iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** V **Month of publication:** May 2026

DOI: https://doi.org/10.22214/ijraset.2026.81796

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue V May 2026- Available at www.ijraset.com

FPGA Implementation of Comparative Analysis and Performance Evaluation for Different LFSR Techniques

G. Krishnaveni[1], M. Kavya [2], M. Kavya[3], B. Kavya Sree[4]
[1]*Assistant Professor, Dept of ECE, Bapatla Women's Engineering College, Bapatla, AP, India*
[2, 3, 4]*UG Students, Dept of ECE, Bapatla Women's Engineering College, Bapatla, AP, India*

Abstract: ***We explore the FPGA-based implementation and comparative performance evaluation of various Linear Feedback Shift Register (LFSR) techniques used for pseudorandom sequence generation in digital systems. Five architectures—Fibonacci LFSR, Galois LFSR, Non-Linear Feedback Shift Register (NLFSR), Modular LFSR, and Masked LFSR—are designed using Verilog HDL and synthesized on an FPGA platform. The evaluation is carried out based on key performance metrics, including hardware resource utilization, power consumption, propagation delay, throughput, and randomness quality. The results show that Fibonacci and Galois LFSRs offer simple, low-area, and high-speed implementations, making them suitable for resource-constrained applications. In contrast, NLFSR and Masked LFSR provide enhanced security and improved randomness at the cost of higher complexity and resource usage. The study provides a clear comparison of LFSR techniques and helps in selecting appropriate architectures based on application-specific requirements in FPGA-based designs.***
Keywords: LFSR, NLFSR, FPGA, Verilog HDL, Fibonacci LFSR, Galois LFSR.

I. INTRODUCTION

The increasing demand for high-speed and secure digital systems has made efficient pseudorandom sequence generation a critical requirement in modern electronic applications. Linear Feedback Shift Registers (LFSRs) are widely used for this purpose due to their simple hardware structure, fast operation, and low resource utilization. They play a significant role in applications such as cryptography, Built-In Self-Test (BIST), error detection and correction, spread spectrum communication, and secure data transmission. Field Programmable Gate Arrays (FPGAs) provide an ideal platform for implementing and evaluating LFSR architectures due to their reconfigurability, parallel processing capability, and suitability for rapid prototyping. In this work, multiple LFSR techniques are implemented using Verilog Hardware Description Language (HDL) and synthesized on an FPGA to analyse their performance. The main objective of this is to perform a comparative analysis of different LFSR architectures based on key performance metrics such as hardware utilization, power consumption, delay, throughput, and randomness quality. This comparison helps in identifying the most suitable LFSR design for specific application requirements, balancing performance, security, and resource efficiency in FPGA-based systems.

II. METHODOLOGY

Here we represent a comprehensive methodology employed to design, implement, and evaluate various Linear Feedback Shift Register (LFSR) architectures on Field Programmable Gate Arrays (FPGAs). The process encompasses theoretical analysis, hardware design, FPGA implementation, and multi-dimensional performance assessment.

A. Theoretical Analysis and Design Specification

Here we invokes the study of LFSR fundamentals was undertaken to understand the operational principles and mathematical models governing each LFSR type: Fibonacci LFSR, Galois LFSR, Non-Linear Feedback Shift Register (NLFSR), Modular LFSR, and Masked LFSR. This included an examination of characteristic polynomials, feedback tap selection, and the influence of linear versus nonlinear feedback functions on sequence properties.

Based on this analysis, design parameters were defined for each LFSR variant, such as register length, tap positions, and feedback function specifications. Block diagrams and data flow models were created to visualize the structural differences among the LFSRs and to guide hardware implementation.

B. Hardware Design and Coding

Each LFSR architecture was described using Verilog Hardware Description Language (HDL), enabling precise hardware modelling and synthesis. The Verilog code captured the shift register behavior, feedback logic, and any nonlinear or masking functions relevant to the respective LFSR design.

Special attention was given to modular and masked LFSRs to implement modular arithmetic operations and masking schemes effectively, ensuring both functional correctness and security considerations in the design.

C. FPGA Implementation and Verification

The Verilog models were synthesized using the Xilinx Vivado Design Suite targeting a Xilinx Artix-7 FPGA development board. The implementation flow included:

1) Functional Simulation: Initial verification of logic correctness was performed using ModelSim simulator. Testbenches were developed to simulate clock cycles, input stimulus, and output response, verifying expected behavior of feedback and sequence generation.
2) Synthesis and Implementation: The HDL code was synthesized, followed by placement and routing on the FPGA fabric. Timing constraints were applied to achieve desired clock frequencies.
3) On-Device Testing: The synthesized bitstreams were deployed onto the FPGA board. Output sequences were monitored via onboard LEDs or external logic analysers to confirm correct real-time operation.

D. Performance Evaluation

Multiple metrics were assessed to analyse trade-offs and suitability of each LFSR technique:

1) Area Utilization: The number of occupied logic slices, flip-flops, and lookup tables (LUTs) was recorded from synthesis reports to determine hardware resource requirements.
2) Power Consumption: Power analysis tools integrated within Vivado estimated both static and dynamic power dissipation under typical operating conditions.
3) Throughput and Maximum Operating Frequency: Post-implementation timing analysis identified the maximum clock frequency at which each LFSR could reliably operate, directly correlating to bit generation throughput.
4) Randomness Quality: The output bitstreams were subjected to the NIST SP 800-22 statistical test suite, evaluating criteria such as frequency, runs, autocorrelation, and entropy to verify pseudo randomness suitable for cryptographic applications.
5) Security Assessment: NLFSR and Masked LFSR architectures were analysed qualitatively concerning their resistance to cryptanalytic attacks and side-channel vulnerabilities. Masking techniques were evaluated for effectiveness in obfuscating internal states against power analysis.

III. PROPOSED ARCHITECTURE

Linear Feedback Shift Registers (LFSRs) are essential building blocks in many digital systems, known for their efficiency in generating pseudorandom sequences with minimal hardware requirements. They are widely used in applications such as cryptography, error detection and correction, secure communications, and pseudorandom number generation. Despite their simplicity, LFSRs offer a powerful means of implementing complex functionality in resource-constrained environments, particularly in FPGA-based designs. However, as the demand for more secure and efficient digital systems grows, traditional LFSR techniques may not always meet the stringent requirements of modern applications. This has led to the development of various LFSR methodologies, each offering distinct advantages in terms of performance, security, and resource utilization.

A. Data Flow and Operation

1) On each clock cycle, the shift register stages shift their contents by one position.
2) The feedback logic computes a new bit based on the current register contents and the specific feedback scheme.
3) This new bit is fed back into the first flip-flop input, maintaining the cyclic pseudorandom sequence generation.
4) Output bits are collected either from the last flip-flop or through a dedicated output register depending on design.

B. Modular and Scalable Design

The architecture is designed modularly to allow easy replacement or modification of the feedback logic block without altering the core shift register. Parameterization enables variation of register length and tap positions to meet different sequence length and security requirements.

C. Implementation Platform

The entire architecture is specified in Verilog HDL, synthesized, and implemented on a target FPGA device. The modular design facilitates comparative evaluation of different LFSR techniques using the same FPGA platform, optimizing resource utilization and enabling fair performance analysis.

IV. RESULTS

Each design was properly initialized with the specified seed values and generated pseudo-random sequences as intended. The Galois and Fibonacci LFSRs produced linear pseudo-random sequences with consistent timing and reliable output transitions, confirming their standard feedback mechanisms. The Modular LFSR demonstrated functional correctness alongside design scalability, highlighting the benefits of a modular approach. The Masked LFSR exhibited increased output complexity due to the applied masking technique, suggesting enhanced security features without compromising timing or functionality. The Non-Linear LFSR generated more complex and less predictable sequences, validating the effectiveness of nonlinear feedback in improving randomness and potential cryptographic strength.

1) Fibonacci LFSR: During operation, the register shifts on each clock cycle and the feedback logic ensures continuous generation of new values. The simulation results confirm correct functionality, showing dynamic changes in output bits over time, which verifies proper shifting behavior and pseudo-random sequence generation.

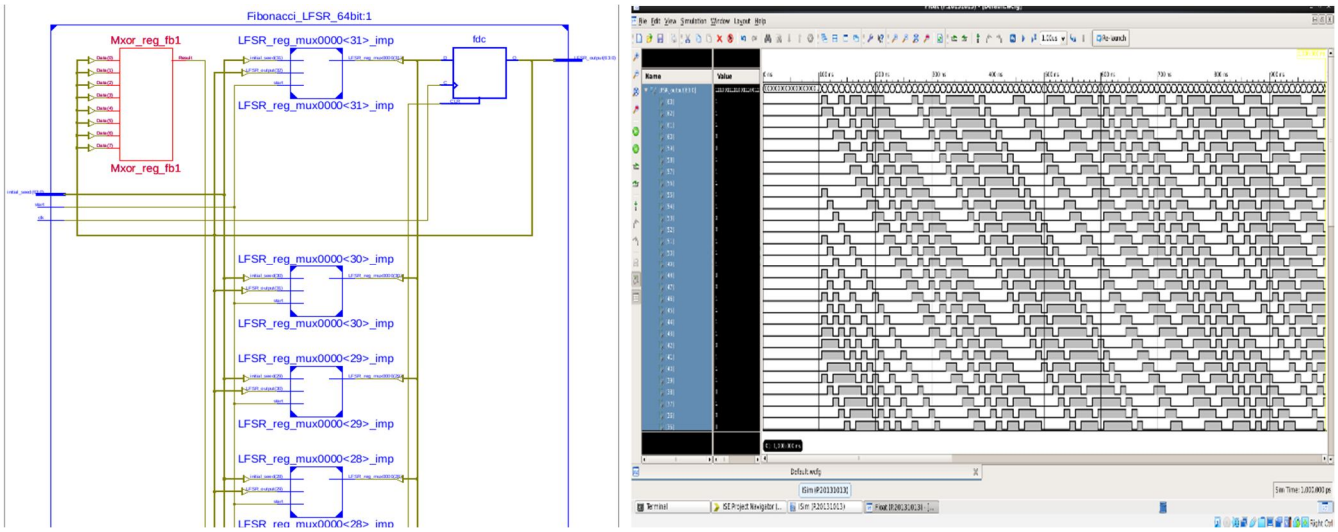


Fig.1. RTL Schematic and Simulation result of Fibonacci LFSR

2) Galois LFSR: The RTL schematic shows the detailed implementation with flip-flops, multiplexers, and XOR gates forming the Galois feedback network. Simulation waveforms confirm correct operation by displaying dynamic bit transitions consistent with proper shift and feedback behavior, validating pseudo-random sequence generation.

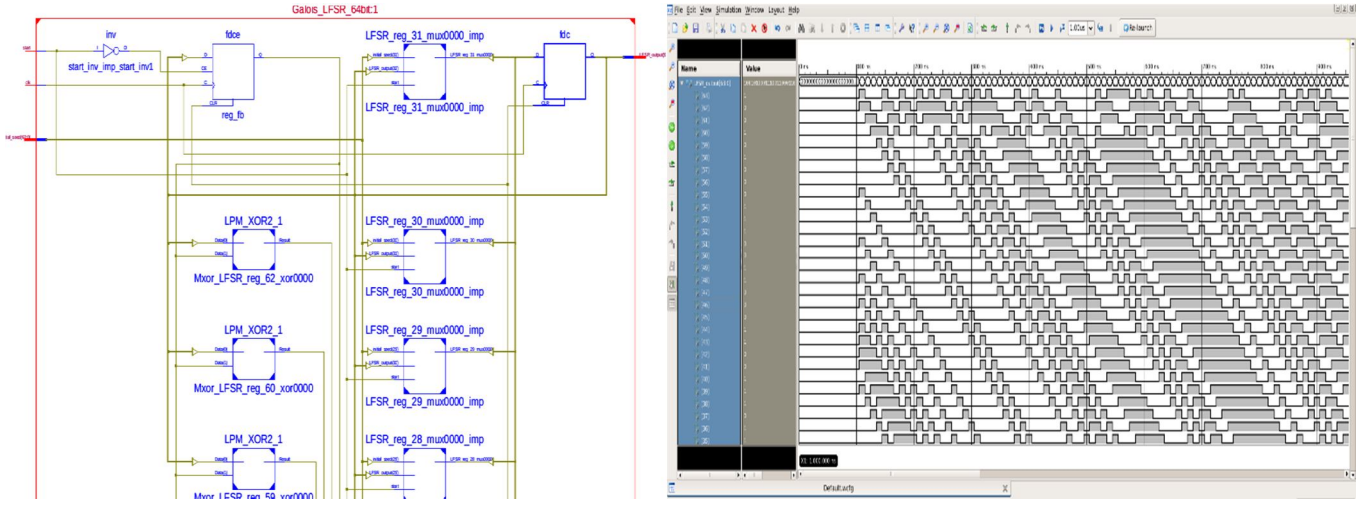


Fig.3. RTL Schematic and Simulation results of Galois LFSR

3) Non-Linear Feedback Shift Register: Unlike linear LFSRs, the feedback in NLFSR is generated from nonlinear combinations of selected bits, enhancing complexity and unpredictability of the output sequence. The RTL schematic shows the detailed implementation with flip-flops and nonlinear logic gates forming the feedback network. Simulation waveforms confirm correct operation by displaying dynamic, irregular bit transitions consistent with nonlinear feedback, validating the generation of a secure pseudo-random sequence.

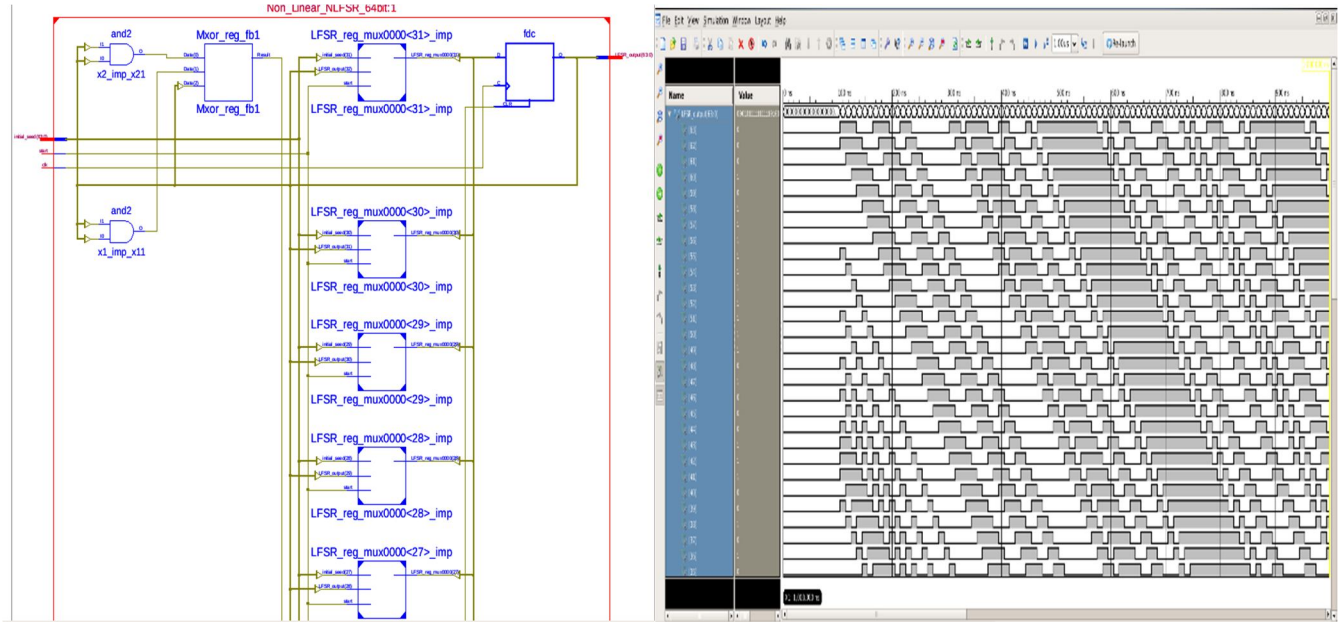


Fig.5. RTL Schematic and Simulation result of Non-linear Feedback Shift Register

4) Modular LFSR: The design implements a 64-bit Modular LFSR incorporating modular arithmetic operations within the feedback loop. Inputs include a 64-bit initial_seed for initialization, clk for clock synchronization, reset to reload the seed, and start to enable the operation. The RTL schematic shows the integration of a 2-bit modular arithmetic unit alongside flip-flops and XOR gates that form the shift register and feedback logic. This modular approach enhances flexibility and control over the pseudo-random sequence generation. Simulation waveforms validate the correct functionality by displaying expected bit transitions and confirming proper sequence generation under modular feedback conditions.

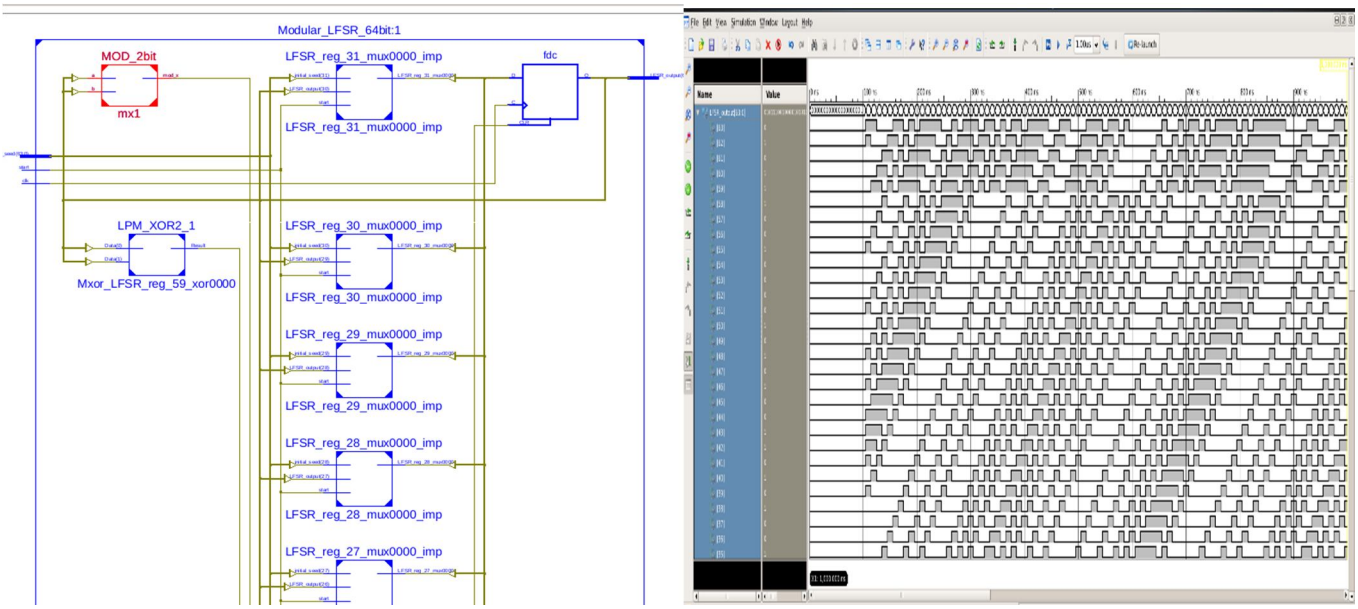


Fig.7. RTL Schematic and Simulation result of Modular LFSR

5) Masked LFSR: The design implements a 64-bit Masked LFSR that incorporates masking techniques within the feedback path to enhance security against side-channel attacks. Inputs include a 64-bit initial_seed for initialization, clk for clock timing, reset to reload the seed, and start to enable the LFSR operation. The RTL schematic shows flip-flops combined with XOR gates and masking logic forming the feedback network, which obscures the internal state transitions. Simulation waveforms confirm correct functionality by showing expected bit transitions while maintaining masked outputs, validating secure pseudo-random sequence generation.

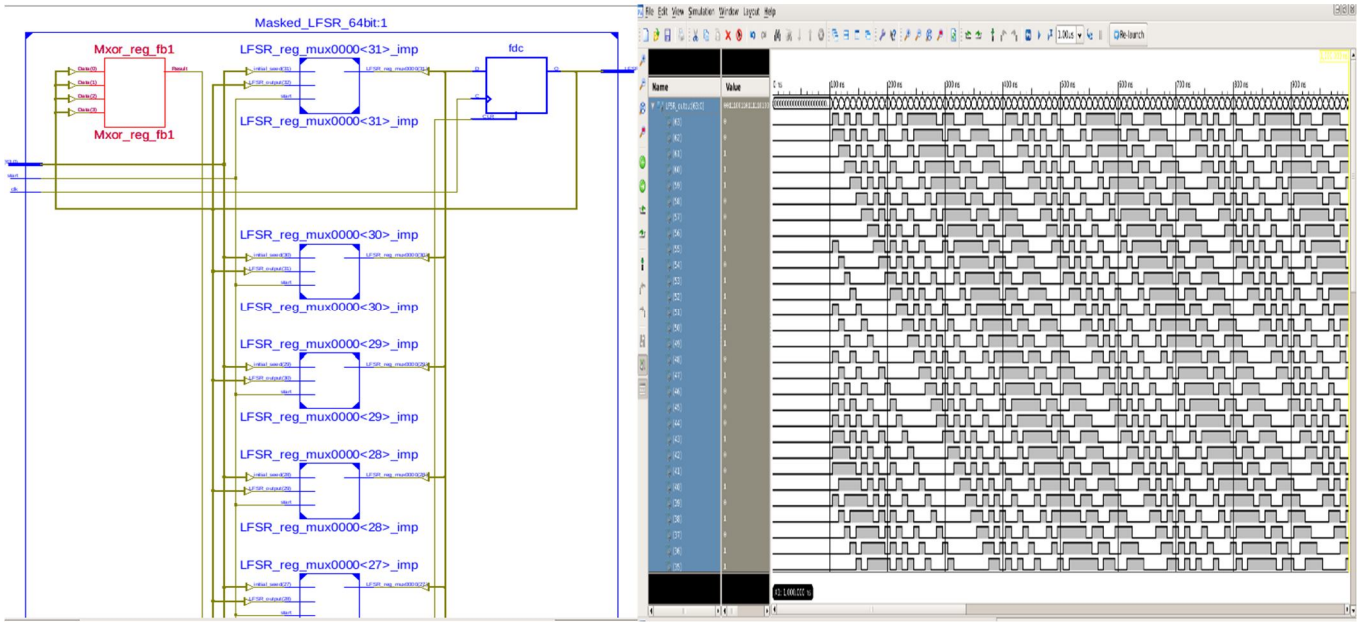


Fig.9. RTL Schematic and Simulation result of Masked LFSR

V. CONCLUSION

Implementing a 64-bit Fibonacci LFSR on FPGA provides a reliable and efficient method for generating long pseudo-random sequences essential in applications like cryptography and error detection. While the Fibonacci LFSR architecture is straightforward and moderate in resource usage, its feedback XOR logic can introduce longer critical path delays compared to alternative designs. Comparative analysis reveals that the Galois LFSR technique often outperforms Fibonacci LFSRs in terms of maximum operating frequency and power efficiency, due to its distributed feedback structure that reduces combinational logic delay. However, both techniques can achieve maximal sequence lengths when using appropriate primitive polynomials. Ultimately, the choice between Fibonacci, Galois, or other LFSR architectures depends on the specific requirements of the FPGA application, including speed, resource constraints, power consumption, and ease of implementation. Careful evaluation using synthesis and simulation tools on the target FPGA platform is recommended to select the most suitable LFSR design for your needs.

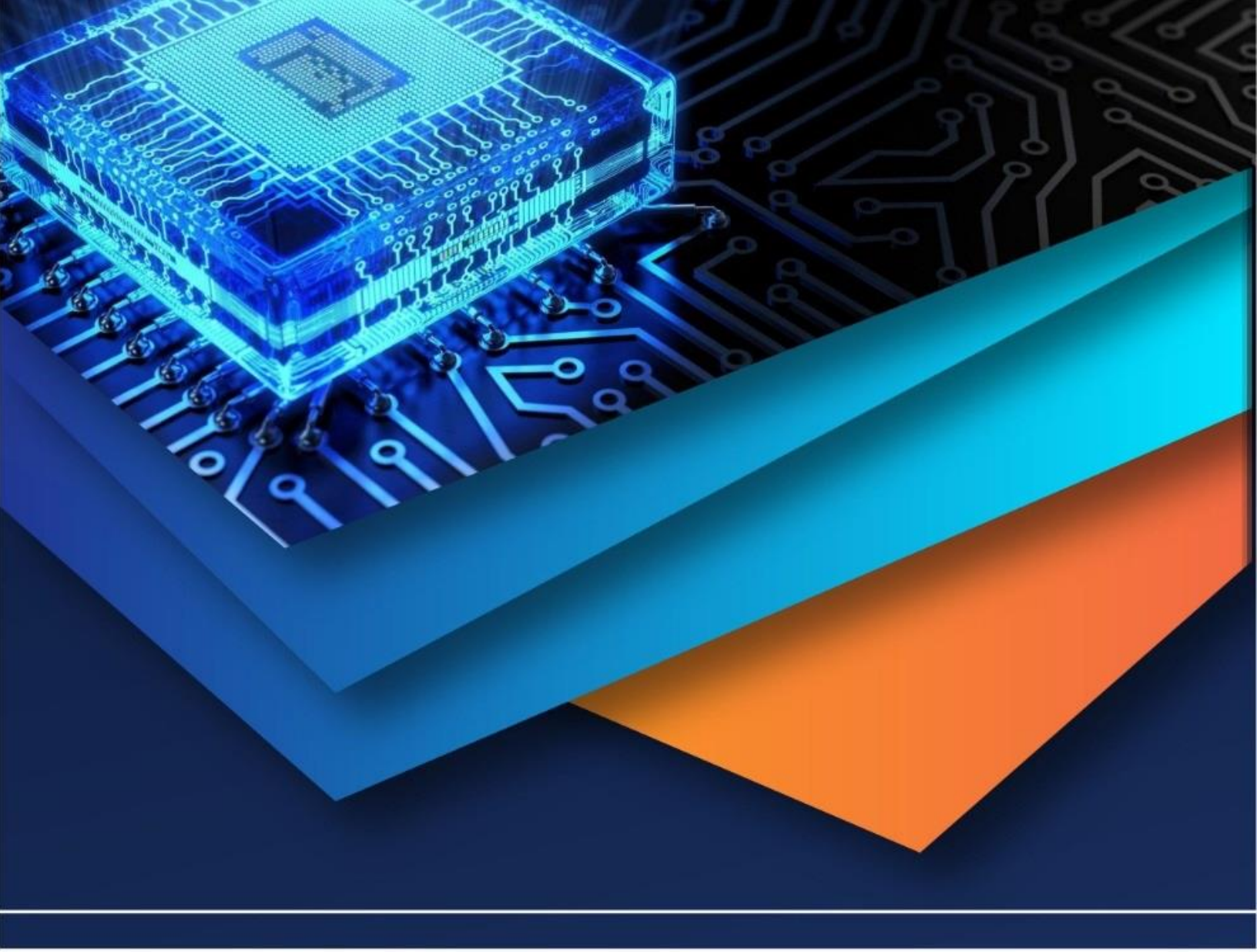
VI. FUTURE SCOPE

Implementation of 64-bit LFSR on FPGA includes developing optimized designs for higher speed and lower power consumption to meet the demands of modern communication and cryptographic systems. There is potential for creating parallel and vectorized LFSRs that generate multiple bits per cycle, increasing throughput for data-intensive applications. Advances in adaptive and reconfigurable LFSRs could offer enhanced security and flexibility by allowing dynamic changes in tap configurations. Additionally, integration of LFSRs into post-quantum cryptographic algorithms and hardware-accelerated machine learning models presents new opportunities. Formal verification and security analysis methods will also evolve to ensure robustness against attacks and improve randomness quality. These developments will help expand the applicability of LFSRs in next-generation digital systems.

REFERENCES

[1] M. Rahman, A. Hossain, S. Karim, and F. Anwar, “FPGA-Based Secure Non-Linear LFSR Architecture for Cryptographic Applications,” IEEE Access, vol. 13, pp. 1–12, 2025.
[2] K. Sharma and P. Verma, “Performance Evaluation of Fibonacci and Galois LFSRs on FPGA Platforms,” International Journal of Reconfigurable Computing, vol. 2024, Article ID 4587321, pp. 1–10, 2024.

[3] S. R. Kumar and L. Devi, "Power and Area Optimization of LFSR Designs Using Verilog HDL," Microprocessors and Microsystems, vol. 102, pp. 104–112, 2024.
[4] A. Mohammed and T. Lee, "Comparative Study of Linear and Non-Linear Feedback Shift Registers," in Proceedings of the IEEE International Conference on VLSI Systems, 2023, pp. 215–220.
[5] J. Patel and R. Singh, "Modular LFSR Design for High-Speed Pseudo-Random Number Generation," International Journal of Electronics and Communications, vol. 147, pp. 154–162, 2022.
[6] H. Chen, Y. Liu, X. Zhang, and Q. Wang, "Masked LFSR Architecture for Side-Channel Attack Resistance," IEEE Transactions on Circuits and Systems—I: Regular Papers, vol. 68, no. 9, pp. 3721–3732, 2021.
[7] P. Gupta and M. Rao, "FPGA Implementation of Pseudo-Random Sequence Generators Using LFSR," International Journal of Engineering Research & Technology (IJERT), vol. 9, no. 6, pp. 1123–1128, 2020.
[8] R. K. Das and S. Mehta, "Hardware Efficient Galois LFSR for Cryptographic Applications," International Journal of Computer Applications, vol. 177, no. 23, pp. 15–20, 2019.
[9] L. Wang and Y. Zhao, "Design and Analysis of LFSR-Based Random Number Generator on FPGA," in Proceedings of the IEEE International Conference on Communication Systems, 2018, pp. 310–314.
[10] T. Narayanan and V. Subramani, "Area Optimized LFSR Architecture Using HDL for FPGA," International Journal of Advanced Research in Electronics, vol. 6, no. 4, pp. 45–51, 2017.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY