106010006 黃詩瑜 電機 21

Lab 5: Timers

1. **Construct a 30-second down counter with pause function. When the counter goes to 0, all the LEDs will be lighted up. You can use one push button for reset and one other for pause/start function.**

1.1 Implement a periodic 30-second down counter and demo with the FPGA board. 1.2 Implement Prelab 1.3 and demo with the FPGA board.

1.3 Combine 1.2 and 1.3 to finish the experiment.

Specification:

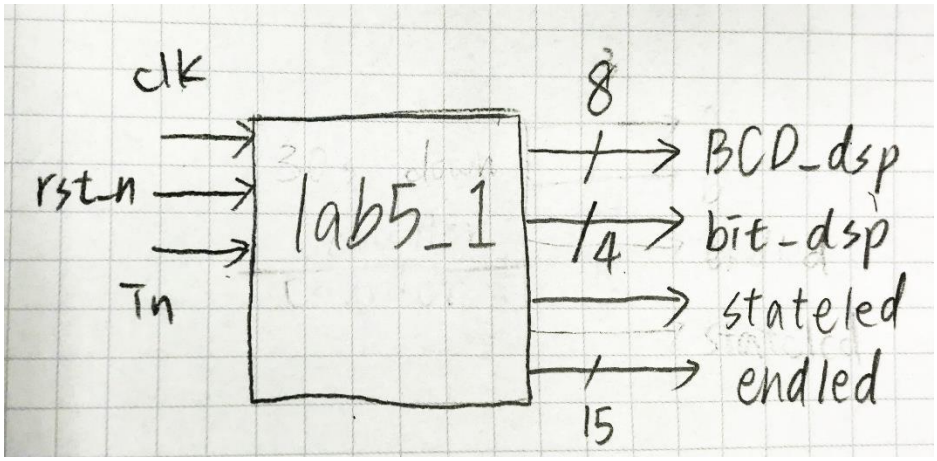
Input: clk,rst_n,in

Output: BCD_dsp[7:0],bit_dsp[3:0],endled[14:0],stateled

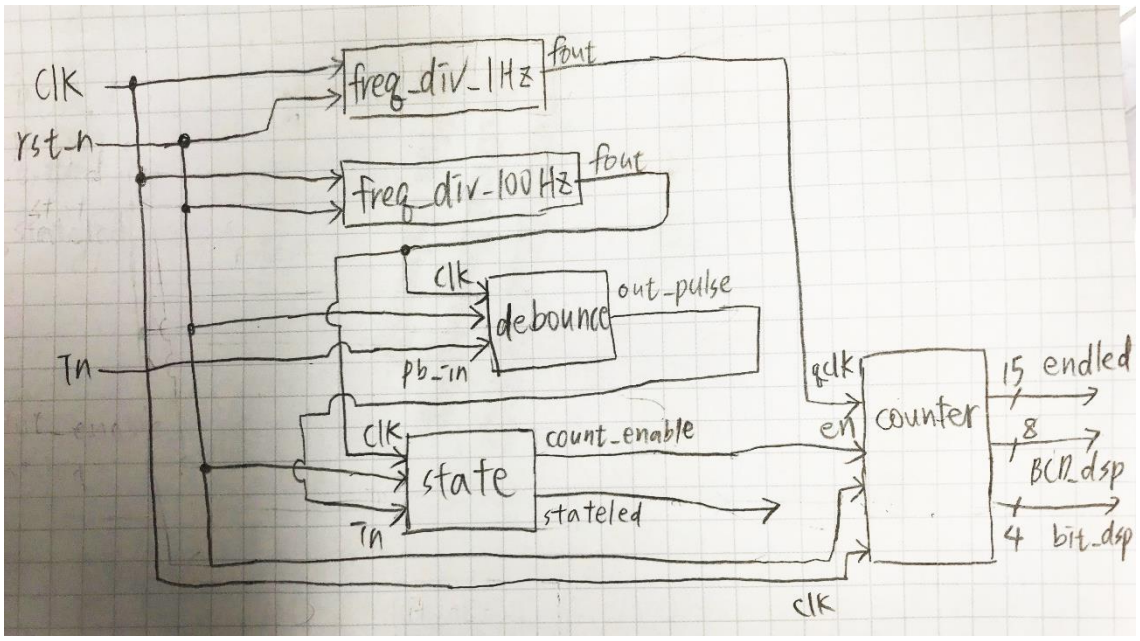
I/O	clk	rst_n	in	stateled	bit_dsp[0]	bit_dsp[1]	bit_dsp[2]	bit_dsp[3]
LOC	W5	U17	T17	U16	U2	U4	V4	W4

BCD_dsp[0]	BCD_dsp[1]	BCD_dsp[2]	BCD_dsp[3]	BCD_dsp[4]	BCD_dsp[5]	BCD_dsp[6]	BCD_dsp[7]
V7	U7	V5	U5	V8	U8	W6	W7

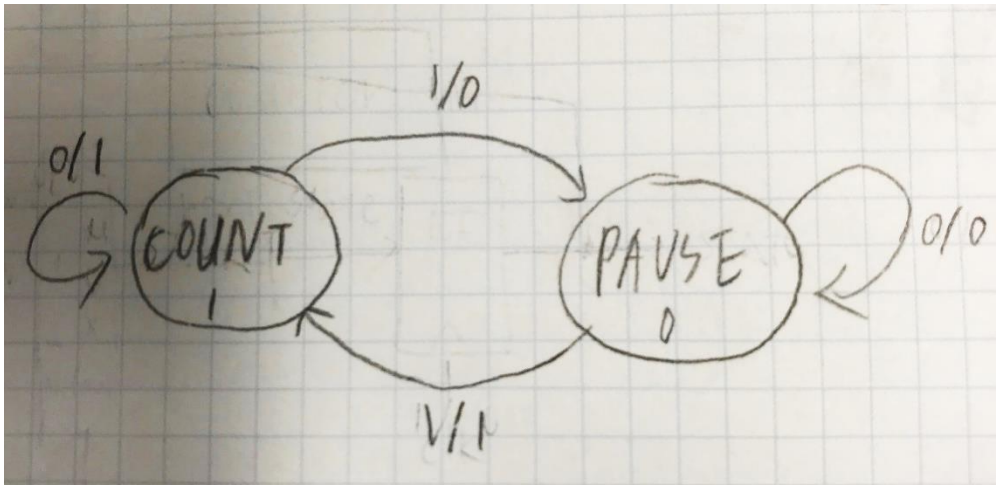
endled[0]	endled[1]	endled[2]	endled[3]	endled[4]	endled[5]	endled[6]	endled[7]	endled[8]	endled[9]	endled[10]	endled[11]	endled[12]	endled[13]	endled[14]
E19	U19	V19	W18	U15	U14	V14	V13	V3	W3	U3	P3	N3	P1	L1



Implementation:



第一題修改 prelab5 的某些部分，其他主要設計都是一樣的，開始/暫停按鈕為 input in，把它接到 debounce 的 input pb_in，我把 debounce 和 one pulse 的功能合併在 debounce 檔案裡面，這兩個功能的 clock 都是接 100Hz，output 為 out_pulse，是過濾掉按鈕的不穩定與變成 one pulse 的訊號，再把它接到 state 的 input in，控制 state 的轉換。



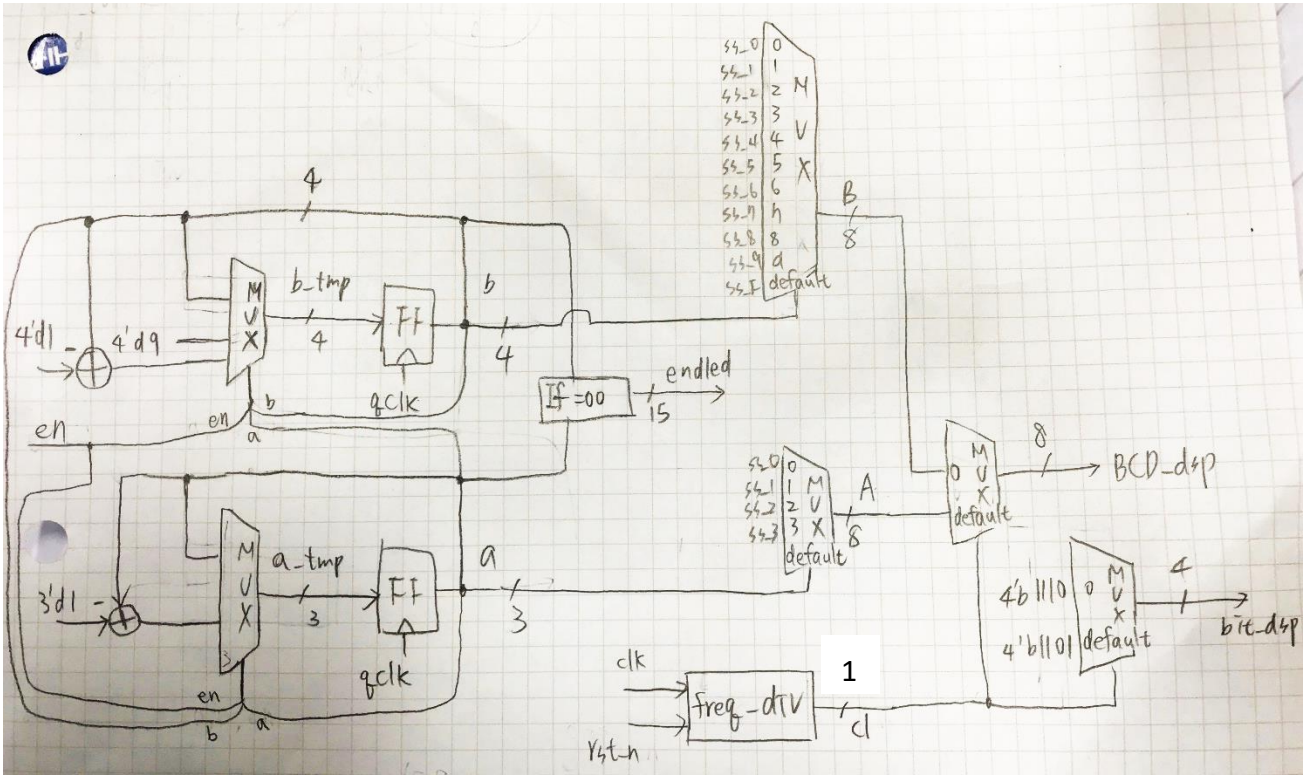
state 這個部分，先定義了 state 的值

```
`define    STAT_PAUSE    0
`define    STAT_COUNT    1
```

`define STAT_DEF 0
`define ENABLED 1
`define DISABLED 0

一開始設定 state 為 PAUSE，當 clock 開始跑之後，FF 會把 next_state 的值給 state，一開始 push button 還沒按下，in 就是 0，所以 state 還是在 PAUSE，當按下 button 後，in 變 1，`STAT_COUNT 就會傳給 next_state，`ENABLED 傳給 count_enable，count_enable 是 1-bit output 他會接到另一個 source counter 的 input en，en 為 1，就能開始倒數，當 state 變成 COUNT 後又按下 button，in 變 1，`STAT_PAUSE 傳給 next_state，DISABLED 傳給 count_enable，en 變成 0，counter 就會暫停。

設一個 bit 的 stateled 接到 state，當接到 FPGA 板上的 LED，就能知道當下的 state。



counter 的部分，跟之前實驗很像，只是加入了開始/暫停的控制，個位數的部分是由 4-bit b 控制，將 FF 的 output 設為 4 個 bits b，一開始 b 的值變成設為 0，因為是從 30 開始倒數，有一個 MUX 判斷，當 push button 還沒按下，en=0，b_tmp 的值會是 b，所以不變，也就是暫停的狀態，按下 button 後，en 會變成 1，就會開始倒數，當 b 不是 0，會將 b-1 設給 b_tmp，過 1 秒再將 b_tmp 的值送給 b，當 b 一直向下數，減到 0 時，b_tmp 就會變成 9，再將值給 b，這樣 b 就會從 9-8-7-….數到 0 後，變回 9 開始數了。

十位數的部分由 3-bit a 控制，將 FF 的 output 設為 3 個 bits a，一開始 a 的值變成設為 3，也會有一個 MUX 判斷，當 push button 還沒按下，en=0，a_tmp 的值會是 a，也是暫停的狀態，按下 button 後，en 會變成 1，就會開始倒數，當個位數 b 不為 0 時，十位數不變，a_tmp 為 a，

若 b 為 0 但 a 不是 0，代表個位數數到 0 了，十位數要借位，就將 a_tmp 設為 a-1，過 1 秒後 a_tmp 的值會送給 a，a 再回到判斷的地方，當 b 為 0 且 a 為 0，代表個位和十位數都是 0，就將 a_tmp 設為 a，b_tmp 設為 b，他們就會在 00 停止， 藉由這些 blocks，3-bit a 和 4-bit b 就能從 30 開始倒數到 00 了。

FF 部分的 code:

```
always @(posedge qclk or posedge rst_n)
if (rst_n)begin
  b <= 4'd0;
  a <= 3'd3;
end
```

rst_n 是接另一個按鈕，當按下按鈕時，rst_n 為 1，a 和 b 的值就會被設為 3 和 0，也就是 reset 的功能。

當 a 和 b 都是 00 時，會讓 15-bit endled 變成 1，當他接到 FPGA 板上的燈，就能讓 LED 燈都亮，代表倒數完了。

為了要把數字顯示再 7-seg LED 燈上，先將控制 7-seg LED 要顯示的數字的 8 個 bits define 給對應的 SS_0~9

```
`define SS_0 8'b00000011
`define SS_1 8'b10011111
`define SS_2 8'b00100101
`define SS_3 8'b00001101
`define SS_4 8'b10011001
`define SS_5 8'b01001001
`define SS_6 8'b01000001
`define SS_7 8'b00011111
`define SS_8 8'b00000001
```

`define SS_9 8'b00001001

再把 3-bit a 和 4-bit b 分別接到 MUX 上，output 是 8 個 bits A 和 B，這樣就能選擇相對應的 8 個 bits 去控制 7-seg LED。因為 7-seg 一次只能顯示相同的數字，所以要利用 LED 燈快速閃爍與視覺停留呈現一次顯示 2 個不同數字的感覺，用 frequency divider，output cl 會接到兩個 MUX，一個是選擇亮 7-seg 的最右邊兩個燈的哪一個，output 是 4-bit bit_dsp，會接到 FPGA 板的 U2,U4,V4,W4，可以控制 7-seg 亮的燈，1110 代表亮最右邊的燈，其他暗，1101 讓右邊第二個燈亮其他暗。另一個 MUX 是選擇亮的那個燈要顯示的數字，output 是 8-bit BCD_dsp，A,B 接到這個 MUX 中，當 cl 快速變化，就能看到 2 個不同的數字。

討論:

雖然只要將 prelab 做好的修改一下就行了，但是在 debounce 和 one pulse 要接甚麼頻率的 clock 試了很久，跟助教討論完之後都改成 100Hz 才順利讓按鈕可以比較穩定。

2. The same function as Exp. 1. Instead of using two push buttons for reset/pause/start, try to use just one push button to finish the design. (Hint: You can press the push button longer to represent the reset)

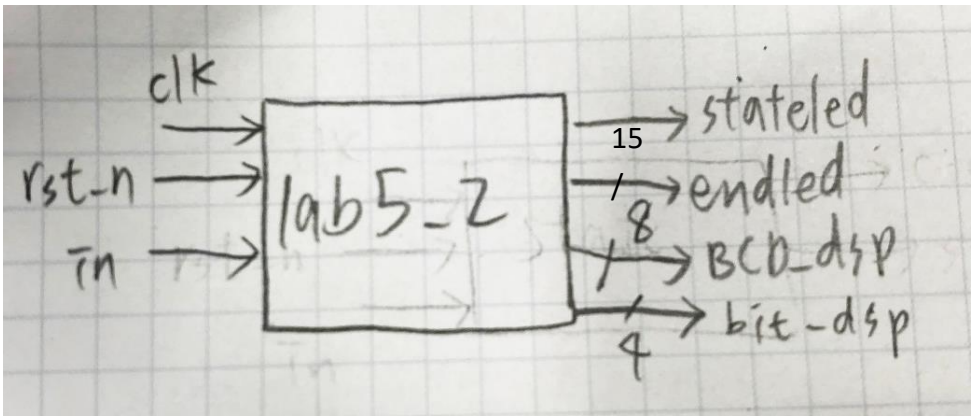
Specification:

Input: clk,rst_n,in
Output: BCD_dsp[7:0],bit_dsp[3:0],endled[14:0],stateled

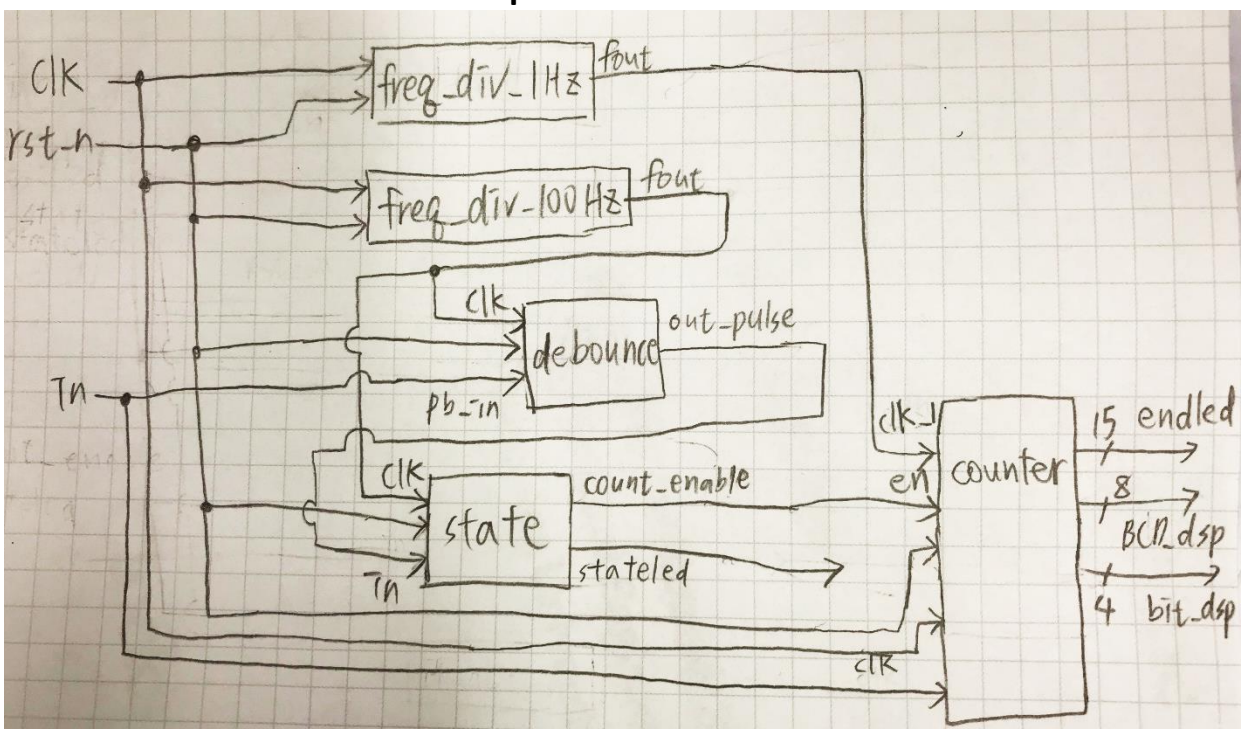
I/O	clk	rst_n	in	stateled	bit_dsp[0]	bit_dsp[1]	bit_dsp[2]	bit_dsp[3]
LOC	W5	V17	T17	U16	U2	U4	V4	W4

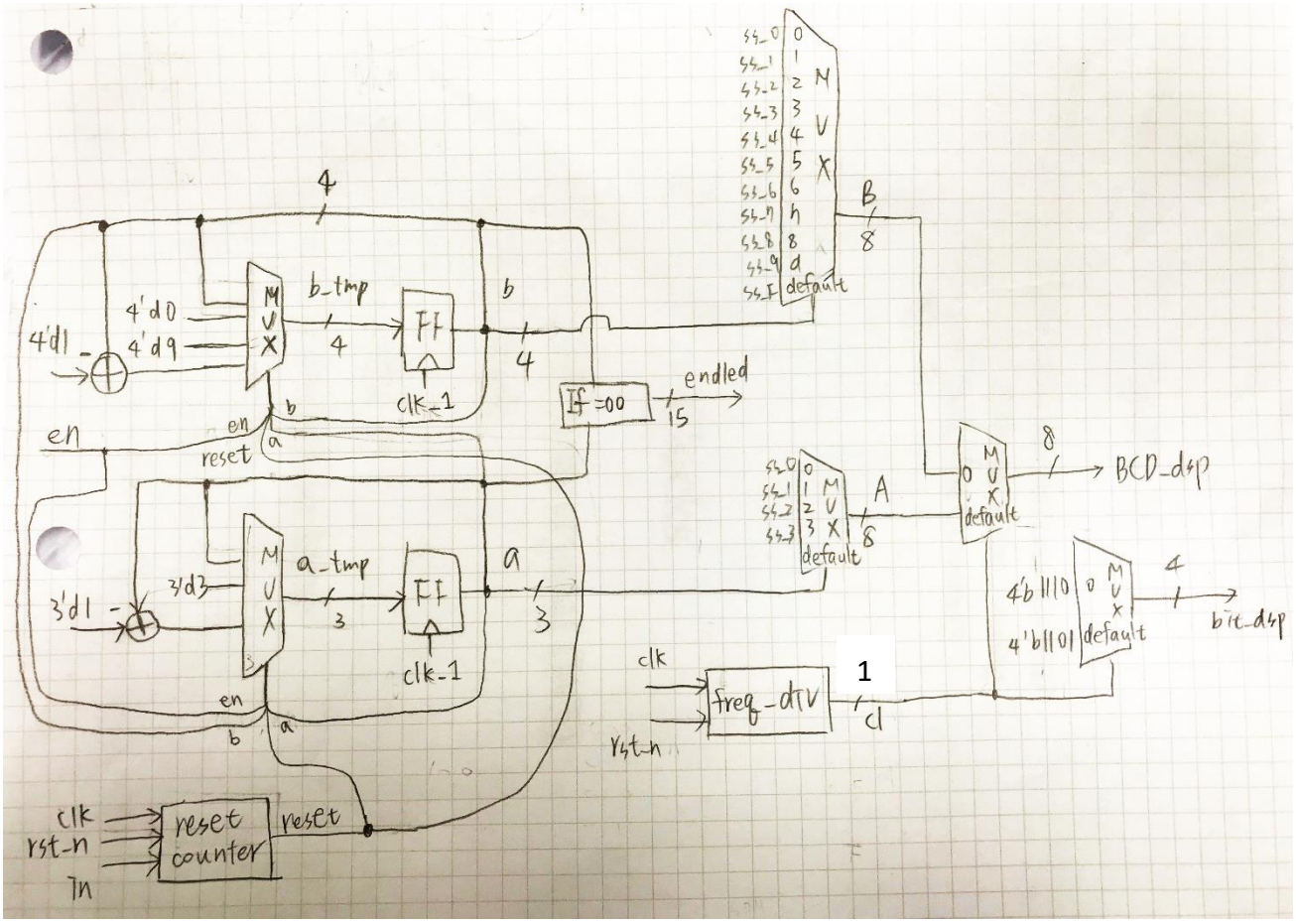
BCD_dsp[0]	BCD_dsp[1]	BCD_dsp[2]	BCD_dsp[3]	BCD_dsp[4]	BCD_dsp[5]	BCD_dsp[6]	BCD_dsp[7]
V7	U7	V5	U5	V8	U8	W6	W7

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



Implementation:





要由同一按鈕控制 reset/開始/暫停，所以 rst_n 改成接在一個 switch，剩一個按鈕 input in 來控制，大部分的設計都跟第一題一樣，只是在 counter 中把 input in 也接進去，設計一個 reset counter，clock 為 1Hz，一開始 reset 的值為 0，當按鈕按下，in 就變 1，reset 就會加 1，當按鈕長按 2 秒時，reset 就會加到 2，把 reset 接到倒數 30 秒的十位和個位 a,b 中的 MUX，當 reset 為 2 時，b_tmp 就會設為 0，a_tmp 為 3，也就是從 30 開始倒數，長按 reset 的功能就設計好了。

討論:

老師說所有 reset 都是同一個 reset，後來助教說可以把 reset 接在一個 switch

上，我才做得出來，長按切換的功能，原本想說是不是要在加一個 state，後來才想到可以設計成一個 counter 去數，這題真的花了很長的時間才完成。

3. (Bonus) Use two push buttons to control a multi-function timer (mode selection, reset, start, stop). The stop timer has two modes: 30-second/1-minute countdown. When being reset, the seven-segment display shows the digits 30/1:00. When the timer counts to 0, it will stop.

3.1 List the specification of the detector.

Specification:

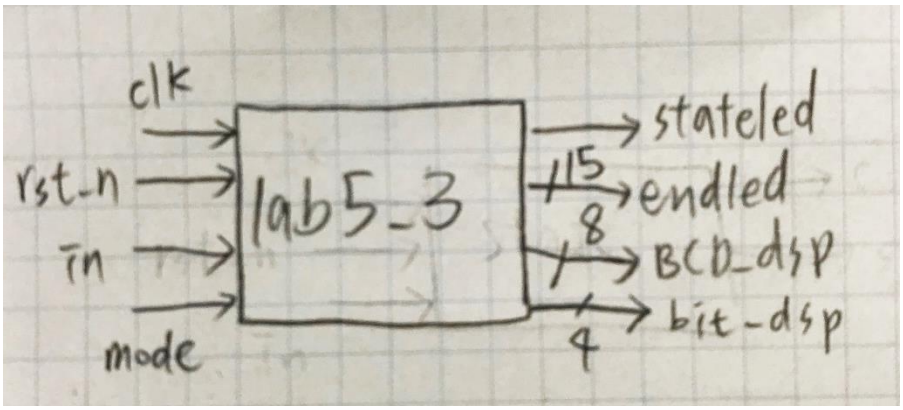
Input: clk,rst_n,in,mode

Output: BCD_dsp[7:0],bit_dsp[3:0],endled[14:0],stateled

I/O	clk	rst_n	in	mode	stateled	bit_dsp[0]	bit_dsp[1]	bit_dsp[2]	bit_dsp[3]
LOC	W5	V17	T17	U17	U16	U2	U4	V4	W4

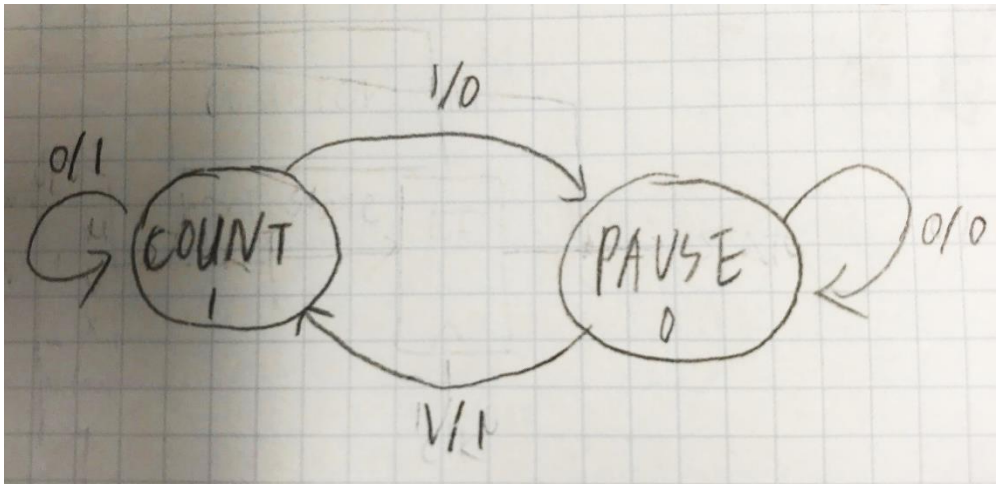
BCD_dsp[0]	BCD_dsp[1]	BCD_dsp[2]	BCD_dsp[3]	BCD_dsp[4]	BCD_dsp[5]	BCD_dsp[6]	BCD_dsp[7]
V7	U7	V5	U5	V8	U8	W6	W7

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



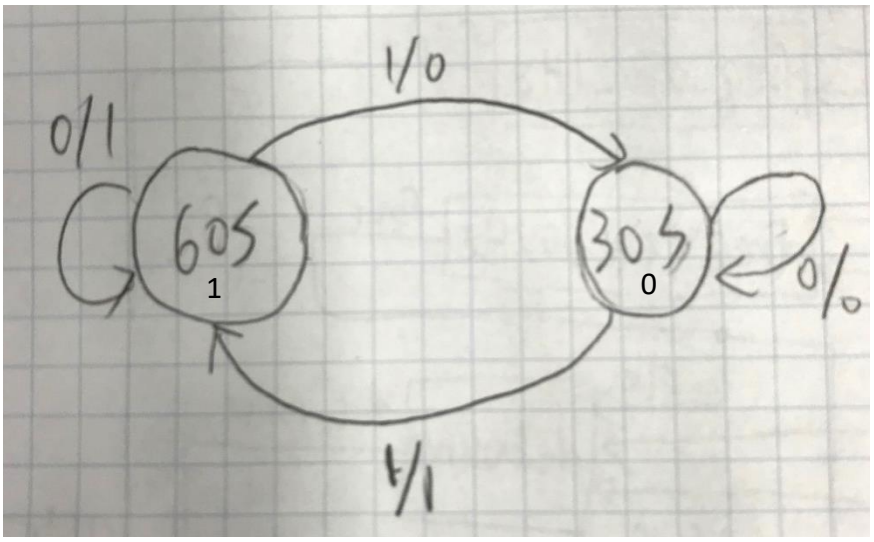
3.2 Design the FSM used in this design.

State:



開始/暫停的 state 不變，多增加了一個 state2 控制 30s/60s 的轉換。

state2:



state2 的部分，定義了

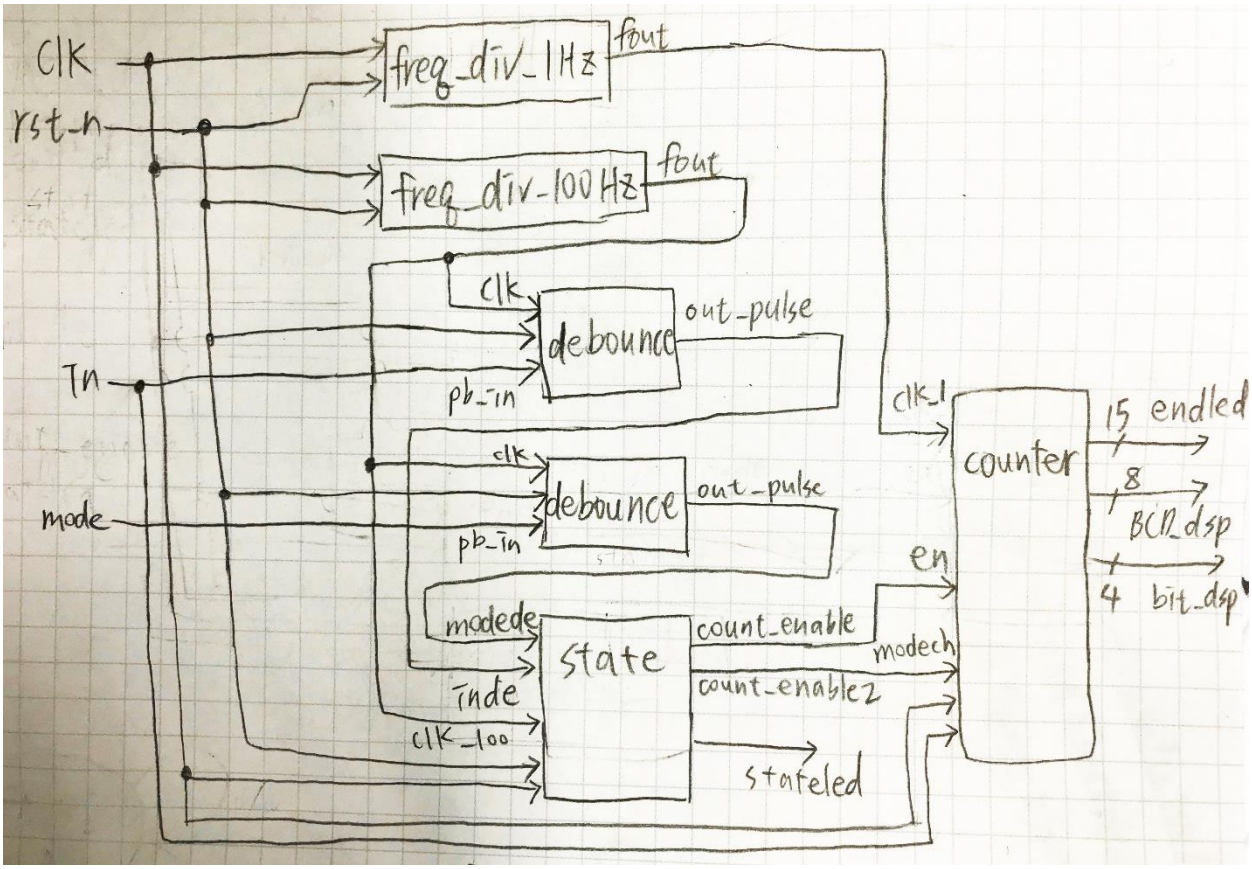
```
`define    STAT_30    0
`define    STAT_60    1
`define    STAT_DEF    0
`define    thi    0
`define    six    1
```

一開始設定 state2 為 30s，當 clock 開始跑之後，FF 會把 next_state2 的值給 state2，一開始控制 30s/60s 的按鈕還沒按下，所以 state2 還是在 30s，當按下 button 後，`STAT_60 就會傳給 next_state2，`six 傳給 count_enable2，

count_enable2 是 1-bit output 他會接到另一個 source counter 的 input modech (下方有圖)，modech 為 1，b_tmp 設為 0，a_tmp 為 0，c_tmp 為 1 (c 控制百位數，a 控制十位數，b 控制個位數)，當 state2 變成 60s 後又按下 button，`STAT_30 傳給 next_state2，`thi 傳給 count_enable2，modech 變成 0，b_tmp 就會設為 0，a_tmp 為 3，c_tmp 為 0，當長按另一個按鈕進行 reset 後， 就可以看到由 30s 倒數變成 60s，或是 60s 變 30s。

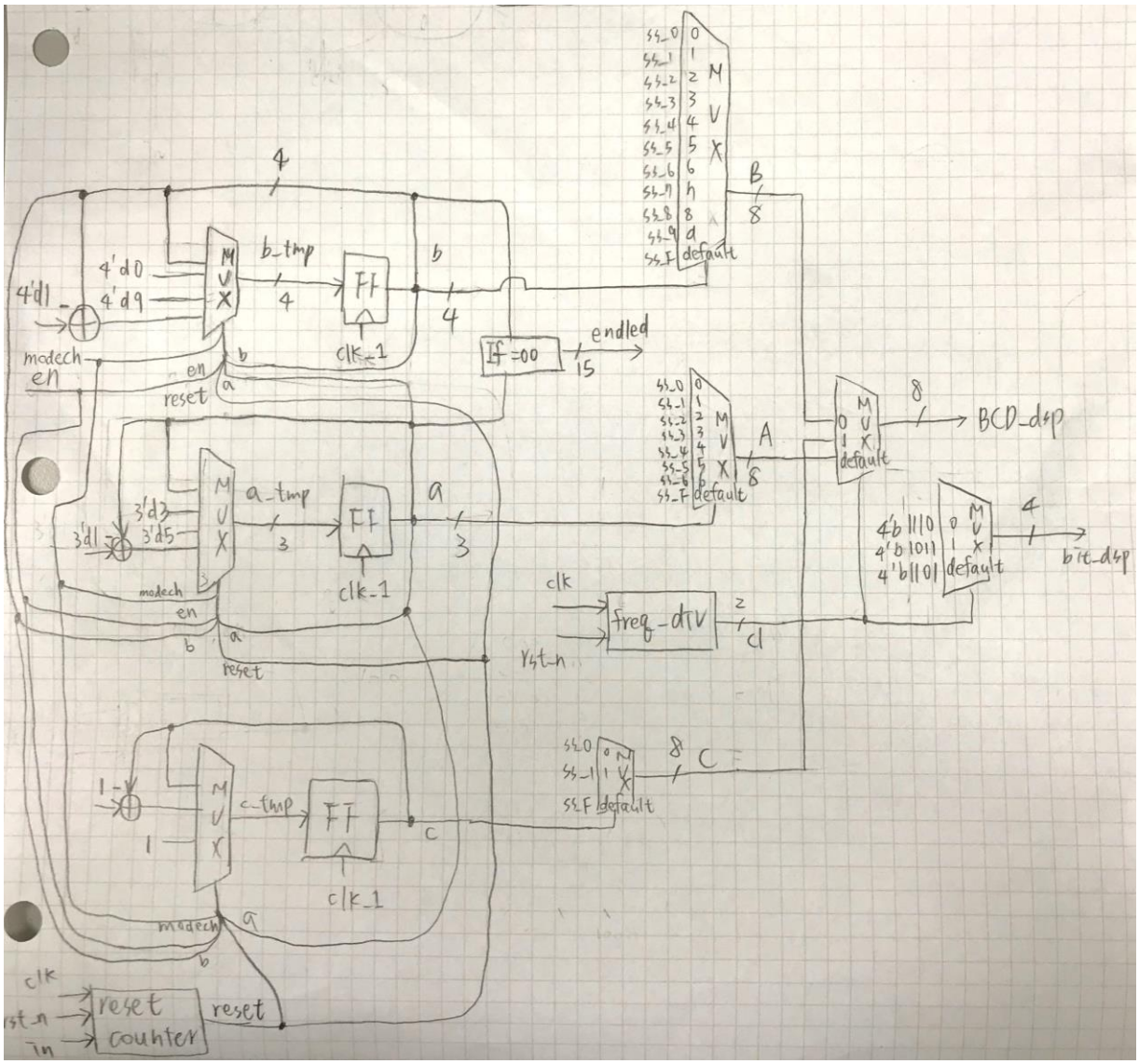
3.3 Draw the block diagram/logic schematic.

3.4 Implement the stop timer with FPGA demo board.



大部分設計都跟上題一樣，只是多了一個 input mode，是用來切換 30s/60s 倒數的按鈕，mode 會接到 debounce 裡，出來的 out_pulse 是經過去除按鈕的不穩定

和 one pulse 處理的訊號，把它接到 state 的 input modede，就可以控制 state2 的轉換，會產生一個對應的 output count_enable2，它會接到 counter 的 Input modech。



Input modech 可以控制 30s/60s 的切換，它會接到倒數的百位,十位和個位 c,a,b 中的 MUX，切換 30s/60s 的模式後，要 reset 後才能看到變化，十位數字 a 的部分，MUX 多了 3’d5 的選項，A 的 MUX 則是多了控制 LED 的 8-bit SS_4~6，百位數字 c 也接到了一個 MUX 當中選擇 SS_0 和 SS_1，freq_div 接出 2 個 bit，控制

7-seg LED 燈要顯示的數字和燈。

討論:

這題的切換我原本是做 30s 切換 60s，後來問老師，老師說一定要是 30s 和 1m，就改了一下，還好只花了一點時間。

結論:

這次實驗真的是做了好幾天，還請教了很多同學，主要是卡在 reset 那邊，能感受到實驗越來越難了，之後有問題要都找助教和老師討論一下，才能獲得更多的想法和資訊。