

Archives available at journals.mriindia.com**International Journal of Advanced Electrical and Electronics Engineering**

ISSN: 2278-8948

Volume 13 Issue 02, 2024

FPGA-Based Intelligent Traffic Monitoring and Signal Control System for Smart Cities

Oisin Usmonov

Department of Computer Science and Engineering, Basra Institute of Business Technology, Iraq
oisin.usmonov@bibt-iq.org**Peer Review Information***Submission: 22 March 2024**Revision: 19 April 2024**Acceptance: 12 May 2024***Keywords***Field-Programmable Gate Array, Intelligent Transportation System, Smart City, Traffic Monitoring, Adaptive Signal Control***Abstract**

Urban traffic congestion requires monitoring and signal control that can react within seconds while remaining energy-efficient, deterministic, and deployable near the intersection. This methodology paper reviews field-programmable gate array (FPGA) approaches for traffic video processing and combines them with contemporary adaptive traffic-signal concepts to formulate a practical smart-city architecture. The proposed method places image acquisition, filtering, vehicle detection, tracking, queue estimation, and traffic-state extraction in a deeply pipelined FPGA datapath, while a lightweight adaptive controller allocates green time according to normalized queue length, flow, occupancy, and emergency priority. A fail-safe finite-state machine enforces minimum green, amber, and all-red clearance constraints. The framework also supports an edge gateway for logging, model updates, and city-level analytics without continuously transmitting raw video. A reproducible four-approach traffic simulation is used to illustrate the evaluation procedure. Across 30 seeded runs of a 2400 s scenario, the adaptive policy reduced mean waiting time from 20.97 s to 12.46 s and mean total queue from 13.19 to 7.85 vehicles relative to a fixed-time baseline. These values are simulation outputs under stated assumptions, not field measurements. The reviewed literature and proposed methodology indicate that FPGA acceleration is particularly suitable where low latency, predictable timing, and energy-aware edge processing are more important than unrestricted software flexibility.

Introduction

Traffic congestion at urban intersections is a coupled sensing-and-control problem. A controller must estimate the current traffic state, distinguish transient fluctuations from sustained queues, and update signal phases without violating safety clearances. Smart-city deployments add further constraints: cameras and sensors may operate continuously; network bandwidth can be limited; privacy concerns discourage unnecessary transmission of raw imagery; and the control loop must remain dependable even when cloud connectivity is

unavailable. These requirements motivate edge computing close to the road rather than a purely centralized architecture [2,13,14].

FPGAs are attractive for this edge role because image-processing stages can execute concurrently as fixed streaming pipelines. Earlier work demonstrated real-time FPGA vehicle extraction for traffic-surveillance sensor nodes [1] and later integrated FPGA/ASIC nodes into low-power urban monitoring networks [2]. More recent research has shown that smart-city video analytics on FPGAs can reduce energy per operation relative to CPU/GPU alternatives for

traffic-flow processing [3], while FPGA accelerators have also been applied to modern vehicle detection and tracking pipelines [9].

At the control level, the research direction has moved from fixed-cycle plans toward data-driven, IoT-enabled, fuzzy, and reinforcement-learning strategies. Traffic signal control studies report that learned or adaptive policies can exploit queue and occupancy observations to allocate time more effectively under non-stationary demand [4-8,10-13]. The challenge is to connect these intelligent policies to a hardware platform that can deliver measurements and safe actions with bounded latency. The objective of this paper is therefore to define a methodology for an FPGA-based intelligent traffic monitoring and signal-control system, not merely to describe a stand-alone traffic-light timer.

The proposed system is intended for a four-approach urban intersection and can be extended to coordinated corridors. Its design objectives are: (i) real-time extraction of vehicle count, queue length, lane occupancy, and approximate flow; (ii) deterministic low-latency execution of vision primitives in FPGA fabric; (iii) adaptive phase selection using interpretable traffic-demand variables; (iv) explicit emergency-vehicle priority; (v) local operation during network outages; and (vi) privacy-preserving transmission of traffic statistics rather than continuous raw video. The scope is methodology-level: it specifies data paths, decision rules, safety constraints, and an evaluation protocol that can be implemented on a Zynq-class FPGA/SoC or a comparable device.

A central design choice is hardware-software partitioning. Pixel-rate operations such as filtering, background subtraction or compact neural-network inference are mapped to programmable logic; state aggregation and higher-level policy calculations may execute in logic, a soft processor, or the embedded CPU. This partition follows the evidence that FPGA pipelines are efficient for regular image workloads [1-3,9], while intelligent signal-control studies often require flexible state handling and policy updates [6-8,11-13]. The resulting architecture is therefore heterogeneous rather than FPGA-only in the narrow sense.

Literature Review

Wojcikowski, Zaglewski, and Pankiewicz [1] provided an important foundation by implementing moving-vehicle extraction for an automatic traffic-surveillance sensor network directly on FPGA hardware. Their algorithm was intentionally adapted to pipelined realization and produced vehicle-related parameters from outdoor video. A subsequent hardware-software

sensor-network study [2] showed how local processing can reduce communication needs and support distributed city traffic monitoring. Together, these studies established the feasibility of placing traffic perception at the edge rather than streaming all imagery to a central server.

Arif et al. [3] shifted attention from an isolated sensor node to the broader smart-city computing problem. Their work described FPGA implementations of traffic-camera image processing and emphasized energy efficiency for large-scale urban video analytics. This is relevant to smart intersections because traffic-monitoring infrastructure is continuously active; reducing computation and communication energy can materially affect operating cost. The work also highlights the development trade-off: FPGAs can provide strong energy-performance characteristics, but hardware design and high-level-synthesis optimization require more engineering effort than conventional CPU code.

On the traffic-control side, Joo, Ahmed, and Lim [4] investigated reinforcement learning for smart-city traffic signal control, while Neelakandan et al. [5] combined IoT traffic prediction with signal-control logic. Tomar, Sreedevi, and Pandey [6] reviewed traffic-light synchronization and identified adaptive control, communication, and intelligent-vehicle integration as major directions.

FPGA vision research has simultaneously progressed toward more computationally demanding perception. Zhai et al. [9] presented an FPGA-based vehicle detection and tracking accelerator, illustrating how detection and multi-object tracking can be accelerated in hardware for traffic-flow statistics. This provides a bridge between classical background-subtraction pipelines and neural perception. In practice, the appropriate algorithm depends on device resources and deployment conditions: a lightweight classical pipeline is attractive for low-cost intersections, while a quantized detector can improve robustness where occlusion, vehicle diversity, and complex backgrounds dominate.

Majstorovic et al. [10] reviewed urban traffic-signal control under mixed traffic flows that include human-driven, connected, and automated vehicles. Their synthesis is important because future intersections will receive heterogeneous information: camera observations, infrastructure sensors, connected-vehicle messages, and emergency priority requests. Multi-agent reinforcement learning has also been investigated for coordinated intersections. Kolat et al. [11] used a cooperative multi-agent formulation, showing why coordination becomes necessary when queues and spillback propagate across neighboring

junctions. For a single FPGA edge node, this implies that the local controller should expose compact traffic-state messages so that corridor-level coordination can be added without redesigning the pixel-processing pipeline.

Two 2024 studies further motivate the proposed architecture. Mani et al. [12] demonstrated an FPGA-oriented emergency-vehicle classification system, supporting the use of accelerated perception for priority handling. Ouyang, Zhan, and Lv [13] compared reinforcement-learning

algorithms for traffic-signal control, reinforcing the practical need to evaluate policies against clear baseline metrics such as delay and queue length. At the wider smart-city level, Omrany et al. [14] identified data analysis, communication, security/privacy, interoperability, energy, and scalability as recurring IoT-enabled smart-city challenges. These concerns justify local traffic-state extraction, limited transmission of raw imagery, and modular interfaces between perception, control, and city platforms.

Table 1. Representative literature informing the proposed methodology

Ref.	Year	Main focus	Computing/control approach	Methodological relevance
[1]	2012	Traffic surveillance	FPGA video pipeline	Real-time vehicle extraction
[2]	2013	Urban sensor network	FPGA/ASIC edge nodes	Distributed low-power monitoring
[3]	2020	Smart-city video analytics	FPGA acceleration	Energy-efficient traffic processing
[6]	2022	Signal synchronization review	Adaptive/connected methods	Control requirements and trends
[9]	2023	Vehicle detection/tracking	FPGA accelerator	Hardware-accelerated perception
[11]	2023	Multi-intersection control	Multi-agent RL	Coordination concept
[12]	2024	Emergency classification	FPGA-based perception	Priority vehicle handling
[13]	2024	Signal-control comparison	Q-learning / DQN	Baseline evaluation metrics

The literature reveals a separation between two mature research streams. FPGA studies show fast and energy-aware traffic perception [1-3,9,12], whereas intelligent signal-control studies optimize phase timing using richer traffic states [4-8,10,11,13]. Fewer works describe a complete, safety-constrained methodology that joins these streams in one edge architecture. The gap addressed here is therefore the integration method: how measurements are generated in hardware, normalized into control variables, converted into green-time decisions, constrained by a fail-safe state machine, and evaluated against a fixed-time baseline. The design intentionally avoids dependence on continuous cloud connectivity and keeps raw video local unless event evidence must be retained.

Methodology

1. System Architecture and Data Flow

Figure 1 summarizes the proposed architecture. A roadside camera supplies a continuous frame stream; optional inductive, radar, or IoT sensors may be fused when available. The FPGA first performs region-of-interest selection and image

conditioning. A streaming perception pipeline then produces vehicle detections and tracks, which are accumulated into lane-level queue length, flow, and occupancy. These values form the traffic-state vector. The adaptive controller converts the state into a demand score for each active movement and requests the next phase or green-time extension. All requested actions pass through a deterministic safety finite-state machine (FSM) that guarantees amber and all-red intervals before conflicting movements are released.

2. FPGA Traffic-Monitoring Pipeline

The video path is implemented as a sequence of streaming stages so that different pixels or objects are processed concurrently. For a classical low-resource implementation, the stages are: RGB-to-grayscale conversion, spatial filtering, adaptive background update, foreground segmentation, morphology, connected-component labeling, lane-region assignment, and centroid-based tracking. The same interface can accept a quantized neural detector when a larger FPGA/SoC is available, as

suggested by recent FPGA detection/tracking work [9]. The key hardware principle is that complete frames should not be copied repeatedly between processor memory and logic; AXI-stream or an equivalent streaming interface is preferred, with line buffers used for neighborhood operations.

For each approach i , the perception block produces: q_i (estimated queued vehicles), f_i (vehicles crossing the stop line per unit time), o_i (lane occupancy ratio), and v_i (optional mean

speed). Queue estimation may be derived from the number of tracked objects whose speed remains below a small threshold within a queue region. Flow is updated when a tracked centroid crosses a virtual stop-line detector. These compact variables are more useful to the signal controller than raw frames and are also less privacy-sensitive for city-level transmission [3,14].

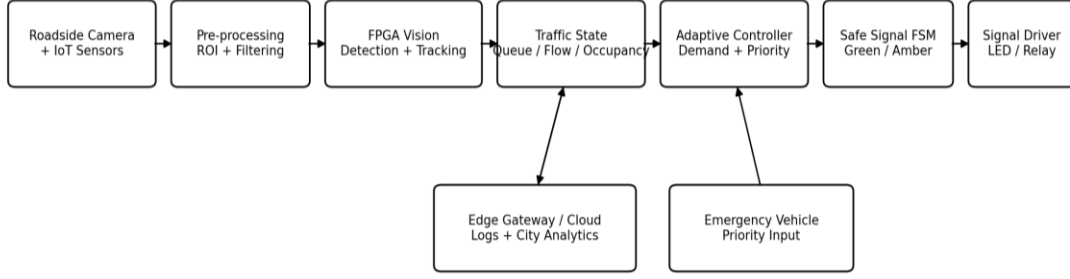


Figure 1. Proposed FPGA-based intelligent traffic monitoring and signal-control architecture.

3. Traffic-State Validation and Fault Handling

Before control, raw estimates pass through temporal validation. A short moving median or exponential filter suppresses single-frame spikes; physically impossible jumps are rejected; and sensor disagreement raises a confidence flag. If camera confidence falls below a threshold because of darkness, obstruction, or heavy rain, the controller can degrade to conservative fixed-time operation or use auxiliary sensor counts. This fallback is essential because an intelligent signal system must remain safe when perception is uncertain. The fallback plan and all-red clearance are stored locally so that a network or cloud failure cannot force an unsafe intersection state.

4. Demand Score and Adaptive Green Time

Each approach is assigned a normalized demand score. One implementable form is $D_i = w_q \cdot (q_i / q_{ref}) + w_f \cdot (f_i / f_{ref}) + w_o \cdot o_i + w_p \cdot p_i$, where p_i is an emergency-priority flag and the weights sum to one. Queue length receives the largest weight during congestion, while flow and occupancy prevent the controller from overreacting to a single stopped vehicle. For a two-phase intersection, north-south and east-west scores are aggregated. The phase with the higher validated demand may extend its green after a minimum-green constraint has been met. Green time is bounded as $g = \text{clip}(g_{min} + K \cdot D, g_{min}, g_{max})$. The exact weights are tunable from historical data or can be replaced by a trained fuzzy/RL policy without changing the FPGA measurement interface [4,6-8,13].

5. Emergency-Vehicle Priority

Emergency priority is treated as a supervised exception rather than an unconstrained override. A priority request may come from camera classification, a secure V2X message, RFID, or another authenticated source. FPGA-accelerated emergency-vehicle classification reported in 2024 [12] supports the feasibility of local perception. When $p_i = 1$, the demanded phase receives the highest score, but the controller still completes the active minimum green and inserts amber/all-red clearance before changing right-of-way. A maximum pre-emption duration prevents starvation of all other approaches.

6. Fail-Safe Signal State Machine

The physical traffic lights are driven only by a hardware FSM. Legal states include GREEN_NS, AMBER_NS, ALL_RED, GREEN_EW, AMBER_EW, and an emergency/fault state that defaults to a preconfigured safe pattern. The intelligent controller cannot directly assert conflicting green outputs; it can only request a phase transition or extension. This separation between optimization and safety is important for verification. Assertions can be written to guarantee that north-south and east-west greens are never active simultaneously and that every conflicting phase change contains the required clearance interval.

7. Hardware-Software Co-Design Procedure

Step 1 - Capture representative intersection video under daylight, night, rain, and peak/off-

peak traffic; define lane regions and virtual stop lines.

Step 2 - Implement and validate the perception algorithm in software; establish detection, queue, flow, and occupancy ground truth on a labeled subset.

Step 3 - Move pixel-rate kernels to FPGA logic using HDL or high-level synthesis; apply fixed-point quantization, pipelining, loop unrolling, and on-chip buffering were beneficial.

Step 4 - Connect lane statistics to the adaptive controller and enforce minimum/maximum green, amber, all-red, and fallback rules in the safety FSM.

Step 5 - Verify logic in simulation, then perform hardware-in-the-loop tests with recorded video before any live signal connection.

Step 6 - Evaluate traffic performance against fixed-time control using identical arrivals; separately report FPGA latency, frame rate, power, resource utilization, and perception accuracy after synthesis/measurement.

8. Reproducible Traffic-Simulation Protocol Used Here

To provide a concrete Results section without inventing field data, a simple queue simulation was executed for 2400 s with four approaches.

Vehicle arrivals were Poisson processes with piecewise-changing rates to create a north-south peak followed by an east-west peak. Each green approach discharged at 0.55 vehicle/s. The fixed-time baseline used 30 s green per major phase with 3 s clearance intervals. The adaptive policy reconsidered demand every 5 s, used a minimum green of 10 s, a maximum green of 45 s, and the same 3 s clearance. Thirty random seeds were averaged. The primary metrics were mean waiting time, mean total queue, maximum total queue, and throughput. This traffic model evaluates the control logic only; FPGA synthesis metrics must be obtained on the actual target board.

Results

Table 2 presents the mean results of the 30 seeded simulation runs. The adaptive controller reduced average waiting time by 40.57% and mean total queue by 40.46% relative to fixed-time control. The maximum total queue was reduced by 34.51%. Throughput changed only slightly (+0.26%) because both controllers had enough long-run service capacity to discharge most arrivals; the major benefit was therefore reduced delay and queue accumulation rather than a large increase in total completed vehicles.

Table 2. Simulation comparison of fixed-time and proposed adaptive control

Metric	Fixed-time	Adaptive	Change
Average waiting time (s/vehicle)	20.97	12.46	-40.57%
Mean total queue (vehicles)	13.19	7.85	-40.46%
Maximum total queue (vehicles)	36.22	23.72	-34.51%
Throughput (vehicles/hour)	2260.84	2266.79	+0.26%

The result pattern is consistent with the rationale of adaptive traffic control reported across the literature [4,7,8,11,13]: when demand shifts between approaches, a fixed split can waste green time on a lightly loaded direction while queues grow elsewhere. In the present simulation, the adaptive policy detects the larger queue and reallocates service subject to minimum and maximum green constraints. Because clearance losses occur whenever phases change, excessively frequent switching would be counterproductive; this is why the methodology includes minimum green and bounded decision intervals.

The simulation does not demonstrate FPGA performance by itself. Hardware claims should be reported only after synthesis and board measurement. Appropriate FPGA metrics include end-to-end frame latency, sustained frames per

second, logic/LUT utilization, block RAM, DSP use, clock frequency, dynamic power, and energy per processed frame. Prior FPGA traffic studies establish that real-time and low-power operation is feasible [1-3,9], but exact values depend strongly on image resolution, detector complexity, numerical precision, and device family. For a publication-quality experimental paper, Table 2 should therefore be accompanied by a measured hardware table once the design is implemented.

Conclusion and Discussion

This paper developed a methodology for an FPGA-based intelligent traffic monitoring and signal-control system suitable for smart-city intersections. The principal contribution is architectural integration: a streaming FPGA perception pipeline transforms video into

compact lane statistics; an adaptive controller converts those statistics into demand-aware timing requests; emergency priority is supported through a constrained override; and a separate fail-safe FSM guarantees safe signal transitions. The edge gateway is used for logging, coordination, and analytics, but safe local operation does not depend on continuous cloud access.

The literature through 2024 shows complementary progress in FPGA traffic perception [1-3,9,12], IoT-enabled traffic management [5,14], and adaptive/RL signal control [4,6-8,10,11,13]. The simple simulation in this paper illustrates how the proposed evaluation can quantify delay and queue reduction under changing demand; it should not be interpreted as a substitute for field trials. Future work should synthesize the full pipeline on a selected FPGA/SoC, test adverse-weather perception, compare classical and quantized neural detectors, verify the safety FSM formally, and evaluate coordinated multi-intersection operation with V2X data. Security, interoperability, privacy, and maintainability should be treated as deployment requirements rather than post-processing additions [14].

References

- M. Wojcikowski, R. Zaglewski, and B. Pankiewicz, "FPGA-Based Real-Time Implementation of Detection Algorithm for Automatic Traffic Surveillance Sensor Network," *Journal of Signal Processing Systems*, vol. 68, pp. 1-18, 2012. DOI: 10.1007/s11265-010-0569-3.
- M. Wojcikowski, R. Zaglewski, B. Pankiewicz, M. Klosowski, et al., "Hardware-Software Implementation of a Sensor Network for City Traffic Monitoring Using the FPGA- and ASIC-Based Sensor Nodes," *Journal of Signal Processing Systems*, vol. 71, pp. 57-73, 2013. DOI: 10.1007/s11265-012-0681-7.
- Arif, F. A. Barrigon, F. Gregoretti, J. Iqbal, L. Lavagno, M. T. Lazarescu, L. Ma, M. Palomino, et al., "Performance and energy-efficient implementation of a smart city application on FPGAs," *Journal of Real-Time Image Processing*, vol. 17, pp. 729-743, 2020. DOI: 10.1007/s11554-018-0792-x.
- H. Joo, S. H. Ahmed, and Y. Lim, "Traffic signal control for smart cities using reinforcement learning," *Computer Communications*, vol. 154, pp. 324-330, 2020. DOI: 10.1016/j.comcom.2020.03.005.
- S. Neelakandan, M. A. Berlin, S. Tripathi, V. B. Devi, I. Bhardwaj, and N. Arulkumar, "IoT-based traffic prediction and traffic signal control system for smart city," *Soft Computing*, vol. 25, no. 18, pp. 12241-12248, 2021. DOI: 10.1007/s00500-021-05896-x.
- Tomar, I. Sreedevi, and N. Pandey, "State-of-Art Review of Traffic Light Synchronization for Intelligent Vehicles: Current Status, Challenges, and Emerging Trends," *Electronics*, vol. 11, no. 3, art. 465, 2022. DOI: 10.3390/electronics11030465.
- S. Damadam, M. Zourbakhsh, R. Javidan, and A. Faroughi, "An Intelligent IoT Based Traffic Light Management System: Deep Reinforcement Learning," *Smart Cities*, vol. 5, no. 4, pp. 1293-1311, 2022. DOI: 10.3390/smartcities5040066.
- Tan, Q. Yuan, W. Guo, N. Xie, F. Liu, J. Wei, and X. Zhang, "Deep Reinforcement Learning for Traffic Signal Control Model and Adaptation Study," *Sensors*, vol. 22, no. 22, art. 8732, 2022. DOI: 10.3390/s22228732.
- Zhai, B. Li, S. Lv, and Q. Zhou, "FPGA-Based Vehicle Detection and Tracking Accelerator," *Sensors*, vol. 23, no. 4, art. 2208, 2023. DOI: 10.3390/s23042208.
- Z. Majstorovic, L. Tisljaric, E. Ivanjko, and T. Caric, "Urban Traffic Signal Control under Mixed Traffic Flows: Literature Review," *Applied Sciences*, vol. 13, no. 7, art. 4484, 2023. DOI: 10.3390/app13074484.
- P. Mani, P. R. G. Komarasamy, N. Rajamanickam, M. Shorfuzzaman, and W. M. Abdelfattah, "Enhancing Sustainable Transportation Infrastructure Management: A High-Accuracy, FPGA-Based System for Emergency Vehicle Classification," *Sustainability*, vol. 16, no. 16, art. 6917, 2024. DOI: 10.3390/su16166917.
- C. Ouyang, Z. Zhan, and F. Lv, "A Comparative Study of Traffic Signal Control Based on Reinforcement Learning Algorithms," *World Electric Vehicle Journal*, vol. 15, no. 6, art. 246, 2024. DOI: 10.3390/wevj15060246.
- H. Omrany, K. M. Al-Obaidi, M. Hossain, N. A. M. Alduais, H. S. Al-Duais, et al., "IoT-enabled smart cities: a hybrid systematic analysis of key research areas, challenges, and recommendations for future direction," *Discover Cities*, vol. 1, art. 2, 2024. DOI: 10.1007/s44327-024-00002-w.