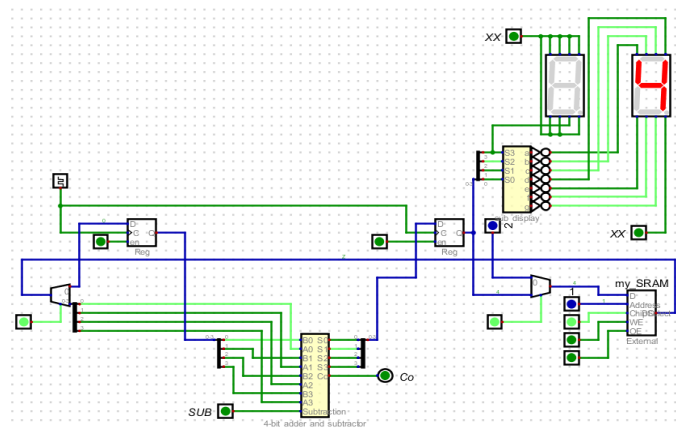
圖十三：模擬 2-d

5. 使多工器的選擇腳位為 1，將結果回傳至 SRAM，並存入任意位置，在此沿用上次存取位置，減少腳位變更次數。



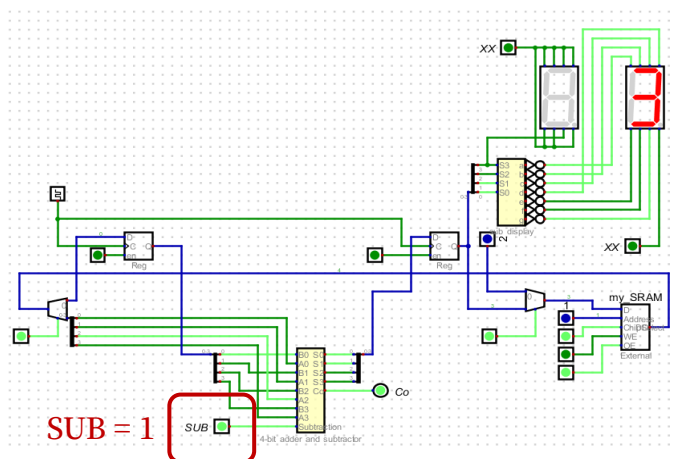
圖十四：模擬 2-e

6. 重複步驟 4、5，來進行不斷加 1 的動作。



圖十五：模擬 2-f

7. 打開加減法器的 SUB 腳位，讓電路變為減法，重複步驟 4、5。



圖十六：模擬 2-g

IV. 實驗結果與成果討論

A. 作業 1

將電路燒錄製電路板後進行測試，以指撥開關為輸入 CLK、L、LSh、RSh、D、SI，LED 為輸出 Q。

1. 寫入：

L=1，寫入模式，撥動 CLK 後，Q =
D = 0101。



圖：實際操作 1-a

2. 左移：

LSh = 1、SI = 0，左移，撥動 CLK 後，Q = 1010。



圖：實際操作 1-b

LSh = 1、SI = 0，再次左移，撥動 CLK 後，Q = 0100。



圖：實際操作 1-c

3. 右移

延續上一個狀態 Q = 0100，RSh = 1、SI = 1，換為右移，並由 1 遞補，撥動 CLK 後，Q = 1010。



圖：實際操作 1-d

RSh = 1、SI = 1，再次右移，撥動 CLK 後，Q = 1101。



圖：實際操作 1-e

B.作業 2

由於電路板上開關的彈跳問題，電路燒錄後一直無法正常運作，故未能提供實際測試的照片。

V.實驗心得

我覺得本次的實驗第二題如果用拉線是比較繁雜的動作內容，要很清楚每一步動作到底在做了什麼，存入了什麼數值，但是瞭解以後就會覺得很踏實。