OPEN ACCESS

Saudi Journal of Engineering and Technology
Abbreviated Key Title: Saudi J Eng Technol
ISSN 2415-6272 (Print) |ISSN 2415-6264 (Online)
Scholars Middle East Publishers, Dubai, United Arab Emirates
Journal homepage: https://saudijournals.com

Original Research Article

Smart Traffic Light Controller using Vivado

Kumbhar Trupti Ravikumar[1], Saket Thapa[2*], Rishabh Pratiush[2]

[1]Asst. Prof., Electronics and Communication, Acharya Institute of Technology, Bengaluru, India
[2]Electronics and Communication, Acharya Institute of Technology, Bengaluru, India

DOI: https://doi.org/10.36348/sjet.2024.v09i12.004 | **Received:** 18.10.2024 | **Accepted:** 06.12.2024 | **Published:** 16.12.2024

***Corresponding author:** Saket Thapa
Electronics and Communication, Acharya Institute of Technology, Bengaluru, India

Abstract

This paper presents an adaptive traffic light control system designed to enhance traffic flow efficiency using real-time vehicle count data. Implemented in Verilog HDL, the system dynamically adjusts signal timings based on vehicle density in each lane, prioritizing heavily congested lanes to reduce delays and improve throughput. The design employs logical circuit modeling and synthesis for optimal performance. By minimizing congestion and accommodating varying traffic conditions, the proposed system demonstrates significant potential for real-world applications in urban traffic management and intelligent transportation systems.

Keywords: Verilog, Vivado, IP integrator, IP Block Design.

I. INTRODUCTION

The adaptive traffic light control system presented in this design leverages the IP integrator feature of Xilinx Vivado to create a modular and scalable block design. This implementation uses custom Verilog modules for traffic light control logic and integrates them into a block design, enabling seamless interconnection with other hardware IPs such as timers, GPIOs, or processors [1].

The system dynamically prioritizes traffic lanes based on real- time vehicle count inputs received from external sensors or emulated signals. The traffic_light_control module processes the inputs and updates the traffic signals for each lane (North, South, East, West) using a state-based approach. This block can be interfaced with external hardware like cameras or sensors for vehicle detection and can be extended with additional features like ambulance detection.

By utilizing Vivado's IP integrator, the design ensures efficient hardware resource utilization, easy integration, and compatibility with FPGA-based implementations. This approach simplifies prototyping and testing on platforms such as Zynq SoCs, where hardware acceleration can further enhance performance [2].

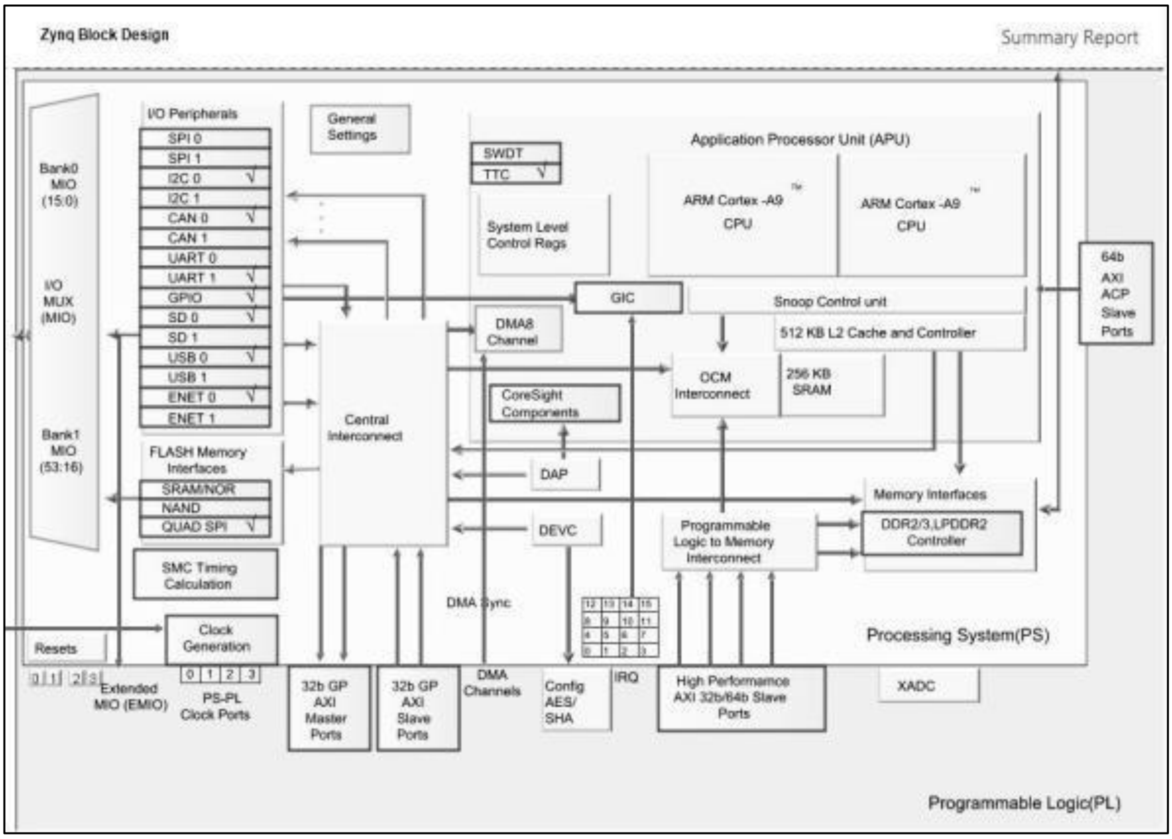


Figure 1: Zynq Processing Unit

II. LITERATURE REVIEWS

The field of traffic control plays an important part in our life so many papers and researches are published to solve the traffic problems. Some of these papers are explained below:

Marin Zhilevski, Mikho Mikhov, Madlena Zhilevska, the paper presents a design methodology for a traffic light control system using Verilog HDL. The study is divided into two sections: theoretical analysis of operational modes and practical aspects such as synthesis, simulation, and implementation using Xilinx's ISE Design Suite. Three control modes are explored: **normal traffic**, **increased traffic**, and **emergency traffic**. A Verilog-based control program is synthesized and simulated.

The findings can inform the design and deployment of adaptive traffic light systems in various urban traffic scenarios.[1] Pranaav Jothi M, Srivatsan M, Krishna Kumar P, Prakash P; Kasthuri P, the study introduces a fully automated and efficient traffic light control system for a four-way intersection based on a Moore machine. Implemented on an Artix-7 xc7a100tcsg324-1 FPGA, the system utilizes Xilinx Vivado software and Verilog HDL for development. It achieves a maximum operating frequency of 10 MHz, demonstrating the capability to manage traffic effectively. This work highlights the advantages of FPGA-based systems for real-time traffic control due to their speed, reliability, and scalability in complex intersection scenarios.[2]

Liang Zhichao, Gu Wen, Yang Yuhang, Wang Ying, this article explores the implementation of an intelligent traffic light system aimed at optimizing urban traffic flow and alleviating congestion. The system employs real-time monitoring of road conditions and traffic flow to dynamically control signal light states. It features a logical circuit design scheme implemented on FPGA. Key components include an introduction to system functions and characteristics, design principles, and implementation methodologies. Experimental results confirm the system's feasibility and effectiveness, showcasing its potential for practical application in urban traffic management.

III. METHODOLOGY

Traditional traffic control systems rely on fixed-timing mechanisms, which are often inefficient under varying traffic conditions. This adaptive approach leverages real-time vehicle count data from four lanes—North, South, East, and West—to dynamically control traffic signals. The system is modeled and simulated using Verilog HDL, ensuring compatibility for hardware deployment.

A. Design methodology

The traffic light state for each lane is encoded using a 3-bit representation, where 3'b001 indicates Red, 3'b010 indicates Yellow, and 3'b100 indicates Green.

The system employs adaptive logic to dynamically compare vehicle densities across all lanes at every clock cycle, assigning the Green signal to the lane with the highest vehicle count while defaulting all other lanes to Red. To ensure fairness, the algorithm resolves ties in vehicle counts using predetermined rules or a round-robin arbitration approach, which can be extended for enhanced functionality. Additionally, the system incorporates a reset functionality, initializing all traffic lights to Red upon system reset or manual initialization.

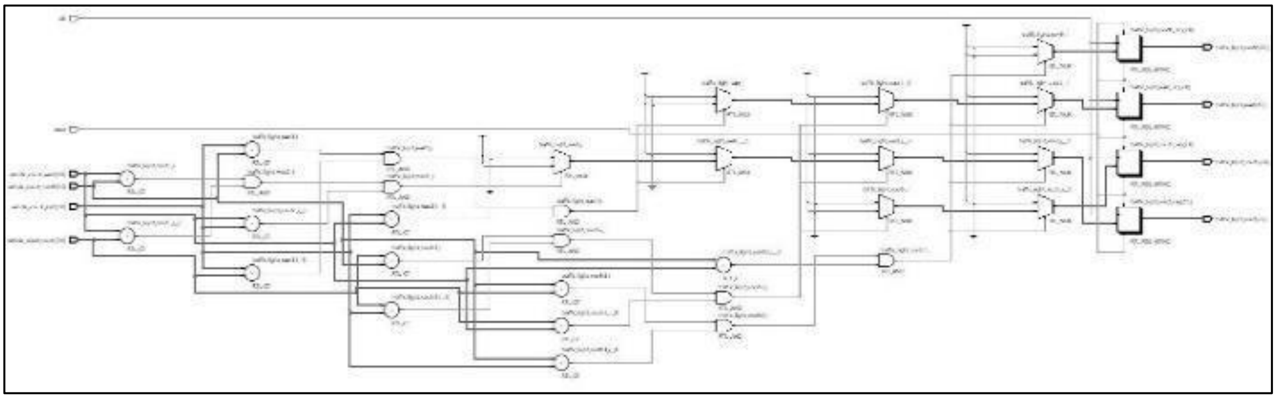


Figure 2: RTL design

B. Implementation and Synthesis

The system is implemented using Verilog HDL and synthesized with industry-standard Electronic Design Automation (EDA) tools, such as Xilinx Vivado and Cadence. This design is fully compatible with FPGA deployment, ensuring real-time responsiveness and efficient operation in hardware.

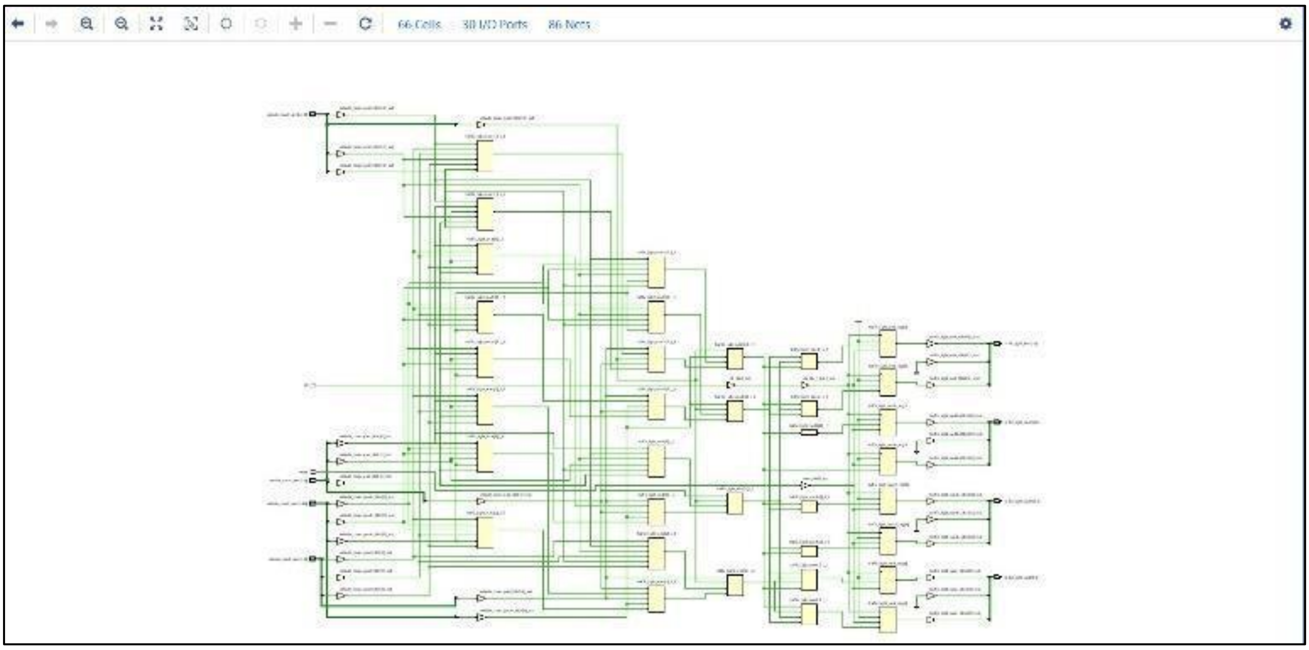


Figure 3: Synthesized Internal design

C. Simulation Methodology

The system operates with a 100 MHz clock signal (clk), corresponding to a 10 ns period, and utilizes a reset signal (reset) to initialize all traffic lights to their default state. Randomized traffic input is generated using $urandom_range (0, 15) to emulate real-world variability in vehicle counts for all four lanes. A $monitor statement is employed to log the simulation's time, vehicle counts, and traffic light states, offering valuable insights into system performance and behavior under different conditions.

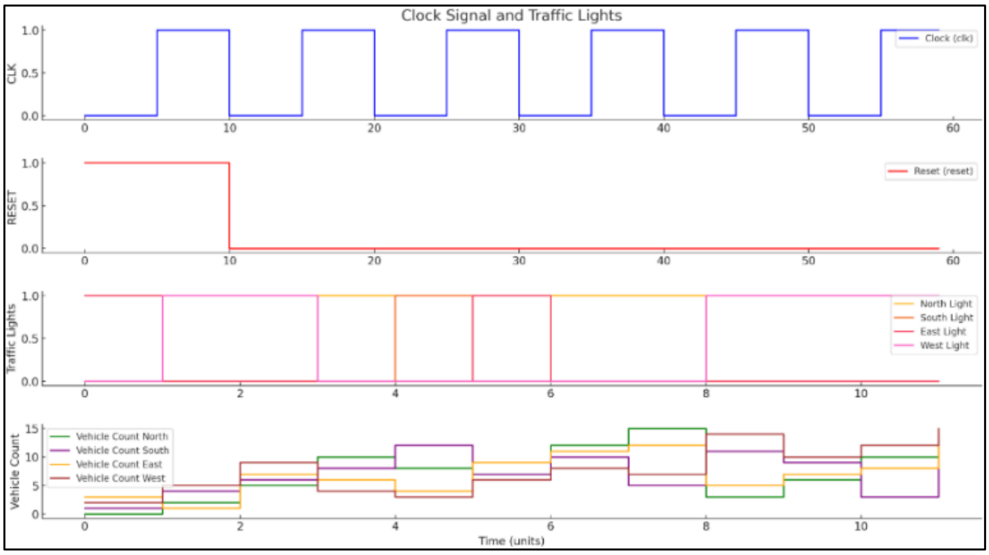
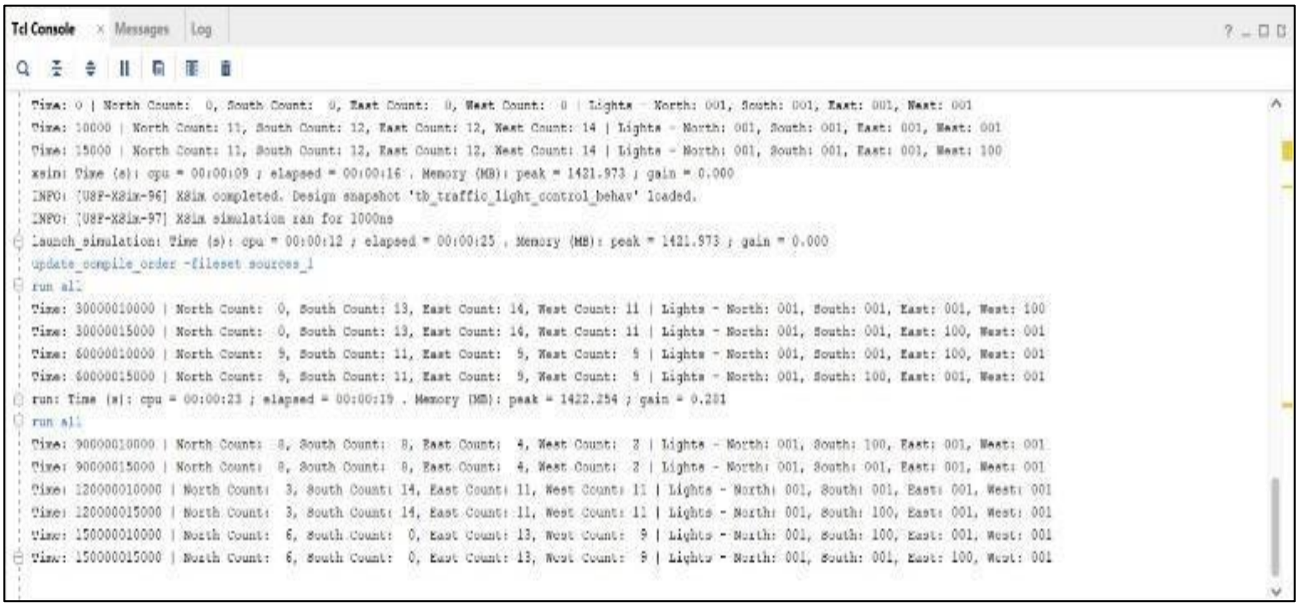


Figure 4: Waveform

D. Testing and Validation

The design is verified through functional testing using the testbench module, ensuring correct traffic light behaviour across various vehicle count scenarios. Performance metrics such as latency—the time required to switch traffic lights in response to changes in vehicle density—and fairness—the proper allocation of green lights across lanes without any lane being starved—are evaluated. Additionally, the design is modular, allowing for easy scalability to incorporate additional lanes or integrate emergency vehicle detection capabilities in the future.



```
Time: 0 | North Count:  0, South Count:  0, East Count:  0, West Count:  0 | Lights - North: 001, South: 001, East: 001, West: 001
Time: 10000 | North Count: 11, South Count: 12, East Count: 12, West Count: 14 | Lights - North: 001, South: 001, East: 001, West: 001
Time: 15000 | North Count: 11, South Count: 12, East Count: 12, West Count: 14 | Lights - North: 001, South: 001, East: 001, West: 100
xsim: Time (s): cpu = 00:00:09 ; elapsed = 00:00:16 . Memory (MB): peak = 1421.973 ; gain = 0.000
INFO: [USF-XSim-96] XSim completed. Design snapshot 'tb_traffic_light_control_behav' loaded.
INFO: [USF-XSim-97] XSim simulation ran for 1000ns
launch_simulation: Time (s): cpu = 00:00:12 ; elapsed = 00:00:25 . Memory (MB): peak = 1421.973 ; gain = 0.000
update_compile_order -fileset sources_1
run all
Time: 30000010000 | North Count:  0, South Count: 13, East Count: 14, West Count: 11 | Lights - North: 001, South: 001, East: 001, West: 100
Time: 30000015000 | North Count:  0, South Count: 13, East Count: 14, West Count: 11 | Lights - North: 001, South: 001, East: 100, West: 001
Time: 60000010000 | North Count:  9, South Count: 11, East Count:  5, West Count:  5 | Lights - North: 001, South: 001, East: 100, West: 001
Time: 60000015000 | North Count:  9, South Count: 11, East Count:  5, West Count:  5 | Lights - North: 001, South: 100, East: 001, West: 001
run: Time (s): cpu = 00:00:23 ; elapsed = 00:00:19 . Memory (MB): peak = 1422.254 ; gain = 0.281
run all
Time: 90000010000 | North Count:  8, South Count:  8, East Count:  4, West Count:  2 | Lights - North: 001, South: 100, East: 001, West: 001
Time: 90000015000 | North Count:  8, South Count:  8, East Count:  4, West Count:  2 | Lights - North: 001, South: 001, East: 001, West: 001
Time: 120000010000 | North Count:  3, South Count: 14, East Count: 11, West Count: 11 | Lights - North: 001, South: 001, East: 001, West: 001
Time: 120000015000 | North Count:  3, South Count: 14, East Count: 11, West Count: 11 | Lights - North: 001, South: 100, East: 001, West: 001
Time: 150000010000 | North Count:  6, South Count:  0, East Count: 13, West Count:  9 | Lights - North: 001, South: 100, East: 001, West: 001
Time: 150000015000 | North Count:  6, South Count:  0, East Count: 13, West Count:  9 | Lights - North: 001, South: 001, East: 100, West: 001
```

Figure 5: TCL Console

IV. DESIGN APPROACH

A. Functional block

The Traffic Light Control IP block is a priority-based dynamic traffic light controller. The block dynamically adjusts traffic light states based on the vehicle count in each lane. The system ensures optimized traffic flow by prioritizing the lane with the highest vehicle count.

B. Block Diagram

The IP block for the Traffic Light Control System consists of five sub-modules: the Input Interface, which captures vehicle counts from each lane; the Control Logic, which implements a priority-based decision-making algorithm to determine the traffic flow; the State Machine, which manages transitions between traffic light states; the Output Interface, which generates the traffic light signals for all four lanes; and the Clock & Reset Control, which ensures synchronized operation and initializes the system for proper functioning.

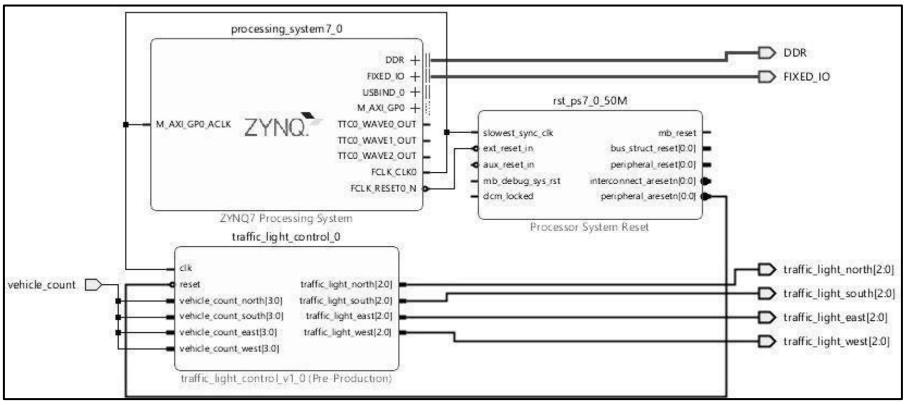


Figure 6: IP Block design

C. Interface Definitions

Input Ports

Signal	Width	Description
clk	1	Clock signal for synchronization.
reset	1	Asynchronous reset for initialization.
Vehicle_count	4	Vehicle count for North, South, East, West.

Output Ports

Signal	Width	Description
traffic_light	3	Traffic light signals for North, South, East, and West in {Red, Yellow, Green} format.

D. IP Configuration

The IP block is designed to support parameterization, allowing flexibility to configure the number of lanes (default is 4, but it can be extended), the clock frequency (e.g., 50 MHz or 100 MHz), and the traffic light timings with adjustable durations for green, yellow, and red lights to suit various system requirements.

E. Design Implementation

The design implementation includes a State Machine with states {RESET, RED, YELLOW, GREEN} where transitions are determined by vehicle counts and timing constraints. A Priority Algorithm is incorporated to compare vehicle counts using combinational logic and resolve ties based on a predefined priority order (e.g., North > South > East > West).

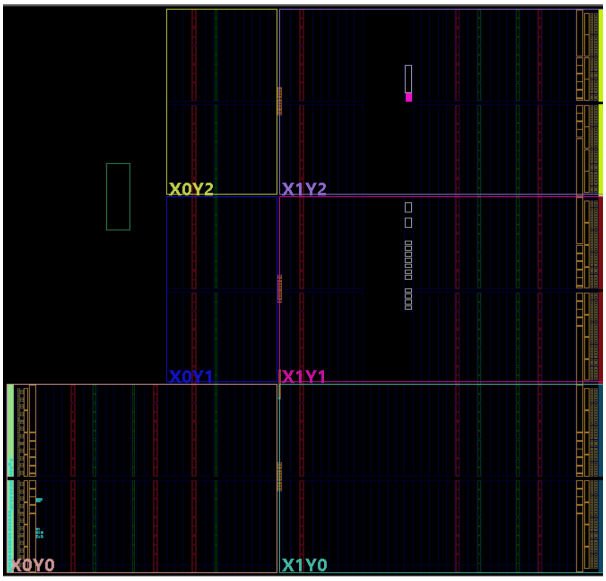


Figure 7: Synthesized design

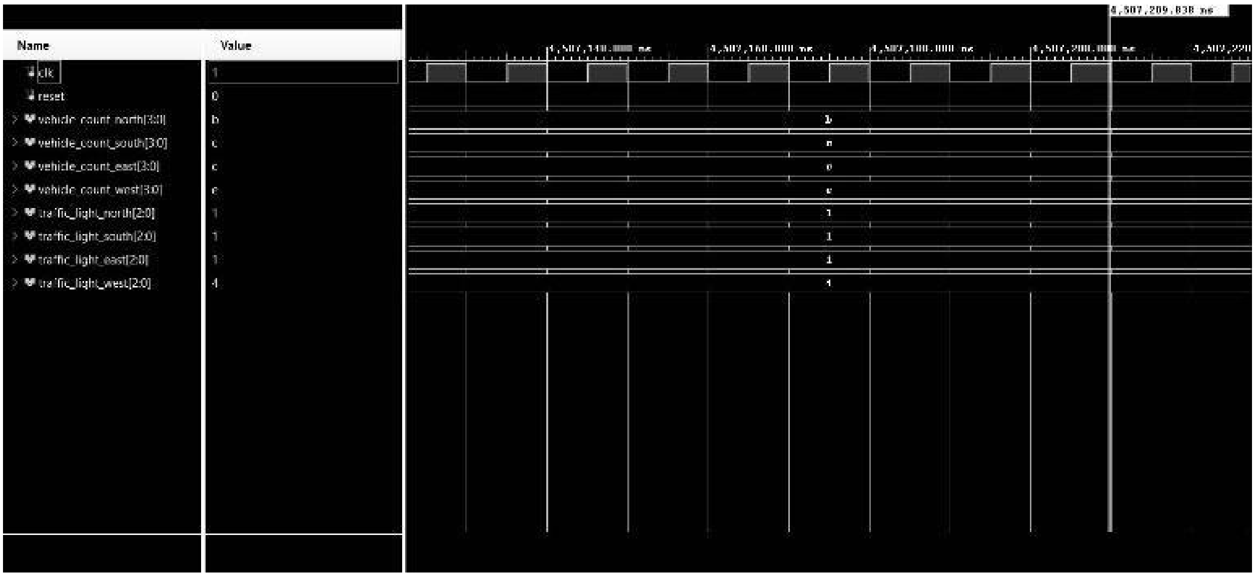
F. Verification Methodology

The verification methodology involves designing a testbench that uses constrained random stimulus and functional coverage to validate key scenarios, including reset behavior, tie-breaking logic for equal vehicle counts, and dynamic traffic light transitions under random traffic conditions. Waveform analysis is performed using industry-standard tools like Xilinx Vivado.

G. Reusability and Scalability

The design ensures reusability and scalability through parameterization for different configurations, standardized interfaces like AMBA or AXI4-Lite for seamless integration into larger SoCs, and packaging the module as reusable IP using tools like Xilinx Vivado IP Integrator.

V. SIMULATION RESULTS

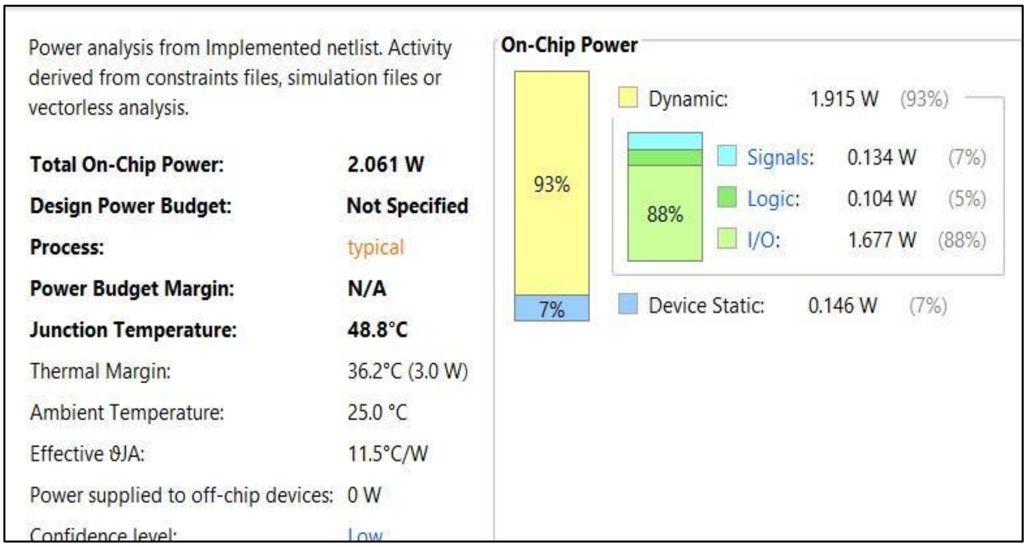


Name	Value
clk	1
reset	0
vehicle_count_north[3:0]	b
vehicle_count_south[3:0]	c
vehicle_count_east[3:0]	c
vehicle_count_west[3:0]	c
traffic_light_north[2:0]	1
traffic_light_south[2:0]	1
traffic_light_east[2:0]	1
traffic_light_west[2:0]	4

Figure 8: Simulation Result

The waveform above represents the behavior of the traffic light controller module:

- **Clock Signal (clk):** A square wave toggling every 5 time units, providing the timing reference for the module.
- **Reset Signal (reset):** High for the first 10 time units, initializing the system.
- **Traffic Light Outputs:** Representing which lane's light is green, based on vehicle counts:
- **North Light:** Green when the north vehicle count is highest. South Light: Green when the south vehicle count is highest.
- **East Light:** Green when the east vehicle count is highest.
- **West Light:** Green when the west vehicle count is highest. Vehicle Counts: Simulated inputs for vehicle density in each lane, varying over time.



Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

Total On-Chip Power:	**2.061 W**
Design Power Budget:	**Not Specified**
Process:	typical
Power Budget Margin:	**N/A**
Junction Temperature:	**48.8°C**
Thermal Margin:	36.2°C (3.0 W)
Ambient Temperature:	25.0 °C
Effective ϑJA:	11.5°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low

On-Chip Power

Dynamic:	1.915 W	(93%)
Signals:	0.134 W	(7%)
Logic:	0.104 W	(5%)
I/O:	1.677 W	(88%)
Device Static:	0.146 W	(7%)

Figure 9: Power Consumption

VI. APPLICATION

1) Optimizes traffic light timings based on real-time vehicle density, reducing congestion and improving flow at intersections.
2) Integrates into smart city systems, utilizing IoT sensors to dynamically adjust traffic lights.
3) Can be adapted to detect emergency vehicles, clearing their path for quicker response times.
4) Reduces idle times, lowering fuel consumption and emissions, supporting sustainable energy goals.

VII. CONCLUSION

The traffic light control system dynamically adjusts signals based on vehicle counts at four intersections, prioritizing lanes with higher traffic for improved flow. Using clock-driven mechanisms, it ensures continuous operation and adaptability. The design can be further enhanced with features like emergency vehicle detection and inter-light communication. Simulation results confirm its effectiveness in optimizing traffic flow. This model provides a foundation for real-time, adaptive traffic management systems using SOC design.

REFERENCE

1. https://ieeexplore.ieee.org/document/10156804
2. https://ieeexplore.ieee.org/document/9865755
3. https://ieeexplore.ieee.org/document/10263447