

lab5 Report

lab4_4

Design Specification

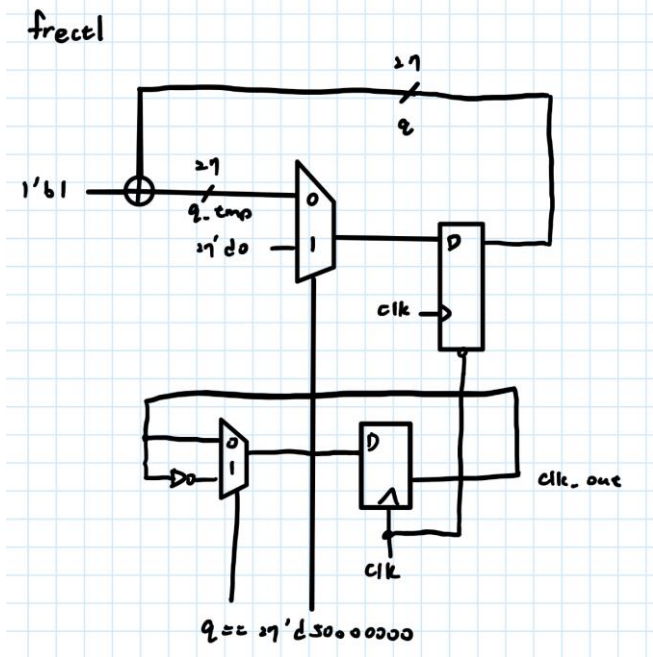
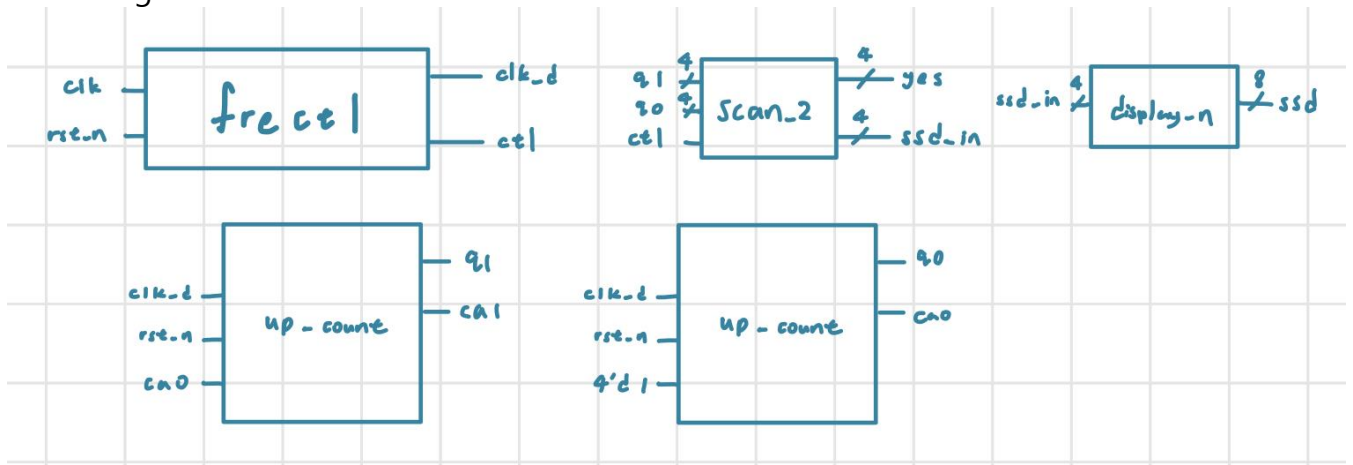
For a 2-digit BCD up counter display on the seven-segment display

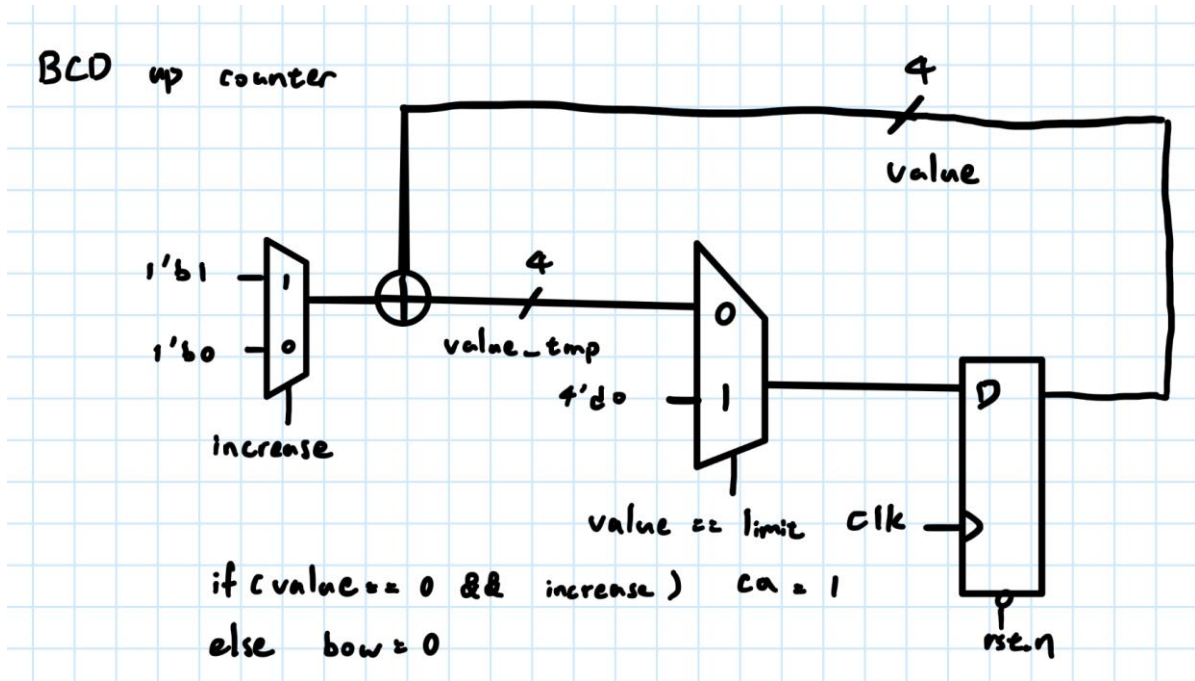
Input: clk, rst_n

Output: [7:0] ssd, [3:0] yes.

Design Implementation

Block diagram





I/O	fcystal	yes 3	yes 2	yes 1	yes 0	ssd7	ssd6	ssd5	ssd4	ssd3	ssd2	ssd1	ssd0	rst_n
Site	W5	V19	U19	E19	U16	V14	U14	U15	W18	V19	U19	E19	U16	R2

Discussion

這個題目其實就是利用類似 lab4_3 所做出的 BCD down counter 而此題則是修改為 up counter 並改成 2digit 將個位數的 carry 連接到十位數的 count_enable，也就是當個位數有 carry 時，十位數才會往上數一，就可以達到題目所要求之效果。而因為在 lab4_3 裡面因為題目不需用到太複雜的功能，我只寫了沒有停止功能，也沒有紀錄 borrow，所以在做這個子題的時候，多花了一些時間理解兩者之間的連接。

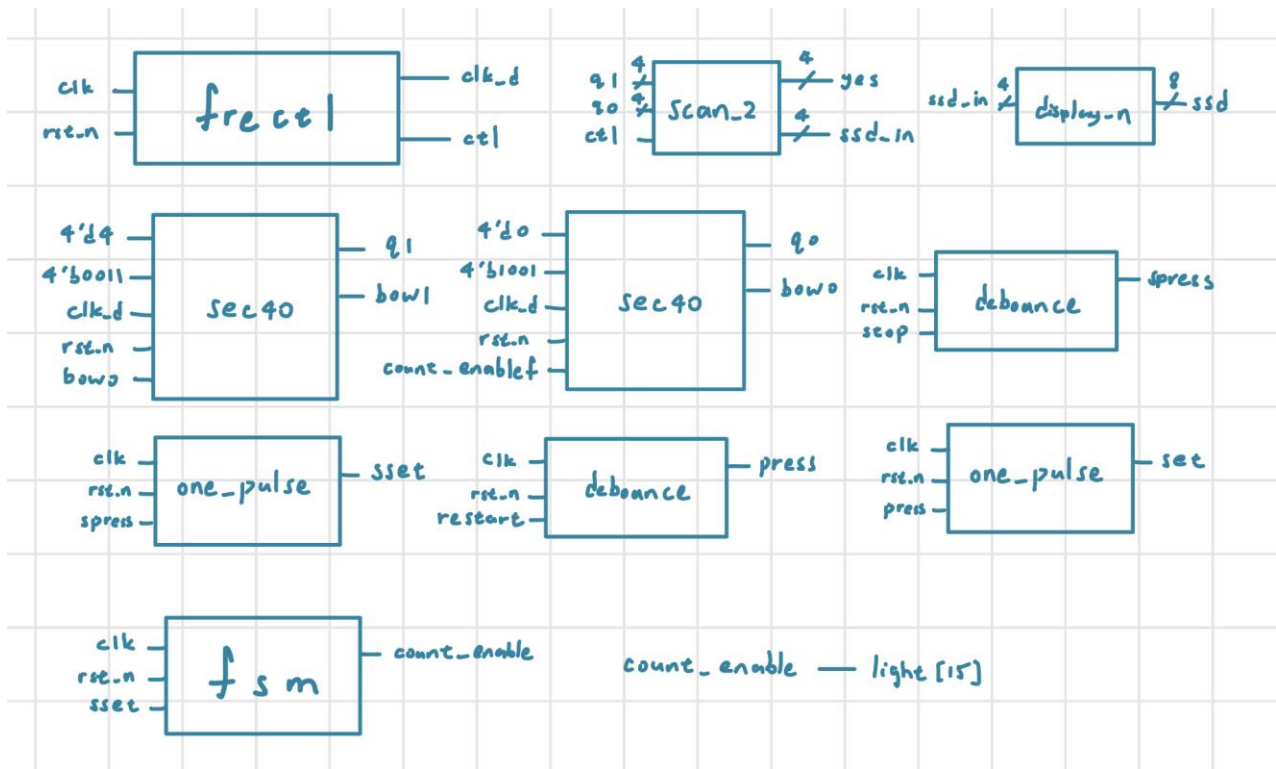
Lab5_1

Design Specification

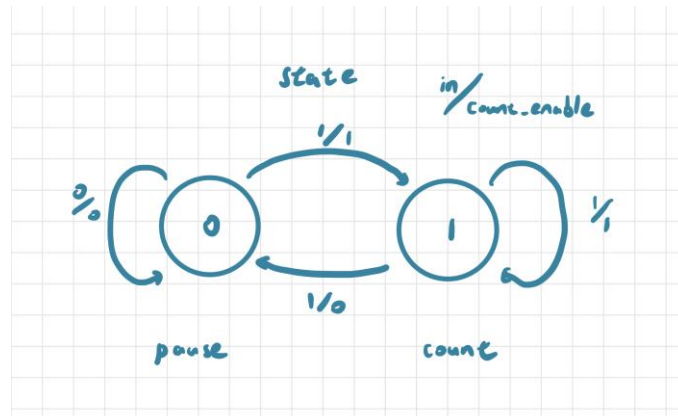
For a 40sec down counter
 Input: clk, rst_n, restart, stop
 Output: [7:0] ssd, [3:0] yes, [15:0] light

Design Implementation

Block diagram

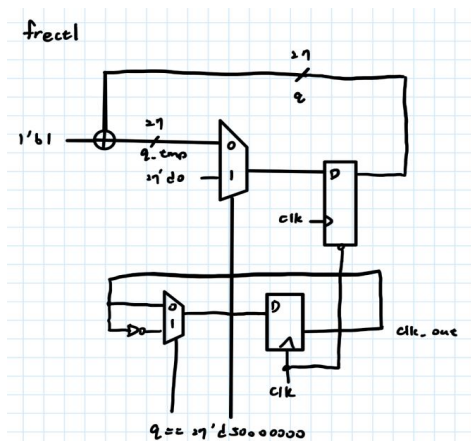


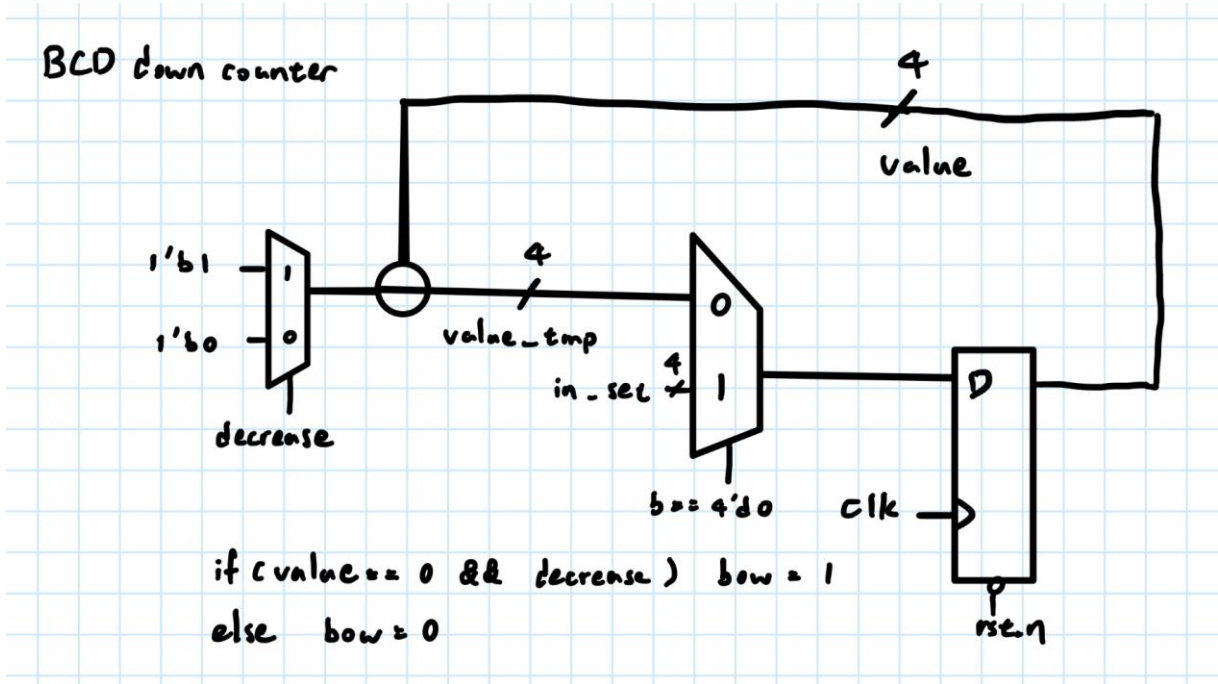
fsm



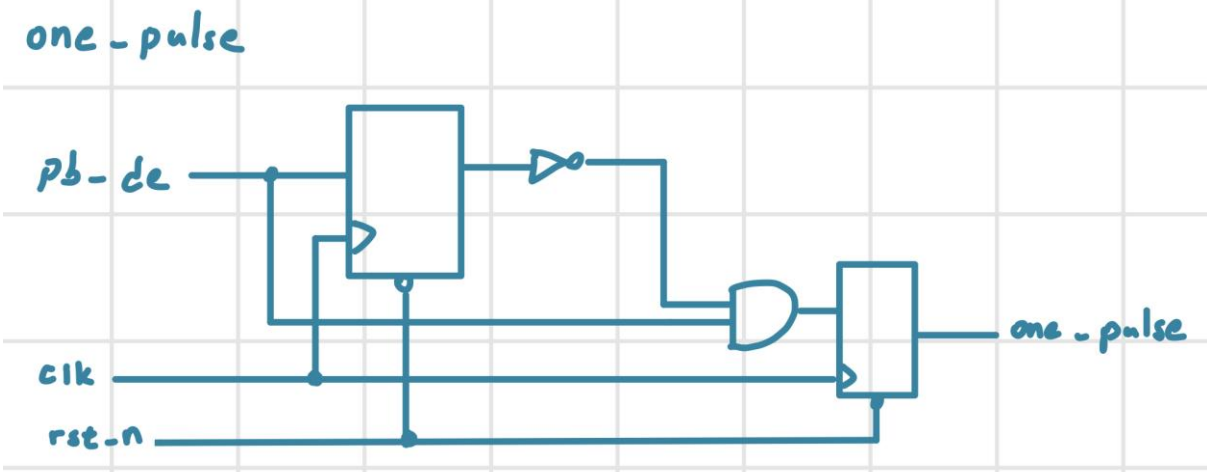
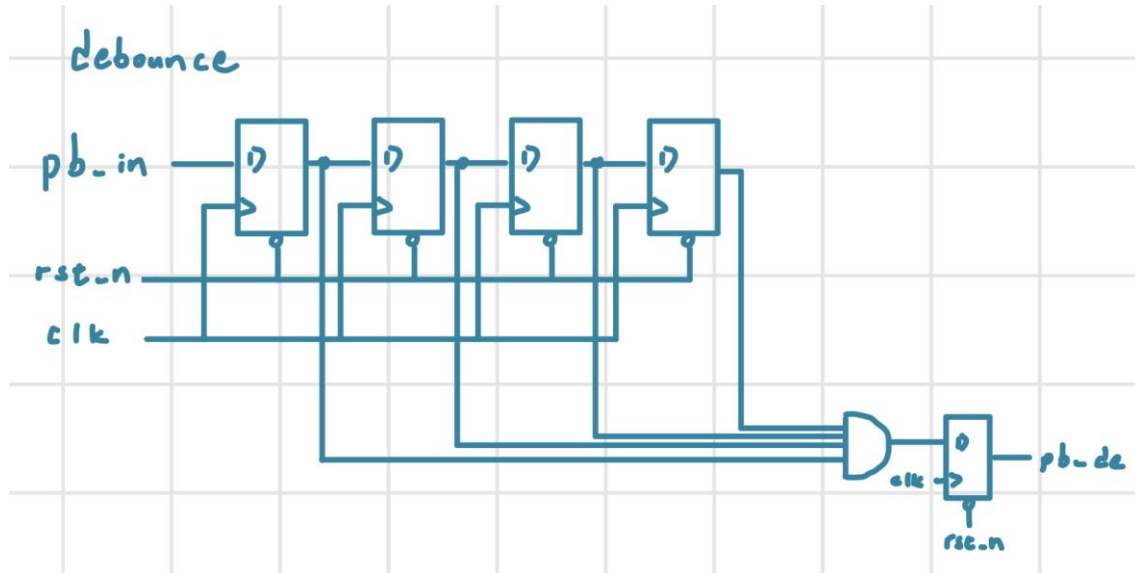
- If state = 0 (pause) and signal is 0, then keep the state.
- If state = 0 (pause) and signal is 1, then jump to next state (state = 1) (start).
- If state = 1 (start) and signal is 0, then keep the state.
- If state = 1 (start) and signal is 1, then jump to next state (state = 0) (pause)

frectl





display	i[3:0]				SSD[7:0]							
	i3	i2	i1	i0	A	B	C	D	E	F	G	H
0	0	0	0	0	0	0	0	0	0	0	1	1
1	0	0	0	1	1	0	0	1	1	1	1	1
2	0	0	1	0	0	0	1	0	0	1	0	1
3	0	0	1	1	0	0	0	0	1	1	0	1
4	0	1	0	0	1	0	0	1	1	0	0	1
5	0	1	0	1	0	1	0	0	1	0	0	1
6	0	1	1	0	0	1	0	0	0	0	0	1
7	0	1	1	1	0	0	0	1	1	1	1	1
8	1	0	0	0	0	0	0	0	0	0	0	1
9	1	0	0	1	0	0	0	0	1	0	0	1
A	1	0	1	0	0	0	0	1	0	0	0	1
B	1	0	1	1	1	1	0	0	0	0	0	1
C	1	1	0	0	0	1	1	0	0	0	1	1
D	1	1	0	1	1	0	0	0	0	1	0	1
E	1	1	1	0	0	1	1	0	0	0	0	1
F	1	1	1	1	0	1	1	0	0	0	0	1
G	1	1	1	1	0	1	1	1	0	0	0	1



I/O	fcystal	yes 3	yes 2	yes 1	yes 0	ssd7	ssd6	ssd5	ssd4	ssd3	ssd2	ssd1	ssd0	rst_n
Site	W5	V19	U19	E19	U16	V14	U14	U15	W18	V19	U19	E19	U16	R2

light [15]	light [14]	light [13]	light [12]	light [11]	light [10]	light [9]	light [8]
L1	P1	N3	P3	U3	W3	V3	V13

light [7]	light [6]	light [5]	light [4]	light [3]	light [2]	light [1]	light [0]
V14	U14	U15	W18	V19	U19	E19	U16

Discussion

本實驗要做出一個 40 秒的倒數計時器，就會直接想到 lab4_3 的 BCD down counter，仿照 lab4_4 的接線方式將十位數的初始值設為 4 個位數為 0，最困難的步驟應該是 Fsm 的處理，第一次使用上卡頓了一下，不過還好只有兩個 state 相對輕鬆，設定按下 stop 時，轉

換 state 使 count_enable 為 0 讓倒數計時暫停

Lab5_2

Design Specification

For a stopwatch

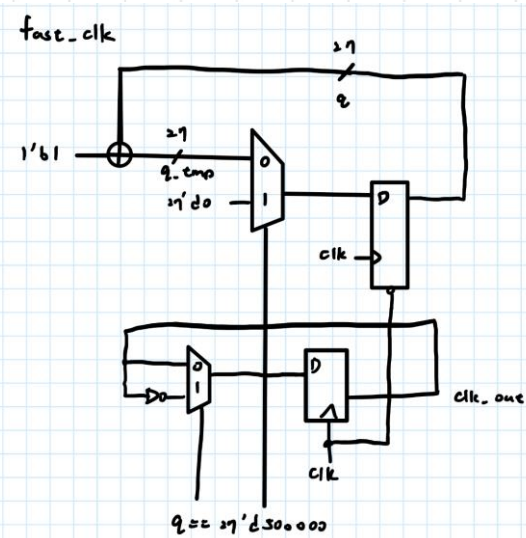
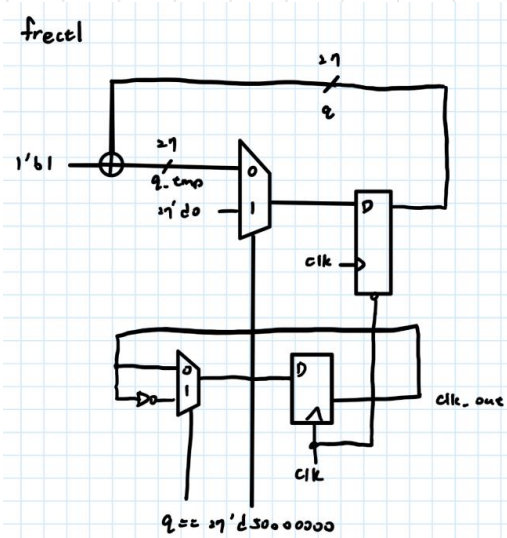
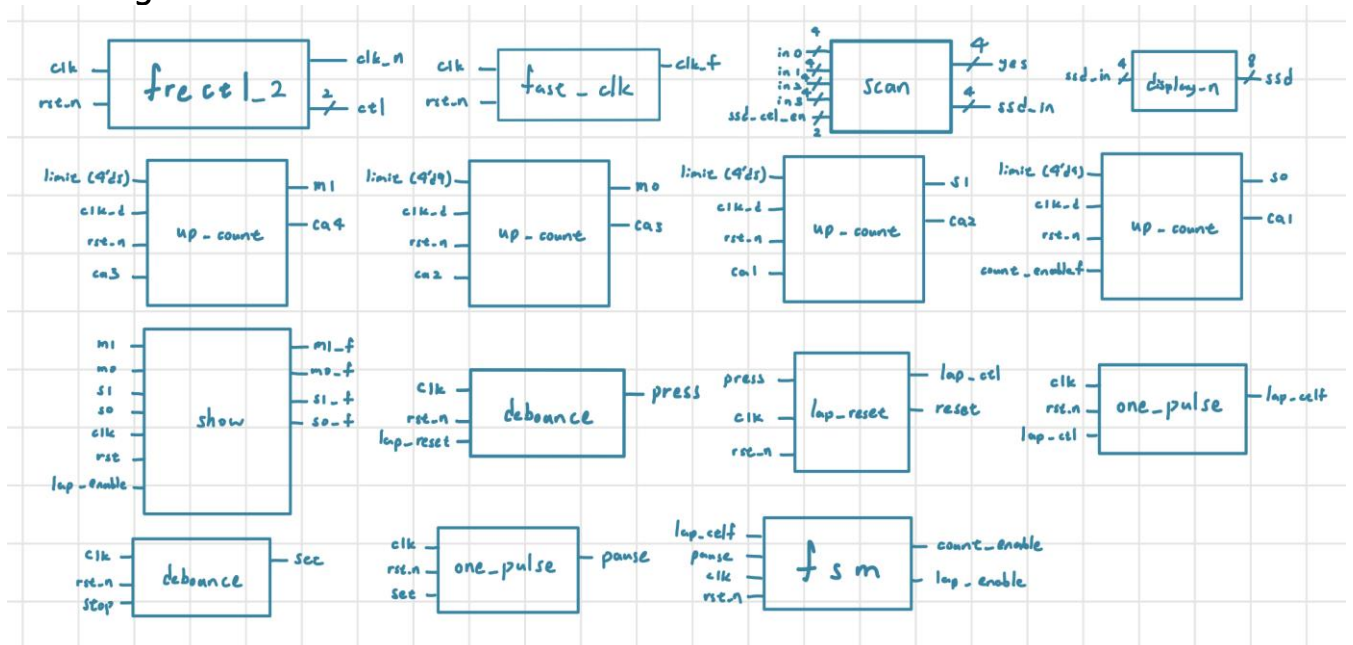
Input: clk, rst_n, lap_reset, stop, clk_ctl

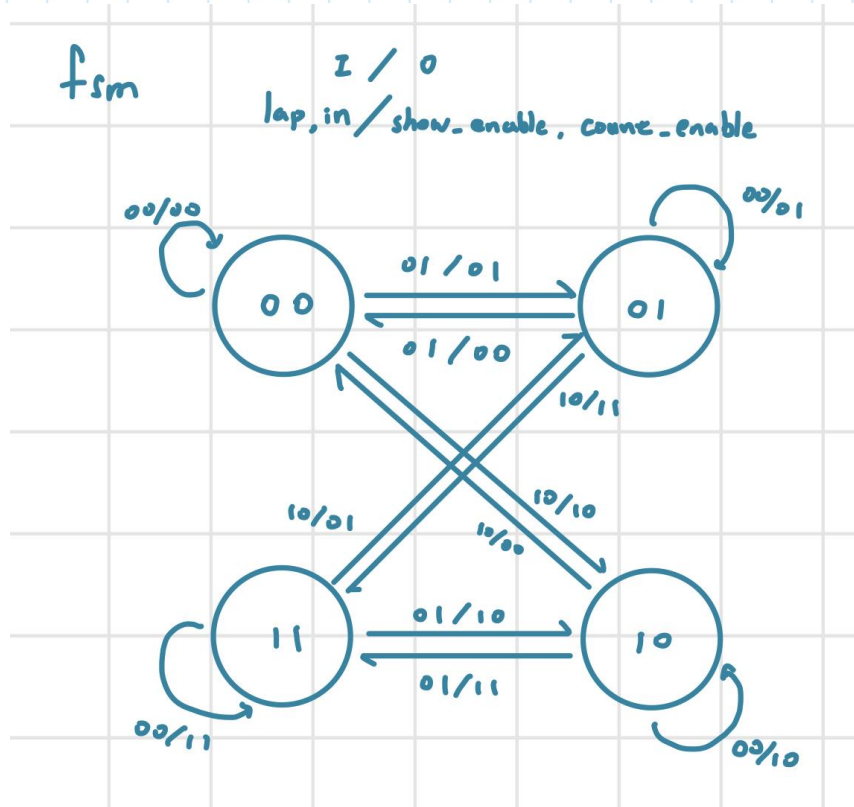
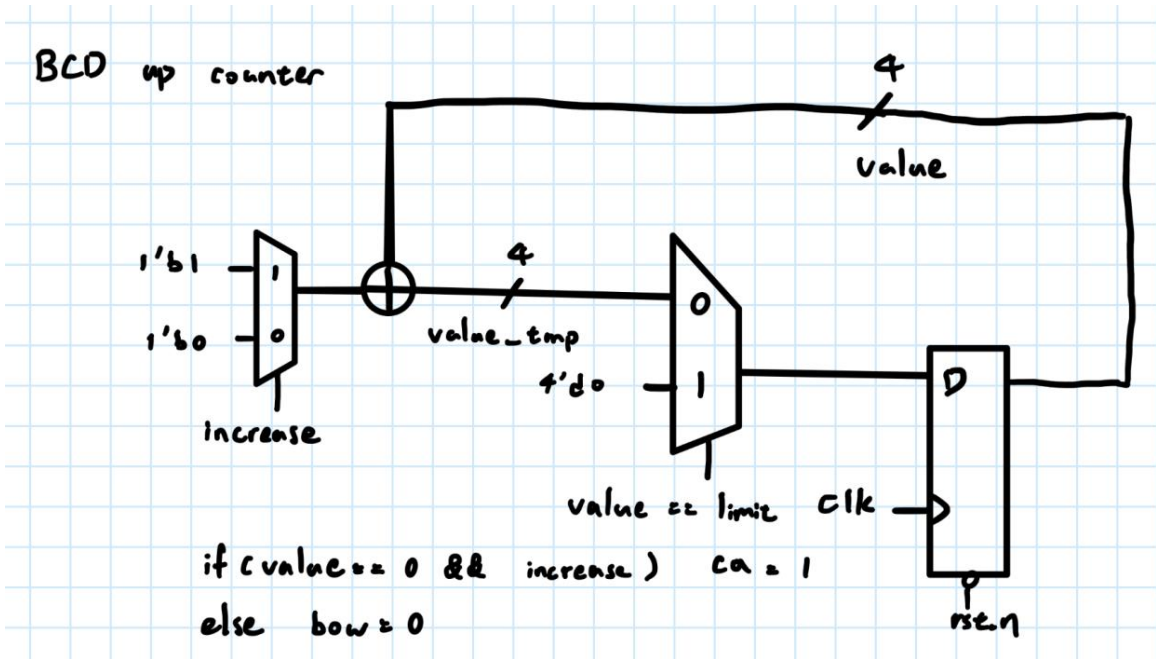
Output: [7:0] ssd, [3:0] yes,

[1:0] light_state, light_lap, light_press(for checking)

Design Implementation

Block diagram

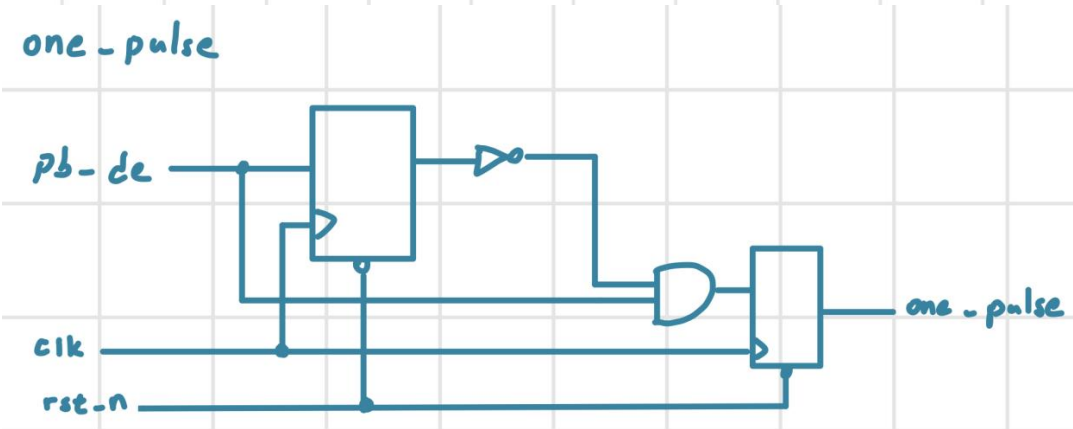
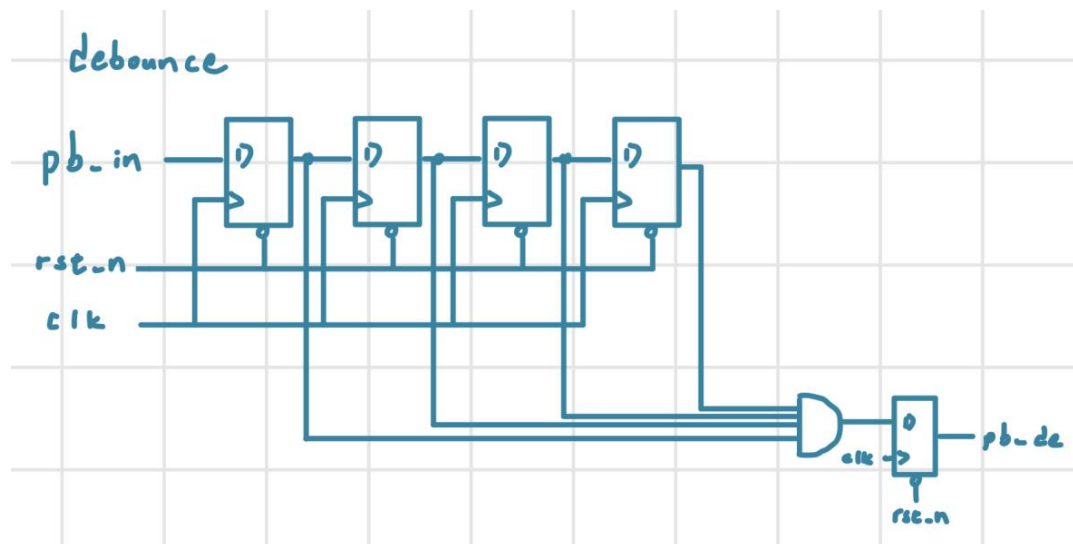




因為 in 跟 lap 不會同時為 1，所以沒有 11 的這種 input

Show

是利用 if else 選擇要輸出 counter 的值還是 lap 的值，為了顯示即時的數字所以使用 100M 的 clk



I/O	fcystal	yes 3	yes 2	yes 1	yes 0	ssd7	ssd6	ssd5	ssd4	ssd3	ssd2	ssd1	ssd0	rst_n
Site	W5	V19	U19	E19	U16	V14	U14	U15	W18	V19	U19	E19	U16	R2

light_state[1]	light_state[0]	light_press	light_lap	lap_reset	stop
L1	P1	P3	N3	T18	U18

Discussion

想法是先接好 4 個 BCD up counter，使 FPGA 可以順利地從 00:00 數到 59:59，然後因為多了 lap 所以有 fsm 出現了 4 個 state，讓難度提高了一些，關於 lap 的功能，想法是利用 fsm 輸出一個 show_enable 來決定要表現出哪一組數字，我的想法是在做出一個 register 去儲存按下 lap 當下的數字，最後再用選擇器去選擇要輸出的數字，而暫停的功能則可以沿襲 lab5_1 的做法，最後一部分則是設計長按是 reset 短按是 lap，想法是在把 lap_reset 接到另外一組 up counter 的 count_enable 使得按下去的時候 counter 開始數，數到一個特定的數字時則輸出 reset 如果 counter 小於特定的數且不等於 0 時則輸出 lap。

Lab5_3

Design Specification

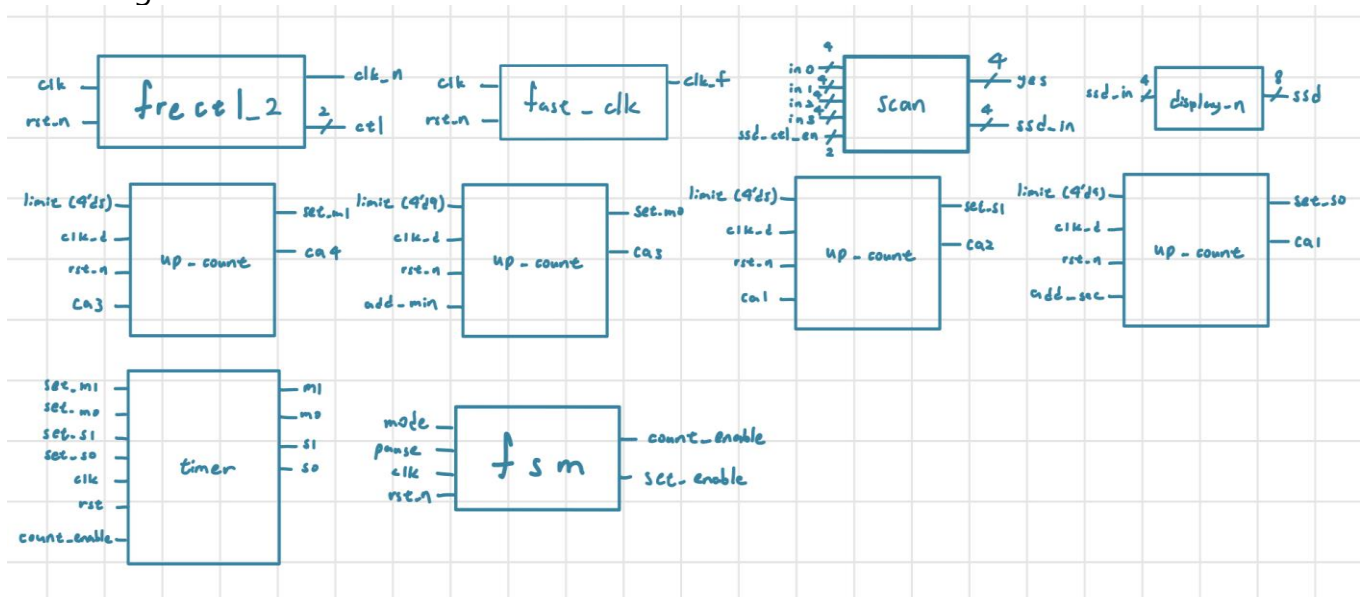
For a timer

Input: clk, rst_n, clk_ctl, mode, pause_resume, start_stop, add_min, add_sec

Output: [7:0] ssd, [3:0] yes, [15:0] light

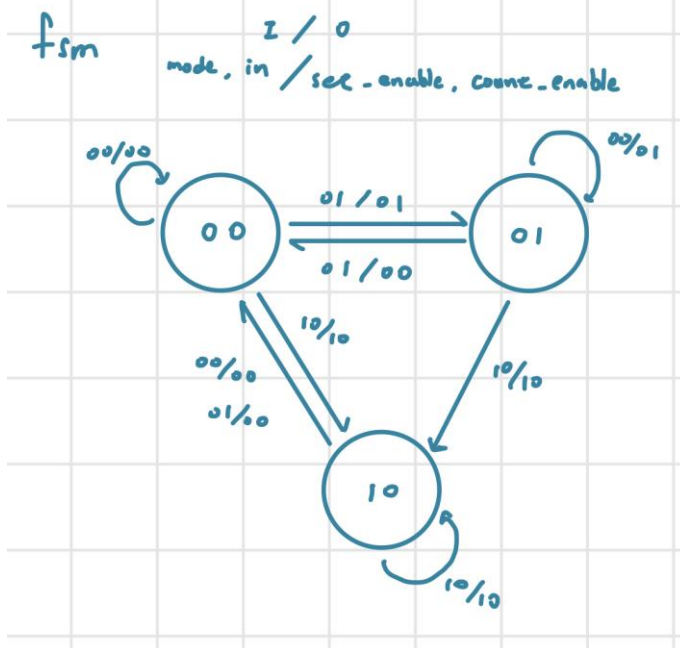
Design Implementation

Block diagram



按鈕皆有接 one pulse 跟 debounce 只是省略

因為 in 跟 mode 不會同時為 1，所以沒有 11 的這種 input



Timer

裡面包四個 BCD down counter，rst 接到 start/stop 使得每一次按下 start/stop 的時候 timer 的值都會跑回 set 所給定的值，而當 count_enable 等於 1 時就正常的 down count，並在數到 0 的時候停下來且發光

I/O	fcystal	yes 3	yes 2	yes 1	yes 0	ssd7	ssd6	ssd5	ssd4	ssd3	ssd2	ssd1	ssd0	rst_n
Site	W5	V19	U19	E19	U16	V14	U14	U15	W18	V19	U19	E19	U16	R2

light [15]	light [14]	light [13]	light [12]	light [11]	light [10]	light [9]	light [8]
L1	P1	N3	P3	U3	W3	V3	V13
light [7]	light [6]	light [5]	light [4]	light [3]	light [2]	light [1]	light [0]
V14	U14	U15	W18	V19	U19	E19	U16

mode	pause_resume	start_stop	add_min	add_sec
V17	T17	W19	T18	U17

Discussion

這個 lab 與 lab5_1 較相同只是多了設定時間的功能，所以第一個想法就是在開一個 register 儲存設定的值，並利用上數器去控制設定值，設定完畢以後再透過按鈕將值送進 timer 也就是 down counter 裡面，使他倒數，數到 0 時則停下來且 led 全亮。

Conclusion

這次的 lab 難度直接大大的提升了一個階級，因為 module 的數量變多，所以接線時的難度就更高了，所以到 lab5_3 時，選擇再利用一個 module 分別把四個 up counter 與 down counter 包起來，使版面更為簡潔。且因為狀態較為多，寫 fsm 的時候要更加小心，但完成 block diagram 及 fsm 整個 lab 都會變得流暢許多。

References

老師的ppt