

邏設期末專題

專題題目：呼吸型超能 ALU 計算機

一、 專題構想

牙膏與牙齦是是一對好兄弟，哥哥牙齦在召喚峽谷學院就讀電子科表現的超級無敵良好，是個學霸，小他三歲的弟弟牙膏對他充滿仰慕之情。今天恰好是召喚峽谷的校際期末考周，弟弟牙膏在這學期修了邏輯設計這門課程，牙膏覺得他快完蛋了，因為他的期中考題是**實現 ALU 電路**，考試時共有 12 題，分別是實現出四則運算、邏輯運算與移位運算共 12 種，老師限用自己設計好的電路實行運算並實現出來，不可以用紙與筆。弟弟牙膏心想，我邏輯設計已經重修兩次了，為了過邏輯設計這門課，他已經兩年沒有玩召喚峽谷了，所以他決定請哥哥牙齦幫忙，牙齦當然非常樂意。此時牙齦心想，弟弟牙膏是個白癡，功課不好就算了，就連玩個召喚峽谷都是會滑起來叫 HASAKI 的傻子，都被鱷魚單殺了 13 次了還說召喚峽谷是佛心公司，但牙齦也不想讓牙膏失望，**因此牙齦決定設計一個笨蛋都看得懂怎麼用的電路**，讓弟弟牙膏在考場上能夠順利通關。

為此，牙齦請了我們組別 20 來幫助他，左思右想後，那乾脆使用 **pyzq-z2** 來實作好了，這麼好的板子剛好落入我們手中，當然要好好利用，因此組別 20 與牙齦討論過後決定實作一個笨蛋都看得懂怎麼用的 ALU 計算機，名稱為**呼吸型超能 ALU 計算機**。

二、 預期方向與設計

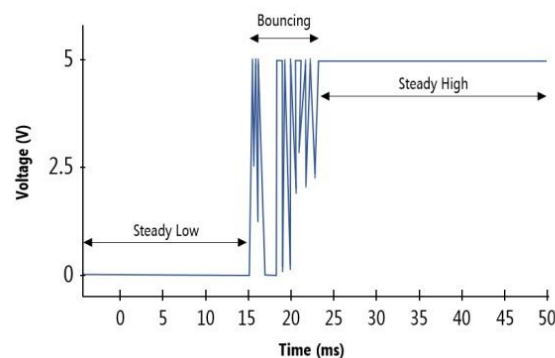
需要用到的東西有 pyzq-z2、公公線、麵包板、兩顆七段顯示器、電阻。預期方向是設計一個狀態機，使用 led 燈表示狀態機目前狀態，透過 pyzq-z2 的按鈕去控制輸入訊號與想要的運算模式，共設計 12 種，然後經由狀態機的控制與推進，再來輸出結果。

三、 實作方式

程式碼內容共有三段分別是 topp.v、debounce_cir.v、seg7.v，還有提供 tbb.v 作為測資。**要注意若單純要跑模擬看波型的話只需放 seg7.v 跟 tbb.v**，因為 topp.v 是用來接 .xdc 檔的腳位，debounce_cir.v 是因為按鈕要防跳彈所以是物理現象，這兩者都沒辦法用 testbench.v 測試。

首先是 topp.v，它就是作為最大的模塊，負責把 debounce_cir.v、seg7.v 拿去與 .xdc 作為溝通的橋樑，本次實作用到的 4 個按鈕是 btn[3:0] 其中 btn[2:0] 需要接上防跳彈電路，所以它的裡頭就會有 1 個 seg7.v 的模塊與 3 個 debounce_cir.v 的模塊。

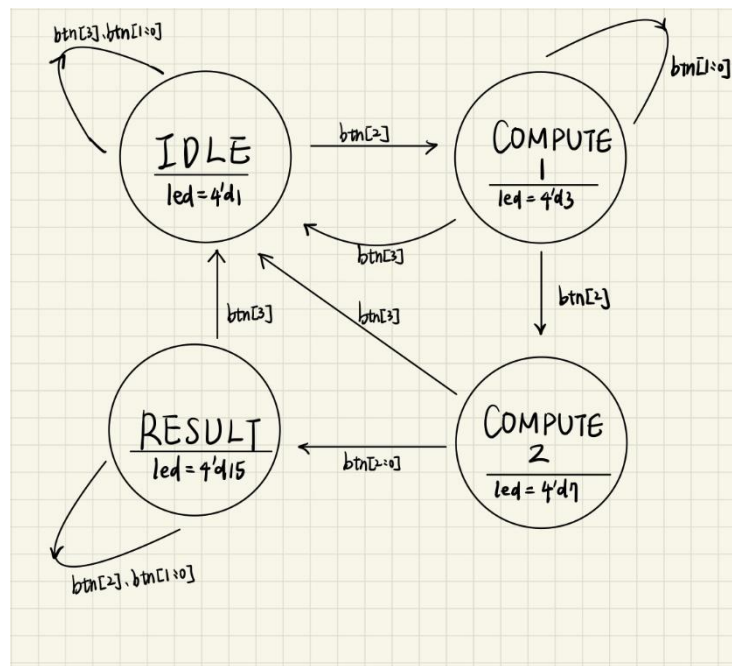
再來是 debounce_cir.v，序向電路的 clock 的速度很快 (125MHz)，若直接將按鈕訊號接到 topp.v 會發生按一下按鈕但觸發多次的問題，為此需要一個電路來防止此物理現象發生，以下圖[1]是示意圖。



圖[1]

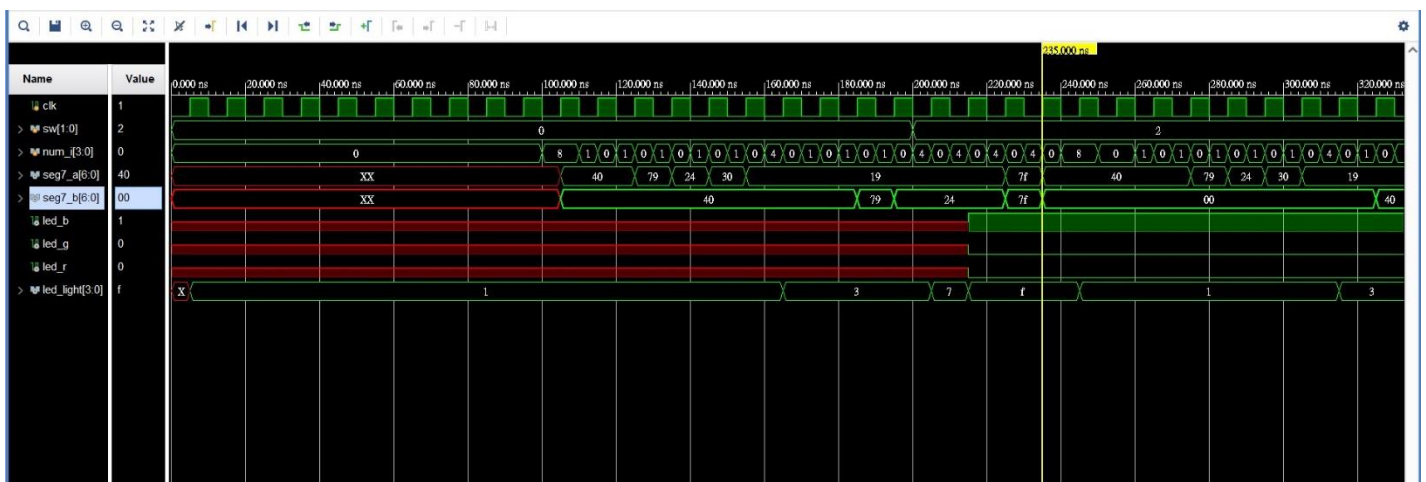
最後是主程式 seg7.v，根據 .xdc 檔給的腳位名稱，本專題共使用 btn[3:0]、sw[1:0]、led[3:0]、led4_b、led4_g、led4_r、ar[13:0]。首先設計**米利狀態機**，如圖[2]所示，共有四個狀態，分別是 **IDLE**、**COMPUTE1**、**COMPUTE2**、**RESULT**。軟體上，IDLE 為初始狀態，等待第一個值進入，每按 btn[0] 就會把站寄存器加 1，按 btn[1] 就會減 1，收到第一個值後進入 COMPUTE1，等待第二個值進入，同樣地收到第二個值後進入 COMPUTE2，此狀態很重要，因為是負責運算 ALU 的狀態，此時調整 sw[1:0] 與按下 btn[2]~btn[0]，若按下 btn[2] 會根據 sw[1:0] 的值，由小到大實現加減乘除；按下 btn[1] 是邏輯運算，由小到大分別是 OR、AND、XOR、XNOR；按下 btn[0] 是移位運算，由小到大是左移、右移、左移後 NOT、右移後 NOT，ALU 運算完後，最後進入 RESULT 狀態，只要按下 btn[2] 就能顯示結果。硬體上，代表狀態機的狀態我使用 led[3:0]，只要在

IDLE，亮 led[0]；在 COMPUTE1，亮 led[1:0]；在 COMPUTE2，亮 led[2:0]；最後 RESULT 亮 led[3:0]，led4_b、led4_g、led4_r 是 RGB 燈，為了突破 sw[1:0] 只能做 4 種 ALU 運算的限制，因此用藍綠紅燈分別代表算數、邏輯、移位運算，總共 12 種，只要按下 btn[2:0] 就會亮起各自代表的顏色，七段顯示器我用的是共陽極，0 為 5V，1 為 0V，運用 16 進位，把第一顆接到 ar[13:7] 代表第一個位數，第二顆接到 ar[6:0] 代表第二個位數。最後，我在每個狀態下皆把 btn[3] 設為 reset，如果執行不順利或錯誤按下 btn[3] 就會歸零。



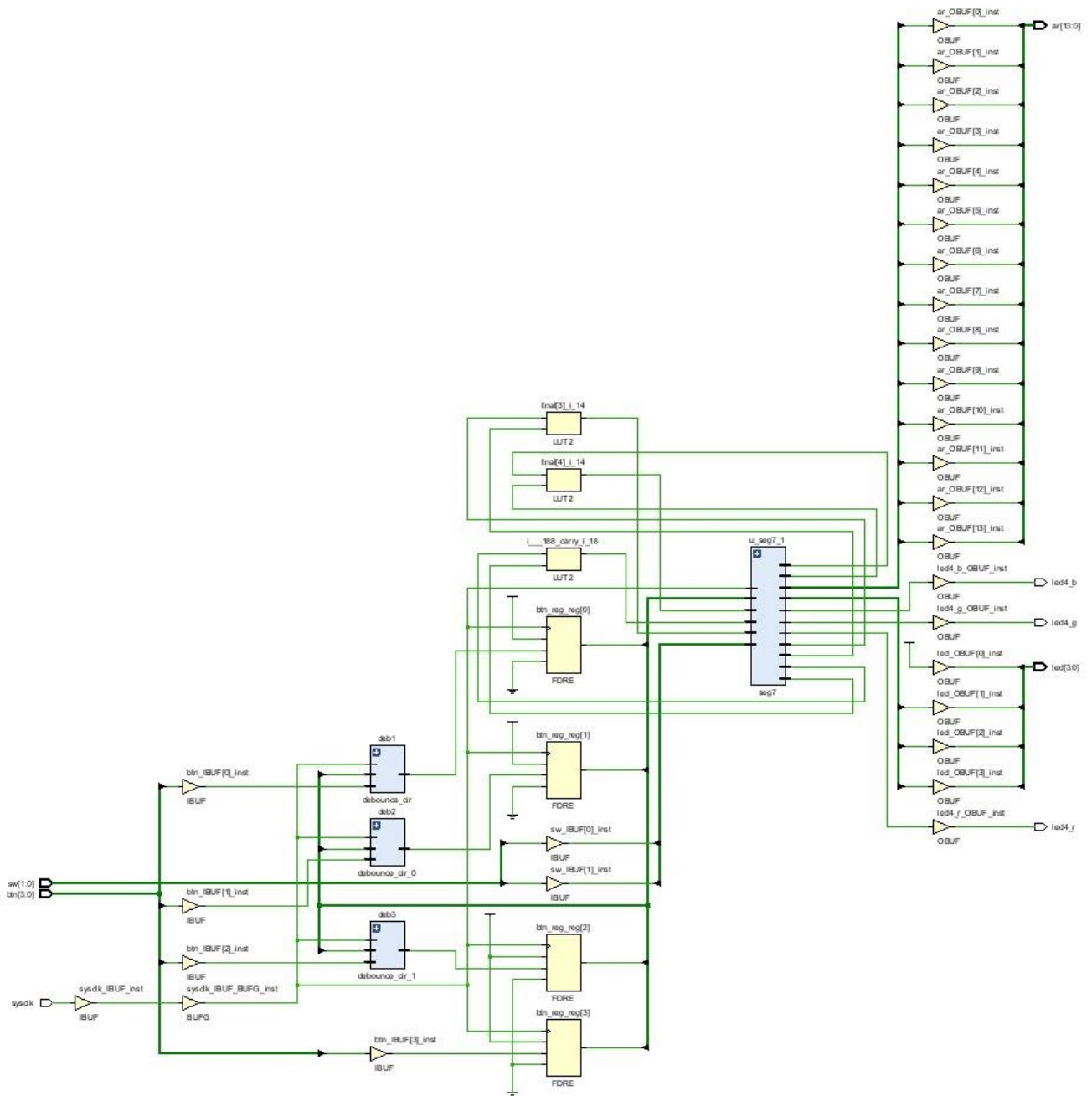
四、實作成果

以下圖[3]為本次模擬圖(seg7.v 與 tbb.v)，可以看到狀態機(led)的轉移。



圖[3]

以下圖[4]先放本次專題電路的合成圖，有點大張。



圖[4]

由合成的電路圖可以確定接腳設定都是正確的，之後的繞線與比特流也沒什麼問題，所以就可以驗證硬體上的結果是否正確。

因為無法用拍照傳達 demo 的樣子，故以下連結是測試結果的影片，

<https://youtu.be/VrkcWe8Lk0U>

影片中共有六筆測資，如下：

1. $9 + c = 15$ 亮藍燈
2. $6 * b = 42$ 亮藍燈
3. $7 \text{ OR } a = 0F$ 亮綠燈
4. $b \text{ AND } 7 = 03$ 亮綠燈
5. $3 \gg 5 = 60$ 亮紅燈
6. $8 \ll 2 = 02$ 亮紅燈

五、 未來展望

本次專題利用狀態機的方法去實作 ALU 電路，透過 LED 燈跟 RGB 燈去展現狀態機的當前狀態，除了突破了很多按鍵上的限制，還有適當的色彩變化，讓使用者方便判斷自己正在做什麼運算，如果沒有設計狀態機的話，很難看出是否做出正確的運算與重置功能。

目前的缺點還是在於按鈕的部分，關於防跳彈的電路，縱使有設計過但不是很穩定，有時候因為物理現象可能會按到兩次或是沒按到，關於這點我覺得可以再上網找更好的寫法去撰寫。再來是使用者介面的部分，也許之後可以利用像是外接 4*4 鍵盤的方式去進行，本次專題是利用 **按鈕去作出計數器** 來實現想要的數字，如果可以改良成 4*4 鍵盤，只要按一下就能馬上出現想要的數字了。還有一項是七段顯示器的接線，可以利用 **sweeping** 的方式去節省 arduino 的腳位，因為 125MHZ 的關係，一個週期只有 8ns，兩個七段顯示器輪流輸出，因為速度很快，看起來就會是兩個同時亮起。還有一些瑣碎的細節像是優化電路設計、運算速度、降低功率等都是可以再考量的事情。

最後，也許可以利用四位數的七段顯示器去做這個專題，當時沒有顧慮清楚，如果用四位數的七段顯示器的話，儲存空間就能更大，能做的運算就會更多，甚至可以做更多輸入的運算。本次專題到這裡就結束了，希望牙膏能夠使用這個**呼吸型超能 ALU 計算機**去完成他邏輯設計的期末考，祝他考試順利~