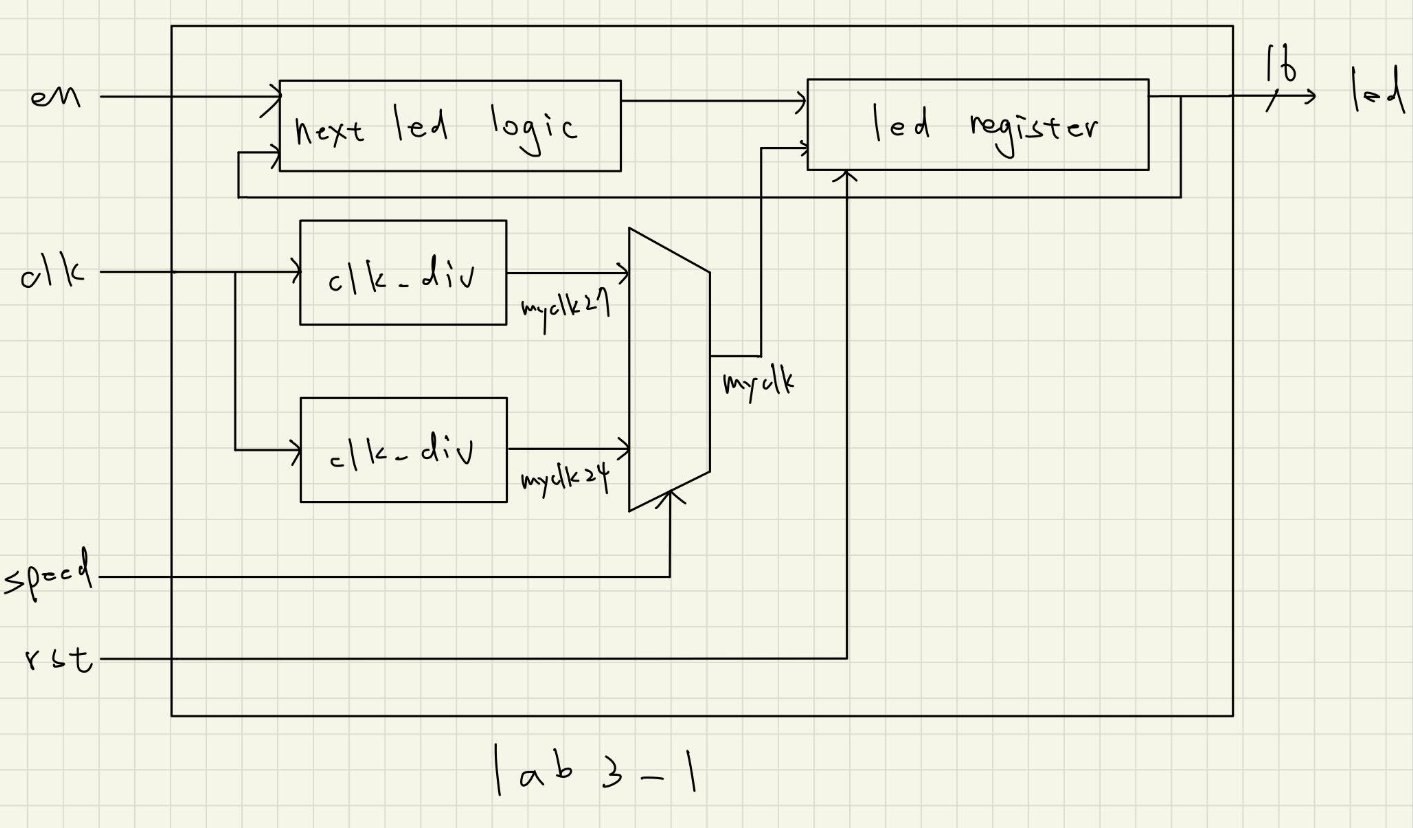
|  |  |
| --- | --- |
| **Lab 2** | |
| 學號: 110062131 | 姓名: 馬毓昇 |

1. **Lab Implementation**

Lab3\_1: 可以用clock\_divider調整速率的LED跑馬燈。



Lab3\_1 segment:

1. clock\_divider

一張含有 文字 的圖片

自動產生的描述

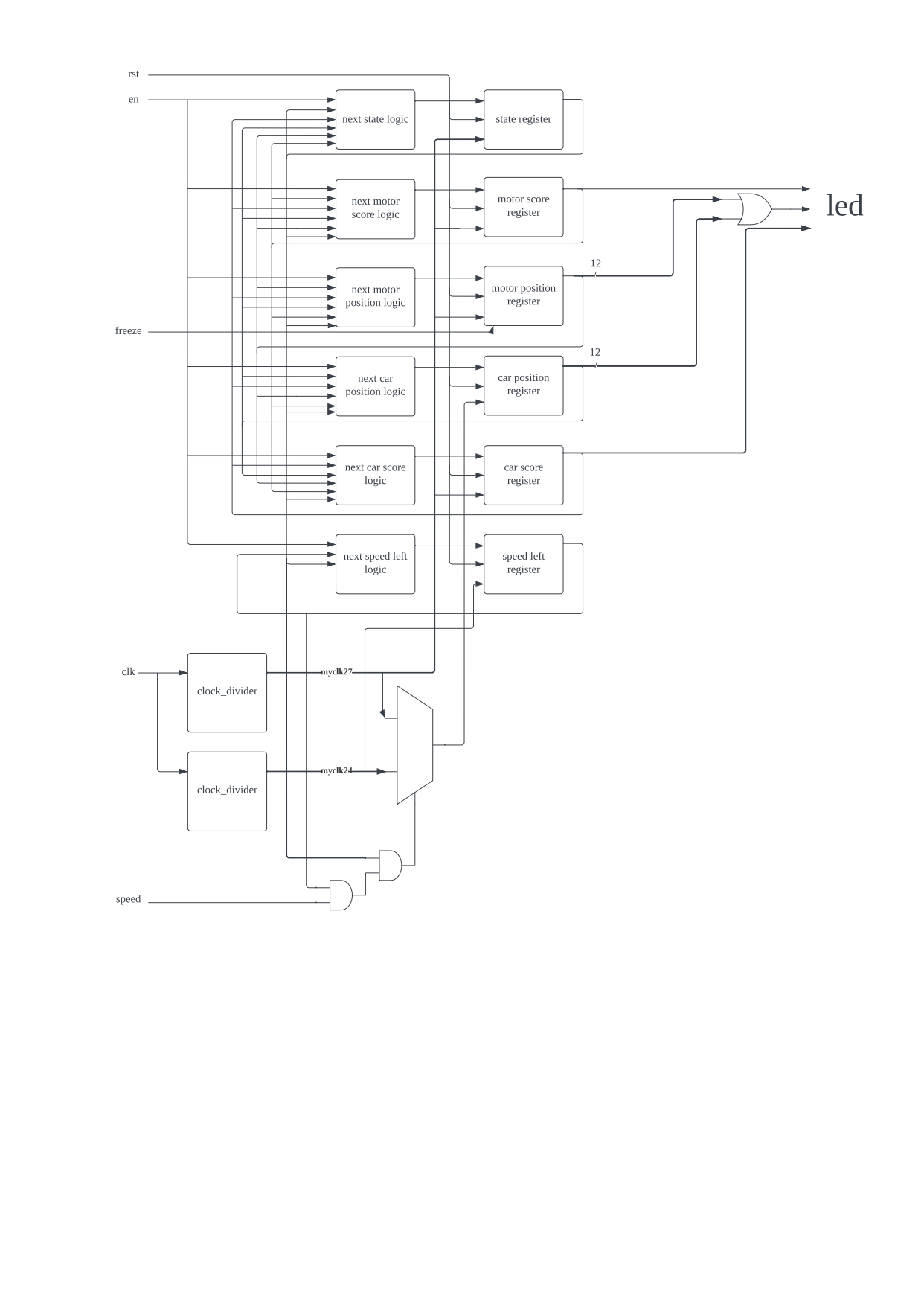
2. 用三元運算子來改變clk的頻率，也就是led燈的更新速度

一張含有 文字 的圖片

自動產生的描述



Lab3\_2: 一個LED賽車遊戲，配有記分板顯示以及freeze跟speed up兩個特殊功能。



Lab3\_2 segment:

1. 7 states，INIT\_WAIT做dummy state來實現勝利後新比賽一開始的等待



一張含有 文字 的圖片

自動產生的描述

2. 用concat的方式實作led接線





3. 車子移動：使用位元運算子”>>”來模擬車子往右走(以motor為例，car以此類推)



一張含有 文字 的圖片

自動產生的描述



4. freeze功能：freeze==1時讓car position不變(見上圖)

5. speed功能：全部的更新(state, motor position, motor score…)都用myclk27來更新。只有car position與speed\_left用myclk，myclk則是用三元運算子依speed\_left狀態(小於6大於0時，共5個cycle)來決定要接入myclk24或myclk27，來達成加速的效果。(state == RACING說明只有在賽跑時才能提速，其他狀態(INIT, FINISH, WIN)car position皆只能用myclk27更新，如此一來便不會影響到其他狀態的時序) 



一張含有 文字 的圖片

自動產生的描述



6. speed\_left用myclk24來算，初始化為6，到0停止，中間共經過5個cycle。RACING時每speed一cycle減一，如果回到speed == 0，依舊把speed\_left減到0為止。可以實現speed只要是HIGH一次就一定會加速滿5個cycle(不論加速中是否把speed變回LOW)與一回合只能用一次的效果。



一張含有 文字 的圖片

自動產生的描述

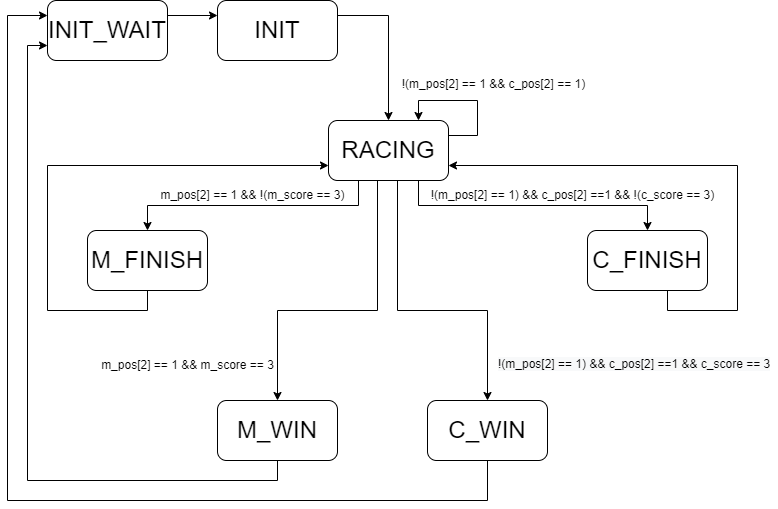


7. 算分，簡單的counter，car score進car finish時加一，motor同理

一張含有 文字 的圖片

自動產生的描述

Lab3\_2 FSM:



1. INIT -> RACING:

進INIT後的下個cycle直接進RACING。

2. RACING -> FINISH(M\_F, C\_F):

當各自到達終點(摩托車燈到led[2]or車子燈到led[2])而自己又還沒贏3場就進FINISH。

3. FINISH(M\_F. C\_F) -> RACING:

進FINISH後的下個cycle直接進RACING。

4. RACING -> WIN(M\_W, C\_W):

當達成各自勝利條件(摩托車到終點且已贏3場or車子到終點且已贏3場)就進WIN。

5. WIN(M\_W, C\_W) -> INIT\_WAIT:

進WIN後的下個cycle直接進INIT\_WAIT。

6. INIT\_WAIT -> INIT:

進INIT\_WAIT後的下個cycle直接進INIT。5, 6搭配實現勝利後新比賽的等待。



1. **Que stions and Discussions**

A: 做一個counter(叫freeze\_left)存freeze剩餘的cycle數，每一輪初始化freeze\_left = 3，比賽中freeze == 1時就每一cycle把freeze\_left減一，如果freeze\_left <= 0就覆蓋掉freeze的影響。

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



B: 我是做method 2，method 1我會用myclk24 + delay\_counter的方式設計，讓delay\_counter以myclk24更新，而其他部分(led control、car、motor)則會在myclk24 && delay\_counter == 3’b000時更新。至於車子加速的部分，則是把delay\_counter的判斷拿掉就可以了。

delay\_counter



myclk24

一張含有 文字 的圖片

自動產生的描述



如果delay\_counter==0就更新，不然就hold住

車子加速把delay\_counter判斷拿掉

一張含有 文字 的圖片

自動產生的描述



1. **Problem Encountered**

1. 最初一畫完FSM圖就直接把整份code打出來，但做出來的燈號呈現一個混亂的狀態，根本無從debug。後來把整份code都刪掉，從最簡單的幾個功能開始實作：單純的motor由左而右、motor到終點分數加一與復位、motor得到4分進勝利狀態。確保上述這三個功能與燈號是正常的之後，再一步步補上car的部分與freeze和最難的speed。依照這樣的步調慢慢實作花不到幾個小時做成功了，反觀我前一天一股腦地做，花了一整晚都沒做出來，算是學到了一個經驗。

2. FSM clocking一開始是嘗試做method 1的，但發現會有兩個問題：第一，無論如何car的動作都對不上motor，我想應該是因為我的實作方式是只有car跑myclk24，其餘全部還是跑myclk27，因此delay\_counter更新(依據myclk24)出來的慢8倍的myclk27都會比實際上的myclk27還要再慢幾個myclk24，所以應該要整份code都用同樣的myclk24 + delay\_counter邏輯才對。

3. 延續上面，第二，en=0再en=1會很明顯看得出來car相對motor的時序變調了，這問題很明顯就出在我原本的做法delay\_counter是用myclk24來更新的，相較於比較慢的myclk27，如果在myclk27是HIGH或LOW的過程中把en回復到1，delay\_counter(依據myclk24)就會偷跑更新，解決辦法應該也是跟第一一樣，整份code都要用myclk24 + delay\_counter的邏輯才行。

原本實作的錯誤方式，除了car與delay\_counter都是用myclk27，後來改做method 2了：

delay\_counter



myclk27

一張含有 文字 的圖片

自動產生的描述



myclk24

一張含有 文字 的圖片

自動產生的描述



如果delay\_counter==0就更新，不然就hold住

一張含有 文字 的圖片

自動產生的描述



1. **Suggestions**

這次lab沒有practice太難了，希望以後lab都能盡量有practice code讓我們參考。