

Power-Optimized Verilog Based Architecture for Real Time Bus Ticketing System

Y. Mamillu

M.Tech, Assistant Professor, Department of Electronics and Communication Engineering, Seshadri Rao Gudlavalleru Engineering College Gudlavalleru, Andhra Pradesh, India

M. Jayantha Siva Srinivas, M. Sai Priya, M. Dinesh, M. Vani

Final Year Students, Department of Electronics and Communication Engineering Seshadri Rao Gudlavalleru Engineering College Gudlavalleru, Andhra Pradesh, India

Abstract - Efficient ticketing systems are essential to improve the speed and convenience of public transportation. Traditional manual ticketing methods often result in delays and inefficiency during passenger boarding. Automated ticketing systems implemented using hardware solutions can significantly reduce processing time and improve the operational efficiency. This study aims to design and implement a power-optimized automatic bus ticketing system using Verilog Hardware Description Language (HDL) and deploy it on a field programmable gate array (FPGA) platform. The proposed system was developed using Verilog HDL and implemented in a Xilinx Vivado design environment targeting an Artix-7 FPGA. The architecture consists of four main modules: Ticket selection, rupees calculation, return processing, and display interface. To reduce dynamic power consumption, flip-flop-based clock gating was incorporated into the design. Functional simulations are conducted in Vivado to verify the accuracy of each module. In addition, power, area, and timing analyses were conducted to compare the conventional architecture with the optimized clock-gated implementation. Simulation and synthesis results demonstrate that the clock-gated design achieves noticeable power reduction while maintaining efficient resource utilization and timing performance. The proposed FPGA-based automatic bus ticketing system provides a fast, reliable, and power-efficient solution suitable for real-time public transportation applications. This architecture also offers scalability to future intelligent fare-collection systems.

Keywords - Verilog HDL, Ticket Selection Module, Rupees Calculation Module, Return Processing Module, Display Interface, Clock Gating, Flipflop Based Clock Gating, Xilin Vivado.

I. INTRODUCTION

Urban transit systems require the fast and accurate collection of fares. Manual ticketing causes queuing delays, calculation errors, and high staffing demands, which intensify with the passenger volume. Hardware-driven automation removes these inefficiencies by directly implementing fare logic in programmable silicon. The Verilog Hardware Description Language (HDL) is an established language for register-transfer-level (RTL) design and is well-suited to FPGA-based implementations [10], [11].

Previous studies have addressed fair automation across multiple domains. Field programmable gate array (FPGA) and Internet of Things (IoT)-based architectures improve transaction speed and collection reliability [1], [2]. Travel-pattern studies derive service-planning insights from automated fare collection (AFC) records [3], [7]. Security, privacy, and hardware reliability analyses address the system trustworthiness [4], [5], [8]. Mobile ticketing platforms [6] and bill-acceptor hardware for rail transit [9] have extended this collection to alternative payment modes. Verilog-based vending machine designs [10], [11] offer the closest structural precedent; however, none of these studies have addressed power consumption.

As FPGA technology nodes shrink, the dynamic power increases, and the clock distribution network, which switches on every cycle regardless of active computation, becomes the dominant source. Flip-flop-based clock gating eliminates this waste by maintaining low clock time for any idle module. Registering the enable signal through a D flip-flop before the AND gate suppresses glitches, which would otherwise create spurious transitions in downstream registers and yield reliable savings without hold-time violations. No published bus ticketing design applies this technique to complete the fare-collection pipelines.

The proposed design addresses this gap by combining a complete Verilog HDL ticketing architecture with flip-flop-based clock gating in an Artix-7 field-programmable gate array (FPGA). The route selection, coin counting, fare validation, change-dispensing, and multiplexed display outputs were unified within a single RTL design. The outcome is a 34 % power reduction, a 6.8°C drop in junction temperature, an unchanged critical path of 5.610 ns, and an area overhead of one LUT.

Section II describes the materials and methods, including design tools, target hardware, and system architecture. Section III presents simulation results and a quantitative performance comparison. Section IV interprets the findings and identifies

the limitations of this study. Section V draws conclusions and outlines directions for future research.

II. MATERIALS AND METHODS

A. Materials

1) Verilog HDL

Gateway Design Automation developed Verilog in 1984, combining HiLo HDL constructs with the C language syntax. A commercial simulator reached the market in 1985, followed by a Verilog XL gate level simulator in 1987. After Cadence Design Systems acquired its gateway in 1990, it transferred the language to Open Verilog International to prevent industry-wide migration to VHDL. After being standardized as IEEE 1364, Verilog remains the dominant HDL for the RTL design and simulation.

2) Xilinx Vivado

Xilinx Vivado, now under AMD, is an integrated field-programmable gate array (FPGA) development environment that supports Verilog and VHDL. The tool unifies synthesis, implementation, bitstream generation, and device programming in a single workflow. The IP Integrator, built-in functional simulator, and detailed timing and resource utilization reports have accelerated design and verification. Vivado is a standard development platform for academic and industrial field-programmable gate array (FPGA) design centers worldwide.

3) Artix-7 FPGA

Built on 28-nm High-K Metal Gate technology, the Artix-7 family targets low-power, cost-sensitive applications. Devices range from 15,000 to 215,000 logic cells and incorporate digital signal processing (DSP) slices, Block RAM, high-speed transceivers, and clock management resources. The low power consumption, sufficient logic capacity, and competitive costs make Artix-7 a target platform for this ticketing system.

4) Flip-Flop-Based Clock Gating

The clock distribution network is the primary source of on-chip dynamic power because it switches on every clock edge, even when no computation is active. Clock gating reduces this cost by disabling the clock to inactive logic. As illustrated in Fig. 1, the flip-flop-based implementation registers the enable signal through a D flip-flop on the rising clock edge and then drives one terminal of an AND gate to produce the gated clock. Registering the enable before the gate suppresses glitches and prevents spurious transitions in the downstream registers, thereby ensuring reliable power savings without hold-time violations.

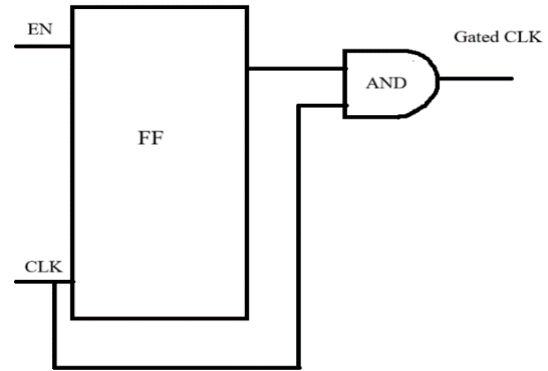


Fig. 1. Block diagram of flip-flop-based clock gating (D flip-flop and AND gate).

B. Methods: System Architecture

This ticketing system accepts 10-rupee, 5-rupee, and 3-rupee coins. The design comprises four functional modules: Ticket Selection, Rupees Calculation, Return Processing, and Display Interface, together with a Flip-Flop Clock module and a top-level integration file. When a passenger selects a route, specifies the ticket quantity, inserts coins, the system computes the fare, verifies the payment, calculates the change, and updates the display. The Flip-Flop Clock module distributes a gated clock to all the modules, suppressing the switching activity in any module that is currently idle.

(a) Ticket Selection Module

The Ticket Selection Module, whose block diagram is shown in Fig. 2, records passengers' route and quantity selections. The decoder accepts two path inputs (path_1 and path_2) together with two quantity inputs (qua_1 and qua_2), producing a PATH [1:0] route bus, PIN [1:0] quantity signal, and PRI [3:0] fare value. A synchronous reset returns all outputs to their default states. PATH [1:0], PIN [1:0], and PRI [3:0] propagated directly to the downstream calculation modules.

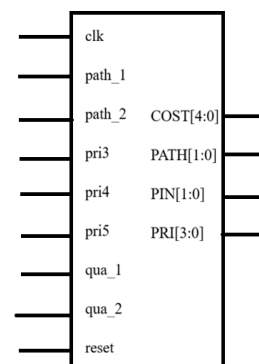


Fig. 2. Block diagram of the Ticket Selection Module.

(b) Rupees Calculation Module

The Rupees Calculation Module depicted in Fig. 3 counts the inserted coins. TEN_IN detects 10-rupee coin insertions and increments counter TEN_OUT [3:0] and FIVE_IN detects 5-rupee insertions and increments FIVE_OUT [3:0]. OUT [3:0] accumulates the weighted running total. A synchronous reset clears all counters and each counter advances on the active clock edge following each insertion event. The return-processing module reads OUT [3:0] for comparison.

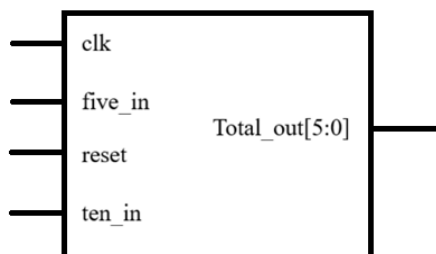


Fig. 3. Block diagram of the Rupees Calculation Module.

(c) Return Processing Module

The return-processing module illustrated in Fig. 4 verifies the payment and calculates the change. PATH_IN, COST_IN, and COIN_IN drive the internal comparator to block ticket issuances when COIN_IN falls below COST_IN. When the deposited amount reaches the fare, the module asserts a change [5:0] with an outstanding balance, raises valid_tic [5:0] to authenticate the ticket, and sets disp_tic to signal transaction completion. The gated clock drives all the operations to ensure fault-tolerant execution.

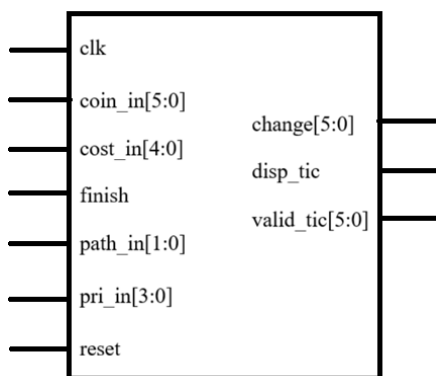


Fig. 4. Block diagram of the Return Processing Module.

(d) Display Interface Module

The Display Interface Module shown in Fig. 5 drives a six-digit, seven-segment display through time-division multiplexing. The route, quantity, fare, deposited amount, and change cycle across the six display positions were in rapid succession. Three subcomponents realize this function: a Mod-6 counter, a 6-to-1 selector, and a binary-coded decimal (BCD)-to-seven-segment decoder.

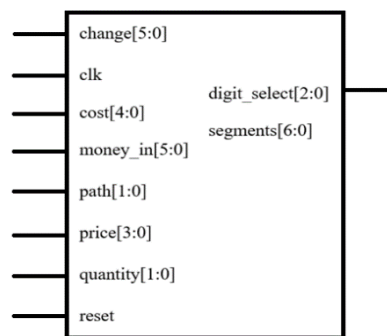


Fig. 5. Top-level block diagram of the Display Interface Module.

1)Mod-6 Counter (Fig. 6): The Mod-6 counter is a synchronous three-bit circuit that advances through states 000 to 101. After reaching state 101, it asserts the carry signal and resets it to 000 at subsequent clock edges. This six-state cycle governs the display scanning rate and digit-address sequence.

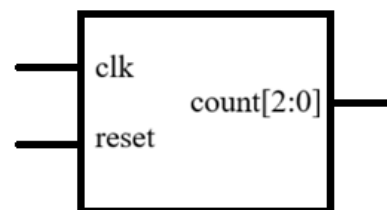


Fig. 6. Block diagram of the Mod-6 Counter.

2)6-to-1 Selector (Fig. 7): The 6-to-1 selector uses the Mod-6 counter output to route one of the six data inputs to the display driver during each time slot. As the switching rate exceeds the persistence threshold of human vision, all six-digit positions appear to simultaneously lit to the viewer. Sharing one display path among all the six digits considerably reduces the circuit area.

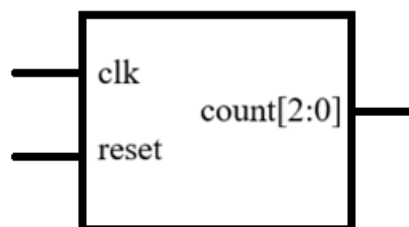


Fig. 7. Block diagram of the 6-to-1 Selector.

3)BCD-to-7-Segment Decoder (Fig. 8): The decoder maps each four-bit BCD input to a seven-bit segment pattern that activates the corresponding digit (0-9) on the seven-segment display. Invalid BCD codes were suppressed in the hardware to prevent erroneous output. Fig. 9 presents a detailed top-level view of the complete Display Interface Module.

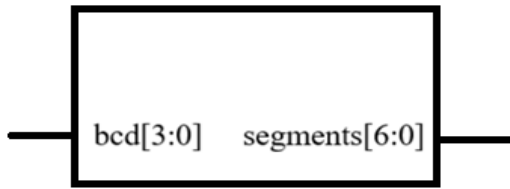


Fig. 8. Block diagram of the BCD-to-7-segment decoder.

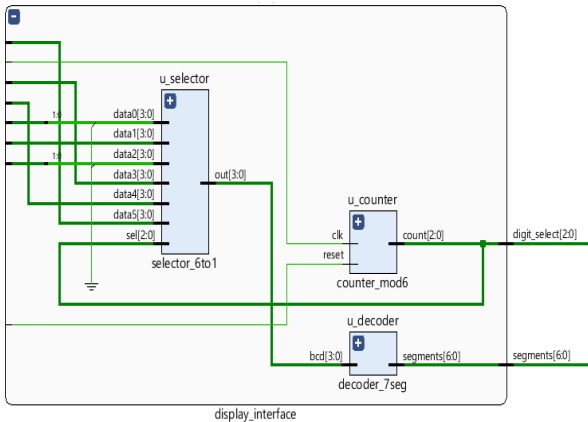


Fig. 9. Detailed top-level block diagram of the Display Interface Module.

(e) Flip-Flop Clock Module

The Flip-Flop Clock Module (Fig. 10) derives two signals from the raw clock input: gated_clk, held low whenever the enable is deasserted, and q_out, the registered flip-flop state. Distributing only gated_clk to all downstream modules ensures that idle logic produces no switching power.

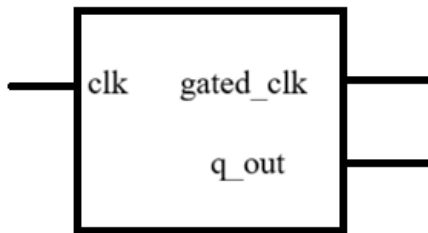


Fig. 10. Block diagram of the Flip-Flop Clock Module.

(f) Top-Level Module

The top-level module (Fig. 11) connects all five functional blocks, routes each inter-module signal, and distributes gated_clk throughout the design hierarchy. The module serves as the sole entry point for synthesis and simulation.

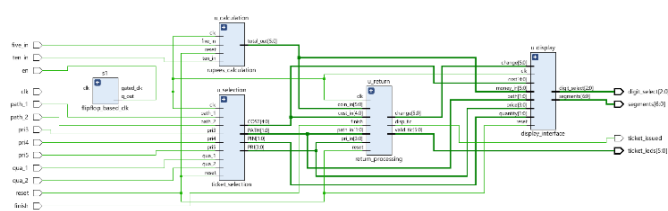


Fig. 11. Block diagram of the Top-Level Module.

III. RESULTS

Each module was independently simulated in Xilinx Vivado before the end-to-end integration testing to confirm the correctness of the full system.

1) Ticket Selection Module

Fig. 12 shows the waveform of the ticket selection module. PATH [1:0] responds correctly when path_1 or path_2 is asserted, PIN [1:0] tracks the quantity inputs, and PRI [3:0] reflects the expected fare for every route-quantity combination. All outputs are clear within one clock cycle upon reset assertion, as specified.

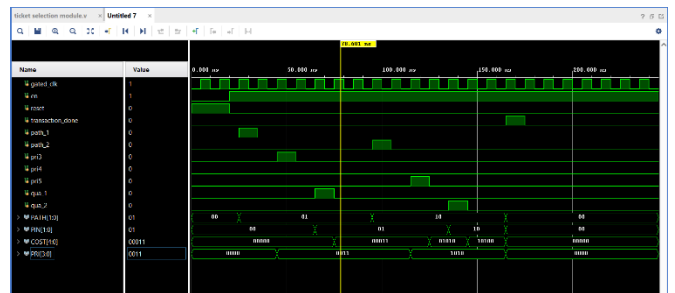


Fig. 12. Simulation waveform of the Ticket Selection Module.

2) Rupees Calculation Module

Fig. 13 illustrates the waveform of the Rupees Calculation Module. Five five_out [3:0] and ten_out [3:0] increments on every active clock edge following coin insertion and total_out [5:0] accumulated the weighted sum in real time. All counters were clear about reset assertions. The waveform matched the RTL specifications for each coin sequence tested.

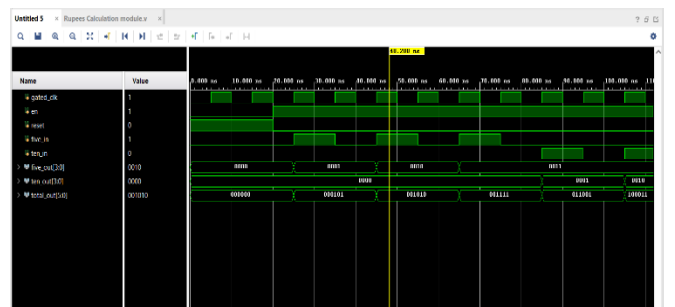


Fig. 13. Simulation waveform of the Rupees Calculation Module.

3) Return Processing Module

Fig. 14 shows the waveform of the return processing module. path_in, pri_in, and cost_in response to route and fare changes as specified. Once coin_in reaches cost_in, change[5:0] reflects the exact remaining balance, valid_tic[5:0] asserts ticket authentication, and disp_tic signals transaction completion. The module withholds the ticket issuance until coin_in meets or exceeds cost_in, as required by the specification.

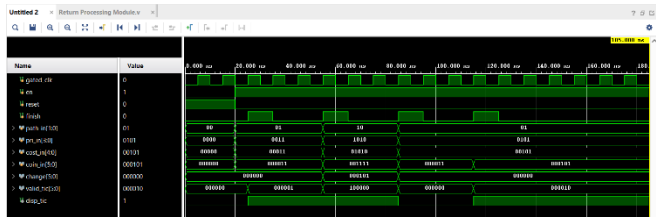


Fig. 14. Simulation waveform of the Return Processing Module.

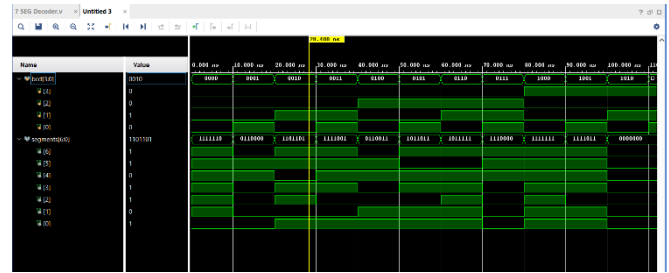


Fig. 18. Simulation waveform of the BCD-to-7-segment decoder.

4) Display Interface Module

Fig. 15 shows the top-level Display Interface waveform: digit_select advances through all six positions in sequence, and segments[6:0] produce the expected character at each step. Fig. 16 shows the Mod-6 counter transitioning from states 101 to 000 without glitches. Fig. 17 confirms the 6-to-1 selector behavior: sel = 000 routes data0 (0001) and sel = 001 routes data1 (0010) as specified. Fig. 18 shows the clean-segment transitions for all BCD inputs from 0 to 9.

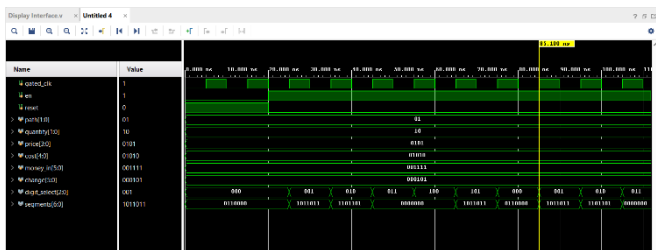


Fig. 15. Simulation waveform of the Display Interface Module (top level).

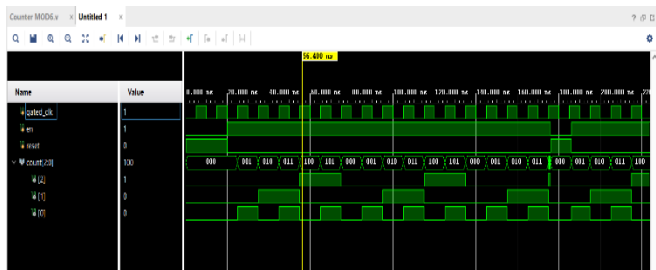


Fig. 16. Simulation waveform of the Mod-6 Counter.

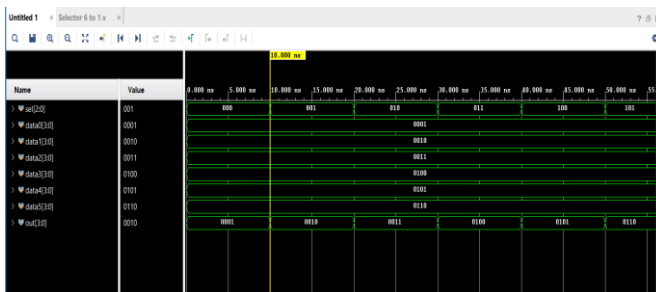


Fig. 17. Simulation waveform of the 6-to-1 Selector.

5) Top-Level Simulation

Fig. 19 shows the waveform of the integrated system. The reset simultaneously clears all the registers. Route and quantity selections propagate through the fare pipeline within a clock cycle. digit_select, and segments[6:0] drive the display accurately under the gated_clk. ticket_leds[5:0] and ticket_issued assert their output upon valid payments. The waveform confirmed end-to-end functional correctness.

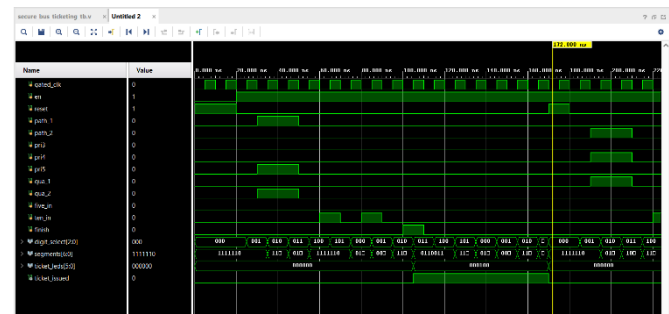


Fig. 19. Top-level simulation waveform of the integrated ticketing system.

IV. PERFORMANCE COMPARISON

Table I compares the existing design (without clock gating) with the proposed design (with flip-flop-based clock gating) in terms of the power, area, and timing metrics derived from the Xilinx Vivado synthesis and implementation reports for the Artix-7 FPGA.

TABLE I. Performance Comparison: Existing vs. Proposed Design

Parameter	Existing System	Proposed System
Area (LUTs)	44	45
Total Power (W)	10.525	6.904
Static Power (W)	0.175	0.155
Dynamic Power (W)	10.350	6.749
Delay (ns)	5.610	5.610
Net Delay (ns)	2.430	2.430
Logic Delay (ns)	3.181	3.181

i) Power

Table I shows a 34% drop in the total on-chip power from 10.525 W to 6.904 W. Figs. 20 and 21 present the power analysis reports for the existing and proposed designs, respectively. The dynamic power constituted approximately 98% of the total power consumption in both designs with I/O switching as the dominant contributor. By eliminating the toggle activity in idle logic, clock gating directly reduces this dominant source. The junction temperature fell from 44.7°C to 37.9°C. The static power remained marginal for both the designs (0.175 W vs. 0.155 W).

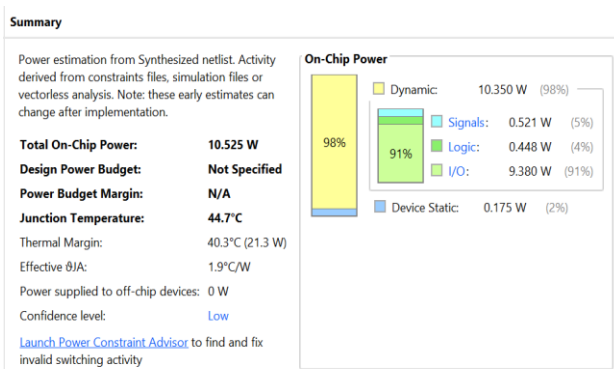


Fig. 20. Power report of the existing system (without clock gating).

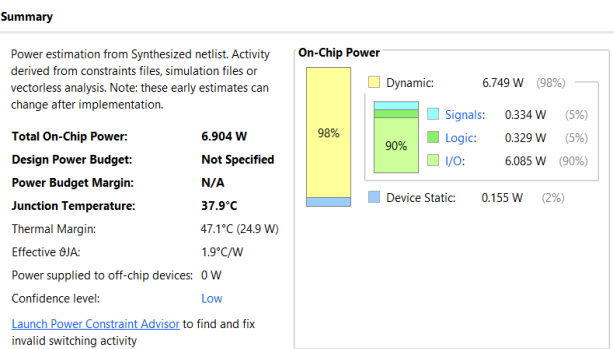


Fig. 21. Power report of the proposed system (with flip-flop clock gating).

ii) Area

Figs. 22 and 23 show the area-utilization reports for both designs. Incorporating the flipflop_based_clk block increased the Slice LUT count from 44 to 45, Slice Register count from 33 to 34, I/O block (IOB) count from 29 to 30, and global clock buffer (BUFGCTRL) count from one to two. All other submodules retained identical resource counts. The total area overhead is one LUT, which is a negligible cost relative to a power reduction of 34 %.

Name	Slice LUTs (134600)	Slice Registers (269200)	Bonded IOB (400)	BUFGCTRL (32)
bus_ticketing_system_top	44	33	29	1
u_calculation (rupees_calculation)	7	6	0	0
u_display (display_interface)	11	3	0	0
u_counter (counter_mod6)	11	3	0	0
u_return (return_processing)	0	11	0	0
u_selection (ticket_selection)	26	13	0	0

Fig. 22. Area utilization report of the existing system.

Name	Slice LUTs (134600)	Slice Registers (269200)	Bonded IOB (400)	BUFGCTRL (32)
bus_ticketing_system_top_EXT	45	34	30	2
s1 (flipflop_based_clk)	0	1	0	0
u_calculation (rupees_calculation)	7	6	0	0
u_display (display_interface)	11	3	0	0
u_return (return_processing)	0	11	0	0
u_selection (ticket_selection)	26	13	0	0

Fig. 23. Area utilization report of the proposed system.

iii) Timing

Figs. 24 and 25 present the timing reports for both the designs. The critical path from register u_selection/PRI_reg to the segments [] output measured 5.610 ns in both implementations, comprising a logic delay of 3.181 ns and routing delay of 2.430 ns. Display paths from u_display to digit_select [] measured 3.52 ns in both cases. Clock gating left all timing parameters unchanged.

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Require
Path 1	∞	5	5	9	u_selection_l_reg[0]/C	segments[2]	5.610	3.181	2.430	
Path 2	∞	5	5	7	u_selection_l_reg[1]/C	segments[0]	5.601	3.181	2.421	
Path 3	∞	5	5	7	u_selection_l_reg[1]/C	segments[1]	5.601	3.181	2.421	
Path 4	∞	5	5	9	u_selection_l_reg[0]/C	segments[3]	5.566	3.191	2.376	
Path 5	∞	5	5	9	u_selection_l_reg[0]/C	segments[4]	5.566	3.191	2.376	
Path 6	∞	5	5	9	u_selection_l_reg[0]/C	segments[5]	5.566	3.191	2.376	
Path 7	∞	5	5	9	u_selection_l_reg[0]/C	segments[6]	5.556	3.181	2.376	
Path 8	∞	2	2	12	u_display/_t_reg[0]/C	digit_select[0]	3.521	2.876	0.646	
Path 9	∞	2	2	11	u_display/_t_reg[1]/C	digit_select[1]	3.521	2.876	0.646	
Path 10	∞	2	2	9	u_display/_t_reg[2]/C	digit_select[2]	3.521	2.876	0.646	

Fig. 24. Timing report of the existing system.

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock
Path 1	∞	5	5	9	u_selection/PRI_reg[0]/C	segments[2]	5.610	3.181	2.430	∞	
Path 2	∞	5	5	7	u_selection/PRI_reg[1]/C	segments[0]	5.601	3.181	2.421	∞	
Path 3	∞	5	5	7	u_selection/PRI_reg[1]/C	segments[1]	5.601	3.181	2.421	∞	
Path 4	∞	5	5	9	u_selection/PRI_reg[0]/C	segments[3]	5.566	3.191	2.376	∞	
Path 5	∞	5	5	9	u_selection/PRI_reg[0]/C	segments[4]	5.566	3.191	2.376	∞	
Path 6	∞	5	5	9	u_selection/PRI_reg[0]/C	segments[5]	5.566	3.191	2.376	∞	
Path 7	∞	5	5	9	u_selection/PRI_reg[0]/C	segments[6]	5.556	3.181	2.376	∞	
Path 8	∞	2	2	12	u_display/_t_reg[0]/C	digit_select[0]	3.521	2.876	0.646	∞	
Path 9	∞	2	2	11	u_display/_t_reg[1]/C	digit_select[1]	3.521	2.876	0.646	∞	
Path 10	∞	2	2	9	u_display/_t_reg[2]/C	digit_select[2]	3.521	2.876	0.646	∞	

Fig. 25. Timing report of the proposed system.

V. DISCUSSION

Flip-flop-based clock gating achieved 34% power reduction with only one-LUT overhead and no change to the 5.610 ns critical path, demonstrating that the technique is both effective and resource-efficient for FPGA-based fare-collection systems. Preserving the critical path confirms that the design maintains real-time responsiveness and is ready for deployment in live-transit environments.

No prior study in this domain combined complete fare automation with hardware-level power management. FPGA fare systems [1], [2] and mobile ticketing platforms [6] meet functional objectives but omit power optimization. Verilog-based vending machine designs [10], [11] share structural similarities yet include no power analysis. Bill acceptor hardware [9] targets a peripheral rather than a complete fare pipeline. The proposed design surpasses each of the previous method by combining automation and clock gating.

Modular architecture scales without restructuring core pipeline. Extending the path-decode logic in the ticket selection module increases route capacity. Expanding the counters in the Rupees Calculation module accommodates additional coin denomination. Return processing, Display Interface, and clock-gating modules require no modifications in either case.

Two limitations apply to the current implementation. First, the design supports only two routes and three coin denominations (ten, five, and three rupees). Second, the correctness was established through functional simulations rather than physical board testing. Future studies should address these limitations in the future.

VI. CONCLUSIONS

A Verilog HDL-based automatic bus ticketing system was designed, simulated, and synthesized on an Artix-7 FPGA, and the results were verified using Xilinx Vivado. The four-module architecture (Ticket Selection, Rupees Calculation, Return Processing, and Display Interface) automates the complete fare-transaction cycle. Flip-flop-based clock gating reduced the total on-chip power by 34% (10.525 to 6.904 W), lowered the junction temperature from 44.7°C to 37.9°C, and added one LUT to the design, while the critical path was held at 5.610 ns. The functional simulation validated the correct operation under all test conditions.

Hardware clock management is a practical low-overhead strategy for power-constrained embedded transit systems. Future work should expand the route and denomination support, incorporate contactless payments, and validate the design of physical FPGA hardware for live transit deployment.

ACKNOWLEDGMENT

The authors thank Mr. Y. Mamillu for project guidance and continuous laboratory access between January and March 2025. This research received no external funding and was conducted as academic coursework. The simulation data supporting the findings of this study were generated in Xilinx Vivado, and are available from the corresponding author upon reasonable request.

REFERENCES

- [1] S. Aditya, S. Khaleel Ahmed, P. Nikhith Sree Sai, and D. Durga Prasad, "Verilog-Based Automatic Bus Ticketing System," in Proc. Int. Conf. Automation and Computation (AUTOCOM), Vijayawada, India, 2024.
- [2] K. S. Gill, A. Sharma, V. Anand, and S. Gupta, "Automated Fare Collection System for Public Transport using Intelligent IoT based System," in Proc. ICECONF, Chennai, India, 2023.
- [3] Y. Chen, Y. Zhao, and K. L. Tsui, "Clustering-based Travel Pattern Recognition in Rail Transportation System Using Automated Fare Collection Data," in Proc. PHM-Qingdao, Qingdao, China, 2019.
- [4] Z. Liu, S. Liu, and T. Chen, "Metro automatic fare collection system safety and security," in Proc. ICEAA, Verona, Italy, 2017, pp. 604-607.
- [5] A. P. Isern-Deya, A. Vives-Guasch, M. Mut-Puigserver, M. Payeras-Capella, and J. Castella-Roca, "A Secure Automatic Fare Collection System for Time-Based or Distance-Based Services with Revocable Anonymity for Users," *The Computer Journal*, vol. 56, no. 10, pp. 1198-1215, Oct. 2013.
- [6] D. G. Chandra, R. Prakash, and S. Lamdharia, "Mobile Ticketing System for Automatic Fare Collection Model for Public Transport," in Proc. CICN, Mathura, India, 2013, pp. 600-603.
- [7] Q. Tu, X. Zhang, and Y. Zhang, "Evaluation Method of Passenger Pressure State in Metro Hub based on AFC Data," in Proc. IEEE ITSC, Auckland, New Zealand, 2019, pp. 86-91.
- [8] N. J. Meyerhoff and J. S. Koziol, "Reliability of Fare Collection Hardware in Subways," in Proc. Annual Reliability and Maintainability Symp., San Francisco, CA, USA, 1984, pp. 33-38.
- [9] Z. Rong, C. Xiangxian, T. Zhifeng, and B. Jidong, "Design of Bill Acceptor for Automatic Fare Collection of Rail Transit," in Proc. Enterprise Systems Conf., Shanghai, China, 2014, pp. 179-183.
- [10] R. Sati, V. Mishra, and G. Verma, "Simulation of Vending Machine Design using Verilog HDL," in Proc. ASIANCON, Ravet, India, 2022.
- [11] K. Sathiyapriya, P. Nirmala, M. Deepika, J. Revanth, M. Saravanan, and M. Mohankumar, "Depiction of FPGA Based Vending Machine Using Mealy Model," in Proc. ICSTCEE, Bengaluru, India, 2022.