電機一乙 數位邏輯實習

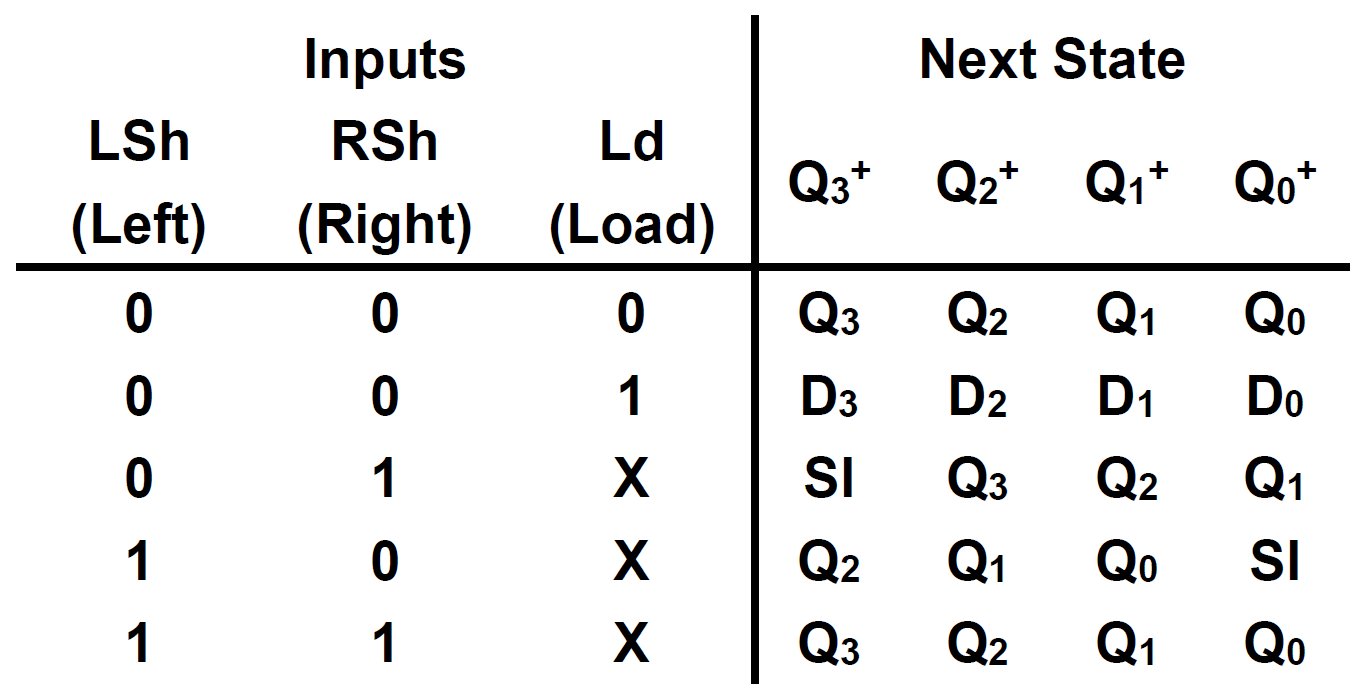
第九週實習作業報告

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。



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。

四位元全加減法器

第一組位址儲存每次累加的數字

第二組位址儲存結果，再傳回加減法器進行累加減法。

SRAM電路

A

B

S

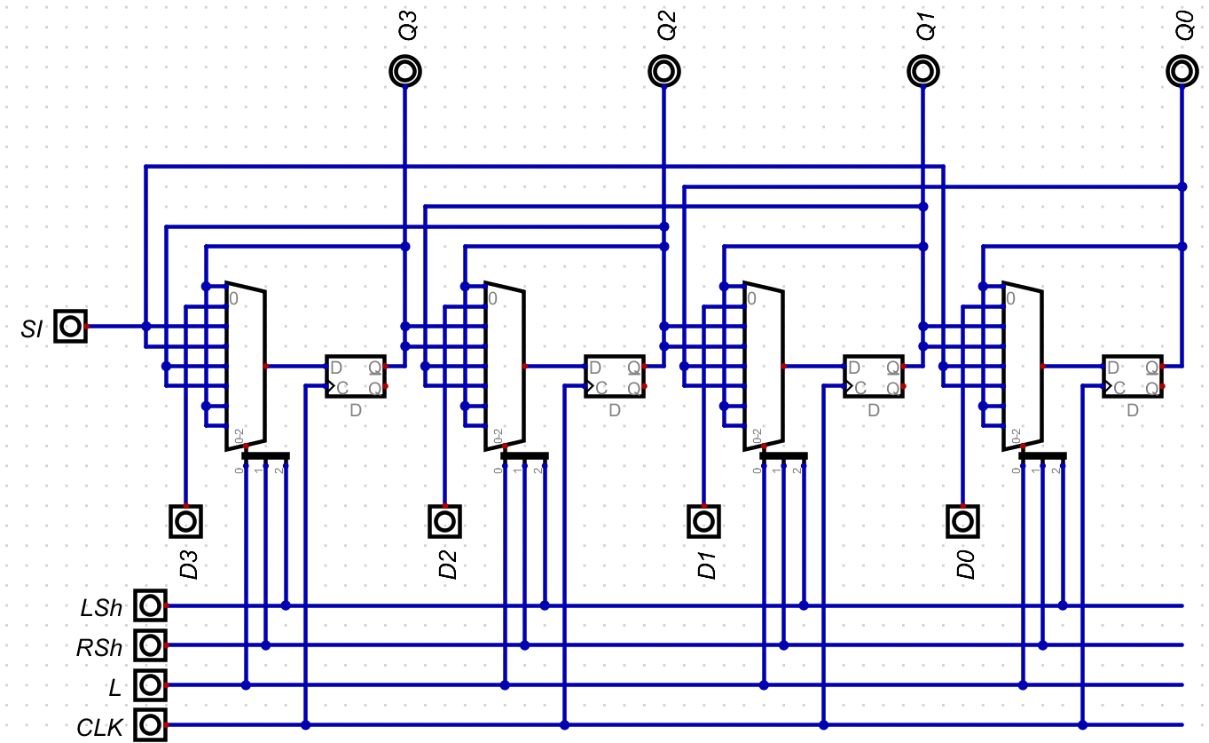
圖二：累加減法器電路示意圖

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。

為了完成上述功能，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。



自訂數值

控制腳位

輸出

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));

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");

signal Address\_int: integer;

signal Address\_temp: std\_logic\_vector(1 downto 0);

begin

Address\_int <= to\_integer(unsigned(Address));

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

else "ZZZZ" when OE = '0'

else SRAM\_internal(Address\_int) when OE = '1';

SRAM\_internal(Address\_int) <= D when WE = '1';

end SRAM\_simple;

腳位設定

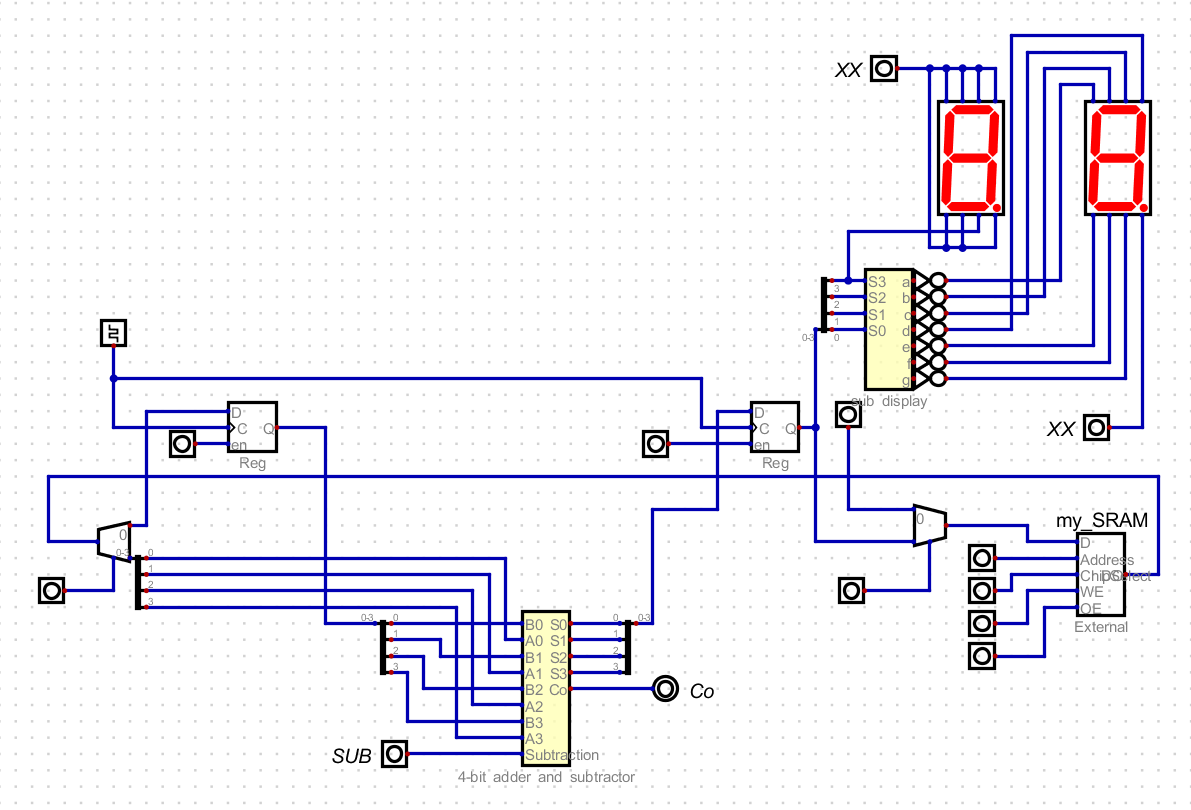
記憶體

位置選擇

輸出

寫入

完成SRAM後，參考投影片，配合暫存器、多工器、解多工器及加減法器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。



七段解碼器

儲存結果

儲存加數

SRAM

加減法器

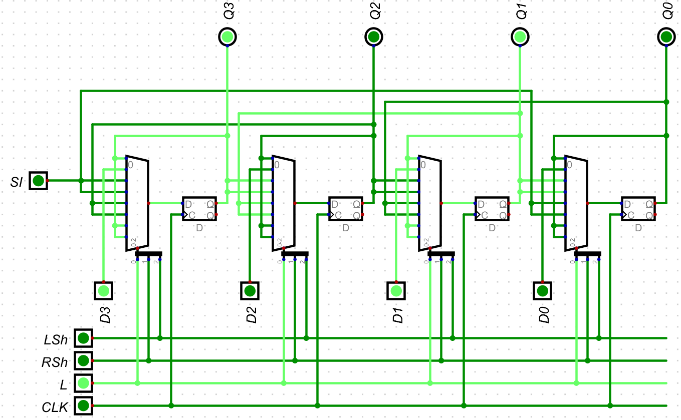
圖四：累加減法器電路圖

III.模擬驗證

1. *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：

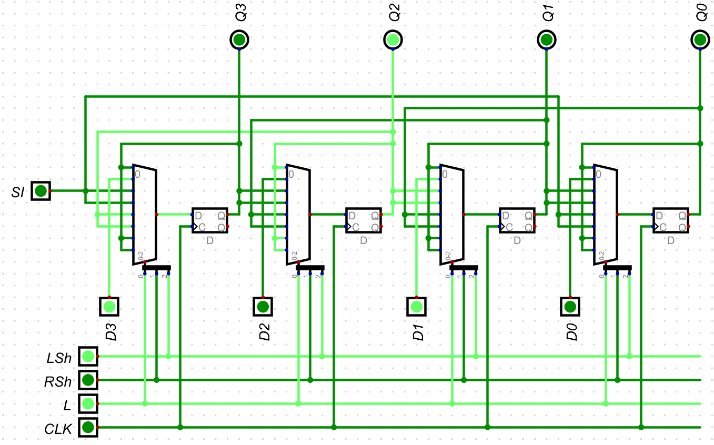


圖五：模擬1-a

當L為1，由CLK觸發後

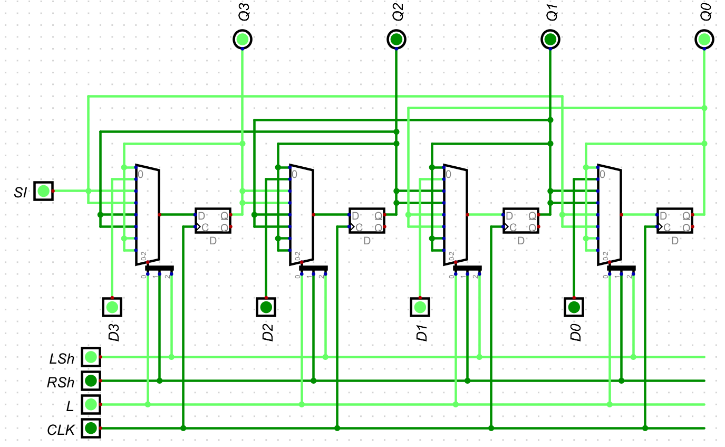
D = 1010存入正反器。此時輸出Q為1010。

1. 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。

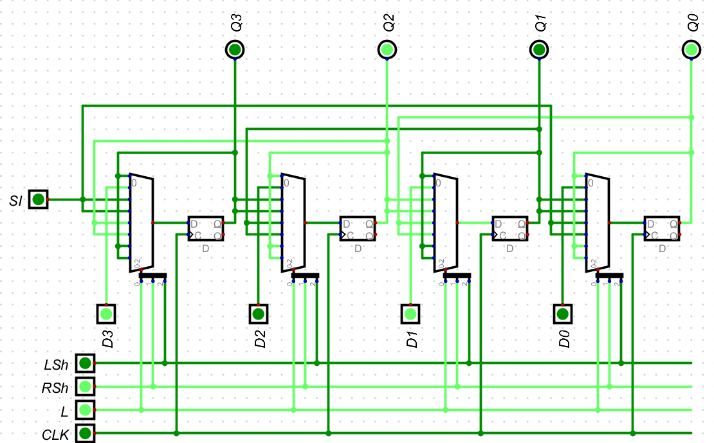


圖七：模擬1-c

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。

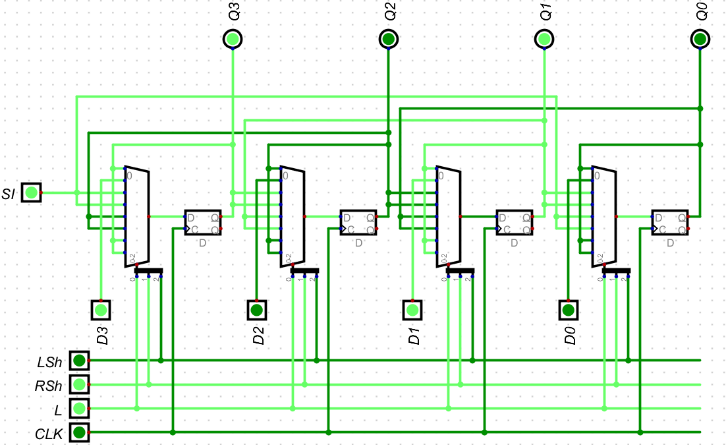
2. 右移:

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。



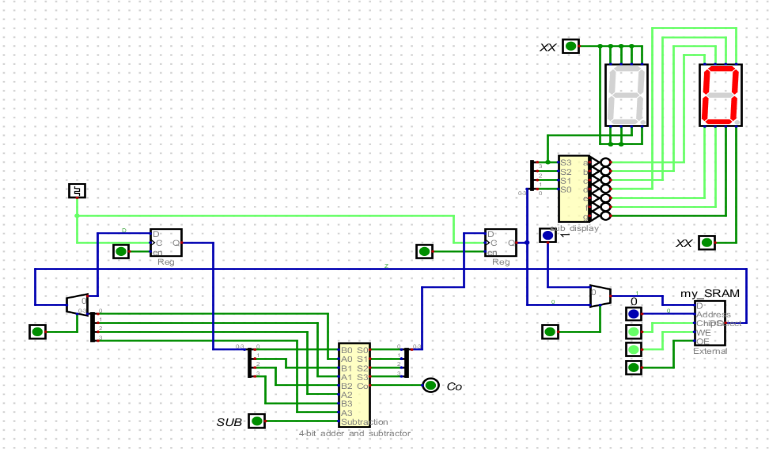
圖九：模擬1-e

RSh維持為1，SI改為1，由CLK觸發後再次右移，此時輸出Q為1010。

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。

*B.作業2*

利用Digtal測試，驗證電路無誤。

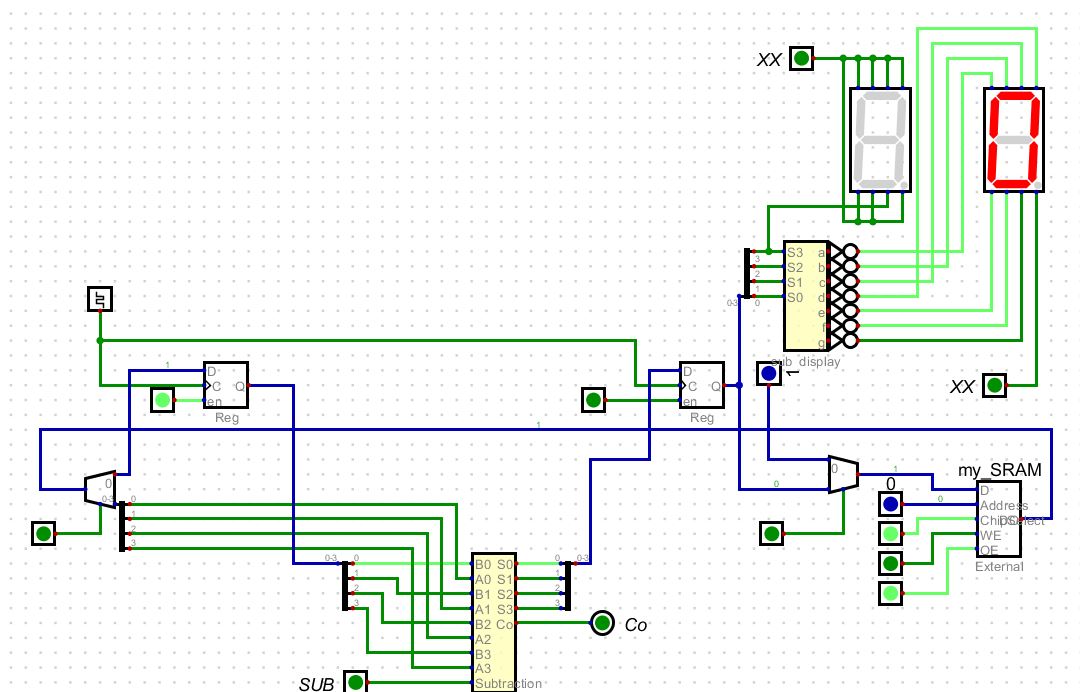
1. 將加數1存入SRAM的位置0。

Address = 0

WE = 1

圖十：模擬2-a

1. SRAM的位置0輸出後，第一顆暫存器通電，存入位置0的加數。



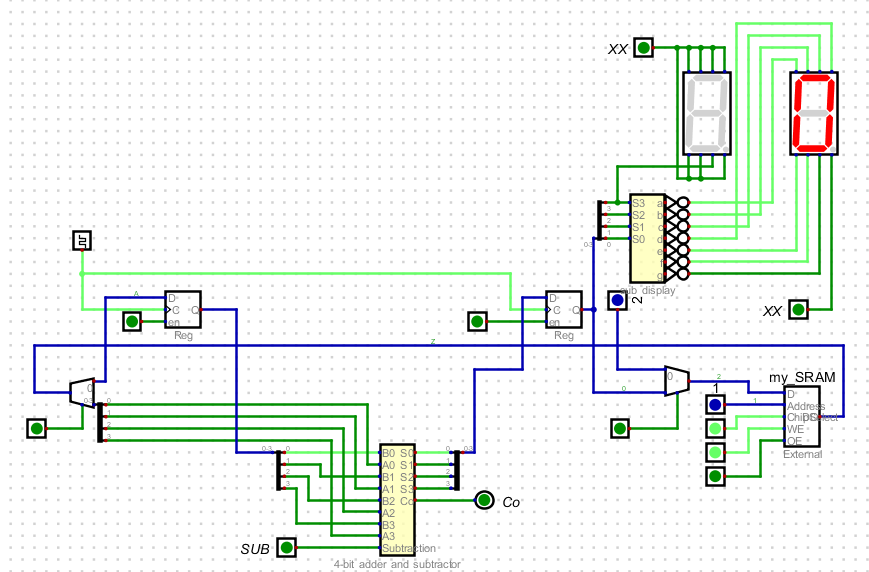
EN = 1

Address = 0

OE = 1

圖十一：模擬2-b

1. 將被加數2存入SRAM的位置1。

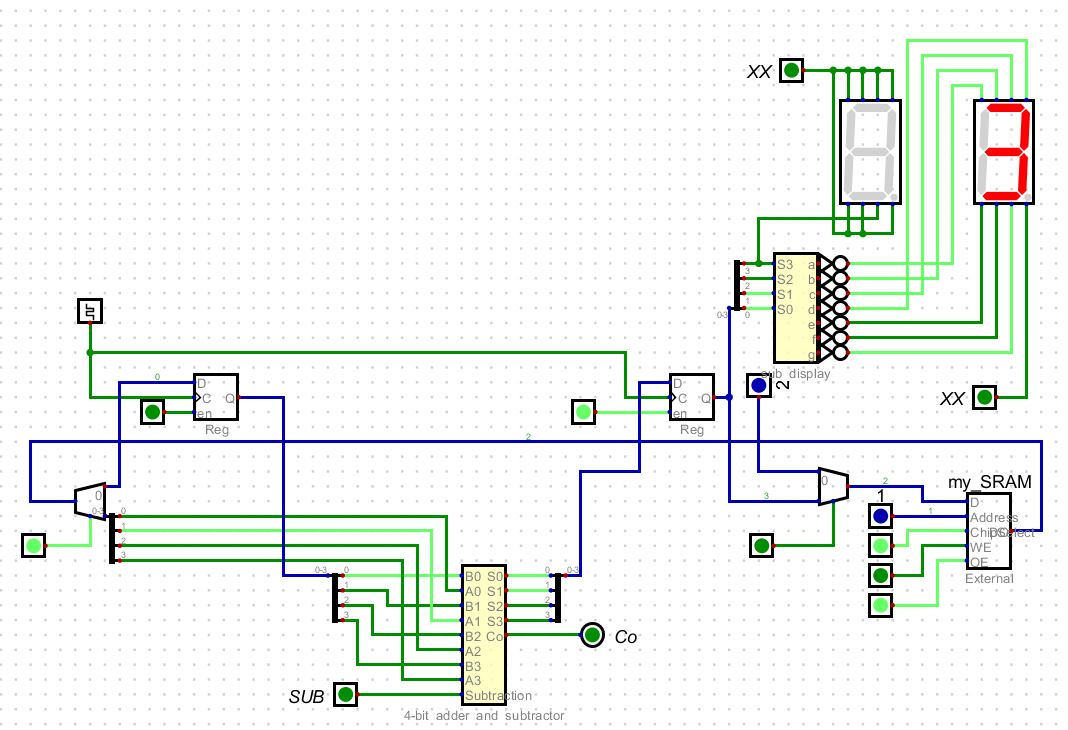


Address = 1

WE = 1

圖十二：模擬2-c

1. 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



EN = 1

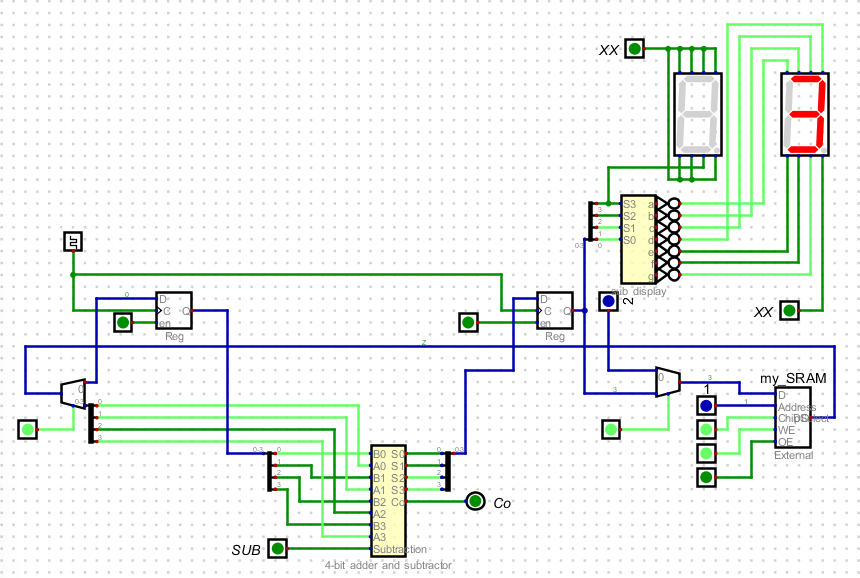
Address = 1

OE = 1

S = 1

圖十三：模擬2-d

1. 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。



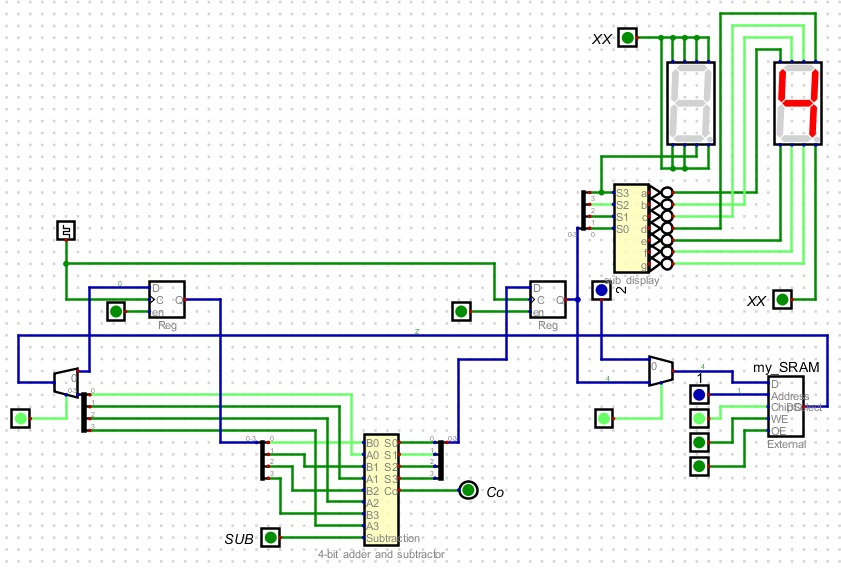
Address = 1

WE = 1

圖十四：模擬2-e

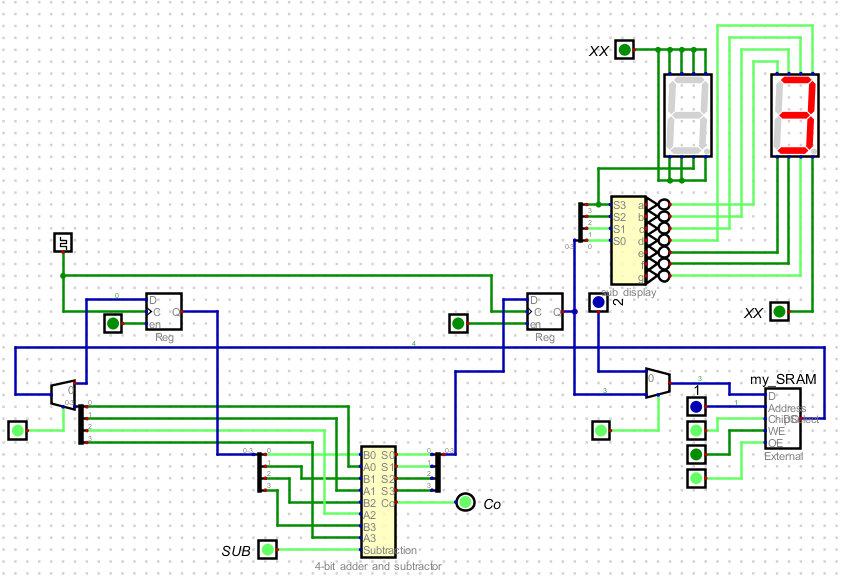
S = 1

1. 重複步驟4、5，來進行不斷加1的動作。



圖十五：模擬2-f

1. 打開加減法器的SUB腳位，讓電路變為減法，重複步驟4、5。



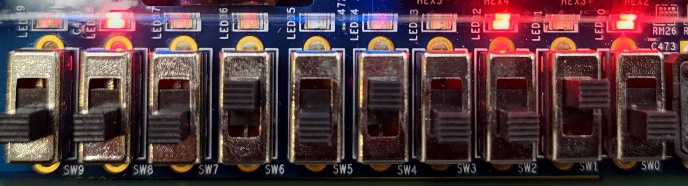
SUB = 1

圖十六：模擬2-g

IV.實驗結果與成果討論

1. *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。



Q = 0101

CLK

L=1

LSh=0

RSh=0

D=0101

SI=0

圖：實際操作1-a

1. 寫入：

L=1，寫入模式，撥動CLK後，Q = D = 0101。

1. 左移：



Q = 1010

CLK

L=1

LSh=1

RSh=0

D=0101

SI=0

圖：實際操作1-b

LSh = 1、SI = 0，左移，撥動CLK後，Q = 1010。



Q = 0100

CLK

L=1

LSh=1

RSh=0

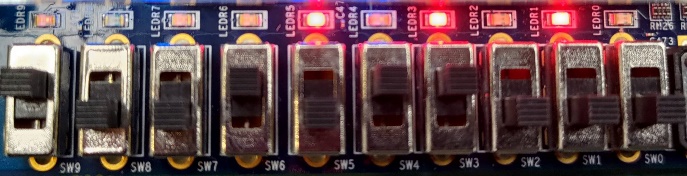
D=0101

SI=0

圖：實際操作1-c

LSh = 1、SI = 0，再次左移，撥動CLK後，Q = 0100。

1. 右移



Q = 1010

CLK

L=0

LSh=0

RSh=1

D=0101

SI=1

圖：實際操作1-d

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。



Q = 1101

CLK

L=0

LSh=0

RSh=1

D=0101

SI=1

圖：實際操作1-e

RSh = 1、SI = 1，再次右移，撥動CLK後，Q = 1101。

*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。