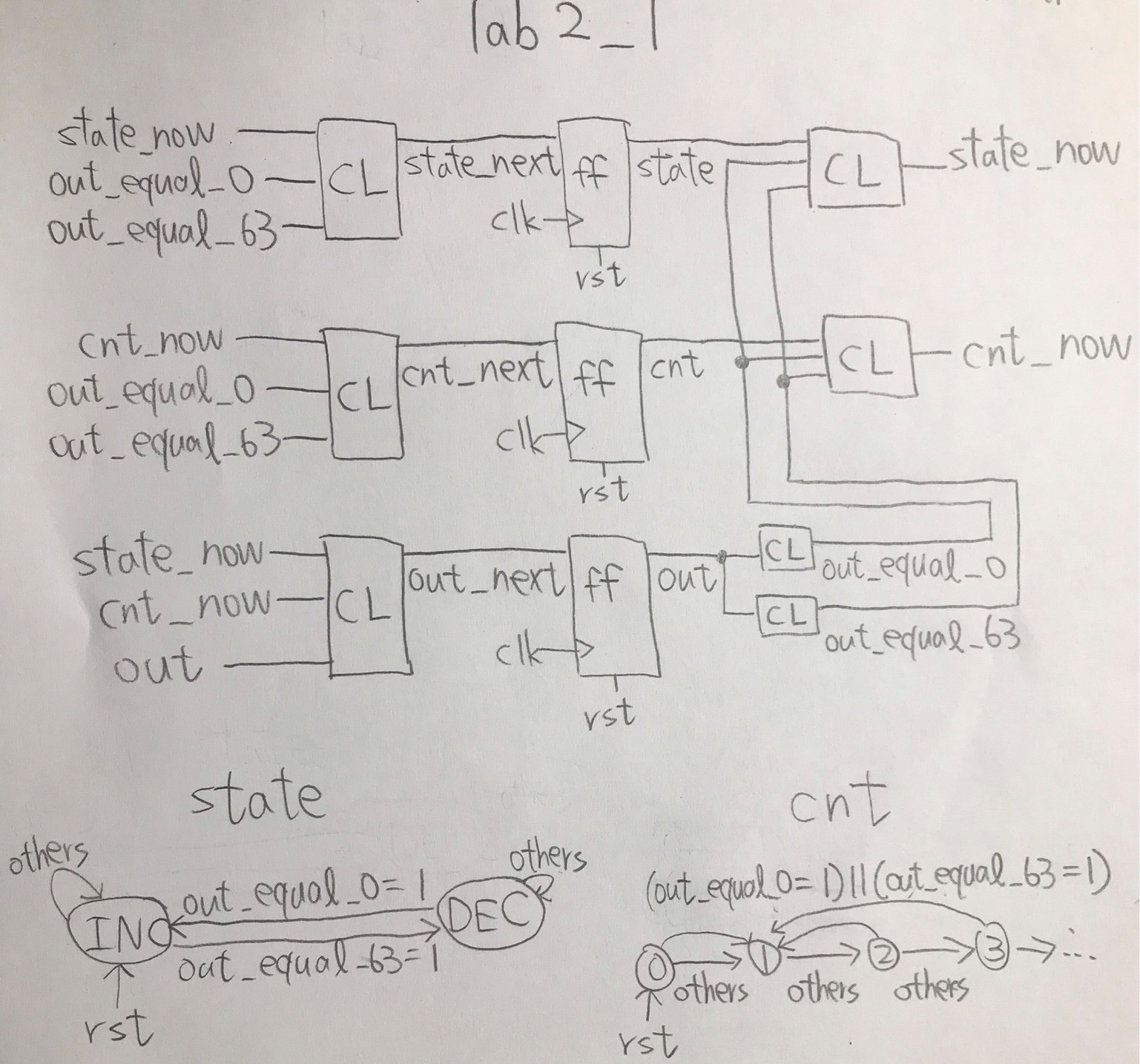
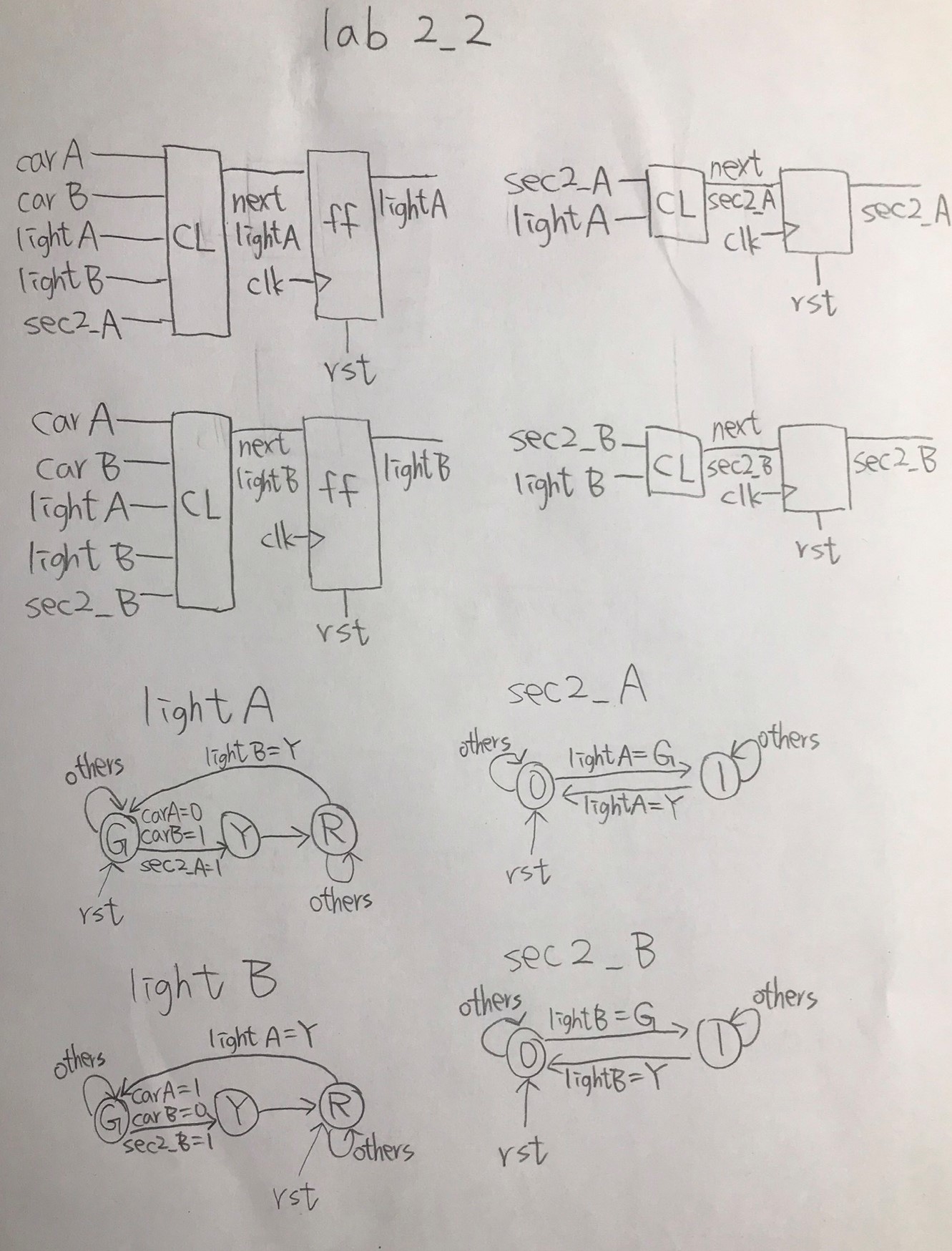
|  |  |
| --- | --- |
| **Lab 2** | |
| 學號: 108032053 | 姓名: 陳凱揚 |

1. 實作過程





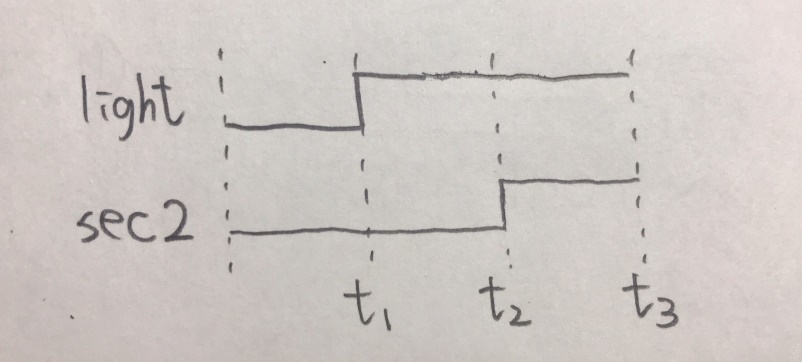
在lab2\_1中，我製作了3個flip-flop，分別是state(紀錄目前狀態是增加還是減少)、cnt(紀錄目前n的值)、out(輸出結果)，而我設定out為0時，state應該是INC(增加)、cnt為0；out為63時，state應該是DEC(減少)、cnt為0。state和cnt都需要根據最新的out值來改變，因此我增加了兩個訊號state\_now和cnt\_now，並在計算state\_next、cnt\_next、out\_next時，使用這兩個訊號，而不是原先的state和cnt。畫出block diagram後，我先將sequencial circuit的3個filp-flop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描述完整個設計。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。

1. 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，題目給的關係式是和，要先轉換為和，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next\_sec2 = 1，黃燈亮時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。



1. 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。