

# Lab 1

學號: 109062202

姓名: 陳禹辰

## 1. 實作過程

Lab2\_1 :

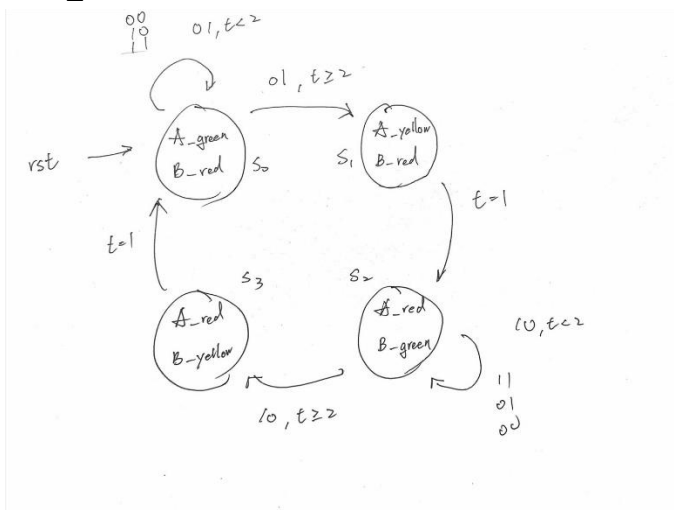
```

always @(*) begin
  case (state)
    sequence_a:
      if(out == 63)begin
        next_state = sequence_b;
        out_next = 62;
      end else begin
        next_state = sequence_a;
        out_next = (out <= cnt) ? out + cnt : out - cnt;
      end
    sequence_b:
      if(out == 0)begin
        next_state = sequence_a;
        out_next = 1;
        cnt = 1;
      end else begin
        next_state = sequence_b;
        out_next = out - 2 ** (cnt - 58);
      end
  endcase
end

```

Lab2\_1 我是用 FSM 的方法去寫，分成兩個 state，分別是 sequence\_a 代表從 0 到 63 跟 sequence\_b 從 63 到 0，然後在用 sequence block 讓他在每個 posedge 時觸動，將 out\_next assign 到 out 上。Lab2\_1 的 testbench 我則是讓他在向上數時 rst 一次然後就讓 counter 一直跑了。

Lab2\_2 :



Lab2\_2 我則是設計有四個 state，分別代表 A 綠燈 B 紅燈、A 黃燈 B 紅燈、A 紅燈 B 綠燈、A 紅燈 B 黃燈四種，然後再透過輸入來判斷什麼時候要換 state 並在每個 posedge 時將下一個 state 輸入，做法跟 lab2\_1 差不多。

## 2. 學到的東西與遇到的困難

遇到的困難是如果設計的 FSM 將輸出直接寫到 combination block 內然後又用 input 來判斷什麼時候要換 state，那在 input 有改變時，就有可能連帶影響到 output 也跟著改變，所以我後來改成在 posedeg 時才判斷 state 的變化，但感覺應該是可以再多寫一個 always block 來把事情分開做。這次是學到了 FSM 如何用 verilog 設計，以及 counter 的設計及應用。

## 3. 想對老師或助教說的話

這次因為是兩周用兩個 lab，再加上 lab1 的難度其實不高，所以在時間上還蠻充裕的，能夠花足夠的時間處理並找到問題。

附上笑話

