






# Normalized Linearly-Combined Chaotic System: Design, Analysis, Implementation, and Application

MD SAKIB HASAN  (Member, IEEE), ANURAG DHUNGEL , PARTHA SARATHI PAUL ,  
MAISHA SADIA  (Graduate Student Member, IEEE),  
AND MD RAZUAN HOSSAIN  (Graduate Student Member, IEEE)

The University of Mississippi, University, MS 38677 USA

CORRESPONDING AUTHOR: MD SAKIB HASAN (e-mail: mhasan5@olemiss.edu).

**ABSTRACT** This work presents a general framework for developing a multiparameter 1-D chaotic system for uniform and robust chaotic operation across the parameter space. This is important for diverse practical applications where parameter disturbance may cause degradation or even complete disappearance of chaotic properties. The wide uninterrupted chaotic range and improved chaotic properties are demonstrated with the aid of stability analysis, bifurcation diagram, Lyapunov exponent (LE), Kolmogorov entropy, Shannon entropy, and correlation coefficient. We also demonstrate the proposed system's amenability to cascading for further performance improvement. We introduce an efficient field-programmable gate array-based implementation and validate its chaotic properties using comparison between simulation and experimental results. Cascaded normalized linearly-combined chaotic system (NLCS) exhibits average LE, chaotic ratio, and chaotic parameter space of 1.364, 100%, and  $1.1 \times 10^{12}$ , respectively, for 10-bit parameter values. We provide a thorough comparison of our system with prior works both in terms of performance and hardware cost. We also introduce a simple extension scheme to build 2-D robust, hyperchaotic NLCS maps. We present a novel reconfigurable multiparameter pseudorandom number generator and validate its randomness using two standard statistical tests, namely, National Institute of Standards and Technology SP 800-22 and FIPS PUB 140-2. Finally, we outline six potential applications where NLCS will be useful.

**INDEX TERMS** Chaos, chaotic map, encryption, field-programmable gate array (FPGA), Lyapunov exponent (LE), reconfigurable random number generator, robust chaos, security.

## I. INTRODUCTION

Chaos can be defined as a phenomenon that occurs when the temporal evolution of a deterministic nonlinear dynamic system becomes aperiodic and highly sensitive to its initial state. In the chaotic region, two initial states, starting infinitesimally close to each other, will eventually follow two drastically different time trajectories, which will never repeat themselves. Starting with Lorenz's seminal work in 1963 [1], chaos has attracted a lot of attention in the last 60 years in different disciplines, such as physics, biology, chemistry, and engineering [2]. In recent years, researchers have leveraged the dual properties of chaotic systems, namely, "deterministic aperiodicity" and "acute susceptibility to initial state

perturbation" for diverse applications, such as random number generation [3], [4], [5], [6], data encryption [7], [8], [9], reconfigurable logic [10], [11], physical unclonable function (PUF) [12], side-channel attack mitigation [13], secure communication [14], [15], [16], modeling of astronomical phenomenon [17], logic obfuscation [18], and so on.

Based on the number of state variables or dimension, chaotic systems can be broadly divided into the following two groups: 1) one-dimensional (1-D); and 2) multidimensional (multi-D) systems. Based on the nature of time evolution, chaotic systems can be classified into the following two groups: 1) continuous-time; and 2) discrete time. It has been shown that a continuous-time nonlinear dynamic system has

to have at least three state variables to show chaotic behavior whereas for discrete-time system there is no such restriction [2]. Familiar examples of 1-D discrete-time maps are sine map, tent map, logistic map, and so on. On the other hand, Henon map (discrete-time) and Lorenz system (continuous time) are examples of multi-D chaotic systems.

A 1-D discrete-time chaotic system consists of a nonlinear block, called a chaotic map, which defines one or multiple control parameter-dependent evolution of a single state variable in discrete-time steps. Conventionally, these 1-D systems have been studied using classic mathematical functions, such as logistic map, tent map, sine map, etc. These traditional 1-D maps are useful as they offer simplicity in implementations. However, the chaotic region of these 1-D maps is limited. Moreover, a good chaotic entropy is not promised over the whole range of that limited chaotic window. As a result, any change in the operating condition or parameter value may degrade the chaotic properties or even deflect the system from the desired chaotic region to an undesired nonchaotic (fixed point or periodic orbit) region. Researchers have been exploring various schemes for an improved chaotic map by manipulating multiple existing 1-D maps (henceforth referred to as seed maps). The schemes include dynamic reconfiguration of control parameter [19], [20], cascading of multiple seed maps [6], [21], use of discrete wheel-switching technique [22], averaging of multiple seed maps [23], sine transformation of a combination of multiple maps [24], modulation and coupling [25], exponential chaotic model [15], and so on. Recently, a new paradigm of designing hyperchaotic maps based on discrete memristor model has attracted the interest of the research community. Memristor was postulated as the fourth fundamental circuit element by Chua in 1971 [26] and experimentally demonstrated in 2008 by HP Labs [27]. By coupling existing 1-D maps with discrete memristor model, researchers have reported 2-D [28], [29] and 3-D [30] hyperchaotic maps with complex dynamics along with their potential usage in secure communication [29] and image encryption [30]. All of the aforementioned techniques result in an improved chaotic performance by widening the chaotic window and/or increasing the chaotic entropy at the cost of increased overhead.

In this work, we propose a general framework of a multiparameter 1-D robust chaotic system called the normalized linearly-combined chaotic system (NLCS) where the output of  $n$  number of 1-D seed maps are linearly combined with arbitrary coefficients and then normalized using a simple algorithm to produce the final output. We use stability analysis using Jacobian at equilibrium points along with the bifurcation plot to demonstrate the wide chaotic region across the entire parameter space (EPS). Then, the excellent chaotic properties are illustrated with the aid of established entropy metrics. The performance analysis shows that NLCS provides an uninterrupted chaotic window, along with uniformly high entropy, over the EPS. We also show an efficient hardware implementation in field-programmable gate array (FPGA) and validate the experimental results against the simulation results

from MATLAB. We introduce a simple extension scheme to build 2-D maps with robust, hyperchaotic, and uniformly excellent properties across the parameter space. Finally, we propose a new reconfigurable multiparameter pseudorandom number generator (PRNG) and outline six potential applications for the proposed system.

In summary, our main contributions in this work are as follows.

- 1) We present a general framework named NLCS for developing arbitrary number of new multiparameter 1-D chaotic system.
- 2) We demonstrate the uniformly excellent chaotic operation of four new NLCS maps across the parameter space using stability analysis, bifurcation diagram, Lyapunov exponent (LE), KE, Shannon entropy (SE), and correlation coefficient (CC). We also show the proposed system's amenability to further improvement in chaotic performance and parameter space using cascading.
- 3) We develop an efficient design for FPGA-based hardware implementation and present a thorough comparison against prior works in terms of chaotic performance and implementation metrics.
- 4) We introduce a simple extension scheme to build 2-D hyperchaotic maps with uniformly excellent properties and demonstrate it using three representative examples.
- 5) We present a new reconfigurable multiparameter PRNG and validate its excellent randomness property using two standard statistical test suites. We also outline six application scenarios where the particular attributes of the proposed system will be useful.

The rest of this article is organized as follows. Three seed maps used in this work are introduced in Section II. The proposed scheme, NLCS is presented in Section III along with the derivation of five representative NLCS systems. Section IV evaluates the chaotic performance with LE, KE, SE, and CC. Section V presents an extension of the proposed system for further performance enhancement. An efficient hardware implementation in FPGA along with its validation against simulation results is discussed in Section VI. Section VII introduces some global metrics to compare the proposed system with previous works. Section IX presents a new reconfigurable PRNG using NLCS along with performance evaluation using statistical tests. Section X outlines six promising applications. Finally, Section XI concludes this article.

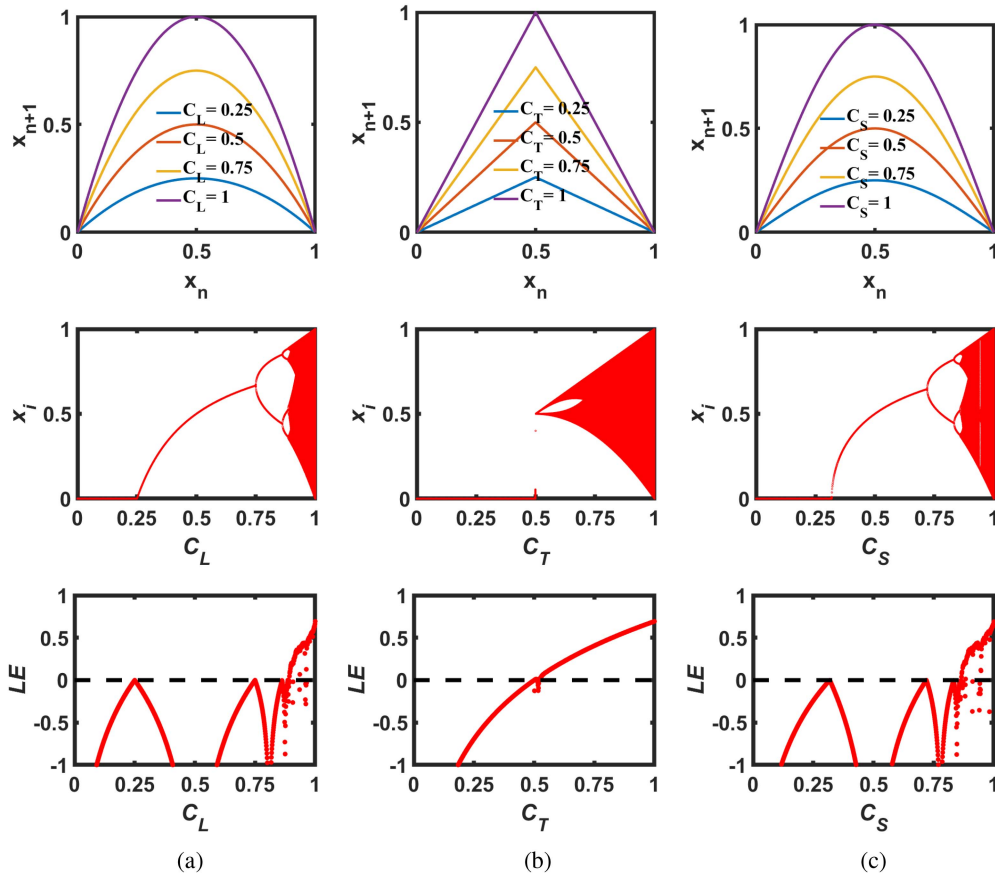
## II. TRADITIONAL SEED MAPS

This section reviews three existing 1-D chaotic maps namely, logistic, tent, and sine maps as background. They will be used as seed maps to generate new chaotic maps in Section IV. For ease of comparison, we are using the normalized versions of these seed maps such that their domain, range, and parameter values are within [0, 1].

Logistic map can be mathematically defined as

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

where  $C_L$  is the control parameter and  $C_L \in [0, 1]$ .



**FIGURE 1.** Transfer curve (first row), Bifurcation diagram (second row), and LE (third row) of three seed maps. (a) Logistic ( $\mathcal{L}$ ). (b) Tent ( $\mathcal{T}$ ). (c) Sine ( $\mathcal{S}$ ).

Tent map can be mathematically defined as

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

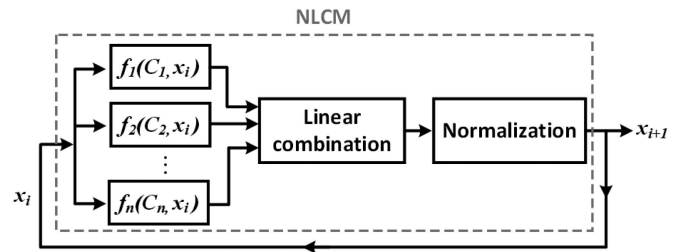
where  $C_T$  is the control parameter and  $C_T \in [0, 1]$ .

Sine map can be mathematically defined as

$$x_{i+1} = \mathcal{S}(x_i) = C_S \sin(\pi x_i) \quad (3)$$

where  $C_S$  is the control parameter and  $C_S \in [0, 1]$ .

The effect of a control parameter on a dynamical system can be visualized with a bifurcation diagram where for each parameter value, a long sequence of steady-state output values is plotted. The chaotic property in the output is evaluated with a widely used metric called LE. A positive LE demonstrates the existence of chaotic behavior [2]. Fig. 1 plots the transfer curves, bifurcation diagrams, and LEs of the logistic, sine, and tent maps with the change of their control parameters. As can be observed, the logistic, sine, and tent maps have chaotic behaviors when  $C_L \in [0.89, 1]$ ,  $C_S \in [0.87, 1]$ , and  $C_T \in (0.5, 1)$ , respectively. It should be noted that the logistic and sine maps do not have robust chaos as periodic windows exist in their chaotic ranges, but the tent map has robust chaos when  $C_T \in (0.5, 1)$ .

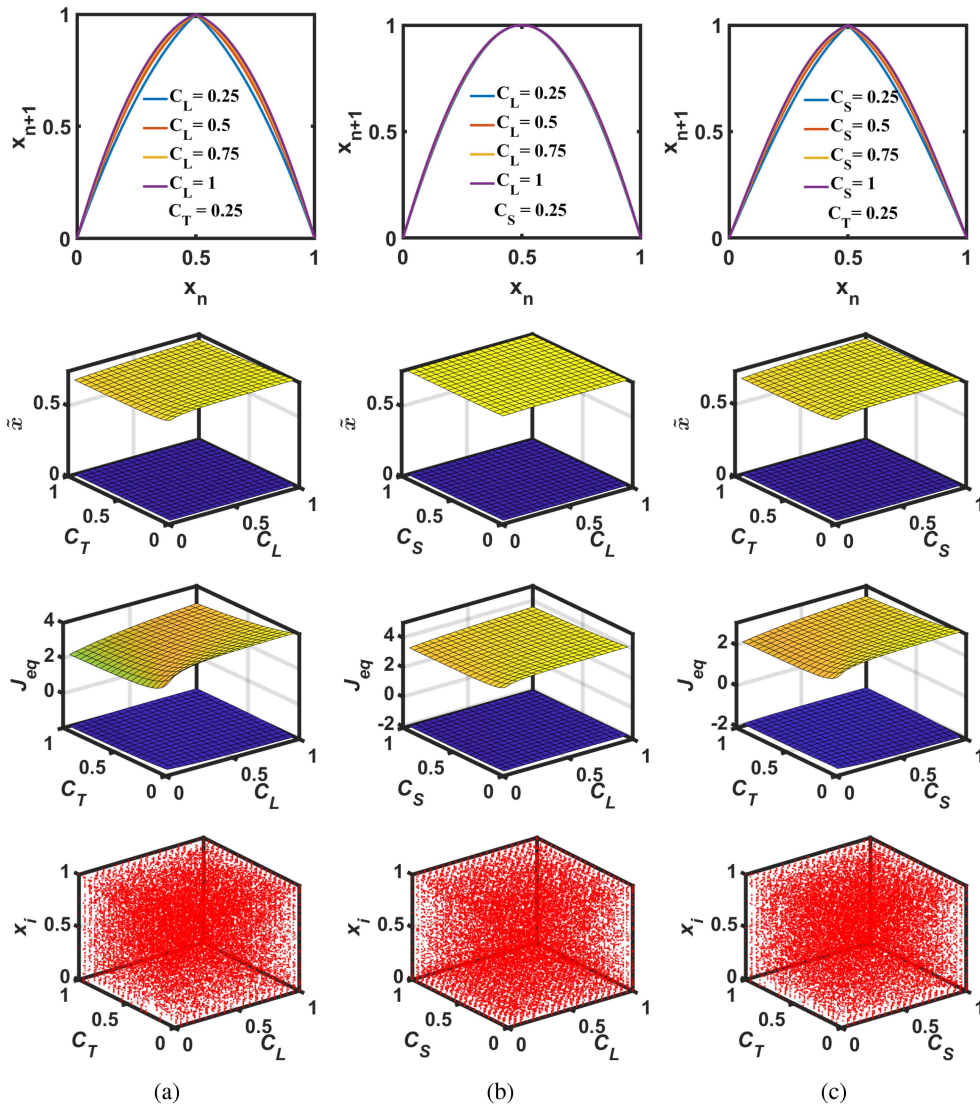


**FIGURE 2.** Schematic of the NLCM scheme. Here,  $f_n$  and  $C_n$  denote the  $n$ th mapping function and the corresponding control parameter, respectively.  $x_i$  is the  $i$ th iteration value of the state variable.

### III. PROPOSED CHAOTIC SYSTEM

Fig. 2 shows the block diagram of the proposed NLCM. The output of the map function, normalized linearly-combined chaotic map (NLCM) is fed back to the input after each iteration. Inside NLCM, the output of the seed maps are linearly combined and then normalized to produce the final output. Given,  $n$  seed maps,  $f_1(C_1, x_i), f_2(C_2, x_i), \dots, f_n(C_n, x_i)$ , the output of the linear combination block,  $LC$  is

$$LC = \sum_{j=1}^n a_j f_j(x_i). \quad (4)$$



**FIGURE 3.** Transfer curves (first row), equilibrium points (second row), corresponding Jacobian (third row), and Bifurcation diagrams (fourth row) of three NLCS maps. (a)  $LT$ . (b)  $LS$ . (c)  $ST$ .

Here,  $a_1, a_2, \dots, a_n$  are the coefficients of the linear combination of seed maps  $f_1, f_2, \dots, f_n$ , respectively. The functionality of the normalization block,  $N(LC)$ , can be expressed as

$$N = \frac{LC - L}{H - L}. \quad (5)$$

Here,  $H = \max(LC)$  and  $L = \min(LC)$  over the range of  $x_i$  from 0 to 1. Both  $L$  and  $H$  are functions of the parameters of the seed maps and coefficients of the linear combination.

The fundamental insight behind this framework can be conveyed using the transfer curves three NLCS systems developed using different combinations of the seed maps. As shown first row of Fig. 3, all the transfer curves cover the entire output range  $[0, 1]$  while retaining high slope across the parameter space for each value of the state variable. This is in stark contrast to the constituent seed maps as shown in the first row of Fig. 1 where the slope and output range vary

significantly with the change in parameter value. Since the chaotic performance has a strong dependence of the average slope of the trajectory, we expect our system to have uniformly excellent entropic properties across the parameter space as will be demonstrated later in Section IV using LE, KE, and SE. Moreover, we expect that the slight change of the transfer curve as a result of any parameter variation is sufficient for generating completely uncorrelated long-term sequence for different parameter values since a chaotic system is extremely susceptible to tiniest perturbation in initial condition or parameter value (popularly known as the “butterfly effect”). This hypothesis will be proved with the help of CC in Section IV and we will leverage this to build a novel reconfigurable PRNG in Section IX.

The proposed NLCS can be formed with any number of seed maps with different values of coefficients for the linear combination. In Section III-A, we will explore three such



maps with two constituent seed maps and unity coefficients. Then, in Section III-B, we will consider the case of three seed maps with unity coefficients. In the final Section III-C, we will consider the case of two seed maps with coefficients other than 1.

**A. TWO SEED MAPS WITH UNITY COEFFICIENTS**

First, we will consider three combinations of two seed maps while keeping the coefficients  $a_1 = a_2 = 1$ .

**1) LOGISTIC-TENT**

If the two constituent seed maps are logistic and tent maps, then for  $a_1 = a_2 = 1$ ,  $LC = \mathcal{L}(x_i) + \mathcal{T}(x_i)$ ,  $H = C_L + C_T$ , and  $L = 0$ . The final expression for  $\mathcal{LT}$  map can be written as

$$x_{i+1} = \begin{cases} x_i(4C_L(1-x_i)+2C_T)/(C_L+C_T); & x_i < 0.5 \\ (1-x_i)(4C_Lx_i+2C_T)/(C_L+C_T); & x_i \geq 0.5. \end{cases} \quad (6)$$

The equilibrium points of the  $\mathcal{LT}$  map are the roots of the following equation:

$$\tilde{x} = \begin{cases} \tilde{x}(4C_L(1-\tilde{x})+2C_T)/(C_L+C_T); & \tilde{x} < 0.5 \\ (1-\tilde{x})(4C_L\tilde{x}+2C_T)/(C_L+C_T); & \tilde{x} \geq 0.5. \end{cases} \quad (7)$$

Solving (7), we can find that there are two equilibrium points over the range [0, 1]. The equilibrium point of a dynamic system can be either stable or unstable. A stable point implies a fixed point whereas an unstable point implies a periodic or chaotic oscillation. The stability of a fixed point can be determined by the magnitude of the eigenvalues of the Jacobian matrix (a derivative of the map function with respect to the state variable) at that equilibrium point. If at least one eigenvalue has a magnitude greater than 1 then the system is unstable. For a 1-D system, the eigenvalue can be simply determined by the value of the Jacobian at the equilibrium point. The Jacobian for  $\mathcal{LT}$  map can be expressed as

$$J(x) = \begin{cases} (4C_L(1-2x)+2C_T)/(C_L+C_T); & x < 0.5 \\ (4C_L(1-2x)-2C_T)/(C_L+C_T); & x \geq 0.5. \end{cases} \quad (8)$$

The second and third subplots of Fig. 3(a) show the two equilibrium points and their corresponding Jacobian values, respectively. The magnitudes of the Jacobian at both equilibrium points are greater than 1 clearly indicating an unstable state. The fourth subplot shows the corresponding bifurcation diagram, which illustrates chaotic operation across the entire 2-D parameter space. This is consistent with the instability of equilibrium points indicating robust chaos for all possible combinations of parameter values.

**2) LOGISTIC-SINE**

If the two constituent seed maps are logistic and sine maps, then for  $a_1 = a_2 = 1$ ,  $LC = \mathcal{L}(x_i) + \mathcal{S}(x_i)$ ,  $H = C_L + C_S$ , and

$L = 0$ . The final expression for  $\mathcal{LS}$  map can be written as

$$x_{i+1} = (4C_Lx_i(1-x_i) + C_S\sin(\pi x_i))/(C_L + C_S). \quad (9)$$

Here,  $C_L$  and  $C_S$  are two parameters of the system and  $C_L, C_S \in [0, 1]$ . For a particular combination of parameters, there are two equilibrium points, which can be determined by solving for the roots of the following equation:

$$\tilde{x} = (4C_L\tilde{x}(1-\tilde{x}) + C_S\sin(\pi\tilde{x}))/C_L + C_S. \quad (10)$$

The Jacobian for  $\mathcal{LS}$  map can be expressed as

$$J(x) = (4C_L(1-2x) + \pi C_S\cos(\pi x_i))/C_L + C_S. \quad (11)$$

The second and third subplots of Fig. 3(b) show the two equilibrium points and their corresponding Jacobian values. The magnitudes of the Jacobian at both equilibrium points are greater than 1 clearly indicating an unstable state. The fourth subplot shows the corresponding bifurcation diagram, which shows that the outputs are chaotic across the entire 2-D parameter space. This is consistent with the instability of equilibrium points and demonstrates wide robust chaos for all possible combinations of parameter values.

**3) SINE-TENT (ST)**

If the two constituent seed maps are sine and tent maps, then for  $a_1 = a_2 = 1$ ,  $LC = \mathcal{S}(x_i) + \mathcal{T}(x_i)$ ,  $H = C_S + C_T$ , and  $L = 0$ . The final expression for  $\mathcal{ST}$  map can be written as

$$x_{i+1} = \begin{cases} (C_S\sin(\pi x_i)+2C_Tx_i)/(C_S+C_T); & x_i < 0.5 \\ (C_S\sin(\pi x_i)+2C_T(1-x_i))/(C_S+C_T); & x_i \geq 0.5. \end{cases} \quad (12)$$

Here,  $C_L$  and  $C_S$  are two parameters of the system and  $C_L, C_S \in [0, 1]$ . For a particular combination of parameters, there are two equilibrium points that can be determined by solving for the roots of the following equation:

$$\tilde{x} = \begin{cases} (C_S\sin(\pi\tilde{x})+2C_T\tilde{x})/(C_S+C_T); & \tilde{x} < 0.5 \\ (C_S\sin(\pi\tilde{x})+2C_T(1-\tilde{x}))/C_S+C_T); & \tilde{x} \geq 0.5. \end{cases} \quad (13)$$

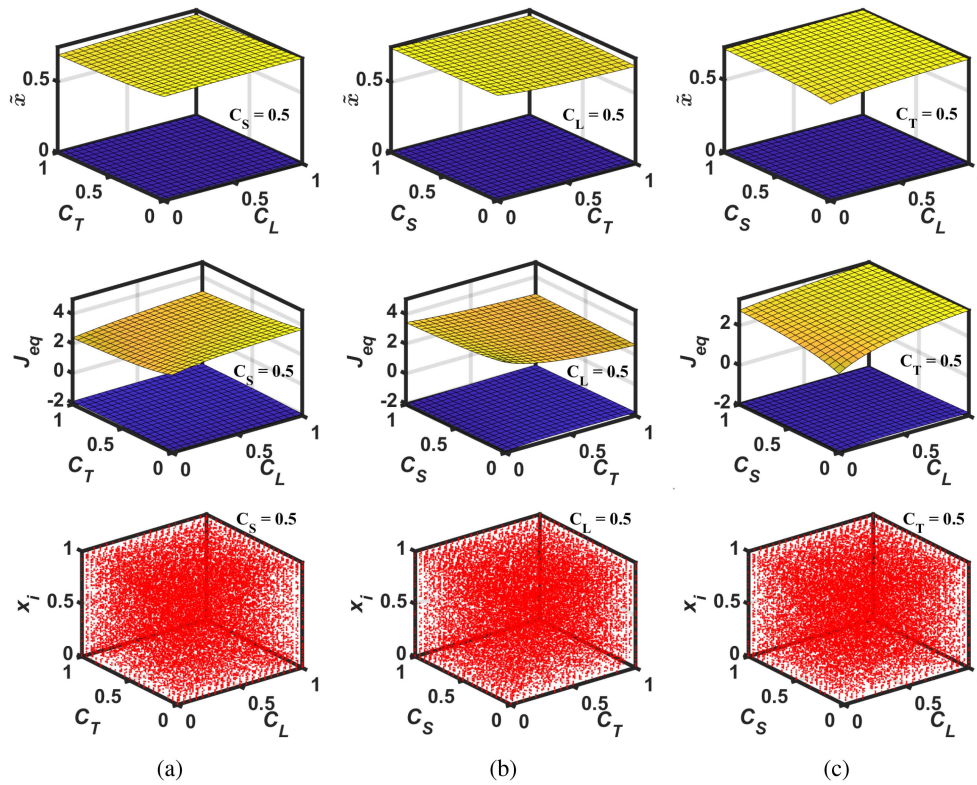
The Jacobian for  $\mathcal{ST}$  map can be expressed as

$$J(x) = \begin{cases} (\pi C_S\cos(\pi x)+2C_T)/(C_S+C_T); & x < 0.5 \\ (\pi C_S\cos(\pi x)-2C_T)/(C_S+C_T); & x \geq 0.5. \end{cases} \quad (14)$$

The second and third subplots of Fig. 3(c) show the two equilibrium points and their corresponding Jacobian values, respectively. The magnitudes of the Jacobian at both equilibrium points are greater than 1 clearly indicating an unstable state. The fourth subplot shows the corresponding bifurcation diagram demonstrating chaos across the entire 2-D parameter space. This is consistent with the instability of equilibrium points indicating wide robust chaos across the EPS.

**B. THREE SEED MAPS WITH UNITY COEFFICIENTS**

Here, we consider an NLCS system consisting of three seed maps with unity coefficients, i.e.,  $a_1 = a_2 = a_3 = 1$ , and



**FIGURE 4.** Equilibrium points (first row), corresponding Jacobian (second row), and bifurcation diagrams (third row) of  $\mathcal{LTS}$  maps while keeping one parameter fixed at 0.5 and varying the other two parameters. (a)  $C_L - C_T$ . (b)  $C_T - C_S$ . (c)  $C_L - C_S$ .

name it logistic–tent–sine ( $\mathcal{LTS}$ ) map. Here,  $LC = \mathcal{L}(x_i) + \mathcal{T}(x_i) + \mathcal{S}(x_i)$ ,  $H = C_L + C_T + C_S$ , and  $L = 0$ . The final expression for  $\mathcal{LTS}$  map can be written as

$$x_{i+1} = \begin{cases} \frac{4C_L x_i(1-x_i) + 2C_T x_i + C_S \sin(\pi x_i)}{C_L + C_T + C_S}; & x_i < 0.5 \\ \frac{(4C_L x_i(1-x_i) + 2C_T(1-x_i) + C_S \sin(\pi x_i))}{C_L + C_T + C_S}; & x_i \geq 0.5. \end{cases} \quad (15)$$

The two equilibrium points can be determined by solving for the roots of the following equation:

$$\tilde{x} = \begin{cases} \frac{4C_L \tilde{x}(1-\tilde{x}) + 2C_T \tilde{x} + C_S \sin(\pi \tilde{x})}{C_L + C_T + C_S}; & \tilde{x} < 0.5 \\ \frac{(4C_L \tilde{x}(1-\tilde{x}) + 2C_T(1-\tilde{x}) + C_S \sin(\pi \tilde{x}))}{C_L + C_T + C_S}; & \tilde{x} \geq 0.5. \end{cases} \quad (16)$$

The Jacobian for  $\mathcal{LTS}$  map can be expressed as

$$J(x) = \begin{cases} \frac{4C_L(1-2x) + 2C_T + \pi C_S \cos(\pi x)}{C_L + C_T + C_S}; & x < 0.5 \\ \frac{4C_L(1-2x) - 2C_T + \pi C_S \cos(\pi x)}{C_L + C_T + C_S}; & x \geq 0.5. \end{cases} \quad (17)$$

The first two rows of Fig. 4 show the two equilibrium points and their corresponding Jacobian values while varying two parameters and keeping the third one fixed. In all cases, the magnitudes of the Jacobian at both equilibrium points are greater than 1 clearly indicating an unstable state. The third row shows the corresponding bifurcation diagrams, which show that the generated sequences are chaotic across the entire 3-D parameter space. This is consistent with the instability of equilibrium points indicating wide robust chaos for all possible combinations of parameter values.

### C. TWO SEED MAPS WITH NONUNITY COEFFICIENTS

Previous sections explored the linear combination of seed maps with unity coefficients. Here, for brevity, we consider one example with nonunity coefficients, but we have verified that similar results can be obtained for other combinations as well. For nonunity coefficients, we use superscript to indicate the coefficients in an ordered pair, e.g.,  $NLCS^{(a_1, a_2)}$ . Let us consider the  $\mathcal{LT}$  map with  $a_1 = 2$  and  $a_2 = 3$ . Then,  $LC = 2\mathcal{L}(x_i) + 3\mathcal{T}(x_i)$ ,  $H = 2C_L + 3C_T$ , and  $L = 0$ . The final expression for  $LT^{(2,3)}$  map can be written as

$$x_{i+1} = \begin{cases} x_i(8C_L(1-x_i) + 6C_T)/(2C_L + 3C_T); & x_i < 0.5 \\ (1-x_i)(8C_L x_i + 6C_T)/(2C_L + 3C_T); & x_i \geq 0.5. \end{cases} \quad (18)$$

The equilibrium points of this system are the roots of the following equation:

$$\tilde{x} = \begin{cases} \tilde{x}(8C_L(1-\tilde{x}) + 6C_T)/(2C_L + 3C_T); & \tilde{x} < 0.5 \\ (1-\tilde{x})(8C_L \tilde{x} + 6C_T)/(2C_L + 3C_T); & \tilde{x} \geq 0.5. \end{cases} \quad (19)$$

The Jacobian for this map can be expressed as

$$J(x) = \begin{cases} (8C_L(1-2x) + 6C_T)/(2C_L + 3C_T); & x < 0.5 \\ (8C_L(1-2x) - 6C_T)/(2C_L + 3C_T); & x \geq 0.5. \end{cases} \quad (20)$$

Fig. 5(a) and (b) shows the two equilibrium points and their corresponding Jacobian values. The magnitudes of the Jacobian at both equilibrium points are greater than 1 clearly

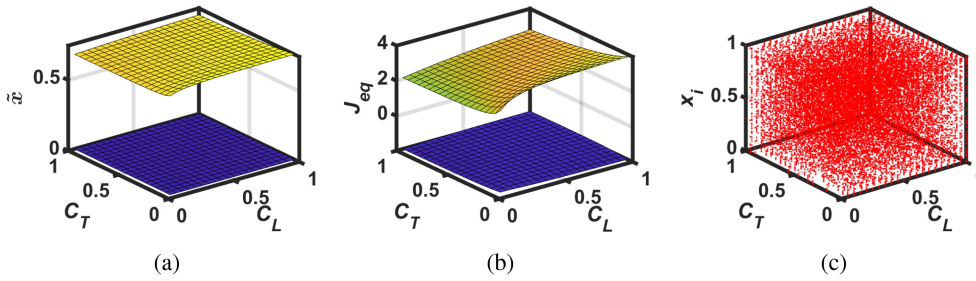


FIGURE 5.  $\mathcal{LT}$  map with  $\alpha_1 = 2$  and  $\alpha_2 = 3$ . (a) Equilibrium points. (b) Jacobian. (c) Bifurcation diagram.

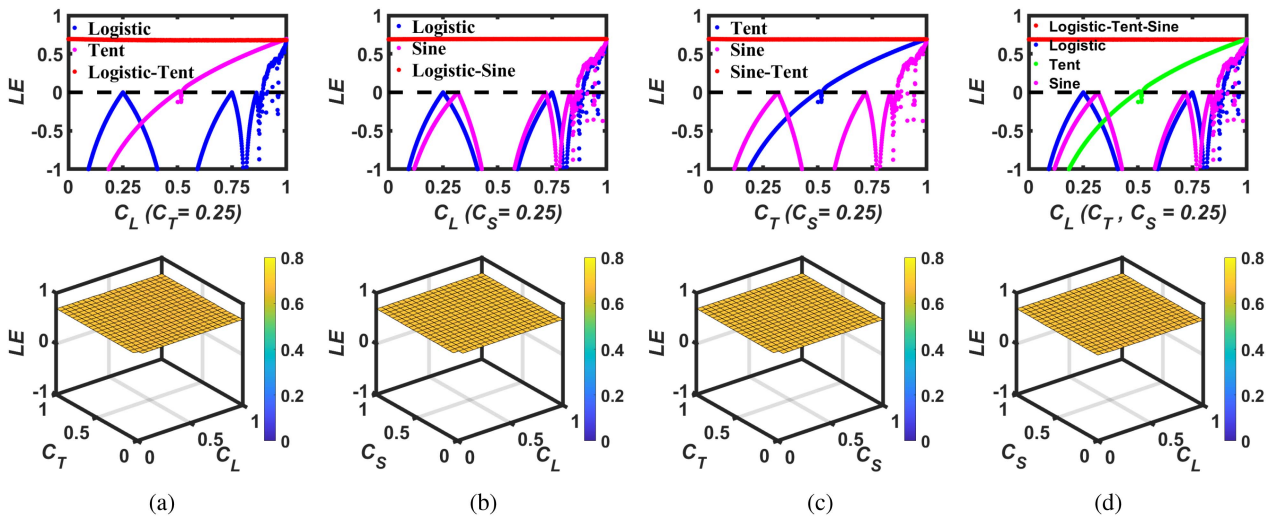


FIGURE 6. LE plot of four NLCS maps. (a)  $\mathcal{LT}$ . (b)  $\mathcal{LS}$ . (c)  $\mathcal{ST}$ . (d)  $\mathcal{LTS}$ .

indicating an unstable state. Fig. 3(c) shows the corresponding bifurcation diagram, which demonstrates chaos across the entire 2-D parameter space. This is consistent with the instability of equilibrium points and clearly illustrates robust chaos across the entire 2-D parameter space for nonunity coefficients.

#### IV. PERFORMANCE ANALYSIS

##### A. LYAPUNOV EXPONENT

The sensitive dependence on the initial condition is a defining characteristic of a chaotic system. Two neighboring trajectories of a chaotic sequence, starting from slightly different initial conditions, diverge exponentially fast, on average. The most widely-used metric to quantify that sensitive dependence on initial conditions is LE. For a discrete-time chaotic system, LE is defined as

$$LE = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=0}^{n-1} \ln|f'(x_i)|. \quad (21)$$

The value of LE is negative for fixed points and periodic orbits whereas for chaotic attractors, its value is positive [2]. Fig. 6

shows the results for four NLCS systems. The LE of each map is calculated with 14 000 steady-state iterations (after discarding first 1000 points) for each control parameter value. The first row presents a comparison of LE values between NLCS and its constituent seed maps with one or more control parameters fixed to a constant value while the other one is varied along the  $x$ -axis. It is clear from these 2-D plots that in the NLCS systems, LE value remains almost steadily close to the maximum LE achievable by the seed maps over the whole operational range. The second row in Fig. 6 shows 3-D LE plots for four NLCS systems where we can observe a uniformly high LE across the EPS.

##### B. KOLMOGOROV ENTROPY

KE captures the generation rate of new information. In this work, we follow the estimation method by Grassberger et al. in [31], which partitions the phase space of an  $F$ -dimensional dynamic system into  $\epsilon^F$ -sized boxes. We are measuring the state of a trajectory,  $\vec{X}(t)$ , at intervals of time,  $\tau$ . There is a probability measure,  $p(i_1, i_2, \dots, i_d)$  that defines the joint probability of  $\vec{X}(t)$  being in the box  $i_1$  at  $t = \tau$ , in  $i_2$  at  $t = 2\tau$ ,



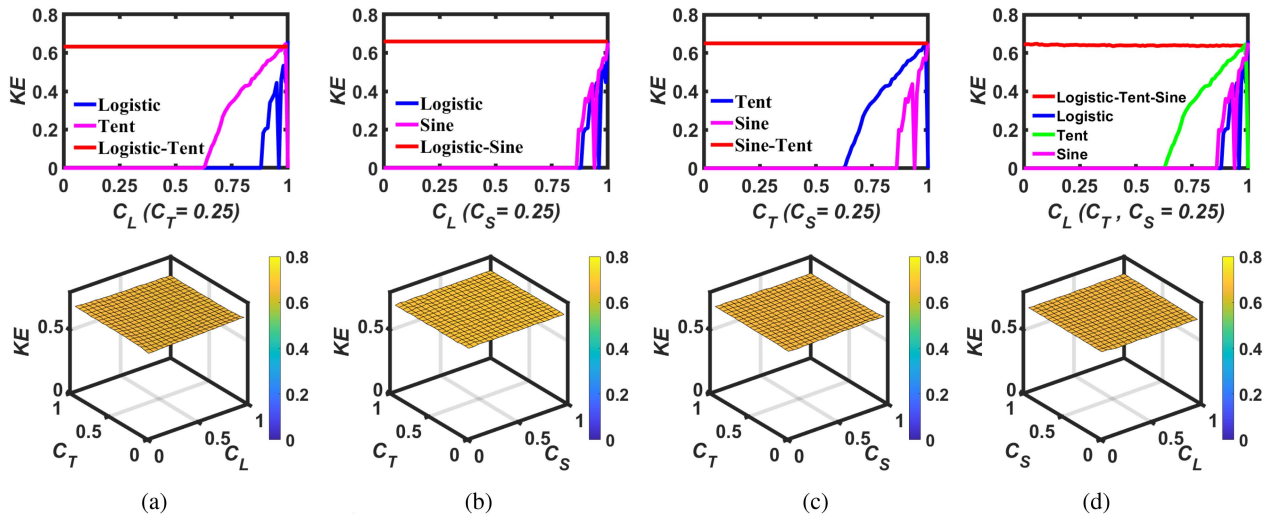


FIGURE 7. KE plot of four NLCS maps. (a)  $LT$ . (b)  $LS$ . (c)  $ST$ . (d)  $LTS$ .

and so on. Then, the KE is defined as

$$KE = - \lim_{\tau \rightarrow \infty} \lim_{\epsilon \rightarrow \infty} \lim_{d \rightarrow \infty} \frac{1}{n} \sum_{i_1, \dots, i_d} p(i_1, i_2, \dots, i_d) \times \ln(p(i_1, i_2, \dots, i_d)). \quad (22)$$

KE is 0 for an ordered sequence,  $\infty$  for a random sequence, and a positive nonzero constant for a chaotic sequence where the higher positive value of KE indicates a better chaotic performance [32]. Fig. 7 shows a uniformly high KE across the EPS for all NLCS schemes, whereas the constituent seed maps show a nonuniform distribution of positive nonzero value in a very narrow region. The KE of each map is calculated with 14 000 steady-state iterations for each parameter value.

### C. SHANNON ENTROPY

SE is a widely used metric to measure the amount of uncertainty in a random process. If the range of values of signal  $X$  is divided into  $n$  equally spaced bins, then the SE can be written as

$$SE = - \sum_{i=1}^n \Pr(x_i) \log_2 \Pr(x_i). \quad (23)$$

Here,  $\Pr(x_i)$  is defined as the probability of signal value located in the  $i$ th bin. SE can be used to check the randomness of a discrete-time sequence. We set  $n = 2^{10} = 1024$  and used (23) to calculate the SE for each control parameter with 14 000 steady-state iterations. The theoretical maximum value is  $\log_2 n = \log_2 1024 = 10$ , which occurs when the sequence values are uniformly distributed across the whole range ( $[0, 1]$ ). The value of SE increases with the amount of ergodicity involved in the sequence. Fig. 8 shows the SE values of NLCS and corresponding seed maps. It is clear from the SE plots that NLCS offers a very high SE value over the whole operational range.

### D. CORRELATION COEFFICIENT

A defining feature of a chaotic system is its extreme sensitivity to slight perturbation in the initial state, i.e., initial condition or parameter values. This sensitive dependence on the initial state can be measured using a well-known metric called CC. Equation (24) shows the expression of Pearson's CC that can be used to determine the correlation between two sequences,  $X$  and  $Y$

$$C_o = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y}. \quad (24)$$

Here, “ $E[\cdot]$ ” indicates the expectation operator while  $\mu$  and  $\sigma$  represent the mean value and standard deviation, respectively. The value of the CC is close to  $+1/-1$  if  $X$  and  $Y$  are highly correlated whereas, a close to 0 CC indicates an extremely low correlation between the data sequences. To measure the initial state dependence using the CC, two sets of steady-state discrete-time data sequences are generated from the same chaotic oscillator with a particular control parameter but with two slightly different initial states. Then, CC is calculated using (24). If the system is chaotic for that particular control parameter then the tiniest variation in the initial state will result in two very different sequences and as a result, we will get a CC close to 0. On the other hand, if the operating point is nonchaotic then the two steady-state sequences will be very similar and result in a CC close to  $+1/-1$ .

Fig. 9 shows the plots of calculated CC for four NLCS systems demonstrating acute sensitivity to initial value perturbation since the CC value is very close to 0 across the EPS. We did a similar experiment to measure the system's susceptibility to parameter perturbation. In this case, we have generated two long sequences with identical initial conditions while slightly varying the parameter value. Fig. 10 shows the parameter sensitivity results for the four NLCS systems demonstrating high susceptibility to tiniest parameter perturbation across the EPS. Therefore, NLCS can be used as a



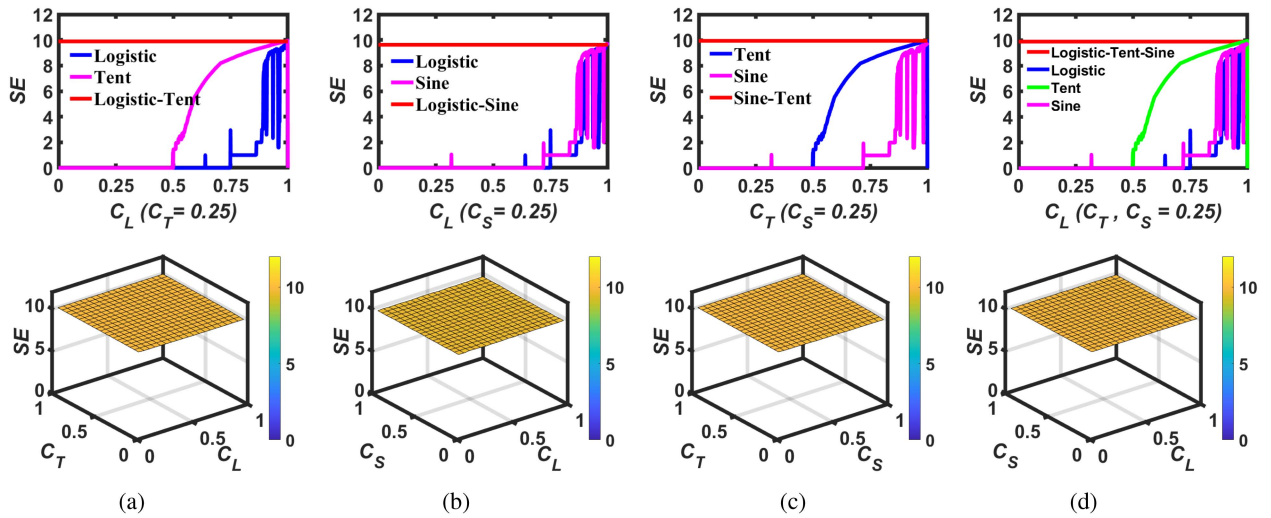


FIGURE 8. SE plot of four NLCS maps. (a)  $LT$ . (b)  $LS$ . (c)  $ST$ . (d)  $LTS$ .

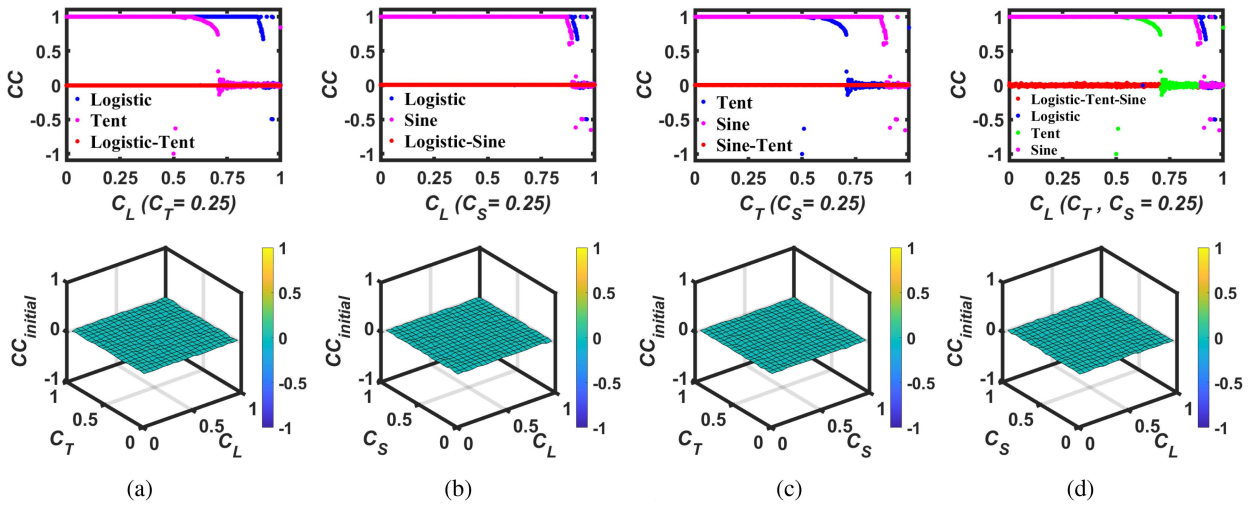


FIGURE 9. CC ( $CC_{initial}$ ) plot demonstrating initial value sensitivity of four NLCS maps. (a)  $LT$ . (b)  $LS$ . (c)  $ST$ . (d)  $LTS$ .

reconfigurable chaotic oscillator since each parameter configuration will generate a completely unique sequence (see Fig. 10) with excellent entropic properties as demonstrated in Figs. 6–8. Later, in Section IX, this attribute will be leveraged to build a new reconfigurable PRNG.

### V. PERFORMANCE IMPROVEMENT WITH CASCADING

It was shown in [21] that cascading multiple 1-D maps can significantly improve chaotic properties. Later, it was shown that this is true under certain constraints and not all maps are amenable to performance improvement via cascading [6]. As it turns out, cascading is particularly suitable for all combinations of NLCS maps. Cascading of two maps with independent parameter exponentially expands the parameter space while uniformly improving entropy metrics across the extended space. The schematic of the cascaded normalized

linearly-combined chaotic system (CNLCS) is presented in Fig. 11. The performance improvement in CNLCS is illustrated using two entropy measures, LE and KE for the cascaded connection of two NLCS maps. The constituent NLCS maps can be identical or different. Figs. 12 and 13 present a comparison between NLCS and CNLCS based on the LE and KE values, respectively. The plots show that CNLCS with two NLCS maps increase both LE and KE by almost a factor of two. Similarly, it can be shown that cascading  $n$  number of maps improves these entropy measures by almost a factor of  $n$ . In addition, each new cascaded map increases the number of parameters and exponentially extends the chaotic space.

### VI. HARDWARE IMPLEMENTATION USING FPGA

Recently, FPGA has gained popularity for implementing different types of chaotic systems [33], [34]. We have chosen

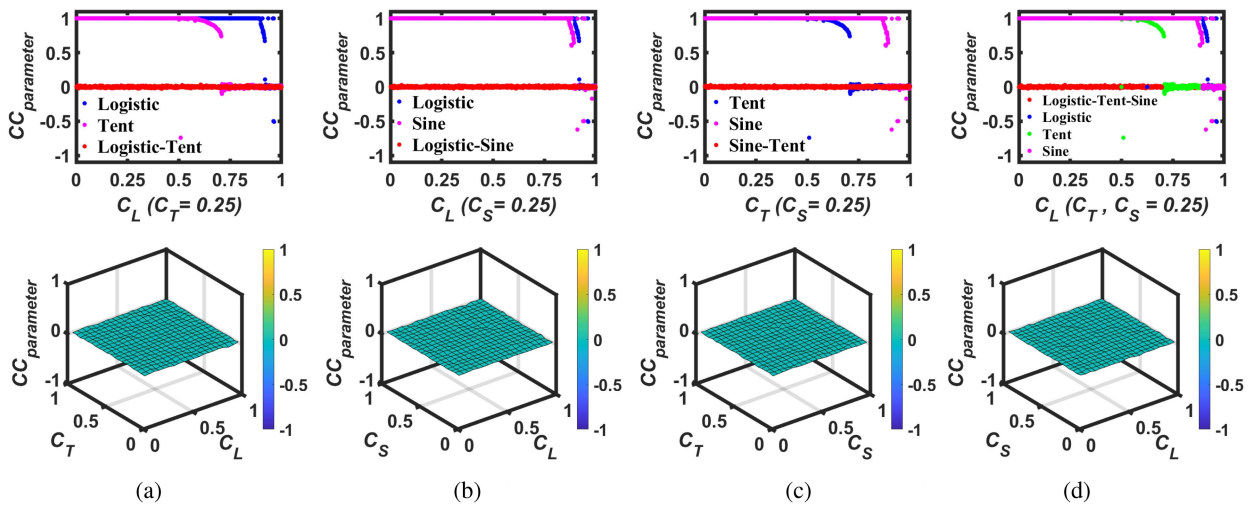


FIGURE 10. CC ( $CC_{parameter}$ ) plot demonstrating parameter sensitivity of four NLCS maps. (a)  $LT$ . (b)  $LS$ . (c)  $ST$ . (d)  $LTS$ .

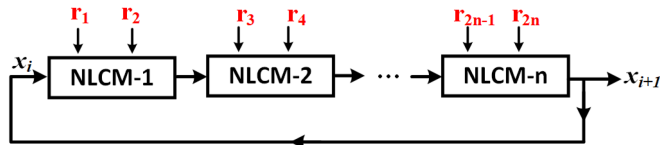


FIGURE 11. CNLCS scheme.

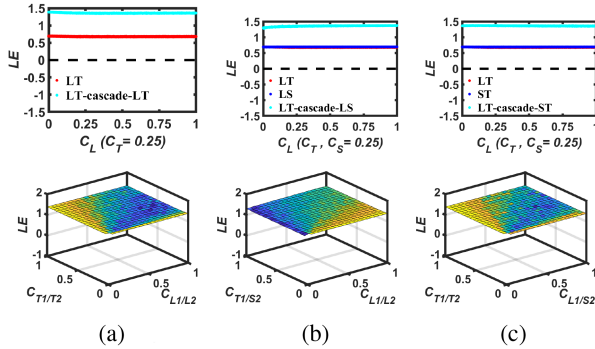


FIGURE 12. LE plots of different NLCS and CNLCS systems. (a)  $LT-LT$ . (b)  $LT-LS$ . (c)  $LT-ST$ .

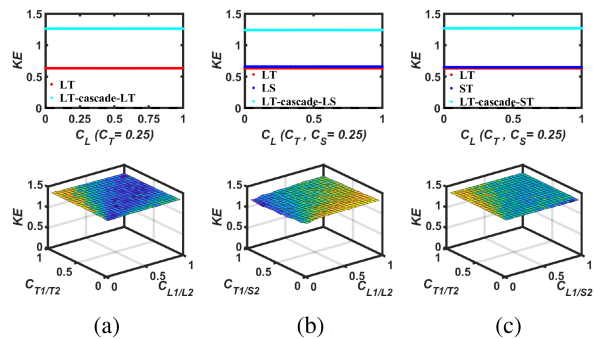


FIGURE 13. KE plots of different NLCS and CNLCS systems. (a)  $LT-LT$ . (b)  $LT-LS$ . (c)  $LT-ST$ .

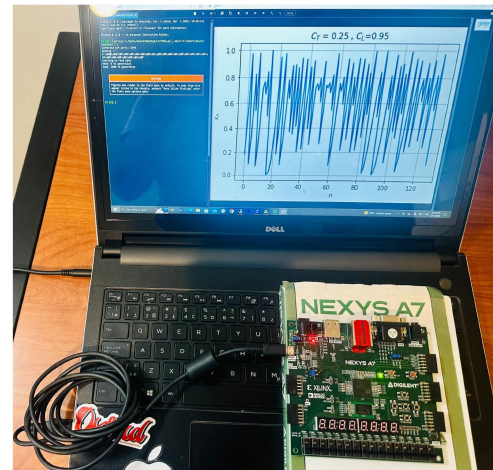


FIGURE 14. Experimental setup for FPGA implementation.

a Nexys A7 FPGA board as our hardware platform due to their affordability, reconfigurability, and high performance. We have implemented four types of NLCS system, namely,  $LT$ ,  $LS$ ,  $ST$ , and  $LTS$  in Nexys A7 FPGA board. An external device (e.g., our PC) communicates with the FPGA with universal asynchronous receiver-transmitter (UART) protocol, which is used for data collection for postprocessing and visualization in our computer. Fig. 14 shows our experimental setup. The hardware architecture and the FPGA implementation result are discussed in the following. Here, we have used  $LT$  as a specific example to explain some of the details but the principles are applicable for any NLCS.

### A. NUMBER REPRESENTATION

Since we are dealing with real numbers exclusively in the range  $[0, 1]$ , we have decided to develop our own fixed-point

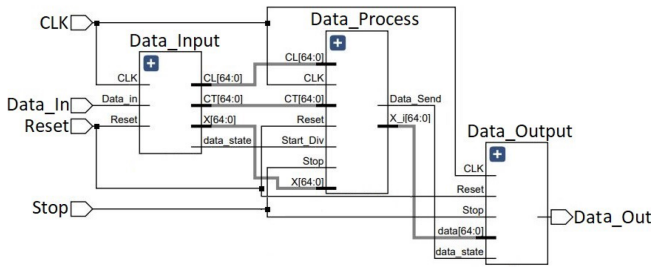


FIGURE 15. Schematic of the FPGA Implementation.

number representation system with  $n + 1$  binary bits ( $n : 0$ ) under the assumption of binary point after the  $n$ th bit. If we consider the  $n + 1$  bits as a binary integer, then the binary number 0 and  $2^n$  correspond to the real numbers 0 and 1, respectively. The following results are obtained for  $n = 64$ , which gives us a uniform high resolution ( $1/2^{64}$ ) across the entire range with less overhead compared to standard 64-bit IEEE-754 floating-point representation [35], which is meant to represent a much wider range of numbers, i.e.,  $[-2 \times 2^{1023}, +2 \times 2^{1023}]$  and consequently, can give the highest resolution of  $(1/2^{52})$  due to its 52-bit mantissa. Moreover, the finite precision of a digital system implies that it will never be possible to obtain an ideal infinitely aperiodic sequence since the system is bound to reach a previous state after a finite number of iterations, which dictates a periodic repetition due to the system’s deterministic nature. Hence, in practice, we strive to obtain the highest possible period out of a chaotic system. Due to our number representation scheme, the highest period achievable by our system is  $2^{63}$  compared to  $2^{52}$  in 64-bit IEEE-754 floating-point representation.

**B. HARDWARE ARCHITECTURE**

The proposed chaotic oscillators are designed in Verilog hardware description language and implemented in Nexys A7 FPGA board. As shown in the block diagram of Fig. 15, it has input and output communication modules to communicate with external devices. The circuit has four input ports and one output port as described in the following.

- 1) CLK: Provides the clock input for the digital circuits.
- 2) Data\_In: An UART Protocol enabled input pin, which accepts data from external devices.
- 3) Stop: A control input to stop all processes in the FPGA.
- 4) Reset: A control input to reset the system.
- 5) Data\_Out: An output port uses UART protocol to communicate individual data produced by the chaotic oscillator with external devices.

The system has three distinct parts as described in the following.

- 1) Data\_Input: This input processing module accepts information from an external device using UART protocol and outputs the initial condition for the chaotic map. External device can run the algorithm shown in Fig. 16. This sends  $3 \times n + 24$  bits of data under the

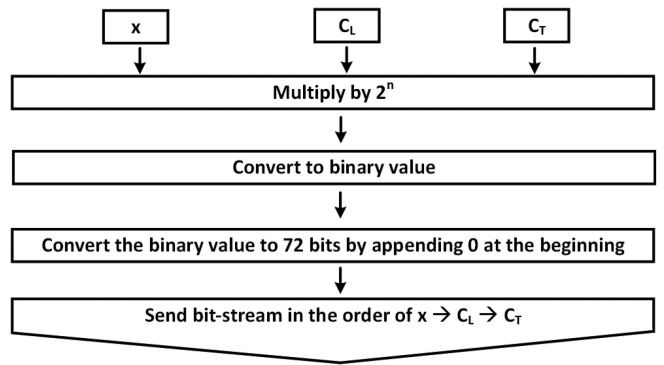


FIGURE 16. Algorithm for communication between the external device and Data\_Input module.

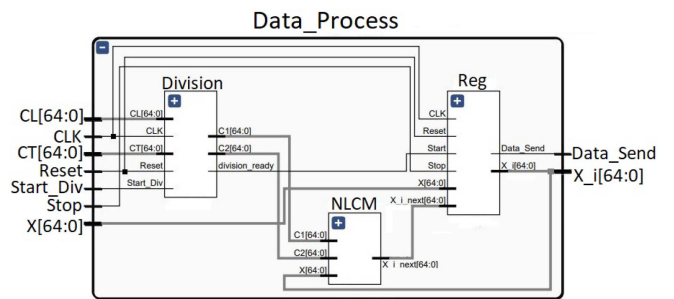


FIGURE 17. Schematic of the Data\_Process module.

UART protocol. The Data\_Input module receives the data and outputs the initial condition and parameters for the chaotic map ( $x_0, C_L$ , and  $C_T$ ), which are all  $n + 1$  bits in size. It also outputs a completion trigger bit to notify the next module in the pipeline to accept the initial condition and parameters.

- 2) Data\_Process: This data processing module is built as a finite-state machine to implement the proposed scheme, as shown in Fig. 17. It accepts the initial condition and parameters and outputs the iterated sequence according to the chaotic map. The normalization step in the proposed scheme requires division, which is slow compared to other operations. However, we observe that the normalizing factor does not change throughout iterations, and consequently, we need to do this only once before the iteration starts, which does not reduce the running throughput of the system. It still has a higher latency for the first output but that is less significant compared to throughput for iterated maps since we usually use these systems to generate a very long sequence of outputs. We elaborate this mechanism using  $\mathcal{LT}$  map function as an example. The transfer function of  $\mathcal{LT}$  map from (6) can be rewritten as

$$x_{i+1} = C_1 \times M_1(x_i) + C_2 \times M_2(x_i). \quad (25)$$

Here,  $C_1 = \frac{C_L}{C_L + C_T}$ ,  $C_2 = \frac{C_T}{C_L + C_T}$ ,  $M_1(x_i) = \mathcal{L}(x_i)$  for  $C_L = 1$  and  $M_2(x_i) = \mathcal{T}(x_i)$  for  $C_T = 1$ .  $C_1$  is calculated



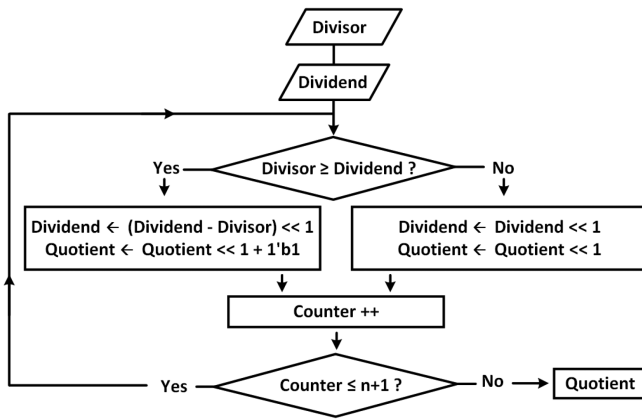


FIGURE 18. Division algorithm for NLCM.

using the highly efficient division algorithm shown in Fig. 18.  $C_2$  is calculated by subtracting  $C_1$  from 1. The inputs  $C_L$  and  $C_T$  and the outputs  $C_1$  and  $C_2$  from the division submodule are all  $\in [0, 1]$ , which satisfy the original assumption behind our chosen number representation. The circuit ignores calculation for  $C_L, C_T > 1$ . This saves  $n$ -bit register and  $n$  clock cycles in the operations performed by the circuit. The Reg submodule stores the current state of the chaotic system, i.e.,  $x_i$  and the NLCM submodule calculates the next state based on the current state from Reg module and precalculated  $C_1$  and  $C_2$  from the division submodule using (25).

- 3) Data\_Output: This module outputs the value produced by each iteration of the chaotic map. It is triggered by the Data\_Process module to send each new output of the iterated map. It appends the  $n + 1$ -bit data with leading zeros and converts it to  $n + 8$ -bit data. This  $n + 8$ -bit data is sent via UART protocol to the external device. The highly optimized hardware implementation ensures a throughput which is almost the same as the constituent seed maps (only fractionally lower due to an extra addition operation) while providing much better chaotic properties as shown in the following section.

### C. FPGA IMPLEMENTATION RESULT

Fig. 19 shows a comparison between MATLAB simulation and FPGA implementation results of  $\mathcal{LT}$  map for an initial condition  $x_0 = 0.75$  and parameter values,  $C_L = 0.90$  and  $C_T = 0.25$ . The series diverges after 50 iterations. This is due to our choice of a fixed-point number representation system for FPGA implementation (see Section-VI-A), which is different from the 64-bit IEEE-754 floating-point representation used in MATLAB simulation. We chose this representation to achieve a higher resolution in the desired range and modified arithmetic modules for efficient implementation. The tiny fluctuations resulting from this difference are amplified by the high susceptibility of the chaotic system to the slightest perturbation, which leads to the eventual divergence of these

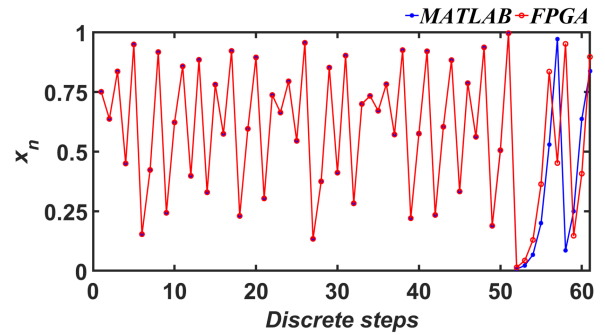


FIGURE 19. Comparison between MATLAB and FPGA implementation results for  $\mathcal{LT}$  map ( $x_0 = 0.75, C_L = 0.90, C_T = 0.25$ ).

two sequences. This divergence is not significant since, theoretically speaking, neither of the two implementations is more correct than the other one. In fact, our FPGA implementation has higher resolution in the desired range as pointed out in Section VI-A compared to MATLAB. The more important question for practical application is whether chaotic entropy values are similar in both implementations. To explore this, we have created two sets of discrete-time sequences with each sequence consisting of 14 000 steady-state values, one with MATLAB simulation and the other one with FPGA. Each set contains chaotic sequences for different parameter values. We have calculated the LE, KE, and SE values from the generated sequences for both cases, and this entire process is repeated for four NLCS maps, namely,  $\mathcal{LT}$ ,  $\mathcal{LS}$ ,  $\mathcal{ST}$ , and  $\mathcal{LTS}$ . Fig. 20 clearly shows an almost identical match between results from MATLAB and FPGA, thereby validating the potential of this efficient hardware implementation for diverse security applications.

### VII. COMPARISON WITH PRIOR WORKS

The first advantage of the proposed design is its much wider chaotic region, i.e., increase in the quantity of chaotic design space. The second advantage is the almost uniform high chaotic properties across the entire chaotic range, i.e., improvement of quality of chaotic operation. If a system has  $p$  parameters and each parameter can have  $N$  distinct values, then the EPS can be defined as  $\text{EPS} = N^p$  [36]. A subset of this space is chaotic, which we call chaotic parameter space (CPS). We use a metric [36] named chaotic ratio (CR), which is defined as the ratio of CPS to EPS

$$\text{CR}(\%) = \frac{\text{CPS}}{\text{EPS}} \times 100. \quad (26)$$

For quality assessment, we are averaging LE, SE, KE, and the absolute value of CC across the chaotic region to come up with a single global metric for each entropy measure. Higher average LE (ALE), average KE (AKE), and average SE (ASE) imply better entropic properties. Similarly, a lower average CC magnitude (ACC) closer to zero implies more initial state sensitivity, i.e., better chaotic quality. We also report the maximum value of LE, KE, and SE (MLE, MKE, and MSE) and



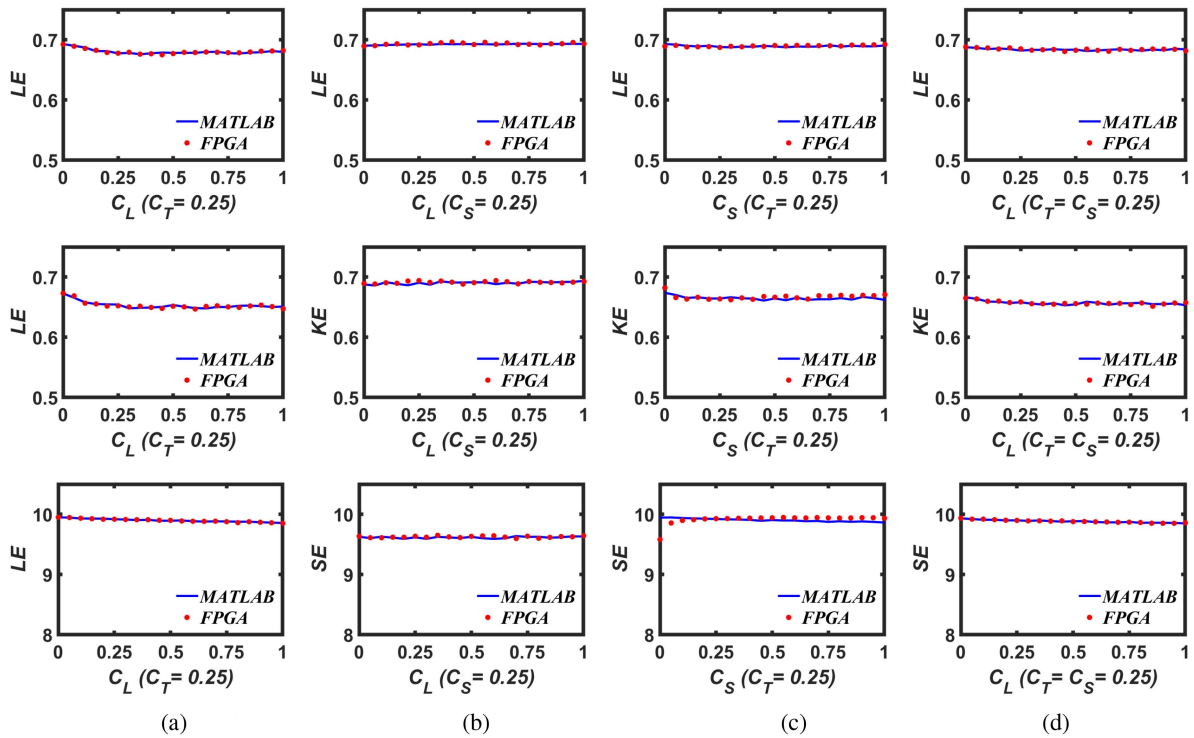


FIGURE 20. Comparison of entropy metrics between MATLAB simulation and FPGA implementation. (a)  $\mathcal{L}\mathcal{T}$ . (b)  $\mathcal{L}\mathcal{S}$ . (c)  $\mathcal{S}\mathcal{T}$ . (d)  $\mathcal{L}\mathcal{T}\mathcal{S}$ .

TABLE 1. Comparison of Chaotic Performance

Design/metrics	ALE	MLE	AKE	MKE	ASE	MSE	ACC <sub>init</sub>	mCC <sub>init</sub>	ACC <sub>par</sub>	mCC <sub>par</sub>	CPS ( $N = 2^{10}$ )	CR (%)	ANDR (%)
Logistic	0.388	0.694	0.397	0.693	8.93	9.69	0.244	$1.1 \times 10^{-4}$	0.247	$4.14 \times 10^{-5}$	101.27	9.89	76.25
Tent	0.393	0.693	0.334	0.67	7.63	9.95	0.391	$1.47 \times 10^{-5}$	0.387	$4.4 \times 10^{-5}$	502.28	49.05	42.29
Sine	0.408	0.689	0.417	0.681	8.96	9.7	0.195	$4.55 \times 10^{-5}$	0.193	$3.12 \times 10^{-6}$	121.73	11.89	75.09
ZBC(LT) [7]	0.684	0.71	0.658	0.693	9.88	9.95	0.0072	$4.55 \times 10^{-5}$	0.0068	$1.32 \times 10^{-6}$	1024	100	99.89
DPCCS(LT) [19]	0.45	0.683	0.464	0.964	8.94	9.91	0.272	$2.05 \times 10^{-5}$	0.272	$5.05 \times 10^{-5}$	1024	100	59.74
ECM(LT) [15]	0.676	0.695	0.653	0.695	9.68	9.74	0.007	$3.32 \times 10^{-5}$	0.0064	$4.46 \times 10^{-5}$	$1.05 \times 10^6$	100	99.99
CNLCs(LT-LT)	<b>1.364</b>	<b>1.387</b>	<b>1.304</b>	<b>1.329</b>	<b>9.91</b>	<b>9.95</b>	<b>0.007</b>	<b><math>3.97 \times 10^{-6}</math></b>	<b>0.0067</b>	<b><math>7.55 \times 10^{-6}</math></b>	<b><math>1.1 \times 10^{12}</math></b>	<b>100</b>	<b>99.99</b>

Boldface was used for our proposed system. The goal is to distinguish our current work from the other works in the table which are references we compare our work against.

the minimum absolute value of two types of CC (mCC). In addition, the dynamic swing range of the steady-state output inside the chaotic region should be as close to the highest output range ( $R$ ) as possible to ensure the maximum unpredictability. For capturing this aspect of chaotic operation, we use a metric [36] named average normalized dynamic range (ANDR) defined as

$$\text{ANDR}(\%) = \left( \frac{1}{\text{CPS}} \sum_{i \in \text{CPS}} \frac{V_{\max}^i - V_{\min}^i}{R} \right) \times 100. \quad (27)$$

Table 1 compares our proposed design, CNLCS (in bold) with the three basic seed maps as well as three previous works, namely ZBC [7], dynamic parameter-control chaotic system (DPCCS) [19], and exponential chaotic model (ECM) [15] using the abovementioned metrics and it shows significant improvement considering all aspects of the chaotic operation.

In addition, we have implemented the prior works along with our proposed system in the Nexys A7 FPGA board and compared the hardware implementation metrics, such as resources, power consumption, and speed (clock cycle/iteration)

of our system against 6 prior works as shown in Table 2. Tables 1 and 2 show that our proposed system provides significantly superior chaotic performance with moderate hardware cost. For example, techniques, such as ECM [15], achieve uniformly robust chaos using logarithms and exponentiation, which are computationally much more expensive compared to NLCS. In addition, unlike these prior works, two NLCS maps can be easily combined without any additional hardware to form a multiparameter 2-D robust, hyperchaotic system as will be shown in Section VIII.

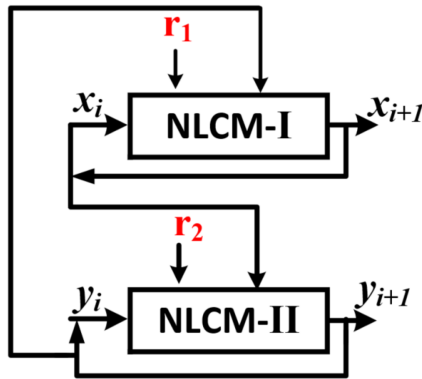
### VIII. EXTENSION TO ROBUST AND HYPERCHAOTIC 2-D MAPS

A dynamical system is hyperchaotic when it has more than one positive LE, i.e., its trajectories will diverge in several directions. Since this dynamic behavior is more complex than chaotic behavior, it has attracted the attention of researchers in recent times [37], [38]. In this section, we propose a simple cross-coupling technique to design a 2-D robust, hyperchaotic system with uniformly excellently chaotic properties. As shown in Fig. 21, two NLCS maps, NLCS-I and

**TABLE 2. Comparison of Hardware Implementation Metrics in Artix 7 FPGA With 100 MHz Clock**

Design	LUT	FF	DSP	BUFG	LUTRAM	Power (mW)	Speed (cycle/iteration)
Logistic	1005	316	48	3	-	190	2
Tent	319	316	16	3	-	128	2
Sine	2055	316	176	3	-	406	2
ZBC(LT) [7]	1408	387	64	3	-	211	2
DPCCS(LT) [19]	1354	518	64	3	-	203	2
ECM(LT) [15]	9904	14564	127	4	1646	511	141
<b>CNLCS(LT-LT)</b>	<b>2941</b>	<b>791</b>	<b>128</b>	<b>3</b>	-	<b>350</b>	<b>2</b>

Boldface was used for our proposed system. The goal is to distinguish our current work from the other works in the table which are references we compare our work against.



**FIGURE 21. Schematic of 2-D NLCS.**

NLCS-II are cross-coupled where the state variable of one map is connected to the second parameter of the other map. The resulting 2-D map has two independent parameters and two state variables  $x_i$  and  $y_i$ . The method is general and we can choose any NLCS maps as map I and II. For example, a 2-D map  $LT-LS$  implies that  $LT$  and  $LS$  are used as NLCS-I and NLCS-II, respectively (see Fig. 21).

Fig. 22 shows the LE values for three different 2-D maps, namely  $LT-LS$ ,  $LT-ST$ , and  $LS-ST$ , generated using the abovementioned scheme. A 2-D map has two LE values ( $\lambda_1$  and  $\lambda_2$ ) and as shown in Fig. 22, both LE values for our 2-D NLCS systems are positive across the EPS with uniformly high LE values exhibiting both robust and hyperchaotic behavior.

Similar to 1-D NLCS, the performance and parameter space of this 2-D extension can also be improved via a simple cascading scheme, as shown in Fig. 23. We have kept same parameter for two maps in cascade to keep the analysis simple, but in general there can be four independent parameters and the configuration space increases exponentially with the number of parameters. The doubling of both LE values across the EPS due to this cascading mechanism for all three 2-D systems are shown in Fig. 24.

### IX. NOVEL RECONFIGURABLE PRNG USING NLCS

PRNGs are used as critical security primitives in cryptographic application and information security [39], [40]. The defining properties of chaotic systems, namely deterministic aperiodicity and acute susceptibility to any perturbation in initial condition render them ideal candidates for building

**TABLE 3. Six Different Parameter Configurations for the Proposed PRNG**

PRNG	$r_1$	$r_2$	$r_3$	$r_4$	$r_5$	$r_6$
1	0.25	1	0.25	1	0.25	1
2	0.8	0.4	0.02	0.6	0.15	0.95
3	0.01	0.07	0.2	0.15	0.15	1
4	0.9	0.25	0.02	0.7	0.01	0.5
5	0.05	1	0.9	0.2	0.3	0.1
6	0.04	0.3	0.8	0.6	0.1	0.5

PRNGs [3], [41], [42]. Here, we present a new reconfigurable multiparameter PRNG leveraging the robust chaotic operation, uniformly high entropy, and availability of multiple independent parameters in NLCS.

The schematic of the proposed PRNG is shown in Fig. 25. We have two parallel chaotic oscillators, one using NLCS and the other one using CNLCS. We are using the LT map as the NLCS in the construction of this PRNG. At every iteration, we extract 8 bits (13:30) from the 64-bit output and XOR them to produce the final 8-bit output, i.e., a throughput of 8 bits/iteration. NLCS provides two parameters ( $r_1$  and  $r_2$ ) whereas CNLCS provides four additional parameters ( $r_3$ ,  $r_4$ ,  $r_5$ , and  $r_6$ ). Due to the uniform chaotic properties of NLCS and CNLCS, this PRNG is reconfigurable across the entirety of its six-dimensional parameter space. To illustrate the reconfigurability, we have chosen six different parameter configurations, which are shown in Table 3 and for each configuration, the excellent randomness of the proposed PRNG has been verified using two statistical randomness tests, namely National Institute of Standards and Technology (NIST) and FIPS.

#### A. NIST SP 800-22

This test suite from the NIST offers 15 statistical subtests to measure the randomness in a sequence [43]. For each one of the six configurations, we ran the test with 100 bit-streams generated from 100 different initial condition with each bit-stream having a length of 1 million bits. The significance level was set to 0.01. Hence, a sequence with 100 million bits (containing 100 bit-streams) will pass a particular test if at least 96 out of the 100 bit-streams generate a  $p$ -values greater than 0.01. The test suite allocates each of the 100 generated  $p$ -values in 10 subintervals from 0 to 1 and evaluates the uniformity in the distribution with  $\chi^2$ -test. The sequence

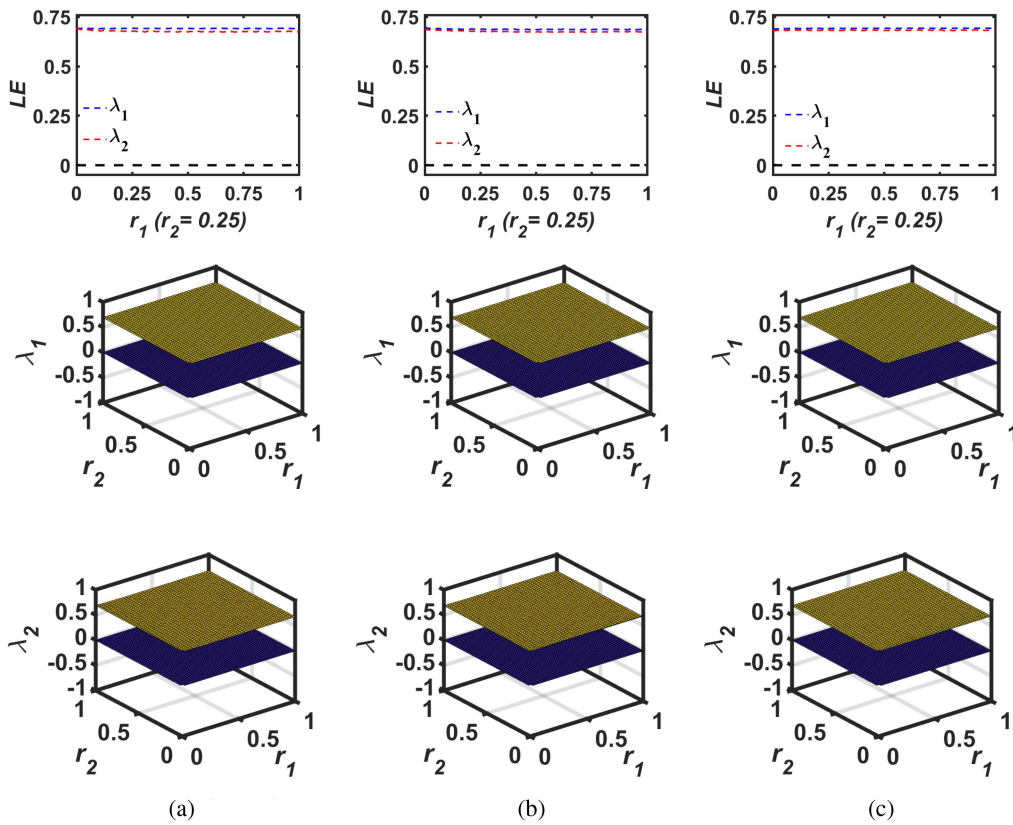


FIGURE 22. LE plots of different 2D-NLCS systems. (a)  $\mathcal{LT}\text{-}\mathcal{LS}$ . (b)  $\mathcal{LT}\text{-}\mathcal{ST}$ . (c)  $\mathcal{LS}\text{-}\mathcal{ST}$ .

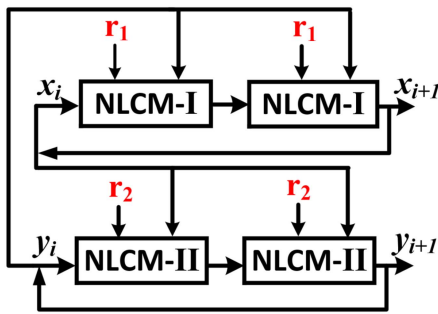


FIGURE 23. Cascaded 2DNLCS scheme.

under test can be considered uniform if the  $p$ -value generated from the  $\chi^2$ -test (refers to  $p\text{-value}_T$ ) is greater than or equal to 0.0001. Table 4 shows that the proposed reconfigurable PRNG passes all requirements of 15 subtests for six different parameter configurations.

**B. FIPS PUB 140-2**

The Federal Information Processing Standards Publications FIPS PUB 140-2 test suite was developed by NIST [44]. FIPS tests the randomness of a binary sequence by dividing the sequence into 20 000-bit blocks. Hence, for a test sequence with 100 million bits, there will be 5000 blocks in total. The

blocks are subjected to 4 subtests namely, Monobit, Poker, Runs, and Long run. The Monobit test counts the number of 1s in each 20 000-bit block. To pass the test, this number must be within the range of [9725, 10 275]. The Poker test divides each 20 000-bit block into 5000 successive 4-bit segments. The 4-bit segment can have 16 possible values. The occurrences of 16 values are counted and stored. This subtest examines the uniformity of the 4-bit segment. Runs test counts and stores the maximum sequence of consecutive 1s or 0s in a 20 000-bit block. A run of 26 or more of either 1s or 0s is defined as a long run. The total number of long runs in a 20 000-bit block is counted as the total failure. Table 5 shows the FIPS test result for each one of the six configurations of the proposed PRNG. The second column (from the left) of Table 5 shows the total number of blocks passing the test out of the total 5000 blocks and the last four columns show the number of failed blocks under corresponding subtests. The results show close to 100% success implying great randomness.

**X. APPLICATIONS**

We outline the following six application scenarios where the particular attributes of the proposed NLCS system will be useful.

- 1) *Reconfigurable random number generator*: Random number generator are used in many applications including but not limited to Monte Carlo simulations,

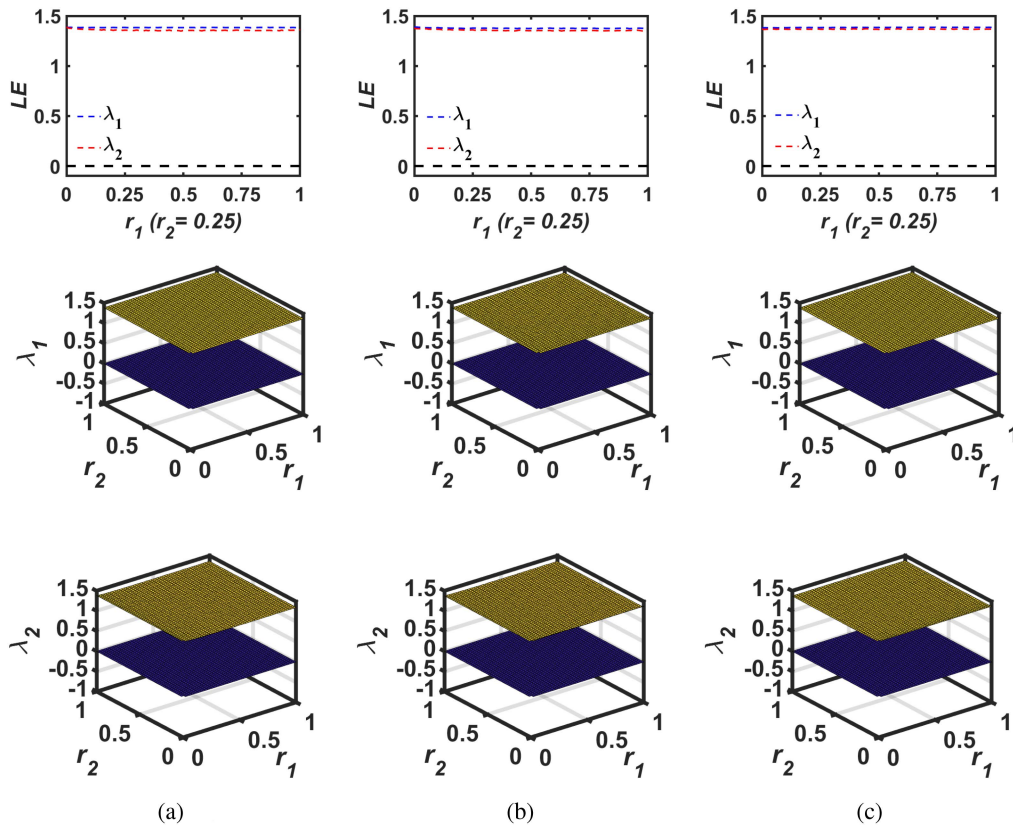


FIGURE 24. LE plots of three different cascaded 2D-NLCS systems. (a)  $LT-LS$ . (b)  $LT-ST$ . (c)  $LS-ST$ .

TABLE 4. NIST Results (1 to 6 Are Six Different Configurations; \*Shows Average of Multiple Tests)

NIST TEST	Pass rate(%)						P-value $_T$					
	1	2	3	4	5	6	1	2	3	4	5	6
Frequency	99	98	99	100	98	99	0.046	0.367	0.898	0.575	0.172	0.289
Block frequency	97	98	98	100	99	99	0.006	0.834	0.456	0.202	0.637	0.367
Cumulative sums*	99	98	99	100	98.5	99	0.938	0.208	0.692	0.468	0.305	0.679
Runs	96	100	99	99	99	99	0.514	0.964	0.575	0.071	0.936	0.249
Longest runs of ones	100	100	100	100	100	100	0.637	0.76	0.834	0.514	0.956	0.456
Rank	98	99	99	99	99	100	0.35	0.115	0.534	0.554	0.946	0.74
FFT	100	99	99	97	99	97	0.74	0.817	0.616	0.898	0.097	0.35
Non-overlapping template*	99.03	99.04	99.02	99.98	99.06	99.02	0.504	0.5	0.483	0.501	0.515	0.469
Overlapping template	97	99	99	98	100	97	0.213	0.817	0.401	0.991	0.419	0.29
Universal	99	100	98	99	97	99	0.063	0.596	0.172	0.658	0.72	0.109
Approximate entropy	99	100	99	100	98	99	0.304	0.163	0.911	0.575	0.401	0.335
Random excursion*	98.68	99.77	98.8	98.38	99.78	99.08	0.276	0.451	0.548	0.42	0.232	0.504
Random excursion variant*	99.81	100	98.93	98.83	99.9	99.26	0.289	0.464	0.531	0.644	0.286	0.275
Serial*	99.5	100	100	99.5	98.5	100	0.554	0.385	0.673	0.408	0.658	0.787
Linear complexity	100	97	99	98	99	100	0.514	0.29	0.384	0.437	0.991	0.456

test pattern generation, scientific experiments, cryptography, and telecommunication systems [42], [45], [46], [47], [48]. Due to their excellent ergodic properties, chaotic maps have been extensively used in designing PRNG [4], [30], [49]. Many chaotic random number generators are designed for a fixed parameter, i.e., for the same seed, it always generates the same sequence, which makes them vulnerable to adversarial attacks [3], [6]. As shown in Section IX, NLCS can be used to build reconfigurable PRNG with excellent randomness across

a very large design space. For a specific seed, a run-time change in configuration by even a single bit of any of the six control parameters will produce a completely uncorrelated yet equally good random sequence, which gives this design a significant immunity against adversarial attacks [50]. Besides, the uniformly high chaotic entropy across parameter space makes NLCS-based PRNG immune against performance degradation due to parameter disturbance. Moreover, the hardware implementation metrics of NLCS (see Table 2) along with



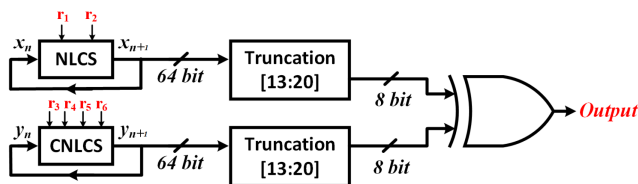


FIGURE 25. Schematic of the proposed PRNG.

TABLE 5. FIPS Test Results

PRNG	Total success	Monobit	Poker	Runs	Long run
<b>1</b>	<b>4998</b>	-	-	1	1
<b>2</b>	<b>4997</b>	1	-	1	1
<b>3</b>	<b>4997</b>	-	1	1	1
<b>4</b>	<b>4998</b>	-	1	-	1
<b>5</b>	<b>4997</b>	1	1	1	-
<b>6</b>	<b>4997</b>	-	1	1	1

Boldface was used for stylistic reason, to highlight total results as opposed to individual result shown without boldface.

the simplicity of the proposed PRNG makes it suitable for resource-constrained edge computing in Internet of Things in contrast to some prior works requiring much higher computational resources [30], [49].

- 2) *Secure communication*: During the last three decades, many researchers have leveraged chaotic dynamics for developing secure communication systems [15], [29], [49], [51]. As shown in [15], a discrete-time chaotic system with uniformly robust chaotic dynamics (ECM) can be a perfect candidate for improving the system’s immunity against channel noise. As shown in Tables 1 and 2, NLCS is superior to ECM both in terms of chaotic performance and hardware cost and as such, it will be an even better building block for developing such secure communication systems.
- 3) *Image encryption*: Since digital image has a lot of information redundancy, traditional stream/block cipher based well-known encryption methods, such as digital encryption standard [52], advanced encryption standard [53], etc., may not be the optimum choice for such data. To circumvent this issue, there has been a significant body of research on developing image encryption algorithms based on chaotic maps [30], [42], [54]. Usually, a secure key is used as the initial condition and/or parameter value of chaotic maps to generate a long sequence of unpredictable values, which are then used to encrypt the input image using a particular algorithm [21], [25], [54], [55], [56]. The success of any such algorithm depends on a large part on the entropic quality of the chaotic map. Given the excellent entropic properties of NLCS across the entire parameter range with low hardware cost, it can be easily integrated with any such algorithm for image encryption application.
- 4) *Reconfigurable computing*: Starting from the seminal 1998 paper [57], researchers have been exploring how

the chaotic dynamics can be utilized to build flexible and reconfigurable computing blocks sometimes called “chaogates” [11], [58]. The aperiodic iteration inside chaotic region means that we can extract a large number of functions from a single chaotic system [59], [60]. As shown in [36] and [61], the CPS plays a key role in expanding the reconfigurability of such system. This can be leveraged for logic locking [18] to prevent integrated circuit (IC) counterfeiting, and reverse engineering, which have become a serious threat in the current IC supply chain. Since multiparameter robust NLCS offers chaotic operation across a large parameter space, it can be a perfect candidate for building chaos-based reconfigurable computing platforms.

- 5) *Side-channel attack mitigation*: Starting with the seminal work of Kocher [62], Side-channel attack has emerged as a serious threat to computer security in recent years where information leaked through side-channels, such as power consumption, electromagnetic emanation, timing information, keystroke behavior, etc., have been used by adversary to extract valuable secret information [63], [64], [65], [66]. Obfuscation via Chaos-based reconfigurable logic has been proposed and explored as a mitigation technique in several recent works [13], [67], [68]. However, for this mitigation technique to be successful, we need a wide chaotic region with good entropic properties [13], which makes NLCS a suitable candidate for such applications.
- 6) *multi-D and multiparameter hyperchaotic system*: It has been shown that 1-D chaotic systems with their relatively simpler orbit can be susceptible to signal estimation attack [69] and dynamic degradation in a digitized platform [70], [71]. This is a hindrance toward their adoption in cryptographic applications where high level of security is required [72]. The state space of a chaotic system increases exponentially with the number of dimensions and a multi-D chaotic system becomes hyperchaotic when it has more than one positive LE [49], [73], [74]. This gives rise to a significantly more complex trajectory compared to 1-D chaotic system [49], [56] and can find use in different applications [29], [30], [49], [75]. Oftentimes, simpler 1-D maps are chosen instead of these hyperchaotic maps due to their prohibitively higher cost of hardware implementation. As shown in Section VIII, NLCS can be easily extended to a 2-D hyperchaotic map with uniformly high and robust entropic properties across an exponentially larger parameter space and state space at the same throughput while incurring only twice the hardware cost of its 1-D counterpart. The same design principle can be easily extended to build even higher dimensional hyperchaotic NLCS systems. These multi-D NLCS maps can be a promising low-cost robust hyperchaotic alternative for diverse security applications [30], [49].

## XI. CONCLUSION

A general framework called NLCS for developing arbitrary number of new multiparameter 1-D and 2-D chaotic system from existing seed maps is presented in this work. The chaotic performance is analyzed using stability analysis and bifurcation diagram along with four established metrics, namely, Lyapunov exponent, KE, SE, and CC. Unlike the seed maps, the entropy values in NLCS remain uniformly high across the whole range and the value is always close to the maximum achievable value from the constituent seed maps. The CPS and ergodic properties are further enhanced by cascading multiple maps. We have shown an efficient FPGA-based hardware implementation. The comparison of performance and hardware cost with seed maps and prior literature shows the superior properties of NLCS. Moreover, we introduced a simple extension scheme to build 2-D maps with robust, hyperchaotic, and uniformly excellent properties across the parameter space. We presented a new reconfigurable multiparameter PRNG and validated its excellent randomness property using two standard statistical tests, namely, NIST SP 800-22 and FIPS PUB 140-2. Finally, we outlined six application scenarios where the particular attributes of the proposed system will be useful.

## REFERENCES

- [1] E. N. Lorenz, "Deterministic nonperiodic flow," *J. Atmospheric Sci.*, vol. 20, no. 2, pp. 130–141, 1963.
- [2] S. H. Strogatz, *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*. Boca Raton, FL, USA: CRC Press, 2018.
- [3] C.-Y. Li, Y.-H. Chen, T.-Y. Chang, L.-Y. Deng, and K. To, "Period extension and randomness enhancement using high-throughput reseeding-mixing PRNG," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 20, no. 2, pp. 385–389, Feb. 2012.
- [4] Z. Hua, B. Zhou, and Y. Zhou, "Sine chaotification model for enhancing chaos and its hardware implementation," *IEEE Trans. Ind. Electron.*, vol. 66, no. 2, pp. 1273–1284, Feb. 2019.
- [5] R. Agrawal, L. Bu, E. Del Rosario, and M. A. Kinsky, "Towards programmable all-digital true random number generator," in *Proc. Great Lakes Symp. VLSI*, 2020, pp. 53–58.
- [6] P. S. Paul, M. Sadia, M. R. Hossain, B. Muldrey, and M. S. Hasan, "Design of a low-overhead random number generator using CMOS-based cascaded chaotic maps," in *Proc. Great Lakes Symp. VLSI*, 2021, pp. 109–114.
- [7] Y. Zhou, L. Bao, and C. P. Chen, "A new 1D chaotic system for image encryption," *Signal Process.*, vol. 97, pp. 172–182, 2014.
- [8] Z. Hua and Y. Zhou, "Image encryption using 2D logistic-adjusted-sine map," *Inf. Sci.*, vol. 339, pp. 237–253, 2016.
- [9] Z. Hua, Z. Zhu, S. Yi, Z. Zhang, and H. Huang, "Cross-plane colour image encryption using a two-dimensional logistic tent modular map," *Inf. Sci.*, vol. 546, pp. 1063–1083, 2021.
- [10] B. Kia, K. Mobley, and W. L. Ditto, "An integrated circuit design for a dynamics-based reconfigurable logic block," *IEEE Trans. Circuits Syst. II: Exp. Briefs*, vol. 64, no. 6, pp. 715–719, Jun. 2017.
- [11] W. L. Ditto and S. Sinha, "Exploiting chaos for applications," *Chaos: Interdiscipl. J. Nonlinear Sci.*, vol. 25, no. 9, Art. no. 097615, 2015.
- [12] K. Gołofit and P. Z. Wiczcerek, "Chaos-based physical unclonable functions," *Appl. Sci.*, vol. 9, no. 5, 2019, Art. no. 991.
- [13] M. S. Hasan, M. B. Majumder, A. S. Shanta, M. Uddin, and G. S. Rose, "A chaos-based complex micro-instruction set for mitigating instruction reverse engineering," *J. Hardware Syst. Secur.*, vol. 4, no. 2, pp. 69–85, 2020.
- [14] R. Trejo-Guerra, E. Tlelo-Cuautle, C. Cruz-Hernández, and C. Sánchez-López, "Chaotic communication system using Chua's oscillators realized with CCHs," *Int. J. Bifurcation Chaos*, vol. 19, no. 12, pp. 4217–4226, 2009.
- [15] Z. Hua and Y. Zhou, "Exponential chaotic model for generating robust chaos," *IEEE Trans. Syst., Man, Cybern.: Syst.*, vol. 51, no. 6, pp. 3713–3724, Jun. 2021.
- [16] C. Liang, Q. Zhang, J. Ma, and K. Li, "Research on neural network chaotic encryption algorithm in wireless network security communication," *EURASIP J. Wireless Commun. Netw.*, vol. 2019, no. 1, 2019, Art. no. 151.
- [17] J. F. Lindner, V. Kohar, B. Kia, M. Hippke, J. G. Learned, and W. L. Ditto, "Strange nonchaotic stars," *Phys. Rev. Lett.*, vol. 114, no. 5, 2015, Art. no. 054101.
- [18] A. S. Shanta, M. B. Majumder, M. S. Hasan, and G. S. Rose, "Physically unclonable and reconfigurable computing system (PURCS) for hardware security applications," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 40, no. 3, pp. 405–418, Mar. 2021.
- [19] Z. Hua and Y. Zhou, "Dynamic parameter-control chaotic system," *IEEE Trans. Cybern.*, vol. 46, no. 12, pp. 3330–3341, Dec. 2016.
- [20] P. S. Paul, A. Dhungel, M. Sadia, M. R. Hossain, B. Muldrey, and M. S. Hasan, "Self-parameterized chaotic map: A hardware-efficient scheme providing wide chaotic range," in *Proc. IEEE 28th Int. Conf. Electron., Circuits, Syst.*, 2021, pp. 1–5.
- [21] Y. Zhou, Z. Hua, C.-M. Pun, and C. P. Chen, "Cascade chaotic system with applications," *IEEE Trans. Cybern.*, vol. 45, no. 9, pp. 2001–2012, Sep. 2015.
- [22] Y. Wu, Y. Zhou, and L. Bao, "Discrete wheel-switching chaotic system and applications," *IEEE Trans. Circuits Syst. I: Regular Papers*, vol. 61, no. 12, pp. 3469–3477, Dec. 2014.
- [23] M. S. Hasan, P. S. Paul, M. Sadia, and M. R. Hossain, "Integrated circuit design of an improved discrete chaotic map by averaging multiple seed maps," in *Proc. SoutheastCon*, 2021, pp. 1–6.
- [24] Z. Hua, B. Zhou, and Y. Zhou, "Sine-transform-based chaotic system with FPGA implementation," *IEEE Trans. Ind. Electron.*, vol. 65, no. 3, pp. 2557–2566, 2017.
- [25] H. Zhu, Y. Zhao, and Y. Song, "2D logistic-modulated-sine-coupling-logistic chaotic map for image encryption," *IEEE Access*, vol. 7, pp. 14081–14098, 2019.
- [26] L. Chua, "Memristor-the missing circuit element," *IEEE Trans. Circuit Theory*, vol. CT-18, no. 5, pp. 507–519, Sep. 1971.
- [27] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, "The missing memristor found," *Nature*, vol. 453, no. 7191, pp. 80–83, 2008.
- [28] B. Bao, K. Rong, H. Li, K. Li, Z. Hua, and X. Zhang, "Memristor-coupled logistic hyperchaotic map," *IEEE Trans. Circuits Syst. II: Exp. Briefs*, vol. 68, no. 8, pp. 2992–2996, Aug. 2021.
- [29] H. Li, Z. Hua, H. Bao, L. Zhu, M. Chen, and B. Bao, "Two-dimensional memristive hyperchaotic maps and application in secure communication," *IEEE Trans. Ind. Electron.*, vol. 68, no. 10, pp. 9931–9940, Oct. 2021.
- [30] Q. Lai, L. Yang, and G. Chen, "Design and performance analysis of discrete memristive hyperchaotic systems with stuffed cube attractors and ultraboosting behaviors," *IEEE Trans. Ind. Electron.*, to be published, doi: [10.1109/TIE.2023.3299016](https://doi.org/10.1109/TIE.2023.3299016).
- [31] P. Grassberger and I. Procaccia, "Estimation of the Kolmogorov entropy from a chaotic signal," *Phys. Rev. A*, vol. 28, no. 4, 1983, Art. no. 2591.
- [32] R. Frigg, "In what sense is the Kolmogorov-sinai entropy a measure for chaotic behaviour? bridging the gap between dynamical systems theory and communication theory," *Brit. J. Philosophy Sci.*, vol. 55, pp. 411–434, 2004.
- [33] E. Tlelo-Cuautle, J. Rangel-Magdaleno, A. D. Pano-Azucena, P. Obeso-Rodelo, and J. C. Nuñez-Perez, "Fpga realization of multi-scroll chaotic oscillators," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 27, no. 1–3, pp. 66–80, 2015.
- [34] M. Tuna, M. Alçin, İ. Koyuncu, C. B. Fidan, and İ. Pehlivan, "High speed FPGA-based chaotic oscillator design," *Microprocessors Microsystems*, vol. 66, pp. 72–80, 2019.
- [35] D. Zuras et al., *IEEE Standard for Floating-Point Arithmetic*, IEEE Standard 754-2019, 2008.
- [36] M. Sadia, P. S. Paul, M. R. Hossain, B. Muldrey, and M. S. Hasan, "Robust chaos with novel 4-transistor maps," *IEEE Trans. Circuits Syst. II: Exp. Briefs*, vol. 70, no. 3, pp. 914–918, Mar. 2023.
- [37] Z. Hua, Y. Zhang, and Y. Zhou, "Two-dimensional modular chaotification system for improving chaos complexity," *IEEE Trans. Signal Process.*, vol. 68, pp. 1937–1949, Mar. 12, 2020.
- [38] M. Ma et al., "A locally active discrete memristor model and its application in a hyperchaotic map," *Nonlinear Dyn.*, vol. 107, no. 3, pp. 2935–2949, 2022.

- [39] I. V. Chugunkov, M. A. Ivanov, E. A. Gridneva, and N. Y. Shestakova, "Classification of pseudo-random number generators applied to information security," in *Proc. IEEE Conf. Russian Young Researchers Elect. Electron. Eng.*, 2017, pp. 370–373.
- [40] Y. Dodis, D. Pointcheval, S. Ruhault, D. Vergniaud, and D. Wichs, "Security analysis of pseudo-random number generators with input: /dev/random is not robust," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2013, pp. 647–658.
- [41] M. Garcia-Bosque, A. Pérez-Resca, C. Sánchez-Azqueta, C. Aldea, and S. Celma, "Chaos-based bitwise dynamical pseudorandom number generator on FPGA," *IEEE Trans. Instrum. Meas.*, vol. 68, no. 1, pp. 291–293, Jan. 2019.
- [42] R. B. Naik and U. Singh, "A review on applications of chaotic maps in pseudo-random number generators and encryption," *Ann. Data Sci.*, pp. 1–26, 2022. [Online]. Available: <https://doi.org/10.1007/s40745-021-00364-7>
- [43] L. E. Bassham et al. *A statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*. Gaithersburg, MD, USA: National Institute of Standards and Technology, 2010.
- [44] *FIPS PUB140-2: Security Requirements for Cryptographic Modules*, Inf. Technol. Lab., Nat. Inst. Standards Technol., Gaithersburg, MD, USA, 2001.
- [45] J. E. Gentle, *Random Number Generation and Monte Carlo Methods*, vol. 381. Berlin, Germany: Springer, 2003.
- [46] M. Kennedy, R. Rovatti, and G. Setti, *Chaotic Electronics in Telecommunications*. Boca Raton, FL, USA: CRC Press, 2000.
- [47] P. Panagiotou, N. Sklavos, E. Darra, and I. D. Zaharakis, "Cryptographic system for data applications, in the context of Internet of Things," *Microprocessors Microsystems*, vol. 72, 2020, Art. no. 102921.
- [48] M. Herrero-Collantes and J. C. Garcia-Escartin, "Quantum random number generators," *Rev. Modern Phys.*, vol. 89, no. 1, 2017, Art. no. 015004.
- [49] Y. Zhang, H. Bao, Z. Hua, and H. Huang, "Two-dimensional exponential chaotic system with hardware implementation," *IEEE Trans. Ind. Electron.*, vol. 70, no. 9, pp. 9346–9356, Sep. 2023.
- [50] S. Sutar, A. Raha, D. Kulkarni, R. Shorey, J. Tew, and V. Raghunathan, "D-PUF: An intrinsically reconfigurable dram PUF for device authentication and random number generation," *ACM Trans. Embedded Comput. Syst.*, vol. 17, no. 1, pp. 1–31, 2017.
- [51] G. Kaddoum, "Wireless chaos-based communication systems: A comprehensive survey," *IEEE Access*, vol. 4, pp. 2621–2648, 2016.
- [52] F. Pub, *Data Encryption Standard (DES)*. Gaithersburg, MD, USA: FIPS PUB, 1999, pp. 46–3.
- [53] P. FIPS, *197: Advanced Encryption Standard (AES)*. Gaithersburg, MD, USA: National Institute of Standards and Technology, vol. 26, 2001.
- [54] Z. Hua, Y. Zhou, C.-M. Pun, and C. P. Chen, "2D sine logistic modulation map for image encryption," *Inf. Sci.*, vol. 297, pp. 80–94, 2015.
- [55] Z. Hua, Y. Zhou, and H. Huang, "Cosine-transform-based chaotic system for image encryption," *Inf. Sci.*, vol. 480, pp. 403–419, 2019.
- [56] Q. Lai and H. Zhang, "A new image encryption method based on memristive hyperchaos," *Opt. Laser Technol.*, vol. 166, 2023, Art. no. 109626.
- [57] S. Sinha and W. L. Ditto, "Dynamics based computation," *Phys. Rev. Lett.*, vol. 81, no. 10, 1998, Art. no. 2156.
- [58] W. L. Ditto, A. Miliotis, K. Murali, S. Sinha, and M. L. Spano, "Chagates: Morphing logic gates that exploit dynamical patterns," *Chaos: An Interdiscipl. J. Nonlinear Sci.*, vol. 20, no. 3, 2010, Art. no. 037107.
- [59] B. Kia, J. F. Lindner, and W. L. Ditto, "A simple nonlinear circuit contains an infinite number of functions," *IEEE Trans. Circuits Syst. II: Exp. Briefs*, vol. 63, no. 10, pp. 944–948, Oct. 2016.
- [60] B. Kia, J. F. Lindner, and W. L. Ditto, "Nonlinear dynamics as an engine of computation," *Philos. Trans. Roy. Soc. A: Math., Phys. Eng. Sci.*, vol. 375, no. 2088, 2017, Art. no. 20160222.
- [61] M. Sadiq, P. S. Paul, and M. S. Hasan, "Compact analog chaotic map designs using SOI four-gate transistors," *IEEE Access*, vol. 11, pp. 64782–64795, 2023.
- [62] P. C. Kocher, "Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems," in *Proc. Adv. 16th Annu. Int. Cryptol. Conf.*, 1996, pp. 104–113.
- [63] S. Mangard, E. Oswald, and T. Popp, *Power Analysis Attacks: Revealing the Secrets of Smart Cards*, vol. 31. Berlin, Germany: Springer, 2008.
- [64] R. Spreitzer, V. Moonsamy, T. Korak, and S. Mangard, "Systematic classification of side-channel attacks: A case study for mobile devices," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 1, pp. 465–488, Jan./Feb. 2017.
- [65] M. Randolph and W. Diehl, "Power side-channel attack analysis: A review of 20 years of study for the layman," *Cryptogr.*, vol. 4, no. 2, p. 15, 2020. [Online]. Available: <https://doi.org/10.3390/cryptography4020015>
- [66] D. Chen et al., "MAGLeak: A learning-based side-channel attack for password recognition with multiple sensors in IIoT environment," *IEEE Trans. Ind. Inform.*, vol. 18, no. 1, pp. 467–476, Jan. 2022.
- [67] G. S. Rose, "A chaos-based arithmetic logic unit and implications for obfuscation," in *Proc. IEEE Comput. Soc. Annu. Symp. VLSI*, 2014, pp. 54–58.
- [68] M. B. Majumder, M. S. Hasan, M. Uddin, and G. S. Rose, "Chaos computing for mitigating side channel attack," in *Proc. IEEE Int. Symp. Hardware Oriented Secur. Trust*, 2018, pp. 143–146.
- [69] W. Xiaofu et al., "A general efficient method for chaotic signal estimation," *IEEE Trans. Signal Process.*, vol. 47, no. 5, pp. 1424–1428, May 1999.
- [70] C. Fan, Q. Ding, and C. K. Tse, "Counteracting the dynamical degradation of digital chaos by applying stochastic jump of chaotic orbits," *Int. J. Bifurcation Chaos*, vol. 29, no. 8, 2019, Art. no. 1930023.
- [71] C. Li, B. Feng, S. Li, J. Kurths, and G. Chen, "Dynamic analysis of digital chaotic maps via state-mapping networks," *IEEE Trans. Circuits Syst. I: Regular Papers*, vol. 66, no. 6, pp. 2322–2335, Jun. 2019.
- [72] D. Arroyo, R. Rhouma, G. Alvarez, S. Li, and V. Fernandez, "On the security of a new image encryption scheme based on chaotic map lattices," *Chaos: Interdiscipl. J. Nonlinear Sci.*, vol. 18, no. 3, 2008, Art. no. 033112.
- [73] O. Rossler, "An equation for hyperchaos," *Phys. Lett. A*, vol. 71, no. 2/3, pp. 155–157, 1979.
- [74] T. Matsumoto, L. Chua, and K. Kobayashi, "Hyper chaos: Laboratory experiment and numerical confirmation," *IEEE Trans. Circuits Syst.*, vol. 33, no. 11, pp. 1143–1147, Nov. 1986.
- [75] Y. Deng and Y. Li, "A 2D hyperchaotic discrete memristive map and application in reservoir computing," *IEEE Trans. Circuits Syst. II: Exp. Briefs*, vol. 69, no. 3, pp. 1817–1821, Mar. 2022.



**MD SAKIB HASAN** (Member, IEEE) received the B.Sc. degree in electrical and electronic engineering from the Bangladesh University of Engineering and Technology, Dhaka, Bangladesh, in 2009 and the Ph.D. degree in electrical engineering from the University of Tennessee, Knoxville, TN, USA, in 2017.



He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, University of Mississippi, University, MS, USA. His research interests include security solutions using nonlinear dynamics, neuromorphic computing, semiconductor device modeling, and VLSI design.

**ANURAG DHUNGEL** is currently working toward the M.S. degree in engineering science with the Department of Electrical and Computer Engineering, University of Mississippi, University, MS, USA.

His research focuses on nonlinear dynamics, chaos-based hardware security applications, digital design, and deep learning.



**PARTHA SARATHI PAUL** received the M.Sc. degree in electrical and electronic engineering from the Bangladesh University of Engineering and Technology, Dhaka, Bangladesh, in 2014 and the M.Sc. degree in electrical and computer engineering from Oregon State University, Corvallis, OR, USA, in 2017. He is currently working toward the Ph.D. degree in engineering science with the Department of Electrical and Computer Engineering, University of Mississippi, University, MS, USA.

His research area focuses on the mixed-signal circuit design for chaos-based hardware security applications like random number generators and reconfigurable logic.





**MAISHA SADIA** (Graduate Student Member, IEEE) received the B.Sc. and M.Sc. degrees in electrical engineering in 2017 and 2019, respectively, from the University of Mississippi, University, MS, USA, where she is currently working toward the Ph.D. degree with the Department of Electrical and Computer Engineering.

Her research interests include vehicular ad hoc networks and chaos-based hardware security applications.



**MD RAZUAN HOSSAIN** (Graduate Student Member, IEEE) received the B.Sc. (Eng.) degree in electrical, electronics and communication engineering from the Department of Electrical Electronic and Communication Engineering, Military Institute of Science and Technology, Dhaka, Bangladesh, in 2015 and the M.Sc. degree from the Department of Electrical and Computer Engineering, North Dakota State University, Fargo, ND, USA, in 2019. He is currently working toward the Ph.D. degree in engineering science with the Department of Electrical and Computer Engineering, University of Mississippi, University, MS, USA.

His current research focuses on neuromorphic computing and nonlinear dynamics.