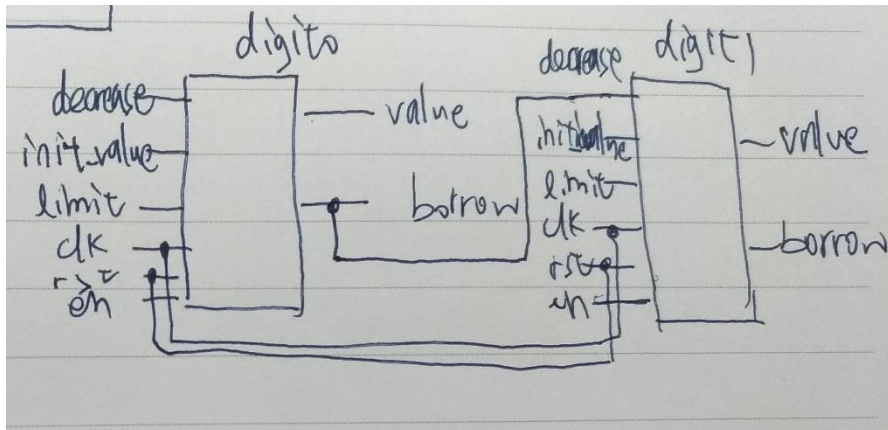


Pre-labs:

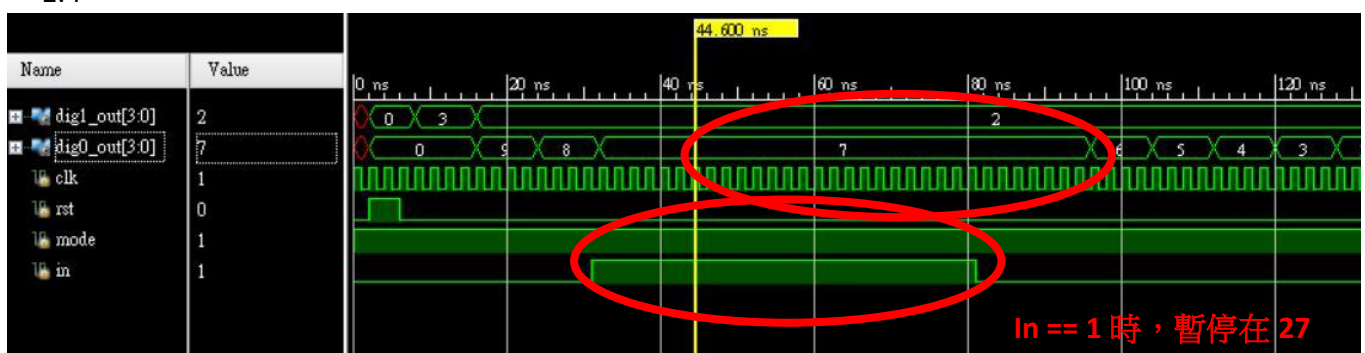
- 1.1 計時器需要：一個 input clk，一個 input rst，一個 input enable(控制暫停或開始計時)，還有一個 mode 來控制 15 或是 30 秒倒數計時。Output 有輸出的 digit1 及 digit0，還有題目需要的 16-bit Led。
- 1.2 當 en 開始時，digit0 會先作用，等到 digit0 == 0 時，digit1 的 en 會開始作用，即 digit1 少一位，少一位的同時 digit0 會回到原本的極限值(ex. 9)而繼續開始 digit0 的運算。而當 digit1 == 0 && digit0 == 0 時，led 會全部變為 1；按 rst 時，會依照輸入的 mode 選擇回到 15 或是 30 秒。



- 1.3 這題的 FSM 需要至少有兩個 state，state 01 代表運算中，state 10 代表 freeze(pause)，區分方法就是 in_latch 的長短，而再設一個 state 00 代表 reset 時的狀態，只要按超過三秒 (in_latch == 3'b111)，或是按 rst 就會跳到 state 00。最後輸出 count_enable 和 freeze_enable 來顯示現在要進行的動作。

present state	input			Next stage	output	
	in	case 1	case 2		count_en	freeze_en
00	0	x	x	00	0	0
00	1	x	x	01	1	0
01	0	x	x	01	1	0
01	1	0	0	00	0	0
10	0	0	1	10	1	1
10	1	x	x	10	1	1
11	0	x	x	01	1	0
11	1	x	x	01	1	0

1.4



Experiments

1.0 完成 exp1 除了 pre-lab1 之外還需要一些東西，需要一個大 module 將整個程式涵蓋住，並增加輸出 SSD 的小 module，將 FSM 的 count_enable 接到 stopwatch 的 en，輸出的 digit0 以及 digit1 要經過 ssd 的轉換再輸出。



2.0 第二題要將 rst 和 pause(freeze)功能合併成一個按鍵，這時需要先從 FSM 來修改，我在 FSM 新增一個 output 叫做 rst_enable，並把 input 的 rst 刪除，當需要 reset 的時候就把 rst_enable 打開，也就是在 state 01(count)的時候，只要按超過三秒(in_latch == 3'b111)時，rst_enable = 1，並回到初始狀態 state 00。這時候在大 module(stopwatch_disp)裡面，把所有的 rst 用經過 FSM 輸出的 rst_enable 代替，再將 input 的 rst 刪除，就可以達成 rst 跟 pause 在同個按鈕，當短按時是 pause，長按時是 reset 的功能。

```
`STAT_COUNT:
if((in) && (in_latch==3'b111)) begin
    next_state = `STAT_DEF;
    count_enable = `DISABLED;
    freeze_enable = `DISABLED;
    rst_enable = `ENABLED;
end
else if((in) && (in_latch==3'b001)) begin
    next_state = `STAT_FREEZE;
    count_enable = `ENABLED;
    freeze_enable = `ENABLED;
    rst_enable = `DISABLED;
end
```

3.0 要做到顯示 1:00 的倒數計時器，要先從 stopwatch 裡面修改，新增一位 digit2，當其中一個 mode 時顯示 1:00，另一個 mode 顯示 0:30。其餘就跟 exp2 裡一樣。最後在輸出的時候，加入 dig2_out 就可以了。