|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | 258057-15121013434242 | | 2022/10/05  實驗四  序向邏輯練習  姓名：王嘉羽 學號：00957116  班級：資工3B  E-mail：vayne20011125@gmail.com | |  | |  | |  | | --- | | 注意繳交時一律轉PDF檔繳交期限為上完課後當週五晚上12點前一人繳交一份檔名：學號\_HW?.pdf檔名請按照作業檔名格式進行填寫未依照格式不予批改 | |  | |  | |

1. **Counter + register**

* **實驗說明：**

設計2個controller和1個加法器，將其中一個counter數到9，另一個counter數到4，然後相加，最後存入一個暫存器w中，輸出暫存器w的值

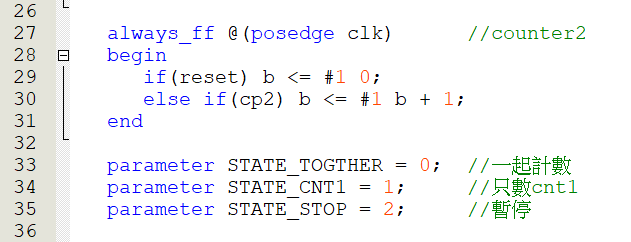
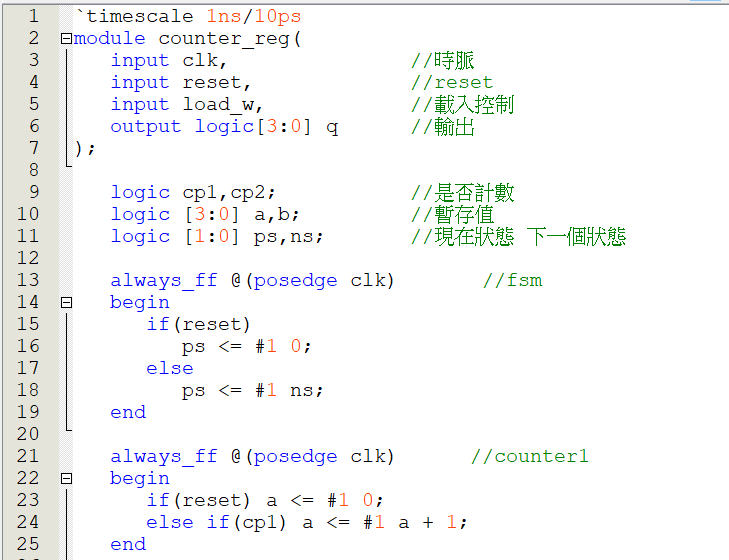
* **系統硬體架構方塊圖（接線圖）：**

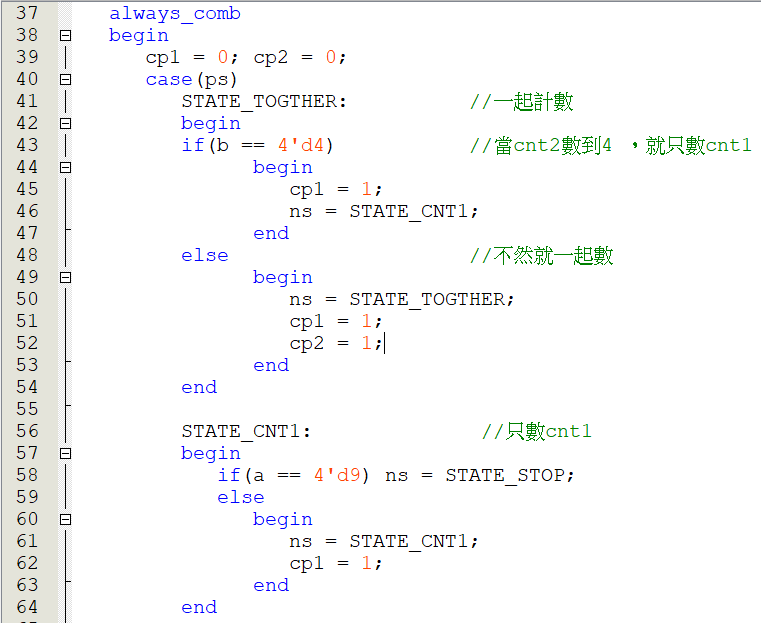


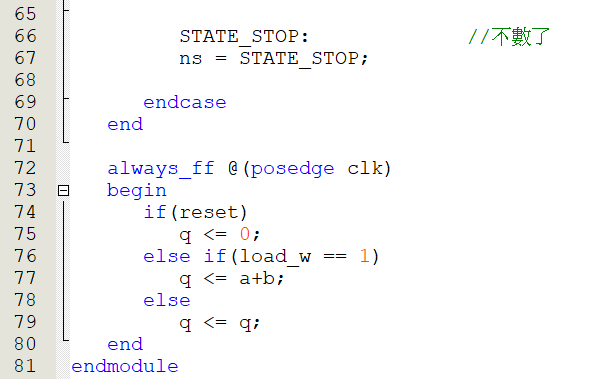
* **系統架構程式碼、測試資料程式碼與程式碼說明(.sv檔及.do檔都要截圖)**

**截圖請善用win+shift+S**

* 硬體程式碼



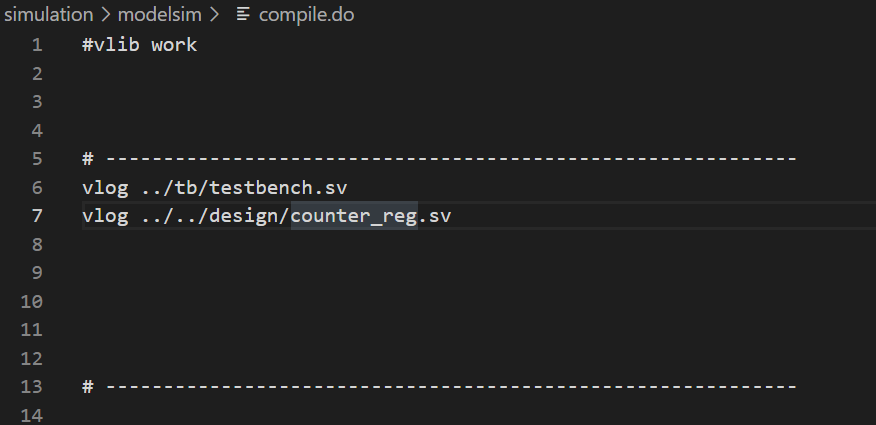




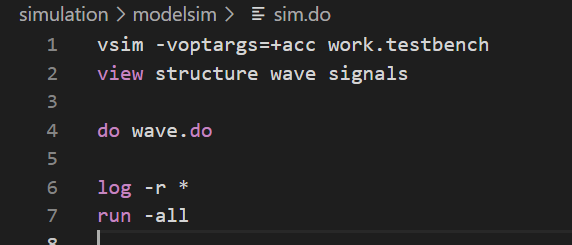
* testbench



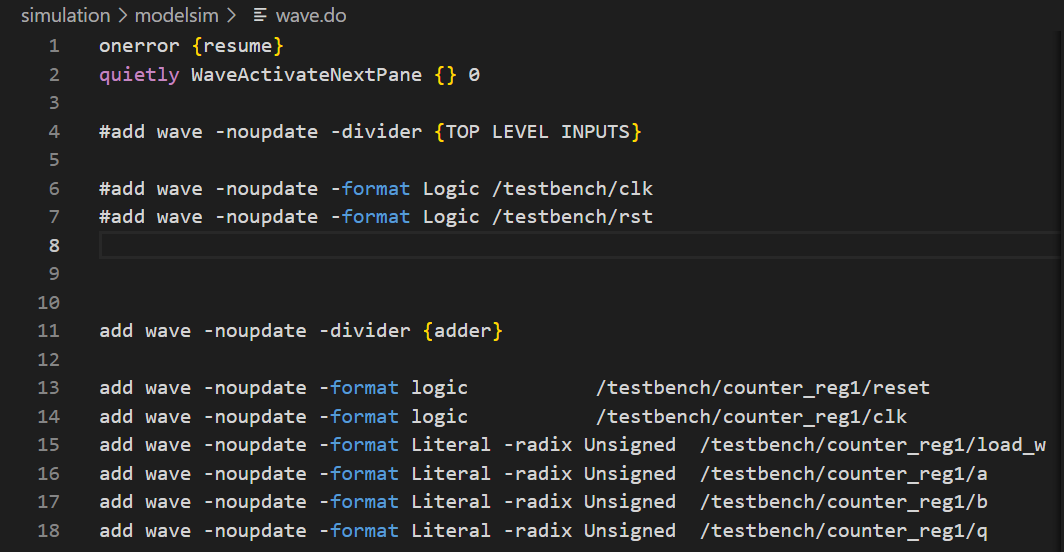
* compile.do



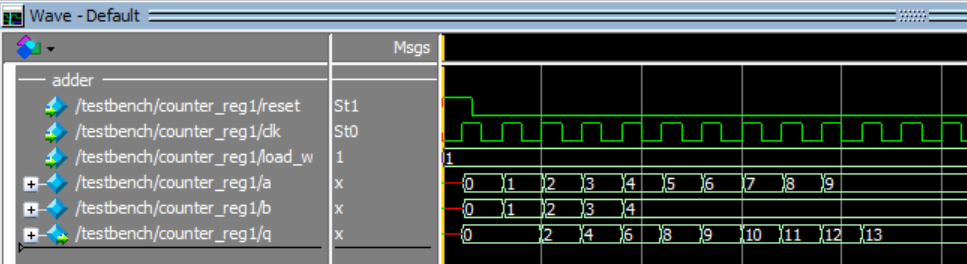
* sim.do



* wave.do



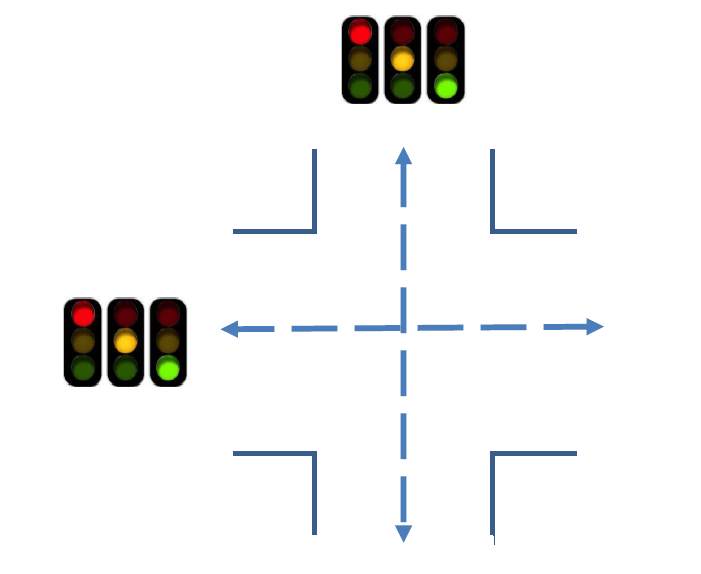
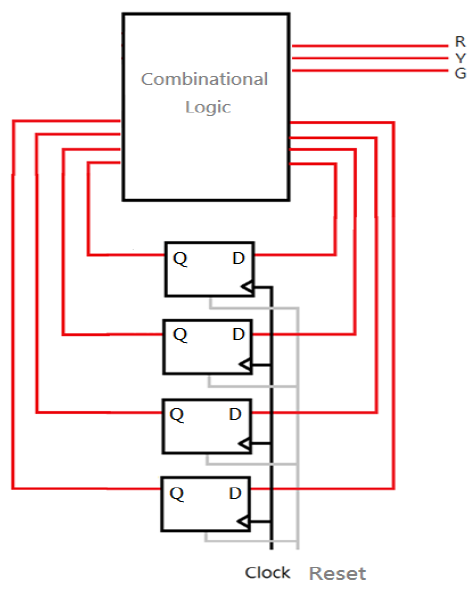
* **模擬結果與結果說明：**

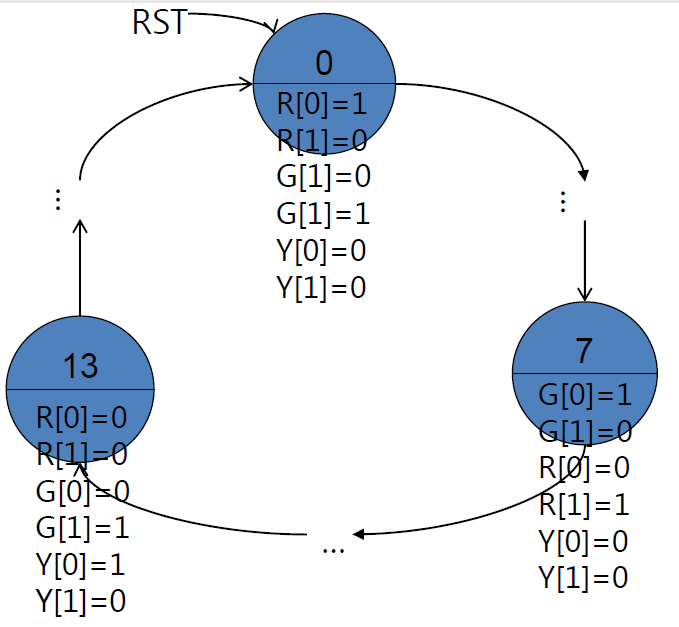


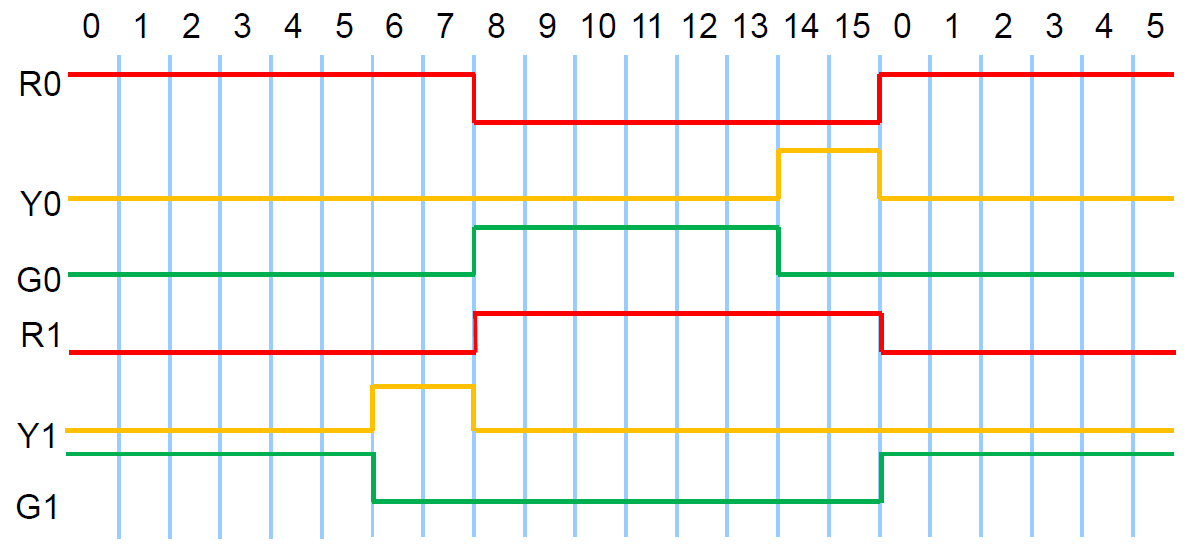
a(cnt1)從0~9，b(cnt2)從0~4然後加起來，當load\_w = 1時，會將上一次的結果載入

1. **紅綠燈**

* **實驗說明：**
  1. 用FSM(Finite State Machine)實作紅綠燈
  2. 第一組紅綠燈(R[0], Y[0], G[0]) 由紅燈為起點依序變換為 綠燈→黃燈..
  3. 第二組紅綠燈(R[1], Y[1], G[1]) 根據地一組紅綠燈的狀態顯示 綠燈→黃燈→紅燈..
  4. 以R表示紅燈、Y表示黃燈、G表示綠燈。
  5. 1表示燈亮，0表示燈滅，最後輸出[1:0]R、[1:0]Y、[1:0]G。
* **系統硬體架構方塊圖（接線圖）：**

****

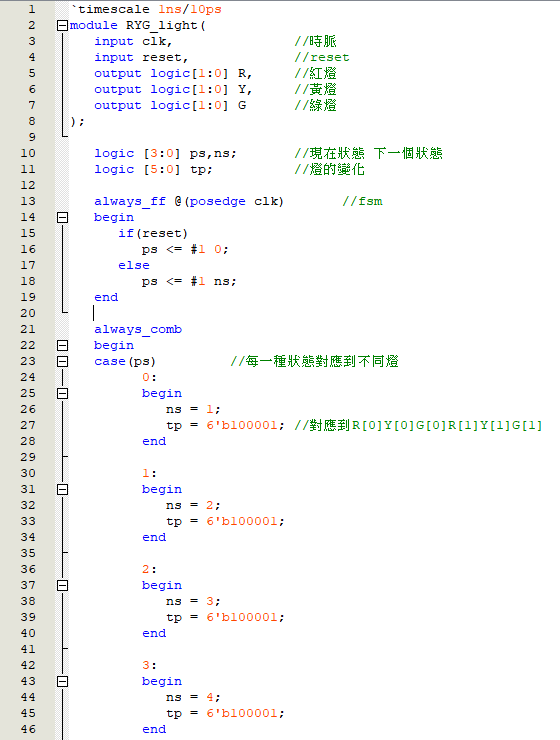
****

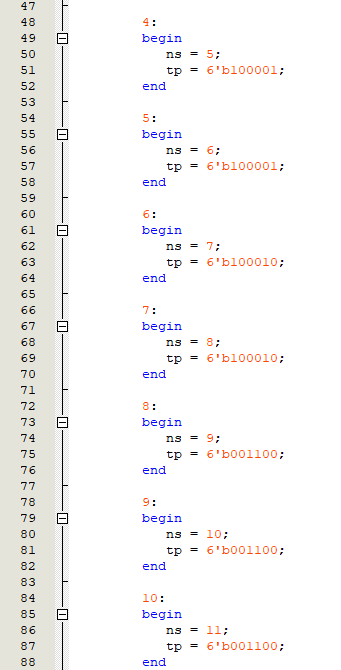
****

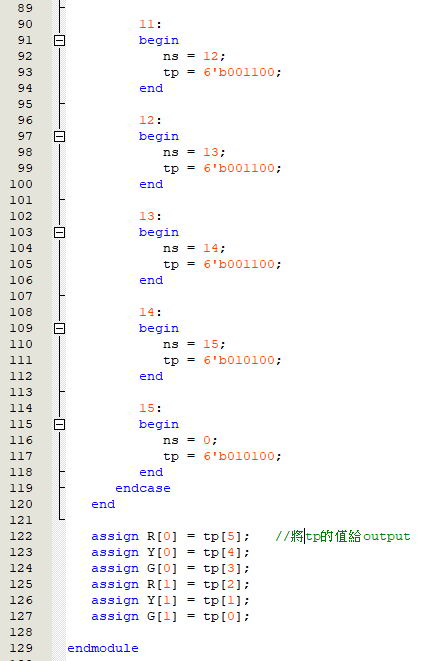
* **系統架構程式碼、測試資料程式碼與程式碼說明(.sv檔及.do檔都要截圖)**

**截圖請善用win+shift+S**

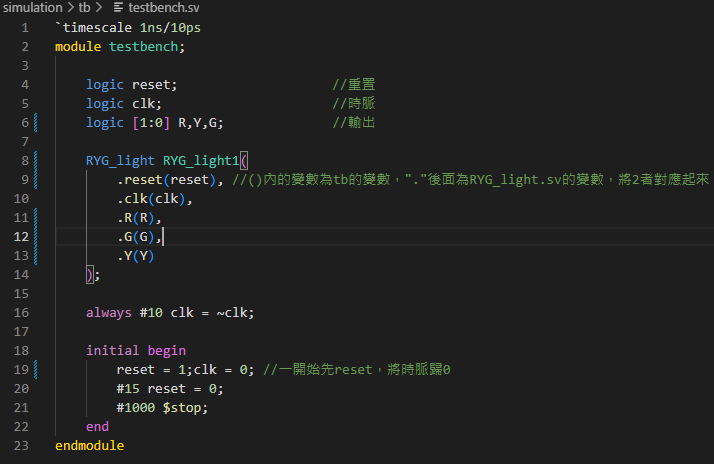
* 硬體程式碼



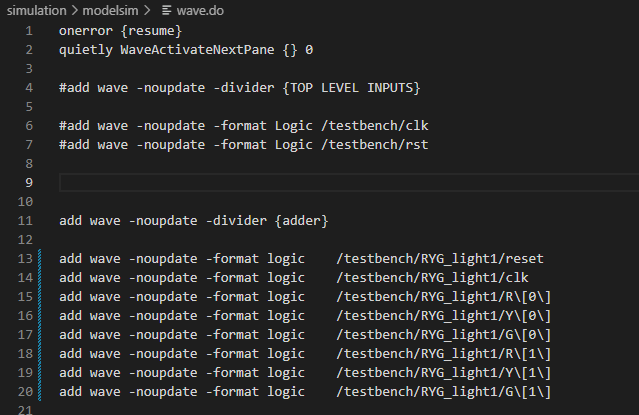




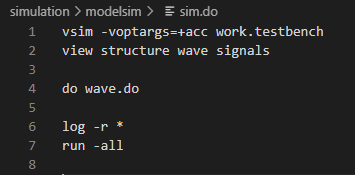
* testbench



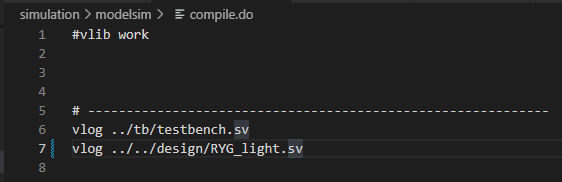
* wave.do



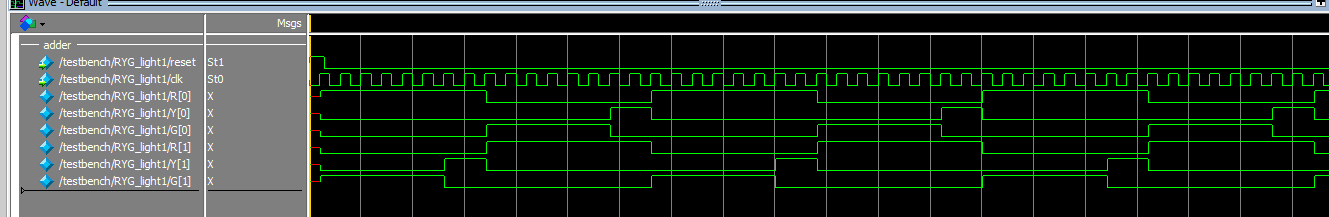
* Sim.do

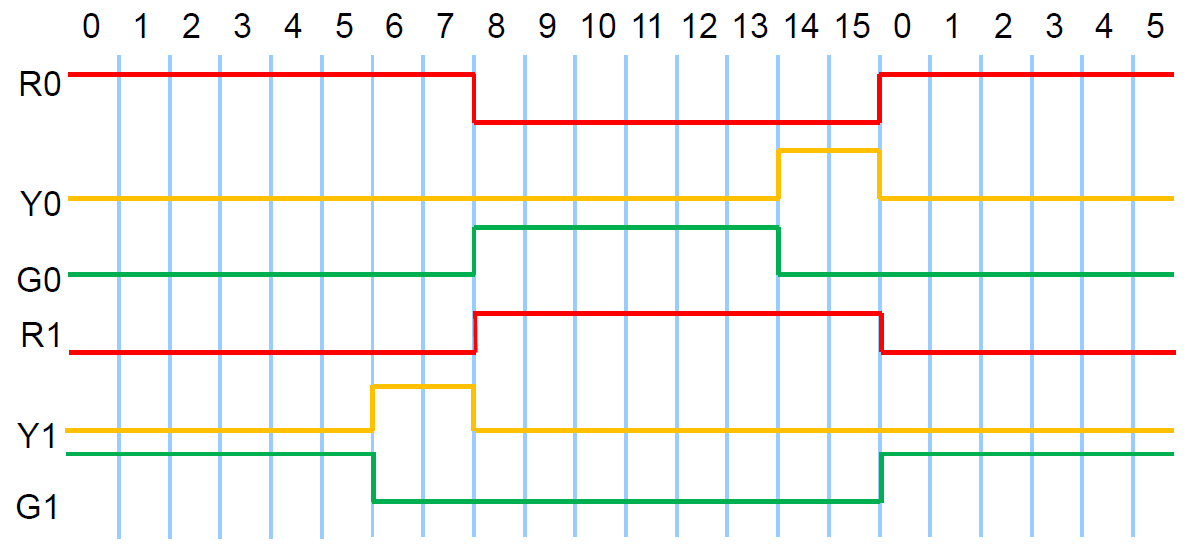


* Compile.do

****

* **模擬結果與結果說明：**

****

****

和模擬圖長得一模一樣!!!

* **結論與心得：**

寫了紅綠燈後更加了解fsm的強大之處，如果沒有用fsm，code一定會很複雜。Fsm很單純的就是把狀態寫出來，然後照著狀態給值，就成功了!這個單元也是我最喜歡的單元，我覺得很好玩!!­­­­