Alqalam Journal of Medical and Applied Sciences. 2026;9(5):1087-1194
https://doi.org/10.54361/ajmas.269522

Original article

Design of Traffic Light Controller using Verilog HDL

Ghaled Bel Kasem[1], Omar Murajia[2]

[1]*Department of Computer Science, The University of Tobruk, Libya*
[2]*Department of Information Technology, Higher Institute of Science and Technology Emsaad, Libya.*
Corresponding Email. *khaled.belkasem@tu.edu.ly*

Abstract
The traffic at road crossings and intersections is regulated by turning on and off the red, green, and yellow lights in a certain order. The Traffic Light Controller is meant to create a series of digital data known as switching sequences, which may be used to operate the traffic lights of a conventional four-way intersection in a predetermined order. It is also recommended that day and night operations be implemented. It is becoming more crucial in modern urban traffic management and control in order to decrease road accidents and traffic jams. It is a sequential machine that must be examined and programmed in several steps. Analysis of current sequential machines in traffic light controllers, timing and synchronization, and the introduction of operation and flashing light synthesis sequences are all part of the device. Design the circuit, write code, simulate, synthesize, and implement in hardware are the approaches utilized in this project. Quartus Prime 18.1 Software was selected for this project to create a schematic using schematic edit and write code using the Verilog HDL (Hardware Description Language) text editor.
Keywords. Traffic Light Controller, Verilog, HDL.

Introduction

Traffic control is a challenging problem in many cities. This is due to the large number of vehicles and the high dynamics of the traffic system. Poor traffic systems are a big reason for accidents and time losses. In this, it will reduce the waiting time of the vehicles at traffic signals. The Traffic Light Control (TLC) system is also based on a microcontroller and a microprocessor [1]. But the disadvantage of with microcontroller or microprocessor is that it works on a fixed time, which functions according to the program, which does not have the flexibility of modification on a real-time basis [2]. In a traffic light controller, the density of traffic is sensed by using IR sensors throughout the day and night, and accordingly, time is allotted for users to pass. Other advantages of this system are: i) The system senses emergency vehicles on the individual road; moreover, it gives priority to the traffic of that road where the emergency vehicles are sensed. ii) Find out the defaulter who crosses the red signal by capturing images using a camera. In this, we are using an FPGA with traffic sensors to control traffic according requirement means we can change the program if it requires, and thus reduces the waiting time.

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 is delayed, and thereby the costs go on increasing. To solve these congestion problems, we have to build new facilities & infrastructure, but at the same time make it smart. The only disadvantage of making new roads on facilities is that it makes the surroundings more congested. So, for that reason, we need to change the system rather than making new infrastructure twice. Therefore, many countries are working to manage their existing transportation systems to improve mobility, safety, and traffic flows in order to reduce the demand for vehicle use. Therefore, many studies about traffic light systems have been done in order to overcome some complicated traffic phenomena, but existing research has been limited to the present traffic system in well-travelled traffic scenarios. The time of allocation is fixed from east to west or the opposite way, and from north to south way in crossroads. Field Programmable Gate Arrays (FPGAs) are extensively used in rapid prototyping and verification of a conceptual design, and are also used in electronic systems when the mask production of a custom IC becomes prohibitively expensive due to the small quantity. Many system designs that used to be built in custom silicon VLSI are now implemented in Field Programmable Gate Arrays. This is because of the high cost of building a mask for the production of a custom VLSI, especially for a small quantity.

Research Problem

The research problem lies in the fact that traditional traffic signal systems, especially those based on microcontrollers or microprocessors, operate on fixed timings that do not adapt to actual changes in traffic density.

This leads to several issues, including:

- Increased traffic congestion
- Longer waiting times at intersections
- Poor response to emergency situations (such as ambulances)
- Reduced overall traffic management efficiency

Moreover, these systems lack flexibility for real-time adaptability, making them unsuitable for modern urban

Received*: 11-03-2026 -* ***Accepted****: 09-05-2026 -* ***Published****: 16-05-2026*

environments with fluctuating traffic flow.

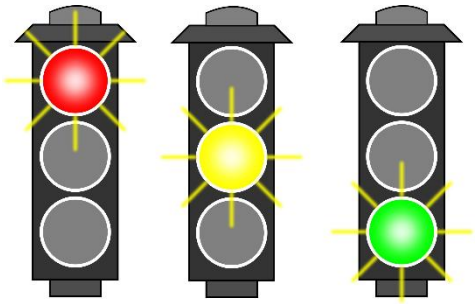

Figure 1. Traffic Light Controller.

Research Objective

This research aims to:

Design and implement an intelligent traffic signal control system using FPGA technology and Verilog HDL.

Develop a system capable of:

- Adapting to traffic density using sensors
- Reducing waiting time at intersections
- Improving traffic flow
- Giving priority to emergency vehicles
- Simulating the system and verifying its performance using design tools (such as Quartus Prime).

In general, the goal is to improve traffic management efficiency compared to traditional systems through an adjustable and higher-performing system.

The system design was chosen using an FPGA and Verilog HDL for the following reasons:

High Flexibility

The system design can be easily modified without changing the hardware.

Supports future updates and development.

High Performance

FPGA provides higher processing speed compared to microcontrollers.

Suitable for systems that require immediate response.

Parallel Processing

Multiple operations can be executed at the same time, which is important in traffic control systems.

Research Method

Field Programmable Gate Array (FPGA)

An FPGA is an Integrated Circuit consisting of an array of uncommitted elements; the interconnection between these elements is user-programmable. Using Random Access Memory, high-density logic is provided. FPGA is advantageous compared to a microcontroller in terms of the number of IO (input & output) ports and performance [3]. FPGA, an inexpensive solution compared to ASIC design, is effective with respect to cost in the case of production of many units, but for fabrication in a small number of units, it is always costly and time-consuming. The Design flow of an FPGA, shown in (Figure 1) is used to implement the traffic light controller using an FPGA. The circuit description can be done using HDLs, followed by the functional simulation and synthesis [4]. 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 an HDL for circuit description to code the TLC module.

FPGA Design Flow

A simplified version of the design flow is given in the following diagram.

Received*: 11-03-2026 -* ***Accepted****: 09-05-2026 -* ***Published****: 16-05-2026*

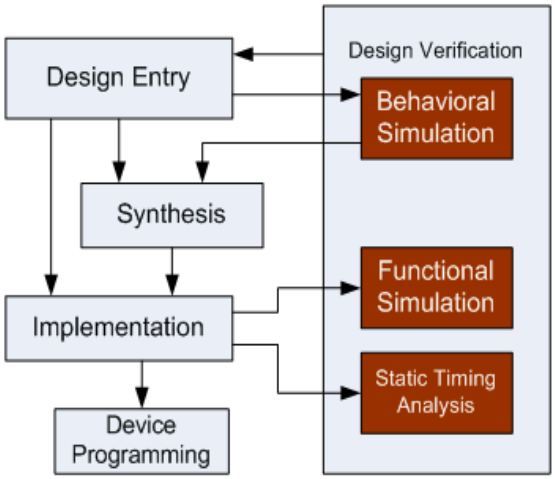

Figure 2. FPGA Design Flow.

Design Entry

There are different techniques for design entry. Schematic-based, Hardware Description Language, and a combination of both, etc. [5]. The selection of a method depends on the design and designer. If the designer wants to deal more with Hardware, then Schematic entry is the better choice. When the design is complex or the designer thinks of the design in an algorithmic way, then HDL is the better choice. Language-based entry is faster but lags in performance and density.

Synthesis

The process that translates VHDL or Verilog code into a device netlist format. i.e., a complete circuit with logical elements (gates, flip flops, etc.) for the design. If the design contains more than one sub-design, ex. to implement a processor, we need a CPU as one design element and RAM as another, and so on, then the synthesis process generates a netlist for each design element [5]. The synthesis process will check code syntax and analyse the hierarchy of the design, which ensures that the design is optimized for the design architecture the designer has selected.

Implementation

This process consists of a sequence of three steps

1. Translate
2. Map
3. Place and Route.

Device Programming

Now the design must be loaded on the FPGA. But the design must be converted to a format so that the FPGA can accept it. This can be done using a cable. The selection of cables depends on the design.

Design Verification

Verification can be done at different stages of the process steps.

Behavioral Simulation (RTL Simulation)

This is, first of all, simulation steps; those are encountered throughout the hierarchy of the design flow. This simulation is performed before the synthesis process to verify RTL (behavioral) code and to confirm that the design is functioning as intended [6]. Behavioral simulation can be performed on either VHDL or Verilog designs. In this process, signals and variables are observed, procedures and functions are traced, and breakpoints are set. This is a very fast simulation, which allows the designer to change the HDL code if the required functionality is not met within a short time period. Since the design is not yet synthesized to the gate level, timing and resource usage properties are still unknown.

Functional Simulation

Post Translate Simulation, functional simulation gives information about the logic operation of the circuit. The designer can verify the functionality of the design using this process after the Translate process [7]. If the functionality is not as expected, then the designer has to make changes in the code and again follow the design flow steps.

Received*: 11-03-2026 -* ***Accepted****: 09-05-2026 -* ***Published****: 16-05-2026*

Static Timing Analysis

This can be done after the MAP processes the Post MAP timing report lists signal path delays of the design derived from the design logic [8]. Post Place and Route timing report incorporates timing delay information to provide a comprehensive timing summary of the design.

Design of Traffic Light Controller

A 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 East, South, and West directions will be allowed to move in sequence. The advantage of writing Traffic Light Controller program is that in a program, modifications as per requirements can be done easily i.e., suppose the traffic on main road should be allowed for more time and for side roads the traffic should be allowed for less time; then the clock is divided in such a way that for main road the clock period will be more and for side roads the clock period will be less, this is because the main road traffic is heavy when compared to the side road traffic. In general, the TLC System will have 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 to be stopped in a few seconds.

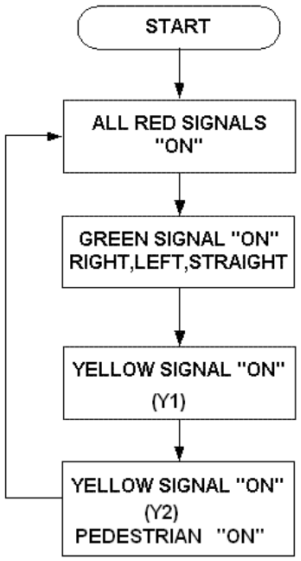

Figure 3. TLC Flow Chart.

Explanation of the Traffic Light Controller

In this structure, there are four traffic signals, represented by R1, R2, R3, and R4, to be controlled. All four signals have the same priority as they are all main roads.

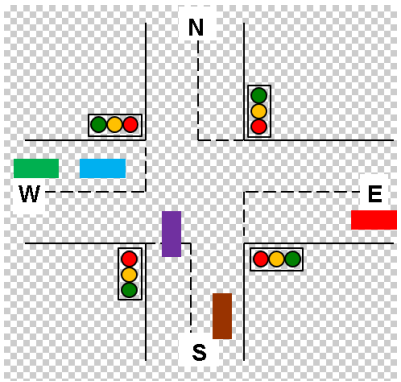

Figure 4. Traffic Signals at Junction.

First of all, the signal controller is in the reset mode, where the signal of road (R1) is green, whereas all the other roads, R2, R3, and R4, are red. This state we have assigned as S0.

Later, the controller sends the control to state S1, where the R1 is yellow, whereas all the other signals are still red only. In this state, the controller checks whether the sensor at road R2, which is X2, is low or not. If the sensor gives a low signal that there is no traffic on that road, then that signal on road R2 is skipped, transferring control to the state S4, where the signal on road R3 is turned on, whereas the rest of the signals are showing red. On the one hand, if the traffic is present on the road R2, then the control is sent to state

Received*: 11-03-2026 -* ***Accepted****: 09-05-2026 -* ***Published****: 16-05-2026*

S2 switches on the signal on road R2 to green, and the rest of the signals are red only when the control is with state S2. After showing the green signal, the signal light changes from green to yellow for the signal on the road R2 while all the other signals continue to be in red light mode only, which is the operation of state S3.

Again, when the controller is in state S3, it checks for the response of sensor X3 on road R3. If the output of the sensor is low, the control of the system will be transferred to state S6, skipping the working of the signal on road R3; otherwise, the control is given to the corresponding next state S4.

When in S4, the traffic signal of road R3 turns green; on the other hand, the signals of roads R1, R2, and R4 remain red. The control is then transferred to state S5.

When the control is in state S5, it checks for the output of the sensor X4 on the road R4. Depending on the output of X4, the further state change takes place accordingly. If low, then the control is transferred to state S0, skipping the operation of the signal on road R4. Otherwise, the control is in state S6. When the controller is in state S5, there is a change of signal on road R3 from green to yellow.

When the control is in state S6, the signal of road R4 turns green, whereas all the other signals turn green or remain in a red signal only. The control is then shifted to state S0.

In state S7, the signal of road R4 turns from green to yellow. Simultaneously, the sensor on the first road, R1, which is X1, is checked for its output. If the signal is low, then the control is shifted directly to state S2; otherwise, the control is shifted to default state S0.

These states are not mandatory. The number of states, the order of the lights, and the delay can be specified by the user. This is one of the advantages of this project.

TLC State Diagram

The TLC state diagram shown in (Figure 2.3) illustrates that whenever cnt=00 and dir=00, then the green light in the north direction will be ON for a few seconds, and the red signal light in all other directions, namely west, south, and east, will be ON. When cnt=01 and dir=00, then the yellow light (y1) will be ON for a few seconds, and when cnt=01, the yellow light (y2) and the pedestrian north will be ON, and then dir is incremented by one and cnt is assigned to zero. So, when cnt=00 and dir=01, the green light in the east direction will be ON for a few seconds, and all red lights in other directions will be ON. Whenever cnt=01 and dir=01, then the yellow light (y1) will be ON for a few seconds, and when cnt=01, the yellow light (y2) and pedestrian east will be ON, and then dir is incremented by one and cnt is assigned to zero.

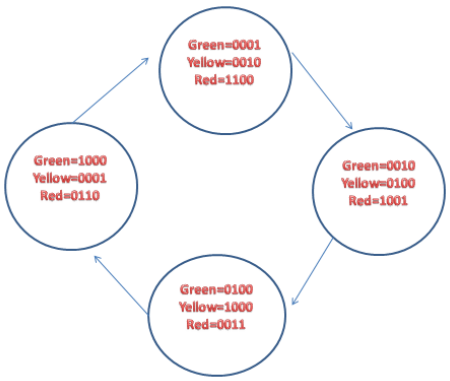

Figure 5. TLC State Diagram.

So, whenever cnt=00 and dir=10, the green light in the south direction will be ON for a few seconds, and all red lights in other directions will be ON. Whenever cnt=01 and dir=10, then the yellow light (y1) will be ON for a few seconds, and when cnt=01, the yellow light (y2) and the pedestrian south will be ON, and then dir is incremented by one and cnt is assigned to zero. So, whenever cnt=00 and dir=11, the green light in the west direction will be ON for a few seconds, and all red lights in other directions will be ON. Whenever cnt=01 and dir=11, then the yellow light (y1) will be ON for a few seconds, and when cnt=01, the yellow light (y2) and pedestrian west will be ON, and then dir. is assigned to 00, and cnt is assigned to zero. This sequence repeats, and the traffic flow will be controlled by assigning time periods in all four directions.

(Figure 6) depicts a general four-road structure which consists of north, east, west, and south directions, each with a set of three lights, namely green, yellow, and red. The green light in a direction will be ON when the left, straight, and right sides are set to be free for traffic in that direction.

Received*: 11-03-2026 -* ***Accepted****: 09-05-2026 -* ***Published****: 16-05-2026*

https://doi.org/10.54361/ajmas.269522

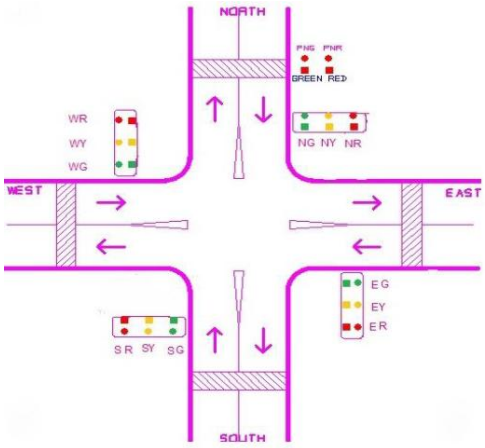

Figure 6. Structure of Road.

The figure shows the design of the traffic light model. To distinguish each lane and the traffic signal lights, they are labelled separately with North, East, South, and West. Signal lights at each lane have their own set of traffic light signals: “Red, Yellow, and Green”. The operation of this signal light is similar to a common traffic light signal. Along with these specifications, each lane has a light to represent a sensor of the corresponding road. A linear sensor or electromagnetic sensor is suitable for the design of a real traffic light system. The first sensor detects the presence of vehicles, and the second sensor determines the volume of the traffic corresponding to that lane. Through the two sensors, we will know the expected time for the green signal to turn on and when the signal light at each lane should be changed to green.

Results and Discussion

RTL Schematic

The figure below shows the RTL Schematic of the Traffic Light Controller.

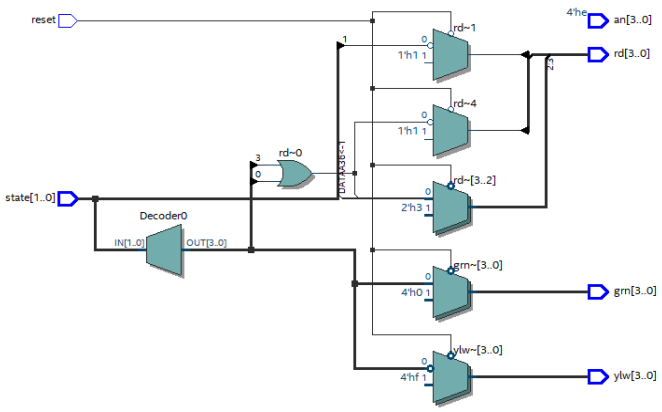

Figure 7. RTL Schematic.

TLC State Machine

The figure below shows the TLC State Machine of the Traffic Light Controller.

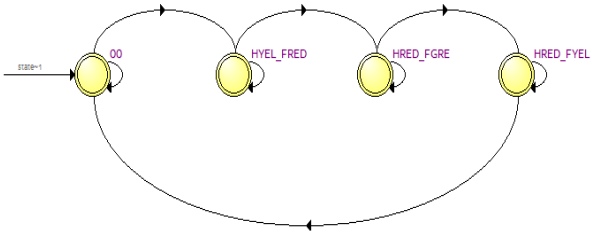

Figure 8. TLC State Machine.

Technology Schematic

The below figure shows the Technology Schematic of the Traffic Light Controller.

Received*: 11-03-2026 -* ***Accepted****: 09-05-2026 -* ***Published****: 16-05-2026*

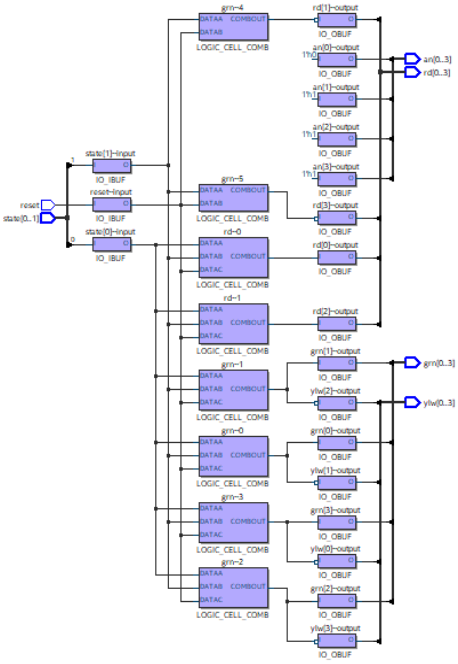

Figure 9. Technology Schematic.

Waveform

The figure below shows the Waveform of the Traffic Light Controller when the test bench is applied to the source code.

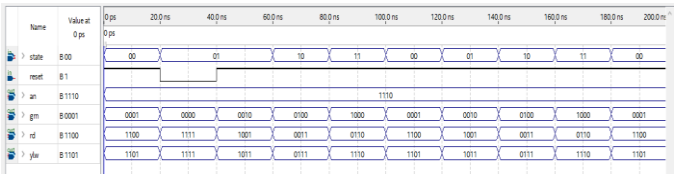

Figure 10. Waveform.

The design and simulation results (RTL, state diagram, and waveform) showed that the traffic signal control system built using FPGA and Verilog HDL operates correctly according to a Finite State Machine (FSM) model, where the transitions between signal states (Green – Yellow – Red) occur in an organized and precise manner. The system responds to sensor signals to detect the presence of vehicles in each direction. Empty lanes are skipped, reducing wasted time. Priority is given to lanes with higher traffic density. The waveform results also showed: Accuracy in signal timing (Timing Accuracy). Correct synchronization between different signals. No conflicts between directions, enhancing traffic safety. Secondly, Quantitative Analysis based on the system behavior as indicated in the simulation, an approximate quantitative analysis can be presented as follows:

Waiting Time in the Traditional System

a fixed time for each direction (e.g., 30–60 seconds) regardless of vehicle presence. In the proposed system, the time is dynamic and depends on traffic density. Estimated reduction in waiting time: 30% – 50%.

Signal Utilization Efficiency Traditional System: A green signal may be given to an empty road → waste of time. Proposed System: The signal is activated only when there are vehicles.

Response Time

Traditional System: Slow response (depends on a fixed time cycle). FPGA-based System: Near-instantaneous response.

Congestion Reduction Thanks to intelligent time distribution among routes:
Significant decrease in traffic queues.
Improved vehicle flow, especially on main roads.
In general, the proposed system reduces time consumption and increases traffic productivity efficiency.

CONCLUSION

The modern ways of multi-way traffic management improve the traffic condition to a large extent. Advanced signaling controllers contribute to the improvement of urban traffic, which is proportional to the complexity

Received*: 11-03-2026 -* ***Accepted****: 09-05-2026 -* ***Published****: 16-05-2026*

of the controller. These more complex controllers can be well handled using state machines. Methods to reduce the states in the state machine also help in reducing the required hardware, thus leading to a low-power and area-efficient design.

Future Work

The future work of this project is it can be directly applied in real time by employing a greater number of such circuits.

REFERENCES

1. Nath S, Pal C, Sau S, Mukherjee S, Roy A, Guchhait A, Kandar D. Design of an Intelligent Traffic Light Controller with VHDL. In: International Conference on Radar, Communication and Computing; 2017 Dec 21-22. p. 92-97.
2. Han T, Lin C. Design of an intelligence traffic light controller (ITLC) with VHDL. In: TENCON '02. Proceedings. IEEE Region 10 Conference on Computers, Communications, Control and Power Engineering; 2016 Oct 28-31. Vol 3. p. 1749-1752.
3. El-Medany WM, Hussain MR. FPGA-Based Advanced Real Traffic Light Controller System Design. In: 4th IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications; 2017. p. 100-105.
4. Singh S, Badwaik SC. Design and Implementation of FPGA-Based Adaptive Dynamic Traffic Light Controller. In: International Conference on Emerging Trends in Networks and Computer Communication (ETNCC); 2018 Apr 22-24; Udaipur, India.
5. Sabri MFM, Husin MH, Abidin WAWZ, Tay KM, Basri HM. Design of FPGA-Based Traffic Light Controller system. Sarawak (Malaysia): Universiti Malaysia Sarawak, Dept. of Electronic Engineering; 2011.
6. Shridhar J, Ruchin, Whig P. Design and Simulation of Power Efficient Traffic Light Controller (PTLCY). In: International Conference on Computing for Sustainable Global Development; 2019. p. 348-352.
7. Singhi PK, Danief P. Advanced Real Traffic Light Controller System Design Using Cortex-M0 IP on FPGA. In: Conference on Advanced Communication Control and Computing Technologies; 2018. p. 1023-1026.
8. Albagul A, Hrairi M, Wahyudi, Hidayathullah MF. Design and development of sensor based traffic light system. Am J Appl Sci. 2017;3(3):1745-1749.

Received*: 11-03-2026 -* ***Accepted****: 09-05-2026 -* ***Published****: 16-05-2026*