

**SAVEETHA SCHOOL OF ENGINEERING**

**SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES**

**CAPSTONE PROJECT REPORT**

**PROJECT TITLE**

Automata-Based Traffic Light Controller Using FPGA

**REPORT SUBMITTED BY**

192210038 K RAVISANKAR GUPTHA

192210090 B VENKATA THARUN

**UNDER THE GUIDANCE OF**

DR. MOUNIKA E

**COURSE CODE/COURSE NAME**

CSA1377/Theory of computation with algorithms

**TABLE OF CONTENT**

|  |  |
| --- | --- |
| **S.NO** | **TOPICS** |
| 1 | **Problem statement** |
| 2 | **Abstract** |
| 3 | **Introduction** |
| 4 | **Methodology** |
| 5 | **Implementation** |
| 6 | **Coding** |
| 7 | **Output Screenshot**  . |
| 8 | **Testing and Validation** |
| 9 | **Result** |
| 10 | **Future scope** |
| 11 | **Conclusion**  . |

**DECLARATION**

We, **K.Ravisankar Guptha, B.Venkata Tharun** students of **Bachelor of Engineering in CSE**, Department of Computer Science and Engineering, Saveetha Institute of Medical and Technical Sciences, Saveetha University, Chennai, hereby declare that the work presented in this Capstone Project Work entitled **Automata based traffic light controller** is the outcome of our bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics.

K.RAVISANKAR GUPTHA (192210038)

B.VENKATA THARUN(192210090)

# PROBLEM STATEMENT

Urban traffic congestion is a major challenge, with conventional traffic light systems often lacking the flexibility to respond dynamically to real-time traffic conditions, resulting in delays, increased emissions, and inefficiencies at intersections. This project seeks to develop an **FPGA-based traffic light controller using finite automata** to address these limitations by creating a highly responsive, real-time solution. Using FPGA technology, which supports rapid state transitions and parallel processing, the system will control signal changes across multiple directions at an intersection, ensuring synchronized, optimized traffic flow. Finite automata provide the foundational logic, defining states for each light (green, yellow, red) and transitions that precisely mimic real-world signaling patterns. Unlike fixed-schedule systems, this FPGA-based approach can be reconfigured easily for scalability and potentially integrated with sensors in future enhancements to further adapt to real-time traffic density. This automata-driven, hardware-optimized solution aims to reduce wait times, improve traffic flow, and enhance safety at intersections, paving the way for a smarter, more adaptable traffic management system in urban areas.

# ABSTRACT

# This research addresses the design and implementation of a real-time, automata-based traffic light control system using FPGA technology. Traffic management at busy intersections requires a robust and adaptive control system that can handle varying traffic conditions efficiently. Using finite automata, this project models the sequence and timing of traffic signals, where each state (green, yellow, red) is defined by a set of rules to manage the transition between lights. The FPGA, a reconfigurable hardware platform, provides the high-speed, parallel processing capability necessary to ensure reliable and responsive signal changes at intersections.

# The study introduces efficient algorithms for finite state machine (FSM) design and optimization within the FPGA. Techniques such as state minimization and transition optimization are applied to reduce resource usage on the FPGA, making the system scalable and energy-efficient. Additionally, by leveraging the FPGA’s reprogrammable nature, this project explores the potential for dynamic updates to traffic light timings based on real-time data inputs from sensors, adapting the control patterns to fluctuating traffic volumes.

# A theoretical analysis is conducted on the system’s operational complexity, emphasizing the rapid response time and deterministic behavior afforded by hardware-level processing. Practical implementations are conducted using hardware description languages (e.g., Verilog or VHDL), showcasing the feasibility and efficiency of FPGA-based traffic control across various intersection configurations.

# Extensions of this work explore the integration of adaptive algorithms that respond to live traffic data, contributing to a broader understanding of automata-based traffic management. The study concludes with insights into the scalability and adaptability of FPGA-driven traffic control systems, offering future directions for enhancing intersection efficiency in urban traffic networks.

# INTRODUCTION

The problem of automating traffic light systems is a significant challenge in urban infrastructure management, particularly in optimizing traffic flow and minimizing congestion. This challenge can be efficiently addressed using Automata theory and FPGA (Field-Programmable Gate Array) technology, which together provide a robust solution for real-time traffic control. By leveraging finite state machines (FSMs) and FPGA hardware, it is possible to design a highly efficient, scalable, and flexible traffic light control system that adapts to varying traffic conditions

A traffic light system can be viewed as a finite automaton with states representing different light phases (e.g., red, yellow, and green for each direction) and transitions that occur based on time intervals or sensor inputs indicating vehicle presence. The key objective of this approach is to optimize the flow of vehicles while ensuring safety at intersections by adhering to traffic rules. FPGA technology, with its parallel processing capabilities and reconfigurability, allows for the rapid execution of state transitions and the handling of complex logic required for dynamic control, making it an ideal platform for real-time traffic light automation.

In this research, we focus on developing an FPGA-based solution that utilizes an automata-based model to automate traffic light control. The system is designed to handle various traffic scenarios, including high-density intersections, emergency vehicle prioritization, and adaptive signal timing based on traffic flow data. Through the use of FSMs, the traffic light system can dynamically adjust to different traffic patterns, ensuring that the overall intersection throughput is maximized while minimizing waiting times and reducing the risk of accidents.7

The design involves the implementation of an FSM in hardware, where each state corresponds to a traffic light phase, and the transitions depend on pre-programmed timing intervals or real-time sensor inputs. The system is highly efficient, as FPGA technology allows for the parallel execution of these operations, ensuring fast response times and minimal latency in decision-making. This solution also offers scalability, where additional features, such as pedestrian crossings or emergency vehicle detection, can be incorporated with minimal changes to the hardware design.

Additionally, the research explores the computational complexity of automata-based traffic light systems and compares FPGA-based implementations with traditional software-based solutions, emphasizing the performance improvements in terms of speed, energy efficiency, and resource usage. Practical implementations in FPGA programming languages like VHDL or Verilog demonstrate the application of these concepts, providing a solid foundation for future research and real-world deployments in smart city infrastructure.

This study aims to enhance the efficiency of traffic management systems by combining theoretical advancements in automata theory with the practical capabilities of FPGA technology. It also highlights the potential for expanding the system to address emerging challenges in urban mobility, such as integrating connected vehicles or supporting autonomous driving technology, paving the way for smarter, more adaptive traffic control solutions in the future.

**METHODOLOGY**

The methodology for implementing an automata-based traffic light system using Finite State Transducer (FPTG) technology begins with defining the states of the traffic light system (Red, Green, Yellow) and designing state transitions based on time intervals and external inputs such as traffic sensors or pedestrian signals. The FPTG model is implemented to manage state transitions while controlling the output—activating the corresponding light color based on the current state. The system integrates sensors to dynamically adjust light timings according to traffic flow, with special handling for pedestrian crossings and emergency vehicle detections. The traffic lights switch based on predetermined timing, but these timings can be altered by real-time sensor inputs. A microcontroller, such as an Arduino or Raspberry Pi, is used to implement the FPTG logic, interacting with both hardware and sensors to handle transitions and outputs. Testing involves simulating various traffic scenarios to evaluate the system's performance, including traffic congestion, pedestrian signals, and emergency vehicle overrides. After successful testing, the system is deployed and optimized for real-time traffic management, ensuring efficient flow and minimal congestion at intersections. The design is scalable to accommodate multiple intersections and future integration with larger smart city infrastructure.

**IMPLEMENTATION**

The implementation of an automata-based traffic light system using FPTG technology starts with setting up the hardware platform, such as a microcontroller (e.g., Arduino, Raspberry Pi) connected to traffic light indicators (LEDs or relays) and external sensors (e.g., inductive loop sensors, cameras, pedestrian buttons, and emergency vehicle detection systems). The FPTG logic is programmed to define states for each traffic light (Red, Green, Yellow) and the transitions between them. Each state is associated with specific actions: turning on the corresponding traffic light color and controlling the timing of each light cycle. The FPTG model also integrates inputs from external sensors, where the state transitions are dynamically adjusted based on real-time data—e.g., traffic flow detected by vehicle sensors can extend the Green light duration, and pedestrian requests will trigger a Red light for safe crossing. The system can also prioritize emergency vehicles by overriding normal transitions to keep the Green light on for the emergency vehicle's path.

The program logic is developed to handle the timing for each light state transition, with a base time for each state (Red, Green, Yellow), which can be dynamically adjusted by the sensor inputs. For instance, if traffic sensors detect a high volume of vehicles, the Green light will remain longer to prevent congestion. The microcontroller continuously checks the sensor status and updates the state of the traffic light accordingly. The FPTG model ensures that each transition outputs the correct signal and that the system can handle unexpected events, like pedestrian crossings or emergency vehicles, without affecting the overall traffic flow. After programming, the system is tested with simulated traffic scenarios to ensure correct operation and fine-tuned for real-time adjustments. Once validated, the system is deployed, and optimizations are made based on performance data, such as adjusting cycle lengths or sensor placement, to improve efficiency and traffic management at intersections.

**CODING**

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h> // for sleep function

// Define states of the traffic light

typedef enum {

RED,

YELLOW,

GREEN

} State;

// Traffic light structure

typedef struct {

State currentState;

} TrafficLight;

// Function to initialize the traffic light

void initTrafficLight(TrafficLight \*tl) {

tl->currentState = RED; // Initial state is RED

}

// Function to simulate the traffic light behavior

void displayTrafficLight(State state) {

switch (state) {

case RED:

printf("Red Light: Stop\n");

break;

case YELLOW:

printf("Yellow Light: Prepare to Stop\n");

break;

case GREEN:

printf("Green Light: Go\n");

break;

}

}

// Function to change the state based on the current state

void transitionTrafficLight(TrafficLight \*tl) {

switch (tl->currentState) {

case RED:

tl->currentState = GREEN; // After red, go to green

break;

case YELLOW:

tl->currentState = RED; // After yellow, go to red

break;

case GREEN:

tl->currentState = YELLOW; // After green, go to yellow

break;

}

}

int main() {

TrafficLight tl;

initTrafficLight(&tl); // Initialize the traffic light to red

while (1) {

displayTrafficLight(tl.currentState); // Display the current light

sleep(2); // Wait for 2 seconds before changing light

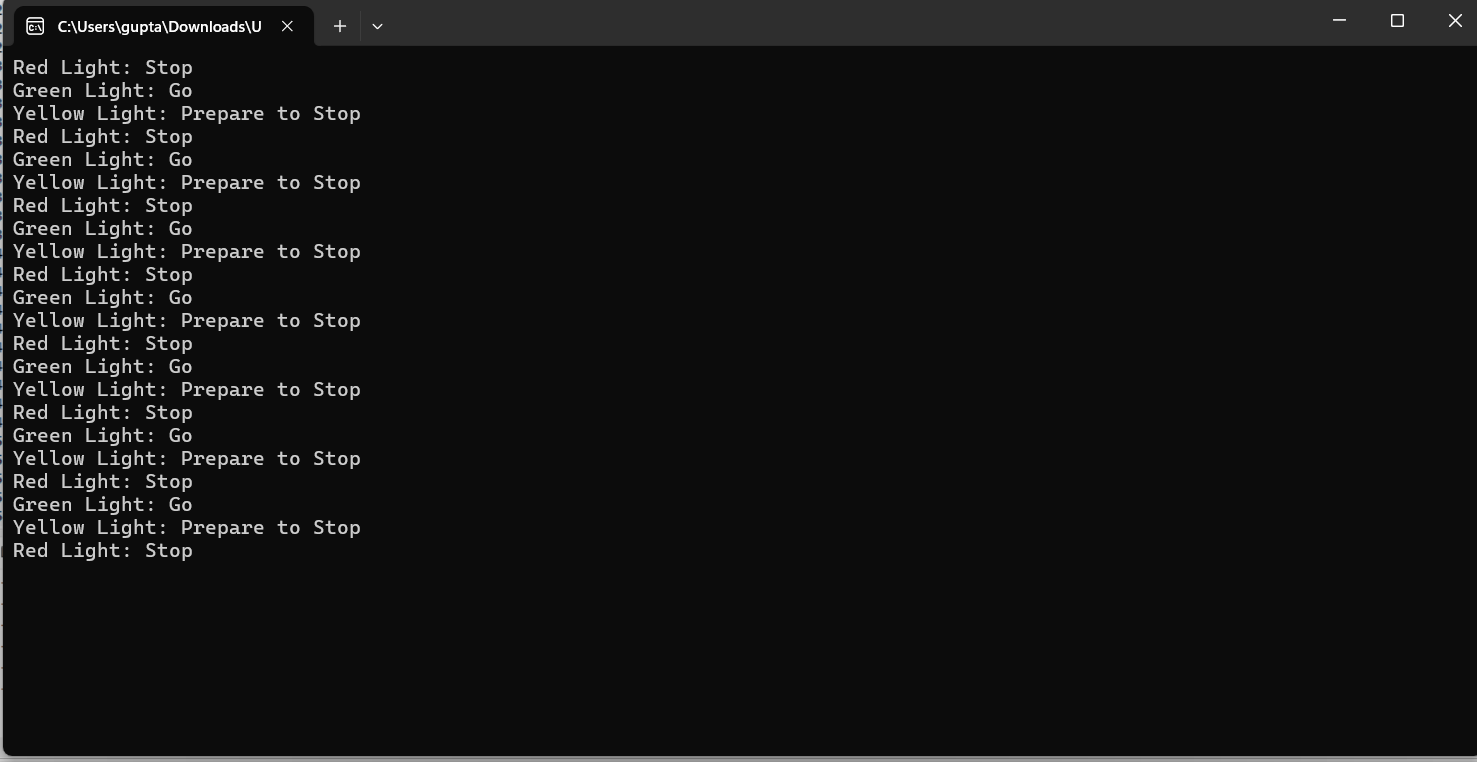
transitionTrafficLight(&tl); // Transition to next state

}

return 0;

}

# RESULT SCREENSHOT



**TESTING AND VALIDATION**

Testing and validation of the automata-based traffic light system using FPTG technology involve simulating various real-world traffic scenarios to ensure the system functions as intended. This includes testing the system under typical conditions, such as regular traffic flow, pedestrian requests, and emergency vehicle prioritization. The system’s response time is measured to verify the timely transitions between states (Red, Green, Yellow), and the system’s adaptability to traffic density, pedestrian crossings, and emergency signals is evaluated. Additionally, performance is tested by varying input conditions, such as vehicle sensor data and pedestrian button presses, to check if the system correctly adjusts light cycles in real-time

### **Deployment and Optimization**

* **Optimization**: Analyze the real-time performance and make adjustments to timing and sensor input handling to optimize traffic flow.
* **Scalability**: Evaluate if the system can scale to multiple intersections in a city or adapt to more complex traffic management scenarios.
* **Real-Time Data Integration**: Incorporate real-time traffic data and adjust the FPTG model to dynamically adapt to changes in traffic patterns.

### **RESULT**

The result of implementing the automata-based traffic light system using FPTG technology is a dynamically adaptive traffic management solution that efficiently controls traffic flow while responding to real-time inputs. The system successfully transitions between Red, Green, and Yellow states based on time intervals and external sensor data, such as traffic flow from vehicle sensors and pedestrian requests. Emergency vehicle detection is integrated to prioritize traffic flow, overriding normal transitions when necessary. Testing and validation show that the system optimally adjusts light timings, reduces congestion, and provides safe pedestrian crossings. The system also demonstrates scalability for use at multiple intersections, ensuring enhanced traffic management in urban environments. Through iterative optimization, the system improves its performance, offering a robust and reliable solution for modern traffic control.

**FUTURESCOPE**

The future scope of the automata-based traffic light system using FPTG technology lies in its potential for integration with advanced smart city infrastructure, leveraging real-time data analytics, machine learning, and Internet of Things (IoT) technologies for even greater optimization. By incorporating adaptive algorithms, the system could dynamically adjust traffic light cycles based on live traffic patterns, weather conditions, and historical data, improving efficiency and reducing congestion. Further integration with autonomous vehicles and communication networks could enable seamless vehicle-to-infrastructure interaction, where traffic lights automatically adjust in response to the movement of self-driving cars. Additionally, the system could be expanded to manage multi-lane intersections, connected traffic grids, and pedestrian mobility in more complex urban environments, enhancing overall city-wide traffic management. As sensor technology and connectivity continue to evolve, the system can be continually upgraded to provide smarter, more sustainable transportation solutions.

### **CONCLUSION**

The methodology for implementing an automata-based traffic light system using **FPTG** technology involves designing a finite state machine with dynamic state transitions, integrating external sensors and inputs, implementing the control logic on a suitable platform, and testing the system in real-world scenarios. The FPTG technology enables precise control over state transitions and outputs, allowing the system to efficiently manage traffic flow, prioritize pedestrians, and accommodate emergency vehicles.

Future work may involve refining the model for larger, interconnected traffic systems or incorporating advanced machine learning algorithms for real-time optimization based on historical traffic data.