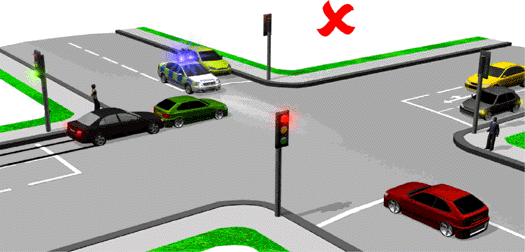
**Traffic Lights Controller**

**Design, Code and Simulation**

By -

**Abhishek Jaisingh ( 14114002 )**

A simple traffic light controller can be implemented by a state machine that has a state diagram such as the one shown in Figure. Its state progresses according the value of the timer used . When the value of timer reaches a specific value, the state of the system changes .

A set of 4 traffic lights have been designed for a 4 – way junction as shown.  
The states are defined in terms of the output . Timer goes from 0 to 100 and then it is reset to 0. Each light is programmed to run as follows:

1. It is GREEN for 20 s.
2. Then it remains YELLOW for 5s.

During this time all other lights remain RED.

This process is continued for all the lights successively.

**Finite State Machine**

**( Moore Machine )**

**State Diagram**

T = 20

T = 100 =>

T = 0

T = 50

T = 70

T = 95

T = 75

T = 45

T = 25

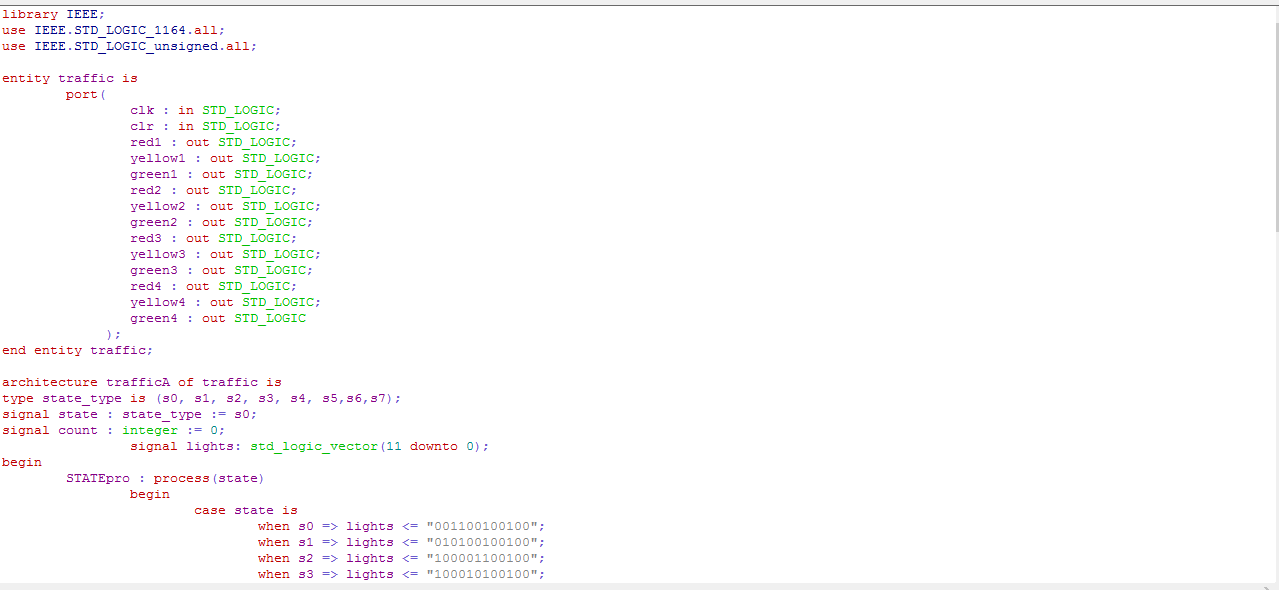
*State Diagram*

**State Table**

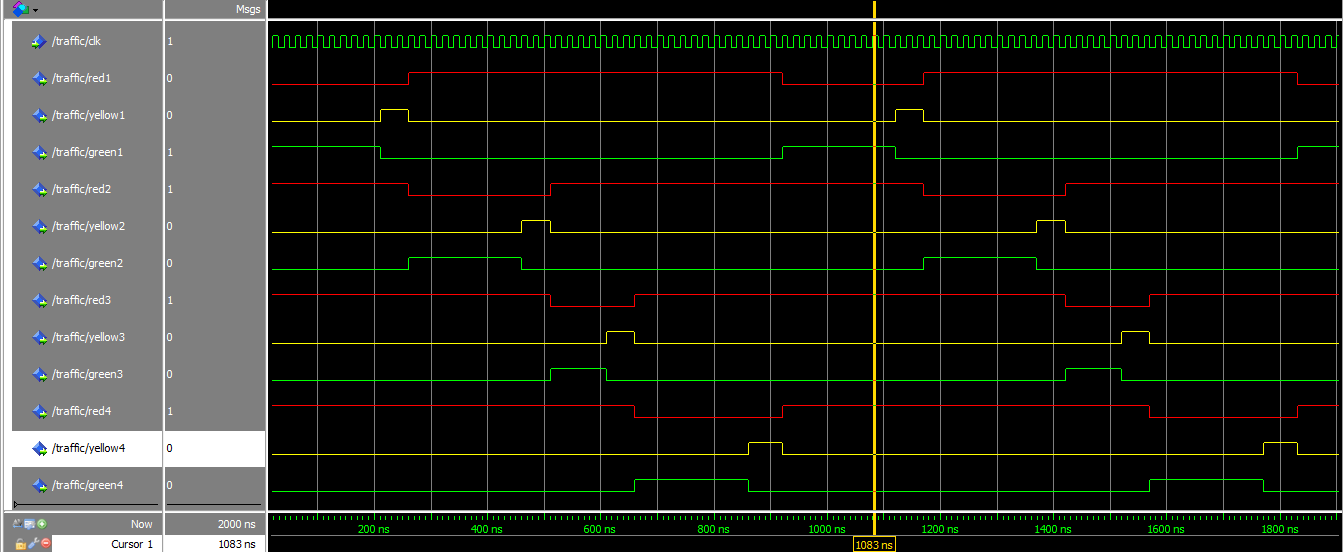
|  |  |  |  |
| --- | --- | --- | --- |
| Time (input) | Current  State | Next  State | Output |
| 0 | **S0** | **S0** | **1234** |
| 20 | **S0** | **S1** | **1234** |
| 25 | **S1** | **S2** | **1234** |
| 45 | **S2** | **S3** | **1234** |
| 50 | **S3** | **S4** | **1234** |
| 70 | **S4** | **S5** | **1234** |
| 75 | **S5** | **S6** | **1234** |
| 95 | **S6** | **S7** | **1234** |
| 100 | **S7** | **S0** | **1234** |

The color in output represents output light.

time = 0 after time becomes 100.

**Code**



**Input / Output Waveforms**