TRAFFIC LIGHT CONTROLLER

Izuchukwu George Enekwa
dept. electronics engineering
Hochschule Hamm-Lippstadt
Germany
izuchukwu-george.enekwa@stud.hshl.de

Patrick Nonki
dept. electronics engineering
Hochschule Hamm-Lippstadt
Germany
Patrick.nonki@stud.hshl.de

Obi Nnamdi Elijah
dept. electronics engineering
Hochschule Hamm-Lippstadt
Germany
Nnamdi-elijah.obi@stud.hshl.de

Abdul Azeez Olanlokun
dept. electronics engineering
Hochschule Hamm-Lippstadt
Germany
Abdul-azeez.olanlokun@stud.hshl.de

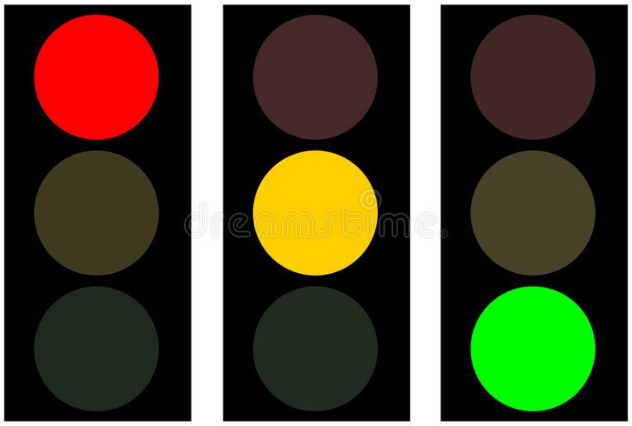
***Abstract*—This paper presents the design and simulation of a traffic light controller at intersections between highway and remote street. In the remote street, there is a sensor that detects whether there is any vehicle on the street. Traffic lights on the high way turn to YELLOW and then RED when vehicles are detected on the street way. This then allows the vehicles from the street way to cross the high way successfully. On the other hand, traffic lights on the highway are always GREEN, while traffic lights on the street are always RED. In both cases, the time period is 3 seconds for the yellow light and 10 seconds for the red light. These include VHDL for modeling and finite state machines, serial communication, uploading and simulating the VHDL design code. We achieved the desired result of our system which is in accordance with our project design and requirements.**

I. Introduction

Signaling devices positioned at pedestrian crossings, intersections, and other locations to control the flow of competing traffic flows known as traffic lights or traffic lamps. In most cities around the world, traffic lights are used to control traffic flow. Road users are assigned right of way by use of lights in standard colors (Red - Yellow - Green), which use a universal color code (and a precise sequence, for color blind drivers). At busy intersections, traffic lights are used to distribute delays more evenly among the various users. The increasing volume of traffic in the cities has a direct impact on congestion and the time it takes to reach certain destinations. But not only the amount of traffic but also how you deal with this traffic has a large impact. It is not sufficient to merely build roads, since they will always reach an end point, such as a junction or bottleneck. Bottlenecks cannot be prevented. However, there is still room for improvement in the way junctions are controlled. Junctions are controlled by traffic lights. Traffic lights though, are most of the time not adaptive. Typical traffic light controllers use a fixed-cycle that doesn't consider how much traffic is coming from any direction; it simply switches configurations at a set interval. This often results in road users waiting at an empty junction with only one road user waiting for a red sign. There has already been an improvement, by placing sensors at the intersections in front of traffic lights so that the controller will only cycle between occupied lanes, so that no one has to wait at a red light at an empty intersection. Theoretical approaches to improving traffic light control include machine learning algorithms. These algorithms provide information samples on ways to predict the future driving behavior of road users, thus enabling traffic light controllers to calculate future waiting times for road users for each action they take. When the controller has the actions and waiting times, the best action would be to choose the one where the expected waiting time is the shortest. We cannot assume that all the information can be gathered or that the information gathered is 100 percent accurate in the real world. Thus decisions have to be made based on estimates, incomplete or erroneous information. This implies working with partial observability. In this paper we have worked with a simple traffic light controller using VHDL, having them control vehicles at intersections between highway and remote street.

.

II. Systems overview



Traffic light systems use the traditional red, yellow, and green colored indicators to designate who has the right of way at an intersection, crossing, or street crossing. Additionally, it coordinates with pedestrian signals to designate right-of-way for pedestrian crossings [1]. A traffic light, also referred to as a traffic signal, stop light, or stop-and-go light, is a signaling device placed at a traffic intersection, pedestrian crossing, or other location to indicate when it is safe to drive, ride, or walk using a universal color code (and a precise sequence, for those who are colors blind)[2]. Currently, a red signal meant that all traffic has to halt. Crosstown traffic had to slow down at a yellow light, and it may go forward or go when the light turned green. The surrounding towns' use of a different method made it even more challenging to comprehend this perplexing color scheme. All participants in the transportation and road traffic system will receive the best outcome thanks to the creation of an intelligent control structure [1]. Road crossings can be managed in a variety of ways. The right-hand rule, a roundabout, or a policeman's signal can help direct traffic in the simplest situations or when it is heavier. However, especially in large cities, it is impossible to avoid using traffic signals in difficult situations where there are multiple lines of traffic at the intersection. One more problem arises when railroad lines and roadways intersect, which happens rather frequently. occurs in traffic situations in suburbs. The most typical method to deal with this kind of The typical cycle lighting control is at an intersection. In more advanced control, sensors monitor the flow of traffic in various directions, and the signals obtained from this operate the traffic lights. The control in this system adjusts to the traffic [3, 4]. The overwhelming quantity of variables and the requirement for extensive computation are the overall issue. Using fuzzy approaches could help to simplify this issue. Numerous simulations have been conducted in recent years, and real control systems based on straightforward fuzzy rules have also been developed [5,6,7,8,9,10,11,12,13,14]. However, it does make sense to employ fuzzy approaches comprising hierarchy and use interpolation to lessen the complexity in the most complex scenarios when there are numerous lanes, maybe more than one road intersection, and a railroad participate [13, 14].

A. *Problem solve*

Traffic control signals are crucial tools for managing traffic on the road when they are used correctly. They decide which traffic movements have the right-of-way, greatly affecting how the traffic moves. One or more of the following benefits will result from traffic control signals that are correctly created, placed, operated, and maintained:

- Provide orderly movement of traffic
- Minimize complicating movement
- Coordinated for continuous movement
- Provide driver confidence by assigning right way

Many people believe that traffic control signals are the solution to all inter-sectional traffic issues. This misconception has resulted in the installation of traffic control signals in numerous places where they are not required, thus compromising the efficiency and safety of traffic for cars, bicycles, and pedestrians. Even when warranted by traffic and road circumstances, traffic control signals may be badly constructed, ineffectively located, incorrectly operated, or poorly maintained. While properly built and justified traffic signals might be beneficial, they also have drawbacks. Even if the signal is justified, there will always be some drawbacks.

III. Systems Modelling

Using VHDL modeling, we created a controller for traffic signals at a four-way intersection using a Finite state machine. There are two lanes of traffic at this intersection going east, west, north, and south. Additionally, there is a designated left and right turn lane for each way. Every roadway, including NS (North-South) and EW (East-West), includes a sensor to show whether a car is at the intersection or if one is approaching.

A. *System Objective*

A four-phase signal intersection must be managed by the traffic light operator. If we just take the straight-ahead direction, every roadway includes NS (North-South) and EW (East-West). while NS and SN Street is always on red. The green signal allows all vehicle on the EW and WE highway traffic drive. The NS and SW are only activated when a vehicle is sensed. Following the completion of NS street, East to West or West to East traffic now follows EW street. We can see that during NS street, vehicles can go to left or right direction according to North-South direction parallel to main direction because EW street is OFF during this. If we were to consider another possibility to follow traffic in another direction rather than straightaway direction. The same is true of EW Street.

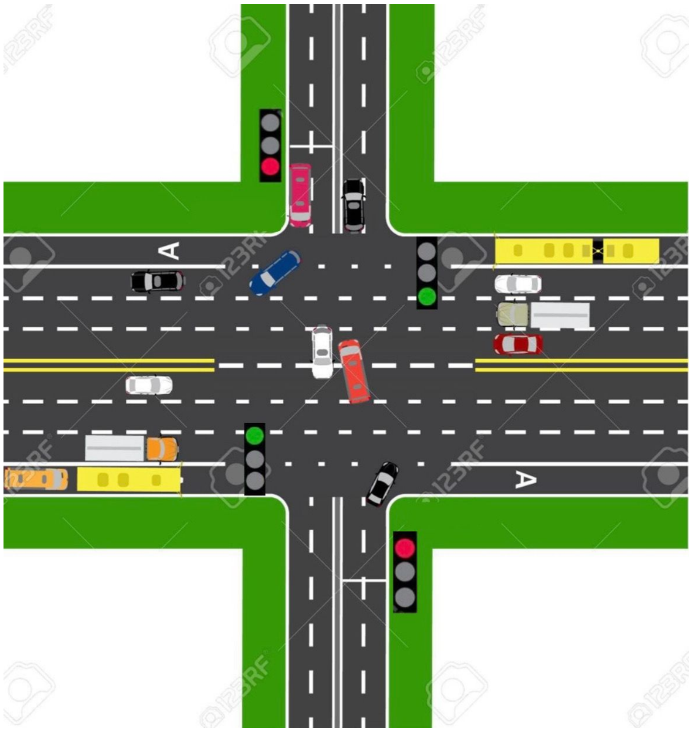


Fig. 1. Scenario of highway and remote street

To understand the scenario perfectly, the image fig.1 illustrates the system. After movement on the North and South happens, the system returns the highway signals to green. Basically, we assigned priority to the highway traffic, the remote street comes on only if a vehicle is detected or sensed. Of course delays (time) was set to each traffic signal.

IV. Implementation

Shows a traffic light controller where a rural road and a highway is controlled by a traffic light controller in VHDL. To determine whether there are any vehicles on the Remote street, there is a sensor on the remote street. Traffic lights on the high way turn from YELLOW to RED if vehicles are spotted on the farm way, allowing the vehicles to cross the high way. Otherwise, the high-way traffic light is always GREEN and remote street traffic signal is always RED. A finite state machine can be used to create a straightforward traffic light controller. According to the value of the timer being used, its state advances. The status of the system changes when the timer value reaches a particular value. The output is used to define the states. The timer runs from 0 to 120 before being reset to 0. To ensure that each traffic light is activated (the GREEN bit is set) at least once, this process is repeated for each light in turn. Every traffic signal is set up to function as follows:

- Initially it is GREEN.
- After that, it stays YELLOW for 3s when a vehicle is sensed from the remote street.
- For the RED light, it is 10s
- After cars from NORTH and SOUTH successfully cross the intersection, the lights are kept RED during again.

A. Finite State Machine

Finite state machine (FSM) is also referred as a synchronous sequential circuit which contains a finite number of states.

There are two types of FSMs:

- Moore State Machine: The only factor affecting output is the
  current conditions
- Mealy State Machine: The results are based on both the both the current conditions and inputs.

In our project, we used an FSM of the Moore type.

In our project, there are 6 states altogether, and the diagram of the state

is below.

B. Terms and Notations

For the following FSM diagram in fig.2 We have the notation ”T” as time in seconds, the ”S” as states, ”start” as Green, ”next” as yellow and ”stop” as red. Also RSV refers to ”Remote Street Vehicle”.

S0 is the initial state at time T=0. At time T=20, a state transition occurs, and S1 is the new state. This cycle continues until time T=120, or when all of the traffic lights have turned green at least once. The cycle then continues when the counter and state have both been reset to 0, respectively.

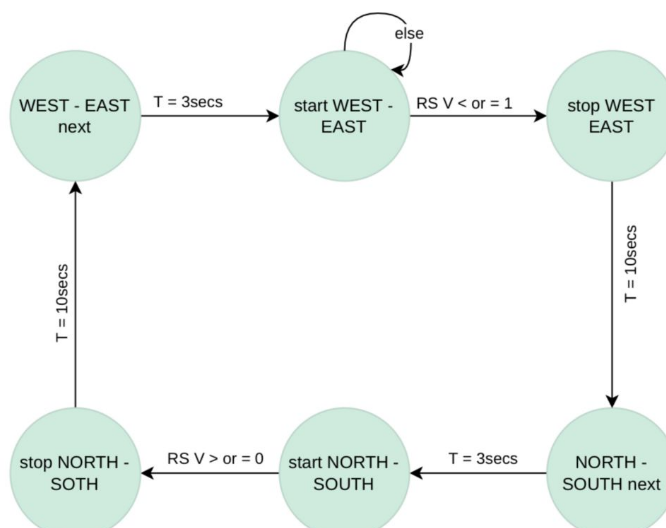


Fig. 2. Finite State Machine.

State 0 is the initial state at time T=0. At S1, the system transitions to stop West and East traffic signals if RSV (remote street vehicle) is detected [RSV ¿ or = 1] at T= 10secs. Meanwhile if no RSV is detected, the WEST and EAST signals remain Green. At North and South the yellow light is T = 3secs. We move to the next state where [RSV ¡ or = 0] changes at time T=3secs, which means the system returns priority to the initial state. The cycle then continues when the counter and state have both been reset to 0, respectively. The following VHDL code mapped from the FSM is shown here.

Traffic_Light_C_TB_Musketeers - Notepad

File Edit View

```
-- Team Musketeers for Hardware Engineering Lab
-- VHDL project Work: VHDL code for traffic light controller
-- Submitted to Engineer Ali Hayek
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

-- Testbench to verify VHDL code for traffic light controller

ENTITY tb_traffic_light_controller IS
END tb_traffic_light_controller;

ARCHITECTURE behavior OF tb_traffic_light_controller IS
    -- we declare the component for the traffic light controller
    COMPONENT traffic_light_controller
    PORT(
         sensor : IN  std_logic;
         clk : IN  std_logic;
         rstn : IN  std_logic;
         Highway_light : OUT  std_logic_vector(2 downto 0);
         RemoteStreet_light : OUT  std_logic_vector(2 downto 0)
        );
    END COMPONENT;
   signal sensor : std_logic := '0';
   signal clk : std_logic := '0';
   signal rstn : std_logic := '0';
  --Outputs
   signal Highway_light : std_logic_vector(2 downto 0);
```

Ln 17, Col 30

Traffic_Light_C_TB_Musketeers - Notepad

File Edit View

```
   signal Highway_light : std_logic_vector(2 downto 0);
   signal RemoteStreet_light : std_logic_vector(2 downto 0);
   constant clk_period : time := 10 ns;
BEGIN
 -- We represent an instance to the traffic light controller
   trafficlightcontroller : traffic_light_controller PORT MAP (
          sensor => sensor,
          clk => clk,
          rstn => rstn,
          Highway_light => Highway_light,
          RemoteStreet_light => RemoteStreet_light
        );
   -- We define the Clock process
   clk_process :process
   begin
  clk <= '0';
  wait for clk_period/2;
  clk <= '1';
  wait for clk_period/2;
   end process;
   stim_proc: process
   begin
  rstn <= '0';
  sensor <= '0';
      wait for clk_period*10;
  rstn <= '1';
  wait for clk_period*20;
  sensor <= '1';
```

Ln 53, Col 26

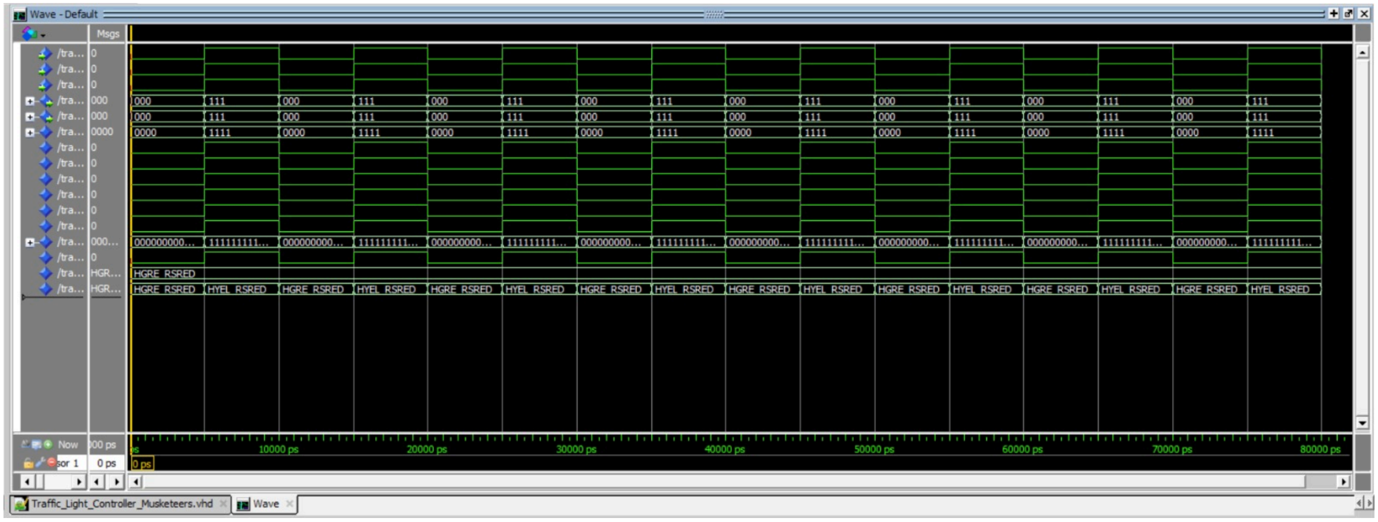
Traffic_Light_C_TB_Musketeers - Notepad

File Edit View

```
          sensor => sensor,
          clk => clk,
          rstn => rstn,
          Highway_light => Highway_light,
          RemoteStreet_light => RemoteStreet_light
        );
   -- We define the Clock process
   clk_process :process
   begin
  clk <= '0';
  wait for clk_period/2;
  clk <= '1';
  wait for clk_period/2;
   end process;
   stim_proc: process
   begin
  rstn <= '0';
  sensor <= '0';
      wait for clk_period*10;
  rstn <= '1';
  wait for clk_period*20;
  sensor <= '1';
  wait for clk_period*100;
  sensor <= '0';
      wait;
   end process;

END;
```

V. SIMULATION



VI. CONCLUSION

For roadways to run smoothly and effectively, a traffic light controller is absolutely necessary. We have modelled and created a Moore Type FSM and Static Priority Arbiter Circuit-based Traffic Light Controller system. Depending on whether or not there is traffic on the roadways, the planned circuit can prioritize which traffic light has to be triggered (GREEN).

REFERENCES

[1] han Huang, Adel W. Sadek and Yunjjie Zhao, Assessing the Mobility and Environmental Benefits of Reservation-Based Intelligent Intersections Using an Integrated Simulator, IEEE, Transactions on Intelligent Transportation Systems, accepted in January, 20, 2012.

[2] . Wen, ”A dynamic and automatic traffic light control system for solving the road congestion problem”, Expert Systems with Applications, Vol. 34, Issue 4, May 2008, pp. 2370-2381.

[3] ousaf Saeed, M. Saleem Khan, Khalil Ahmed, Abdul Salam Mubashar, A Multi-Agent Based Autonomous Traffic Lights Control System Using Fuzzy Control, International Journal of Scientific Engineering Research Volume 2, Issue 6, June2011 1, ISSN 2229-5518.

[4] o Chen and Harry H. Cheng, A Review of the Applications of Agent Technology in Traffic and Transportation Systems, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, VOL. 11, NO. 2, JUNE 2010.