Lab 05

105060016 謝承儒

一.Lab05-1 : 30-second down counter

1. Design Specification :

A. Input : clk_crystal　　//輸入的頻率(100MHz)
　　rst　　//當=1 時，使數字回到 30 並暫停
　　push　　//代表暫停/啟動的按鈕
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
　　[15:0]LEDs　　//當數到 00 時，所有 LED 會亮
C. Wire : clk_d　　//除頻後的頻率(1Hz)，用來驅動其他功能
　　[1:0] ssd_ctl　　//從除頻器輸出，解碼後為 display_c
　　en　　//當=0，為暫停狀態；當=1，為倒數狀態
　　decrease　　//當=0，無法再倒數(已到 00)；反之則為可以繼續數，接在個位數的 downcounter
　　br0　　//借位，相當於 decrease，接在十位數的 downcounter
　　[3:0] d0　　//個位數的值
　　[3:0] d1　　//十位數的值
　　[7:0] display0　　//個位數的顯示碼
　　[7:0] display1　　//十位數的顯示碼

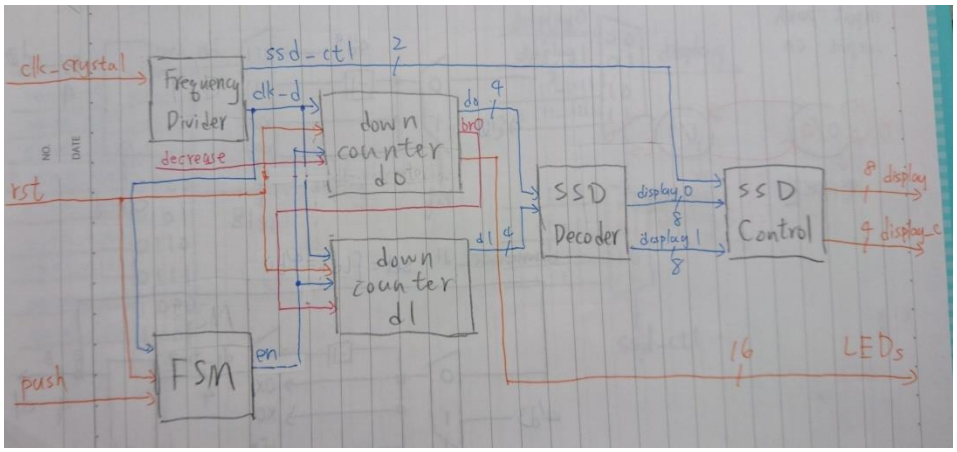

圖 1 Lab05-1 的區塊圖

2. Design Implementation :

A. Logic diagram :
   a. decrease = ~(en & (d0==4'b0000) & (d1==4'b0000))

B. Logic diagram :

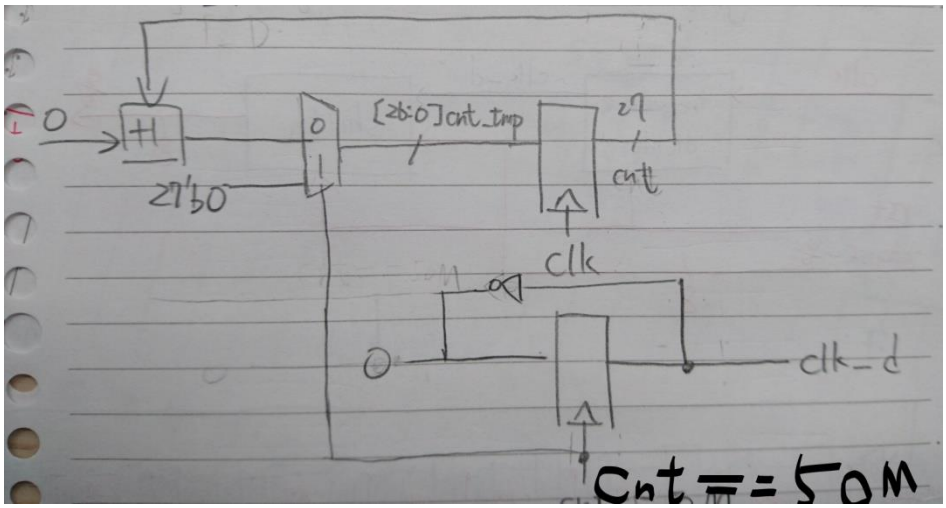

圖 2 Lab05-1 的除頻器(100MHz->1Hz)

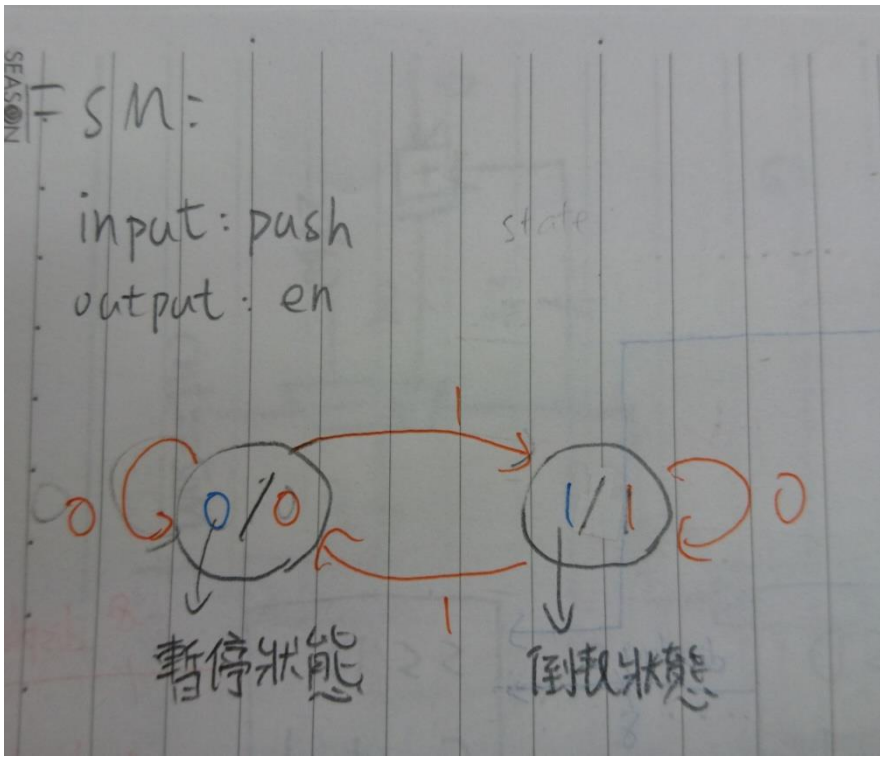

圖 3 Lab05-1 的 FSM

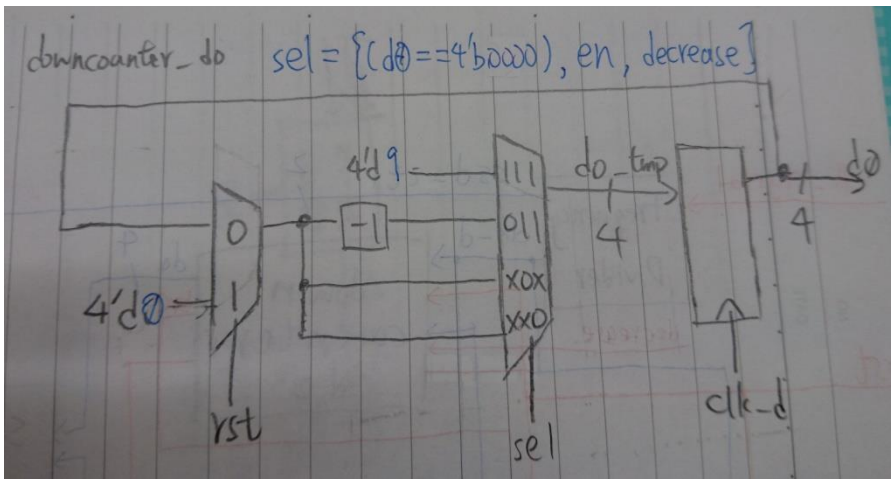

圖 4 Lab05-1 的 downconter_d0(個位數)

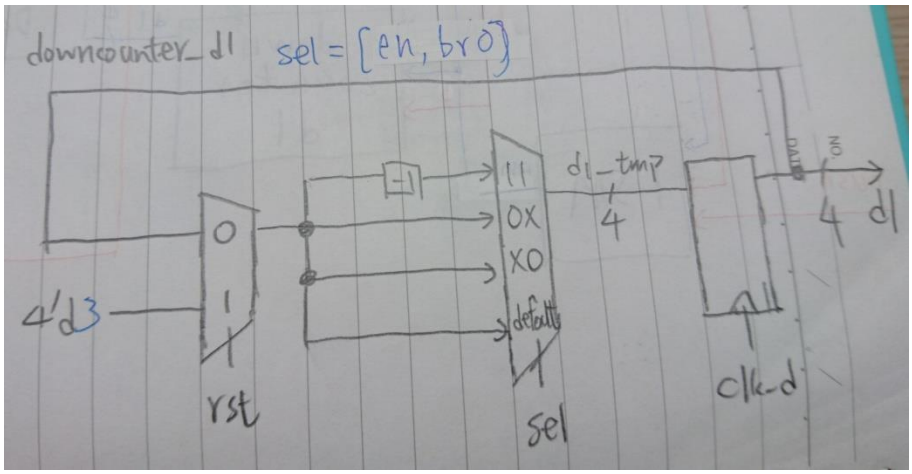

圖 5 Lab05-1 的 downconter_d1(個位數)

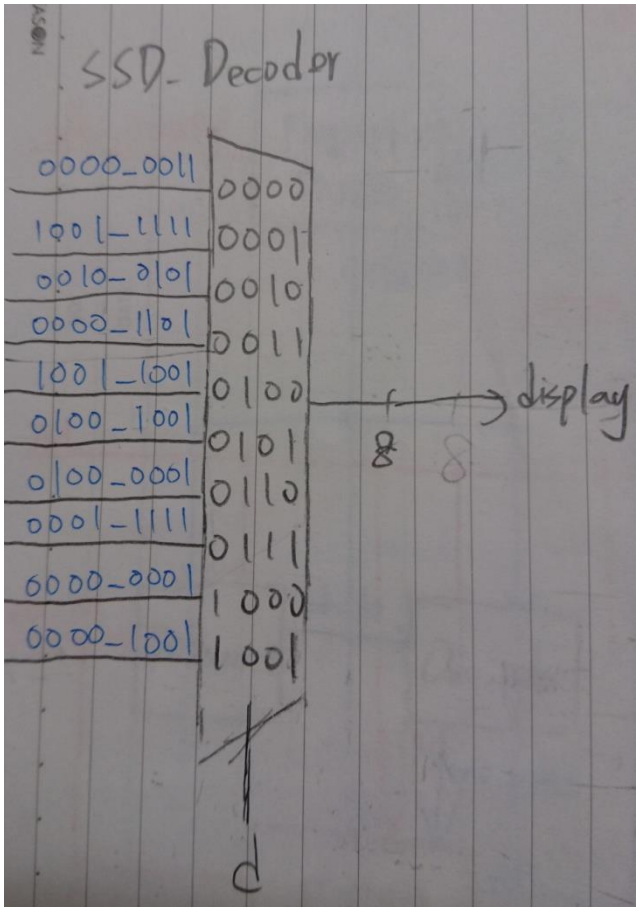

圖 6 Lab05-1 的 SSD Decoder

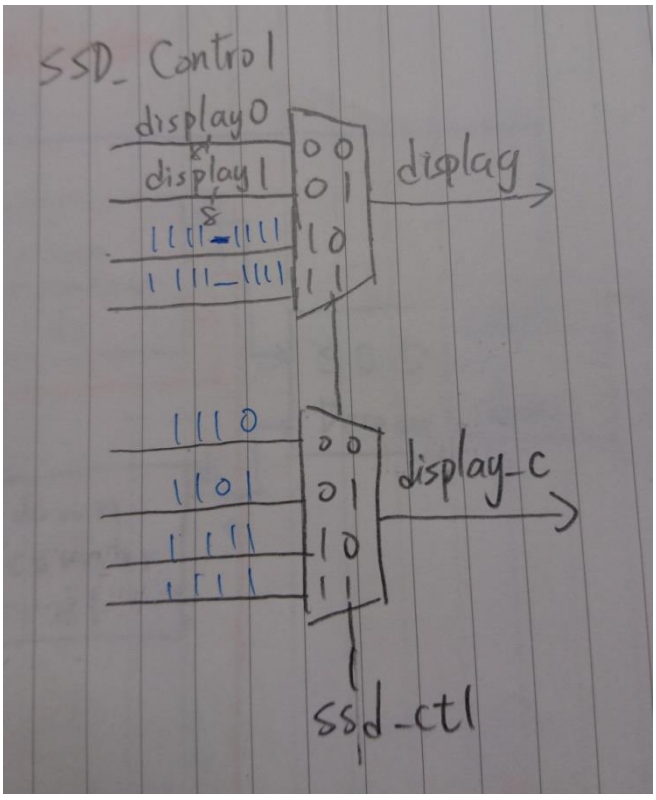

圖 7 Lab05-1 的 SSD Control

C. Pin assignment :
   a. Input :
      1) clk = W5
      2) rst = T17
      3) push = W19
   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) LEDs[0] = U16
      14) LEDs[1] = E19
      15) LEDs[2] = U19
      16) LEDs[3] = V19
      17) LEDs[4] = W18
      18) LEDs[5] = U15
      19) LEDs[6] = U14
      20) LEDs[7] = V14
      21) LEDs[8] = V13
      22) LEDs[9] = V3
      23) LEDs[10] = W3
      24) LEDs[11] = U3
      25) LEDs[12] = P3
      26) LEDs[13] = N3
      27) LEDs[14] = P1
      28) LEDs[15] = L1

3. Discussion :

A. 思考過程 :

這次要做的是頻率為 1Hz 的 30 秒倒數計時器。按照圖 1 的區塊圖來製作。

a. 除頻器:用來製造 1Hz

跟之前的 Lab 相同，用輸入的 clk(clk_crystal)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。除此之外，將其中的兩位輸出(ssd_ctl)，作為待會 SSD_Control 的其中一個 Input，此次選擇 cnt[20:19]。

b. FSM:用來決定是暫停/倒數狀態

state 有 0 和 1 兩種狀態。

當為 0 時，就代表為暫停狀態且輸出 en=0 來讓 Down Counter 停止；當為 1 時，就代表為倒數狀態且輸出 en=1 來讓 Down Counter 倒數。

若按下按鈕後(push=1)，就會換到另一個狀態(0->1 或 1->0)；反之，若沒按下(push=0)，就甚麼事都沒發生(0->0 或 1->1)。

若 rst 的話，state 便回到 0(暫停)。

c. Down Counter

分為個位數和十位數兩個 Down Counter

1) 個位數(d0)

輸入的有 en、decrease 兩個，其中 en 代表是在暫停或倒數狀態，而 decrease 代表是否為能數(Ex:到 00)。

若為倒數狀態並還能繼續數(en=1 且 decrease=1)，就判斷是否數到 0，若非 0 就直接減 1(Ex:29->28)；若為 0 就代表需要借位(Ex:20->19)，將 9 輸進下一個，並輸出 br0=1，作為 d1 的 decrease。

若 rst 的話，值便回到 0。

2) 十位數(d1)

基本上和個位數沒什麼差別，只需將 d0 輸出的 br0，當成 decrease 輸入進來就可以。

若 rst 的話，值便回到 3。

d. SSD Decoder

將 Down Counter 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-bit2 的顯示碼(display0、display1)，將它們輸入到下個 SSD Control。

e. SSD Control

把在除頻器得到的 2-bit 的 ssd_ctl 輸入，因為 2-bit 可以得到 4 種

不同的結果(00、01、10、11)，可以作為選擇 SSD 位數的依據，而且因為其變化速度極快，讓本來是依序做的動作，看起來像是同時並行的。

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，不過因為這是只用到後面 2 位，因此當 ssd_ctl 為 10、11 時，便把 display 和 display_c 所有 bit 都設成 1，讓它們不會亮，避免 bug 出現。

B. 過程中的 Bug :

若是按的時間過久，便會因為經過兩次 clk 的 posedge，而回到原本的狀態(Ex:0->1->0)；但按得太短，就會因為沒有經過 posedge，而沒有變化。必須按的時間剛好通過 1 次 posedge，才能順利變換狀態。這問題可以利用 One Pulse 來替換原本的 push 解決，在 Lab05-2 便是這樣。

二.Lab05-2 : 30-second down counter (短按暫停長按回到 30)

1. Design Specification :

A. Input : clk_crystal //輸入的頻率(100MHz)
rst //當=1 時，回到 30 並暫停
pb_in //按鈕，短按暫停/倒數，長按回到 30

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
LEDs //數到 00 讓所有 LED 亮起

C. Wire : clk_d //除頻後頻率(1Hz)，用來驅動 downcounter
clk_100 //除頻後頻率(100Hz)，驅動 FSM 相關功能
[1:0] ssd_ctl //從除頻器輸出，解碼後為 display_c
pb_de //經過 Debounce 後的 pb_in
one_pulse //經過 One_pulse 處理後，產生為期 1 週期的 1
en //0 為暫停，1 為倒數
rst_2 //功能同 rst，由 FSM 輸出

decrease	//當=0，無法再倒數(已到 00)；反之則為可以繼續數，接在個位數的 downcounter
br0	//借位，功能同 decrease，接在十位數的 downcounter
[3:0] d0	//個位數的值
[3:0] d1	//十位數的值
[7:0] display0	//個位數的顯示碼
[7:0] display1	//十位數的顯示碼

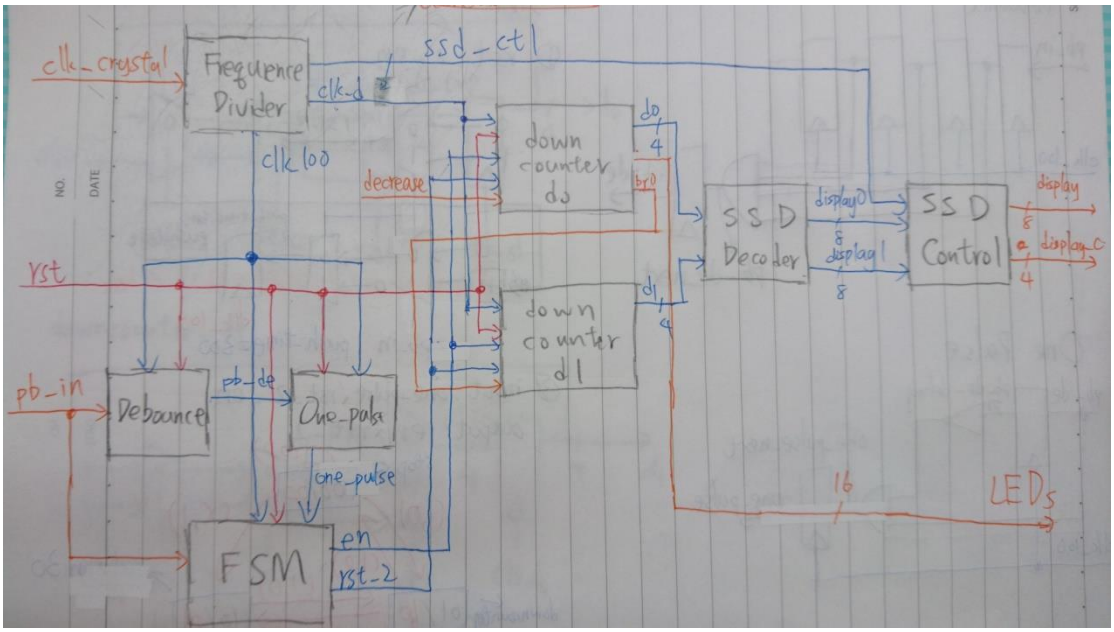

圖 8 Lab05-2 的區塊圖

2. Design Implementation :

A. Logic diagram :

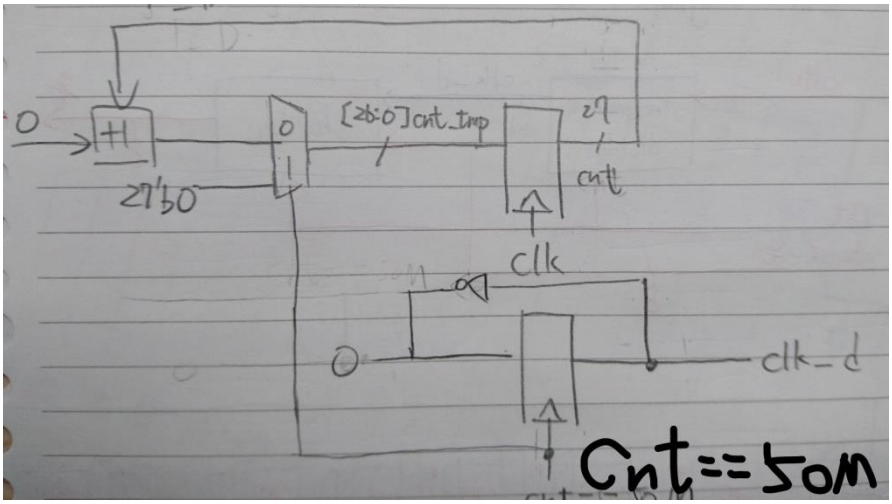

圖 9 Lab05-2 的除頻器(100MHz->1Hz)

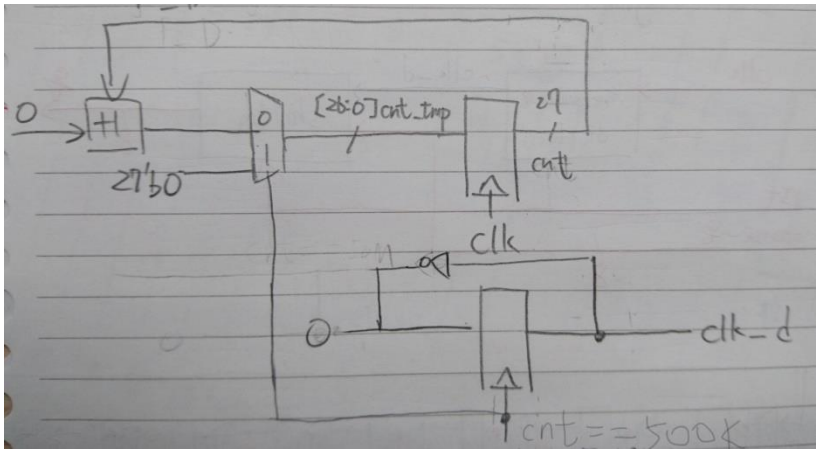

圖 10 Lab05-2 的除頻器(100MHz->100Hz)

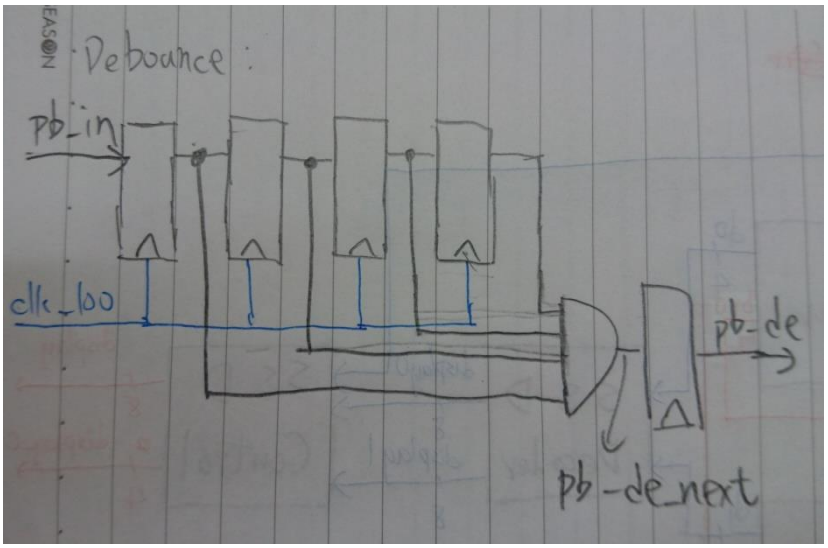

圖 11 Lab05-2 的 Debounce

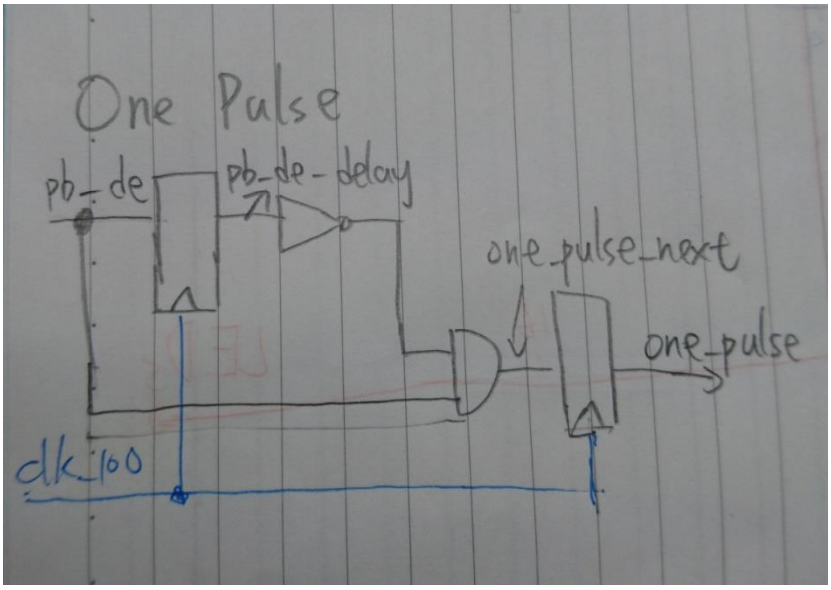

圖 12 Lab05-2 的 One pulse

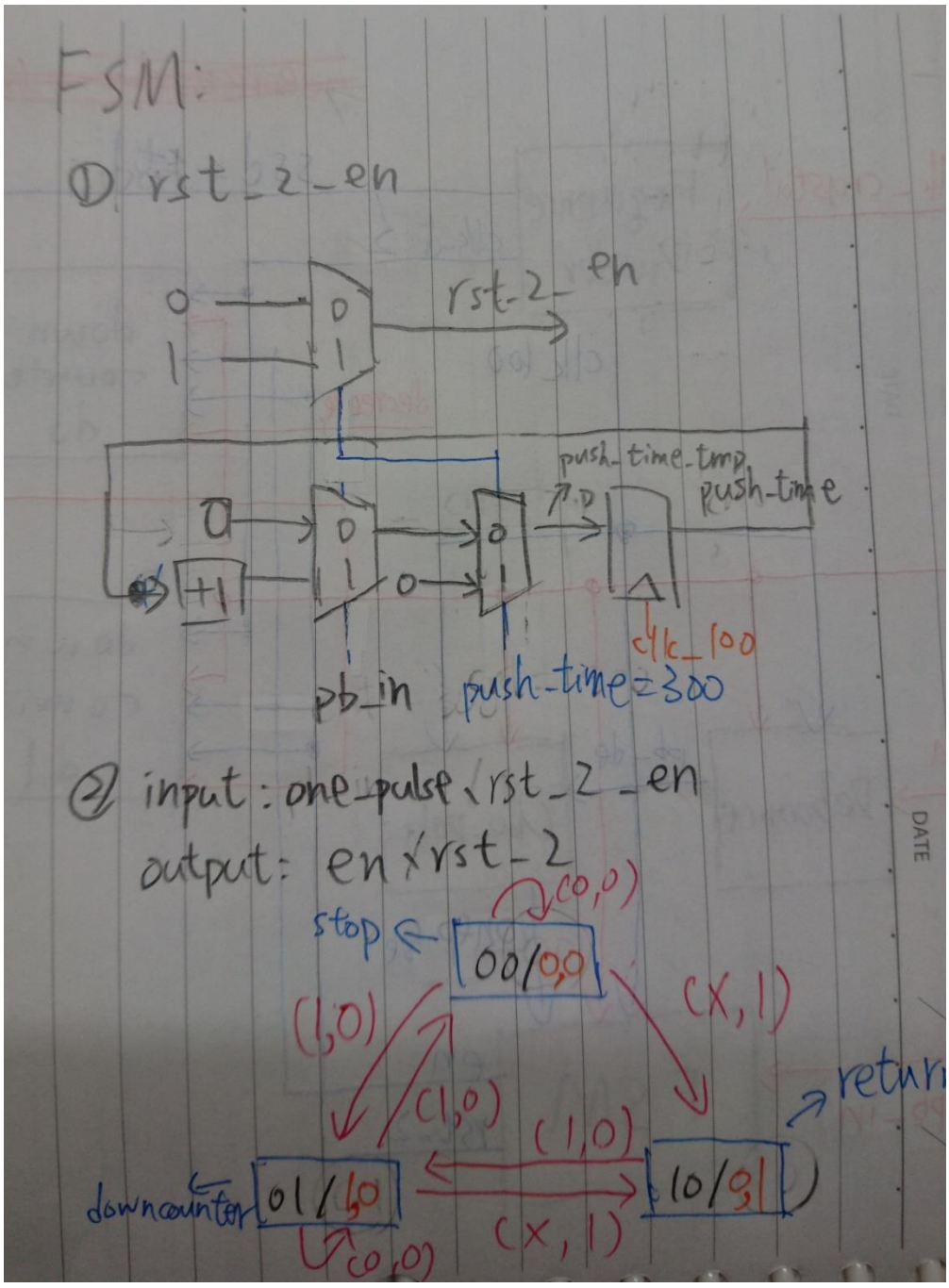

圖 13 Lab05-2 的 FSM

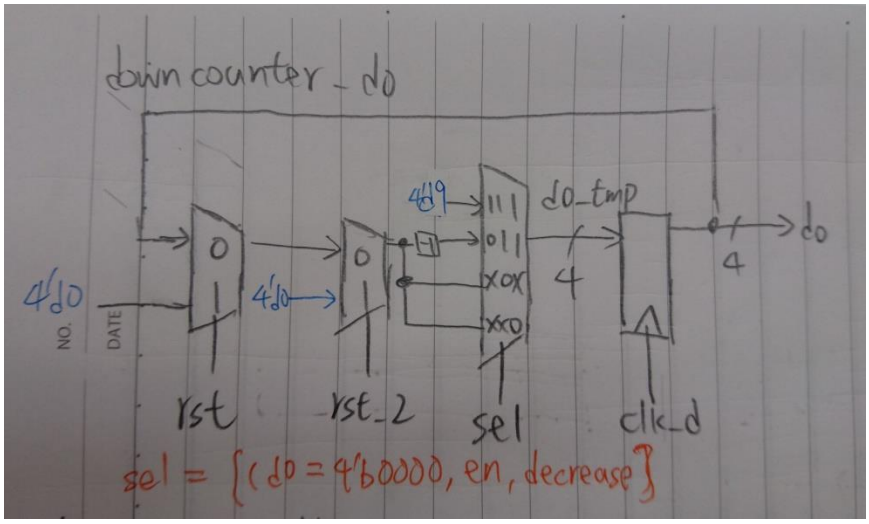

圖 14 Lab05-2 的 downcounter_d0(個位數)

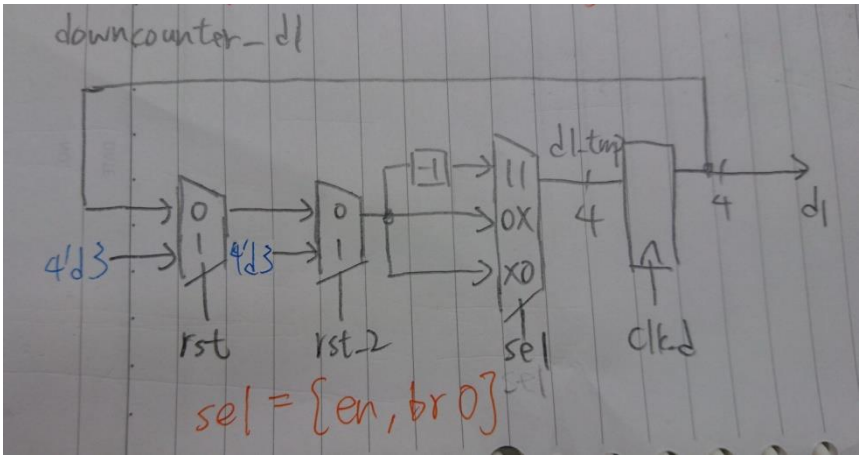

圖 15 Lab05-2 的 downcounter_d1(十位數)

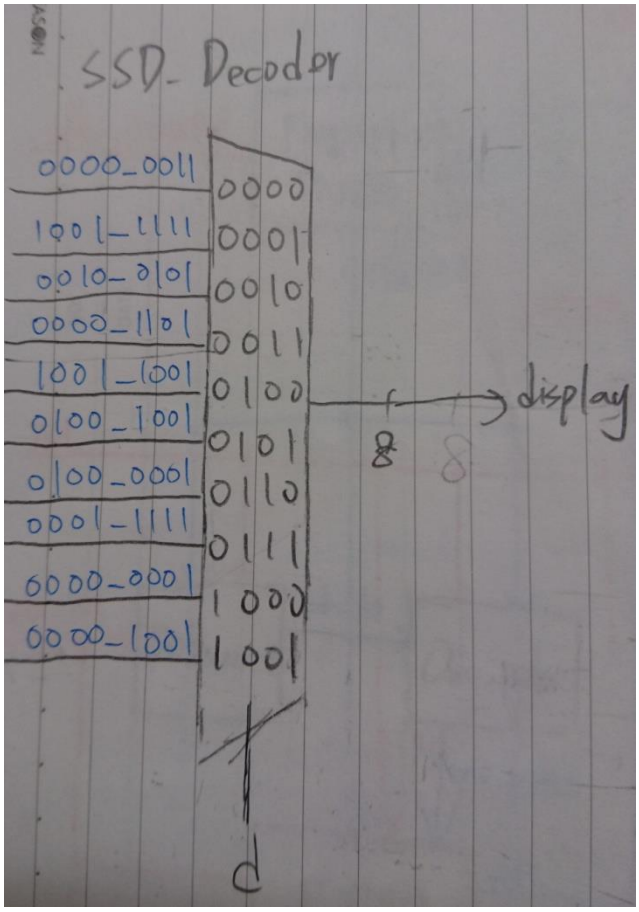

圖 16 Lab05-2 的 SSD Decoder

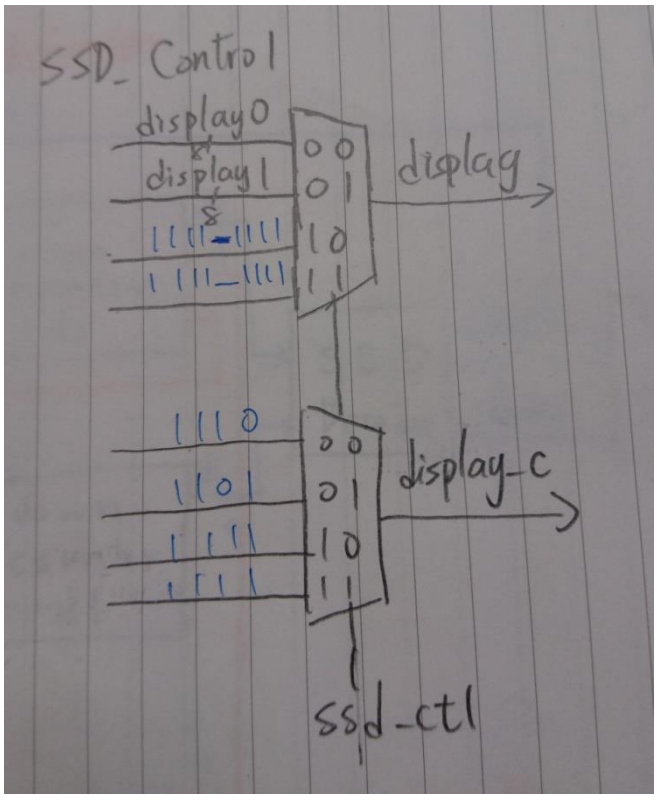

圖 17 Lab05-2 的 SSD Control

B. Pin assignment :
    a. Input :
        1) clk : W5
        2) rst : T17
        3) pb_in : W19
    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) LEDs[0] = U16
        14) LEDs[1] = E19
        15) LEDs[2] = U19
        16) LEDs[3] = V19
        17) LEDs[4] = W18
        18) LEDs[5] = U15
        19) LEDs[6] = U14
        20) LEDs[7] = V14
        21) LEDs[8] = V13
        22) LEDs[9] = V3
        23) LEDs[10] = W3
        24) LEDs[11] = U3
        25) LEDs[12] = P3
        26) LEDs[13] = N3
        27) LEDs[14] = P1
        28) LEDs[15] = L1

3. Discussion :

A. 思考過程 :

這次要做的是頻率為 1Hz 的 30 秒倒數計時器，而且將暫停/倒數和回到 30 做在同一個按鈕，利用短按(暫停/倒數)和長按(回到 30)來達到目的。基本上和 Lab05-1 一樣，但多了新的頻率(100Hz)，和 Debounce、One Pulse 來製造短按的效果，並修改 FSM。

a. 第二個除頻器 : 製造 100Hz

只需將除頻器中的條件"cnt==50M"改為"cnt==500k"便可以製造出 100Hz 的頻率。

b. Debounce : 消除不穩定的震盪

當我們按下按鈕時，其內部的彈簧會發生震盪，有時表面上只按 1 次，實際上內部發生的震盪導致按了 3 次。

利用一個 4-bit 的 Shfiter，每次 clk 都將現在按鈕的狀態輸入(pb_in)，若 4 個都為 1，則輸出 pb_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=>短按

將剛得到的 pb_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，這可以來當作短按的效果。

d. 修改 FSM

此部分分為兩個

1) 計算按住時間的 Counter->長按

設計一個 Counter，每次 clk posedge 時，若為按住狀態(pb_in=1)，則將裡面的值 cnt 加 1。當 cnt 加到 300(3 秒)，便輸出 rst_2_en=1，作為待會 FSM 的其中一個 input。

2) FSM

Input : one_pulse、rst_2_en

Output : en、rst_2

State : 00(暫停)、01(倒數)、10(回到 30)

若 state=00，則輸出 en=0、rst_2=0。此時若 one_pulse=1，則 state 跳到 01；若 rst_2_en=1，則 state 跳到 10；其餘情況便仍然維持原樣。

若 state=01，則輸出 en=1、rst_2=0。此時若 one_pulse=1，則 state 跳到 00；若 rst_2_en=1，則 state 跳到 10；其餘情況便仍然維持原樣。

若 state=00，則輸出 en=0、rst_2=0。此時若 one_pulse=1，則 state 跳到 01；其餘情況便仍然維持原樣。

B. 過程中的 Bug :
起初只將 100Hz 的頻率接上 Debounce、One Pulse，沒有接上 FSM，如此在 1Hz 的 FSM 中，為期 0.01 秒的 One Pulse 便會顯得太短，無法改變 state。之後將 FSM 也接上 100Hz 的頻率就解決了。

三.Conclusion :

這次實驗雖然只有 2 個，但耗時反而比之前實驗久，果然實驗越到後面就越複雜，準時完成實驗越來越困難啦。

四.Reference :

1. 老師給的實驗講義
讓我知道 code 怎麼打，還有如何處理按鈕的震盪問題，跟製造 1 週期的 1。