

## Homework #4

E14096724 鄭喆嚴

- (1) A brief description of your design, including its particular functionality.

要設計一個 traffic light controller 並且 state diagram 至少要有 5 states，我設計的是一個輸入訊號有 clk、reset，輸出訊號有 Horizontal\_Green、Horizontal\_Yellow、Horizontal\_Left、Horizontal\_Red、Vertical\_Green、Vertical\_Yellow、Vertical\_Left、Vertical\_Red。HG，Horizontal 代表的是水平的路、Vertical 代表的是垂直的路，Green 代表的是綠燈、Yellow 代表的是黃燈、Left 代表的是紅燈加上左轉燈、Red 代表的是紅燈。在水平的路和垂直的路皆有四種燈號時再加上一開始的閒置狀態一共會有 7 states，而由於有 7 個 state 所以 state 和 next\_state 兩個都必須要是 3-bits，並且有使用一個 Counter 來計算各個燈號的時間。

一開始還沒 reset 時是閒置狀態 S0 全部燈號都是暗的，reset 完第一個 state 是 S1 是水平的路是綠燈而垂直的路是紅燈，在 Counter 數完綠燈的時間後會跳到下一個 state 是 S2 是水平的路是黃燈而垂直的路是紅燈，會一直跳到下一個 state 並循環回 S0

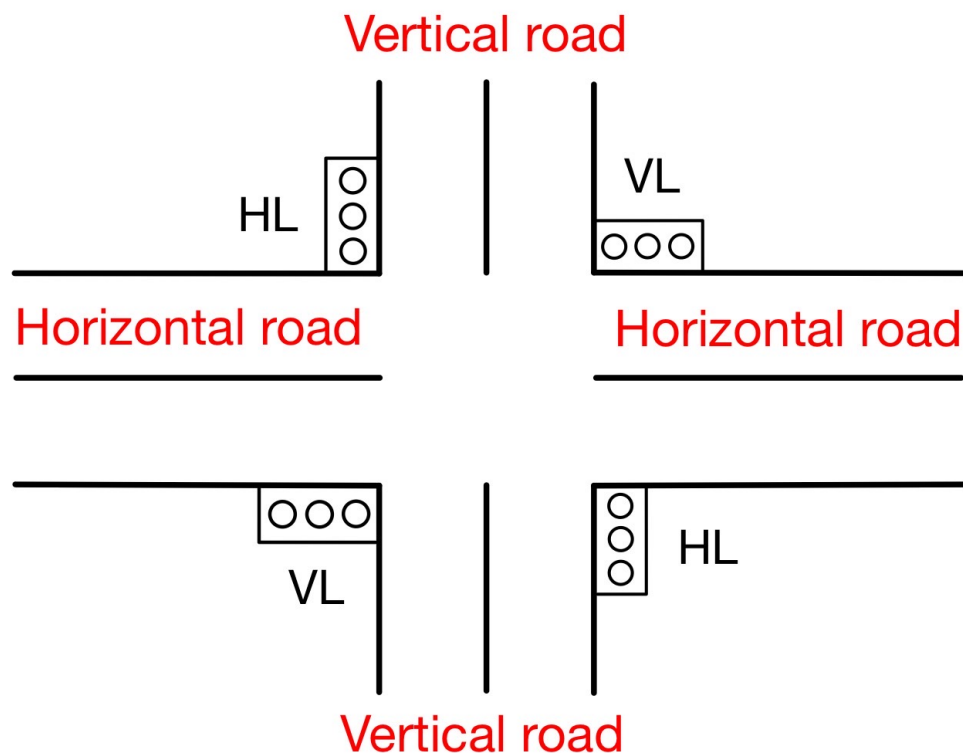


Fig. 1. traffic light controller 的路口示意圖

(2) Testbench and Timing specification for your design

我設計的每個燈號的時間是 horizontal road 的綠燈為 30 秒、黃燈為 5 秒、左轉燈為 10 秒，而 horizontal road 的紅燈是當 vertical road 亮綠燈、黃燈、左轉燈時亮；vertical road 的綠燈為 30 秒、黃燈為 5 秒、左轉燈為 10 秒，而 vertical road 的紅燈是當 horizontal road 亮綠燈、黃燈、左轉燈時亮

testbench 中有測試 reset 是否對 traffic light controller 有用，並且有測試是否會在經過每個燈號的 clock cycle 後會進入到下個 state，還有在最後測試是否會從最後一個 state 回到第一個 state。

(3) Simulation results before and after the synthesis (including some waveforms)

before the synthesis:

[illegible]

Fig. 2. TLC synthesis 之前測試的測資全部符合

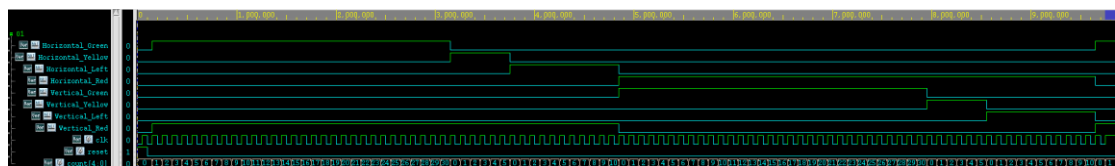


Fig. 3. TLC synthesis 之前 testbench 前段的波形圖

[illegible]

在 synthesis 之後在一開始會有一些 delay，還有在切換的時候會有一些和 synthesis 之前不同的 static 1-hazard 和 static 0-hazard。

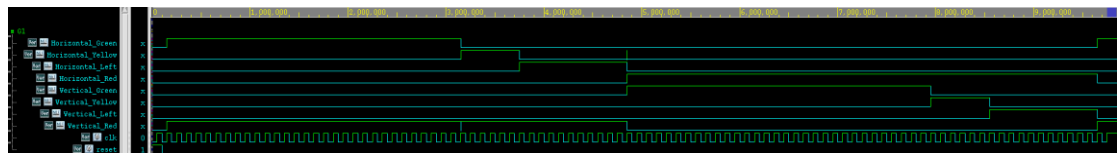


Fig. 5. TLC synthesis 之後 testbench 前段的波形圖

(4) Provide the synthesis results such as power, area, speed, etc.

timing report:

```
*****
Report : timing
        -path full
        -delay max
        -max_paths 1
        -sort_by group
Design : TLC
Version: Q-2019.12
Date   : Mon Nov 13 11:48:23 2023
*****

Operating Conditions: slow   Library: slow
Wire Load Model Mode: top

Startpoint: state_reg[1]
            (rising edge-triggered flip-flop clocked by clk)
Endpoint: count_reg[1]
          (rising edge-triggered flip-flop clocked by clk)
Path Group: clk
Path Type: max
--

Des/Clust/Port      Wire Load Model      Library
-----
TLC                  tsmc18_wl10          slow

Point              Incr              Path
-----
clock clk (rise edge)                2.50              2.50
clock network delay (ideal)           0.00              2.50
state_reg[1]/CK (DFFRHQX1)           0.00              2.50 r
state_reg[1]/Q (DFFRHQX1)            1.76              4.26 r
U53/Y (NAND2X1)                       0.56              4.82 f
U54/Y (NOR2X1)                        0.58              5.39 r
U51/Y (AOI2BB1X1)                     0.39              5.78 r
U50/Y (AOI21X1)                       0.14              5.92 f
U48/Y (OAI211X1)                      0.61              6.54 r
U62/Y (NOR2BX1)                       0.23              6.77 f
count_reg[1]/D (DFFRHQX1)             0.00              6.77 f
data arrival time                      6.77

clock clk (rise edge)                7.50              7.50
clock network delay (ideal)           0.00              7.50
count_reg[1]/CK (DFFRHQX1)            0.00              7.50 r
library setup time                    -0.40              7.10
data required time                     7.10
-----
data required time                     7.10
data arrival time                      -6.77
-----
slack (MET)                           0.33

***** End Of Report *****
```

Fig. 6. timing report 的截圖

## area report:

```

*****
Report : area
Design : TLC
Version: Q-2019.12
Date   : Mon Nov 13 11:46:59 2023
*****

Information: Updating design information... (UID-85)
Library(s) Used:

    slow (File: /home/ncku_class/vsd2023/vsd202300/Desktop/vsd2023/synopsys/slow.db)

Number of ports:          10
Number of nets:           62
Number of cells:          57
Number of combinational cells: 49
Number of sequential cells: 8
Number of macros/black boxes: 0
Number of buf/inv:        11
Number of references:     21

Combinational area:        688.564811
Buf/Inv area:              73.180801
Noncombinational area:    558.835205
Macro/Black Box area:     0.000000
Net Interconnect area:    7046.701996

Total cell area:          1247.400016
Total area:               8294.102012

***** End Of Report *****

```

Fig. 7. area report 的截圖

## power report:

```

*****
Report : power
        -analysis_effort low
Design : TLC
Version: Q-2019.12
Date   : Mon Nov 13 11:47:40 2023
*****

Library(s) Used:

    slow (File: /home/ncku_class/vsd2023/vsd202300/Desktop/vsd2023/synopsys/slow.db)

Operating Conditions: slow   Library: slow
Wire Load Model Mode: top

Design      Wire Load Model      Library
-----
TLC         tsmc18_w110          slow

Global Operating Voltage = 1.62
Power-specific unit information :
  Voltage Units = 1V
  Capacitance Units = 1.000000pf
  Time Units = 1ns
  Dynamic Power Units = 1mW      (derived from V,C,T units)
  Leakage Power Units = 1pW

Cell Internal Power = 112.8081 uW (75%)
Net Switching Power = 37.4826 uW (25%)
-----
Total Dynamic Power = 150.2908 uW (100%)

Cell Leakage Power = 36.3634 nW

Power Group      Internal Power      Switching Power      Leakage Power      Total Power ( % ) Attrs
-----
io_pad           0.0000           0.0000           0.0000           0.0000 ( 0.00%)
memory          0.0000           0.0000           0.0000           0.0000 ( 0.00%)
black_box       0.0000           0.0000           0.0000           0.0000 ( 0.00%)
clock_network   0.0000           0.0000           0.0000           0.0000 ( 0.00%)
register        0.1044           1.7061e-02       1.7766e+04       0.1215 ( 80.81%)
sequential      0.0000           0.0000           0.0000           0.0000 ( 0.00%)
combinational   8.4106e-03       2.0422e-02       1.8598e+04       2.8851e-02 ( 19.19%)
-----
Total           0.1128 mW       3.7483e-02 mW     3.6363e+04 pW     0.1503 mW

***** End Of Report *****

```

Fig. 8. power report 的截圖

## (5) Superlint result

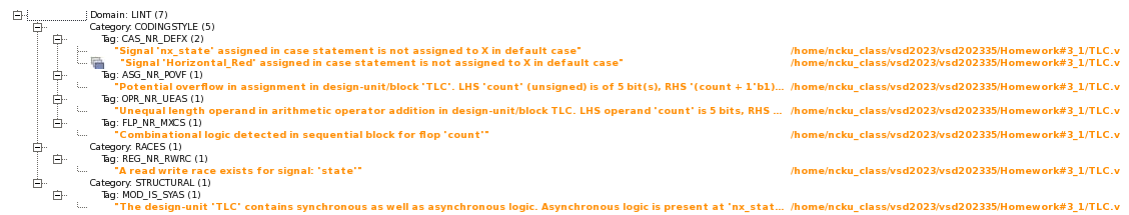


Fig. 9. Superlint result 的截圖

## (6) Discussions

這次的作業中使用了 Synthesis 和 Superlint，但是我的上個作業的程式碼在 Superlint 有許多錯誤和警告，所以我決定重寫一份來讓這次作業中的 Synthesis 和 Superlint 可以順利進行，在重寫的過程中我修改了許多之前在 Superlint 中看到的錯誤和警告，這讓我更熟悉了 Verilog 程式的語法，另外在 debug 的時候也讓我夠熟悉使用波形圖來修正自己程式中的錯誤，這次的作業讓我學習到了許多相關的知識。