Lab7_1

Design Specification

✓ For a stopwatch:

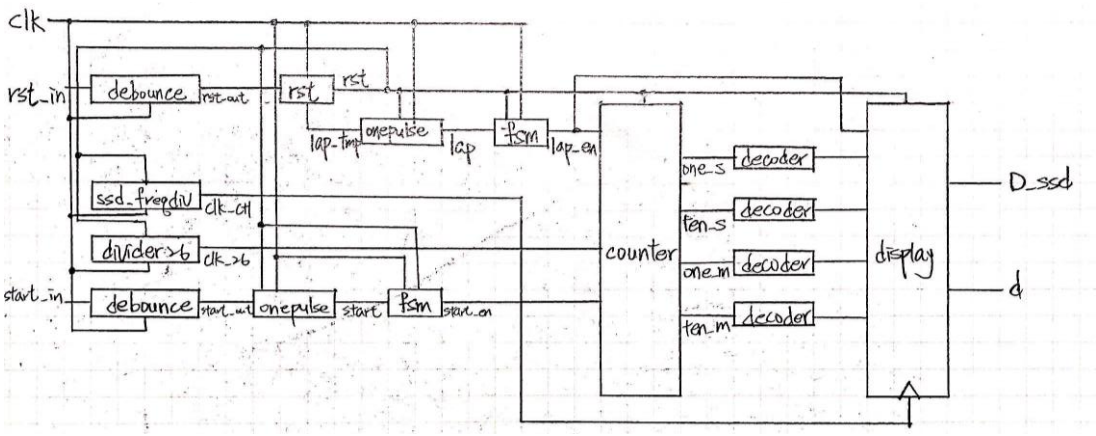
Input:

start_in // 開始倒數
rst_n // control rst_n button
clk
clk_c // 控制秒的頻率(非必要，因此無在 block diagram 中顯示)

Output:

D_ssd[7:0] // 7-segment display
d[3:0]

✓ Draw the block diagram of the design.



Design Implementation

✓ 本題由 ssd_freqdiv、divider26、debounce、rst、onepulse、fsm、decoder、counter、display 九個 module 組成。由於這次是用按鈕控制，因此有 debounce、rst、onepulse 三個 module。Debounce 的功能是產生穩定的波型；rst 的功能是區分長按與短按；onepulse 的功能是製造一個 button 的訊號，讓按下 button 後可以一直保持在下個 state。以上皆在之前的 lab 有說明過，因此不再贅述。ssd_freqdiv、divider26 則是做出需要的頻率；decoder 是將 binary 轉為 BCD 以呈現再 7-segment display 上；counter 的功能為計算秒與分，主要組成為 BCD upcounter；fsm 則是控制長按短按的狀態，以上在之前的 lab 皆有做過，因此此次重點將放在 display 中的 lab。

✓ Display

這次實驗重點為碼錶分圈。Input 的 lab 和 rst_n 為同一個按鈕控制，其中 lab 為短按。和之前的 display 不同之處為 lab7 的 display 中多加了一個 mux，控制輸出為分圈的數字還是正在數的數字。若 lab = 1，則輸出分圈數字；lab = 0，則輸出正在數的數字。每當 lab 再次為 1 時，分圈的數字皆會改變。

✓ I/O pin

I/O	D_ssd[7]	D_ssd[6]	D_ssd[5]	D_ssd[4]	D_ssd[3]	D_ssd[2]	D_ssd[1]	D_ssd[0]
VOC	W7	W6	U8	V8	U5	V5	U7	V7
I/O	d[3]	d[2]	d[1]	d[0]	clk	rst_n	start_in	
VOC	W4	V4	U4	U2	W5	W19	T17	

Lab7_2

Design Specification

✓ For a downcounter:

Input: clk,

rst_n,

start, // 開始倒數

pause, // 暫停倒數

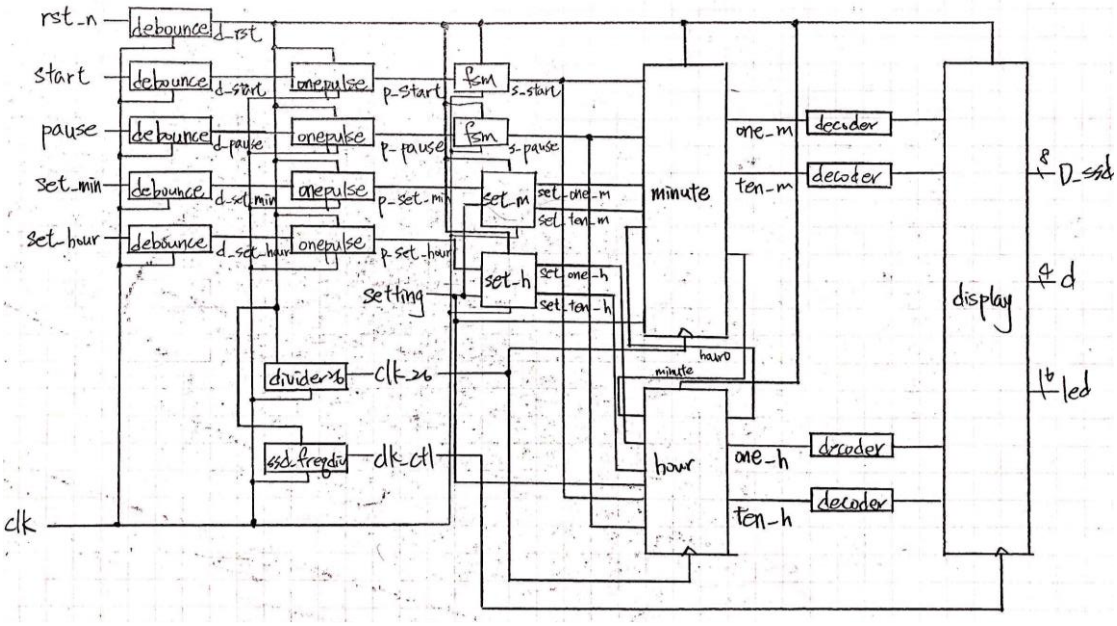
setting, // 允許設置時間

set_min, // 設置分

set_hour // 設置時

Output: D_ssd[7:0], d[3:0]

✓ Draw the block diagram of the design.

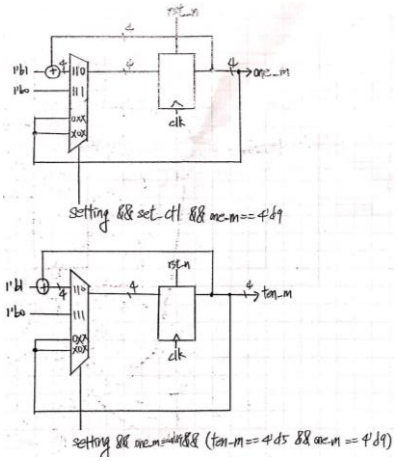


Design Implementation

✓ 此設計類似先前做過的時鐘與倒數計時器的結合，其中的 minute 與 hour 類似 lab6 中的時鐘，但為 down counter。本次的重點為時間的設置，因此以下將說明 set_m、set_h 的設計。

✓ set_m、set_h

其原理即為 up counter，當 setting = 1 時，允許設置時間，藉由 set_ctl 的按鈕，每按一下時間變加一。



上圖為分的設置，若是時(set_h)只要將進位的條件更改即可。

set_m:

MUX 條件			MUX output
setting	set_ctl	one_m == 4'd9	one_m
1	1	0	one_m + 1
1	1	1	0
0	X	X	one_m
X	0	X	one_m

MUX 條件			MUX output
setting	one_m == 4'd9	one_m == 4'd9 && ten_m == 4'd5	ten_m
1	1	0	ten_m + 1
1	1	1	0
0	X	X	ten_m
X	0	X	ten_m

set_h:

MUX 條件			MUX output
setting	set_ctl	one_h == 4'd9	one_h
1	1	0	one_h + 1
1	1	1	0
0	X	X	one_h
X	0	X	one_h

MUX 條件			MUX output
setting	one_h == 4'd9	one_h == 4'd3 && ten_h == 4'd2	ten_h
1	1	0	ten_h + 1
1	0	1	0
0	X	X	ten_h
X	0	X	ten_h

其 output 為設定好的時間，將其作為 minute 與 hour 的 input 即為倒數計時器的時間設定。

✓ I/O pin

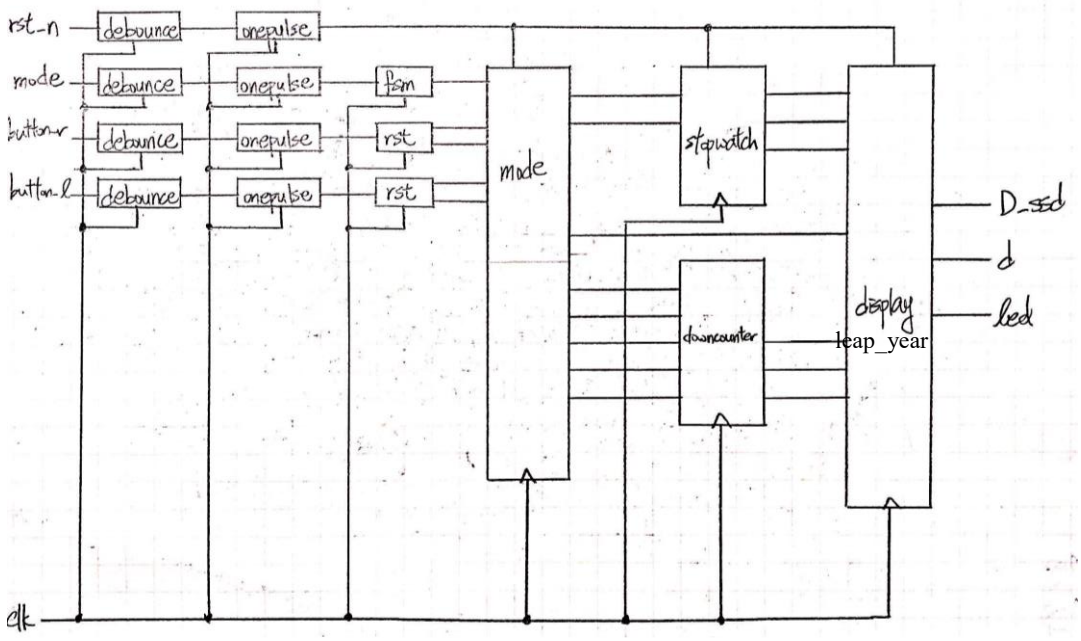
I/O	led[15]	led[14]	led[13]	led[12]	led[11]	led[10]	led[9]	led[8]
VOC	L1	P1	N3	P3	U3	W3	V3	V13
I/O	led[7]	led[6]	led[5]	led[4]	led[3]	led[2]	led[1]	led[0]
VOC	V14	U14	U15	W18	V19	U19	E19	U16
I/O	D_ssd[7]	D_ssd[6]	D_ssd[5]	D_ssd[4]	D_ssd[3]	D_ssd[2]	D_ssd[1]	D_ssd[0]
VOC	W7	W6	U8	V8	U5	V5	U7	V7
I/O	d[3]	d[2]	d[1]	d[0]	clk	start	pause	set_min
VOC	W4	V4	U4	U2	W5	T18	U17	T17
I/O	set_hour	setting	rst_n					
VOC	W19	V17	U18					

Lab7_3

Design Specification

✓ Input: clk,
rst_n,
mode, // 模式切換
button_r,
button_l
Output: D_ssd[7:0], d[3:0], led[15:0]

✓ Draw the block diagram of the design.



Design Implementation

✓ 第三題為第一題與第二題的結合，其中的 mode 功能為控制模式與按鈕的作用。其餘 module 皆與第一題與第二題差不多，因此不再討論。

✓ mode
input:
clk,
rst_n,
mode,
button_r_l, // 右鍵長按
button_r_s, // 右鍵短按
button_l_l, // 左鍵長按
button_l_s, // 左鍵短按
output:
lap,
start_in,
start,
pause,
setting,
set_min,
set_hour
一開始的時候 output 皆為 0，此處我設計當 mode = 0 時為 stop watch 模式，mode = 1 時為 down counter 模式。

mode	setting	set_min	set_hour	start	pause	lap	start_in
0	0	0	0	0	0	button_r_s	button_l_s
1	0	0	0	button_r_s	button_l_s	0	0
1	1	button_r_s	button_l_s	0	0	0	0

其中當 mode = 1 時，setting <= button_r_l。藉由 mode 的控制，此處將決定按鈕所輸出對應到的功能為何。

✓ I/O pin

I/O	led[15]	led[14]	led[13]	led[12]	led[11]	led[10]	led[9]	led[8]
VOC	L1	P1	N3	P3	U3	W3	V3	V13
I/O	led[7]	led[6]	led[5]	led[4]	led[3]	led[2]	led[1]	led[0]
VOC	V14	U14	U15	W18	V19	U19	E19	U16
I/O	D_ssd[7]	D_ssd[6]	D_ssd[5]	D_ssd[4]	D_ssd[3]	D_ssd[2]	D_ssd[1]	D_ssd[0]
VOC	W7	W6	U8	V8	U5	V5	U7	V7
I/O	d[3]	d[2]	d[1]	d[0]	clk	rst_n	mode	button_r
VOC	W4	V4	U4	U2	W5	T18	U18	T17
I/O	button_l							
VOC	W19							

Discussion

在第一題的 lap 設置中，我原本想在 counter 中先做好分圈，再將其輸出到 display 中顯示，但遇到的問題是因為會延遲一個 clock 所以分圈的值會多一秒，且這樣要多接線，造成不必要的浪費。因此最後我將其做在 display 中，當 lap = 1 時即控制顯示為分圈時的數字。

第二題為將時鐘改為倒數，並可以自己設定時間。在這題沒遇到太大的問題，主要是最後當歸零時要全部的 led 燈全亮，由於我原本的控制跟時與分的設定寫在一起，一直無法讓他在正確的時間亮，之後將其單獨分開判斷便成功了。

第三題需要把第一題與第二題中的 debounce、onepulse、fsm、rst 接在外面再做為 mode 的 input，無法直接藉由判斷 mode 的值來決定按鍵功能。此處須注意 mode 的 output 與 down counter、stop watch 的關聯。

Conclusion

這次的實驗幾乎都是重組之前做過的 module 並加以小修改，比較困難的是第三題不能直接用 mode 的值來決定按鍵功能，跟軟體的想法很不一樣。

