Lab07\_Electronic\_Clock\_II\_report

106061146 陳兆廷

Experiments:

1. 題目要求一個碼表，可以計時、凍結當前lap時間，因此需要一個有enable功能且有分、秒的clock、一個FSM來計算當前的state、一個除頻器，以及一個大module和SSD顯示功能。(圖一)

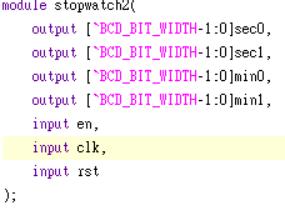
在clock module裡面，需要有輸出的各位分、秒、輸入的en、clk、rst，而內容很簡單，就如同上次的時鐘內容，在enable時開始計時即可(圖二)。

在FSM裡面，輸入要利用兩個按鈕(in1、in2)來控制開始，還有一個clk，輸出值則有判斷是否要計時的count\_enable(接上clock的en)、凍結畫面的freeze\_enable、是否重製的reset\_enable(接上所有的rst)，而其所有的state有如圖(四)四種，state diagram如圖(五)。

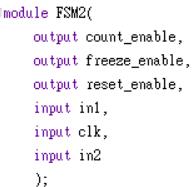
最後用大的display module將各個小module合併，輸入值為in1、in2兩個botton以及clk，輸出值為控制SSD的ssd\_ctl還有一個8bit的SSD，就大功告成。



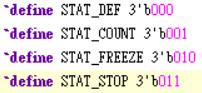
圖(一)



圖(二)



圖(三)



圖(四)

S2

S1

S4

S3

in1短按(start)

In1長按(Pause)

In2短按(lap)

in1長按(Pause)

In1短按(resume)

In2長按(reset)

In2長按(reset)

圖(五)

這個作業遇到的難題是FSM的撰寫，在lap計算時如何讓時鐘凍結在那個畫面且又繼續計時。約花了我2個小時。

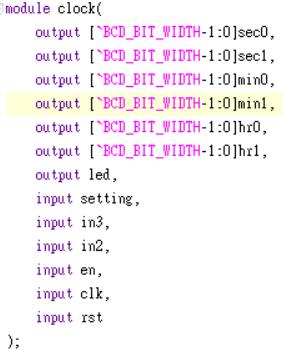
1. 這題和bonus是一樣的，因此我合在一起寫。

題目要求以一個I/O switch控制setting，可以自訂時間的倒數計時器(至23:59)以及碼表兩種，可以用3個按鈕來控制。

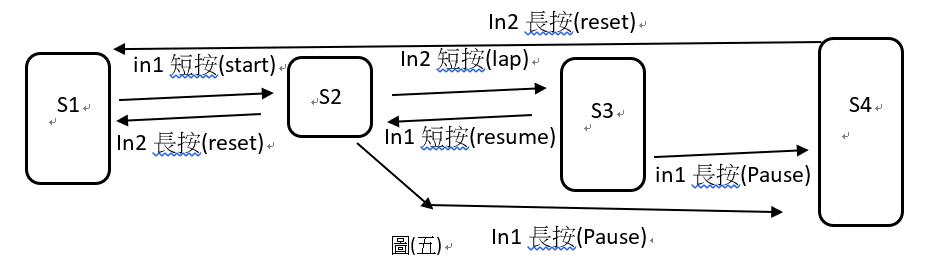
因此我對第一題稍作修改，在clock裡面新增hr0、hr1、led，並新增一個setting，當setting == 1時，用兩個輸入值min\_enable、hr\_enable設定初始值，當進入倒數計時模式、開始計時前可以設定初始的時間，且在倒數至0時0分0秒時led會變成1，如圖(六)，接著再依照setting來控制正數或是倒數計時，分別寫出碼表及倒數計時器。

在FSM裡，我新增幾條線和幾個輸出，在reset時(S1)，當setting是在倒數計時時，in2及in3分別代表min\_enable及hr\_enable，傳入clock設置初始值，而其餘便如同第一題，因為在clock裡已經寫好依照setting來正數或是倒數，只要count\_enable == 1就會動作。(圖七)

最後在display module裡面，只要新增輸入in1、in2、in3、setting，新增輸出led，並控制當時間小於一小時時顯示分及秒，時間大於一小時時顯示時及分，再輸入至SSD輸出即可。



圖(六)



In2短按(min++)

In3短按(hr++)

圖(七)

這題在構想時，一度多用了幾個state，原本想在FSM控制正數或是倒數，所以多了state\_stopwatch和state\_timer，又多一個state\_setting，後來搞了很久，發現在clock裡面控制兩種方法最為簡單，也可以直接及時輸出時、分、秒的各位數字，再修正了許多Bug，像是無法重新設定等，耗費約三個小時。

這次的lab又讓我再次練習各種state及FSM的應用。