數位邏輯實習

班級：電機一甲
專題名稱：數位存錢筒
組員：鍾恩貴、鄧勝宥

前言

存錢筒，又名錢筒、撲滿、儲蓄罐、存錢罐或悶葫蘆罐，是儲存硬幣的容器。從前多以竹筒或陶瓷製造，現代亦使用塑膠、人造橡膠或金屬等其他材質。傳統的撲滿除了竹筒採用原型外，其餘多製作成豬的造形，其大肚子象徵著財富的積累與富足，近年來撲滿的造形則比較多樣化。

舊時的撲滿除了頂端有一個用於投入硬幣的狹長小縫隙外，別無其他出入口，如果真的要用裡面的錢，就必須破壞它。這個設計可以教育小孩儲蓄的概念：用錢時必須慎重考慮。現在的撲滿多在底部開有小洞，並用橡膠封住，以便將錢取出。

在傳統的中國家庭，父母常會給孩子一個存錢筒，以幫助孩子養成存錢和理性消費的習慣。演變至今，許多存錢筒雖然已被忽視，但是我們認為還是會需要存錢筒來養成良好的儲蓄習慣，同時存錢筒不僅是拿來當作存錢用途之外，也是有守財的功用的。除此之外，使用對的存錢筒也會讓好運和財運提升喔！於是我們決定利用這學期所學，設計出實用的數位存錢筒。

個人貢獻

鄧勝宥	鍾恩貴
網路文獻查找	統整資料
模擬驗證	規劃動作流程
企劃書撰寫	繪製 Quartus 電路
影片報告	企畫書撰寫

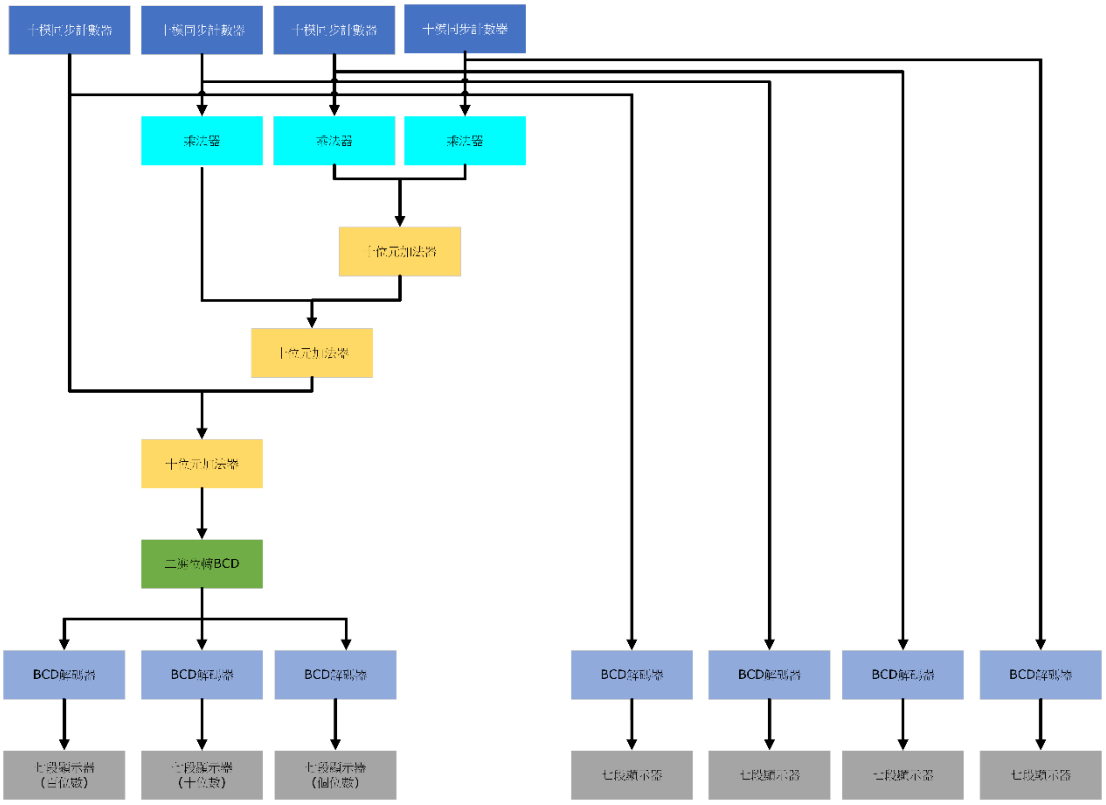
研究動機

從小到大我們都有都有儲蓄的習慣，我們被教導一點一滴的累積，積少成多小錢也會成大錢。而人類的隨著歷史的演進和科技的進步，也出現了各式各樣用來累積金錢的工具及設施，小到個人或多人所使用的存錢筒，大到提供大量的人進行存入提出和借貸的銀行。而隨著網路的發展科技的進步金錢也開始數位化，人們可以利用電腦或手機連上網路進行金勤的利用。新年家中會把一年當中存下來的錢拿出來清點，由於太多算起來費時，那時我就想到如果可以從一開始就開始管理的話之後清點就可以不用再浪費多餘的時間在計算零錢上。

存錢筒是一個傳統且簡便的儲蓄工具，結合現代科技的應用，藉著自己親手存入然後隨時觀察金額的變化也有以往所感受不到的成就感。廣義上他是一個錢幣蒐集裝置，但在用途上不只可以當存錢筒，一些小商家找零後的零錢也可以投入其中方便往後清點。

因此我們利用這用這學期所學，用按鈕模擬硬幣投入存錢筒，七段顯示器顯示各硬幣數量及總金額，製作出一目了然的智慧存錢筒，使零錢不再成為商家的負擔，顧客的噩夢。

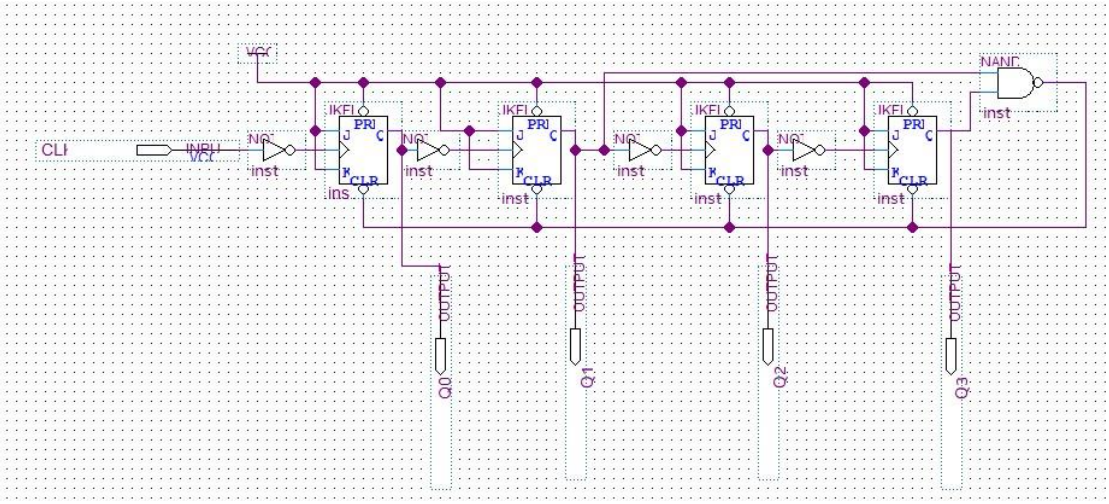
設計原理分析



首先四個十模計數器對應 4 種不同的硬幣，當硬幣投入後計數器就會計算共有幾枚硬幣，接下來透過乘法器依據硬幣數量程上相對應的幣值(10 元乘 10，50 元乘 50，1 元不用乘)，最後將乘出來的數值兩兩相加後得出零錢的總金額，最後將所得到的二進位數值轉換成 BCD 碼透過解碼器將總金額顯示在七段顯示器上。此外四個十模計數器也分別對應到一組解碼器跟七段顯示器用來顯示目前各硬幣的數量

(一)所使用到的電路

1.十模計數器

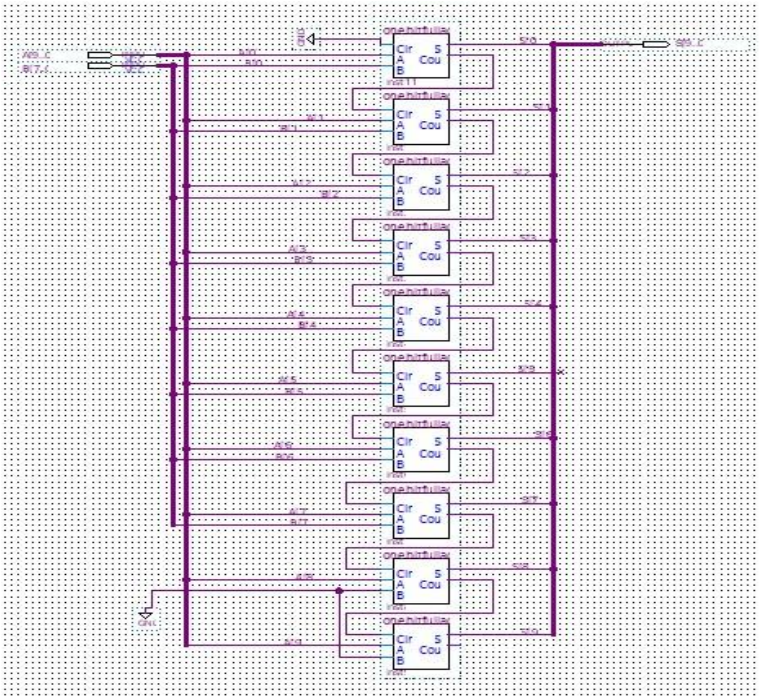


此電路為十模計數器，功能為計算零錢的個數，這次一共使用到 4 個，分別對應 4 種硬幣。

CLK	Q3	Q2	Q1	Q0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

模十計數器真值表

2. 加法器



這次所使用的加法器為 10 位元的，功能是將計數器所得硬幣數量經過乘法器後所得的值相加

3. 乘法器

```
 1  library IEEE;
 2  use ieee.std_logic_1164.all;
 3  use ieee.std_logic_unsigned.all;
 4 ⊟entity multiplier10 is
 5 ⊟port (A: in  std_logic_vector(3 downto 0);
 6 |       B: in  std_logic_vector(5 downto 0);
 7 |        M: out std_logic_vector(9 downto 0));
 8 |end multiplier10;
 9 ⊟architecture  mtp10 of multiplier10 is
10 ⊟begin
11 |
12 |   M <= A * B;
13 |
14  end mtp10;
```

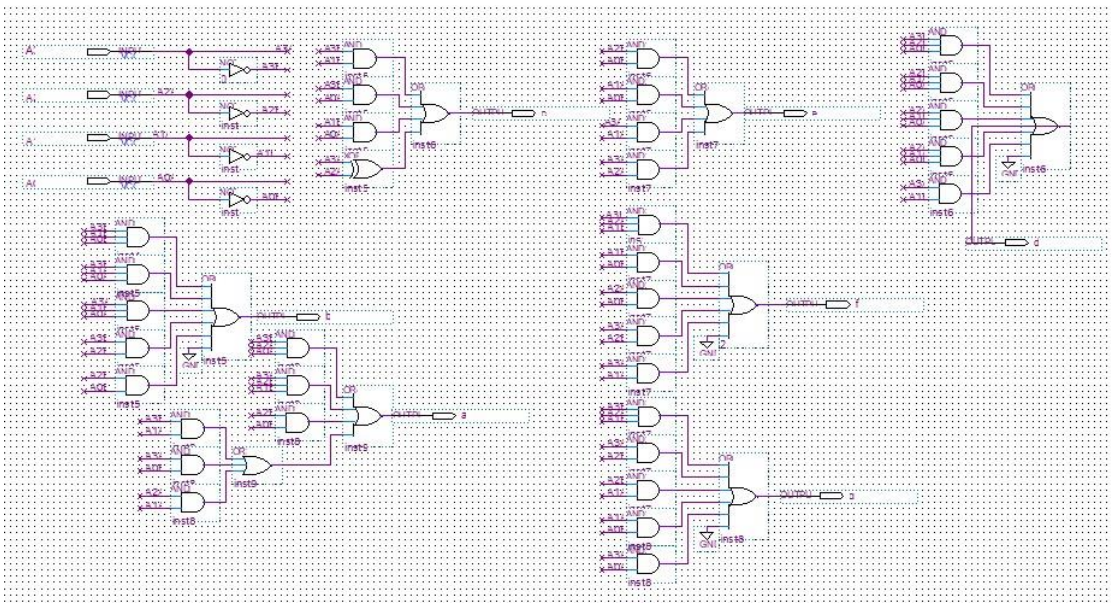
由於這次乘法器是使用十位元在運算，若是拉電路的話會導致效率低下，經過考量決定複雜的電路都採用 VHDL 語言撰寫。乘法器在電路中的作用是將計數器數到的硬幣數量程已相對應的的數字，得出結果及為該硬幣的總金額。

4.二進制轉 BCD 碼(8421)

```
module binaryBCD (
input [9:0] binary,
output reg [3:0] Hundreds,
output reg [3:0] Tens,
output reg [3:0] Ones
);

integer i;
always @ (binary)
begin
Hundreds = 4'd0;
Tens = 4'd0;
Ones = 4'd0;

for (i=9; i>=0; i=i-1)
begin

       if(Hundreds >= 5)
       Hundreds = Hundreds+3;
       if (Tens >= 5)
       Tens = Tens + 3;
       if (Ones >= 5)
       Ones = Ones + 3;

       Hundreds = Hundreds << 1;
       Hundreds[0] = Tens[3];
       Tens = Tens << 1;
       Tens[0] = Ones[3];
       Ones = Ones << 1;
       Ones[0] = binary[i];
       end
end

endmodule
```

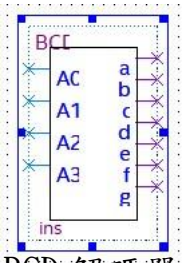
二進位轉 BCD 碼是用於最後將總金額計算完成後會獲得一串二進制十位元的數字，而想將數字顯示至七段顯示器上轉換成 BCD 碼是最簡單的方法，這個電路是採用 Verlilog 的語法去寫出來的，更直觀更符合邏輯。

5. BCD 解碼器

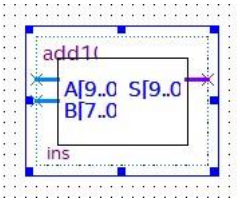


這次專題所使用的 BCD 解碼器是先前實習所留下來的電路，在這裡解碼器的功能為將轉換後的 BCD 碼送到解碼器內部，這樣在輸出端接上七段顯示器即可正常顯示

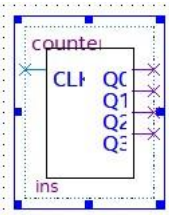
二. 集成模組圖示



BCD 解碼器



加法器



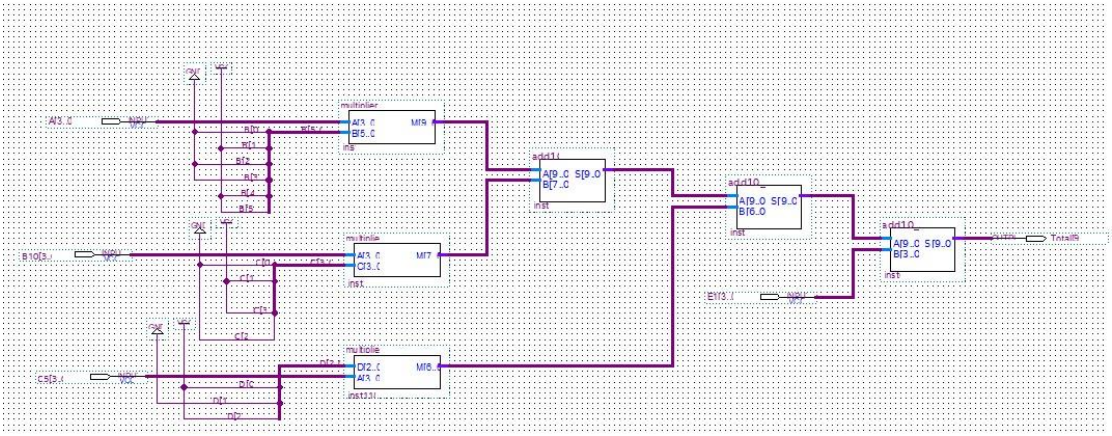
計數器

二進制轉 BCD 圖示

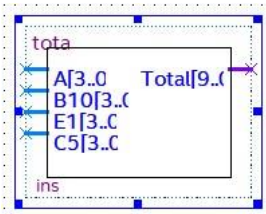
乘法器圖示

三. 總電路

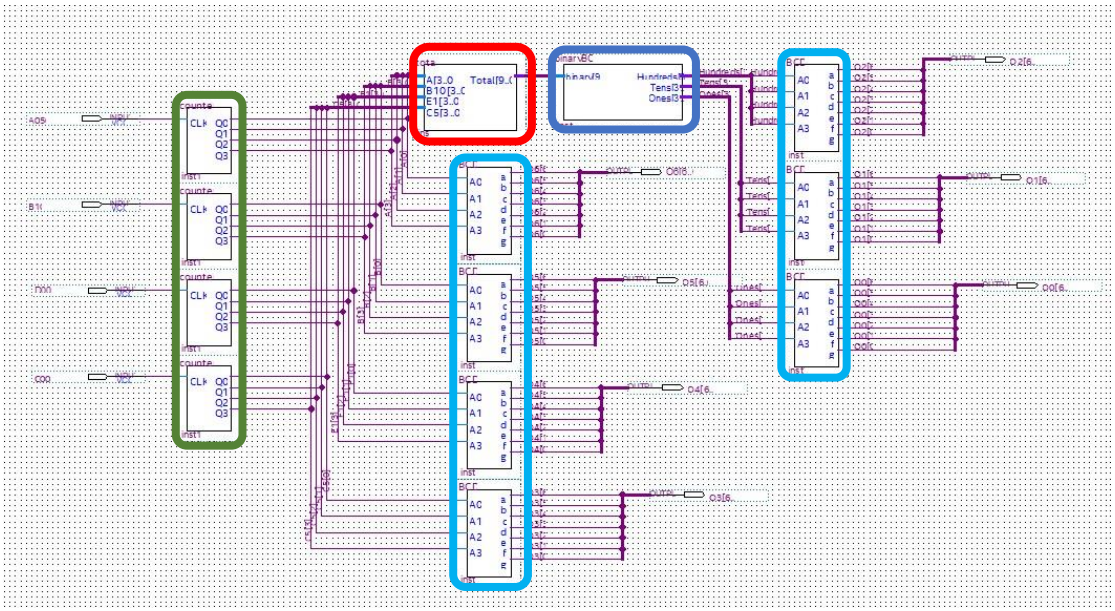
1. 計算金額電路



此專題我先負責計算金額的電路封裝在一個封包裡面，為保持電路整潔



2. 總電路



上圖為這次專題的集成電路圖，紅色框框為負責計算總金額的邏輯方塊，將綠色框框中的計數器所接收到的數量分別送至計算金額以及藍色框框的 BCD 解碼器，而計算金額的部分計算完成後將數值送到深藍色框框轉換成 BCD 碼在送到 BCD 解碼後透過七段顯示器輸出即可總金額及各硬幣的數量

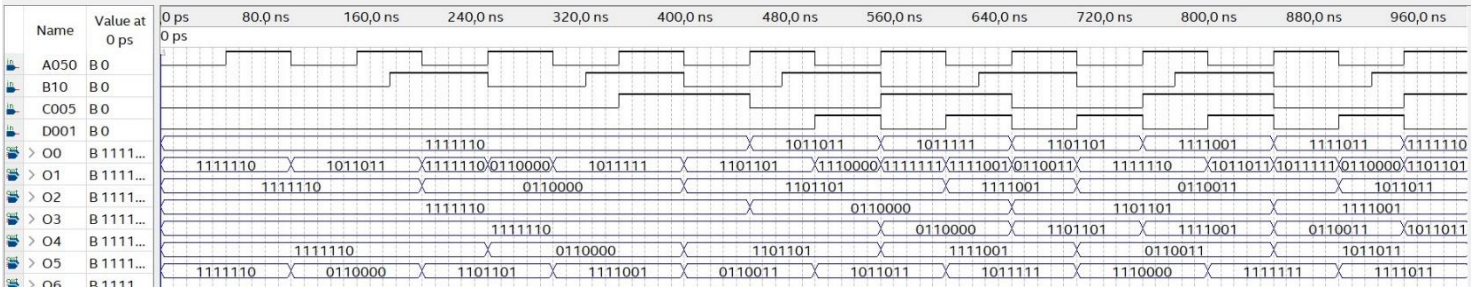
軟硬體系統

由於疫情之因素，所以專題呈現的方式就只有使用軟體模擬驗證的方式呈現，故這次只使用到軟體部分

軟體： Quartus Prime Lite Edition (version 20.1)

實驗結果比較

模擬驗證結果



腳位自訂義：

輸入

A050➔模擬 50 元硬幣投入
B10➔模擬 10 元硬幣投入
C005➔模擬 5 元硬幣投入
D001➔模擬 1 元硬幣投入

輸出

O0➔七段顯示器-總金額個位數
O1➔七段顯示器-總金額十位數
O2➔七段顯示器-總金額百位數
O3➔七段顯示器-1 元硬幣數量
O4➔七段顯示器-5 元硬幣數量
O5➔七段顯示器-10 元硬幣數量
O6➔七段顯示器-50 元硬幣數量

動作說明:

在模擬驗證中先模擬 50 元投入 1 次，在投入前 O0~O6 都顯示為 0，投入後可以觀察到 O5 顯示為 1、O1 顯示為 5，代表投入 1 枚 50 元硬幣，總計 50 元，在 240 ns 處共計投入 2 枚 50 元硬幣 1 枚 10 元硬幣，共計 110 元，在 O2 顯示為 1、O1 顯示為 1、O6 顯示為 2、O5 顯示為 1，以此類推…

結論

參考文獻

1. http://yhhuang1966.blogspot.com/2019/06/counter.html

2. https://www.cyut.edu.tw/~yfahuang/chap04.pdf

3. https://www.twblogs.net/a/5ef592b4f1151da65ed36a20

4.上課所用之簡報 PPT