International Journal of Research Publication and Reviews, Vol 6, Issue 5, pp 17554-17561 May 2025

WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

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

Dr Ch . Raja Rao [1] , P. Gopichand[2], R . Avinash[3] , N.Shivaramakrishna[4]

[1].Associate Professor & COE, Department Electronics and Communication Engineering, Guru Nanak Institutions Technical Campus, Ibrahimpatnam)
[2,3,4,] Department of Electronics and Communication Engineering, Guru Nanak Institutions Technical Campus, Ibrahimpatnam)

ABSTRACT :

Pseudo-random bit generators (PRBGs) are fundamental components in various computational applications, including simulations, cryptography, and statistical sampling. Linear Congruential Generators (LCGs) and their combinations, such as Combined Linear Congruential Generators (CLCGs) and Dual-CLCGs, are widely used due to their simplicity and speed. However, existing Dual-CLCG methods often exhibit statistical weaknesses, predictable patterns, and limited period lengths, which restrict their suitability for security-sensitive or high-fidelity simulation tasks. This paper presents the design of a modified Dual-CLCG algorithm aimed at improving statistical randomness properties and extending the period length. The proposed method incorporates non-linear transformations and state mixing techniques applied to the outputs of two independent LCGs. We detail the design principles, algorithm, and theoretical analysis of expected improvements. Furthermore, we propose an architectural design for hardware or software implementation and analyze its complexity. Evaluation using standard statistical test suites, such as NIST SP 800-22, alongside performance and period analysis, demonstrates that the proposed modified Dual-CLCG significantly enhances randomness compared to the standard method, offering a more robust solution for applications requiring higher degrees of statistical quality.

.

KEYWORDS:. : Pseudorandom bit generator (PRBG), LCG, VLSI architecture ,CLCG

INTRODUCTION

With the rapid expansion of the Internet of Things (IoT), data security and privacy have become more crucial than ever. IoT devices often deal with sensitive user data and operate in distributed and sometimes vulnerable environments. Ensuring the confidentiality, integrity, and authenticity of data transmitted over these networks is a significant challenge. One of the core components in establishing a secure communication framework is the Pseudo-Random Bit Generator (PRBG). PRBGs are essential in various fields such as cryptography, secure communications, statistical simulations, and random sampling, where unpredictability and high-quality randomness are paramount. These generators are responsible for producing sequences of bits that exhibit the properties of true randomness. A high-quality PRBG ensures that the generated bit sequence is unpredictable and uniformly distributed, which is vital for encryption algorithms, key generation, and secure protocols. Need for Efficient Hardware-Based PRBGs As hardware resources are often limited in IoT devices, there is a growing demand for PRBGs that are not only secure and statistically robust but also efficient in terms of area, power consumption, and speed. Hardware-based implementations of PRBGs are particularly favorable in embedded and real-time applications because they offer high-speed operation, low latency, and better resistance to side-channel attacks compared to software-based solutions.
To address these challenges, this project proposes a hardware-efficient and statistically strong PRBG known as the Modified Dual Combined Linear Congruential Generator (Dual-CLCG). This new method significantly enhances the performance of existing PRBG architectures while maintaining strong cryptographic properties.

Overview of Modified Dual-CLCG PRBG

What is a Dual-CLCG?
A Linear Congruential Generator (LCG) is a simple algorithm for generating pseudo-random numbers using a recurrence relation. While LCGs are fast and easy to implement, their randomness properties can be limited. To overcome these limitations, Combined Linear Congruential Generators (CLCGs) use multiple LCGs with different parameters and combine their outputs. This approach significantly improves the statistical quality and extends the period of the generated sequence.
A Dual-CLCG uses two LCGs with distinct seeds and combines their outputs using mathematical operations such as addition or XOR to improve randomness. However, conventional dual-CLCG implementations often involve complex logic and can be resource-intensive.
Proposed Modification and Key Contributions

The Modified Dual-CLCG architecture proposed in this work introduces several key innovations:

1. Simplified Logic Design
   - The output stage uses a single XOR gate, minimizing logic complexity and reducing chip area and power consumption.
2. High-Efficiency Random Bit Generation
   - The generator produces a maximum sequence length of 2^n bits with only one clock cycle of initial latency, ensuring high throughput.
3. NIST-Compliant Randomness
   - The design passes all 15 statistical tests from the NIST SP800-22 test suite (version 2.1.2), validating its high-quality randomness and statistical soundness.
4. Polynomial-Time Unpredictability
   - The probabilistic analysis shows that determining the initial seed requires solving an n^{24} complexity problem, making the generator secure and unpredictable in polynomial time.
5. Flexible and Scalable Hardware Design
   - The architecture is implemented using Verilog HDL and synthesized on 90nm CMOS technology using Cadence tools, supporting various word sizes (8-bit, 16-bit, and 32-bit) to meet different application requirements.
6. Optimized Performance Metrics
   - The proposed design is evaluated for initial latency, maximum bit generation frequency, output-to-output latency, chip area utilization, and power dissipation, showing substantial improvements over conventional PRBG designs.

Combined linear congruential generators, as the name implies, are a type of PRNG (pseudorandom number generator) that combine two or more LCGs (linear congruential generators). The combination of two or more LCGs into one random number generator can result in a marked increase in the period length of the generator which makes them better suited for simulating more complex systems. The combined linear congruential generator algorithm is defined

$$Xi \equiv \left(\sum_{j=1}^{K} (-1)^{j-1} Y_{ij}\right) (\mathrm{mod}(m1 - 1))$$

Where m1m1 is the modulus of the LCG, Yi, jYi, j is the iith input from the jjth LCG and XiXi is the iith random generated value. L'Ecuyer describes a combined linear generator that utilizes two LCGs in Efficient and Portable Combined Random Number Generators for 32-bit processors. Algorithm for a PRBG. The objective of this modification is to improve the efficiency and statistical properties of the generated random bit sequences. The Dual-CLCG algorithm combines the outputs of two Linear Congruential Generators (LCGs) with distinct seeds to produce random bits. Linear Congruential Generators are a common choice for generating pseudo-random sequences due to their simplicity and speed. However, their periodicity and statistical properties can be limited when used individually. The Dual-CLCG algorithm overcomes these limitations by leveraging the combined power of two LCGs while carefully selecting seed values and parameters.

EXISTING SYSTEM:

The Dual-Coupled Linear Congruential Generator (Dual-CLCG) is a cryptographic pseudorandom bit generator designed to enhance security in applications like IoT devices. Unlike standard LCG-based methods, it introduces inequality comparisons to generate bits at non-uniform intervals, improving randomness and security. However, the existing system struggles with high memory usage, latency issues, and irregular bit generation, making it inefficient for high-performance computing. Recent advancements have optimized the clock rate, reduced latency, and improved cryptographic strength, allowing it to successfully pass all 15 benchmark randomness tests from NIST, ensuring better security and hardware efficiency.

ARCHITECTURAL MAPPING OF THE EXISTING DUAL-CLCG METHOD

The dual-CLCG method is a dual coupling of four linear congruential generators proposed by Katti et al and is defined mathematically as follows:

$$x_{i+1} = a_1 \times x_i + b_1 \bmod 2^n \quad (1)$$

$$y_{i+1} = a_2 \times y_i + b_2 \bmod 2^n \quad (2)$$

$$p_{i+1} = a3 \times p_i + b3 \bmod 2^n \quad (3)$$

$$q_{i+1} = a4 \times q_i + b4 \bmod 2^n \quad (4)$$

$$Z_i = \begin{cases} 1 & \text{if } x_{i+1} > y_{i+1} \text{ and } p_{i+1} > q_{i+1} \\ 0 & \text{if } x_{i+1} < y_{i+1} \text{ and } p_{i+1} < q_{i+1} \end{cases} \quad (5)$$

The output sequence Zi can also be computed in an alternativeway as described in [15], i.e.,

Zi = Bi if Ci =0 (6)

Where,

$$B_i = \begin{cases} 1, & \text{if } x_{i+1} > y_{i+1} \\ 0, & \text{else} \end{cases}; \quad C_i = \begin{cases} 1, & \text{if } p_{i+1} > q_{i+1} \\ 0, & \text{else} \end{cases} \quad (7)$$

Here, a1, b1, a2, b2, a3, b3, a4 and b4 are the constant parameters; x0, y0, p0 and q0 are the initial seeds. Followingare the necessary conditions to get the maximum period.

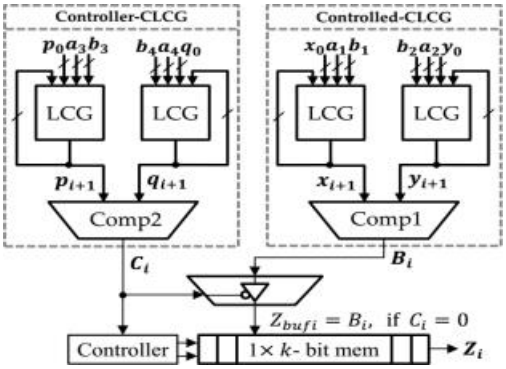


Fig. 1. Architectural mapping of the existing dual-CLCG method.

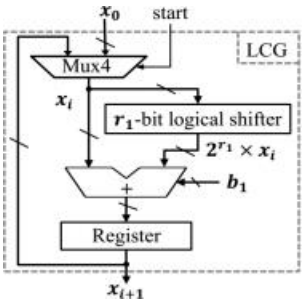


Fig. 2. Architecture of the linear congruential generator.

(i) b_1, b_2, b_3 and b_4 are relatively prime with $2^n(m)$.

(ii) (a_1-1), (a_2-1), (a_3-1) and (a_4-1) must be divisible by 4.

Following points can be observed from the dual-CLCG method i.e.

1. The output of the dual-CLCG method chooses the value of B_i when C_i is 'zero'; else it skips the value of B_i and does not give any binary value at the output.
2. As a result, the dual-CLCG method is unable togenerate pseudorandom bit at each iteration.

Drawbacks of the Existing Dual-CLCG Method

While the Dual-CLCG method offers improved security compared to basic LCG and standard CLCG approaches, it still comes with several limitations—especially when considering hardware efficiency and statistical performance. The key drawbacks are outlined below:

High Hardware Resource Requirement

The existing design relies heavily on control circuitry and flip-flops (memory elements). Specifically, it needs k flip-flops to generate k pseudorandom bits. Given that for a randomly chosen n-bit seed, the value of $k \approx 2^{n-1}$, the architecture requires 2^{n-1} flip-flops. This leads to significant hardware overhead, making it less efficient for compact or low-power systems like IoT devices.

Significant Initial Latency

The time delay between input and the first valid output—known as initial clock latency—is quite large. To generate the maximum sequence length, the dual-CLCG method requires 2^n clock cycles before producing the first output bit, which may not be acceptable for applications needing fast random bit generation.

Dependence on Seed Characteristics

The length of the pseudorandom sequence generated is not always optimal. It is affected by the number of zeros in the internal seed values (Ci), often resulting in a shorter-than-expected sequence, typically around 2^{n-1} for random seed inputs. This limits the randomness quality and reduces the usable range of the generator.

Assumption-Based Operation

The architecture assumes a balanced distribution of 0s and 1s in the sequence generated by the controller logic. This assumption doesn't always hold in practice, which may impact the consistency and reliability of the output bitstream.

Fails Key Randomness Tests

Perhaps most critically, the existing dual-CLCG implementation fails five of the fifteen key randomness tests in the NIST SP800-22 statistical test suite. These failures indicate that the bit sequences produced are not statistically random enough for cryptographic applications, undermining the generator's credibility in secure systems

PROPOSED SYSTEM:

The proposed system for the Dual-Coupled Linear Congruential Generator (Dual-CLCG) aims to overcome the limitations of the existing model by introducing a modified architecture that enhances randomness, reduces latency, and optimizes hardware efficiency. This improved system integrates a uniform clock rate for pseudorandom bit generation, ensuring minimal hardware complexity and one initial clock delay. Additionally, it employs a splitting structure in the adder circuit rather than a single adder, significantly improving timing performance while minimizing area overhead. These enhancements make the modified Dual-CLCG more suitable for cryptographic applications, IoT security, and high-performance computing.

Modified Dual-CLCG Using Carry-Save Binary Adder Technique

In modern cryptographic systems, especially for applications like secure IoT devices, efficient pseudorandom bit generation plays a crucial role. Linear Congruential Generators (LCGs) and their enhanced versions, such as CLCG, Dual-CLCG, and MDCLCG, are widely used for this purpose. A key operation in these systems is three-operand modular addition, which is essential for both randomness and performance. The proposed approach enhances the Dual-CLCG method by integrating a Carry-Save Adder (CSA) technique to improve speed, area, and randomness quality.

Three-Operand Addition: The Need for Efficiency

In traditional PRBG architectures, performing a three-operand binary addition is critical. This operation is common in modular arithmetic systems and significantly impacts the performance of pseudorandom bit generators. The naive approach uses two stages of two-operand adders, which is both slow and hardware-intensive.

To address this, a Carry-Save Adder (CSA) is employed. The CSA efficiently adds three binary numbers in two steps:

Carry-Save Stage: Each full adder in this stage receives three bits (ai, bi, ci) and simultaneously computes a sum bit and a carry bit. This results in two outputs: a "partial sum" and a "carry".

Final Addition: The carry output is left-shifted and added to the partial sum using a ripple carry adder (RCA) to produce the final result.

Despite its advantages, CSA introduces critical path delay due to the carry propagation in the final stage, which increases linearly with the bit-width. The total delay and area are given by:

Delay: $T_{CS3A} = (n+1)T_{FA} = 3T_X + 2NT_G$

Area : $A_{CS3A} = 2NA_{FA} = 4NA_X + 2NT_G$

Where T_G and A_G refer to the delay and area of basic logic gates and T_X and A_X correspond to XOR gates.

Addressing Delay: Parallel Prefix Adders

To mitigate the delay introduced by ripple carry in CSA, the architecture also considers parallel prefix adders like Han-Carlson (HC), which offer logarithmic delay characteristics. These adders significantly reduce critical path delays, particularly for larger bit-widths (n > 16).

Several fast adder architectures have been explored:

- Han-Carlson: Fast and area-efficient.
- Ultra-Fast Adder: Slightly faster than HC but uses twice the area.
- Hybrid Han-Carlson: Combines Brent-Kung and Kogge-Stone adders for a balanced performance between delay and gate count.

Modified Dual-CLCG with CSA Integration

The enhanced pseudorandom bit generator uses a modified Dual-CLCG design, where four LCG blocks operate in parallel. Each LCG follows a recurrence relation, taking in initial seeds x0,y0,p0,q0 and constants b1,b2,b3,b4b_1, b_2, b_3, b_4b1,b2,b3,b4, and uses shifting values r1,r2,r3,r4r_1, r_2, r_3, r_4r1,r2,r3,r4 to increase randomness.

Key Components of the Architecture:

Multiplexer-Controlled Seed Selection:

A 2:1 multiplexer chooses between initial seeds and iteratively updated values. This switching is controlled by a START signal, initiating the pseudorandom generation.

1. Binary Comparators:

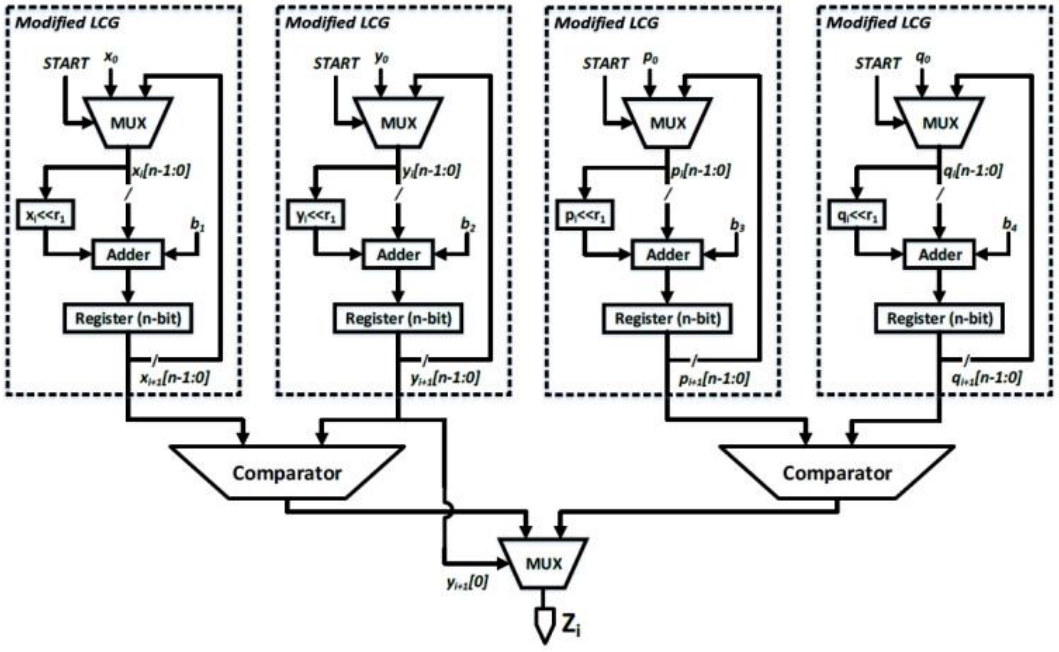
Two n-bit comparators evaluate iterative seed values (xi+1,yi+1) and (pi+1,qi+1) to generate control bits Bi_and Ci which influence the final output.

2. Output Bit generation using MUX and Comparators :
   A final multiplexer uses the least significant bit of y_{i+1} to select between B_i and C_i,enhancing unpredictability and randomness.

4.CSA-Based Addition for Each LCG Block:

Each LCG's computation incorporates CSA logic, accelerating the modular arithmetic operations required in each iteration.

BLOCKDIAGRAM:



System Architecture

The modified Dual-Coupled Linear Congruential Generator (Dual-CLCG) system architecture improves upon the traditional approach by incorporating four parallel Linear Congruential Generators (LCGs), each optimized with additional components such as multiplexers (MUX), adders, registers, and shift operations to enhance randomness and computational efficiency. Instead of relying on a single adder circuit, the design integrates a splitting structure in the adder circuit, effectively improving timing performance and reducing latency while minimizing overall hardware complexity. The inclusion of comparators ensures that pseudorandom bits are generated in a more structured manner, overcoming the irregularity observed in earlier models. Furthermore, a final multiplexer selects the output bit at a uniform clock rate, ensuring stability and predictability in cryptographic applications, particularly in IoT devices and secure communication systems. This refined system architecture allows for more efficient VLSI implementation, optimizing power consumption and area overhead without compromising security standards. With these improvements, the modified Dual-CLCG achieves a balance between computational feasibility and cryptographic strength, making it a valuable choice for modern applications demanding robust pseudorandom bit generation.

METHODOLOGY

The methodology for the modified Dual-Coupled Linear Congruential Generator (Dual-CLCG) is centered around optimizing the existing structure to achieve enhanced randomness, reduced latency, and improved hardware efficiency. This is accomplished by integrating four parallel Linear Congruential Generators (LCGs), each equipped with dedicated multiplexers (MUX), adders, registers, and shift operations. These components work in tandem to enhance computational accuracy and ensure the generation of high-quality pseudorandom bits. A key innovation in this approach is the use of a splitting structure in the adder circuit, replacing the traditional single adder design. This architectural change significantly improves timing performance by allowing for more efficient processing and reduced latency. Moreover, the splitting structure achieves these benefits while minimizing area overhead, making it an optimal solution for hardware implementation. The generation of pseudorandom bits in the modified Dual-CLCG follows a structured and well-defined process. Comparators play a crucial role in analyzing intermediate outputs to ensure that the generated bits exhibit randomness and stability. This structured approach overcomes the inconsistencies and irregularities observed in earlier designs, where bit generation could be unpredictable or biased. To further enhance the reliability of the output, a final multiplexer is employed to select the output bit at a uniform clock rate. This ensures that the pseudorandom bits are generated consistently, without the timing variations that could compromise the security or performance of cryptographic applications. The methodology also prioritizes VLSI (Very Large Scale Integration) implementation, with a focus on optimizing power consumption and reducing hardware complexity. By carefully balancing these factors, the modified Dual-CLCG achieves a design that is both efficient and secure. This makes it an ideal candidate for a wide range of cryptographic applications, including IoT security, secure data transmission, and high-performance computing. The refined structure of the modified Dual-CLCG allows it to deliver robust and efficient pseudorandom bit generation, meeting the stringent requirements of modern cryptographic systems. Its ability to balance computational feasibility with cryptographic strength ensures that it remains a valuable tool in the development of secure communication systems and other applications that rely on high-quality randomness. With these advancements, the modified Dual-CLCG stands out as a reliable and efficient solution for generating pseudorandom bits in demanding environments..

Results and simulation :

The RTL schematic is abbreviated as the register transfer level it denotes the blue print of the architecture and is used to verify the designed architecture to the ideal architecture that we are in need of development. The hdl language is used to convert the description or summery of the architecture to the working summery by use of the coding language i.e verilog ,VHDL. The RTL schematic even specifies the internal connection blocks for better analyzing. The figure represented below shows the RTL schematic diagram of the designed architecture

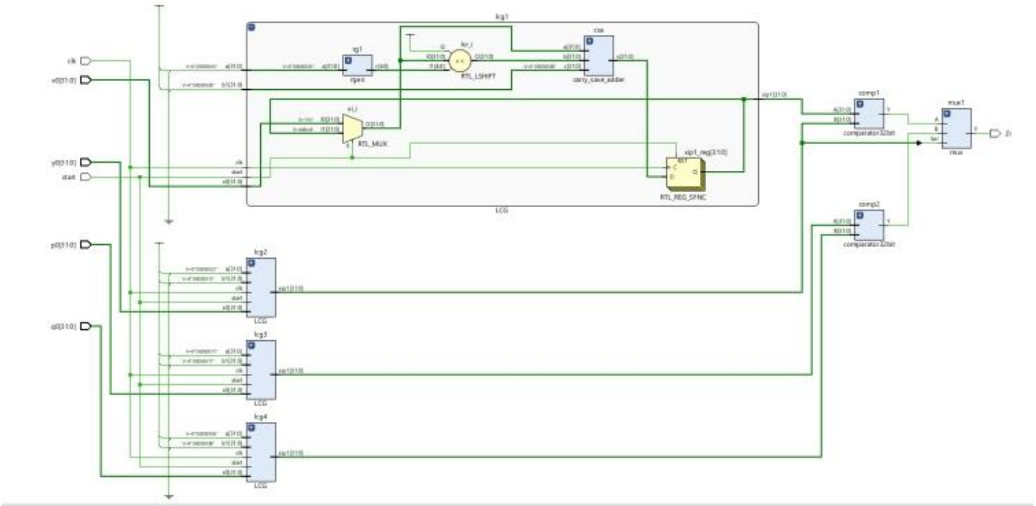


FIG :Schematic diagram for DUAL CLCG Random bit generator

Simulation

The simulation is the process which is termed as the final verification in respect to its working whereas the schematic is the verification of the connections and blocks. The simulation window is launched as shifting from implementation to the simulation on the home screen of the tool, and the simulation window confines the output in the form of wave forms output. Here it has the flexibility of providing the different radix number systems.

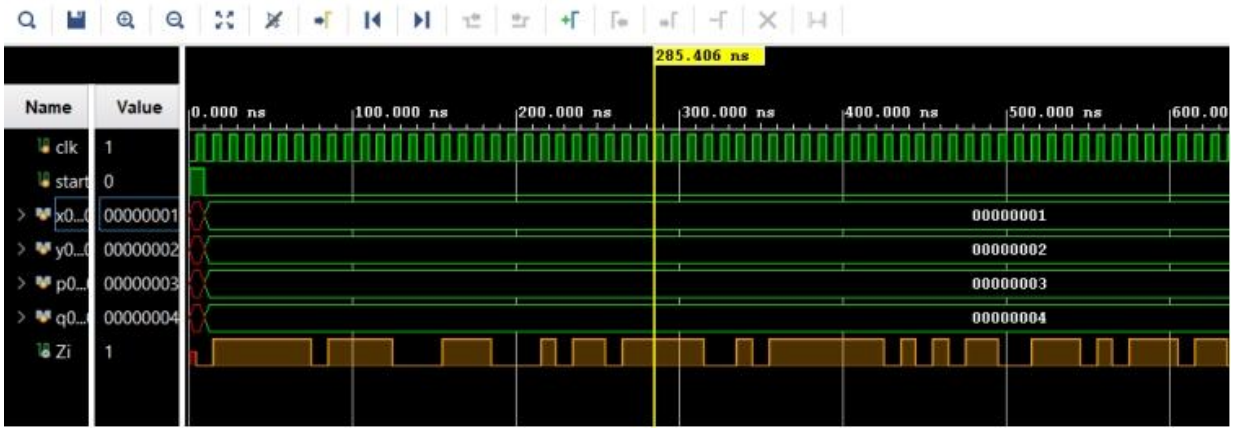


FIG : Simulation wave form of modified Dual CLCG using CSA

Other paarameters

Existed Parameters

```
Minimum period: 4.431ns (Maximum Frequency: 225.683MHz)
Minimum input arrival time before clock: 4.818ns
Maximum output required time after clock: 2.660ns
Maximum combinational path delay: No path found
```

Proposed Results

```
Minimum period: 7.535ns (Maximum Frequency: 132.714MHz)
Minimum input arrival time before clock: 7.610ns
Maximum output required time after clock: 2.666ns
Maximum combinational path delay: No path found
```

CONCLUSION

Modified Dual-CLCG using CS3A method involves dual coupling of four LCGs that makes it more secure than LCG based PRBGs. However, it is reported that this method has the drawback of generating pseudorandom bit at more delay. Proposed architecture of the new modified dual- CLCG method is significantly reduced the parameter. The proposed architecture of the modified dual- CLCG using three operand PPA method is working with high frequency resultant it would be reduced the delay of the design . Based on the performance analysis in terms of hardware complexity, randomness and security, it is observed that 32- bit hardware architecture of the proposed modified dual- CLCG method is optimum and can be useful in the speed of hardware security and IoT applications.

ACKNOWLEDGMENT

We thank all the authors for their outstanding assistances in this paper.

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 computa- tion model on cloud for big data feature learning,” IEEE Trans. Comput., 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 Secur. 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. Comput. 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. Comput., 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. Comput., 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] Sidorenko and B. Schoenmakers, "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] 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," in Proc. Int. Conf. Internet Technol. Secured Trans., London, U.K., vol. 11, Nov. 2010, pp. 1–6.

[14] R. S. Katti and R. G. Kavasseri, "Secure pseudo-random bit sequence generation using coupled linear congruential generators," in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), Seattle, WA, USA, May 2008, pp. 2929–2932.