International Journal of Computational and Electronic Aspects in Engineering
Volume 6, Issue 1, March 2025, pp 24-34
https://doi.org/10.26706/ijceae.6.1.20250205

Efficient Real-Time Key Generation for IoT Using Multi-Dimensional Chaotic Maps

Harith G. Ayoub[1], **Osama A. Qasim**[2], **Zaid A. Abdulrazzaq**[3], **Mohammed S. Noori**[4]

Northern Technical University, (NTU), Mosul, Iraq

[1] **Corresponding Author:** harithga@ntu.edu.iq
[2] osama.hassani@ntu.edu.iq
[3] zaid.a.abdulrazzaq@ntu.edu.iq
[4] Moh.sami@ntu.edu.iq

Abstract: The paper presents a new approach for triple key generation techniques with Henon, Lozi and Duffing chaotic maps in FPGA. Three key streams generated by XORing chaotic maps with pseudo random number generators, the results of XORing PN1 with PN2 is called gold code generator. The algorithm was built in embedded Xilinx System Generator tool (XSG) and the results were validated using MATLAB/SIMULINK environment. NIST, Normality and Dieharder test randomness tests were employed for key streams and verified as the best randomness in comparison with recent work.

Keywords: FPGA, PN, XSG, Chaos, Gold Code.

RAME PUBLISHERS

INTERNATIONAL JOURNAL OF COMPUTATIONAL AND ELECTRONICS ASPECTS IN ENGINEERING

ISSN: 2395-4116 (PRINT)
ISSN: 2395-4124 (ONLINE)

A Quarterly Publishing Journal

Article – Peer Reviewed
Received: 20 January 2025
Accepted: 10 March 2025
Published: 22 March 2025

Cite this article: Harith G. Ayoub, Osama A. Qasim, Zaid A. Abdulrazzaq, Mohammed S. Noori, "Efficient Real-Time Key Generation for IoT Using Multi-Dimensional Chaotic Maps", *International Journal of Computational and Electronic Aspects in Engineering*, RAME Publishers, vol. 6, issue 1, pp. 24-34, 2025.
https://doi.org/10.26706/ijceae.6.1.20250205

1. Introduction

The prevalence of digital images in various fields has led to a growing need for secure transmission and storage of confidential information. Images transmitted through shared or public networks are particularly vulnerable to attacks, posing a significant challenge in terms of protection [1–4]. Cryptography used for protection, which is the process of converting known images to undefined images to make them non-knowledgeable or unpredictable by attackers. Recent research in cryptography has focused on the use of pseudo- random number generators (PRNGs) based on chaotic systems due to their high randomness factor and sensitivity to initial conditions [5–10]. Chaotic maps and image encryption involve the application of chaotic dynamics in information security. These maps are deterministic and generate uniformly distributed values [11-14]. The complexity and cost of traditional image encryption algorithms, as well as their susceptibility to attacks, have prompted the use of hardware implementation, specifically field programmable gate array (FPGA) [15–18].

A digital image is a two-dimensional array containing MxN pixels; in this array, M is the number of rows or the height of the image, and N is the number of columns or the width of the image. Each pixel in this digital array can take only specific numerical values, and the set of these values depends on the type of image Different pixels in the image carry information that can have various meanings depending on the application it is used for. It may be a binary image which carries only two specific values usually zero and one, i.e., a single bit per pixel, or grayscale images carrying 8 bits per pixel for different gray levels, or it may be a color image that carries the information about colors [19–21].

This paper discusses the use of various chaotic systems—Lozi, Henon, and Duffing maps—to generate three key streams system to encrypt each channel of a three- dimensional color image. The key generation process also enhanced by incorporating pseudo and gold code number generators. Additionally, all operations accelerated using the ZYNQ702 FPGA with the Xilinx System Generator tool (XSG).

2. Main Contributions

A. Design Henon, Duffing and Lozi chaotic maps and checked the sequence generated by them.

B. Build the models in XSG form and export the results to MATLAB/SIMULINK environment to be ready for test.

C. Implement the XSG model in using ZYNQ702 evolution board through VIVADO software tool and get maximum frequency and silicon area.

D. Apply NIST, Normality and Dieharder randomness Tests for checking the unpredictability distribution of generated keys.

3. Methods

A. Proposed key generator System

The proposed key generator algorithm illustrated in the following Fig.1, Focusing on symmetric secret keys shared between the sender and receiver in a communication system. After the synchronization process, keys need for encryption begins to prepare for the send/receive process.

Generating three channel streams could be utilized for channel digital image first ciphered by key stream1, second with key stream, third with key stream 3.

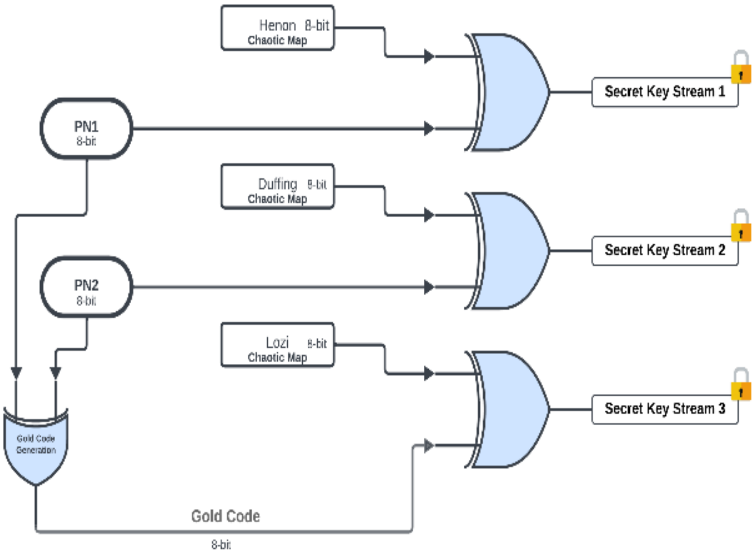


Figure 1. Three Key Streams Producing Using Proposed Approach.

A. PRBG

Pseudorandom binary generators (PRBGs) or random binary sequence generators play a critical role in cryptographic applications. Whenever the need arises for a binary sequence with high random behavior, a PRBG is employed before actual use. Even if the requirement is not exactly this, a spoof of indispensable confidential information, plain text, pseudorandom, or random key must be used. For proper performance of any cryptographic algorithm, appropriate PRBGs and pseudorandom numbers are essential for a fault-tolerant application in cryptosystems for end-to-end security, confidentiality, and integrity [22][23]. Because of the importance of random binary sequences, methods have been developed to test such genera- tors. Also, some hardware and software has been developed to generate such random sequences for non-security applications, like simulation studies [36-37]. The relevance of pseudorandom generators in cryptography, and XSG as it has relevance in cryptography [24].

B. FPGA-ZYNQ implementation via Xilinx System Generator

Xilinx System Generator (XSG) is an integrated development environment based on the Simulink MATLAB program for the development of custom hardware IP Cores and FPGA (Field Programmable Gate Array) programming in Xilinx de- vices. In this platform, designer-friendly and run-time efficient VHDL (VHSIC (Very High-Speed Integrated Circuit)

Hard- ware Description Language) and Simulink blocks libraries are provided. The Simulink-XSG comprises two main components: ”black box” Simulink block parameters and subsystem blocks from the Xilinx System Generator library that provide additional ’smart’ properties for automatic hardware/software co-simulation acceleration [25][26].

The Xilinx System Generator for DSP and HDL integrates seamlessly with the Xilinx Vivado Design Suite. The Xilinx Vivado Design Suite is built for complex designs and supports the latest nodes. It offers system and device level design and joins hardware and software support. It includes toolsets for synthesis, place, route, and FPGA configuration management. Specialized tools like hardware co-simulation, partial reconfiguration, and hardware design languages like Verilog, VHDL, or System Verilog are also integrated [27].

Fig.2, shows hardware flow design using Xilinx system generator, this work implemented using VIVADO/XSG environment VIVADO 2020.2 associated with MATLAB/SIMULINK 2020 a.

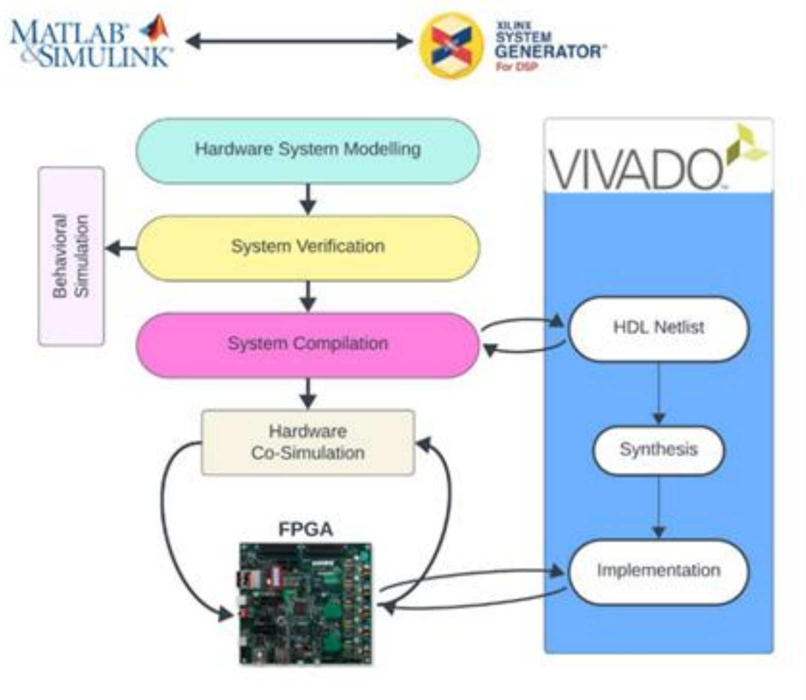


Figure 2. Hardware Flow Design Using Xilinx System Generator.

C. FPGA-ZYNQ implementations of the proposed method

FPGA ZYNQ702 evaluation board provide real time implementation of key generation system. FPGA is an abbreviation of field programmable gate array provide effective level of parallelism and pipelining capabilities; additional property of FPGA over ASIC is the flexibility of recon- figuration. In this paper, Xilinx system generator (XSG) tool employed FPGA implementation to validate the results performance of the proposed system.

4. Results and Discussion

A. Hardware Architecture of proposed system

This section provides detailed explanation about hardware implementation using XSG for proposed key generation system in Fig.3 as following:

input (1): data input/output unsigned fix 8-bit integer rep- resent pixel value of red channel of image.

input (2): data input/output unsigned fix 8-bit integer rep- resent pixel value of green channel of image.

input (3): data input/output unsigned fix 8-bit integer rep- resent pixel value of green channel of image.

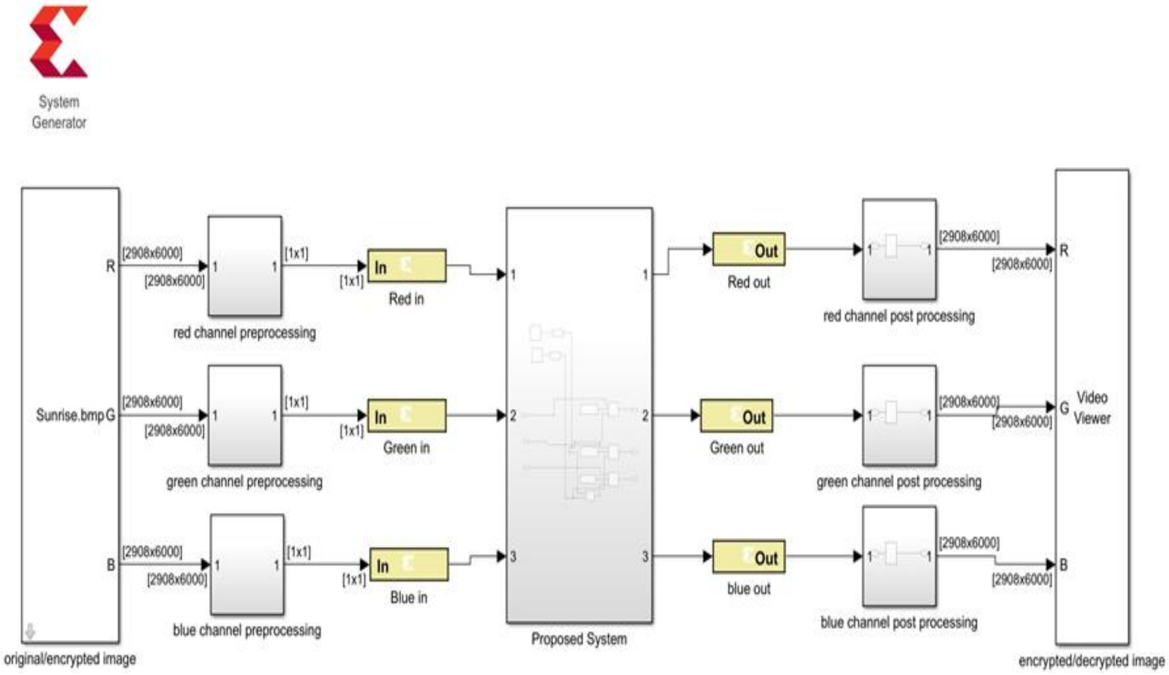

System Generator
Sunrise.bmp
R
G
B
original/encrypted image
[2908x6000]
[1x1]
red channel preprocessing
green channel preprocessing
blue channel preprocessing
In
Red in
Green in
Blue in
Proposed System
Out
Red out
Green out
blue out
red channel post processing
green channel post processing
blue channel post processing
Video Viewer
encrypted/decrypted image

Figure 3. Proposed Hardware System.

Channel pre-processing: this block convert 2-D array to one pixel scalar consists of two parts shown in figure firstly convert 2-D to 1-D array, secondly convert 1-D to scalar shown in Fig.4.

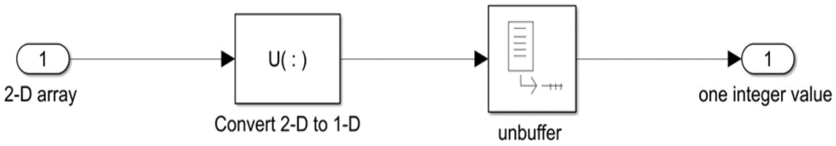


Figure 4. Image Pre-processing.

Channel post processing: this block gets back array form of the image from input pixels consists of three parts shown in Fig.5 firstly buffer involve gathering of the pixel in buffer, secondly reshape involve convert the elements stored in buffer to 2-D array form, thirdly, uint8 block for validate the data to unsigned integer eight.

Channel pre-processing: this block convert 2-D array to one pixel scalar consists of two parts shown in figure firstly convert 2-D to 1-D array, secondly convert 1-D to scalar shown in Fig.5.

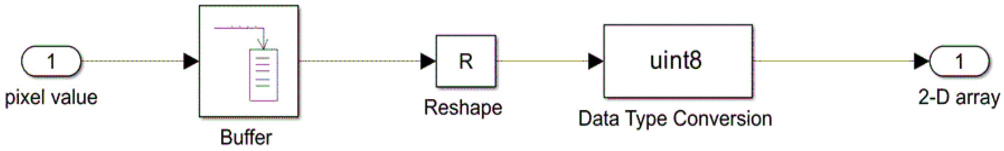


Figure 5. Image Post-processing.

Channel post processing: this block gets back array form of the image from input pixels consists of three parts shown in fig. 5 firstly buffer involve gathering of the pixel in buffer, secondly reshape involve convert the elements stored in buffer to 2-D array form, thirdly, uint8 block for validate the data to unsigned integer eight.

Red-in, green-in, blue-in: yellow blocks for input image data from Simulink to FPGA. -red-out, green-out, blue-out: yellow blocks for forward image data from FPGA to Simulink. The next figure Fig.6 shows the general architecture of XSG blocks of proposed system.

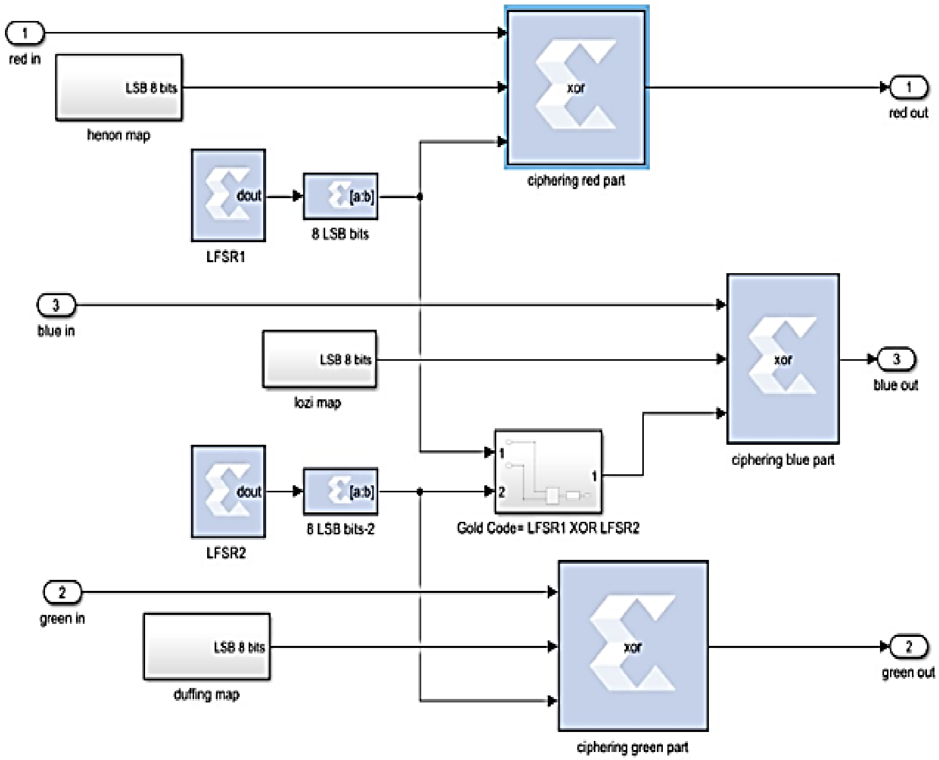


Figure 6. XSG Blocks of Proposed System.

LFSR stands for linear feedback shift register use for generating pseudo random number generator (PN). Red-in XORed with key generated by (Henon map XOR LFSR1). Green-in XORed with key generated by (Duffing map XOR LFSR2). Blue-in XORed with key generated by (Lozi map XOR Gold- code). Gold code can be generated by two normal pseudo random number generator shows in Fig.7.

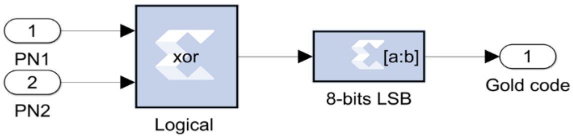


Figure 7. Gold Code Generation in XSG.

5. Chaotic Maps Utilized for Proposed Approach

A. Henon map

The Henon map is one of the most understandable two-input systems and it lies in the third plane of the biological body. In the aforementioned equations (1,2) are given as the control parameters, which are selected for our aim, for example, encryption as cross iteration as 1.4 and second iteration 0.3 [28-29].

$$x(n + 1) = 1 + a(x(n))2 + y(n). \quad (1)$$

$$y(n + 1) = bx(n). \quad (2)$$

Algorithm illustrates in Fig.8 based on finite precision discrete chaotic maps is proposed for generation of pseudo-random signals. The output of the presented chaos generator not only achieves faster performance, but also achieves higher quality of chaotic properties. This algorithm is based on the piecewise linear dynamics of the standard 2D map [30] [31].

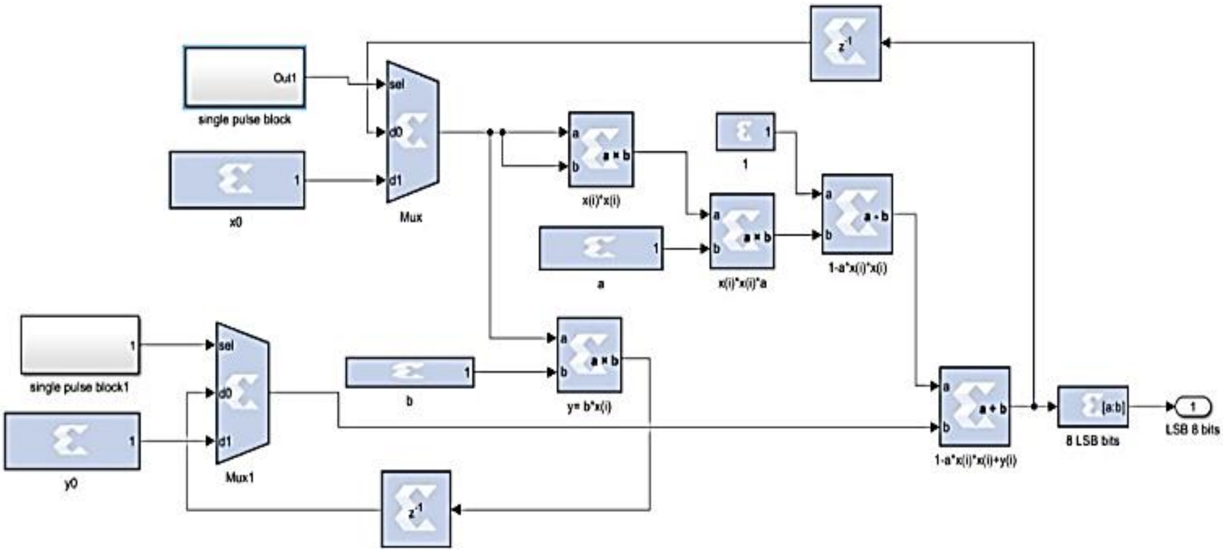


Figure 8. Hardware Implementation of Henon.

B. Duffing map

The Duffing map guarantees large Lyapunov exponent values, and it is the simplest map in one that can do so. Hence, it has been widely utilized in information security. In the 1980s, Miller utilized the Duffing map to develop a symmetric-key block cipher algorithm [32].

Fig.9 illustrates using a Duffing map for a block cipher, it is not that the plaintext sufficiently mixed results in the ciphertext. This is because the Duffing map does not support good performance for any permutation in a limited sequence space-time, especially for large plaintext [33]. Duffing map could be show in these equations (3,4):

$$x(n + 1) = y(n). \qquad (3)$$

$$y(n + 1) = -bx(n) + ay(n) - (y(n))3. \qquad (4)$$

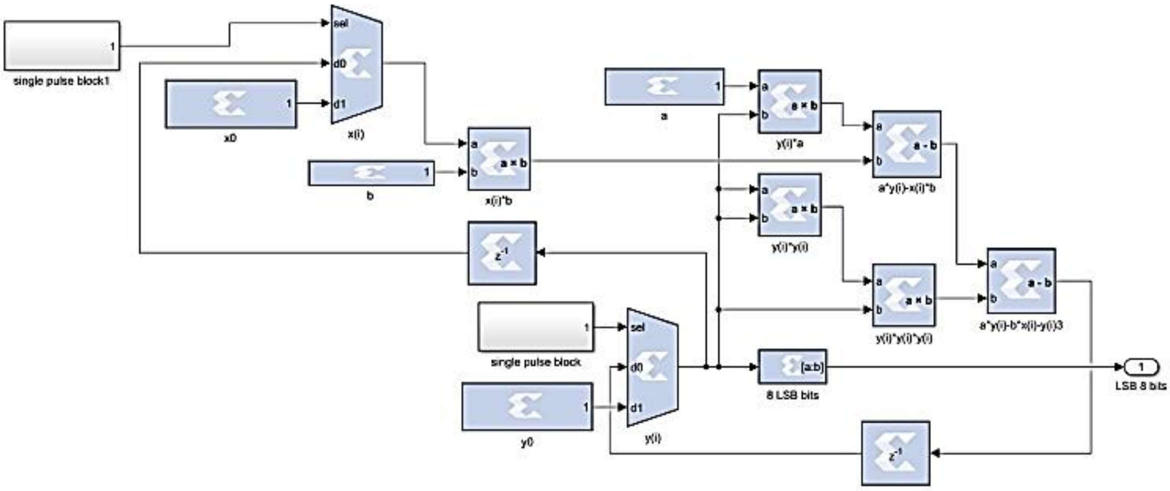


Figure 9. Hardware Implementation of Duffing.

C. Lozi map

One of the numerous sophisticated chaos systems is the Lozi chaos map. The Lozi map is a two-dimensional discrete dynamical map. There has been a lot of interest in certain newly developed maps of chaos that may be practically used [34].

The mathematical format of the Lozi Map showed in equations below portrays the change of a location because of two sequential steps; this includes an ego-centric conversion with regard to an origin and to an additional output. Algorithm illustrates in Fig.10 The unique properties of the Lozi's chaos dynamical systems described by equations (5,6) make it suitable for a variety of applications in different fields such as control theory, pseudorandom number generators, digital communication, physics, and cryptography [35].

$$x(n + 1) = 1 + y(n) - a|x(n)|. \qquad (5)$$

$$y(n + 1) = bx(n). \qquad (6)$$

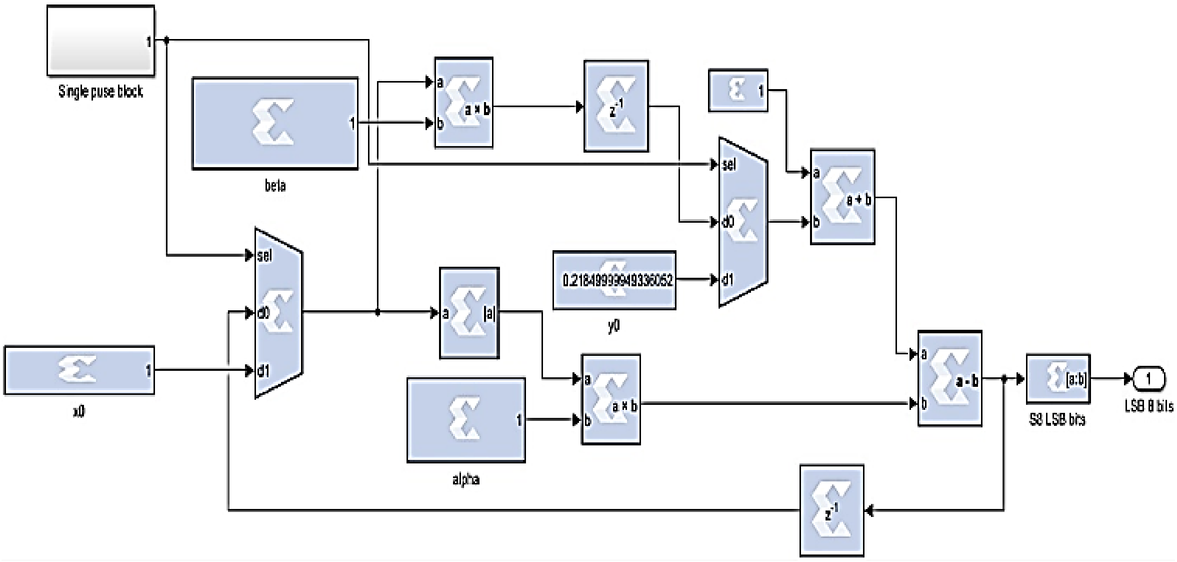


Figure 10. Hardware Implementation of Lozi.

The single pulse as shown in Fig.11 consists of three components: register initial '1', logical XOR block, assert block for sampling time, used to control the selector of multiplexer giving an initial value 1 to take the value of initial x0 then after the first clock the feedback value goes through.

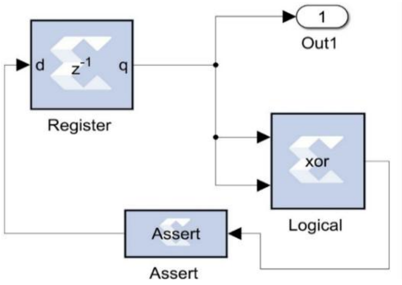


Figure 11. Hardware Implementation of Single Pulse Block.

6. Hardware Synthesize and Implementation

This step involved compilation of the design to HDL Netlist file to get ready for using Xilinx VIVADO software tool. The ZYNQ702 FPGA evaluation board operate at 667 MHZ as a maximum frequency, after synthesize process it has been clear the suitable period to meet timing paths requirements was 6 ns resulting worse negative slacks (WNS) 2.129 ns then the maxi- mum frequency for design can be calculated from the equation (7).

$$fmax = 1/\,(T - WNS). \qquad (7)$$

The throughput (the number of processing bits per second) of the hardware system considered an important measurement that can be calculated from the equation (8).

$$Throughput = (N \times fmax)/latency. \qquad (8)$$

The maximum frequency was 258.33 MHZ with high through- put near 6199 Mb/s.

Table.1 declare estimated of ZYNQ702 FPGA board silicon area of the proposed key generation system as the following:

A. *LUT : represent utilization of Look-Up Tables.*

B. *LUTRAM: represent utilization of Look-Up Table RAMS.*

C. *FF : represent utilization of Flip-Flops.*

D. *BRAM: represent utilization of Block RAMs.*

E. *DSP : represent utilization of Digital Signal Processing blocks.*

F. *IO : represent utilization of Input/output buffers.*

G. *BUFG : represent utilization of Global Buffers*

H. *MMCM: represent utilization of Mixed-Mode Clock Manager.*

Table 1. Logic Area Utilization.

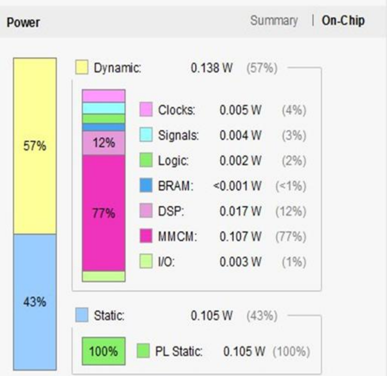
Resource	Utilization	Available	Percentage
LUT	888,	53200	167
LUTRAM	1	17400	0.01
FF	1314	106400	1.23
BRAM	2	140	1.43
DSP	36	220	16.36
IO	2	200	1
BUFG	4	32	12.5
MMCM	1	4	25

Table.2 present comparative study of present work with ref. [38] declaring that proposed work the fastest in terms of frequency and throughput.

Table 2. Speed Comparison with Recent Resource.

Work	FPGA	F(MHZ)	Throughput (Mb/s)
Proposed	ZYNQ702	258.33	6199
[38]	ZYNQ702	142.8	3408

Fig.12 presents the power estimation for each of component mentioned in table 1, the total static power is 0.105 watt while the dynamic power is 0.138 watt.



Dynamic: 0.138 W (57%)

Clocks: 0.005 W (4%)
Signals: 0.004 W (3%)
Logic: 0.002 W (2%)
BRAM: <0.001 W (<1%)
DSP: 0.017 W (12%)
MMCM: 0.107 W (77%)
I/O: 0.003 W (1%)

Static: 0.105 W (43%)

PL Static: 0.105 W (100%)

Figure 12. Power Utilization.

7. Key Analysis Using NIST Test Suit

NIST 800-22 [39]. Involved 15 tests used to examine randomness distribution for the generated key streams they have statistical p-value calculated by particular methods. Assume α=0.01, as acritical probability the p-value for each test must be > α to passed the randomness test otherwise said the test failed, if the p-value of any test was 1 then the sequence considered excellent random, if the, if the p-value of any test was 0 then the sequence considered non-random. Table 3 show 12 tests all passed, that is indication that the streams are unpredictable, independent distribution statistically, the table also contain comparison with [40] to show the strength of the proposed system.

Table 3. NIST Test of Generated Keys

Test name	p-value (proposed)	Mohamed et al. (2023)	Comparison
Frequency (Monobit) Test	0.9251	0.8755390	Proposed
Frequency Test within a Block	0.8494	0.3924558	Proposed
Test for the Longest Run of Ones in a Block	0.4874	0. 6371194	Mohamed
Binary Matrix Rank Test	0.3905	0.4372742	Mohamed
Non-overlapping Template Matching Test	0.3023	0. 1463590	Proposed
Overlapping Template Matching Test	0.9919	0.5449921	Proposed
Linear Complexity Test	0.6862	0.1922722	Proposed
Serial Test	0.8586	0.0713232	Proposed
Approximate Entropy Test	0.56749	0.0125474	Proposed
Cumulative Sums (Cusum) Test (forward)	0.63747	0. 6371194	Proposed
Cumulative Sums (Cusum) Test (reverse)	0.8626	0. 6371194	Proposed
Runstest	0.9486	0. 6371194	Proposed

8. Conclusion

The paper presented a detailed novel ZYNQ-FPGA implementation of symmetric keys generation based on multi-dimensional chaotic maps. FPGA design accomplished using Xilinx system generator (XSG) tool to build three chaotic maps: Henon, Duffing, Lozi with 32 bits fixed point numbers then design two PN sequence generator in the same tool (XSG) PN1, PN2 then generating gold code by XORing them. The work employed the 8 LSB bits of each because they are the most changeable bits in computations. The algorithm started inside Simulink with first secret key resulted with XORing PN1 with Henon map then second secret key resulted with XORing PN2 with Duffing map then third secret key stream resulted with XORing gold code with Lozi map, the gold code is the a sequency By utilizing ZYNQ702 evaluation board to meet timing requirements of the design the work achieved high frequency of 258.33 MHZ and high throughput of 6199 Mb/s. two types of analysis involved: statistical analysis and key analysis. Both analyses compared with recent work proved an effective security performance of the proposed system. The authors will search for utilizing the generated keys for digital image encrypting/decrypting.

References

[1] Ö. Aslan, S. S. Aktuğ, M. Ozkan-Okay, A. A. Yilmaz, and E. Akin, “A comprehensive review of cyber security vulnerabilities, threats, attacks, and solutions,” *Electronics*, vol. 12, no. 6, p. 1333, 2023.

[2] O. Evsutin, A. Melman, and R. Meshcheryakov, “Digital steganography and watermarking for digital images: A review of current research directions,” *IEEE Access*, vol. 8, pp. 166589–166611, 2020.

[3] Alanezi et al., “Securing Digital Images through Simple Permutation-Substitution Mechanism in Cloud-Based

Smart City Environment," *Security and Communication Networks*, vol. 2021, no. 1, p. 6615512, 2021.

[4] T. Suhail and H. G. Ayoub, "A new method for hiding a secret file in several WAV files depends on circular secret key," *Egyptian Informatics Journal*, vol. 23, no. 4, pp. 33–43, 2022.

[5] U. Zia et al., "Survey on image encryption techniques using chaotic maps in spatial, transform, and spatiotemporal domains," *International Journal of Information Security*, vol. 21, no. 4, pp. 917–935, 2022.

[6] S. A. Baker, "Preserving Big Data Privacy in Cloud Environments Based on Homomorphic Encryption and Distributed clustering," *NTU Journal of Engineering and Technology*, vol. 3, no. 1, Mar. 2024, doi: 10.56286/ntujet.v3i1.861.

[7] N. Jumaah, "Embedded Reversibility Data in an Encrypted Photograph: A Case Study," *International Journal of Computational and Electronic Aspects in Engineering*, vol. 5, no. 3, pp. 73–79, Sep. 2024.

[8] S. I. Hamad, "Use of Chaos in Key Schedules for A Symmetrical Encryption Algorithm without Data Loss," *International Journal of Computational and Electronic Aspects in Engineering*, vol. 5, no. 4, pp. 194–202, Dec. 2024.

[9] W. A. Hashim, "Quantum Cryptography-Enabled Cloud Security (QCECS) Framework," *International Journal of Computational and Electronic Aspects in Engineering*, vol. 5, no. 4, pp. 214–228, Dec. 2024.

[10] A. H. Kjwan, "Adaptive Covert Communication Framework for 6G Networks Integrating Quantum Cryptography and AI-Augmented Physical Layer Security," *International Journal of Computational and Electronic Aspects in Engineering*, vol. 5, no. 4, pp. 203–213, Dec. 2024.

[11] Z. S. Karam, R. H. Ali, and B. O. Al-Nashy, "Exploring Emerging Strategies for Countering Computer Malware Attacks: A Comprehensive Survey of Tools and Techniques," *International Journal of Computational and Electronic Aspects in Engineering*, vol. 4, no. 2, pp. 25–37, Jun. 2023, doi: 10.26706/ijceae.4.2.20239750.

[12] Masood, J. Ahmad, S. A. Shah, S. S. Jamal, and I. Hussain, "A novel hybrid secure image encryption based on julia set of fractals and 3D Lorenz chaotic map," *Entropy*, vol. 22, no. 3, p. 274, 2020.

[13] A. Salih, Z. A. Abdulrazaq, and H. G. Ayoub, "Design and Enhancing Security Performance of Image Cryptography System Based on Fixed Point Chaotic Maps Stream Ciphers in FPGA," *Baghdad Science Journal*, vol. 21, no. 5 (SI), p. 1754, 2024.

[14] S. Parikibandla and S. Alluri, "Low area field-programmable gate array implementation of PRESENT image encryption with key rotation and substitution," *ETRI Journal*, vol. 43, no. 6, pp. 1113–1129, 2021.

[15] N. Qureshi, S. Qayyum, M. N. Ul Islam, and G. Jeon, "A secure data parallel processing based embedded system for internet of things computer vision using field programmable gate array devices," *International Journal of Circuit Theory and Applications*, vol. 49, no. 5, pp. 1450–1469, 2021.

[16] C.-H. Yang and Y.-S. Chien, "FPGA implementation and design of a hybrid chaos-AES color image encryption algorithm," *Symmetry*, vol. 12, no. 2, p. 189, 2020.

[17] W. Burger and M. J. Burge, *Digital Image Processing: An Algorithmic Introduction*, Springer Nature, 2022.

[18] Ding, K. Ma, S. Wang, and E. P. Simoncelli, "Comparison of full-reference image quality models for optimization of image processing systems," *International Journal of Computer Vision*, vol. 129, no. 4, pp. 1258–1281, 2021.

[19] Salvi, U. R. Acharya, F. Molinari, and K. M. Meiburger, "The impact of pre-and post-image processing techniques on deep learning frameworks: A comprehensive review for digital pathology image analysis," *Computers in Biology and Medicine*, vol. 128, p. 104129, 2021.

[20] Toktas and U. Erkan, "2D fully chaotic map for image encryption constructed through a quadruple-objective optimization via artificial bee colony algorithm," *Neural Computing and Applications*, pp. 1–25, 2022.

[21] I. Žeger, S. Grgic, J. Vuković, and G. Šišul, "Grayscale image colorization methods: Overview and evaluation," *IEEE Access*, vol. 9, pp. 113326–113346, 2021.

[22] X. Wang, S. Lin, and Y. Li, "Bit-level image encryption algorithm based on BP neural network and gray code," *Multimedia Tools and Applications*, vol. 80, pp. 11655–11670, 2021.

[23] Ntnguyen, et al., [Reference details not found in the uploaded file.].

[24] B. Al-Roithy and A. Gutub, "Trustworthy image security via involving binary and chaotic gravitational searching within PRNG selections," *International Journal of Computer Science and Network Security*, vol. 20, no. 12, pp. 167–176, 2020.

[25] T. Etem and T. Kaya, "A novel true random bit generator design for image encryption," *Physica A: Statistical Mechanics and its Applications*, vol. 540, p. 122750, 2020.

[26] A. A. Yazdeen, S. R. Zeebaree, M. M. Sadeeq, S. F. Kak, O. M. Ahmed, and R. R. Zebari, "FPGA implementations for data encryption and decryption via concurrent and parallel computation: A review," *Qubahan Academic Journal*, vol. 1, no. 2, pp. 8–16, 2021.

[27] B. M. Krishna, G. R. Chowdary, C. Santhosh, S. Kalam, and K. Naidu, "Implementation of Xilinx system generator based image processing algorithms through FPGA," *AIP Conference Proceedings*, 2024.

[28] F. A. BOUAZA Youcef, "Image filtering design and implementation based on Xilinx System Generator with Hardware Co-Simulation and VHDL with FPGA IP Core Generator," 2023.

[29] A. Alhomoud, "Real Time FPGA Implementation of a High Speed for Video Encryption and Decryption System with High Level Synthesis Tools," *International Journal of Advanced Computer Science & Applications*, vol. 15,

no. 1, 2024.

[30] S. V. Gonchenko, K. A. Safonov, and N. G. Zelentsov, “Antisymmetric Diffeomorphisms and Bifurcations of a Double Conservative Hénon Map,” *Regular and Chaotic Dynamics*, vol. 27, no. 6, pp. 647–667, 2022.

[31] A. Kumar and M. Dua, “A novel exponent–sine–cosine chaos map-based multiple-image encryption technique,” *Multimedia Systems*, vol. 30, no. 3, p. 141, 2024.

[32] R. B. Naik and U. Singh, “A review on applications of chaotic maps in pseudo-random number generators and encryption,” *Annals of Data Science*, vol. 11, no. 1, pp. 25–50, 2024.

[33] X. Chen et al., “Pseudorandom Number Generator Based on Three Kinds of Four-Wing Memristive Hyperchaotic System and Its Application in Image Encryption,” *Complexity*, vol. 2020, no. 1, p. 8274685, 2020.

[34] S. Kanwal, S. Inam, S. Quddus, and F. Hajjej, “Research on color image encryption approach based on chaotic Du3ffing map,” *Physica Scripta*, vol. 98, no. 12, p. 125252, 2023.

[35] C. Li, G. Luo, and C. Li, “A parallel image encryption algorithm based on chaotic Duffing oscillators,” *Multimedia Tools and Applications*, vol. 77, pp. 19193–19208, 2018.

[36] S. Zhang, H. Zhang, and C. Wang, “Dynamical analysis and applications of a novel 2-D hybrid dual-memristor hyperchaotic map with complexity enhancement,” *Nonlinear Dynamics*, vol. 111, no. 16, pp. 15487–15513, 2023.

[37] H. Ning, G. Zhao, Y. Dong, and Y. Ma, “A Novel Two-Dimensional Dynamic Pseudo-Random Coupled Map Lattices System Based on Partitioned Elementary Cellular Automata,” *Applied Sciences*, vol. 12, no. 23, p. 12399, 2022.

[38] M. Gafsi, N. Abbassi, M. A. Hajjaji, J. Malek, and A. Mtibaa, “Xilinx Zynq FPGA for hardware implementation of a chaos-based cryptosystem for real-time image protection,” *Journal of Circuits, Systems and Computers*, vol. 30, no. 11, p. 2150204, 2021.

[39] K. M. Shafi, P. Chawla, A. S. Hegde, R. Gayatri, A. Padhye, and C. Chandrashekar, “Multi-bit quantum random number generator from path-entangled single photons,” *EPJ Quantum Technology*, vol. 10, no. 1, p. 43, 2023.

[40] M. ES-SABRY et al., “Securing images using high dimensional chaotic maps and DNA encoding techniques,” *IEEE Access*, 2023.