**Lab 5 110060027朱豐蔚**

1. **prelab (40s down counter with pause function)**

**Design Specification**

Input: clk, reset, bottom, bottom\_reset.

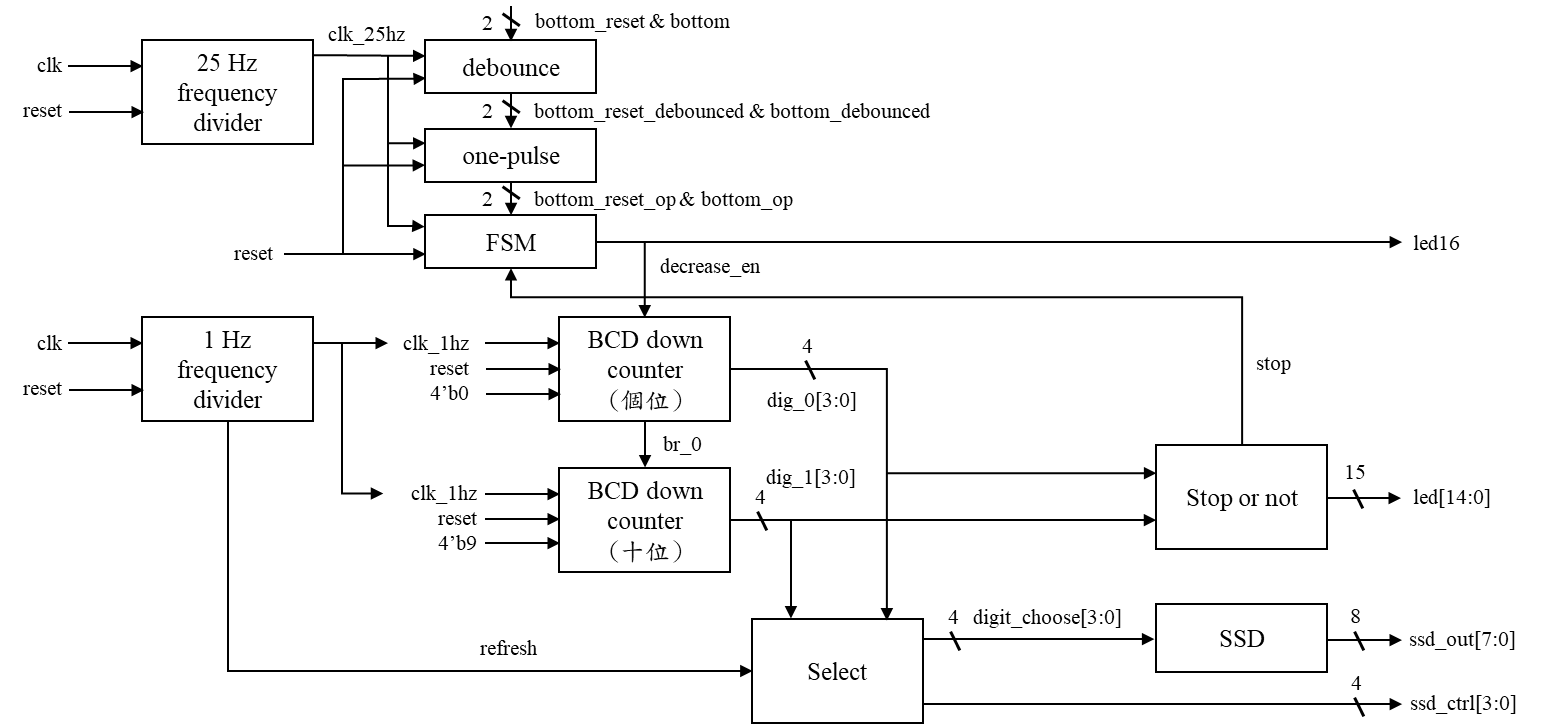
Output: ssd\_out[7:0], ssd\_ctrl[3:0], led[14:0], led16.

**Design Implementation**

Specification of the frequency divider:

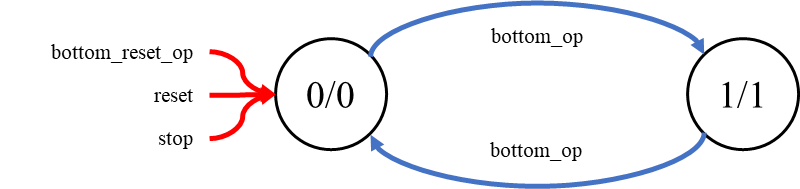
使用一數50M次的counter來對原本100M Hz的clk進行除頻，使輸出為1 Hz的訊號clk\_1hz，以提供後續數秒使用。

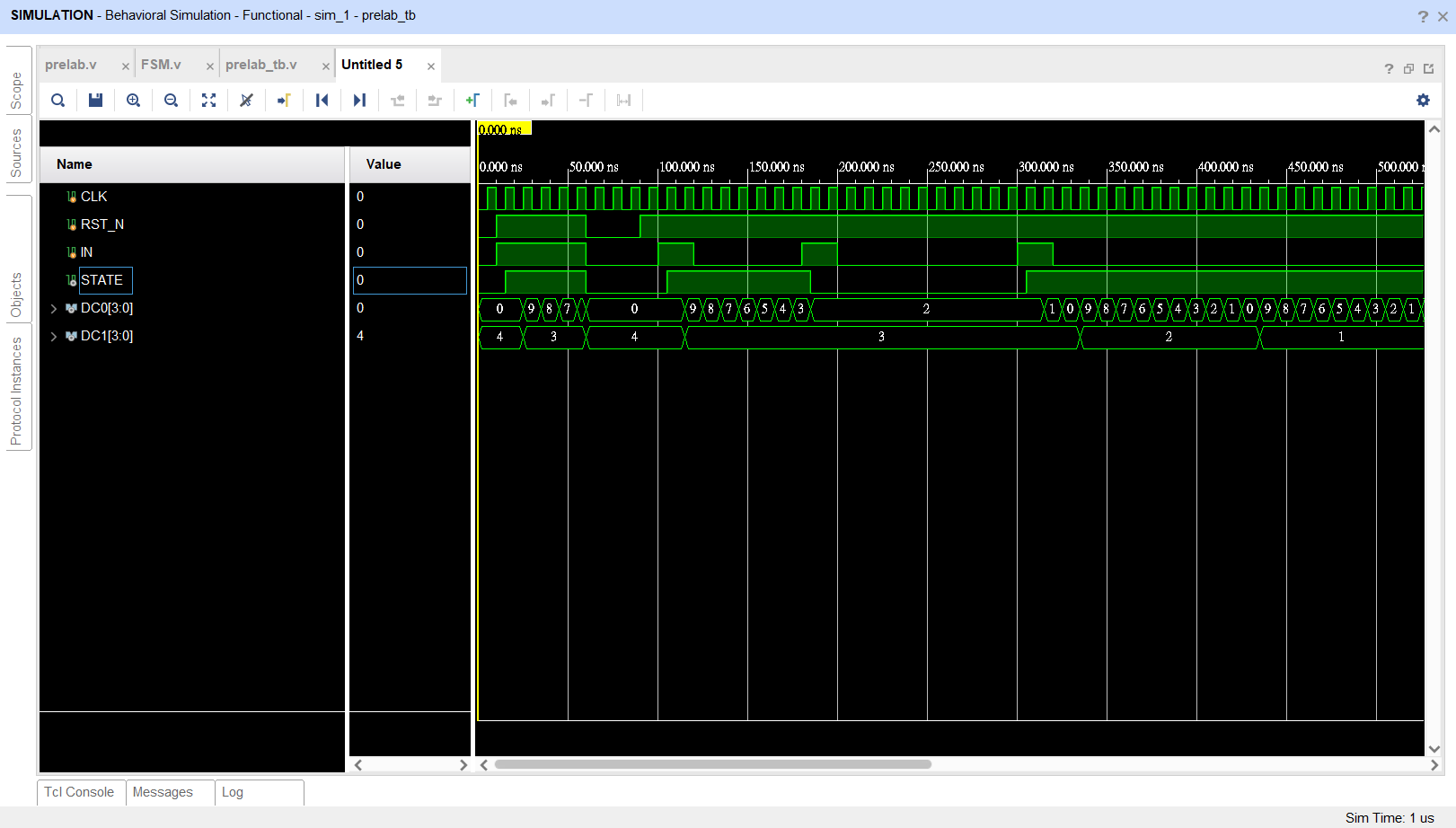
另使用一數2M次的counter來對原本100M Hz的clk進行除頻，使輸出為25 Hz的訊號clk\_25hz，以提供debounce、one-pulse、FSM使用。

 Block diagram:

FSM design:

考慮此題，僅有停止數、繼續數兩種狀態，故只需要一位元即可表示所有state。

使用與debounce、one-pulse相同的clk\_25hz作為判斷速度，每當收到bottom\_op訊號時切換state，當收到reset或bottom­\_reset\_op或stop時重置FSM，使FSM回到停止數的狀態，而最後輸出decrese\_en作為state的判斷。

simulation waveform:

**Discussion:**

跟著prelab的設計走，第一次接觸按鈕設計的我們，很快就可以上手並找到debounce、one-pulse模組中的錯誤，prelab中要求的state也不多，很方便我們進行FSM的撰寫，我覺得這次prelab的難度恰到好處。

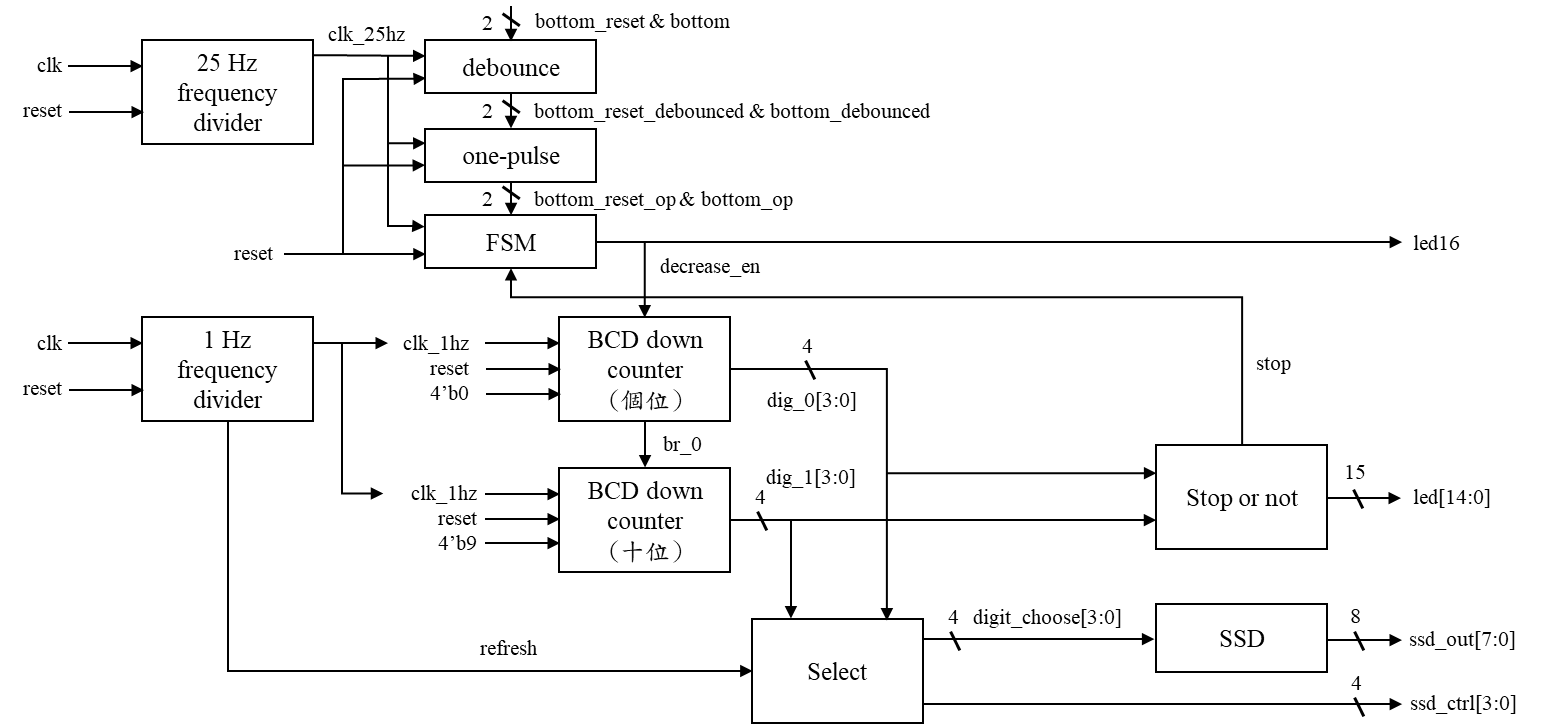
1. **exp\_1 (40s down counter with pause function)**

**Design Specification**

Input: clk, reset, bottom, bottom\_reset.

Output: ssd\_out[7:0], ssd\_ctrl[3:0], led[14:0], led16.

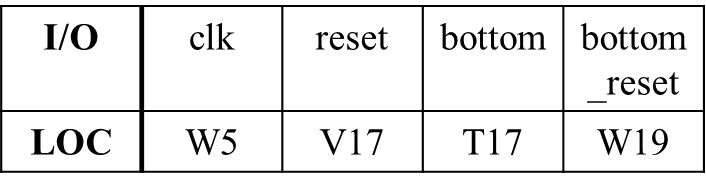
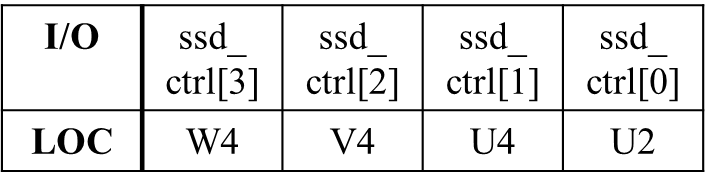
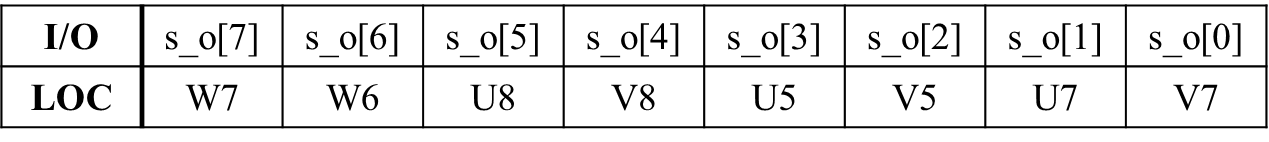
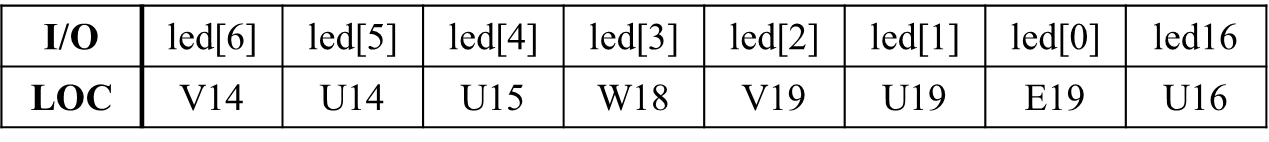
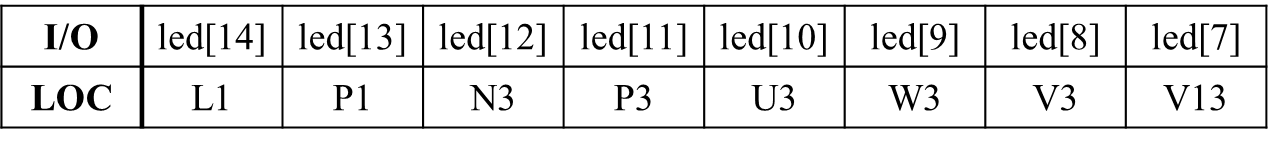
**Design Implementation**

 Block diagram:

Design flow:

將prelab中測試完全的FSM模組套入進來，並加上帶有enable輸入的BCD down counter，使用decrease\_en作為此counter的enable。此外，與Lab4-4相似，透過兩個counter分別去紀錄十位數與個位數，其中將br\_0作為串連兩個counter的進位引導，最後引入select、SSD模組以進行數字的獨立顯示。

在題目另外有要求當數到0時，停止記時並點亮所有led，我使用if的寫法，當dig\_1跟dig\_0都為4’b0時，使stop變成1，led全亮即可達成題目所求。

I/O pins assignment :

**Discussion:**

我覺得這題是承上啟下的，承接著Lab4-4的兩位元counter，而往下延伸出FSM與按鈕的debounce、one-pulse，但不一樣的地方是此題的counter為down counter，FSM也較為簡單，我認為這題的難度是恰到好處的。

在此題的撰寫當中，我遇到最大的困難就是在debug上，因為我是一次性把所有模組寫完再進行implement的，所以當我按按鈕沒有反應時，我完全不知道是哪個環節出了問題，花了很長的時間才找到是因為我在one-pulse模組中多打了一個 ~。這次經驗也告訴了我，一次做太多反而不是一件好事，應該要逐步慢慢前進，做多少先看多少，才能夠提早發現錯誤，及時修正。

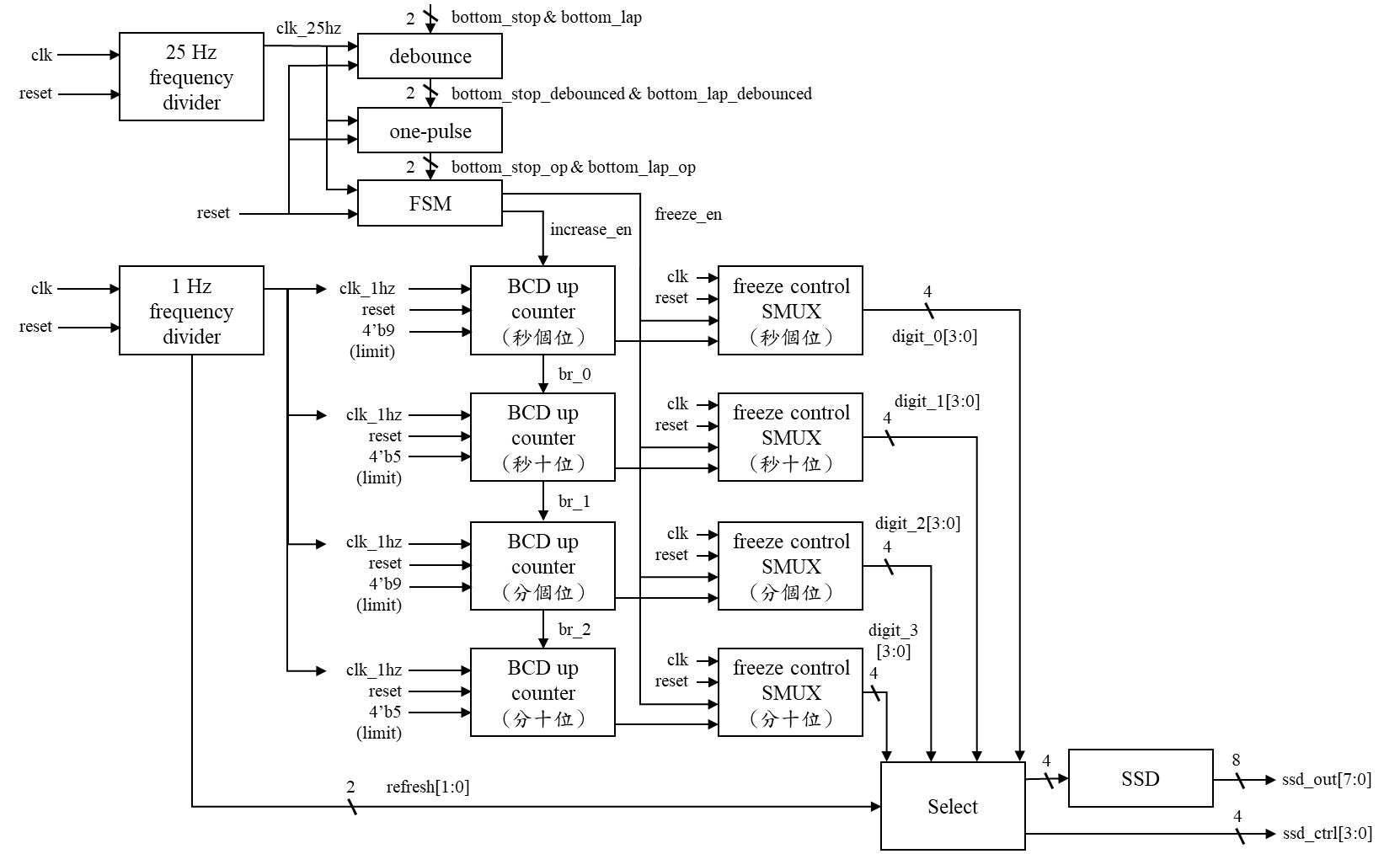
1. **exp\_2 (stopwatch)**

**Design Specification**

Input: clk, reset, bottom\_stop, bottom\_lap.

Output: ssd\_out[7:0], ssd\_ctrl[3:0].

**Design Implementation**

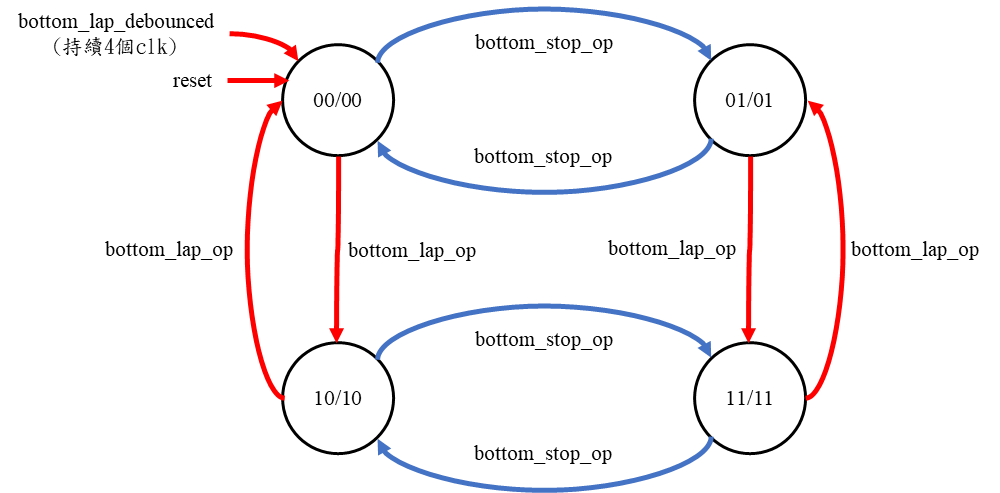
 Block diagram:

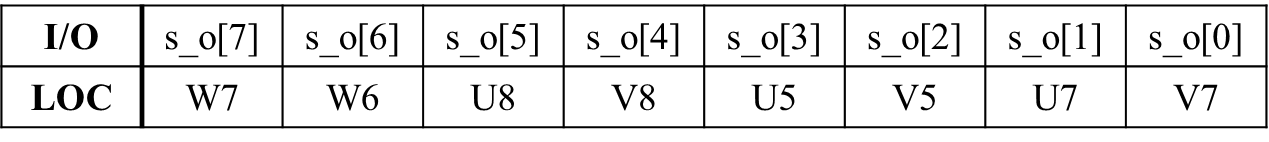
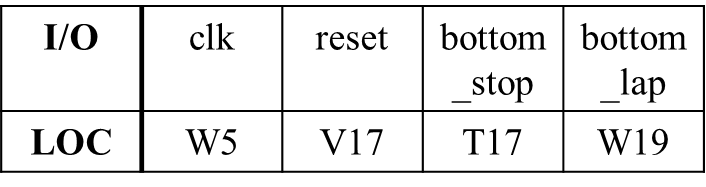
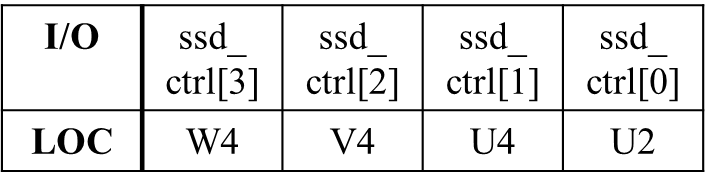
Design flow:

相比第一題，這題的複雜處在於state變多了，考慮是否count、是否freeze，總計就有4種state，因此需要兩位元來表示。

在FSM的設計中，如下圖，設定00為不數且不freeze、01為數且不freeze、11為數且freeze、10為不數且freeze，因此，令output中的第一項為freeze\_en，第二項為count\_en，即可達成題目的要求。

另外，freeze控制，我使用了一個SMUX來做控制，若freeze­\_en = 1，就讓輸入等於輸出，在下一個clk後，就會維持不變；而當freeze­\_en = 0，就讓輸入等於counter中的值，即可正常顯示出來。



I/O pins assignment :

**Discussion:**

在這題的撰寫當中，我遇到最大的問題就是長按reset的設計，因為原先我一直是使用bottom\_lap的one-pulse訊號作為window的判斷輸入，但一直發現沒反應，最後我另外增加4個燈號作為判斷window狀態的輸出，才發現window要使用bottom\_lap的debounced訊號才可成功。

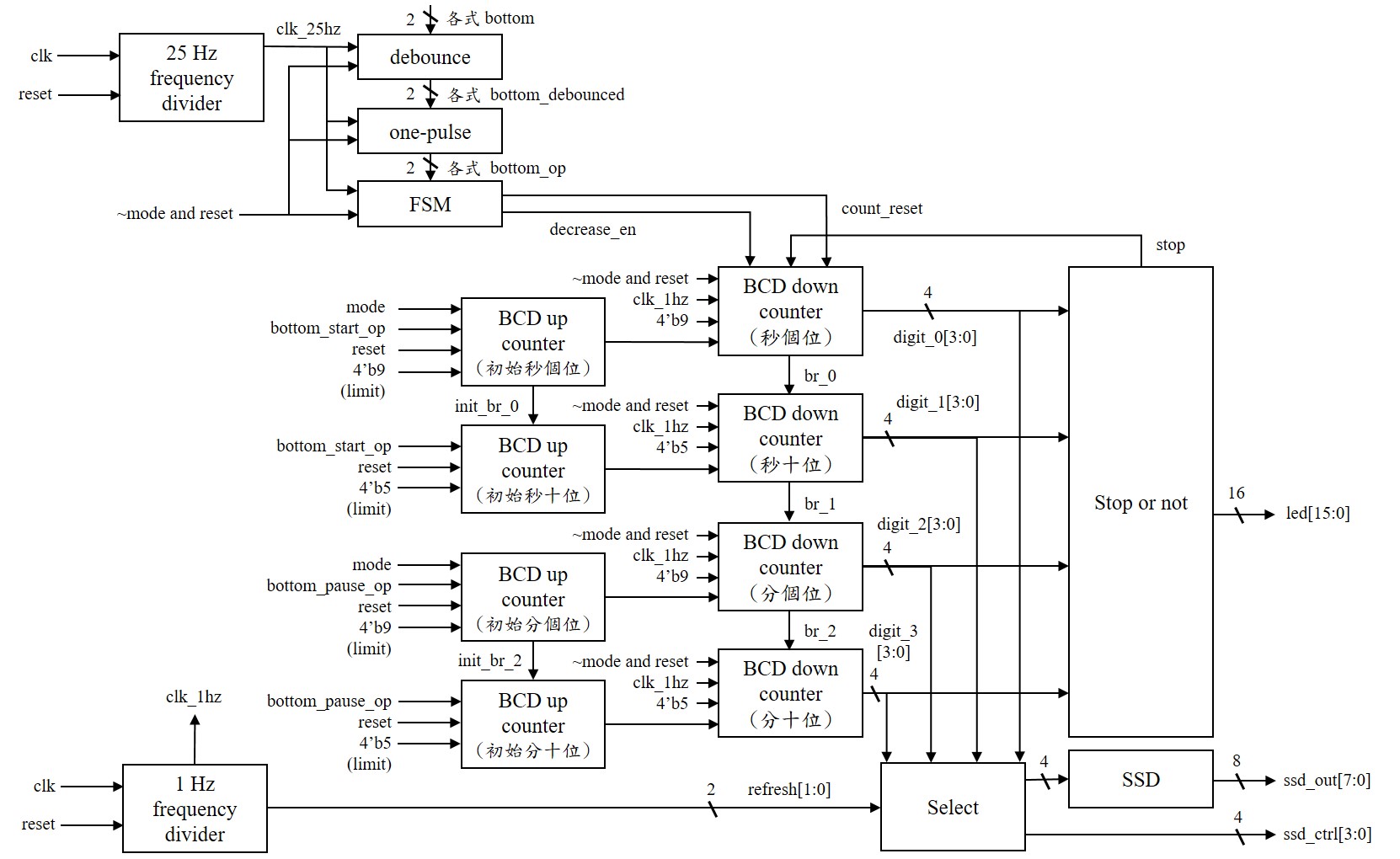
1. **exp\_3 (timer with setting function)**

**Design Specification**

Input: clk, reset, mode, bottom\_start, bottom\_pause.

Output: ssd\_out[7:0], ssd\_ctrl[3:0], led[15:0].

**Design Implementation**

 Block diagram:

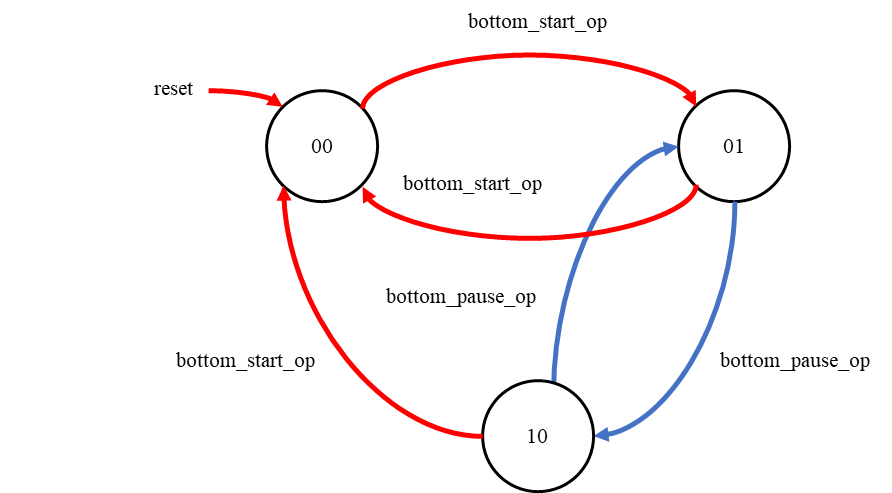
Design flow:

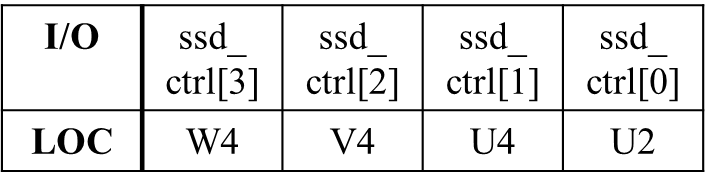
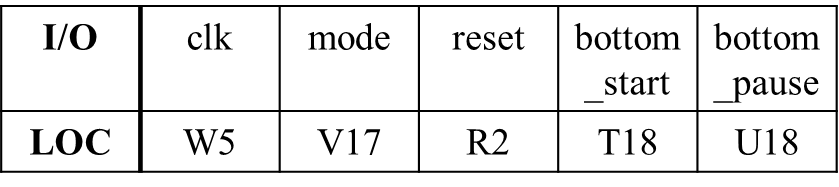
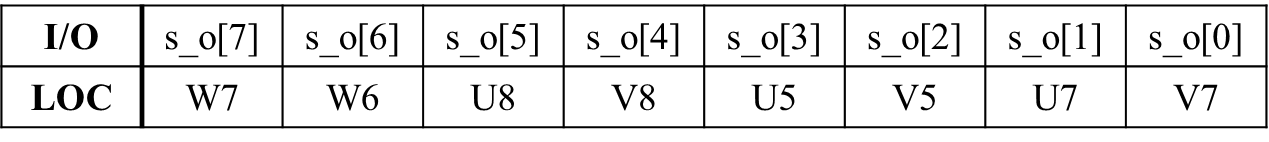
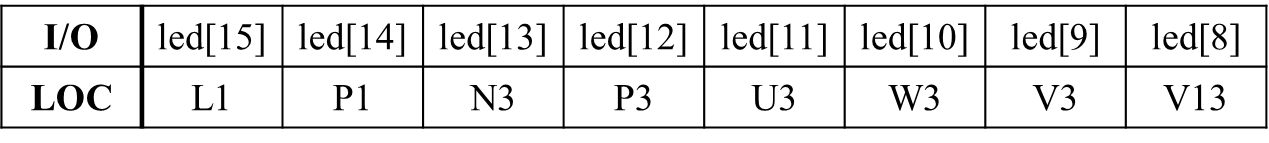
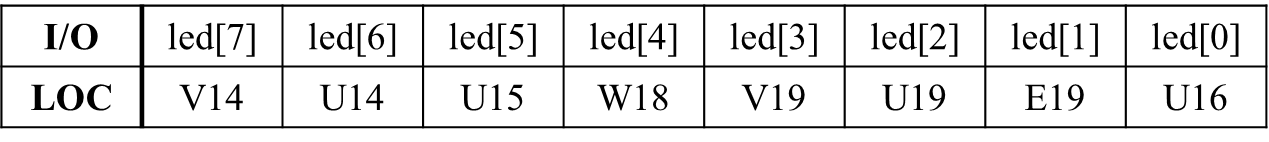
首先先考慮此題所需要的state，有reset後的初始態，訂為00，在來就是可以數的狀態，訂為01，最後是暫停態，訂為10。

當在初始態按下start後就會進到數與暫停的循環，此時pause鍵會讓state在01、10之間互相切換，此時控制counter是否數的訊號即為count\_en。而不管在01還是10，再次按下start就會回到00態，此時讓count\_reset變為1，讓counter重置，回到設定的初始值。

接著要設計模式的切換與設定初始值方式，我想到的方法是另外創立4個counter去作為timer的初始值紀錄，於是設定只會影響到初始值counter，再由其值去做為timer的initial輸入，即可在不干擾timer的情況下達成設定效果。

同exp\_1，一樣會需要設立當所有數值歸0時的led輸出，我一樣選用if的寫法，當所有數值都為0時，產生stop訊號，讓timer不再倒數，且讓所有led燈發亮。

最後的數值呈現，由於沒有freeze功能，所以不需要如exp\_2一樣加入SMUX去選擇，可以直接將數值輸入到select當中，再藉由refresh訊號去做數值的獨立顯示。

****I/O pins assignment:

**Discussion:**

我認為exp\_3相比於exp\_2，在FSM的撰寫上較為簡單，因為state數僅有3個，且彼此的切換簡單易懂，但更困難的地方是在設定模式的創建，需要引入switch作為模式的切換，還需要考慮如何加入設定的功能。

我首先設想了兩種設定方式，第一種如上述，將初始值counter獨立出來，而第二種則是合在一起，但合在一起的方式第一是難寫，第二是輸入太多太雜容易互相干擾，因此我選擇了第一種方式作為我的設定寫法。

**======================================================================**

**Conclusion**

在這次Lab中，我最大的收穫就是學會了按鈕的寫法，透過debounce消除雜訊，在利用one-pulse使訊號單一不重複，這兩個模組缺一不可。尤其是在exp\_2中的長按功能，更讓我了解到了debounced訊號與one-pulse訊號個有其優缺點與應用之處，one-pulse雖然不會重複切換state，但同時卻無法處理長按這種需要紀錄時間長度的function，因此，如何靈活運用這兩種訊號的型態應該會是今後最大的課題之一。

此外，我也想到one-pulse的這種寫法並不侷限於按鈕，應該試用任何長訊號來源，因此，更可以透過one-pulse讓switch也有按鈕的效果，這點可以應用於Lab6的撰寫當中，豐富了我們處理題目的方法與手段。

**References**

FSM編撰方式: (from 第五週上課講義)

Freeze功能編撰方式: (from 第五週上課講義)