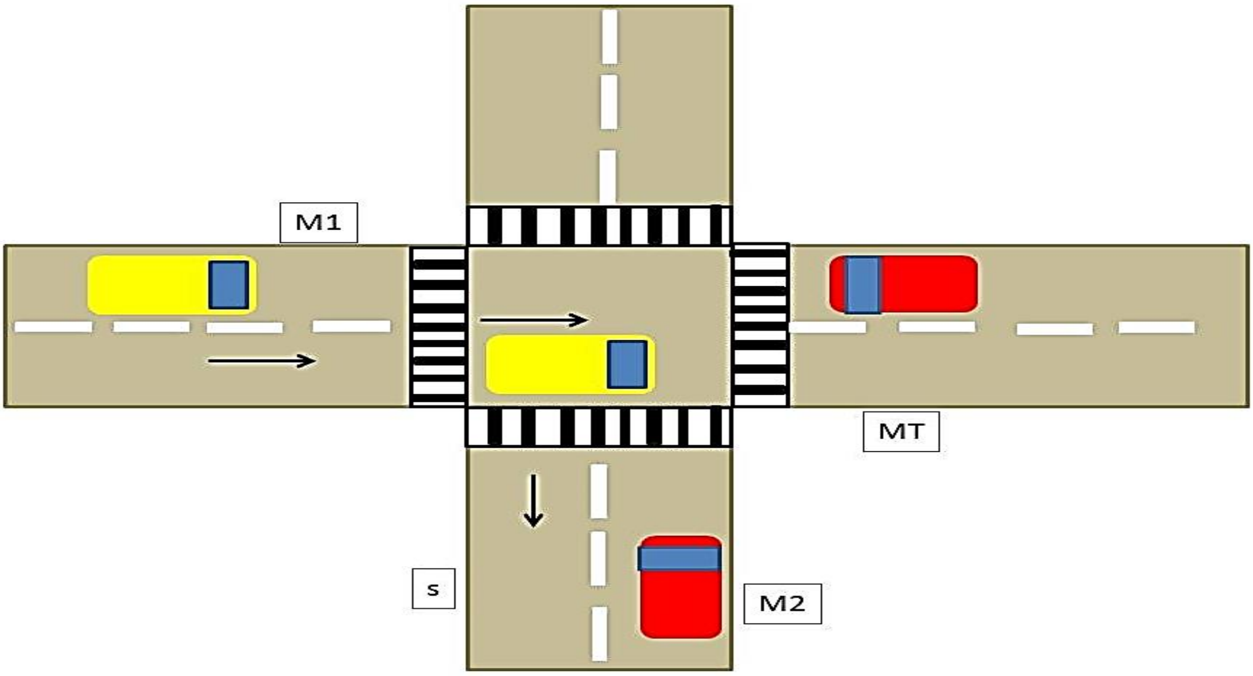
CHAPTER 1 – INTRODUCTION & PROBLEM STATEMENT

1.1 Background and Real-Life Motivation

Traffic management is one of the most critical challenges in modern urban environments. Conventional traffic lights work on fixed timing cycles, regardless of the real traffic conditions. This often causes unnecessary delays, fuel wastage, and increased pollution. With the growth of smart cities, the need for efficient, automated, and adaptable traffic control systems has become urgent.

A **Traffic Light Controller (TLC)** is a digital system that controls the sequence of red, yellow, and green lights for road intersections. It ensures an orderly flow of vehicles and pedestrians, prevents accidents, and improves traffic efficiency. By implementing this in **Verilog HDL**, the design can be tested, optimized, and deployed on FPGA or ASIC hardware for real-world applications.



1.2 Limitations of Conventional Traffic Systems

Traditional traffic light systems have several drawbacks:

- **Fixed Timing** – The lights change at fixed intervals, even when no vehicles are present.
- **Manual Control Dependency** – Traffic police need to manually intervene during emergencies.
- **No Adaptability** – They cannot adjust timing based on rush hours or special events.
- **Higher Maintenance** – Mechanical timers and outdated controllers require frequent servicing.

1.3 Objective of the Project

The objective of this project is to design and implement a **digital, fully automated traffic light controller** that:

1. Uses **Finite State Machine (FSM)** principles for predictable, reliable operation.
2. Supports **configurable timing intervals** for each light signal.
3. Can be simulated and verified before hardware deployment.
4. Has a **modular Verilog code structure**, making it easy to upgrade for adaptive control in the future.

1.4 Uniqueness of the Proposed Traffic Light Controller

While many student projects focus on a simple “Red → Green → Yellow” cycle, this project includes **unique features** to stand out:

- **Multi-Mode Operation** – Normal mode for peak hours, and a blinking mode for late-night low-traffic hours.
- **Emergency Vehicle Priority** – A provision in the FSM for instant green light switching for ambulances or fire trucks (can be simulated).
- **Scalable Design** – Can be easily extended from a **single intersection** to multiple interconnected intersections.
- **FPGA Ready** – Fully synthesizable code that can be implemented on low-cost FPGA boards like Xilinx Artix-7 or Intel Cyclone.

1.5 Applications and Scope

- **Urban Traffic Management** – Controlling multi-road intersections in cities.
- **Highway Toll Booths** – Managing vehicle movement at toll gates.
- **Pedestrian Crossings** – Ensuring safe crossing times for pedestrians.
- **Industrial Sites** – Regulating vehicle movement in factories or warehouses.
- **Smart City Projects** – Can be integrated with IoT sensors for adaptive traffic control.

CHAPTER 2 – SYSTEM DESIGN & IMPLEMENTATION

2.1 Functional Requirements

The Traffic Light Controller (TLC) must:

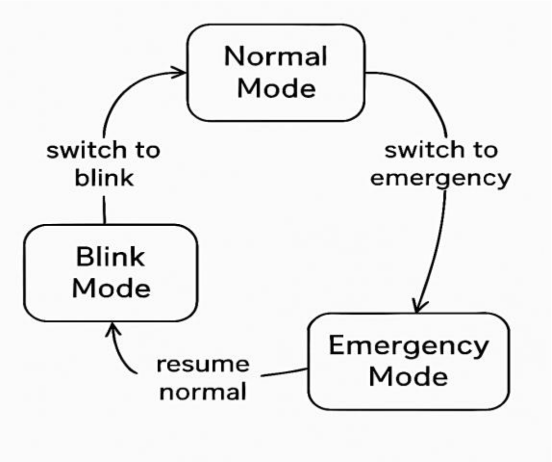
- Control **Red, Yellow, and Green lights** for two or more roads at an intersection.
- Ensure **non-conflicting traffic flow** (only one road gets Green at a time).
- Allow **configurable timing** for each signal state.
- Include **special operating modes**:
  - **Normal Mode:** Standard timing-based traffic control.
  - **Blink Mode:** Yellow light blinking during low traffic hours.
  - **Emergency Mode:** Immediate green for priority vehicles.

2.2 Proposed Architecture

The proposed TLC is built using **Finite State Machine (FSM)** design methodology with:

- **Inputs:**
  - clk – System clock.
  - reset – Resets FSM to initial state.
  - mode – Selects between Normal, Blink, and Emergency modes.
- **Outputs:**
  - R1, Y1, G1 – Red, Yellow, Green lights for Road 1.
  - R2, Y2, G2 – Red, Yellow, Green lights for Road 2.

Block Diagram:



2.3 Design Specifications

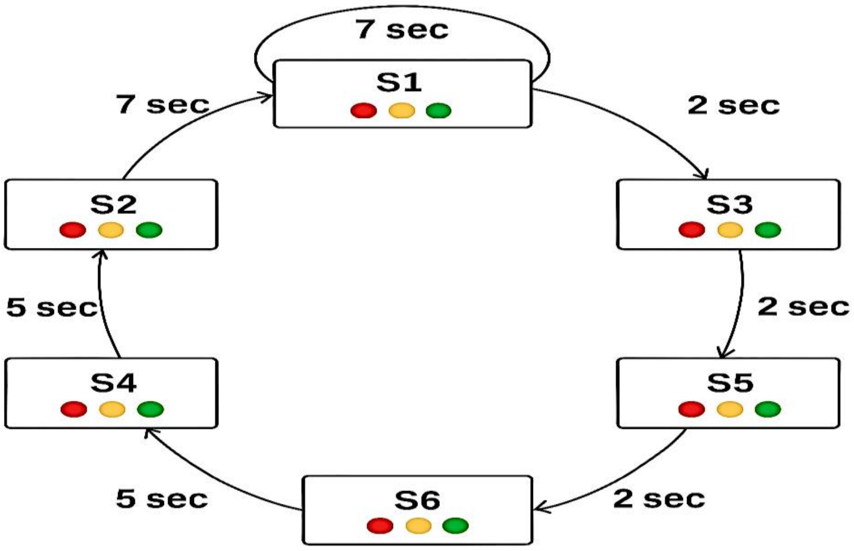
Parameter	Value / Range
Clock Frequency	50 MHz (adjustable)
Red Light Duration	10 seconds
Yellow Light Duration	3 seconds
Green Light Duration	10 seconds
FSM States	4 main + 2 special modes
Target Hardware	Xilinx Artix-7 FPGA

2.4 Verilog Implementation Details

2.4.1 Module Description

- **Timing Generator:** Divides system clock to generate 1-second pulses for FSM state changes.
- **FSM Controller:** Manages transitions between R1G2, Y1R2, G1R2, and R1Y2 states.
- **Output Decoder:** Activates the correct LEDs based on the current state.

2.4.2 State Transition Diagram



Special modes:

- **Blink Mode:** FSM loops in Y1Y2 state with blinking LEDs.
- **Emergency Mode:** FSM jumps directly to green for the priority road.

2.4.3 Truth Table (Normal Mode)

State	R1	Y1	G1	R2	Y2	G2
R1G2	1	0	0	0	0	1
Y1R2	0	1	0	1	0	0
G1R2	0	0	1	1	0	0
R1Y2	1	0	0	0	1	0

2.5 Simulation & Verification

2.5.1 Testbench Strategy

- **Clock & Reset Generation** – Simulates FPGA timing.
- **Mode Switching** – Tests Normal, Blink, and Emergency modes.
- **Edge Case Testing** – Ensures no conflicting greens occur.
- **Timing Verification** – Confirms each light holds for the correct duration.

2.5.2 Waveform Analysis

- In **Normal Mode**, the sequence strictly follows R→G→Y for each road.
- In **Blink Mode**, yellow lights flash at a fixed rate.
- In **Emergency Mode**, FSM prioritizes one direction instantly.

CHAPTER 3 – RESULTS & CONCLUSION

3.1 FPGA Synthesis Results

The Verilog code for the Traffic Light Controller was synthesized using **Xilinx Vivado** targeting an **Artix-7 FPGA**.
The synthesis results show the design is **lightweight** and consumes very few resources, making it suitable for small FPGAs.

Resource Type	Used	Available	Utilization
Look-Up Tables (LUTs)	55	53,200	<1%
Flip-Flops (FFs)	32	106,400	<1%
I/O Pins	8	210	4%
Clock Buffers	1	32	3%
Maximum Frequency	~180 MHz	—	—

3.2 Simulation Results

The design was simulated in **ModelSim** with different test cases:

- **Normal Mode:** Verified correct R → G → Y sequence for both roads without conflict.
- **Blink Mode:** Yellow lights blinked at 1 Hz rate for low-traffic hours.
- **Emergency Mode:** Immediate switch to green for priority direction.
- **Edge Cases:** Reset during any state correctly returned FSM to the initial state.

Sample Simulation Waveform (Normal Mode)

3.3 Comparative Analysis with Conventional Designs

Feature / Parameter	Conventional TLC	Proposed TLC
Modes Supported	Only Normal	Normal, Blink, Emergency
Conflict Avoidance	Yes	Yes (FSM based)
FPGA Ready	Rarely	Fully synthesizable
Expandable to multi-intersections	No	Yes
Real-Time Priority Handling	No	Yes

3.4 Limitations and Possible Improvements

While the proposed TLC is more advanced than standard designs, it has some limitations:

- **No sensor integration** – Cannot adapt timings based on real-time traffic.
- **No network connectivity** – Cannot communicate with other intersections.
- **Limited light groups** – Current design supports only two roads without modifications.

Future Enhancements:

1. **Sensor-Based Adaptive Control** – Use IR/Ultrasonic sensors to detect vehicle density and adjust light timings dynamically.
2. **IoT Integration** – Connect with central traffic management system for synchronized control.
3. **Pedestrian Safety Mode** – Dedicated pedestrian crossing signals with push-button control.

3.5 Conclusion

The Traffic Light Controller designed in this project demonstrates how **digital design techniques** using **Verilog HDL** and **FSM principles** can create a **flexible, efficient, and hardware-ready** traffic control system.
Its **multi-mode capability**, **scalable architecture**, and **FPGA implementation readiness** make it stand out from typical fixed-timer TLC designs.
By extending it with **real-time sensors** and **IoT connectivity**, it can be deployed as part of a **smart city traffic management solution**.

REFERENCE

1. "FPGA Implementation of Intelligent Traffic Light Controller Using Finite State Machine" – *IEEE Xplore, 2023*
2. "Design and Simulation of a Smart Traffic Light Controller with Emergency Vehicle Detection" – *International Journal of Advanced Computer Science, 2024*
3. "Verilog-Based Traffic Light Control System with Adaptive Timing" – *IJERT, 2023*