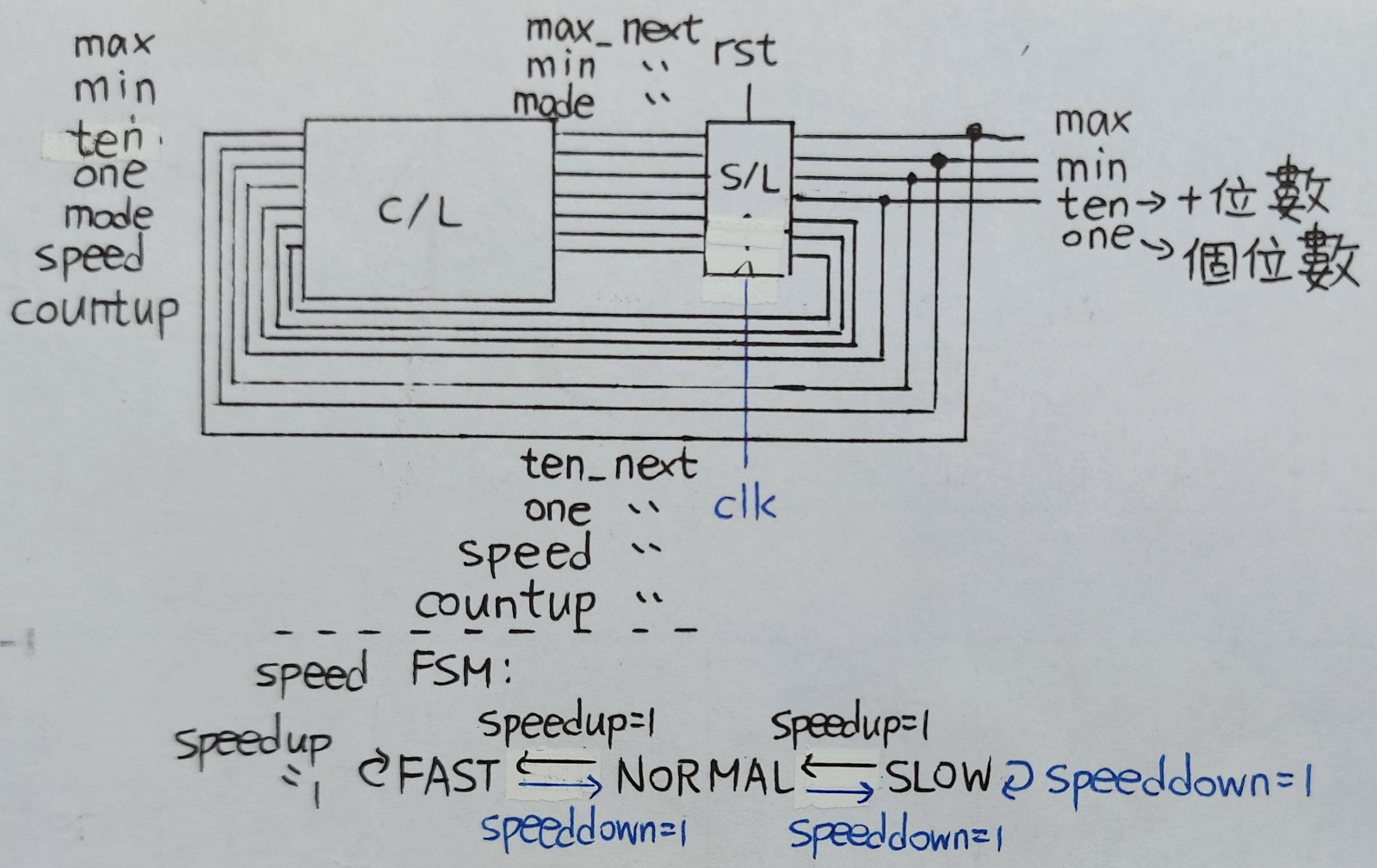
|  |  |
| --- | --- |
| **Lab 4** | |
| 學號: 109062318 | 姓名: 簡弘哲 |

1. 實作過程

Lab4-1:



Ten負責記錄十位數，one則記錄個位數(實際上可以用一個長度為2的陣列來記錄會更為簡潔)，要留意數到boundary number(0,9)時十位數的變換，下圖中的code有特判boundary case(00,99)，如果數到極限的話就維持住。

一張含有 文字 的圖片

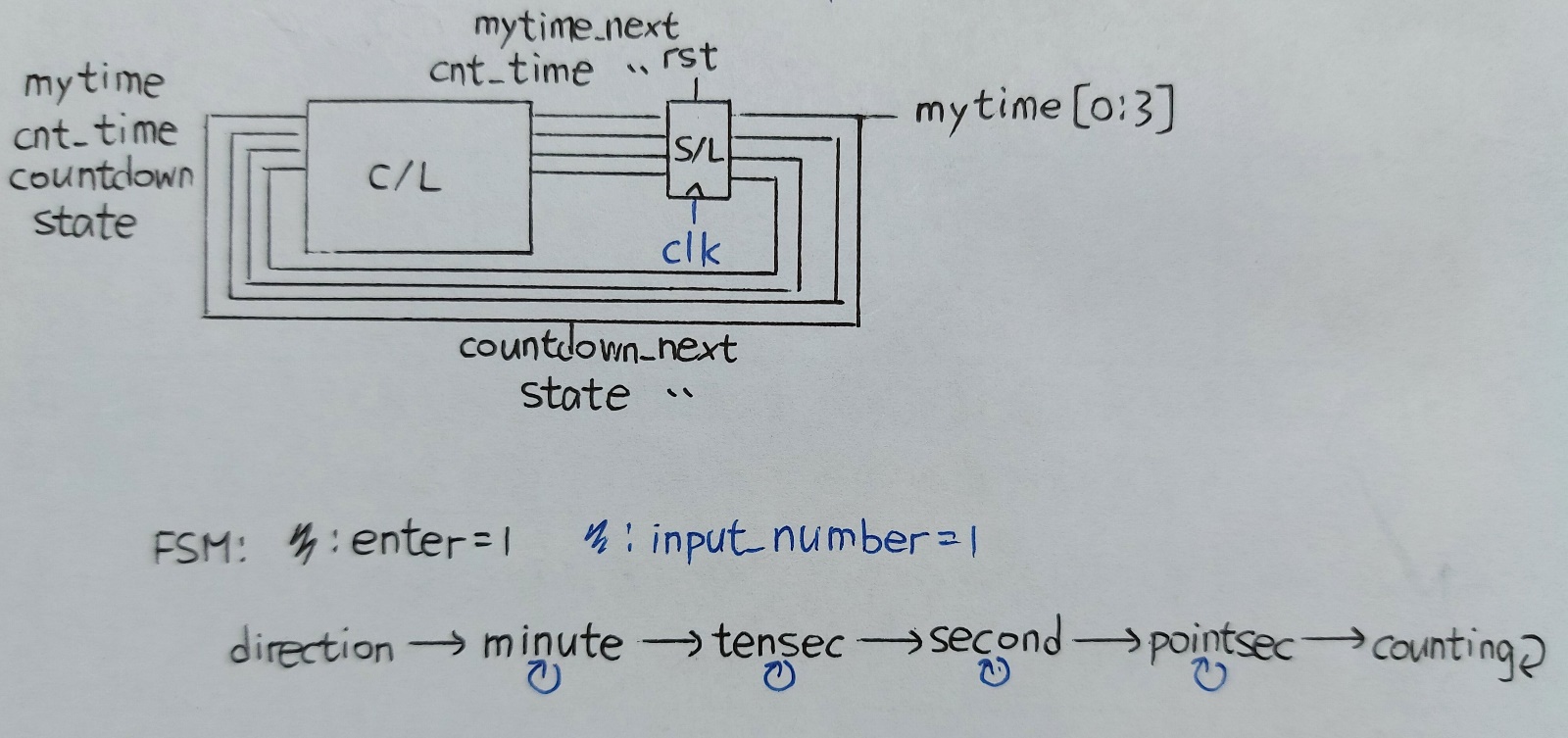
自動產生的描述

後來發現max,min不需要設max\_next,min\_next這兩個變數，分別用一句assign並檢查目前的十位數及個位數就可判斷是否需要亮起。

Mode記錄現在是start還是pause，在數秒的時候需要檢查現在的mode有沒有等於start。

我是利用fsm去控制speed的速度並分為3個state，分別是FAST,NORMAL,SLOW。(事實上也可以利用if判斷以及一些對speed+1、-1的動作就可達成目的，會更為簡單，只不過當時設計時沒有想到這點)

Lab4-2:



Mytime是目前用來顯示在7-segment的時間

Cnt\_time用來記錄碼表需要數到的終點，如果是countdown則cnt\_time是0000，否則的話就是使用者所設定的時間

State(3bits)用來記錄目前所在的狀態(總共6個)，分別是

設定direction(direction),

設定minute(minute),

設定10秒(tensec),

設定1秒(second),

設定0.1秒(pointsec)

以及數秒(counting)

圖中的黑色箭頭代表按下enter按鈕，而藍色箭頭則代表按下input\_number按鈕(在direction state應補上指向自己的箭頭，代表按下countdown按鈕切換start/pause)

數秒部分的實作:

藉由比較mytime(現在時間)跟cnt\_time(終點)來判斷是否數到終點，數秒的動作是用巢狀if並檢查boundary case(0,9)，array index 0指的是分鐘、1是10秒、2是1秒、3是0.1秒。一張含有 文字 的圖片

自動產生的描述、

Bonus:

一張含有 文字 的圖片

自動產生的描述

已知板子的clk 1s會100M上下，那經過0.1s就已經10M上下了，所以counter就數10M，數到10M的時候就代表已經過了0.1s，超過10M就把num重設為0。

在num==10M的時候將clk\_div輸出1，num!=10M的時候就維持0。

因為num=10M只會維持一個clk cycle，clk\_div也是，所以我的clock divider有把onepulse的效果一同做進去。

1. 學到的東西與遇到的困難

學到的東西:

1.最好不要有clk,rst之外的posedge信號例如en,dir

2.如何設計0.1s精確的clock divider

3.為了顯示時間而使用了陣列與for loop，以及了解陣列[MSB:LSB]跟[LSB:MSB]的差異

遇到的困難:

在寫4-2時把獨立的功能分成很多always block，但這些always block的=左手邊有共同的變數導致multi net driven，後來把=左手邊有相同變數的always block合成一個大的block就解決了。

2.

在4-2每一個state都設一個用來顯示7-segment的陣列，導致需要管理每一個時間的狀態而難以debug。

顯示7-segment設1個(mytime)

Number setting state設2個(goal,set\_time)

counting state設2個(countup\_time,countdown\_time)

3.

clk的速度沒調好導致按下按鈕完全沒反應或是不靈敏，以及忘記在combinational block裡忘了寫if(…\_onepulse)導致按鈕不會work

4.

4-2一開始不知道如何往上數，後來想到要maintain一個變數記錄數到的終點，如果是往下數那終點就是0，反之就是使用者設定的數字，在進入counting state前設定好。如下面兩圖:(mytime為現在時間,cnt\_time\_next則是要數到的終點)

一張含有 文字 的圖片

自動產生的描述一張含有 文字 的圖片

自動產生的描述

1. 想對老師或助教說的話

覺得lab一次比一次難、花更多心力與時間