Received: May 2, 2018

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Low Area FPGA Implementation of PRNG-LCC-CSLA Architecture Based on Chaotic Circuit

Baby Honnenahalli Thammannagowda[1*] **Sujatha Bangalore Ramchandra[2]**

[1]*Government Engineering College, Hassan, India*
[2]*Malnad College of Engineering, Hassan, India*
* Corresponding author's Email: babygowda@gmail.com

Abstract: In the present days, internet communication is adapting data hiding and cryptography that often needs billions of pseudo random numbers. The Pseudo Random Number Generator (PRNG) is used to fulfill privacy and security needs in most of the applications. In this paper, the PRNG using Lorenz Chaotic Circuit with Carry Select Adder (PRNG-LCC-CSLA) method used to generate the pseudo random numbers. The LCC implemented by using Verilog to generate the pseudo random numbers, and randomness of the LCC results also checked in National Institute of Standards Technology (NIST) testing scheme. Encryption process is implemented by employing binary value of both text and image. Decryption process is implemented by employing same LCC binary value of encrypted data. At last, the Matlab is used to observe the text and image data. The Field Programmable Gate Array (FPGA) results showed how Look Up Table (LUT), slice, flip-flop and frequency improved and Application Specific Integrated Circuit (ASIC) results showed area, power, delay, Area Power Product (APP) and Area Delay Product (ADP) improvements in the PRNG-LCC-CSLA method compared to the existing methods.

Keywords: Pseudo-random number generator, Cryptography security, Carry select adder, Lorenz's chaotic circuit and NIST- statistical tests.

1. Introduction

Presently reliable communication channel is required to carry digital data (information) across the transmission channel. The security of the digital data becomes more important in the communication field. The demand for faster and secure encryption techniques becomes crucial to avoid the loss of data. To understand the similarities between chaos and cryptography, a number of chaotic encryption schemes have been developed in the previous years. The media encryption technique is a challenging task because of the significant level of sophistication developed by forgers and other cyber criminals. The advanced encryption techniques are used for data storage, secure transmission and retrieval of digital images required for different applications like medical, military, homeland security and so on [1]. Generally, the information security depends upon the quality of the Pseudo Random Number (PRN) employed for the protocols. The PRN Generator (PRNG) is used for secured internet communication like the generation of any key stream, keys generation of an asymmetric cryptosystem, keys used for keyed hash functions and so on. The numerous PRNGs technologies have been developed, which are more secure to store the data. But, this system is very slow and insecure in the communication field [2]. The Internet security field is concerned with the use of watermarking methods for data hiding. Nowadays, the watermarking method is often cited as a possible solution for digital rights management problems and to counteract piracy of digital work in an internet based entertainment world [3].

The chaos theory describes the behavior of the certain nonlinear dynamic systems with specific conditions that are sensitive to the initial values. The benefit of the chaotic encryption methods is robust, easy implementation, high secure and fast encryption process [4, 5]. The symmetric cipher

method based on the substitution-diffusion architecture employs 1-dimension (1D) logistic map and 2-dimension (2D) standard map. The private key algorithm contains the original conditions and the number of iterations of the chaotic maps and system parameters. But the encrypted cipher text is not fully secure [6, 7]. The PRNG based chaotic circuit has been designed in FPGA hardware and used a different cryptography technique. But this method is not matched for high-speed applications and also does not secure the information [8]. To overcome this problem, the PRNG-LCC-CSLA method is introduced to reduce the hardware utilization of the entire system. In this work, LCC is implemented using Verilog to generate the pseudo random numbers and randomness of the LCC output is checked with NIST suite. Encryption is implemented using binary value corresponding to both Text and Image, and pseudo random number output of LCC. Decryption is done for the encrypted data using same LCC binary value. MATLAB is used to observe the TEXT data and Image files. ASIC synthesis done in Cadence tool for different technology such as 180nm and 45nm. In the ASIC implementation, the area, power, and delay is reduced in PRNG-LCC-CSLA method. As well as the FPGA synthesis is done in Xilinx tool for various devices like virtex-4, virtex-5, and virtex-6. In the FPGA implementation, reduced the number of LUT, flip flop and slice of PRNG-LCC-CSLA.

The rest of this paper is organized as follows. In section-2 related works are discussed, section-3 describes the implementation of PRNG-LCC-CLSA methodology. In the Section-4 results are discussed and section-5 concludes the overall work.

2. Related work

H.I. Hsiao and J. Lee [9] proposed the chaos based encryption algorithms that significantly composed of two kinds of aspects: diffusion and confusion. In this paper, amplitude phase frequency model has been implemented to secure the color images. This method possessed the key sensitivity that reached the order of 10^{-10} to get a sufficient security strength needed to protect the color image. Amplitude Phase Frequency Model (APFM) requires more memory to store the information of decrypted data.

H. Shimakage and Y. Tamura [10] presented chaotic oscillation circuits for RNG. In the simulation, an RCSJ model and the current bias have been used. Current biases were applied to the Josephson junctions and computed the time evaluation signal of the voltage when that was irradiated with an external microwave. This work was very complicated to evaluate the random number.

X. Fang, Q. Wang, C. Guyeux, and J.M. Bahi [11] proposed FPGA acceleration of a PRNG based chaotic iteration. In this paper, chaotic iteration was redesigned in FPGA to improve the generation rate. From the chaotic pseudo random sequence, different kind of performance was analyzed. To perform the cryptography, XOR shift was implemented that had a number of the multiplier, which occupied more area. A number of digital adders were required to design the multiplier. This paper does not identify and mention hardware requirements like LUT, slices, Flip-flops, and Frequency. So, it's very critical to find the hardware usage of this work.

İ. Koyuncu, and A.T. Özcerit, [12] presented the new high-speed FPGA based chaotic true RNG. In this paper, chaotic system was modeled numerically with the help of fourth order of the Runge- Kutta method. This work has been implemented using the Pspice software. The main drawback of this proposed method was difficult to generate the random number in analog based chaotic circuit compared to digital based chaotic circuit. This method failed to analysis the FPGA performance for different Virtex devices such as Virtex4, and Virtex5.

J.M. Bahi, X. Fang, C. Guyeux, and Q. Wang [13] presented True Random Number generators (TRNGs) based on Beat Frequency Detection (BFD) method was implemented for FPGA applications. In this paper, ring oscillator based BFD-TRNG was implemented based on different FPGA device. The main advantage of the proposed method was to improve randomness through dynamic partial reconfiguration. Main problem of this paper was that when the ring oscillators were free running, it's difficult to control the design bias.

The main problem in the VLSI implementation is minimizing the area, power and delay parameters. The principle objective of this work is to reduce power, area and delay of the overall PRNG-LCC-CSLA structure, reduce the hardware cost and maximize the speed of the architecture by using CSLA adder and also to increase the randomness of the pseudo random numbers generated.

3. PRNG-LCC-CSLA methodology

In this work, PRNG-LCC-CSLA methodology has been implemented to develop the security of the input data with minimum number of hardware utilization. CSLA adder is introduced instead of

using digital adder in XOR shift circuits to reduce the hardware utilization.

The block diagram of PRNG-LCC-CSLA is shown in Fig. 1. MATLAB tool is used to read input image, encrypted image and the decrypted image. The rest of the coding section like chaotic circuits, encryption and decryption process are performed by Xilinx tool. By using Lorenz's chaotic circuit, the generated pseudo random number is used as the KEY value. The key is used to encrypt the input and convert into cipher text. The same key is required to decrypt cipher text on the receiver side. Fig. 2 shows the architecture required for generating the PRNG by using the Lorenz chaotic system and XOR shift structure to perform encryption. This work is implemented in the FPGA.

3.1 Pseudo random number generator

The PRNG employs a non-deterministic source (entropy source), processing function (entropy distillation process) to provide "Randomness" [14]. The use of a distillation method requires conquering any weakness in entropy source, which results in the production of non-random numbers. The entropy sources contain some physical quantity such as the timing of user processes, noise in the electrical circuit, quantum effect in the semiconductor. Different combinations of this input may be employed to achieve randomness.

The Fig. 2 shows block diagram of Lorenz chaotic system. An FPGA is used for efficient and secure implementation of the symmetric cryptography protocol and procedures. These reconfigurable devices intend to fill the gap between the software and hardware.

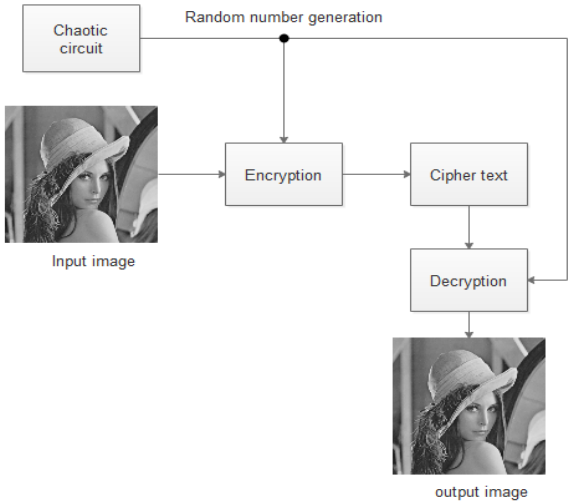


Figure.1 Block diagram of PRNG-LCS-CSLA

The digital programmable software implementation by using FPGA provides high performance and more flexibility compared to conventional hardware performance. The traditional chaotic circuits such as autonomous (chua's circuit) and non-autonomous (Resistor Inductor -diode circuit) based chaotic circuits are not suitable for pseudo random number generator due to that circuit designs are simple and basic chaotic circuits. So, the Lorenz chaotic system has been employed for designing chaotic software key, which output given to the input of the encryption and decryption operation in the experimental research. The result of any PRNG requires strict randomness criteria which are computed by the statistical tests. The advantage of the cryptographic method is the output of the PRNGs is unpredictable. However, some physical sources (data and time vectors) are quite predictable. These issues may be mitigated by combining results from various types of sources to employ input for an PRNG. However, the output from the PRNG is still deficient when computed by statistical tests. The PRNG is capable of generating sequences of numbers that are used for cryptographic applications and Randomly Generated Key (RGK) leads to more system security.

The chaos synchronization process employs chaotic signals to develop a level of security for communication systems. Chaos is not a complete disorder- its disorder in a deterministic dynamic system that is always predictable in a short-time. The Chaotic signal is derived from non-linear dynamic methods such as aperiodic, broadband and deterministic, uncorrelated and it appears randomly in the time-domain.

3.2 The PRNG by employing Lorenz chaotic circuit

The Lorenz system is a typical three-dimensional (3-D) chaotic model. Its dynamical Eq. (1) follows the first order non-linear differential formulations system. The Fig. 2 consists of three different stage process that stages depend on one stage to another stage. Each stage has two inputs such as a and b, which are stored in the different Registers like Register 1, Register2 and Register 3. and X[n], Y[n], Z[n] are the LCC output.

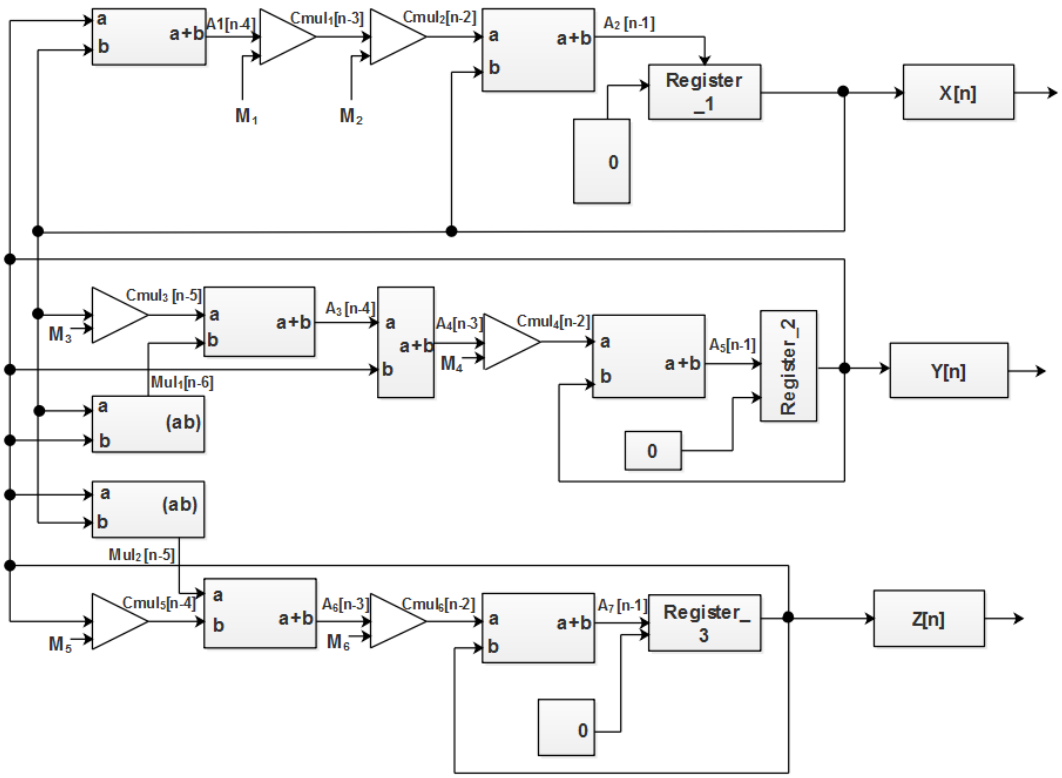


Figure.2 Block diagram of Lorenz chaotic system

Stage 1:

$A_1[n-4] = Y[n] + X[n]$

$Cmul_1[n-3] = A_1[n-4] \times M_1$

$Cmul_2[n-2] = Cmul_1[n-3] \times M_2$

$A_2[n-1] = Cmul_2[n-2] + X[n]$

$Register_1 = A_2[n-1]$

$X[n] = Register_1$

Stage 2:

$Mul_1[n-6] = X[n] \times Y[n]$

$Cmul_3[n-5] = X[n] \times M_3$

$A_3[n-4] = Mul_1[n-6] + Cmul_3[n-5]$

$A_4[n-3] = A_3[n-4] + \mathrm{Y[n]}$

$Cmul_4[n-2] = A_4[n-3] + M_4$

$A_5[n-1] = Cmul_4[n-2] + Y[n]$

$Register_2 = A_5[n-1]$

$Y[n] = Register_2$

Stage 3:

$Mul_2[n-5] = X[n] \times Y[n]$

$Cmul_5[n-4] = Z[n] \times M_5$

$A_6[n-3] = Mul_2[n-5] + Cmul_5[n-4]$

$Cmul_6[n-2] = A_6[n-3] \times M_6$

$A_7[n-1] = Cmul_6[n-2] + z[n]$

$Register_3 = A_7[n-1]$

$Z[n] = Register_3$

From this three stages, LCC has been successfully generated the PRNG which possible from digital designs [11].

3.3 Pseudo random number generation test

NIST test is a statistical package containing fifteen tests. These tests implemented to test the

randomness of the binary sequences offered by their software or hardware based on PRNG and cryptographic Random. These NIST tests concentrate on different types of the non-randomness. The fifteen tests are Frequency, Block frequency, Runs, Longest-run, Rank, DFT, Non-overlapping, Overlapping, Universal statistical, Linear- complexity, Serial, Approximate Entropy, Cumulative sums and Random excursions.

3.3.1. Frequency test

The frequency test is the propagation of 'ones' and 'zeros' for the overall sequence. The benefit of this test is to define whether the number of 0s and 1s in a Pseudo Random Sequence (PRS) are approximately the same. The frequency test measures the closeness of occurrence of zeros and ones is ½. Every subsequent test is based on the passing of frequency test.

3.3.2. Frequency block test

The frequency block test is the propagation of 1s within M-bit blocks. The advantage of frequency block test is to define if the frequency of 1s in M-block is accurately$M/2$, as would be expected under an assumption of randomness.

3.3.3. Runs test

In runs test, a number of runs in RS are un-interrupted sequence of similar-bits. Run of length k contains k identical bits. It is constrained before and after with a bit of opposite value. Advantage of the runs test is to consider if the number of the runs of 1s and 0s of numerous lengths is approximately equal.

3.3.4. Longest run test

This test is the longest run of ones with M-bit blocks. The advantage of this test to determine if the length of the longest run of 1s with the tested sequence is consistent with the length of the longest run of 1s that would be expected in RS.

3.3.5. Binary matrix rank test

The binary matrix rank test is the rank of dis-joint sub-matrices of the entire pseudo random sequence. The advantage of this test is to check the linear dependence among fixed length sub-strings of the real sequence.

3.3.6. Discrete Fourier transform (DFT) test

This test is mainly concentrated on DFT of the RS. The benefit of DFT test is to direct the periodic features in tested sequence that would indicate a deviation from the assumption of randomness. The expectation is to identify whether the quantity of peaks exceeding the 95 % of the threshold is essentially not the same as 5 %.

3.3.7. Non-overlapping test

The non-overlapping test is the number of occurrences of pre-determined target strings. The advantage of the non-overlapping test is to detect generators, which produce several occurrences of a given non-periodic pattern.

3.3.8. Overlapping test

The overlapping test is the number of rates of pre-specified target strings. The non-overlapping and the overlapping test utilize m-bit window for searching a particular m-bit design.

3.3.9. Universal statistical test

This test concentrated on the number of bits in the matching patterns. The benefit of this test is to detect whether or not the sequence can be compressed without data loss. In most of the cases, compressible sequence is measured to be non-random.

3.3.10. Linear- complexity test

The linear complexity test is the Length Feedback Shift Register (LFSR). The benefit of this test is to define if the sequence is complex enough to be considered RS. Furthermore, the RSs are characterized by longer LFSRs.

3.3.10. Serial test

The serial test is the concentration of the frequency of all possible overlapping m-bit pattern across the entire pseudo random sequence. The benefit of this test is to determine if the number of rates of the 2^m M - bit overlapping patterns is approximately used for RS.

3.3.11. Approximate-entropy test

The approximate-entropy test is the frequency of all possible overlapping m-bit patterns across the entire RS. The main advantage of this test is to compare the frequency of overlapping blocks of 2-

consecutive or adjacent lengths $(m, m+1)$ against the expected outcome of a pseudo random sequence.

3.3.12. Cumulative sums Test (CST)

CST is the concentration of the maximum excursion of the RS considered by the cumulative sum of adjusted (-1, +1) digits in the sequence. The purpose of CST is to consider whether the cumulative sum of "Partial Sequence" occurring in the test sequence is much large or small compared to the expected behavior of cumulative sum used for the pseudo random sequences.

3.3.13. Random execution test

The random execution test is concentration of the number of cycles having exactly *K* visits in a CS pseudo random sequence. The CS pseudo random sequence is derived from partial sums after (0, 1) sequence is transferred to the appropriate (-1, +1) sequence.

3.4 Encryption using chaotic PRNG for image and text

In this section, image and text files employed to perform the encryption operation. Both files converted into the binary value that is given as the input along with LCC output to the encryption block. The resultant output will be in the form of binary data and will be represented as encrypted image using MATLAB tool.

3.5 Decryption

The decryption method is same as the encryption method in terms of the modular element by the element multiplication. The encrypted value and LCC output perform XOR operation to get decrypted value. Finally, the decrypted value is similar to input of encryption value. After performing encryption, the noise if added in an encrypted file still provides exact decrypted file. The optimized CSLA used in LCC to reduce the hardware utilization which is explained in section 3.6.

3.6 CSLA design

In this method, a new area efficient CSLA is employed as an alternative of the normal adder, which is presented in Fig 3. This CSLA achieves fast mathematic (addition) operation in various data processing technique. To minimize the area, power dissipation, and delay in the PRNG-LCC designed by using CSLA. The CSLA is manipulating in many computational structures to cut the carry propagation delay. The elementary knowledge of this work includes Binary- to- Excess Converter (BEC) instead of the Ripple Carry Adder (RCA) with Cin=1.

The essential idea of this work is to employ BEC an alternative of RCA with Cin = 1 in the normal CSLA to improve low- area and power consumption. The benefit of this BEC logic comes from the fewer number of logic-gates compared to the n-bit design. The group-2 has one 2-b RCA, which has one- FA and one- Half Adder (HA) for Cin =0. Instead of another 2-b RCA with Cin = 1, 3-b BEC is employed, it adds 'one' to the output from 2-b RCA.

An input arrival time is lesser than the multiplexer selection input arrival time. Based on the selection line input Cin, CSLA adder gives either BEC output or multiplexer output. The multiplexer delay and mux-selection arrival time are derived from the different kind of groups. Finally, ASIC performance parameters like area, power, and delay as well as the FPGA performance parameters such as LUT, slices, and flip-flop are reduced in CSLA technique compared to the existing methods.

4. Result and discussion

The PRNG-LCC-CSLA design timing diagram is verified in Modelsim 10.1c using Verilog code. RTL schematic was taken from Synplify pro tool.

FPGA performance was analyzed for different devices of Virtex-4, Virtex-5 and Virtex-6 by using Xilinx ISE tool. In PRNG-LCC-CSLA work, ASIC implementation results are taken from Cadence tool of different libraries such as 180nm as well as 45nm. The NIST tests are used for checking Randomness of LCC circuit. In this paper, the LCC architecture design has implemented by employing CSLA instead of normal adder, which takes less area, power and delay.

4.1 Image encryption and decryption

Initially, the Lena image is read in MATLAB as shown in Fig. 4 and stored in a text file in the structure of "image_data.txt".

Then, Lorenz chaotic circuit was designed in Verilog to generate the pseudo random numbers. To perform the encryption, simple XOR operation was employed using 2 inputs i.e. output from LCC as a KEY value and the binary values generated from the image (image_data.txt). After performing the Encryption, the encrypted values were stored in a text file in the structure of "Encryption.txt".

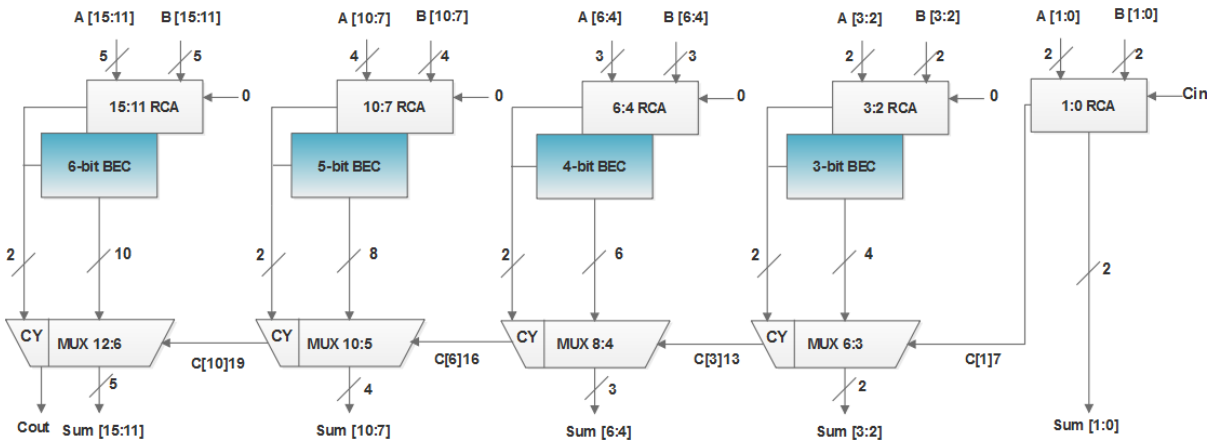


Figure.3 Block diagram of low-CSLA

The output of the LCC and Encrypted text file are required to perform a Decryption operation. The decrypted output was stored in the structure of "Decryption.txt". The text value of Image_data.txt, Encryption.txt and Decryption.txt is shown in Fig. 5 (a), (b) and (c). Image size represented as 128*128 which is equal to 16384.

Figure.4 input Lena image

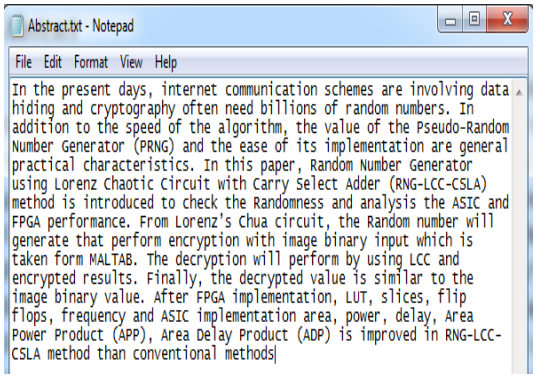
1	11000101	1	xxxxxxxx	1	xxxxxxxx
2	11000101	2	11000101	2	11000101
3	11000100	3	11001111	3	11000101
4	11000100	4	11110110	4	11000100
5	11000011	5	10001100	5	11000100
6	11000111	6	10010100	6	11000011
7	11000001	7	10101000	7	11000111
8	11000100	8	11010010	8	11000001
9	11001111	9	00010111	9	11000100
10	11010111	10	10010100	10	11001111
11	11010010	11	10100011	11	11010111
12	11001011	12	10001010	12	11010010
13	11000100	13	01010100	13	11001011
14	11001010	14	00010011	14	11000100
15	11000011	15	01111010	15	11001010
16	11000101	16	11110000	16	11000011
17	11001011	17	11111100	17	11000101
18	11001011	18	10100000	18	11001011
19	11001100	19	01001010	19	11001011
20	11001100	20	10111111	20	11001100

(a) (b) (c)

Figure.5 (a) Image_data.txt, (b) Encryption.txt, and (c) Decryption.txt

4.2 Text encryption and decryption

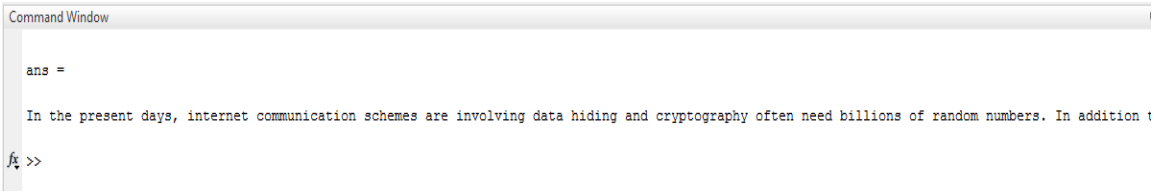
In this section, "Abstract" text was employed to perform encryption and decryption operation, which is as shown in Fig. 6 (a). The text was read from MATLAB that converted into binary value, which is as shown in Fig. 6 (b). These binary values are used to perform encryption and decryption process similar to the image data. The decrypted value was read in MATLAB to retrieve the "Abstract" text value which is as shown in Fig. 6 (c).



(a)

1	01001001
2	01101110
3	00100000
4	01110100
5	01101000
6	01100101
7	00100000
8	01110000
9	01110010
10	01100101
11	01110011
12	01100101
13	01101110
14	01110100
15	00100000
16	01100100
17	01100001
18	01111001
19	01110011
20	00101100

(b)



(c)

Figure.6 (a) Abstract text, (b) Abstract binary value, and (c) Decrypted Text

4.3 Randomness test

The LCC output is employed to check the randomness test. The randomness results are as given in Table 1.

4.4 ASIC synthesis

ASIC synthesis was implemented in Cadence tool for different technology like 180nm and 45nm. From this tool, the parameter performance was calculated such as area, power and delay. The comparison of the area, power, delay, APP and ADP for different technologies such as 180nm and 45nm presented in Table 2. In existing method [15], the normal digital adder was employed to perform the multiplication operation, which occupied more area. In the PRNG-LCC-CSLA method, the CSLA was employed in the multiplication, which used less area as compared to normal digital adder. The CSLA adder used in PRNG-LCC-CSLA architecture minimized the area, power, delay, APP and ADP than the conventional method. We have implemented existing method and proposed method. Based on implementation results, we have taken the performances value.

The comparison graph of the area, power, area power product and area-delay product is as shown in Fig. 7, 8, 9 and 10 respectively. The results were drawn by using 180nm & 45nm technology. From the graph it can be clearly concluded that the PRNG-LCC-CSLA method consumes less area, less power, less area power product and less area-delay product than conventional methods.

The reduction percentage of area, power, delay, APP, and ADP is as given in Table 3. PRNG-LCC-

CSLA architecture results were calculated using both 180nm & 45nm technology. In 180nm technology, 26.12 % of area, 78.79 % of power, 8.43 % of APP and 26.12 % of ADP minimized whereas in 45nm technology, 27.14 % of area, 75.95 % of power, 82.23 % of APP and 34.43 % of ADP reduced, compared to conventional methods.

Table 1. NIST statistical results

Statistical Test	P-value	Result
Frequency	0.4518	PASS
Block Frequency	0.7152	PASS
Runs	0.7815	PASS
Longest Run	0.0912	PASS
Rank	0.3024	PASS
DFT	0.1523	PASS
Non-overlapping transform	0.0023	PASS
Overlapping templates	0.9924	PASS
Universal	0.7215	PASS
Linear complexity	0.8941	PASS
Serial	0.1354	PASS
Approximate entropy	0.9536	PASS
Cumulative sums	0.5312	PASS
Random excursions	0.0174	PASS

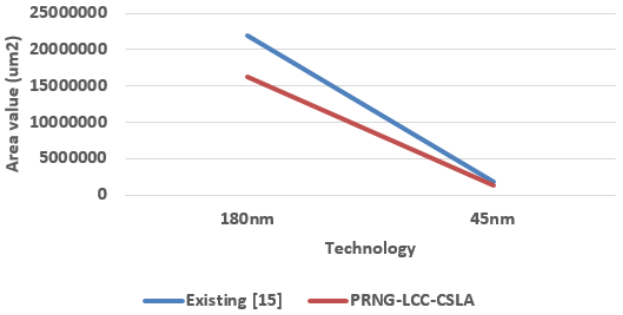


Figure.7 Area performance of different bits and tabs for 180nm & 45nm technology

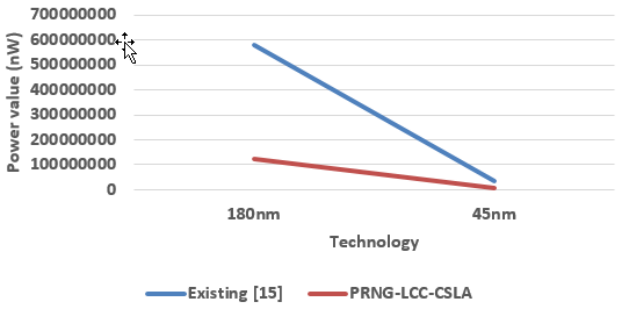


Figure.8 Power performance of different bits and tabs for 180nm & 45nm technology

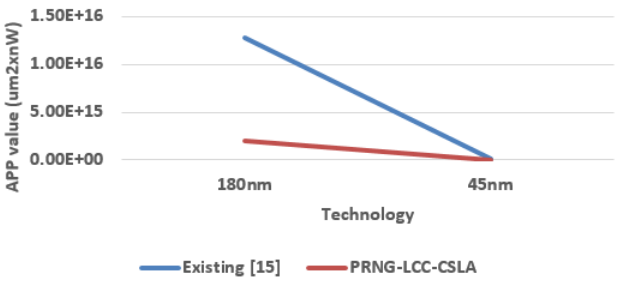


Figure.9 APP performance of different bits and tabs for 180nm & 45nm technology

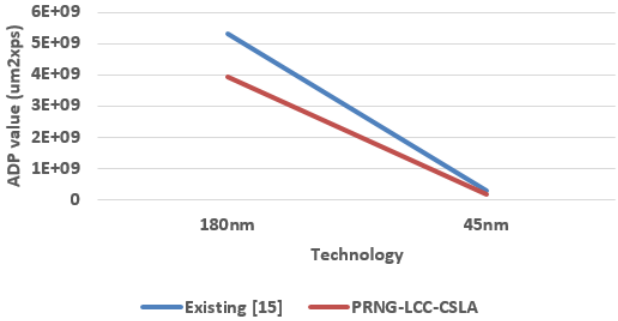


Figure.10 ADP performance of different bits and tabs for 180nm & 45nm technology

Table 2. The performance of area, power consumption and delay of PRNG-LCC-CSLA method for 180nm & 45nm Technology.

Technology	Method	Area (μm^2)	Power (nW)	Delay (ps)	APP (μm^2 x nW)	ADP (μm^2 x ps)
180nm	Existing [15]	22003133	576930454	242	1.269427751111238e+16	5324758186
	PRNG-LCC-CSLA	16255209	122360924	242	1989002393053116	3933760578
45nm	Existing [15]	1799703	36517169	160	65720058600807	287952480
	PRNG-LCC-CSLA	1329563	8780221	142	11673856973423	188797946

Table 3. Reduced percentage of area, power, delay, APP, and ADP for PRNG-LCC-CSLA method

Technology	Reduced % of Area	Reduced % of power	Reduced % of APP	Reduced % of ADP
180nm	26.12	78.79	8.43	26.12
45nm	27.14	75.95	82.23	34.43

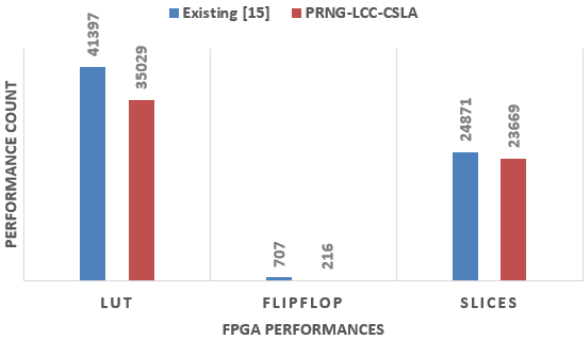


Figure.11 FPGA performance of the Virtex 4

Table 4. Comparison of the FPGA performances for different Xilinx FPGA devices

FPGA					
Target FPGA	Circuit	LUT	Flip-flop	Slice	Frequency (MHz)
Virtex4 xc4vlx200	Existing [15]	41397/178176	707/178176	24871/89088	5.05
	PRNG-LCC-CSLA	35029/178176	216/178176	23669/89088	3.549
Virtex5 xc5vlx330T	Existing [15]	30962/207360	409/207360	11540/51840	6.797
	PRNG-LCC-CSLA	28628/207360	216/207360	9817/51840	5.52
Virtex6 xc6vlx760dt	Existing [15]	35413/474240	370/948480	9692/118560	7.825
	PRNG-LCC-CSLA	31526/474240	208/948480	8068/118560	5.594

4.5 FPGA synthesis

FPGA synthesis was implemented in Xilinx tool for different devices such as Virtex-4, Virtex-5 and Virtex-6. From this tool, the performance parameter like LUT, flip-flop, Slices and Frequency were calculated.

Table 4 shows the comparison of the LUTs, flip-flops, slices and frequency for different Xilinx devices like vertex 4, 5 and 6. From this table, it is clear that the LUT, flip-flop and slices reduced in PRNG-LCC-CSLA method compared to the existing method. Due to the reduction of those parameters, the area has been minimized in filter architecture.

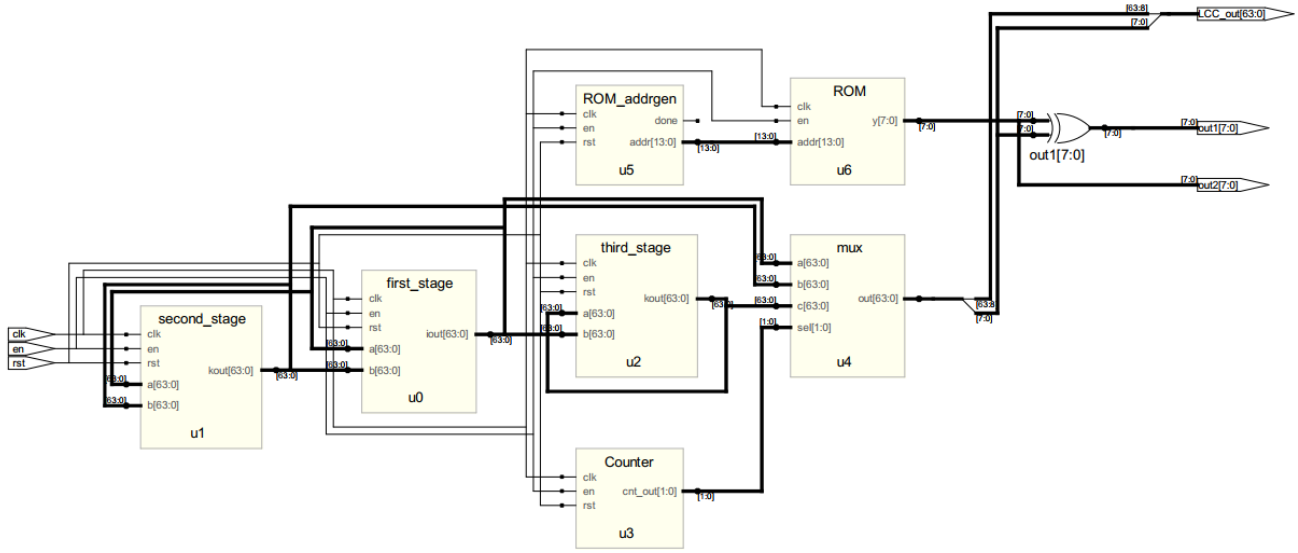


Figure.12 RTL schematic diagram of PRNG-LCC-CSLA

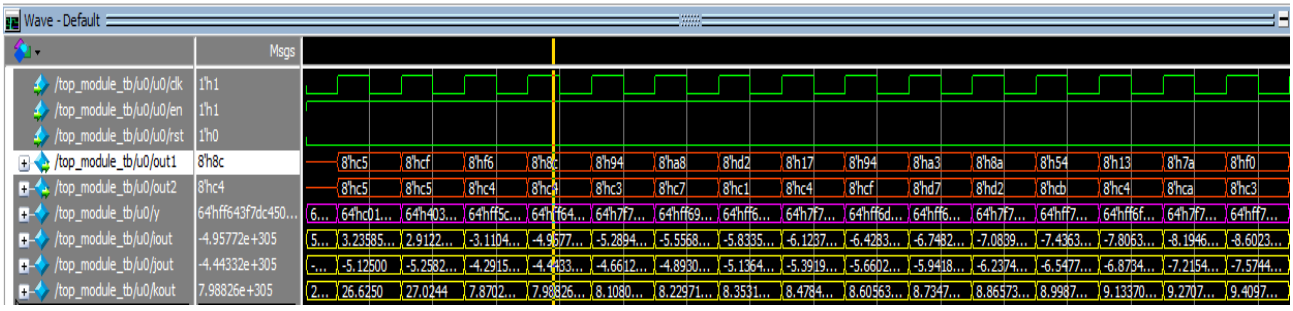


Figure.13. Output waveform of PRNG-LCC-CSLA method

FPGA performance of Virtex-4 device is as shown in fig. 11 for verification purpose. The graph shows FPGA values of LUT, Flip-flop, slices and Frequency for all Virtex devices. It indicates improved FPGA performance in PRNG-LCC-CSLA design than conventional design.

The RTL schematic of PRNG-LCC-CSLA is as shown in fig. 12, which was taken from Synplify pro software using Verilog code. This architecture has separate code for each block such as a First stage, second stage, third stage, counter, ROM and MUX. Input binary value is stored in a ROM. From this input, Encryption and Decryption have been performed

Fig. 13 shows the output waveform of PRNG-LCC-CSLA method. The LCC three stage outputs such as i, j, k, that outputs are given to the MUX. In the MUX circuit, the three outputs are converted into single output (y). This "y" represent pseudo random number. Out1 calculated for performing XOR operation with y and key. This is called as encryption output. This encryption output performed the XOR operation with key which gave decryption output (out2). This decrypted binary value is given to the MATLAB for getting decrypted image.

5. Conclusion

In this paper, PRNG-LCC-CSLA architecture was implemented in ModelSim software by using Verilog code. With the help of LCC, the pseudo random number has been generated to implement encryption and decryption operation. Image and text files have been taken for encryption and decryption. With FPGA implementation, quantity of LUT, slices and flip-flops utilized are minimized in PRNG-LCC-CSLA. In ASIC 180nm technology 26.12 % of area, 78.79 % of power, 8.43 % of APP and 26.12 % of ADP minimized in PRNG-LCC-CSLA. Also in 45nm technology 27.14 % of area, 75.95 % of power, 82.23 % of APP and 34.43 % of ADP reduced in PRNG-LCC-CSLA method compared to conventional method.

For future work different chaotic circuit can be used to improve the randomness and circuit optimization can be performed to reduce the area,

power and delay, for encryption and decryption by considering different images.

References

[1] A. Zaghloul, T. Zhang, M. Amin, and A.A. Abd El-Latif, "Color encryption scheme based on adapted quantum logistic map", In: *Proc. of 6th International Conf. On Digital Image Processing (ICDIP 2014)*, vol. 9159, p. 915922, 2014.

[2] A. Mohanty, K.B. Sutaria, H. Awano, T. Sato, and Y. Cao, "RTN in Scaled Transistors for On-Chip Random Seed Generation", *IEEE Transactions on Very Large Scale Integration Systems*, Vol.25, No.8, pp.2248-2257, 2017.

[3] R. Wang, G. Xu, B. Liu, Y. Cao, and X. Li, "Flow Watermarking for Antinoise and Multistream Tracing in Anonymous Networks", *IEEE MultiMedia*, Vol.24, No.4, pp.38-47, 2017.

[4] H. Liu, A. Kadir, and X. Sun, "Chaos-based fast colour image encryption scheme with true random number keys from environmental noise", *IET Image Processing*, Vol.11, No.5, pp.324-332, 2017.

[5] M. Moniruzzaman, M.A.K. Hawlader, and M.F. Hossain, "An image fragile watermarking scheme based on chaotic system for image tamper detection", In: *Proc. of International Conf. On Informatics, Electronics & Vision*, pp. 1-6, 2014.

[6] A. Akhshani, S. Behnia, A. Akhavan, S.C. Lim, and Z. Hassan, "An Image Encryption Approach Using Quantum Chaotic Map", I*nt. J. Adv. Computer Sci. Appl*, pp.2250-37654, 2014.

[7] C. Lin, X. Shen, and M. Lei, "Generation of plaintext-independent private key based on conditional decomposition strategy", *Optics and Lasers in Engineering*, Vol.86, pp.303-308, 2016.

[8] S. Venkateswarlu, G.M. Deepa, and G. Sriteja, "Implementation of Cryptographic Algorithm on FPGA", *International Journal of Computer Science and Mobile Computing*, Vol.2, No.4, 2013.

[9] H.I. Hsiao and J. Lee, "Color image encryption using chaotic nonlinear adaptive filter", *Signal Processing*, Vol.117, pp.281-309, 2015.

[10] H. Shimakage and Y. Tamura, "Chaotic Oscillations in Josephson Junctions for Random Number Generation", *IEEE Transactions on Applied Superconductivity*, Vol.25, No.3, pp.1-4, 2015.

[11] X. Fang, Q. Wang, C. Guyeux, and J.M. Bahi, "FPGA acceleration of a pseudorandom number generator based on chaotic iterations", *Journal of Information Security and Applications*, Vol.19, No.1, pp.78-87, 2014.

[12] I. Koyuncu and A.T. Özcerit, "The design and realization of a new high speed FPGA-based chaotic true random number generator", *Computers & Electrical Engineering*, Vol.58, pp.203-214, 2017.

[13] J.M. Bahi, X. Fang, C. Guyeux, and Q. Wang, "Suitability of chaotic iterations schemes using XORshift for security applications", *Journal of Network and Computer Applications*, Vol.37, pp.282-292, 2014.

[14] A.P. Johnson, R.S. Chakraborty, and D. Mukhopadyay, "An improved DCM-based tunable true random number generator for Xilinx FPGA", *IEEE Transactions on Circuits and Systems II: Express Briefs*, Vol.64, No.4, pp.452-456, 2017.

[15] C. Tanougast, A. Dandache, M.S. Azzaz, and S. Sadoudi, "Hardware Design of Embedded Systems for Security Applications", In: *Proc. of International Conf. on Embedded Systems-High Performance Systems, Applications and Projects, InTech*, pp.233-260, 2012.