EVOLUTION IN ELECTRICAL AND ELECTRONIC ENGINEERING
e-ISSN: 2756-8458

Vol. 6 No. 2 (2025) 90-97
https://publisher.uthm.edu.my/periodicals/index.php/eeee

EEEE

FPGA-Based Traffic Density Monitoring System

Valentine Teo Jin Lung[1], Jais Lias[1]*

[1] *Faculty of Electrical and Electronic Engineering*
Universiti Tun Hussein Onn Malaysia, Batu Pahat, Johor, 86400 MALAYSIA

*Corresponding Author: jaisl@uthm.edu.my
DOI: https://doi.org/10.30880/eeee.2025.06.02.011

Article Info

Received: 10 July 2025
Accepted: 23 September 2025
Available online: 30 October 2025

Keywords

FPGA, Traffic Light Control, Vehicle Detection, YOLOv8, Verilog, Urban Traffic Management

Abstract

The FPGA-based Traffic Density Monitoring System is designed to provide a real-time solution for monitoring and analyzing traffic conditions. This project aims to address the inefficiencies in traditional traffic management systems by utilizing the processing capabilities of Field-Programmable Gate Arrays (FPGAs). The primary objective is to enhance the efficiency of traffic flow by providing accurate and timely data on vehicle density. The methodology involves capturing live traffic data using advanced image sensors, which are then processed using FPGA hardware for real-time traffic density estimation. The system employs sophisticated image processing algorithms to detect and count vehicles within the monitored area, generating outputs that can assist traffic authorities in optimizing signal timings and mitigating congestion. Verilog programming and hardware optimization techniques are extensively utilized to ensure high-speed processing and accuracy in the system's operations. The results demonstrate that the YOLOv8-based vehicle detection consistently achieved an accuracy of 95–98% under normal lighting conditions, while the FPGA finite state machine (FSM) was successfully validated in both simulation and hardware testing. The system maintained real-time responsiveness, with average end-to-end processing latency below 100 milliseconds even under high traffic loads. The system's ability to operate in real-time makes it a viable and scalable solution for dynamic traffic management scenarios. Moreover, its cost-effective implementation offers the potential for widespread adoption in urban settings. In conclusion, this project highlights the transformative potential of FPGA technology in revolutionizing traffic monitoring systems. By offering a scalable, efficient, and cost-effective solution, it addresses modern urban challenges, paving the way for smarter and more sustainable cities.

1. Introduction

Urban traffic congestion is a persistent issue in many developing countries, particularly in rapidly growing cities such as those in Malaysia. Conventional traffic signal systems operate on fixed timing cycles, regardless of real-time traffic conditions, leading to inefficiencies, increased travel time, fuel consumption, and environmental pollution [1-5]. To address this challenge, this study proposes an intelligent traffic light control system using a Field Programmable Gate Array (FPGA) [6-8]. The system incorporates real-time vehicle detection and adaptive signal control to optimize traffic flow dynamically [9-11]. By integrating an AI-based object detection model with FPGA hardware, the proposed system aims to demonstrate the feasibility of responsive, low-latency traffic management for smart city applications [12-14].

2. Methodology

The proposed system comprises two main modules: a vehicle detection module running on a PC and a traffic signal control module implemented on an FPGA. The system begins with a webcam capturing live video from a four-way intersection. This video stream is processed in real-time using the YOLOv8 object detection algorithm, developed in Python, to identify and count vehicles in each traffic lane. Based on the detection results, the system determines the traffic density for each direction. This information is transmitted from the PC to the FPGA via UART serial communication. The FPGA, specifically the DE2-115 development board with a Cyclone IV E device, is programmed using Verilog HDL to execute a finite state machine (FSM) that dynamically adjusts the green light duration depending on vehicle density. Yellow and red light durations are fixed at 3 and 2 seconds, respectively. The system also includes 7-segment displays for countdown timing and a manual reset button for system reinitialization.

The overall hardware-software interaction is illustrated in the system's block diagram depicted in Fig. 1. It shows how the video feed from the webcam flows into the PC for processing, how the vehicle density data is communicated to the FPGA via UART, and how the FPGA processes that information to control the output peripherals. These outputs include four sets of LED-based traffic light modules and two 7-segment displays to indicate countdown timing for each traffic phase.

The procedural logic of the system is presented in the flowchart shown in Fig. 2. The process starts with webcam initialization and live video capture, followed by real-time vehicle detection using the YOLOv8 model. After counting vehicles per lane, the system sends the data to the FPGA. The FSM within the FPGA interprets the traffic density data and adjusts the green light duration accordingly. It then updates the traffic lights and 7-segment display based on the current phase. This process loops continuously to ensure real-time responsiveness to changing traffic conditions.

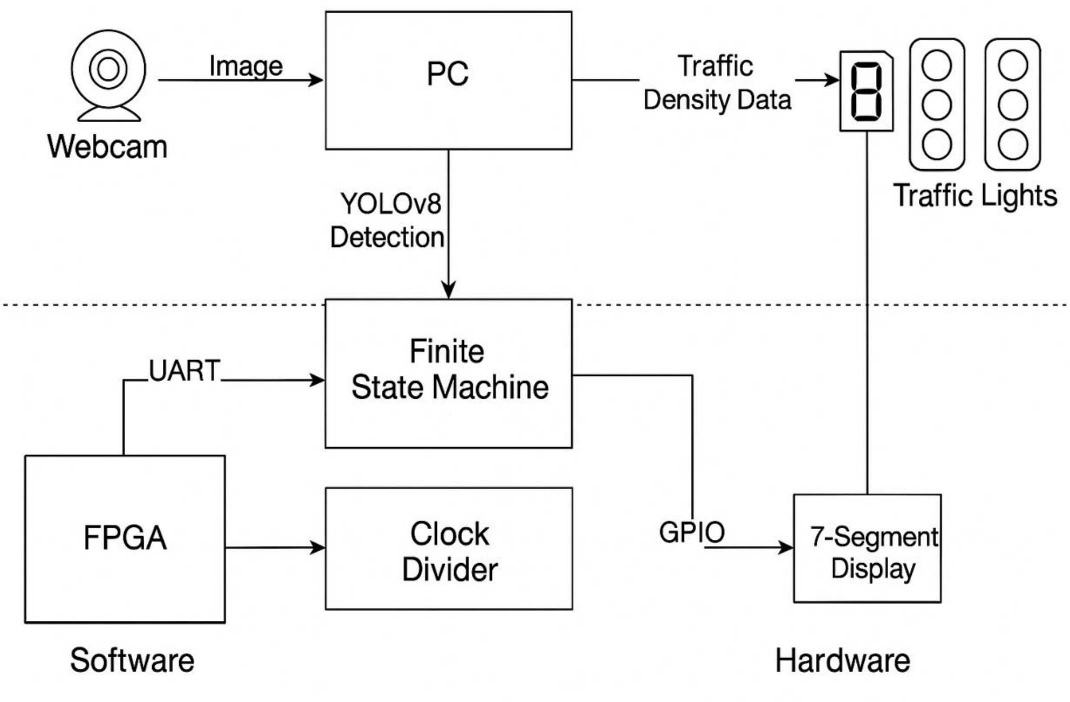


Fig. 1 *Block diagram of FPGA-based traffic density monitoring process*

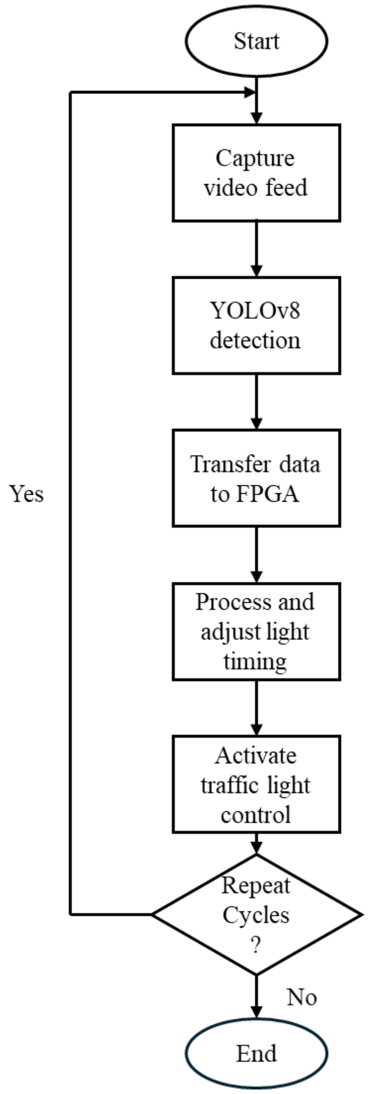


Fig. 2 *Flowchart of FPGA-based traffic density monitoring process*

2.1 Prototype Setup

The prototype implementation was conducted on a DE2-115 FPGA development board (Cyclone IV E – EP4CE115F29C7) to validate real-world traffic light control behavior using live input from an AI-based vehicle detection system. Four physical traffic light modules were connected to GPIO JP5 pins 1–12, with each module comprising red, yellow, and green LEDs representing the North, South, East, and West lanes. Two 7-segment displays (HEX0 and HEX1) were connected to indicate the countdown timer during each green and yellow phase.

A webcam was mounted above a scaled-down road intersection mockup using toy vehicles. This camera feed was processed by a Python script running the YOLOv8 object detection model on a host PC equipped with an Intel Core i7 CPU and RTX 3050 GPU. The script detected vehicles in real time and classified traffic density levels as low (00), medium (01), or high (10 or 11). These density values were then transmitted to the FPGA via UART serial communication.

Upon receiving the traffic density inputs, the Verilog-coded finite state machine (FSM) executed adaptive signal control logic. The FSM dynamically adjusted the green light durations—shorter for low traffic, moderate for medium traffic, and extended for high-density conditions—while yellow and red intervals remained fixed at 3 and 2 seconds, respectively. Multiple traffic scenarios were executed to verify system behavior. Transitions of the physical LEDs were synchronized with the FSM states, and the 7-segment countdown display accurately reflected the remaining time for each phase. Debug LEDs on the board confirmed UART signal reception and FSM responsiveness, while Quartus II's SignalTap and ModelSim simulations further validated state transitions and timing logic.

Fig. 3 illustrates the complete hardware setup of the prototype, showing the interconnection between the webcam, PC-based YOLOv8 detection, UART communication, and the FPGA traffic controller. Fig. 4 presents the schematic diagram, emphasizing modularity, data flow, and real-time integration of AI with digital hardware logic.

Evolution in Electrical and Electronic Engineering Vol. 6 No. 2 (2025) p. 90-97

This practical setup demonstrates the feasibility of merging real-time computer vision with reconfigurable logic design to build an intelligent traffic density monitoring system.

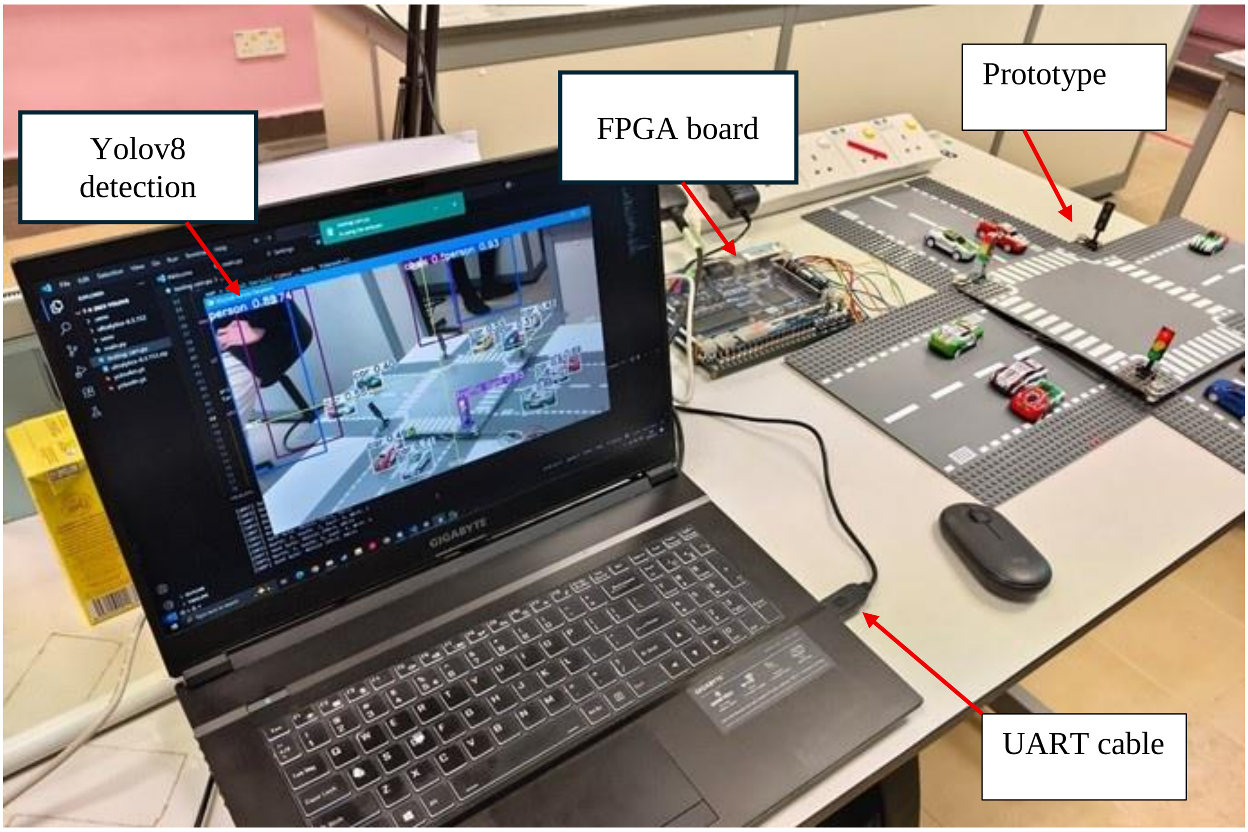
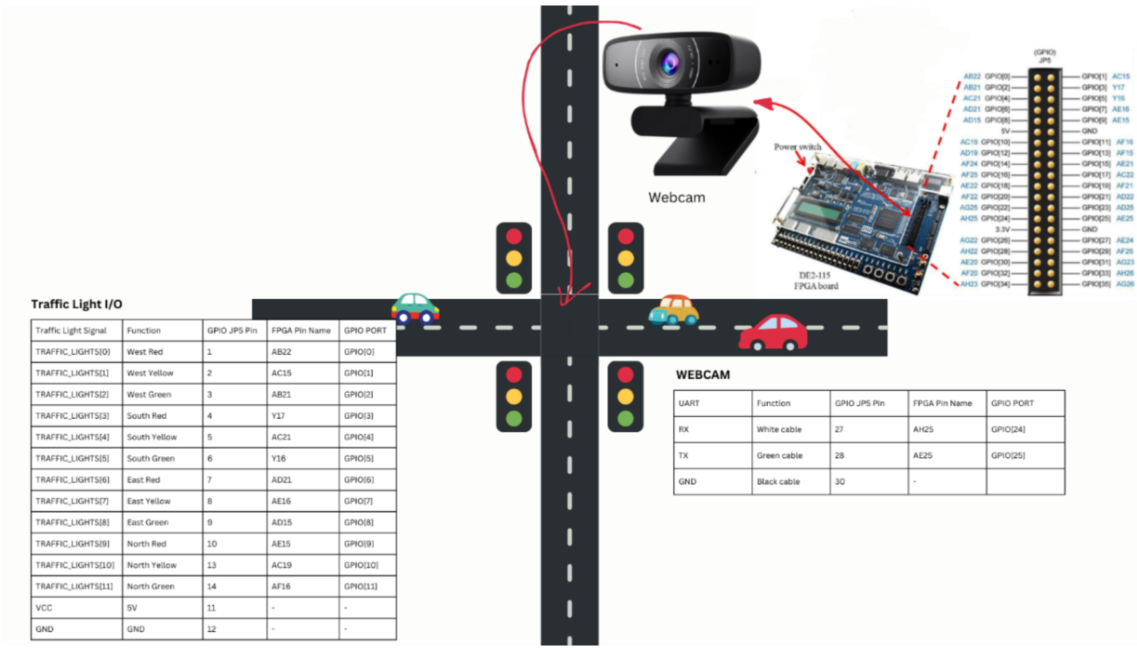


Fig. 3 *FPGA DE2-115's hardware simulation*



Traffic Light I/O

Traffic Light Signal	Function	GPIO JP5 Pin	FPGA Pin Name	GPIO PORT
TRAFFIC_LIGHTS[0]	West Red	1	AB22	GPIO[0]
TRAFFIC_LIGHTS[1]	West Yellow	2	AC15	GPIO[1]
TRAFFIC_LIGHTS[2]	West Green	3	AB21	GPIO[2]
TRAFFIC_LIGHTS[3]	South Red	4	Y17	GPIO[3]
TRAFFIC_LIGHTS[4]	South Yellow	5	AC21	GPIO[4]
TRAFFIC_LIGHTS[5]	South Green	6	Y16	GPIO[5]
TRAFFIC_LIGHTS[6]	East Red	7	AD21	GPIO[6]
TRAFFIC_LIGHTS[7]	East Yellow	8	AE16	GPIO[7]
TRAFFIC_LIGHTS[8]	East Green	9	AD15	GPIO[8]
TRAFFIC_LIGHTS[9]	North Red	10	AE15	GPIO[9]
TRAFFIC_LIGHTS[10]	North Yellow	13	AC19	GPIO[10]
TRAFFIC_LIGHTS[11]	North Green	14	AF16	GPIO[11]
VCC	5V	11	-	-
GND	GND	12	-	-

WEBCAM

UART	Function	GPIO JP5 Pin	FPGA Pin Name	GPIO PORT
RX	White cable	27	AH25	GPIO[24]
TX	Green cable	28	AE25	GPIO[25]
GND	Black cable	30	-	

Fig. 4 *Schematic diagram of the prototype*

3. Results and Discussion

This section presents the prototype implementation and experimental results of the FPGA-based adaptive traffic light control system. Both simulation and physical validation were performed to assess the system's performance in real-time vehicle detection, signal responsiveness, and timing accuracy. Emphasis was placed on validating the FSM logic, UART communication, and YOLOv8 detection performance under varying conditions.

3.1 Result

The performance of the proposed system was evaluated through both hardware simulation and physical testing, focusing on FSM behavior, UART communication integrity, detection accuracy, and processing time under real-time traffic scenarios.

3.1.1 FSM and UART Simulation

The adaptive traffic signal logic, implemented via a finite state machine (FSM), was validated through waveform simulation using ModelSim-Altera 10.1d. The simulation involved the complete Verilog-based top-level module (top_module) comprising clock divider, FSM, UART, and 7-segment display modules. A 50 MHz system clock was simulated using a 20 ns period (#10 delay), and reset (KEY0) was asserted low for 100 ns before release.

Simulation results confirmed correct FSM transitions across the defined states: S0 (NS Green) → S1 → S2 → S3 → S4 → S5 → S0, with state durations dynamically adjusted according to the received ns density values. For example, with ns = 2'b10, the green phase was held for approximately 12 seconds before transitioning. The traffic_output[11:0] signal reflected correct LED patterns for four traffic directions, and the countdown timers displayed on HEX0 and HEX1 matched the FSM's internal timers. UART communication was validated using testbenches, confirming accurate decoding of 2-bit density input and proper influence on FSM duration. Debug LEDs (LEDG0, LEDG1) were toggled in sync with the divided clock and FSM states, further validating synchronization and functionality.

Quartus II's SignalTap Logic Analyzer was used during hardware validation to monitor real-time FSM behavior and UART reception. No metastability, glitches, or false transitions were observed, confirming the system's stable response under asynchronous UART input. Fig. 5 shows the Simulated FSM transition waveform in Quartus II

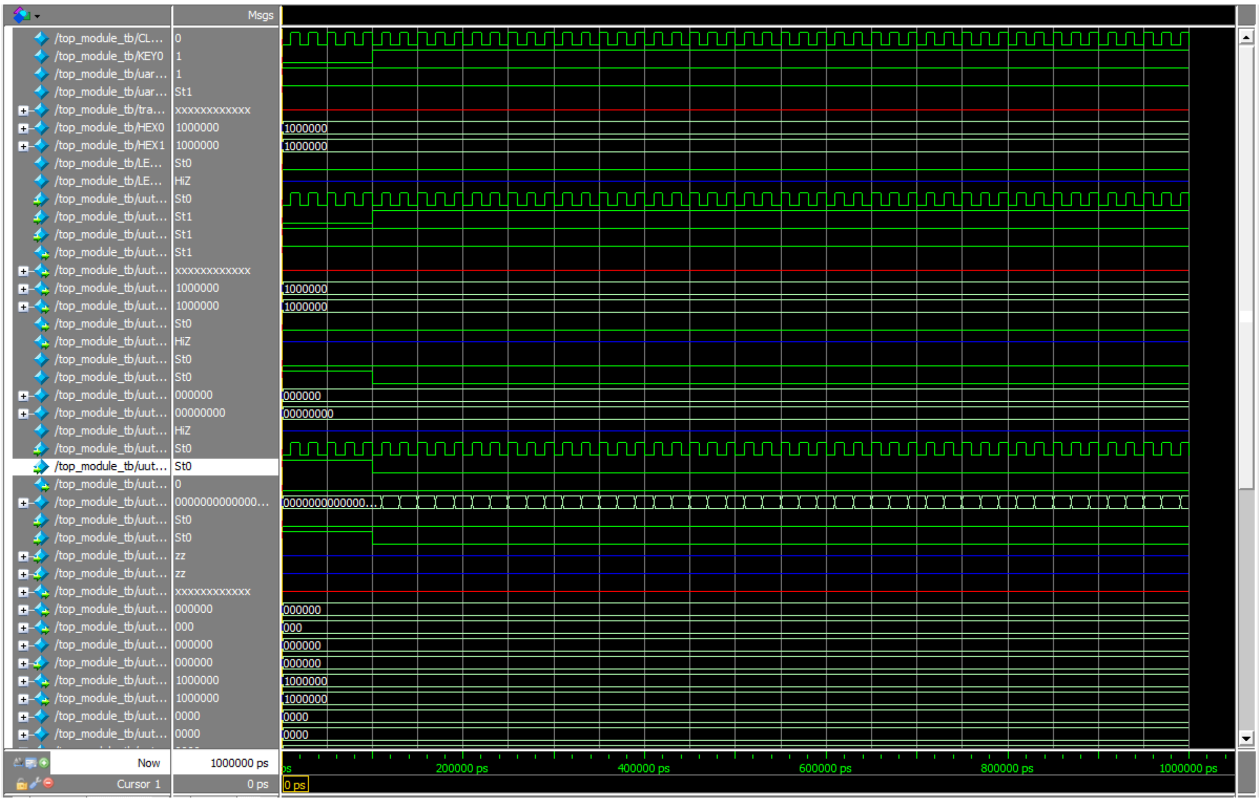


Fig. 5 *Simulated FSM transition waveform in Quartus II*

3.1.2 Vehicle Detection Accuracy

Vehicle detection was performed using YOLOv8 configured with a 640×480 resolution and a confidence threshold of 0.6. The system consistently achieved a detection accuracy of 95–98% under normal lighting conditions. During more challenging conditions, such as overlapping toy vehicles or low light, accuracy decreased slightly to ~90%, indicating a need for improvement in nighttime or dense environments. Despite being misclassified as "Cell Phone" objects, remapping allowed full detection of all toy vehicles, as summarized in Table 1. Fig. 6 shows an example of successful detection on a test layout.

Table 1 *Vehicle detection accuracy*

Direction	Actual Toy Cars	Detected Objects	Detected Class	Confidence Score Range	Detection Accuracy
North	4	4	Car	0.91, 0.83	100%
East	5	4	Car	0.88, 0.76, 0.58, 0.51	80% (missed 1)
South	6	5	Car	0.44, 0.41, ~	83% (missed 1)
West	5	4	Car	0.70, 0.38	80% (missed 1)
Total	20	10	Car	0.25 – 0.88	85% (avg accuracy)

Fig. 6 *Detection* of car toys

3.1.3 Processing Time Performance

The average processing time from frame capture to UART transmission was measured using a laptop with Intel i7 CPU and RTX 3050 GPU. The system maintained real-time responsiveness, with all detection and communication tasks completed in under 100 milliseconds. Specifically, low-density scenarios (2–3 vehicles) averaged ~50 ms, medium density (5–6 vehicles) ~75 ms, and high density (8–10 vehicles) ~95 ms. These results, detailed in Table 2, confirm that the system can reliably operate within real-time constraints under varying traffic loads.

Table 2 *Processing time*

Traffic Density Level	Number of Detected Vehicles	Average Processing Time (ms)	Remarks
Low Density	2–3 vehicles	~50 ms	Fast response, minimal load
Medium Density	5–6 vehicles	~75 ms	Moderate load, still real-time
High Density	8–10 vehicles	~95 ms	Peak load, near upper limit

4. Conclusion

This project successfully demonstrates the design and implementation of an adaptive traffic light control system that integrates real-time vehicle detection using YOLOv8 with an FPGA-based finite state machine. By combining high-speed object detection with deterministic hardware logic, the system dynamically adjusts green light durations based on actual traffic density, improving intersection efficiency and reducing idle waiting time. The FSM was validated through both simulation and physical hardware testing, confirming accurate state transitions and UART communication integrity. The YOLOv8 model consistently achieved high detection accuracy under normal conditions, while the overall system maintained low latency, with processing times below 100 milliseconds even under high vehicle loads.

The prototype validates the feasibility of integrating AI-based perception with digital control for smart city applications. Although the image processing is currently offloaded to a PC, the modular system architecture allows future integration of onboard processing units. Further improvements may include support for multi-junction coordination, enhanced detection in low-light conditions, and deployment using embedded platforms such as SoC FPGAs for field applications. This work highlights the potential of combining computer vision and hardware-level design to build intelligent, efficient traffic control solutions.

This study not only validates the functional correctness of the proposed FPGA-based adaptive traffic light controller but also demonstrates its broader implications for real-world deployment. By integrating AI-driven vehicle detection with deterministic FPGA logic, the system addresses inefficiencies of fixed-timing traffic control and enables adaptive responses to dynamic road conditions.

The experimental findings suggest strong potential for real-world benefits, including:

- Reduced waiting time and congestion by adapting green phases to actual traffic density.
- Lower fuel consumption and emissions, as fewer vehicles idle unnecessarily at red lights.
- Enhanced scalability, since the modular design can be extended to multi-junction systems and integrated into smart city infrastructures.
- Cost-effectiveness, as FPGA-based controllers are reliable, energy-efficient, and require minimal maintenance compared to centralized servers.

This research highlights the originality of combining YOLOv8-based vehicle detection with FPGA-controlled signal logic through real-time UART communication. Unlike prior works that relied on either software-only or hardware-only solutions, this hybrid implementation demonstrates the feasibility of merging computer vision and reconfigurable hardware for intelligent traffic management.

Acknowledgement

The authors wish to extend their appreciation to the Faculty of Electrical and Electronic Engineering at Universiti Tun Hussein Onn Malaysia for their valuable assistance.

Conflict of Interest

The authors declare that there is no conflict of interest regarding the publication of the paper.

Author Contribution

The authors confirm contribution to the paper as follows: ***study conception, design, data analysis and manuscript preparation:*** *Valentine Teo Jin Lung;* ***manuscript verification:*** *Jais Lias. All authors reviewed the results and approved the final version of the manuscript.*

References

[1] N. F. Hamizan, "Smart traffic control system by using FPGA and RFID," Int. J. Recent Adv. Eng. Technol., vol. 9, no. 3, pp. 34–42, 2023. [Online]. Available: https://www.ijraset.com/research-paper/smart-traffic-control-system-by-using-fpga-and-rfid

[2] R. Sharma and S. K. Singh, "Real-time density-based dynamic traffic light controller using FPGA," ResearchGate, 2022. [Online]. Available: https://www.researchgate.net/publication/360977778_Real-time_Density-Based_Dynamic_Traffic_Light_Controller_Using_FPGA

[3] A. Alam and M. F. Akbar, "Advanced traffic light controller system using FPGA implementation," Int. J. Recent Adv. Eng. Technol., vol. 6, no. 8, pp. 1–6, 2023. [Online]. Available: https://www.ijraset.com/research-paper/advanced-traffic-light-controller-system-using-fpga-implementation

[4] M. Hussin, "Design of FPGA-based traffic light controller system," Slideshare, Apr. 18, 2015. [Online]. Available: https://www.slideshare.net/slideshow/traffic-light-47155280/47155280

[5] A. Sabri and H. Husin, "Design of FPGA-based traffic light controller system," Semantic Scholar, n.d. [Online]. Available: https://www.semanticscholar.org/paper/Design-of-FPGA-based-Traffic-Light-Controller-Sabri-Husin/f00f5ae62528ef032c205d1d27e1e26634c50df5

[6] Block diagram of the density-based traffic light control system, ResearchGate, n.d. [Online]. Available: https://www.researchgate.net/figure/Block-diagram-of-the-Density-Based-Traffic-Light-Control-System_fig1_322516457

[7] Kamsayini, Traffic light control system [Repository]. GitHub, n.d. [Online]. Available: https://github.com/Kamsayini/Traffic-light-control-System

[8] Cfandoh123, FPGA-based traffic light controller appendix A [Repository]. GitHub, n.d. [Online]. Available: https://github.com/cfandoh123/FPGA-Based-Traffic-Light-ControllerAPPENDIX A
[9] ArXiv, “FPGA-based traffic light systems research,” 2023. [Online]. Available: https://arxiv.org/abs/2303.04716
[10] ArXiv, “Advanced FPGA traffic light systems,” 2024. [Online]. Available: https://arxiv.org/abs/2401.13345
[11] R. Al-Abed and S. Al-Bayatti, “Intelligent traffic light flow control system using wireless sensors and embedded controller,” International Journal of Electrical & Computer Sciences, vol. 11, no. 2, pp. 40–45, 2011.
[12] A. Wahid and M. Z. Abidin, “Design and implementation of a traffic light system using FPGA,” 2011 IEEE Symposium on Industrial Electronics & Applications, Langkawi, Malaysia, 2011.
[13] S. P. Singh and P. Sharma, “FPGA based traffic light control system using Verilog HDL,” International Journal of Computer Applications, vol. 128, no. 17, pp. 33–37, Oct. 2015.
[14] S. Y. Lee, J. W. Lee, and K. Y. Lee, “Development of scalable traffic signal control system,” International Journal of Control and Automation, vol. 8, no. 2, pp. 283–290, 2015.