ATTRACTOR INFLUENCED PRNG FOR CRYPTOGRAPHIC KEY GENERATION ON FPGA

Hemanth Kumar Nalajala, Rajesh B, Sivaraman R, Sridevi A,
Amirtharajan Rengarajan and Sundararaman Rajagopalan

School of EEE, SASTRA Deemed to be University, Thanjavur, Tamilnadu

ABSTRACT

Random numbers play a significant role while implementing the crypto architectures on reconfigurable hardware. Chaotic attractors are reliable sources of deterministic random number generation due to their large keyspace capability. Chaos exhibits random perturbations in floating-point units which is a challenge while replicating the system of equations on hardware. The conversion of floating-point numbers to binary representation will take many quantization possibilities which certainly affect the randomness and sensitivity to initial conditions. This work aims to implement the chaotic Rössler attractor based Pseudo Random Number Generation (PRNG) on FPGA through simulation and real time experimentation. This attractor has been realized on Altera Cyclone II EP2C20F484C7 FPGA using hardware primitives. It required 201 LUTs with a power dissipation of 78.48 mW and time duration of 4 µS to generate 32,768 random bits. The randomness of this attractor was evaluated through entropy, correlation, bit distribution and NIST SP 800 – 22 analyses.

KEYWORDS

Chaos, PRNG, Quantization effects, Reconfigurable Hardware, Rössler attractor, Differential equations & FPGA

1. INTRODUCTION

Advancements in networking have revolutionized the communication in a commendable manner. Considering the huge amount of data being shared over the internet, it is necessary to preserve the various facets of information security. Cryptographic algorithms have a significant role in enhancing the confidentiality of data by all means [1]. In cryptography, key generation occupies a primary role for both the symmetric as well as asymmetric approaches. Random keys play a decisive role in improving the quality of cipher generated. Generally, keys are generated through Random Number Generators (RNGs) which are classified into two categories namely Pseudo Random Number Generator (PRNG) and True Random Number Generator (TRNG) [2]. Noticeably, PRNGs have been utilized widely in data encryption because of their high level of randomness. They employ mathematical equations or fixed architecture driven by a unique applicable initial condition and/or seed value [3]. Recently, hardware accelerated PRNG implementations are in greater demand due to the high speed requirements. Especially, Field Programmable Gate Array (FPGA) based PRNG algorithms have attained a substantial requirement due to their unique characteristics such as reconfigurability, algorithm agility, concurrency, faster time to market, easy prototyping and other on – chip / off – chip capabilities [4].

N. Ramasubramanian et al.(Eds) : SRAIC-2019
pp. 59-70, 2019. © CS & IT-CSCP 2019 DOI: 10.5121/csit.2019.91505

Some of the PRNG techniques are based on Linear Congruential Generator (LCG) [5], Quadratic Congruential Generator (QCG) [6], Linear Feedback Shift Register (LFSR) [7], Cellular Automata (CA) [8], etc. The utilization of the concept of chaos in information security brings enormous advantages because of its inherent properties such as sensitivity to initial condition and large perturbations [9]. Chaos can be exploited either in continuous or in discrete forms. Chaotic maps comprise 1D discrete equation(s) which are driven by suitable initial conditions and seed to generate random numbers.

In various works, logistic map [10], Tent map [11], Henon map [12], Bernoulli map [13] have been the frequently used 1D chaotic systems for random number generation. Despite the yield of good randomness from the chaotic maps, their keyspace is limited which make them vulnerable to brute force attacks. To enhance the keyspace along with randomness, chaotic attractors are suggested. Chaotic attractors are multi – dimensional continuous chaotic systems which consist of more number of control parameters. Lorenz, Lu, Chen and Rössler attractors have been identified as good random number generators due to their versatility while implementing their architecture on FPGA.

Zidan et al. proposed a PRNG based on Lorenz and Chen chaotic attractors which were implemented on Xilinx Virtex 4 FPGA. This implementation has been verified through lyapunov exponent and autocorrelation confidence region. This work required 658 slices of configurable logic blocks and 97 flip-flops to accommodate the design at an operating frequency of 13.17 MHz [14]. Azzaz et al. implemented the Lorenz attractor on Xilinx Virtex II FPGA using Runge – Kutta 4th order approach which utilizes 1926 slices yielding 124 Mbps as throughput for the operating frequency of 15.5 MHz [15]. Further, Schmitz and Zhang designed a continuous chaotic system through Rössler attractor using VHDL on Xilinx ZYNQ 7000 series FPGA. This implementation consumed 433 slices and 96 registers. In addition, this design utilized 12 DSP blocks to achieve 2850 Mbps of throughput [16]. Zhang has suggested yet another implementation of Lorenz attractor on Xilinx Spartan 3E and ZYNQ 7020 FPGAs which required 1029 and 338 slices respectively. It also required 8 DSP blocks with 0.153 mW of power dissipation for the 32-bit implementation of attractor [17]. Rezk et al. enhanced the Lorenz and Lu attractor's implementation using hardwired shifting and multiplexing schemes. This PRNG was designed using VHDL and implemented on Xilinx XC5VLX50T FPGA whose randomness was evaluated through NIST SP 800 – 22 batteries of tests. This design operated at a frequency of 78MHz and consumed 100 slices and 96 flip-flops to generate pseudo random numbers [18].

In general, the transformation of differential equations into time domain representation has been performed through three different approaches namely Euler's method, Mid – point method and Runge – Kutta 4th order approach. The above-mentioned implementations have utilized all the three approaches where Euler's method has significant advantage in terms of area and throughput. With this inference, the proposed work focuses on the implementation of Rössler attractor on Cyclone II FPGA using Verilog HDL with Euler's method. The significant advantages of the proposed work are:

- Lightweight continuous chaotic system
- NIST SP 800 – 22 verified random source
- Moderate throughput with 50 MHz operating clock frequency

2. PRELIMINARIES

In this work, the architecture of chaotic Rössler attractor for designing PRNG on FPGA is proposed. The chaotic attractors are discrete dynamical systems which have attributes namely more sensitivity to initial conditions, deterministic, ergodicity, stochasticity and periodicity. The

randomness of chaotic system has been tested through bifurcation and Lyapunov exponent analysis. The Rössler attractor is known for its simplicity and more chaotic span. It is a system comprising three non-linear ordinary differential equations defined by Otto Rössler [19]. These differential equations define a continuous-time dynamical system that exhibit chaotic dynamics. The set of three dimensional equations of Rössler attractor are given in equations (1 – 3):

$$\frac{dx}{dt} = -y - z \qquad (1)$$

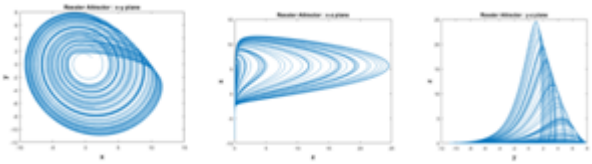
$$\frac{dy}{dt} = x + ay \qquad (2)$$

$$\frac{dz}{dt} = b + z(x - c) \qquad (3)$$

Where a, b and c are the control parameters and x0, y0 and z0 are the initial conditions which trigger the process of generation of time series. Table 1 presents the control parameters chosen and assumed initial conditions of Rössler attractor whereas Fig.1 (a – c) depict the responses of complex behavior in x, y and z directions.

Table 1. Control parameters and initial conditions.

Parameters	Case -1	Case - 2	Case - 3
a	0.2	0.2	0.15
b	0.1	0.2	0.20
c	14	5.7	5.7
x_0	1	1	0.1
y_0	1	1	5.0000000001
z_0	0	0	25



(a) (b) (c)

Figure 1. Complex behavior of Rössler attractor: (a) X - Y plane (b) Y - Z plane and (c) Z - X plane [19].

3. PROPOSED METHODOLOGY

In this proposed work, the Rössler attractor has been designed with the combination of multiplier, adder and subtractor whose architectural representation is shown in Fig. 2. The following steps were carried out to generate random numbers by implementing the attractor design on FPGA.

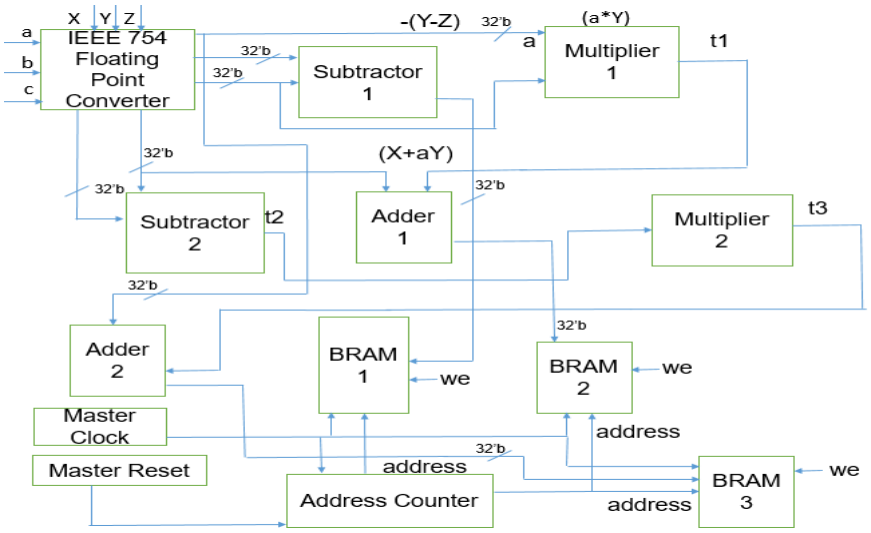


Figure 2. Block diagram of the proposed work.

Step I: Convert the control parameters and initial conditions of Rössler attractor from floating point to binary values using IEEE 754 single precision conversion.

Step II: From the observation of Fig. 2, design requires two subtractors, adders and multipliers. Design the primitive components using Quartus II 13.0 mega functions.

Step III: Develop the Finite State Machine (FSM) using HDL to control and drive the operation of X, Y and Z planes. Also, generate the necessary control signals for memory and end of operation.

Step IV: Create the first state of FSM to perform the required mathematical operations of attractor and make the second state to feedback the results to the attractor variables to \ generate the random numbers continuously.

Step V: Store the random values in Block Random Access Memory (BRAM) of FPGA for further analysis.

Rössler attractor has been designed using Verilog HDL and the functionality was verified through Modelsim 6.0 platform as shown in Fig. 3. Quartus II 13.0 EDA tool was used for implementing the design on FPGA after creating the equivalent Register Transfer Level (RTL) schematic where the architecture has been represented in terms of logic gates and registers as portrayed in Fig. 4. Figure 5 (a – c) show the BRAM screen shots of ten 32 – bit random numbers generated by the FPGA through Rössler attractor with the parameters of case – 3.

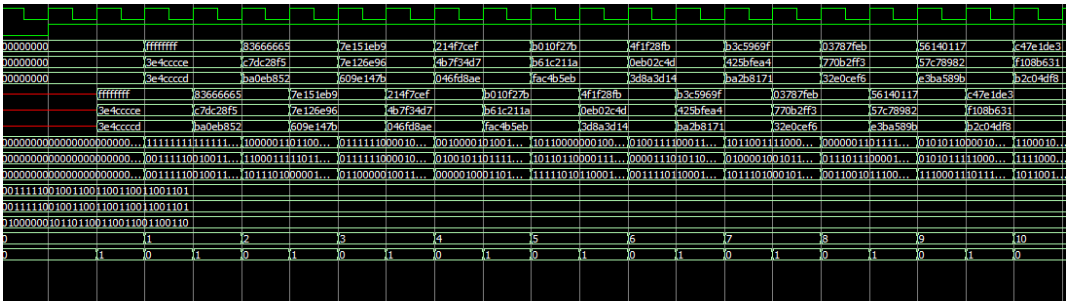


Figure 3. Simulation result of Rössler attractor.

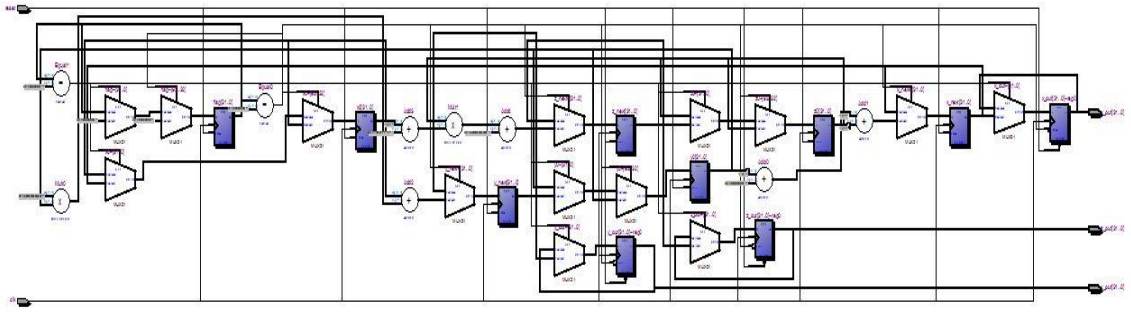


Figure 4. RTL diagram of Rössler attractor.

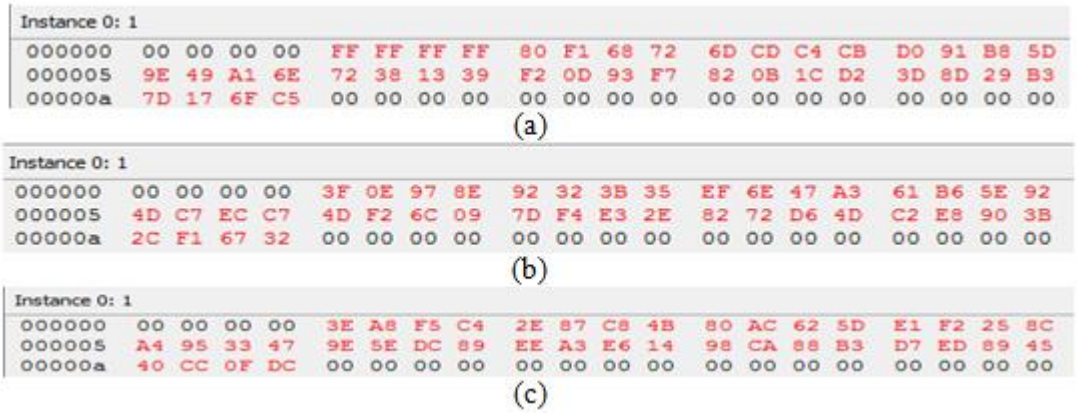


Figure 5. BRAM shots of Rössler attractor for Case-3 (a) X – Plane (b) Y – Plane and (c) Z – Plane.

4. RESULTS AND DISCUSSION

Statistical properties of the proposed PRNG were verified through NIST SP 800 – 22 batteries of test with three different sets of control parameters. In addition, entropy, correlation and bit distribution analyses were carried out to verify the randomness.

4.1. NIST SP 800 – 22 Batteries of Test

It is a statistical package consisting of many tests that have been developed to test the randomness of the binary sequences produced by random number generators [20]. Some tests are divided into many subtests. NIST tests were performed with 1,00,000 of random bits for all the three cases of control parameters with the following constraints.

Block frequency Test – Block length (M) = 128
Non-overlapping Template – Block length (m) = 9
Overlapping Template – Block length (m) = 9
Approximate Entropy Test – Block length (m) = 8
Serial Test – Block length (m) = 10
Linear Complexity Test – Block length (M) = 500

Table 2, 3 and 4 present the results for NIST SP 800 – 22 tests for Case 1, 2 and 3 of control parameters wherein case – 3 yields better results because of its sensitivity and stochasticity.

Table 2. NIST Results of random numbers generated using Case -1 control parameters.

Tests	Case – 1 (a = 0.2, b =0.1, c = 14, x_0 = 1, y_0 =1 & z_0 = 0)					
	X	Statu	Y	Status	Z	Status
Frequency	0.00000	Failed	0.21330	Passed	0.21330	Passed
Block Frequency	0.21330	Passed	0.35048	Passed	0.35048	Passed
Cumulative Sums – I	0.00000	Failed	0.74991	Passed	0.74981	Passed
Cumulative sums – II	0.00000	Failed	0.21330	Passed	0.213330	Passed
Runs	0.00000	Failed	0.12232	Passed	0.12232	Passed
Longest Runs	0.00232	Failed	0.73991	Passed	0.73991	Passed
Rank	0.01430	Passed	0.53414	Passed	0.53414	Passed
FFT	0.35048	Passed	0.78591	Passed	0.73991	Passed
Non Overlapping Template	0.00001	Failed	0.92341	Passed	0.91141	Passed
Overlapping Template	0.00887	Passed	0.03517	Passed	0.00004	Failed
Approximate Entropy	0.35048	Passed	0.73991	Passed	0.73991	Passed
Serial - I	0.00138	Failed	0.00001	Failed	0.73991	Passed
Serial - II	0.21330	Passed	0.00018	Failed	0.35048	Passed
Linear Complexity	0.73991	Passed	0.35048	Passed	0.35048	Passed

Table 3. NIST Results of random numbers generated using Case -2 control parameters

Tests	Case –2 (a = 0.2, b = 0.2, c = 5.7, x_0 = 1, y_0 =1 & z_0 = 0)					
	X	Status	Y	Status	Z	Status
Frequency	0.00000	Failed	0.00000	Failed	0.00000	Failed
Block Frequency	0.02365	Passed	0.21330	Passed	0.91141	Passed
Cumulative Sums – I	0.00000	Failed	0.00000	Failed	0.00000	Failed
Cumulative Sums – II	0.00000	Failed	0.00000	Failed	0.00000	Failed
Runs	0.00000	Failed	0.00000	Failed	0.00887	Passed
Longest Runs	0.06688	Passed	0.00000	Failed	0.00000	Failed
Rank	0.00000	Failed	0.00000	Failed	0.00000	Failed

FFT	0.00000	Failed	0.00000	Failed	0.00887	Passed
Non Overlapping Template	0.00000	Failed	0.00000	Failed	0.91141	Passed
Overlapping Template	0.35048	Passed	0.91141	Passed	0.02431	Passed
Approximate Entropy	0.00000	Failed	0.00000	Failed	0.00000	Failed
Serial - I	0.00000	Failed	0.00000	Failed	0.00000	Failed
Serial - II	0.00001	Failed	0.06688	Passed	0.00887	Passed
Linear Complexity	0.21330	Passed	0.03517	Passed	0.73991	Passed

Table 4. NIST Results of random numbers generated using Case -3 control parameters.

Tests	**Case –3 (a = 0.15, b =0.20, c = 5.7, x_0 = 0.1, y_0 =5.0000000001 & z_0 = 25)**					
	X	**Statu**	**Y**	**Status**	**Z**	**Status**
Frequency	0.19921	Passed	0.73991	Passed	0.73991	Passed
Block Frequency	0.21330	Passed	0.21330	Passed	0.21330	Passed
Cumulative Sums – I	0.44512	Passed	0.53414	Passed	0.73991	Passed
Cumulative Sums – II	0.12526	Passed	0.99146	Passed	0.21330	Passed
Runs	0.03887	Passed	0.53414	Passed	0.00887	Passed
Longest Runs	0.06688	Passed	0.73991	Passed	0.73991	Passed
Rank	0.01430	Passed	0.53414	Passed	0.53414	Passed
FFT	0.12232	Passed	0.02791	Passed	0.35048	Passed
Non Overlapping Template	0.99146	Passed	0.91141	Passed	0.91141	Passed
Overlapping Template	0.21330	Passed	0.99146	Passed	0.91141	Passed
Approximate Entropy	0.12232	Passed	0.06688	Passed	0.73991	Passed
Serial - I	0.35048	Passed	0.73991	Passed	0.73991	Passed
Serial - II	0.73991	Passed	0.53414	Passed	0.06688	Passed
Linear Complexity	0.91141	Passed	0.35048	Passed	0.91141	Passed

From the above results, it was inferred that the Case – 3 passed all the tests in NIST SP 800 – 22 which ensured its strength of randomness.

4.2. Entropy Analysis

Entropy is a fundamental measure of uncertainty which describes the amount of probability of 0's and 1's in a random sequence [21]. It is used to determine the equi - distribution property of random numbers. For a strong set of random numbers, the entropy value must be close to 1. To analyze the equi – distribution property, 32 – bit random numbers from Rössler attractor were divided as 8, 16 and 32 – bits with respect to the LSB and MSB positions. The results of entropy analysis are listed in Table 5. From the Table 5, it is observed that the 32-bit random numbers possess near 1 entropy when compared to 16-bit and 8-bit segments of all the cases.

Table 5. Entropy analysis.

Bits	*Case 1(X)*	*Case 1(Y)*	*Case 1(Z)*	*Case 2(X)*	*Case 2(Y)*	*Case 2(Z)*	*Case 3(X)*	*Case 3(Y)*	*Case 3(Z)*
32	0.999788	0.999636	0.999953	0.999304	0.999304	0.999306	0.999901	0.999998	0.999981
0 – 16	0.993846	0.999925	0.999915	0.999791	0.997365	0.997114	0.999439	0.999924	0.999945
17 –31	0.999845	0.999947	0.999965	0.997915	0.997385	0.997224	0.999479	0.999984	0.999947
9 – 24	0.999876	0.999998	0.999998	0.997925	0.997686	0.997254	0.999785	0.999995	0.999993
0 – 7	0.999876	0.999998	0.999998	0.997925	0.997686	0.997254	0.999785	0.999995	0.999993
8 – 15	0.988025	0.999998	0.99991	0.974932	0.988897	0.988543	0.989123	0.999432	0.999763
16 –23	0.988032	0.999993	0.99997	0.974923	0.988843	0.988345	0.989345	0.999561	0.999761
24 –31	0.988024	0.999994	0.99993	0.974943	0.988854	0.988567	0.989321	0.999287	0.999376

4.3. Correlation Analysis

This metric determines the data dependencies between the random numbers. To become a cryptographically strong PRNG, the correlation must be very low. This analysis was performed for Rössler attractor generated random numbers with Case – 3 control parameters and the results are depicted in Fig. 6 (a – c). The figures convey no existence of correlation among the generated numbers. The data distribution of random sequences generated by Rössler attractor are presented in Fig. 7 (a – c) to ensure the presence of randomness.

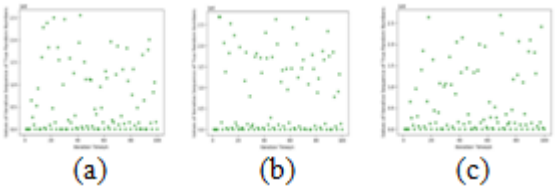


Figure 6. Correlation analysis: (a) X – Plane (b) Y – Plane and (c) Z – Plane.

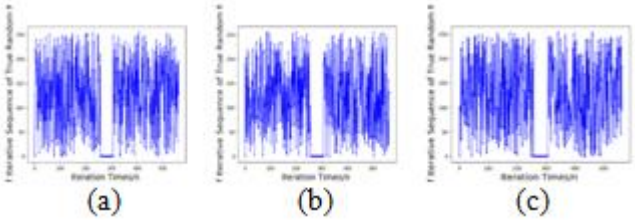


Figure 7. Data distribution analysis: (a) X – Plane (b) Y – plane and(c) Z – Plane.

4.4. Hardware Analysis

As the attractor is implemented on FPGA, it is necessary to evaluate the hardware efficiency in terms of the utility of standard measures such as Look Up Tables (LUTs), combinational logics, dedicated logic registers, on – chip memory bits and power dissipation. The proposed implementation consumes only 2% of total logic elements of Cyclone II FPGA EP2C20F484C7 to construct the Rössler attractor where the total power dissipation is 78.48 mW obtained through Power Play Power Analyzer in Quartus II 13.0 EDA tool for 12.5% toggling rate. Table 6 presents the hardware analyses of the proposed design.

Table 6. Hardware analysis.

Target FPGA	Cyclone II EP2C20F484C7 FPGA
Total Logic Elements	201/18,752 (1%)
Total Combinational Functions	289/18,752 (2%)
Dedicated logic registers	289
Total registers	98/315 (31%)
Total memory bits	131072
Embedded Multiplier 9-bit elements	12/52 (23%)
Total power dissipation (mW)	78.48mW
Time taken for 1024 × 32 bits	4.0 µS

To analyze the performance efficiency, the proposed work has been compared with other earlier works of attractor designs on various FPGAs which are presented in Table 7. From the comparison, this design is superior in terms of logic elements consumption and throughput.

Table 7. Performance comparison.

Criteria	Proposed work	Ref. [14]	Ref. [15]	Ref. [16]	Ref. [17]	Ref. [18]
Attractor	Rössler	Lorenz	Lorenz	Rössler	Lorenz	Lorenz + Lu
Number of Equations	3	3	3	3	3	4
Random Number Size	32 Bits	32 Bits	32 Bits	32 Bits	32 Bits	32 Bits
Target FPGA	Cyclone II	Virtex II	Virtex IV	ZYNQ 7000	ZYNQ 7020	Virtex V
LUTs	201	2718	287	433	868	276
Registers	289	791	96	96	96	96
DSP Blocks	12(Embedded multiplier)	-	8	12	8	8
Frequency(MHz)	50	15.59	53.53	70.9	36.3	78.149
NIST SP 800 - 22	PASS	-	-	-	-	PASS

5. CONCLUSION

The Rössler attractor based 32 – bit PRNG has been implemented on Cyclone II FPGA using Verilog HDL and Quartus II 13.0 EDA tool. This proposed design was realized through Euler's method which is the best way to represent differential equations in time domain. This work consumes 201 LUTs to accommodate the Rössler attractor design which requires only 4.0 µS to generate 32,768 random bits. Statistical characteristics of the proposed design have been verified through NIST SP 800 – 22 tests, entropy and correlation analyses. Future work will be on developing an image security system using the Rössler attractors.

ACKNOWLEDGEMENTS

The authors would like to thank SASTRA Deemed University for providing infrastructure through the Research & Modernization Fund (Ref. No: R&M / 0026 / SEEE – 010 / 2012 – 13) to carry out the research work.

REFERENCES

[1] Y. Hardy and W.-H. Steeb, "Cryptography," in Classical and Quantum Computing: with C++ and Java Simulations, Basel: Birkhäuser Basel, 2001, pp. 215–228.

[2] A. THESEN, "Chapter IX - Random Number Generators," in Computer Methods in Operations Research, A. THESEN, Ed. Academic Press, 1978, pp. 194–213.

[3] S. Z. Li and A. Jain, Eds., "Pseudo-Random Number Generator," in Encyclopedia of Biometrics, Boston, MA: Springer US, 2009, p. 1100.

[4] A. J. Elbirt, W. Yip, B. Chetwynd, and C. Paar, "An FPGA-based performance evaluation of the AES block cipher candidate algorithm finalists," IEEE Trans. Very Large Scale Integr. Syst., vol. 9, no. 4, pp. 545–557, 2001.

[5] S. Tezuka, “Linear Congruential Generators,” in Uniform Random Numbers: Theory and Practice, Boston, MA: Springer US, 1995, pp. 57–82.

[6] S. Strandt, “Quadratic Congruential Generators With Odd Composite Modulus,” in Monte Carlo and Quasi-Monte Carlo Methods 1996, 1998, pp. 415–426.

[7] M. H. Weik, “Linear-Feedback Shift Register,” in Computer Science and Communications Dictionary, Boston, MA: Springer US, 2001, p. 896.

[8] L. Petrica, “FPGA optimized cellular automaton random number generator,” J. Parallel Distrib. Comput., vol. 111, pp. 251–259, 2018.

[9] J. Pejaś and A. Skrobek, “Chaos-Based Information Security,” in Handbook of Information and Communication Security, P. Stavroulakis and M. Stamp, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 91–128.

[10] S. L. Chen, T. Hwang, and W. W. Lin, “Randomness enhancement using digitalized modified logistic map,” IEEE Trans. Circuits Syst. II Express Briefs, vol. 57, no. 12, pp. 996–1000, 2010.

[11] A. Ilyas, A. Luca, and A. Vlad, “A study on binary sequences generated by tent map having cryptographic view,” 2012 9th Int. Conf. Commun. COMM 2012 - Conf. Proc., pp. 23–26, 2012.

[12] M. Suneel, “Cryptographic pseudo-random sequences from the chaotic Hénon map,” Sadhana, vol. 34, no. 5, pp. 689–701, Oct. 2009.

[13] D. J. Driebe, “The Bernoulli Map,” in Fully Chaotic Maps and Broken Time Symmetry, Dordrecht: Springer Netherlands, 1999, pp. 19–43.

[14] M. A. Zidan, A. G. Radwan, and K. N. Salama, “The effect of numerical techniques on differential equation based chaotic generators,” Proc. Int. Conf. Microelectron. ICM, pp. 1–4, 2011.

[15] M. S. Azzaz, C. Tanougast, S. Sadoudi, and A. Dandache, “Real-time FPGA implementation of Lorenz’s chaotic generator for ciphering telecommunications,” in IEEE North-East Workshop on Circuits and Systems and TAISA Conference, NEWCAS-TAISA ’09, 2009, pp. 1–4.

[16] J. Schmitz and Z. Lei, “Rössler-based chaotic communication system implemented on FPGA,” Can. Conf. Electr. Comput. Eng., pp. 1–4, 2017.

[17] L. Zhang, “System generator model-based FPGA design optimization and hardware co-simulation for Lorenz chaotic generator,” 2017 2nd Asia-Pacific Conf. Intell. Robot Syst. ACIRS 2017, no. 2, pp. 170–174, 2017.

[18] A. A. Rezk, A. H. Madian, A. G. Radwan, and A. M. Soliman, “Reconfigurable chaotic pseudo random number generator based on FPGA,” AEU - Int. J. Electron. Commun., vol. 98, pp. 174–180, 2019.

[19] D. T. Maris and D. A. Goussis, “The ‘hidden’ dynamics of the Rössler attractor,” Phys. D Nonlinear Phenom., vol. 295–296, no. M, pp. 66–90, 2015.

[20] L. E. Bassham et al., “A statistical test suite for random and pseudorandom number generators for cryptographic applications,” Natl. Inst. Stand. Technol., Gaithersburg, MD, USA, Tech. Rep., no. April, 2010.

[21] R. Sivaraman, S. Rajagopalan, A. Sridevi, J. B. B. Rayappan, M. P. V. Annamalai, and A. Rengarajan, “Metastability-Induced TRNG Architecture on FPGA,” Iran. J. Sci. Technol. Trans. Electr. Eng., vol. 2, 2019.

AUTHORS

Hemanth Kumar Nalajala is currently pursuing his B. Tech (Electronics and Communication Engineering) in SASTRA Deemed to be University, Thanjavur. His research areas include information security and FPGA implementations of cryptographic algorithms.

Rajesh B is currently pursuing his B. Tech (Electronics and Communication Engineering) in SASTRA Deemed to be University, Thanjavur. His research areas include information security and FPGA implementations of cryptographic algorithms.

Sivaraman R received his B.Tech (Electronics and Communication Engineering) in 2014, from SRC, SASTRA University, Kumbakonam and M.Tech in VLSI Design from SASTRA University, Thanjavur in 2016. He is currently working as research scholar in school of electrical and electronics engineering, SASTRA University, India. His research areas include information security, digital system design of hardware peripherals and embedded system. He has published 16 research articles in the national and international journals.

Sridevi A completed her B. Tech in Electronics & Communication Engg. and M. Tech in Communication& Networking in the years 2009 and 2015 respectively. She has one year of industrial experience and two years of teaching experience. She is currently pursuing her Ph.D. in the domain of Multimedia information security at SASTRA Deemed University, Thanjavur. Her research areas include information security, Multimedia communication, Machine Learning and embedded system. She has published 13 research articles in the national and international journals.

Dr. R. Amirtharajan was born in Thanjavur, Tamil Nadu province India, in 1975. He received B.E. degree in Electronics and Communication Engineering from P.S.G. College of Technology, Bharathiyar University, Coimbatore, India in 1997. M.Tech. and Ph. D. from SASTRA University Thanjavur, India in 2007 and 2012 respectively. He joined SASTRA University, Thanjavur, Tamil Nadu, India (Previously Shanmugha College of Engineering) as a Lecturer in the Department of Electronics and Communication Engineering since 1997 and is now Associate Professor, His research interests include Image Processing, Information Hiding, Computer Communication and Network Security. So far, he filed one international patent; he has published more than 125+ research articles in national and international journals and 28 IEEE conference papers with 4 Best Paper Awards. He also holds the Certificate of Appreciation from IBM in 2009 for Great Mind Challenge, Mentor IBM Academic Initiative Program. Recently, he received the Founder Chancellor Award for the best Ph.D. thesis for 2013 from SASTRA University and he received the SASTRA Anukul Puraskar for Higher Involvement in Research and Education Award for 2011-2012 and 2013-2014. He serves as a Life Member in CRSI, SSI, IAENG, and IACSIT. He also served as the TPC Member and Review Member for more than 30+ IEEE and Springer supported international conferences apart from more than 10 peer reviewed journals. He had been working on funded project in the field of steganography supported by DRDO, Government of India, New Delhi, India.

Sundararaman Rajagopalan completed his B.Tech in Electronics & Instrumentation Engg., M.Tech in Advanced Communication Systems and Ph.D thesis in Steganography System on Reconfigurable Hardware in the years 2005, 2007 and 2016 respectively. He is currently working as Asst. Professor in the Dept. of ECE, SASTRA University. Presently he is carrying out a DRDO, Govt. of India funded project on True Random Key Generation. He carried out a funded project by DRDO on steganography as a Co-Principal Investigator between the years 2010 - 2013. He has published 15 articles on information security. Also he was a WIPRO ULK faculty resource guide team Contributor for FPGA based experiments with Unified Technology Learning Platform (UTLP). He was also a faculty guide for IBM Remote Mentoring Projects for the years 2011 and 2012.