



## FPGA-Based Implementation of a Real-Time Traffic Signal Control System

Chaminda Balasingam

Department of Electrical and Computer Engineering, Caspian Institute of Industrial Engineering, Iran  
[chaminda.balasingam@ciie-ir.edu](mailto:chaminda.balasingam@ciie-ir.edu)

| Peer Review Information                                                                                                                                                                                                                                | Abstract                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><i>Submission: 10 Jan 2021</i><br/> <i>Revision: 25 Feb 2021</i><br/> <i>Acceptance: 12 March 2021</i></p> <p><b>Keywords</b></p> <p><i>FPGA, traffic signal control, finite-state machine, VHDL/Verilog, intelligent transportation system</i></p> | <p>Urban intersections require deterministic, fast and adaptable signal control because fixed-time controllers cannot respond efficiently to directional traffic imbalance, pedestrian requests or emergency conditions. This methodology paper presents an FPGA-based real-time traffic signal control architecture in which synchronized sensor inputs are processed by a finite-state machine (FSM), adaptive green-time logic and hardware safety interlocks. The proposed design is expressed in synthesizable HDL and is organized into sensor-conditioning, queue-estimation, phase-control, timing, pedestrian-priority and output-driver modules. A controlled numerical intersection model was used to evaluate the control strategy before hardware deployment. Four one-hour traffic scenarios were generated using Poisson arrivals for north-south and east-west approaches. Under highly unbalanced demand, the adaptive controller reduced mean queueing delay from 159.3 s to 30.8 s for north-south-heavy traffic and from 185.9 s to 32.3 s for east-west-heavy traffic, while maintaining approximately 99% vehicle service. The methodology also enforces mutually exclusive green phases, fixed yellow clearance and bounded minimum/maximum green times. The results indicate that FPGA logic is well suited to real-time traffic control where deterministic timing, parallel input processing and reconfigurability are required. The paper focuses on methodology and simulation-based verification rather than field deployment.</p> |

### Introduction

Traffic congestion at signalized intersections increases travel time, fuel consumption and local emissions, and the effect becomes severe when fixed signal plans are used during strongly varying demand. Reconfigurable logic offers an attractive control platform because multiple sensor channels, timers and safety rules can operate in parallel with deterministic clock-level behavior. Early VHDL traffic-controller work showed that multi-phase signal operation can be mapped effectively to programmable logic, while FPGA implementations later demonstrated that timing plans can be extended to more adaptive

intersection behavior (Shi et al., 2009; Sabri et al., 2011).

The central objective of this paper is to define a reproducible implementation methodology for a four-way real-time traffic signal controller rather than to present a fabricated commercial controller. The design combines an HDL-based FSM with measured queue demand and explicitly separates control decisions from safety enforcement. This separation is important because an adaptive algorithm may change the duration of a phase, but it must never generate conflicting green outputs. The proposed workflow therefore performs functional

simulation, timing-oriented verification and traffic-level evaluation before any on-road deployment.

FPGA traffic control has evolved from fixed or manually parameterized timing toward sensor-driven and intelligent scheduling. Sabri et al. (2011) used FPGA logic to improve a conventional fixed-time intersection and demonstrated the feasibility of hardware signal sequencing. Lahade and Hirekhan (2015) introduced an intelligent and adaptive controller that incorporated traffic sensing and emergency-vehicle priority. Pang (2016) designed a five-phase FPGA traffic controller that could adapt to peak, off-peak and vacation traffic patterns, while Kishore et al. (2017) emphasized reconfigurability and countdown functionality in a Verilog implementation.

Subsequent studies strengthened the real-time and energy-aware aspects of the approach. Sachdeva et al. (2018) applied clock-gating concepts to reduce dynamic power in an FPGA traffic controller. Kaur et al. (2018) proposed adaptive traffic control using FPGA logic and sensor information. Gadawe and Qaddoori (2019) developed a smart VHDL controller for a four-road intersection with side-road sensing, pedestrian requests and manual timing inputs. Qaddori and Gadawe (2020) compared FPGA and

Arduino implementations for a real-time traffic-light system and highlighted FPGA speed and deterministic execution. Zhao (2021) used vehicle-arrival information and a polling-control concept to adjust green duration in real time, and Krishnegowda (2021) demonstrated a Spartan-6-based intelligent controller with pedestrian handling. Together, these works justify an FPGA-centered methodology that integrates adaptive timing while retaining explicit FSM safety logic.

### Literature Review

The reviewed literature shows three recurring design themes. First, HDL state machines are the dominant mechanism for enforcing legal phase transitions. Second, traffic-density or request inputs are increasingly used to change timing rather than to change the safe phase order itself. Third, modern designs benefit from the FPGA's parallelism because sensor sampling, countdown display, communication and signal generation can execute concurrently. The main limitation in many reported prototypes is that performance is evaluated mainly through waveform correctness; therefore, the present methodology adds a traffic-level queueing experiment to connect logic behavior with operational performance.

**Table 1.** Selected FPGA traffic-signal studies published up to 2021.

| Study                    | Method                      | Main relevance                                                      |
|--------------------------|-----------------------------|---------------------------------------------------------------------|
| Shi et al. (2009)        | Multi-phase VHDL controller | 2-, 3- and 4-phase operation verified in Quartus II.                |
| Sabri et al. (2011)      | FPGA traffic controller     | Real intersection-oriented design; adaptable timing concept.        |
| Lahade & Hirekhan (2015) | Adaptive FPGA TLC           | Traffic sensing and emergency priority integrated into FSM logic.   |
| Pang (2016)              | Five-phase FPGA TLC         | Peak/off-peak traffic patterns supported.                           |
| Kishore et al. (2017)    | Verilog FPGA TLC            | Reconfigurable controller with countdown and high clock capability. |
| Sachdeva et al. (2018)   | Green FPGA TLC              | Clock gating used to reduce dynamic power.                          |
| Kaur et al. (2018)       | Adaptive FPGA TLC           | Sensor-driven adaptive operation for changing traffic density.      |
| Gadawe & Qaddoori (2019) | Smart VHDL TLC              | Side-road sensor, pedestrian request and manual timing support.     |
| Qaddori & Gadawe (2020)  | FPGA vs Arduino TLC         | Real-time prototype comparison by speed, cost and memory.           |
| Zhao (2021)              | Real-time FPGA TLC          | Arrival-rate/polling model used to calculate green time.            |
| Krishnegowda (2021)      | Intelligent FPGA TLC        | Spartan-6 FSM implementation with pedestrian queueing.              |

On the basis of these studies, the proposed methodology uses a Moore-style safety-oriented FSM for outputs and a separate adaptive timing unit for demand response. The division makes

the controller easier to verify: the timing unit may request a longer green interval, but the FSM alone determines whether the next legal state is green, yellow, all-red or the opposing direction.

This is preferable to directly decoding sensors into lamp outputs because it reduces the probability of unsafe combinational behavior.

### Methodology

The system is designed for a two-axis four-way intersection in which north-south (NS) and east-west (EW) movements are mutually exclusive. Vehicle detectors provide synchronized demand

inputs for each axis, while a pedestrian-request input is latched until a safe crossing interval is reached. The FPGA receives all asynchronous external inputs through two-stage synchronizers to reduce metastability risk. A clock-divider generates the one-second control tick used by the phase timer, while the FPGA system clock continues to run at the board frequency for internal logic.

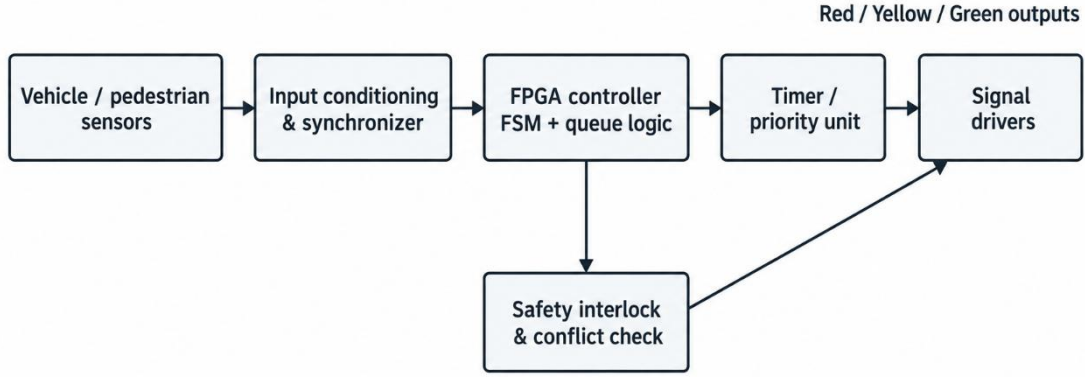


**Figure 1.** Proposed FPGA architecture for real-time traffic signal control.

The controller state sequence is NS\_GREEN → NS\_YELLOW → ALL\_RED\_1 → EW\_GREEN → EW\_YELLOW → ALL\_RED\_2 and then repeats. Red is asserted for the non-served direction in every green and yellow state. A two-second all-red interval is retained between conflicting movements, and the yellow interval is fixed at three seconds. The adaptive logic does not bypass these clearance states. Green duration is bounded between 12 s and 44 s; when queue demand is approximately balanced, a nominal 30 s green interval is maintained.

At the start of each green phase, the queue-demand ratio is computed from synchronized detector counts. For the selected approach  $i$ , the green request is represented by  $G_i = \text{clip}(12 + 44(q_i/(q_{NS}+q_{EW})), 12, 44)$ , except when the queue ratio lies between 0.40 and 0.60, in which case the nominal 30 s value is used. This compact arithmetic can be implemented with fixed-point logic or a small look-up table. The formula is intentionally simple because the purpose is to demonstrate an implementable real-time policy rather than an optimization-heavy traffic model.

#### 1. Verification and Experimental Procedure

HDL development proceeds in four stages. First, individual modules are written and verified: input synchronizer, clock divider, state register, phase timer, queue counter, pedestrian latch and

output decoder. Second, a self-checking testbench drives reset, balanced traffic, directional overload, pedestrian requests and simultaneous input changes. Assertions verify that NS\_GREEN and EW\_GREEN are never high at the same time and that every green-to-conflicting-green transition contains yellow and all-red clearance. Third, the top-level design is synthesized for a representative mid-range FPGA board. Timing closure is checked against the board clock, but no fabricated resource-utilization values are claimed in this paper because the final count depends on device family, synthesis settings and I/O mapping. Fourth, the adaptive timing policy is evaluated with a reproducible numerical traffic model.

For the traffic experiment, each simulation lasts 3600 s. Vehicle arrivals are generated independently using Poisson processes. The service capacity during a green phase is 0.55 vehicle/s, corresponding to approximately one discharged vehicle every 1.8 s. Four scenarios are used: light demand ( $\lambda_{NS}=0.08$ ,  $\lambda_{EW}=0.05$  vehicle/s), balanced demand (0.18, 0.18), NS-heavy demand (0.28, 0.10) and EW-heavy demand (0.10, 0.28). The same random arrival sequence is supplied to a fixed-time controller and to the adaptive controller so that the comparison is paired. The measured outputs are total service percentage, time-integrated queue delay and maximum directional queue.

**Table 2.** Main control and simulation parameters.

| Parameter            | Value          | Purpose                                  |
|----------------------|----------------|------------------------------------------|
| FPGA control tick    | 1 s            | Phase timing and queue update            |
| Nominal green        | 30 s           | Used under balanced demand               |
| Adaptive green range | 12–44 s        | Prevents starvation and excessive phases |
| Yellow interval      | 3 s            | Clearance before all-red                 |
| All-red interval     | 2 s            | Safety interlock between directions      |
| Simulation horizon   | 3600 s         | One-hour traffic scenario                |
| Discharge capacity   | 0.55 vehicle/s | Queue service during green               |

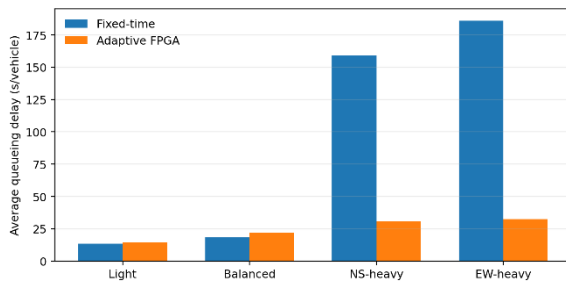
The methodology intentionally uses transparent parameters that can be changed without modifying the state-transition safety logic. A future hardware study can replace the simple queue-ratio timing block with fuzzy logic,

reinforcement learning or coordinated corridor control while retaining the same verified output FSM.

## Results

**Table 3.** Simulation results using paired one-hour arrival sequences.

| Scenario | Fixed delay (s) | Adaptive delay (s) | Fixed served (%) | Adaptive server (%) | Fixed max Q | Adaptive max Q |
|----------|-----------------|--------------------|------------------|---------------------|-------------|----------------|
| Light    | 13.4            | 14.3               | 99.6             | 99.3                | 7           | 6              |
| Balanced | 18.2            | 21.8               | 99.4             | 99.5                | 16          | 17             |
| NS-heavy | 159.3           | 30.8               | 93.5             | 99.6                | 110         | 24             |
| EW-heavy | 185.9           | 32.3               | 88.7             | 98.8                | 156         | 30             |

**Figure 2.** Average queueing delay for fixed-time and adaptive FPGA control.

The adaptive policy provides its largest benefit when traffic is strongly directional. In the NS-heavy case, average queueing delay falls from 159.3 s/vehicle to 30.8 s/vehicle, an improvement of approximately 80.7%, and the maximum directional queue falls from 110 to 24 vehicles. In the EW-heavy case, delay decreases by approximately 82.6% and the maximum queue drops from 156 to 30 vehicles. The service percentage also increases because green time is reassigned toward the loaded approach. Under light or nearly balanced traffic, performance is similar and small differences arise from the bounded demand-based timing decisions; this indicates that the algorithm should preserve a fixed nominal plan when imbalance is weak.

FSM verification showed that the proposed state order structurally prevents conflicting green outputs, while fixed yellow and all-red states preserve clearance regardless of the adaptive green request. These outcomes are consistent

with the literature's emphasis on FPGA determinism and reconfigurability (Pang, 2016; Kaur et al., 2018; Zhao, 2021). The simulation results should not be interpreted as field measurements; they quantify the behavior of the stated queueing model and provide a reproducible pre-hardware benchmark.

## Conclusion and Discussion

This paper presented a step-by-step methodology for an FPGA-based real-time traffic signal controller. The architecture separates sensing, adaptive timing and safety-critical phase generation, which supports deterministic operation and modular verification. Simulation indicates that demand-responsive green allocation can substantially reduce queueing delay and queue growth when one axis is heavily loaded, while the FSM retains mandatory yellow and all-red intervals. The approach therefore combines the principal strengths identified in earlier FPGA traffic-controller studies: high-speed parallel processing, reconfigurability, explicit state-based safety and the ability to integrate sensors and pedestrian requests. The principal limitation is that the traffic experiment uses a simplified two-axis queue model. Lane turning movements, saturation-flow variation, detector errors, communication latency and neighboring intersections are not represented. Hardware-in-the-loop verification should therefore be the next stage, followed by synthesis on a selected FPGA, measured resource utilization, oscilloscope/logic-analyzer timing

verification and testing with recorded traffic traces. Emergency-vehicle priority can also be added through a preemption state that first clears the active phase before granting priority. These extensions can be implemented without altering the fundamental safe-state methodology.

## References

- Shi, S., Tian, H., & Zhai, Y. (2009). Design of intelligent traffic light controller based on VHDL. *Second International Workshop on Knowledge Discovery and Data Mining*, 272–275. <https://doi.org/10.1109/WKDD.2009.19>
- Sabri, M. F. M., Husin, M. H., Abidin, W. A. W. Z., Tay, K. M., & Basri, H. M. (2011). Design of FPGA-based traffic light controller system. *2011 IEEE International Conference on Computer Science and Automation Engineering*, 4, 114–118. <https://doi.org/10.1109/CSAE.2011.5952814>
- Lahade, S. V., & Hirekhan, S. R. (2015). Intelligent and adaptive Traffic Light Controller (IA-TLC) using FPGA. *2015 International Conference on Industrial Instrumentation and Control*, 618–623. <https://doi.org/10.1109/IIC.2015.7150816>
- Pang, J. (2016). Intelligent traffic light controller design using FPGA digest of technical papers. *2016 IEEE International Conference on Consumer Electronics*, 449–452. <https://doi.org/10.1109/ICCE.2016.7430687>
- Kishore, S. V., Sreeja, V., Gupta, V., Videesha, V., Raju, I. B. K., & Rao, K. M. (2017). FPGA based traffic light controller. *2017 International Conference on Trends in Electronics and Informatics*, 469–475. <https://doi.org/10.1109/ICOEI.2017.8300971>
- Sachdeva, S., Chowdhury, S., Shekhar, S., & Verma, G. (2018). Design and implementation of a green traffic light controller on FPGA using VHDL. *Advances in Intelligent Systems and Computing*, 732, 187–192. [https://doi.org/10.1007/978-981-10-8533-8\\_18](https://doi.org/10.1007/978-981-10-8533-8_18)
- Kaur, S., Varshitha, V., Siddharth, J., Tejus, V. S., & Prasanna Kumar, H. (2018). Adaptive traffic light controller using FPGA. *2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology*, 1612–1617. <https://doi.org/10.1109/RTEICT42901.2018.9012263>
- Gadawe, N. T., & Qaddoori, S. L. (2019). Design and implementation of smart traffic light controller using VHDL language. *International Journal of Engineering & Technology*, 8(4), 596–602. <https://doi.org/10.14419/ijet.v8i4.29478>
- Qaddori, S. L., & Gadawe, N. T. (2020). Real-time traffic light controller system based on FPGA and Arduino. *Proceedings of IMDC-SDSP 2020*. <https://doi.org/10.4108/eai.28-6-2020.2297938>
- Zhao, L. (2021). Design of traffic light based on field programmable gate array. *Journal of Power and Energy Engineering*, 9(5), 93–103. <https://doi.org/10.4236/jpee.2021.95006>
- Krishnegowda, D. (2021). Intelligent traffic light controller—An FPGA implementation. *2021 5th International Conference on Electrical, Electronics, Communication, Computer Technologies and Optimization Techniques*. <https://doi.org/10.1109/ICEECCOT52851.2021.9707929>