

Lab 07

105060016 謝承儒

一.Lab07-1：有凍結功能的分秒計時器

(凍結:表面上不動，實際上內部仍然還在計時。)

1. Design Specification :

- A. Input : clk //輸入的頻率(100MHz)
 - rst //當=1 時，使數字回到 00 並暫停
 - pb_s //暫停/開始按鈕
 - pb_f //凍結/重製按鈕
- B. Output : [7:0] display //七段顯示器的顯示碼
 - [3:0] display_c //控制 4 個顯示器中哪個會改變
- C. Wire : clk_1 //除頻後的頻率(1Hz)
 - clk_100 //除頻後的頻率(100Hz)
 - [1:0] ssd_ctl //從除頻器輸出，解碼後為 display_c
 - pb_s/f_de //經過 Debounce 後的 pb_s/f
 - one_pulse_s/f //經過 One_pulse 處理後，產生為期 1 週期的
 - freeze_en、rst_f、count_en //從 FSM 輸出，分別代表是否凍結/重置/計時
 - co_s1/ co_min / co_m1 //代表進位，分別是秒十位數、分個位數、分十位數
 - [3:0] sec_s0、sec_s1、min_m0、min_m1 //秒和分的個位數、十位數
 - [3:0] d0、d1、d2、d3 //經過是否凍結的區塊後，準備顯示在顯示器的值
 - [7:0] display0、display1、display2、display3 //d 經過解碼後得到的顯示碼

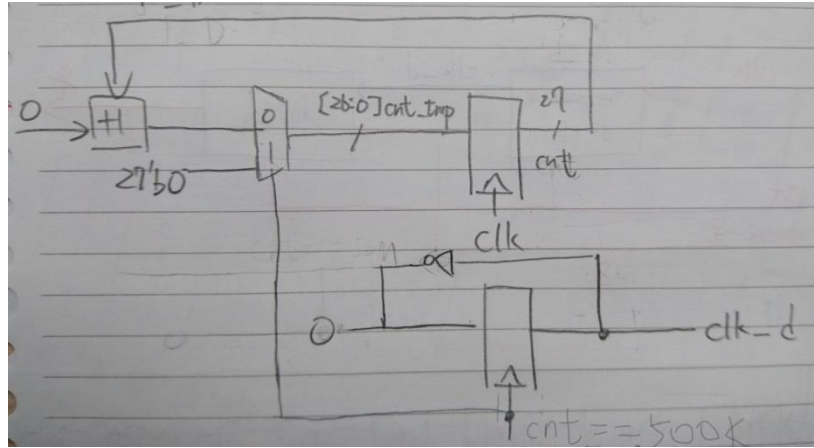


圖 3 Lab07-1 的除頻器(100MHz→100Hz)

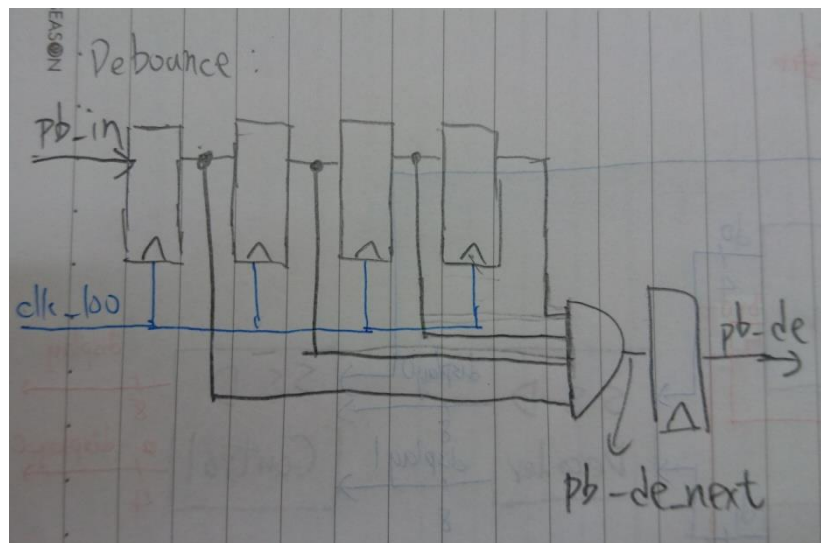


圖 4 Lab07-1 的 Debounce

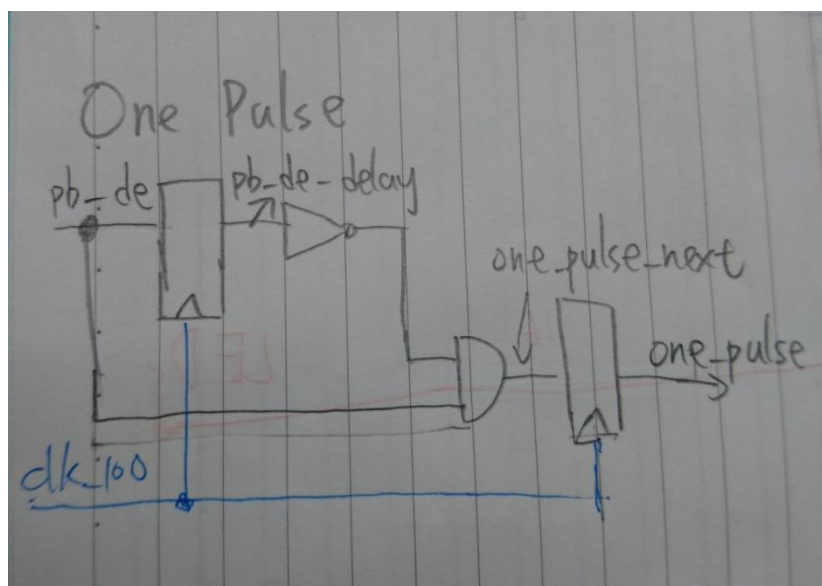


圖 5 Lab07-1 的 One pulse

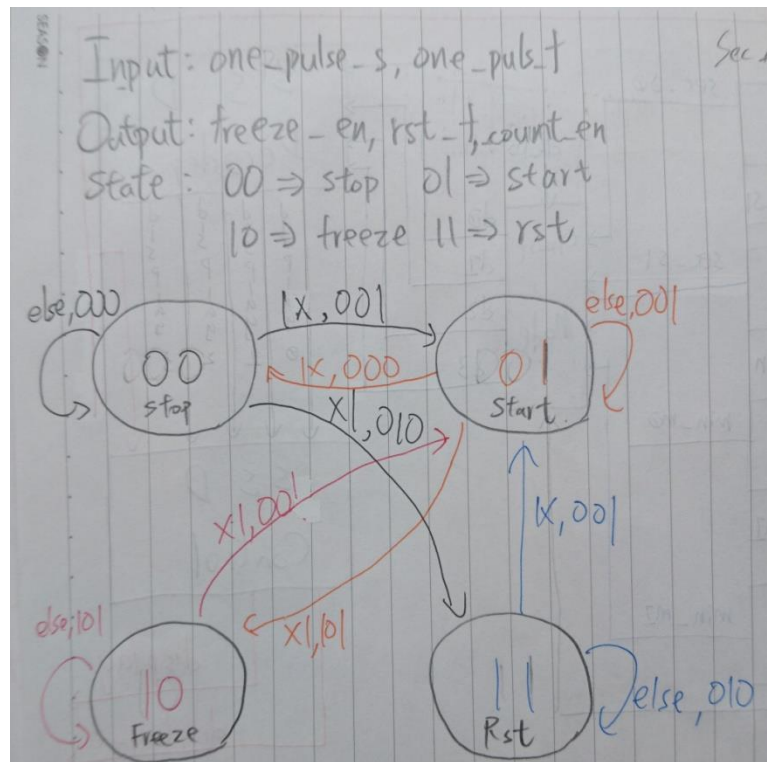


圖 6 Lab07-1 的 FSM

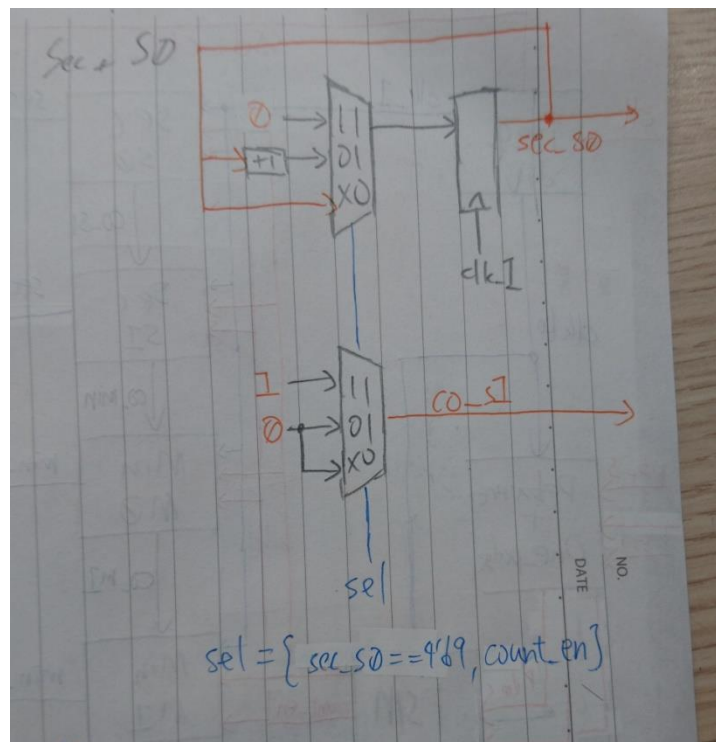


圖 7 Lab07-1 的 sec_s0 的 up counter(個位數)

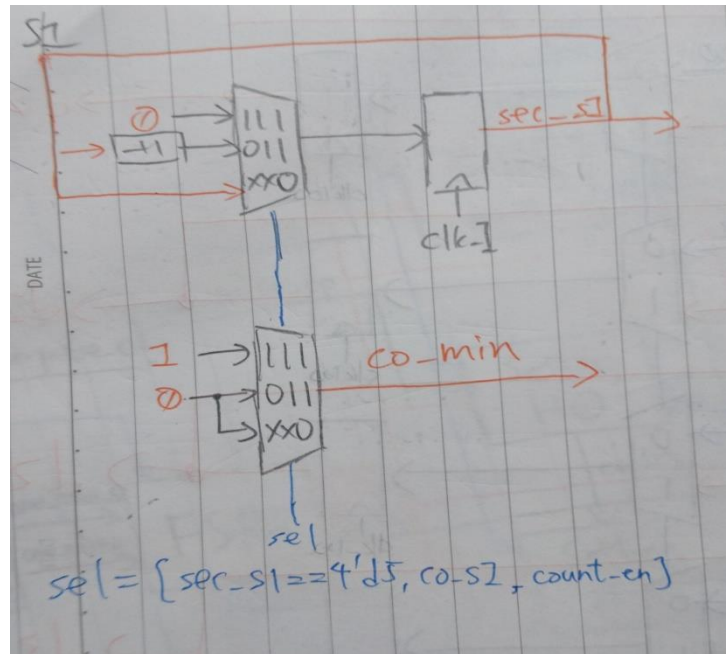


圖 8 Lab07-1 的 `sec_s1` 的 up counter(十位數)

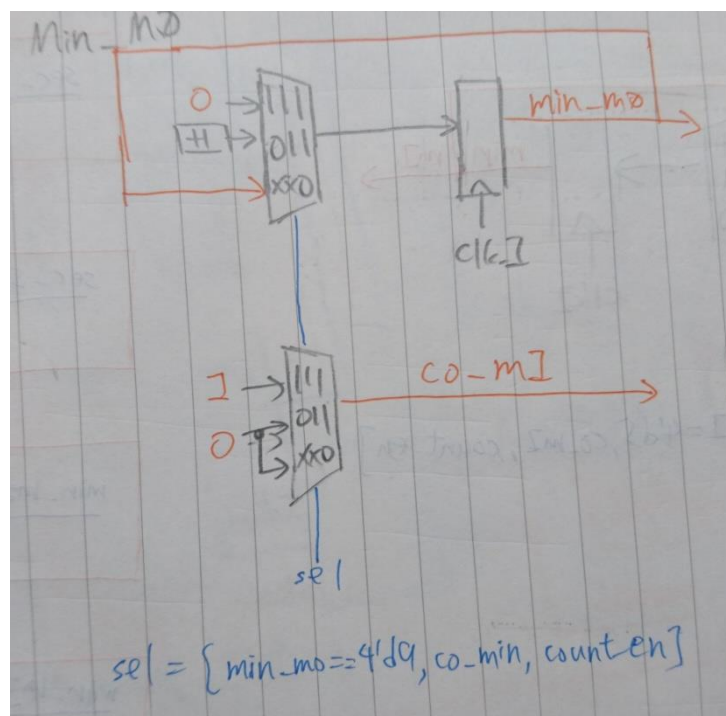


圖 9 Lab07-1 的 `min_m0` 的 up counter(個位數)

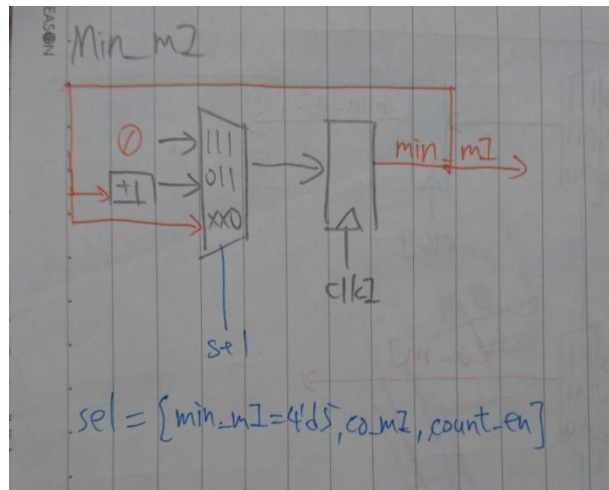


圖 10 Lab07-1 的 min_m0 的 up counter(十位數)

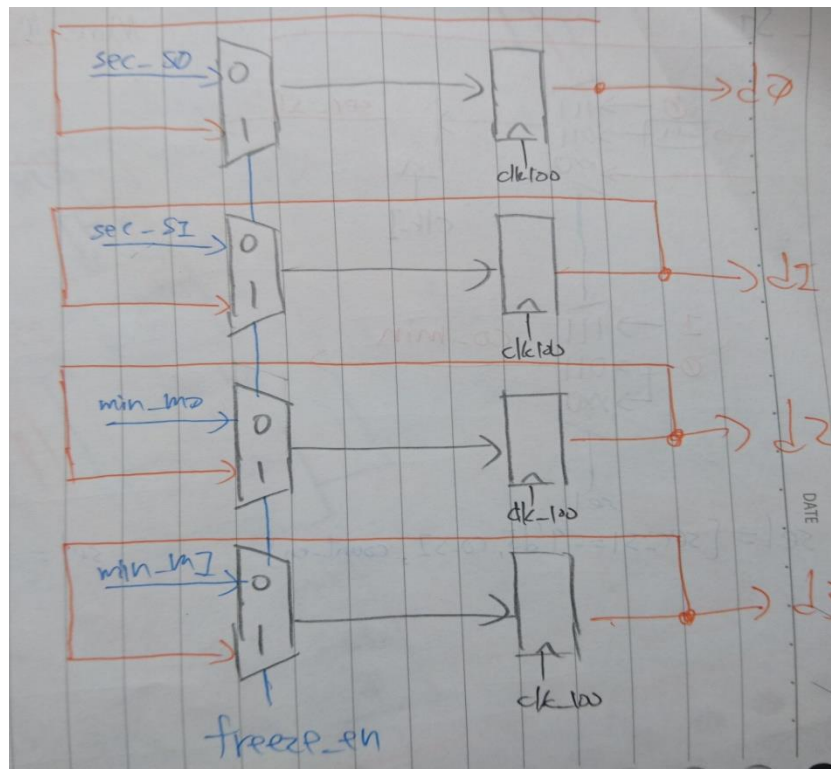


圖 11 Lab07-1 的 choose_mode

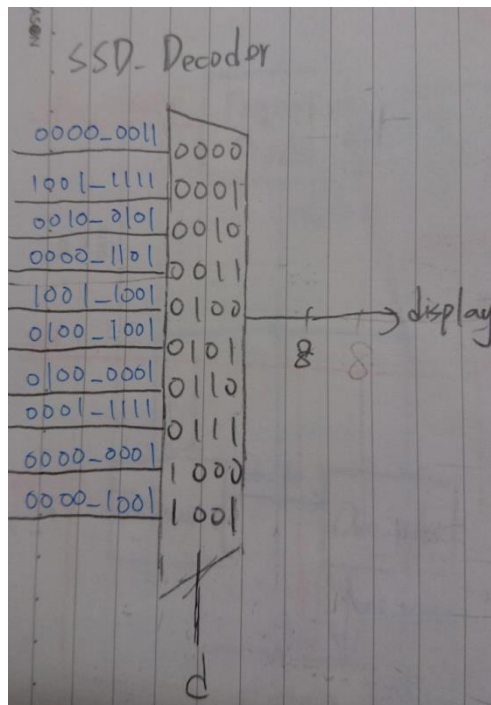


圖 12 Lab07-1 的 SSD Decoder

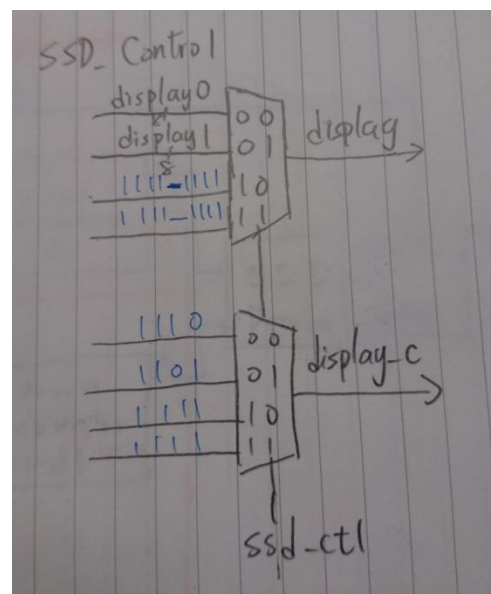


圖 13 Lab07-1 的 SSD Control

B. Pin assignment :

a. Input :

- 1) clk = W5
- 2) rst = R2
- 3) pb_f = W19
- 4) pb_s = T17

b. Output :

- 1) display[0] = V7
- 2) display[1] = U7
- 3) display[2] = V5
- 4) display[3] = U5
- 5) display[4] = V8
- 6) display[5] = U8
- 7) display[6] = W6
- 8) display[7] = W7
- 9) display_c[0] = U2
- 10) display_c[1] = U4
- 11) display_c[2] = V4
- 12) display_c[3] = W4

3. Discussion :

A. 整體運作過程 :

- a. 將 clk 輸入進除頻器(freq_div)，得到 1Hz 和 100Hz 的新 clk。1Hz 的用來驅動 counter；100Hz 的用來驅動 Debounce、One_pulse、FSM。
此外，輸出 ssd_ctl 來做為待會 SSD_Control 的其中一個 Input。
- b. 將經過處理後的 one_pulse_s、one_pulse_f 輸入到 FSM，得到 freeze_en、rst_f、count_en，用來在後面的 counter 中作為條件來達到不同狀態。
- c. 將 4 個 counter 的值(sec_s0/1、min_m0/1)輸入到 choose_mode，並利用 freeze_en 來判斷是否處於凍結狀態，然後輸出要顯示出來的值(d0~3)。
- d. 最後，把 d0~3 解碼成顯示碼(display0~3)，再加上 ssd_ctl 來控制顯示板(display_c)和將每個值逐一輸出(display)。

B. 各 block 的構想：

a. **除頻器**:用來製造 1Hz、100Hz

跟之前的 Lab 相同，用輸入的 `clk(clk_c)`來驅動 27-bit 的 Counter(其值為[26:0]cnt)。在 `cnt=50M` 時，讓 `clk_d = ~clk_d` 且 `cnt` 歸零，達到 1Hz 的效果。

而將條件改成 `cnt=500K`，便可以製造出 100Hz。

除此之外，將其中的兩位輸出(ssd_ctl)，此次選擇 `cnt[15:14]`。

b. **Debounce**：消除不穩定的震盪

利用一個 4-bit 的 Shift register，每次 `clk` 都將現在按鈕的狀態輸入 (`pb_s/f`)，若 4 個都為 1，則輸出 `pb_s/f_de=1`。因為是由電腦模擬出來的，不會有不穩定的震盪。

c. **One Pulse**：製造只會為期 1 週期的 1=>只會觸發一次效果

將剛得到的 `pb_s/f_de` 輸入進 Flip Flop(FF)，只有當 `clk posedge` 時，FF 才會打開讓值通過，因此 FF 後的值(`pb_de_delay`)會和進來前的(`pb_de`)有 1clk 的誤差。

將 `~(pb_de_delay)`和 `pb_de` 做 and，便可以製造出只會為期 1 週期的 1。

d. **FSM**:用來決定暫停、計時、凍結、重置 4 種狀態

因為有 4 種狀態，所以 `state` 是 2-bit，00 為暫停、01 為計時、10 為凍結、11 為重置。

狀態的變換是經由兩個按鈕來控制；其中一個(`pb_s`)控制暫停還是計時；另一個則是凍結/重置(`pb_f`)，按下時，若為暫停狀態則會重置，若為計時狀態則會凍結。

e. **sec/min up counter**

基本上只是一個簡單的 BCD up counter，只是需要注意秒和分都是六十進位制，當為 59 時，下一個則是 00。

f. **choose_mode**: 是否處於凍結狀態

把剛剛在 FSM 得到的 `freeze_en` 作為選擇依據。

當 `freeze_en=0` 時，就代表為一般的計時，直接將輸入的各值 (`sec_s0/1`、`min_m0/1`)輸出就好。

當 `freeze_en=1` 時，就代表是處於凍結狀態，把上個瞬間輸出的值(`d1~3`)再次輸出。

g. **SSD Decoder**：將數字轉換成顯示碼

將 `Choose_mode` 得到的值(`D0`、`D1`)輸入，作為多工器的依據(在 code 裡可用 case 得到相同效果)，解碼成 8-bit 的顯示碼 (`display0`、`display1`)，將它們輸入到下個 SSD Control。

h. SSD Control : 決定哪塊板子的值改變

把在除頻器得到的 2-bit 的 `ssd_ctl` 輸入，作為選擇 SSD 四個顯示器的依據。

當 `ssd_ctl_en=00` 時，便將 `display0` 輸出(`display`)，並將 `display_c` 輸出為 `1110`。如此顯示器上便只有最後一位可以改變，保留其他三個的字母，其他 `ssd_ctl` 的情況也是一樣。

C. 過程中的 Bug :

一開始，在 `freeze` 的狀況下，再按一次凍結鍵應回到計時狀態，但反而會先重置後，再開始計時。把兩個按鍵的 `Debounce` 和 `One_pulse` 分開做就正常了。

二.Lab07-2：可設定時間的倒數器

1. Design Specification：

- A. Input : clk //輸入的頻率(100MHz)。
rst //當=1 時，重置時間
setting //是否在設定狀態
pb_p_hr
//設定時為加小時按鈕，倒數時為暫停/倒數
pb_s_min
//設定時為加小時按鈕，倒數時為開始倒數
- B. Output : [7:0] display //七段顯示器的顯示碼。
[3:0] display_c //決定哪個顯示器改變。
[15:0]LEDs //16 個 LED 燈
- C. Wire : clk_d //除頻後頻率(40Hz)
clk_f //除頻後頻率(100Hz)
[1:0] ssd_ctl //從除頻器輸出，解碼後為 display_c。
pb_s/p_de、one_pulse_s/p
//經過處理後的按鍵，給 FSM 相關的 block，100Hz
pb_cs/cp、one_pulse_cs/cp
//經過處理後的按鍵，給 setting 和 counter，40Hz
rst_f、de_en、set_en、light
//從 FSM 輸出、代表是否歸零、倒數、設定、發亮
[3:0] set_m0,set_m1,set_h0,set_h1
//代表設定的值
co_m1、co_hr、co_h1
//代表是否進位到下一個時間單位
[3:0] min_m0/1、hr_h0/1
//分、時的個、十位數值
br_m1、br_h0、br_h1
//代表借位
[7:0] display0、display1、display1、display1
//各個位數的顯示碼。

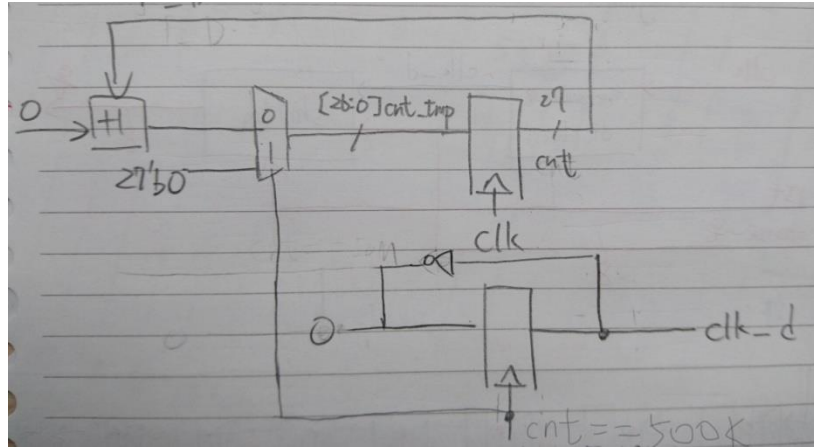


圖 16 Lab07-2 的除頻器(100MHz->100Hz)

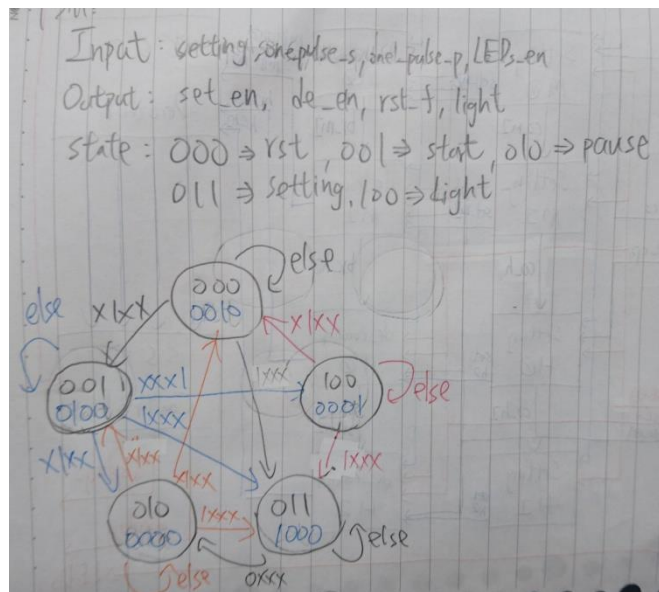


圖 17 Lab07-2 的 FSM

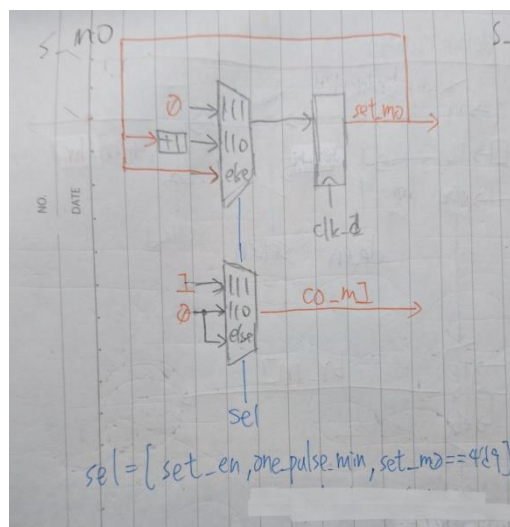


圖 18 Lab07-2 的 set_m0

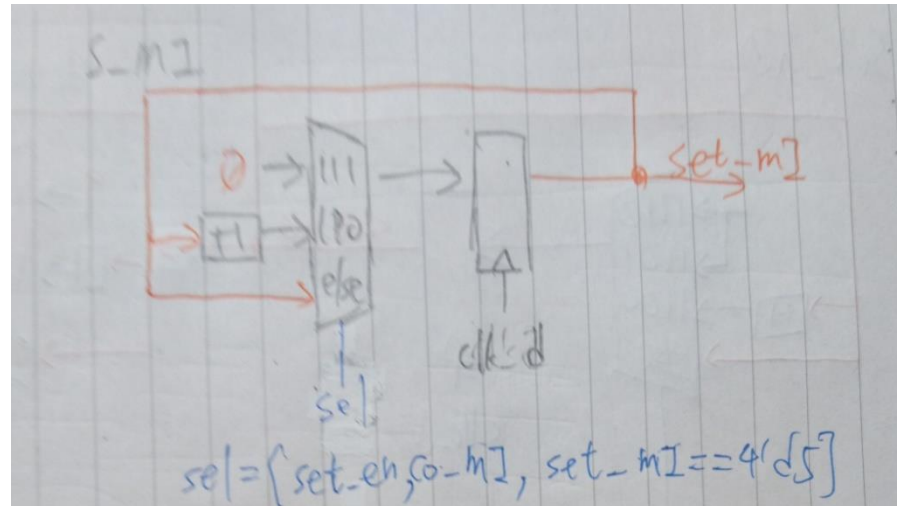


圖 19 Lab07-2 的 set_m1

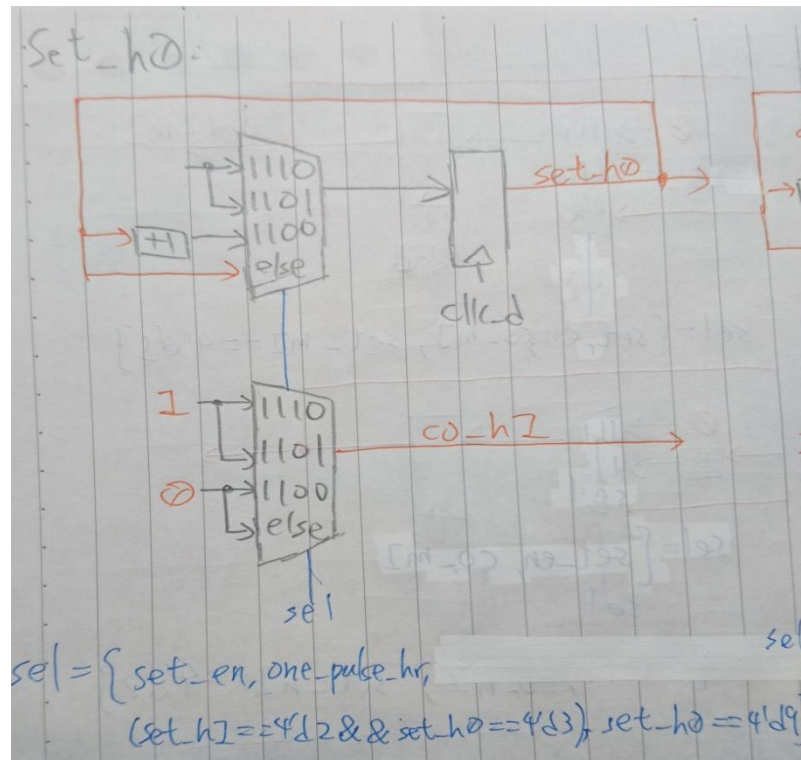


圖 20 Lab07-2 的 set_h0

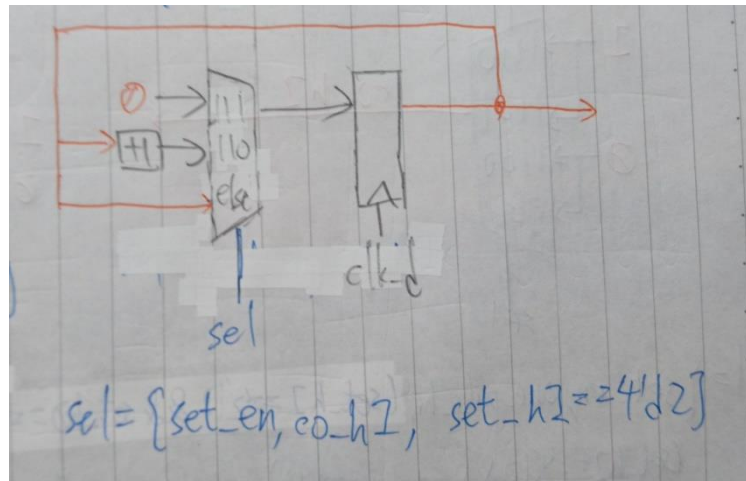


圖 21 Lab07-2 的 set_h1

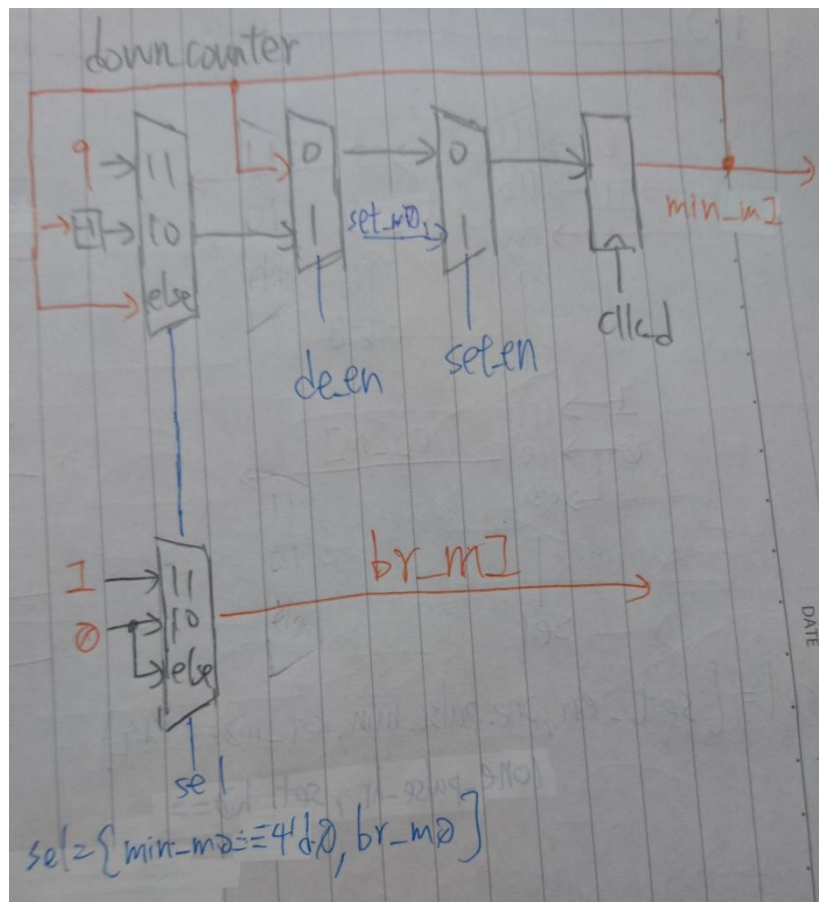


圖 22 Lab07-2 的 down counter

(各位數的結構差異不大)

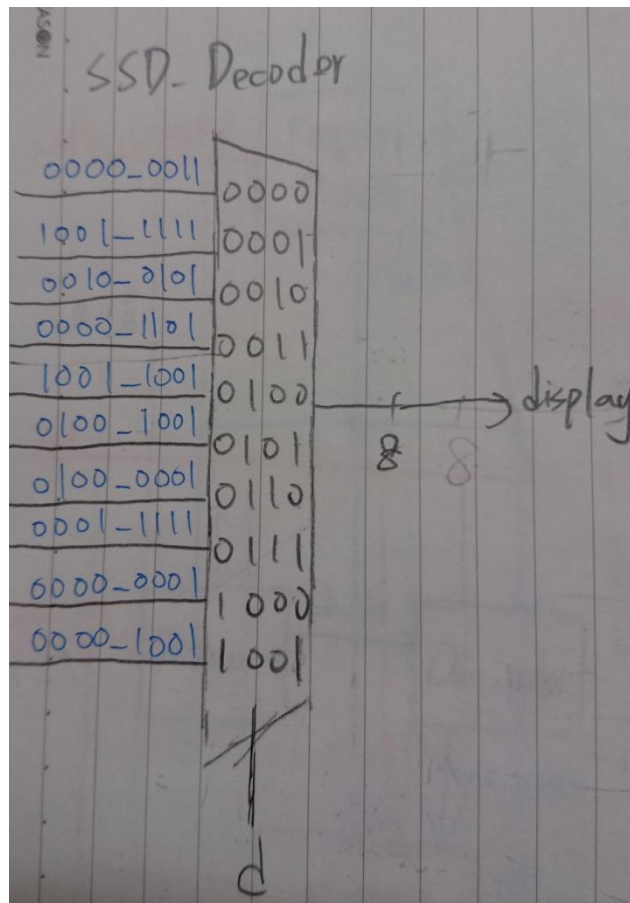


圖 23 Lab07-2 的 SSD Decoder

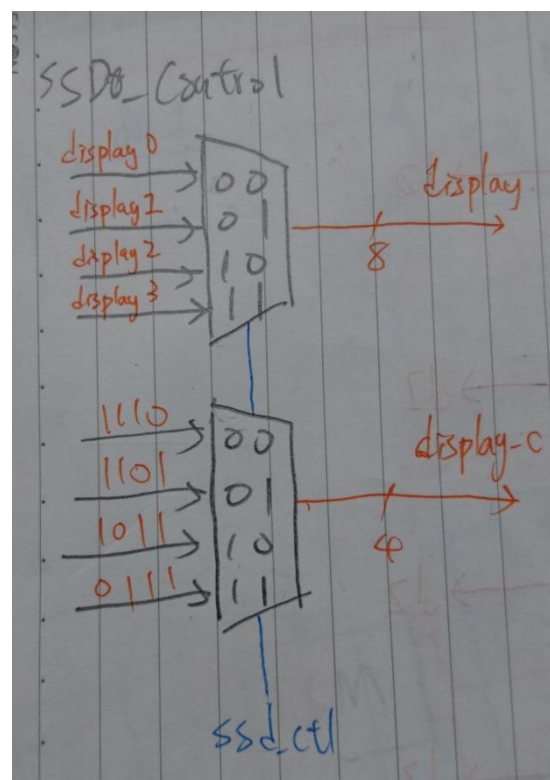


圖 24 Lab07-2 的 SSD Control

C. Pin assignment :

a. Input :

- 1) clk : W5
- 2) rst : V17
- 3) setting : R2
- 4) pb_p_hr: W19
- 5) pb_s_min : T17

b. Output :

- 1) display[0] = V7
- 2) display[1] = U7
- 3) display[2] = V5
- 4) display[3] = U5
- 5) display[4] = V8
- 6) display[5] = U8
- 7) display[6] = W6
- 8) display[7] = W7
- 9) display_c[0] = U2
- 10) display_c[1] = U4
- 11) display_c[2] = V4
- 12) display_c[3] = W4
- 13) [15:0] LED = L1~U16(全部的 LED 燈的 Pin 腳)

三. Discussion :

A. 整體運作過程 :

- a. 將 clk 輸入進除頻器(freq_div)，得到 40Hz(clk_d)和 100Hz(clk_f) 的新 clk。40Hz 的用來驅動 down counter 和 setting 相關的 block(debounce_c、ome_pulse_c)；100Hz 的用來驅動和 FSM 相關的 block(debounce、ome_pulse)。
此外，輸出 ssd_ctl 來做為待會 SSD_Control 的其中一個 Input。
- b. 把 clk_f、和經過處理的按鍵(one_pulse_s/p)輸入到 FSM 來決定各種狀態，並輸出 set_en、de_en、rst_f、light 來作為後面 block 的條件。
- c. 把 clk_d、和經過處理的按鍵(one_pulse_cs/cp)輸入到各個 setting counter block，得到各個 set 的值(set_m0、m1、h0、h1)。
- d. 把 set_m0、m1、h0、h1 分別輸入到其所屬的 down counter，得到各位數的值(min_m0/1、hr_h0/1)。
- e. 將 min_m0/1、hr_h0/1 輸入到 SSD Decoder 解碼得到顯示碼

(display0~3)。

- f. 最後，將 display0~3 輸入到 SSD Control，再加上 ssd_ctl 來控制顯示板(display_c)和把每個值逐一輸出(display)。

B. 各 block 的構想：

部分 block 可以沿用上個實驗 Lab07-1 的結構，而下面是說的是新 block。

- a. **Debounce_c+One_pulse_c**：用於設定有關的 block(Setting)和原本的不同，這裡使用的 clk 是 clk_d，而不是 clk_f。因為輸出出來的 one_pulse_cs/cp 是要用在 Setting blocks 上，加上 Setting 是和 Down counter 連動的，所以才採用相同的 clk 來避免 bug。
- b. **Setting**：設定時/分的值
這些 block 和一般的 BCD up counter 沒有太大的差別，主要是+1的機制不是由 clk 來控制，而是改為按鈕來控制。此外，當從 59 分到 00 分時，並沒有進位，其他的就沒什麼改變。
- c. **修改 FSM**：
這次我將 state 分成 5 種狀態，000 是歸零、001 是開始到數、010 是暫停、011 是設定、100 是數完然後發光。
利用 setting,one_pulse_s/p,LEDs_en 來讓 state 互相轉換。
- d. **Down counter**：倒數計時器
其實這和 up counter 相同，只是把+1 改成-1 和進位轉成退位就可以達成了。

C. 過程中的 Bug：

起初在設定值時，只要一按按鍵便會直接加到超過 9，讓顯示器上顯示 F，儘管之後把按鍵 One_pulse 化也一樣。

最後，按照同學說得把 setting 和 down counter 分開來寫 bug 就消失了。

四.Conclusion：

上次才說有上手的感覺，結果這次就被打臉了。這次花了許多時間在消除 bug，甚至為此改變整體結構才成功完成實驗。

五.Reference：

1. 老師給的實驗講義
讓我知道大概的架構是時麼。