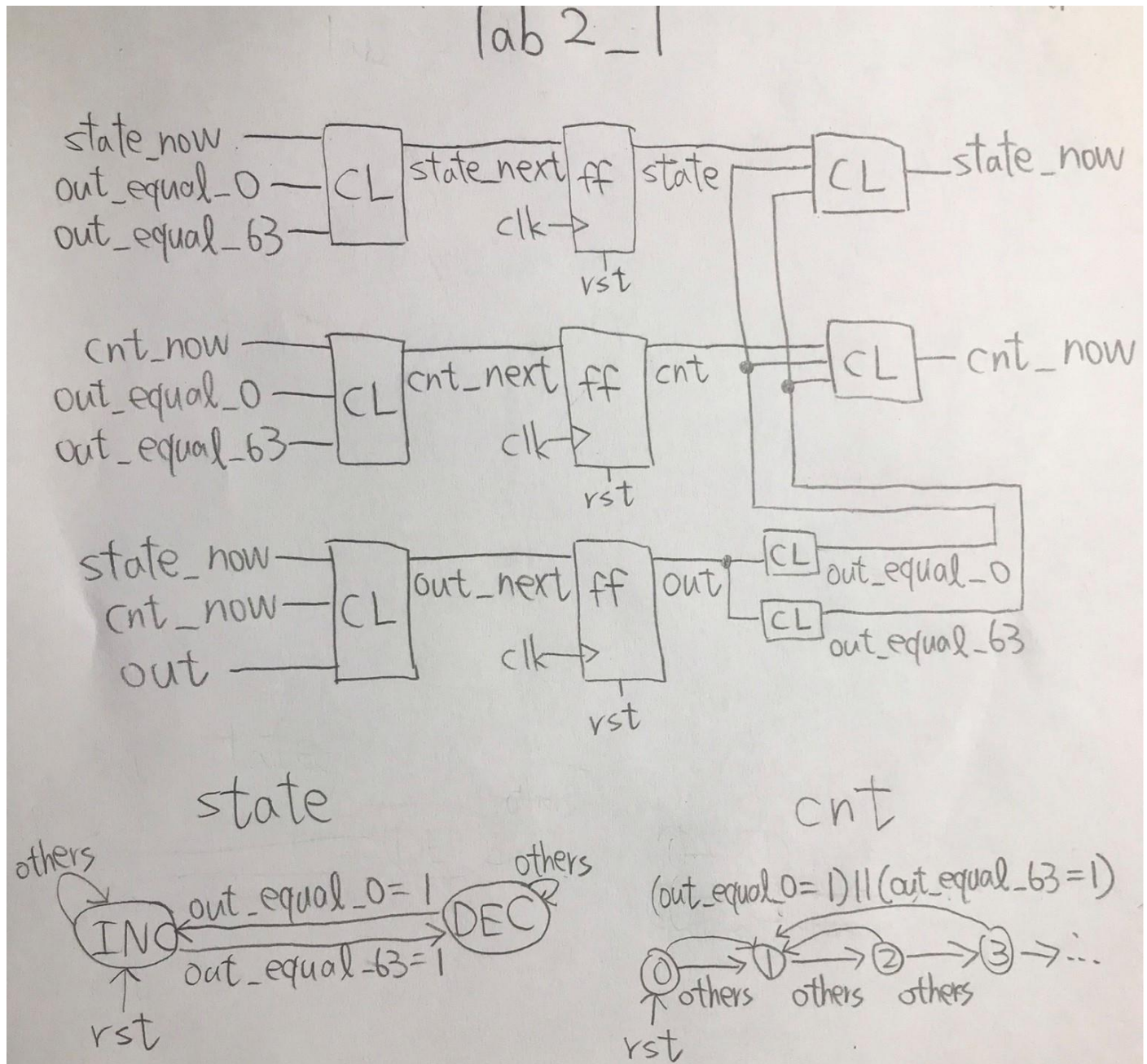


## Lab 2

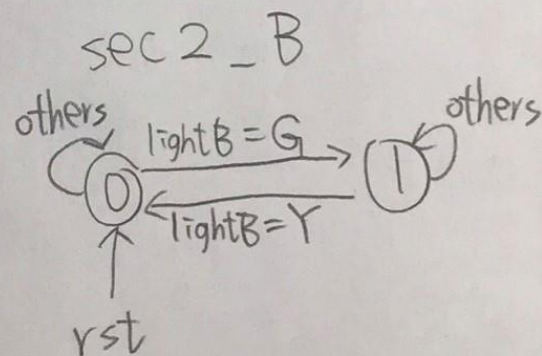
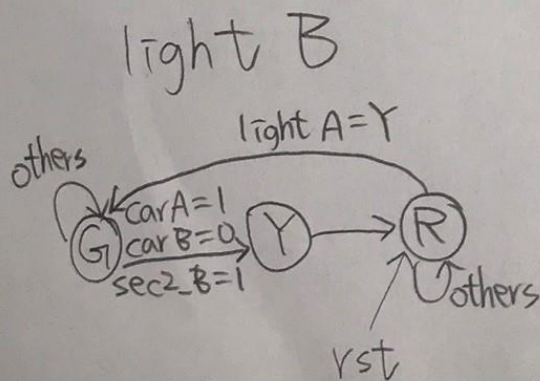
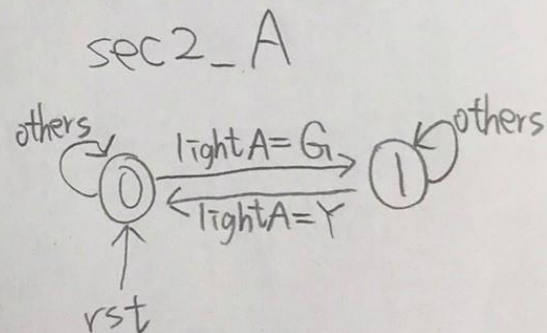
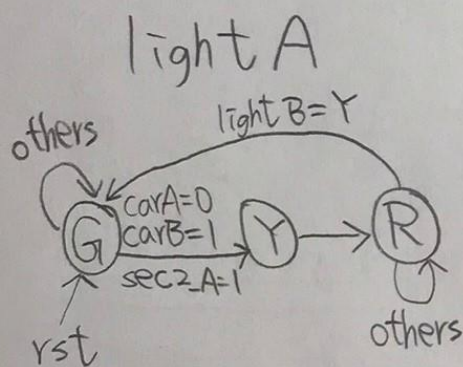
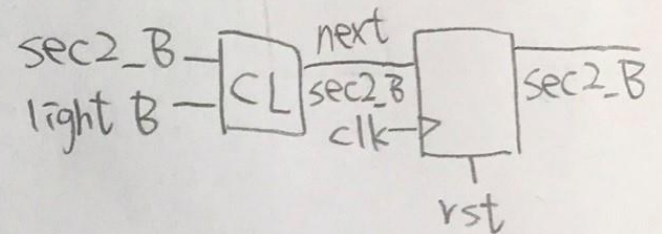
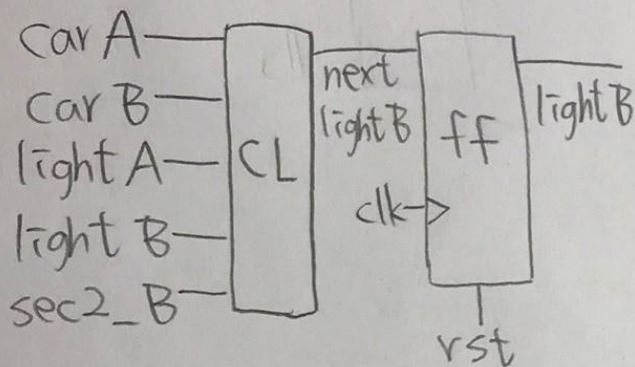
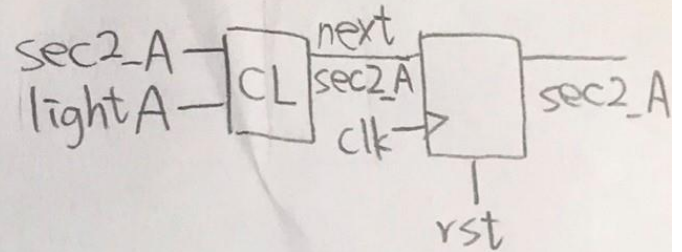
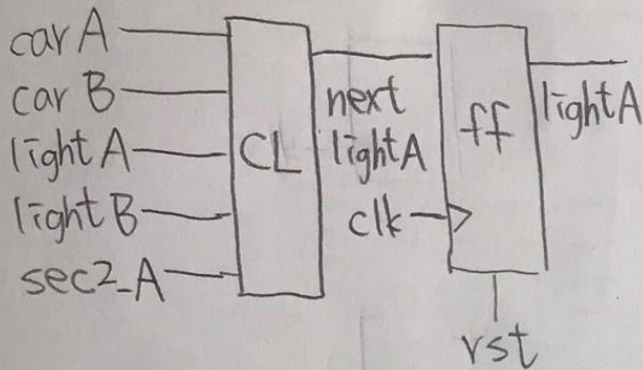
學號: 108032053

姓名: 陳凱揚

### 1. 實作過程



## lab 2\_2



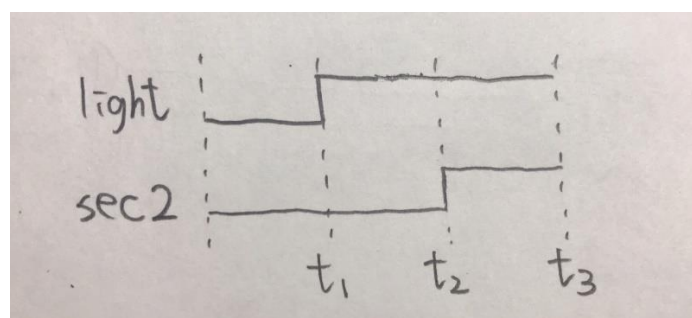
在 lab2\_1 中，我製作了 3 個 flip-flop，分別是 state(紀錄目前狀態是增加還是減少)、cnt(紀錄目前 n 的值)、out(輸出結果)，而我設定 out 為 0 時，state 應該是 INC(增加)、cnt 為 0；out 為 63 時，state 應該是 DEC(減少)、cnt 為 0。state 和 cnt 都需要根據最新的 out 值來改變，因此我增加了兩個訊號 state\_now 和 cnt\_now，並在計算 state\_next、cnt\_next、out\_next 時，使用這兩個訊號，而不是原先的 state 和 cnt。畫出 block diagram 後，我先將 sequential circuit 的 3 個 flip-flop 以一個 always block 描述，接著再分別描述每一個 combinational circuit，就能夠很快地以 verilog 描述完整設計。而設計 testbench 時，我在一開始先拉起 rst，測試是否所有狀態都回到初始值且不改變，接著完整跑一遍整個週期(0, 1, ..., 63, 62, ..., 0, 1, ...)，測試數列的順序是否正確，最後再分別於 state 的兩個不同狀態(INC, DEC)時，拉起 rst，測試是否回到初始狀態。

在 lab2\_2 中，我製作了 4 個 flip-flop，分別是 lightA、lightB、sec2\_A、sec2\_B，並依據題目敘述畫出他們的 state diagram，比較需要注意的部分是只要 light 是綠燈，對應的 next\_sec2 就會是 1；只要 light 是黃燈，對應的 next\_sec2 就會是 0。

## 2. 學到的東西與遇到的困難

我覺得 lab2\_1 裡最重要的部分在於如何轉換 state，因為必須等到 out 出來後，才能知道下一個週期是否需要轉換 state，並將 cnt 歸零，如果沒有 state\_now 和 cnt\_now 的話，計算就會發生錯誤。另外，題目給的關係式是  $a_n$  和  $a_{n-1}$ ，要先轉換為  $a_{n+1}$  和  $a_n$ ，比較方便轉換成程式碼。

在 lab2\_2 裡，我覺得先完全看懂題目敘述非常重要，如果能完全清楚燈號轉換的邏輯，state diagram 就能很快地畫出來，code 部分也能很快產生，整個設計過程會非常有效率。另外，我覺得題目敘述的綠燈亮 2 個 cycle 是一個小陷阱，下意識會想判斷  $\text{sec} == 2$  是否成立，但實際上，只要使用一個 1bit 且由 clk trigger 的 sec2，當綠燈亮時  $\text{next\_sec2} = 1$ ，黃燈亮時  $\text{next\_sec2} = 0$ ，就能達到判斷 2 個綠燈 cycle 的效果，如下圖 t3 時，綠燈就恰好亮了 2 個 cycle。



### 3. 想對老師或助教說的話

放連假時還要辛苦上班的人叫做甚麼？

- .
- .
- .
- .
- .
- .
- .
- .
- .
- .
- .
- .
- .
- .
- .
- .
- .
- .
- .
- .

廉價勞工。