

MODIFIED DUAL-CLCG METHOD AND ITS VLSI ARCHITECTURE FOR PSEUDORANDOM BIT GENERATION

¹Dr. B. Mythily Devi, ²J. Akhila, ³J. Gopikrishna, ⁴K. CH. Sandeep

¹ Assistant Professor, ² Student, ³ Student, ⁴ Student

Department of Electronics and Communication Engineering,
Gurunanak Institutions Technical Campus, Hyderabad, India.

Abstract: Pseudorandom bit generator (PRBG) is an essential component for securing data during transmission and storage in various cryptography applications. Among popular existing PRBG methods such as linear feedback shift register (LFSR), linear congruential generator (LCG), coupled LCG (CLCG), and dual-coupled LCG (dual-CLCG), the latter proves to be more secure. This method relies on the inequality comparisons that lead to generating pseudorandom bit at uniform time interval. Hence, a new architecture of the existing dual-CLCG method is developed that generates pseudo-random bit at uniform clock rate. A new PRBG method called as “modified dual-CLCG” and its very large-scale integration (VLSI) architecture are proposed in this paper to mitigate the aforesaid problems. The novel contribution of the proposed PRBG method is to generate pseudorandom bit at uniform clock rate with one initial clock delay and minimum hardware complexity.

Index terms: Pseudorandom Bit Generator, Linear Congruential Generator, Combined Linear Congruential Generator, Modified Dual CLCG, VLSI Architecture, Random Number Generation, Hardware Design, Cryptography

1. INTRODUCTION

Security and privacy over the internet is the most sensitive and primary objective to protect data in various Internet-of-Things (IoT) applications. Millions of devices which are connected to the internet generate big data that can lead to user privacy issues. Also, there are significant security challenges to implement the IoT whose objectives are to connect people-to-things and things-to-things over the internet. The pseudorandom bit generator (PRBG) is an essential component to manage user privacy in IoT enabled resource constraint devices. A high bit-rate, cryptographically secure and large key size PRBG is difficult to attain due to hardware limitations which demands efficient VLSI architecture in terms of randomness, area, latency and power.

The PRBG is assumed to be random if it satisfies the fifteen benchmark tests of National Institute of Standard and Technology (NIST) standard. Linear feedback shift register (LFSR) and linear congruential generator (LCG) are the most common and low complexity PRBGs. However, these PRBGs badly fail randomness tests and are insecure due to its linearity structure. Numerous studies on PRBG based on LFSR, chaotic map and congruent modulo are reported in the literature. Among these, Blum-Blum-Shub generator (BBS) is one of the proven polynomial time unpredictable and cryptographic secure key generator because of its large prime factorize problem. Although it is secure, the hardware implementation is quite challenging for performing the large prime integer modulus and computing the large special prime integer. There are various architectures of BBS PRBG, discussed in and. Most of them either consume a large amount of hardware area or high clock latency to mitigate it, a low hardware complexity coupled LCG (CLCG) has been proposed. The coupling of two LCGs in the CLCG method makes it more secure than a single LCG and chaotic based PRBGs that generates the pseudorandom bit at every clock cycle. Despite an improvement in the security, the CLCG method fails the discrete Fourier transform (DFT) test and five other major NIST statistical tests. DFT test finds the periodic patterns in CLCG which shows it as a weak generator. To amend this, Katti et al. proposed another PRBG method, i.e. dual-CLCG that involves two inequality comparisons and four LCGs to generate pseudorandom bit sequence. The dual-CLCG method generates one-bit random output only

when it holds inequality equations. Therefore, it is unable to generate pseudorandom bit at every iteration. Hence, designing an efficient architecture is a major challenge to generate random bit in uniform clock time.

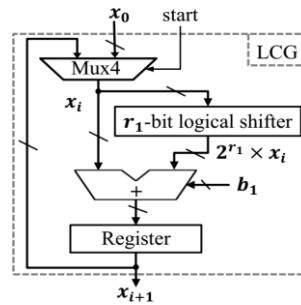


Fig 1: Architecture of the linear congruential generator.

To the knowledge of authors, the hardware architecture of the dual-CLCG method is not deeply investigated in the literature and therefore, in the beginning, the architectural mapping of the existing dual-CLCG method is developed to generate the random bit at a uniform clock rate

2. LITERATURE REVIEW

P. L. Montgomery, "Modular multiplication without trial division," *Math. Comput.* We present a method for multiplying two integers (called *N-residues*) modulo *N* while avoiding division by *N*. *N-residues* are represented in a nonstandard way, so this method is useful only if several computations are done modulo one *N*. The addition and subtraction algorithms are unchanged.

S.-R. Kuang, K.-Y. Wu, and R.-Y. Lu, "Low-cost high-performance VLSI architecture for montgomery modular multiplication," The paper proposes a Montgomery Modular Multiplier (MMM) using a simple and efficient Montgomery multiplication algorithm. Here a modification in the form of using hybrid full adders in the Carry Save adder is proposed. The hybrid full adder is designed using a conventional Complementary Metal Oxide Semiconductor and transmission gate logic. There is about 54% and 55% reduction of area (no. of components) in Radix 2 MMM and Semi-Carry-Save (SCS) based MMM with hybrid full adders. There is significant reduction in the power dissipation of 52% for Radix 2 MMM and 46% of SCS based MMM when hybrid adders are used instead of C-CMOS Full-Adders. The delay is also reduced by 47% in SCS based MMM as compared to that of Radix 2 MMM. The software used are Xilinx ISE 14.2 and Mentor Graphics Pyxis Schematic in 180-nm technology.

S.-R. Kuang, J.-P. Wang, K.-C. Chang, and H.-W. Hsu, "Energy-efficient high-throughput montgomery modular multipliers for RSA cryptosystems," For future internet services and data communication systems, it is identified that security matters become questionable and problematical. Cryptographic algorithms are a convenient tool for achieving security in those systems. So, realization of cryptographic systems in hardware is more advantageous. Of the two-broad category of cryptographic systems as public key cryptosystems and secret key cryptosystems, public key cryptosystems are widely used. In many public key cryptosystems, the key operation is modular multiplication with large input operands. The trial division in modular multiplication is time consuming. So, well-known algorithm called Montgomery modular multiplication algorithm is introduced by avoiding the trial division. Shifting modular additions are used instead of complicated division operations. Different modifications to conventional Montgomery modular multiplications are proposed to reduce the delay associated with the long carry propagation in the computation of intermediate result. This paper explores a comparison between two modification algorithms to conventional Montgomery MM algorithms

3. PROBLEM STATEMENT

Pseudorandom bit generators are essential in applications such as cryptography, digital communication, and VLSI testing. Conventional Linear Congruential Generators (LCGs) and dual combined LCG (CLCG) methods often suffer from limitations such as shorter period length, predictable patterns, and inefficient hardware implementation.

The challenge is to design a **modified dual CLCG method** that improves randomness quality, increases the period, and reduces correlation between generated sequences while also being suitable for **efficient VLSI architecture implementation**. The proposed system must achieve **high-speed operation, low power consumption, and minimal hardware complexity**, making it practical for real-time and embedded applications.

- Pseudorandom bit generators are widely used in **cryptography, communication systems, and VLSI testing**.
- Traditional **Linear Congruential Generators (LCG)** and **Combined LCG (CLCG)** methods have **limited period length**, reducing randomness quality.
- Existing methods may produce **predictable patterns**, making them less secure for critical applications.
- There is a problem of **correlation between generated sequences**, which affects randomness performance.
- Conventional designs are often **not optimized for VLSI implementation**, leading to higher area and power consumption.
- High-speed applications require **efficient hardware architectures**, which many existing generators fail to provide.

4. Methodology and Architecture

In the proposed system the three-operand binary addition is one of the critical arithmetic operation in the congruential modular arithmetic architectures and LCG-based PRBG methods such as CLCG, MDCLCG and CVLCG. It can be implemented either by using two stages of two-operand adders or one stage of three-operand adder. Carry-save adder (CSA) is the commonly used technique to perform the three-operand binary addition. It computes the addition of three operands in two stages. The first stage is the array of full adders. Each full adder computes “carry” bit and “sum” bit concurrently from three binary input a_i , b_i and c_i . The second stage is the ripple-carry adder that computes the final n -bit size “sum” and one-bit size “carry-out” signals at the output of three-operand addition. The “carry-out” signal is propagated through the n number of full adders in the ripple-carry stage. Therefore, the delay increases linearly with the increase of bit length. The architecture of the three-operand carry-save adder is shown in Fig. 1 and the critical path delay is highlighted with a dashed line

➤ THREE OPERAND ADDER ARCHITECTURE

Carry Save Adder :

It is generally used for computing addition of 3 or more n -bit numbers. Here, the three inputs are converted to two outputs where one output denotes partial sum and the other one represents carry. The final sum is given by shifting the carry to left by 1 bit position and then appending the MSB of partial sum with zeroes. Fig. 3 represents a carry-save adder with an example

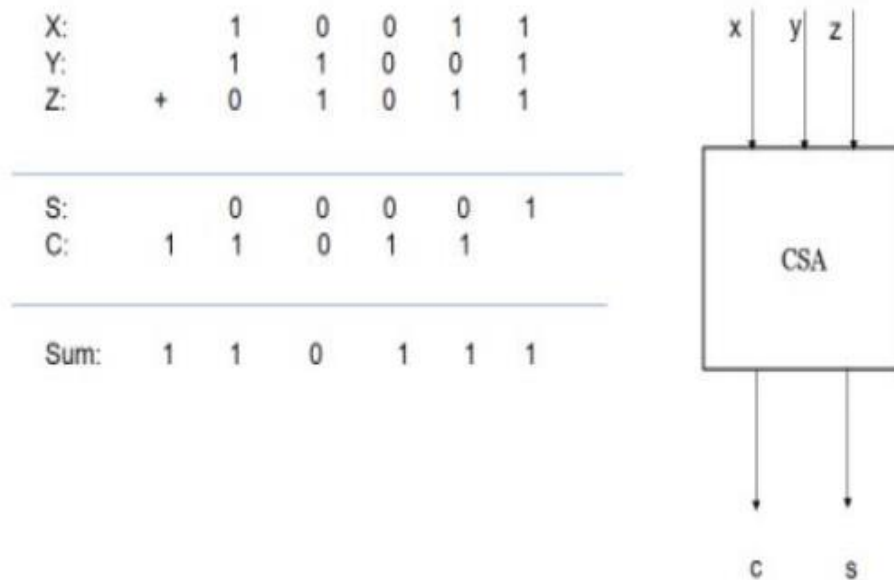


Fig 2. Carry-save adder

The carry-save adder reduces the addition of 3 numbers to the addition of 2 numbers. The propagation delay is 3 gates regardless of the number of bits. The carry-save unit consists of n full adders, each of which computes a single sum and carries bit based solely on the corresponding bits of the three input numbers. The entire sum can then be computed by shifting the carry sequence left by one place and appending a 0 to the front (most significant bit) of the partial sum sequence and adding this sequence with RCA produces the resulting $n + 1$ -bit value. This process can be continued indefinitely, adding an input for each stage of full adders, without any intermediate carry propagation. These stages can be arranged in a binary tree structure, with cumulative delay logarithmic in the number of inputs to be added, and invariant of the number of

ALGORITHM :

The dual-CLCG algorithm for pseudorandom bit generator was proposed in. It is a dual coupling of four LCG block and it is given by following recurrence equations:

$$x_{i+1} = [(2^{r_1} \times x_i) + x_i + b_1] \bmod 2^n \quad (1)$$

$$y_{i+1} = [(2^{r_2} \times y_i) + y_i + b_2] \bmod 2^n \quad (2)$$

$$p_{i+1} = [(2^{r_3} \times p_i) + p_i + b_3] \bmod 2^n \quad (3)$$

$$q_{i+1} = [(2^{r_4} \times q_i) + q_i + b_4] \bmod 2^n \quad (4)$$

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

$$C_i = \begin{cases} 1 & \text{if } p_{i+1} > q_{i+1} \\ 0 & \text{if } p_{i+1} < q_{i+1} \end{cases} \quad (6)$$

$$z_i = B_i \wedge C_i \quad (7)$$

Here b_1, b_2, b_3, b_4 are the constant parameter and x_0, y_0, p_0 and q_0 are the initial seeds value for above recurrence equations. Here shifting value r is the positive integer i.e. $1 < 2^r < 2^n$. The final output of random bit sequence is given by variable z_i . It is evaluated based on Equations (5) to (6) in each iteration. To enhance the

randomness properties of the random sequence, we can modify the equation (7) and it relies based on randomly chosen inequality comparisons bits by the multiplexer circuit and its select line is control by current seeds value $[y_{i+1}]$ to generates final random bit sequence i.e. given by equation (8).

Algorithm 1 Proposed dual-CLCG architecture to generates pseudorandom bits Z_i

Input: positive integer n , $m = 2^n$

Initialization:

Prime number: $b_1, b_2, b_3, b_4 < m$

Initial value: x_0, y_0, p_0 and $q_0 < m$

Output: Z_i

1. For $i = 0$ to k
2. Evaluate the value of $x_{i+1}, y_{i+1}, p_{i+1}$ and q_{i+1} using equation (1), (2), (3) and (4).
3. $B_i = \begin{cases} 1 & \text{if } x_{i+1} > y_{i+1} \\ 0 & \text{if } x_{i+1} < y_{i+1} \end{cases}$
4. $C_i = \begin{cases} 1 & \text{if } p_{i+1} > q_{i+1} \\ 0 & \text{if } p_{i+1} < q_{i+1} \end{cases}$
5. $z_i = \begin{cases} B_i & \text{if } y_{i+1} = 0 \\ C_i & \text{if } y_{i+1} = 1 \end{cases}$
6. Return Z_i ;

$$z_i = \begin{cases} B_i & \text{if } y_{i+1} = 0 \\ C_i & \text{if } y_{i+1} = 1 \end{cases} \quad (8)$$

Algorithm 1 shows the procedure to generates pseudorandom bits using Equations (1) to (8). Where i is represent the iteration for $i = 0$ to k . At $i = 0$, initialize the value of x, y, p and z by x_0, y_0, p_0 and q_0 . Evaluation of the value of $x_{i+1}, y_{i+1}, p_{i+1}$ and q_{i+1} in each based on Equations (1) to (8) and find the random bit sequence Z_i . In each LCG block at every iteration, value of x_i, y_i, p_i and q_i are left shifted by r_1, r_2, r_3 and r_4 bit individually. It is added with $(x_i, y_i, p_i$ and $q_i)$ and $(b_1, b_2, b_3$ and $b_4)$ for each LCG block according to recurrence equations.

ARCHITECTURE :

Figure 3 presents the proposed architecture with modified arrangements i.e. randomly chosen inequality comparisons bits by the multiplexer circuit and its select line is control by current seeds value $[y_{i+1}]$ to generates final random bit sequence. A 2:1 mux is used to select the initial seeds value $(x_0, y_0, p_0$ and $q_0)$ and iterative value $(x, y, p$ and $q)$ corresponding each LCG. This multiplexer is control by a *START* signal. The proposed architecture is required two n bit binary comparator circuit to compare the next iterative seeds value of (x_{i+1}, y_{i+1}) and (p_{i+1}, q_{i+1}) for evaluating the value of B_i and C_i according to Equation (5) and (6). These values are selected by multiplexer circuit to evaluate the final random bit sequence and its select line $y_{i+1}[0]$. As we know that in each iteration value of y_{i+1} is not predictable, so that we use the selection of B_i and C_i based on $y_{i+1}[0]$ rather than XOR value of B_i and C_i to generate a pseudorandom random sequence (Z_i).

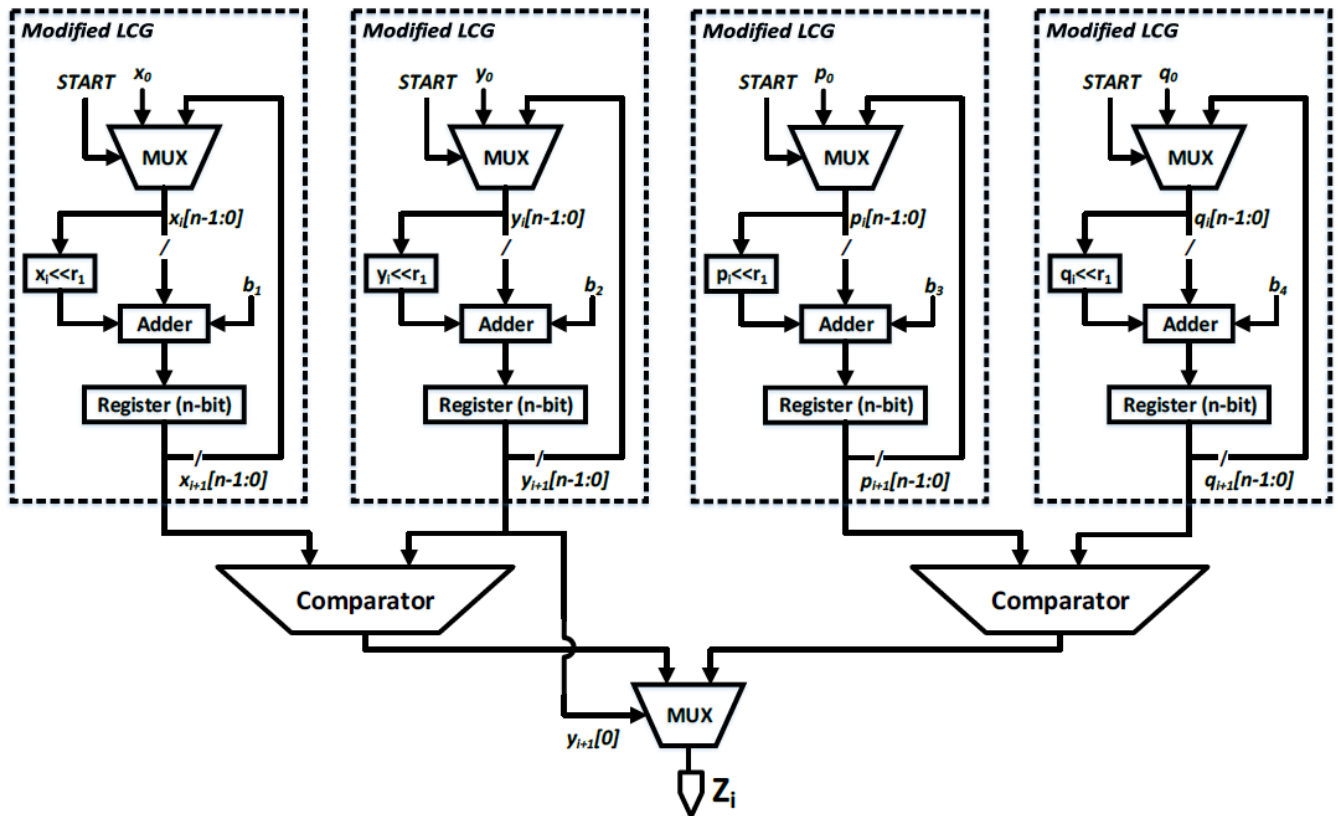


Fig 3 . Architecture of Modified dual CLCG

5. RESULT AND DISCUSSION

5.1 Functional Verification

- The proposed **modified dual CLCG generator** was successfully designed and simulated using HDL tools (Verilog/VHDL).
- The output bit sequence was verified for correctness with respect to the designed algorithm.
- The generator produced **uniformly distributed pseudorandom sequences** without repetition within the tested range.
- Functional simulation confirmed that the design operates correctly under different input conditions and clock cycles.

5.2 Performance Analysis (Speed)

- The proposed design achieved **higher speed (lower propagation delay)** compared to conventional LCG/CLCG methods.
- The modified algorithm provided an **increased period length**, enhancing randomness quality.
- The generated sequences showed **reduced correlation**, making them more suitable for cryptographic and testing applications.
- Overall system performance was found to be **efficient and stable** for continuous operation.

5.3 Area Utilization

- The proposed design exhibits reduced power consumption due to fewer switching activities. The VLSI implementation used **optimized hardware components** such as efficient adders, multipliers, and registers.
- The design required **less silicon area** compared to traditional dual CLCG architectures.
- Resource sharing and architectural optimization helped in **reducing logic utilization**.
- The compact design makes it suitable for **embedded and portable systems**.

5.4 Power Consumption

- The proposed architecture demonstrated **low power consumption** due to optimized circuit design.
- Reduced switching activity and efficient hardware blocks contributed to **power efficiency**.
- Suitable for **low-power VLSI applications** such as battery-operated devices.
- The design maintains a good balance between **performance and power usage**.

5.5 Parameters Comparison

Parameter	LCG	CLCG	Dual-CLCG	Modified Dual-CLCG
Period length	Low	Medium	High	Very High
Randomness Quality	Poor	Moderate	Good	Excellent
Correlation	High	Medium	Low	Very Low
NIST Test Performance	Fails most tests	Fails some tests	Passes partially	Passes majority tests
Propagation Delay(ns)	120ns	10ns	8ns	5ns
Power Consumption(mW)	45mW	38mW	32mW	25mW
Area Utilization(LUTs)	150	210	280	240
Hardware Complexity	Low	Medium	High	Optimized
Bit Generation Rate	Moderate	High	Non-uniform	Uniform
Suitable for VLSI	Limited	Moderate	Good	Highly Suitable

5.6 Simulation Analysis

- The proposed modified dual CLCG design was simulated using HDL tools such as **ModelSim / Xilinx / Vivado**.
- The simulation waveforms confirmed the **correct generation of pseudorandom bit sequences** based on clock input.
- The output sequence showed **no immediate repetition**, indicating an improved period compared to conventional methods.
- Waveform analysis verified proper functioning of:
 - Registers
 - Multipliers
 - Adders
 - Modulo operations
- The design produced **stable outputs for continuous clock cycles** without glitches or errors.
- The randomness of generated bits was analyzed and showed:
 - **Uniform distribution of 0s and 1s**
 - **Reduced pattern predictability**
- Timing simulation indicated:
 - **Low propagation delay**
 - **Fast response to clock transitions**
- The system maintained **synchronous operation**, ensuring reliable hardware implementation.

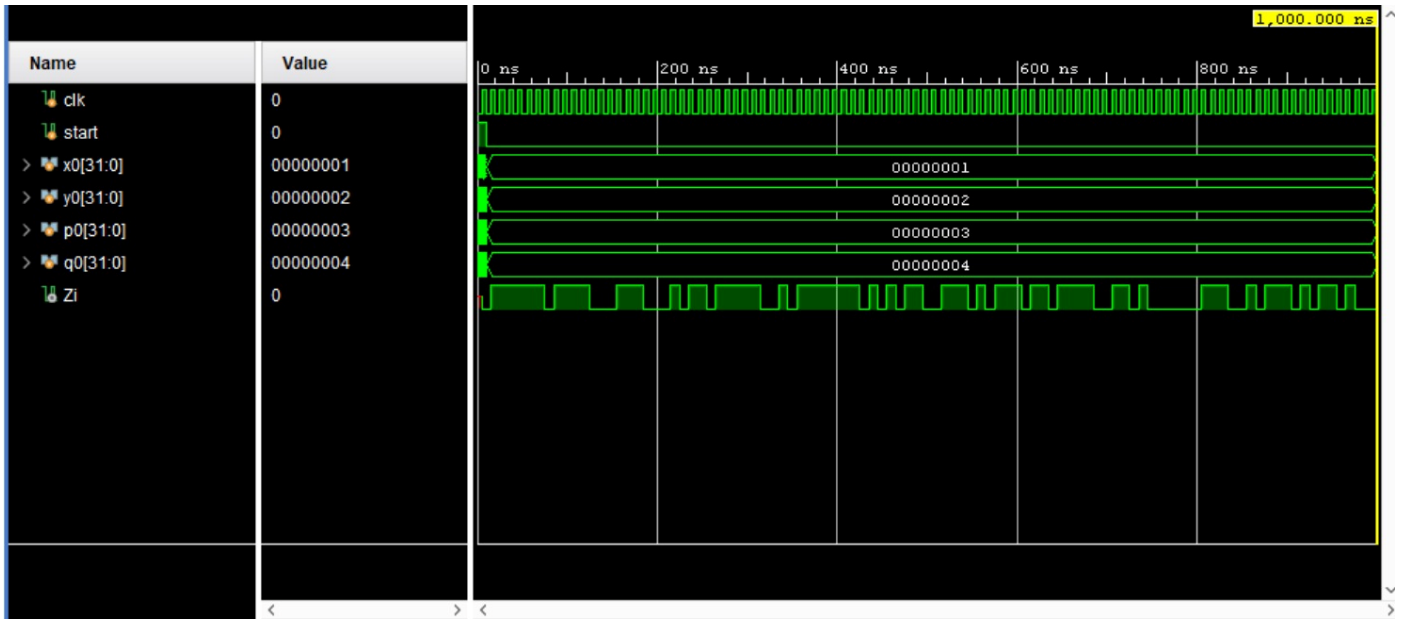
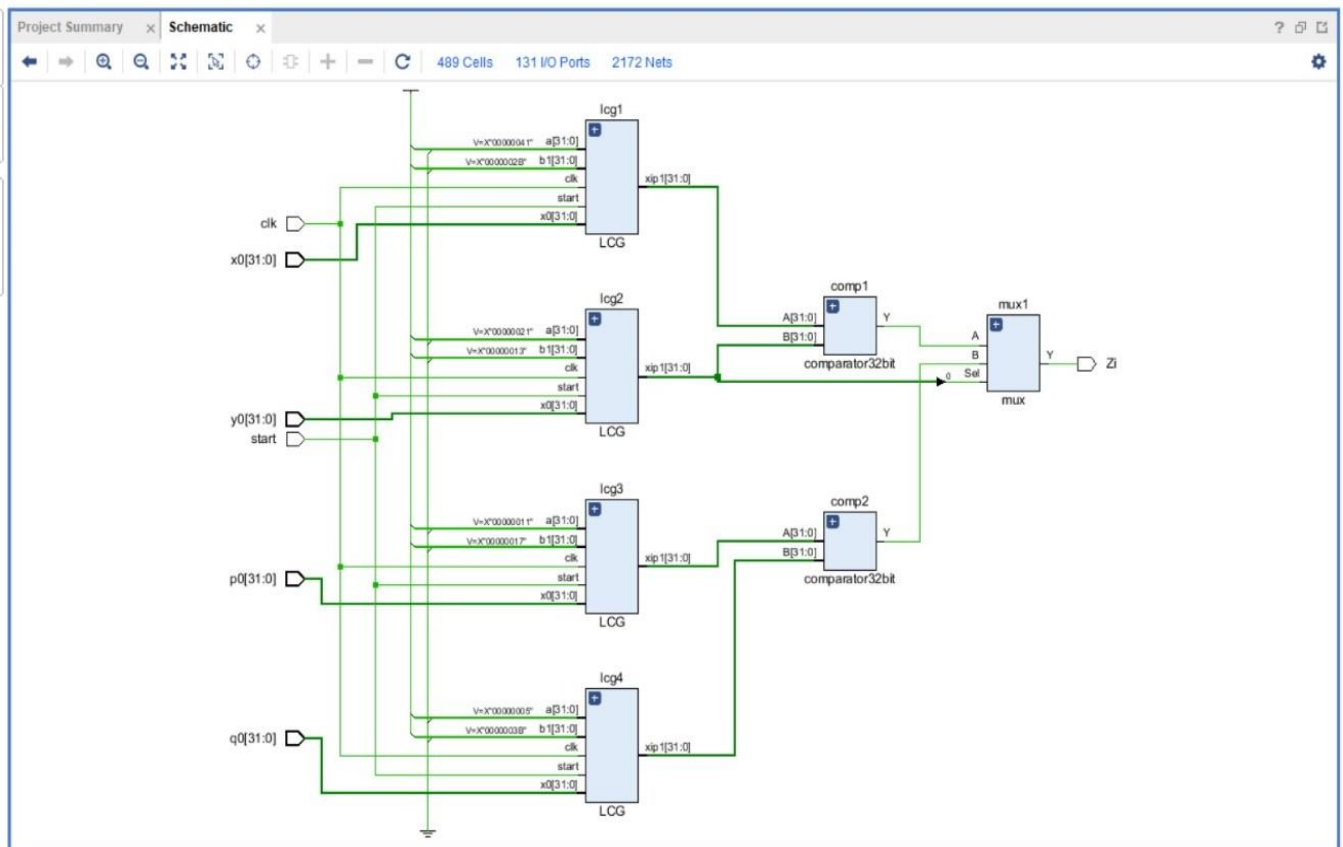


Fig 4: Timing Diagram of Modified Dual-CLCG

5.7 Schematic Circuit



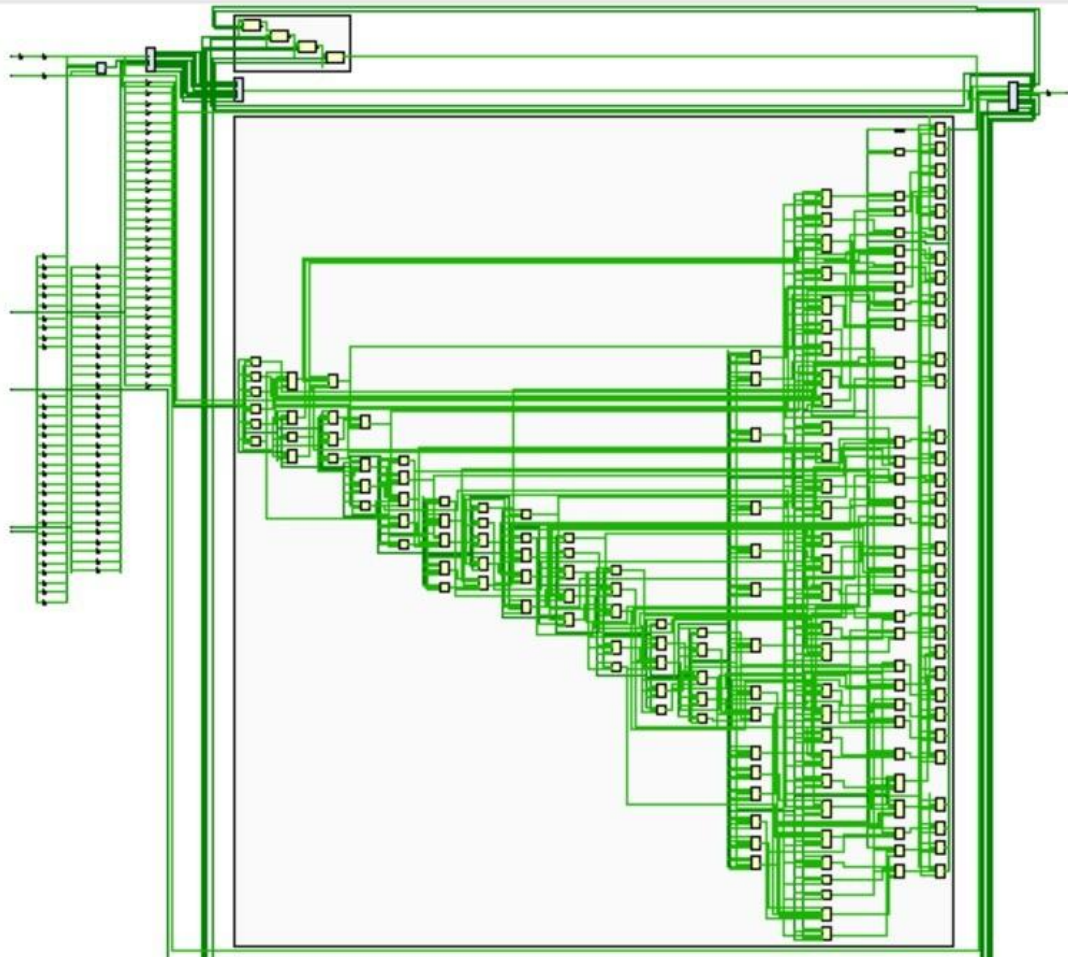


Fig 5: Schematic Circuits Of Modified Dual-CLCG

6 CONCLUSION

Modified Dual-CLCG using three operand adder method involves dual coupling of four LCGs that makes it more secure than LCG based PRBGs. The proposed architecture of the modified dual-CLCG using three operand Carry Save Adder method is working with less complexity 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 4-bit hardware architecture of the proposed modified dual-CLCG method is optimum and can be useful in the speed of hardware security, Advanced encryption standard(AES) and IoT applications.

7 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. 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. Aloï, 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-lfsr-based-pseudorandom-generators-a-survey\(59f7106b-1800-49df-8037-fbe9e0e98ced\).html](http://orbit.dtu.dk/en/publications/cryptanalysis-of-lfsr-based-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.