



Article

Optimizing Chaotic Behavior: Systematic Shifting and Operations for Robust 1-D Chaotic Maps

Mrittika Chowdhury, Ziyi Niu [†], Shuai Song [†], Anurag Dhungel and Md Sakib Hasan ^{*}

Department of Electrical and Computer Engineering, University of Mississippi, Oxford, MS 38677, USA; mchowdh2@go.olemiss.edu (M.C.); zniu@go.olemiss.edu (Z.N.); ssong2@go.olemiss.edu (S.S.); adhungel@go.olemiss.edu (A.D.)

^{*} Correspondence: mhasan19@umassd.edu

[†] These authors contributed equally to this work.

Abstract

In this work, we present a systematic framework to optimize robust 1-D chaotic maps, focusing on expanding the uninterrupted chaotic region and enhancing chaotic properties throughout the entire parameter space. To achieve these objectives, we propose three distinct techniques, each involving systematic manipulations of chaotic seed maps. These manipulations include shifts and operations such as multiplication and division, which result in significant improvements in their chaotic characteristics. The effectiveness of the proposed methods is demonstrated through a comprehensive analysis using bifurcation plots, the maximum Lyapunov exponent, the correlation coefficient, Shannon entropy, the average Lyapunov exponent, and the chaotic ratio. The results illustrate the attainment of an extensive and uninterrupted chaotic range, alongside enhanced chaotic behavior due to the application of shifted maps. Additionally, in this work we also investigate the impact of combining general shifted maps with halfway-shifted maps, showing that their product leads to further improvements in chaotic properties and their division widens the chaotic ratio. In the last proposed method, the combined product division (CPD) maps achieve the highest overall performance, attaining a maximum Lyapunov exponent of 1.3567 and an average Lyapunov exponent of 1.3498, while maintaining a perfect chaotic ratio ($CR = 1$) across the entire parameter space. To demonstrate hardware feasibility, some of the proposed maps were also implemented on an FPGA, and the results were compared with MATLAB R2022b simulations. The close match between the two validates the practicality of implementing these chaotic systems in hardware. The proposed techniques have potential applications in areas such as random number generation, chaos-based cryptography, and secure communication, among others. To prove this, the optimized maps are leveraged to design a chaos-based pseudo-random number generator (PRNG) that passes statistical tests including NIST SP 800-22 (all 15 sub-tests passed) and TestU01 (38/38 Rabbit, 17/17 Alphabit, 102/102 BlockAlphabit), validating cryptographic-grade randomness.



Academic Editor: Costas Psychalinos

Received: 12 May 2026

Revised: 14 July 2026

Accepted: 24 July 2026

Published: 27 July 2026

Copyright: © 2026 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 \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Keywords: nonlinear dynamics; chaotic system; discrete map; security; robust chaos; FPGA; PRNG

1. Introduction

Nonlinear dynamic systems have garnered significant interest across various scientific and technological fields in recent years. Chaos theory, a key component of nonlinear dynamics, has seen extensive application in disciplines ranging from mathematics and

physics to computer science, meteorology, ecology, astronomy, and economics since its inception [1]. Chaos is characterized by aperiodic behavior in deterministic nonlinear dynamic systems that are highly sensitive to initial conditions, a phenomenon often termed the “butterfly effect”. This implies that small variations in the starting state of a system can lead to dramatically different outcomes, despite the deterministic nature of these systems, which allows for repeatable aperiodic sequences under consistent system parameters. Unlike stochastic systems, which also display aperiodic behavior but through random processes, chaotic systems are entirely deterministic [2]. The unique properties of chaotic systems make them particularly useful in various security applications, such as pseudo-random number generators (PRNGs) [3,4], image encryption [5,6], mitigation of side-channel attacks [7,8], and enhancing hardware security through physical unclonability [9].

In the field of chaotic systems, they can be categorized by the timing of their governing equations (either continuous-time or discrete-time) and by the dimensionality of their state variables (either one-dimensional or multi-dimensional). For instance, one-dimensional discrete-time maps, such as the logistic [10], sine [3], and tent maps [10], are characterized by a single equation that evolves the state variable at discrete time intervals. In contrast, a system like the Lorenz system [11] exemplifies multi-dimensional continuous-time chaotic systems, which are defined by three interlinked ordinary differential equations continuously evolving multiple state variables. Conversely, the Hennon map [12], another multi-dimensional system, progresses through discrete time steps, illustrating the diversity within chaotic systems.

Multi-dimensional chaotic systems are typically capable of generating complex, high-entropy chaos which is beneficial for robust applications; however, they also demand greater computational resources and increased system complexity, presenting significant implementation challenges [13]. For example, in [14] Yalcin et al. presented several classical and quantum chaotic maps and implemented them on an FPGA for pseudo-random number generation; however, the approach relies on a multi-stage model-based workflow, HDL generation, and fixed-point hardware realization, which increases the implementation complexity relative to lightweight one-dimensional chaotic constructions. The study by Damaj et al. [15] confirms that multi-stage chaotic implementations are not lightweight: increasing the discretization order and precision raises the need for intermediate variables, arithmetic units, larger FPGA area, and higher power consumption, thereby making such designs less attractive for resource-constrained applications. On the other hand, one-dimensional discrete-time chaotic systems are favored for certain applications due to their minimal computational requirements and ease of hardware implementation. These systems traditionally employ classic mathematical functions, such as the logistic or tent maps. Their simplicity facilitates straightforward implementations, but this comes with limitations in terms of security because the predictability of these systems can be compromised. The constrained chaotic behavior inherent to these maps can lead to predictability, especially as operational conditions or parameter values shift, potentially driving the system from chaotic to non-chaotic states (fixed points or periodic orbits) [16]. In [17], Nazish et al. emphasized that although one-dimensional chaotic maps are simple and hardware-friendly, their chaotic behavior is typically confined to narrow parameter intervals and may collapse into periodic or non-chaotic states under parameter variation or finite-precision implementation, which limits their suitability for secure applications. In [18], Alawida et al. show that 1-D chaotic maps remain lightweight and straightforward to implement, but their security is weakened by constrained chaotic behavior and limited unpredictability, which can compromise their suitability for robust cryptographic use.

To address these security vulnerabilities and enhance the utility of one-dimensional maps in secure applications, researchers are investigating ways to expand the chaotic

range and functionality. This is accomplished through various innovative approaches, such as dynamically altering control parameters [3,19], integrating multiple foundational chaotic maps [10,20], or applying discrete wheel-switching techniques [21]. Other strategies include the sine transformation of multiple map combinations [22], as well as advanced modulation and coupling methods [23], and the use of exponential chaotic models [24]. These methodologies aim to enrich the system's operational variability and complexity, thereby increasing the difficulty for potential attackers to decipher or predict the system's dynamics, enhancing its applicability in security-sensitive environments.

While the aforementioned techniques enhance chaotic performance—either by expanding the chaotic window or increasing chaotic entropy—they often come with added complexity and overhead. Cascade-type constructions, for instance, have been shown to preserve chaos and yield Lyapunov exponents that combine contributions from subsystems [10], and can broaden the chaotic parameter range under favorable configurations. However, these improvements are not always uniform across the entire parameter space, and the extent of chaotic enhancement in cascaded maps has been shown to depend strongly on the choice of parameters and their permutation across subsystems [25], which can limit their reliability in applications requiring consistent chaos across the full parameter range. In [26], Ul Islam et al. proposed a novel one-dimensional Exposine Chaotic Map (ECM) and its hardware-efficient approximation, the Approximate Exposine Chaotic Map (AECM), to broaden the chaotic range and improve sensitivity, entropy, and unpredictability for secure applications. However, because the map combines sine and exponential nonlinearities, the direct implementation is computationally expensive, so the authors had to introduce Taylor–Padé approximation and a lookup-table-based realization to reduce hardware complexity and runtime overhead. In [27], Li et al. proposed a one-dimensional complex discrete hyperchaotic map aimed at strengthening the chaotic properties needed for secure applications. Their design improves dynamical complexity, unpredictability, and sensitivity to initial conditions, making it more suitable for cryptographic use than many conventional 1D maps. However, the added nonlinear structure also brings practical trade-offs, including greater implementation complexity and hardware overhead, which they address through a hardware-oriented realization strategy. In [28], Silva et al. proposed a hardware-efficient chaotic PRNG based on a Sugeno-approximated sine map and posit arithmetic to improve randomness and mitigate dynamical degradation in digital implementations; however, the enhanced robustness comes with a practical trade-off, since the design relies on approximation, perturbation, and non-native posit arithmetic, which introduces implementation complexity and conversion overhead in hardware. In [29], Sarra et al. introduced the 1D-Powered Chebyshev Quadratic Map (1D-PCQM) to enhance nonlinearity, widen the chaotic window, and improve cryptographic robustness. At the same time, the use of a hybrid multi-map encryption framework and dynamic switching/diffusion increases algorithmic complexity and computational overhead, reflecting the usual trade-off between stronger chaos and practical efficiency. In [30], Alexan et al. proposed a secure and efficient multi-image encryption scheme that integrates a 5D hyperchaotic system, a dynamically generated S-box, Langton's Ant, and Arnold's Cat Map to enhance confusion, diffusion, and permutation while maintaining real-time efficiency. They reported strong resistance to statistical, differential, and randomness-based attacks, but their design trades off increased algorithmic complexity, memory usage, and dependence on precise chaotic parameter synchronization for improved security. These studies highlight the need for alternative approaches that can deliver strong and reliable chaotic behavior without incurring significant trade-offs.

Furthermore, while several works have demonstrated FPGA implementations of chaos-based PRNGs, these implementations are often tied to specific map structures with

limited chaotic range. For instance, Garcia-Bosque et al. [31] proposed a chaos-based bitwise dynamical PRNG based on the logistic map implemented on a Virtex 7 FPGA using 32-bit fixed-point precision, dynamically changing the chaotic parameters to prevent short-period orbits and improve randomness. Yalcin et al. [14] implemented classical and quantum chaotic maps on FPGAs for pseudo-random number generation using a multi-stage model-based workflow. De la Fraga et al. [32] demonstrated hardware implementation of PRNGs based on chaotic maps, including Bernoulli, tent, and zigzag maps on FPGAs, validating randomness using the NIST SP 800-22 test suite. More recently, Luo et al. [33] proposed a high-speed PRNG based on an n -dimensional non-degenerate chaotic system with FPGA implementation. Hasan et al. [16] further demonstrated FPGA implementation of flipped-product chaotic maps for PRNG applications, validating their entropic properties against software simulation. However, these implementations are often constrained by narrow chaotic windows, limited parameter flexibility, or high hardware overhead, motivating the need for a more robust and generalizable hardware-compatible chaotic framework.

In this study, we introduce four advanced strategies for constructing more robust chaotic systems. The first approach involves individual adjustments of each 1D seed map based on specific control parameter values, after which these modified curves are aggregated to create the comprehensive map. The second technique modifies each seed map by performing a half-shift and then multiplying this adjusted map by its original shifted form. These two were reported in our previous work [34]. While these operations share a structural similarity with cascade-type constructions [10], in that both combine multiple maps to enhance chaotic properties, the key distinction lies in the systematic use of the shift parameter b and the modular operation, which ensures consistent chaotic performance across the entire parameter space, a property not consistently achieved by existing cascade methods. Building on this foundation, a third technique, the halfway-shifted division, involves dividing one of the two maps by their cumulative total. In the fourth and most complex method, we integrate both product and division techniques, termed Quarterway-Shifted Product Divider maps, where halfway-shifted product maps undergo further division. We assess the efficacy of these newly proposed maps through various methods such as bifurcation plots, both individual and average Lyapunov exponents, and the chaotic ratio. Our analysis demonstrates that these shifted maps sustain a broader and more consistent chaotic window than their original counterparts over the complete parameter spectrum. Remarkably, the Quarterway-Shifted Product Divider maps show the most superior performance.

The remainder of the paper is organized as follows: Section 2 introduces the traditional seed maps, with particular emphasis on the log and tent maps. Section 3 outlines the various performance parameters used to evaluate chaotic behavior. Section 4 presents the newly proposed chaotic systems and analyzes their performance in comparison to the traditional maps. Section 5 details the results obtained from FPGA implementation. In Section 6, the application of the proposed maps in pseudo-random number generation is explored, and their effectiveness is validated using two different statistical test suites. Finally, Section 7 provides the concluding remarks and summarizes the key contributions of the work.

2. Traditional Seed Maps

In this section, we review two well-known one-dimensional chaotic maps: the logistic map and the tent map. These maps are used as seed functions in generating the new chaotic maps discussed in subsequent sections. For the sake of consistency and ease of comparison, we consider the normalized versions of these chaotic maps, where all their domain, range, and parameter values are confined within the interval $[0, 1]$.

The transfer curve illustrates the relationship between the input and output of a map for various values of the control parameter. To understand how the control parameter influences the behavior of a dynamical system, a bifurcation diagram is often used. In this diagram, the steady-state output values corresponding to a range of control parameter values are plotted. This visualization helps in understanding the system's transitions, especially the period-doubling process. As the control parameter is varied, the system progresses from a stable fixed point to a periodic regime and eventually into a chaotic region. To quantify the chaotic nature of the output, the Lyapunov exponent (LE) is commonly used. A positive LE indicates that the system exhibits chaotic behavior. In this context, we explore the characteristics of these two maps by analyzing their transfer curves, bifurcation diagrams, and Lyapunov exponent plots.

2.1. Logistic Map

The logistic map is a classic example of a one-dimensional chaotic map. It is mathematically defined as follows:

$$x_{i+1} = L(x_i) = 4rx_i(1 - x_i) \quad (1)$$

Here, x_i represents the state of the system at time step i , and r is the control parameter that determines the behavior of the system. The parameter r lies within the range $0 \leq r \leq 1$. The logistic map is frequently used due to its simplicity and its ability to exhibit complex and unpredictable behavior under certain conditions.

The equation shows that the next value of x is derived by applying the function $L(x_i)$ to the current state, which involves a nonlinear interaction between the current state and the control parameter. The system's evolution is sensitive to initial conditions, making it a hallmark of chaotic dynamics. The transfer curve, bifurcation, and Lyapunov exponent plots of the log map are shown in Figure 1a–c. As can be seen, the logistic map has chaotic properties at $r \in [0.89, 1]$.

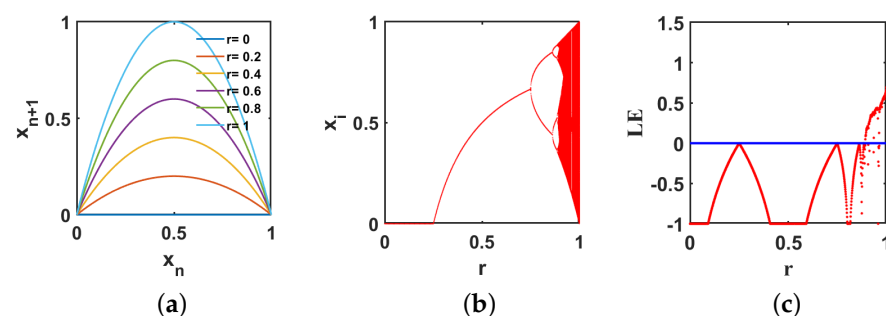


Figure 1. (a) Transfer curve, (b) bifurcation diagram, and (c) LE of logistic map.

2.2. Tent Map

The tent map is another simple yet powerful chaotic map used to generate chaotic behavior in systems. It is mathematically defined as follows:

$$x_{i+1} = T(x_i) = \begin{cases} 2rx_i & \text{when } x_i < 0.5 \\ 2r(1 - x_i) & \text{when } x_i \geq 0.5 \end{cases} \quad (2)$$

In this map, the control parameter r lies in the range $0 \leq r \leq 1$. The behavior of the tent map depends on whether the current state x_i is less than or greater than 0.5. If $x_i < 0.5$, the function behaves as a straight line with slope $2r$, and if $x_i \geq 0.5$, the map flips over and behaves as a descending linear function. The tent map is known for its simplicity and can also produce chaotic sequences, particularly when the control parameter r is large.

The piecewise nature of the tent map leads to distinct behaviors in different regions of the domain, providing an interesting contrast to the continuous logistic map. This map is widely used in studies of chaotic dynamics due to its ease of computation and clear bifurcation patterns. The transfer curve, bifurcation, and Lyapunov exponent plots of the log map are shown in Figure 2a–c. As can be seen, the tent map has chaotic properties at $r \in [0.5, 1]$.

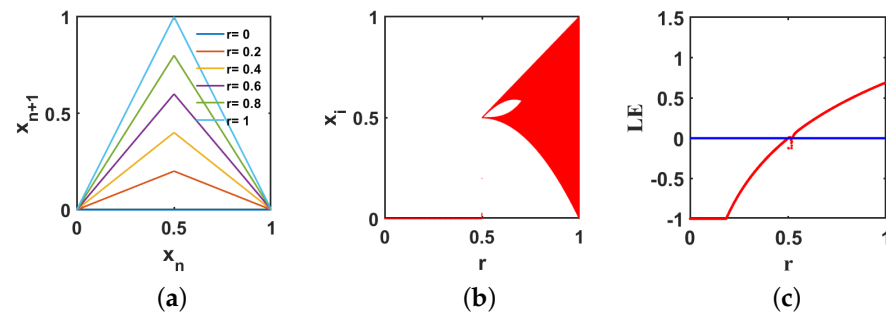


Figure 2. (a) Transfer curve, (b) bifurcation diagram, and (c) LE of tent map.

3. Performance Parameters for Chaotic Maps

To generate a discrete-time sequence, the process begins by initializing an oscillator with an initial value, followed by running the feedback loop iteratively. After discarding a few initial iterations, the steady-state sequence values are recorded. To assess the chaotic behavior of the resulting sequence, two methods are used: (i) a qualitative approach, visually inspecting the sequence through a bifurcation diagram; and (ii) a quantitative evaluation of chaotic entropy using three established metrics—the Lyapunov exponent (LE), correlation coefficient (CC), and Shannon entropy (SE). The performance of the proposed chaotic maps in the sections below are thoroughly evaluated by examining their bifurcation diagrams, Lyapunov exponents, average Lyapunov exponent, chaotic ratio, correlation coefficient, and Shannon entropy.

3.1. Bifurcation Diagram

The bifurcation diagram provides valuable insight into the evolution of output values in a chaotic oscillator as the control parameters are varied. It tracks the system's transition from periodic behavior to chaos as the control parameters change, highlighting the bifurcation points along the way. For instance, Figure 1b and Figure 2b present two-dimensional bifurcation diagrams for the logistic and tent maps, respectively, where regions marked in dark red represent chaotic behavior. In contrast, other parts of the diagram exhibit stable fixed points or periodic oscillations, forming repetitive and predictable patterns. In our study, we used the shifting parameter b as the bifurcation parameter and examined how the output values change with variations in b .

3.2. Lyapunov Exponent

The Lyapunov exponent (LE) is a key measure of the sensitivity of a chaotic system to its initial conditions. In chaotic systems, even small differences in initial conditions can lead to exponentially diverging trajectories over time. The Lyapunov exponent, denoted as λ , serves as a crucial and widely used tool for quantifying this sensitivity. For a discrete-time chaotic system, the Lyapunov exponent is defined as shown in Equation (3):

$$\lambda = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=0}^{n-1} \ln |f'(x_i)| \quad (3)$$

The value of the Lyapunov exponent (λ) is negative when the system exhibits fixed points or periodic orbits. On the other hand, for chaotic attractors, the Lyapunov exponent is positive [2]. For instance, Figure 1c plots the LE of the logistic map and tent map, respectively.

3.3. Correlation Coefficient

A key characteristic of chaotic systems is their extreme sensitivity to small changes in the initial conditions, such as slight variations in the initial state or parameter values. This sensitivity is often quantified using a well-known metric called the correlation coefficient (CC). Equation (4) presents the formula for Pearson's CC, which is used to determine the correlation between two sequences X and Y :

$$C_o = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y} \quad (4)$$

In this equation, $E[\cdot]$ denotes the expectation operator, while μ and σ represent the mean and standard deviation, respectively. If the CC value is close to +1 or −1, it indicates a strong correlation between the sequences X and Y . Conversely, a CC value near 0 suggests that the two sequences are minimally correlated. To assess the dependence of the system on its initial state, two sets of steady-state data sequences are generated from the same chaotic system with a given control parameter, but with slightly different initial conditions. The CC is then calculated using the formula in the equation mentioned. For a chaotic system, even the smallest variation in the initial state results in two vastly different sequences, leading to a CC value close to 0. In contrast, for a non-chaotic system, the sequences will be very similar, and the CC will be close to +1 or −1.

3.4. Shannon Entropy

The Shannon entropy (SE) is a measure commonly used to quantify the uncertainty present in a random process. If the range of values for a signal X is divided into n equally spaced bins, the SE can be expressed as

$$SE = - \sum_{i=1}^n \Pr(x_i) \log_2 \Pr(x_i) \quad (5)$$

In this equation, $\Pr(x_i)$ represents the probability of the signal value falling into the i -th bin. SE serves as a tool to assess the randomness of a discrete-time sequence. In this study, we set $n = 2^{10} = 1024$ and use Equation (5) to calculate the SE for each control parameter over 14,000 steady-state iterations. The theoretical maximum value of SE is $\log_2 n = \log_2 1024 = 10$, which occurs when the sequence values are uniformly distributed over the entire range $[0, 1]$. The SE value increases with the degree of ergodicity in the sequence.

3.5. Chaotic Ratio

In addition to the standard metrics described above, we introduce the chaotic ratio (CR) as a measure specific to the proposed shifted maps. Unlike the seed maps, where chaotic behavior is evaluated over the control parameter r , the proposed maps fix r at $r_{\max}$ and instead vary the shift parameter $b \in [0, 1]$. The CR is therefore defined as the fraction of shift parameter values for which the Lyapunov exponent is strictly positive ($\lambda > 0$):

$$CR = \frac{\sum_b \mathbf{1}[\lambda(b) > 0]}{N_b} \quad (6)$$

where $\mathbf{1}[\cdot]$ is the indicator function, N_b is the total number of shift parameter values evaluated over $b \in [0, 1]$, and $\lambda(b)$ is the Lyapunov exponent computed at each value of b with r fixed at r_{max} . A CR value of 1 indicates that chaotic behavior ($\lambda > 0$) is sustained at every evaluated value of b , while lower values indicate the presence of non-chaotic windows within the parameter range. So, $CR = 1$ does not imply uniform chaotic density across the parameter space—the degree of chaotic spread may still vary across the range of b , with some regions exhibiting a narrower chaotic attractor than others while still maintaining $\lambda > 0$.

4. Proposed Chaotic Systems

This section presents four distinct methods for generating chaotic maps that enhance chaotic behavior and extend the range of chaotic windows. Starting with the logistic and tent maps as seed maps, we apply various transformations to improve their chaotic properties. The first method involves shifting these seed maps to create modified versions with better chaotic behavior. In the second method, we combine the maps by applying a halfway shift followed by multiplication, resulting in a new variant with improved dynamics. The third method introduces division between the shifted and halfway-shifted maps, further refining the chaotic behavior. Finally, the fourth method involves dividing two halfway-shifted product maps from the second method, yielding a more complex and chaotic system. Throughout these transformations, the chaotic ratio and Lyapunov exponent (LE) values exhibit significant improvements, demonstrating the efficacy of the proposed techniques in enhancing the chaotic properties of the maps.

4.1. Shifted Map

In the proposed approach, we consider a one-dimensional chaotic map with a single control parameter, $f(r, x)$, where the evolution is given by $x_{i+1} = f(r, x_i)$, and the map's domain is $[l, h]$. The goal is to determine the value of r , referred to as r_{max} , for which the Lyapunov exponent reaches its maximum. Using this value of r_{max} , we define the shifted one-dimensional map with a parameter b as follows:

$$sh_f(b, x) = f(r_{max}, ((x - b - l) \bmod (h - l)) + l) \quad (7)$$

The domain of the map remains $[l, h]$, and the range of the shifted parameter is $[0, h - l]$. In this work, the seed maps are normalized such that $l = 0$ and $h = 1$, so the general expression $((x - b - l) \bmod (h - l)) + l$ simplifies to $(x - b) \bmod 1$. Since the modular operation $(x - b) \bmod 1$ is a bijection on the domain $[l, h]$, the derivative of the shifted map with respect to x satisfies

$$\frac{\partial}{\partial x} sh_f(b, x) = f'(r_{max}, (x - b) \bmod 1) \quad (8)$$

The derivative structure of the shifted map is therefore identical to that of the original seed map evaluated at r_{max} , regardless of the value of b . For maps with a uniform derivative structure, such as the tent map, this results in a constant LE across the entire range of b , as confirmed by the shifted tent map results, where $Max_LE = Avg_LE = 0.6931$ and $CR = 1$. For maps with more complex derivative profiles, such as the logistic map, the LE remains near $\lambda_f(r_{max})$ across a significantly broader range of b compared to the original unshifted map, resulting in a wider chaotic window.

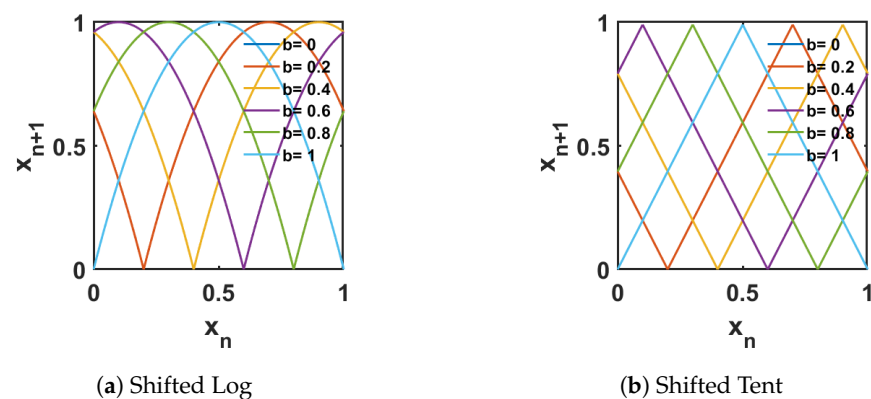
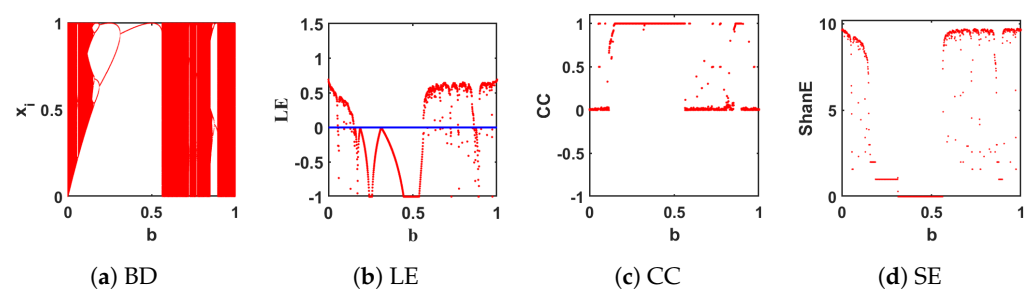
In this work, we consider two simple normalized seed maps: the logistic map $L(r, x)$ and the tent map $T(r, x)$, which are summarized in Table 1.

Table 1. Mathematical expressions for the two shifted maps.

Map Name	Mathematical Expression
Shifted logistic	$L_s(b, x) = L(1, (x - b) \bmod 1)$
Shifted tent	$T_s(b, x) = T(1, (x - b) \bmod 1)$

The maximum Lyapunov exponent (LE) values are consistently observed at a control parameter value $(C_L, C_T) = 1$ for all two maps. Subsequently, by applying a shift determined by the parameter b , and varying b , we obtain the final shifted versions of these maps. The mathematical expressions for the shifted maps are provided in Table 1.

Figure 3a,b illustrates the transfer curves of the shifted maps. The $b = 0$ and $b = 1$ curves are identical due to the periodicity of the shift, so the $b = 1$ curve visually overlaps and hides the $b = 0$ curve in Figure 3. Figures 4a–d and 5a–d present the performance parameters of the shifted logistic and shifted tent maps, respectively, including the 2-D bifurcation diagrams (BDs), Lyapunov exponent (LE), correlation coefficient (CC), and Shannon entropy (SE), where each parameter is evaluated by varying the control parameter b . In the BD plots, the introduction of shifted dynamics significantly enlarges the dark red regions, indicating a substantial increase in chaotic behavior compared to the original seed maps. The LE plots reveal that for the shifted logistic map, LE values are generally positive across a wide range of b , while the shifted tent map consistently achieves its maximum positive LE values throughout, signifying strong chaotic dynamics. The CC plots and SE plots for both shifted maps further validate these findings; they show near-zero values across a broad parameter range, confirming enhanced chaotic behavior. These observations are consistent with the corresponding BD and LE plots and demonstrate that the shifted maps exhibit chaotic operation over a much wider range than their original counterparts.

**Figure 3.** Transfer curves of different shifted maps.**Figure 4.** Performance parameters of shifted log.

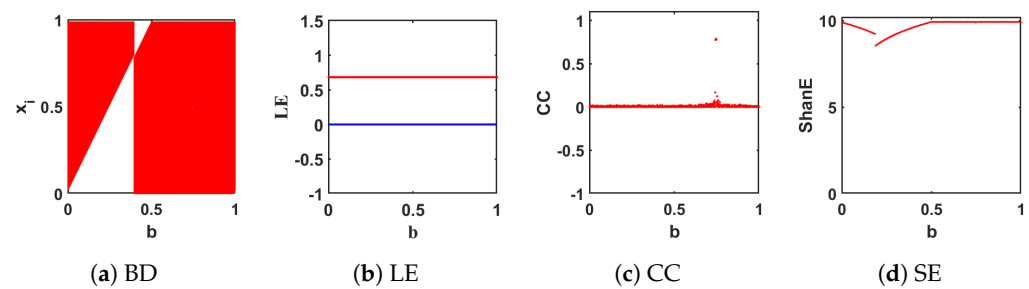


Figure 5. Performance parameters of shifted tent.

4.2. Halfway-Shifted Product Maps

Shifting the seed maps leads to improvements in chaotic behavior compared to the original maps. However, the desired chaotic properties, such as achieving the highest Lyapunov exponent (LE) values across the entire window, are not yet fully realized. To address this, we introduce an additional enhancement in the proposed method by performing a halfway shift on each map, which further contributes to the development of a more robust chaotic map.

In this approach, we first apply a halfway shift to one of the seed maps (either the logistic or tent map). Then, we combine the resulting map with one of the previously shifted maps. The product of the halfway-shifted map and the shifted map is calculated as follows:

If $w = h - l = 1$, then

$$f_s g_{h_s}(r_1, r_2, b, x) = \frac{1}{A} \cdot f_s(r_1, b, x) \cdot g_s(r_2, b + \frac{w}{2}, x) \quad (9)$$

Here, A is a constant value, r_1 and r_2 are dependent on which maps we are using. For log maps, we know the highest Lyapunov exponent is found at $r_1 = 1$ and for tent maps $r_2 = 0.99$. Or we can simply write it as follows:

$$\text{HSP}_{fg}(b, x) = \frac{1}{A} \cdot f_s(b, x) \cdot g_s(b + \frac{w}{2}, x) \quad (10)$$

Applying the product rule, the derivative of HSP_{fg} with respect to x is

$$\frac{\partial}{\partial x} \text{HSP}_{fg}(b, x) = \frac{1}{A} \left[f'_s(b, x) \cdot g_s\left(b + \frac{1}{2}, x\right) + f_s(b, x) \cdot g'_s\left(b + \frac{1}{2}, x\right) \right] \quad (11)$$

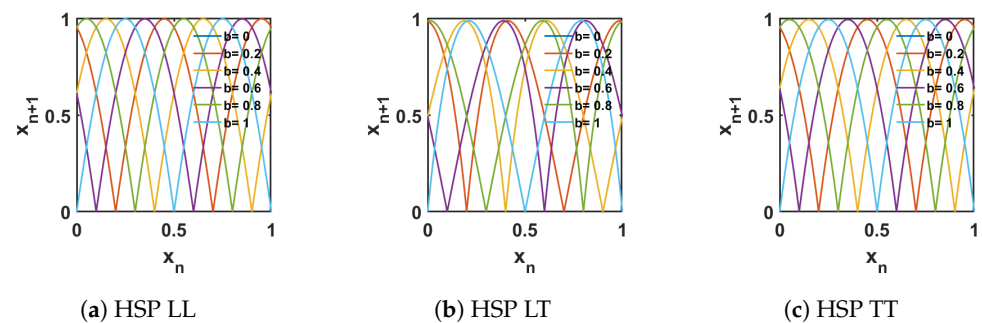
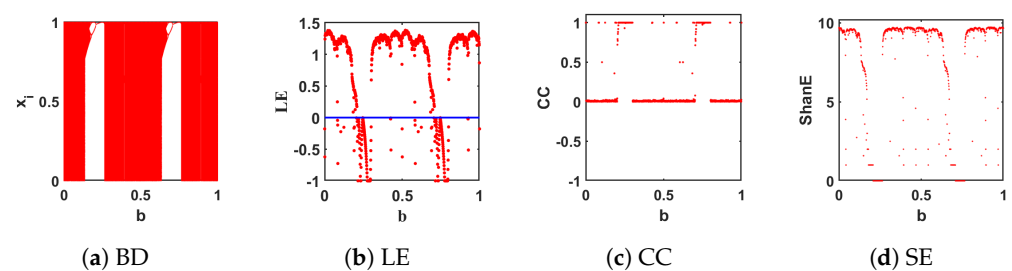
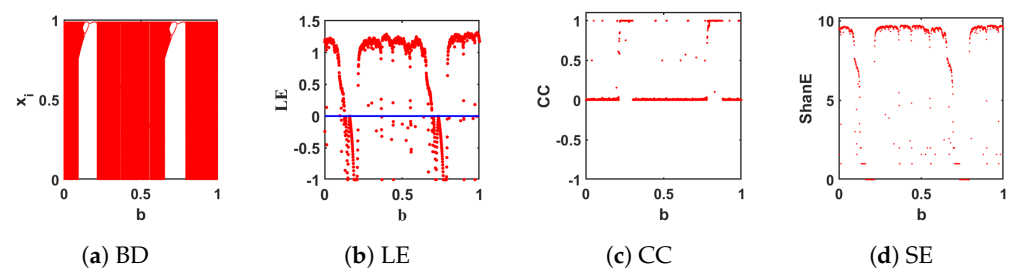
Since both f_s and g_s are normalized maps with outputs in $[0, 1]$ and both operate at $r_{\max}$, the cross-terms in the product rule consistently amplify the derivative magnitude compared to either individual shifted map alone. This amplification is reflected in the significantly higher LE values observed in the HSP maps (approximately 1.31–1.34) compared to the individual shifted maps (approximately 0.69). The half-period offset of $\frac{1}{2}$ between the two component maps ensures complementary phase coverage across the parameter space, maximizing this amplification effect. It should be noted however that the product operation introduces a trade-off: while the maximum LE is substantially increased, the chaotic ratio decreases to approximately $CR = 0.77$ – 0.78 , which motivates the further enhancements discussed next.

This results in four possible combinations. Three notable combinations, namely, $L_s L_{h_s}$, $L_s T_{h_s}$, and $T_s T_{h_s}$, are highlighted. The mathematical expressions for these halfway-shifted product maps are provided in Table 2.

Table 2. Mathematical expressions for the three halfway-shifted product maps.

Map Name	Mathematical Expression
Halfway-shifted product LL	$\text{HSP}_{LL}(b, x) = (1/A) * L_s(b, x) * L_s(b + 1/2, x); A = 0.5625$
Halfway-shifted product LT	$\text{HSP}_{LT}(b, x) = (1/A) * L_s(b, x) * T_s(b + 1/2, x); A = 0.385$
Halfway-shifted product TT	$\text{HSP}_{TT}(b, x) : (1/A) * T_s(b, x) * T_s(b + 1/2, x); A = 0.246$

Figure 6a–c shows the transfer curves of the halfway-shifted product (HSP) maps. As with the shifted maps, the $b = 0$ and $b = 1$ curves coincide, so the $b = 1$ curve overlaps and hides the $b = 0$ curve in Figure 6. Figures 7a–d, 8a–d and 9a–d present the performance results of three HSP variants: log-log (HSP LL), log-tent (HSP LT), and tent-tent (HSP TT). Each figure includes a 2-D bifurcation diagram (BD), Lyapunov exponent (LE), correlation coefficient (CC), and Shannon entropy (SE) plotted against the control parameter b . In the BD plots, we can see that the chaotic regions are significantly wider than in the original and shifted maps, indicating stronger chaotic behavior. The LE plots show mostly positive values for all three variants, meaning that the systems remain chaotic over a large range of b . The CC plots are close to zero across wide intervals, and the SE plots show high entropy values—both signs of unpredictable and complex behavior. Altogether, these results confirm that the HSP maps produce more consistent and stronger chaos than the earlier versions, making them more suitable for applications requiring high randomness and complexity.

**Figure 6.** Transfer curves of different halfway-shifted product (HSP) maps.**Figure 7.** Performance parameters of HSP LL.**Figure 8.** Performance parameters of HSP LT.

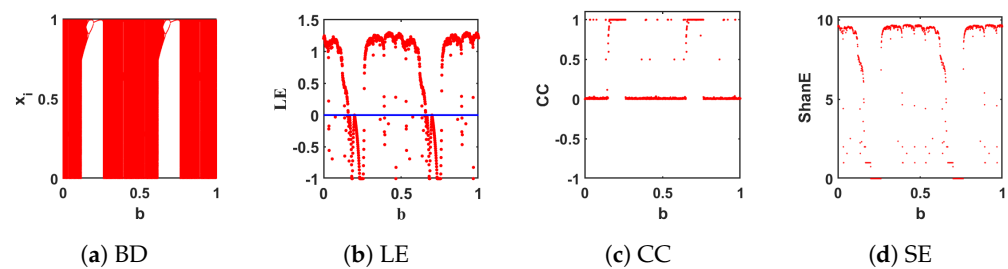


Figure 9. Performance parameters of HSP TT.

4.3. Halfway-Shifted Division

The chaotic behavior improves significantly with the halfway-shifted product method compared to the previously discussed maps, and the chaotic ratio also increases. However, chaos still does not persist across the entire parameter window. To achieve this, we introduce another enhancement to our previous method by incorporating the division approach between the maps.

In this step, we combine one shifted map and one halfway-shifted map and perform division between them, as follows:

If $w_1 = f_s(r_1, b, x)$ and $w_2 = g_s(r_2, b + w, x)$, then

$$\text{HSD}_{fg}(r_1, r_2, b, x) = \frac{w_1}{w_1 + w_2} \quad (12)$$

The normalization structure $\frac{w_1}{w_1 + w_2}$ ensures the output is always bounded in $(0, 1)$. The derivative of the HSD map with respect to x takes the form:

$$\frac{\partial}{\partial x} \text{HSD}_{fg} = \frac{w'_1 w_2 - w_1 w'_2}{(w_1 + w_2)^2} \quad (13)$$

where $w'_1 = \frac{\partial w_1}{\partial x}$ and $w'_2 = \frac{\partial w_2}{\partial x}$. The cross-derivative structure $w'_1 w_2 - w_1 w'_2$ sustains non-zero sensitivity to initial conditions across the full range of b . The half-period offset between w_1 and w_2 ensures that when one component is in a periodic region of its parameter space, the other is in a complementary chaotic region, and the division operation blends their outputs in a way that eliminates periodic windows—which is why all the HSD variants consistently achieve a perfect chaotic ratio ($CR = 1$) across the entire parameter space.

This results in four different combinations: HSD_{LL} (halfway-shifted division log-log), HSD_{LT} (halfway-shifted division log-tent), HSD_{TL} (halfway-shifted division tent-log), and HSD_{TT} (halfway-shifted division tent-tent). For completeness, we note that each pair corresponds to a distinct ordering of the seed maps within the division structure, thereby producing slightly different dynamical responses while preserving the overall construction. The corresponding mathematical expressions for these halfway-shifted division maps are provided in Table 3, which lists the exact functional forms used in our analysis.

Table 3. Mathematical expressions for the halfway-shifted division maps.

Map Name	Mathematical Expression
Halfway-shifted division LL	$\text{HSD}_{LL}(b, x) = L_s(1, b, x) / (L_s(1, b, x) + L_s(1, b + 0.5, x))$
Halfway-shifted division TL	$\text{HSD}_{TL}(b, x) = T_s(0.99, b, x) / (T_s(0.99, b, x) + L_s(0.99, b + 0.5, x))$
Halfway-shifted division TT	$\text{HSD}_{TT}(b, x) = T_s(0.99, b, x) / (T_s(0.99, b, x) + T_s(0.99, b + 0.5, x))$

Figure 10a–c illustrates the transfer curves of the halfway-shifted division (HSD) maps. The same periodicity applies here, so the $b = 0$ curve is hidden beneath the overlapping $b = 1$ curve in Figure 10. Figures 11a–d, 12a–d and 13a–d display the performance parameters for three HSD variants: log-log (HSD LL), tent-log (HSD TL), and tent-tent

(HSD TT). These figures include the 2-D bifurcation diagram (BD), Lyapunov exponent (LE), correlation coefficient (CC), and Shannon entropy (SE), all plotted with respect to the control parameter b . The BD plots show a remarkably large spread of chaotic regions compared to not only the original and shifted maps but also the halfway-shifted product (HSP) maps. The LE plots maintain high positive values across almost the entire range of b , indicating strong and sustained chaotic behavior. In addition, the CC plots hover close to zero, and the SE plots remain consistently high, both pointing to enhanced unpredictability and information richness. These combined results demonstrate that the HSD maps deliver superior chaotic performance across all evaluated metrics, surpassing the behavior of earlier designs and making them highly effective for chaos-based applications that demand strong randomness and complex dynamics.

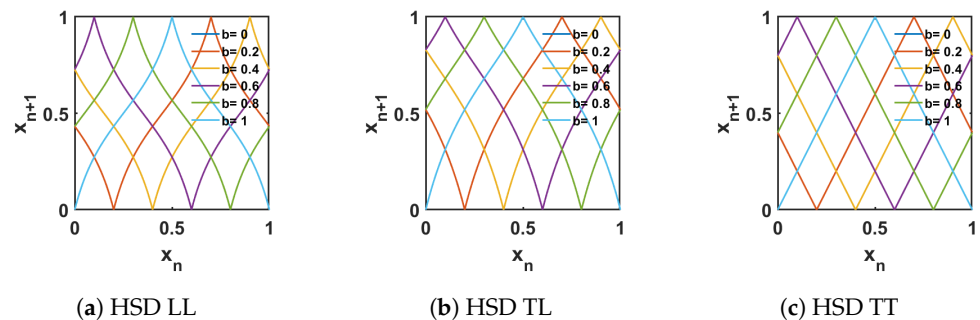


Figure 10. Transfer curves of different halfway-shifted division (HSD) maps.

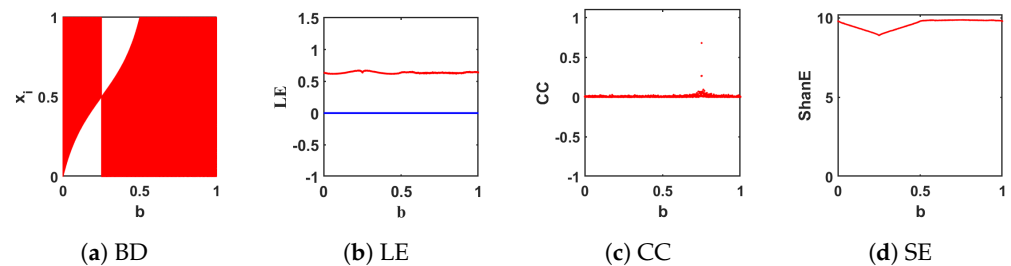


Figure 11. Performance parameters of HSD LL.

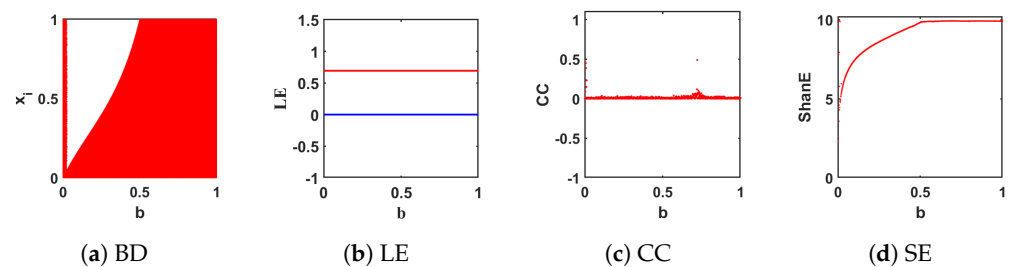


Figure 12. Performance parameters of HSD TL.

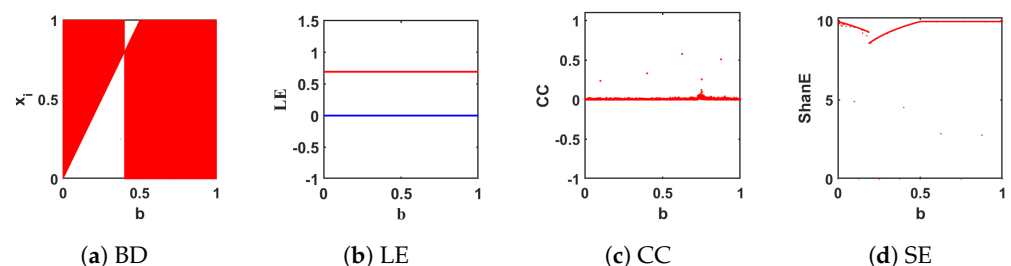


Figure 13. Performance parameters of HSD TT.

4.4. Combined Product Divider

In the halfway shifted division maps, the desired chaotic ratio value 1 is achieved but in the process the maximum LE value decreases to around 0.6709 for the halfway-shifted

division log-log. Moreover, the highest LE is not achieved throughout the whole window, which can be realized from the average LE values. So a third type of enhancement is introduced. Initially, we consider two halfway-shifted products, and then apply the division operation between them. This approach results in 16 possible combinations, as there are four options for each of the two products, leading to a total of $4 \times 4 = 16$ combinations.

Let the halfway-shifted product of functions f and g be defined as

$$\text{HSP}_{fg}(r_1, r_2, b, x, A_1) = \frac{1}{A_1} \cdot f_s(r_1, b, x) \cdot g_s(r_2, b + \frac{1}{2}, x) \quad (14)$$

Similarly, the halfway-shifted product of functions h and k is defined as

$$\text{HSP}_{hk}(r_3, r_4, b, x, A_2) = \frac{1}{A_2} \cdot h_s(r_3, b, x) \cdot k_s(r_4, b + \frac{1}{2}, x) \quad (15)$$

Next, we define the combined product divider (CPD) as follows:

$$\text{CPD}_{fg_hk}(b, x) = \frac{w_1}{w_1 + w_2} \quad (16)$$

where $w_1 = \text{HSP}_{fg}(r_1, r_2, b, x, A_1)$ and $w_2 = \text{HSP}_{hk}(r_3, r_4, b + \frac{1}{4}, x)$.

When the same map combination is used for both w_1 and w_2 (i.e., $fg = hk$), the scaling factors A_1 and A_2 cancel in the division:

$$\text{CPD}(b, x) = \frac{\frac{1}{A} \cdot P(b, x)}{\frac{1}{A} \cdot P(b, x) + \frac{1}{A} \cdot P(b + \frac{1}{4}, x)} = \frac{P(b, x)}{P(b, x) + P(b + \frac{1}{4}, x)} \quad (17)$$

where $P(b, x) = f_s(b, x) \cdot g_s(b + \frac{1}{2}, x)$ is the unnormalized product, showing that the CPD map's chaotic properties are independent of the choice of scaling factor A . Furthermore, the quarterway shift of $\frac{1}{4}$ applied to w_2 is motivated by the frequency-doubling effect of the product operation: since the HSP map has twice the oscillation frequency of the individual shifted maps (as seen in the transfer curves in Figure 14), a quarterway shift achieves the same complementary half-period offset between w_1 and w_2 that the halfway shift achieves between the component maps in the HSP. This ensures that the division in CPD benefits from the same periodic-window elimination mechanism as the HSD maps, while simultaneously inheriting the high LE amplification from the HSP structure.

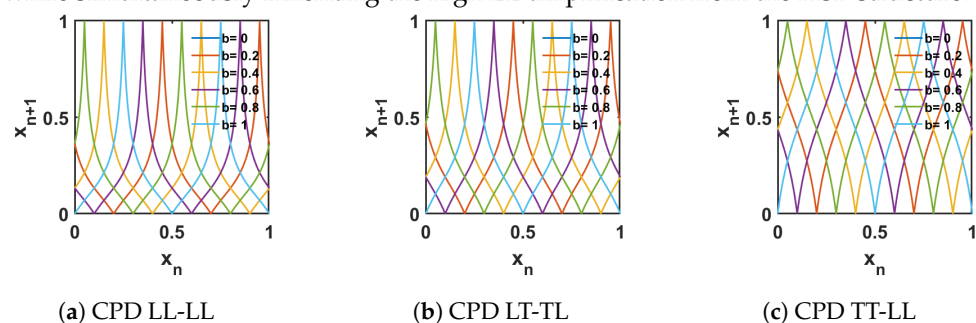


Figure 14. Transfer curves of different combined product division (CPD) maps.

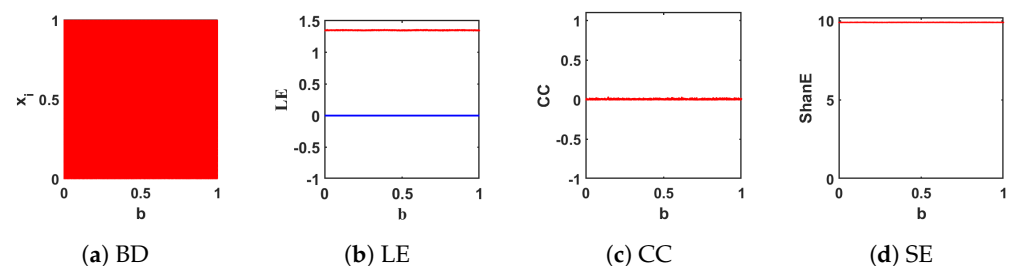
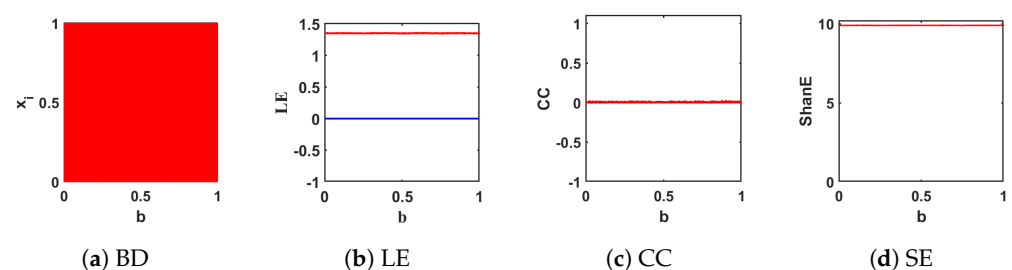
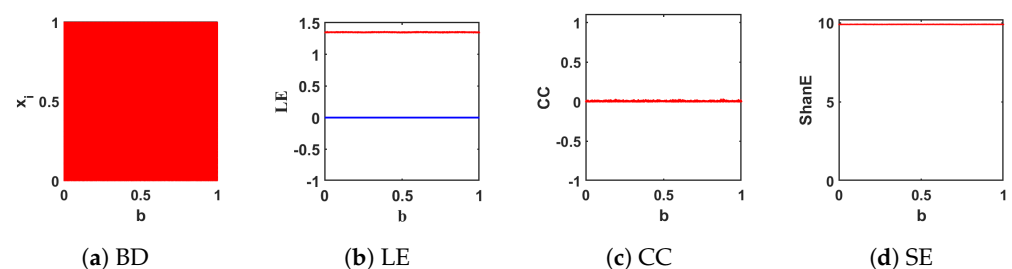
The scaling factors A_1 and A_2 can be chosen as they were for the HSP. However, even when $A_1 = A_2 = 1$, the system should function correctly, and we can compare the global metrics across the different combinations. The corresponding mathematical expressions for some of these combined product division maps are provided in Table 4.

$$\text{CPD}_{LL}(b, x) = \frac{\text{HSP}_{LL}(b, x)}{\text{HSP}_{LL}(b, x) + \text{HSP}_{LL}(b + 0.25, x)} \quad (18)$$

Table 4. Mathematical expressions for the combined product division maps.

Map Name	Mathematical Expression
Combined product division LL-LL	$CPD_{LL-LL}(b, x) = \frac{HSP_{LL}(b, x)}{HSP_{LL}(b, x) + HSP_{LL}(b + 0.25, x)}$
Combined product division LT-TL	$CPD_{LL-TT}(b, x) = \frac{HSP_{LT}(b, x)}{HSP_{LT}(b, x) + HSP_{TT}(b + 0.25, x)}$
Combined product division TT-LL	$CPD_{TT-LL}(b, x) = \frac{HSP_{TT}(b, x)}{HSP_{TL}(b, x) + HSP_{LL}(b + 0.25, x)}$

Figure 14a–c illustrates the transfer curves of the combined product division (CPD) maps. As before, the $b = 0$ curve coincides with and is hidden by the $b = 1$ curve in Figure 14 due to the same shift periodicity. Figures 15a–d, 16a–d and 17a–d show the performance characteristics of three CPD variants: log-log (CPD LL-LL), log-tent to tent-log (CPD LT-TL), and tent-tent to log-log (CPD TT-LL). Each figure presents the 2-D bifurcation diagram (BD), Lyapunov exponent (LE), correlation coefficient (CC), and Shannon entropy (SE), plotted against the control parameter b . Among all the proposed systems, the CPD maps exhibit the most ideal and robust chaotic behavior. The BD plots are fully saturated, showing an uninterrupted and dense chaotic regime across the entire parameter space. The LE plots maintain uniformly high positive values throughout, confirming strong and persistent sensitivity to initial conditions. The CC values remain close to zero across the full b -range, and the SE values are consistently at their maximum, indicating perfect unpredictability and information complexity. These combined observations establish CPD maps as the most reliable and effective chaotic systems among all the proposed designs, ideal for high-security and high-randomness applications.

**Figure 15.** Performance parameters of CPD LL-LL.**Figure 16.** Performance parameters of CPD LT-TL.**Figure 17.** Performance parameters of CPD TT-LL.

The quantitative analysis in Table 5 reveals a clear evolution in chaotic map performance through successive enhancements. Beginning with the base maps, the shifted tent demonstrates ideal chaotic consistency ($CR = 1$) with Lyapunov exponents (LE s) of 0.6931, outperforming the shifted logistic map's lower average LE (0.5069) and partial chaos ($CR = 0.5145$). The halfway-shifted product maps achieve dramatic LE amplification (1.3178–1.3342) at the cost of reduced chaotic consistency ($CR = 0.7702$ – 0.7842), with HSP LT reaching the highest max LE (1.3342). This trade-off is resolved in halfway-shifted division maps, which restore perfect CR (1) while maintaining moderate LE values (0.6709–0.6931), particularly in the HSD TT variant, matching the shifted tent's ideal CR with improved avg LE (0.6931). The ultimate enhancement comes from the combined product division maps, which simultaneously deliver both the highest chaotic intensity (max LE up to 1.3567 in CPD LL-LL) and perfect consistency ($CR = 1$), representing a 96% max LE increase and 166% avg LE improvement over the base maps.

Table 5. Performance parameter values for different chaotic maps.

Parameter	Sh L	Sh T	HSP LL	HSP LT	HSP TT	HSD LL	HSD LT	HSD TT	CPD LL-LL	CPD TT-LL
Max LE	0.6918	0.6931	1.3178	1.3342	1.3221	0.6709	0.6931	0.6931	1.3567	1.3324
Avg LE	0.5069	0.6931	1.1068	1.0910	1.0770	0.6364	0.6031	0.6931	1.3498	1.3200
CR	0.5145	1	0.7842	0.7842	0.7702	1	1	1	1	1

To place the proposed maps in the context of existing chaos enhancement methods, Table 6 compares the performance of the best proposed variant, CPD LL-LL, against representative existing approaches using three key metrics: maximum LE , average LE , and chaotic ratio. The seed maps (logistic and tent) serve as the baseline, while the cascade chaotic system (ccs) [10] and the dynamic parameter-control chaotic system (DPCCS) [3] represent well-established chaos enhancement methods, and the enhanced flipped-product chaotic system (EFPCS) [16] represents a recent study in the field.

Table 6. Performance comparison with existing chaotic map methods.

Method	Max LE	Avg LE	CR
Logistic map	0.694	0.388	0.098
Tent map	0.693	0.393	0.49
CCS [10]	1.27	0.788	0.143
DPCCS [3]	0.683	0.45	1
EFPCS [16]	1.326	0.949	1
CPD LL-LL (proposed)	1.3567	1.3498	1

Data for logistic map, tent map, CCS, and DPCCS are taken from [16].

Table 6 compares the proposed CPD LL-LL map against representative existing approaches using maximum LE , average LE , chaotic ratio, and FPGA realizability. The results demonstrate that the proposed CPD maps achieve the highest maximum LE and average LE among the compared methods while simultaneously maintaining a perfect chaotic ratio ($CR = 100\%$) and FPGA hardware validation, confirming superior overall chaotic performance.

5. FPGA Implementation Results

To validate the hardware feasibility of the proposed chaotic maps, we implemented the designs on a Nexys A7 FPGA board and executed the full evaluation flow on hardware. The Lyapunov exponent (LE), used as the primary metric for chaotic behavior, was measured across 21 bifurcation-parameter values, with the parameter sweeping from 0.00025 to 1 in non-uniform steps.

In the FPGA implementation, a fixed-point format with $n + 1$ binary bits ($n : 0$) is used, where the binary point is placed after the n th bit. Choosing $n = 64$ yields a uniform high resolution of $\left(\frac{1}{2^{64}}\right)$ across the entire range. This format reduces overhead compared to the IEEE-754 floating-point standard, which supports a significantly wider range $([-2 \times 2^{1023}, +2 \times 2^{1023}])$ but achieves a resolution of $\left(\frac{1}{2^{52}}\right)$ due to its 52-bit mantissa [35].

The core Verilog implementation of the shifted tent map is presented in Listing 1. The module is parameterized with a 64-bit fixed-point representation ($\text{BIT} = 64$), where the integer value 1 is encoded as a left-shifted constant ($65'b1 \ll \text{BIT}$). The shift operation is realized through fixed-point arithmetic: first, 1 is added to the input state x , and then the shift parameter b is subtracted, effectively implementing the modular shift $(x - b) \bmod 1$. The modular wrap-around is achieved implicitly through bit truncation, where only the lower BIT bits of the result are retained (lines 40–41), naturally enforcing the $[0, 1)$ domain boundary without requiring explicit conditional logic. The control parameter r is set to its fixed-point equivalent of 0.99, which corresponds to the maximum Lyapunov exponent condition for the tent map. Finally, the shifted state is passed to the tent map submodule, which computes the next state x_n , with multiplication by 4 handled internally within the submodule.

Listing 1: Core Verilog implementation of the Shifted Tent map.

```

1 module shifted_tent #(parameter BIT=~64) // Module parametrization
2 (
3     // Inputs
4     input  [BIT : 0] x,
5     input  [BIT : 0] b,
6     // Outputs
7     output [BIT : 0] x_n
8 );
9
10 reg  [BIT      : 0] one=~65'b1<<BIT;
11 wire [BIT+1    : 0] one_plus_x;
12 wire [BIT+1    : 0] m;
13 wire [BIT      : 0] m_final;
14 reg  [BIT      : 0] r=~65'b
15     b11111101011100001010001110101110000101000111101011100000000
16     00000; // fixed-point representation of r=~0.99
17
18 assign one_plus_x=~one + x;
19 assign m=~one_plus_x - b;
20 assign m_final[BIT - 1 : 0]~=~m[BIT - 1 : 0]; // bit truncation
21     for mod operation
22 assign m_final[BIT]~=~1'b0;
23
24 tent u1(m_final, r, x_n);
25 // this includes multiplication by 4
26 endmodule

```

For reference, the comparative results against a MATLAB simulation are shown in Figures 18a,b and 19a–c, where the close agreement between the hardware and software confirms the practicality of the proposed implementations.

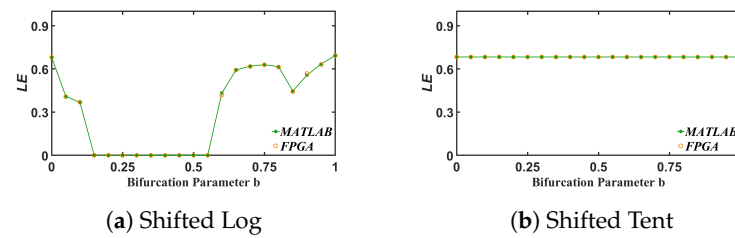


Figure 18. FPGA plot of different shifted maps.

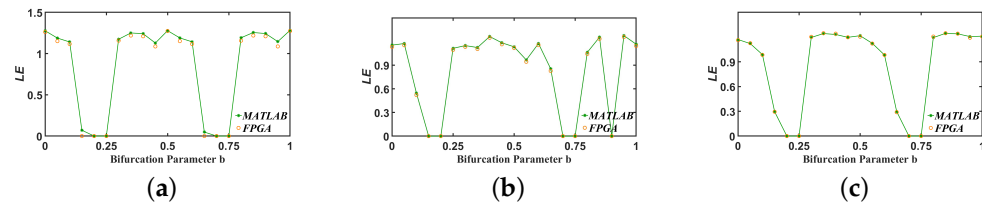


Figure 19. FPGA plot of different halfway-shifted product maps. (a) Halfway shifted product LL. (b) Halfway shifted product LT. (c) Halfway shifted product TT.

For the shifted maps, the LE values obtained from the MATLAB simulations and FPGA implementations match exactly. A similar level of agreement is observed for the halfway-shifted product map T-T. However, small discrepancies appear in the LE values of the other two halfway-shifted maps, LL and LT. These differences are primarily due to the numerical representation: the FPGA implementation relies on a fixed-point system, while the MATLAB simulations use the 64-bit IEEE-754 floating-point format [36]. Although the observed differences are minimal, they are amplified by the inherent sensitivity of chaotic systems to small perturbations, often resulting in divergent behavior over time. However, these variations are not considered critical, as neither implementation is inherently superior in accuracy. In fact, the FPGA implementation provides even higher resolution within the intended range, reinforcing the practicality and robustness of the proposed maps for high-resolution, hardware-based application.

The hardware cost of implementing the proposed chaotic maps using a 64-bit floating-point representation on the Nexys A7 FPGA board is evaluated through a detailed resource-utilization analysis in Tables 7–9. In particular, we account for core components such as slice look-up tables (LUTs), logic LUTs, slice registers, DSP blocks, memory LUTs, and bonded IOBs, together with power metrics that include both total on-chip power and dynamic power consumption. Taken together, these tables provide a comprehensive summary of resource and power usage across all the proposed chaotic map variants, offering a clear view of the implementation cost and its distribution among the major architectural elements.

Table 7. FPGA resource utilization summary for shifted maps and halfway-shifted product maps.

Logic Utilizing	Sh L	Sh T	HSP LL	HSP LT	HSP TT
Slice registers (126,800)	505	505	505	505	505
Slice LUTs (63,400)	7062	7228	16,340	20,572	20,557
DSPs (240)	27	36	72	81	90
Memory LUTs (19,000)	8	8	8	8	8
Logic LUTs (63,400)	7054	7220	16,332	20,564	20,549
Bonded IOB (210)	5	5	5	5	5

Table 8. FPGA resource utilization summary for halfway-shifted division maps and combined product divider maps.

Logic Utilizing	HSD LL	HSD LT	HSD TT	CPD LL-LL	CPD TT-LL
Slice registers (126,800)	505	505	505	509	509
Slice LUTs (63,400)	24,989	24,570	25,040	28,585	45,146
DSPs (240)	54	63	72	72	162
Memory LUTs (19,000)	8	8	8	8	8
Logic LUTs (63,400)	24,981	24,562	25,032	28,577	45,138
Bonded IOB (210)	5	5	5	5	5

Table 9. Power utilization summary for different chaotic maps.

Power Utilizing	Sh L	Sh T	HSP LL	HSP LT	HSP TT	HSD LL	HSD LT	HSD TT	CPD LLLL	CPD TTLL
Total on-chip power(W)	0.336	0.349	0.603	0.664	0.672	0.885	0.965	0.938	0.811	1.483
Dynamic power(W)	0.239	0.251	0.504	0.565	0.573	0.785	0.865	0.838	0.712	1.381

The results clearly indicate that for each map—whether shifted, product-based, division-based, or hybrid—the utilization of logic and memory resources remains well below the maximum capacity of the target FPGA device. For instance, slice LUT usage remains under 72% even for the most complex combined product division (CPD) maps, and DSP block usage is modest across all implementations. Furthermore, power consumption also remains within acceptable limits, with dynamic power peaking at 1.381 W for the CPD TT-LL map, which is still within operational thresholds for modern low-power FPGA platforms. These findings underscore the efficiency and scalability of the proposed designs. Despite the increased chaotic complexity introduced by the shifted and cascaded map structures, the resulting hardware implementations remain lightweight.

6. Application: Pseudo-Random Number Generator

High-quality pseudo-random number generators (PRNGs) are crucial components in information security and cryptographic applications [37,38]. The deterministic aperiodicity and sensitive dependence on initial conditions inherent in chaotic systems have been widely exploited to generate sequences that exhibit apparent randomness [31,39,40]. Owing to their enhanced chaotic characteristics, the proposed shifted, HSP (halfway-shifted product), HSD (halfway-shifted division), and CPD (combined product division) maps offer promising avenues for constructing high-quality PRNGs. In this work, we present a novel PRNG design leveraging HSD and CPD maps. As a representative example, we demonstrate the results using the tent-tent map configuration; however, the general structure can be readily adapted to any of the proposed maps.

A schematic of the proposed PRNG is presented in Figure 20. It consists of two parallel chaotic oscillators: one based on the HSD tent-tent map and the other on the CPD tent-tent map. At each iteration, the 64-bit outputs from both oscillators are truncated to retain the least significant 10 bits, which are subsequently XORed to produce the final 10-bit output, achieving a throughput of 10 bits per iteration. The XOR operation combines the two independent chaotic streams to reduce any residual patterns in the individual outputs and improve the overall randomness of the final sequence. The bitstreams were generated through MATLAB simulation by iterating the mathematical expressions of the proposed maps and extracting the resulting numerical sequences for statistical evaluation. The effectiveness of the proposed PRNG was validated through two standard statistical randomness test suites, namely, NIST and TestU01.

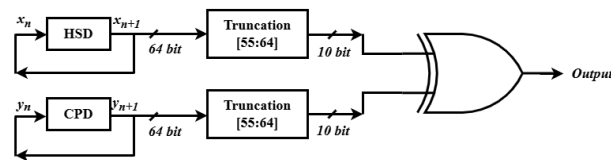


Figure 20. Schematic of the proposed PRNG.

6.1. NIST SP 800-22 Test Suite

The NIST Special Publication 800–22 test suite provides 15 statistical sub-tests designed to evaluate the randomness of binary sequences [41]. In our analysis, we generated 100 independent bitstreams, each initiated from a distinct initial condition, with every bitstream containing 1 million bits. The significance level was set at 0.01. According to the NIST criteria, a sequence comprising 100 million bits (i.e., 100 bitstreams) passes a given sub-test if at least 96 out of the 100 streams produce p -values greater than 0.01.

Additionally, the suite segments the 100 p -values into 10 equally spaced intervals between 0 and 1, and evaluates the uniformity of their distribution using a chi-squared (χ^2) test. A sequence is considered uniformly distributed if the p -value resulting from the χ^2 test (denoted as p_{value_T}) is greater than or equal to 0.0001. The NIST test results, summarized in Table 10, indicate that both the HSD and CPD sequences generated using the TTF-based PRNG successfully pass all 15 sub-tests.

Table 10. NIST test results and p -values (* shows average of multiple tests; ** sample Size is 65 binary sequences).

NIST TEST	Pass Rate (%)	p -Value
Frequency	99	0.43
Block frequency	99	0.88
Cumulative sums *	100	0.77
Runs	98	0.75
Longest runs of ones	100	0.79
Rank	100	0.28
FFT	99	0.19
Non-overlapping template *	99.53	0.46
Overlapping template	100	0.16
Universal	100	0.88
Approximate entropy	97	0.92
Random excursion **	64.62	0.42
Random excursion variant **	64.72	0.28
Serial *	100	0.51
Linear complexity	99	0.35

6.2. TestU01

TestU01 is a widely adopted library for empirical statistical testing, implemented in ANSI C [42]. For this evaluation, we employed three standard test batteries: Rabbit, Alphabit, and BlockAlphabit. The tests were conducted on sequences with bit lengths of 2^{20} and 2^{30} .

The Rabbit battery includes 38 sub-tests for 2^{20} and 40 sub-tests for 2^{30} , while Alphabit consists of 17 sub-tests. BlockAlphabit extends Alphabit by applying the same 17 tests across six blocks, totaling 102 sub-tests. A sequence is deemed to pass a sub-test if the resulting p -value falls within the interval [0.001, 0.999].

Table 11 summarizes the pass rates for each test battery across both sequence lengths. The results confirm the strong statistical quality of the bitstreams generated using HSD and CPD with tent-tent-based PRNGs.

Table 11. TestU01 results.

Bit Length	Rabbit	Alphabit	Block Alphabit
2^{20}	38/38	17/17	102/102
2^{30}	39/40	17/17	101/102

7. Conclusions

This study introduces an innovative approach to enhancing the robustness of chaotic systems using the logistic and tent maps as representative seed maps. The proposed methods involve strategically shifting the seed maps to improve chaotic performance, followed by multiplication and division operations to further augment the system's behavior. The chaotic characteristics of the system were evaluated using bifurcation plots, Lyapunov exponent analyses, correlation coefficient, Shannon entropy, average Lyapunov exponent values, and chaotic ratios. The results demonstrate that the proposed system successfully maintains a continuous chaotic window across the entire operational range, unlike the individual seed maps, which exhibit limited chaotic windows. Moreover, the introduction of halfway-shifted product maps, such as logistic-logistic, logistic-tent, and tent-tent combinations, yields noticeable improvements in the system's performance.

Building on this foundation, the halfway-shifted division maps offer even better performance by increasing the chaotic ratio and reducing structural symmetry, resulting in stronger chaotic properties. Furthermore, the combined product division maps demonstrate the most significant improvements, delivering a chaotic behavior that spans the entire parameter space while maintaining high entropy and a consistently positive Lyapunov exponent.

To assess hardware feasibility, selected proposed maps were also implemented on an FPGA platform. The Lyapunov exponent values obtained from the hardware implementation were closely aligned with those from MATLAB simulations, thereby confirming the practical applicability of the designs. In addition, the correspondence between hardware and software results remained consistent across operating points and parameter sweeps, which strengthens the overall validation. Despite minimal variations attributable to fixed-point and numerical representation formats, the FPGA-based results demonstrated that high-resolution and resource-efficient implementation of the proposed chaotic maps is both feasible and effective for real-world applications, providing a clear path from algorithmic development to deployable hardware.

The expanded chaotic window, enhanced entropy, and hardware realizability of the proposed methods make them highly suitable for practical use in secure communication, cryptographic systems, and other applications where robust and resilient chaotic behavior is essential.

Author Contributions: Conceptualization, M.C. and M.S.H.; methodology, M.C., Z.N., and M.S.H.; software, M.C., Z.N., S.S., A.D., and M.S.H.; validation, M.C., Z.N., S.S., A.D., and M.S.H.; formal analysis, M.C., Z.N., and M.S.H.; investigation, M.C.; resources, M.C., Z.N., S.S., and A.D.; data curation, M.C.; writing—original draft preparation, M.C.; writing—review and editing, M.C., and M.S.H.; visualization, M.C., Z.N., S.S., and M.S.H.; supervision, M.S.H.; project administration, M.C. and M.S.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable

Data Availability Statement: The data supporting the findings of this study are contained within the article. Additional data and simulation results are available from the corresponding author upon reasonable request.

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

References

1. Ding, D.; Wang, W.; Yang, Z.; Hu, Y.; Wang, J.; Wang, M.; Niu, Y.; Zhu, H. An n-dimensional modulo chaotic system with expected Lyapunov exponents and its application in image encryption. *Chaos Solitons Fractals* **2023**, *174*, 113841. [\[CrossRef\]](#)
2. Strogatz, S.H. *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*; CRC Press: Boca Raton, FL, USA, 2018.
3. Hua, Z.; Zhou, Y. Dynamic parameter-control chaotic system. *IEEE Trans. Cybern.* **2015**, *46*, 3330–3341. [\[CrossRef\]](#) [\[PubMed\]](#)
4. Da Silva, S.S.; Cardoso, M.; Nardo, L.; Nepomuceno, E.; Hübner, M.; Arias-Garcia, J. A new chaos-based PRNG hardware architecture using the HUB fixed-point format. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 1–8. [\[CrossRef\]](#)
5. Pak, C.; Huang, L. A new color image encryption using combination of the 1D chaotic map. *Signal Process.* **2017**, *138*, 129–137. [\[CrossRef\]](#)
6. Zhou, Y.; Bao, L.; Chen, C.P. A new 1D chaotic system for image encryption. *Signal Process.* **2014**, *97*, 172–182. [\[CrossRef\]](#)
7. Xiao, D.; Li, T.; Li, Y.; Lu, H.; Liu, J. A countermeasure for advanced encryption standard against power analysis attack using chaotic system. In Proceedings of the 2021 10th International Conference on Internet Computing for Science and Engineering, Guilin, China, 9–11 July 2021; pp. 60–66.
8. Pareschi, F.; Scotti, G.; Giancane, L.; Rovatti, R.; Setti, G.; Trifiletti, A. Power analysis of a chaos-based random number generator for cryptographic security. In Proceedings of the 2009 IEEE International Symposium on Circuits and Systems (ISCAS), Taipei, Taiwan, 24–27 May 2009; pp. 2858–2861.
9. Chowdhury, M.; Hasan, M.; Hoque, T.; Hasan, M.S. Physically Unclonable and Reconfigurable Circuits for IP Protection: Opportunities and Challenges. In Proceedings of the 2024 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), Knoxville, TN, USA, 1–3 July 2024; pp. 817–820.
10. Zhou, Y.; Hua, Z.; Pun, C.M.; Chen, C.P. Cascade chaotic system with applications. *IEEE Trans. Cybern.* **2014**, *45*, 2001–2012. [\[CrossRef\]](#) [\[PubMed\]](#)
11. Gonzales, O.A.; Han, G.; De Gyvez, J.P.; Sánchez-Sinencio, E. Lorenz-based chaotic cryptosystem: A monolithic implementation. *IEEE Trans. Circuits Syst. I Fundam. Theory Appl.* **2000**, *47*, 1243–1247. [\[CrossRef\]](#)
12. Al-Shameri, W.F.H. Dynamical properties of the Hénon mapping. *Int. J. Math. Anal.* **2012**, *6*, 2419–2430.
13. Zhang, Y.; Liu, Y.; Wang, T.; Song, J.; Shen, T. Complex-valued chaotic model with high chaos complexity and provable Lyapunov exponent. *Chaos Solitons Fractals* **2025**, *201*, 117282. [\[CrossRef\]](#)
14. Ors Yalcin, B.; Hematpour, N.; Faruk Birgul, O. Design of the new classic and quantum chaotic maps and implementation on FPGA. *Arab. J. Sci. Eng.* **2025**, *50*, 17441–17487. [\[CrossRef\]](#)
15. Damaj, I.; Zaher, A.; Lawand, W. Optimizing FPGA implementation of high-precision chaotic systems for improved performance. *PLoS ONE* **2024**, *19*, e0299021. [\[CrossRef\]](#) [\[PubMed\]](#)
16. Hasan, M.S.; Paul, P.S.; Dhungel, A.; Sadia, M.; Hossain, M.R. Design, Analysis, and Application of Flipped Product Chaotic System. *IEEE Access* **2022**, *10*, 125181–125193. [\[CrossRef\]](#)
17. Nazish, M.; Javid, M.; Banday, M.T. Enhanced logistic map with infinite chaos and its applicability in lightweight and high-speed pseudo-random bit generation. *Cybersecurity* **2025**, *8*, 24. [\[CrossRef\]](#)
18. Alawida, M. Enhancing logistic chaotic map for improved cryptographic security in random number generation. *J. Inf. Secur. Appl.* **2024**, *80*, 103685. [\[CrossRef\]](#)
19. Paul, P.S.; Dhungel, A.; Sadia, M.; Hossain, M.R.; Muldrey, B.; Hasan, M.S. Self-parameterized chaotic map: A hardware-efficient scheme providing wide chaotic range. In Proceedings of the 2021 28th IEEE International Conference on Electronics, Circuits, and Systems (ICECS), Dubai, United Arab Emirates, 21 November–1 December 2021; pp. 1–5.
20. Paul, P.S.; Sadia, M.; Hossain, M.R.; Muldrey, B.; Hasan, M.S. Design of a low-overhead random number generator using cmos-based cascaded chaotic maps. In Proceedings of the 2021 on Great Lakes Symposium on VLSI, Virtually, 22–25 July 2021; pp. 109–114.
21. Wu, Y.; Zhou, Y.; Bao, L. Discrete wheel-switching chaotic system and applications. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2014**, *61*, 3469–3477. [\[CrossRef\]](#)
22. Hua, Z.; Zhou, B.; Zhou, Y. Sine-transform-based chaotic system with FPGA implementation. *IEEE Trans. Ind. Electron.* **2017**, *65*, 2557–2566.
23. Zhu, H.; Zhao, Y.; Song, Y. 2D logistic-modulated-sine-coupling-logistic chaotic map for image encryption. *IEEE Access* **2019**, *7*, 14081–14098. [\[CrossRef\]](#)
24. Hua, Z.; Zhou, Y. Exponential chaotic model for generating robust chaos. *IEEE Trans. Syst. Man Cybern. Syst.* **2019**, *51*, 3713–3724. [\[CrossRef\]](#)
25. Wu, Q.; Zhang, F.; Hong, Q.; Wang, X.; Zeng, Z. Research on cascading high-dimensional isomorphic chaotic maps. *Cogn. Neurodynamics* **2021**, *15*, 157–167.

26. Islam, M.O.U.; Malik, S.A.; Partila, P.; Frnda, J. A novel exposine chaotic map for secure image encryption with applications in adaptive cruise control. *Array* **2026**, *29*, 100710. [[CrossRef](#)]
27. Li, X.; Zhao, J.; Zhang, Y. Generation of one-dimensional complex discrete hyperchaotic maps with hardware implementation. *Chaos Solitons Fractals* **2025**, *200*, 116976. [[CrossRef](#)]
28. Da Silva, S.S.; Nardo, L.G.; Nepomuceno, E.; Yudi, J.; Rego, B.S.; Arias-Garcia, J. A Hardware-Efficient Chaotic PRNG Exploring Posit Arithmetic for Secure Image Encryption. *IEEE Access* **2025**, *13*, 209813–209828. [[CrossRef](#)]
29. Sarra, B.; Sun, H.; Dua, M.; Dua, S.; Dhingra, D. A novel 1D powered Chebyshev quadratic map-based image encryption using dynamic permutation-diffusion. *Sci. Rep.* **2026**, *16*, 9469. [[CrossRef](#)] [[PubMed](#)]
30. Alexan, W.; Shabasy, N.H.E.; Ehab, N.; Maher, E.A. A secure and efficient image encryption scheme based on chaotic systems and nonlinear transformations. *Sci. Rep.* **2025**, *15*, 31246. [[CrossRef](#)] [[PubMed](#)]
31. Garcia-Bosque, M.; Pérez-Resca, A.; Sánchez-Azqueta, C.; Aldea, C.; Celma, S. Chaos-based bitwise dynamical pseudorandom number generator on FPGA. *IEEE Trans. Instrum. Meas.* **2018**, *68*, 291–293. [[CrossRef](#)]
32. 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](#)]
33. Luo, Y.; Fan, C.; Xu, C.; Li, X. Design and FPGA implementation of a high-speed PRNG based on an nD non-degenerate chaotic system. *Chaos Solitons Fractals* **2024**, *183*, 114951. [[CrossRef](#)]
34. Hasan, M.S.; Chowdhury, M.; Niu, Z.; Song, S.; Dhungel, A. 1-D Robust Chaotic Maps Through Systematic Shifting and Halfway Shifted Product. In Proceedings of the 2024 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), Knoxville, TN, USA, 1–3 July 2024; pp. 319–324.
35. Hasan, M.S.; Dhungel, A.; Paul, P.S.; Sadia, M.; Hossain, M.R. Normalized Linearly-Combined Chaotic System: Design, Analysis, Implementation and Application. *IEEE Open J. Ind. Electron. Soc.* **2023**, *4*, 486–505. [[CrossRef](#)]
36. Agrawal, A.; Mueller, S.M.; Fleischer, B.M.; Sun, X.; Wang, N.; Choi, J.; Gopalakrishnan, K. DFloat: A 16-b floating point format designed for deep learning training and inference. In Proceedings of the 2019 IEEE 26th Symposium on Computer Arithmetic (ARITH), Kyoto, Japan, 10–12 June 2019; pp. 92–95.
37. Chugunkov, I.V.; Ivanov, M.A.; Gridneva, E.A.; Shestakova, N.Y. Classification of pseudo-random number generators applied to information security. In Proceedings of the 2017 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), St. Petersburg and Moscow, Russia, 1–3 February 2017; pp. 370–373.
38. Dodis, Y.; Pointcheval, D.; Ruhault, S.; Vergniaud, D.; Wichs, D. Security analysis of pseudo-random number generators with input: /dev/random is not robust. In Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security, Berlin, Germany, 4–8 November 2013; pp. 647–658.
39. Li, C.Y.; Chen, Y.H.; Chang, T.Y.; Deng, L.Y.; To, K. Period extension and randomness enhancement using high-throughput reseeding-mixing PRNG. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2011**, *20*, 385–389.
40. Naik, R.B.; Singh, U. A review on applications of chaotic maps in pseudo-random number generators and encryption. *Ann. Data Sci.* **2024**, *11*, 25–50.
41. Rukhin, A.; Soto, J.; Nechvatal, J.; Smid, M.; Barker, E.; Leigh, S.; Levenson, M.; Vangel, M.; Banks, D.; Heckert, A.; et al. *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*; US Department of Commerce, Technology Administration, National Institute of Standards and Technology: Gaithersburg, MD, USA, 2001; Volume 22.
42. Lécuyer, P.; Simard, R. TestU01: AC library for empirical testing of random number generators. *ACM Trans. Math. Softw. (TOMS)* **2007**, *33*, 1–40.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.