Lab06 (109061256 陳立萍)

Lab6_1: Implement a stopwatch function (00:00 -59:59) with the

FPGA board:

1.1 Use the four (Seven-Segment Displays, SSDs) as the display. The left two digits represent the minute and the right two digits represent the second.

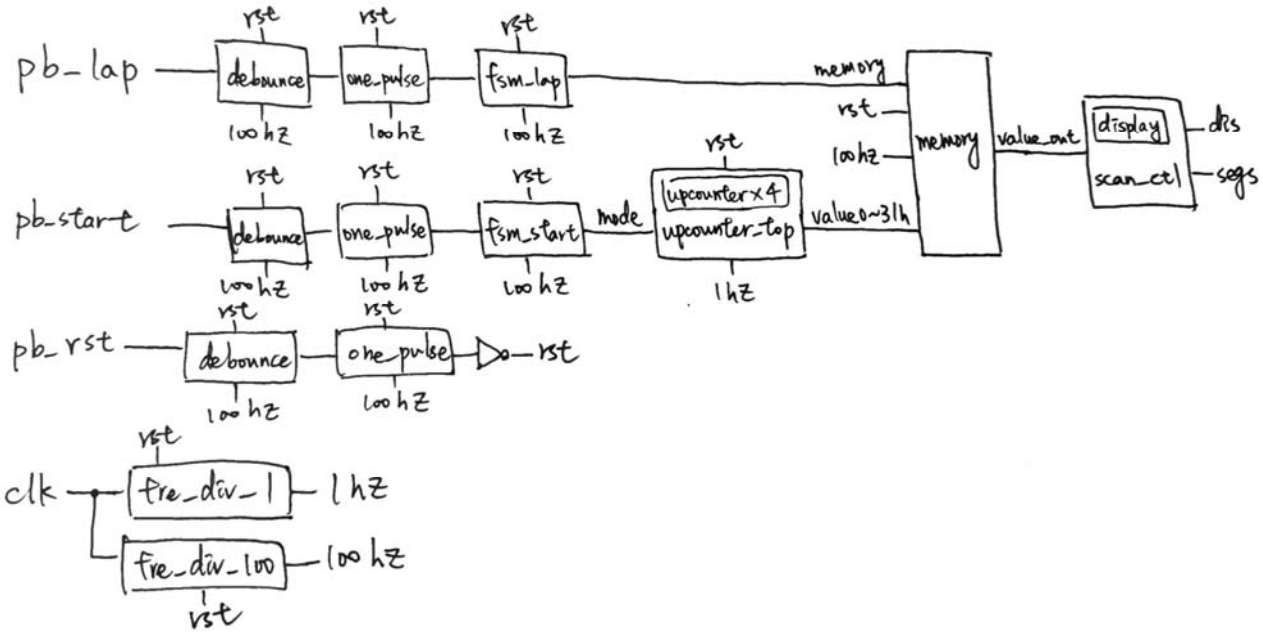
1.2 Use two push buttons to control the function. Use one button to control start/stop and the other to control the lap and reset. When the stopwatch counts, press the 'lap' button will freeze the SSDs but the stopwatch continues counting, and when press the 'lap' button again, the SSDs will start to show current time.

Design Specification

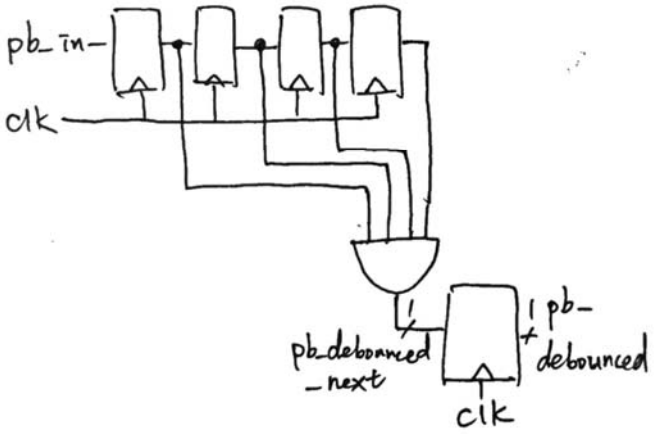
input : pb_rst(reset), pb_start(開始暫停), pb_lap(lap), clk(100Mhz)

output : [7:0]segs(七段段顯示器圖形), [3:0]dis(四個七段顯示器)

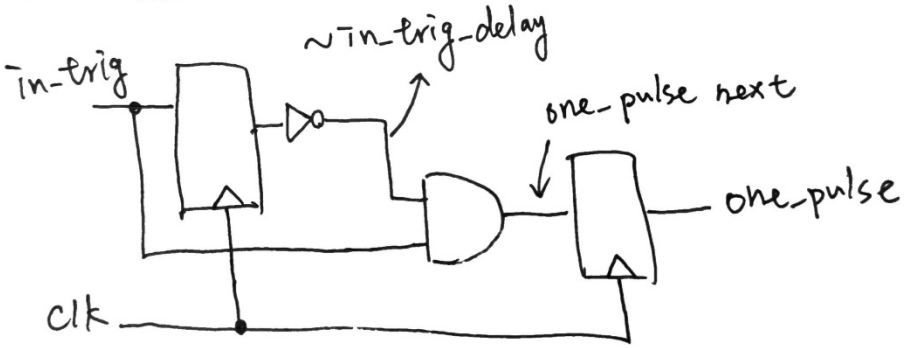
logic diagram



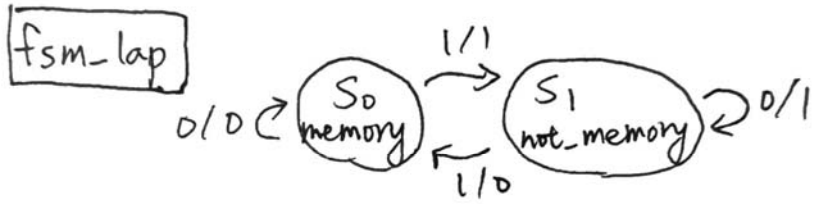
Debounce

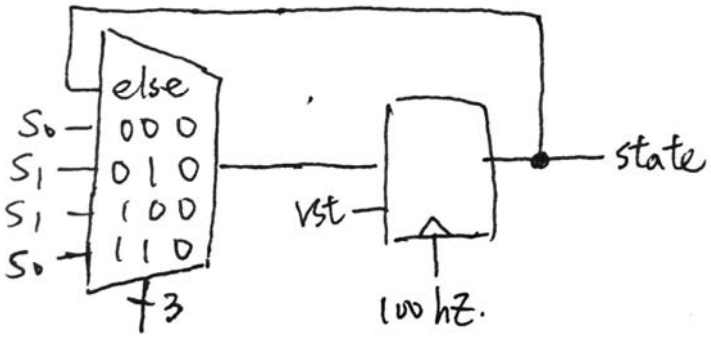


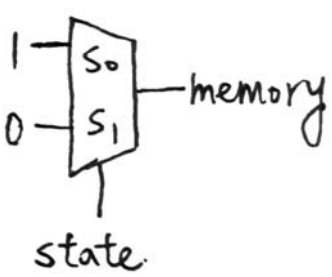
one_pulse



fsm_lap

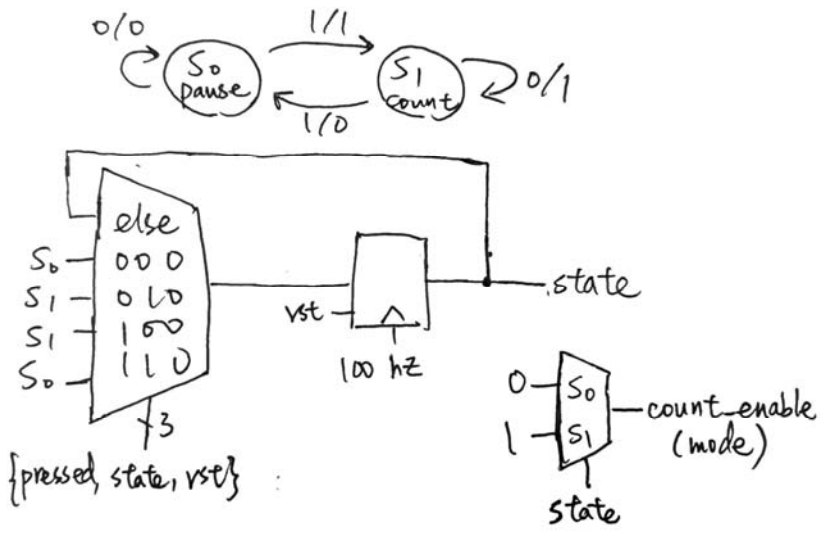




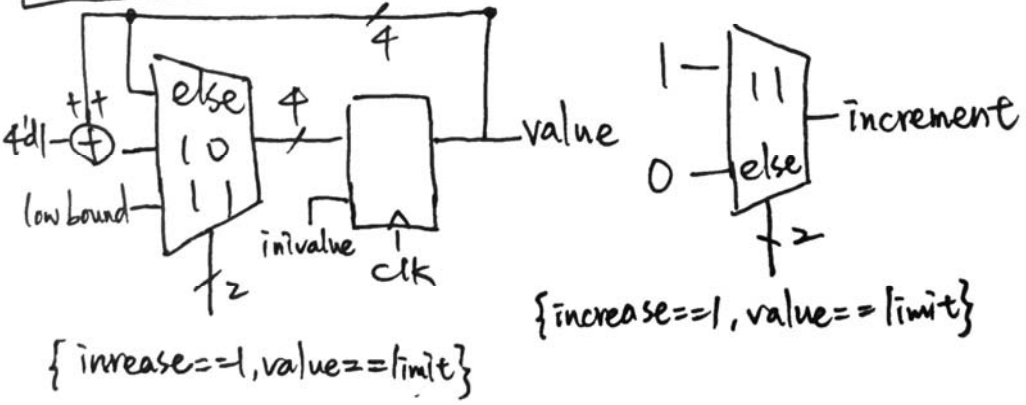


{pressed, state, rst}

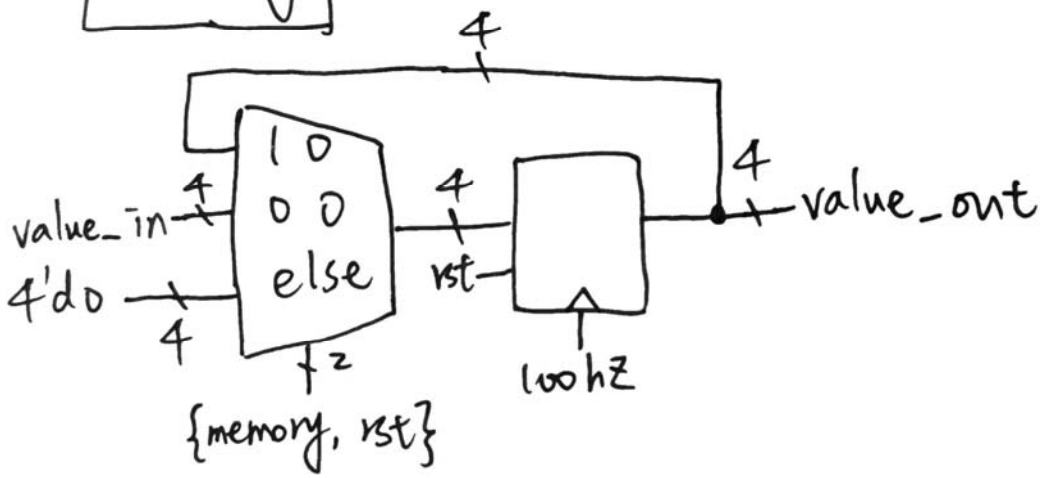
fsm_start



value and increment



memory



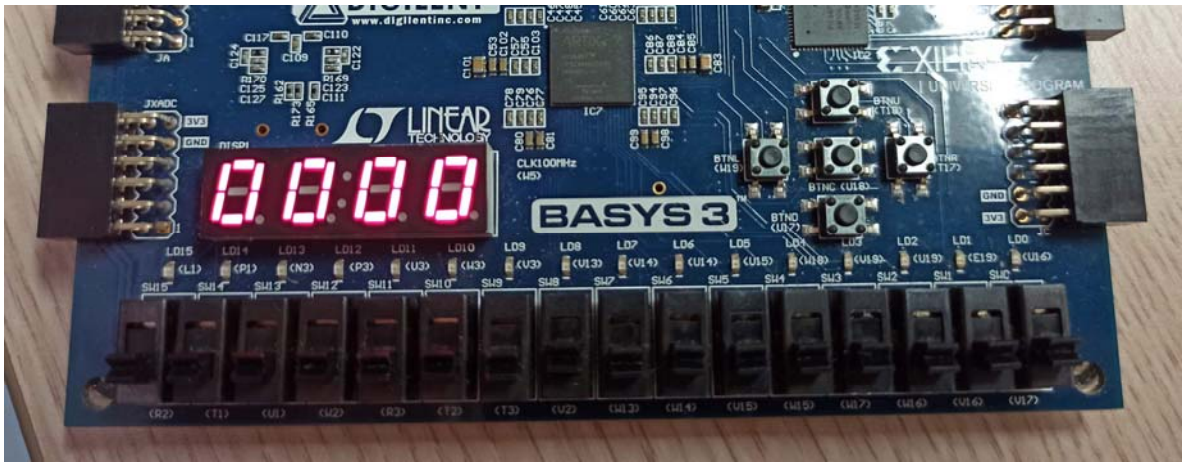
I/O pin assignment:

dis[3]	dis[2]	dis[1]	dis[0]	clk	pb_rst	pb_start	pb_lap
W4	V4	U4	U2	W5	U18	T17	W19

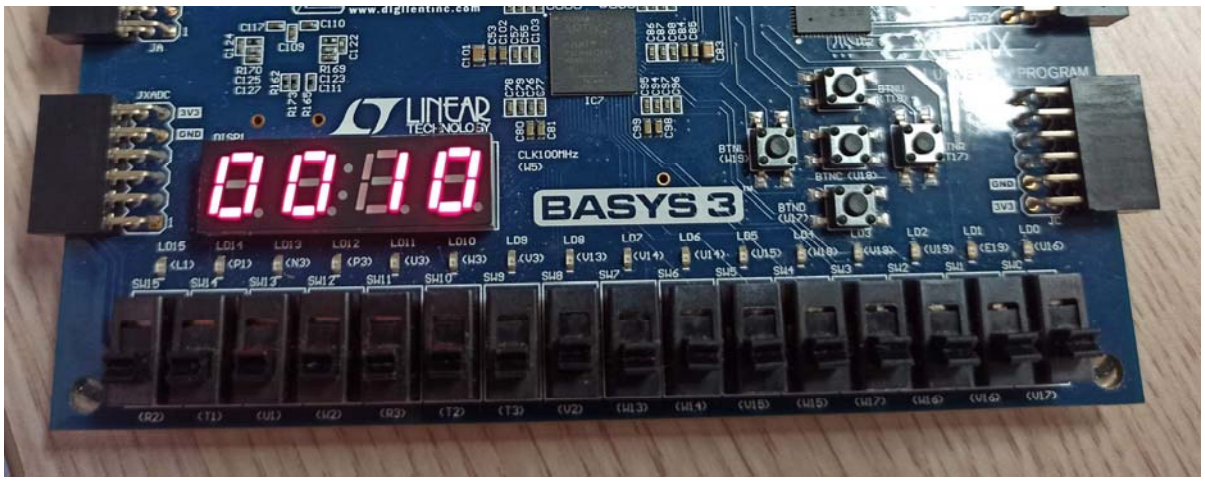
segs[7]	segs[6]	segs[5]	segs[4]	segs[3]	segs[2]	segs[1]	segs[0]
W7	W6	U8	V8	U5	V5	U7	V7

Design Implementation

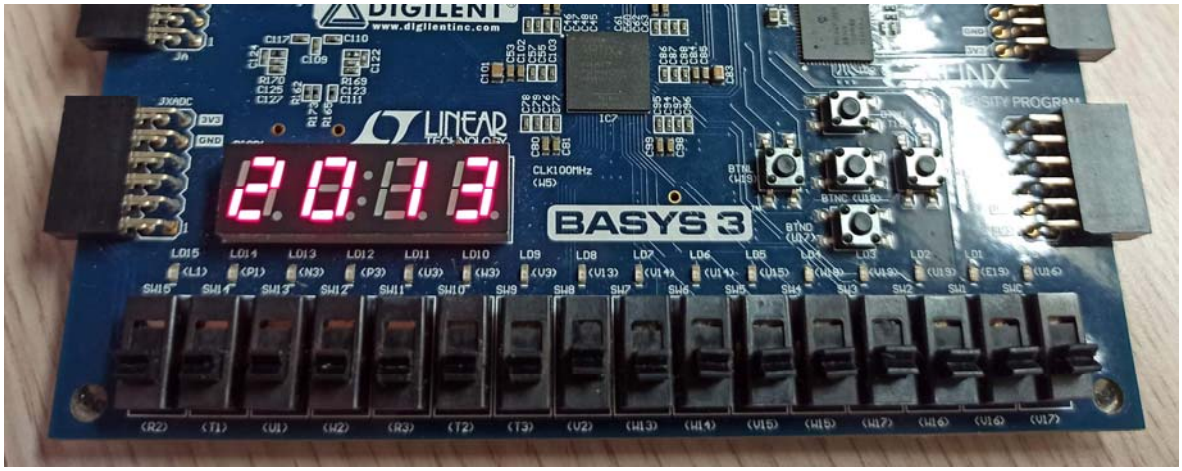
start from 00 / reset

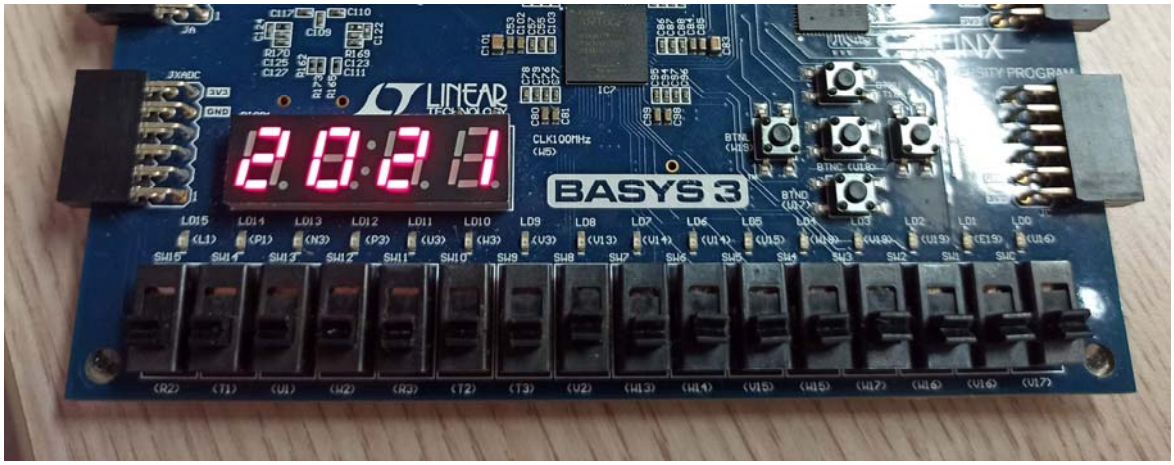


up counting…/stop/start



lap/continue counting





設計方法：

這個題目裡共用了 12 個.v 檔。

其中 fre_div_1.v:製造 1hz,◦fre_div_100.v:製造 100hz,◦debounce.v:消除按下按鈕後的 0101 震盪◦one_pulse.v: 用來製造一個 clock 長度的 pulse，以確保輸入的訊號只經過一個 clock，也就是我們製造的 one_pulse 訊號◦scan_ctl:製造視覺暫留,◦display.v：七段顯示器輸出。。

以上各個設計原理之前已介紹過，不在此贅述。

此題新增的 module 如下：

＊fsm_lap.v:

用來製造要不要記憶的 memory 訊號給 memory.v。此 module 為一個 FSM，狀態分為 STATE_MEMORY（記憶）和 STATE_NOT_MEMORY（不記憶）。若在不記憶的狀態下，偵測到按鍵按下，則會直接跳到記憶狀態，同時將 memory 設定為 1。若此時在記憶狀態下，當偵測到按鍵按下，則狀態會直接跳到不記憶，同時將 memory 設定為 0，此 memory 訊號之後會連接到 memory.v 作為記不記憶的 enable。另外若偵測到 reset 訊號，則會直接跳到不記憶狀態。

＊fsm_start.v：與之前的開始暫停的 FSM 相同，但多了 mode 輸出給 upcounter_top.v。

＊upcounter.v:

分為以下 4 種狀況：

1.若此位元值等於此位元最大值（個位數為 9，十位數為 5），但是還要加一時：則此時會傳入此位元最小值（個位數為 0，十位數為 0）同時將此位元的 increment (進位)設為 1，此進位訊號會傳到更高一位作為更高一位的加一 enable。

2.若此位元沒有等於此位元最大值，且要加一，則直接將當時的值加一。

3.若不加一則維持在此原來的值

4.若偵測到 reset 訊號，則此位元輸入 inivalue (初始值 0000)，其餘則維持原來狀態。

＊upcounter_top.v:

將分的個位十位與秒的個位十位傳入 upcounter 做運算。並整合各個 upcounter。首先，秒的個位的加一由 mode 訊號控制，mode 訊號為 fsm_start 產生，當 mode 為 1 的時候代表 start 也就是開始數，同時個位的 increment（進位）會連結到秒的十位作為加一的控制訊號。同理秒的十位的 increment（進位）會連結到分的個位作為加一的控制訊號，分的個位 increment（進位）會連結到分的十位作為加一的控制訊號。而若秒的個位加一訊號 mode 為 0 代表暫停狀態，則會停止不倒數。各級的輸出由 value 接收。

＊memory.v:

memory 的輸出為 4 個 value_out 值，此四個值會傳到 scan_ctl 作 display 顯示在七段顯示器上。

memory.v 運作分為以下 3 種狀況：

1.收到的 memory 訊號為 0 時，則此時計數器數到的值會一直不斷傳入 memory 的 value_out，也就是 value_out 會和此時計數器的值同步

2.收到 memory 訊號為 1 時，則 value_out 會將此時原本的值一直保留，也就是將 flipflop 的值輸出拉回 flipflop 輸入，值一直保持不變，當之後遇到 memory 變回 0 後，會將此時計數器數到的值輸入，並且 value_out 值會一直和計數器同步。

3.若偵測到 reset 訊號，則會直接將 4 個 bit 記憶的值清空為 0000。

＊lab6_1_top.v

最後再用 lab6_1_top 將所有的功能整合並接線，完成此題所有要求。

Lab6_2: Implement a timer (can support as long 23:59) with the following functions:

2.1 Use one DIP switch as the 'setting' control. When the 'setting' is ON, you can use two buttons to set the minute and second.

2.2 Use other two buttons to control the timer operation. One button forstart/stop and the other button for pause/resume.

2.3 When the time goes to 0, light up all the LEDs.

Design Specification

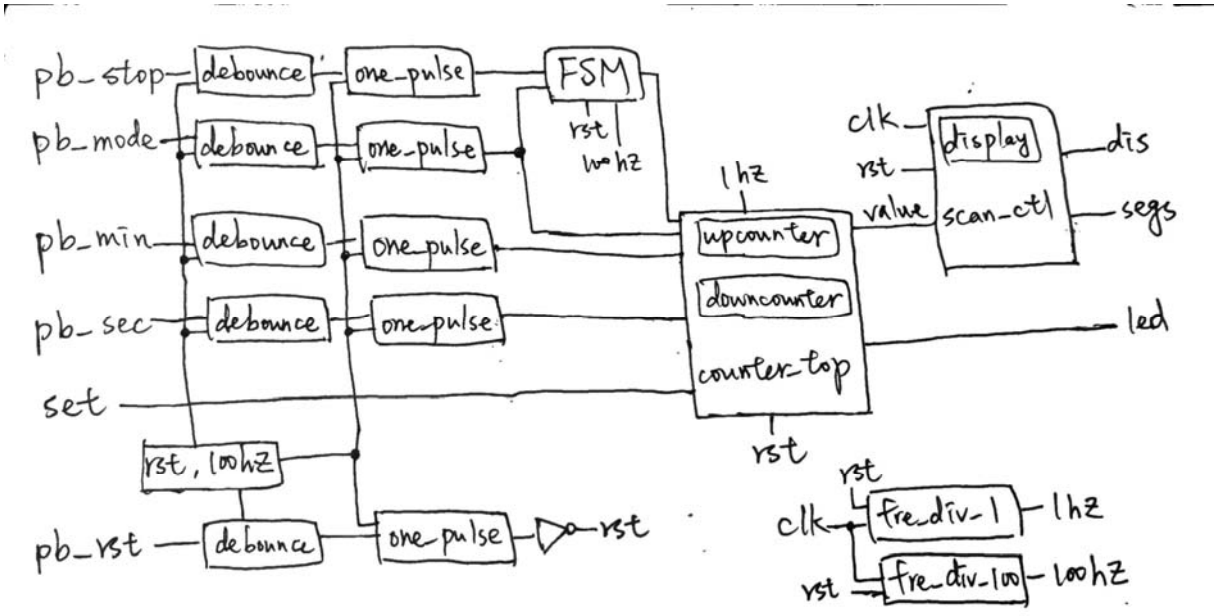
input : pb_stop(開始暫停), clk (100Mhz), set(設定時間), pb_rst,(reset),

pb_mode(pause / resume), pb_min(設定分), pb_sec(設定秒)

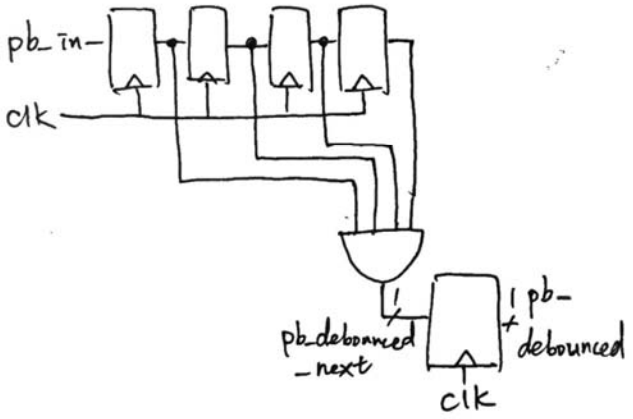
output : [7:0]segs(七段段顯示器圖形), [3:0]dis(四個七段顯示器),

[15:0]led(LED)

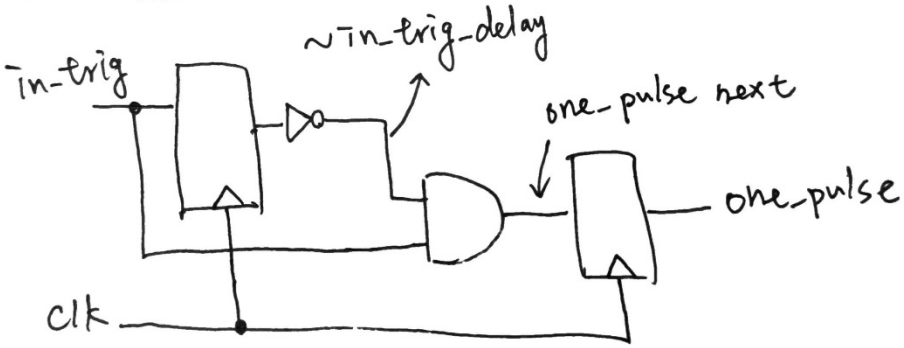
logic diagram



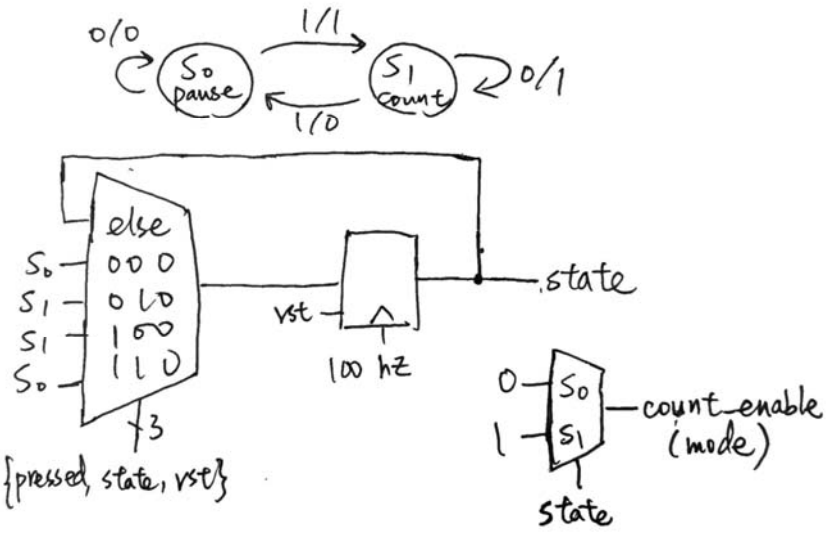
Debounce

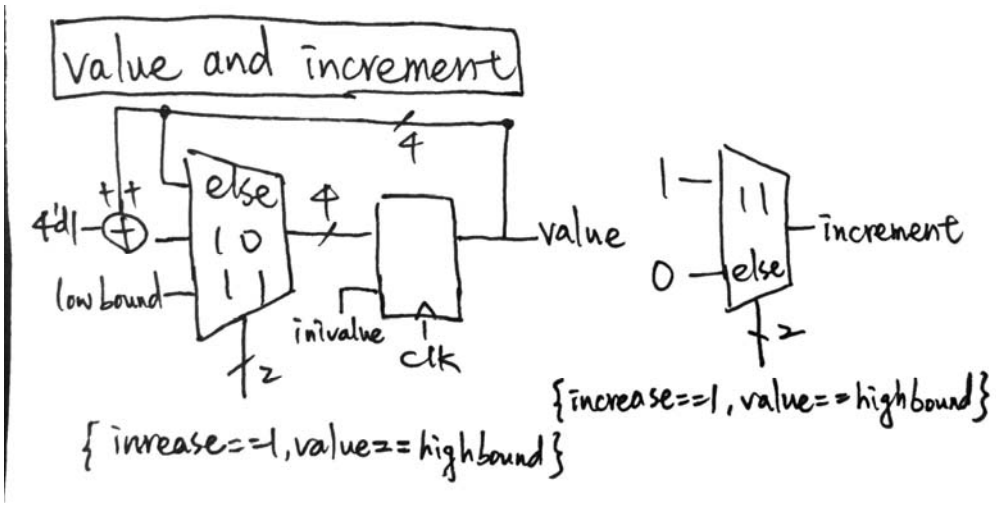


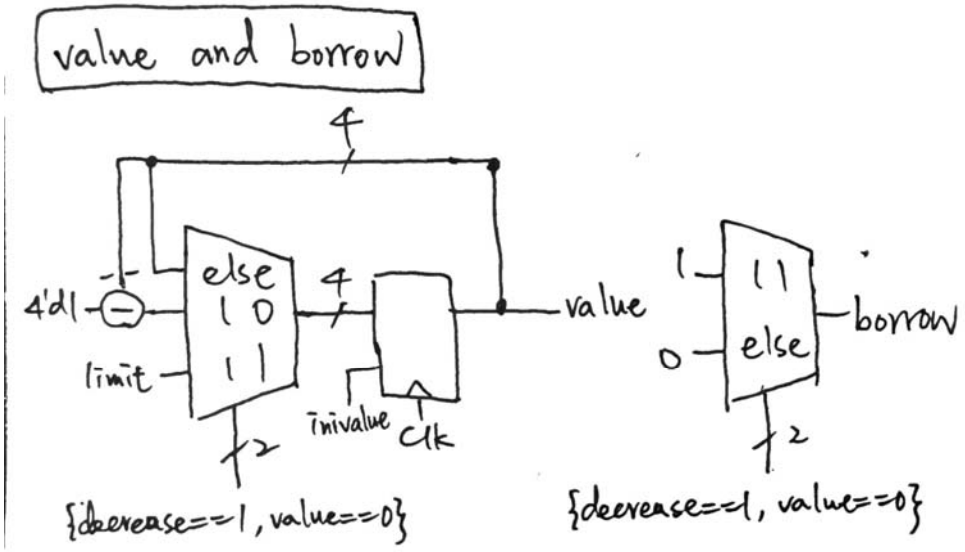
one_pulse



FSM







I/O pin assignment:

dis[3]	dis[2]	dis[1]	dis[0]	clk
W4	V4	U4	U2	W5

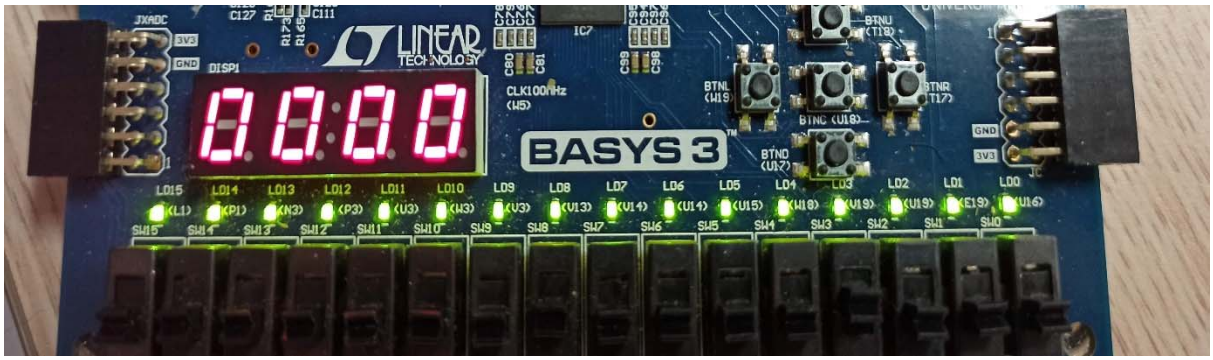
pb_stop	pb_mode	pb_min	pb_sec	pb_rst	set
T18	U17	W19	T17	U18	R2

led[15]	led[14]	led[13]	led[12]	led[11]	led[10]	led[9]	led[8]
L1	P1	N3	P3	U3	W3	V3	V13
led[7]	led[6]	led[5]	led[4]	led[3]	led[2]	led[1]	led[0]
V14	U14	U15	W18	V19	U19	E19	U16

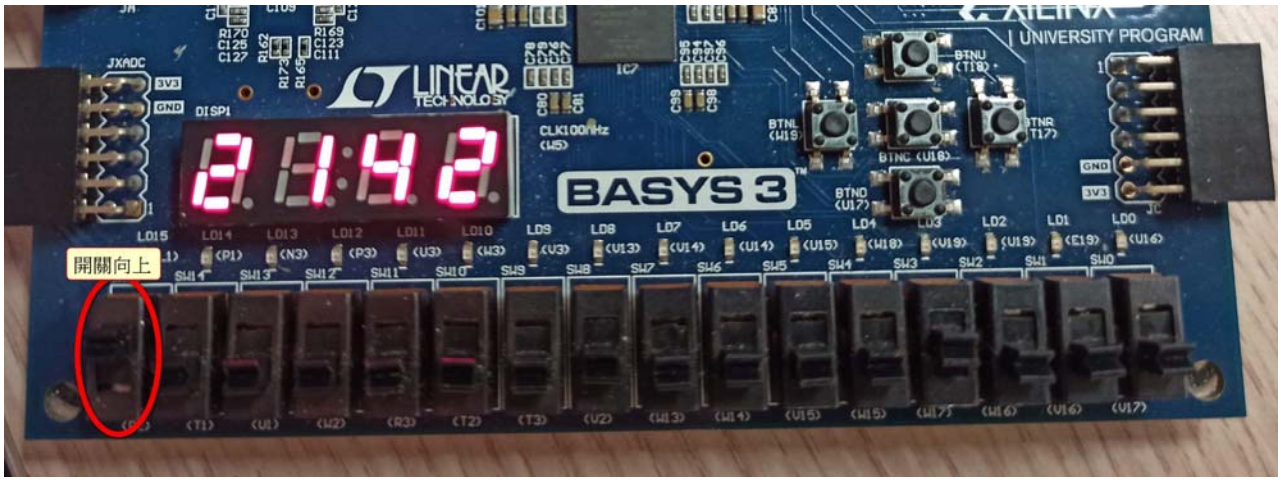
segs[7]	segs[6]	segs[5]	segs[4]	segs[3]	segs[2]	segs[1]	segs[0]
W7	W6	U8	V8	U5	V5	U7	V7

Design Implementation

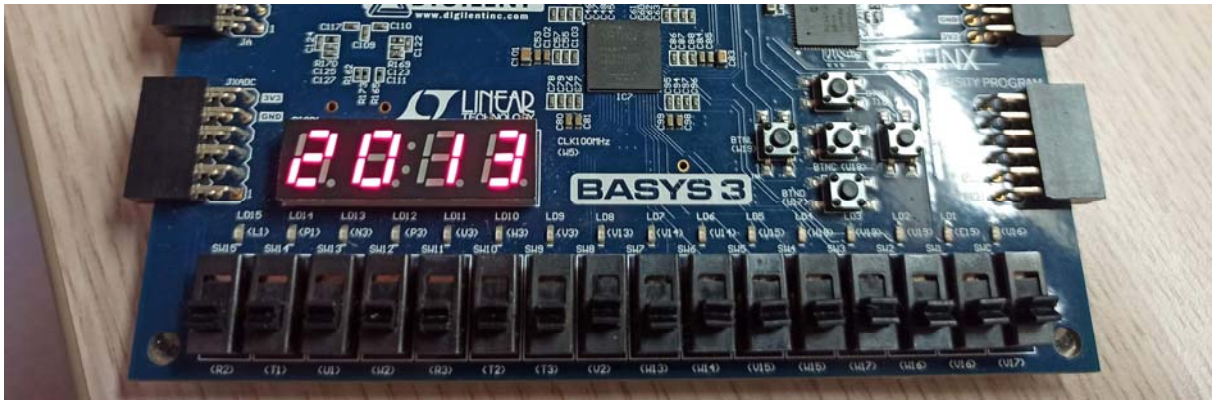
start from 00/count down to 00/reset, all LED light up



set time



down counting…/ resume and pause/stop/start



設計方法：

這個題目裡共用了 11 個.v 檔。

其中 fre_div_1.v:製造 1hz,◦fre_div_100.v:製造 100hz,◦debounce.v:消除按下按鈕後的 0101

震盪。one_pulse.v: 用來製造一個 clock 長度的 pulse，以確保輸入的訊號只經過一個 clock，也就是我們製造的 one_pulse 訊號。scan_ctl:製造視覺暫留,。display.v：七段顯示器輸出。。以上各個設計原理之前已介紹過，不在此贅述。

此題新增的 module 如下：

＊FSM.v: 與之前的開始暫停的 FSM 相同，但多了 mode 輸出給 upcounter_top.v。

＊upcounter.v:

分為以下 4 種狀況：

1.若此位元值等於此位元最大值（個位數為 9，十位數為 5），但是還要加一時：則此時會傳入此位元最小值（個位數為 0，十位數為 0）同時將此位元的 increment (進位)設為 1，此進位訊號會傳到更高一位作為更高一位的加一 enable。

2.若此位元沒有等於此位元最大值，且要加一，則直接將當時的值加一。

3.若不加一則維持在此原來的值

4.若偵測到 reset 訊號，則此位元輸入 inivalue (初始值 0000)，其餘則維持原來狀態。

＊downcounter.v:

分為以下 5 種狀況：

1.若此位元值等於此位元最小值（個位數為 0，十位數為 50），但是還要減一時：則此時會傳入此位元最大值（個位數為 9，十位數為 5）同時將此位元的 borrow (借位)設為 1，此借位訊號會傳到更高一位作為更高一位的減一 enable。

2.若此位元沒有等於此位元最小值，且要減一，則直接將當時的值減一。

3.若不減一則維持在此原來的值

4.若偵測到 reset 訊號，則此位元輸入 inivalue (初始值 0000)，其餘則維持原來狀態。

5.若偵測到設定訊號時，則會將 upcounter 計數到的值輸入到此位元。

＊counter_top.v:

將分的個位十位與秒的個位十位傳入 upcounter 做運算，再將 upcounter 計數到的值傳給各自的 downcounter，並整合各個 upcounter 與 downcounter。首先，秒的個位的加一由 set_sec 訊號控制，set_sec 訊號為 setting 訊號去 and set_sec_next 訊號後產生，setting 訊號為使用者由 DIP switch 輸入，代表此時可不可以調時間，而 set_sec_next 則為使用者按下 button 的訊號。當 set_sec 為 1 的時候代表可以加一也就是當偵測到為設定模式且按下秒加一的按鈕，同時個位的 increment（進位）會連結到秒的十位作為加一的控制訊號。同理，分的個位的加一由 set_min 訊號控制，set_min 訊號為 setting 訊號去 and set_min_next 訊號後產生，setting 訊號為使用者由 DIP switch 輸入，代表此時可不可以調時間，而 set_min_next 則為使用者按下 button 的訊號。當 set_min 為 1 的時候代表可以加一也就是當偵測到為設定模式且按下分加一的按鈕，同時個位的 increment（進位）會連結到分的十位作為加一的控制訊號。

而若 setting 訊號為 0 代表此時無法調整時間，只有開始暫停功能。另外，因為題目為最大值到 23：59 所以分的部分在加一時不能數到 24，所以在這裡還做了分是否數到 24 的判斷，若成立則會將 rst_min 變成 1，傳到 upcounter 裡將分的部分歸零。

各級的輸出由 val_set 接收，再傳入 downcounter 作為開始倒數的值。在四個 downcounter

裡，秒的個位的減一由 start 訊號控制，start 訊號為 FSM 產生並和(~setting)做 and 運算產生，意思是要倒數且不坐在設定時間狀態。
當 start 為 1 的時候代表 start 也就是開始倒數，同時個位的 borrow（借位）會連結到秒的十位作為減一的控制訊號。同理分個位的 borrow（借位）會連結到分的十位作為減一的控制訊號。
在這個 module 裡，還控制了如果倒數到 0000 則要停止，並亮起 16 個 LED 燈。
＊lab6_2_top.v
最後再用 lab6_2_top 將所有的功能整合並接線，完成此題所有要求。

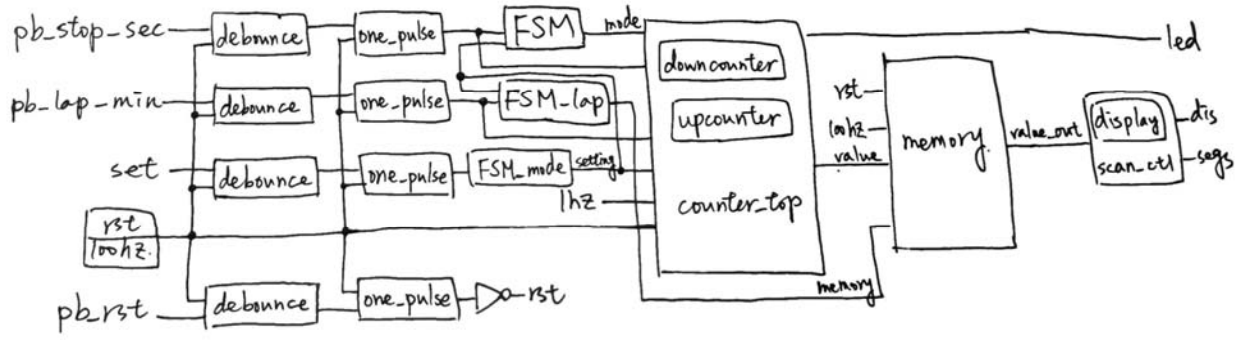
3.Lab6_3_bonus : Integrate the above two functions together with only three buttons.

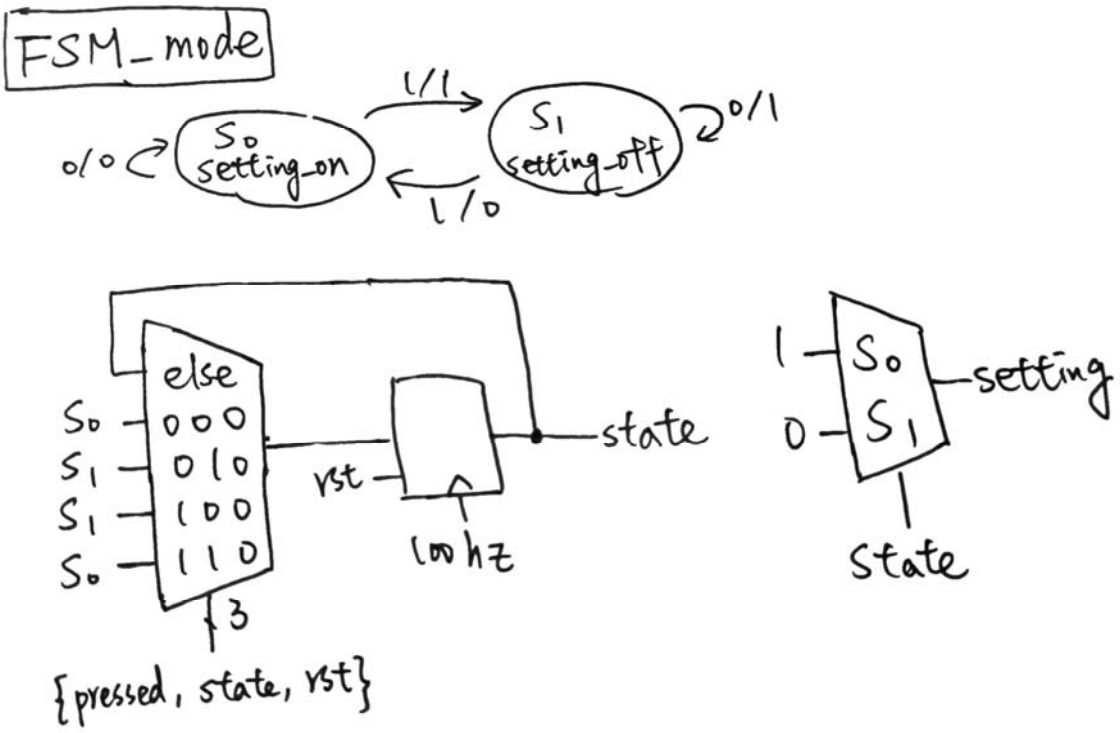
Design Specification

input : set(設定時間/開始暫停切換), clk (100Mhz), pb_ rst (reset),
pb_lap_min(lap and set min), pb_stop_sec(stop and set sec)

output : [7:0]segs(七段段顯示器圖形), [3:0]dis(四個七段顯示器),
reg [15:0]led (LED)

logic diagram





I/O pin assignment:

dis[3]	dis[2]	dis[1]	dis[0]	clk
W4	V4	U4	U2	W5

pb_lap_min	pb_stop_sec	pb_rst	set
W19	T17	U18	U17

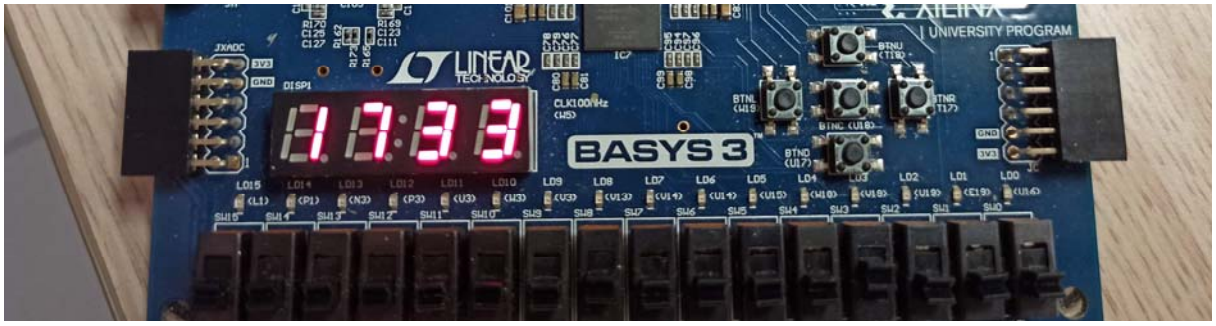
led[15]	led[14]	led[13]	led[12]	led[11]	led[10]	led[9]	led[8]
L1	P1	N3	P3	U3	W3	V3	V13
led[7]	led[6]	led[5]	led[4]	led[3]	led[2]	led[1]	led[0]
V14	U14	U15	W18	V19	U19	E19	U16

segs[7]	segs[6]	segs[5]	segs[4]	segs[3]	segs[2]	segs[1]	segs[0]
W7	W6	U8	V8	U5	V5	U7	V7

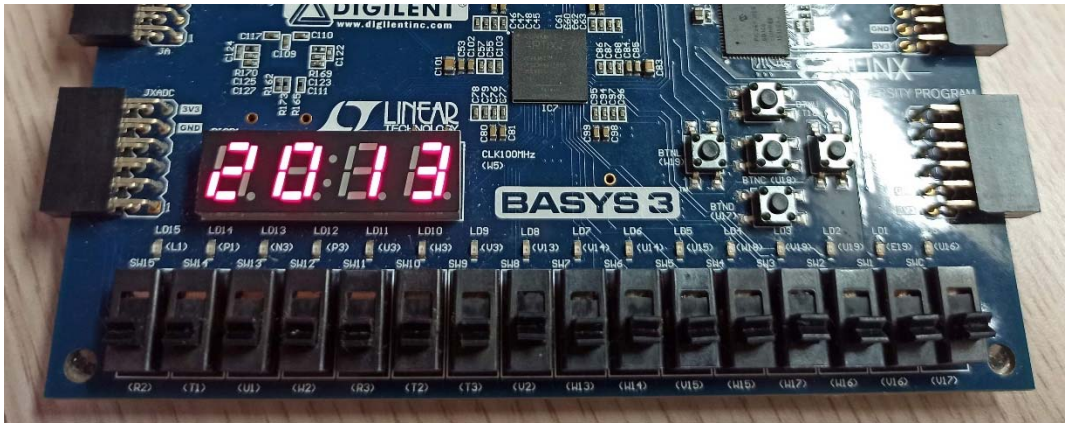
Design Implementation

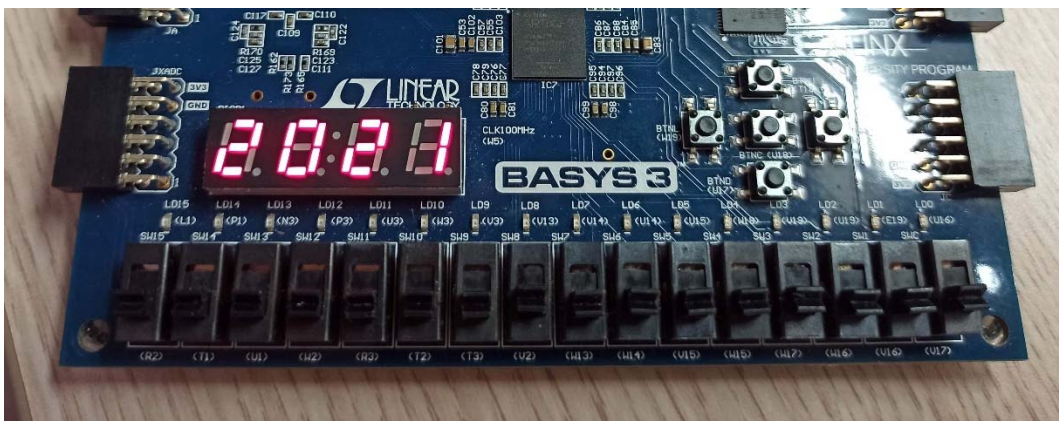
start from 00/count down to 00/reset, all LED light up

set time/down counting…/ resume and pause

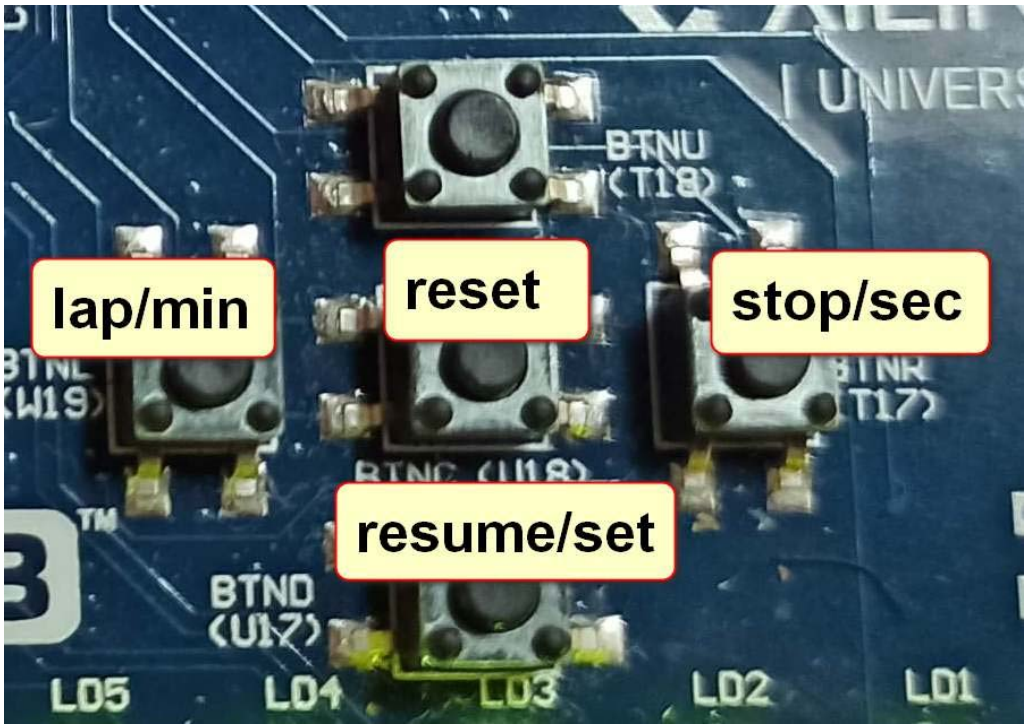


lap/continue counting/stop/start





開關功能



設計方法：

這個題目裡共用了 14 個.v 檔。

其中 fre_div_1.v:製造 1hz, fre_div_100.v:製造 100hz, debounce.v:消除按下按鈕後的 01 震盪, one_pulse.v: scan_ctl:製造視覺暫留, display.v 各個設計原理已介紹過，不在此贅述。

此題是要將前 2 題作結合，並只用 3 個按鈕，因此 counter_top.v, upcounter.v, downcounter.v, memory.v 都可以直接使用前兩題的.v 檔，在此不再重複介紹

新增與修改的 module 如下：

＊FSM.v:

比前 2 題的 FSM_start 多了（~setting）的判斷，也就是要在不設定(setting 為 0)模式下，這個 FSM 才會運作，才有開始暫停功能。

＊FSM_lap.v:

比第一題的 fsm_lap 多了 setting 的判斷，也就是要在設定(setting 為 1)模式下才可以調整時間。

＊FSM_mode.v:

因為第二題是用 switch 做時間設定控制，因此 set 可以一直維持在 1 或是 0，可是按鈕只有在按下的那一瞬間才會轉換 0 或 1，因此這個 module 就是用來產生 setting 訊號的 0 和 1。首先，若狀態在 setting_on（設定），則此時 setting 為 1，當偵到按鈕被按下一次後，狀態會直接跳到 setting_off（不設定），且此時 setting 為 0，而若狀態在 setting_off（不設定），則此時 setting 為 0，當偵到按鈕被按下一次後，狀態會直接跳到 setting_on（設定），且此時 setting 為 1。另外這裡還設定了若是 reset 被按下則會跳到 setting_off

（不設定）。

＊lab6_3_top.v

最後再用 lab6_3_top 將所有的功能整合並接線，在此調整分與 lap 功能共用一個按鈕，調整秒與開始暫停共用另一個按鈕，set 用第三個按鈕，另外 reset 依題目可以再多設一個按鈕，接著就完成此題所有要求。

Discussion

Lab6_1

這題要小心 memory 功能的記憶值，要記憶的話就要不斷把 flipflop 輸出接回輸入，不記憶則一直輸入計數器的值，因為一開始一直搞錯要記憶與不記憶要輸入的值，導致根本不會停止並記憶，修改完後的結果與題意相符。

Lab6_2

這題其實只是要重覆用一個 module 很多次，因此要小心接線，在 counter_top 裡 include 了 4 次 upcounter 與 4 次 downcounter，8 行寫在一起超容易接錯線，幸好 vivado 有自動繪 logic diagram 的功能，才能快速找到接錯的點。另外我在這個.v 檔裡將 val_set[2]和 val_set[3]寫相反原本要 val_set[2]=4 且 val_set[3]=2 時要歸零，但後來變成 val_set[3]=4 且 val_set[2]=2，導致他一直無法數到 23 後就變 00，害我浪費了 1 個半小時。不過結果後來就正常了。

Lab6_3_bonus

這題一開始我一直以為很難，因為只能用三個按鈕，卻要有一堆功能，不過後來實作的時候發現其實只多了一個 set 處理而已，其他的地方基本上都是前兩題就已經有的，只是在接線的時候因為有 2 個功能共用一個按鈕，接錯線，導致秒從分的地方輸出。另外，在做這題的時候，我因為兩個英文單字中間多一個底線，而導致 bug 一直找不到，幸好後來刪除一個底線後就成功了。

Conclusion

這次 lab 真的好難，花了好多的時間，因為 module 數目與上一次相比整整多了快 2 倍，一大堆線與 input, output, wire, reg 變數。常常變數型別設錯，不然就是接錯線，打錯字。做邏輯設計真的要很有耐心，而且每一步驟都要細心處理，不然浪費很多時間在做錯的功能。最後看到 bonus 那題出來時是我最感動的時候，因為在 3 個按鍵裡包含了 6 個功能，幾乎和外面賣的計時器有 7~8 成像了，功能完整，又不需要用到太多的按鍵，只差在不會叫而已。

References

教授授課頭影片：語法運用，符號運用，設計觀念。