Lab7.1 Use the four SSDs as the display. The left two digits represent the minute and the right two digits represent the second. Use two push buttons to control the function. Use one button to control start/stop and the other to control the lap and reset.

1.Design specification:

功能:具有開始暫停及時間記憶計時器

輸入: pb_mode (開始/暫停計時)

pb_lap (記憶現在分秒)

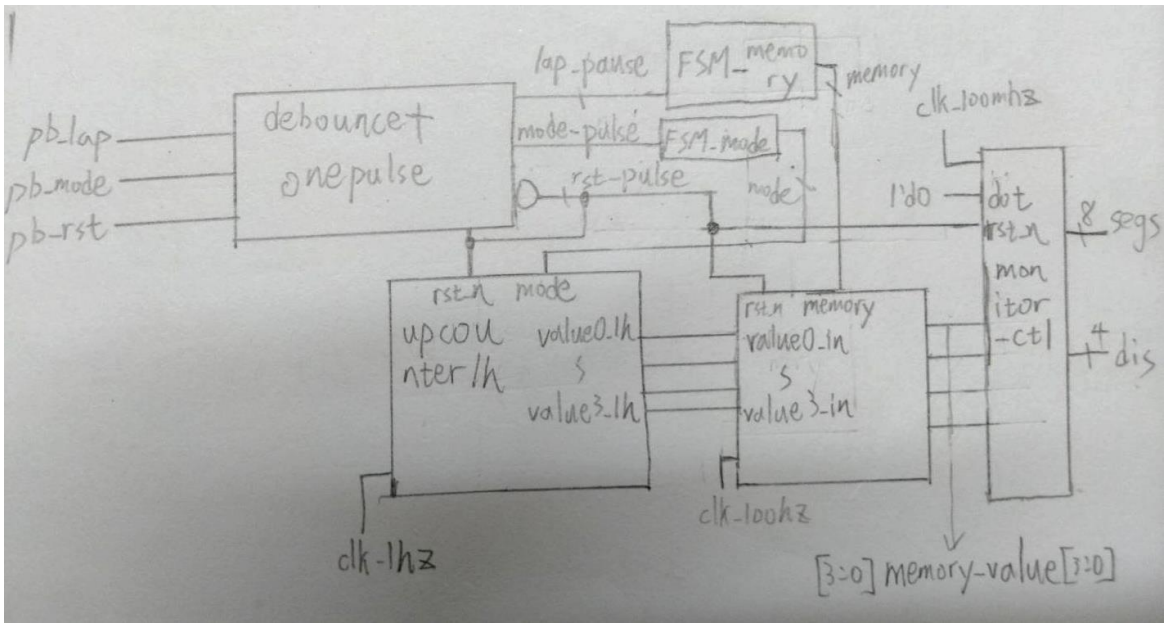
pb_rst (重置計時器)

clk_100mhz (石英震盪器輸入)

輸出: [7:0]seg (控制每個七段顯示器針腳)

[3:0]dis (控制七段顯示器快慢)

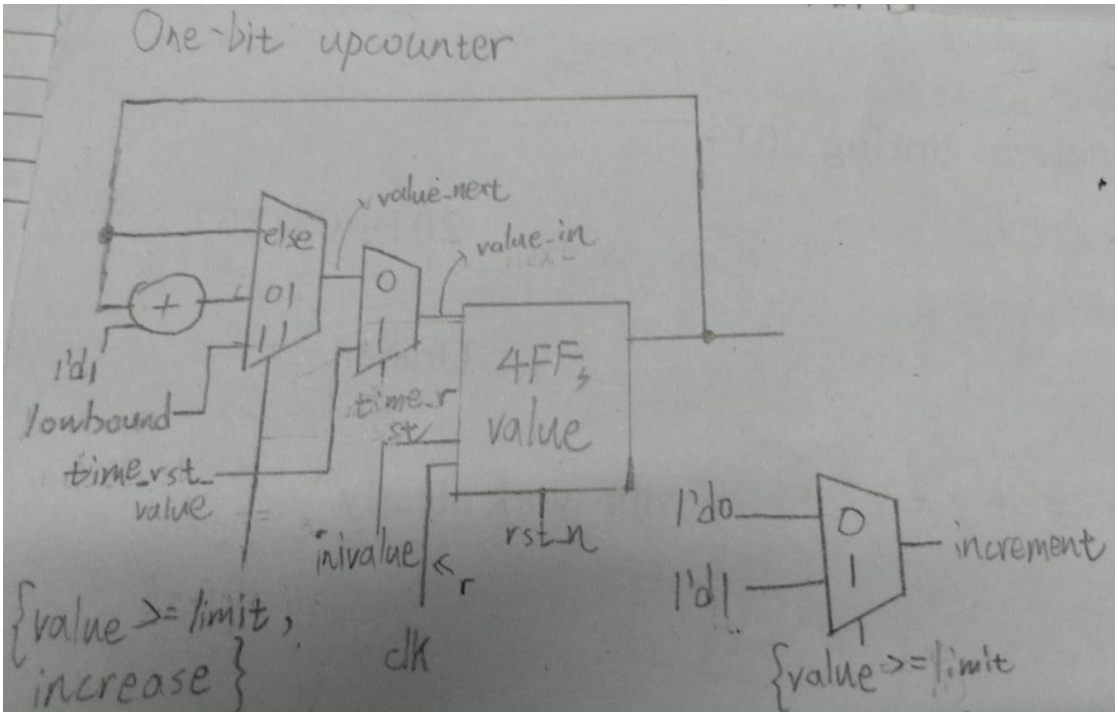
2. Design Implementation



這一次的 lab 題目可以把題目拆成開始/暫停及記憶/向上的 Finite state machine，向上計時系統、記憶時間系統和位數顯示器系統，按鍵 pb_lap 處理後用來控制 FSM_memory，控制 memory 模組要不要記憶。按鍵 pb_mode 處理後用來控制 FSM_mode，控制 upcounter 模組要開始計時或暫停計時。按鍵 pb_lap 處理後用來重置整個電路板。當 pb_rst 按下後，upcounter1h 及兩個 FSM 都會歸零及暫停，及解除記憶功能。當按一次 pb_mode 時，upcounter1h 會開始向上計時，在按一次則暫停計時。其輸出的四位值會被送到 memory 模組。Memory 模組原本會輸出 upcounter1h 的值到顯示器處理器 monitor_ctl，在按一次時會儲存目前的值並持續輸出該值，以此類推。

FSM_mode、FSM_lap 及 monitor_ctl 的結構和上一個 lab 的 FSM 結構一樣，只是 state 功用不一樣，所以不另行贅述。

Upcounter1h:



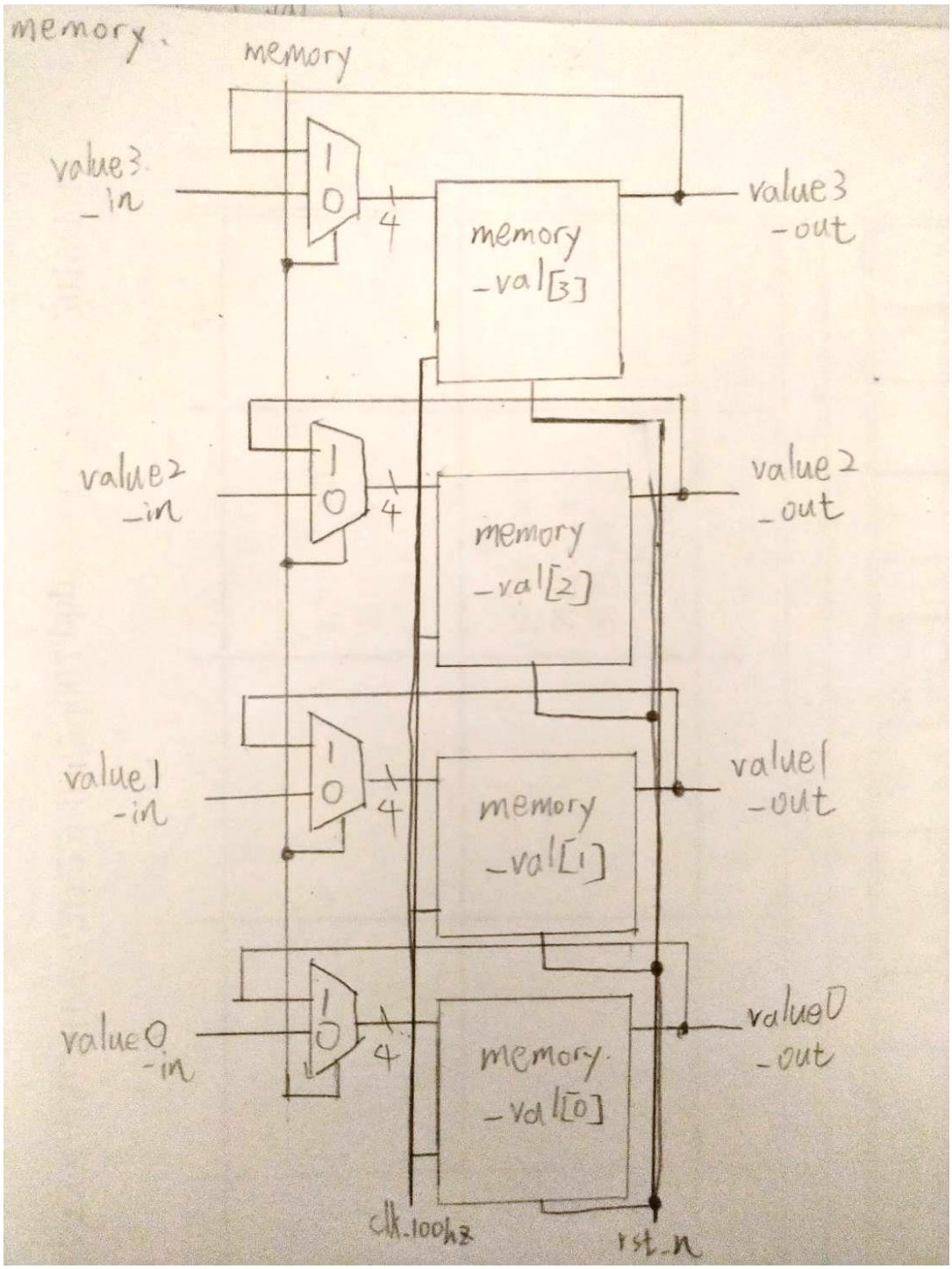
由於本題會用到多位元向上計數器，所以向上計數器應該要具有本位元上數和判斷進位的功能。所以會有一個 increase 是用來偵測要不要使本位元加一，當前位元為九時或是功能上有需要，就會輸入高電位到 increase，啟動上加功能，使得此位元的值加一，輸出至 value_next。而當 increase 為低電位時，則會保持原本的值。而判斷是不是該進位的時候，就由 increase 及值為 limit 時判斷，假設兩項條件皆成立時，increment(進位)為一輸出，同時本位元會輸入 lowbound 值，也就是該位元的最小值，然後會啟動下位元的 increase 使其加一。而當值不為零但 increase 為一時，此多工器會輸出比暫存器的值再加一的值。increase 為零時則不加值。

接著，value_next 後的多工器則是選擇要不要使暫存器變回 time_rst_value 的值，由 time_rst 決定。time_rst 的觸發條件，則由 upcounter 外的組合邏輯決定，在這個 lab 是以某兩個計數器的 digit 大小大於等於某特定值決定。

Inivalue 則是在 rst 觸動時會使得暫存器的值變回 inivalue 輸入的值，不論其原本暫存器的值為多少。

Upcounter1h 則是將四個計時器接在一起，接法就如同上個 lab 的時及分。順帶一提，當 FSM_mode 輸入 0 時，upcounter1h 會暫停，當 FSM_mode 輸入 1 時，upcounter1h 會開始計時。

Memory:



Memory 的結構很簡單，四個暫存器用來存值並且一直輸出至顯示器的處理器 monitor_ctl。當 FSM_lap 輸入 0 時，會一直輸入由 upcounter1h 輸出的值，使暫存器的值和 upcounter1h 同步。而 FSM_lap 輸入 1 時，暫存器則會一直保留目前儲存的值，使其一直輸出在 FSM_lap 由 0 到 1 時，儲存在暫存器的值。

3. I/O pin assignment

	變數接收處	晶片 I/O 點	描述
輸入	clk_100mhz	W5	石英震盪器時脈輸入
	pb_lap	T17	右邊的按鈕
	pb_rst	U18	中間的按鈕
	pb_mode	W19	左邊的按鈕
輸出	seg[7]	W7	控制七段顯示器腳位
	seg[6]	W6	
	seg[5]	U8	
	seg[4]	V8	
	seg[3]	U5	
	seg[2]	V5	
	seg[1]	U7	
	seg[0]	V7	
	dis[3]	W4	控制每個七段顯示器開關
	dis[2]	V4	
	dis[1]	U4	
	dis[0]	U2	

Lab7.2 Implement a timer (can support as long as 23:59) Use one DIP switch as the ‘setting’ control. When the ‘setting’ is ON, you can use two buttons to set the hour and minute. Use other two buttons to control the timer operation. One button for start/stop and the other button for pause/resume.

1.Design specification:

功能:具有開始暫停及時間記憶計時器

輸入: pb_mode (開始/暫停計時)

pb_stop (停止計時器)

pb_rst (重置計時器)

pb_hour(設置小時)

pb_min(設置分鐘)

sw_set(切換設定/計時模式)

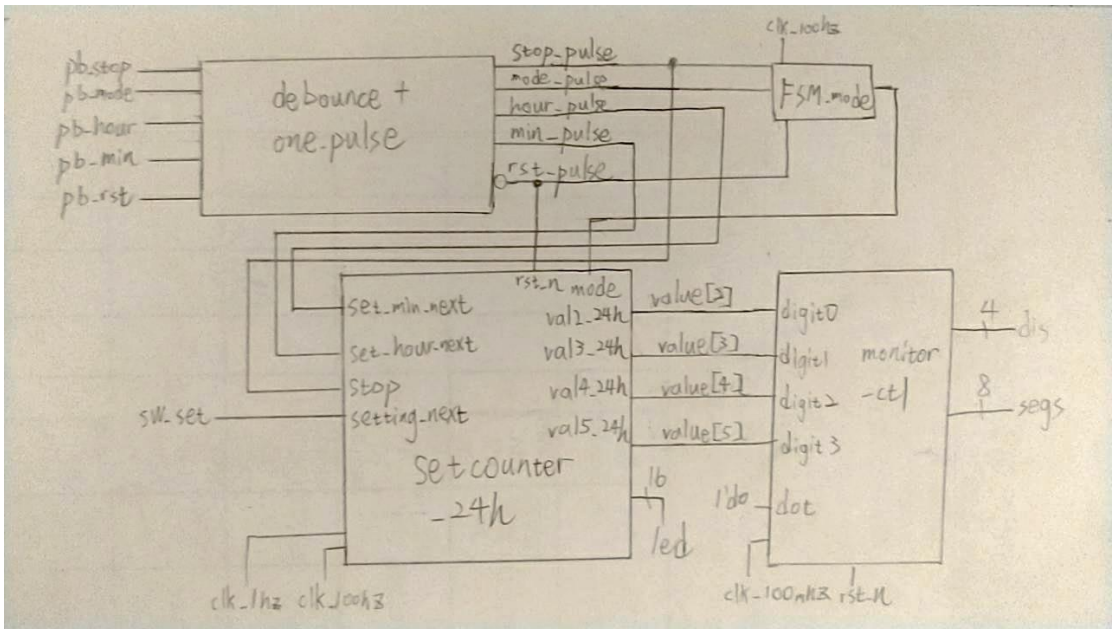
clk_100mhz (石英震盪器輸入)

輸出: [7:0]seg (控制每個七段顯示器針腳)

[3:0]dis (控制七段顯示器快慢)

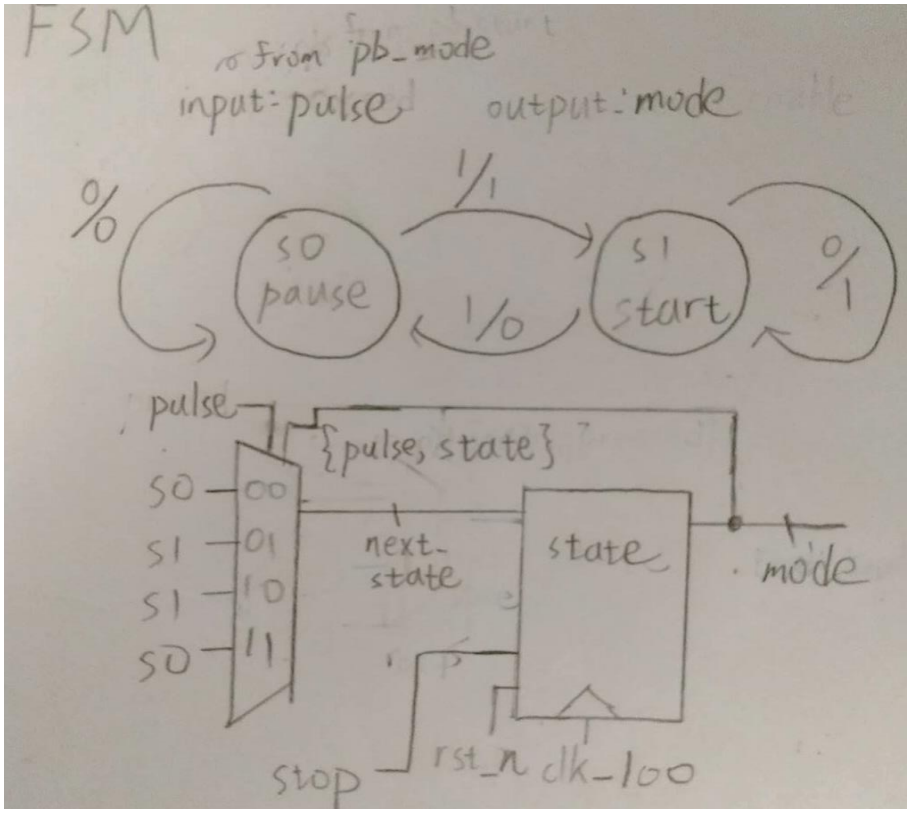
[15:0]led(控制 led 開關)

2. Design implementation



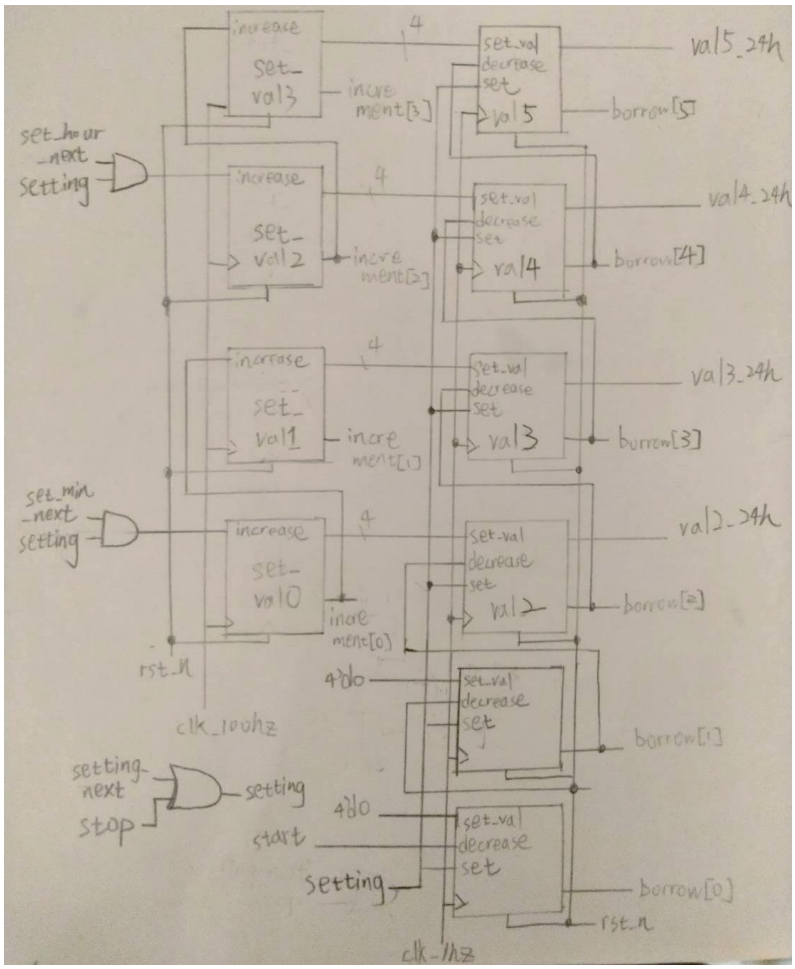
輸入的按鈕經過 debounce 及 one_pulse 後變成一個 100hz 的突波。然後 FSM_mode 會接收 stop_pulse、mode_pulse、及 rst_pulse 的訊息，告知計時器要開始倒數或是不倒數。核心部分 setcounter_24h，即負責倒數計時及時間的設定。在 sw_set 的開關向下時，進入設定模式，系統會讀取時和分的按鍵訊息，每按一下時，開始計時的時或分會向上調一。sw_set 的開關向上時，則會進入計時模式，可以藉由 pb_mode 開始、暫停或藉 pb_stop 停止計時。Pb_rst 按下時，會使得調整的倒數計時及正在倒數的時間全部歸零。

FSM_mode



當 pb_mode 訊號傳到 FSM_mode 後，下一個 state 會根據現有的 state 及 input 來做調整。State 有分為 pause 及 start。當 pause 時收到 pulse 的高電位訊號，它會跳到 start 狀態。當 pause 狀態時收到低電位訊號，它會仍在 pause 狀態。當 start 狀態時收到高電位訊號，它會跳到 pause 狀態。當 start 狀態時收到低電位訊號，它會仍在 start 狀態。而當 stop 訊號或是 rst_n 訊號為一時，state 會直接跳到 pause 狀態。在 start 狀態的 FSM_mode 會輸出 mode 值零，反之為一。mode 訊號會傳到後面的計時器，控制計時的開關。

Setcounter_24h:



這個模組總共用到了十個 4bit 的暫存器，其中左邊四個是用來存使用者設定的時及分值，後面六個是用來倒數計時及輸出時間到顯示模組所用。

在 sw_set 撥上去時，setting_next 為一，此時 setting 的輸入會一直為一，處在設定狀態，此時，如果 set_min_next 為一(按下 pb_min 時)，set_val[0]的 increase 會變為一直到該 counter 的值加一(下一個正緣)為止。Set_hour_next 為同理。而在左邊四個暫存器的數字，會直接傳到右邊上四個 counter 模組 set_val 輸入端中，由於 set 在此模式中皆為一，所以值會一直被傳到右四個模組中。下面兩個 counter 在此模式中，會被輸入零，代表秒數為零。

然後當 set 值為零時，進入計時模式，在這模式中 setting_next 值為零，所以右邊 counter 不再讀取左邊 counter 的數字。當 start 啟動時(FSM_mode 為一時)，計時器會開始倒數，其原理和之前的 lab 都一樣，反之則是暫停，不再向下數。但當 pb_stop 被按下去時，setting 值會變高電位一個時脈，使得左邊的原始值被輸到右邊去，同時讓 FSM_mode 變為暫停狀態，不再倒數，達成重新計時的功能。Rst_n 輸入時所有的暫存器直接歸零，重新開始。

最後，當所有倒數計時的 digit 為零時，會使得面板上的 led 燈全亮，其餘則全暗。

3. I/O pin assignment

	變數接收處	晶片 I/O 點	描述
輸入	clk_100mhz	W5	石英震盪器時脈輸入
	pb_min	T17	右邊按鈕
	pb_hour	W19	左邊按鈕
	pb_stop	T18	上面按鈕
	pb_mode	U17	下面按鈕
	pb_rst	U18	中間按鈕
輸出	seg[7]	W7	控制七段顯示器腳位
	seg[6]	W6	
	seg[5]	U8	
	seg[4]	V8	
	seg[3]	U5	
	seg[2]	V5	
	seg[1]	U7	
	seg[0]	V7	
	dis[3]	W4	控制每個七段顯示器開關
	dis[2]	V4	
	dis[1]	U4	
	dis[0]	U2	
	led[15]	L1	控制 LED 亮暗
	led[14]	P1	
	led[13]	N3	
	led[12]	P3	
	led[11]	U3	
	led[10]	W3	
	led[9]	V3	
	led[8]	V13	
	led[7]	V14	
	led[6]	U14	
	led[5]	U15	
	led[4]	W18	
	led[3]	V19	
	led[2]	U19	
	led[1]	E19	
	led[0]	U16	

心得及討論:

這一次的 lab 還比上次複雜，但是將大問題細分成如怎麼向上計時、如何讓時進位到日等小問題後，會發現所有的問題都可以分成 sequential logic 和 combinational logic，使用模組或是多工器來解決。所以以後再打類似或更大的 lab 時，要先將問題寫在紙上在畫圖，而不是直接畫圖，這樣腦袋裡的邏輯才會清晰。另外，testbench 也很重要，我很多的 bug 都是靠 testbench 解出來的呢。