

## Lab 4

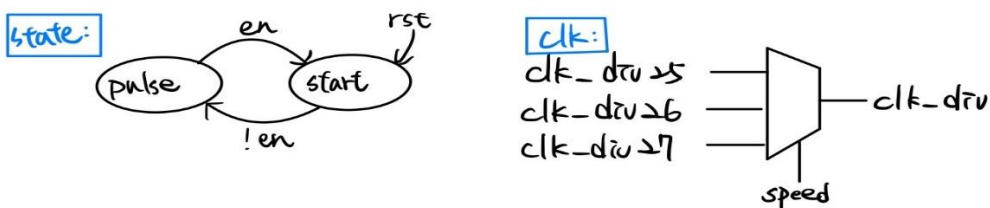
學號: 109000205

姓名: 蕭皓隆

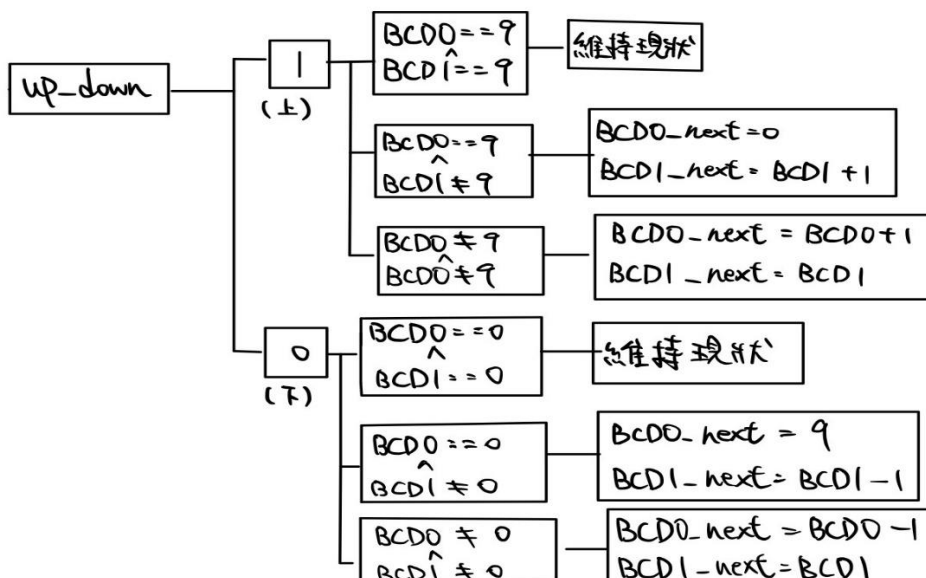
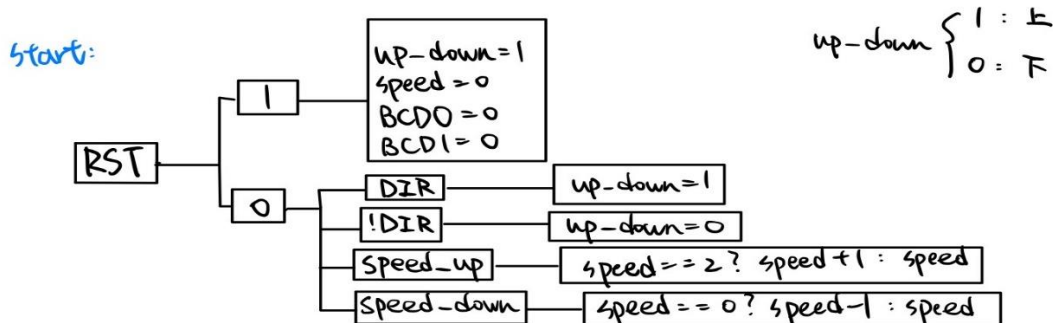
### 1. 實作過程

#### 4-1:

定義兩個 state，分別為執行(start)與暫停(pulse)，若  $en=1$  時  $state = start$ ， $en=0$  時  $state = pulse$ 。利用按鈕控制相對應的參數，再把參數傳入 counter，不讓按鈕直接控制 counter，避免邏輯混亂以及降低 debug 的難度。值得注意的第一點是在 4-1 由於 dir 按著時要下數，所以訊號不需經過 onepulse 處理，這樣才能達到想要的效果；第二點則是偵測按鈕的 clk 不能與 counter 的 clk 相同，這樣可能會造成按鈕無法被正常偵測。



**pulse:**  
所有變數維持現狀



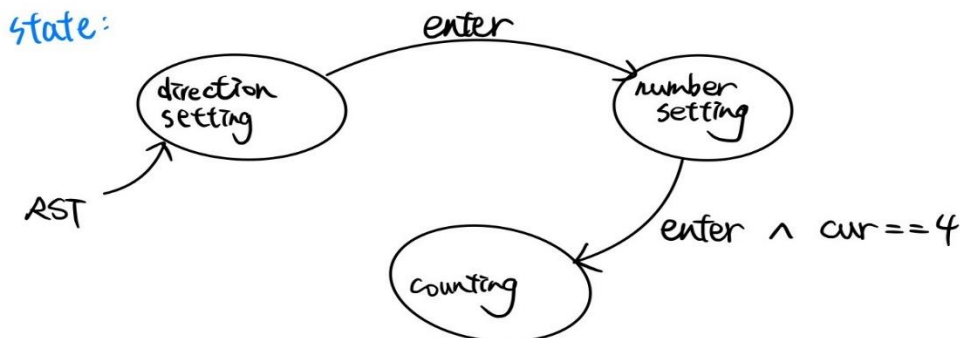
**4-2:**

利用 FSM 實作，分別為三個 state: Direction\_Setting, Number\_Setting, Counting。

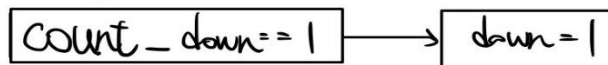
Direction\_Setting：偵測 count\_down 是否有被按下，若 count\_down==1 則 down=1 並讓 led 訊號等於 1 (down 用來記錄是否往下數)

Number\_Setting：先讓 7-segment 變成 0000，並利用 cur 紀錄目前調整的為第幾位數，並對當前位數做設定，並把當前數字指派給 goal，若 enter 按下則把 cur + 1，若 cur==4 又按下 enter 時則跳至 Counting\_state。

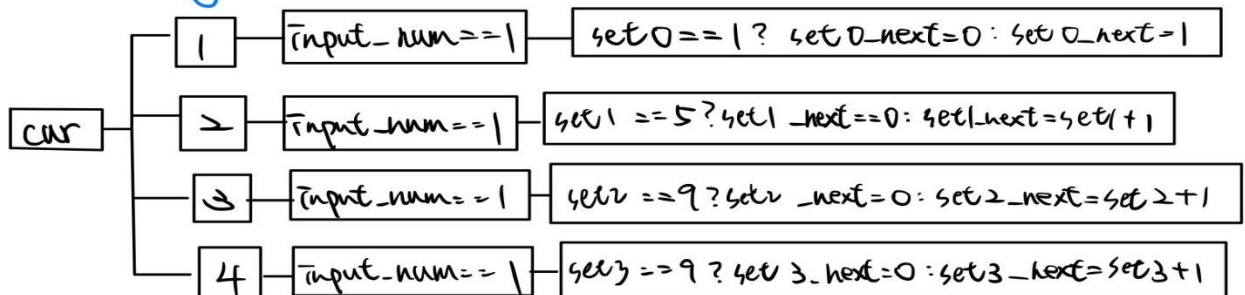
Counting：若 down==1 則先讓 7-segment 變成 goal，down==0 則變成 0000。若上數到 goal 則停下；若下數到 0000 停下來。上數隨時判斷各位數是否有到達 goal。



direction-setting:



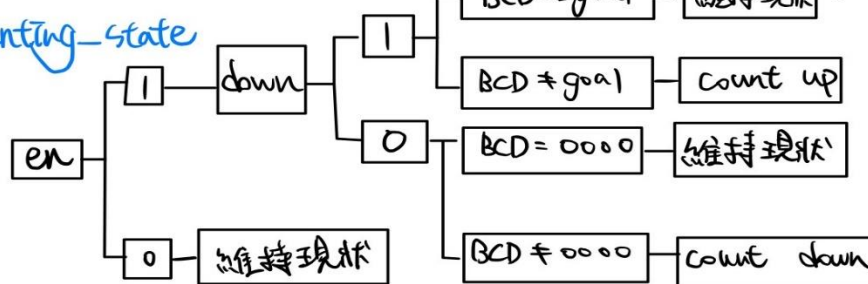
number-setting:



goal {  
 0 = set0  
 1 = set1  
 2 = set2  
 3 = set3

down {  
 1 : BCD {  
 0 : 0  
 1 : 1  
 2 : 2  
 3 : 3  
 0 : goal0  
 1 : goal1  
 2 : goal2  
 3 : goal3

counting-state



## 2. 學到的東西與遇到的困難

學到的東西：學到 onepulse 跟 debounce 的原理以及實作方法，也了解到甚麼時候該讓訊號經過 onepulse 處理甚麼時候不用。另一個學到的重要觀念是 clock\_divider，在 n 等於多少時能正常顯示 7-segment 以及能讓 counter 以正常速度運作都要經過摸索。

遇到的困難：剛開始有點難適應 7-segment 的邏輯，會不知道從何下手，但看到能正常顯示之後蠻有成就感的。

## 3. 想對老師或助教說的話

大家都知道酒精吧，乙醚就是乙醇加硫酸製成的醚類。那大家知道甲醚嗎？

甲醚就是當你不想吃飯的時候就可以吃麵(呷咪)