Design and Implementation of an Advanced FSM-Based Traffic Light Control System

Bishmoy Paul
Department of Electrical and Computer Engineering
Santa Clara University
bpaul@scu.edu

ABSTRACT

This paper presents the design and implementation of two enhanced traffic light controller systems using modern state-based Verilog HDL, improving upon traditional gate-level implementations. The first controller implements a simplified four-state finite state machine with manual state transitions triggered by external signals, providing a straightforward control mechanism for intersection management. The second design offers a more sophisticated automatic controller featuring timer-based state transitions, multiple operational modes (normal, increased traffic, emergency), and adaptive timing based on traffic sensor inputs. Both controllers utilize a state machine architecture with clearly defined states for horizontal and vertical traffic flows. Unlike previous gate-level designs that require explicit signal routing, our state-based approach significantly improves code readability, maintainability, and design flexibility. Synthesis results using Synopsys Design Compiler with the SAED32nm_EDK library demonstrate efficient resource utilization while maintaining timing constraints. Comprehensive verification through Synopsys VCS simulation confirms the proper functionality across all operation modes, including emergency vehicle prioritization and congestion-based timing adjustments. This work demonstrates how modern HDL methodologies can enhance traffic management systems with greater adaptability to varying traffic conditions while maintaining implementation simplicity.

I. INTRODUCTION

Traffic management is a critical challenge in modern urban development, with significant implications for road safety, traffic flow efficiency, energy consumption, and environmental impact. As cities continue to grow and vehicle ownership increases worldwide, the development of intelligent traffic light systems becomes essential for managing congestion and optimizing roadway utilization.

While hardware description languages (HDLs) like Verilog have long been used to implement traffic controllers, many existing designs rely on gate-level implementations that are difficult to modify, maintain, and adapt to changing traffic requirements.

This paper presents two enhanced traffic light controller designs that leverage modern state-based Verilog HDL methodologies to improve upon traditional approaches. The first design implements a simplified four-state finite state machine (FSM) with manual state transitions, providing a clear and intuitive control mechanism for basic intersection management. The second, more advanced controller introduces timer-based automatic state transitions, multiple operational modes (normal, increased traffic, and emergency), and adaptive timing based on real-time traffic sensor inputs.

Our work builds upon the research of Zhilevski et al. [1], who demonstrated the feasibility of using Verilog HDL for traffic light control systems with multiple operation modes. While their implementation relied on gate-level descriptions with explicit signal routing, our state-based approach significantly improves code readability, maintainability, and design flexibility.

The primary contributions of this paper include:

1. An improved design approach using state-based Verilog (instead of gate-level logic)
2. A practical implementation with three operating modes and traffic-adaptive timing
3. Synthesis results using Synopsys Design Compiler with SAED32nm_EDK library
4. Verification results confirming functionality across all operating conditions

The remainder of this paper is organized as follows: Section II provides background on traffic light control systems and discusses related work in the field. Section III presents our design methodology and the architectural details of both controller implementations. Section IV describes the simulation and verification process. Section V presents synthesis results and discusses performance metrics. Section VI concludes the paper and outlines potential future work in this area.

II. BACKGROUND

Traffic management systems are critical for urban infrastructure, addressing issues of congestion, safety, and environmental impact. Traditional traffic light controllers operate on fixed timing sequences regardless of actual traffic conditions, leading to inefficiencies during non-peak hours and inadequate responses to fluctuations in traffic patterns [2].

Previous implementations of traffic light controllers using Hardware Description Languages (HDLs) have demonstrated feasibility but often rely on gate-level descriptions with explicit signal routing. Zhilevski et al. [1] developed a traffic light system using Verilog HDL with three operational modes (normal, increased traffic, and emergency), but their implementation used low-level gate logic that limits maintainability and adaptability.

Recent advances in intelligent traffic systems incorporate sensors and real-time data to dynamically adjust signal timing [2]. However, many such systems still lack the flexibility needed for emergency vehicle prioritization and adaptive response to unusual congestion patterns. State-based design approaches offer significant advantages over gate-level implementations, including improved readability, maintainability, and the ability to easily modify timing parameters and state transitions as traffic requirements evolve.

III. METHODS

The design methodology for our traffic light controllers builds on the foundation established by Zhilevski et al. [1] while advancing beyond their gate-level implementation to a more maintainable state-based approach. The traffic intersection under consideration consists of two perpendicular roads with two-way traffic. While the physical intersection would require eight traffic lights (four for vehicles and four for pedestrians), we simplify our design to focus on just two vehicle traffic lights - one controlling horizontal traffic and one controlling vertical traffic. This simplification is possible because traffic lights for parallel lanes operate synchronously. Additionally, since pedestrian traffic lights simply complement the vehicle signals (green for pedestrians when vehicles have red, and red for pedestrians when vehicles have green or yellow), we focus exclusively on modeling the vehicle traffic light controller.

To validate our state-based approach, we developed two traffic controller implementations of increasing complexity:

System 1: Simplified Controller - Implements basic functionality with manual state transitions, providing fundamental traffic management capabilities without automated timing. This design serves as a proof-of-concept for our state-based methodology.

System 2: Advanced Controller - Extends the first design with automatic timer-based transitions and supports three distinct operational modes:

Normal Operation Mode: The default mode for regular traffic conditions with balanced timing for all directions.

Increased Traffic Mode: An adaptive mode that extends green light duration for congested roads, improving traffic flow during peak hours.

Emergency Mode: A priority mode that facilitates rapid passage of emergency vehicles through the intersection.

We explain the detailed design and implementation of both systems in the subsequent sections.

A. Simplified Traffic Light Controller Design

The simplified controller implements a four-state finite state machine (FSM) with manual state transitions. This design serves as a proof-of-concept for the state-based approach while providing basic traffic light control functionality for an intersection with perpendicular roads.

State Definition and Transitions

The controller defines four distinct states representing different traffic light configurations:

STATE_H_RED_V_GREEN: Horizontal lights are RED while vertical lights are GREEN, allowing vertical traffic to flow.

STATE_H_YELLOW_V_YELLOW: A transition state where horizontal lights are YELLOW (preparing to turn GREEN) while vertical lights are also YELLOW (preparing to turn RED).

STATE_H_GREEN_V_RED: Horizontal lights are GREEN while vertical lights are RED, allowing

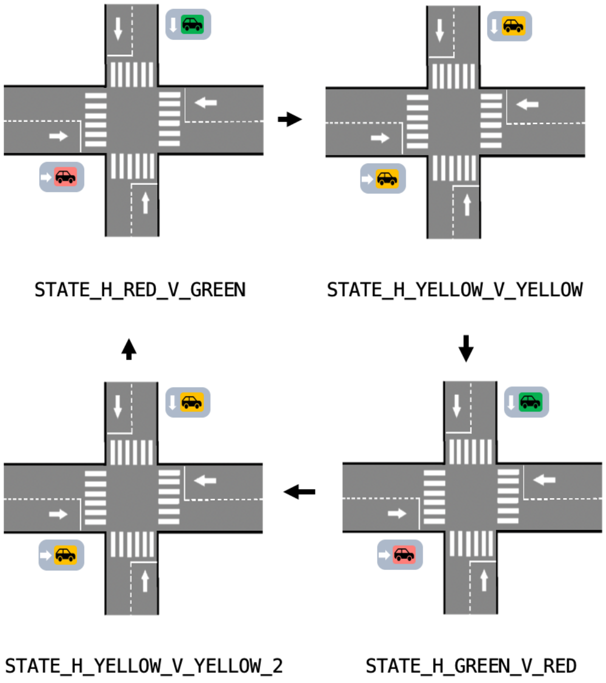


Figure 1: Simplified Traffic System

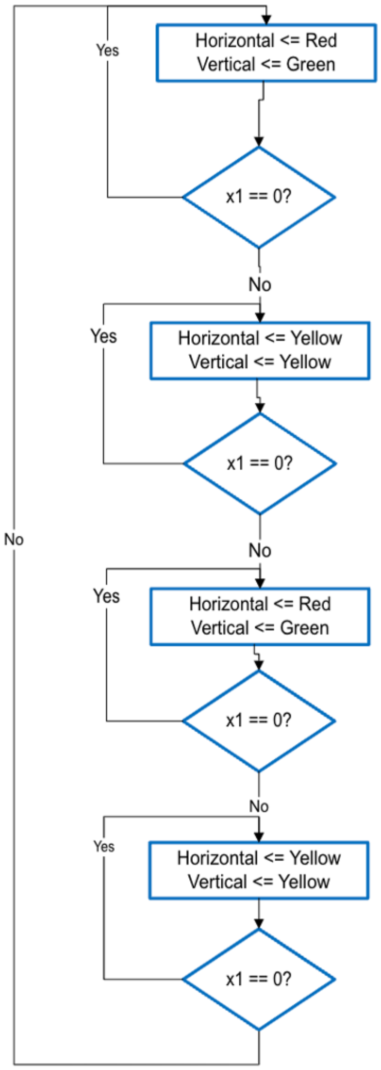


Figure 2: Simplified Controller States

horizontal traffic to flow.

STATE_H_YELLOW_V_YELLOW_2: A second transition state where horizontal lights are YELLOW (preparing to turn RED) while vertical lights are also YELLOW (preparing to turn GREEN).

These states form a cyclic sequence that ensures safe traffic flow at the intersection. State transitions are controlled by an external signal (x1), which allows for manual advancement through the traffic light sequence. When the signal is activated, the controller progresses to the next state in the sequence, completing the full cycle as: RED-GREEN → YELLOW-YELLOW → GREEN-RED → YELLOW-YELLOW → RED-GREEN.

The state diagram illustrated in Fig. 2 shows these transitions and their associated output logic. This straightforward implementation provides clear visualization of the traffic light sequence while demonstrating the benefits of a state-based approach over traditional gate-level design.

B. Advanced Traffic Light Controller Design

Building upon the simplified controller, our advanced traffic light controller implementation features timer-based automatic state transitions, multiple operational modes, and adaptive timing based on traffic conditions. This design provides a comprehensive solution for managing traffic flow efficiently in various scenarios.

State Definition and Transitions

The advanced controller implements a more sophisticated state machine with ten distinct states to handle a complete traffic light cycle (normal transition states are shown in Fig. 3):

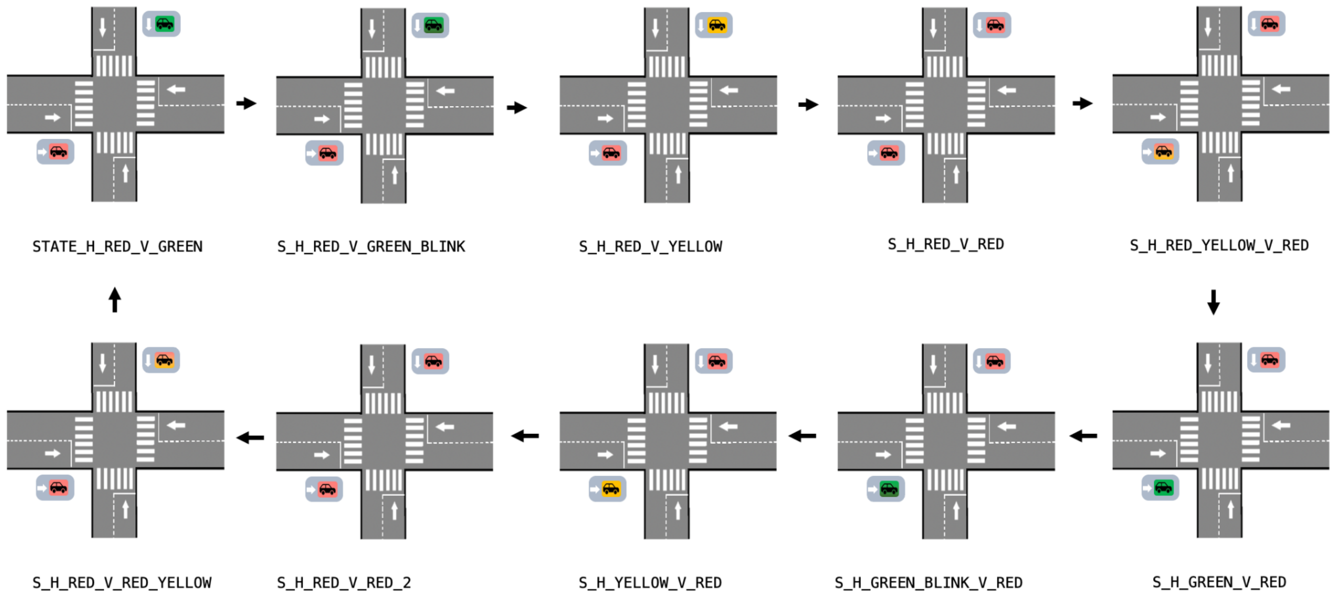


Figure 3: Advanced Traffic System

S_H_RED_V_GREEN: Horizontal lights are RED while vertical lights are GREEN, allowing vertical traffic to flow.

S_H_RED_V_GREEN_BLINK: Horizontal lights remain RED while vertical lights BLINK GREEN, warning that the green phase is ending soon.

S_H_RED_V_YELLOW: Horizontal lights are RED while vertical lights are YELLOW, indicating imminent stop for vertical traffic.

S_H_RED_V_RED: A safety state where both horizontal and vertical lights are RED.

S_H_RED_YELLOW_V_RED: Horizontal lights show RED+YELLOW (preparing to go) while vertical lights are RED.

S_H_GREEN_V_RED: Horizontal lights are GREEN while vertical lights are RED, allowing horizontal traffic to flow.

S_H_GREEN_BLINK_V_RED: Horizontal lights BLINK GREEN while vertical lights remain RED, warning that the green phase is ending soon.

S_H_YELLOW_V_RED: Horizontal lights are YELLOW while vertical lights are RED, indicating imminent stop for horizontal traffic.

S_H_RED_V_RED_2: A second safety state where both directions show RED.

S_H_RED_V_RED_YELLOW: Horizontal lights are RED while vertical lights show RED+YELLOW (preparing to go).

S_EMERGENCY: A special state for emergency situations with flashing YELLOW in both directions.

Unlike the simplified controller, transitions between states occur automatically based on timer values rather than manual control. The controller maintains an internal timer that triggers state progression when predetermined intervals elapse, creating a continuous cycle of traffic light phases. The state diagram for normal operation mode is shown on Fig. 4.

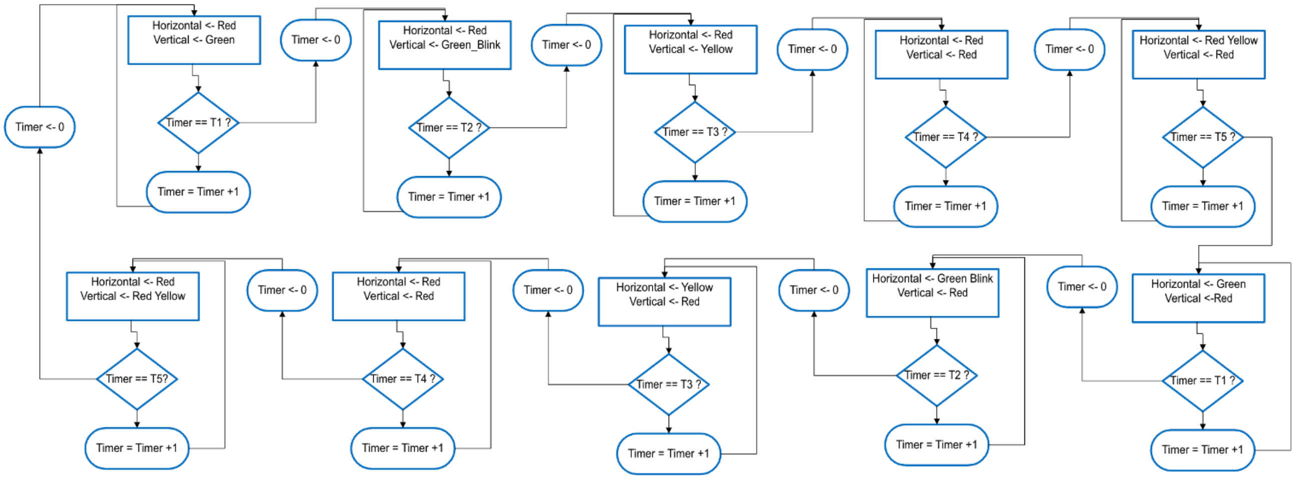
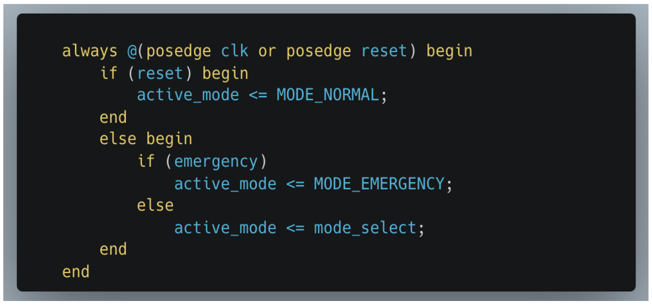


Figure 4: Advanced Traffic State Transitions

Operational Modes

The advanced controller supports three distinct operational modes that affect the timing and state transition behavior:



```
always @(posedge clk or posedge reset) begin
    if (reset) begin
        active_mode <= MODE_NORMAL;
    end
    else begin
        if (emergency)
            active_mode <= MODE_EMERGENCY;
        else
            active_mode <= mode_select;
    end
end
```

Figure 5: Operation Mode Control

1. **Normal Mode**: Standard operation with balanced timing for all directions. In this mode, both horizontal and vertical traffic receive equal green light duration, following a predefined sequence through all states.
2. **Increased Traffic Mode**: This adaptive mode modifies the timing allocation based on traffic sensor input. When congestion is detected on one road (indicated by the traffic sensor), the controller extends the green light duration for that direction by applying a multiplier to the standard timing. This feature helps maintain traffic flow on busy roads during peak hours.
3. **Emergency Mode**: When activated, the controller immediately transitions to the emergency state regardless of its current position in the cycle. In this state, both horizontal and vertical lights flash YELLOW, alerting drivers to yield to emergency vehicles. Once the emergency signal is cleared, the controller returns to normal operation starting from the first state.

A description of how these operation modes are coded into the system is shown on Fig. 5.

Timing and Adaptation

The controller implements a sophisticated timing system with predefined duration parameters for each phase of the traffic light sequence. These timing parameters control how long each state remains active before transitioning to the next state.

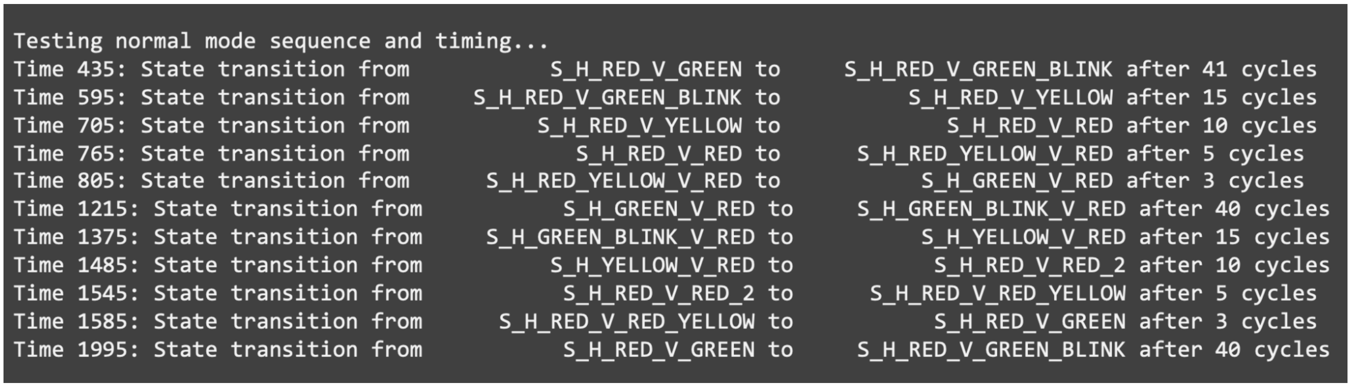
In increased traffic mode, the controller dynamically adjusts these timing parameters based on traffic sensor input. When the sensor detects congestion on the horizontal road, the green light duration for horizontal traffic is extended using a time multiplier. Similarly, when congestion is detected on the vertical road, the vertical green light duration is extended. This adaptive timing mechanism optimizes traffic flow based on real-time conditions.

IV. RESULTS

The functional verification of the advanced traffic light controller demonstrates correct operation across all three operational modes. The testbench results confirm that the controller properly implements the state transitions and timing requirements for each mode:

Normal Mode

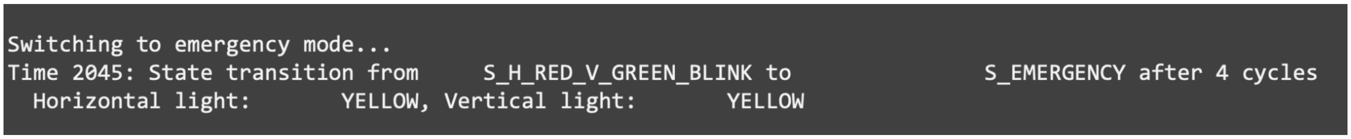
In normal mode, the controller follows the expected sequence through all states with appropriate timing. The test results show a complete cycle beginning with S_H_RED_V_GREEN and progressing through the entire sequence. Each state maintains its specified duration - for example, the green states last for 40 clock cycles, while transition states like yellow signals last for 10-15 cycles. The safety states with dual red signals last for 5 cycles, providing adequate clearance time.



```
Testing normal mode sequence and timing...
Time 435: State transition from           S_H_RED_V_GREEN to     S_H_RED_V_GREEN_BLINK after 41 cycles
Time 595: State transition from     S_H_RED_V_GREEN_BLINK to          S_H_RED_V_YELLOW after 15 cycles
Time 705: State transition from          S_H_RED_V_YELLOW to             S_H_RED_V_RED after 10 cycles
Time 765: State transition from             S_H_RED_V_RED to      S_H_RED_YELLOW_V_RED after 5 cycles
Time 805: State transition from      S_H_RED_YELLOW_V_RED to           S_H_GREEN_V_RED after 3 cycles
Time 1215: State transition from           S_H_GREEN_V_RED to     S_H_GREEN_BLINK_V_RED after 40 cycles
Time 1375: State transition from     S_H_GREEN_BLINK_V_RED to          S_H_YELLOW_V_RED after 15 cycles
Time 1485: State transition from          S_H_YELLOW_V_RED to           S_H_RED_V_RED_2 after 10 cycles
Time 1545: State transition from           S_H_RED_V_RED_2 to      S_H_RED_V_RED_YELLOW after 5 cycles
Time 1585: State transition from      S_H_RED_V_RED_YELLOW to           S_H_RED_V_GREEN after 3 cycles
Time 1995: State transition from           S_H_RED_V_GREEN to     S_H_RED_V_GREEN_BLINK after 40 cycles
```

Emergency Mode

When the emergency signal is activated, the controller immediately transitions to the S_EMERGENCY state regardless of its current position in the sequence. The test results show this transition occurring within 4 clock cycles of the emergency signal activation. In this state, both horizontal and vertical lights display flashing yellow signals, correctly implementing the emergency protocol to alert drivers.

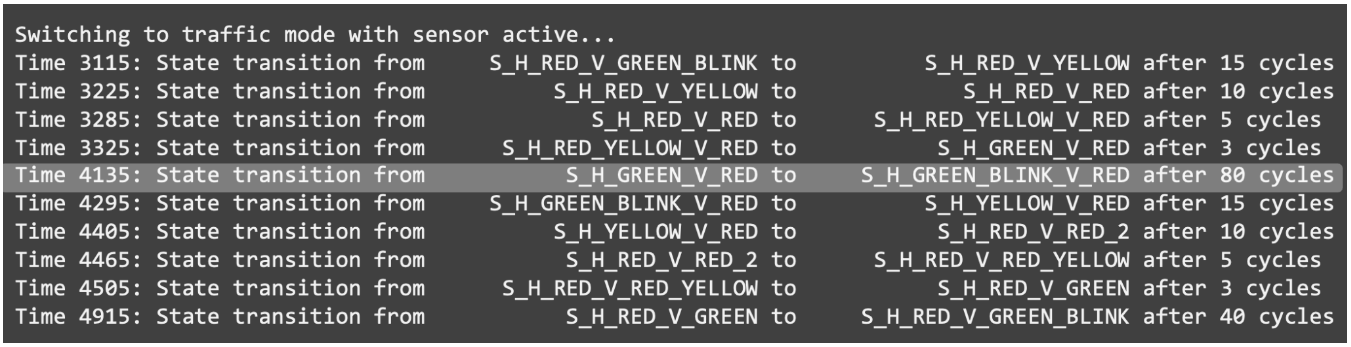


```
Switching to emergency mode...
Time 2045: State transition from     S_H_RED_V_GREEN_BLINK to               S_EMERGENCY after 4 cycles
  Horizontal light:       YELLOW, Vertical light:       YELLOW
```

Increased Traffic Mode

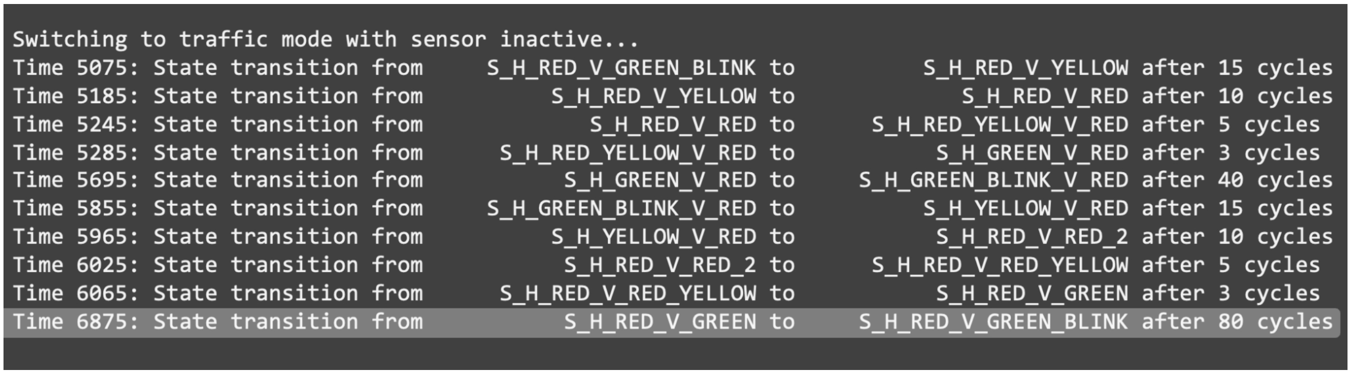
The traffic mode demonstrates adaptive timing based on sensor input:

1. With the traffic sensor active (indicating congestion on the horizontal road), the horizontal green phase is extended to 80 clock cycles, double the normal duration.
2. With the traffic sensor inactive, the vertical green phase is extended to 80 clock cycles while the



```
Switching to traffic mode with sensor active...
Time 3115: State transition from     S_H_RED_V_GREEN_BLINK to         S_H_RED_V_YELLOW after 15 cycles
Time 3225: State transition from          S_H_RED_V_YELLOW to            S_H_RED_V_RED after 10 cycles
Time 3285: State transition from             S_H_RED_V_RED to     S_H_RED_YELLOW_V_RED after 5 cycles
Time 3325: State transition from      S_H_RED_YELLOW_V_RED to          S_H_GREEN_V_RED after 3 cycles
Time 4135: State transition from           S_H_GREEN_V_RED to    S_H_GREEN_BLINK_V_RED after 80 cycles
Time 4295: State transition from     S_H_GREEN_BLINK_V_RED to         S_H_YELLOW_V_RED after 15 cycles
Time 4405: State transition from          S_H_YELLOW_V_RED to          S_H_RED_V_RED_2 after 10 cycles
Time 4465: State transition from           S_H_RED_V_RED_2 to     S_H_RED_V_RED_YELLOW after 5 cycles
Time 4505: State transition from      S_H_RED_V_RED_YELLOW to          S_H_RED_V_GREEN after 3 cycles
Time 4915: State transition from           S_H_RED_V_GREEN to    S_H_RED_V_GREEN_BLINK after 40 cycles
```

horizontal green phase maintains its standard 40-cycle duration.



```
Switching to traffic mode with sensor inactive...
Time 5075: State transition from     S_H_RED_V_GREEN_BLINK to         S_H_RED_V_YELLOW after 15 cycles
Time 5185: State transition from          S_H_RED_V_YELLOW to            S_H_RED_V_RED after 10 cycles
Time 5245: State transition from             S_H_RED_V_RED to     S_H_RED_YELLOW_V_RED after 5 cycles
Time 5285: State transition from      S_H_RED_YELLOW_V_RED to          S_H_GREEN_V_RED after 3 cycles
Time 5695: State transition from           S_H_GREEN_V_RED to    S_H_GREEN_BLINK_V_RED after 40 cycles
Time 5855: State transition from     S_H_GREEN_BLINK_V_RED to         S_H_YELLOW_V_RED after 15 cycles
Time 5965: State transition from          S_H_YELLOW_V_RED to          S_H_RED_V_RED_2 after 10 cycles
Time 6025: State transition from           S_H_RED_V_RED_2 to     S_H_RED_V_RED_YELLOW after 5 cycles
Time 6065: State transition from      S_H_RED_V_RED_YELLOW to          S_H_RED_V_GREEN after 3 cycles
Time 6875: State transition from           S_H_RED_V_GREEN to    S_H_RED_V_GREEN_BLINK after 80 cycles
```

This behavior confirms that the controller correctly implements the timing adaptation mechanism based on real-time traffic conditions.

Synthesis Results

The synthesis process utilized Synopsys Design Compiler with the SAED32nm_EDK library operating under specific timing constraints. A clock with a period of 10 ns was established, providing a 5 ns duty cycle. To model real-world conditions, a clock latency of 0.3 ns was applied, simulating the typical delay experienced in clock distribution networks. Input and output delays were carefully configured with values of 2.0 ns for all inputs and 1.7 ns for all outputs, representing realistic signal propagation times between the controller and external components.

These constraints ensure the design operates reliably within practical timing parameters. The synthesis was optimized for area efficiency, resulting in a compact implementation suitable for integration within larger traffic management systems. The resultant circuit is shown on Fig. 6, and the results are described below:

Timing Performance: The design achieves a positive slack of 7.68 ns in a 10 ns clock period

Area Utilization: The total area consumption is 424.064 μm²

Power Consumption: The design exhibits low power consumption of 13.5221 μW

These synthesis metrics confirm that the state-based design approach not only improves code readability and maintainability but also results in an efficient hardware implementation

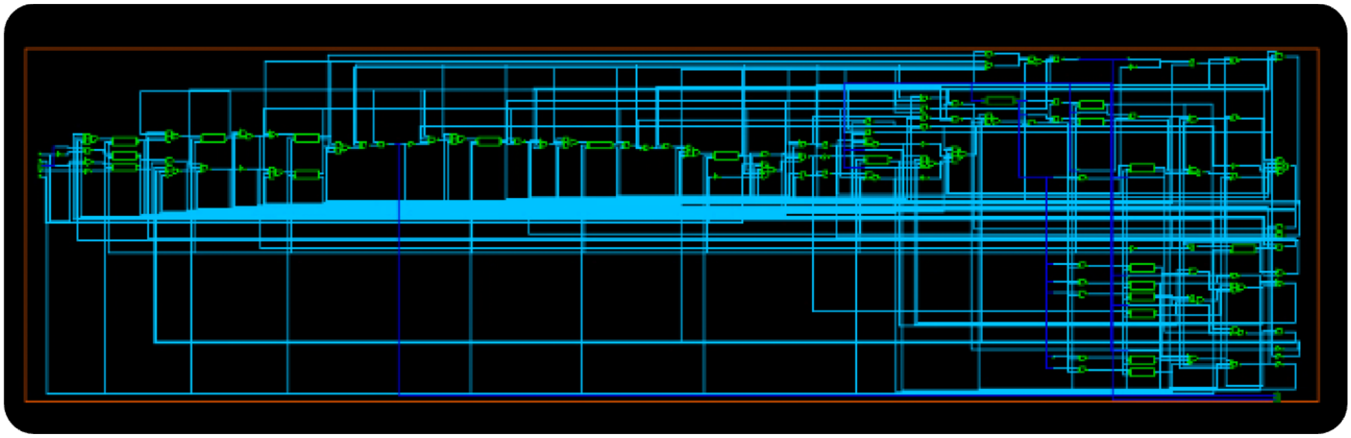


Figure 6: Optimized Circuit

V. DISCUSSION

Our state-based architecture provides clear separation between timing logic, state transitions, and output generation, enhancing maintainability while maintaining excellent hardware efficiency. The synthesis results confirm this with 7.68 ns timing slack, 424.064 µm² area utilization, and 13.5221 µW power consumption. The adaptive timing mechanism effectively addresses varying traffic conditions, doubling green light duration from 40 to 80 clock cycles when congestion is detected. The emergency mode transitions to flashing yellow within 4 clock cycles, ensuring rapid response for emergency vehicles.

Despite these improvements, challenges remain. The current traffic detection uses binary sensor input, providing limited information about actual congestion levels. More sophisticated detection methods could enable proportional timing adjustments rather than fixed multipliers.

VI. CONCLUSION

This paper presented two enhanced traffic light controllers utilizing state-based Verilog HDL implementation. Our key contributions include a maintainable state-based design approach, adaptive timing mechanisms responsive to traffic conditions, emergency vehicle prioritization, and efficient hardware implementation.

Future work could integrate machine learning to optimize timing based on historical traffic patterns, coordinate multiple intersections for "green wave" traffic flow, and incorporate pedestrian detection for responsive crossing times.

VII. REFERENCES

[1] Zhilevski et al. "Design of a Traffic Light System Using Verilog HDL." 2023 5th International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA). IEEE, 2023.

[2] Borges, Dimitrius F., et al. "Traffic light control using hierarchical reinforcement learning and options framework." IEEE Access 9 (2021): 99155-99165.