Grenze International Journal of Engineering and Technology, Jan Issue

A Low Power Dual-CLCG for Pseudorandom Bit Generation

Priyamvada B. S[1] and Dr. Kala Bharathan[2]
[1]PG Student, Dept. of ECE, Vidya Academy of Science and Technology, Kerala, India
Email: bspriyamvada@gmail.com
[2]Associate Professor, Dept. of ECE, Vidya Academy of Science and Technology, Kerala, India
Email: Kalabharathan@vidyaacademy.ac.in

***Abstract*—Pseudo Random Bit Generator (PRBG) is a need factor for getting data through transmission in different cryptographic applications. In the normal existing Pseudo random bit generation systems like Linear Feedback Shift Register (LFSR), Linear Congruential Generator (LCG), Coupled LCG (CLCG), and Dual Coupled LCG (Dual-CLCG), the last ends up being more secure. This technique relies upon correlations that causes pseudorandom bit generation at a non-uniform stretch. The drawbacks listed for this technique are excessive memory usage, high-initial clock latency, failing to grasp the foremost length sequence. In this paper we have proposed a Modified Dual-CLCG technique at a uniform clock rate to produce pseudorandom number. The tale commitment of the arranged PRBG method is to have pseudorandom bit with low power. The present proposed architecture offers significant improvement in terms of power and speed in comparison to other reported architectures. Architecture for LCG is compared with different devices so that Artix 7 shows least power having a value of 0.042W. The execution of PRBG strategy is simulated utilizing VHDL and is reconfigurable utilizing Artix 7 FPGA board.**

***Index Terms*— Pseudorandom bit generator (PRBG), VLSI architecture, Linear Congruential Generator (LCG).**

I. INTRODUCTION

By the transformative advancement of Internet-associated savvy cell phones, the data trade over the uncertain organization channel is an essential worry over a long time. Truth be told, every one of the actual things, for example, medical services framework, climate observing, progressed metering in savvy lattices, and so forth related through the Internet of things (IoT) based devices produce a significant proportion of data which prompts security and assurance issues. Pseudorandom bit generator (PRBG) is the major design block in the cryptography strategy to guarantee the security of information exchange. Most sporadic number generators require detail of a hidden number used as the early phase, which is known as a "seed." The honesty of subjective numbers made by a given estimation can be analyzed by taking a gander at its uproar circle. While creating irregular numbers over some predefined limit, it is frequently important to standardize the dissemination with the goal that every differential zone is similarly populated.

Grenze ID: 01.GIJET.7.1.5_1

A. Random Number Generation and its Types

Random Number Generator (RNG) is a strategy that utilizes various gadgets to produce an arrangement of numbers or images that can't be decently expected over by an irregular possibility. Different utilizations of irregularity that lead to the occasion of the shifted entire various courses for producing arbitrary information, of that some have existed since history

among whose positions square measure notable "exemplary" models, close to the moving of dice, coin flipping, the rearranging of getting a charge out of cards etc. Different random number generators include Blum Blum Shub Generator (BBS) [6], Linear Feedback Shift Registers (LFSR) [5], Middle-square method etc. Indeed, one will exclude that the numbers were generated prior to be the opposer and traced into a memory situated within the device.

There are two types of Random Number Generators: -

1. True Random Number Generators or Non-Deterministic Random Number Generators
2. Pseudo Random Number Generators or Deterministic Random Number Generators.

For some cryptographic applications the genuine arbitrary number generator assumes a significant part which incorporates PIN/secret word age, confirmation conventions, key age, irregular cushioning, and nonce age. The True Random Number Generator circuits that uses a nondeterministic arbitrary interaction, which is for the most part as electrical commotion, as a fundamental wellspring of irregularity. The numbers created by this kind of gadget are not unsurprising by any numerical rationale and basically rely upon the actual data like barometrical clamor. They are also called as Non-Deterministic Random Numbers. The Random Numbers produced by a gadget is unsurprising as it utilizes numerical calculations through programming. These are also called as Deterministic Random Numbers. Pseudorandom number Generator utilizes an equation for producing the grouping of numbers whose properties inexact the properties of arrangements of arbitrary numbers.

B. Need for Pseudorandom Number Generation

With the appearance of PCs, developers perceived the necessity for a method of bringing arbitrariness into a PC infection. Be that as it may, dazzling in light of the fact that it could show up, it's hard to ask a PC to attempt to one thing inadvertently as PC follows the given headings indiscriminately and is along these lines totally unsurprising. It's impossible to think of truly arbitrary numbers from settled issue like workstations hence PRNG could be a strategy created to concoct irregular numbers utilizing a PC.

C. Motivation and Objectives

In spite of its long history, irregular age actually stays an intriguing issue, with the development of the purported Random or Entropy as Service needs. Notwithstanding the regular utilization of these generators in numerous applications as depicted over, their reconciliation into System on Chip becomes profoundly attractive, especially for IoT and Smart Cards. Thusly, the common sense motivation behind flow research works in this field is to give reduced, with high throughput, secure, also, reconfigurable pseudorandom generators for equipment applications. The main objectives of this project are:

1. To develop an efficient PRBG (Pseudo Random Bit Generator) algorithm with high speed computations with low power.
2. To apply the Pseudorandom Bit Generation algorithm for IoT enabled resource constraint devices.

II. PREVIOUS WORKS

Amit Kumar Panda and Kailash Chandra Ray clarifies the Dual-CLCG strategy which includes double coupling of four LCG that makes it safer than LCG based PRBGs. Notwithstanding, it is accounted for that this technique has the disadvantage of producing pseudorandom bit at non-uniform time stretch as it deals with the couple of imbalance cases. In their design, just a solitary XOR rationale is used at the yield stage rather than complex equipment circuits (buffer, information control and memory) utilized in past double CLCG engineering [1]. Accordingly, the equipment intricacy of the proposed engineering of the new changed double CLCG technique is essentially decreased.

Sayed Ahmad Salehi in his work he verified the plan of minimal effort and low-associated SNG circuits utilizing LFSR sharing. To lessen the connection among the produced bit transfers, they permuted the yield of a common LFSR prior to utilizing it as contribution for distinctive SNGs [2]. He additionally proposed a calculation for finding a bunch of m changes that can be shared among m SNGs with least cross connection. They utilized the proposed SNGs in the SC-based usage of a few applications and the results show that, with low equipment intricacy, they acquire better computational exactness contrasted with earlier techniques.

Amit Kumar Panda and Kailash Chandra Ray clarifies the advancement of a proficient equipment engineering for producing pseudorandom bit arrangement by focusing on the IoT empowered applications. At first, the

current double CLCG calculation is concentrated to play out its equipment design [3]. In any case, it encounters the disadvantages of producing pseudorandom bit at an inconsistent time frame that lead to fluctuating in the yield inertness and furthermore neglects to accomplish the most extreme length time frame. The proposed CVLCG technique is created by coupling of two variable-input LCGs, which produces pseudorandom bit at each cycle and furthermore achieves the most extreme length grouping. Additionally, it saves one comparator region in contrast with the current double CLCG design.

Jun Zhou and Zhenfu Cao portrays about the Internet of Things which is increasingly changing into a current figuring administration, requiring Brobdingnagian volumes of data stockpiling and cycle tragically, in light of the unmistakable qualities of asset imperatives, self-association, and short differ correspondence in IoT, it never-endingly falls back on the cloud for re-evaluated capacity and calculation, that has prompted a progression of ongoing troublesome security and protection dangers [4]. During this article, they tend to present the plan and unmistakable security and protection needs for succeeding age versatile innovations on cloud based IoT.

III. PROPOSED METHODOLOGY

The design of the entire system which includes Architecture, Block Diagram and the general circuits for the pseudorandom number generation is given below.

A. Linear Congruential Generator (LCG)

Linear Congruential Generator, the most notable and most prepared computation for making pseudo-randomized numbers. LCG's can be widely used for a variey of applications. LCG mainly related to a rule that yields a grouping of numbers which are pseudo-randomized and are determined with a spasmodic piece wise condition. Pseudorandom random generator algorithms are the oldest and well-known algorithm. The theory behind them is comparatively simple to grasp, and that they are simply enforced and quick, particularly on constituent which may offer standard arithmetic by storage-bit truncation. The architecture of LCG is shown in the Fig 1a.

The Linear Congruential Generator is defined by the recurrence relation:

$$X_{n+1} = (aX_n + c) mod m \qquad (1)$$

where X determines the sequence of pseudorandom values, m is the modulus value a and c are increments and X_0 is the seed.

To begin, the calculation requires an underlying Seed, which must be given by certain methods. The presence of haphazardness is given by performing modulo math.

Theorem (Period): The main three conditions for an LCG to have full period if and only if:

- The only positive integer that (exactly) divides both m and c is 1.
- Let q be a prime number that divides m, then q divides a-1.
- If 4 divide m, then 4 divide a-1.

B. Proposed Method

By the comparison of different random number generators, it can be concluded that Linear Congruential Generator is the best pseudorandom number generator for generating random numbers. The comparison of pseudorandom number generation was done with respect to the best performing random number generators like Blum Blum Shub generator, Linear Feedback Shift Registers, SHA-LFSR etc. The linear random number generators are LCG and LFSR. Non linear random number generator is the Blum Blum Shub generator.

Even though the LCG has its own advantages and disadvantages its most usable and least complicated structure. The PRBGs are insecure due to its linearity structure. The coupling of LCG becomes more easier to generate random number and thus avoids the drawbacks of single LCG. The linearity property can be regained using the dual LCG method which is the coupling of LCG's. The coupling of two LCGs in the CLCG method makes it more secure than a single LCG. Dual LCG's can be used for high security usages. The comparison table for different generators were shown in Table I. The most important criteria's like initialization time, memory usage, CPU usage, security, randomness was compared in order to identify the best random number generator.

The advantages of LCG over other random number generators are shown below.

- Mostly used random number generator.
- LCG is quick and requires negligible memory.
- LCG is easy to implement on both the hardware and software.
- The coupling of LCG makes the random number generation easier.
- LCG is having less complex structure.
- It provides more security when coupled.

- Eventhough they are not having linearity structure they can be modified for better random number generation purposes.

TABLE I. COMPARISON STUDY ON DIFFERENT GENERATORS

Features	BBS	LCG	LFSR	SHA-LFSR
Initialization time	Too Large	Large	Less	Less
Memory Usage	18.1 MB	18.1 MB	24.2 MB	32 MB
CPU Usage	27%	25%	25%	75%
Security	Normal	Low	Low	High
Randomness	Pass the statistical test	Pass the statistical test	Pass the statistical test	Pass the statistical test

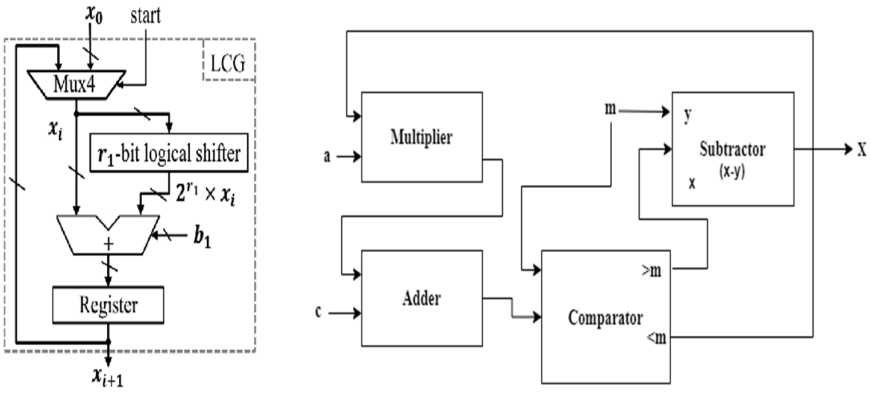


Figure 1. (a) Architecture of Linear Congruential Generator and (b) Block diagram of Linear Congruential Generator

The architecture of LCG which includes Adder, Multiplier, Subtractor, Comparator, Shift Registers and Multiplexers shown in Fig. 1a. Each block was analysed using Xilinx software. Among the different internal block's multiplexer were avoided in order to reduce the hardware complexity as well for power consumption. Fig. 1 b shows block diagram LCG operation in general ignoring the seed. Multiplier, Adder, Comparator and Subtractor are the main blocks for the working of LCG. Multiplier is used to multiple previous random value X (2-bit) with a, then add with increment (4-bit) c. Finally, that will be compared with modulus m (5-bit). If the number is smaller or equal to m then it is considered as a random number. Otherwise, the number then subtracted with m. The result of this is then consider as random number.

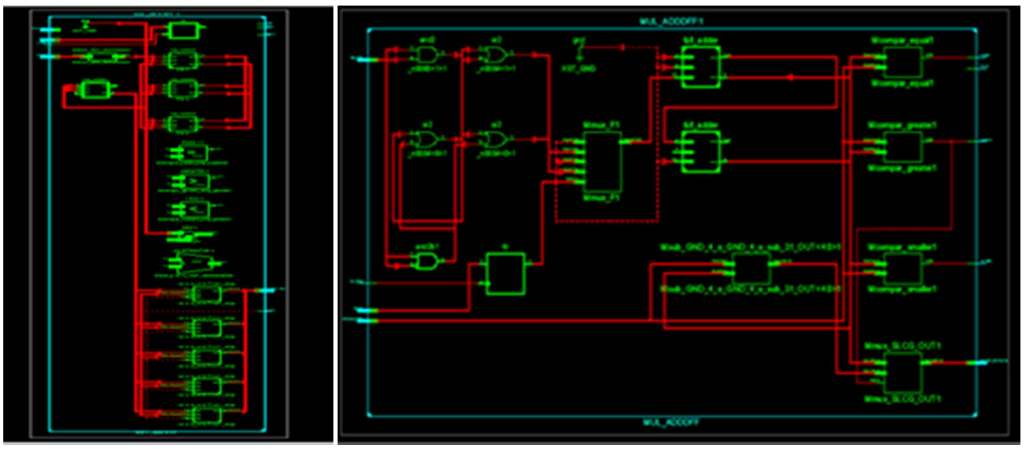
IV. RESULTS AND DISCUSSIONS

A. Analysis of Single Linear Congruential Generator

The analysis of different blocks was done using Xilinx software. The proposed architecture is implemented using VHDL. The code is tested for 5-bit binary numbers. The sub-blocks of Linear Congruential Generator include the Adder (Carry Save Adder), Subtractor (Full Subtractor), Comparator, Multiplier and D-Flip Flop. The RTL (Register Transistor Level) Schematic of single LCG is shown in Fig 2 (a) and (b). Simulated output for single LCG is shown in Fig 3.

B. Power Analysis of Single LCG in different devices like Spartan 3E, Artix 7, Virtex 7

Based on the Simulations of Single Linear Congruential Generator circuit, power a major factor was also analysed in different devices in Xilinx software like Spartan 3E, Artix 7 and Virtex 7. The average power of the basic circuit according to our design were obtained. Different devices have different utilizations in terms of area and power which help us to identify which device will better for the designing session as well for the random number generation. The power obtained for different devices like Artix 7, Spartan 3E, Virtex 7 are 0.042W, 0.082W, 0.143W respectively. The delay for Artix 7, Spartan 3E, Virtex 7 includes 2.785ns, 10.372ns, 2.236ns. From the comparisons on different devices Artix 7 have less power i.e. 0.042W compared with Spartan 3E and Virtex 7. The comparison of power and delay with respect to different devices are shown in Fig 4 (a) and (b).



(a) (b)

Figure 2. RTL Schematic of Single LCG (a) Schematic diagram on Spartan 3E device (b) Schematic diagram on Artix 7 and Virtex 7

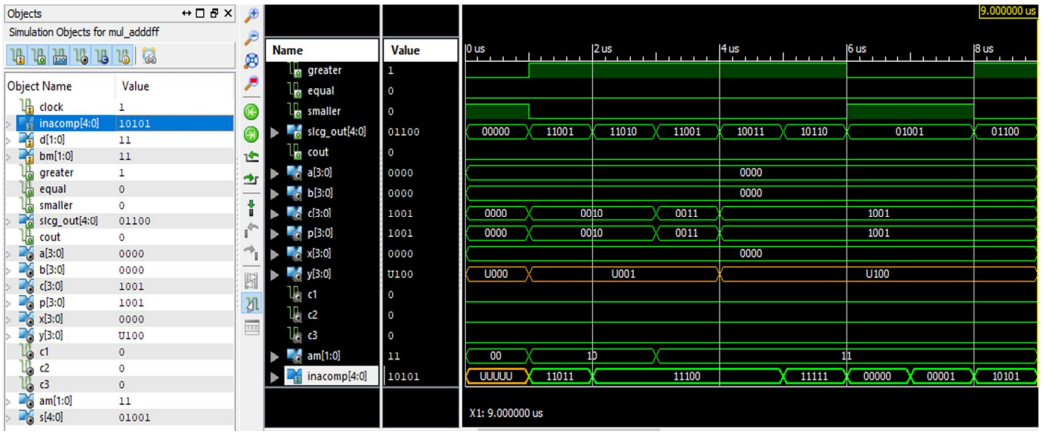
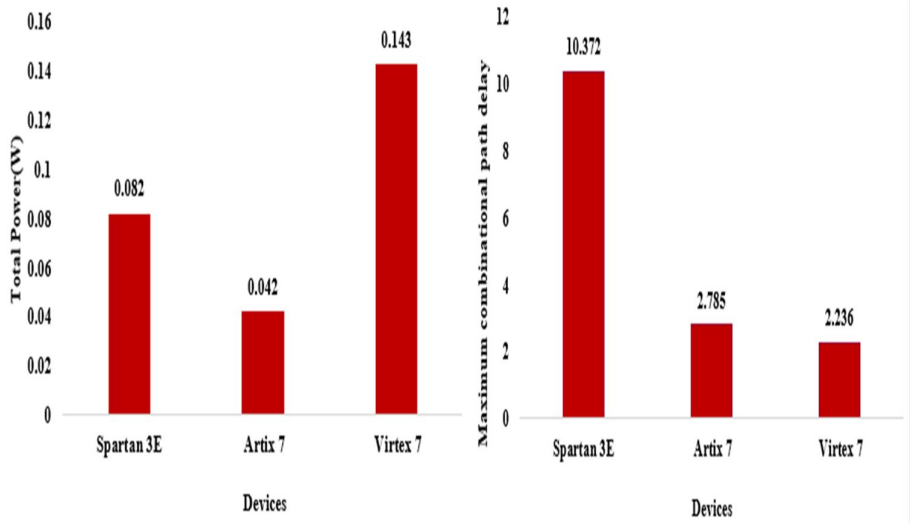


Figure 3. Simulated Output of Single LCG **(d=10,bm=01; d=10,bm=01; d=11,bm=01; d=11,bm=11)**



(a) (b)

Figure 4. Analysis in terms of (a) Total power and (b) Delay

V. Conclusion

In this paper architectural studies relating to the pseudorandom number generation was analysed. The LCG is the least difficult and most productive pseudo random number generator compared to all the other structures for security purposes. The performance in terms of power was obtained as 0.042W in Artix 7 while it shows the maximum speed. By the analysis of path delay the least delayed device will undertake most speed computations. Here Virtex 7 shows the least delay of 2.236ns. From the result analysis it is proven that these architectures can be used in various devices which shows the reconfigurability with respect to FPGA devices. Future applications include Digital Signal Processing, Security purposes, Encryption.

Acknowledgment

I would like to earnestly acknowledge the sincere efforts and valuable time given by my project guide Dr. Kala Bharathan, Associate Professor, Dept., of ECE. Her valuable guidance and feedback have helped me in completing this project.

References

[1] Amit Kumar Panda, Kailash Chandra Ray "A Coupled Variable Input LCG Method and its VLSI Architecture for Pseudorandom Bit Generation" IEEE ,0018-9456,2020.
[2] Sayed Ahmad Salehi, "Low-Cost Stochastic Number Generators for Stochastic Computing", IEEE Transactions on VLSI Systems, 1063-8210,2020.
[3] Amit Kumar Panda, Kailash Chandra Ray, "Modified Dual-CLCG Method and Its VLSI Architecture for Pseudorandom Bit Generation", IEEE Transactions on Circuits and Systems–I: Regular Papers ,1549-8328,2019.
[4] Jun Zhou, Zhenfu Cao, "Security and Privacy for Cloud Based IoT: Challenges, measures and Future Direction", IEEE Communications magazine,2017.
[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-lfsrbasedpseudorandom generators–a-survey(59f7106b-1800-49df-8037- fbe9e0e98ced).html.
[6] C. Ding, “Blum-Blum-Shub generator,” IEEE Electron. Lett., vol. 33, no. 8, p. 667, Apr. 1997.
[7] Zulfikar and Hubbul Walidainy, "Design and Implementations of Linear Congruential Generator into FPGA "International Journal of Electronics Communication and Computer Engineering Volume 5, Issue 4,2278.