William V.S. Tubman University

Tubman Town, East Harper, Maryland County, Liberia

Smart Traffic Controller, Project Documentation.

Course: Digital Computer Circuits

Course code: CSENG 406

Summited by:

Hannibal P. Davis ………… 10793

Mai S.J Reed ………………. 28530

Alphonson Nah………….

Syvester ……………..

Summited to:

Mr. Duah Jeremiah Leakpor

Instructor

Date: May, 2025

PROJECT TITLE

Smart Traffic Intersection Controller with Emergency Override and Adaptive Timing

1.0 INTRODUCTION

Traditional traffic light systems often rely on pre-programmed cycles that may not efficiently respond to real-time traffic conditions. Therefore, in the context of 21^{1st} Century, a smart traffic control system is designed to optimize traffic light cycles based on real-time data and pedestrian crossing support, with special prioritization for emergency vehicles. In this project, a simplified model of an intelligent traffic signal controller for a four-way intersection is designed using fundamental digital logic components within the **Logisim Evolution** environment. It aims to reduce congestion and enable emergency vehicles to navigate intersections more efficiently.

In the absence of employing microcontrollers or software-based logic, the system uses only hardware-level digital design concepts—such as flip-flops, logic gates, and finite state machines (FSMs)—to simulate automated control of traffic signals. This project showcases how principles of digital electronics can solve real-world problems and lays the groundwork for more advanced adaptive traffic control systems.

1.1 Problem Statement

Urban areas face significant traffic congestion, often resulting in delays for emergency vehicles at intersections. This project addresses this problem by automatically adjusting traffic lights to prioritize emergency vehicles and manage vehicle density more effectively.

1.2 Objectives

The key goals of the project are:

- To simulate a real-world traffic light controller for a four-way intersection using digital logic.
- To apply combinational and sequential logic in designing a fully functional finite state machine.

- To ensure safe and realistic traffic signal transitions (Green → Yellow → Red) for each direction.
- To demonstrate pedestrian crossing support.
- Prioritizes emergency vehicles.
- To provide a modular and scalable design for future integration with sensors or adaptive control logic.

1.3. Features

- **Real-Time Traffic Control:** Dynamically adjusts traffic light timings based on real-time vehicle density at each intersection.
- **Pedestrian Crossing:** Allows safe crossing of human beings when the pedestrian button is press.
- **Emergency Vehicle Detection and Prioritization:** Detects emergency vehicles on the road and grants them a green signal to pass through without delay.
- **Adaptive Timing Mechanism:** Uses the vehicle count and rate of increase on each road to calculate optimized green-light durations.

2.0 TOOLS AND TECHNOLOGIES

Tool / Component	**Purpose**
Logisim Evolution 3.9.0	Simulation of digital circuits and real-time visualization
Logic Gates (AND, OR, NOT, NOR)	Implement decision-making and signal routing logic
D Flip-Flops	Store current state in FSM and facilitate clock-based transitions
Registers	Store temporary or intermediate logic values
Clock Source (Manual/OSC)	Drive timing and transitions in the FSM
Pins & LEDs	Represent control inputs and signal outputs for visualization

3.0 SYSTEM DESIGN OVERVIEW

The controller uses an 8-state **Finite State Machine (FSM)** model, where each state corresponds to a unique phase of traffic flow, ensuring that only one direction gets the green signal at a time while others remain red or transition through yellow.

A **3-bit binary counter** using D flip-flops increments on each clock pulse and feeds a **3-to-8 decoder**, which activates only one state output at a time. These state outputs are then logically combined to drive the correct traffic light LEDs.

3.1 State Definitions

Each FSM state controls traffic for one direction, with safe yellow transitions between green and red:

State	North	East	South	West
S0	Green	Red	Red	Red
S1	Yellow	Red	Red	Red
S2	Red	Green	Red	Red
S3	Red	Yellow	Red	Red
S4	Red	Red	Green	Red
S5	Red	Red	Yellow	Red
S6	Red	Red	Red	Green
S7	Red	Red	Red	Yellow

3.2. Circuit Components and Implementation

Key Components:

- **Counter (FSM State Machine)**: Constructed using three D flip-flops configured as a modulo-8 counter.

- **3-to-8 Decoder**: Translates 3-bit binary counter output into a one-hot signal, activating one state at a time.
- **Logic Gates**: AND and NOR gates combine state signals to control each direction's traffic lights (R/Y/G).
- **LED Groups**: Each direction has a trio of LEDs—Green, Yellow, and Red—to represent traffic light output.
- **Clock Source**: Manual button or oscillator circuit to advance the FSM at fixed intervals.

3.3 Operation Sequence

On each rising clock edge:

- The counter advances by 1 (binary).
- The decoder activates one of the eight states (S0–S7).
- Combinational logic processes this state and drives the LED outputs accordingly.
- The system loops back to S0 after S7, forming a continuous control cycle.

4.0 TESTING AND RESULTS

Testing was conducted in Logisim Evolution with the following process:

- Step-by-step state transitions were observed using manual clock pulses.
- Verified exclusive green lighting for only one direction at any time.
- Confirmed that yellow phases always precede a red transition.
- The reset mechanism correctly returned the FSM to the initial state (S0).
- All traffic directions were verified through the complete 8-state loop.
- When the pedestrian crossing button is press, the counter continually reads from 0 – 9 and on stop until the button is press again. This is actually not the expected outcome.
- When an emergency is present, the traffic light continues in its normal states. Somehow, not prioritizing emergency.

Outcome: The circuit achieved the designed and functionality of the 4-ways traffic light, but unable to validate the pedestrian crossing and emergency override functionality and safety logic.

4.1. Challenges Encountered

- **Pedestrian Crossing**: To demonstrate pedestrian crossing support is challenging. However, we managed to solve it partially. The countdown is reading, but in an increasing mode. Also, the does not reading until it is press again.
- **Emergency override**: To enable traffic priority for emergency vehicles (ambulances, fire trucks). This challenge is unsolved, but still working on arriving at solution.
- **Clock Debounce Issue**: Fast clock transitions caused glitches in state detection; solved by using a slower or debounced clock.
- **LED Conflict Resolution**: Conflicting outputs (e.g., unintended multiple greens) were avoided by refining logic gate arrangements, including the strategic use of NOR gates.

5.0 CONCLUSION AND FUTURE ENHANCEMENTS

5.1. Conclusion

The **Smart Traffic Controller** project successfully models an automated traffic light system using purely digital logic circuits. It demonstrates how FSMs and sequential logic can provide real-time decision-making without software programming or microcontroller dependence.

This project not only reinforces core digital electronics concepts—like flip-flops, counters, and logic gates—but also illustrates how these building blocks can be scaled into smart infrastructure applications. Future extensions can incorporate sensors and intelligent control, offering an educational path toward more complex IoT-enabled systems.

5.2. Future Enhancements

This project lays a foundation for multiple real-world improvements:

- **Pedestrian Signals**: Work around finding the solution and ensure achieving pedestrian crossing excellent functionality which should include push-button pedestrian inputs with timing interlocks for safety.
- **Emergency Override**: Ensure traffic priority for emergency vehicles (ambulances, fire trucks) is achieve.
- **Sensor Integration**: Add vehicle detection (e.g., IR or pressure sensors) to adjust timing dynamically based on real traffic.
- **Microcontroller Upgrade**: Replace FSM with programmable microcontroller (e.g., Arduino) for advanced control.
- **Monitoring Dashboard**: Develop a GUI or simulation dashboard to visualize traffic phases and timing in real-time.