

Lab6.1 Finish the time display function supporting 24-hour (00-23). Support two modes: AM/PM and 24-hour.

1.Design specification:

功能:二四小時/十二小時時鐘

輸入: pb_mode (切換 12 時/24 時模式)

sw_rst (重置計數器)

sw_dis (切換顯示時分/秒)

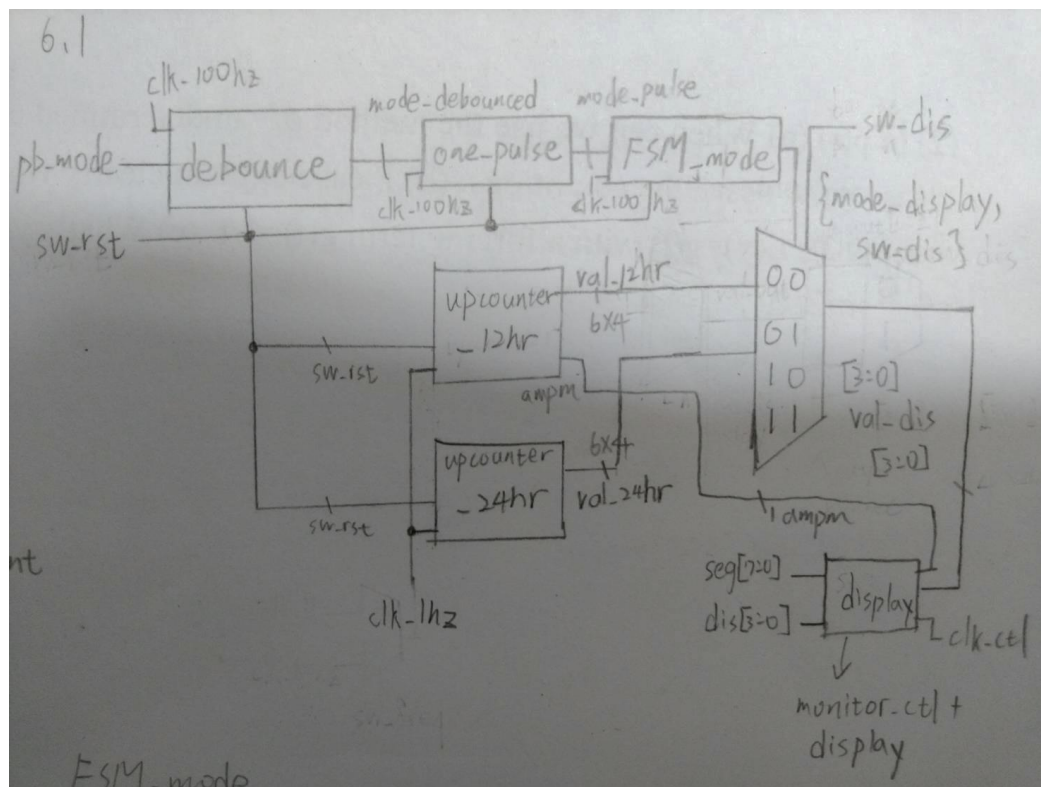
clk_cr (石英震盪器輸入)

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)

註:計時速度為原本的二十倍，便於檢查。

2.Design Implementation:



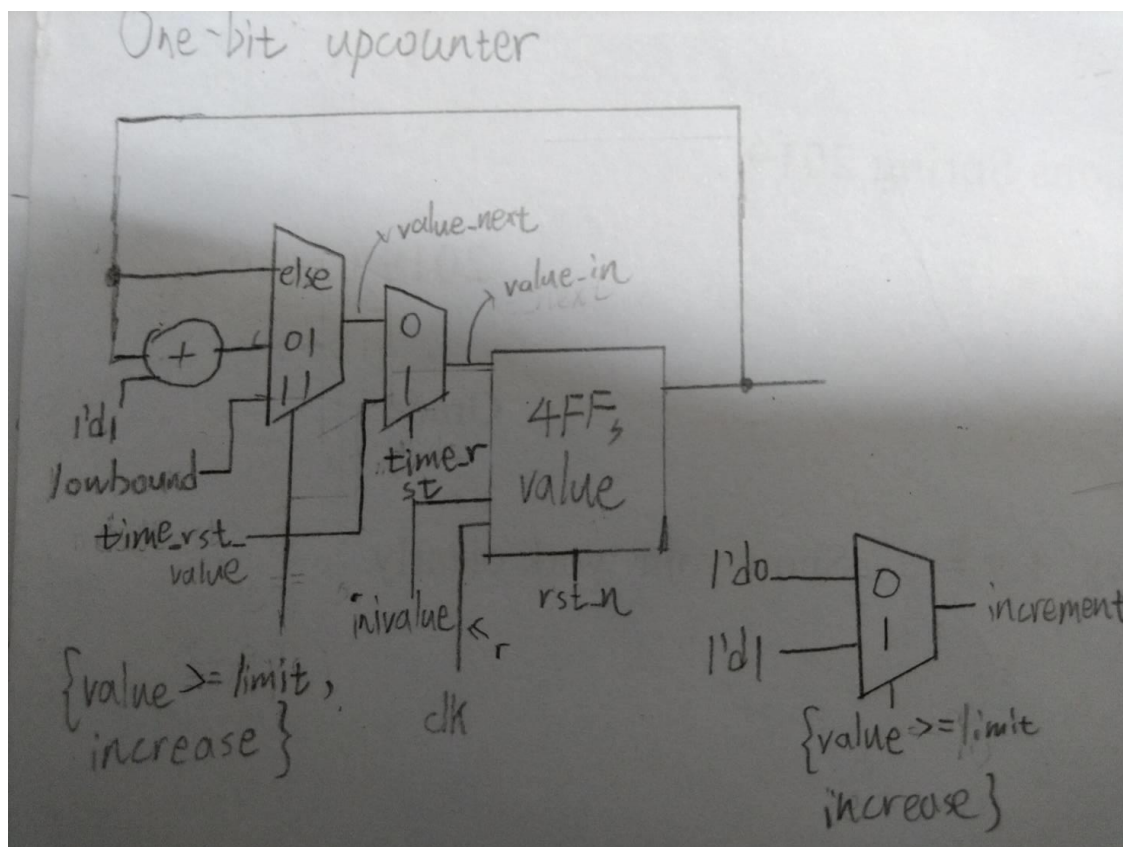
在控制 rst 的 switch 由上向下撥時，upcounter_12hr 及 upcounter_24hr 會回歸到其預設之值(12h 為 11:59:00，24h 為 23:59:00)，然後自動開始如時鐘般往前數，直到下一次 rst 為止。這兩個 block 會各輸出 6*4bit 的數字 val_12hr 及 val_24hr，接著會被送到由 mode_display 和 sw_dis 控制的多工器，控制要輸出到 SSD 的數字。由按鈕 pb_mode 發出的訊號首先會先經由 debounce 及 one_pulse 處理成一個約 1/100s 的電位波，調整 FSM_mode 的 state。State0 按一下會變 state1，再按一下變 state0。當 FSM_mode 的 state 為零時，會選擇 12hr 的數值輸出，當 FSM_mode 的 state 為一時，會選擇 24hr 的數值輸出。當 sw_dis(右邊第一個 switch)為零時輸出秒，一時輸出時和分，如下圖。

2-bit choose

val-dis	[0]	[1]	[2]	[3]
00	4'd0	4'd0	val-12h[1]	val-12h[0]
01	val-12h[5]	val-12h[4]	val-12h[3]	val-12h[2]
10	4'd0	4'd0	val-24h[1]	val-24h[0]
11	val-24h[5]	val-24h[4]	val-24h[3]	val-24h[2]

最後多工器會輸出 4*4bit 的訊號，最後被 monitor_ctl 和 display 模組轉化成 SSD 接受的模式輸出。

1_bit upcounter



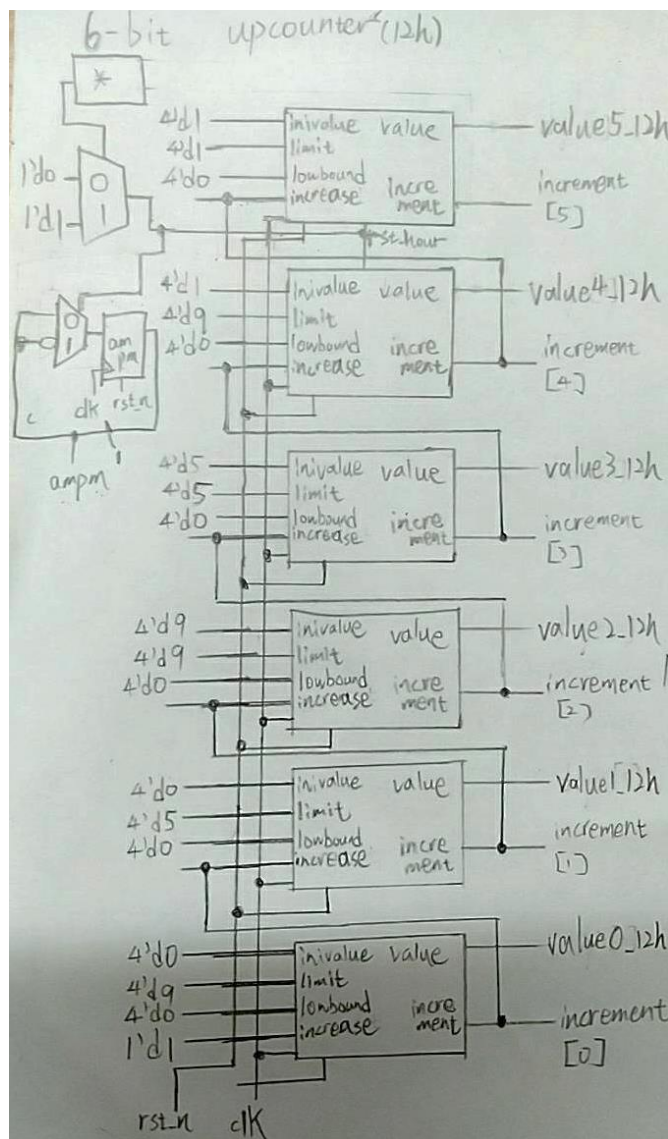
由於本題會用到多位元向上計數器，所以向上計數器應該要具有本位元上數和判斷進位的功能。所以會有一個 **increase** 是用來偵測要不要使本位元加一，當前位元為九時或是功能上有需要，就會輸入高電位到 **increase**，啟動上加功能，使得此位元的值加一，輸出至 **value_next**。而當 **increase** 為低電位時，則會保持原本的值。而判斷是不是該進位的時候，就由 **increase** 及值為 **limit** 時判斷，假設兩項條件皆成立時，**increment**(進位)為一輸出，同時本位元會輸入

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。

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

Upcounter_12h 及 upcounter_24h 模組



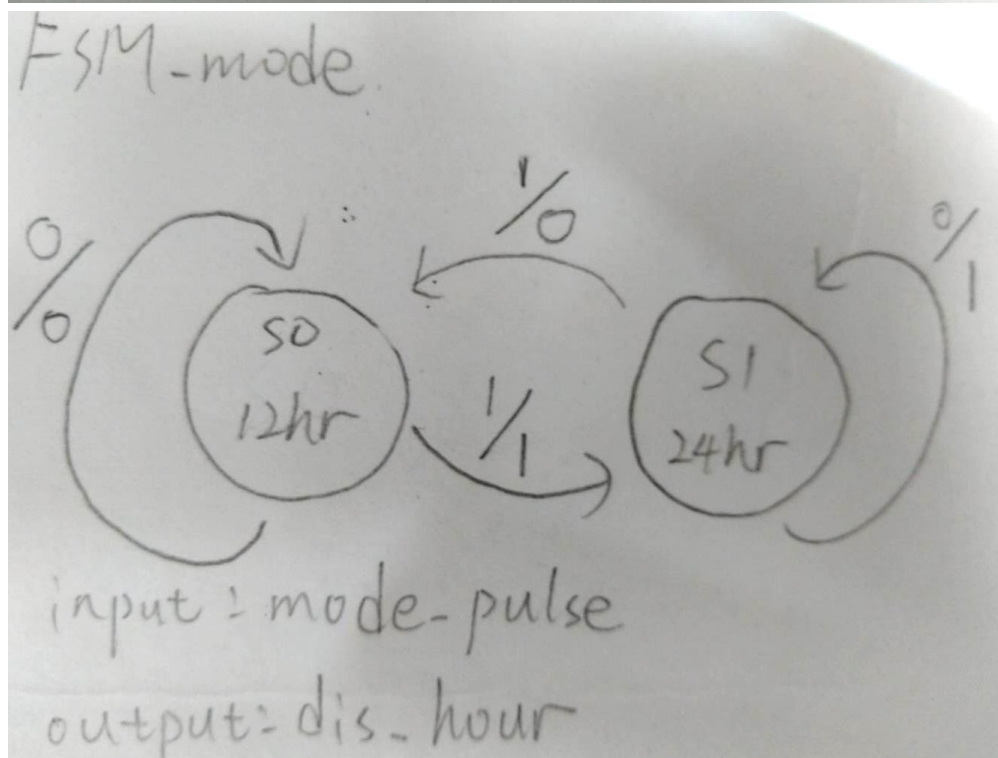
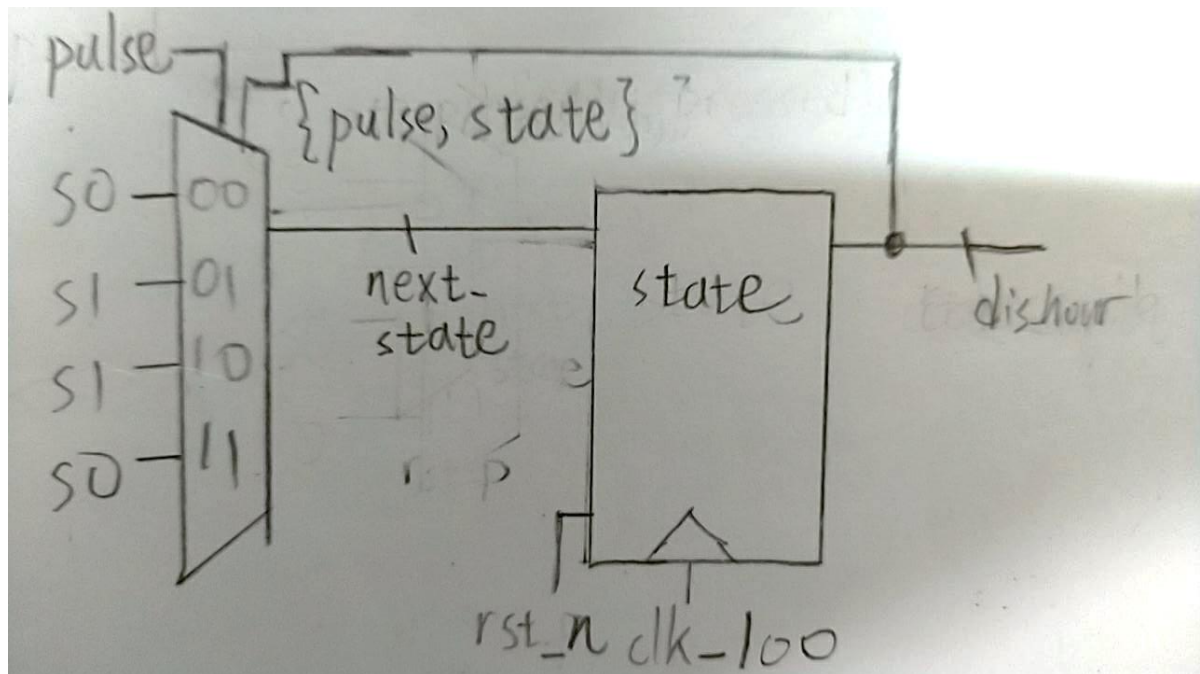
將六個 upcounter 接在一起，並且設立好 limit(12h 是 195959, 24h 是 295959)，lowbound(皆為零)，及 invalue(12h 是 115900，24h 是 235900)，將每個計數器的 increment 接到下一個計數器的 increase。

至於小時的歸零則不太一樣，12h 的要能在 12 點時轉換到 1 點，但是不能直接將小時的 limit 設為 12，這樣計數器在兩點時會直接跑到十點。所以計數器

中的 `time_rst` 就派上用場了，將時鐘的輸出在外面用比較器(用星號表示)比較。如果時鐘計數器輸出的值分別為 115959(為時歸零前的最後一秒)，則在下一個正緣時把 01 分別輸到由上往下數第一第二個計數器中。24h 的計數器也以此類推(比較數值為 235959)

而在歸零的同時，比較器也傳輸訊息到掌管 ampm 顯示的暫存器(1 代表 am，0 代表 pm，rst 時預設為 0 (pm))，使暫存器的訊號存入與原本相反的信號，切換 ampm。

FSM_mode



pb_mode 的作用是在按下 rst 時會使得 SSD 顯示十二小時制的值，按一下後會切換到二十四小時制，再按一下就變回十二小時制。

首先將 pb_mode 輸入的信號轉成一個脈衝(原理同上個 lab)，輸進去 FSM_mode。當訊號傳到 FSM_mode 後，下一個 state 會根據現有的 state 及 input 來做調整。State 有分為 12h 及 24h。當 12h 時收到高電位訊號，它會跳到 24h 狀態。當 12h 狀態時收到低電位訊號，它會仍在 12h 狀態。當 24h 狀態時收到高電位訊號，它會跳到 24h 狀態。當 24h 狀態時收到低電位訊號，它會仍在 24h 狀態。而在 12h 訊號的 FSM_mode 會輸出 dis_hour 值零，dis_hour 訊號傳到後面的多工器，會選擇要用 counter_12h 的數值，而 dis_hour 為一時則選擇 counter_24h 的數值。

3. I/O pin assignment

	變數接收處	晶片 I/O 點	描述
輸入	clk_cr	W5	石英震盪器時脈輸入
	sw_rst	V16	重置時鐘
	sw_dis	V17	切換時分/秒顯示
	pb_mode	W19	切換十二/二十四小時制
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	

Lab6.2 For the date functions in clock (no leap year), we have the following functions:

- o Day (Jan/March/May/July/Aug/Oct/Dec: 1-31, Feb: 28, Apr/June/Sept/Nov: 30),
- o Month (1-12)
- o Year (00-99).

1.Design specification:

功能:年月日計日器

輸入: sw_rst (重置計數器)

sw_dis (切換顯示年/月日)

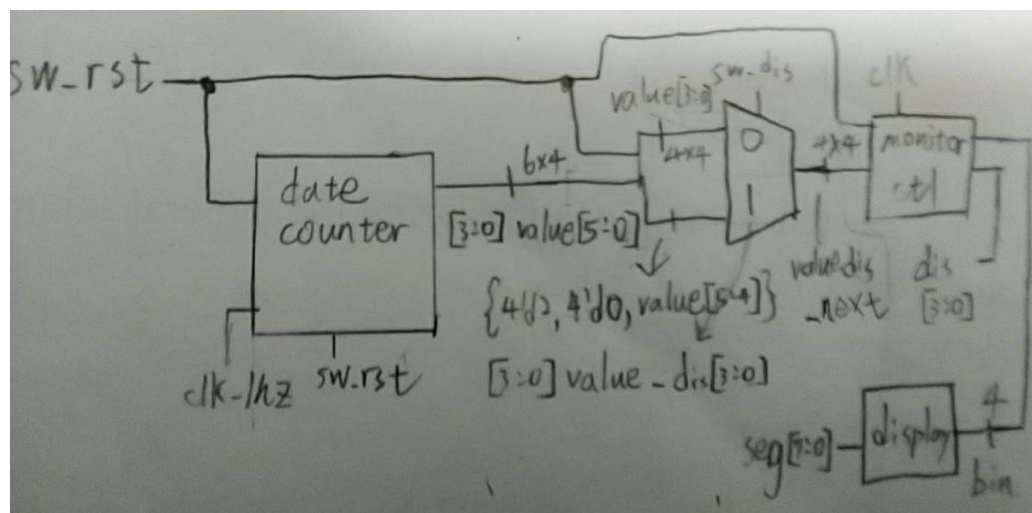
clk_cr (石英震盪器輸入)

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)

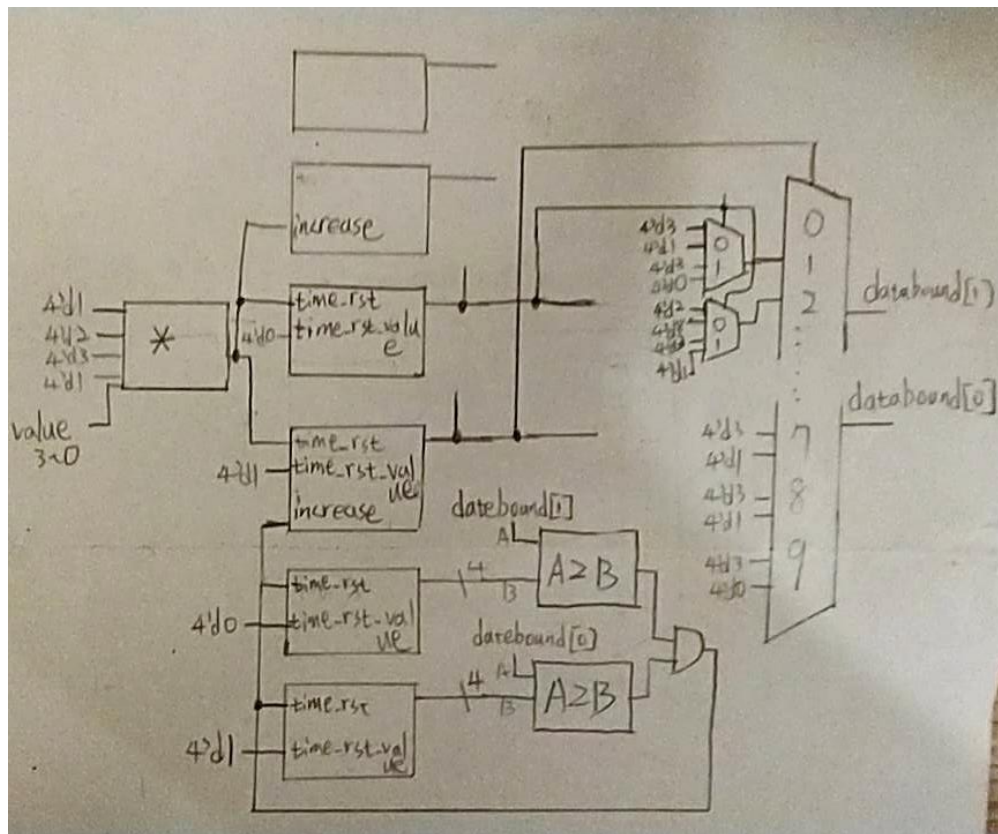
註:計時速度為原本的二十倍，便於檢查。

2.Design Implementation



在 sw_rst 向下撥又向上撥後，datecounter 會重新開始計年月日，其輸出六個 BCD 數字到多工器，此多工器由 sw_dis 控制，當 sw_dis 開關為下時，其輸出月日([3:0]value[3:0])到 monitor_ctl 再到 display，轉換成符合 SSD 顯示輸出的格式。sw_dis 朝上時，則輸出{4'd2,4'd0,[3:0]value[5:4]}的格式到 monitor_ctl。

Datecounter:



這題計數器的大致概念和第一題相像(為了圖像整潔,所以 `clk`, `rst_n` 等上一題有講述過即不再畫),只是多了要判斷月歸零和日歸零的問題。月歸零的判斷方式和上一題的小時歸零很像,是藉由判斷現在是幾月幾號來判斷(`value0` 到 `value3` 的值是否為十二月三十一日),如是則輸出信號到月 `counter` 的 `time_rst` 及年的第一位元的 `increase`,使月份歸至 `01`,年分往上加一。但是日歸零就不一樣了,由於不同的月份會有不同的日數,所以不能直接用一定值比較。所以必須使用一個多工器,對於每個月分都會輸出其對應的日期到比較器,和現在計數器輸出的日期比較,如現在值大於等於 `databound`(現在月份的天數),那就日期重置到 `01`,月份加一。

3. I/O pin assignment

	變數接收處	晶片 I/O 點	描述
輸入	clk_cr	W5	石英震盪器時脈輸入
	sw_rst	V16	重置時鐘
	sw_dis	V17	切換年/月日顯示
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	

Lab6.3 (Bonus) Add the time display support of both AM/PM and 24-hour, and the leap year support. (The year will start from 2000 to 2200 and use 4 SSDs to display.)

1.Design specification:

功能:可顯示年月日時分秒及切換十二/二四小時的時鐘

輸入: pb mode (切換 12 時/24 時模式)

sw rst (重置計數器)

sw_freq(切換 1hz/250000hz 頻率)

[1:0]sw dis (切換顯示年/月日/時分/秒)

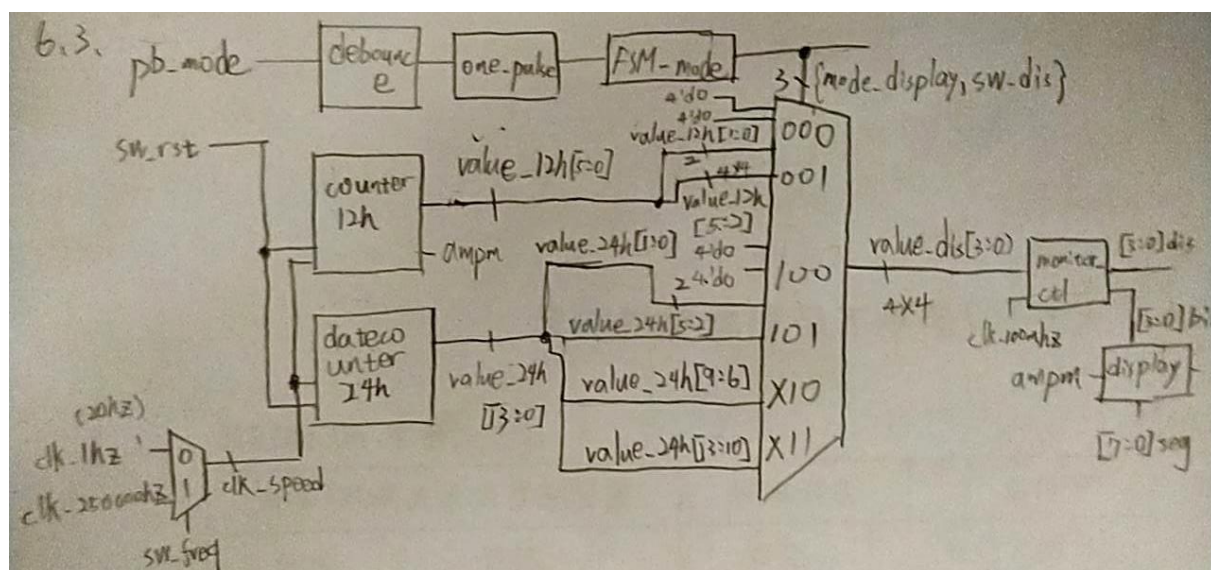
clk cr (石英震盪器輸入)

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 (控制所有 led 亮暗)

2. System implementation



這一題可以說是第一題和第二題的結合與放大版，保留了第一題的 `counter_12h` 和 `FSM_mode`，並且將 `datecounter_24h` 加上了計年月日的功能。然後後面的多工器選擇 `sw_dis` 由 1bit 變成 2bit，當 `mode_display` 為一時，輸出年月日及 `datecounter24h` 的時分秒，為零時則輸出年月日及 `counter12h` 的時分秒。

而後面的 `sw_dis`(右至左數第二、第一個 `switch`)，當輸入為 `11` 時，會輸出年分，`10` 時輸出月日，`01` 時輸出時分，`00` 時則輸出秒，而輸出的值會被 `monitor_ctl` 及 `display` 轉換成 SSD 可輸出的格式。

Datecounter24h 中日計時器被加在時計時之上，時對日的進位也是比較時的兩位數字，閏年的計算則是用年份的每一位數取餘數比較的方式決定，先由**400**年決定，如果第三位數除四餘零，且第四位為二，後兩位為零就潤。如果上述條件不成立，那如後兩位為零就潤。如果前述條件不成立，那第二位數為雙位數且第一位數被四整除或第二位數為奇位數且第一位數被四除於二為潤，

其餘不潤。而閏年的訊號則會被輸到日期被月份選擇的大多工器前尾數為二月份的選擇器做選擇。

3. I/O pin assignment

	變數接收處	晶片 I/O 點	描述
輸入	clk_cr	W5	石英震盪器時脈輸入
	sw_freq	W17	切換 1hz/250000hz 頻率
	sw_rst	W16	重置時鐘
	sw_dis[1]	V16	切換年/月日/時分/秒顯示
	sw_dis[0]	V17	
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	

心得及討論:

這一次的 lab 很複雜，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使用前面的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。