105061254 林士平 邏輯設計實驗報告 Lab5

1.

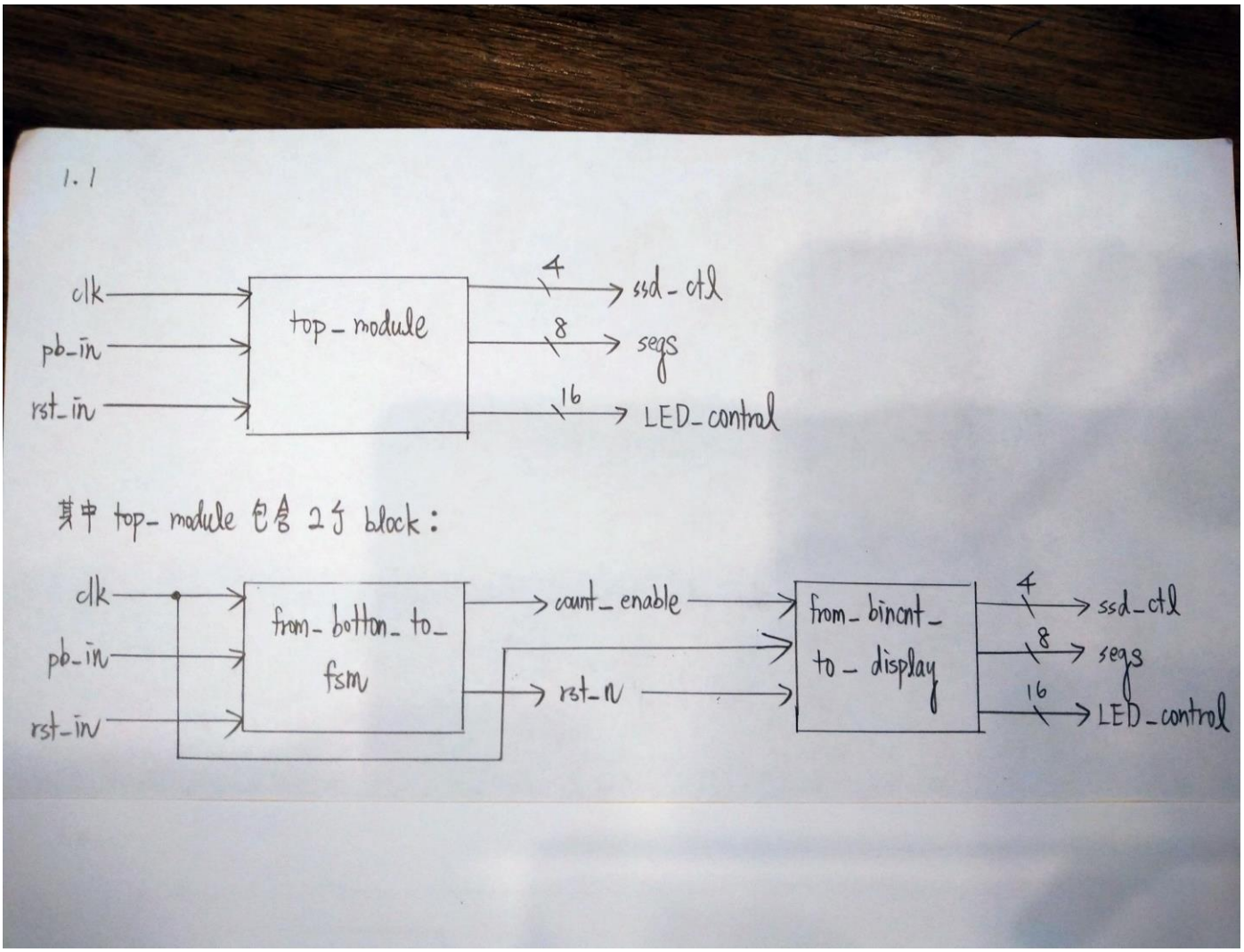
(1) Design specification :

A. Inputs and outputs(表一)：

Inputs	clk, rst_in, pb_in
Outputs	ssd_ctl[3:0], segs[7:0], LED_control[15:0]

↑表一：Inputs and outputs of 1

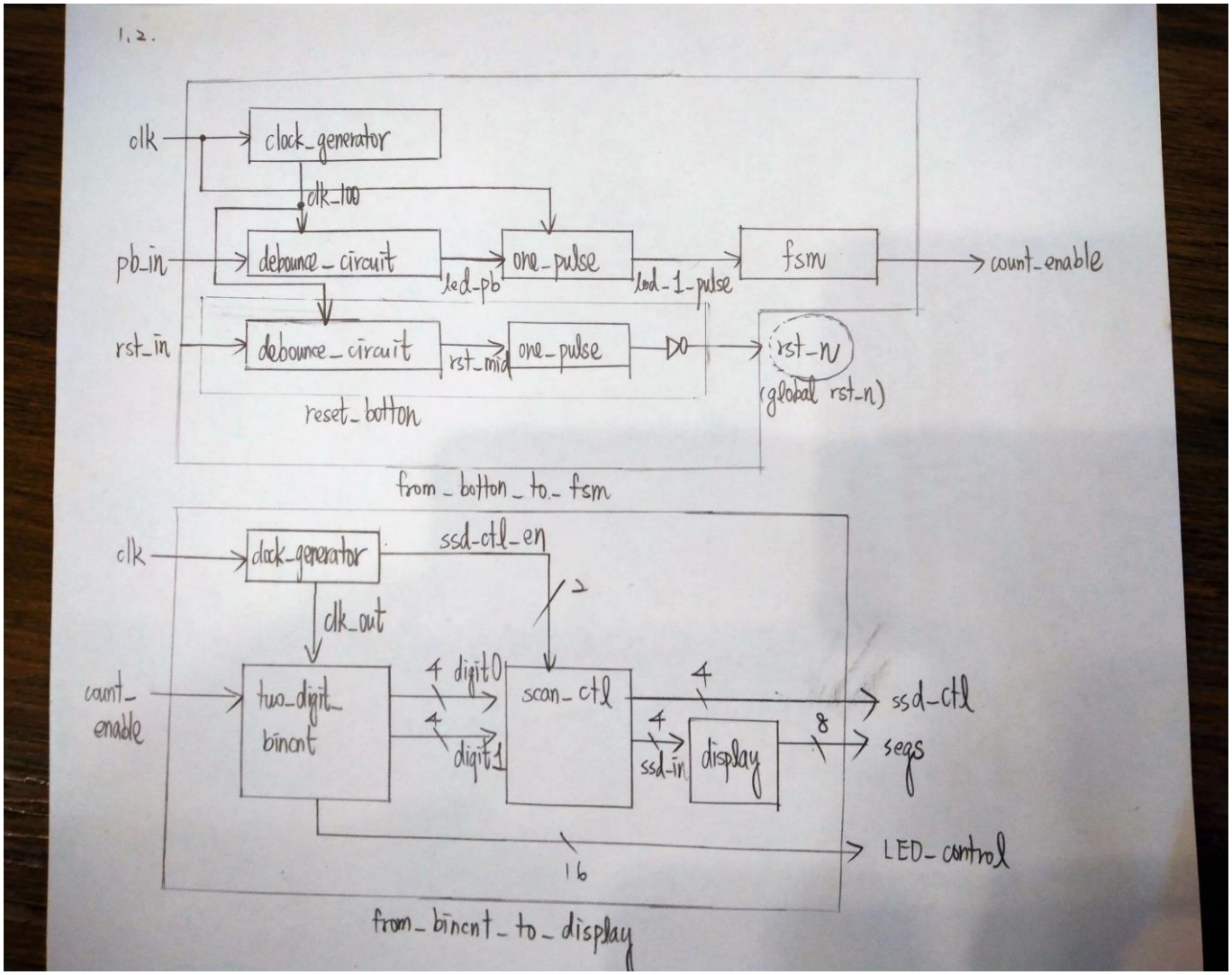
B. Block diagram(function table)(圖一)：



↑圖一：The block diagram of 1

(2) Design implementation :

A. Logic diagram(function table)(圖二)：



↑圖二：logic diagram of 1

B. I/O pin assignment(表二)：

I/O	ssd_ctl[3]	ssd_ctl[2]	ssd_ctl[1]	ssd_ctl[0]	segs[7]	segs[6]	segs[5]
LOC	W4	V4	U4	U2	W7	W6	U8
I/O	segs[4]	segs[3]	segs[2]	segs[1]	segs[0]	LED_ control[15]	LED_ control[14]
LOC	V8	U5	V5	U7	V7	L1	P1
I/O	LED_ control[13]	LED_ control[12]	LED_ control[11]	LED_ control[10]	LED_ control[9]	LED_ control[8]	LED_ control[7]
LOC	N3	P3	U3	W3	V3	V13	V14
I/O	LED_ control[6]	LED_ control[5]	LED_ control[4]	LED_ control[3]	LED_ control[2]	LED_ control[1]	LED_ control[0]
LOC	U14	U15	W18	V19	U19	E19	U16
I/O	pb_in	rst_in	clk				
LOC	T17	U17	W5				

↑表二：I/O pin assignment of 1

C.功能與做法說明：

本題為利用兩個按鈕，在 FPGA 板上實現 30 秒倒數器。其中一個按鈕的功能是 pause/start，另一個按鈕的功能是 reset。我的設計包括三個大模組：top_module、from_botton_to_fsm、from_bincnt_to_display。

from_botton_to_fsm 為從按鈕到 finite state machine 的設計，裡面的功能包括按鈕的 debounce 和 one_pulse，並且將 one_pulse 的結果送到 fsm 做為 state 轉換的控制，fsm 共兩個 state：start 和 pause。在 start state 的時候 count_enable = 1；在 pause state 的時候 count_enable = 0，count_enable 用來決定倒數器要 pause 還是 start。其中有個 reset_botton 模組是專門處理 reset 的事情，包含了 reset 按鈕的 debounce 和 one_pulse，而且 one_pulse 完後要 toggle(因為是 negative edge reset)，處理完的 reset 訊號(rst_n)再輸入各個 module 作為 Flip-Flop 的 negative edge reset。

from_bincnt_to_display 則是從 binary counter 到最後 display 的設計，首先輸入從 fsm 得到的 count_enable 到 counter 決定 counter 是否下數，再將 counter 數出來的結果送到 scan_ctl 和 display 使得七段顯示器得以顯示。其中 two_digit_bnt 的設計非常接近 Lab4 的 bonus，只是在第一個 counter(Udc0)的 decrease_enable 必須加上這樣一段敘述：

```
assign decrease_enable = en & &(~((digit0 == 4'b0) && (digit1 == 4'b0)));
```

表示倒數器在 enable = 1 且還沒數到 00 時才會下數。

最後 top_module 連接 from_botton_to_fsm 和 from_bincnt_to_display，並處理整個 30 秒倒數器的 input 和 output。

2.

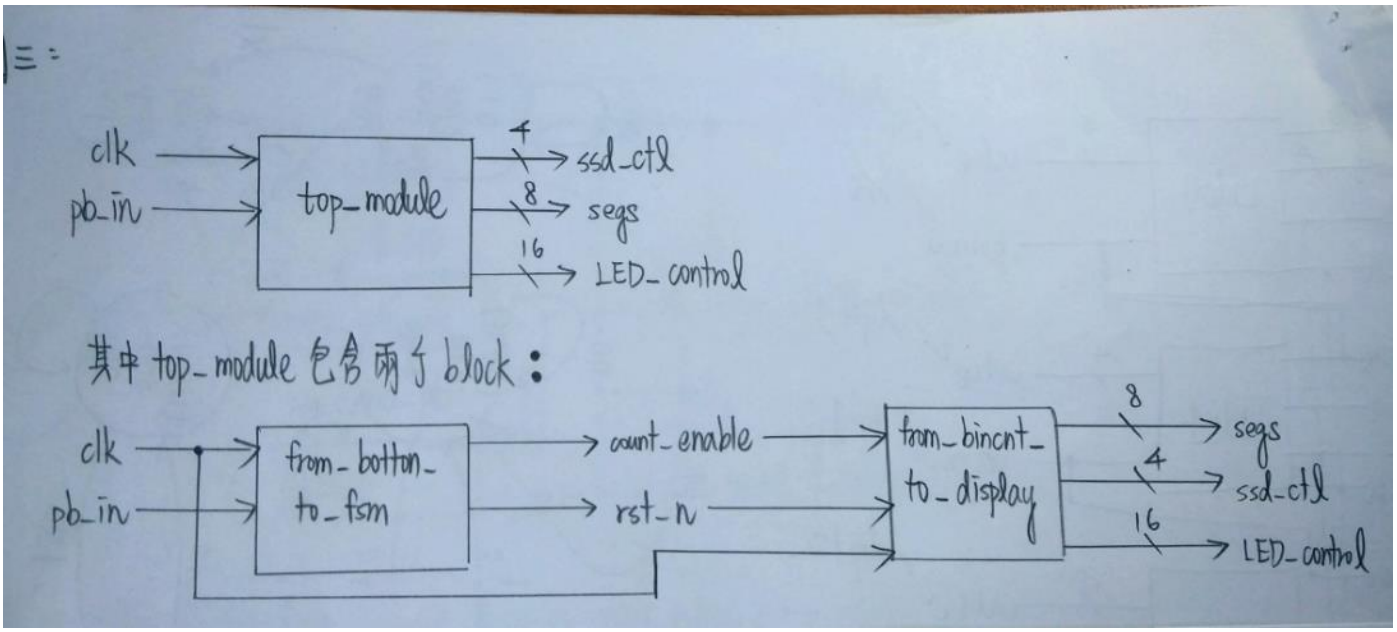
(1) Design specification :

A. Inputs and outputs(表三)：

Inputs	clk, pb_in
Outputs	ssd_ctl[3:0], segs[7:0], LED_control[15:0]

↑表三：Inputs and outputs of 2

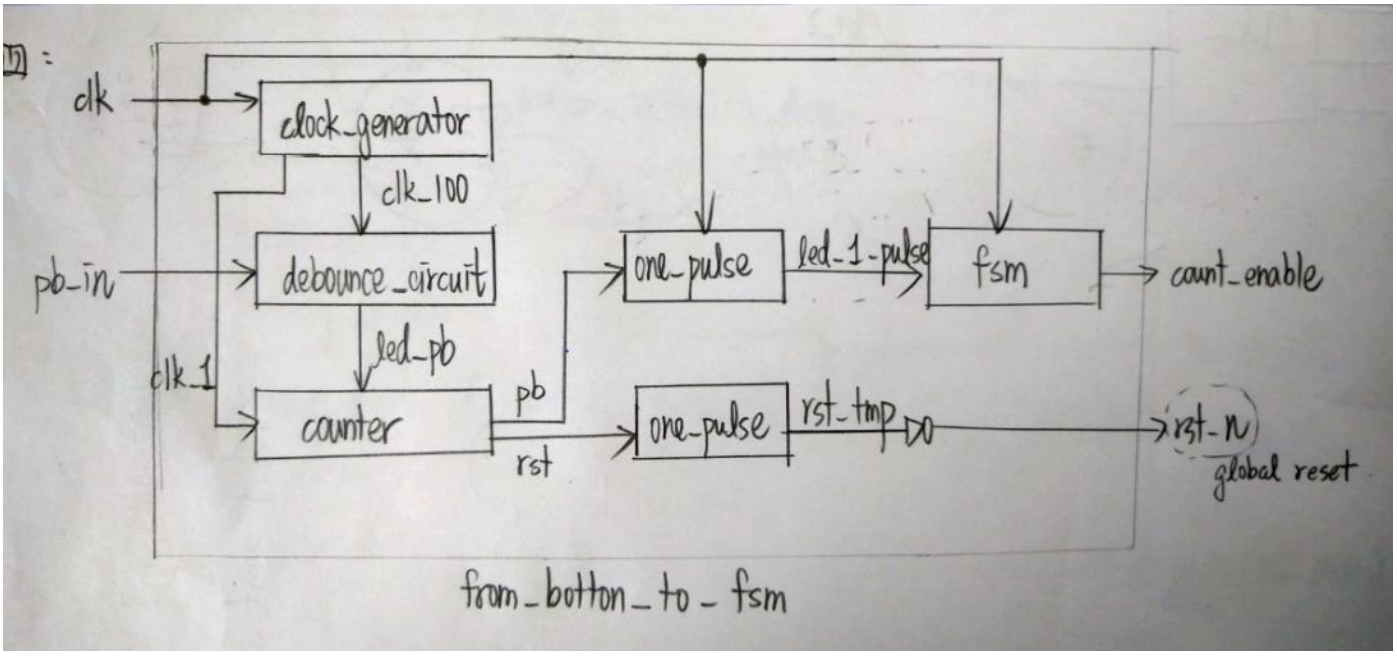
B. Block diagram(function table)(圖三)：



↑圖三：The block diagram of 2

(2) Design implementation :

A. Logic diagram(function table)(圖四)：



↑圖四：logic diagram of 2(本題的 from_bincnt_to_display 和第一題相同(見圖二)，故只畫 from_botton_to_fsm 的部分)

B. I/O pin assignment(表四)：

I/O	ssd_ctl[3]	ssd_ctl[2]	ssd_ctl[1]	ssd_ctl[0]	segs[7]	segs[6]	segs[5]
LOC	W4	V4	U4	U2	W7	W6	U8
I/O	segs[4]	segs[3]	segs[2]	segs[1]	segs[0]	LED_ control[15]	LED_ control[14]
LOC	V8	U5	V5	U7	V7	L1	P1
I/O	LED_ control[13]	LED_ control[12]	LED_ control[11]	LED_ control[10]	LED_ control[9]	LED_ control[8]	LED_ control[7]
LOC	N3	P3	U3	W3	V3	V13	V14
I/O	LED_ control[6]	LED_ control[5]	LED_ control[4]	LED_ control[3]	LED_ control[2]	LED_ control[1]	LED_ control[0]
LOC	U14	U15	W18	V19	U19	E19	U16
I/O	pb_in	clk					
LOC	T17	W5					

↑表四：I/O pin assignment of 2

C.功能與做法說明：

本題和前一題類似，也是要在 FPGA 板上實現 30 秒倒數器，差別是只能用一個按鈕。為了要能夠區分出長按或短按，我加入了一個 counter 模組：當按下按鈕之後它便開始計數，放開的話就會歸零，如果按下按鈕的時間超過一定的數值，就會輸出 rst 訊號，rst 訊號經過 one_pulse 後再 toggle 會成為 rst_n 訊號，並對全部的 Flip-Flop reset；如果按下按鈕的時間不夠長，則輸出 pb 訊號，經過 one_pulse 輸入 fsm 做 state 的轉換。其餘部分和第一題相同。

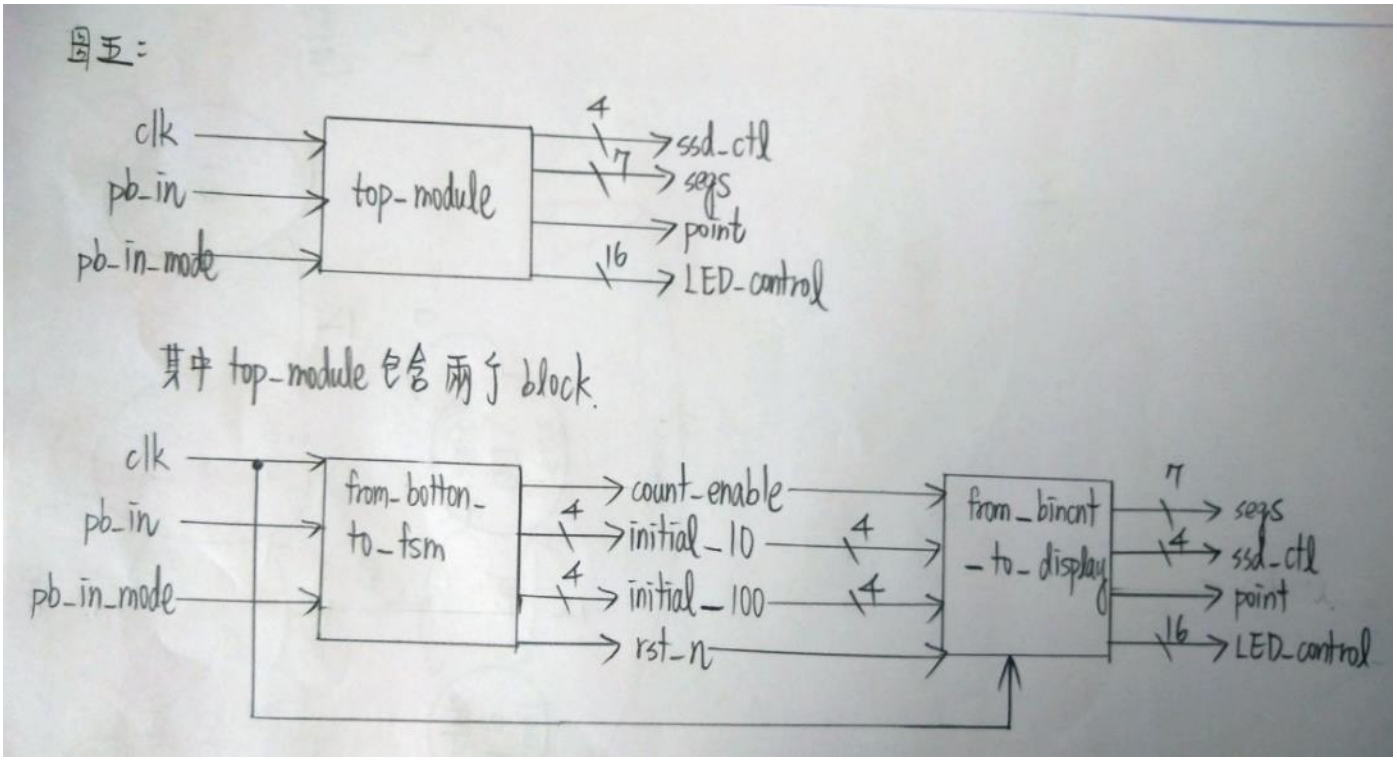
3.

(1) Design specification :

A. Inputs and outputs(表五)：

Inputs	clk, pb_in, pb_in_mode
Outputs	ssd_ctl[3:0], segs[7:0], LED_control[15:0], point
↑表五：Inputs and outputs of 3	

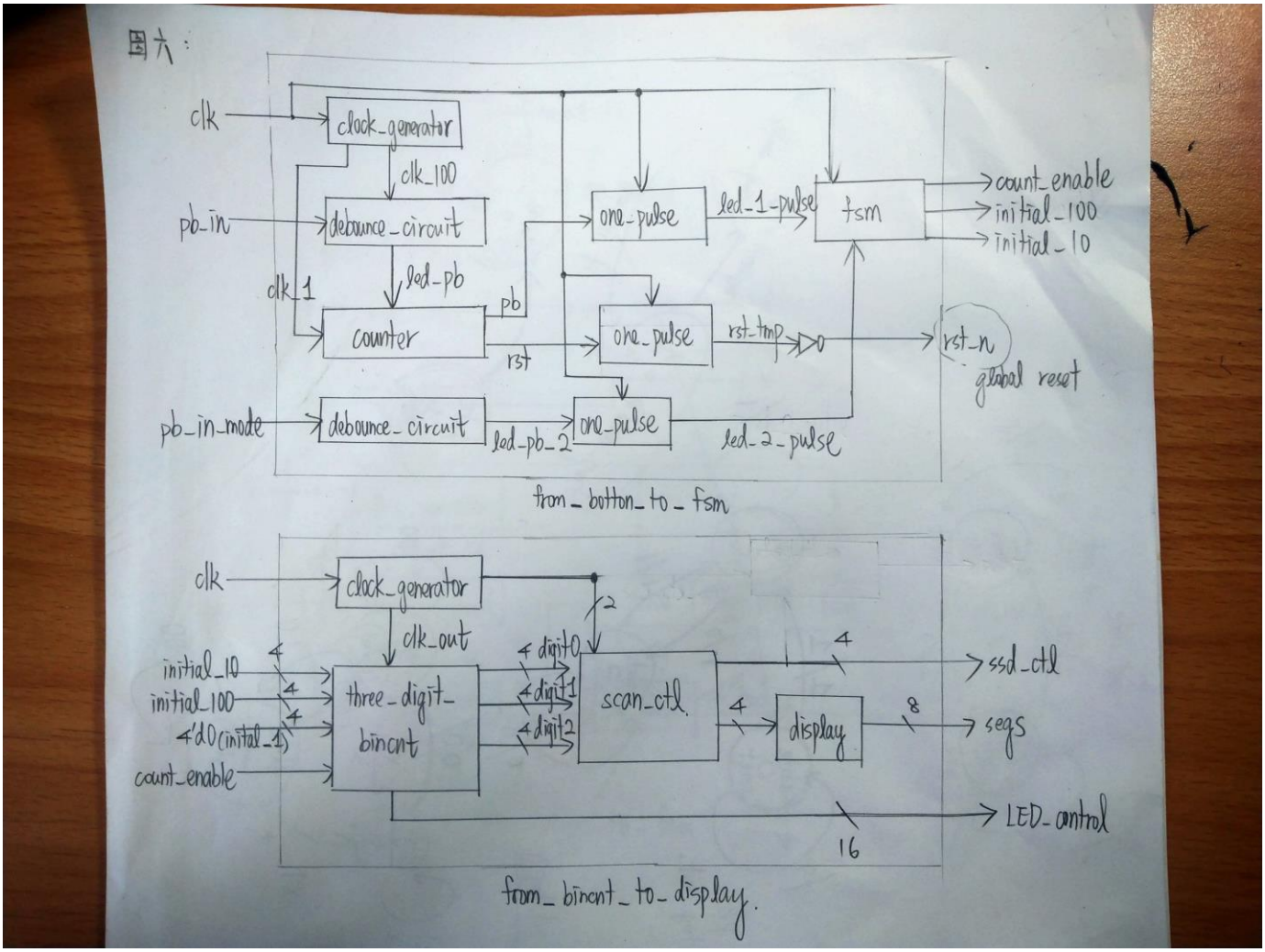
B. Block diagram(function table)(圖五)：



↑圖五：The block diagram of 3

(2) Design implementation :

A. Logic diagram(function table)(圖六)：



↑圖六：logic diagram of 3

B. I/O pin assignment(表六)：

I/O	ssd_ctl[3]	ssd_ctl[2]	ssd_ctl[1]	ssd_ctl[0]	segs[7]	segs[6]	segs[5]
LOC	W4	V4	U4	U2	W7	W6	U8
I/O	segs[4]	segs[3]	segs[2]	segs[1]	segs[0]	LED_ control[15]	LED_ control[14]
LOC	V8	U5	V5	U7	V7	L1	P1
I/O	LED_ control[13]	LED_ control[12]	LED_ control[11]	LED_ control[10]	LED_ control[9]	LED_ control[8]	LED_ control[7]
LOC	N3	P3	U3	W3	V3	V13	V14
I/O	LED_ control[6]	LED_ control[5]	LED_ control[4]	LED_ control[3]	LED_ control[2]	LED_ control[1]	LED_ control[0]
LOC	U14	U15	W18	V19	U19	E19	U16
I/O	pb_in	pb_in _mode	point	clk			
LOC	T17	U17	V7	W5			

↑表六：I/O pin assignment of 3

C.功能與做法說明：

本題應用到第二題判斷長按短按的方法，並且多加入一個按鈕作為 mode select，完成一個 30/1:00 倒數器。判斷長按短按的方式如同第二題所述，和前面不同的是 mode select 的部分。

我的方式是：mode 不同會改變 three_digit_bincnt 的 initial 值，因此當 reset 之後，七段顯示器便會顯示那個 mode 的 initial 值，便達到改變 mode 的效果。這個設計的重點是在 finite state machine，我用兩個 fsm 來完成：第一個 fsm 有三個 state， 一個是 start、一個是 pause，另外我多加入了一個 reset state 來作為 reset 後進入的 state；第二個 fsm 有兩個 state，一個是 30，另一個是 1:00，功能是 mode select，當進入不同的 state 會改變 three_digit_bincnt 的 initial 值。

5. Discussion

我認為這次的實驗重點有兩個，一個是 push botton，另一個是 finite state machine，兩個都花了我相當多時間了解與熟悉，所以這個 Lab 花了我非常多的時間。

首先是 push botton，並不是所有的 push botton 都需要 one pulse，one pulse 的功能在預報中我有提到，是為了要讓 pulse 可以切齊 clock edge，讓 fsm 的 state transition 不會出問題，所以如果 push botton 和 fsm 無關，其實可以不用 one pulse 的，例如：rst_n；另外，判斷長按短按的方法也花了我相當多時間，最後我採用 counter 的方式來判斷，不過我也有聽到別的方法，例如：用 shift register、改變 one pulse 的 clk 等等，以後的 Lab 有機會可以試試。

第二個是 fsm，fsm 的 clk 花了我不少時間，後來發現 fsm 的 clk 和 one pulse 的一定要一樣，不然 state transition 就會有錯；然後是怎麼設計 fsm，大前提是要能夠把 state transition diagram 畫出來，只要畫出來 code 就一定可以打出來。

6. Conclusion

這次的 Lab 模組量暴增，所以打 code 的時候頭腦一定要很清楚，才不會亂掉，更重要的是一開始一定要先在紙上擬好草稿，並且把變數名稱寫清楚。