Project Date: June 15, 2023

TRAFFIC SIGNAL CONTROLLER

Introduction

Consider a four-way crossing. There will be four traffic signals installed at the front of each lane that turn green one by one. Thus a vehicle has to wait for all the three lanes to pass before the signal turns green.

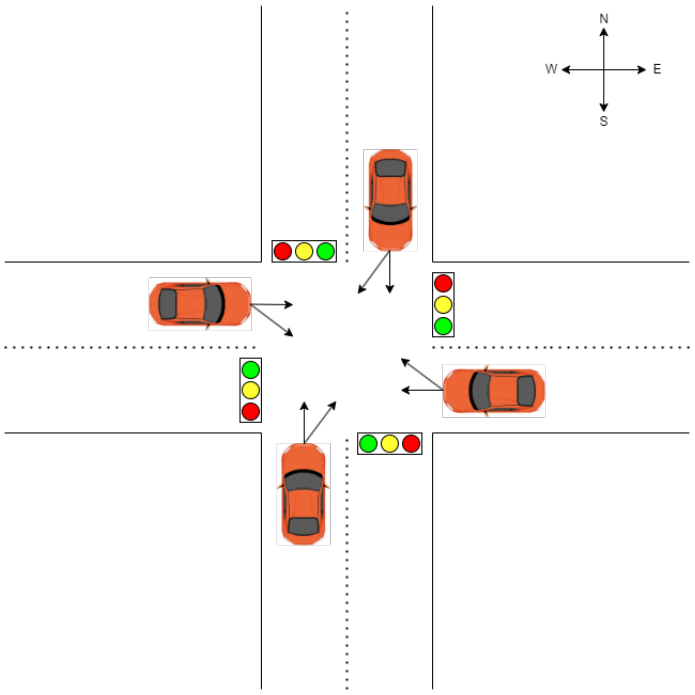

Figure 1: Traffic signal at a 4-way crossing

Design Statement

Design a controller to control the traffic lights of each lane with mentioned functionality.

- There will be 3 traffic lights : RED, YELLOW and GREEN
- The traffic light pattern for each signal is as follows
  $RED \implies YELLOW \implies GREEN \implies YELLOW \implies RED$
- GREEN signal will be turned ON in clockwise direction of the traffic signals
- For an individual signal one of the 3 color must be "ON" all of the time.
- For an individual signal two color can be ON at the same time.
- No two signal can be GREEN at the same time.
- Two adjacent signal will be YELLOW at the same time in clockwise direction.
- For each signal GREEN light will be "ON" for 5 sec and **YELLOW** light will be "ON" for 1 sec.

Problem Statement

1. Create a state diagram for the above controller.
2. Design an FSM using VHDL in behavioral description and structural modelling to produce final output.
3. Use the 50 MHz clock to generate delay of 1 second and 5 seconds.
4. There will be 3 different color for each signal. Instantiate 4:2 encoder that will take your 3 color inputs and produce a corresponding result named Direction_0(LSB) and Direction_1(MSB).

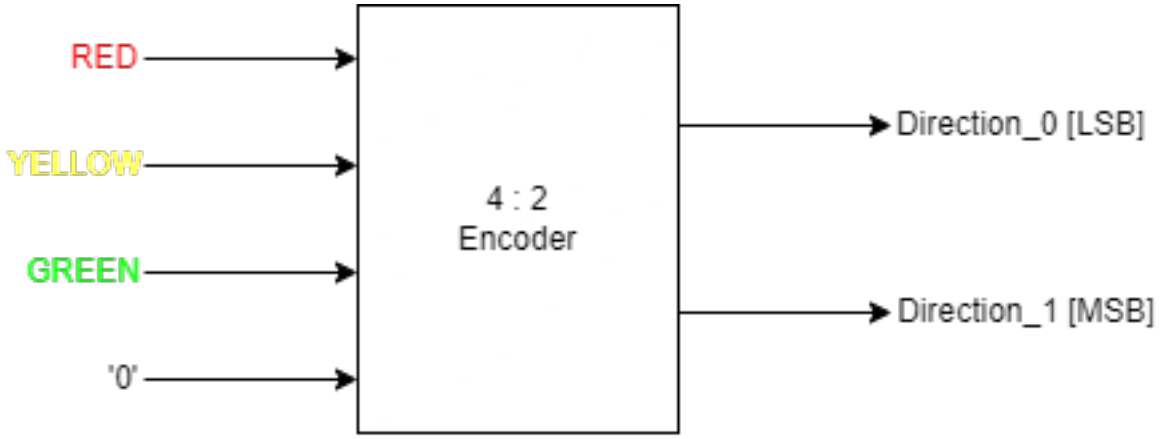

Figure 2: Output Encoder

E.g. For North Signal the corresponding decoder output will be North_0 and North_1.

Signal Value	North_1	North_0
RED = 0; YELLOW = 0; GREEN = 1;	0	1
RED = 0; YELLOW = 1; GREEN = 0;	1	0
RED = 1; YELLOW = 0; GREEN = 0;	1	1

5. Instantiate four 4:2 encoder to depict output for each signal i.e North, South, East and West.
6. Consider the initial state as North traffic signal is giving output "01" and all other traffic signal giving output of "11".
7. Introduce "RESET" functionality using the Push Button on board, so that the system resets to the state as described above.
8. **Pin Mapping:**

$$\begin{array}{ll} North\_1 \implies LED[1] & North\_0 \implies LED[0] \\ East\_1 \implies LED[3] & East\_0 \implies LED[2] \\ South\_1 \implies LED[5] & South\_0 \implies LED[4] \\ West\_1 \implies LED[7] & West\_0 \implies LED[6] \end{array}$$

9. Verify your design using ModelSim and check if the waveforms are appropriate
10. Use the Krypton board to verify the functionality by observing the LEDs.