VSD HW3

F64091130 楊采語

1. Design

本次作業主要設計一個十字路口的紅綠燈，分別設為東西方向及南北方向，主要情況分為 4 種：東西方向紅燈時，南北方向可能是綠燈或黃燈，而當東西方向變綠燈時，南北方向會轉為紅燈，最後東西方向會進入黃燈，再轉為綠燈。另一種特殊情況是當 reset 訊號輸入時，會將狀態回到初始狀態(idle)，然後再接著將東西方向設為紅燈，南北方向設為綠燈。

2. I/O specifications

Pin name	**Input/ Output**	**Bit Width**	**Description**
clk	Input	1	system clock signal
reset	Input	1	active high synchronous reset signal
EW_Red	Output	1	東西方向紅燈
EW_Green	Output	1	東西方向綠燈
EW_Yellow	Output	1	東西方向黃燈
NS_Red	output	1	南北方向紅燈
NS_Green	output	1	南北方向綠燈
NS_Yellow	output	1	南北方向黃燈

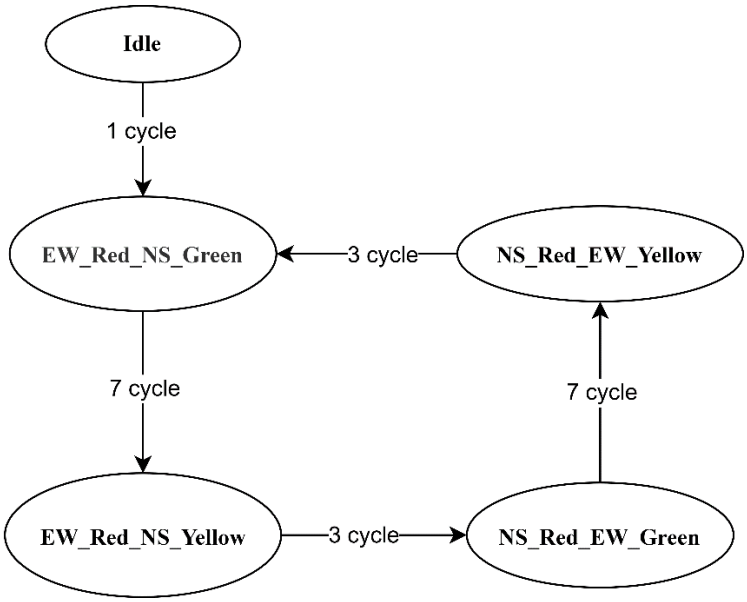
`define G_time 6'd6

`define Y_time 6'd2

3. Testbench

reset	一開始先 reset (100ns)
clk	週期為 100ns(50ns 翻轉 clk 的值)
信號顯示	negedge clk display 當前燈號狀態(如 EW_Red 表示東西方向紅燈亮起)
$finish	在 6000ns (6us)時結束

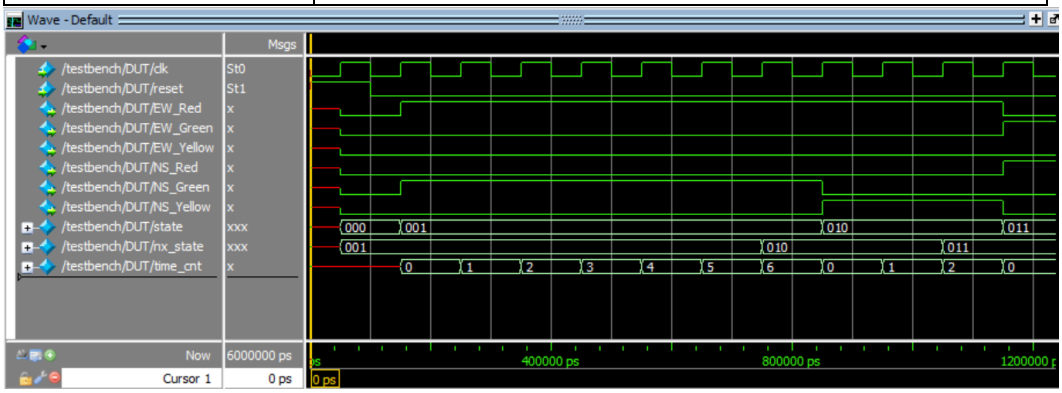
4. State diagram



5. Simulation results

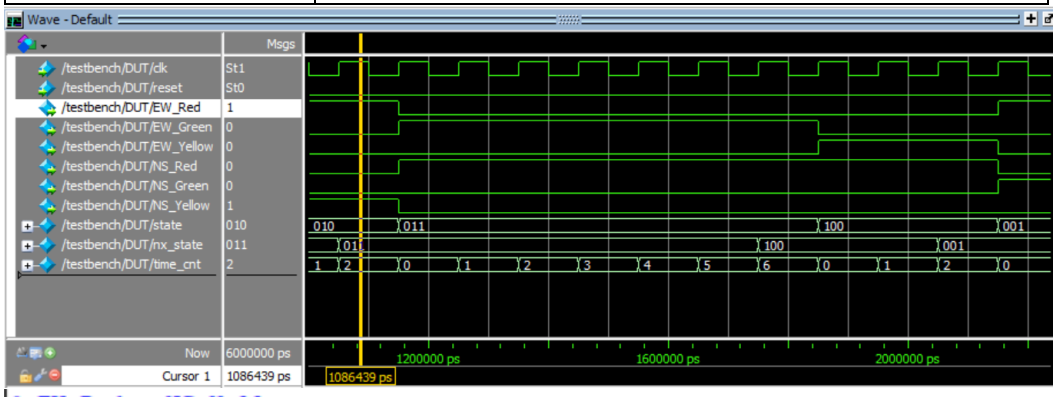
```
          V C S   S i m u l a t i o n   R e p o r t|
Time: 6000000 ps
CPU Time:      0.340 seconds;       Data structure size:   0.0Mb
Mon Oct 28 11:19:17 2024
CPU time: .353 seconds to compile + .427 seconds to elab + .296 seconds to link + .381 seconds in simulation
```

state	Explanation
Idle	一開始的 reset 訊號(第一個 cycle)
EW_Red_NS_Green	東西方向紅燈、南北方向綠燈個 7 個 cycle
EW_Red_NS_Yellow	東西方向紅燈、南北方向黃燈 3 個 cycle



```
# Wait
# Wait
# EW_Red & NS_Green
# EW_Red & NS_Green
# EW_Red & NS_Green
# EW_Red & NS_Green
# EW_Red & NS_Green
# EW_Red & NS_Green
# EW_Red & NS_Green
# EW_Red & NS_Yellow
# EW_Red & NS_Yellow
# EW Red & NS Yellow
```

state	Explanation
EW_Red_NS_Yellow	東西方向紅燈、南北方向黃燈 3 個 cycle
NS_Red_EW_Green	南北方向紅燈、東西方向綠燈個 7 個 cycle
NS_Red_EW_Yellow	南北方向紅燈、東西方向綠燈個 3 個 cycle
EW_Red_NS_Green	東西方向紅燈、南北方向綠燈 7 個 cycle




```
# EW_Red & NS_Yellow
# EW_Green & NS_Red
# EW_Green & NS_Red
# EW_Green & NS_Red
# EW_Green & NS_Red
# EW_Green & NS_Red
# EW_Green & NS_Red
# EW_Green & NS_Red
# EW_Yellow & NS_Red
# EW_Yellow & NS_Red
# EW_Yellow & NS_Red
```

```
# EW_Red & NS_Green
# EW_Red & NS_Green
# EW_Red & NS_Green
# EW_Red & NS_Green
# EW_Red & NS_Green
# EW_Red & NS_Green
# EW_Red & NS_Green
# EW_Red & NS_Yellow
# EW_Red & NS_Yellow
# EW_Red & NS_Yellow
```

6. Discussion

原本在燈光控制時(切換燈的狀態)是使用 posedge clk 來控制，會出現延遲一個 cycle 的狀況，後來發現是因為這個 state 在 posedge clk 才會更新，而燈光控制會吃不到這個 state 這次換的值，所以燈光控制會在下一個 cycle 才被更新，而不會跟 state 一起更換。所以現在把它用組合邏輯 (always @ (*))來根據 state 更換燈光控制的訊號，讓 state、counter、燈光顯示同步。