

FSM-Based Digital Design and RTL Synthesis of a Traffic Light Controller Using Verilog HDL and Cadence EDA Tools

Sudhanshu S. Humaney

Dept. of VLSI Design & Technology

Dr. Babasaheb Ambedkar Technological University
Lonere, Maharashtra, 402103

Dr. Pallavi P. Ingale

Dept. of Electronics & Telecommunication

Dr. Babasaheb Ambedkar Technological University
Lonere, Maharashtra, 402103

Amruta V. Umap

Dept. of VLSI Design & Technology

Dr. Babasaheb Ambedkar Technological University
Lonere, Maharashtra, 402103

Dr. Narendra S. Jadhav

Dept. of Electronics & Telecommunication

Dr. Babasaheb Ambedkar Technological University
Lonere, Maharashtra, 402103

Abstract: Effective traffic management systems demand efficient, deterministic, and fixed-latency control systems for road safety and efficient flow of traffic. Hardware description logic-based traffic controllers outperform software-controlled systems in terms of deterministic performance and latency. In this paper, the finite state machine (FSM)-based traffic light controller for a four-way intersection using Verilog HDL description and synthesis will be demonstrated. The FSM-based traffic light controller will comprise the Moore FSM model with the counter-based timer for efficient traffic signal control. The functions described in the Verilog HDL code will achieve the desired FSM modules and functional verification with the help of the Cadence nelaunch environment. The waveform analysis will then be performed for successful function transitions and signal switching. The Cadence Genus Synthesis Solution will then execute the full-chip RTL synthesis under the worst-case process model. The results after the synthesis will illustrate the excellent hardware utilization efficiency, power consumption, and the large positive timing margin for the FSM-based traffic light controller with the Moore machine model. The work will find applications in embedded systems design and obtain maximum advantages from the complete digital RTL to gate-level digital VLSI design flow.

Keywords— Traffic Light Controller, Finite State Machine, RTL Design, Verilog HDL, Cadence Genus, Digital VLSI, Synthesis Analysis

I. INTRODUCTION

The development of urban infrastructure has led to a remarkable increase in the density of vehicles, making the need for effective traffic control systems imperative. Traffic signal controllers are a basic requirement for smart transport infrastructure, helping to control traffic flow at intersections.

Conventional traffic signal controls are often developed using micro-controllers or programmable logic controllers. Though flexible, such systems are highly dependent on software processing and sequential processing, which might lead to

uncertainties in processing times as well as increased power dissipation. In the context of safety-related applications like traffic signal controls, deterministic logic assumes prime importance [7], [8].

Digital hardware-based controllers provide an effective alternative because of the ability to execute in parallel, predictability in terms of timing, and robustness. Finite State Machines are commonly used for control-oriented digital systems because they offer a systematic way of showing the sequence of actions using a fixed number of states and state transitions. The problem of traffic light control is inherently periodic in nature and thus suitable for the finite state machines approach [5], [7].

In this paper, an FSM-based traffic light control system is modeled and developed as an RTL-level design in Verilog HDL. A whole design cycle involving functional verification with Cadence nelaunch and synthesis with Cadence Genus is discussed. This work focuses on synthesis-level analysis such as timing analysis, area analysis, and power analysis and does not go further to layout level as in most existing literature [1], [2], [4].

II. LITERATURE REVIEW

The literature has described a number of studies pertaining to traffic light control systems. In the past, electro-mechanical timers and simple digital logic were used to implement fixed time controls for traffic lights. Additionally, because microcontrollers are easily programmable, the development of the embedded systems concept led to their use in traffic control systems [7].

When compared to software-controlled traffic systems, FPGA-based traffic control systems created with HDLs have demonstrated superior performance and efficacy. Because FSAs in FPGAs can execute functions or tasks in parallel, they can process tasks with shorter execution times and predictable performance [2], [3], [6].

Recently, there has been interest in using intelligent traffic control systems with sensors and wireless communication systems along with artificial intelligence for adjusting the signal time depending on the density of traffic formed [2], [7].

Despite the improvement in traffic efficiency made possible by adaptive systems, they add new complexities in terms of design, power, and external sensing infrastructure. There has been immense research that has been conducted, largely in simulating models for behavior or prototyping, with lesser relevance to analysis of RTL synthesis with commercial EDA tools [1], [5].

To address this, this research further proposes a systematic method for implementing an RTL with an FSM-centered architecture based on synthesis with the Cadence Genus tool and discusses its results and performance. There is consistency between the relevance of this research regarding the current focus of VLSI design work on timing closure, area, and power issues as well [1], [9].

III. SYSTEM ARCHITECTURE

The proposed controller will manage a four-way traffic light intersection of north, south, east, and west directions. Each direction is assigned with three signal indications: red, yellow, and green. The overall architecture of the system is partitioned into the following blocks of functionality:

- Clock generation and asynchronous reset unit
- FSMCU (Finite State Machine Control Unit)
- Counter-based timing logic
- Logic for decoding the output

The control unit FSM drives the sequence of the states of the traffic light signals. Similarly, a set of logical functions ensures that each phase is on for an adequate time. Output decoding logic will translate the FSM state into the corresponding traffic light signals: The architecture gives the guarantee that only one direction is on at any given time and it never guarantees conflicting movements [5], [6].

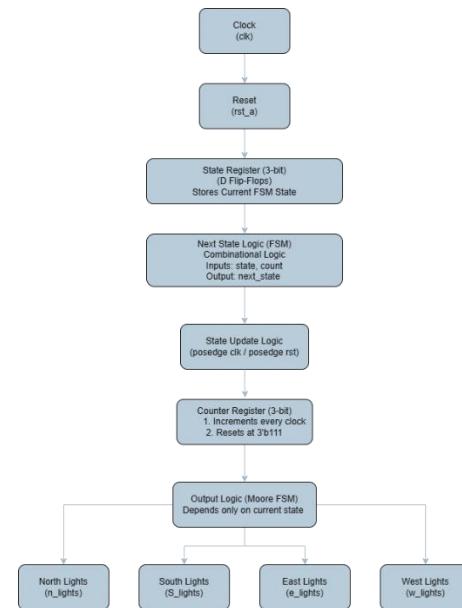


Fig. 1. Block diagram of the FSM-based traffic light controller

IV. FSM DESIGN METHODOLOGY

The controller of a traffic light is implemented as a Moore-type FSM: its output signals depend only on the current state [5], [8]. Such a design decision simplifies significantly the output logic and makes it glitch-free because the outputs are not directly dependent on the inputs. In this FSM there are eight states, including four green and four corresponding yellow states, one pair in each direction:

The state transitions in it are synchronized with the rising edge of the clock. It follows a timing mechanism with a counter that decides on the transitions. More time is allocated for the green states, during which vehicles can move, and the yellow states act like transitional intervals to warn the driver before changing the signal. The FSM is initialized asynchronously to an active-high reset into a safe default state to ensure that the startup behavior will be predictable [4], [6].

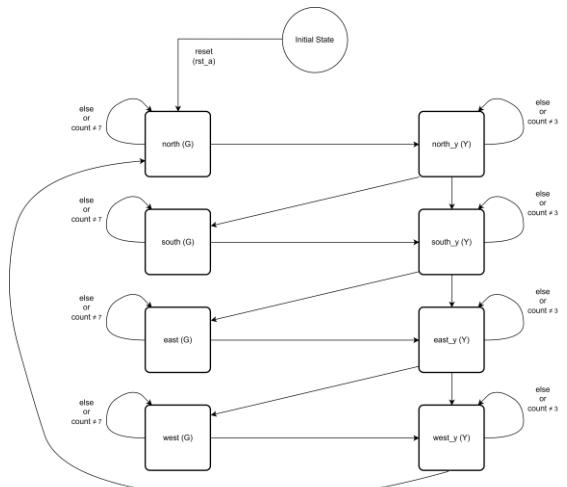


Fig. 2. Block diagram of the FSM-based traffic light controller

Current State	Duration	Next State	Active Direction
North Green	8 cycles	North Yellow	North
North Yellow	4 cycles	South Green	North
South Green	8 cycles	South Yellow	South
South Yellow	4 cycles	East Green	South
East Green	8 cycles	East Yellow	East
East Yellow	4 cycles	West Green	East
West Green	8 cycles	West Yellow	West
West Yellow	4 cycles	North Green	West

The state transition should be cyclic, which will grant every direction a fair and orderly flow of traffic. Similar FSM sequencing strategies for traffic signal control have been reported in prior IEEE studies [5], [6].

V. FUNCTIONAL VERIFICATION USING CADENCE NCLAUNCH

This is the most important functionality before synthesis. Cadence nclaunch provides a whole environment for simulating digital designs, which can be used for functional verification of RTL. Here, there is a dedicated testbench that generates a periodic clock and applies an asynchronous reset during the first few hundred nanoseconds into the simulation.

The simulation runs through several traffic cycles to observe steady-state behavior, ensuring a long-term stable output. Cadence SimVision waveform analysis for correct initialization and proper yellow phase sequencing; mutual exclusivity of traffic signals verifies that there are no illegal states or green signal overlaps in extended simulation runs [8], [9].

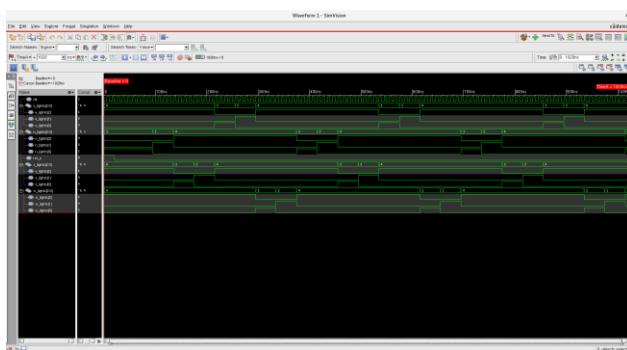


Fig. 3. Simulation waveform captured using Cadence SimVision showing clock, reset, and traffic light outputs for all directions

VI. RTL SYNTHESIS WITH CADENCE GENUS

After successful functional verification, the RTL design is synthesized using the Cadence Genus Synthesis Solution. During the synthesis phase, the design is exercised with a slow process corner and a standard-cell timing library to

analyze the worst-case performance [1], [9]. A clock constraint of 10ns is used.

The synthesis step transfers the RTL description onto a gate-level netlist that uses flip-flops, comb logic gates, and clocking elements. There are no timing violations during this step, and the design has a large positive timing margin. The resultant gate-level schematic graphic representation verifies the FSM hardwired implementation [1], [4].

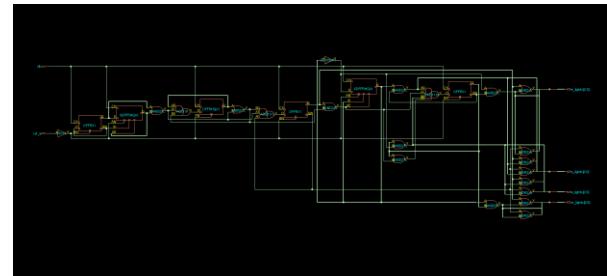


Fig. 4. Gate-level schematic for the traffic light controller using the Cadence Genus Simulator

VII. RESULTS AND PERFORMANCE ANALYSIS

A. Timing Analysis

The post-synthesis timing analysis reveals a critical path delay of 1.294 ns for the worst-case conditions. The design has a clock period of 10 ns with a worst negative slack value of +8.706 ns. This reflects the excellent timing capabilities of the design [1], [9].

B. Area Analysis

The optimized design comprises 27 standard cells with the total cell area of 205.877 units. The area is largely dominated by sequential components. It is expected because the control design is based on the FSM model [5], [6].

C. Power Analysis

The total power consumed has been calculated to be 16.44 μ W. Register power contributes largely because of the clock, while combinational logic and clock contribute less to the total power. Hence, the design is suitable for low-power design [1], [8].

Parameter	Value
Clock Period	10 ns
Critical Path Delay	1.294 ns
Worst Negative Slack	+8.706 ns
Total Cell Count	27
Total Cell Area	205.877
Total Power	16.44 μ W

VIII.DISCUSION

The above-mentioned results from functional verification and RTL synthesis show that the proposed traffic light control system using an FSM can be made more reliable and

deterministic with less hardware cost. The simulation results from the waveform diagrams show the proper phase transition of the traffic signal and validate the Moore FSM design style in which the outputs do not depend on the inputs directly but depend on the current state only [5], [8].

From the synthesis perspective, the design shows a very positive timing margin under worst-case processes. In this case, the critical path of 1.294 ns is much lower than the applied clock period of 10 ns; therefore, the worst negative slack is +8.706 ns. This indicates that the designed controller is able to handle variability in processes, voltage rates, and temperatures without violating functional correctness with regard to performance specifications. Additionally, a very positive margin shows that the design can run with faster clocks depending on the system level [1], [9].

Area analysis for the design shows that the SPL-designed netlist has a small area because it occupies a mere 27 standard cells, and the main contribution to the area is from the sequential modules, which is very normal because it is an FSM-controlled control logic. The simplicity of the combinational logic Once again, it exemplifies the optimized state coding and transition strategy used in the design.

The power analysis also supports that the proposed controller is suitable for embedded systems. The total power consumed by the controller is 16.44 μ W, which is dominated by register operations, especially clock switching. The power consumed by combinatorial logic and clock networks is relatively smaller. This is also as expected, as it indicates that further optimizations can be done to minimize power, which can include clock gating [1], [3].

In particular, synthesis results confirm that FSM-based traffic control logic can be efficiently implemented with RTL design styles and standard-cell synthesis techniques to meet competitive performance with low design complexity.

IX. DESIGN ASSUMPTIONS AND LIMITATIONS

In designing the traffic light controller, certain assumptions are made that ease the control process and help demonstrate clearly the use of FSM in synthesizing RTL. The traffic light controller works in fixed time values specified for both green and yellow light intervals. The fixed time values assumed in this design consider equal traffic flow in all directions [7].

The system does not use any input information from other sensors, like car sensors or pedestrian request signals. This means that there are no varying traffic conditions to which the controller could automatically adjust the timing of the traffic signals. Although this can be a problem in a busy city environment, it helps the system run deterministically.

Another disadvantage comes from being in the synthesis environment. The reported performance in terms of timing, area, and power consumption will be varying based on the choice of standard cell libraries as well as the process corner. There could be varying values based on technology nodes or

choice of standard cell libraries. There could be effects from interconnects approximated by using wire load models in synthesis [1], [9].

Although it has these shortcomings, it is a good base to implement traffic control using FSMs. With the simplified assumptions, it is possible to effectively judge RTL design and synthesis, and it can be used as an example to introduce improvements in the future.

X. CONCLUSION

This paper has provided an all-encompassing FSM-based digital system and RTL synthesis description involving a Verilog HDL implementation of a traffic light controller, which utilized industry-leading Cadence EDA software tools. The traffic light controller has been abstracted as a Moore FSM, which helps to ensure that its outputs are deterministic and glitch-free, an important aspect for a safety-related monitoring system, namely traffic light control [5], [8].

The functional verification done with Cadence nclaunch ensured that the designed FSM is correct in operation. The waveform analysis also ensured that the resetting is done in the correct sequence, in addition to ensuring that the yellow and green phases are done in the correct sequence. The exclusive nature of the traffic light signals in all directions was also confirmed [8], [9].

The RTL synthesis task performed by Cadence Genus was able to implement the design in a gate netlist representation using standard cell libraries, focusing on worst-case operating environments. Analysis of the designed architecture post-synthesis revealed an encouraging critical path of 1.294 ns with a clock period constraint of 10 ns, offering a substantially large positive slack value. This is a clear indication of the good performance of the designed architecture in spite of the detrimental effects of the process, voltage, and temperature factors. The area analysis revealed an optimal architecture with a predominance of sequential elements, justifying the nature of the FSM-controlled architectures. Additionally, the analysis revealed minimal power consumption in the architecture with a predominant share due to the involvement of clock-driven registers [1], [9].

"The significance of the results, obtained from this work, lies in the successful usability of the control logic based on the FSM approach by utilizing the RTL design methodology and analysis. The task deals with the entire design flow from the RTL to gate-level design without the need for physical design, and therefore fills the gap between the theoretical design of the FSM and VLSI design. The design proposed is helpful for reliable traffic light control design and is also an important learning aid for students learning modern VLSI design practices."

XI. FUTURE WORK

Although the designed traffic light controller displays strong functionality and efficient synthesis attributes, there are a number of improvements that can be made to widen the utilization scope of this controller. Real-time traffic sensing methods are among the prominent areas that need to be focused upon in the realms of improvement of this traffic control system. Vehicle detectors, pedestrian request signals, or cameras can be used to achieve this objective [2], [7].

Another promising extension could be implemented by the use of hardware prototyping in an FPGA platform. By implementing the synthesis design in an FPGA, it would be possible to validate the design in real time, calculate the time it takes to execute in its operating environment, as well as evaluate the power consumption based on the use of resources in the FPGA. This would also be an opportunity to validate the scalability of the design in handling more traffic or pedestrian crossings [2], [3].

Power optimization is a key area that still has room for improvement. The methods that could still be used, or rather could still be effectively applied, include clock gating, low power FSM state coding, as well as dynamic frequency scaling that could be used to suppress switching activities, especially in cases where these activities occur unnecessarily in sequential circuit components.

Future work may also involve the extension of the controller to facilitate the coordination of multiple crossings. This may also allow for the synchronization of traffic and the implementation of the green-wave effect, thus improving traffic flow. The extension of the design to intelligent transportation systems and intelligent infrastructures may also increase its relevance.

Finally, moving the design flow forward to the inclusion of physical implementation steps like floor planning, placement, and routing would allow a closer insight into the characteristics related to timing, power, and area. The results can then be compared at the synthesis and the post-layout levels, thus adding to the complete design flow for the VLSI design [], [9].

XII. REFERENCES

- [1] A. Devipriya, G. H., N. A. Reddy, V. Srinivas and R. D., **“RTL Design and Logic Synthesis of Traffic Light Controller for 45 nm Technology,”** 2022 3rd International Conference for Emerging Technology (INCET), 2022, pp. 1–5. DOI: <https://doi.org/10.1109/INCET54531.2022.9824833>
- [2] W. M. El-Medany and M. R. Hussain, **“FPGA-Based Advanced Real Traffic Light Controller System Design”**, Proceedings of the 2007 IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS), Dortmund, Germany, 2007, pp. 100–105. DOI: <https://doi.org/10.1109/IDAACS.2007.4488383>
- [3] S. Singh and S. C. Badwaik, **“Design and Implementation of FPGA-Based Adaptive Dynamic Traffic Light Controller”**, 2011 International Conference on Emerging Trends in Networks and Computer Communications (ETNCC), Udaipur, India, 2011, pp. 324–330. DOI: <https://doi.org/10.1109/ETNCC.2011.6255914>
- [4] M. Faizrizwan, M. Sabri, W. A. W. Z. Abidin, K. M. Tay and H. M. Basri, **“Design of FPGA-Based Traffic Light Controller System”**, 2011 IEEE International Conference on Computer Science and Automation Engineering (CSAE), Shanghai, China, 2011, pp. 114–118. DOI: <https://doi.org/10.1109/CSAE.2011.5952814>
- [5] K. S. Reddy and B. B. Shabarainath, **“Timing and Synchronization for Explicit FSM Based Traffic Light Controller”**, 2017 IEEE 7th International Advance Computing Conference (IACC), Hyderabad, India, 2017, pp. 526–529. DOI: <https://doi.org/10.1109/IACC.2017.0114>
- [6] B. Dilip, Y. Alekhy and P. Divya Bharathi, **“FPGA-Based Intelligent Traffic Light Controller with FSM Design”**, 2012 IEEE Radar, Communication and Computing Conference (RCC), Bangalore, India, 2012, pp. 92–97. DOI: <https://doi.org/10.1109/RCC.2012.6163881>
- [7] T. Han and C. Lin, **“Design of an Intelligent Traffic Light Controller Using HDL”**, Proceedings of the 2002 IEEE Region 10 Conference (TENCON), Beijing, China, vol. 3, 2002, pp. 1749–1752. DOI: <https://doi.org/10.1109/TENCON.2002.1182673>
- [8] M. Papageorgiou, H. Hadj-Salem and F. Middelham, **“Review of Road Traffic Control Strategies”**, Proceedings of the IEEE, vol. 91, no. 12, 2003, pp. 2043–2067. DOI: <https://doi.org/10.1109/JPROC.2003.819610>
- [9] IEEE Computer Society, **“IEEE Standard for Verilog Hardware Description Language”**, IEEE Std 1364-2005, 2006, pp. 1–706. DOI: <https://doi.org/10.1109/IEEESTD.2006.99495>