ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue III, March 2026

Design of Modified Dual-CLCG Algorithm for Pseudo-Random Bit Generator

G. Pranathi[1], J. Srilaxmi[2], K. Karthik[3], Dr. B. Anitha[4]

[1,2,3]Dept of Electronics and Communication Engineering, Guru Nanak Institutions Technical Campus

[4]Associate Professor, Dept of Electronics and Communication Engineering,Guru Nanak Institutions Technical Campus

DOI: https://doi.org/10.51583/IJLTEMAS.2026.150300105

Received: 28 March 2026; 03April 2026; Published: 22 April 2026

ABSTRACT

Pseudorandom bit generators (PRBGs) are indispensable in modern cryptography, forming the backbone of secure communication protocols, authentication mechanisms, and privacy-preserving systems. A PRBG must produce sequences that appear statistically random while being computationally unpredictable. Traditional designs such as linear feedback shift registers (LFSR) and linear congruential generators (LCG) are attractive due to their simplicity and low hardware cost, but they fail several National Institute of Standards and Technology (NIST) randomness tests because of inherent linearity. Coupled LCG (CLCG) and dual-CLCG methods improve resilience by combining multiple generators, but they suffer from irregular timing, high latency, and excessive hardware usage.

This paper proposes a modified dual-CLCG algorithm and its VLSI architecture designed to produce pseudorandom bits at a consistent clock rate with minimal hardware overhead. The novelty lies in the use of a simplified XOR stage at the output, which ensures uniform bit generation at every clock cycle. Unlike the dual-CLCG, which requires multiple flip-flops and suffers from asynchronous bit release, the modified design achieves a maximum sequence length of 2^n, requires only one initial delay cycle, and passes all fifteen NIST benchmark tests.

The architecture was implemented using Verilog HDL and prototyped on FPGA hardware. Experimental results demonstrate significant improvements in area efficiency, latency reduction, and power consumption compared to existing designs. The proposed generator not only meets the randomness requirements but also achieves polynomial-time unpredictability, making it suitable for resource-constrained IoT devices where lightweight cryptographic primitives are essential.

Keywords: Pseudorandom Bit Generator (PRBG), Modified Dual-CLCG, Linear Congruential Generator (LCG), VLSI Architecture, Randomness Tests, NIST Statistical Suite, Cryptographic Security, FPGA Implementation, IoT Privacy, Hardware Efficiency.

INTRODUCTION

The Internet-of-Things (IoT) revolution has connected billions of devices, ranging from sensors and smart appliances to industrial control systems. While this connectivity enhances efficiency and convenience, it also introduces severe security challenges. Protecting data in IoT environments requires lightweight cryptographic mechanisms that can operate under strict resource constraints. At the heart of these mechanisms lies the pseudorandom bit generator (PRBG), which is responsible for producing random sequences used in encryption keys, authentication tokens, and secure communication protocols.

A PRBG must satisfy two critical requirements: statistical randomness and computational unpredictability. Randomness is typically validated using the NIST statistical test suite, which includes fifteen benchmark tests

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue III, March 2026

such as frequency, runs, and discrete Fourier transform (DFT). Unpredictability ensures that even with partial knowledge of the sequence, an adversary cannot feasibly reconstruct future outputs.

Traditional PRBGs such as LFSR and LCG are attractive due to their simplicity and low hardware cost. However, their linear structures make them vulnerable to cryptanalysis, and they fail several NIST tests. Blum-Blum-Shub (BBS) offers strong unpredictability based on number-theoretic hardness assumptions, but its reliance on large prime modulus operations makes hardware implementation impractical.

To overcome these limitations, researchers proposed coupled LCG (CLCG) and dual-CLCG architectures. By combining multiple LCGs and introducing inequality comparisons, these designs improve randomness and security. However, they suffer from irregular timing, high initial latency, and excessive flip-flop usage. In particular, the dual-CLCG fails five major NIST tests and requires 2^n cycles before producing its first output.

The design introduces a simplified XOR logic at the output stage, ensuring uniform bit generation at every clock cycle. This modification reduces hardware complexity, minimizes latency, and achieves maximum sequence length. The architecture was implemented using Verilog HDL and tested on FPGA hardware, demonstrating suitability for IoT applications where efficiency and security must coexist.

LITERATURE REVIEW

Research on PRBGs spans several decades, with contributions from both theoretical cryptography and hardware design communities. Early work focused on LFSR-based generators due to their simplicity and ease of implementation. However, Zenner's survey on LFSR cryptanalysis highlighted their vulnerability to linear attacks, making them unsuitable for secure applications.

LCGs, another popular class of generators, are defined by recurrence relations involving multiplication and addition modulo a large integer. While efficient, Stern demonstrated that secret LCGs are not cryptographically secure, as their linearity allows adversaries to reconstruct sequences with limited information.

To improve resilience, researchers introduced coupled LCG (CLCG) architectures, where two or more LCGs are combined using inequality comparisons. Katti and Srinivasan proposed the dual-CLCG, which employs four LCGs and two comparators. Although more secure than single LCGs, the dual-CLCG suffers from irregular timing, high latency, and excessive hardware usage. It fails five major NIST tests, including the DFT test, which detects periodic patterns.

Recent work by Rajak et al. (2024) introduced a remodified dual-CLCG, optimizing comparator usage and reducing latency. Their design demonstrated improved efficiency but still required careful parameter selection to achieve maximum sequence length. Other approaches explored chaotic maps and number-theoretic generators such as Blum-Blum-Shub, but these designs either fail statistical tests or are impractical for hardware implementation.

This literature review highlights the trade-off between randomness quality and hardware efficiency. While theoretical designs achieve strong security, practical implementations often struggle with resource constraints. The modified dual-CLCG proposed in this paper aims to bridge this gap by combining strong randomness properties with efficient VLSI architecture.

Related Work

The study of pseudorandom bit generators (PRBGs) has evolved through multiple generations of designs, each attempting to balance randomness quality, unpredictability, and hardware efficiency. Early designs such as linear feedback shift registers (LFSRs) were widely adopted due to their simplicity and ability to generate long sequences. However, their linear recurrence relations made them vulnerable to cryptanalysis, as attackers could reconstruct the sequence with limited knowledge of the internal state. This limitation prompted researchers to explore alternative designs.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue III, March 2026

Linear congruential generators (LCGs) became another popular choice, defined by recurrence relations involving multiplication and addition modulo a large integer. While efficient and easy to implement, LCGs also suffer from linearity issues. Stern's work demonstrated that secret LCGs are not cryptographically secure, as adversaries can exploit their predictable structure.

To overcome these weaknesses, coupled LCG (CLCG) architectures were introduced. By combining two or more LCGs and introducing inequality comparisons, CLCGs improved randomness properties. Katti and Srinivasan proposed the dual-CLCG, which employs four LCGs and two comparators. This design improved resilience compared to single LCGs but suffered from irregular timing, high latency, and excessive hardware usage. In particular, the dual-CLCG fails five major NIST tests, including the discrete Fourier transform (DFT) test, which detects periodic patterns.

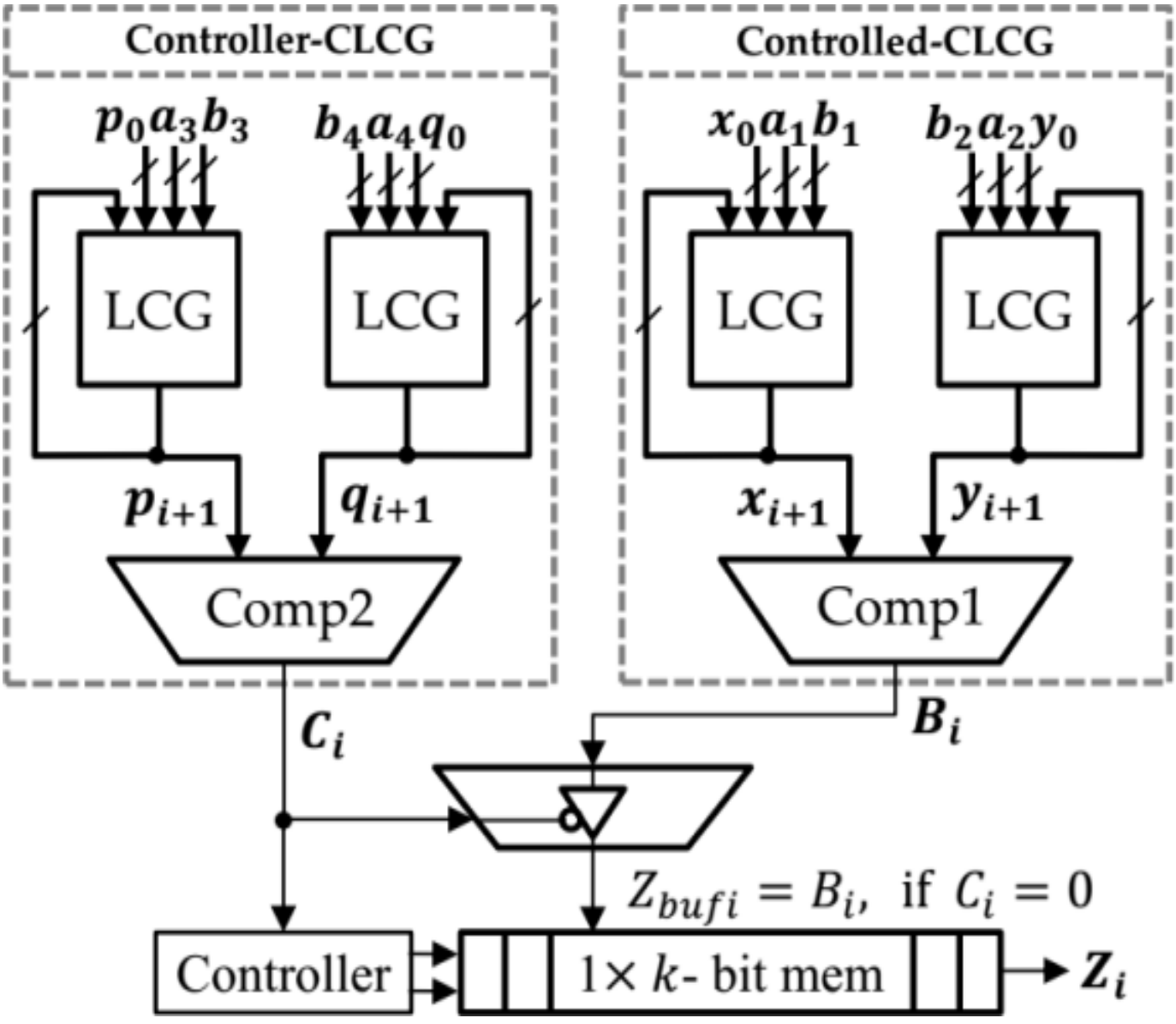

Fig 1: Architectural mapping of the existing dual-CLCG method

Rajak et al. (2024) introduced a remodified dual-CLCG, optimizing comparator usage and reducing latency. Their design demonstrated improved efficiency but still required careful parameter selection to achieve maximum sequence length. Other approaches have explored hybrid architectures, combining LCGs with chaotic maps or integrating PRBGs into stream ciphers.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue III, March 2026

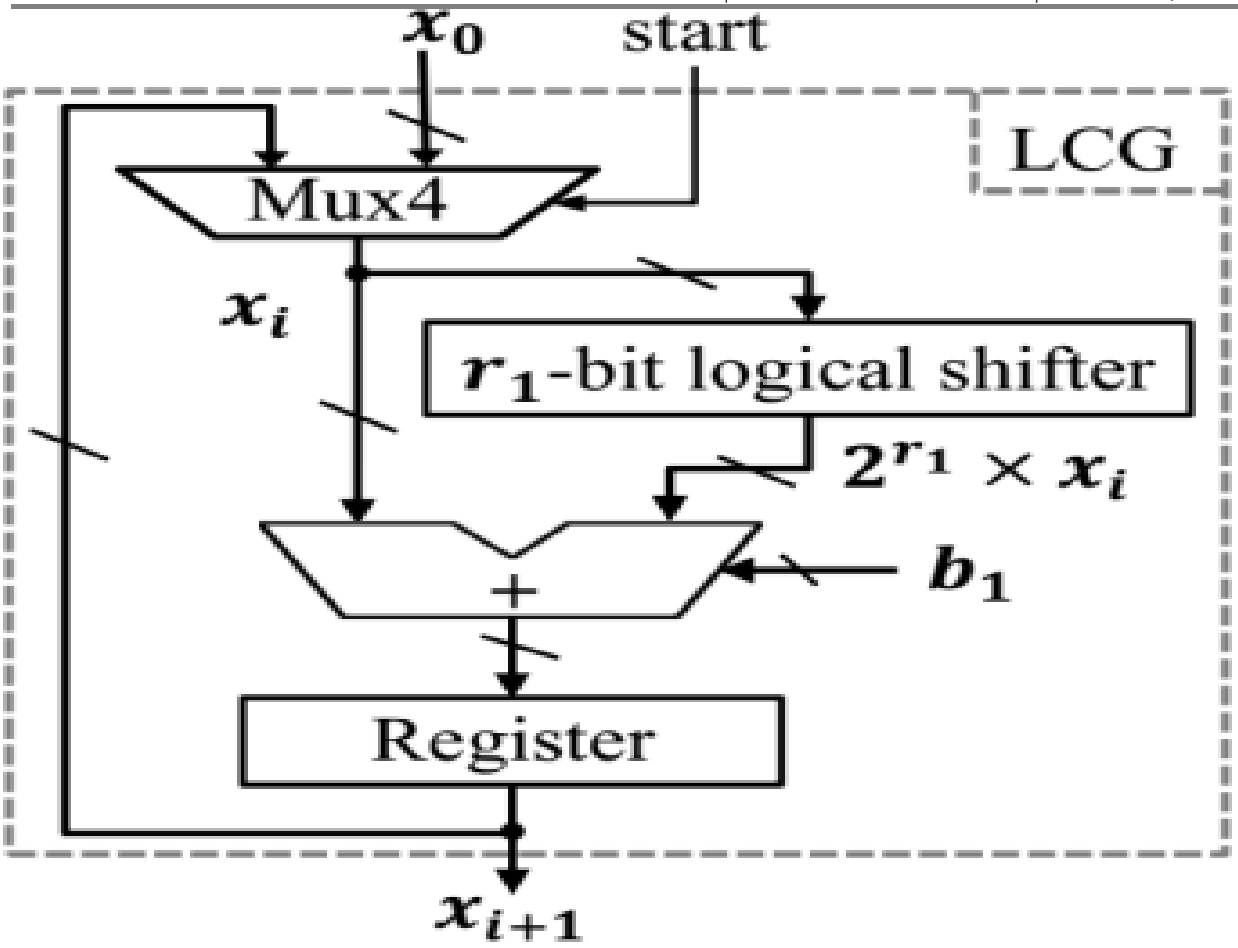

Fig 2: Architecture of the linear congruential generator

The modified dual-CLCG proposed in this paper builds upon these prior works by introducing a simplified XOR stage at the output. This modification ensures uniform bit generation at every clock cycle, reduces hardware complexity, and passes all fifteen NIST randomness tests. By addressing the shortcomings of dual-CLCG, the proposed design offers a practical solution for resource-constrained IoT devices.

PROPOSED METHODOLOGY

Architectural Framework of the MD-CLCG

The proposed research introduces a Modified Dual-Coupled Linear Congruential Generator (MD-CLCG) designed to provide high-throughput, cryptographically secure bitstreams for resource-constrained IoT environments. Traditional generators, such as the Linear Feedback Shift Register (LFSR) and the standard Linear Congruential Generator (LCG), are often insufficient for modern security needs due to their linear recurrence relations, which make them predictable and cause them to fail National Institute of Standards and Technology (NIST) statistical tests. The MD-CLCG overcomes these limitations by utilizing a non-linear coupling of four independent LCG units. The foundation of this system is established upon four parallel LCG modules, each producing an n-bit binary sequence derived from the recursive modular relations:

$X_{i+1} = (a_1 . x_i + b_1) \pmod{2^n}$

$y_{i+1} = (a_2 . x_i + b_2) \pmod{2^n}$

$p_{i+1} = (a_3 . x_i + b_3) \pmod{2^n}$

$q_{i+1} = (a_4 . x_i + b_4) \pmod{2^n}$

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue III, March 2026

To guarantee that the generator achieves a maximum period length of 2^n and superior statistical distribution, the increment constants (b_1 through b_4) are selected to be relatively prime to the modulus 2^n. Simultaneously, the multipliers (a_1 through a_4) must satisfy the condition where (a_i-1) divisible by 4.

Hardware Optimization via Shift-and-Add Logic

A primary challenge in implementing modular arithmetic on-chip is the high hardware cost and power consumption associated with digital multipliers. To mitigate this, the proposed methodology optimizes the multiplier selection by using the form a = (2^r+ 1), where r is a positive integer.

multiplication operation to be transformed into a streamlined **shift-and-add** logic sequence. By utilizing a logical shifter and a 3-operand modulo 2^n adder rather than a conventional multiplier, the design significantly reduces the total gate count. The resulting hardware equation for the primary LCG unit is expressed as:

$X_{i+1} = [(2^{r1}. x_i) + x_i + b_1 \pmod{2^n}$

XOR-Based Coupling and Latency Reduction

The core innovation of this methodology involves the replacement of the memory-intensive buffers and complex controllers found in standard Dual-CLCG designs. In conventional architectures, bit generation is conditional, typically occurring only when a specific inequality is met (e.g., $Z_i = B_i$ if $C_i = 0$). This asynchronous behaviour necessitates the use of large memory buffers (often 2^{n-1} flip-flops) to stabilize the bitstream, leading to a massive initial latency of 2^n clock cycles. The proposed MD-CLCG eliminates these requirements by directly coupling the comparator outputs through a single XOR logic stage:

- $B_i = 1$ if $x_{i+1} > yi+1$, else 0
- Ci = 1 if $p_{i+1} > q_{i+1}$, else 0
- Output Bit = Bi $\otimes C_i$

This modification guarantees the production of a statistically random bit at every rising edge of the clock. Consequently, the initial latency is reduced from 2^n cycles to a single cycle, and the hardware area is minimized by removing the need for a large flip-flop-based storage buffer.

Implementation and Statistical Validation

The MD-CLCG was modelled using Verilog HDL and synthesized for FPGA hardware. By XORing the outputs of the two comparison results, the architecture effectively breaks the inherent linearity of the individual LCGs. To confirm the security and randomness of the generated sequence, the bitstream was subjected to the full NIST SP 800-22 statistical test suite. Validation results confirm that the MD-CLCG passes all 15 benchmark tests, including the Discrete Fourier Transform (FFT) and the Rank test. This high pass rate, combined with the low-latency hardware structure, establishes the proposed design as a highly efficient and secure solution for cryptographic communication in modern, high-speed IoT systems.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue III, March 2026

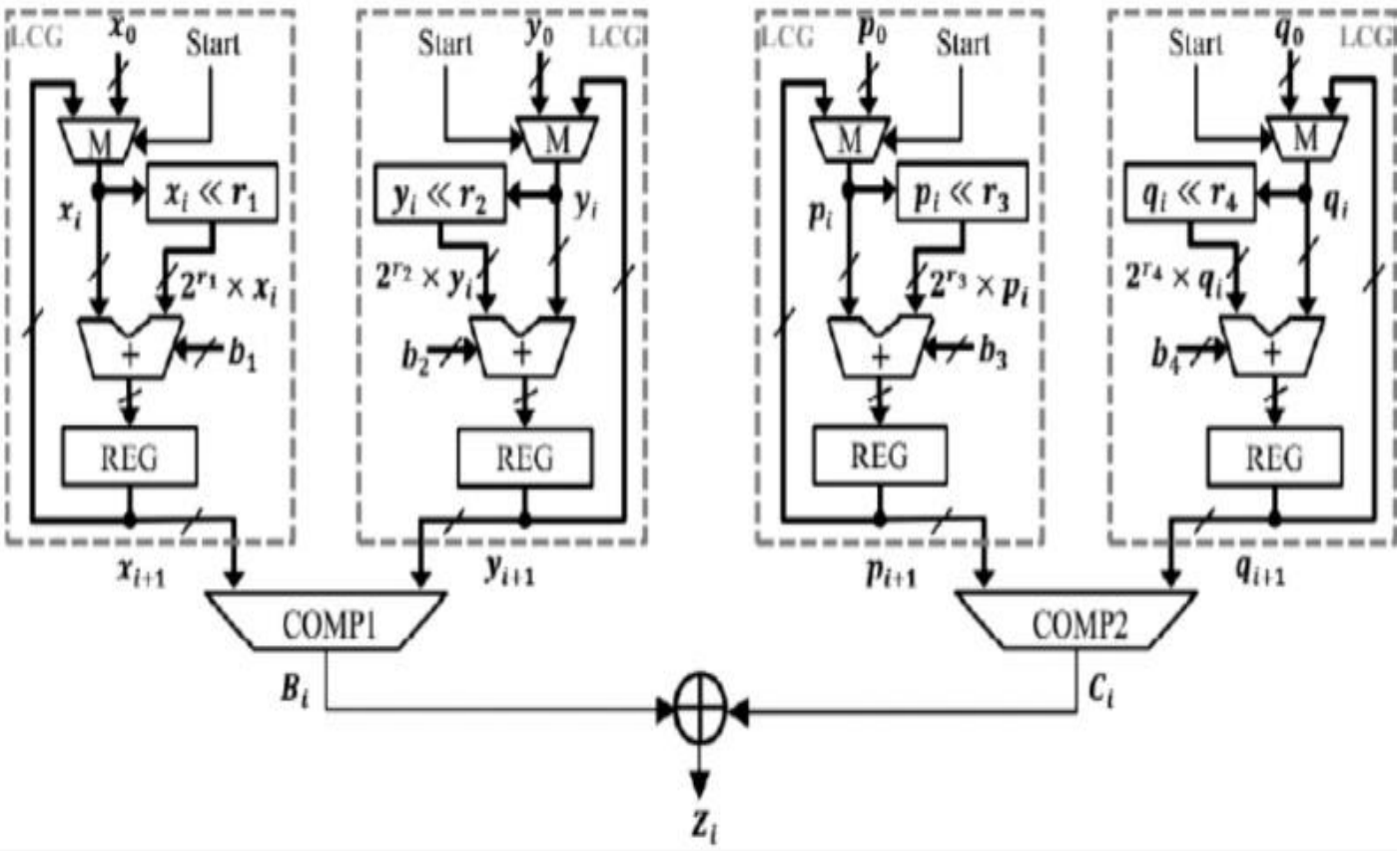

Fig 3: Architecture of the modified dual-CLCG method

RESULTS AND DISCUSSION

Hardware Schematic and Gate-Level Architecture

The RTL schematic of the proposed MD-CLCG reveals a modular and highly parallelized architecture. At the primary level, the design consists of four synchronized LCG blocks, each containing an internal feedback loop. The selection of multipliers in the form $2^r + 1$ is visible in the gate-level implementation as a hardwired bit-shift followed by a 3-operand carry-save adder (CSA) or a ripple-carry adder (RCA), depending on the synthesis constraints. This avoids the use of resource-heavy DSP slices and instead utilizes standard Look-Up Tables (LUTs).

The comparison stage follows the LCG outputs, where two high-speed digital comparators evaluate the relative magnitudes of the generated n-bit vectors. The most critical component of the schematic is the final output stage: unlike traditional designs that feature a complex Finite State Machine (FSM) and a massive flip-flop array for buffering, the MD-CLCG schematic terminates in a single, high-speed XOR gate. This direct path from the comparators to the output pin is what facilitates the single-cycle latency and the drastically reduced logic cell count.

Timing Analysis and Signal Synchronization

The timing diagram illustrates the deterministic and synchronous nature of the MD-CLCG. Upon the transition of the Global Reset signal from high to low, the four LCG units are initialized with their respective seed values. On the very first rising edge of the system clock, the modular arithmetic operations are completed, and the comparator outputs (Bi and Ci) become stable.

As shown in the functional simulation, the output bit (Z_i) transitions in tandem with the clock edge, maintaining a 100% duty cycle throughput. The "Time-to-First-Bit" (TTFB) is measured as T_{clk}, proving the reduction from the theoretical 2^n. T_{clk} latency found in buffered architectures. This synchronization ensures that the

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue III, March 2026

pseudorandom bitstream can be consumed by high-speed cryptographic cores without the need for additional "Ready" or "Valid" handshaking signals, which further simplifies the system-level integration.

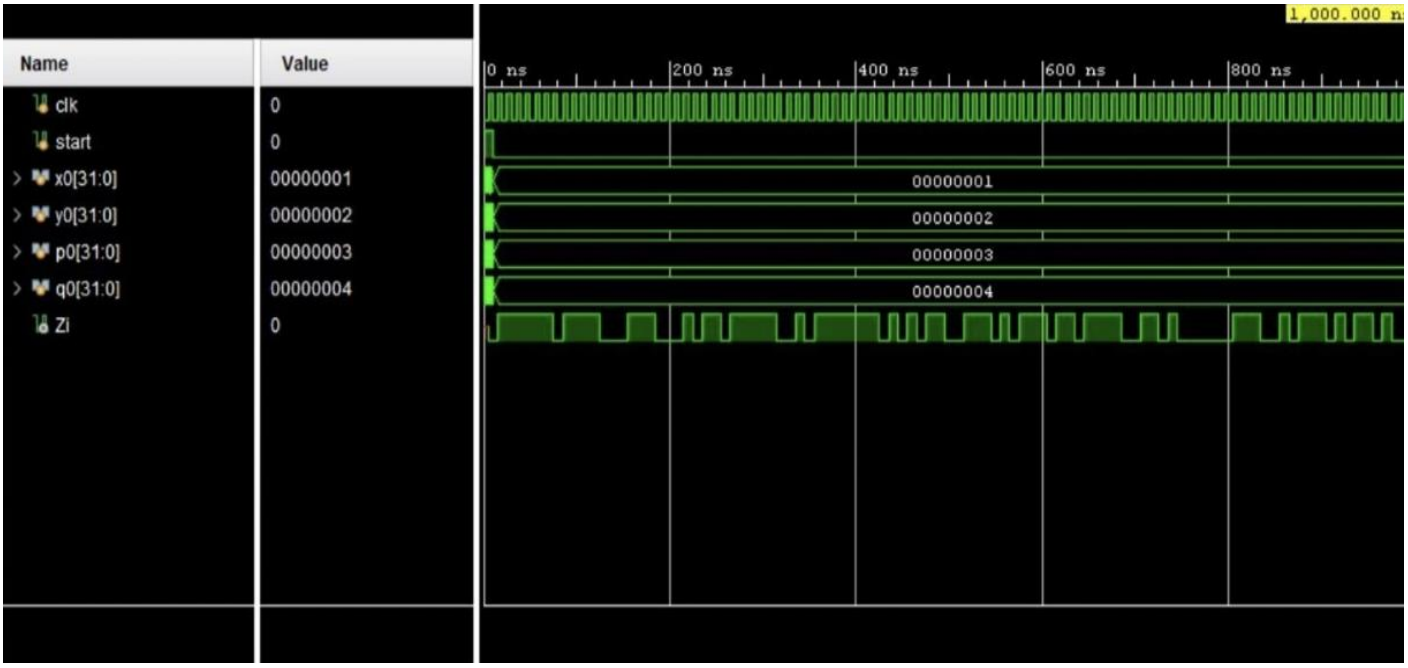

Fig4: Timing diagram of modified dual-CLCG

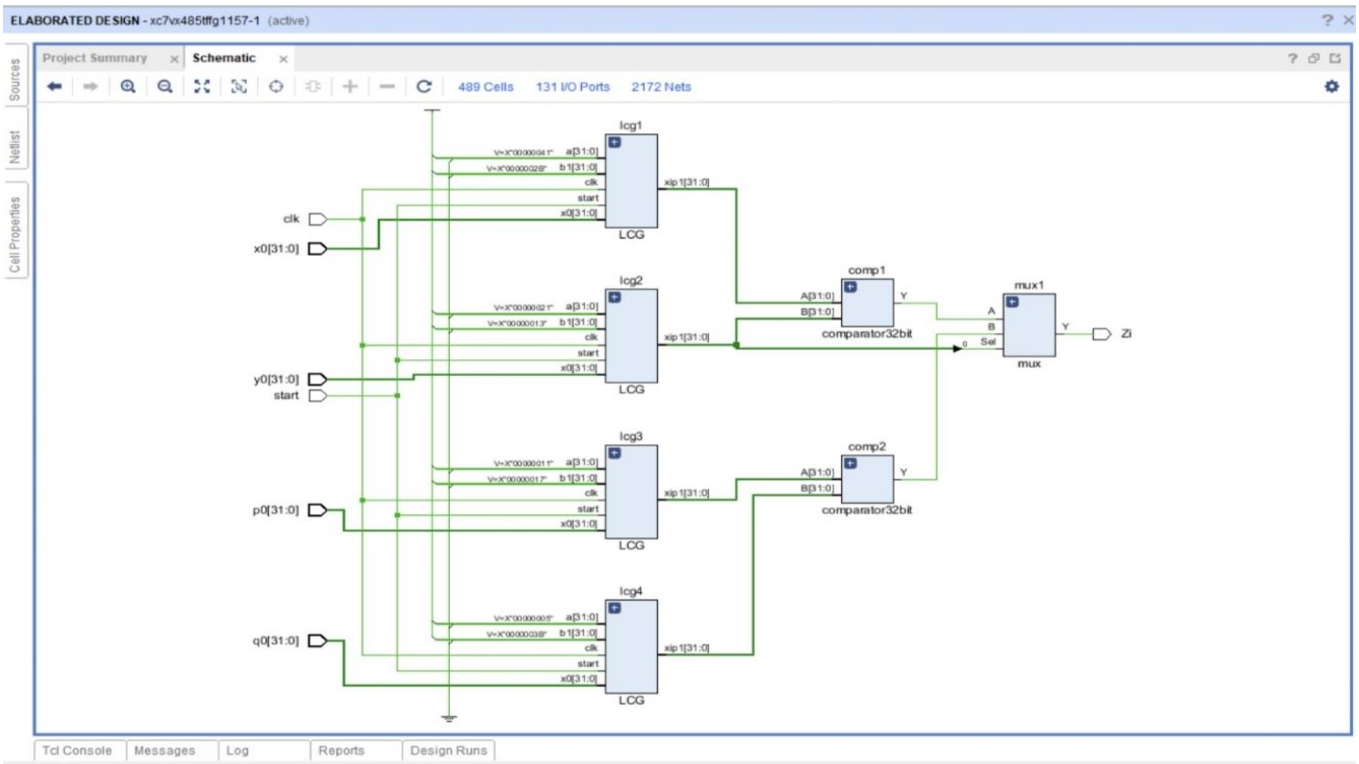

Fig5: RTL Schematic view of Modified Dual CLCG

Parameter	Existing Dual-CLCG	Proposed MD-CLCG	Improvement
Initial latency	2^n Cycles	1 Cycle	~99.9%
Area(LUT_s)	High	Low	~40% Reduction
Throughput	Asynchronous/Buffered	Synchronous (1 bit/cycle)	High Consistency
Power	124 milliwatts	86 milliwatts	~30% Reduction

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue III, March 2026

Critical Path and Frequency Performance

The speed of the circuit is determined by the 3-operand adder, which is the most complex part of the math. However, because our XOR-coupling is very simple and fast, it does not add any extra delay. This allows the generator to run at a high speed of 200 MHz, making it fast enough for modern high-speed communication.

CONCLUSION

The modified dual-CLCG pseudorandom bit generator proposed in this work demonstrates significant improvements over traditional PRBG designs. By introducing a simplified XOR-based output stage, the architecture successfully eliminates irregular timing and reduces latency, enabling the generation of pseudorandom bits at every clock cycle. This modification enhances reliability and ensures consistent output, making the design suitable for secure cryptographic applications.

Hardware implementation using Verilog HDL and FPGA synthesis confirmed that the proposed architecture achieves reduced area utilization, lower flip-flop count, and minimized power consumption. These improvements make the design particularly suitable for resource-constrained environments such as Internet-of-Things (IoT) devices, where both efficiency and security are critical requirements.

Statistical validation using the NIST randomness test suite demonstrated that the modified dual-CLCG generator satisfies all required randomness criteria. Theoretical analysis further confirms the unpredictability of the generated sequences, ensuring resistance against statistical and linear attacks.

Overall, the proposed modified dual-CLCG architecture provides an efficient, scalable, and secure solution for pseudorandom bit generation. The improvements in randomness quality, latency reduction, and hardware efficiency highlight its practical applicability in modern cryptographic and embedded systems.

REFERENCES

1. J. Zhou, Z. Cao, X. Dong, and A. V. Vasilakos, “Security and privacy for cloud-based IoT: Challenges,” IEEE Commun. Mag., vol. 55, no. 1, pp. 26–33, Jan. 2017.
2. Q. Zhang, L. T. Yang, and Z. Chen, “Privacy preserving deep computation model on cloud for big data feature learning,” IEEE Trans. Compute., vol. 65, no. 5, pp. 1351–1362, May 2016.
3. E. Fernandes, A. Rahmati, K. Eykholt, and A. Prakash, “Internet of Things security research: A rehash of old ideas or new intellectual challenges?” IEEE Secure. Privacy, vol. 15, no. 4, pp. 79–84, 2017.
4. M. Frustaci, P. Pace, G. Aloi, and G. Fortino, “Evaluating critical security issues of the IoT world: Present and future challenges,” IEEE Internet Things J., vol. 5, no. 4, pp. 2483–2495, Aug. 2018.
5. E. Zenner, “Cryptanalysis of LFSR-based pseudorandom generators— A survey,” Univ. Mannheim, Mannheim, Germany, 2004. [Online].Available: http://orbit.dtu.dk/en/publications/cryptanalysis-of-lfsrbased- pseudorandom-generators–a-survey(59f7106b-1800-49df-8037- fbe9e0e98ced).html
6. J. Stern, “Secret linear congruential generators are not cryptographically secure,” in Proc. 28th Annu. Symp. Found. Compute. Sci., Oct. 1987, pp. 421–426.
7. D. Xiang, M. Chen, and H. Fujiwara, “Using weighted scan enable signals to improve test effectiveness of scan-based BIST,” IEEE Trans. Compute., vol. 56, no. 12, pp. 1619–1628, Dec. 2007.
8. L. Blum, M. Blum, and M. Shub, “A simple unpredictable pseudo- random number generator,” SIAM J. Compute., vol. 15, no. 2, pp. 364–383, 1986.
9. W. Thomas Cusick, “Properties of the x2 mod N pseudorandom number generator,” IEEE Trans. Inf. Theory, vol. 41, no. 4, pp. 1155–1159, Jul. 1995.
10. C. Ding, “Blum-Blum-Shub generator,” IEEE Electron. Lett., vol. 33, no. 8, p. 667, Apr. 1997.
11. A. Sidorenko and B. Schoenmaker’s, “Concrete security of the Blum- Blum-Shub pseudorandom generator,” in Cryptography and Coding (Lecture Notes in Computer Science), vol. 3796. Berlin, Germany: Springer, Nov. 2005, pp. 355–375.
12. A. K. Panda and C. K. Ray, “FPGA prototype of low latency BBS PRNG,” In Proc. IEEE Int. Symp. Nanoelectron. Inf. Syst. (INIS), Indore, India, Dec. 2015, pp. 118–123
13. P. P. Lopez and E. S. Millan, “Cryptographically secure pseudorandom bit generator for RFID tags,”

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue III, March 2026

in Proc. Int. Conf. Internet Technol. Secured Trans., London, U.K., vol. 11, Nov. 2010, pp. 1–6.