

Lab 5: Timers

106061218 李丞恩

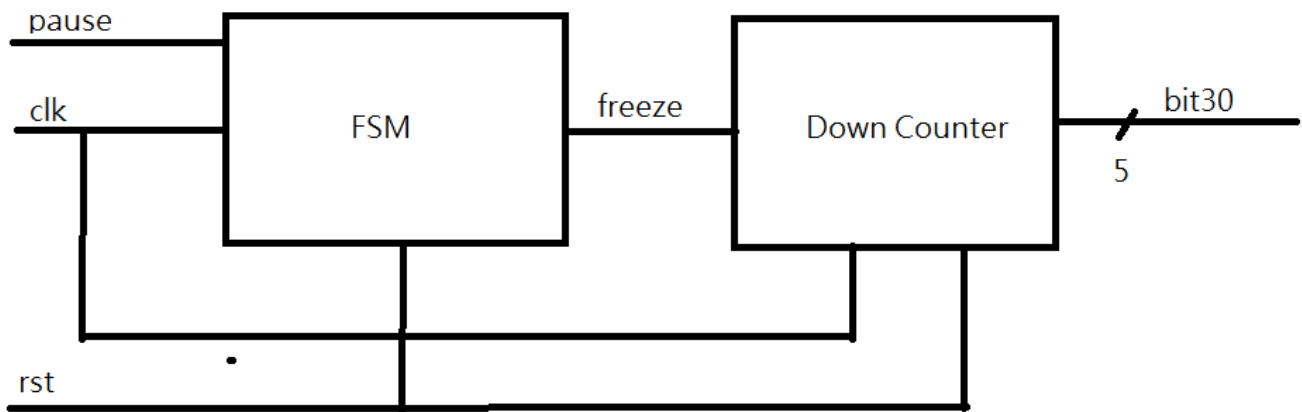
Pre-lab

Design Specification

(1) I/O ports

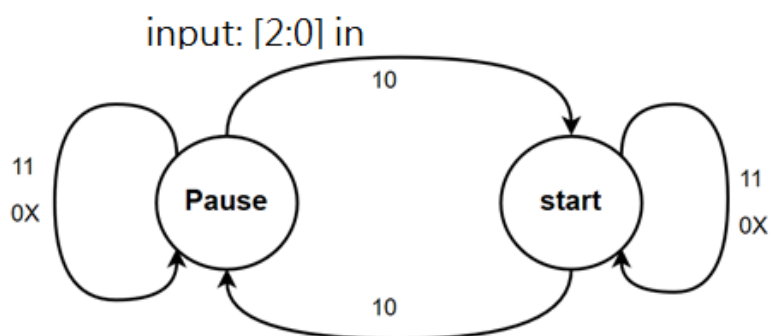
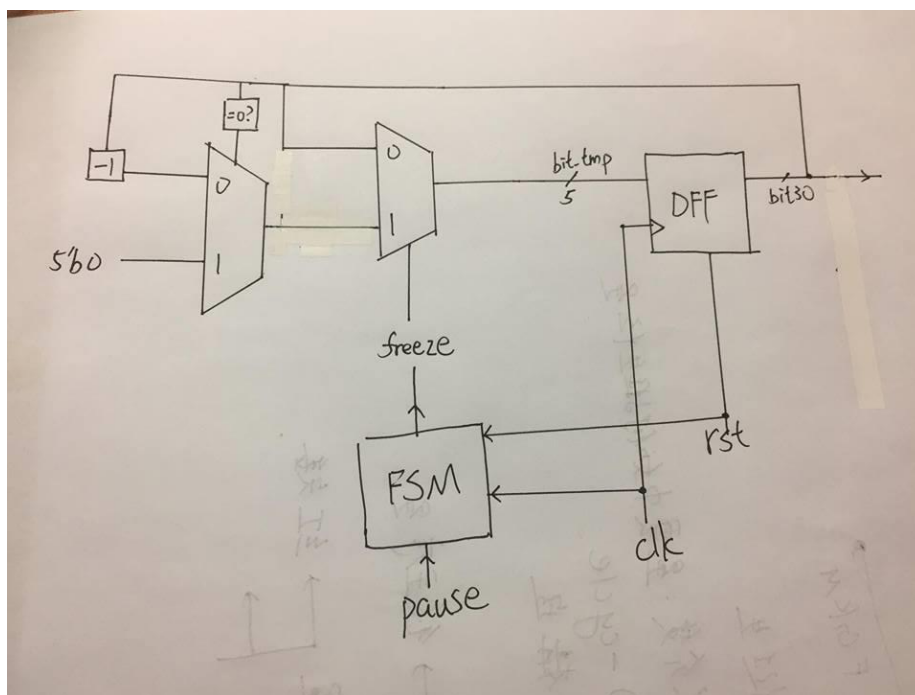
input rst reset 或開關
input clk 全域時脈
input pause 按鈕
output reg [4:0] bit30 即 30 秒倒數計時

(2) Block Diagram



Design Implementation

(1) Logic Diagram



(2) Verilog codes

這次我的 code 在原本的檔案內有打註解，在此就不另外說明。

(3) Verification

Test Bench 跑出來的結果如圖示：

由圖可知控制暫停與否的 freeze 訊號為 0 時，counter 就會停止作用。



2. 30-second down counter with pause function.

Design Specification

(1) I/O ports

Input:

Clk: global clock

Rst: 接在 button 上，採正緣觸發。

pb_in: 接在另一個 button 上，負責暫停。

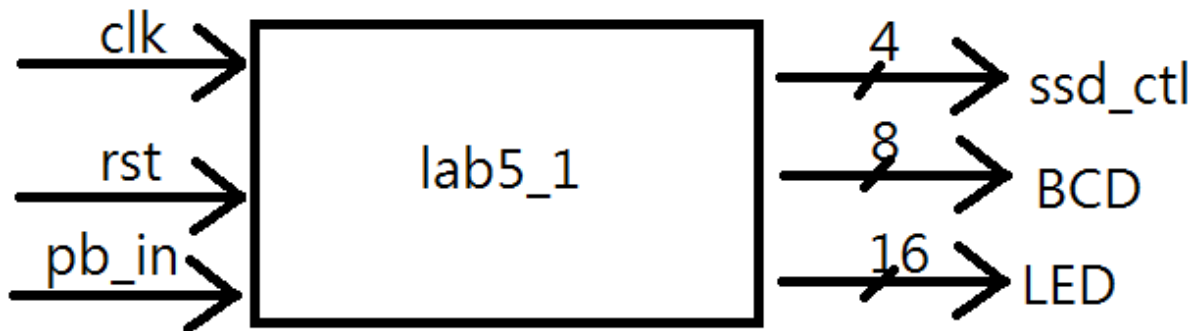
output

[7:0] BCD

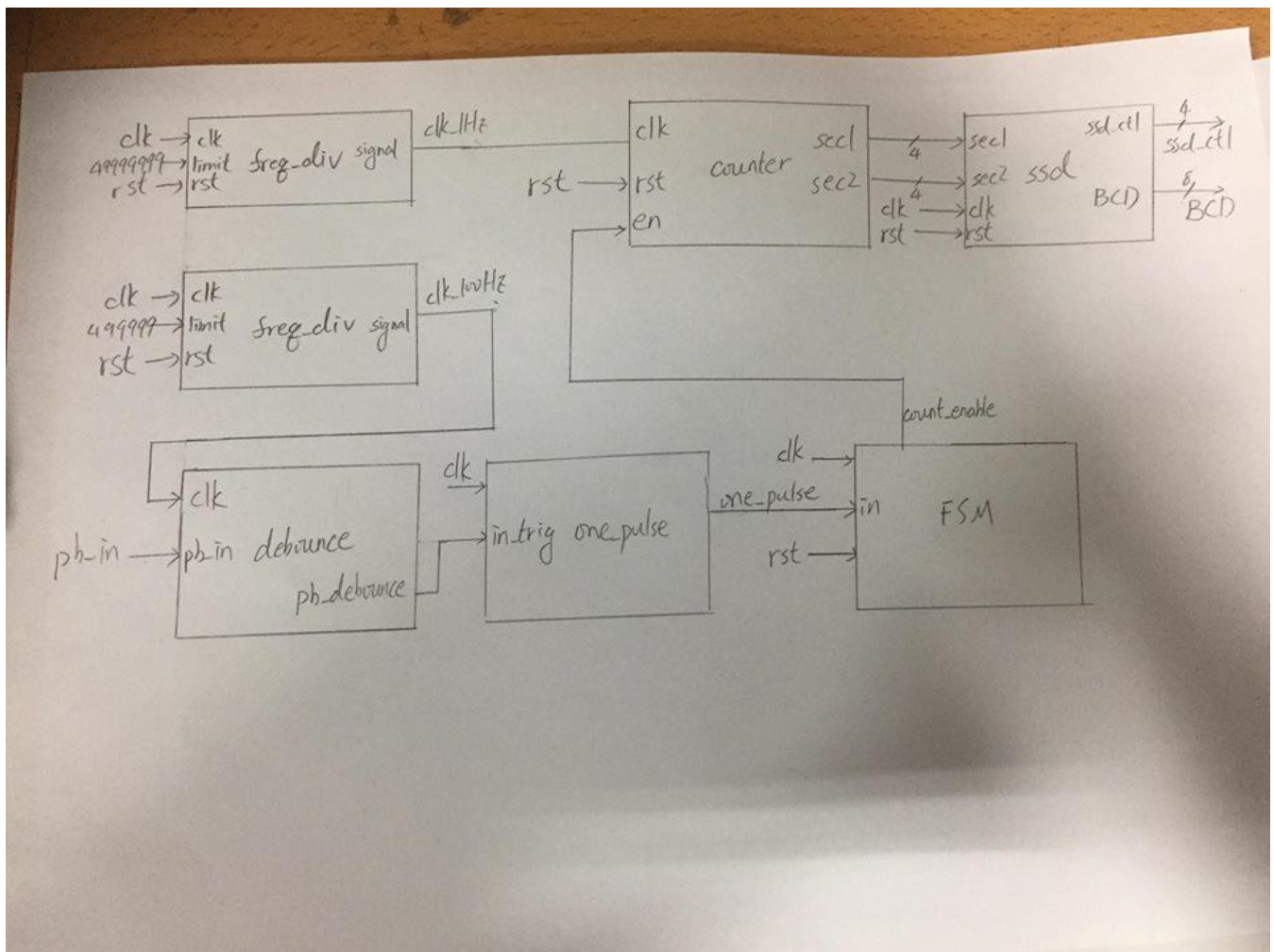
[3:0] ssd_ctl

[15:0] LED: 倒數到 0 的時候全部變 0

(2) Block Diagram



Design Implementation



Logic Diagram

每個 module 的功用如下：

Freq_div：普通的除頻器。藉由給予不同的上限值來產出不同頻率的 clock。在這個 lab 中，產生 Counter 用的 1Hz 的 clock 和 debounce 要用的 100MHz 的 clock。

Debounce：老師給的範例程式，負責把按下去的雜訊處理乾淨。

One_pulse：老師給的範例程式，負責產生脈衝丟給 FSM

FSM：只有兩種 state：倒數與暫停。轉換的依據是經由按壓按鈕的訊號經以上兩個 module 處理後所產生的脈衝。

Counter：非常普通的 30 秒下數器，但有把 FSM 的 state 接進來達成按一次按鈕以暫停的功能。

Ssd：負責把剩餘秒數顯示在七段顯示器上。

3. The same function as 2. Instead of using two push buttons

Design Specification

(1) I/O ports

Input:

Clk: global clock

pb_in: 接在另一個 button 上，負責暫停。

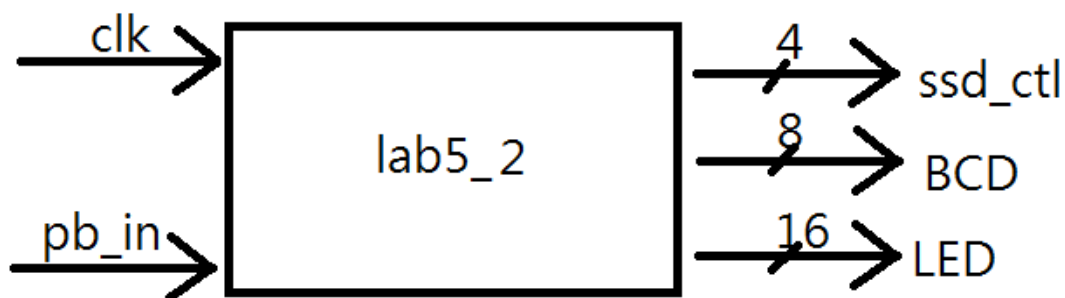
output

[7:0] BCD

[3:0] ssd_ctl

[15:0] LED: 倒數到 0 的時候全部變 0

(2) Block Diagram



Design Implementation

這一題大抵與上一題相同，最大的相異之處在於把 input 中的 reset 拿掉了。每個 module 的功用如下：

Freq_div：普通的除頻器。藉由給予不同的上限值來產出不同頻率的 clock。在這個 lab 中，產生 Counter 用的 1Hz 的 clock 和 debounce 要用的 100MHz 的 clock。

Debounce：老師給的範例程式，負責把按下去的雜訊處理乾淨。

One_pulse：老師給的範例程式，負責產生脈衝丟給 FSM

FSM：只有兩種 state：倒數與暫停。轉換的依據是經由按壓按鈕的訊號經以上兩個 module 處理後所產生的脈衝。

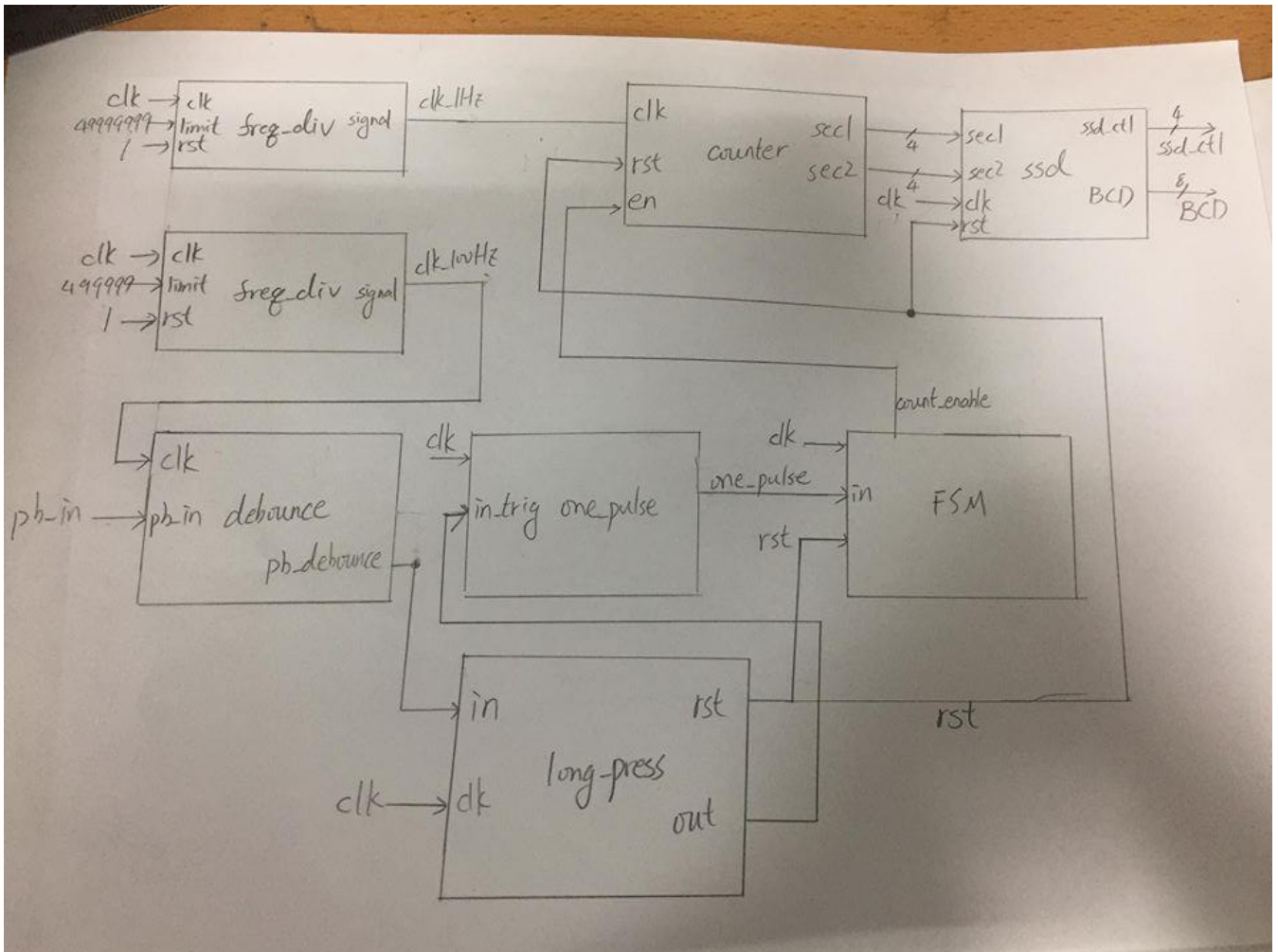
Counter：非常普通的 30 秒下數器，但有把 FSM 的 state 接進來達成按一次按鈕以暫停的功

能。

Ssd：負責把剩餘秒數顯示在七段顯示器上。

Long_press：reset 的產生器，同時也是一個上數器。Reset 的預設值是 0，只要經過 debounce 的信號為 1，上數器就會一直上數，到一定值後就會把 reset 變成 1。

Logic Diagram:



Discussion

寫 prelab 時遇到很多問題。首先，counter 完全不倒數，後來發現原因是若倒數結束時把全部的暫態令成 0 要寫在 FSM 傳出來的 freeze 信號控制的 if 條件句下，而不能額外寫成一個組合邏輯。另外就是對 top module 的概念不熟悉導致配線錯誤，致使 counter 完全不聽 freeze 信號的指揮而暫停。以後再遇到這種需要寫很多 module 的電路，應該要把每個 module 獨立寫完，並寫一個 top module 做組裝。組裝時應注意要把所有 input output 都確實接到正確的信號，不能有所偏差。

後來嘗試寫了這個 lab 寫了兩個禮拜，我就對關於 push_button 的一切徹底絕望了。

直到兩個月後的今天，我，變強了。於是我把之前寫的東西重新接過一遍，結果發現只要我把 reset 改成正緣觸發，一切就沒事了，原來如此。

Conclusion

從禮拜一弄到禮拜四 QQ

寫電路的順序：先把各個獨立的 module 寫好，再從 top module 把所有 module 組裝成一個電路。

不只 key board，原來 push button 也要用正緣觸發的 reset，難怪學長整個學期的 lab 都是 posedge reset。

References

《Verilog 硬體描述語言數位電路》，鄭信源，儒林圖書公司(2016)。
另外也要感謝熱心助人的李佳璋同學、陳永憲同學、丁文淵同學。