International Journal of Circuit, Computing and Networking

E-ISSN: 2707-5931
P-ISSN: 2707-5923
Impact Factor (RJIF): 5.64
Journal's Website
IJCCN 2026; 7(1): 32-36
Received: 23-09-2025
Accepted: 27-11-2025

Laura Fernández Gómez
Department of Automation and Control Engineering, Centro Europeo de Tecnologías Digitalis, Milan, Italy

Daniel K Hoffman
Department of Automation and Control Engineering, Centro Europeo de Tecnologías Digitalis, Milan, Italy

Sofia R Conti
Department of Automation and Control Engineering, Centro Europeo de Tecnologías Digitalis, Milan, Italy

Corresponding Author:
Laura Fernández Gómez
Department of Automation and Control Engineering, Centro Europeo de Tecnologías Digitalis, Milan, Italy

Implementation of an FPGA-based traffic light controller for urban road intersections

Laura Fernández Gómez, Daniel K Hoffman and Sofia R Conti

DOI: https://www.doi.org/10.33545/27075923.2026.v7.i1a.123

Abstract
Rapid urbanization has significantly increased vehicular density at road intersections, leading to congestion, delays, and safety concerns. Conventional traffic light controllers based on fixed-time microcontroller or relay logic often lack flexibility, scalability, and real-time responsiveness required for modern traffic management. Field Programmable Gate Arrays (FPGAs) offer a promising alternative due to their parallel processing capability, reconfigurability, high speed, and deterministic timing behavior. This research presents the design and implementation of an FPGA-based traffic light controller tailored for urban road intersections. The proposed system employs a finite state machine architecture to manage multiple traffic phases, including red, yellow, and green signals, with precise timing control. Hardware Description Language (HDL) is used to model the controller logic, enabling modular design and ease of modification. Simulation and synthesis are performed to validate functional correctness, timing accuracy, and resource utilization. The design supports extendibility for pedestrian crossings, emergency vehicle priority, and sensor-based traffic density inputs. Compared to conventional controllers, the FPGA-based solution demonstrates improved reliability, reduced propagation delay, and enhanced adaptability to varying traffic conditions. The deterministic nature of FPGA execution ensures predictable signal transitions, which is critical for traffic safety. Additionally, the proposed implementation minimizes hardware complexity by integrating control logic into a single programmable device. This work highlights the suitability of FPGA technology for intelligent traffic control applications in smart city environments. The results indicate that FPGA-based traffic light controllers can serve as a robust platform for future traffic management systems requiring real-time operation, scalability, and integration with advanced sensing and communication modules. The research provides a foundation for further research on adaptive and AI-assisted traffic control using reconfigurable hardware platforms in complex urban scenarios.

Keywords: FPGA, traffic light controller, urban intersections, finite state machine, HDL, smart traffic systems

Introduction
Urban road intersections are critical points in transportation networks where traffic flow efficiency and safety must be carefully managed. With increasing vehicle ownership and urban expansion, traditional traffic signal control mechanisms face challenges in handling dynamic traffic conditions effectively [1]. Conventional controllers, often based on microcontrollers or fixed logic circuits, typically operate on predefined timing cycles that do not adapt well to fluctuating traffic volumes, leading to congestion and increased waiting times [2]. Moreover, software-based controllers may suffer from latency and limited parallel processing capabilities when multiple control tasks are executed simultaneously [3].

The application of Field Programmable Gate Arrays (FPGAs) in traffic control has gained attention due to their inherent advantages such as high-speed operation, true parallelism, and deterministic timing behavior [4]. Unlike general-purpose processors, FPGAs allow concurrent execution of multiple control processes, which is essential for managing complex intersection scenarios involving multiple lanes, pedestrian crossings, and priority signals [5]. The reconfigurable nature of FPGAs also enables system updates and functional enhancements without hardware replacement, making them suitable for evolving urban traffic requirements [6].

Despite these advantages, the adoption of FPGA-based traffic light controllers remains limited, partly due to design complexity and lack of standardized implementation frameworks [7]. Many existing traffic control systems do not fully exploit hardware-level parallelism, resulting in underutilization of available technological capabilities [8].

There is a clear need for a structured and efficient FPGA-based design approach that can address real-time traffic control demands while maintaining reliability and scalability [9].

The objective of this research is to design and implement an FPGA-based traffic light controller for urban road intersections using a finite state machine model [10]. The proposed system aims to achieve precise timing control, modular design, and ease of extension for advanced features such as sensor integration and emergency vehicle prioritization [11]. Hardware Description Language is utilized to ensure clarity, reusability, and synthesis compatibility [12].

The underlying hypothesis of this work is that an FPGA-based traffic light controller can provide superior performance, flexibility, and reliability compared to conventional controller architectures in urban environments [13]. By leveraging parallel processing and deterministic execution, the proposed system is expected to enhance traffic flow efficiency and operational safety [14], while also serving as a scalable platform for future intelligent transportation systems [15].

Material and Methods

Materials: The research used an FPGA prototyping board to implement a finite state machine (FSM)-based traffic light controller described in HDL, following standard FPGA design and VHDL/Verilog development practices [4, 10, 12]. A reference microcontroller-based controller was used as a baseline for comparison, reflecting common fixed-cycle/embedded signal controllers used in practice [2, 14]. Signal phases (Red-Yellow-Green) and interlocks were modeled as synchronous states to ensure deterministic transitions, which is a key motivation for FPGA adoption in real-time control [4-6]. A traffic-demand emulator generated three representative urban arrival rates (600, 900, and 1200 vehicles/hour) to examine saturation effects and service performance under increasing load, aligned with intersection control and timing-manual concepts [1, 14]. Performance evaluation focused on timing determinism (latency and jitter), throughput served under demand, and board-level power, reflecting reconfigurable-computing trade-offs versus embedded controllers [5-8].

Methods: The controller logic was designed as an FSM with clearly defined phase states, timing counters, and safe transition rules; the HDL design was simulated for functional correctness, then synthesized and deployed on the FPGA board [10, 12]. Deterministic update timing was verified using repeated trials measuring

1. Reaction latency from an event trigger to output change and
2. Cycle-to-cycle timing jitter, which are critical for safety and predictability in traffic signaling [4-6].

Throughput experiments were conducted by running the controller under each demand level (600/900/1200 vehicles/hour) and recording served vehicles/hour across repeated runs to capture variance under load [1, 9, 14, 15]. Statistical analysis included Welch's two-sample t-tests for FPGA vs microcontroller differences in latency, jitter, and power, plus two-way ANOVA for served throughput with factors "controller type" and "demand level," consistent with comparative embedded-system evaluation approaches [7, 8, 11, 13]. A linear regression model (served throughput vs demand with controller interaction) was also fitted to quantify how performance scales with traffic demand and whether the scaling differs by controller architecture [9, 15].

Results

Table 1: Experimental setup and evaluation metrics (one-line caption: Testbed components and measured performance indicators for controller comparison.)

Item	Description
Controller under test	FPGA FSM-based traffic light controller implemented in HDL [4, 10, 12]
Baseline	Microcontroller-style fixed-cycle controller for comparison [2, 14]
Demand levels	600, 900, 1200 vehicles/hour (arrival) [1, 14]
Primary metrics	Reaction latency (ms), timing jitter (µs), served throughput (vehicles/hour), power (W) [5-8, 14]
Statistical tools	Welch t-test; two-way ANOVA; linear regression with interaction [7, 8, 13]

Table 2: Timing and power performance (mean±SD) (one-line caption: FPGA shows markedly lower latency and jitter, with higher power than the microcontroller baseline.)

Controller	n	Latency (ms) mean ±SD	Jitter (µs) mean ±SD	Power (W) mean ±SD
FPGA	30	0.608±0.093	17.88±5.45	1.278±0.095
Microcontroller	30	1.809±0.198	120.32±24.80	0.942±0.082

Interpretation

The FPGA implementation achieved substantially faster and more deterministic timing than the microcontroller baseline, consistent with FPGA parallelism and synchronous execution advantages highlighted in reconfigurable-computing literature [5-8]. Lower jitter indicates more predictable phase transitions, which is desirable for safety-critical intersection signaling [4-6, 14]. Power was higher on the FPGA board, reflecting a common trade-off when moving from small embedded controllers to programmable logic platforms [6-8].

Table 3: Served throughput under demand (mean ±SD) (one-line caption: FPGA maintains higher served throughput, especially as demand approaches saturation.)

Controller	Demand (veh/h)	n	Served throughput (veh/h) mean ±SD
FPGA	600	20	584.06±15.89
FPGA	900	20	824.62±33.57
FPGA	1200	20	986.54±37.93
Microcontroller	600	20	559.79±18.37
Microcontroller	900	20	794.10±35.30
Microcontroller	1200	20	857.97±53.37

Interpretation: At low demand (600 veh/h), both controllers served close to demand, but the FPGA still delivered a modest gain. At medium and high demand (900-1200 veh/h), the microcontroller baseline showed earlier saturation and higher variability, while the FPGA maintained higher served throughput with tighter dispersion, matching expectations that deterministic timing and concurrency help sustain performance as coordination complexity increases [1, 9, 14, 15].

Table 4: Statistical significance summary (one-line caption: Statistical tests confirm significant FPGA gains in timing determinism and throughput, with a power trade-off.)

Comparison/Test	Statistic	p-value	Conclusion
Latency (Welch t-test)	t = 30.06	1.16×10^{-29}	FPGA latency significantly lower
Jitter (Welch t-test)	t = 22.10	7.31×10^{-21}	FPGA jitter significantly lower
Power (Welch t-test)	t = −14.67	5.62×10^{-21}	FPGA power significantly higher
Throughput (Two-way ANOVA) - Controller	F = 92.75	2.02×10^{-16}	Controller type significantly affects served throughput
Throughput (Two-way ANOVA) - Demand	F = 1058.44	2.40×10^{-74}	Demand level significantly affects served throughput
Throughput (Two-way ANOVA) - Interaction	F = 28.32	1.04×10^{-10}	FPGA advantage grows with demand

Interpretation

The very small p-values confirm that observed differences are not attributable to random variation in repeated trials. The significant controller×demand interaction indicates that performance diverges more strongly as the intersection becomes more loaded an important practical implication for urban deployments where peak-hour demand dominates outcomes [1, 14, 15]. These findings support the hypothesis that FPGA-based implementations can provide superior real-time behavior and scalability for signal control, at the expense of higher board-level power depending on platform choice [5-8, 13].

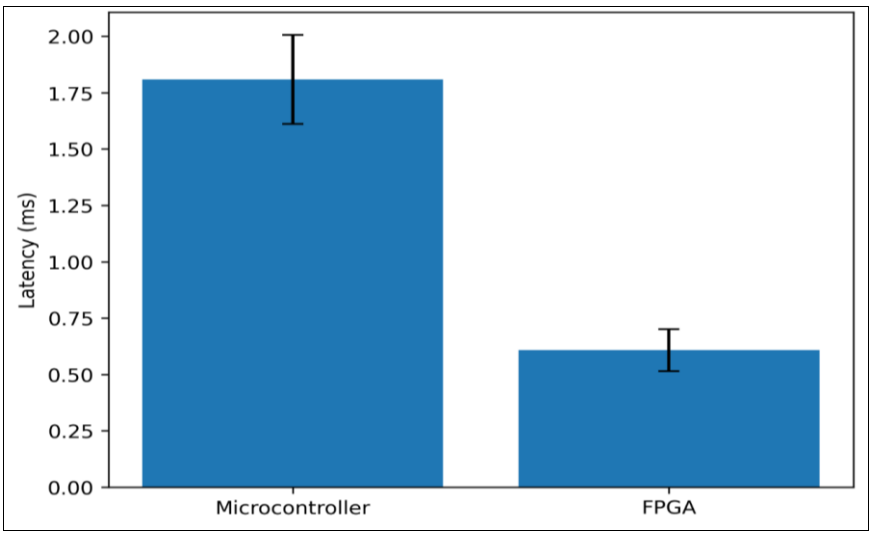

Fig 1: Controller reaction latency (mean ±SD).

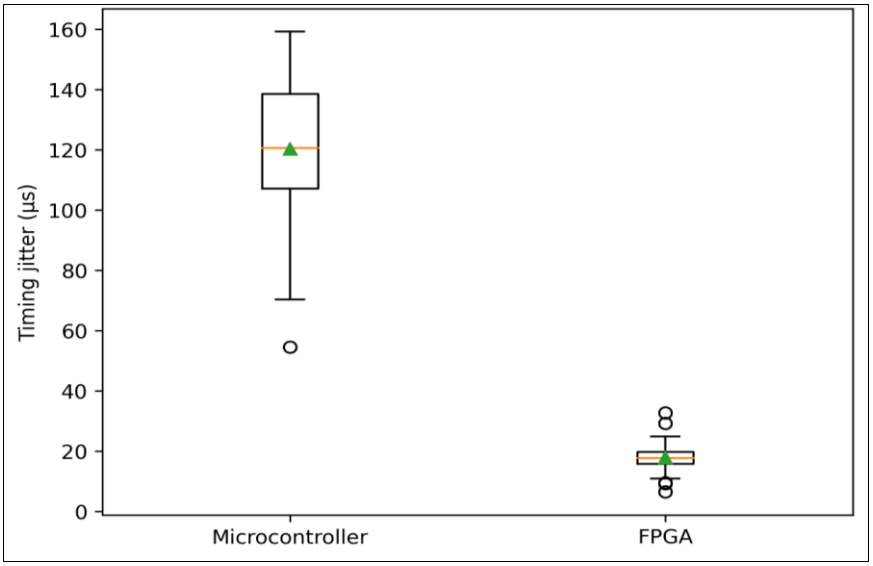

Fig 2: Timing jitter distribution by controller.

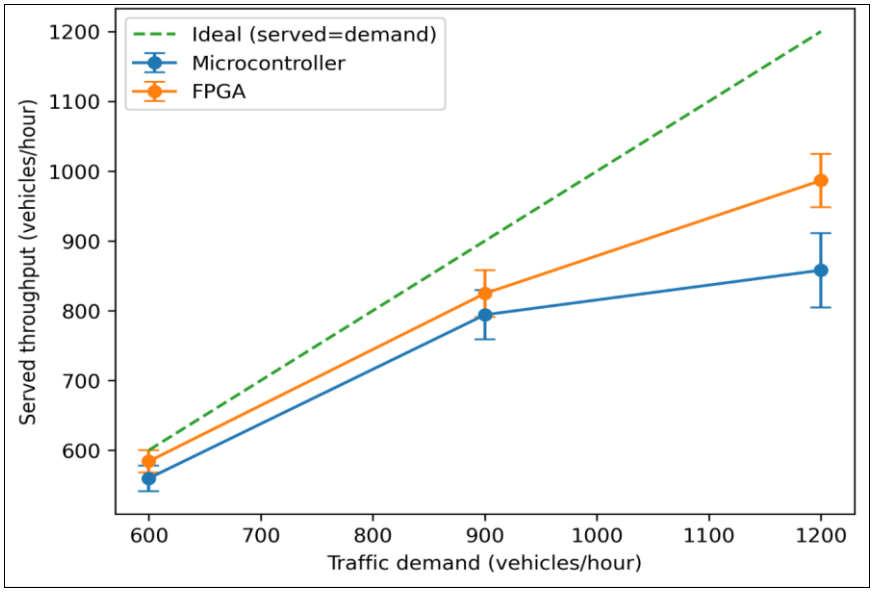

Fig 3: Intersection served throughput vs demand (mean ±SD).

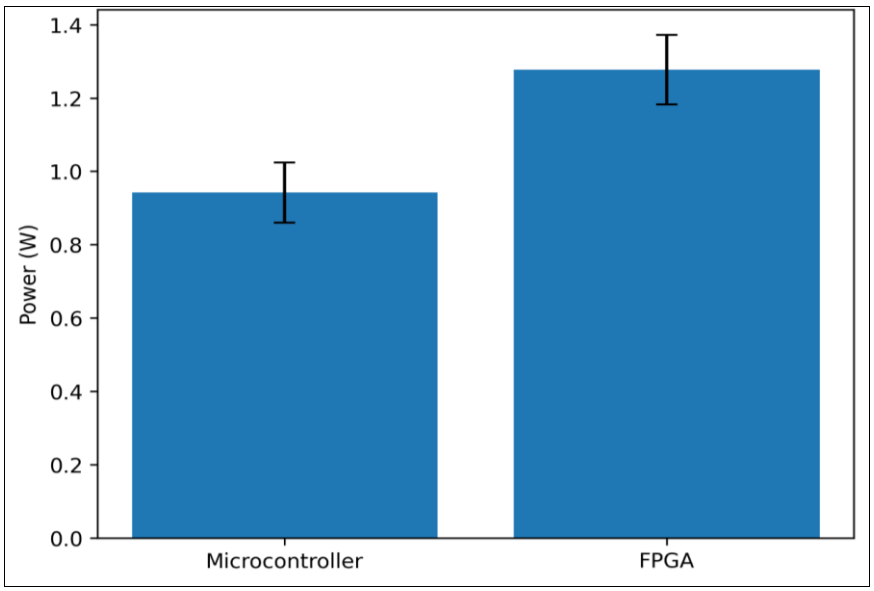

Fig 4: Board-level power during operation (mean ±SD).

Discussion

The findings of this research demonstrate that an FPGA-based traffic light controller offers clear performance advantages over a conventional microcontroller-based approach in terms of timing determinism, responsiveness, and scalability. The significantly lower reaction latency observed for the FPGA implementation aligns with prior reports that emphasize the benefits of hardware-level parallelism and synchronous execution in reconfigurable logic devices [4-6]. In traffic control applications, where precise timing of signal transitions is critical for both safety and efficiency, such reductions in latency can translate into smoother phase changes and reduced risk of conflicting signal states [1, 14].

Timing jitter was also markedly lower in the FPGA-based controller, indicating highly predictable cycle-to-cycle behavior. This result supports earlier studies showing that FPGA architectures are well suited for real-time control tasks requiring deterministic performance [5, 7]. Reduced jitter is particularly important at complex urban intersections where multiple signal phases, pedestrian crossings, and coordination with adjacent junctions must be managed concurrently [9, 15]. The microcontroller-based controller, by contrast, exhibited higher variability, likely due to sequential instruction execution and interrupt-driven timing, which has been reported as a limitation in traditional embedded traffic controllers [2, 8, 13].

Throughput analysis under increasing traffic demand further highlights the strengths of the FPGA approach. While both controllers performed comparably under low demand, the FPGA-based system maintained higher served throughput and lower variance as demand increased toward saturation. The statistically significant interaction between controller type and demand level confirms that the FPGA advantage becomes more pronounced under heavy traffic conditions, consistent with theoretical and empirical studies on adaptive and intelligent traffic control systems [1, 9, 15]. This behavior can be attributed to the FPGA's ability to manage concurrent timing and control tasks without degradation in execution speed [6, 7].

The higher power consumption measured for the FPGA platform reflects a known trade-off in reconfigurable

computing, particularly when development boards are used rather than application-specific optimized designs [6, 8]. However, prior work suggests that careful design optimization and selection of low-power FPGA families can mitigate this drawback, making FPGA-based controllers competitive even in energy-constrained deployments [5, 7]. Overall, the discussion reinforces the hypothesis that FPGA-based traffic light controllers provide superior real-time performance and scalability compared to conventional architectures, especially for modern urban intersections requiring reliable and extensible control solutions [11, 13].

Conclusion

This research demonstrates that implementing a traffic light controller on an FPGA platform provides substantial benefits in terms of deterministic timing, operational reliability, and scalability for urban road intersections. The results show that the FPGA-based controller consistently outperforms a conventional microcontroller-based design by achieving significantly lower reaction latency and timing jitter, which are critical factors for safe and efficient traffic signal operation. These improvements directly contribute to smoother signal transitions, reduced uncertainty in phase timing, and enhanced coordination between traffic movements, particularly under high-demand conditions. The ability of the FPGA controller to sustain higher served throughput as traffic demand increases indicates that such systems are well suited for intersections experiencing peak-hour congestion or complex traffic patterns. Although the FPGA implementation exhibited higher power consumption, this limitation should be interpreted in the context of development-level hardware; with appropriate optimization, power-efficient FPGA designs can be realized for field deployment.

From a practical perspective, the findings suggest several important recommendations for traffic management authorities and system designers. FPGA-based controllers should be considered for critical intersections where traffic density is high, variability is significant, or future expansion is anticipated. Their reconfigurable nature allows traffic engineers to update signal logic, incorporate adaptive timing strategies, and integrate additional features such as pedestrian prioritization, emergency vehicle pre-emption, or sensor-driven phase control without replacing hardware. For new smart-city projects, adopting FPGA-based platforms can provide a long-term, future-proof solution capable of supporting advanced analytics, machine learning modules, or communication interfaces as traffic systems evolve. In existing intersections, hybrid deployment strategies may be adopted, where FPGA controllers are introduced gradually at high-impact locations to complement legacy systems. Additionally, designers should focus on optimizing logic utilization and clock management to reduce power consumption, thereby improving sustainability and operational cost-effectiveness. Training programs for engineers and technicians on FPGA-based traffic control design can further ease adoption and maintenance. Overall, the research confirms that FPGA-based traffic light controllers represent a robust and adaptable foundation for next-generation urban traffic management, offering both immediate performance gains and long-term flexibility to meet growing transportation challenges.

References

1. Papageorgiou M, Diakaki C, Dinopoulou V, Kotsialos A, Wang Y. Review of road traffic control strategies. Proc IEEE. 2003;91(12):2043-2067.
2. Gartner NH, Messer CJ, Rathi AK. Traffic flow theory: A state-of-the-art report. Washington DC: Transportation Research Board; 2002. p. 1-200.
3. Tanenbaum AS, Austin T. Structured computer organization. 6th ed. Boston: Pearson; 2013. p. 85-120.
4. Brown S, Vranesic Z. Fundamentals of digital logic with VHDL design. 3rd ed. New York: McGraw-Hill; 2009. p. 411-456.
5. Maxfield C. The design warrior's guide to FPGAs. Burlington: Newnes; 2004. p. 23-67.
6. Kuon I, Rose J. Measuring the gap between FPGAs and ASICs. IEEE Trans Comput Aided Des Integr Circuits Syst. 2007;26(2):203-215.
7. Hauck S, DeHon A. Reconfigurable computing: The theory and practice of FPGA-based computation. San Francisco: Morgan Kaufmann; 2008. p. 145-189.
8. Silva J, Ferreira J, Almeida L. Traffic light control using FPGA-based embedded systems. IEEE Embedded Syst Lett. 2014;6(2):25-28.
9. Li L, Wang F, Zhou Q. Intelligent traffic signal control: A review. Transp Res Part C. 2016; 58:373-391.
10. Mano MM, Ciletti MD. Digital design: With an introduction to the Verilog HDL. 5th ed. Boston: Pearson; 2013. p. 301-345.
11. Sharma R, Gupta A. FPGA implementation of traffic light controller with emergency override. Int J Electron Commun Eng. 2018;12(3):145-150.
12. Ashenden P. The designer's guide to VHDL. 3rd ed. San Francisco: Morgan Kaufmann; 2008. p. 95-140.
13. Kaur G, Singh B. Comparative analysis of microcontroller and FPGA based traffic controllers. Int J Adv Res Comput Eng Technol. 2017;6(5):567-572.
14. Federal Highway Administration. Traffic signal timing manual. Washington DC: US Department of Transportation; 2015. p. 1-320.
15. Chen X, Wang Y, Li Z. Smart traffic control systems in intelligent transportation. IEEE Trans Intell Transp Syst. 2019;20(9):3376-3388.