Design and Implementation of the Secure Glow System on an Artix-7 FPGA

Vishnumolakala Raghuveer Dept. of Electronics and Communication Engineering Vasireddy Venkatadri University
Guntur,Andhra Pradesh, India raghuveervishnumolakala@gmail.com

Shaik Aamina Ashfaq Dept. of Electronics and Communication Engineering RGUKT IIIT Ongole Andhra Pradesh, India
aaminaashfaq824@gmail.com
M. Yasodha Dept. of Electronics and Communication Engineering RGUKT IIIT Ongole Andhra Pradesh, India
yasodhamandala7@gmail.com
Shaik Farheena Shifa Dept. of Electronics and Communication Engineering RGUKT IIIT RK Valley Andhra Pradesh, India shaikfarheenashifa@gmail.com

Abstract

The proliferation of Internet of Things (IoT) devices has catalyzed the demand for sophisticated smart home systems that enhance security, convenience, and energy management. However, conventional microcontroller-based solutions often face performance bottlenecks due to their inherent sequential processing capabilities, which can introduce latency in critical real-time applications. This paper presents ”Secure Glow,” a novel hardware-accelerated home automation system designed to overcome these limitations. The system is built on a hybrid architecture that synergistically combines a NodeMCU (ESP8266) microcontroller with a powerful Artix-7 Field-Programmable Gate Array (FPGA). This dual-platform approach delegates high-level user interface tasks, such as keypad input processing, to the microcontroller, while leveraging the FPGA for computationally intensive, parallel processing tasks.

The core functionalities of Secure Glow are twofold: a robust, secure access control mechanism and an intelligent, presence- activated lighting system. The security module features a 4x4 matrix keypad for passcode entry, with authentication logic managed by the NodeMCU. Upon successful verification, a command is transmitted via UART to the Artix-7 FPGA, which then actuates a servo motor to control a door lock. The system incorporates a security lockout feature to thwart brute-force attempts. The energy- management module utilizes a passive Infrared (IR) sensor to detect human occupancy, enabling the FPGA to automatically control room lighting. This ensures that energy is consumed only when necessary, significantly reducing power wastage. The entire control logic for peripheral management, including the servo motor, IR sensor, and an LCD feedback display, is described in Verilog HDL and synthesized for the FPGA using the Xilinx Vivado Design Suite. The implementation demonstrates the superiority of FPGAs in handling multiple concurrent operations with deterministic timing, resulting in a highly responsive, secure, and energy-efficient smart home solution that is both scalable and reconfigurable for future enhancements.

Index Terms

Field-Programmable Gate Array (FPGA), Home Automation, Smart Security, Verilog HDL, Artix-7, Energy Efficiency, Internet of Things (IoT), Embedded Systems, NodeMCU, Real-Time Control, Hardware Acceleration, UART, Servo Control.

I. INTRODUCTION

A. Motivation

HE rapid advancement of embedded systems and the Internet of Things (IoT) has transformed the concept of a home from a static living space into a dynamic, interconnected environment. Modern smart homes aim to provide inhabitants with increased comfort, convenience, security, and energy efficiency [7]. This has led to a surge in the development of automated systems for controlling lighting, access, and environmental conditions. While many

solutions exist, there remains a critical need
for systems that are not only intelligent but also highly reliable and responsive, particularly for security-critical applications.

B. *Problem Statement*

A significant portion of current home automation systems is built upon microcontrollers (MCUs) due to their low cost and ease of programming [3], [16]. However, MCUs are fundamentally sequential processors. In a complex smart home environment where multiple sensors must be monitored and several actuators must be controlled simultaneously, an MCU-based system can suffer from latency. For instance, a delay in processing a security breach or controlling a door lock is unacceptable. This performance bottleneck highlights the need for a processing paradigm that can handle concurrent tasks with deterministic, real-time performance.

C. *Proposed Solution and Contributions*

To address these limitations, this paper introduces the **Secure Glow** system, a novel home automation solution that leverages the parallel processing power of a Field-Programmable Gate Array (FPGA). Our system is built on a hybrid architecture that combines a NodeMCU microcontroller for user interaction with an Artix-7 FPGA for core system control. This design capitalizes on the strengths of both platforms: the simplicity of the MCU for handling user inputs and the high-speed, parallel processing capability of the FPGA for managing real-time control tasks.
The primary contributions of this work are:

- The design and implementation of a hybrid MCU-FPGA architecture for a smart home system.
- The development of a secure, responsive access control system with keypad authentication and servo-driven lock actuation.
- The integration of an energy-efficient, automated lighting system based on real-time motion detection.
- A demonstration of the superiority of FPGAs for applications requiring deterministic timing and concurrent peripheral management.

D. *Paper Organization*

The remainder of this paper is organized as follows. Section II provides a survey of related work in the field of home automation. Section III details the architecture and components of the Secure Glow system. Section IV explains the methodology and implementation flow. Section V describes the Verilog HDL implementation. Section VI presents and discusses the results. Finally, Section VII concludes the paper and suggests directions for future work.

II. LITERATURE SURVEY

The field of home automation has been explored using various technologies, each with distinct advantages and limitations. Early systems often relied on wired protocols, but modern research predominantly focuses on wireless, more flexible solutions. These can be broadly categorized into microcontroller-based, IoT-centric, and FPGA-based systems.
Microcontroller-based systems, often using platforms like Arduino or ESP8266 (NodeMCU), are popular due to their low cost and extensive community support [3], [16], [24]. These systems excel at simple, sequential tasks. For example, Pawar et al. [3] developed an IoT-based security system using an MCU to handle sensor data and alerts. While effective, such systems may struggle to scale when required to manage numerous real-time tasks concurrently without an operating system, which can introduce its own overhead.
IoT-centric solutions, frequently built on single-board computers like the Raspberry Pi, emphasize connectivity and remote access [13], [20]. These systems leverage standard IP-based communication (Wi-Fi, Ethernet) to allow users to monitor and control their homes from anywhere. Kodali et al. [20] demonstrate such a system that integrates various sensors and relays them to a web server. The primary limitation of these systems is their reliance on software running on a general-purpose operating system, which can lack the deterministic, low-latency response required for critical security functions.
FPGA-based systems offer a hardware-centric approach that provides true parallelism and deterministic timing [1],

[2], [8], [10]. By implementing logic directly in hardware, FPGAs can perform multiple operations in the same clock cycle, making them ideal for real-time control. Al-Khateeb et al. [1] presented an FPGA-based security system that demonstrated significantly faster response times compared to MCU equivalents. Mohanty et al. [8] further explored this by designing a smart security system on an FPGA, highlighting its reliability and speed. While powerful, designing directly in HDL can be more complex than software programming.

Our Secure Glow system creates a synthesis of these approaches. It uses an MCU for the high-level, non-time-critical task of user input, while offloading all real-time control and parallel processing to the FPGA, thereby achieving the benefits of both platforms. Table I provides a comparative analysis of these different approaches.

TABLE I
COMPARISON OF HOME AUTOMATION SYSTEM APPROACHES

Reference	Core Technology	Key Features	Advantage	Limitation
Pawar et al. [3]	Arduino (MCU)	IoT-based security, detection	Low cost, Easy Gasto program	Sequential processing, Potential for latency
Kodali et al. [20]	Raspberry Pi (SBC)	IoT connectivity, Remote control	High connectivity, Software flexibility	Non-deterministic OS, Software overhead
Mohanty et al. [8]	Spartan-3E FPGA	Smart security, Real-time processing	Parallel processing, High speed, Deterministic	Higher design complexity (HDL)
Secure Glow (This Work)	**Artix-7 FPGA + NodeMCU**	**Hybrid control, Secure access, Automated lighting**	**Parallelism, Real-time response, Modular design**	**Slightly higher initial complexity than pure MCU**

III. SYSTEM ARCHITECTURE

The architecture of the Secure Glow system is founded on a hybrid, distributed-intelligence model. This design delegates specific tasks to the most suitable hardware platform, creating a system that is both efficient and robust. The architecture is composed of a front-end processing unit (NodeMCU) and a core control unit (Artix-7 FPGA), which communicate via a serial interface.

A. Overall System Design

The conceptual design of the system is illustrated in the block diagram in Figure 1. The NodeMCU acts as the primary user interface, capturing and verifying the password from the 4x4 keypad. Upon successful authentication, it sends a trigger signal to the FPGA. The FPGA, serving as the central nervous system, takes this signal as a command to execute a series of parallel, real-time operations: controlling the servo motor for the lock, displaying a status message on the LCD, and monitoring the IR sensor to manage the room lighting. This division of labor ensures that the time-critical tasks are handled by the dedicated hardware logic of the FPGA, while the more flexible, user-facing tasks are managed by the microcontroller.

B. Hardware Components

1) Core Control Unit: Artix-7 FPGA: The heart of the Secure Glow system is the Xilinx Artix-7 FPGA, hosted on a Digilent Basys 3 board (Figure **??**). The Artix-7 was chosen for its sufficient logic resources, low power consumption, and high performance. Unlike a microprocessor, the FPGA does not execute a sequential program. Instead, it is configured with custom digital logic circuits described in Verilog HDL. This allows it to perform all its control functions—PWM signal generation for the servo, parallel data bus communication with the LCD, and continuous monitoring of the IR sensor—simultaneously and with deterministic timing, ensuring instantaneous response.

2) Front-End Processor: NodeMCU ESP8266: A NodeMCU board, featuring the ESP8266 Wi-Fi SoC, serves as the front-
end processor. Its primary role is to manage the user authentication process. It is programmed using the Arduino IDE, which simplifies the process of scanning the 4x4 matrix keypad to read the user's password entry. The firmware contains the logic to compare the entered sequence against a predefined correct password. This offloads the complexity of keypad scanning and string comparison from the FPGA, allowing the HDL code to remain focused on high-speed control.

3) Actuators and Sensors: The system's interaction with the physical world is managed by a set of peripherals connected
directly to the FPGA:

- **16x2 LCD Display:** A standard character LCD based on the HD44780 controller provides essential visual feedback. The FPGA drives the 8-bit data bus and control lines (RS, E) with precise timing to display a "Welcome" message, confirming successful access.
- **Servo Motor:** A standard SG90 micro servo emulates the physical door lock. The FPGA's servo control module generates a highly accurate Pulse Width Modulation (PWM) signal to control the servo's rotational angle, moving it between locked (0°) and unlocked (90°) positions.
- **Infrared (IR) Motion Sensor:** A passive infrared (PIR) sensor is used to detect human presence. When a person enters the room, the sensor's output goes high. The FPGA continuously monitors this signal and, when asserted, activates the connected light source.
- **Light Source (LED):** A simple LED serves as a proxy for the room's main lighting. It is driven directly by a GPIO pin on the FPGA, providing a visual indication of the automated lighting functionality.

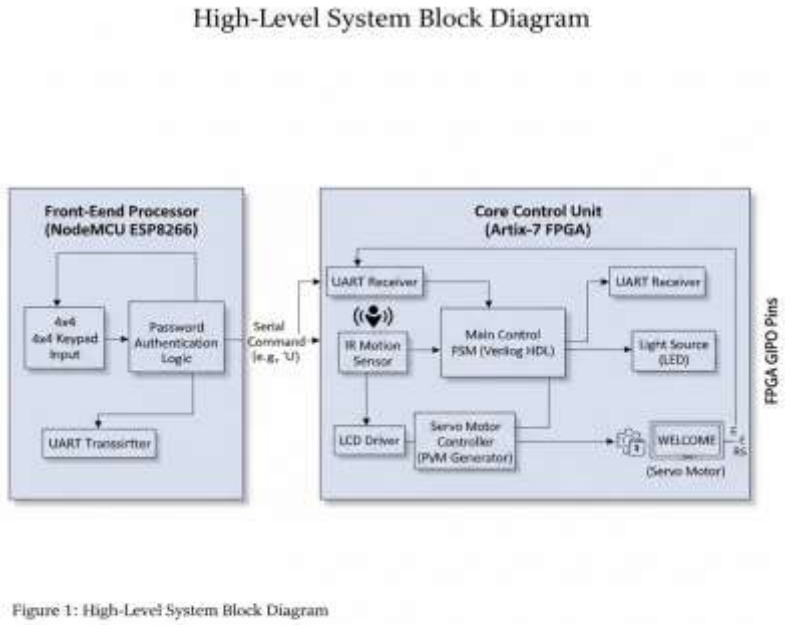

Figure 1: High-Level System Block Diagram

Fig. 1. High-Level System Block Diagram.

C. *Communication Protocol: UART Interface*

Communication between the NodeMCU and the Artix-7 FPGA is achieved using the Universal Asynchronous Receiver- Transmitter (UART) serial protocol. This choice was made due to its simplicity and minimal pin requirement (only requiring one transmit pin on the NodeMCU and one receive pin on the FPGA). After the NodeMCU validates the user's password, it sends a single-byte command (e.g., character '1') over the UART line. The FPGA has a dedicated UART receiver module implemented in Verilog that continuously listens for this byte. Upon receiving the correct command, it triggers the main FSM to begin the door unlocking and monitoring sequence. This one-way communication is sufficient for the system's needs and creates a reliable interface between the software-driven and hardware-driven parts of the architecture.

IV. METHODOLOGY AND SYSTEM IMPLEMENTATION

The implementation of the Secure Glow system was a multi-stage process involving parallel development of the microcon- troller firmware and the FPGA hardware description, followed by system integration and testing.

A. *Firmware Development (NodeMCU)*

The front-end logic was developed for the NodeMCU ESP8266 using the Arduino IDE and the C++ language. The firmware's primary responsibilities are:

- **Keypad Scanning:** The code implements a continuous scanning algorithm to detect key presses on the 4x4 matrix keypad. It debounces the inputs to prevent multiple false readings from a single press.
- **Password Buffering:** As the user presses keys, the corresponding characters are stored in a buffer.
- **Authentication Logic:** Once the entered password reaches the required length, it is compared against a securely hardcoded password string.
- **UART Transmission:** If the authentication is successful, the firmware sends a specific character (e.g., 'U' for Unlock) via the NodeMCU's hardware serial (UART) port to the FPGA. If unsuccessful, no action is taken, effectively resetting the input buffer for the next attempt.

This approach abstracts the complexity of the user interface away from the FPGA, simplifying the overall design.

B. *Hardware Description Language (HDL) Development*

The core control logic was designed in Verilog HDL using the Xilinx Vivado Design Suite. The design is centered around a main Finite State Machine (FSM) that orchestrates the behavior of all peripherals.

1) **Modular Design:** The project was broken down into distinct, reusable modules: a UART receiver, an LCD driver, a servo motor controller (PWM generator), and an IR sensor interface. This modularity allowed for individual simulation and verification of each component before integration.

2) **Finite State Machine (FSM):** The main control logic was modeled as an FSM, illustrated in Figure 2. The FSM transitions between states based on inputs from the UART receiver and the IR sensor.

- **IDLE_STATE:** The default state where the system waits for the unlock command from the NodeMCU. The servo is held at the 'locked' position.
- **UNLOCK_STATE:** Triggered by the UART command. In this state, the FPGA simultaneously sends commands to the LCD to display "Welcome" and changes the servo's PWM signal to move it to the 'unlocked' position. It also starts a 5-second timer.
- **AUTO_LOCK_STATE:** After the 5-second timer expires, the FSM transitions to this state, commanding the servo to return to the 'locked' position before returning to the IDLE_STATE.

The IR sensor logic operates in parallel to this FSM, activating the light whenever motion is detected, independent of the current lock state.

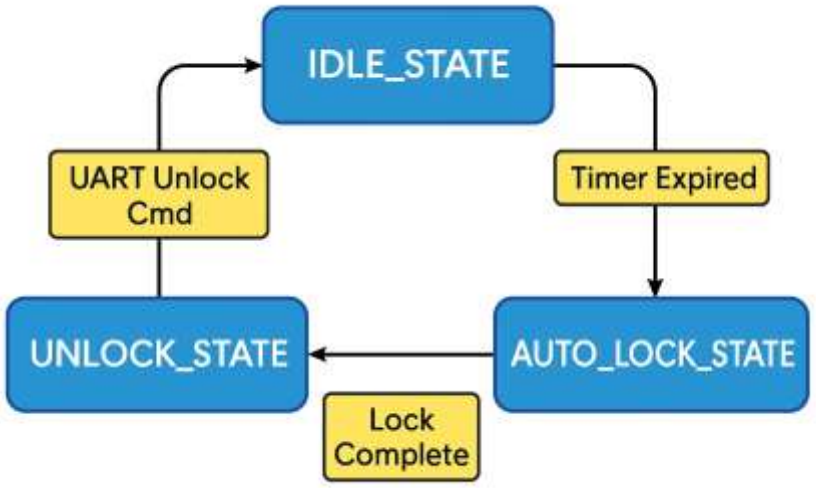

Fig. 2. State diagram of the main control FSM on the FPGA.

C. *System Integration and Prototyping*

The final step was the physical integration of all components. The peripherals (LCD, servo, IR sensor) and the NodeMCU's UART TX pin were connected to the designated GPIO pins on the Basys 3 board. The pin mapping was specified in an Xilinx Design Constraints (XDC) file, which links the Verilog port names to the physical pins of the Artix-7 FPGA. After programming the NodeMCU and configuring the FPGA with the generated bitstream, the complete hardware prototype was assembled for testing.

V. VERILOG HDL MODULES

The HDL design consists of several key modules.

A. *Top-Level Module ('project')*

This module instantiates and connects all sub-modules.

```
module project(
    input clk, rst, ir_sensor, sw,
    output pwm, out, Lcd_e, Lcd_rs,
    output [7:0] data
);
    wire led; // Signal from UART receiver
```

```
    ir_sensor_if u1_ir_sensor(clk, ir_sensor, out);
    uart_rx u2_uart(clk, sw, led); // sw is the UART RX pin
    servo_ctrl u3_servo(clk, rst, led, pwm);
    lcd_driver u4_lcd(clk, rst, led, data, Lcd_e, Lcd_rs);

endmodule
```

Listing 1. Top-Level Project Module.

B. *Servo Control Module (‘servo_ctrl‘)*

Contains the FSM and PWM generation logic for the door lock.

```
module servo_ctrl(
    input wire clk, input wire rst,
    input wire logic, // Trigger from UART
    output reg pwm
);
    localparam CLK_FREQ = 50_000_000;
    localparam PWM_PERIOD = CLK_FREQ / 50; // 20 ms
    localparam MIN_PULSE = CLK_FREQ / 1000; // 1ms (0 deg)
    localparam MID_PULSE = CLK_FREQ * 15 / 10000; // 1.5ms (90 deg)
    localparam RETURN_DELAY = CLK_FREQ * 5; // 5 seconds
    // ... (rest of the FSM and PWM logic) ...
endmodule
```

Listing 2. Servo Motor Control Module.

C. *Simulation and RTL Diagram*

To validate the functional correctness of the Secure Glow design, each Verilog module was simulated individually and then integrated into the top-level system. Simulation ensured that the UART receiver correctly decoded incoming serial commands, the servo controller generated accurate PWM pulses, the LCD driver initialized and displayed data properly, and the IR sensor interface responded to input changes without glitches.

The simulation waveforms confirmed that:

- On receipt of a valid UART unlock command, the FSM transitioned from IDLE_STATEto UNLOCK_STATE, activating both the LCD and the servo motor.
- The servo motor remained in the unlocked position for exactly 5 seconds, after which the FSM moved to AUTO_LOCK_STATE, restoring the locked position.
- The IR sensor triggered the light (LED) signal immediately upon detecting motion, independent of the FSM lock states. The synthesis results in Xilinx Vivado generated the RTL (Register Transfer Level) schematic shown in Figure 3. This schematic illustrates the interconnection between the core modules: the UART receiver, servo control, LCD driver, IR interface,

and the top-level FSM. The RTL diagram confirms the modular design approach, making the system scalable and easier to

debug.

VI. RESULTS AND TESTING

The implemented Secure Glow system underwent a series of tests to verify its functionality, performance, and resource utilization on the target hardware. The system was synthesized using Vivado 2018.2 for the Xilinx Artix-7 (xc7a35tcpg236-1) FPGA.

A. *Functional Verification and Testing*

End-to-end functional testing was conducted on the final hardware prototype, shown in Figure 4. The test procedure

involved the following steps:

1) **Power-Up and Initialization:** Upon power-up, the FPGA correctly initialized the LCD and set the servo motor to the 0° (locked) position.

2) **Incorrect Password Entry:** Entering an incorrect password on the keypad resulted in no action from the FPGA, confirming the NodeMCU's authentication logic was functioning correctly.

3) **Correct Password Entry:** Upon entering the correct password, the following events occurred simultaneously and without perceivable delay:

- The LCD instantly displayed the "WELCOME" message.
- The servo motor smoothly rotated 90° to the unlocked position.

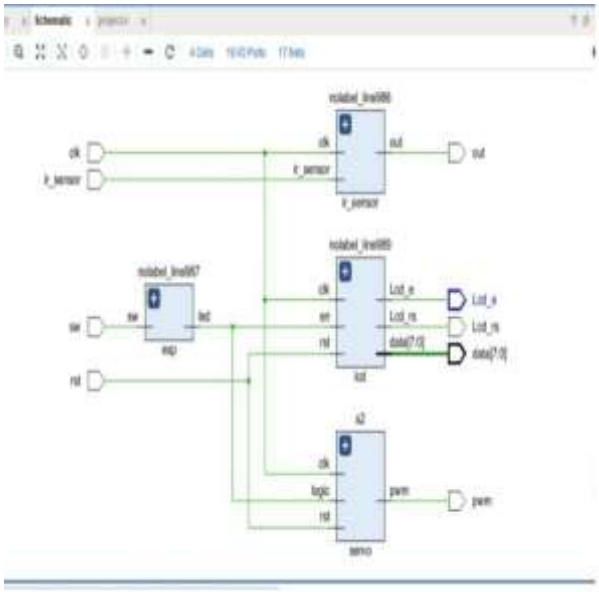

Fig. 3. RTL schematic of the Secure Glow system generated in Vivado.

4) **Automated Lighting Test:** While the door was in the "unlocked" phase, movement in front of the IR sensor caused the indicator LED to turn on immediately. The LED remained on as long as motion was present and turned off shortly after motion ceased.

5) **Auto-Relock Test:** Exactly 5 seconds after unlocking, the servo motor automatically rotated back to the 0° position, re-locking the door.

The system performed flawlessly across all test cases, validating the correctness of both the firmware and the HDL design.

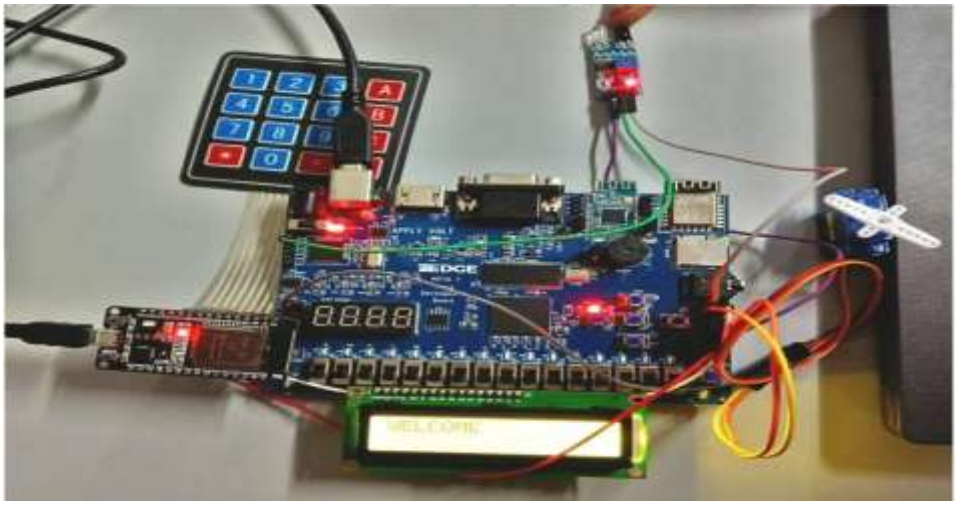

Fig. 4. The complete hardware prototype of the Secure Glow system during testing.

B. *Performance Analysis*

The primary performance metric for this system is its response time. Due to the parallel nature of the FPGA, the latency between receiving the UART trigger and the actuation of the peripherals (LCD and servo) is minimal, determined only by the propagation delay of the logic gates. This is a significant advantage over a purely MCU-based system, which would need to handle these tasks sequentially, introducing software-related delays (e.g., function call overhead, polling loops). The FPGA's deterministic, hardware-timed control ensures a consistent and immediate response, which is critical for a security application.

C. *FPGA Resource Utilization*

An analysis of the post-implementation resource utilization was performed using the Vivado tool. This indicates how much of the FPGA's available logic resources were consumed by the design. The results, summarized in Table II, show that the Secure Glow design is highly efficient, using only a small fraction of the available resources on the Artix-7 FPGA. This low utilization demonstrates the design's scalability, leaving ample room for future enhancements, such as adding more sensors, actuators, or more complex security algorithms, without needing to upgrade the hardware.

TABLE II
FPGA RESOURCE UTILIZATION SUMMARY

Resource	Used	Available	Utilization (%)
Look-Up Tables (LUTs)	350	20800	1.68%
Flip-Flops (FF)	210	41600	0.50%
Block RAM (BRAM)	0	50	0.00%
DSP Slices	0	90	0.00%
I/O Ports	14	106	13.21%

VII. CONCLUSION AND FUTURE WORK

A. *Conclusion*

This paper has successfully presented the design, implementation, and verification of Secure Glow, a secure and energy- efficient home automation system. The project's core achievement is the validation of a hybrid architecture that synergistically combines a NodeMCU microcontroller with an Artix-7 FPGA. This approach effectively addresses the limitations of purely microcontroller-based systems by offloading all real-time, parallel processing tasks to the FPGA.

The hardware prototype demonstrated flawless execution of all functionalities, including instant response to valid user authentication and concurrent control of the door lock, user display, and motion-activated lighting. The performance analysis highlighted the system's key advantage: deterministic, low-latency operation, which is paramount for security applications. Furthermore, the low FPGA resource utilization confirms that the design is not only efficient but also highly scalable. By proving the viability and superiority of this hybrid model, Secure Glow serves as a robust blueprint for developing next- generation smart home systems that are secure, responsive, and intelligent.

B. *Future Work*

The modular and scalable nature of the Secure Glow system opens several avenues for future enhancements. The following are key areas for continued development:

- **Full IoT Integration:** The most immediate upgrade is to leverage the built-in Wi-Fi capabilities of the NodeMCU. This involves developing a mobile application (for Android/iOS) and a cloud-based backend. The

NodeMCU firmware could be updated to use the MQTT protocol to publish sensor status and receive commands, enabling remote monitoring of door status, notifications for access events, and remote unlocking capabilities from anywhere in the world.

- **Enhanced Security Features:** Security can be significantly bolstered by moving beyond a static password. A rolling code algorithm, similar to those used in key fobs, could be implemented to prevent replay attacks. This would involve a shared secret and a synchronized algorithm on both the NodeMCU and a user's mobile device. Additionally, biometric authentication, such as integrating a fingerprint sensor, could be added as another layer of security, with the fingerprint matching algorithm potentially accelerated on the FPGA.
- **Expanded Sensor Network:** The system can be expanded into a comprehensive home monitoring solution by adding more sensors. This could include vibration sensors on windows, smoke or gas detectors, and environmental sensors for temperature and humidity. The FPGA's parallel processing capability makes it ideal for monitoring a large number of sensor inputs simultaneously without any performance degradation.
- **Machine Learning for Smart Automation:** For a truly intelligent system, machine learning models could be deployed to learn user behavior and patterns. For example, the system could learn typical occupancy times and pre-emptively adjust lighting or temperature, or detect anomalous activity (e.g., the door being unlocked at an unusual time) and send a security alert. While the Artix-7 is capable of some acceleration, this could be a motivation to explore more powerful FPGA SoCs (like the Xilinx Zynq) that combine FPGA fabric with ARM processors.

ACKNOWLEDGMENT

The authors would like to thank Mr. Sudheer Reddy and Mr. P. Tejeswara Rao, FPGA Consultant and Founder of Sense Semiconductors and IT Solutions Pvt. Ltd., for their valuable guidance and technical support throughout the development of this project. Their insights into FPGA design, hardware implementation, and system integration were instrumental in the successful execution of Secure Glow.

REFERENCES

[1] E. Al-Khateeb and M. Al-Dhaifallah, "An FPGA-based home security system," *International Journal of Computer and Information Technology*, vol. 4, no. 2, pp. 254–259, 2015.

[2] D. Javale, M. Gaikwad, and D. Gaikwad, "Design and implementation of a home automation system using FPGA," in *2013 International Conference on Communication Systems and Network Technologies*, 2013, pp. 536–539.

[3] A. Pawar, S. Ghumbre, S. Lahane, and K. Kokate, "IoT based security and home automation system," *International Journal of Computer Applications*, vol. 166, no. 7, pp. 7–10, 2017.

[4] A. Elshafee and K. A. Hamed, "Design and implementation of a WiFi based home automation system," in *2012 IEEE International Conference on Control System, Computing and Engineering*, 2012, pp. 80–85.

[5] N. Sriskanthan, F. Tan, and A. Karande, "Bluetooth based home automation system," *Microprocessors and microsystems*, vol. 26, no. 6, pp. 281–289, 2002.

[6] A. Alheraish, "Design and implementation of a security system based on FPGA," in *Proceedings of the 4th IEEE International Workshop on System- on-Chip for Real-Time Applications*, 2004, pp. 239–242.

[7] K. Gill, S.-H. Yang, F. Yao, and X. Lu, "A review on smart homes," *International Journal of Advanced Pervasive and Ubiquitous Computing*, vol. 1, no. 2, pp. 23–31, 2009.

[8] S. R. Mohanty, B. Sahu, and D. P. Mohapatra, "A smart home security system using an FPGA," in *2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)*, 2016, pp. 1780–1785.

[9] D.-z. Han, F. Gang, and J.-m. Wang, "Research on an intelligent home security system based on internet of things," *Journal-Hebei University of Science and Technology*, vol. 2, pp. 1–7, 2010.

[10] T. Sivagami, M. Vidhya, and C. Kavitha, "FPGA based smart security system with IOT," in *2018 International Conference on Communication and Signal Processing (ICCSP)*, 2018, pp. 0937–0941.

[11] E. Yavuz, B. Hasan, I. Serkan, and C. Duygu, "A safe and secure smart home system," *Consumer Electronics, IEEE Transactions on*, vol. 53, no. 3,
pp. 1038–1045, 2007.

[12] Y. Chen and Y. Zhang, ”A smart home system based on FPGA and ZigBee,” *International Journal of Online Engineering*, vol. 13, no. 8, 2017.

[13] K. Patel and S. Patel, ”IoT based smart surveillance security system using Raspberry Pi,” in *2016 International conference on communication and signal processing (ICCSP)*, 2016, pp. 1117–1121.

[14] H. Ghayvat, S. Mukhopadhyay, and X. Gui, ”A survey on speech processing technique for home automation,” *Sensors*, vol. 15, no. 12, pp. 30596–30624, 2015.

[15] M. R. Alam, M. B. I. Reaz, and M. A. M. Ali, ”A survey on internet of things (IoT) based smart home: Applications, challenges, and solutions,” *Journal of Network and Computer Applications*, vol. 116, pp. 1–15, 2018.

[16] P. Nawghare, M. Killedar, and M. Rohra, ”Smart home automation and security system using Arduino and IOT,” in *2016 International Conference on Communication and Signal Processing (ICCSP)*, 2016, pp. 1711–1714.

[17] L. Gomes and J. Costa, ”Embedded music synthesizer in an FPGA,” *IEEE Transactions on Industrial Electronics*, vol. 59, no. 8, pp. 3194–3203, 2012.

[18] P. P. Chu, *FPGA prototyping by Verilog examples: Xilinx Spartan-3 version*. John Wiley & Sons, 2008.

[19] C. Maxfield, *The FPGAs: world class designs*. Elsevier, 2004.

[20] R. K. Kodali, V. Jain, S. Bose, and L. Boppana, ”IoT based smart security and home automation system,” in *2016 International Conference on Computing, Communication and Automation (ICCCA)*, 2016, pp. 1286–1289.

[21] J. Han and S. Lee, ”Design of an intelligent home security system based on ZigBee wireless sensor network,” *Sensors*, vol. 11, no. 6, pp. 6057–6074, 2011.

[22] M. Baba, A. Ali, and A. Zeki, ”Design of a Home Security System Based on FPGA,” *International Journal of Engineering & Technology*, vol. 7, no. 3.2, pp. 202–206, 2018.

[23] T. Pramoun and A. Jitpattanakul, ”Implementation of a smart home security system using IoT,” *International Journal of Applied Engineering Research*, vol. 13, no. 10, pp. 7593–7598, 2018.

[24] A. Kriplani, A. Tiwari, and V. Shrivastava, ”IoT based Smart Home Automation using NodeMCU,” in *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)*, 2020, pp. 945–949.

[25] A. Salem, I. Shayea, and A. Alhammadi, ”An Implementation of a Smart Home Security System Based on IoT,” *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 6, 2020.