The Dynamic Analysis of a Novel Reconfigurable Cubic Chaotic Map and Its Application in Finite Field

Chuanfu Wang, Yi Di, Jianyu Tang, Jing Shuai, Yuchen Zhang and Qi Lu *

School of Information and Communication Engineering, Hubei University of Economics, Wuhan 430205, China; 20201039@hbue.edu.cn (C.W.); diyi8710@hbue.edu.cn (Y.D.); tjy@hbue.edu.cn (J.T.); 00001626@hbue.edu.cn (J.S.); 20130095@email.hbue.edu.cn (Y.Z.)

* Correspondence: rickylq2021@hbue.edu.cn

Abstract: Dynamic degradation occurs when chaotic systems are implemented on digital devices, which seriously threatens the security of chaos-based pseudorandom sequence generators. The chaotic degradation shows complex periodic behavior, which is often ignored by designers and seldom analyzed in theory. Not knowing the exact period of the output sequence is the key problem that affects the application of chaos-based pseudorandom sequence generators. In this paper, two cubic chaotic maps are combined, which have symmetry and reconfigurable form in the digital circuit. The dynamic behavior of the cubic chaotic map and the corresponding digital cubic chaotic map are analyzed respectively, and the reasons for the complex period and weak randomness of output sequences are studied. On this basis, the digital cubic chaotic map is optimized, and the complex periodic behavior is improved. In addition, a reconfigurable pseudorandom sequence generator based on the digital cubic chaotic map is constructed from the point of saving consumption of logical resources. Through theoretical and numerical analysis, the pseudorandom sequence generator solves the complex period and weak randomness of the cubic chaotic map after digitization and makes the output sequence have better performance and less resource consumption, which lays the foundation for applying it to the field of secure communication.

Keywords: cubic chaotic map; period; chaotic attractor; reconfigurable

1. Introduction

Chaos is a well-known dynamic behavior in physics. The most famous characteristic of a chaotic system is that the evolution of the system is sensitive to the initial conditions, which is also known as the butterfly effect [1]. The chaotic system is nonlinear, sensitive to initial value, aperiodic, ergodic, and noise-like. These special chaotic behaviors are consistent with “confusion” and “diffusion” in Shannon’s information theory, which provides a theoretical basis for generating pseudorandom sequences via chaotic systems [2–5]. Because the future evolution behavior of the chaotic system is aperiodic and unpredictable, it also is widely used in the field of chaotic secure communication [6–12].

With the development of numerical analysis in finite field, it is found that the finite precision effect will cause the dynamic degradation of a chaotic system in real number field, which makes the cryptographic algorithm based on the chaotic system have weak cryptographic security [13]. In a survey of this phenomenon, Li systematically studied the changing state of the chaotic system from real number field to finite field, established the mathematical model of a chaotic system in finite field, and presented the method of analyzing chaotic performance in digital field [14]. Since then, the degradation characteristics of digital chaotic systems have been studied, and many methods to resist the degradation behavior of chaos have been proposed. Yang studied the mathematical model of Logistic chaotic map in $\mathbb{Z}(2^m)$ field and deduced the maximum dynamic orbit [15]. Souza constructed the mathematical model of Arnold’s cat chaotic map in $\mathbb{Z}(3^m)$ field and constructed a new pseudorandom sequence generator [16]. Miyazaki found that...
the properties of sequences generated by digital chaotic systems largely depend on the inherent truncation method in finite fields [17]. The non-smooth probability distribution function of robust logistic map (RLM) trajectories gives an uneven binary distribution in the randomness test. To overcome this disadvantage in RLM, control of chaos (CoC) is proposed for the smooth probability distribution function of RLM [18]. Li proposed the corresponding state mapping network (SMN) model of the chaotic system in finite field, studied the dynamic properties of logistic chaotic map and tent chaotic map in finite field, and proved the scale-free properties of logistic chaotic map in digital field [19].

With the rapid development of large-scale integrated circuits, the speed and quantity of information collection, analysis, and transmission are greatly improved. The throughput of the pseudorandom sequence is as high as $10^{11}$ bit/s. If the pseudorandom sequence is not repeated within 10 years, the period of the pseudorandom sequence should not be less than $2^{65}$, and the period of pseudorandom sequence with encryption capability is generally greater than $2^{80}$. In earlier times, it was thought that the digital chaotic sequence generated by the digital chaotic system retained the great period behavior of the original chaotic map [20]. However, compared with the chaotic system, Alvarez found that the orbit of the digital chaotic system is degenerated to some extent, leading to the emergence of short period behavior [21]. In order to overcome the complex periodic behavior of the degenerate chaotic system in digital field, Deng proposed a feedback method to enhance the performance of digital chaotic sequence by using the idea of mixing digital and analog chaotic systems, which increases the period of digital chaotic sequence efficiently [22]. Zheng introduced a disturbance source to disturb the digital chaotic system to resist the complex periodic behavior of the output sequence [23]. Chen proposed a dynamical perturbation-feedback mixed control (DPFMC) method based on a novel pseudorandom sequence by combining feedback and perturbation [24]. Based on introducing the feedback term, Wang proposed a general method to enhance the period of the digital chaotic sequence by using control theory and effectively increased the period of digital chaotic sequence in theory [25]. In addition, Lin proposed the construction method of nondegenerate high-dimensional chaotic systems and resisted the short period behavior of the digital chaotic system by increasing the dimension [26]. Guyeux innovatively proposed the idea of constructing the chaotic system in finite field and called the chaotic system in finite field a chaotic iterative system, which overcomes the influence of truncation error effect on period [27]. Wang proposed a high-dimensional chaotic iterative system, which further enhanced the ability to resist the complex periodic behavior of digital chaotic sequences [28]. For the digital chaotic system, Yang transformed the digital field into other special fields and used the properties of different fields to increase the period of the digital chaotic sequence [15]. In the design of a digital circuit, the state space of a finite state machine is always a finite value. Under the finite state space, the orbit of the digital chaotic system will gradually deviate from the theoretical true orbit of the corresponding chaotic system. The method of cascaded chaotic systems is proposed systematically by Zhou, which can effectively improve the period length of the digital chaotic sequence and give a specific application example [29]. Shakiba proposed a cascaded modulation-coupled hyperchaos to further enhance the periodicity of digital chaotic sequences [30].

The above research promotes the research of improving the period of digital chaotic sequence and lays a theoretical foundation for the methods of enhancing the complex period behavior of digital chaotic sequence. In the field of digital information encryption, such as pseudorandom sequence generator, chaotic stream cipher, and chaotic image encryption, the accurate period is an important feature of the security of chaos-based cryptographic modules. Compared with current enhanced methods of the period of digital chaotic sequence, this paper proposes a novel special cubic chaotic map and accurately controls the period of output sequence of the digital cubic chaotic map to comprehensively overcome the weak pseudorandom sequence phenomenon caused by finite precision effect. On this foundation, a reconfigurable pseudorandom sequence generator based on the digital cubic chaotic system is proposed. Since pseudorandom sequence generator is an
important part of the chaotic stream cipher, chaotic image encryption, and other chaotic ciphers, the period of output sequence plays an important role in improving the security of cryptographic modules based on the digital chaotic system.

This paper is organized as follows. Section 2 introduces a special cubic chaotic map in real number field, and its chaotic behavior is analyzed, especially the chaotic attractors. Section 3 discusses the degradation of the cubic chaotic map, and its dynamic behavior is analyzed, especially the attractors in finite field. Section 4 presents a novel reconfigurable pseudorandom sequence generator based on the digital cubic chaotic system and analyzes its period and randomness. Finally, the last section concludes our work.

2. A Special Cubic Chaotic Map

2.1. New Cubic Chaotic Map

For chaos-based cryptosystems, one-dimensional chaotic maps are used widely for their simple algebraic structure, especially linear or quadratic chaotic maps, but the security of linear and quadratic chaotic maps is relatively low. For the high-dimensional chaotic maps, although they have high security, the hardware resource consumption of a cryptosystem based on high-dimensional chaotic map is large. In contrast, the research on the dynamic behaviors of the cubic chaotic system is relatively less, which also is a one-dimensional chaotic map with a simple algebraic structure. In addition, the one-dimensional cubic chaotic map has higher order, stronger nonlinearity, and small hardware resource consumption. Cubic map has been deeply studied, especially when the parameter is greater than zero [31,32],

$$f(x) = ax^3 + (1 - a)x, \ a \in [0, 4].$$ (1)

Considering the tradeoff between resource and security performance, a special form of the cubic chaotic map is proposed in this paper based on the classical cubic chaotic [32]. The combined form of the cubic chaotic map is:

$$x_n+1 = a(bx_n^3 - tx_n), \ n \in \mathbb{N}, \ x_n \in \mathbb{R}.$$ (2)

where parameter $a \in \{-1, 1\}$, $b \in (1, 10)$ and $t \in (1, 3)$.

2.2. Fixed Point Analysis

The stability of fixed points in the discrete iterative system $x_{n+1} = f(x_n)$ with an iterative variable $n$ can reflect certain special evolution behaviors of dynamic system. According to the theory of dynamics, the stability of the fixed point is determined by the derivative of the function $f(x_n)$ corresponding to the discrete iterative map at the fixed points. Let $x_{n+1} = x_n$, Map (2) is changed into:

$$x_n = a(bx_n^3 - tx_n), \ n \in \mathbb{N}, \ x_n \in \mathbb{R}.$$ (3)

For the cubic polynomial equation, its root is easy to obtain because of the lack of constant term.

$$\begin{cases} \ x_n = 0 \\ \ x_n^2 = \frac{tn+1}{mt} \end{cases}$$ (4)

The function corresponding to the discrete iterative Map (2) is:

$$f(x) = a(bx^3 - tx), \ n \in \mathbb{N}, \ x_n \in \mathbb{R}.$$ (5)

Its derivative function is:

$$f^{(1)}(x) = 3abx^2 - ta, \ n \in \mathbb{N}, \ x_n \in \mathbb{R}.$$ (6)
Bringing the fixed points into Equation (6), we can obtain the derivative value at the fixed points:

\[
\begin{align*}
    f^{(1)}(x_n = 0) &= -ta \\
    f^{(1)}(x_n^2 = \frac{ta+1}{ab}) &= 2ta + 3
\end{align*}
\]  

(7)

When parameter \( a \in \{-1, 1\} \), \( b \in (1, 10) \), and \( t \in (1, 3) \), the absolute values of derivatives at fixed points \( x_n^2 = \frac{ta+1}{ab} \) are hard to determine. However, we can find that the absolute value of the derivative at fixed points \( x_n = 0 \) is greater than 1. Therefore, the cubic Map (2) has at least an unstable fixed point, which can make it a chaotic map under certain parameter \( a, b, \) and \( t \). The stability of fixed points affects the whole dynamic behavior of the discrete iterative map. When \( a = 1 \), Figure 1 shows the influence of parameters \( b \) and \( t \) on the sequence generated by Map (2).

![Figure 1](image.png)

(a) The influence of \( b \) on \( x_n \)
(b) The influence of \( t \) on \( x_n \)

**Figure 1.** When \( a = 1 \), the influence of parameters on iteration variable \( x_n \).

From Figure 1a, we can see that with the change of parameter \( b \), the dynamic behavior of Map (2) does not change, but the value range of the iterative variable \( x_n \) changes. When \( a = 1 \) and \( b = 1 \), iterative variable \( x_n \in [-2, 2] \). When \( a = 1 \) and \( b = 4 \), iterative variable \( x_n \in [-1, 1] \). When \( a = 1 \) and \( b = 10 \), iterative variable \( x_n \in [-0.6, 0.6] \). As the parameter \( b \) increases, the iteration variable \( x_n \) decreases. Therefore, the parameter \( b \) affects the amplitude of the iteration variable \( x_n \). Through a large number of statistical experiments, we find that when \( b = 4 \), which is more conducive to the realization by digital circuits. Contrary to the effect of the parameter \( b \), the dynamic behavior of Map (2) changes with the change of the parameter \( t \). When \( a = 1 \), with the change of parameter \( t \), the iterative variable shows the period-doubling bifurcation, which is a classical dynamic behavior that leads to chaos. When \( a = 1 \) and \( t = 3 \), the iterative variable of this special cubic map has the largest value range.

When \( a = -1 \), Figure 2a shows the influence of parameter \( b \) on the sequence generated by Map (2), and Figure 2b shows the influence of parameter \( t \) on the sequence generated by Map (2). From Figure 2, we also find that parameter \( b \) affects the amplitude of iteration variable \( x_n \), and parameter \( t \) affects the dynamic behaviors of iteration variable \( x_n \). When \( a = -1 \), with the change of parameter \( t \), the iterative variable also shows the period-doubling bifurcation, which is a classical dynamic behavior that leads to chaos. But when \( a = -1 \), the specific evolution process is different from the situation of \( a = 1 \). Different maps can be realized by changing only one symbol in the algebraic structure of Map (2), which also reflects that Map (2) is highly reconfigurable and has a special symmetry.
By systematically analyzing the influence of parameters on the fixed points, we find that when parameter \( b = 4 \), Map (2) has a good range of iteration variable \( x_n \), which is more conducive to the realization by digital circuits. By selecting the appropriate parameter \( b \), Map (2) can be rewritten as:

\[
x_{n+1} = a(4x_n^3 - tx_n)
\]

where parameter \( a \in \{-1, 1\} \), \( x_n \in [-1, 1] \). (8)

2.3. Lyapunov Exponent Spectrum

Lyapunov exponent, also known as Lyapunov characteristic exponent, represents the numerical characteristics of the average exponential divergence rate of adjacent trajectories in phase space. It is an important numerical value used to identify certain dynamic behaviors, especially chaotic behavior. When the Lyapunov exponent of the iterative map is positive, the iterative map shows a chaotic behavior, and the iterative map can be called a chaotic map. The specific expression of the Lyapunov exponent is shown in Definition 1.

**Definition 1.** The Lyapunov exponent of a discrete dynamic system \( x_{n+1} = F(x_n) \) with the iterative variable \( n \) is mathematically defined as follows:

\[
\lambda = \lim_{n \to \infty} \left( \frac{1}{n} \ln \frac{|F^n(x + \Delta) - F^n(x)|}{\Delta} \right),
\]

where \( \Delta \) is a small positive number. It is relatively complicated to calculate the Lyapunov exponent by its definition. At present, the Jacobi method is commonly used to calculate the Lyapunov exponent of the discrete dynamic system. Since parameter \( b \) does not affect the dynamic behaviors of Map (2), we mainly analyze the variation of Lyapunov exponent with parameter \( t \), which is shown in Figure 3.

By numerical comparison, we find that the Lyapunov exponent spectrum of Map (2) is independent of the parameter \( a \), but only related to the parameter \( t \). In theory, the absolute values of derivatives of iteration variable \( x_n \) are:

\[
|f^{(1)}(x_n)| = |3abx_n^2 - t| = |3bx_n^2 - t|,
\]

which is the most important value to calculate the Lyapunov exponent in the Jacobi method. From Equation (10), we find that the value \( |f^{(1)}(x_n)| \) has nothing to do with the parameter \( a \). However, the parameter \( a \) participates in the whole iterative operation of Map (2). Therefore, the parameter \( t \) determines whether Map (2) has chaotic behavior, and the parameter \( a \) determines the evolution process of a specific chaotic behavior. By systematically analyzing the influence of parameters on dynamic behavior, we find that when parameter \( t = 3 \), Map (2) has the largest Lyapunov exponent, that is \( \lambda = 1.1, \)
and perfect chaotic behavior. By selecting the appropriate parameter $t$, Map (2) can be rewritten as:

$$x_{n+1} = a(4x_n^3 - 3x_n), \ n \in \mathbb{N}, \ x_n \in \mathbb{R},$$

where parameter $a \in \{-1, 1\}, \ x_n \in [-1, 1]$.

![Figure 3](image-url)

**Figure 3.** The Lyapunov exponent spectrum of Map (2) under different parameter $a$.

### 2.4. Symmetry Analysis

The symmetry of a discrete map mainly depends on its corresponding function, which limits the dynamic behavior of iteration. For Map (11), it is a discrete chaotic iterative map based on a polynomial function, and the highest order of the polynomial is 3. Therefore, the function corresponding to chaotic Map (11) is a cubic polynomial function.

$$f(x) = a(4x^3 - 3x), \ n \in \mathbb{N}, \ x_n \in \mathbb{R},$$

where parameter $a \in \{-1, 1\}$. The diagram of Function (12) is shown in Figure 4.

![Figure 4](image-url)

**Figure 4.** The diagram of Function (12) under different parameter $a$.

As shown in Figure 4, Function (12) has obvious symmetry, which is about the origin. This can also be seen from the algebraic form of the Function (12). For $-x$, its function value is $f(-x) = -a(4x^3 - 3x)$, such that $f(-x) = -f(x)$. Therefore, Function (12) is an odd function, symmetric about the origin. In addition, for the chaotic Map (11), its dynamic behavior also has symmetry. For amplitude, from Figures 1 and 2, the value of the iteration variable $x_n$ is symmetric with respect to the $x$-axis. When $a = 1$ and $a = -1$, the algebraic form of a chaotic Map (11) also has symmetry in operation, which provides the basis for the reconfigurable design of Map (11).
2.5. Chaotic Attractors and Sequence Characteristics

Lyapunov exponent is a one-dimensional numerical index to describe chaotic behavior, and chaotic behavior can also be shown by attractor, which is a high-dimensional description method. For a one-dimensional chaotic map, the attractor can be a two-dimensional image to describe chaotic behavior. In addition, time series and iterative graphs can well reflect the chaotic characteristics of the chaotic map, especially iterative graphs, which can reflect the specific behavior of attractors of a one-dimensional chaotic map.

When \( a = 1 \), chaotic Map (11) can be expressed as:

\[
x_{n+1} = (4x_n^3 - 3x_n), \quad n \in \mathbb{N}, \quad x_n \in \mathbb{R},
\]

(13)

where iterative variable \( x_n \in [-1, 1] \). Through numerical simulation, the time series and iterative graph of Map (13) are shown in Figure 5.

![Figure 5](image.png)

(a) The time series of Map (13)  
(b) The iterative graph (attractor) of Map (13)

Figure 5. The time series and iterative graph of Map (13).

Since Map (13) has a positive Lyapunov exponent, it is a chaotic map. When \( x_0 = 1 \), the output sequence of Map (13) is shown in Figure 5a. It can be seen from Figure 5a that the output sequence of Map (13) is bounded and random-like. Figure 5b is the iterative graph of the output sequence of Map (13) under \( x_0 = 1 \), which reflects the stretching and folding characteristics of the chaotic system. When \( a = -1 \), Map (11) can be expressed as:

\[
x_{n+1} = -(4x_n^3 - 3x_n), \quad n \in \mathbb{N}, \quad x_n \in \mathbb{R},
\]

(14)

where iterative variable \( x_n \in [-1, 1] \). Maps (13) and (14) have a high degree of similarity. The arithmetic in the whole iterative map is consistent, and the only difference is one symbol. Therefore, Maps (13) and (14) have high reconfigurability and symmetry. Using digital circuits to realize these two systems at the same time can reuse multiple identical units, greatly reducing the consumption of hardware resources. Compared with Maps (13) and (14), it also has good chaotic characteristics. Its time series and iterative graph are shown in Figure 6.

When \( x_0 = 1 \), the output sequence of Map (14) is shown in Figure 6a. It can be seen from Figure 6a that the output sequence of Map (14) is also bounded and random-like. But when \( x_0 = 1 \), the evolution of the output sequence of Maps (13) and (14) is different, which can also be reflected in the iteration graph of the sequence. It can be seen from Figure 6b that the chaotic attractor of Map (14) is similar to the Lorenz chaotic attractor, which is divided into two parts, but the attractor of Map (13) is mainly concentrated in one part and similar to the chaotic attractor of Logistic map. Although the form of Maps (13) and (14) are highly similar, the evolution of the output sequence is different. Because of the different sequence characteristics, when using a digital circuit to realize these two maps,
two sequences with different performances can be obtained while repeatedly using the same modules.

Figure 6. The time series and iterative graph of Map (14).

3. The Degeneration of Cubic Chaotic Map in Finite Field

3.1. The Model of Digital Cubic Chaotic Map in Finite Field

Although Map (11) has good chaotic characteristics, the digital circuit has a certain influence on the realization of a chaotic system. For the chaotic Map (11), when it is realized by a digital circuit, the original model will change, and the influence of the digital circuit must be added. Compared with the floating point operation, the fixed point operation is faster, and the logic consumption is less. In this paper, the chaotic Map (11) is realized by the form of an unsigned fixed point form. The precision of digital circuit is set as \( N \), and for decimal \( x_n \), the expression of its unsigned fixed point number is:

\[
\overline{x_n} = \left\lfloor \frac{2^N x_n}{2} \right\rfloor \mod 2^N, \quad n \in \mathbb{N}, \overline{x_n} \in \mathbb{N},
\]

where term \( \left\lfloor \frac{2^N x_n}{2} \right\rfloor \) represents the integer part of \( \frac{2^N x_n}{2} \). Multiplying both sides of Equation (11) by \( 2^{3N} \), we can obtain:

\[
2^{3N} x_{n+1} = 2^{3N} a (4x_{n}^3 - 3x_{n}) \mod 2^N, \quad n \in \mathbb{N}, \quad x_n \in \mathbb{R}. \tag{16}
\]

Merging congeneres,

\[
2^{2N} 2^N x_{n+1} = a (42^{3N} x_{n}^3 - 32^{2N} 2^N x_{n}) \mod 2^N, \quad n \in \mathbb{N}, \quad x_n \in \mathbb{R}. \tag{17}
\]

Converting decimal \( x_n \) to an unsigned fixed point number \( \overline{x_n} \):

\[
2^{2N} \overline{x_{n+1}} = a (4\overline{x_{n}}^3 - 32^{2N} \overline{x_{n}}) \mod 2^N, \quad n \in \mathbb{N}, \overline{x_n} \in \mathbb{N}. \tag{18}
\]

Dividing both sides of Equation (18) by \( 2^{2N} \), we can get:

\[
\overline{x_{n+1}} = a \left( \left\lfloor \frac{4\overline{x_{n}}^3}{2^{2N}} \right\rfloor - 3\overline{x_n} \right) \mod 2^N, \quad n \in \mathbb{N}, \overline{x_n} \in \mathbb{N}. \tag{19}
\]

By replacing symbol \( \overline{x_n} \) with the symbol \( x_n \), the digitized chaotic Map (11) is obtained:

\[
x_{n+1} = a \left( \left\lfloor \frac{4x_{n}^3}{2^{2N}} \right\rfloor - 3x_n \right) \mod 2^N, \quad n \in \mathbb{N}, \quad x_n \in \mathbb{N}. \tag{20}
\]

where the iterative variable \( x_n \in [0, 2^N - 1], a \in \{-1, 1\} \).
3.2. The Lyapunov Exponent in Finite Field

Due to the influence of the digital circuit, the digital cubic chaotic map is not an iterative map composed of real numbers in real number field, and its output sequence is no longer an aperiodic sequence in infinite field. From the mathematical meaning of the Lyapunov exponent, it represents the numerical characteristics of the average exponential divergence rate of adjacent trajectories in phase space. For the density of real numbers, we can find a third different real number interval in the two different real numbers. However, in finite field, this property is difficult to achieve. For the digital cubic chaotic Map (20), we cannot find a third different positive integer number interval in the two different positive integer numbers. For example, we cannot find a third different positive integer number interval in positive integer number 1 and 2. Therefore, the Lyapunov exponent can describe the chaotic characteristics of the dynamic iterative map in real number field, but it is difficult to analyze the digital chaotic system effectively. It does not seem sensible to use the Jacobi method to calculate the Lyapunov exponent of Map (20).

3.3. The Dynamic Behaviors of Digital Cubic Chaotic Map

In addition to the Lyapunov exponent, time series and iterative graphs can well reflect the chaotic characteristics of the chaotic map, especially iterative graphs, which can reflect the specific behavior of chaotic attractor of the one-dimensional chaotic map. However, for the proposed digital cubic chaotic map, due to the influence of numerical simulation, it cannot clearly show the characteristics of all sequences, especially when the precision is high. When \( a = 1 \), Map (20) can be expressed as:

\[
x_{n+1} = \left( \left\lfloor \frac{4 \times x_n^3}{22N} \right\rfloor - 3x_n \right) \mod 2^N, \quad n \in \mathbb{N}, \quad x_n \in \mathbb{N}.
\]  

(21)

In order to intuitively analyze the dynamic characteristics of Map (21), when \( x_0 = 1 \), the precision \( N = 4, N = 12, \) and \( N = 20 \) respectively, its time series and iterative diagram are shown in Figure 7.

As shown in Figure 7a, when \( N = 4 \), the output sequence of the digital chaotic Map (21) evolves into a constant sequence with the growing times of iteration, and the value of the output sequence eventually is 0, which can also be reflected in the iterative graph of the sequence. In Figure 7b, when \( N = 4 \), the iterative graph of the digital chaotic Map (21) finally stays at the origin 0. When \( N = 12 \), as shown in Figure 7a, the output sequence of the digital chaotic Map (21) evolves with the iteration of variable and finally becomes a periodic sequence with period 4. The periodic sequence is \( \{940, 1474, 437, 2804\} \), which can also be reflected in the iterative graph of the sequence. In Figure 7b, when \( N = 12 \), the iterative path of the digital chaotic Map (21) finally forms a closed loop and sequence points jump among four values \( 940, 1474, 437, \) and \( 2804 \). As shown in Figure 7a, when \( N = 20 \), the output sequence of the digital chaotic Map (21) shows random behavior with the evolution of iteration times, but it is still a periodic sequence. Because of the small number of iterations, we cannot see its period intuitively, but through longer iterations, it will eventually become a periodic sequence, and the same is true for the corresponding diagram of sequence iteration in Figure 7b. It can be seen from Figure 7 that compared with the original chaotic Map (11), the chaotic behavior of the digital chaotic Map (21) is degraded. The digital chaotic Map (21) only has a period attractor, it is no longer a chaotic map. When \( a = -1 \), Map (20) can be expressed as:

\[
x_{n+1} = -\left( \left\lfloor \frac{4 \times x_n^3}{22N} \right\rfloor - 3x_n \right) \mod 2^N, \quad n \in \mathbb{N}, \quad x_n \in \mathbb{N}.
\]  

(22)

where iterative variable \( x_n \in [0, 2^N - 1] \). When \( x_0 = 1 \), precision \( N = 4, N = 10 \) and \( N = 20 \) respectively, its time series and iterative diagram are shown in Figure 8. Due to the inherent limitations of digital circuits, Map (22) also exhibits the same degradation behavior as Map (21). When \( N = 10 \), the period of the output sequence of Map (22) is
small, but it is difficult to quickly distinguish from the time series; thus, the repeated period sequence is marked with a red box in the time series.

(a) When precision $N = 4$, $N = 12$, and $N = 20$, the time series of Map (21)

(b) When precision $N = 4$, $N = 12$, and $N = 20$, the iterative graph (attractor) of Map (21)

**Figure 7.** The time series and iterative graph (attractor) of Map (21) under different precision.
Figure 8. The time series and iterative graph (attractor) of Map (22)

(a) When precision $N = 4$, $N = 12$, and $N = 20$, the time series of Map (22)

(b) When precision $N = 4$, $N = 12$, and $N = 20$, the iterative graph (attractor) of Map (22)

Figure 8. The time series and iterative graph (attractor) of Map (22) under different precision.
By comparing Figures 7 and 8, we can see that there are obvious differences in the function graphs of the corresponding functions. In both cases of \( a = 1 \) and \( a = -1 \), the functions corresponding to Map (11) are the cubic function. However, the digitized Map (20) is no longer a typical cubic function, and the corresponding functions of Map (11) are different in the two different cases of \( a = 1 \) and \( a = -1 \). The deformation degree of Map (11) is large, especially \( a = -1 \). By comparing Figure 8 with Figures 6 and 7 with Figure 5, respectively, the chaotic behavior of the chaotic map realized by the digital circuit is degraded. Furthermore, by analyzing the sequence behavior with different digital circuit precision, we can find that with the increase of precision, the behavior of Map (20) is closer to the original chaotic Map (11). But for digital circuits, the hardware resources are limited, and the precision cannot be infinite. Therefore, in the case of limited resources, the dynamic behavior of Map (20) needs to be optimized to be suitable for the digital application, such as a pseudorandom sequence generator. Nevertheless, in many designs of the chaos-based pseudorandom sequence generator, the degradation of chaotic dynamics due to digital circuit implementation is often ignored, especially the complex periodic phenomenon. In this paper, we analyzed the influence of digital circuits on the complex periodic behavior of the digital cubic chaotic Map (20), and we present an effective method to resist the complex periodic behavior.

4. A Reconfigurable Pseudorandom Sequence Generator

4.1. The Analysis of Determining Periodic Sequence Source

For the digital cubic chaotic Map (20), its output sequence has complex periodic behavior. When the precision is high, the period of output pseudorandom sequence cannot be found effectively. However, for pseudorandom sequences, period and randomness are two important indexes, which need to be analyzed carefully. As seen from Figures 7 and 8, the period of the output sequence of Map (20) shows short period behavior, which is not suitable for an ideal pseudorandom sequence. Therefore, it is necessary to optimize the period of the output sequence of Map (20). Let the general iterative map in the digital field be \( x_{n+1} = F(x_n) \mod 2^N \). For a map \( x_{n+1} = F(x_n) \mod 2^N \), when different inputs are corresponding to the same output, that is \( F(x_i) = F(x_j) = k \mod 2^N, \ i \neq j \), sequence point \( x_i \) and sequence point \( x_j \) are not in one cycle. In this case, the more the sequence points are, the smaller the output period of the iterative map \( x_{n+1} = F(x_n) \mod 2^N \) will be, and multiple sequences with different periods will be generated. Therefore, to effectively increase the period of the output sequence of Map (20), Map (20) can avoid the situation that different inputs correspond to the same output, and an ideal way is to make Map (20) be a one-to-one mapping. However, the rounding function of the term \( \left\lfloor \frac{4x^3}{2^N} \right\rfloor \) in Map (20) seriously affects the functional properties of the corresponding function of Map (20). In this paper, parameters are introduced into the linear term to compensate for the truncation effect due to the elimination of the rounding function. In addition, for Map (20), when \( x_0 = 0, F(x_n) = 0, n = 1, 2, 3, \ldots \). Therefore, Map (20) always has a fixed point of 0. In order to avoid this short period behavior situation, it is necessary to offset a constant term parameter to Map (20). By analyzing the specific form of Map (20), the iterative Map (20) is optimized:

\[
x_{n+1} = a(4x^3_n - 3cx_n) + d \mod 2^N, \ n \in \mathbb{N}, \ x_n \in \mathbb{N},
\]  

where \( a \in \{-1, 1\}, \ d \neq 0 \). The corresponding function of Map (23) is:

\[
f(x) = a(4x^3 - 3cx) + d \mod 2^N, \ x \in \mathbb{N}.
\]  

The range of parameter \( c \) in Function (24) can be determined by analyzing the precondition of one-to-one mapping.

**Lemma 1.** When \( 3c \) is odd, the Function (24) is a one-to-one mapping.
Proof of Lemma 1. The lemma is proved by the contradiction. First, suppose that the function \( f(x) \) is not a one-to-one mapping over \( F_{2^N} \), that is, there are two different numbers \( t_1 \) and \( t_2 \) such that \( f(t_1) = f(t_2) \). Let \( f(t_1) = f(t_2) \), bring \( t_1 \) and \( t_2 \) into function \( f \), and we can obtain:
\[
a(4t_1^3 - 3ct_1 + d) \mod 2^N = a(4t_2^3 - 3ct_2 + d) \mod 2^N.
\]
(25)

Merging congeners in Equation (25),
\[
a((t_1 - t_2)(4(t_1^2 + t_1t_2 + t_2^2) - 3c)) \mod 2^N = 0 \mod 2^N.
\]
(26)

For the two different numbers \( t_1, t_2 \in F_{2^N}, (t_1 - t_2) \in F_{2^N} \). Since \( a \in \{-1, 1\} \), the parameter \( a \) does not affect the parity of Equation (26). In Equation (26), the term \( 4(t_1^2 + t_1t_2 + t_2^2) \) is an even number. When \( 3c \) is odd, the term \( (4(t_1^2 + t_1t_2 + t_2^2) - 3c) \) in Equation (26) is an odd number. Then, the term \( (4(t_1^2 + t_1t_2 + t_2^2) - 3c) \) cannot be a multiple of \( 2^N \), and \( t_1 = t_2 \). However, this is contradictory to the assumption that there are two different numbers \( t_1 \) and \( t_2 \). Therefore, Function (24) is a one-to-one mapping. \( \square \)

Lemma 2. ([33]). For \( x_{n+1} = F(x_n) \mod 2^N \), if the \( k \)-th column of the output function \( F(x_n) \) depends only on the first \( k \)-th columns of the \( x_n \), then for each cycle in the \( (N - 1) \)-th iteration of length \( l \), there are either two cycles of length \( l \) or one cycle of length \( 2l \) in the \( N \)-st iteration. Consequently, if \( F \) has any fixed point then for any \( i > 0 \) there are at least \( 2^{i+1} \) points that belong to cycles of length at most \( 2^i \).

First, the Boolean function of the iterative Map (23) is analyzed in the view of binary. There are only multiplication, addition, and subtraction in Map (23). For any number \( g \), according to the properties of \( \mod 2^N \), we can obtain \( 2^N - g \mod 2^N = \tilde{g} \mod 2^N \), where the symbol “\( \sim \)” is bitwise negation. Here, a concrete example of the symbol “\( \sim \)” is given. When \( N = 8 \), suppose \( g = 93 \), then the binary representation of \( g \) is \( g(2) = (0, 1, 0, 1, 1, 1, 0, 1) \), \( \tilde{g}(2) = (0, 1, 0, 1, 1, 1, 0, 1) = (1, 0, 1, 0, 0, 0, 1, 0) \). According to the conversion between decimal system and binary system, \( \tilde{g} = 162 \), then \( 2^8 - 93 \mod 2^8 = 163 \mod 2^8 = 162 + 1 \mod 2^8 \). Therefore, addition and subtraction can be converted to each other for the properties of \( \mod 2^N \), and there are only multiplication and addition in Map (23). In the addition of two multibit numbers, the carry of each single bit addition is added to the next-insignificance bit, and the multiplication also has a similar operation process. Multiplication and addition both satisfy the condition of Lemma 2.

Theorem 1. When \( ac = 1 \) and the parameter \( d \) is odd, the output sequence of Map (23) has a definite period, and the period is \( 2^N \).

Proof of Theorem 1. When \( N = 1 \), Map (23) is:
\[
x_{n+1} = a(4x_n^3 - 3cx_n) + d \mod 2, \quad n \in \mathbb{N}, \quad x_n \in \mathbb{N},
\]
(27)

where \( x_n \in \{0, 1\} \). Let the initial variable \( x_0 = 0 \), and bring \( x_0 = 0 \) into Equation (27), then we can obtain \( x_1 = d \mod 2 \). Since \( d \) is odd, \( x_1 = d \mod 2 = 1 \). When \( x_1 = 1 \), bringing \( x_1 = 1 \) into Equation (27), then we can obtain \( x_2 = 4a - 3c + d \mod 2 \). It is known from the precondition that \( a \in \{-1, 1\} \); thus, \( 4a \) is even. From Lemma 1, we already know that 3c is an odd number. From the condition of Theorem 1, we can know that parameter \( d \) is odd. Therefore, the sum of two odd numbers \( d \) and \( 3c \) is even, then the sum of two even numbers \( -3c + d \) and \( 4a \) is even, that is \( x_2 = 4a - 3c + d \mod 2 = 0 \). When \( N = 1 \), Map (23) can produce a sequence with period 2. According to Lemma 2, when system precision \( N > 1 \), the period of the output sequence of Map (23) is the exponential power of 2. When \( N > 1 \), by using Lemma 2 many times, we can obtain that if \( x_{2^N-1} \neq x_0 \), the period of the sequence generated by Map (23) is \( 2^N \). For any initial variable \( x_0 \), we have:
\[
x_1 = 4ax_0^3 - 3acx_0 + d \mod 2^N
\]
(28)
\[ x_2 = 4ax_1^3 - 3acx_1 + d \mod 2^N \]
\[ = 4ax_1^3 - (3ac)4ax_0^3 + (3ac)^2x_0 - (3ac)d + d \mod 2^N \]  
(29)
\[ x_3 = 4ax_2^3 - 3acx_2 + d \mod 2^N \]
\[ = 4ax_2^3 - (3ac)4ax_1^3 + (3ac)^24ax_1^3x_0 - (3a)^34acx_1^3 + (3a)^2d - (3a)d + d \mod 2^N \]  
(30)
\[ x_{2N-1} = 4a\left( \sum_{i=1}^{2^{N-1}-1} (-3ac)^{2^{N-1}-1-i}x_i^3 \right) + d\left( \sum_{i=1}^{2^{N-1}-1} (-3ac)^{i-1} \right) + 4a(-3ac)^{2^{N-1}-1} x_0^3 + (-3ac)^{2^{N-1}} x_0 \mod 2^N \]  
(31)

Let the initial variable \( x_0 = 0 \), we can obtain:

\[ x_{2N-1} = 4a\left( \sum_{i=1}^{2^{N-1}-1} (-3ac)^{2^{N-1}-1-i}x_i^3 \right) + d\left( \sum_{i=1}^{2^{N-1}-1} (-3ac)^{i-1} \right) \mod 2^N \]  
(32)

Since \( 4a \) is an even number, the term \( 4a\left( \sum_{i=1}^{2^{N-1}-1} (-3ac)^{2^{N-1}-1-i}x_i^3 \right) \) in Equation (32) also is an even number. From the condition of Theorem 1, we can know that parameter \( ac = 1 \), then the term \( \left( \sum_{i=1}^{2^{N-1}-1} (-3ac)^{i-1} \right) \) in Equation (32) is even. Therefore, the product of an even number \( \left( \sum_{i=1}^{2^{N-1}-1} (-3ac)^{i-1} \right) \) and an odd number \( d \) is even, and the sum of an even number \( d\left( \sum_{i=1}^{2^{N-1}-1} (-3ac)^{i-1} \right) \) and even numbers \( 4a\left( \sum_{i=1}^{2^{N-1}-1} (-3ac)^{2^{N-1}-1-i}x_i^3 \right) \) is even. However, term \( 4a\left( \sum_{i=1}^{2^{N-1}-1} (-3ac)^{2^{N-1}-1-i}x_i^3 \right) \) and term \( d\left( \sum_{i=1}^{2^{N-1}-1} (-3ac)^{i-1} \right) \) are polynomial on the power of \( (3ac) \). Then, \( x_{2N-1} \) is polynomial on the power of \( (3ac) \). For each \( x_n, 1 \leq n \leq 2^N-1 \), they have a common factor \( (3ac) \). Since \( (3ac) \) is an odd number, \( x_{2N-1} \) cannot be the multiple of \( 2^N \). According to the properties of \( \mod 2^N \),
\[ x_{2N-1} = 4a\left( \sum_{i=1}^{2^{N-1}-1} (-3ac)^{2^{N-1}-1-i}x_i^3 \right) + d\left( \sum_{i=1}^{2^{N-1}-1} (-3ac)^{i-1} \right) \mod 2^N \neq 0, \]  
that is, \( x_{2N-1} \neq x_0 \). In conclusion, the period of the sequence generated by iterative Map (23) is \( 2^N \). \( \square \)

For Map (23), according to the selection of parameter \( a \) and parameter \( c \), it can be changed into two similar maps. When \( a = 1 \) and \( c = 1 \), Map (23) can be changed into Map (33).
\[ x_{n+1} = (4x_n^3 - 3x_n) + d \mod 2^N, \quad n \in \mathbb{N}, \quad x_n \in \mathbb{N}. \]  
(33)

When \( a = -1 \) and \( c = 1 \), Map (23) can be changed into Map (34).
\[ x_{n+1} = -(4x_n^3 + 3x_n) + d \mod 2^N, \quad n \in \mathbb{N}, \quad x_n \in \mathbb{N}. \]  
(34)

By simplifying Map (34), we can obtain:
\[ x_{n+1} = -4x_n^3 - 3x_n + d \mod 2^N, \quad n \in \mathbb{N}, \quad x_n \in \mathbb{N}. \]  
(35)

Although the output sequences of Maps (33) and (35) vary with the evolution of iteration, the periods of sequences generated by both of them are \( 2^N \). By combining the form of Maps (33) and (35), a comprehensive map is obtained:
\[ x_{n+1} = a4x_n^3 - 3x_n + d \mod 2^N, \quad n \in \mathbb{N}, \quad x_n \in \mathbb{N}, \]  
(36)
where \( a \in \{-1, 1\} \). For any odd number \( d \), Map (36) can output a sequence with period \( 2^N \). Therefore, Map (36) can be regarded as a family of iterative map sets that can output sequences with period \( 2^N \) under precision \( N \), in which the elements are Map (36) with an
odd number \( d \) under a certain precision. In this set, when \( d = 1 \), Map (36) under precision \( N \) has a simpler form; thus, we can obtain a definite map:

\[
x_{n+1} = a4x_n^3 - 3x_n + 1 \mod 2^N, \quad n \in \mathbb{N}, \quad x_n \in \mathbb{N},
\]

where \( a \in \{-1, 1\} \). Chengqing Li, Bingbing Feng, Shujun Li, Jürgen Kurths, and Guanrong Chen observed that the effectiveness of the improvement method for digital chaotic system mainly depends on the period distribution under low precision \[19\]. However, many proposed improvement methods ignore the analysis under low precision. In contrast, the map proposed in this paper also has a good periodic distribution under low precision. When precision \( N \) is low, all periodic behaviors of the map in the digital field can be found by traversal search. In Figure 9, when precision \( N = 4 \), all periodic behaviors of Maps (20) and (37) are shown completely.

![Figure 9](image_url)

(a) When \( a = 1 \), periodic behavior of Map (20) is on the left and periodic behavior of Map (37) is on the right

(b) When \( a = -1 \), periodic behavior of Map (20) is on the left and periodic behavior of Map (37) is on the right

**Figure 9.** When \( N = 4 \), all periodic behaviors of Maps (20) and (37).

As can be seen from Figure 9, when \( a = 1 \), Map (20) shows the complex periodic behavior, and no matter which initial value starts, it will eventually become a periodic sequence with period 1 and show only period point 0 after several iterations. This short period behavior is not acceptable in the design of a pseudorandom sequence generator, and
the sequence cannot be used as a pseudorandom sequence in the cryptosystem. However, when \( a = 1 \), Map (37) shows excellent periodic behavior. Its output sequence forms a perfect closed-loop, the period reaches the ideal value \( 2^4 \), and shows ergodicity. When \( a = -1 \) Map (20) shows more complex periodic behavior than when \( a = 1 \). Five sequence points 4, 11, 13, 5, and 14 form a small closed-loop, which represents a periodic sequence \( (4, 11, 13, 5, 14) \) with period 5. Starting from one of the remaining initial values, after several iterations, it will eventually become a periodic sequence with period 1 and show only period point 0. This short period behavior and multiple period behavior is also not acceptable in the design of a pseudorandom sequence generator, and the sequence is hard to be used as a pseudorandom sequence in the cryptosystem. As in the case of \( a = 1 \), when \( a = -1 \) the output sequence of Map (37) forms a perfect closed loop, and the period reaches the ideal value \( 2^4 \). Since the values of the output sequences of Map (37) are different with the evolution of iteration in the two cases of \( a = 1 \) or \( a = -1 \), they belong to two different periodic sequences with the largest period.

When precision \( N \) is higher, the difference between all periodic behaviors of Maps (20) and (37) is more obvious. Due to the limitation of computer precision and storage devices, the periodic behavior of Maps (20) and (37) cannot be directly described when the precision \( N \) is high. Therefore, when precision \( N = 17 \), the periods of the output sequences of Maps (20) and (37) are analyzed and distinguished. When the system precision is high, the period of sequence can be estimated by autocorrelation detection. Autocorrelation detection can reflect the dependence relationship of a signal between two different moments, which is an important detection method to evaluate the sequence period. The expression of the autocorrelation function is as follows:

\[
R_z(m) = \frac{1}{K} \sum_{n=0}^{K-1} Z_n Z_{n+m},
\]

(38)

where \( R_z(m) \) and \( K \) represent the autocorrelation function and the length of detection sequence, respectively. The autocorrelation detection results of the output sequences of Maps (20) and (37) are shown in Figure 10.

For Map (20), when \( a = 1 \) or \( a = -1 \), a 1000-length sequence is selected from the output sequence for autocorrelation detection. However, for Map (37), when \( a = 1 \) and \( a = -1 \), a \( 2^{17} \)-length sequence is selected from the output sequence for autocorrelation detection. In Figure 10, the distance between two peaks can be approximately expressed as the period of the sequence. For Map (20), there are many peaks in Figure 10a,b. Through autocorrelation detection, whether \( a = 1 \) or \( a = -1 \), the period of output sequence generated by Map (20) is much less than 1000. By measuring the distance between the two peaks, we find that whether \( a = 1 \) or \( a = -1 \), the period of output sequence generated by Map (20) is less than 200. In contrast, for Map (37), only one peak can be seen in Figure 10a,b respectively. Since the length of the test sequence is \( 2^{17} \), whether \( a = 1 \) or \( a = -1 \), the period of output sequence generated by Map (37) is greater than or equal to \( 2^{17} \). According to the theory of digital circuits and cryptography, when the system precision is \( N \), the maximum period of the sequence is \( 2^N \). When system precision \( N = 17 \), whether \( a = 1 \) or \( a = -1 \), the period of output sequence generated by Map (37) is less than or equal to \( 2^{17} \). Therefore, when system precision \( N = 17 \), whether \( a = 1 \) or \( a = -1 \), the period of output sequence generated by Map (37) is equal to \( 2^{17} = 131072 \). When system precision \( N = 17 \), whether \( a = 1 \) or \( a = -1 \), the period of the sequence generated by Map (37) is at least 655 times larger than that of the sequence generated by Map (20). The proposed Map (37) in this paper greatly improves the period of the sequence generated by the digital cubic chaotic map in finite field.

4.2. The Design of a Reconfigurable Pseudorandom Sequence Generator

In addition to period behavior, as a rule, the weakest statistical property the sequence must necessarily satisfy to be considered as pseudorandom in any reasonable meaning is a uniform distribution, that is, each term of the sequence must occur with the same
frequency. From Theorem 1, the period of the output sequence of Map (36) is $2^N$, and thus is its special form Map (37). For Maps (33) and (35), each term of the sequence occurs with the same frequency within a period, and the probability of occurrence is $1/2^N$, which satisfies the uniform distribution in pseudorandom sequence theory. Since the output sequence of Map (36) has a large period, good randomness, excellent algebraic form, and high symmetry, a reconfigurable pseudorandom sequence generator is designed based on Map (36). When $N = 32$, the principle block diagram is shown in Figure 11.

(a) When $a = 1$, autocorrelation detection of Map (20) is on the left and autocorrelation detection of Map (37) is on the right

(b) When $a = -1$, autocorrelation detection of Map (20) is on the left and autocorrelation detection of Map (37) is on the right

Figure 10. When $N = 17$, autocorrelation detection of Maps (20) and (37).

As shown in Figure 11, the proposed reconfigurable pseudorandom sequence generator implements both Maps (33) and (35), where module $d - 3x_n$ and module $x_3^n$ are shared

Figure 11. A reconfigurable pseudorandom sequence generator.
reconfigurable modules. By reducing the repeated use of these two modules, hardware resources can be greatly saved. Compared with only one map, the number of output sequences can be further increased, and the randomness of output sequences can be enhanced by implementing two maps at the same time. The output sequences of Maps (33) and (35) undergo the same processing: module \( P(x) \), module \( B(x) \), and module \( H(x) \). For an integer number \( s \) under precision \( N \), it has a binary vector form \( \{ s[N-1], s[N-2], \ldots, s[0] \} \). The relationship between decimal \( s \) and binary \( s[k] \), \( k = 0, 1, 2, \ldots, N - 1 \), is as follows:

\[
s = \sum_{k=0}^{N-1} s[k]2^k, \quad s \in \mathbb{N}, \quad s[k] \in \{0,1\}.
\]  

(39)

In Figure 11, the specific operation method of the module \( P(x) \) is as follows:

\[
P(x) = x \oplus (x_{<13}) \oplus (x_{<23}),
\]  

(40)

where \( x_{<i} \) denotes \( i \)-bit left rotation of a value \( x \), and \( \oplus \) denotes bitwise xor operation. The module \( P(x) \) is a linear diffusion operation, it can associate each bit of the binary vector of \( x \) with other bits. The function of the module \( B(x) \) is bit extraction, its specific operation is:

\[
B(x) = \{ x[31], x[30], x[16], x[1], x[0] \}.
\]  

(41)

For an integer number \( x \), the module \( B(x) \) will extract specific 5 bits from the binary vector of \( x \), that is \( \{ x[31], x[30], x[16], x[1], x[0] \} \). In Figure 7, the module \( H(x) \) is a special nonlinear function, which has the property of five inputs and one output. For a five bits input \( \{ h_4, h_3, h_2, h_1, h_0 \} \), the specific form of function \( H(x) \) is:

\[
H(x) = h_1 \oplus h_4 \oplus h_2h_3 \oplus h_1h_4 \oplus h_0h_2 \oplus h_3h_4 \oplus h_1h_3 \oplus h_0h_1h_3h_4 \oplus h_0h_1h_3h_4.
\]  

(42)

Since the output sequences of Maps (33) and (35) undergo the same processing: module \( P(x) \), module \( B(x) \), and module \( H(x) \), these three different modules also are the reconfigurable modules in the proposed reconfigurable pseudorandom sequence generator. By calculating the Walsh spectrum of the function \( H(x) \), we find that the function is balanced, that is, the number of 0 and 1 in its output sequences are equal. Since Module \( P(x) \) is a linear diffusion operation and reversible, it is balanced. The module \( B(x) \) extracts specific bits from the binary vector of \( x \), it does not affect the characteristics of the sequence. For the module \( H(x) \), it also is a balanced Boolean function. Therefore, the output sequence of the proposed reconfigurable pseudorandom sequence generator is balanced, and the period is \( 2^N \). When \( N = 32 \), the period of the output sequence is \( 2^{32} \).

4.3. The Implementation and Performance Analysis

After designing the reconfigurable pseudorandom sequence generator, we implement it in hardware via FPGA. When \( b = 1 \), the hardware implementation diagram of a specific reconfigurable pseudorandom sequence generator is shown in Figure 12, the time series waveform is shown in Figure 13, and the consumption of hardware resources is shown in Table 1.

From Figure 13, the output of the proposed reconfigurable pseudorandom sequence generator does show a noise-like behavior. In addition to using probability distribution to analyze the randomness of sequence, randomness detection is also another important method. At present, the main test method of randomness in the world is NIST-sp800 test suite, which is the national standard for measuring randomness in the world and published by the National Institute of Standards and Technology of the United States [34]. NIST-sp800 test suite focuses on a variety of models for sequence randomness detection, including many approximately independent statistical tests, such as linear complexity, approximate entropy, etc. The version of NIST-sp800 test suite used in this paper is 2.1.2, and 100 groups
of sequences with a length of 1,000,000 are detected. The detection results are shown in Table 2.

![Top-level block diagram of reconfigurable pseudorandom sequence generator](image1.png)

**Figure 12.** The hardware implementation diagram of reconfigurable pseudorandom sequence generator.

![RTL level circuit of reconfigurable pseudorandom sequence generator](image2.png)

**Figure 13.** The output sequence waveform of reconfigurable pseudorandom sequence generator.

| Total Logic Elements | Total Combinational Functions | Dedicated Logic Registers | Lpm_Mult |
|----------------------|-------------------------------|---------------------------|----------|
| 1494                 | 1493                          | 34                        | 2        |

In Table 2, the minimum pass rate for each statistical test with the exception of the random excursion (variant) test is approximately = 96 for a sample size = 100 binary sequences. The minimum pass rate for the random excursion (variant) test is approximately = 62 for a sample size = 66 binary sequences. In Table 2, if the ratio is greater than 1, it means passing the corresponding subtest in NIST-sp800 test suite. In the parameter setting of NIST-sp800 test suite, the block length of Block frequency test is 128; the block length of NonOverlappingTemplate test is 9; the block length of OverlappingTemplate test is 9; the block length of ApproximateEntropy test is 10; the block length of Serial test is 16; the block length of LinearComplexity test is 500. Among the 15 subtests, Cumulative Sums test, NonOverlappingTemplate test, RandomExcursions test, RandomExcursionsVariant...
test, and Serial test have multiple test output results. In Table 2, only the lowest U-value in multiple results of these 5 different subtests are showed.

Table 2. Results of the NIST-sp800 test suite.

| Subtest                  | U-Value  | Ratio  | Result |
|--------------------------|----------|--------|--------|
| Frequency                | 0.090936 | 99/96  | Success|
| Block frequency          | 0.779188 | 100/96 | Success|
| Cumulative Sums          | 0.289667 | 99/96  | Success|
| Runs                     | 0.009535 | 100/96 | Success|
| LongestRun               | 0.739918 | 98/96  | Success|
| Rank                     | 0.935716 | 100/96 | Success|
| FFT                      | 0.262249 | 99/96  | Success|
| NonOverlappingTemplate   | 0.012650 | 99/96  | Success|
| OverlappingTemplate      | 0.616305 | 99/96  | Success|
| Universal                | 0.145326 | 98/96  | Success|
| ApproximateEntropy       | 0.834308 | 99/96  | Success|
| RandomExcursions         | 0.002043 | 65/62  | Success|
| RandomExcursionsVariant  | 0.032000 | 65/62  | Success|
| Serial                   | 0.224821 | 99/96  | Success|
| LinearComplexity         | 0.055361 | 97/96  | Success|

The p-value is a statistical test value for the randomness of a group of measured sequences for NIST-sp800 test suite, and it is a positive number less than or equal to 1. The larger p-value is, the stronger the randomness of the measured sequence is. For a group of test sequences, p-value is the index to determine whether the sequence can pass all 15 subtests in NIST-sp800 test suite. If p-value is determined to be equal to 1, the sequence appears to be completely random, and p-value less than 0.01 indicates that the sequence appears to be completely non-random. In Table 2, U-value is a distribution of p-value of 100 groups of sequences, and U-value of each subtest can be seen from Table 2. As shown in Table 2, the output sequence has passed the test of NIST-sp800 test suite and shows good randomness. Since period and randomness can be determined, the proposed reconfigurable pseudorandom sequence generator not only has low resource consumption but also has good performances.

For the chaos-based pseudorandom sequence generator, which has two main characteristics, one is period, the other is randomness. Therefore, the period and randomness are the core indexes. However, randomness has been widely concerned, but periodicity has not. In addition, with the development of chaos-based cryptosystem, increased attention has been paid to the consumption of hardware resources and reconfigurability has become an important principle in the design of pseudorandom sequence generators, stream cipher, and block cipher, especially stream cipher. For the design of stream cipher, pseudorandom sequence generator is a required important module, and its reconfigurability has gained extensive attention and been studied. Therefore, reconfigurability is regarded as the third important index of performance for the pseudorandom sequence generator based on the chaotic map.

At present, the pseudorandom sequence generators based on chaotic map is an important part of the chaos-based cryptosystem, and the short period characteristic has gradually become a common attack method to crack chaos-based cryptosystems. Compared with randomness, the period is often ignored by most designers of chaos-based cryptosystems. Most of the proposed chaos-based cryptosystems do not analyze the period of the output sequence, and certain chaos-based cryptosystems suggest special algorithms for searching the period of output sequence via computer under low precision; only a few chaos-based cryptosystems give the proof of definite period of the output sequence in mathematics under any precision. This paper selects several existing typical and great chaos-based pseudorandom sequence generators and compares them with the methods proposed in
this paper. The comparison results between the proposed and other existing methods are shown in Table 3.

Table 3. The comparison results between the proposed and other existing methods.

| Sequence Generator | Period       | NIST-sp800 Test Suite | Reconfigurability |
|---------------------|--------------|------------------------|-------------------|
| Method in [15]      | $\approx 10^8(K = 20)$ | Success                | No                |
| Method in [16]      | $2(3^{M-1})$ | Success                | No                |
| Method in [17]      | 656($N = 19$) | Success                | No                |
| Method in [18]      | Not analyzed | Success                | No                |
| The proposed        | $2^N$        | Success                | Yes               |

In Table 3, although all five methods have passed the random test, the periods of their output sequences are completely different. For Method in [15] and Method in [16], by transforming the accuracy between different systems, we can get the following relationship: $N = K(\log_3 3), N = 2M(\log_2 3)$. For the method proposed by [15], a special algorithm of searching sequence period is given, and only an approximate value of the period is given under precision $K = 20$. By comparison, the period of output sequence generated by the proposed method in this paper is $2^{K(\log_3 3)} \approx 10^9$. For the method proposed by [16], the period of the output sequence is $2^{(3^{M-1})}$. Since $N = 2M(\log_2 3)$, the period of output sequence generated by the proposed method in this paper is greatly larger than the method proposed by [16]. For the method proposed by [17], an algorithm of exhaustive search is given, and the period of the output sequence is 656 under precision $N = 19$. However, when $N = 19$, the period of output sequence generated by the proposed method in this paper is $2^{19} = 524288$. For the method proposed by [18], the period of the output sequence is not analyzed, only the randomness is tested. Except for period comparison, all five methods have passed the randomness test, and only the method proposed in this paper is reconfigurable. As can be seen from Table 3, the results of the period, randomness, and reconfigurability analysis demonstrate that the proposed one is better than the related work in terms of security and performance.

In addition to chaos-based pseudorandom sequence generators, there are many standard pseudorandom sequence generators [35], such as linear congruence generator (LCG), linear feedback shift register sequence generator (LFSR), well-known trinomial-based generalized feedback shift register sequence generator (GFSR), Mersenne twist generator (MT19937), subtract with borrow generator (SWB), and stream cipher algorithm. There are differences in their random performance, mainly whether they have encryption ability. AES in stream cipher modes (OFB, CTR, KTR) is the pseudorandom sequence generator with encryption ability. Besides NIST-sp800, Testu01 has the majority of types of randomness tests [35]. Offering a rich variety of empirical tests is the purpose of the TestU01 library. In TestU01 user’s guide (compact version) [35], to test a pseudorandom sequence generator, it is recommended to start with the quick battery SmallCrush. If everything is fine, one can try Crush, and finally, the more stringent BigCrush. Testu01 testing has been added to the paper, and the test results are as follows.

In Table 4, the AES-KTR mode uses a counter as key that is increased by 1 at each encryption iteration, and the speed is the required time (in seconds) to generate $10^8$ random numbers. The randomness of the pseudorandom sequence proposed in this paper is higher than that of the ordinary standard pseudorandom sequence, but the speed is slower. Compared with the cryptographic algorithm, the randomness of the cryptographic algorithm is higher than that of the pseudorandom sequence proposed in this paper, and the speed is relatively slow, but they have the ability to encrypt information. However, the method proposed in this paper cannot encrypt the information and has no encryption ability. In Table 4, our implementations of pseudorandom sequence generators are not necessarily the fastest possible. According to the cipher document, we found that the optimized AES can run normally above 10 GB/s, which is a high speed. In view of the
above indicators analysis, we can conclude that the pseudorandom sequence generator based on the proposed method in this paper has good randomness.

Table 4. The comparison results between the proposed and other existing generators.

| Generators                  | Speed | Small Crush | Crush | Big Crush |
|-----------------------------|-------|-------------|-------|-----------|
| LCG(2^{24}, 16,598,013, 12,820,163) | 0.66  | Failure     | Failure | Failure   |
| LCG(2^{31}, 65,539, 0)       | 0.65  | Failure     | Failure | Failure   |
| LCG(2^{32}, 69,069, 1)       | 0.67  | Failure     | Failure | Failure   |
| LCG(2^{46}, 5^{13}, 0)       | 0.75  | Failure     | Failure | Failure   |
| LCG(2^{63}, 5^{19}, 1)       | 0.75  | Success     | Failure | Failure   |
| LFSR113                     | 1.0   | Success     | Failure | Failure   |
| LFSR258                     | 1.2   | Success     | Failure | Failure   |
| GFSR(250, 103)              | 0.9   | Success     | Failure | Failure   |
| MT19937                     | 1.6   | Success     | Failure | Failure   |
| The proposed                | 3.4   | Success     | Success | Failure   |
| SWB(2^{24}, 10, 24)         | 4.3   | Success     | Success | Success   |
| AES (CTR)                   | 5.4   | Success     | Success | Success   |
| AES (OFB)                   | 5.8   | Success     | Success | Success   |
| AES (KTR)                   | 5.2   | Success     | Success | Success   |

5. Conclusions

The complex periodic behavior of the digital chaotic system is a potential factor that affects the application of the digital chaotic system in the pseudorandom sequence generator. This paper studies a combined cubic chaotic map and analyzes the dynamic characteristics of the special cubic chaotic map after digitization. Compared with the chaotic attractor in real number field, we find that the attractor of the proposed cubic chaotic map in digital field changes and finally forms the periodic attractor. For the proposed cubic chaotic map, when \( a = -1 \), the chaotic attractor can be divided into two parts, which is different from that when \( a = 1 \). However, after digitization in finite field when \( a = -1 \), the attractor will deform greatly and become similar to the situation of \( a = 1 \). By offsetting the influence of truncation, for any precision \( N \), we successfully prove the period of output sequence of the digital cubic map is \( 2^N \) in mathematics, which is the theoretical maximum value. Based on the great period behavior, a novel pseudorandom sequence generator is proposed, which is high symmetry and can greatly reduce the consumption of resources. Compared with the existing methods, the proposed pseudorandom sequence generator in this paper has advantages in period and reconfigurability. Through theoretical analysis and random detection, the sequence output by the proposed generator also has good randomness, which can be used to generate good periodic pseudorandom sequences and can be combined with other chaotic functions to further form the required chaotic pseudorandom sequences.

Author Contributions: C.W. and Q.L.; methodology, J.S. and Y.D.; Software, C.W.; writing—original draft preparation, Y.Z. and J.T.; writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Scientific Research Foundation for the PhD (grant number XJ20BS44), Scientific Research Project of Education Department of Hubei Province (grant number Q20192201), Youth Project of Hubei School of Economics (grant number XJ201912), and Hubei Provincial Natural Science Foundation of China (grant number 2019CFB778 and 2020CFB306).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing not applicable.

Conflicts of Interest: The authors declare no conflict of interest.
33. Klimov, A.; Shamir, A. A new class of invertible mappings. In Cryptographic Hardware and Embedded Systems-CHES2002, Proceedings of the 4th International Workshop, Redwood Shores, CA, USA, 13–15 August 2002; Revised Papers; Springer: Berlin/Heidelberg, Germany, 2002.

34. Rukhin, A.; Soto, J.; Nechvatal, J.; Smid, M.; Barker, E. A statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications; NIS Special Publication 800–22, Rev. 1-a; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2010.

35. L’Ecuyer, P.; Simard, R. TestU01 A Software Library in ANSI C for Empirical Testing of Random Number Generators—User’s Guide, Compact Version. 16 May 2013. Available online: http://simul.irc.unreal.ca/testu01/guideshorttestu01.pdf (accessed on 27 July 2021).