View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by UWL Repository

micromachines

MDPI

Article

A Hardware Pseudo-Random Number Generator Using Stochastic Computing and Logistic Map

Junxiu Liu [1], Zhewei Liang [1], Yuling Luo [1,2,*], Lvchen Cao [1], Shunsheng Zhang [1], Yanhu Wang [1] and Su Yang [3]

[1] School of Electronic Engineering, Guangxi Normal University, Guilin 541004, China; j.liu@ieee.org (J.L.); lzw.soldier@gmail.com (Z.L.); 18810775110@163.com (L.C.); shunszhang@163.com (S.Z.); yanhu_huny1010@163.com (Y.W.)

[2] Guangxi Key Lab of Multi-Source Information Mining & Security, Guangxi Normal University, Guilin 541004, China

[3] School of Computing and Engineering, University of West London, London W5 5RF, UK; scott.yang@uwl.ac.uk

* Correspondence: yuling0616@gxnu.edu.cn

Abstract: Recent research showed that the chaotic maps are considered as alternative methods for generating pseudo-random numbers, and various approaches have been proposed for the corresponding hardware implementations. In this work, an efficient hardware pseudo-random number generator (PRNG) is proposed, where the one-dimensional logistic map is optimised by using the perturbation operation which effectively reduces the degradation of digital chaos. By employing stochastic computing, a hardware PRNG is designed with relatively low hardware utilisation. The proposed hardware PRNG is implemented by using a Field Programmable Gate Array device. Results show that the chaotic map achieves good security performance by using the perturbation operations and the generated pseudo-random numbers pass the TestU01 test and the NIST SP 800-22 test. Most importantly, it also saves 89% of hardware resources compared to conventional approaches.

Keywords: stochastic computing; chaos; logistic map; FPGA

check for updates

Citation: Liu, J.; Liang, Z.; Luo, Y.; Cao, L.; Zhang, S.; Wang, Y.; Yang, S. A Hardware Pseudo-Random Number Generator Using Stochastic Computing and Logistic Map. *Micromachines* **2021**, *12*, 31. https://doi.org/10.3390/mi12010031

Received: 28 November 2020
Accepted: 24 December 2020
Published: 30 December 2020

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Pseudo-random numbers have an overwhelming impact on many fields, such as cryptography, digital signatures, image authentication watermarks, and protected communication protocols. The design of pseudo-random number generators (PRNGs) has greatly attracted the attention of many researchers. For the generation of pseudo-random numbers, PRNGs are usually based on the implementation of mathematical algorithms. Popular pseudo-random number generation methods include the mid-square method, linear congruence method, linear and nonlinear feedback shift registers [1]. However, due to the fixed linear structure inside, they are easy to be tracked and predicted, which usually leads to the system being not safe enough. The chaos-based system is highly sensitive to initial conditions and parameters. It has the characteristics of pseudo-randomness and unpredictability, which play an important role in information encryption. Therefore, chaotic systems are considered as an effective way to improve the performance of PRNGs.

Using chaotic systems to design a PRNG was first proposed in [2]. Subsequently, a lot of PRNGs based on chaotic systems were proposed. A method of designing PRNG based on the logistic map is proposed in [3]. Compared to the pseudo-random numbers produced by the congruence method, the random numbers generated by the logistic map is large and aperiodic which appears to have better performance. The PRNG construction model based on the discrete chaotic dynamic system is proposed in [4]. In [5], a PRNG based on the piecewise chaos-based system is proposed. In [6], a unidirectional coupling map lattice composed of logical maps is used to construct a spatiotemporal chaotic system. On this basis, a PRNG based on the spatiotemporal chaos-based map is proposed. In [7],

two PRNGs were constructed using 2D dynamic systems constituted by two symmetrically coupled logical maps. The design of a double chaotic system based on logistic map is proposed in [8], and PRNG is derived from random independent initial conditions. In [9], a PRNG based on the composition of several tent maps with diverse initial parameters is proposed, and it can be further used to develop chaotic stream cyphers. A nonlinear principle is proposed in [10] to generate pseudo-random bit sequences. During the generation process, the parameter of a chaotic map is used to disturb the trajectory of another chaotic map to extend its period length. In [11], a PRNG was proposed based on the three-dimensional Chen chaotic system. In [12], an algorithm for generating multiple pseudo-random sequences using chaotic functions was developed. The initial values of the chaotic system are calculated and indexed by a based-chaos linear congruences function.

All of the above methods are chaos-based PRNGs, and are all realized using software. In addition, many researchers have contributions in hardware implementation of chaos-based PRNGs. In [13], fully digital circuits are used to implement a chaos-based PRNG, and the clock frequency achieves 120 Mhz. In [14], the spatiotemporal chaotic system is digitized, and the highly parallel PRNGs are implemented on the Field Programmable Gate Arrays (FPGAs). In [15], one-dimensional, two-dimensional, and three-dimensional chaotic systems are implemented and verified on the FPGA platform. In [16], a post-processing technique for chaotic mapping in digital systems is proposed, which supports the third-order chaotic system. The chaotic degradation of digital chaotic systems is greatly reduced. In [17], the optimal parameters of the four chaotic maps are analyzed, and the PRNGs are implemented on the FPGA platform using floating point number and fixed-point number, respectively. In [18], the period of digital chaotic mapping is extended by a simple recursive structure and perturbation, and this PRNG is with low-complexity and long-period safety.

Compared with Application-specific Integrated Circuits, FPGAs have the characteristics of high program flexibility and high parallel computing efficiency. Thus, FPGA is usually selected as the platform for digital implementation of PRNG. However, in practical applications, the hardware resources are limited, and PRNG based on chaotic mapping consume more resources, especially the Digital Signal Processor (DSP) resources. Stochastic computing (SC) is based on a form of probability, which optimizes the hardware implementation of PRNG in this paper. Traditionally, the numbers of SC are substituted by the statistical distribution of the stochastic bit stream. The value of traditional number is converted into stochastic bit stream to represent the probability value in the interval [0, 1]. Then a simple gate circuit can be used to perform arithmetic operations of classic calculations. This characteristic allows SC to provide higher fault tolerance and lower hardware resource dissipation, and it can keep almost the same computing performance like conventional computing technology [19]. In this paper, the arithmetic operations of chaotic systems are replaced by the compact hardware of SC to reduce the computational complexity. For the computing characteristics of the digital platform and SC, the logistic map has been changed accordingly as well as its arithmetic expression. The hardware implementation of the computing unit of PRNG is essentially simplified, which maintains low power and resource dissipation. The main contribution of this paper is that the SC method is used to optimize the digital implementation of PRNG, and the performance analysis and hardware resource consumption are given.

The rest of this paper is organized as follows. The background is retrospect in Section 2. The basic mechanism of the proposed chaotic PRBG is presented in Section 3. Section 4 analyses the chaotic properties of the improved logistic map. Section 5 concludes this paper.

2. Background

We review the related works of the logistic map and stochastic computing in this section.

2.1. Logistic Map

The one-dimensional logistic map [20] is a nonlinear discrete system, which is easy to be implemented in hardware due to its simplicity. It is one of the most widely used chaotic maps. Its iterative equation is defined by

$$x_{n+1} = rx_n(1 - x_n), \tag{1}$$

where the range of control parameter r is $0 < r \leq 4$, and the state value for each iteration is $0 < x_n < 4$. The initial value of the iteration is $n = 1, 2, 3, \ldots, x_n$. When the control parameter r changes, the behaviour of the logistic map will be very different. Figure 1 depicts the bifurcation diagram of the logistic map when the initial value is 0.45. It describes the distribution of the iteration values as a function of the parameter r. The abscissa is the parameter r, and the ordinate is the iterative state value x_n. When parameter r increases, the mapping iteration motion will appear in multiple states. When $3.569945672 < r \leq 4$, the values of x_n will be distributed within a certain range, and there is no fixed period. In the meanwhile, the logistic map exhibits a chaotic state.

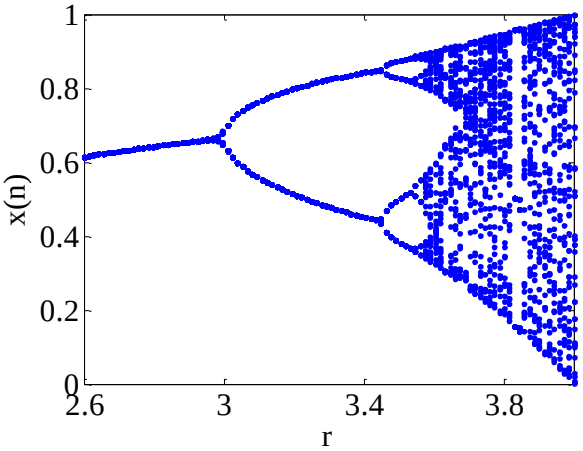

Figure 1. Logistic map.

The chaotic map is highly sensitive to the initial value [21]. The logistic map is iterated 100 times when the control parameter r is 4, and the initial values are $x_0 = 0.45$ and $x_0 = 0.450001$. The results are shown in Figure 2, and it demonstrates the initial condition sensitivity of logistic map. The two chaotic systems are only with extremely small differences in initial values. In the iterative process, the errors are rapidly amplified and the trajectories become irrelevant, which make it extremely difficult to crack the logistic map based encryption system.

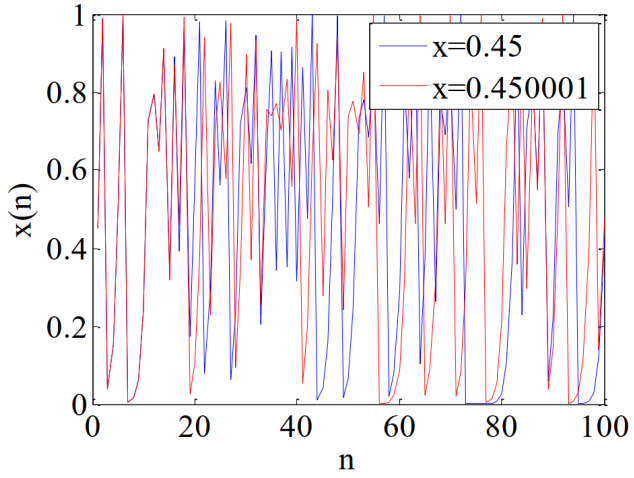

Figure 2. Chaotic sequences.

2.2. Stochastic Computing

The basic rule of SC is that the calculated data are presented in a stochastic bit stream, and then the data are processed in the form of digital probability. There are two representations for converting traditional numbers to a stochastic bit stream: unipolar and bipolar [22]. The number range of unipolar representation is [0, 1] and bipolar is [−1, 1]. In SC, the number represented by the stochastic bit stream is replaced by the probability of occurrence of "1" in this bit stream. For example, three different stochastic bit streams (1,0,0,0), (0,1,0,0) and (0,1,0,0,0,1,0,0) in unipolar. In the representation, the number 0.25 is indicated. For example, there are three different stochastic bit streams (0,0,1,0), (1,0,0,0) and (1,0,0,0,0,0,0,1), and they all represent the number 0.25 in unipolar notation. These bit streams represent the number −0.5 in bipolar. The number represented by the unipolar stochastic bit stream can be expressed by the following formula:

$$X = P(X = 1) = P(X), \tag{2}$$

where X is the value of a traditional number, and $P(X = 1)$ is the probability of the occurrence of "1" in the random bit stream X and is represented by $P(X)$. The number formula represented by the bipolar stochastic bit stream is:

$$X = 2 * P(X = 1) - 1 = 2 * P(X) - 1. \tag{3}$$

The architecture of SC is shown in Figure 3, which usually consists of three main components: stochastic number generator (SNG), stochastic computing element (SCE), and a de-randomizer [23]. The SNG is implemented by using a random number generator and a comparator. On the FPGA platform, we can use a linear feedback shift register (LFSR) instead of the random number generator. SNG is used to convert the binary value into random bit stream. At the same time, a de-randomizer of a binary counter is usually used to decode the output stochastic bit stream into a deterministic binary number [19]. The SCE is an arithmetic operator part of SC, such as multiplier, adder or subtractor. In this section, only unipolar SCE is introduced, because this article only uses a unipolar operation. The circuit structure of SCE is shown in Figure 4.

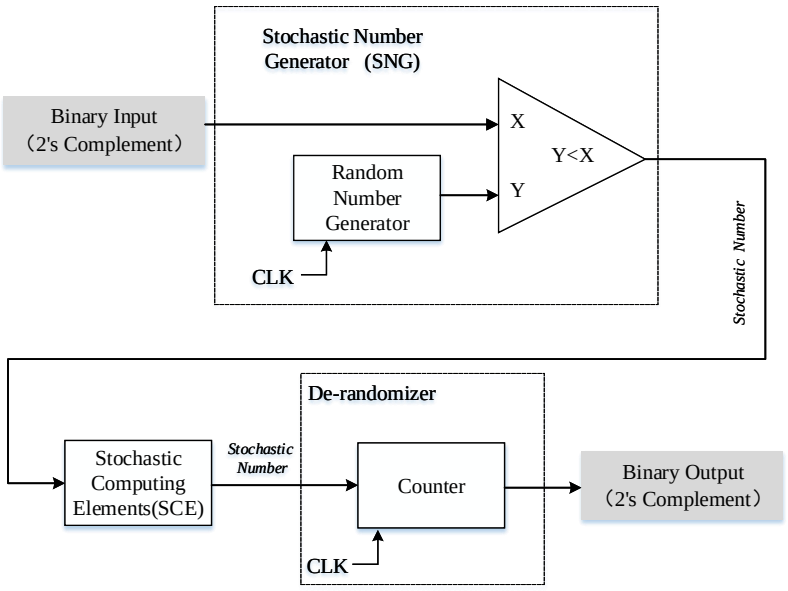

Figure 3. The structure of stochastic computing.

(a). ***Multiplication operation***. In SC, the multiplication of two stochastic bit streams can be realized by a simple gate circuit, which can save a lot of hardware resources in large-scale calculations. Unipolar multiplication only needs to pass the input bit streams

through an AND gate to get the output. It should be noted that the two input bit streams must be guaranteed to be uncorrelated.

(b). ***Addition operation***. In SC, the unipolar bit stream represents values between [0, 1]. The numerical range of the result after the addition should be [0, 2], which is not within the indicated range, so the addition in the random calculation needs to be performed by a special operation, which is called as scaled addition. The scaled addition operation allows the multiplexer to scale the output to the normal range.

(c). ***Subtraction operation***. Its circuit structure is almost the same as an addition operation, where only a NOT gate is used between the second bitstream and the selector.

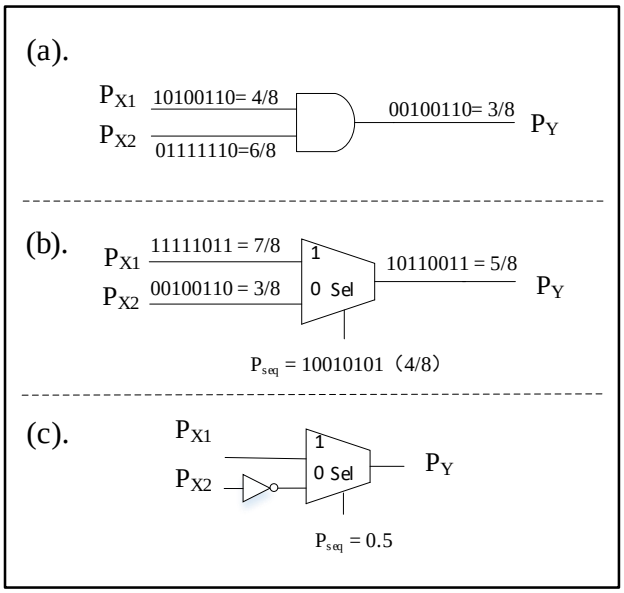

Figure 4. The circuit structure of SCE. (**a**) is a multiplier, (**b**) is an adder, and (**c**) is a subtractor.

3. Enhanced Digital Logistic Map and Hardware Implementation

In this section, an enhanced logistic map is introduced, and it is used to reduce the chaotic degradation caused by digital implementation. Then the mathematical expression of the logistic map is transformed to fit the characteristics of the SC. Finally, the hardware implementation of PRNG is given.

3.1. Enhanced Digital Logistic Map

When the logistic map is in real continuous field, a non-periodic unpredictable sequence can be generated after giving the initial value x_0 and the control parameter r, and $r \in (3.569945672, 4]$. The classical SC has a calculation range of [0, 1]. For a logistic map, only the control parameter r is not within this range, which cannot be represented by SC. Thus, we change the mathematical expression of logistic, and it is shown as

$$\begin{aligned} x_{n+1} &= rx_n(1-x_n) \\ &= (4-d)x_n(1-x_n) \\ &= 4x_n(1-x_n) - dx_n(1-x_n), \end{aligned} \tag{4}$$

where $r = 4 - d$, then $d = 4 - r \in [0, 0.430054328)$. In this manner, a new control parameter can be denoted as d, and all data can be converted into the form of SC.

However, due to the limited precision of a digital system, the chaotic system will generate quantization error in the digital realization process, and the pseudo randomness of the chaotic system is degraded [24]. If the calculation precision is a L-bit binary number, the data space is 2^L. The number of generated iteration values can only be 2^L, which is a finite precision due to the digitization. The problems of short period and high autocorrelation

will reduce the performance of PRNG. Thus, if only using chaotic systems, the performance of PRNG cannot meet the requirements of practical applications.

There are five main ways to promote the degradation of chaotic maps on digital systems: (a) using higher precision [25]. This kind of scheme can prevent the degradation to a certain extent, but the computation costs will grow geometrically, which will affect the speed and consume much more resources when the chaotic system is implemented by hardware. (b) Cascading multiple chaotic systems [26]. This scheme can efficiently extend the period of the digital chaos-based system but may result in poor data distribution. (c) Multiple chaotic system switching [27]. It is often difficult to design an effective switching strategy between multiple chaotic systems. (d) Error compensation method [28]. This scheme can effectively improve the performance of digital chaotic systems, but it is not easy to be extended to high-dimensional chaotic systems. (e) Disturbance methodology [29]. This methodology can prevent the degradation of digital chaotic systems. Disturbance objects including input, output, and system parameters. If the perturbation algorithm is well designed, good performance can be obtained. In this paper, the perturbation methodology is adopted, and we select to disturb the parameter d. The block diagram of the system is shown in Figure 5. The initial value of the parameter d is set to 0. The d in the subsequent iteration is determined by the output of the previous iteration. The output is disturbed to control the parameter d by dividing, and d is controlled within the corresponding range.

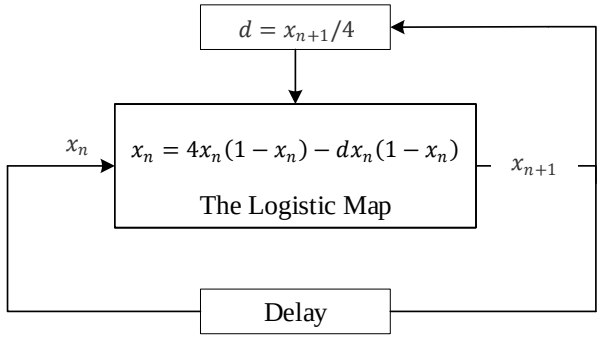

Figure 5. The block diagram of the enhanced logistic map.

3.2. Hardware Implementation of PRNG

Considering the complexity of hardware implementation, SC is used in this paper to replace the arithmetic unit of traditional digital circuits. In the SC, it is first necessary to convert the fixed-point fraction to the stochastic number through SNG. The structure of the SNG is shown in Figure 6. Figure 6a shows the structure of a conventional SNG. x_n is a fixed-point number, and it is compared with a LFSR which has the same number of bits. If the value of the LFSR is greater than x_n, the value of the stochastic sequence at the current time is 1, otherwise, it is 0. Finally, a complete stochastic number is generated. Figure 6b shows the improved SNG structure. When the rising edge of $CLK2$ is detected, LFSR generates a stochastic number. The seed of the LFSR is set to be variable. After the LFSR runs for one cycle, the $CLK1$ becomes high, and value of the seed is increased, $seed = seed + 1$. Even if the chaotic system generates the same iteration value, the stochastic number of SNG conversion will not be the same. The above operation acts as a minor disturbance to prevent chaotic systems from falling into a short cycle.

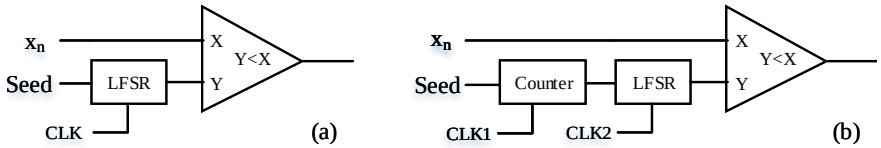

Figure 6. The circuit structure of SNG. (**a**) The conventional SNG; (**b**) The improved SNG.

The hardware structure of the enhanced logistic map is shown in Figure 7. The implementation accuracy is determined by the period length of the LFSR. In Figure 7, a NOT gate is added between SNG_x1 and input x_n, and the resulting stochastic number represents $1 - x_n$. SNG_x produces a stochastic number x_n. Counter represents a decoder in SC that converts a stochastic number into a fixed-point fractional form. The function of Shifter1 is to shift the value of $(1 - x_n) * x_n$ to the left by two bits. Then, the final output value x_{n+1} is obtained by subtracting the output value $u * x_n * \times(1 - x_n)$ of Shifter1 from the output value $4 * x_n * \times(1 - x_n)$ of counter1. The function of Shifter2 is to right shift x_{n+1} by two bits. The output value is in the range of (0, 0.25), and it will be used as the control parameter d for the next iteration.

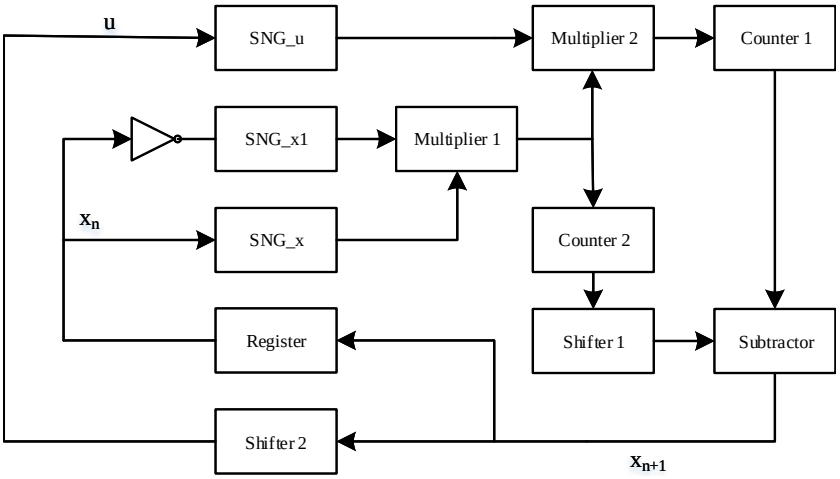

Figure 7. The hardware structure of the PRNG based on the enhanced logistic map.

4. Performance Analysis

4.1. Performance Improvement of Digital Chaotic System

If the precision is limited when implementing a chaotic system, the short cycle phenomenon will occur. The data precision and pseudo-random number generation rate in SC depends on the length of the stochastic sequence, while the length of the stochastic sequence depends on the period length of the LFSR [22]. Specifically, the longer the sequence length of the LFSR, the higher the numerical accuracy and more hardware resources it will consume. Therefore, to achieve a trade-off between the numeric accuracy and hardware resource consumption, the 16-bit LFSR is used to generate random numbers, and the working frequency of proposed system is 100 MHz, which represents the pseudo-random number generation rate is 1.5 KHz, and the effective precision of the fixed-point number is 16 bits. As shown in Figure 8a, the motion trace of the original logical map enters a loop after several iterations. On the contrary, the enhanced logistic map avoids the short period, and the improved system exhibits better randomness, as shown in Figure 8b.

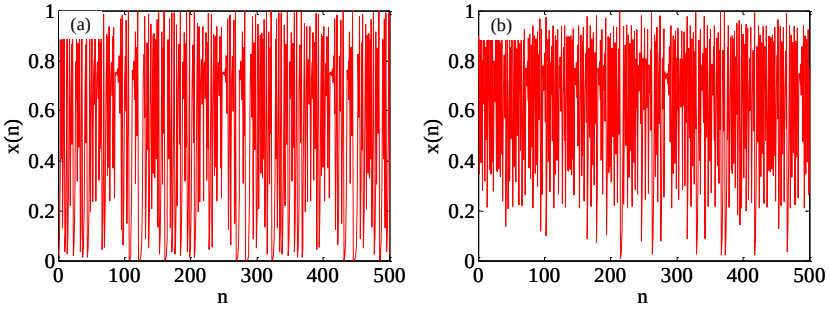

Figure 8. Iterative values produced by the logistic map and the proposed method based on digital implementation. (**a**) The original logistic map; (**b**) The enhanced logistic map.

4.2. Initial Value Sensitivity

Initial value sensitivity is an indispensable indicator of PRNGs. After several iterations, the motion trajectories of the chaotic system will be very different if there are extremely small changes in initial values. The control parameter $d = 0$ in the enhanced logistic map is kept unchanged, and the initial values of the system are taken as $x_0 = 0.25$ and $x_0 = 0.250015$, respectively. The changes of the two sequences after 100 times of iterations are as shown in Figure 9a. The trajectories of the chaotic system have a huge difference after several iterations. Similarly, when the initial value $x = 0.25$ is kept unchanged, a slight variation is given to the initial parameters. The generated sequences are shown in Figure 9b. When the system goes through several iterations, the trajectories show a huge difference. The result indicates that the enhanced logistic map has good initial sensitivity.

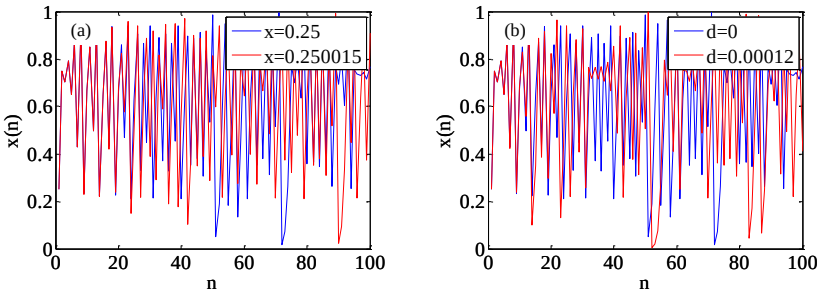

Figure 9. Sequences generated by enhanced logistic map using different initial values. (**a**) Initial values of 0.25 and 0.250015; (**b**) Initial values of 0 and 0.00012.

4.3. Chaotic Attractor

Chaotic attractors usually have a fixed geometric structure, and the structural complexity of the attractor reflects the chaotic degree. In this experiment, 10,000 continuous values were selected for generating the chaotic attractors. Figure 10a shows the chaotic attractor of the original logistic map. Its orbit is in the interval of (0, 1). For the digital realization with limited precision, the logistic map gets stuck in cycles and the iterative sequence cannot achieve full traversal, as shown in Figure 10b. However, after the perturbation, the enhanced logistic map is traversed, as shown in Figure 10c. This chaotic attractor is not a strict curve, but a thicker curve oscillates around it. This is better than the original situation, since the orbit is traversal and more complex.

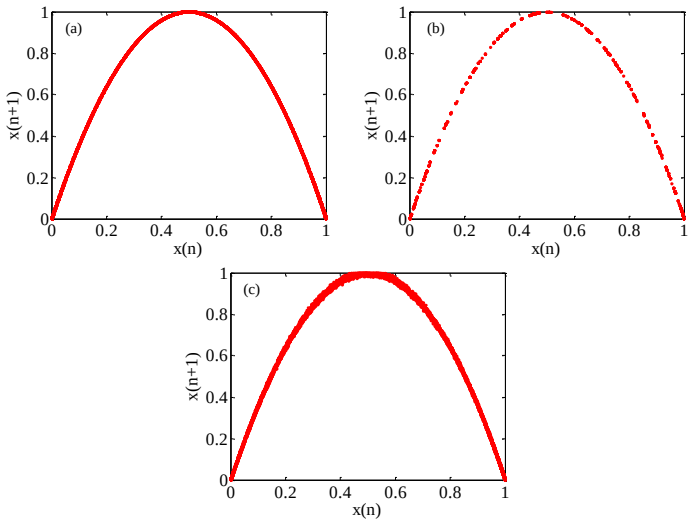

Figure 10. Comparison of chaotic attractors. (**a**) The original logistic map; (**b**) The logistic map of getting stuck in cycles; (**c**) The enhanced logistic map.

4.4. Autocorrelation

Autocorrelation is used to describe the correlation between values at different times in a sequence [30]. Ideally, the autocorrelation of any random sequence is an impulse function. Figure 11a shows the autocorrelation of the output sequence by original logistic map. The correlation between adjacent tracks is strong, which makes the system vulnerable to attack. Figure 11b is the improved autocorrelation, and it shows that the autocorrelation in a sequence is greatly reduced.

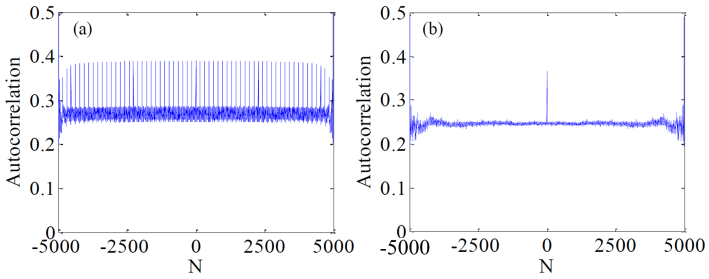

Figure 11. Autocorrelation comparison based on different logistic maps. (**a**) The original logistic map; (**b**) The enhanced logistic map.

4.5. Approximate Entropy

Approximate entropy is often utilized to measure the randomness of binary sequences. According to [31], the approximate entropies of the sequences produced by three systems are obtained and shown in Table 1. The iteration values between the tent map and logistic map are generated on the platform of Matlab 2014a. The parameter of tent map is set as 0.3, and the logistic map parameter is set as 4. The iterative values of the proposed system is calculated by the SC. The approximate entropy based on the proposed system is improved compared with the original logistic map.

Table 1. Approximate entropy comparison based on three chaotic maps.

Chaotic System	Tent Map	Logistic Map	The Proposed System
Approximate entropy	0.5170	0.6506	0.7112

4.6. Histogram of Frequency Distribution

The frequency histogram can reflect the distribution of the chaotic sequence. The frequency distribution of the original logic map is shown in Figure 12a, and the frequency distribution of the enhanced logistic map using the SC is shown in Figure 12b. The frequency distribution of Figure12a is not ideal, since the distribution of values is not uniform, but an obvious improvement is obtained in Figure 12b. Thus, the system proposed in this paper can greatly enhance the ability to resist frequency attacks.

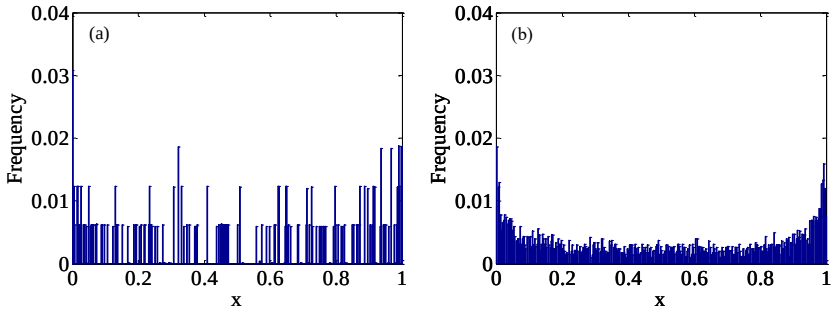

Figure 12. Histogram comparison based on different logistic maps. (**a**) The original logistic map; (**b**) The enhanced logistic map.

4.7. NIST SP 800-22 Analyses

The National Institute of Standards and Technology SP800-22 is a standard suite for measuring the randomness of binary sequences [32]. It consisted of 15 subtests, and these subtests are used to test the same sequence from different aspects. In this test, the data produced by the proposed system are 1,000,000 bits, and the significant level α is set as 0.01. When the P-value is greater than the significant level α, the test is passed, otherwise, the test fails. The analysis results are displayed in Table 2. It can be described that the stochastic bit stream produced by the proposed system has excellent statistical properties.

Table 2. Results of the NIST SP800-22 analyses.

Sub-Tests	*p*-Value	Result
Frequency	0.213309	Pass
Universal	0.554405	Pass
Serial	0.739918	Pass
Rank	0.441376	Pass
FFT	0.902014	Pass
Runs	0.350485	Pass
Longest Run	0.911413	Pass
Block Frequency	0.565945	Pass
Cumulative Sums	0.534146	Pass
Overlapping Template	0.395130	Pass
Non-overlapping Template	0.066882	Pass
Approximate Entropy	0.122325	Pass
Linear Complexity	0.911413	Pass
Random Excursions	0.652391	Pass
Random Excursions Variant	0.381832	Pass

4.8. TestU01 Test

TestU01 [33] is also a commonly used random number test standard. It provides several sets of tests, in which each test contains different subtests. In this experiment, we mainly use the Rabbit and Alphabit tests. The data length is 2^{20} bits. Rabbit test has a total of 38 small tests under this data length, and Alphabit test contains 17 tests. These two tests are mainly used to test hardware implemented PRNGs. Table 3 shows the results of the TestU01 test, and the improved system passes all the tests. Thus, the proposed system has better randomness compared to the original logistic map.

Table 3. Results of the TestU01 test for different PRNGs.

PRNGs	Rabbit	Alphabit
Original logistic map	35/38	15/17
Enhanced logistic map	38/38	17/17

4.9. Area Overhead

The proposed system is performed on the ZedBoard Zynq Evaluation and Development Kit (xc7z020clg484-1) platform, and its working frequency is 100 MHz, and the utilizations of all hardware resource consumption are summarized. The comparison of resource consumption is shown in Table 4, where 373 LUTs and 445 registers are consumed, and the utilization rates are 0.7% and 0.4% of the ZedBoard device, respectively. The PRNG in [34] also uses the logistic map. Compared to it, the consumption of look-up tables (LUTs) is reduced by 89.6%, the consumption of the register is reduced by 62.2%, and the DSP blocks are not used. Results show that the PRNG implemented on the FPGA combined with the SC and the enhanced logistic map occupies the least resources.

Table 4. Comparison of resource consumption based on different PRNGs.

Approaches	Chaotic Generators	LUTs	Registers	DSP Blocks	Frequency (MHz)	Device
[35]	Jerk	716	130	16	50.46	Virtex 4
[36]	Cubic map	387	N/A	24	100	Virtex 6
[34]	Logistic map	3711	1176	N/A	50	Cyclone II
[37]	Loren	3476	N/A	N/A	100	Virtex 6
[38]	Coupled map lattices	2548	2429	32	100	Stratix IV
This work	Logistic map	373	445	0	100	ZedBoard

5. Conclusions

In this paper, a hardware PRNG based on the enhanced logistic map has been presented, and it is optimized by using the technique of SC. Compared with existing approaches, this proposed work reduces 89% of the consumption of hardware resources, especially its implementation does not require DSP blocks which are expensive for FPGA devices. Besides, the pseudo-random numbers generated by the proposed PRNG pass the TestU01 test and the NIST SP 800-22 test. The performance and statistical analysis results demonstrate a high-security performance of the proposed work with relatively low hardware utilizations. Future work will aim to further reduce the required hardware resource for complex digital chaotic system implementations.

Author Contributions: Conceptualization, J.L.; Formal analysis, Z.L. and S.Y.; Funding acquisition, Y.L.; Investigation, Z.L. and Y.L.; Methodology, J.L., Z.L., Y.L., L.C., S.Z., Y.W. and S.Y.; Validation, J.L., L.C., S.Z. and S.Y.; Writing—original draft, Z.L. and Y.L.; Writing—review & editing, J.L., Z.L., Y.L., L.C., S.Z., Y.W. and S.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially supported by the National Natural Science Foundation of China under Grants 61661008 and 61801131, the funding of Overseas 100 Talents Program of Guangxi Higher Education, the Diecai Project of Guangxi Normal University, 2018 Guangxi One Thousand Young and Middle-Aged College and University Backbone Teachers Cultivation Program, and research fund of Guangxi Key Lab of Multi-source Information Mining & Security (19-A-03-02).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Liu, L.; Miao, S.; Hu, H.; Deng, Y. Pseudorandom bit generator based on non-stationary logistic maps. *IET Inf. Secur.* **2016**, *10*, 87–94. [CrossRef]
2. Oishi, S.; Inoue, H. Pseudo-random number generators and chaos. *Trans. Inst. Electron. Commun. Eng. Jpn.* **1982**, *65*, 534–541.
3. Andrecut, M. Logistic map as a random number generator. *Int. J. Mod. Phys. B* **1998**, *12*, 921–930. [CrossRef]
4. Shujun, L.; Xuanqin, M.; Yuanlong, C. Pseudo-random bit generator based on couple chaotic systems and its applications in stream-cipher cryptography. In *Progress in Cryptology—INDOCRYPT 2001, Proceedings of the International Conference on Cryptology in India, Calcutta, India, 10–13 December 2001*; Springer: Berlin, Germany, 2001; pp. 316–329.
5. Meng, F.S.; Yang, C.Z.; An, Z.Y. Chaos-Based Random Number Generators. *J. Comput. Res. Dev.* **2004**, *4*, 1–23.
6. Li, P.; Li, Z.; Halang, W.A.; Chen, G. A novel multiple pseudo random bits generator based on spatiotemporal chaos. *IFAC Proc. Vol.* **2005**, *38*, 1085–1089. [CrossRef]
7. Pellicer-Lostao, C.; López-Ruiz, R. Pseudo-random bit generation based on 2D chaotic maps of logistic type and its applications in chaotic cryptography. In Proceedings of the International Conference on Computational Science and Its Applications, Perugia, Italy, 30 June–3 July 2008; pp. 784–796.
8. Patidar, V.; Sud, K.K.; Pareek, N.K. A pseudo random bit generator based on chaotic logistic map and its statistical testing. *Informatica* **2009**, *33*, 441–452.
9. Cristina, D.A.; Radu, B.; Ciprian, R. A new pseudorandom bit generator using compounded chaotic tent maps. In Proceedings of the International Conference on Communications (COMM), Bucharest, Romania, 21–23 June 2012; pp. 339–342.
10. Wang, X.Y.; Yang, L. Design of pseudo-random bit generator based on chaotic maps. *Int. J. Mod. Phys. B* **2012**, *26*, 1250208–1250217. [CrossRef]
11. Hu, H.; Liu, L.; Ding, N. Pseudorandom sequence generator based on the Chen chaotic system. *Comput. Phys. Commun.* **2013**, *184*, 765–768. [CrossRef]

12. Francois, M.; Grosges, T.; Barchiesi, D.; Erra, R. A new pseudo-random number generator based on two chaotic maps. *Informatica* **2013**, *24*, 181–197. [CrossRef]
13. Yang, H.T.; Huang, J.R.; Chang, T.Y. A chaos-based fully digital 120 MHz pseudo random number generator. In Proceedings of the IEEE Asia-Pacific Conference on Circuits and Systems, Tainan, Taiwan, 6–9 December 2004; pp. 357–360.
14. Mao, Y.; Cao, L.; Liu, W. Design and FPGA implementation of a pseudo-random bit sequence generator using spatiotemporal chaos. In Proceedings of the International Conference on Communications, Circuits and Systems, Guilin, China, 25–28 June 2006; pp. 2114–2118.
15. Mansingka, A.S.; Radwan, A.G.; Salama, K.N. Fully digital 1-D, 2-D and 3-D multiscroll chaos as hardware pseudo random number generators. In Proceedings of the International Midwest Symposium on Circuits and Systems (MWSCAS), Boise, ID, USA, 5–8 August 2012; pp. 1180–1183.
16. Barakat, M.L.; Mansingka, A.S.; Radwan, A.G.; Salama, K.N. Generalized Hardware Post-processing Technique for Chaos-Based Pseudorandom Number Generators. *ETRI J.* **2013**, *35*, 448–458. [CrossRef]
17. de la Fraga, L.G.; Torres-Pérez, E.; Tlelo-Cuautle, E.; Mancillas-López, C. Hardware implementation of pseudo-random number generators based on chaotic maps. *Nonlinear Dyn.* **2017**, *90*, 1661–1670. [CrossRef]
18. Dastgheib, M.A.; Farhang, M. A digital pseudo-random number generator based on sawtooth chaotic map with a guaranteed enhanced period. *Nonlinear Dyn.* **2017**, *89*, 2957–2966. [CrossRef]
19. Brown, B.D.; Card, H.C. Stochastic neural computation. I. Computational elements. *IEEE Trans. Comput.* **2001**, *50*, 891–905. [CrossRef]
20. Boeing, G. Visual analysis of nonlinear dynamical systems: Chaos, fractals, self-similarity and the limits of prediction. *Systems* **2016**, *4*, 37. [CrossRef]
21. Luo, Y.; Ouyang, X.; Liu, J.; Cao, L. An image encryption method based on elliptic curve elgamal encryption and chaotic systems. *IEEE Access* **2019**, *7*, 38507–38522. [CrossRef]
22. Alaghi, A.; Hayes, J.P. Survey of stochastic computing. *ACM Trans. Embed. Comput. Syst. TECS* **2013**, *12*, 1–19. [CrossRef]
23. Wong, M.M.; Wong, D.M.L. Stochastic computing with spiking neural P systems. *J. Univ. Comput. Sci.* **2017**, *7*, 589–602.
24. Ouyang, X.; Luo, Y.; Liu, J.; Liu, Y.; Bi, J.; Qiu, S. Period analysis of chaotic systems under finite precisions. In Proceedings of the International Conference on Systems Engineering (ICSEng), Nevada, LV, USA, 18–20 December 2018; pp. 1–5.
25. Wheeler, D.D.; Matthews, R.A. Supercomputer investigations of a chaotic encryption algorithm. *Cryptologia* **1991**, *15*, 140–152. [CrossRef]
26. Heidari-Bateni, G.; McGillem, C.D. A chaotic direct-sequence spread-spectrum communication system. *IEEE Trans. Commun.* **1994**, *42*, 1524–1527. [CrossRef]
27. Wong, K.W.; Kwok, B.S.H.; Law, W.S. A fast image encryption scheme based on chaotic standard map. *Phys. Lett. A* **2008**, *372*, 2645–2652. [CrossRef]
28. Hu, H.; Xu, Y.; Zhu, Z. A method of improving the properties of digital chaotic system. *Chaos Solitons Fractals* **2008**, *38*, 439–446. [CrossRef]
29. Xing-Yuan, W.; Lin-Lin, W. A new perturbation method to the Tent map and its application. *Chin. Phys. B* **2011**, *20*, 50509–50517.
30. Yoshioka, D.; Tsuneda, A. Design of pseudochaotic maximum length sequences with prescribed autocorrelation obtained from discretized chaos maps. In Proceedings of the International Symposium on Spread Spectrum Techniques and Applications, Bologna, Italy, 25–28 August 2008; pp. 161–165.
31. Pincus, S.M. Approximate entropy as a measure of system complexity. *Proc. Natl. Acad. Sci. USA* **1991**, *88*, 2297–2301. [CrossRef] [PubMed]
32. Bassham, L.E., III; Rukhin, A.L.; Soto, J.; Nechvatal, J.R.; Smid, M.E.; Barker, E.B.; Leigh, S.D.; Levenson, M.; Vangel, M.; Banks, D.L.; et al. *Sp 800-22 rev. 1a. a Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*; National Institute of Standards & Technology: Gaithersburg, MD, USA, 2010.
33. L'Ecuyer, P.; Simard, R. TestU01: AC library for empirical testing of random number generators. *ACM Trans. Math. Softw. TOMS* **2007**, *33*, 1–40. [CrossRef]
34. Hue, T.T.K.; Van Lam, C.; Hoang, T.M.; Al Assad, S. Implementation of secure SPN chaos-based cryptosystem on FPGA. In Proceedings of the International Symposium on Signal Processing and Information Technology (ISSPIT), Ho Chi Minh City, Vietnam, 12–15 December 2012; pp. 129–134.
35. Mansingka, A.S.; Radwan, A.G.; Salama, K.N. Design, implementation and analysis of fully digital 1-D controllable multiscroll chaos. In Proceedings of the ICM 2011 Proceeding, Istanbul, Turkey, 13–14 June 2011; pp. 1–5.
36. Giard, P.; Kaddoum, G.; Gagnon, F.; Thibeault, C. FPGA implementation and evaluation of discrete-time chaotic generators circuits. In Proceedings of the Conference on IEEE Industrial Electronics Society, Montreal, QC, Canada, 25–28 October 2012; pp. 3221–3224.
37. Chung, H.; Miri, A. On the hardware design and implementation of a chaos-based RFID authentication and watermarking scheme. In Proceedings of the International Conference on Information Science, Signal Processing and their Applications (ISSPA), Montreal, QC, Canada, 2–5 July 2012; pp. 460–465.
38. Cao, L.; Luo, Y.; Bi, J.; Qiu, S.; Lu, Z.; Harkin, J.; McDaid, L. An authentication strategy based on spatiotemporal chaos for software copyright protection. *Secur. Commun. Netw.* **2015**, *8*, 4073–4086. [CrossRef]