Traffic light controller

1st Darshit Patel
19BEC092
dept. of EC engineering
Nirma University,Ahmedabad
19BEC092@nirmauni.ac.in

2nd Het Patel
19BEC093
dept. of EC engineering
Nirma University,Ahmedabad
19BEC093@nirmauni.ac.in

***Abstract*—The traffic at road crossings and intersections is regulated by turning on and off the Red, Green, and Amber lights in a specific order. The Traffic Light Controller is meant to generate a series of digital data known as switching sequences, which can be used to operate the traffic lights of a conventional four-way intersection in a predetermined order. It is also proposed that day and night operations be implemented. It plays more and more significant function in modern management and control of urban traffic to reduce the accident and traffic jam in road. It is a sequential machine that must be evaluated and programmed in several steps.Analysis of existing sequential machines in traffic light controllers, timing and synchronisation, and the introduction of operation and flashing light synthesis sequences are all part of the device.**

I. Introduction

In many modern cities around the world, traffic congestion is a major issue. Many crucial difficulties and challenges have arisen as a result of traffic congestion in the world's most populous cities. Traveling between different parts of the city is becoming more challenging for those stuck in traffic. People waste time, miss opportunities, and become frustrated as a result of traffic congestion. Congestion has a direct impact on businesses. Workers lose productivity as a result of traffic congestion, trade opportunities are lost, deliveries are delayed, and costs rise as a result.To alleviate traffic congestion, we must construct new facilities and infrastructure while also making them smart. The sole consequence of building new roads on facilities is that it increases congestion in the surrounding area. As a result, rather than building new infrastructure twice, we need to modify the system. As a result, several countries are attempting to improve the mobility, safety, and traffic flow of their existing transportation networks in order to minimise the demand for automobiles.As a result, many studies on traffic light systems have been conducted in order to overcome some complex traffic phenomena, but existing research on the current traffic system in well-traveled traffic scenarios has been restricted. In crossroads, the time of allocation is set from east to west or in the opposite direction, and from north to south. FPGAs are widely utilised in quick prototyping and verification of conceptual designs, as well as in electrical systems when mask-production of a custom IC becomes prohibitively expensive due to low volume. Many of the system designs that were formerly done in custom silicon VLSI are now implemented in Field Programmable Gate Arrays.This is due to the expensive expense of creating a mask for custom VLSI fabrication, particularly in small quantities.

Identify applicable funding agency here. If none, delete this.

II. Flow Chat

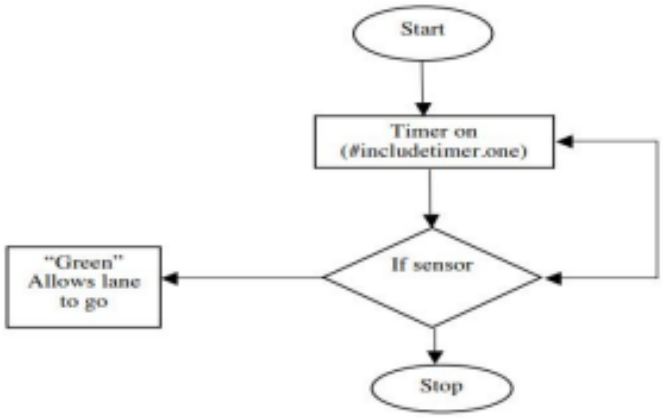

Fig. 1. FLOW CHART

Here above flowchat indicates Traffic light controller. First of all start and synchronize the clock. Here we put sensor to detect a car. So use of the sensor we know that there is traffic present or not. And after help of the sensor state will be change.

III. State Diagram

This indicates the state diagram of our task. Here we take output as a hwy and country road. First of all Initially on the highway all cars are passing and on country road no any cars are passing. That's why initially on the highway green signal and on country road red signal. so initially state is S0. Here we put sensor at the country road and it's detect a car is comming or not. Meaning is that it's simply sense that on country road taffic or not. In our implementation part we define sensor as a variable X. So that we initially on S0 state and X=0 means no any car pass in country road so state will be S0. But if when car is comming to the country road sensor is X=1. So it's detect the car and go to the S1 state. so here in S1 state

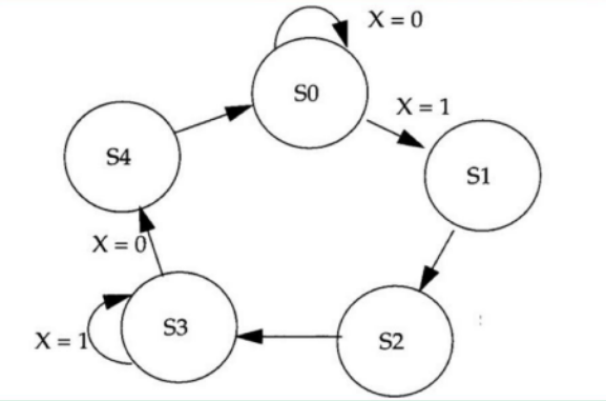

Fig. 2. State diagram

State	Hwy	Country
S0	Green	Red
S1	Yellow	Red
S2	Red	Red
S3	Red	Green
S4	Red	Yellow

Fig. 3. State Table

on highway road signal is yellow and on country road signal is red. And here we give Y2R delay and go to the S2 state. In S2 state both the signals are red. Because of the safety both the signals are red. and here give R2G delay and go the S3 state. In S3 state on hwy signal is red and country road signal is green. Here if X=1 it's indicate traffic on country road and go to the same state. But When no traffic X=0 then go the S0 state and same process will be repeate.

IV. RTL VIEW

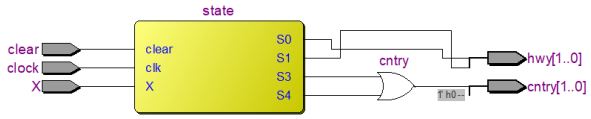

Fig. 4. State Table

V. SIMULATION AND RESULT

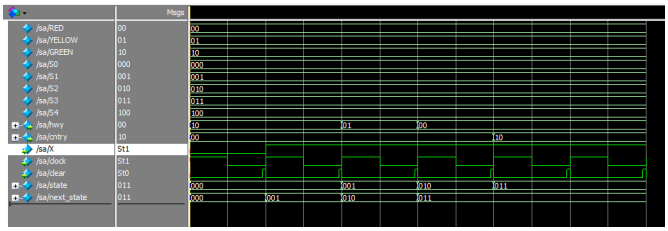

Fig. 5. Waveform

Our this task we implement on Quarus II platform. Implementation done in structuaral style of modelling. As we can see on the graph, we take clock, x, and clear as an input and hwy, country as a output (reg type). Here we also defined state and next state based on the output. When we give an input and we must check the states. In the waveform clearly seen that give input as a clock and make X is 1 or not. Based on this input state will be change and it's clearly shown in the waveform.

VI. CONCLUSION

The contemporary methods of multi-way traffic management vastly enhance traffic conditions. The contribution of advanced signalling controllers to the improvement of urban traffic is proportionate to the controller's complexity. States machines can easily handle these more complicated controllers. Methods for lowering the number of states in a state machine also help to reduce the amount of hardware necessary, resulting in a low-power, small-area design. The project's future aim is that it can be immediately used in real time by utilising a larger number of such circuits.

VII. REFERENCES

- https://github.com/nextseto/VerilogProjects/blob/master/Projec
- https://www.ijmter.com/papers/volume-4/issue-4/traffic-light-controller-using-vhdl.pdf
- Parag K. Lala, "Digital System Design Programmable Logic Devices", Page15-17, Chapter 1, B S Publications

APPENDIX

```
//for sensor X
`define TRUE 1'b1
`define FALSE 1'b0

//Delays
`define Y2RDELAY 3 //Yellow to red delay
`define R2GDELAY 2 //Red to Green Delay

module sa(hwy, cntry, X, clock, clear);
//I/0 ports
output [1:0] hwy,cntry;
//2 bit output for 3 states of signal
//GREEN, YELLOW, RED;
reg [1:0] hwy,cntry;

input X; //sensor
input clock, clear;
//Status of lights
parameter
RED =2'd0,
YELLOW = 2'd1,
GREEN = 2'd2;

//State definition
parameter  S0 = 3'd0,
S1= 3'd1,
S2= 3'd2,
S3= 3'd3,
S4= 3'd4;

//Internal state variables
reg [2:0] state;
reg [2:0] next_state;

always @(posedge clock)
if (clear)
   state <= S0;
else
   state <= next_state;
//Compute values of main signal and country signal
always @(state)
begin
    hwy = GREEN; //Default light assignment
    cntry= RED; //Default light assignment

case (state)
S0:;
S1:hwy= YELLOW;
S2:hwy= RED;
S3:begin
    hwy= RED;
    cntry=GREEN;
end
S4:begin
    hwy= RED;
    cntry=GREEN;
end
endcase
end
```

```
//State machine using case statements
always @(state or X)
begin
    case (state)
    S0: if (X)
            next_state=S1;
    else
            next_state = S0;
    S1: begin //delay some positive edges of clock
            repeat (`Y2RDELAY) next_state=S1;
            next_state =S2;
    end
    S2: begin //delay some positive edges of clock
            repeat (`R2GDELAY) next_state=S2;
            next_state=S3;
    end
    S3:if(X)
            next_state =S3;
    else
        next_state =S4;
    S4: begin //delay some positive edges of clock
            repeat (`Y2RDELAY) next_state=S4;
            next_state=S0;
    end
    default: next_state =S0;
  endcase
 end
endmodule
```