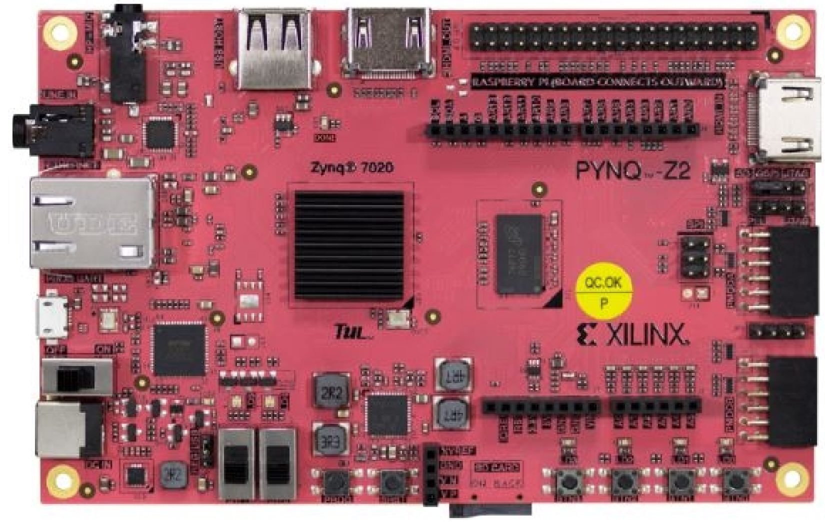
TRAFFIC LIGHT CONTROLLER

Group Members

Siddharth Agarwal
Suyash Bansal
Suvankit Sahoo

Need For Traffic Light Controllers

Traffic congestion is a severe problem in many modern cities around the world. It has been causing many critical problems and challenges in the major and most populated cities.Due to continuous increase in traffic, day-to-day travel is becoming increasingly difficult . Due to these congestion problems, people lose time, miss opportunities, and get frustrated. Traffic congestion directly impacts the companies. Due to traffic congestion, there is a loss in productivity from workers, trade opportunities are lost, delivery gets delayed, and thereby the costs goes on increasing. To solve these congestion problems, we have to build new facilities & infrastructure but at the same time make it smart.



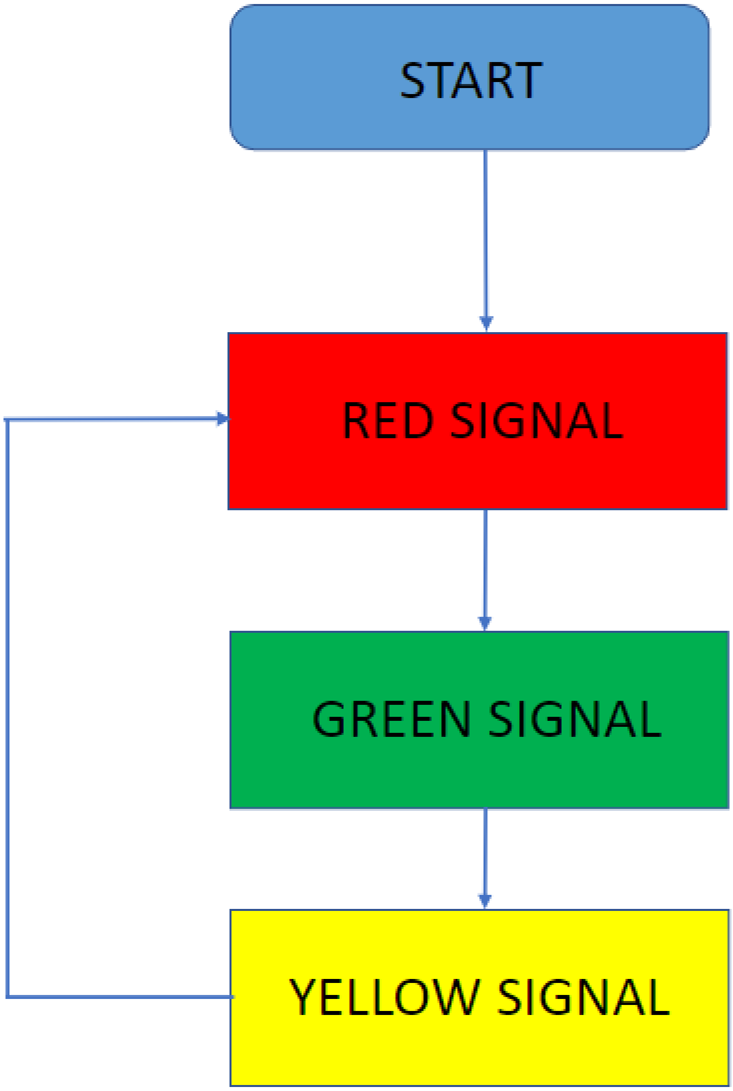
FPGA (Field Programmable Gate Array)

FPGA is the abbreviation of Field Programmable Gate Array. This denotes an integrated circuit which is programmed in the field, i.e. by the system manufacturer. FPGAs can be characterized by the following items:

- High production cost
- Low design density
- Programmable fabric adds significant overhead
- No NRE and Re-Spin cost
- Low development effort
- Low dead-time
- simplified timing
- No test vectors
- Relaxed verification
- Physical design is “hands-off”

FPGA is an Integrated Circuit consisting of an array of uncommitted elements; interconnection between these elements is user-programmable. Using Random Access Memory, high density logic is provided. FPGA is advantageous compared to microcontroller in terms of number of IO (input & output) ports and performance. FPGA, an inexpensive solution compared to ASIC design; is effective with respect to cost in the case of production of large number of units but for fabrication in small number of units it is always costly and time consuming. The Design flow of FPGA shown in Fig. 1 is used to implement the traffic light controller using FPGA. The circuit description can be done using HDLs, followed by the functional simulation and synthesis. The design flow is followed till the timing simulation and then the generated file is downloaded into the target device (FPGA). Verilog is used as HDL for circuit description to code the TLC module. Verilog HDL is used because of the difficulty in writing a VHDL code which has to integrate the source code, ChipScope Pro-Integrated Controller (ICON) and Virtual Input Output (VIO).

TLC Flow Chart

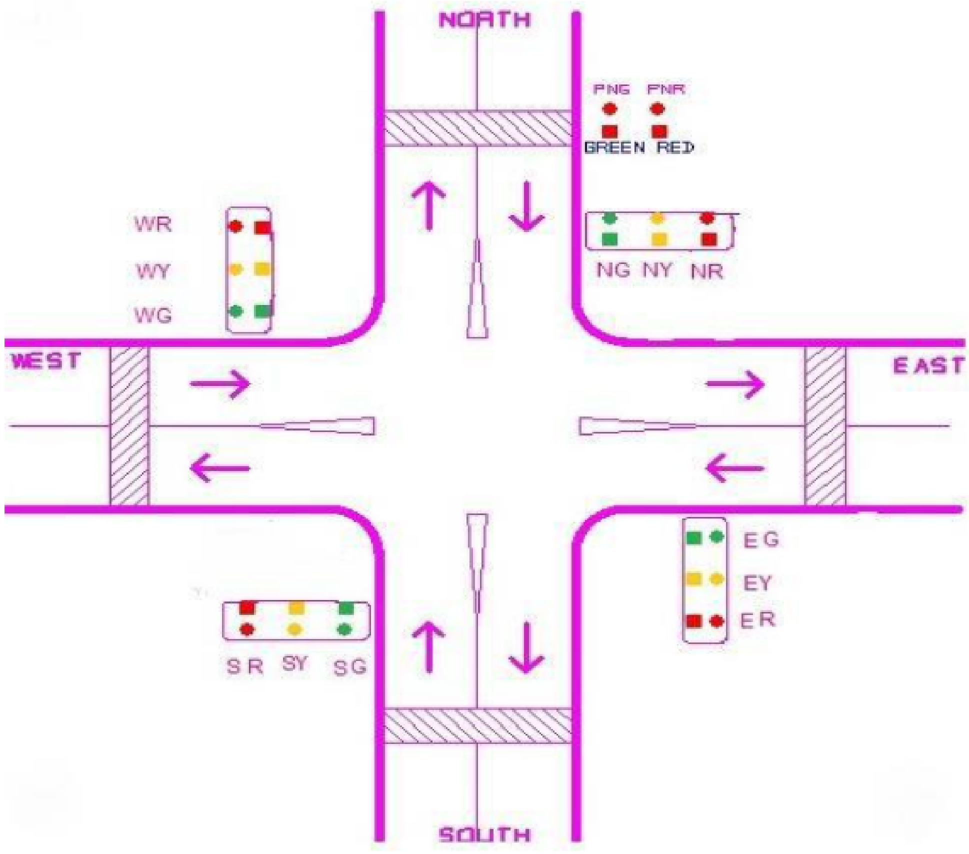


Logic Explanation

Traffic Light Controller can be designed by starting with some arbitrary assumptions. At first the North traffic will be allowed to move and then traffic in the South, East and West direction will be allowed to move in sequence. In general TLC System will be having three lights (red, green and yellow) in each direction where red light stands for traffic to be stopped, green light stands for traffic to be allowed and yellow light stands for traffic is going to be stopped in few seconds.

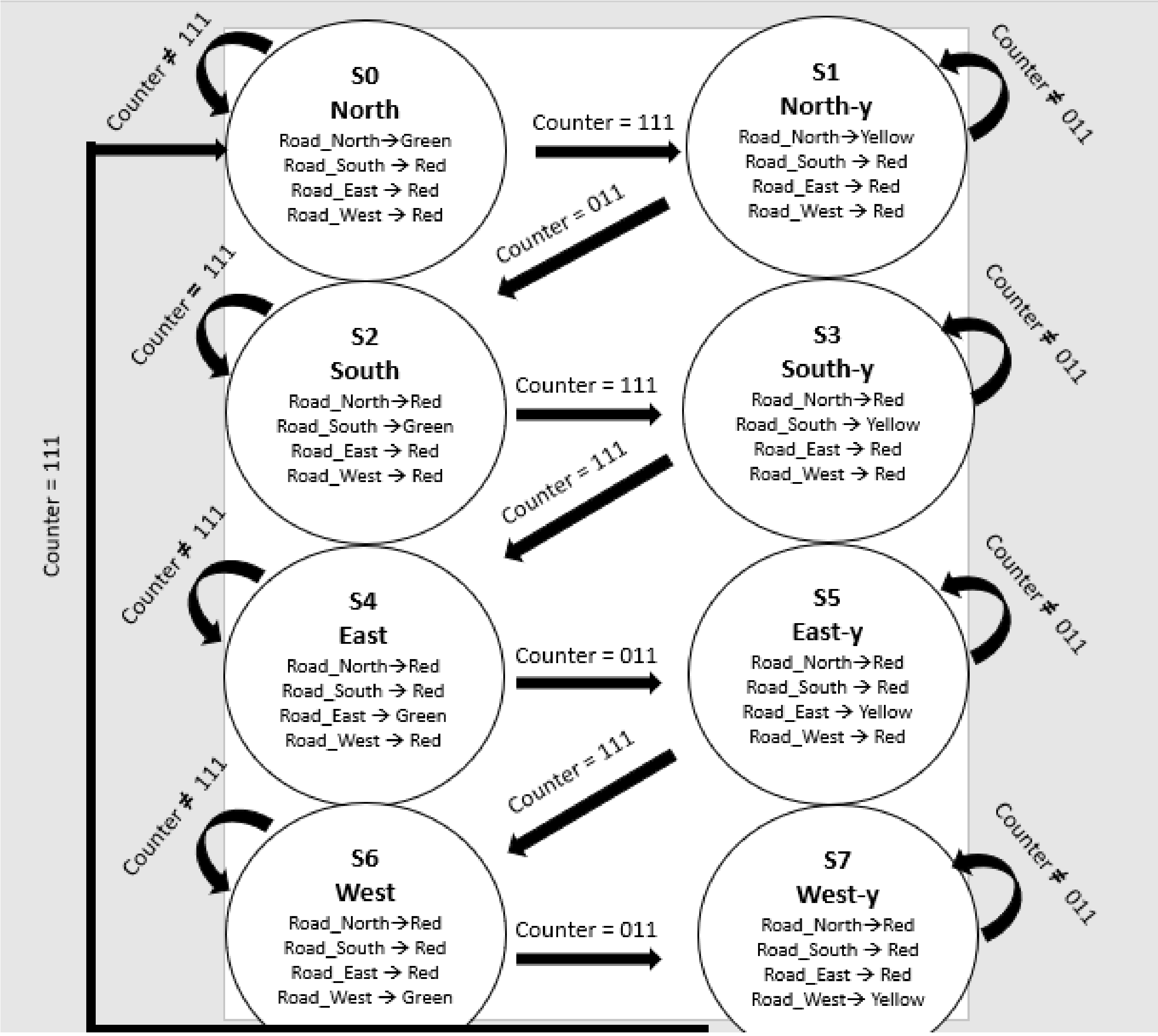
Explanation of Traffic Light Controller

In this structure, there are four traffic signals, represented by North, South, East and West to be controlled. First of all the signal controller is in the reset mode where in the signal of road (North) is green whereas all the other roads South, East and West are red. This state we have assigned as S0.



Later the controller sends the control to state S1 where the North is yellow whereas all the other signals are still red only. In this state the controller waits for some time and then proceed to the next state, state S2 where the South is green and rest all are red. Later the controller sends the control to state S3 where the South is yellow whereas all the other signals are still red only. In this state the controller waits for some time and then proceed to the next state, state S4 where the East is green and rest all are red. Later the controller sends the control to state S5 where the East is yellow whereas all the other signals are still red only. In this state the controller waits for some time and then proceed to the next state, state S6 where the West is green and rest all are red. Later the controller sends the control to state S7 where the West is yellow whereas all the other signals are still red only. From this, the state goes back to the state S0 where North goes green and rest all gets red.

State Machine Diagram


S0
North
Road_North→Green
Road_South → Red
Road_East → Red
Road_West → Red
S1
North-y
Road_North→Yellow
Road_South → Red
Road_East → Red
Road_West → Red
S2
South
Road_North→Red
Road_South →Green
Road_East → Red
Road_West → Red
S3
South-y
Road_North→Red
Road_South → Yellow
Road_East → Red
Road_West → Red
S4
East
Road_North→Red
Road_South → Red
Road_East → Green
Road_West → Red
S5
East-y
Road_North→Red
Road_South → Red
Road_East → Yellow
Road_West → Red
S6
West
Road_North→Red
Road_South → Red
Road_East → Red
Road_West → Green
S7
West-y
Road_North→Red
Road_South → Red
Road_East → Red
Road_West→ Yellow
Counter ≠ 111
Counter = 111
Counter ≠ 011
Counter = 011
Counter = 111
Counter = 111
Counter ≠ 011
Counter = 111
Counter ≠ 111
Counter = 011
Counter ≠ 011
Counter = 111
Counter ≠ 111
Counter = 011
Counter ≠ 011
Counter = 111