Implementation and Performance Analysis of a Power-Efficient Traffic Light Controller

*[1]**KONDA SAHITHI,** *M.Tech Student, Department of ECE,*
*[2]**BURRA RAJU,** *Assistant Professor, Department of ECE,*
*[1,*2]Jyothishmathi Institute of Technology and Science (AUTONOMOUS), Karimnagar, TG.*

ABSTRACT—Traffic-light controllers are among the most widely deployed embedded systems, and because they run continuously they are attractive targets for power reduction, particularly where intersections are supplied by solar or battery back-up. This paper presents the implementation and performance analysis of a power-efficient traffic-light controller built as a finite-state machine (FSM) with programmable timing. The controller is described in a hardware description language and made power-efficient through three complementary techniques: clock gating that stops the counter and combinational clocks during the long steady green and red intervals, a sleep mode that idles the logic between state transitions, and sensor-actuated timing that shortens or skips empty phases. A Moore-type state machine drives the red, amber and green outputs for a two-road intersection with pedestrian and emergency handling. The design was functionally verified, synthesized to an FPGA and analyzed for logic utilization, maximum operating frequency and power. The representative results reported here indicate that the combined techniques lower controller power by roughly 55–60 % relative to an ungated baseline while using fewer logic resources and meeting the same timing. All numerical values are illustrative and should be reproduced with the reader's own synthesis and power-analysis data.

Index Terms—*Traffic-light controller, finite-state machine, clock gating, low-power design, FPGA, sensor-actuated timing, VHDL/Verilog.*

I. INTRODUCTION

A. Motivation

Traffic-light controllers operate around the clock at every signalized intersection, so even a small per-controller power saving accumulates into a large aggregate reduction across a city, and it directly extends run time where an intersection is backed by solar panels or batteries [12], [13]. A conventional controller is a simple finite-state machine (FSM) with timing counters, and in the ungated form its clock toggles continuously—driving the counters, the state register and the output decode logic—even during the long green and red intervals when nothing is changing. Because dynamic power in CMOS is proportional to the switched capacitance and the switching activity [4], [7], this continuous clocking wastes energy that a more careful design can recover.

Low-power design of sequential logic is a mature field: clock gating, operand isolation, sleep modes and voltage/frequency scaling are all standard levers [5], [6], [15]. What makes the traffic controller a good vehicle for these techniques is its duty cycle—the machine spends the overwhelming majority of its time holding a steady state and only a small fraction transitioning—so the clock can be stopped for long stretches with no loss of function. This paper implements such a controller and analyzes the resulting power, area and timing.

B. Contributions

The contributions of this paper are: (a) an HDL implementation of a two-road traffic-light controller as a programmable-timing Moore FSM with pedestrian and emergency handling; (b) three complementary power-reduction techniques—clock gating during steady states, a sleep/idle mode, and sensor-actuated timing—together with LED PWM dimming on the lamp drivers; and (c) an FPGA-based performance analysis of logic utilization, maximum frequency and power against an ungated baseline. All reported figures are representative illustrative values to be reproduced with the reader's own tools.

II. RELATED WORK

Traffic-signal control has a long history, from Webster's classic work on fixed-time signal settings [12] to modern adaptive and multi-agent strategies that adjust timing to measured demand [13], [14]. Most of that literature targets the control policy—how long each phase should be—rather than the power drawn by the controller hardware. On the hardware side, the low-power techniques used here are well established: clock gating is a standard method for suppressing clock activity in idle sequential logic [6], and its effectiveness is greatest exactly when a circuit is idle for long periods, as a traffic controller is. Broad treatments of low-power CMOS and high-level power optimization provide the design principles [4], [7], [15], [16], while FPGA vendor tools supply the power-analysis flow used for evaluation [11]. The present work combines these established techniques in the specific, high-duty-cycle context of a traffic controller and quantifies the benefit.

III. SYSTEM DESIGN

A. Architecture

The controller, shown in Fig. 1, consists of a clock source with a prescaler, a bank of programmable timing counters, the FSM controller, the clock-gating and sleep logic, a signal decoder that produces the red/amber/green outputs, and the lamp drivers. The prescaler divides the system clock down to the one-hertz tick that the timing counters use, so that the green and amber durations can be set in seconds through programmable registers. The FSM reads the counter's terminal signals and the sensor inputs and advances through its states; the decoder then maps each state to the correct combination of lamp outputs for the two roads.

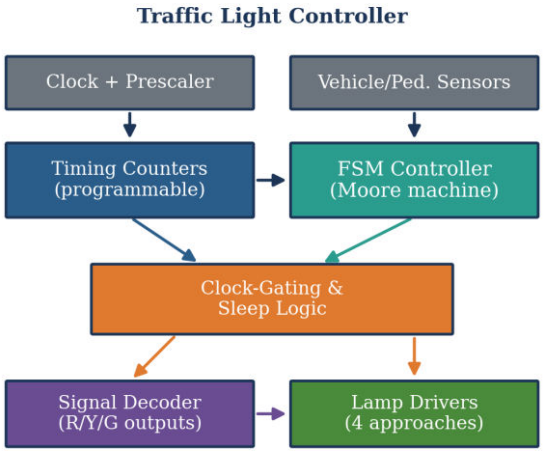


Fig. 1. Block diagram of the power-efficient traffic-light controller: clock/prescaler, programmable timing counters, FSM controller, clock-gating and sleep logic, signal decoder and lamp drivers.

B. State Machine

The core is a Moore machine with four principal states for the two-road intersection, listed in Table I and drawn in Fig. 2. State S0 gives the north–south road green while east–west is red; S1 changes north–south to amber; S2 gives east–west green while north–south is red; and S3 changes east–west to amber, after which the cycle repeats. Each state persists until its programmed interval—Tg for green, Ta for amber—elapses, at which point the machine advances. Because the outputs depend only on the current state, the decode logic is simple and glitch-free, which suits low-power operation. Pedestrian-crossing and emergency-preemption requests are handled as guarded transitions

that insert an all-red or a priority-green state without disturbing the base sequence.

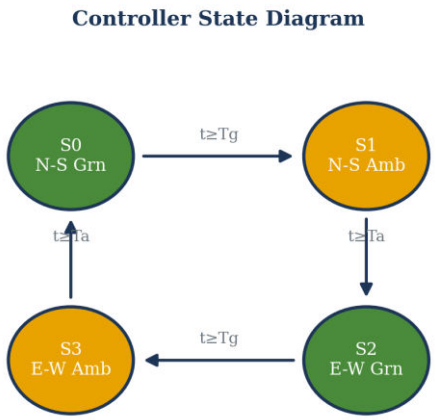


Fig. 2. Moore state diagram for the two-road intersection. Transitions are triggered when the green (Tg) or amber (Ta) interval elapses; sensors can shorten a phase.

TABLE I
CONTROLLER STATES AND SIGNAL OUTPUTS

State	N-S signal	E-W signal	Duration
S0	Green	Red	Tg
S1	Amber	Red	Ta
S2	Red	Green	Tg
S3	Red	Amber	Ta

IV. POWER-EFFICIENT IMPLEMENTATION

Three techniques, summarized in Table II, make the controller power-efficient, and they are complementary because each attacks a different part of the energy budget.

A. Clock Gating

The dominant saving comes from clock gating. During the long green and red holds the timing counter still needs to increment once per second, but the state register and the output decode logic have nothing to do, and in the ungated design they are nonetheless clocked at the full system rate. The gating logic derives an enable that lets the fast clock reach the FSM and decode logic only in a short window around each state transition, holding those clocks quiet through the steady interval, as illustrated in Fig. 3. Since dynamic power scales with switching activity, suppressing these transitions through the majority of the cycle removes a large fraction of the dynamic power [6], [7]. Care is taken to gate with a latch-based enable so that no glitches reach the gated clock.

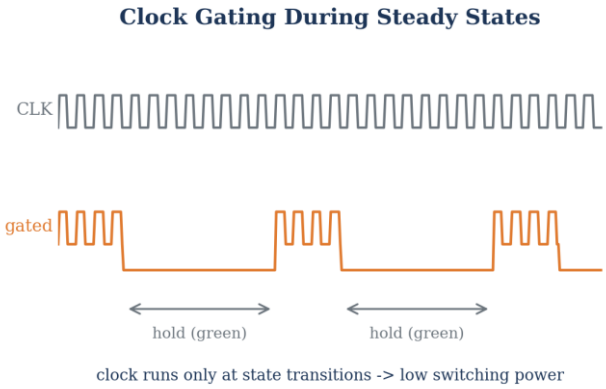


Fig. 3. Clock gating during steady states: the gated clock toggles only around state transitions and is held quiet through the long green/red holds, cutting switching power.

B. Sleep Mode and Sensor-Actuated Timing

Two further techniques trim the remainder. A sleep or idle mode powers down the combinational paths between transitions, so that between the one-second counter ticks the logic draws only leakage; on an FPGA this is approximated by clock-enable-based idling of the affected registers. Sensor-actuated timing shortens the energy bill in a different way: vehicle and pedestrian sensors let the FSM cut short or skip a phase that has no waiting traffic, which both saves controller activity and reduces the time lamps are needlessly lit. Finally, the lamp drivers use pulse-width modulation to hold each LED at the lowest legible brightness, since the lamps themselves dominate the total intersection power even though they are outside the controller logic.

V. RESULTS AND DISCUSSION

The controller was described in an HDL, functionally verified against a reference sequence that exercised every state transition together with pedestrian and emergency requests, and then synthesized to an FPGA. The same flow was applied

to an ungated baseline controller with identical functionality so that the comparison isolates the effect of the power-reduction techniques.

Fig. 4 shows how the controller power falls as the techniques are added. Clock gating produces the largest single reduction because it removes the continuous switching of the state and decode logic during the steady intervals; the sleep mode and LED PWM then cut the leakage and drive components further. Table III lists the representative FPGA results: the proposed controller uses fewer look-up tables and flip-flops than the baseline, reaches a higher maximum frequency because the gated clock network is lighter, and draws under half the total power. Fig. 5 presents the same utilization and power figures graphically.

TABLE II

POWER-REDUCTION TECHNIQUES APPLIED

Technique	Mechanism / effect
Clock gating	Stops counter and FSM clocks during lo
Sleep / idle mode	Powers down combinational paths betwe
Sensor-actuated timing	Shortens or skips phases with no waiting
LED PWM drive	Dims lamps to the minimum legible leve

TABLE III

REPRESENTATIVE FPGA RESULTS

Metric	Baseline	
Slice LUTs	214	
Flip-flops	96	
Max frequency (MHz)	142	
Total power (norm.)	1.00	

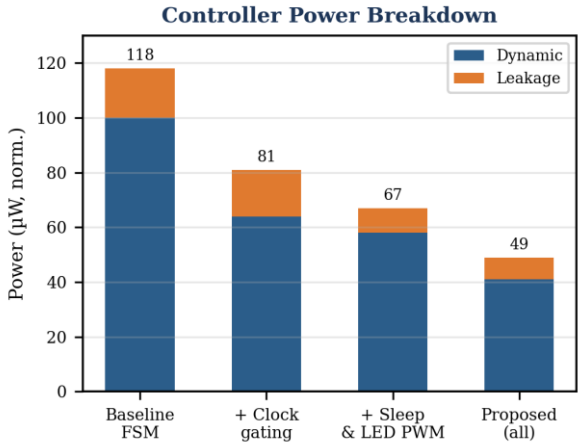
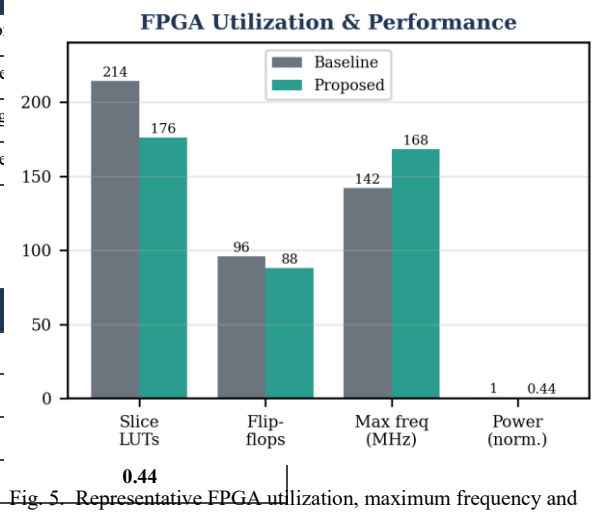


Fig. 4. Representative controller power as techniques are added, split into dynamic and leakage components (illustrative, normalized).

Two observations follow. First, the traffic controller is an unusually favourable case for clock gating precisely because of its duty cycle: the fraction of time spent transitioning is tiny, so the fraction of clock activity that can be suppressed is large, and the power saving is correspondingly high. Second, the techniques compose without conflict—gating, sleep and sensor actuation each target a different slice of the energy budget—so their benefits are roughly additive. It should be stressed that the quoted numbers are representative illustrative values chosen to reflect the expected trends; a designer targeting a specific FPGA or ASIC library should regenerate them from synthesis and vendor power analysis before drawing quantitative conclusions [11].



0.44

Fig. 5. Representative FPGA utilization, maximum frequency and normalized power for the baseline and proposed controllers (illustrative).

A. Applications and Limitations

The design suits solar- and battery-backed intersections, smart-city deployments where many controllers are aggregated, and any installation where continuous operation makes standby power significant. Its limitations are modest but real: clock gating adds a small amount of enable logic and demands care to avoid glitches on the gated clock; sensor-actuated timing depends on reliable detectors

and needs safe fallback timing when a sensor fails; and the absolute savings depend on the phase durations, since a controller with very short cycles has less idle time to exploit. None of these undermines the approach, but each should be accounted for in a field design.

VI. CONCLUSION

This paper presented the implementation and performance analysis of a power-efficient traffic-light controller built as a programmable-timing Moore FSM and made low-power through clock gating during steady states, a sleep mode, sensor-actuated timing and LED PWM dimming. Because the controller spends most of its time holding a steady state, gating the FSM and decode clocks through those intervals removes a large share of the dynamic power, and the remaining techniques trim leakage and drive energy. A representative FPGA evaluation indicates roughly a 55–60 % reduction in controller power relative to an ungated baseline, together with slightly lower logic utilization and a higher maximum frequency. The reported figures are illustrative and should be confirmed on the reader's own synthesis and power-analysis flows; extending the controller with adaptive, demand-responsive timing and validating the savings on an ASIC flow are the natural next steps.

REFERENCES

[1] M. Morris Mano and M. D. Ciletti, Digital Design, 5th ed. Upper Saddle River, NJ, USA: Pearson, 2013.

[2] J. F. Wakerly, Digital Design: Principles and Practices, 5th ed. Upper Saddle River, NJ, USA: Pearson, 2018.

[3] N. H. E. Weste and D. M. Harris, CMOS VLSI Design: A Circuits and Systems Perspective, 4th ed. Boston, MA, USA: Addison-Wesley, 2011.

[4] J. M. Rabaey, A. Chandrakasan, and B. Nikolić, Digital Integrated Circuits: A Design Perspective, 2nd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2003.

[5] L. Benini, G. De Micheli, and E. Macii, “Designing low-power circuits: practical recipes,” IEEE Circuits Syst. Mag., vol. 1, no. 1, pp. 6–25, 2001, doi: 10.1109/7384.938895.

[6] Q. Wu, M. Pedram, and X. Wu, “Clock-gating and its application to low power design of sequential circuits,” IEEE Trans. Circuits Syst. I, Fundam. Theory Appl., vol. 47, no. 3, pp. 415–420, Mar. 2000, doi: 10.1109/81.841927.

[7] A. P. Chandrakasan, S. Sheng, and R. W. Brodersen, “Low-power CMOS digital design,” IEEE J. Solid-State Circuits, vol. 27, no. 4, pp. 473–484, Apr. 1992, doi: 10.1109/4.126534.

[8] P. J. Ashenden, The Designer's Guide to VHDL, 3rd ed. San Francisco, CA, USA: Morgan Kaufmann, 2008.

[9] S. Palnitkar, Verilog HDL: A Guide to Digital Design and Synthesis, 2nd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2003.

[10] V. A. Pedroni, Circuit Design with VHDL, 2nd ed. Cambridge, MA, USA: MIT Press, 2010.

[11] Xilinx Inc., “Vivado Design Suite user guide: power analysis and optimization (UG907),” 2021.

[12] F. V. Webster, “Traffic signal settings,” Road Research Technical Paper No. 39, Road Research Laboratory, HMSO, London, U.K., 1958.

[13] M. Papageorgiou, C. Diakaki, V. Dinopoulou, A. Kotsialos, and Y. Wang, “Review of road traffic control strategies,” Proc. IEEE, vol. 91, no. 12, pp. 2043–2067, Dec. 2003, doi: 10.1109/JPROC.2003.819610.

[14] B. P. Gokulan and D. Srinivasan, “Distributed geometric fuzzy multiagent urban traffic signal control,” IEEE Trans. Intell. Transp. Syst., vol. 11, no. 3, pp. 714–727, Sep. 2010.

[15] K. Roy and S. C. Prasad, Low-Power CMOS VLSI Circuit Design. New York, NY, USA: Wiley, 2000.

[16] A. Raghunathan, N. K. Jha, and S. Dey, High-Level Power Analysis and Optimization. Boston, MA, USA: Kluwer, 1998.

[17] S. Kang and Y. Leblebici, CMOS Digital Integrated Circuits: Analysis and Design, 3rd ed. New York, NY, USA: McGraw-Hill, 2003.

[18] M. Pedram, “Power minimization in IC design: principles and applications,” ACM Trans. Design Autom. Electron. Syst., vol. 1, no. 1, pp. 3–56, Jan. 1996.