ISSN: 2278-4632
Vol-10 Issue-9 No.03 September 2020

Efficient VLSI Architecture for PRBG Using Modified Dual-CLCG

ANJALIDEVI PONAMALA[1], SHAIK CHINNA BAJI[2]
[1]Assistant professor, MVR College of engineering and technology, Paritala 521180
[2]Assistant professor, MVR College of engineering and technology, Paritala 521180

Abstract

The transfer of information is done through wireless these days which made easy to communicate from anywhere easily, but to protect and secure the information from the external hacking world many procedures are invented which gives the compatibility of coding (enciphering). The efficient procedure is Pseudo random bit generator (PRBG). 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 a non-uniform time interval. A new PRBG method called as "Efficient vlsi architecture for PRBG using modified dual-CLCG" are proposed in this paper to claim the cons of previous structure. 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 (PRBG), LCG, VLSI architecture.

I. INTRODUCTION

The PRBG is assumed to be random if it satisfies the fifteen benchmark tests of National Institute of Standard anto 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.

Hence, to overcome these shortcomings in the existing dual-CLCG method and its architecture, a new PRBG method and its architecture are proposed in this paper. The manuscript mainly focuses on developing an efficient PRBG algorithm and its hardware architecture in terms of area, latency, power, randomness and maximum length sequence.

ISSN: 2278-4632
Vol-10 Issue-9 No.03 September 2020

This paper is organized as follows: architectural mapping of the existing dual-CLCG method is performed and The proposed PRBG method along with its randomness properties are discussed. he efficient VLSI architecture of the proposed modified dual-CLCG method. 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 as:

$$Xi \equiv (\sum_{j=1}^{k} (-1\dot{)}^{j-1} Y_{ij})(\mathrm{mod}(m1-1))$$

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.

II. LITERATURE SURVEY

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 a non-uniform time interval. Hence, a new architecture of the existing dual CLCG method is developed that generates pseudo-random bit at uniform clock rate. However, this architecture experiences several drawbacks such as excessive memory usage and high-initial clock latency, and fails to achieve the maximum length sequence. Therefore, 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. Moreover, the proposed PRBG method passes all the 15 benchmark tests of NIST standard and achieves the maximal period of 2^n.

The proposed architecture is implemented using Verilog-HDL and prototyped on the commercially available FPGA device. J. Stern, “Secret linear congruential generators are not cryptographically secure,” The dual-coupled-linear congruential generator (LCG) (dual-CLCG) is a secure pseudorandom bit generator (PRBG) method among various linear feedback shift register (LFSR), LCG, and chaotic-based PRBG methods for generating a pseudorandom bit sequence. The hardware implementation of this method has a bottleneck due to the involvement of inequality equations. Initially, a direct architectural mapping of the dual-CLCG method is performed. Since two inequality equations are involved for coupling, it generates pseudorandom

ISSN: 2278-4632
Vol-10 Issue-9 No.03 September 2020

bit at unequal interval of time that leads to large variation in output latency. In addition, it consumes a large area and fails to achieve the maximal period. Hence, to overcome the aforesaid drawbacks, a new efficient PRBG method, i.e., "coupled-variable input LCG (CVLCG)," and its architecture are proposed. The novelty of the proposed method is the coupling of two newly formed variable input LCGs that generates pseudorandom bit at every uniform clock rate, attains maximum length sequence, and reduces one comparator area as compared to the dual-CLCG architecture.

III. ARCHITECTURAL MAPPING OF THE EXISTING DUAL-CLCG METHOD

The dual-CLCG method is a dual coupling of four linear congruential generators proposed by Kattiet 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 alternative way as described in [15], i.e.,

$$Zi = Bi \text{ if } Ci = 0 \quad (6)$$

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. Following are the necessary conditions to get the maximum period.

ISSN: 2278-4632
Vol-10 Issue-9 No.03 September 2020

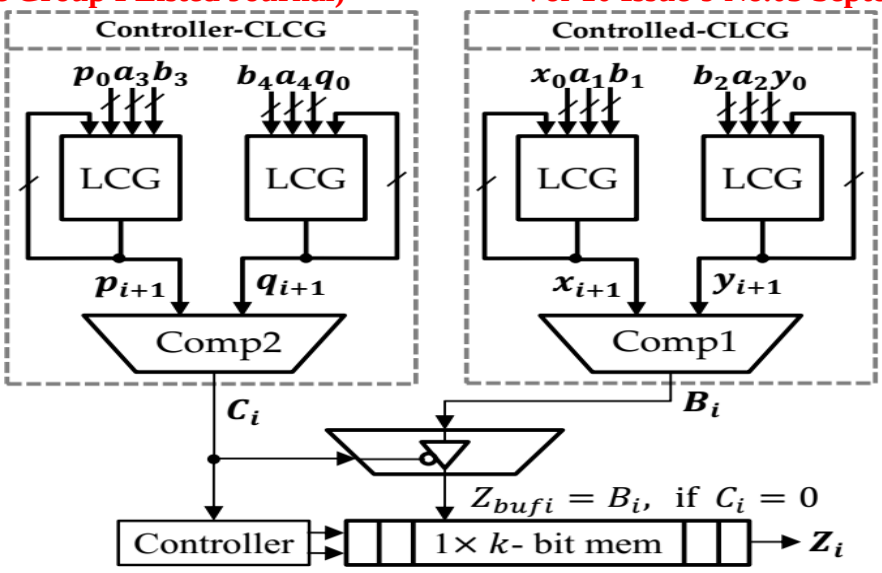


Fig. 1. Architectural mapping of the existing dual-CLCG method.

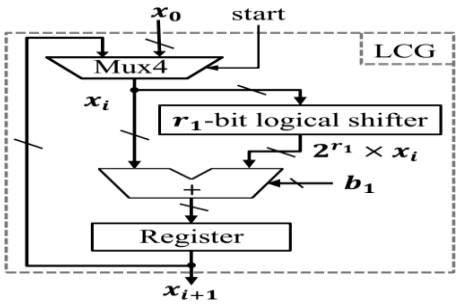


Fig. 2. Architecture of the linear congruential generator.

(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 Bi and does not give any binary value at the output.
2. As a result, the dual-CLCG method is unable to generate pseudorandom bit at each iteration.

Architecture Mapping of the Existing Dual-CLCG Method

The scope of the work presented in is limited to the algorithmic development. However, it lacks the architectural design of the dual-CLCG method. Hence, a new hardware architecture of the existing dual- LCG method is developed to generate pseudorandom bit at an equal interval of time for encrypting continuous data stream in the stream cipher. The architecture is designed with two comparators, four LCG blocks, one controller unit and memory (flip-flops) as shown in Fig. 1. The LCG is the basic functional block in the dual-CLCG architecture that involves multiplication and addition processes to compute n-bit binary random number on every clock

cycle. The multiplication in the LCG equation can be implemented with shift operation, when a is considered as (2^r+1). Here, r is a positive integer, $1 < r < 2^n$. Therefore, for the efficient computation of xi+1, the equation (1) can be rewritten as,

$$xi+1 = (a1 \times xi + b1) \bmod 2^n = [(2^{r1} + 1)xi + b1] \bmod 2^n$$
$$= [(2^{r1} \times xi) + xi + b1] \bmod 2^n$$

The architecture of LCG shown in Fig. 2 is implemented with a 3-operand modulo 2n adder, 2 × 1 n-bit multiplexer and n-bit register. LCG generates a random n-bit binary equivalent to integer number in each clock cycle. Other three LCG equations can also be mapped to the corresponding architecture similar to the LCG equation (1). To implement the inequality equation, a comparator is used that compares the output of two LCGs. The comparator and two linear congruential generators (LCG) are combined to form a coupled-LCG (CLCG). Two CLCGs are used in the dual-CLCG architecture. One is called controller-CLCG which generates C_i and another one is called controlled-CLCG which generates B_i.To perform the operation of Z_i B_i if C_i 0, a 1-bit tristate buffer is employed that selects the B_i (output of controlled-CLCG) only when C_i 0 (the output of controller-CLCG) and it does not select the value of B_i while C_i 1.

If the sequence C_i is considered to be known and the zero-one combinations are in a uniform pattern, then the fixed number of flip-flops can be estimated to generate random bit at every uniform clock cycle. For example:

If,

C_i = (0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1);

and

B_i = (1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1);

then,

$Z_i = B_i$ if $C_i = 0$; $Z_{buf\,i}$ = (1,x,x,1, x,0,0,x, 1,x,0,x)

In this example, it is observed that the number of 0's and 1's are equal in every 4-bit patterns. There are two 0's and two 1's in every pattern. Therefore, a pair of two flip-flops (two 2-bit registers) is sufficient to get the random bit in the uniform clock cycle. However, C_i is not known to the user and is not always in a uniform pattern. Consider an example,

If,

Ci = (**0**, 1, 1, **0**, 1, **0**, **0**, 1, **0**, 1, **0**, 1);

And

Bi = (**1**, 0, **1**, **1**, **0**, 0, 0, **1**, **0**, 0, 1, 0, **1**, 1, **1**, 1);

then,

Zbu f i = (1, **x**, 1, 1, 0, **x**, **x**, 1, 0, **x**, **x**, **x**, 1, **x**, 1, **x**)

ISSN: 2278-4632
Vol-10 Issue-9 No.03 September 2020

No such uniform pattern can be observed in the sequence Ci of the above example. Therefore, the fixed number of flip-flops cannot be estimated in this case.

Considering this problem as mentioned above, k number of bits are generated first and then stored in k-flip flops (termed as k-bit memory) when C_i = '0'. Further, these stored bits are released at every equal interval of clock cycle. If there are k number of 0's in the sequence C_i, then the dual-CLCG architecture can generate maximum of k-random bits in 2^n - clock cycles. It can utilize k- flip-flops to store k different bits of B_i while C_i = '0'. After 2^n clock cycles, it releases these stored bits serially at every two clock cycles (1-bit per two clock cycles). Here, it is assumed that the number of 0's number of 1's 2^{n-1} (for n-bit input seed) in the sequence C_i. This architecture may work even if the number of 0's in the sequence C_i is less than 2^{n-1}. However, when the number of 0's is greater than the number of 1's, then this architecture may not work correctly. The maximum length period of dual-CLCG method depends on the number of 0's in C_i and is nearly 2^{n-1} for randomly chosen n-bit input seed.

The maximum combinational path delay in the dual-CLCG architecture is the three-operand adder with multiplexer. The reason is that three-operand adder with multiplexer has highest critical path delay as compared to the comparator. Therefore,

Area, $A_{DCLCG} = 4A_{LCG} + 2A_{cmp} + A_{tri} + A_{mem} + A_{cntrl}$ Critical path, $T_{DCLCG} = T_{add} + T_{mux}$

IV. PROPOSED PRBG METHOD

To overcome the aforesaid shortcomings in the existing dual-CLCG method and its architecture as highlighted, a new PRBG method and its architecture are proposed in this project. The proposed PRBG method is the modified version of the dual-CLCG method referred as "Modified dual-CLCG", in which the equation (6) of existing dual-CLCG method is replaced with the new equation (5), This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination. Whereas the other four equations from (1) to (4) are same as in the case of dual-CLCG method from equation (1) to (4).

Proposed Modified Dual-CLCG Method and Its Algorithm

The proposed modified dual-CLCG method generates pseudorandom bits by congruential modulo-2 addition of two coupled linear congruential generator (CLCG) outputs and is mathematically defined as follows,

$$x_{i+1} \equiv a_1 \times x_i + b_1 \bmod 2^n \quad (8)$$

$$y_{i+1} \equiv a_2 \times y_i + b_2 \bmod 2^n \quad (9)$$

$$p_{i+1} \equiv a_3 \times p_i + b_3 \bmod 2^n \quad (10)$$

$$q_{i+1} \equiv a_4 \times q_i + b_4 \bmod 2^n \quad (11)$$

The pseudorandom bit sequence Zi is obtained by using the congruential modulo-2 equation (5),

$$Z_i = (B_i + C_i) \bmod 2 = B_i \text{ xor } C_i \quad (12)$$

Where,

ISSN: 2278-4632
Vol-10 Issue-9 No.03 September 2020

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

Here, a1, b1, a2, b2, a3, b3, a4 and b4 are the constant parameters; x0, y0, p0 and q0 are the initial seeds. The necessary conditions to get the maximum length period are same as the existing dual-CLCG method. The proposed modified dual-CLCG method uses the congruential modulo-2 addition of two different coupled LCG outputs as specified in equation (5). Hence, the congruential modulo-2 addition does not skip any random bits at the output stage and produces one-bit random output in each iteration. Since, the coupled LCG has the maximal period, the modulo-2 addition of two coupled-LCG outputs in the modified dual- CLCG have also the same maximum length period of 2n for n-bit modulus operand. To perform the modulo-2 addition operation, it takes only single XOR logic. Therefore, by replacing equation (6) in existing dual CLCG with equation (5), the proposed PRBG method can reduce the large memory area used in the existing dual-CLCG method and also can achieve the full-length period of 2n. The step by step procedure to evaluate the pseudorandom bit sequence in the proposed PRBG method is summarized in the algorithm form in Algorithm 1.

Algorithm 1 Modified Dual-CLCG Algorithm to Generate Pseudorandom Bit Sequence Z_i

Input: n (positive integer), m 2^n.

Initialization:

$b_1, b_2, b_3, b_4 < m$, such that these are relatively prime with m.

$a_1, a_2, a_3, a_4 < m$ s.t. (a_1-1), (a_2-1), (a_3-1) and (a_4-1) must be divisible by 4.

Initial seeds x_0, y_0, p_0 and $q_0 < m$.

Output: Z_i

1. for i 0 to k
2. Compute $x_{i\ 1}$, $y_{i\ 1}$, $p_{i\ 1}$, $q_{i\ 1}$ using equation (1), (9), (3) and (4) respectively;
3. if $x_{i+1} > y_{i+1}$, then $B_i = 1$ else $B_i = 0$;
4. if $p_{i+1} > q_{i+1}$, then $C_i = 1$ else $C_i = 0$;
5. Z_i $(B_i$ $C_i)$ mod 2;
6. Return Z_i ;

Before developing the architecture of the "Modified dual-CLCG" algorithm, the randomness properties are analyzed in the next subsequent sections.

ISSN: 2278-4632
Vol-10 Issue-9 No.03 September 2020

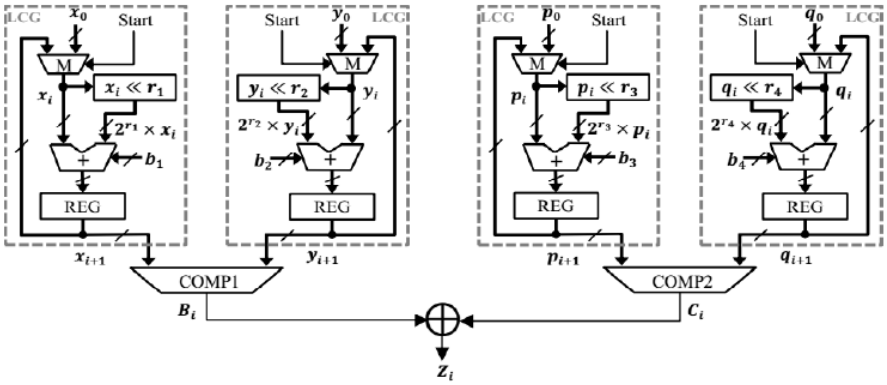


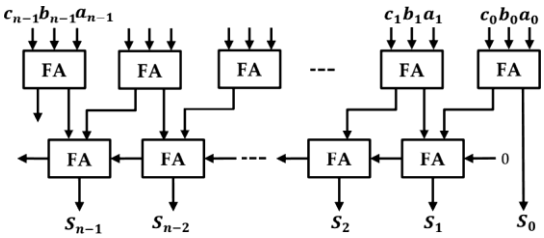
Fig. 3. Proposed architecture of the modified dual-CLCG method.

Complexity Analysis of the Proposed Architecture

The LCG block used in the proposed architecture takes the maximum combinational path delay which is contributed by the combination of one adder and multiplexer delay. The proposed architecture of the modified dual-CLCG method consumes an area of four 2x1 n-bit multiplexers, four n-bit registers, four n-bit three-operand modulo-2^n adders and one 1-bit XOR gate. Therefore, the area and the maximum combinational path delay of the proposed modified dual- CLCG architecture are evaluated as follows,

Area, $A_{prop} = 4\ A_{3oa} + A_{mux} + A_{reg} + 2A_{cmp} + A_X$ Critical path, $T_{prop} = T_{3oa} + T_{mux}$ Σ

Here, A_{3oa} and T_{3oa} represent the area and critical delay of the three-operand adder. The performance of the proposed architecture depends on the efficient implementation of the three-operand adder and the binary comparator. The carry save adder (CSA) is the most efficient and widely adopted adder technique to perform the three-operand modulo-2^n addition [25]. Therefore, the three-operand modulo-2^n adder in the proposed architecture is implemented by



using the carry-save adder which is shown in Fig. 6. In carry-save adder, the three- operand addition is performed in two stages. The first stage is the array of full adders and each of them performs bit wise addition that computes sum and carry bit signal. The second stage is ripple carry adder that computes final sum signal. The area (A_{3CSA}) and critical path delay (T_{3CSA}) of carry save adder are evaluated as follows,

Fig. 4. Three-operand modulo-2^n carry-save adder.

Area, $A_{3CSA} = (2n-1)A_{FA} = (2n-1)\ (2A_X + 3A_G)$

ISSN: 2278-4632
Vol-10 Issue-9 No.03 September 2020

Critical path, $T_{3CSA} = nT_{FA} = 3T_X + 2(n-1)T_G$

Similarly, the binary comparator in the proposed modified dual-CLCG architecture is implemented by the magnitude comparator which is the most common comparator technique [26]. The n-bit binary comparator is designed using the 2-bit magnitude comparator (see Fig. 7(a)). The logic diagram of 2-bit magnitude comparator is shown in Fig. 7(b) that computes A > B (A_{big}) and A < B (B_{big}) signals by comparing two 2-bit binary operands. The number of 2-bit comparator stages for the n-bit magnitude comparator is ($\log_2 n + 1$) and therefore, the critical path delay is in the order of O ($\log_2 n$). Hence, the overall area (A_{cmp}) and critical path delay (T_{cmp}) of the magnitude comparator are evaluated as follows,

$$\text{Area, } A_{cmp} = (n-1)[9A_G + 4A_N]$$
$$\text{Critical path, } T_{cmp} = 4(\log_2 n)T_G$$

Therefore, the overall area and critical path delay of the proposed architecture of the modified dual-CLCG method can be evaluated as follows,

Area, A_{MDCLCG}

$$= 4(A_{3oa} + A_{mux} + A_{reg}) + 2A_{cmp} + A_x)$$
$$= (16n - 7)A_X + 2(27n-15)A_G + 12A_N + 4nA_{FF}$$

Critical path, T_{MDCLCG}

$$= T_{3oa} + T_{mux} = 3T_X + 2nT_G$$

Here, A_G, A_X, A_N and A_{FF} are denoted as area of 2-input basic gate (AND/NAND/OR/NOR), XOR/XNOR gate, NOT gate and flipflop respectively. T_G and T_X denotes the delay of 2-input basic gate (AND/NAND/OR/NOR) and XOR/XNOR gate respectively. Table III summarizes and compares the area and time complexity of the proposed architecture of the modified dual-CLCG method with the architecture of other existing PRBG methods. It reports that the critical path delay in all the LCG based methods depend on the three-operand adder circuit. The area of the proposed architecture is considerably less than the dual-CLCG architecture. It generates a one-bit random output at every clock cycle with one initial clock latency. Whereas, dual-CLCG architecture takes 2^n initial clock latency (input to first output delay) to give the first output bit and further, it takes two clock cycles (output to output delay/output latency) to generate one- bit random output. On the other hand, the architecture of BBS method [12] has the large output latency of 2n 5 clock cycles due to the use of iterative Montgomery modular multiplier.

ISSN: 2278-4632
Vol-10 Issue-9 No.03 September 2020

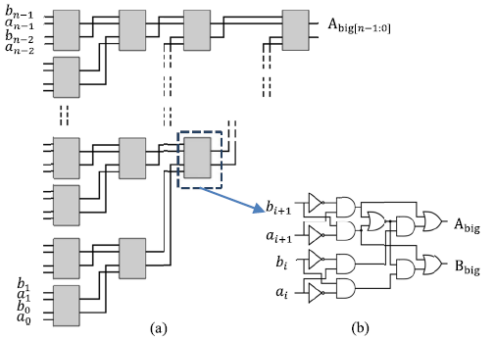


Fig. 5. Magnitude comparator (a) n-bit, (b) Logic diagram of 2-bit comparator.

Example of the Proposed Modified Dual-CLCG Method: Let

= a_1 5, b_1 5, a_2 5, b_2 3, a_3 5, b_3 1, a_4 5,

= b_4=7and m 2^3. The sequences x_i , y_i , p_i and q_i have a period of 8 and are hence full period. If the initial condition or the seed are (x_0, y_0, p_0, q_0) (2, 7, 3, 4) then the generated sequences are

$X_i = (7,0,5,6,3,4,1,2)$; $P_i = (0,1,6,7,4,5,2,3)$;

$Y_i = (6,1,0,3,2,5,4,7)$; $Q_i = (3,6,5,0,7,2,1,4)$;

Therefore, the output sequences B_i and C_i are evaluated as,

$B_i = (1,0,1,1,1,0,0,0)$; $C_i = (0,0,1,1,0,1,1,0)$

The final sequence Z_i generated from the proposed modified dual-CLCG method for n = 3-bit is,

$Z_i = (B_i + C_i) \bmod 2 = B_i \oplus C_i = (1,0,0,0,1,1,1,0)$

5.RESULTS

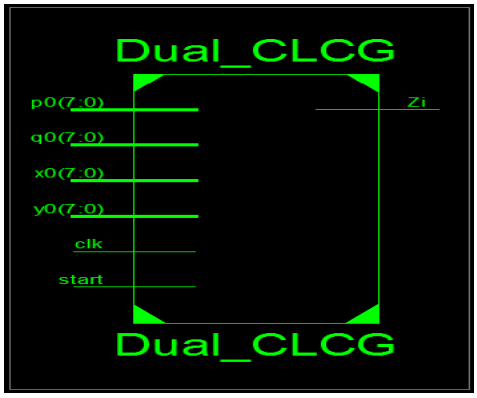


Fig6: RTL Schematic view of Dual CLCG

ISSN: 2278-4632
Vol-10 Issue-9 No.03 September 2020

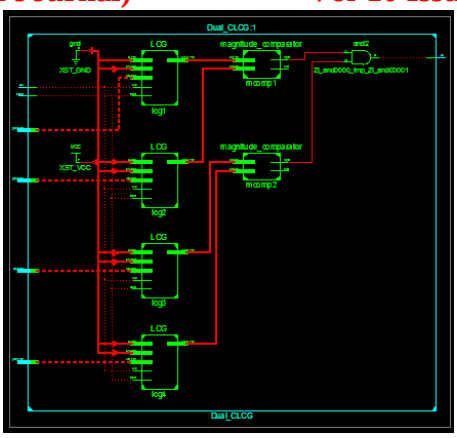


Fig 7: Internal view of RTL Schematic view of Dual CLCG

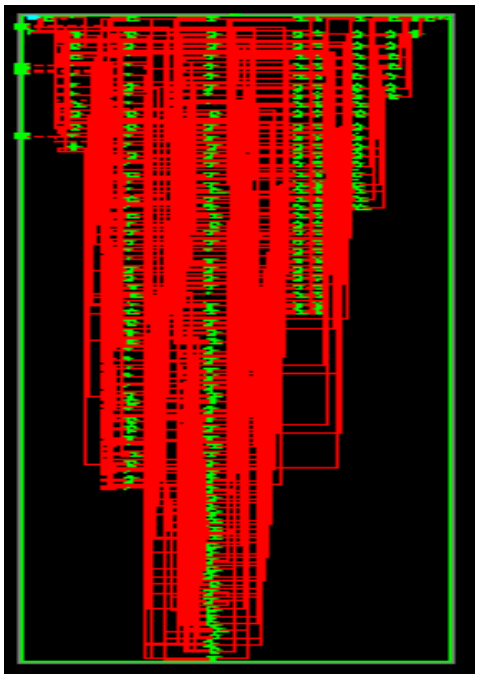


Fig 8: View technology Schematic of Dual CLCG

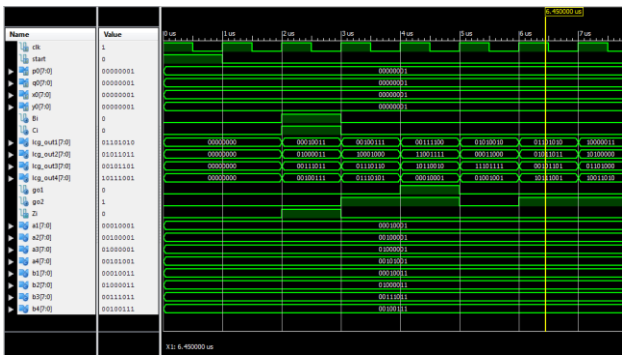


Fig 9: simulated wave form of Dual CLCG

ISSN: 2278-4632
Vol-10 Issue-9 No.03 September 2020

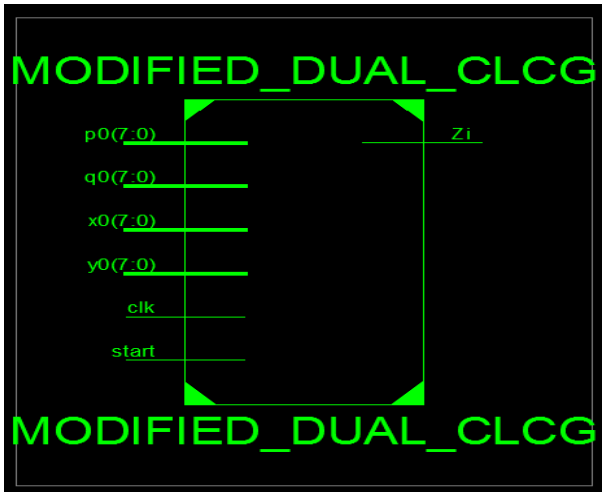


Fig 10: RTL Schematic view of Modified Dual CLCG

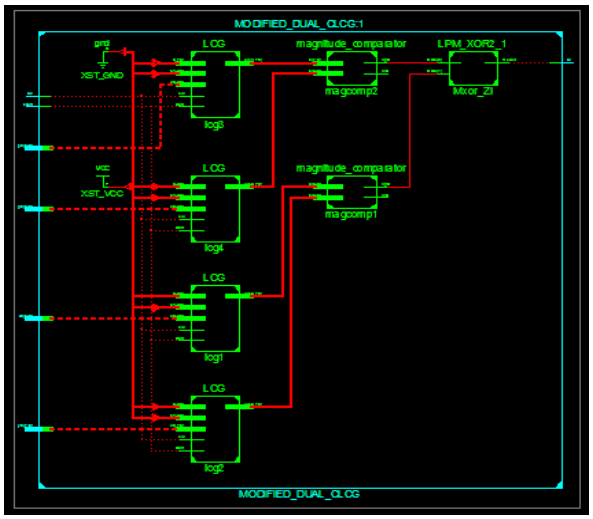


Fig 11: Internal view of RTL Schematic view of Modified Dual CLCG

ISSN: 2278-4632
Vol-10 Issue-9 No.03 September 2020

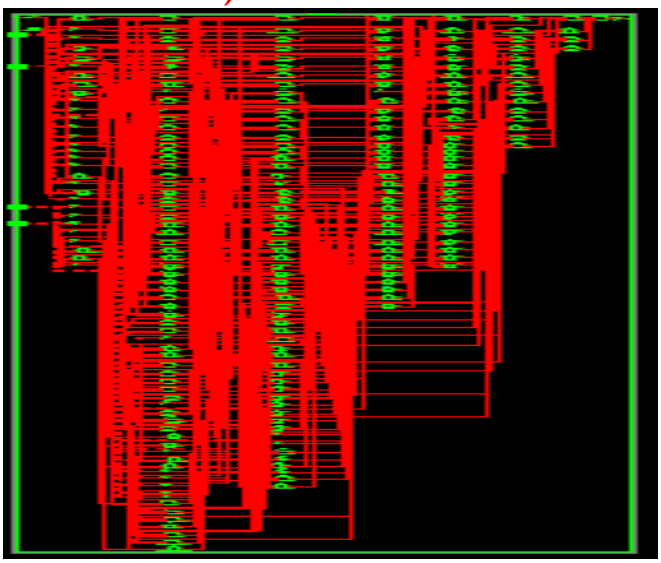


Fig 12: View technology Schematic of Modified Dual CLCG

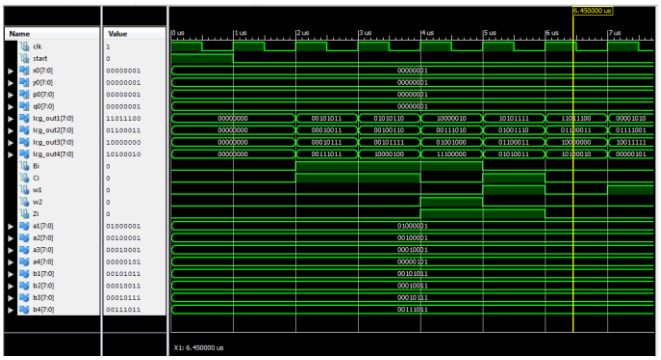
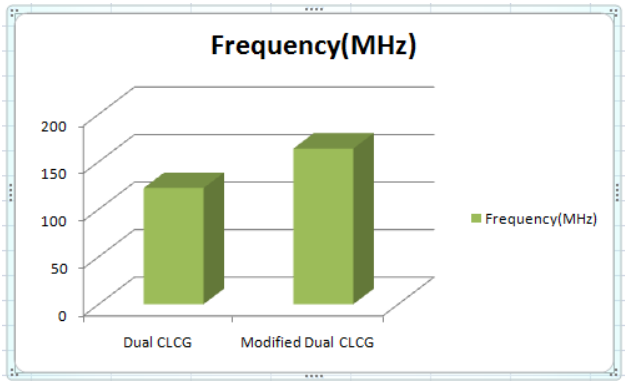


Fig 13: simulated wave form of Modified Dual CLCG

Parameter	Dual CLCG	Modified Dual CLCG
Frequency(MHz)	122.279	163.854

Table 1 : parameter comparison table



ISSN: 2278-4632
Vol-10 Issue-9 No.03 September 2020

Fig14 : Frequency comparison bar graph

CONCLUSION

Dual-CLCG 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 non-uniform time interval as it works on the few inequality cases. Hence, new hardware architecture of the existing dual-CLCG method is developed that generates the pseudorandom bit at uniform clock rate.

The proposed architecture of the modified dual- CLCG method is prototyped using the verilog code and the results are captured in using Xilinx ISE. Based on the performance analysis in terms of hardware complexity, randomness and security, the architecture can be used in the iot applications.

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. 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.