

Lab5 report

107061218 謝霖泳

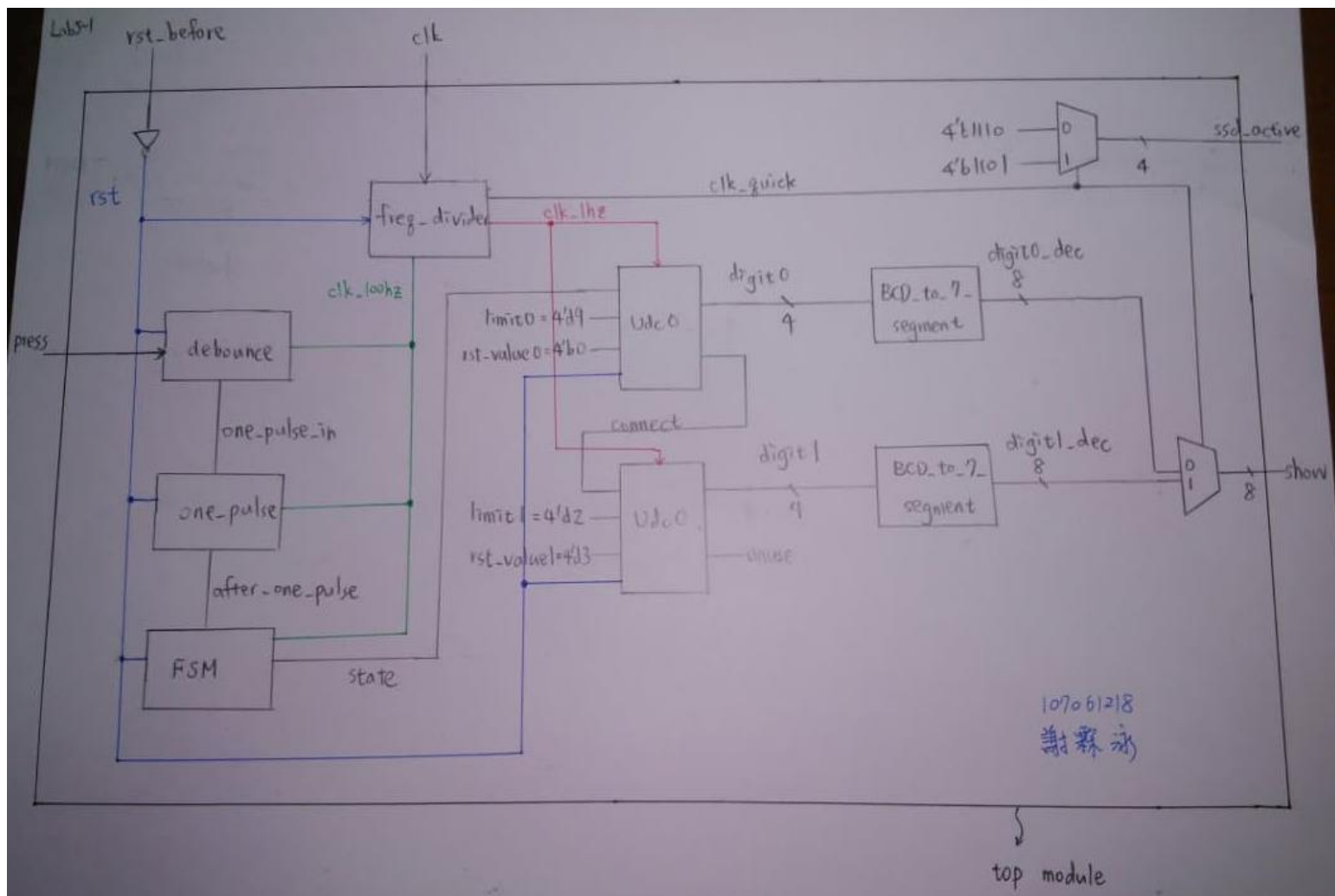
1.

➤ Design Specification

(1) top module

Input: clk, rst_before, press

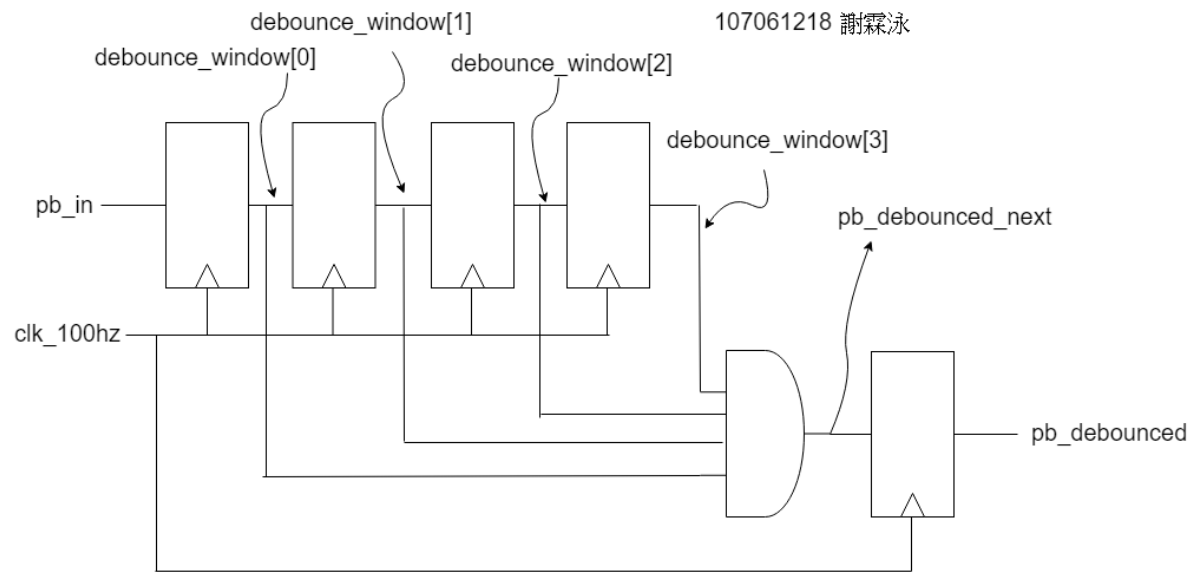
Output: [3:0] ssd_active, [7:0] show



(2) debounce

Input: clk_100hz, rst, pb_in

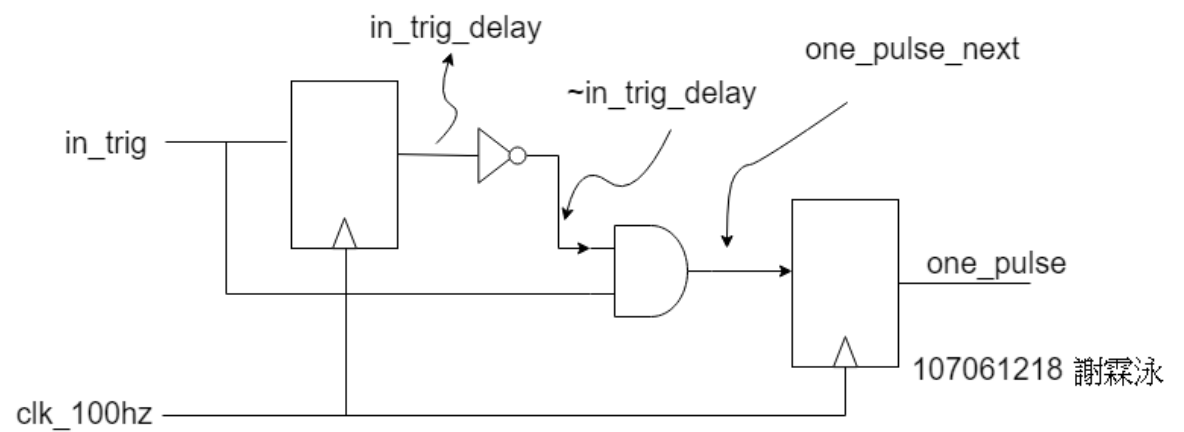
Output: pb_debounced



(3) one_pulse

Input: clk_100hz, rst, in_trig

Output: out_pulse

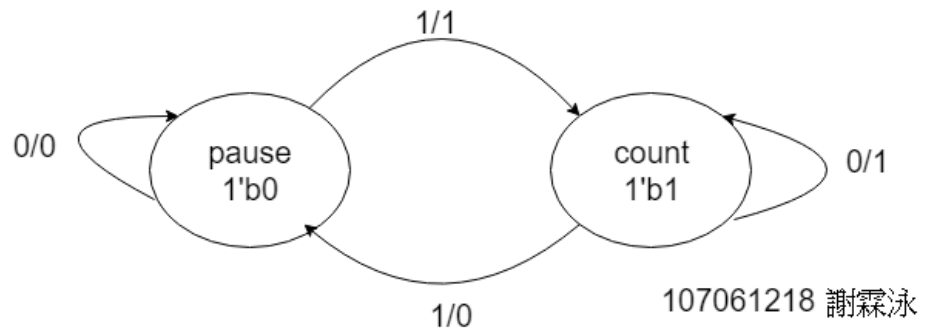


(4) FSM

Input: clk_100hz, rst, press_processed

Output: state

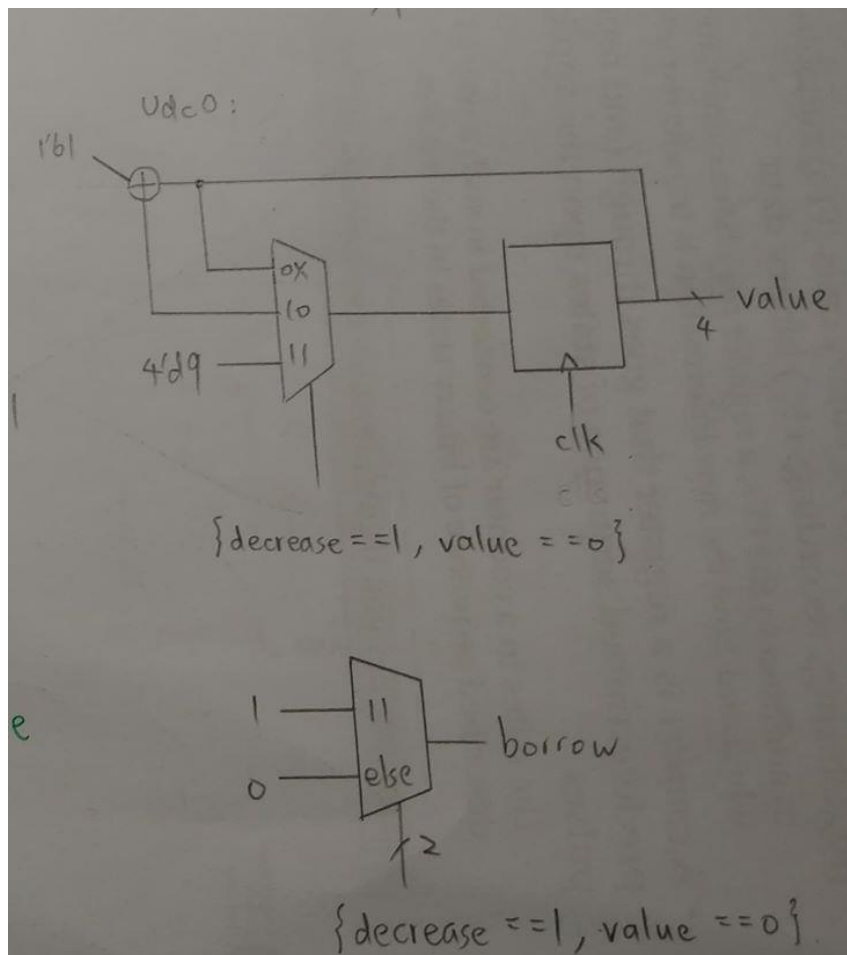
input: press_processed
output: state



(5) Udc0

Input: clk_1hz, rst, decrease, [3:0] rst_value, [3:0] limit

Output: [3:0] value, borrow



(6) BCD_to_7_segment

Input: [3:0] i

Output: [7:0] D_ssd

(7) frequency_divider

Input: clk, rst

Output: clk_out_1hz, clk_out_100hz, clk_for_ssd

➤ Design Implementation

首先，先介紹輸入的部分，設置兩個 button，分別為控制開始/暫停與控制重製的按鈕。控制開始/暫停的那個按鈕為 input press，控制重製的按鈕為 input rst_before。因為一般而言，我們習慣 reset 為 low active，但我們按下按鈕想重製時的時候訊號為 1'b1，沒按時為 1'b0，與我們習慣的模式相反。因此，我將按下按鈕後的 reset 訊號命名為 rst_before，經過一個 inverter 後的訊號才把它稱為 rst 訊號，這樣才不會悖於我們以往對於 reset 的認知。

至於除頻器與 BCD to seven segment decoder 的部分，因為在之前的 lab 已出現過非常多次，因此不再贅述。唯一值得注意的一點是，這裡的除頻器總共輸出三種信號，分別為 1Hz、100Hz 與一變動快速的時脈。1Hz 是給 2-bit synchronous decimal counter 用的，因為每過 1 秒，就要往下數 1；100Hz 的頻率是給 debounce、one pulse 與 FSM 用的，因為這樣才能達到除去雜訊與 one pulse 的功能；至於剩下那個快速變動的時脈，是給七段顯示器用的，要快速地變換顯示的位置與顯示內容，才能達成視覺暫留的效果。因為這次只需要顯示兩位數，因此，只需要 1-bit 的快速變動時脈。

Debounce 的部分，將按鈕得到的原始訊號傳入由 4 個 FF 組成的 shift register 中，並將這 4-bit 的訊號 and 起來，當按鈕訊號產生 1、0 之間快速且不穩定變化的雜訊時，經過 and-gate 後，這段不穩定的訊號就會變成 0，便得到一個乾淨的訊號，達成 debounce 的效果。

➤ Discussion

為了驗證我的設計的正確性，我將最左邊的 LED 燈設為我的 state，換句話說，當我按下控制繼續/暫停的按鈕開始數時，此時狀態為「下數」，所以該 LED 燈亮；當我再次按下該按鈕，數字便會停留不變，此時狀態為「暫停」，故該 LED 燈不亮。

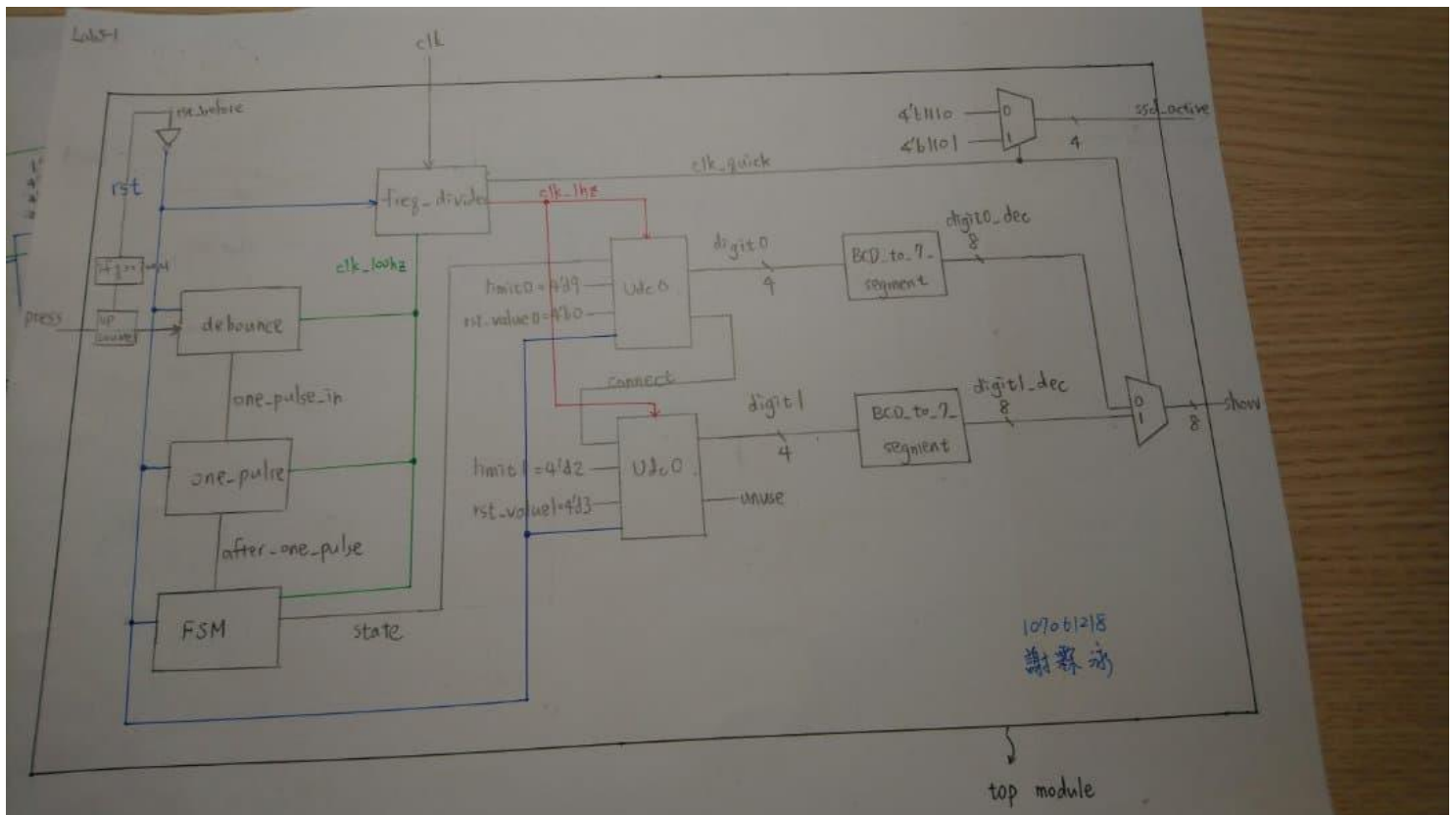
此外，我還多設計了一個功能，就是當倒數到 0 時，所有板子上的 LED 燈都會同時亮起，真的很漂亮。

2.

➤ Design Specification

Input: clk, press

Output: [3:0] ssd_active, [7:0] show



➤ Design Implementation

這一題與第一題大同小異，其差別僅在於此題只能用 1 個 button。因為這個 button 同時要有暫停/繼續與 reset 的功能，因此用長按短按來區分之。

為了區分長按與短按，在 press 訊號進來之後便先經過一個 counter，而 press 訊號也就作為這個 counter 的 enable。當我按下按鈕後，會先執行暫停/繼續的功能，於此同時也會開始計時，這個 counter 的 clock 為晶體震動的 100MHz clock，因此，當我的 q 值數到 300M 時，便會啟動 reset 功能，也就是我現在是「長按」；反之，若未達 300M，代表我按的時間不到 3 秒，所以要執行短按的功能，也就是繼續/暫停。

其餘 one pulse、debounce.....等均與上題相同，於此不再贅述。

➤ Discussion

這次 lab 真的變得複雜許多，讓我真的體會到了一個 top module 裡面包著非常多小 module 的感覺。這次也花費了我更多的時間，主要是因為是第一次使用 button，對於 debounce、one pulse.....等功能尚不夠熟稔，之後

應該會愈來愈好，漸入佳境。

我也體會到，若分成許多小 module 有利於我們 debug，因為哪一塊功能錯誤就改那個 module 就可以了；反之，若全部混在一起可能造成 debug 困難，因為看不出到底是什麼功能產生錯誤。