FEDERAL UNIVERSITY OF MINAS GERAIS
School of Engineering
Graduate Program in Electrical Engineering

Samuel Souza da Silva

A comparison analysis of using different numerical representations in digital chaotic maps

Belo Horizonte

2023

Samuel Souza da Silva

A comparison analysis of using different numerical representations in digital chaotic maps

Dissertation presented to the Graduate Program in Electrical Engineering of the Federal University of Minas Gerais in partial fulfillment of the requirements for the degree of Master in Electrical Engineering.

Supervisor: Prof. Dr. Janier Arias García
Co-supervisor: Prof. Dr. Erivelton Geraldo Nepomuceno

Belo Horizonte
2023

Silva, Samuel Souza da.
S586c A comparison analysis of using different numerical representations in digital chaotic maps [recurso eletrônico] / Samuel Souza da Silva. - 2023.
1 recurso online (64 f. : il., color.) : pdf.

Orientador: Janier Arias García.
Coorientador: Erivelton Geraldo Nepomuceno.

Dissertação (mestrado) - Universidade Federal de Minas Gerais, Escola de Engenharia.

Bibliografia: f. 59-64.
Exigências do sistema: Adobe Acrobat Reader.

1. Engenharia elétrica - Teses. 2. Teoria do caos - Teses. 3. Sistemas caóticos - Teses. 4. Análise numérica - Teses. 5. Cálculos numéricos - Teses. 6 . Estatística - Teses. 7. Números - Representação - Teses.
I. Arias García, Janier. II. Nepomuceno, Erivelton Geraldo.
III. Universidade Federal de Minas Gerais. Escola de Engenharia.
IV. Título.

CDU: 621.3(043)

Ficha catalográfica elaborada pela bibliotecária Ângela Cristina Silva CRB/6 2361
Biblioteca Prof. Mário Werneck, Escola de Engenharia da UFMG

UNIVERSIDADE FEDERAL DE MINAS GERAIS
ESCOLA DE ENGENHARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

FOLHA DE APROVAÇÃO

"A COMPARISON ANALYSIS OF USING DIFFERENT NUMERICAL REPRESENTATIONS IN DIGITAL CHAOTIC MAPS"

SAMUEL SOUZA DA SILVA

Dissertação de Mestrado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do grau de Mestre em Engenharia Elétrica. Aprovada em 04 de julho de 2023. Por:

Prof. Dr. Janier Arias García
DELT (UFMG) - Orientador

Prof. Dr. Erivelton Geraldo Nepomuceno
Centre for Ocean Energy Research and Department of Electronic Engineering (Maynooth University)

Prof. Dr. Jones Yudi Mori Alves da Silva
ENM (UNB)

Prof. Dr. Leonardo Amaral Mozelli
DELT (UFMG)

Documento assinado eletronicamente por **Janier Arias Garcia**, **Professor do Magistério Superior**, em 04/07/2023, às 12:38, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Jones Yudi Mori Alves da Silva**, **Usuário Externo**, em 05/07/2023, às 10:03, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Leonardo Amaral Mozelli**, **Professor do Magistério Superior**, em 05/07/2023, às 10:17, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Erivelton Geraldo Nepomuceno**, **Usuário Externo**, em 10/07/2023, às 08:51, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

A autenticidade deste documento pode ser conferida no site https://sei.ufmg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **2426838** e o código CRC **79250FC7**.

I dedicate this work to my parents, Maria José Oliveira Xavier and Aloizio Souza da Silva, for all the support.

Acknowledgements

I would like to thank my supervisor, Janier Arias García, for his guidance throughout this work. I would like to thank my parents, Maria José and Aloizio Souza, for their support, on which, without them, I would not be able to complete this project. Many thanks for the contributions of Lucas Nardo, Matheus Cardoso, Erivelton Nepomuceno, and Michael Hübner to this work. I also acknowledge the Federal University of Minas Gerais, for all the knowledge I acquired during my academic studies.

"Creativity is the ability to introduce order into the randomness of nature."
—Eric Hoffer

Resumo

Mapas caóticos digitais vêm sendo bastante utilizados no meio científico, a exemplo de aplicações envolvendo simulação de Monte Carlo e geração de números pseudoaleatórios. Contudo, sabe-se que o uso de uma representação finita faz com que mapas caóticos digitais apresentem necessariamente órbitas periódicas que degeneram suas propriedades caóticas. Além disso, nota-se, também, que há na literatura um número significativo de trabalhos que propõem métodos para mitigar o processo de degradação do caos. Por exemplo, pode-se utilizar um registrador de deslocamento com realimentação linear como uma fonte de perturbação para esses sistemas. No entanto, poucos estudos visam a analisar o impacto do uso de diferentes representações numéricas no processo de degradação do caos em mapas caóticos digitais. Dado isso, este trabalho propõe uma arquitetura de *hardware* utilizando o formato HUB (*Half Unit Bias*), em ponto fixo, que acopla o mapa tenda com o mapa de Bernoulli e que apresenta propriedades pseudoaletórias, sendo que, para tal arquitetura, foi realizada uma análise comparando o uso do formato HUB com relação a representação padrão em ponto fixo. Além disso, este trabalho também propõe uma implementação, em *software*, de um novo sistema baseado no mapa seno que, usando a representação posit, utiliza inferência fuzzy de Sugeno para aproximar a função senoidal. Com isso, para tal sistema em *software*, também é proposta uma análise que compara a influência do uso da representação posit em relação a representação padrão IEEE 754 em ponto flutuante. Em ambos os sistemas propostos, a análise comparativa feita acerca das representações numéricas adotadas foi realizada tendo-se por base, sobretudo, propriedades caóticas e pseudoaleatórias dos sistemas, por meio de métricas como, por exemplo, o expoente de Lyapunov e a suíte de testes do NIST SP 800-22, que consiste em um conjunto de testes estatísticos para geradores de números pseudoaleatórios (PRNGs). Resultados mostram que, para a arquitetura de *hardware* proposta, a representação em ponto fixo que usa o formato HUB consome menos recursos em *hardware* e que ela pôde fazer com que o sistema não entrasse em pontos fixos de anulação de caos. Para o sistema baseado no mapa seno, foi constatado que, pelo fato da representação posit possuir uma maior entropia por bit, tal representação foi capaz de ter melhores resultados, em termos de pseudoaleatoriedade, quando comparada com a implementação usando o formato IEEE 754 em ponto flutuante. Ademais, ambos os sistemas propostos passaram em todos os testes do NIST SP 800-22, o que evidencia, portanto, que eles podem ser usados como um PRNG.

Palavras-chave: Representações numéricas. Mapa tenda. Mapa de Bernoulli. Mapa seno. Ponto fixo. *Half unit bias*. IEEE 754 em ponto flutuante. Posit. PRNG.

Abstract

Digital chaotic maps have been used in a variety of applications in the scientific field, such as in the Monte Carlo simulation and in the generation of pseudorandom numbers. However, it is known that the usage of finite precision makes digital chaotic maps necessarily present periodic orbits that degenerate their chaotic properties. Furthermore, it can be noted that there has been a significant number of works that proposes different methods to mitigate chaos degradation in the literature. For instance, one can utilize a linear feedback shift register as a source of perturbation in such systems. Nevertheless, a few papers intend to analyze the impact of using different numeric representations in the process of chaos degradation in digital chaotic maps. Thus, this work aims to provide a hardware architecture using the HUB (Half Unit Bias) format, in fixed-point, that bi-couples the tent map with the Bernoulli map, and presents pseudorandom properties, on which, for such architecture, it was performed a comparative analysis that evaluates the use of the HUB format in contrast with the standard fixed-point representation. In addition, this work also presents an implementation, in software, of a novel system based on the sine map using the posit representation, on which it was utilized the Sugeno fuzzy inference to approximate the sine function. Herewith, for such a system in software, it was also proposed an analysis that compares the use of the posit representation in relation to the standard IEEE 754 floating-point representation. In both proposed systems, the comparative analysis of the numerical representations adopted was made based mainly on chaotic and pseudorandom properties of the systems, through metrics, such as the Lyapunov Exponent and the NIST SP 800-22 test suite, which consists of a set of statistical tests for pseudorandom number generators (PRNGs). Results show that, for the proposed hardware architecture, the fixed-point representation using the HUB format uses fewer hardware resources and it could also avoid the system from falling into chaos annulling conditions. For the system based on the sine map, it was observed that, because the posit representation has a higher entropy per bit, such representation was able to have better results, in terms of pseudorandomness, when compared with the implementation using the IEEE 754 floating-point format. Besides, both proposed systems have passed all the tests of the NIST SP 800-22 test suite, evidencing that they can be used as a PRNG.

Keywords: Numerical representations. Tent map. Bernoulli map. Sine map. Fixed-point. Half unit bias. IEEE 754 floating-point standard. Posit. PRNG.

List of Figures

List of Tables

List of abbreviations and acronyms

2D-CLSM	2D-Cosine-Logistic-Sine map
BT	Bit Transformation
DSP	Digital Signal Processor
ERN	Exactly Represented Number
FF	Flip-flop
FPGA	Field Programmable Gate Array
HUB	Half Unit Bias
IEEE	Institute of Electrical and Electronics Engineers
IO	Input/Output
LFSR	Linear Feedback Shift Register
LSB	Least Significant Bit
LUT	Look-up Table
MBT	Modified Bit Transformation
MFCD	Method for mitigating chaos degradation
MSB	Most Significant Bit
MUX	Multiplexer
NaN	Not a Number
PPGEE	Programa de Pós-Graduação em Engenharia Elétrica
PRBG	Pseudorandom Bit Generator
PRNG	Pseudorandom Number Generator
PWLCM	Piecewise linear chaotic map

TCAS-I	Transactions on Circuits and Systems I
TIM	Transactions on Instrumentation and Measurement
UFMG	Universidade Federal de Minas Gerais
ULP	Unit in the Last Place
Unum	Universal Number
VHDL	VHSIC Hardware Description Language
VHSIC	Very High Speed Integrated Circuit
XNOR	Exclusive NOR
XOR	Exclusive OR

Contents

Chapter 1

Introduction

Chaotic systems present attractive properties to scientific and industrial communities, such as complex behavior, high sensitive dependence on initial conditions, and unpredictability Corless [1994]. For instance, Fortuna et al. [2003] proved that the peculiarity of chaos enhanced the performance of the ultrasonic sensors' instruments that use unique sequences to identify their echo. In addition, Garcia-Bosque et al. [2019] proposed a new chaos-based pseudorandom number generator that can be applied in Monte Carlo simulations while keeping low hardware resource consumption. In Perez-Resa et al. [2019], an encryption scheme based on a chaotic algorithm has been successfully used in the Ethernet 1000Base-X standard. Furthermore, Pisarchik and Zanin [2008] proposed a cryptosystem for color images based on coupling chaotic maps, presenting good confusion and diffusion properties that guarantee high security.

Although there are significant applications involving chaotic systems in the literature, preserving chaotic properties in digital chaotic maps can be a non-trivial task when using finite precision Corless [1994]. Whenever using a word length of n bits, the iterations of the digital chaotic map can only take a maximum of 2^n distinct values for any given number representation, and, because the value of a given iteration determines the value of the next one, its iterations will necessarily enter in periodic cycles Garcia-Bosque et al. [2019], Liu et al. [2018]. Additionally, this process, called dynamical degradation of chaos, can lead to digital chaotic systems with periods that are significantly lower than the maximum number of distinct values available for the number representation adopted Garcia-Bosque et al. [2019], Liu et al. [2018]. Herewith, the dynamical degradation of chaos can significantly degenerate chaotic properties, resulting in systems with poor performance, such as a slow cryptography system that lacks security Baptista [1998].

Nonetheless, if the period of the digital chaotic system is sufficiently large, it can still be used in a variety of applications Cardoso et al. [2021], Pisarchik and Zanin [2008]. Consequently, an intuitive method to mitigate the process of chaos degradation is using higher bit resolutions, although it might result in the use of considerable hardware resources.

Thus, to solve this problem, researchers have been studying different methods to reduce the effects of dynamical degradation. Hereby, the methods for mitigating the effects of chaos degradation can be divided into the following main categories: (1) using higher precision, (2) cascading chaotic maps, (3) perturbing the output of the chaotic system, (4) switching multiple chaotic maps, (5) analog-digital mixed control method, and (6) using a control feedback loop on the system Deng et al. [2015], Liu et al. [2018]. For example, as a common approach, one can use a Linear Feedback Shift Register, a series of combinations of flip-flops and XOR or XNOR logic gates Mishra et al. [2016], as a source of perturbation Cardoso et al. [2021].

In addition, given the advantages and capabilities of reconfigurable hardware, recent papers have presented new hardware architectures for chaotic systems. Nevertheless, optimizing performance and hardware resources can be a challenge when designing such systems, and, therefore, this subject is of interest to recent studies. For instance, Cardoso et al. [2021] proposed a hardware architecture for the exponential chaotic map that uses the Maclaurin series along with the Horner's method. In this way, it was shown that the exponential chaotic map could be implemented in a Field Programmable Gate Array (FPGA) with only one block of addition and one of multiplication to perform mathematical arithmetic operations.

Considering fixed-point numbers, a significant amount of papers have been adopting this representation in digital chaotic systems, such as in Garcia-Bosque et al. [2019], Teh et al. [2018], Cardoso et al. [2021], Hobincu and Datcu [2018]. Specifically, in Teh et al. [2018], a chaos-based hash function using fixed-point arithmetic was implemented, showing that the proposed system has low computational complexity. However, additional logical circuits would be necessary to perform rounding to the nearest number using such a format, which could considerably increase the hardware resources. Besides, most of the applications involving digital chaotic maps in the literature use the fixed-point representation because of its simplicity and low hardware consumption.

The use of the IEEE 754 floating-point representation is also presented in recent papers, as in Cardoso et al. [2021], François et al. [2014], Aboulseoud and Ismail [2019], Hassan and Ismail [2018]. For instance, Aboulseoud and Ismail [2019] introduced a hardware implementation of the generalized fractional-order logistic map using the IEEE 754 floating-point format to achieve high accuracy and precision. This numerical representation significantly increases the dynamic range. However, it is noticeable that algebraic operations are more complex to be accomplished in floating-point arithmetic when compared with the fixed-point format.

Furthermore, new non-conventional formats to represent numbers have been studied in the literature, such as the Half-Unit-Bias (HUB) format Hormigo and Villalba [2014]. In such a format, numbers have implicitly one bit set with the value "1" on the least

significant bit, and, with this, some operations are simpler to be executed. As proof, one can perform the rounding to the nearest approximation with only the cost of truncation, which decreases substantially the hardware resources necessary to perform this operation Hormigo and Villalba [2014].

Moreover, another recent number representation, called posit, has been proposed by Gustafson and Yonemoto [2017] that consists of a type of universal number (unum). Herewith, unums consist of a family of number formats designed to be an alternative to the IEEE 754 floating-point standard. Thus, posit numbers, which is equivalent to unums type III, exhibit a higher dynamic range compared with the IEEE 754 standard counterpart and, at the same time, it presents a tapered accuracy, in the sense that numbers on the interval [-1, 1] have higher accuracy than numbers that have an extremely high magnitude. However, most of the personal computers currently do not natively support this representation.

In such a way, it can be inferred that the number representation being used in digital chaotic maps not only can influence their required resources, but it can also directly influence their iterations. Nevertheless, a few works in the literature aims to analyze the influence of the usage of different numeric representations in the process of chaos degradation on digital chaotic maps. Therefore, this work compares the influence of using the HUB format in fixed-point against the standard fixed-point representation in a system that utilizes the tent map in conjunction with the Bernoulli map. Furthermore, since the method applied in this system could not be performed for the posit representation, a second system, based on the sine map, is presented that compares the usage of the posit representation against the IEEE 754 floating-point representation. In both systems, the comparison analysis was focusing on metrics involving chaotic and pseudorandom properties, such as the Lyapunov Exponent and pseudorandom statical tests.

1.1 General Objective

Analyze the influence of using different numeric representations in digital chaotic systems, focusing on chaotic and pseudorandom properties. To perform it, a novel hardware architecture that bi-couples the tent map with the Bernoulli map was proposed, and it was compared the usage of the HUB format in fixed-point against the usage of the standard format in fixed-point. Additionally, this work presents a software-based implementation of a system based on the Sine Sugeno Fuzzy approximation map that uses the posit representation and compares the performance of this system to that of the IEEE 754 floating-point standard representation.

1.2 Specific objectives

- Highlight the main advantages and disadvantages of each non-conventional representation applied.
- Propose a mechanism to reduce the chaos degradation in both systems, keeping key design parameters, such as performance.
- Perform statistical test suites which demonstrate the pseudorandomness of the proposed systems.
- Demonstrate that the systems can be used as a PRNG (pseudorandom number generator).
- Minimize the resources being used.
- Compare the results with state-of-the-art works.

1.3 Publications

The following publication has been accomplished during the development of this dissertation:

1. S. S. d. Silva, M. Cardoso, L. Nardo, E. Nepomuceno, M. Hübner and J. Arias-Garcia, "A New Chaos-Based PRNG Hardware Architecture Using the HUB Fixed-Point Format," in IEEE Transactions on Instrumentation and Measurement, vol. 72, pp. 1-8, 2023, Art no. 2001208, doi: 10.1109/TIM.2023.3235457.
2. S. S. d. Silva, L. Nardo, E. Nepomuceno, J. Yudi, and J. Arias-Garcia, "A novel Chaos-based PRNG using fuzzy inference approximation and POSIT numerical representation," submitted into a high impact journal, and it is currently under review.

1.4 Textual organization of the dissertation

The rest of this document is divided as follows: section 2 presents preliminary concepts and state-of-the-art to have a better understanding of this work. Section 3 contains details involving the development of the systems. Section 4 has the results and a comparative analysis of the systems regarding different numeric representations. Finally, section 5 shows the conclusions concerning this work.

Chapter 2

Preliminary Concepts and State-of-the-Art

2.1 An introduction to chaotic systems

2.1.1 Definition of chaotic system

It is known that non-linear systems can converge to equilibrium, oscillation, or chaos range Robinson [2009]. Nevertheless, chaos theory has emerged from recognizing that physical systems can appear unpredictable even though they follow deterministic laws Robinson [2009]. Herewith, a chaotic system is a non-linear system that presents high sensitive dependence on initial conditions in the sense that small modifications on the input of the system result in vastly different outputs. Therefore, chaotic systems present a set of characteristics such as complex behavior, and unpredictability Muthuswamy and Banerjee [2015], Rüdisüli et al. [2013].

One classical example of a chaotic system is the logistic map Phatak and Rao [1995], which is given by the equation 2.1:

$$x_{n+1} = \lambda x_n(1 - x_n) \quad \text{with} \quad n \in \mathbb{Z}_0^+ \tag{2.1}$$

Where λ is a control parameter on which $0 < \lambda < 4$ and the initial value x_0 belongs to the interval $[0, 1]$. In addition, depending on the value of λ, the system can assume different states Phatak and Rao [1995], which are:

- For $0 < \lambda < 1$: $x = 0$ is a stable fixed point, in the sense that for any value of x_0 inside its domain interval, there will be an integer number $i \geq 0$ on which, $x_i = 0$.
- For $1 \leq \lambda \leq 3$: $x = (\lambda - 1)/\lambda$ is a stable fixed point.

- For $3 < \lambda < 4$: the logistic map will present period doubling events, and, for specific values of λ on this range, it exhibits chaotic behavior, presenting, therefore, high sensitivity to initial conditions.

The behavior described can be visualized through the bifurcation diagram of the logistic map. The plot of the values taking into account the iterations of a chaotic map as a function of its control parameter, after the transient part, is called a bifurcation diagram. Hence, the bifurcation diagram of the logistic map is shown in Figure 1, reaffirming the aforesaid behavior:

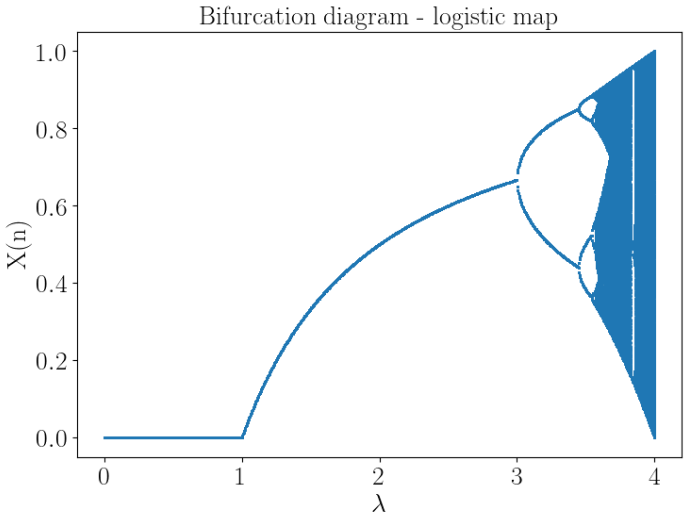

Figure 1 – Bifurcation diagram of the logistic map.

Mathematically, a system is considered chaotic if it presents at least one positive Lyapunov Exponent for a given value of the control parameter. Herewith, the Lyapunov Exponent can be interpreted as the rate of divergence of nearby trajectories in dynamical systems, and, for discrete one-dimensional maps, it can be calculated following the equation 2.2:

$$\lambda_a(x_0) := \lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} \ln \left| \frac{dF}{dx}(x_n) \right| \tag{2.2}$$

Where F corresponds to the mathematical equation of the discrete one-dimensional map.

For instance, it can be demonstrated that the logistic map assumes positive Lyapunov Exponent values when the control parameter is between 3.569946 and 4 Arif et al. [2022]. Thus, for this specific interval, the logistic map is considered chaotic.

2.1.2 A brief history of chaos

The first significant work on chaotic systems was done by Henri Poincaré, a professor at the University of Paris in the 19th century, during his studies on celestial bodies. Poincaré's investigation of the three-body problem revealed chaotic behavior, as it became clear that it was impossible to precisely predict the movements of certain celestial bodies over a long period Muthuswamy and Banerjee [2015]. Subsequently, the ergodic hypothesis of Boltzmann became another influential work in the field. Koopman's efforts to prove the ergodic hypothesis demonstrated that a nonlinear problem with finite dimensions could be translated into a linear problem with infinite dimensions Cvitanović et al. [2016].

However, one of the first definitive observations of deterministic chaos didn't occur until 1927, when Van Der Pol and Van Der Mark proposed a nonlinear oscillator that produced irregular noise at specific frequencies Muthuswamy and Banerjee [2015], Ginoux and Letellier [2012]. In 1950, Edward Lorenz used a computer with only 16KB of internal memory to solve 12 differential equations of an atmospheric model Alligood et al. [1997]. Lorenz found the model too complex, so he simplified it using three differential equations, as described in 2.3.

$$\begin{cases} \dot{x} = \sigma(y - x) \\ \dot{y} = rx - y - xz \\ \dot{z} = xy - bz \end{cases} \tag{2.3}$$

Where x, y, and z are three quantities that depend on time.

The equations in 2.3 revealed the chaotic properties of the model, such as sensitivity to initial conditions, aperiodicity, bounded trajectories, and complex attractor Muthuswamy and Banerjee [2015]. For instance, the attractor of the Lorenz system can be obtained in simulation, as it was obtained, using the Python programming language, in Figure 2.

It is noticeable that chaos can also be found in discrete dynamical systems Devaney [1987]. Moreover, May [1976] popularized the logistic map[1], a simple discrete mathematical equation, in 1976, on which, on his work, he reviewed the dynamics of first-order difference equations highlighting its applications in the context of biological, economic, and social sciences. However, even though the logistic map exhibits a simple mathematical equation, it presents complex dynamics, from stable points to apparently random behavior. Nowadays, it is known that the logistic map can be used in various applications, for instance, in a secure cryptography system with high performance Pareek et al. [2006].

Furthermore, the history of chaotic systems is also intertwined with electronic

[1] Further described in section 2.1.3.

circuits. For example, researchers had been trying to demonstrate the realization of the Lorenz system through electronic circuits since the 1970s. However, it wasn't until 1983 that Leon O. Chua, a professor at the University of California, developed the first chaotic electronic circuit. Since then, numerous electronic circuits exhibiting chaotic behavior have been proposed, such as hysteresis-based chaos generators, and chaos from synchronized oscillators Muthuswamy and Banerjee [2015].

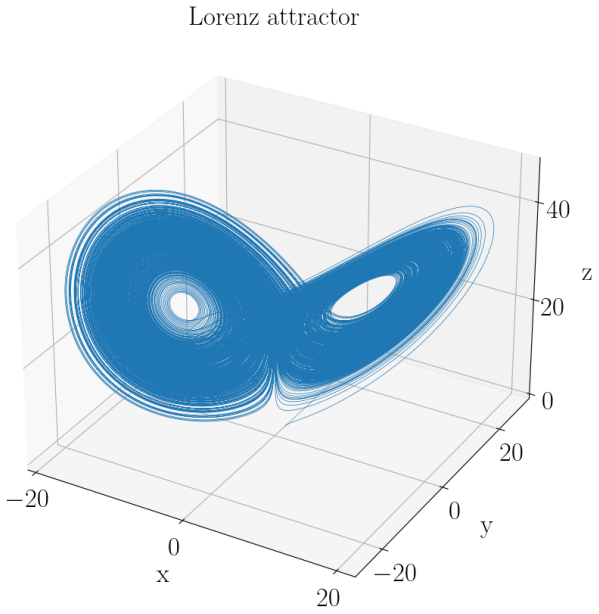

Figure 2 – Lorenz attractor considering $\sigma = 10$, $b = 8/3$, and $r = 28$.

2.1.3 Chaotic maps

Chaotic maps are discrete chaotic systems that have at least one positive Lyapunov Exponent and can be represented by the mathematical equation 2.4, as described in Aguirre [2023]:

$$\boldsymbol{x}_k = F(\boldsymbol{x}_{k-1}) \tag{2.4}$$

Where $\boldsymbol{x}_k \in \mathbb{R}^n$, and F is a mathematical function on which $F : \mathbb{R}^n \to \mathbb{R}^n$.

One example of a one-dimensional chaotic map is the logistic map, which is given by equation 2.1. Thus, from an initial value x_0, the value of the next iteration, x_1, can be obtained through the map recurrence equation, on which, specifically for the logistic map, $x_1 = \lambda \cdot x_0 \cdot (1 - x_0)$. In this way, we can successively compute the iterations x_2, x_3, x_4, ... , x_n, and this set of iterations is called "orbit" of the map. Nevertheless, when using finite precision in a computer, rounding is necessary to be executed. Consequently, the

value obtained after each iteration of a chaotic map represented in a computer will not be necessarily the same as the orbit of the same chaotic map from its mathematical definition. Thus, the set of iterations of a digital chaotic map, $\hat{x}_1$, $\hat{x}_2$, ... , $\hat{x}_n$, is called "pseudo-orbit". For instance, Figure 3 presents the pseudo-orbit of the logistic map obtained in software simulation, using the Python programming language, with $x_0 = 0.1$ and $\lambda = 1.95$, and with floating-point numbers in double precision, on which each interaction corresponds to a red circle on the figure. In this case, it is clear that, for this parameter of λ, the system does not present a chaotic behavior, and it has a stable fixed point at $(\lambda - 1)/\lambda \approx 0.487$. Nevertheless, results show that the logistic map can present a chaotic behavior for values of λ close to the number 4, with, therefore, a positive Lyapunov Exponent. To demonstrate it, Figure 4 presents the time series of the pseudo-orbits of the logistic map for $x_0 = 0.1$ and $\lambda = 3.97$. Notice that there is no apparent correlation between its interactions, even though the system is deterministic.

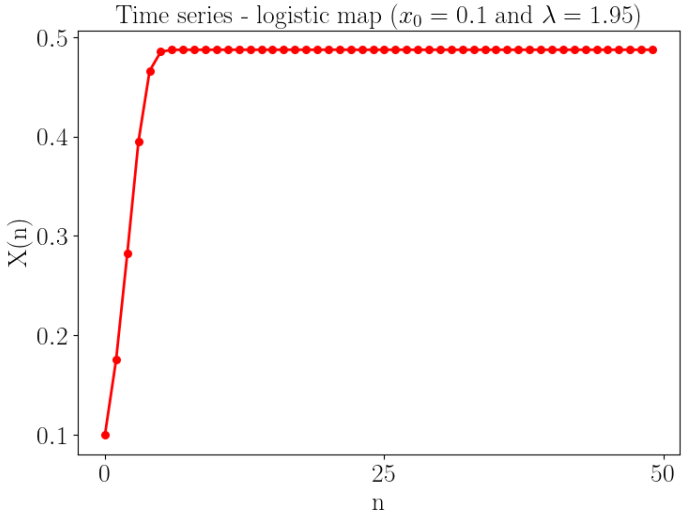

Figure 3 – Time series of the logistic map for $x_0 = 0.1$ and $\lambda = 1.95$.

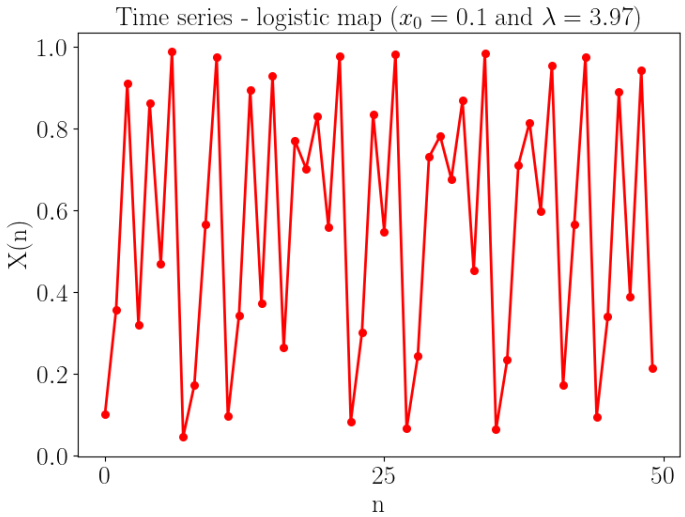

Figure 4 – Time series of the logistic map for $x_0 = 0.1$ and $\lambda = 3.97$.

Besides, in the rest of this subsection, it will be described all the chaotic maps utilized in this work. It is valid to mention that these chaotic maps were chosen based on their simplicity to be implemented in both hardware and software schemes.

2.1.3.1 Tent Map

The tent map is a discrete system that presents chaotic behavior depending on the value of its control parameter. It is represented by 2.5:

$$x_{n+1} = \begin{cases} \mu x_n, & \text{if} \quad x_n \in [0, 1/2) \\ \mu(1 - x_n), & \text{if} \quad x_n \in [1/2, 1), \end{cases} \tag{2.5}$$

where $x_n \in [0, 1)$ and the control parameter $\mu \in [0, 2]$. The chaotic behavior of this map is obtained for $1 < \mu \leq 2$ Yoshida et al. [1983]. Herewith, the time series of the tent map for $x_0 = 0.1$ and $\mu = 1.95$ is shown in Figure 5.

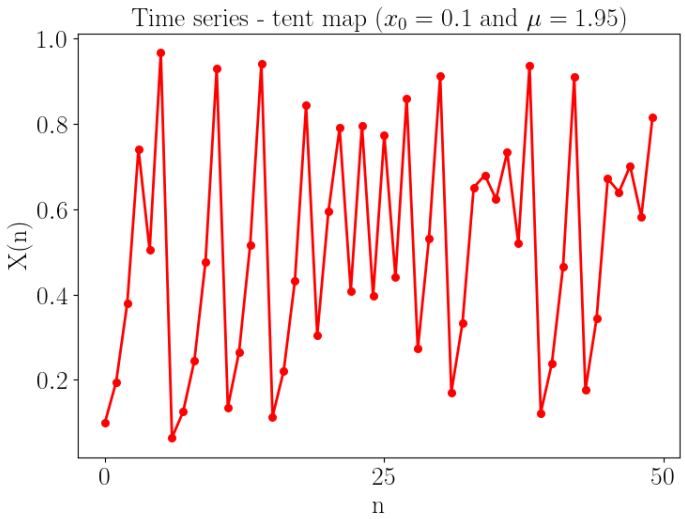

Figure 5 – Time series of the tent map for $x_0 = 0.1$ and $\mu = 1.95$.

Its bifurcation diagram is illustrated in Figure 6. It is clear that, for $\mu \leq 1$, the map does not present a chaotic behavior, having a stable fixed point. Nonetheless, as the μ parameter increases, it can be seen as a continuum on the bifurcation diagram, on which it presents a chaotic behavior.

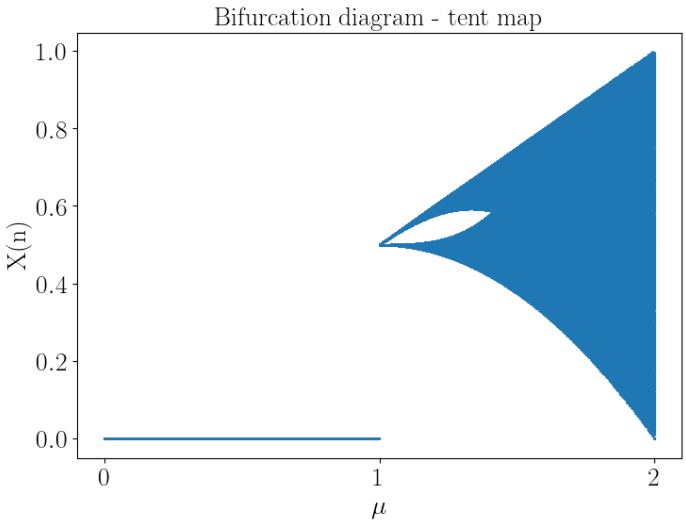

Figure 6 – Bifurcation diagram of the tent map.

Moreover, Figure 7 shows the state space of this map. The name "tent map" is due to the tent-like shape of its attractor.

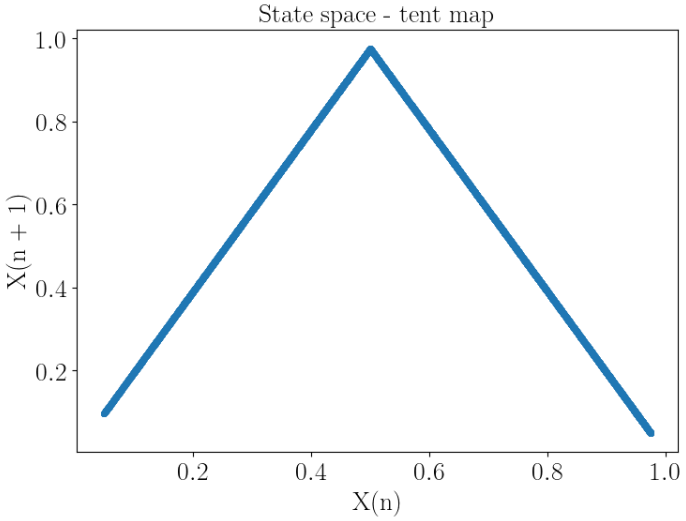

Figure 7 – State space of the tent map for $\mu = 1.95$.

2.1.3.2 Bernoulli Map

The one-dimensional Bernoulli map is a discrete system defined by 2.6:

$$x_{n+1} = \begin{cases} 2x_n, & \text{if} \quad x_n \in [0, 1/2) \\ 2x_n - 1, & \text{if} \quad x_n \in [1/2, 1), \end{cases} \tag{2.6}$$

It is noticeable that $x_n \in [0, 1)$ and the map does not have a control parameter. The Bernoulli map has a positive Lyapunov Exponent, equal to $\ln(2) = 0.693$; therefore, it presents chaotic behavior Driebe [1999]. Its time series is presented in Figure 8.

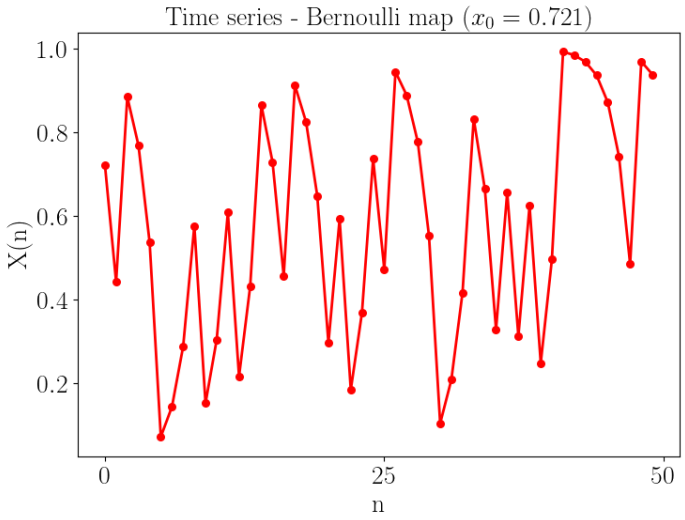

Figure 8 – Time series of the Bernoulli map for $x_0 = 0.721$.

Furthermore, since it does not have a control parameter, obtaining a bifurcation diagram for such a system is not possible. Additionally, the state space of this map is shown in Figure 9.

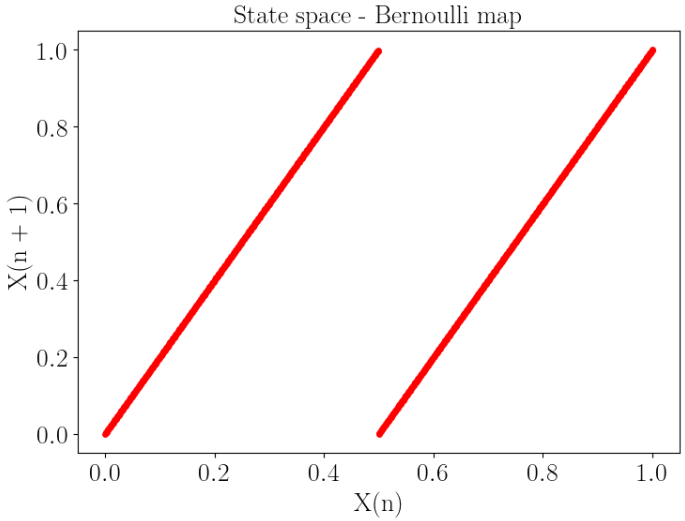

Figure 9 – State space of the Bernoulli map.

2.1.3.3 Sine map

The sine map is given by the equation 2.7:

$$x_{i+1} = \eta \, sin(\pi x_i) \tag{2.7}$$

Where η is the control parameter, and $\eta \in [0, 1]$. It exhibits chaotic behavior when $\eta \in [0.87, 1]$, and its domain is [0, 1], as described in Mansouri and Wang [2020]. The time series for $x_0 = 0.1$ and $\eta = 0.97$ is shown in Figure 10.

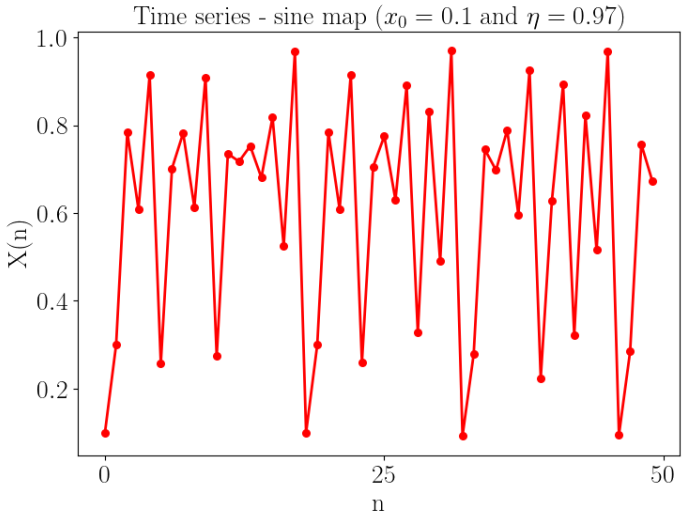

Figure 10 – Time series of the sine map for $x_0 = 0.1$ and $\eta = 0.97$.

Furthermore, the bifurcation diagram and its state space have a similar shape to the logistic map, as can be seen in Figure 11, and Figure 12, respectively.

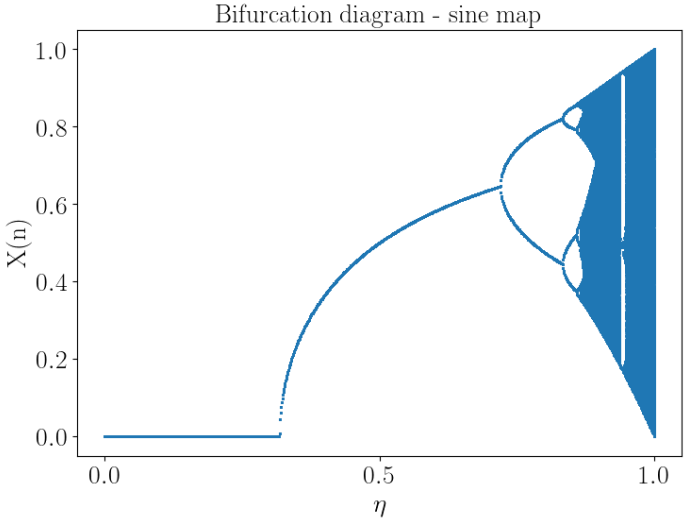

Figure 11 – Bifurcation diagram of the sine map.

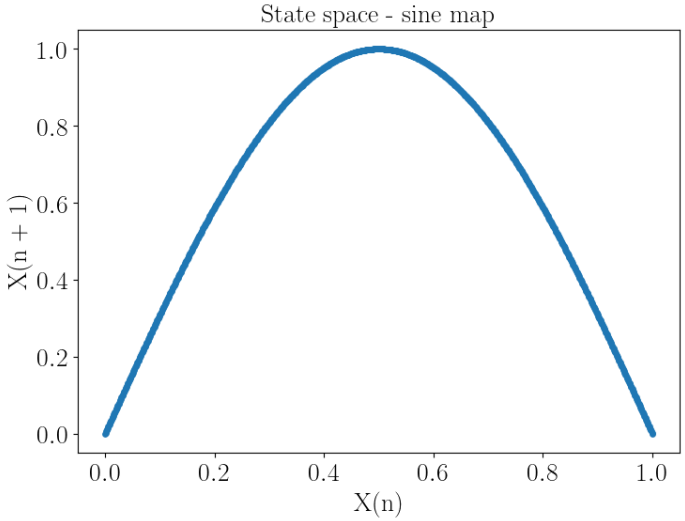

Figure 12 – State space of the sine map for $\eta = 1$.

2.2 Numerical representations

2.2.1 Fixed-point standard representation

A real number, A, represented in fixed-point by a set of "n" bits is divided into the parts shown in Figure 13, on which the most significant bit (MSB) is the signal bit, the m_2 least significant bits (LSBs) are the bits of the fractional part and the $m_1 = n - 1 - m_2$ remaining bits are used for the integer part. Herewith, given a set of "n" bits, $b_{n-1}b_{n-2}\ldots b_1b_0$, on which b_0 is the LSB, b_1 is the second LSB, and so on, the exactly represented number of this set of bits is equal the equation 2.8. Besides, it is valid to mention that most of the applications involving digital chaotic maps in the literature use this numeric representation because of its simplicity and because it demands a low computational cost to perform arithmetic operations.

$$\begin{aligned} A = (-1)^{b_{n-1}} \times (b_{n-2} \times 2^{n-m_2-2} + b_{n-3} \times 2^{n-m_2-3} + \ldots \\ + b_{m_2} \times 2^0 + b_{m_2-1} \times 2^{-1} + b_{m_2-2} \times 2^{-2} + \ldots + b_0 \times 2^{-m_2}) \end{aligned} \tag{2.8}$$

Figure 13 – Fixed-point representation.

For instance, when using a word length of 5 bits, with 1 bit for the sign part, 1 bit for the integer part, and the 3 remaining bits for the fractional part, consider the following bit vector:

$$01110_2 \tag{2.9}$$

For this specific example, we can infer that:

- The signal part is equal to 0, and, therefore, it is a positive number.
- The integer part is 1_2.
- The fractional part is equal to 110_2.

Thus, the bit vector described in 2.9 represents the number given by equation 2.10:

$$01110_2 = (-1)^0 \cdot (1 \cdot 2^1 + 1 \cdot 2^{-1} + 1 \cdot 2^{-2} + 0 \cdot 2^{-3}) = 1.75_{10} \tag{2.10}$$

2.2.2 IEEE 754 floating-point

The standard for floating-point arithmetic (IEEE 754) is a numerical representation largely employed in the industry nowadays. It supports two formats, which are: (1) single and (2) double precision Jiménez et al. [2014]. The single precision format uses a word length of 32 bits, in which the MSB is used for the sign bit, the 23 LSBs are utilized for the mantissa part, and the 8 bits remaining are used for the exponent part, as can be seen in Fig. 14-A. In addition, the double precision representation has a word length of 64 bits, in which the most significant bit is for the signal part, the 52 least significant bits are designated for the mantissa part, and the 11 remaining bits are used for the exponent part, as illustrated in Fig. 14-B.

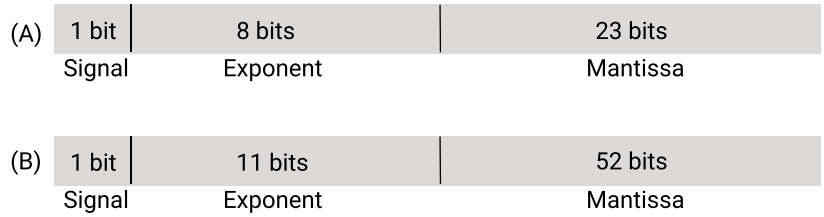

Figure 14 – IEEE 754 Floating-point representation: (A) single precision. (B) double precision.

In both precisions, the sign bit assumes the value "1" for negative numbers. Otherwise, the sign bit will assume the value "0". Furthermore, when using the single precision, the 8-bit exponent has a bias of 127, and the bits of the mantissa are normalized. Whereas for the double precision, the 11-bit exponent has a bias equal to 1023, and the mantissa is also normalized.

The floating-point number FP is represented by equation 2.11.

$$FP = (-1)^S \cdot (1.mantissa) \cdot 2^{exponent-bias} \tag{2.11}$$

For instance, consider the following bit vector in single precision:

$$01000010001011100000000000000000_2 \tag{2.12}$$

For this specific case, the following is inferred:

- The signal part, S, is equal to 0.
- Since it is using single precision, then $bias = 127_{10}$.
- The exponent is given by: $exponent = 10000100_2 = 132_{10}$.
- The mantissa part is $01011100000000000000000_2$ and it represents the fraction 0.359375_{10}.

Thus, the decimal number represented by the bit vector 2.12 is equal to equation 2.13:

$$01000010001011100000000000000000_2 = (-1)^0 \cdot (1.359375) \cdot 2^{132-127} = 43.5_{10} \quad (2.13)$$

However, it is worth noting that this standard has several special values that do not follow equation 2.11. They are:

- Zero: represented when both exponent and mantissa bits are equal to zero. Notice that one can represent positive zero and negative zero using this standard.
- Infinity: when the exponent bits are all set to "1", and the mantissa bits are all zero. In this case, we can also represent positive and negative infinity, depending on the sign bit.
- Not a number (NaN): occurs when the exponent bits are all set to "1" and the mantissa part is non-zero.

2.2.3 Half-Unit-Biased Format

Half-Unit-Biased (HUB) is a representation format where numbers necessarily have an implicit least significant bit constant and equal to one, and it is important to note that this least significant bit does not bring any extra storage cost Hormigo and Villalba [2014]. Herewith, considering the term "Exactly Represented Number" (ERN) as the number value that is represented after performing the rounding in a given operation, the ERN of a bit vector under the HUB format corresponds to the ERN of the same bit vector using the conventional format added a value of half unit in the last place (ulp) Hormigo and Villalba [2014]. As an example, when using one bit for the sign part, one bit for the integer part, and three explicit bits for the fractional part for a HUB number in fixed-point, consider the following bit vector:

$$00110_2 \quad (2.14)$$

For the example of 2.14, we conclude the following:

- The sign part is 0.
- The integer part is 0.
- The explicit bits of the fractional part are equal to 110_2, but it has implicitly one least significant bit equal to one.

Thus, the ERN of the bit vector 00110_2 is given by equation 2.15:

$$00110_2 = (-1)^0 \cdot (0 \cdot 2^0 + 1 \cdot 2^{-1} + 1 \cdot 2^{-2} + 0 \cdot 2^{-3} + 1 \cdot 2^{-4}) = 0.8125_{10} \tag{2.15}$$

One advantage of using the HUB format is that it performs operations, such as the 2's complement, in a simpler way. The 2's complement can be performed by bit-wise inversion. Then, as an example, the 2's complement of the HUB number 00110_2 is given by 11001_2. Furthermore, the ability to round to the nearest number can be done only by the cost of truncation Hormigo and Villalba [2014]. Thus, the HUB format can remarkably reduce the hardware resources needed to perform the round-to-the-nearest operation when compared with the conventional format. In contrast, in applications that need an effective rounding down, additional logic circuits would be necessary to be performed when using this format.

Fig. 15 illustrates, in (A), the process of truncating a number in a conventional fixed-point representation, using three bits for the fractional part. The non-ERNs are shown in the figure in black circles, whereas the ERNs are shown in light blue circles. In this figure, it is clear that rounding is necessary to be done to represent the non-ERN when using only three bits for the fractional part. Therefore, when using the standard representation in (A), the truncation operation is responsible for an effective rounding down. However, due to the implicit least significant bit when using HUB numbers, truncating a number in (B) leads to rounding to the nearest ERN. Additionally, it is important to mention that, in (B), the ERN uses three explicit bits for the fractional part, and their implicit bit is shown in red for convenience.

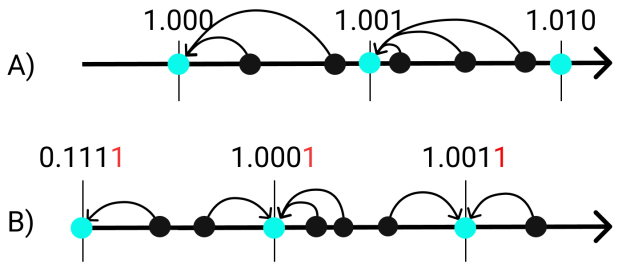

Figure 15 – Process of truncating a number using: (A) standard fixed-point representation for numbers using three bits for the fractional part. (B) the HUB fixed-point format for numbers using three explicit bits for the fractional part. The non-ERNs are represented in black circles, and the ERNs after truncation is represented in blue circles.

Furthermore, to work with the HUB fixed-point format, a general rule can be applied to compute arithmetic operations. Firstly, the bit vector used is extended with an additional least significant bit set to one. Then, one can apply the conventional arithmetic

operations considering the extended bit vector. Finally, after calculating the result of such an arithmetic operation, one can have the output HUB format simply by truncating it, obtaining the desired number of bits Hormigo and Villalba [2014].

2.2.4 Posit

Posit numbers are a data type that has been designed to replace the IEEE 754 standard for floating-point arithmetic Gustafson and Yonemoto [2017], Cook [2018]. The key differences between posit arithmetic and its counterpart are that posit arithmetic never overflows to infinity Gustafson and Yonemoto [2017], Cook [2018], and, at the same time, posit arithmetic provides a higher dynamic range with tapered precision, on which its values mass around 0. Therefore, numbers inside the interval $[-1, 1]$ have more accuracy and precision than extremely high numbers or extremely small numbers Murillo et al. [2020]. As disadvantages, it can be noted that most hardware implementations and personal computers do not natively support this numerical representation, limiting its use.

The posit representation is specified by two numbers, which are: the word length (n), and the maximum number of bits (es) of the exponent part. Thus, it is a common practice to designate a posit number representation using the notation: posit<n, es>. Furthermore, a posit number has the following parts: (1) sign, (2) regime, (3) exponent, and (4) fraction bits.

The sign bit behaves similarly to the IEEE 754 floating-point numbers. The sign bit "0" indicates positive numbers, and "1" indicates negative values. However, if the sign bit is "1", it is necessary to take the twos complement of the rest of the bits, before computing the regime, exponent, and fraction bits of the number.

Moreover, the regime is a sequence of identical bits right after the sign bit. The regime ends when either it achieves the maximum bit length (n) or when the sequence of identical bits ends by an opposite bit. For example, if the first bit after the sign part is "0", the regime part will be composed of a sequence of bits "0", until either the bit "1" is found or when it runs out of bits. It is valid to mention that the opposite bit that indicates the end of the regime sequence is not included in the regime for computing posit values.

The exponent bits are the ones after the regime part if there are any. Their maximum length will be equal to es. Furthermore, if there is a number of bits less than es after the regime part, then the rest of the bits after the regime will be assigned to the exponent part. Besides, the remaining bits, if any, after the sign, the regime, and the exponent bits, are set to the fraction part.

Herewith, the representation of a posit number P is given by 2.16:

$$P = (-1)^S u^k 2^e f \tag{2.16}$$

Here S corresponds to the sign bit; e is the exponent bits taken as an unsigned integer; the value u, known as "used", is represented by equation 2.17; k is given by equation 2.18, and f is the value represented by the fraction bits added a value of 1.

$$u = 2^{2^{es}} \tag{2.17}$$

$$k = \begin{cases} -m, & \text{if regime has m 0's} \\ m - 1, & \text{if regime has m 1's} \end{cases} \tag{2.18}$$

For example, consider the following bit vector using posit<32, 2>:

$$01101011110110100000000000000000_2 \tag{2.19}$$

For this case, the following is inferred:

- The signal part, S, is equal to 0.
- The useed is $u = 2^{2^{es}} = 2^{2^2} = 16_{10}$.
- The regime bits are 11_2, and, therefore, $k = 1_{10}$
- The exponent bits are 10_2, representing the integer value 2_{10}.
- The fraction part is $11110110100000000000000000_2$, and, consequently, $f = 1.962890625$.

Thus, the ERN of the bit vector of 2.19 is shown on equation 2.20:

$$01101011110110100000000000000000_2 = (-1)^0 \cdot 16^1 \cdot 2^2 \cdot 1.962890625 = 125.625_{10} \tag{2.20}$$

In addition, there are only two posit special values, which are:

- Zero: represented by all n bits set to zero.
- $\pm$ infinity: when only the first bit is "1", and the rest of the bits are "0". Note that there is no distinction between positive or negative infinity in this representation.

In addition, it is worth mentioning that, for 32 bits, a common practice in the literature is using posit<32, 2>.

2.3 Sugeno fuzzy inference

The Sugeno fuzzy inference Sugeno [1985], also referred to Takagi-Sugeno-Kang fuzzy inference, is a method for mapping from a given input to an output using fuzzy logic. Herewith, this procedure is typically composed by a set of "N" rules in the following pattern:

$$\text{If } x \text{ is } A_i \text{ and } y \text{ is } B_i, \text{ then } z_i = f(x, y). \tag{2.21}$$

Where x and y are inputs of the system; A_i and B_i are fuzzy sets; z_i is either a linear function of the input values or a constant for the rule i. In addition, to compute a single number from the set of rules, the output z_i is weighted by a value w_i, on which for an "and" rule, as in 2.21, w_i is given by equation 2.22:

$$w_i = t\text{-}norm(F_1(x), F_2(y)) \tag{2.22}$$

Where F_1 and F_2 are the membership functions of the fuzzy sets, and the *t-norm* is usually the minimum operator. Consequently, the final output of the system is presented in equation 2.23:

$$\text{Output} = \frac{\sum_{i=1}^{N} w_i z_i}{\sum_{i=1}^{N} w_i} \tag{2.23}$$

As it will be shown in Section III, this scheme can be used to approximate the sine function.

2.4 State-of-the-Art

To conduct a short introduction of the state-of-the-art works in the literature, it was employed the platform Web of Science, a database and selective citation index which is currently owned by Clarivate Analytics. The search applied the following fields in the platform: "chaotic maps", "chaos degradation", "chaotic systems", "applications". Herewith, a comprehensive number of papers were listed and analyzed.

Considerable research in the literature has applied chaotic maps as a component of pseudorandom number generators, such as those by Liu et al. [2018], Kopparthi et al. [2022], Cardoso et al. [2021]. For instance, Liu et al. [2018] proposed a coupled chaotic model that can be universally applied in such a way that the output of one chaotic map is used to compute the control parameter of another chaotic map. Herewith, the paper presents two coupled models: one that couples two logistic maps, and the other that couples the Chebyshev map with the Baker map. As an application, the paper demonstrated

that the coupled logistic maps can be used as a pseudorandom bit generator (PRBG), passing all statistical tests of the NIST test suite, a well-known set of statistical tests for pseudorandom sequences.

In addition, Kopparthi et al. [2022] proposed a hardware implementation of the Piecewise Linear Chaotic Map (PWLCM), on which XORed shift registers were used to mitigate chaos degradation. The designed PWCLM was implemented in the Xilinx Zynq 7000 FPGA, and the results showed that the proposed system can effectively be used as a PRNG. Moreover, the proposed design's effectiveness in producing random outcomes was validated through statistical and security analyses, such as phase space, key sensitivity, correlation, and information entropy.

Another common application involving chaotic maps relates to cryptography. As an example, in Sharma and Bhargava [2016], it is presented an image encryption scheme designed for gray-scaled images. The proposed scheme generates a chaotic sequence from a two-step iterated logistic map that was used to change pixel positions. Consequently, it was proven that the scheme provides high security and acceptable speed, being, therefore, evidence that the properties of digital chaotic maps, such as high sensitivity and long periods, are suitable for designing such a system.

Other examples include the image algorithm that cascades two chaotic maps provided by Zheng and Bao [2022]. They introduced an improved cascaded two-dimensional map, a 2D-Cosine-Logistic-Sine map (2D-CLSM), offering advantages in terms of complexity and sensitivity. Nevertheless, it is valid to mention that the efficiency of the proposed scheme was reduced due to the implementation of two rounds of encryption, one at the bit level and another at the pixel level.

An application involving instrumentation and measurements is presented in Garcia-Bosque et al. [2019]. A hardware implementation of a chaotic system based on the logistic map is applied, and, to mitigate chaos degradation, the control parameter of the map is dynamically changed. The proposed system was employed in a Virtex 7 FPGA with a total of 510 lookup tables (LUTs) and 120 registers, consuming, therefore, low resources. In addition, it was shown that the proposed system passed all NIST tests, being suitable to be used in the Monte Carlo simulation, an important application involving instrumentation and measurements.

Besides, Table 1 contains chaotic maps used by the state-of-the-art papers cited in this section:

Table 1 – Comparison of state-of-the-art works in terms of chaotic maps being used.

Paper	**Chaotic maps used**
Liu et al. [2018]	Logistic map/ Chebyshev map/ Baker map
Kopparthi et al. [2022]	Piecewise linear chaotic map
Cardoso et al. [2021]	Exponential chaotic map
Hobincu and Datcu [2018]	Generalized Henon Map
Sharma and Bhargava [2016]	Logistic map
Zheng and Bao [2022]	Logistic map / Sine map
Garcia-Bosque et al. [2019]	Logistic map

Table 2 summarizes applications on each state-of-the-art paper as well as their approach to mitigate chaos degradation:

Table 2 – Comparison of state-of-the-art works in terms of application and their MFMCD (method to mitigate chaos degradation).

Paper	**Application**	**MFMCD**
Liu et al. [2018]	PRBG	Coupling chaotic maps
Kopparthi et al. [2022]	PRNG	XORed LFSR
Cardoso et al. [2021]	PRNG	XORed LFSR
Hobincu and Datcu [2018]	PRNG	XORed three variables operation
Sharma and Bhargava [2016]	Image encryption	Two-step logistic map
Zheng and Bao [2022]	Image encryption	Cascading maps
Garcia-Bosque et al. [2019]	Monte Carlo simulation	Changes on the control parameters

It is noteworthy that chaotic maps may be employed not only in the applications referenced in this section but also in a diverse array of other problem domains, including, for instance, optimization Rani et al. [2023], and control theory Chen et al. [2016]. This serves to underscore the significance that such systems hold in contemporary times.

Chapter 3

Development

In this work, we propose two systems to analyze the influence of using different numeric representations on chaos degradation. To do it, the first one, which is called "Bi-coupled system" from now on, is a hardware architecture bi-coupling the tent map with the Bernoulli map. The second one, called "Sine system" from now on, was made in software and it is based on the sine map. Herewith, the next two subsections present the details regarding the development of these systems.

3.1 Bi-coupled System

In this system, it was used VHDL for the description of the digital circuit along with the Xilinx Vivado 2019.1 synthesis tool. In addition, the hardware architecture was mapped into the FPGA device *Artix 7 XC7A100TCSG324*. Furthermore, to reproduce the experimental tests done in this work, one can use the hardware setup shown in Fig. 16, which uses a Nexys 4 board along with a system generator tool and Matlab on a personal computer.

For the fixed-point representation used in this first system, we chose to represent the numbers using 0 bits for the integer part and 32 bits for the fractional part. Such a decision was made based on the fact that the domain of both the tent map and the Bernoulli map belongs to the interval $[0, 1)$.

To implement the tent map in hardware, it was necessary to pay attention to the singularities of the algebraic operations of its recurrence equation. It is known that to compute the subtraction of two real numbers x_1 and x_2, i.e., $x_1 - x_2$, when using the standard fixed-point format, one can simply add x_1 with the 2's complement of x_2. Nonetheless, once the 2's complement of a number in the HUB format is performed by bit-wise inversion Hormigo and Villalba [2016], the subtraction can be obtained by extending the implicit least significant bit of x_1 and the 2's complement of x_2, adding x_1 with the 2's complement of x_2, and then, performing truncation to have the desired

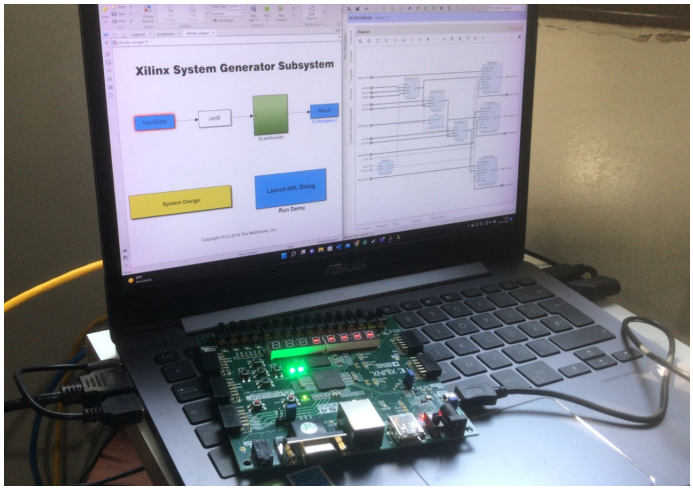

Figure 16 – Experimental setup using the Nexys 4 board with the low-cost FPGA Xilinx *Artix 7 XC7A100TCSG324*.

number of bits. For example, considering $x_1 = 11000000000000000000000000000000_2$ and $x_2 = 01000000000000000000000000000000_2$, we can compute $x_1 - x_2$, in HUB format, using the following steps:

1. Obtain the twos complement of x_2 performing an operation of bit-wise inversion, that is: $10111111111111111111111111111111_2$
2. Extend the implicit least significant bit of x_1, on variable $\hat{x}_1$, as well as the implicit bit of the twos complement of x_2, on variable $\hat{x}_2$, as so we have the following:
   $\hat{x}_1 = 110000000000000000000000000000001_2$ and
   $\hat{x}_2 = 101111111111111111111111111111111_2$.
3. Add $\hat{x}_1$ with $\hat{x}_2$, resulting in: $100000000000000000000000000000000_2$.
4. Truncate the result obtained in the last step to have the desired number of bits, which is 32 bits in this example, resulting in $10000000000000000000000000000000_2$.

Herewith, the proposed tent map using the HUB format can be implemented as shown in Fig. 17-A. Since the subtraction only occurs when $x_n \geq 1/2$, as seen in (2.5), and because of $1/2 = 0.100...0_2$, the most significant bit, $x_n[31]$, is the only bit that matters to verify the aforesaid condition. Thus, if the most significant bit is set to '1', then the input of the tent map is greater than 1/2. Otherwise, the number is less than 1/2.

Seeing Fig. 17-A, the selector of the multiplexer (MUX) is set to "0" if x_n is less than half, making the output of the map be $x_{n+1} = \mu x_n$. Otherwise, the selector of the MUX would be set to "1" and the output of the chaotic map would be equal to $x_{n+1} = \mu(1 - x_n)$. Since all the 32 bits of the word length were being used for the fractional part, and as the control parameter of the tent map needs to belong to the interval $(1, 2]$ to

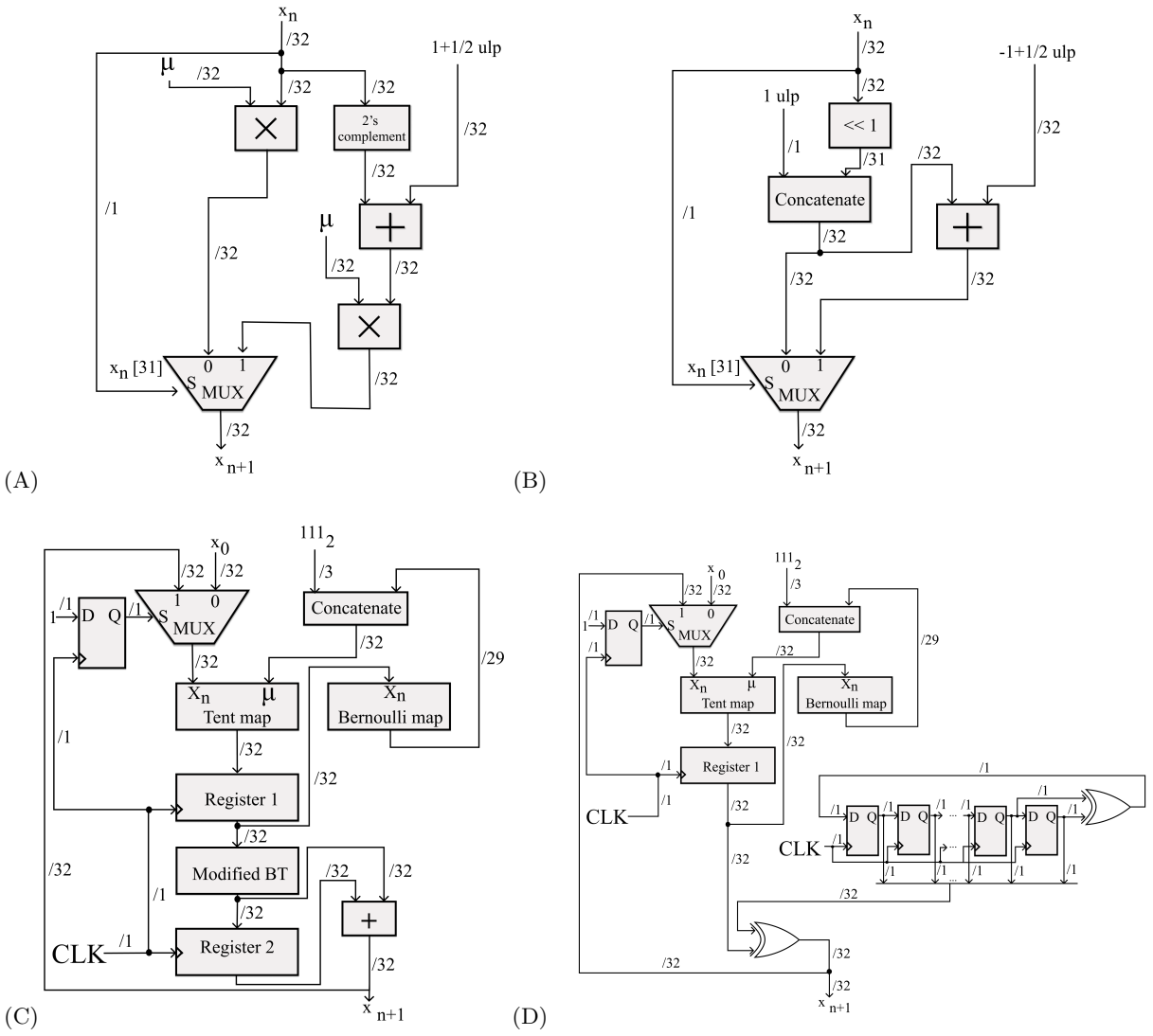

Figure 17 – Proposed hardware architectures using the HUB format for the following cases: (A) Tent map. (B) Bernoulli Map. (C) Bi-coupled map. (D) Bi-coupled map using an LFSR. For the addition and multiplication operations, the following steps were taken: first, explicitly extend the least significant bit of the operands; second, operate using conventional arithmetic; third, truncate the result to have 32 bits. The block "Concatenate" indicates the concatenation of bits, the symbol "<< 1" corresponds to the shift left of one bit, and $x_n[31]$ corresponds to the most significant bit of the word length of 32 bits.

present a chaotic behavior, the parameter μ was computed in such a way that it also had implicitly one bit set for the integer part, which makes sure that the control parameter μ is greater than one and that it belongs to the interval in which the tent map has a positive Lyapunov Exponent. Furthermore, once it is not possible to represent exactly the integer number "1" in the HUB format, the symbol "1 + 1/2 ulp" indicates that the ERN of it consists of the value "1" added half ulp.

For the Bernoulli map, a similar approach was used (see Fig. 17-B). Since it is not possible to represent exactly the number "2" in the HUB format, to approximate

the multiplication by two, the following steps were taken: first, shift left the bits of the represented number by one bit, and then set the least explicit significant bit to "1". The multiplexer in Fig. 17-B works the same way it was used for the tent map.

As will be demonstrated in section 5, empirical tests indicate that the iterations of the hardware architectures of the tent map and Bernoulli map (Fig. 17-A and Fig. 17-B) have short periods for a variety of values of the control parameter, which makes these systems not suitable in certain applications, such as in image encryption. This limited period is due to the dynamical degradation when using finite precision.

Nevertheless, to mitigate the dynamical degradation process, this project applies a bi-coupling approach, which consists of a particular case shown in Liu et al. [2018]. Thus, the output of the tent map influences the input of the Bernoulli map and vice versa. In the proposed system, the least significant 29 explicit bits of the parameter μ of the tent map are set to be equal to the least significant 29 explicit bits of the output of the Bernoulli map, whereas the 3 remaining explicit bits of the μ parameter was set to have a fixed value of "111_2", making the parameter μ assume values in the interval $[1.75, 2]$. This decision was made based on the fact that the Lyapunov Exponent of the tent map is high for values of the control parameter close to "2", as well as it guarantees that the tent map presents a chaotic dynamic behavior in the system. Thus, the control parameter of the tent map of the architecture is dictated by the output of the Bernoulli map. Fig. 17-C shows the proposed architecture done in this work. Note that the control parameter of the tent map is determined based on the iterations of the Bernoulli map; therefore, the parameter μ is not an input of the hardware architecture.

A D flip-flop with an input set to the logical high was used in Fig. 17-C. The flip-flop makes the selector (S) of the MUX to be equal to the logical low in the first iteration, and, therefore, the output of the MUX will be equal to the initial condition (x_0) of the proposed chaotic system. Then, after the first iteration has been computed, the output of the flip-flop will assume the logical high, as it will receive a clock pulse, making the output of the MUX equal to the previous value generated by the hardware architecture.

Furthermore, to make the proposed hardware architecture produce uniformly distributed sequences, an operation called "modified BT" was used, which is slightly different from the bit transformation (BT) suggested in Jiang and Wu [2009]. The "modified BT" operation takes as input an array x, of 32 bits, as described in (3.1). Then, it inverts half of the input bits, as shown in (3.2). The other half, described in (3.3), is also used to compute f_h (see (3.4)), where the bit-wise XOR operation is performed. The result of the "modified BT" operation, called "MBT{x}", shown in (3.5), is obtained by concatenating f_h and f_l. Finally, after computing it, an adder is introduced into the proposed chaotic system, adding two sub-states of the system, contributing to avoiding its dynamical degradation.

$$x = b_{31}b_{30}b_{29}\ldots b_3b_2b_1b_0, \tag{3.1}$$

$$f_l = b_0b_1b_2\ldots b_{15}, \tag{3.2}$$

$$f_h^{'} = b_{31}b_{30}b_{29}\ldots b_{16}, \tag{3.3}$$

$$f_h = f_h^{'} \oplus f_l, \tag{3.4}$$

$$MBT\{x\} = f_h, f_l. \tag{3.5}$$

For comparison purposes, the proposed hardware architecture in Fig. 17-C was also made using the standard fixed-point representation. However, it was not possible to represent this hardware architecture using the IEEE 754 single-precision floating-point standard since the "modified BT" operation can make the system produce values that are outside of the domain of both the tent map and Bernoulli map for such arithmetic.

Additionally, another approach to reduce the dynamical degradation was made in this project, as can be seen in Fig. 17-D. Hence, instead of using the "modified BT" operation along with an adder, this approach does a bit-wise XOR operation between the output of the tent map and the output of a Linear Feedback Shift Register (LFSR), similar to what was done in Cardoso et al. [2021]. Nevertheless, as it will be described in the next section, this approach does not present better results compared with the architecture shown in Fig. 17-C.

The following nomenclature will be used to refer to the proposed hardware architectures in the rest of this dissertation:

- Bi-HUB: the proposed Bi-coupled map using the HUB fixed-point format.
- Bi-Standard: the proposed Bi-coupled map using the standard fixed-point format.
- Bi LFSR-HUB: the proposed Bi-coupled map using the HUB fixed-point format in conjunction with the LFSR.

3.2 Sine System

The second system is based on the sine map, and it was implemented using the Python programming language, as it is a high-level language, and, to perform operations

with posit numbers, the library SoftPosit was employed. As most sine function implementations support only floating-point numbers, an approximation of this function using Sugeno fuzzy inference was applied. Unlike most of the approximations for the sine function, such as the Bhaskara's approximation Stroethoff [2014], the Sugeno fuzzy inference implemented in this work enabled performing the approximation with only simple operations of multiplication and addition. Therefore, while currently implemented in software, the scheme can also be implemented in hardware. Moreover, as will be shown in section 4, the errors introduced by the approximation of the sine function did not affect the pseudorandomness of the system.

We used the posit<32,2> representation, which has a 32-bit length and a maximum of two exponent bits. This format is widely used in the literature Zhang and Ko [2021]. For comparison purposes, we also implemented the sine map using IEEE 754 single-precision floating-point format.

We implemented the sine function using Sugeno fuzzy inference and considering three rules based on the graphical properties shown in Fig. 18, and described as follows:

- if x is A_1, then $z_1 = a_1 x + b_1$;
- if x is A_2, then $z_2 = a_2 x + b_2$;
- if x is A_3, then $z_3 = a_3 x + b_3$;

where A_1, A_2, and A_3 are fuzzy sets with triangle membership functions w_1, w_2, and w_3, as shown in Fig. 19. The membership functions are described by (3.6)-(3.8):

$$w_1(x) = \begin{cases} \frac{x}{\pi} + 1 & \text{if } -\pi < x \leq 0, \\ -\frac{x}{\pi} + 1 & \text{if } 0 < x < \pi, \\ 0 & \text{otherwise.} \end{cases} \tag{3.6}$$

$$w_2(x) = \begin{cases} \frac{x}{\pi} & \text{if } 0 < x \leq \pi, \\ -\frac{x}{\pi} + 2 & \text{if } \pi < x < 2\pi, \\ 0 & \text{otherwise.} \end{cases} \tag{3.7}$$

$$w_3(x) = \begin{cases} \frac{x}{\pi} - 1 & \text{if } \pi < x \leq 2\pi, \\ -\frac{x}{\pi} + 3 & \text{if } 2\pi < x < 3\pi, \\ 0 & \text{otherwise.} \end{cases} \tag{3.8}$$

We chose these membership functions because their sum is always equal to 1 for $0 \leq x \leq 2\pi$, which simplifies the computation of the sine approximation.

The values of z_1, z_2, and z_3 are given by (3.9)-(3.11):

$$z_1(x) = \frac{2}{\pi}x, \tag{3.9}$$

$$z_2(x) = -\frac{2}{\pi}x + 2, \tag{3.10}$$

$$z_3(x) = \frac{2}{\pi}x + 4. \tag{3.11}$$

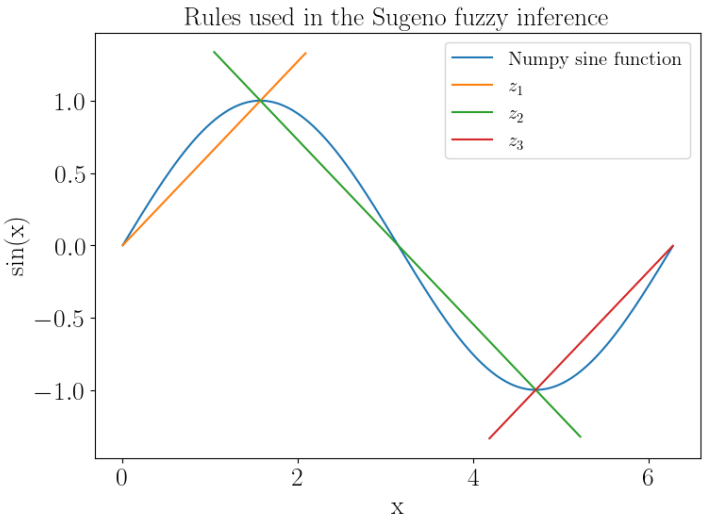

Figure 18 – Rules z_1, z_2, and z_3 used in Sugeno fuzzy inference to approximate the sine function. The sine function is shown in this picture only for reference.

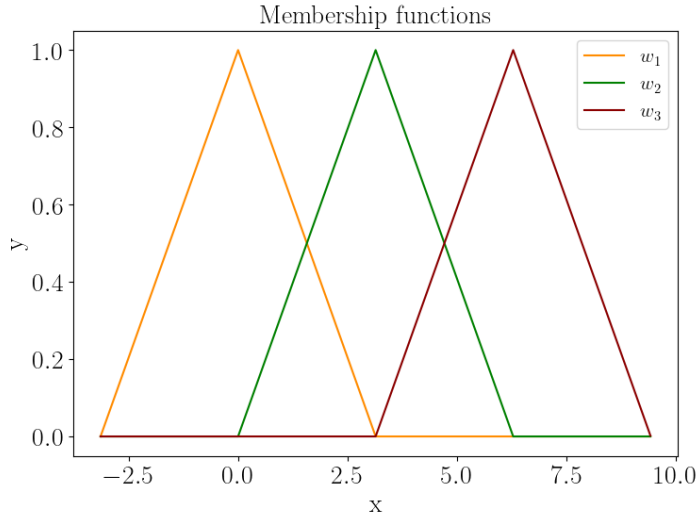

Figure 19 – Membership functions w_1, w_2, and w_3 of the fuzzy sets A_1, A_2, A_3, respectively.

Thus, (3.12) shows how we can compute the approximation of the sine function through Sugeno fuzzy inference. The approximation of the sine function using such an equation is illustrated in Fig. 20.

$$sin(x) \approx \frac{w_1(x)z_1(x) + w_2(x)z_2(x) + w_3(x)z_3(x)}{w_1(x) + w_2(x) + w_3(x)}. \tag{3.12}$$

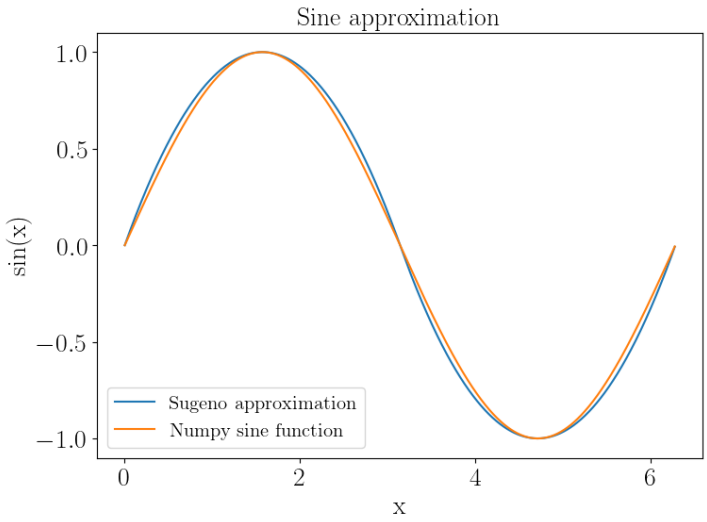

Figure 20 – Sine function approximated in comparison with the function natively implemented in Python.

Since the sine map domain is limited to the interval [0, 1], $w_3(x)$ and $w_3(x)z_3(x)$ will always be equal to zero. Moreover, within this interval, the sum of $w_1(x)$ and $w_2(x)$ is always equal to 1 due to the chosen membership functions. Therefore, (3.12) can be simplified to (3.13) specifically for the sine map domain:

$$sin(x) \approx w_1(x)z_1(x) + w_2(x)z_2(x). \tag{3.13}$$

By combining the sine map model (2.7) with the sine function approximation (3.13), we obtain (3.14):

$$x_{i+1} = \eta\left[w_1(\pi x_i)z_1(\pi x_i) + w_2(\pi x_i)z_2(\pi x_i)\right]. \tag{3.14}$$

Section IV reveals that the sine map shown in (3.14) does not show good results in terms of randomness, not passing in most of the tests in the NIST SP 800-22 test suite. Therefore, a perturbation method as a post-processing is necessary.

The perturbation method used in this paper is similar to the one in Öztürk and Kılıç [2018], which applies a LFSR. We perturb the 27 LSBs by performing a bit-wise XOR operation between the sine map iterations and the LFSR output with a feedback loop. It is worth noting that this method cannot be used with the IEEE standard for floating-point arithmetic. If we disturb the 27 LSBs with a feedback loop, some exponent bits will be disturbed, causing the system to go outside the sine map domain.

Furthermore, the sine map always outputs positive numbers within the interval [0, 1]. As a result, the most significant bits of the sine map iterations - which correspond to the sign and regime bits - will not change significantly after each iteration. This behavior is not desirable for pseudorandom number generators, which should produce bit sequences

with no apparent correlation or consistent repetition. To increase the entropy of the 5 MSBs, which were not disturbed by the LFSR, we applied an XOR operation between these bits and a 5-bit-length mask. This mask was generated by XORing the 5 least significant bits of the last iteration of the scheme and the next iteration of the sine map, as shown in Fig. 21. However, to prevent the system from falling outside the sine map domain, we did not apply a feedback loop to the five most significant bits. By using this approach, we can take all 32 bits of the output of the scheme into consideration, and the output of the complete scheme will be the set of real numbers.

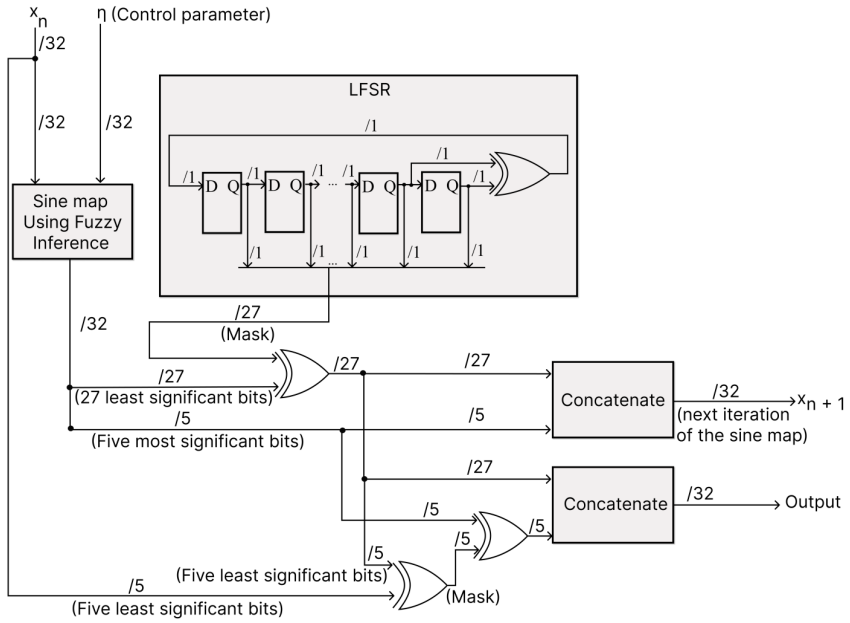

Figure 21 – Diagram block of the complete scheme using the sine map with perturbation.

Chapter 4

Results

4.1 Bi-coupled System

To demonstrate the effects of the perturbation method of this first system, the proposed hardware architectures were all tested to evaluate the period of its pseudo-orbits. Once it is being used a finite precision (32 bits), it can be inferred that the proposed systems have necessarily periodic pseudo-orbits. However, if the period of a digital chaotic system is sufficiently large, it can be used in a variety of applications. Thus, Table 3 shows the period identified after testing each hardware architecture when using the HUB format. A million iteration was generated for testing each hardware architecture, considering the initial condition $x_0 \approx 0.71$. For the tent map, the value of the μ parameter used is $\mu \approx 1.75$. Nonetheless, for the final implementation, the value of the μ parameter changes on each iteration. Moreover, similar results were obtained when testing the same systems for a variety of other initial conditions and also when using the standard fixed-point format.

Table 3 – Period and transient of the chaotic hardware architectures using the HUB format. The term "Transient" is used here to refer to the number of iterations that occurs initially that do not belong to the period of the pseudo-orbit itself. Similar results were obtained when testing for other initial conditions.

Hardware Architecture	**Period**	**Transient**
Tent map (Fig. 17-A) - $\mu \approx 1.75$	20013	41279
Bernoulli map (Fig. 17-B)	1	30
Bi-HUB	Not identified within the sample	Not identified within the sample
Bi LFSR-HUB	63563	17155

Based on Table 3, it is clear that the process of dynamical degradation makes the Bernoulli map converge to one unique value, which is $0.4\overline{9}$ when using the HUB format. Furthermore, Table 3 shows the tent map has periodic pseudo-orbits with a period of 2.0013×10^4, for $x_0 \approx 0.71$ and $\mu \approx 1.75$. Nevertheless, this period is not high enough for some applications. For example, in some image encryption schemes, when encrypting images of size 512×512, it would be necessary to iterate at least 2.62144×10^5 times the

chaotic system, which invalidates this system to be used in this kind of application. In addition, when using the perturbation method proposed in Fig. 17-C, it was not identified a period. Thus, in this case, it is possible to generate a sequence of 1×10^6 values, each value with 32 bits, without a defined period. Therefore, this is evidence that the proposed perturbation method in Fig. 17-C mitigates chaos degradation for this sample, justifying its use. Moreover, the perturbation method using an LFSR is able to mitigate chaos degradation as its period increases in relation to when using only the tent map or the Bernoulli map. However, a period of 6.3563×10^4 is identified in this case, which limits its use in some specific applications.

Table 4 – P-value results after running the SP 800-22 test suite for the proposed hardware architectures, considering the bi-coupled systems.

	Bi-HUB		**Bi-Standard**		**Bi LFSR-HUB**	
Test	**P-value**	**Proportion**	**P-value**	**Proportion**	**P-value**	**Proportion**
Frequency	0.514124	100/100	0.437274	98/100	0.437274	97/100
Block Frequency ($m = 128$)	0.779188	99/100	0.534146	99/100	0.000199	100/100
Cusum-Forward	0.153763	100/100	0.779188	98/100	0.897763	97/100
Cusum-Reverse	0.595549	100/100	0.366918	98/100	0.834308	97/100
Runs	0.071177	98/100	0.021999	98/100	0.000000	83/100
Longest Runs of Ones	0.319084	100/100	0.574903	99/100	0.678686	97/100
Rank	0.534146	100/100	0.739918	100/100	0.779188	99/100
FFT	0.153763	100/100	0.383827	100/100	0.016717	98/100
Non-overlapping Templates	0.595549	99/100	0.678686	99/100	0.016717	98/100
Overlapping Templates ($m = 9$)	0.153763	98/100	0.191687	100/100	0.554420	99/100
Universal	0.657933	99/100	0.616305	98/100	0.000000	95/100
Approximate Entropy	0.699313	99/100	0.224821	98/100	0.045675	97/100
Random Excursions ($x = +1$)	0.897763	56/57	0.057146	64/65	0.153763	52/54
Random Excursions Variant ($x = -1$)	0.062821	56/57	0.723129	64/65	0.236810	53/54
Linear Complexity ($M = 500$)	0.045675	99/100	0.678686	99/100	0.334538	98/100
Serial ($m = 16$, $\nabla\Psi_m^2$)	0.574903	99/100	0.108791	99/100	0.816537	99/100

To validate the proposed hardware architectures, the bi-coupled system was submitted to the NIST SP 800-22 test suite Bassham et al. [2010]. Accordingly, the NIST SP 800-22 test suite consists of a set of 15 statistical tests for random and pseudo-random number generators Bassham et al. [2010]. Each test has as output a number called "P-value", which belongs to the interval $[0, 1)$. If the output of the corresponding test has a P-value greater than a significance level α, typically equal to 0.01, then the sequence is considered random for the given statistical test. Thus, a system is considered to have pseudorandom properties if, for all tests, its P-values are greater than α. Herewith, Table 4 shows the results obtained for the NIST SP 800-22 test suite, considering 100 sequences of one million bits and a significance level $\alpha = 0.01$. Therefore, it is noticeable that, in the Bi-HUB and Bi-Standard cases, the proposed systems have passed all tests, as all P-values obtained were greater than α. Moreover, for the Bi LFSR-HUB case, the system has passed most of the tests, but not all of them, since some tests present P-value $< \alpha$. Herewith, although the bi-coupled system with an LFSR did not pass all tests, it is known that using an LFSR as a source of perturbation is a common practice Cardoso et al. [2021], Ahmed et al. [2018], Garcia-Bosque et al. [2016], and, therefore, the results obtained for the Bi

LFSR-HUB architecture are presented in this document only for compassion purposes.

In contemplation of validating the use of the proposed bi-coupled system as a PRNG, we also submitted the generated bitstream in the ENT test suite, a tool that has been extensively used to prove pseudorandom features of numerical sequences. By running 6 different statistical tests, this series of tests present a satisfactory indicator of the quality of random generators. Table 5 shows the results obtained for each numerical representation used, which reaffirms what was shown by NIST SP 800-22 test suite. These tests confirm that the bi-coupled system using HUB fixed-point or the standard fixed-point format presents outstanding pseudorandom features. Furthermore, the ENT result proves that the proposed PRNG can effectively be used in the Monte Carlo simulation as it got values close to π in the "Monte Carlo value for Pi" test.

Table 5 – ENT test suite results obtained for the hardware architectures and Lyapunov Exponent calculated based on Hegger et al., considering the bi-coupled systems.

Test	**Expected value**	**Bi-HUB**	**Bi-Standard**	**Bi LFSR-HUB**
Entropy (bits per bit)	1.0	1.0	1.0	1.0
Chi-square (abs)	-	3.12	1.06	3.88
Chi-square (percentage)	$10 \sim 90$	7.72	30.31	4.88
Arithmetic Mean	0.5	0.5001	0.5000	0.5001
Monte Carlo Value for Pi	3.14159265	3.14103	3.14051146	3.14490
Serial Correlation	0.0	0.000139	0.000091	-0.001353
Lyapunov Exponent	> 0	5.15388	5.15332	0.8514

Due to the nature of the perturbation method applied, the complexity of calculating the Lyapunov Exponent analytically is very high. Thus, an approximation of the Lyapunov Exponent was evaluated. The Lyapunov Exponent was calculated based on the method described by Kantz [1994]. Table 5 shows the results obtained for the proposed hardware architectures using different numerical representations. In each case, it was used the software available in Hegger et al., and results exhibit that the systems have a positive Lyapunov Exponent, an important factor to indicate the presence of chaotic properties.

The state space of the system is shown in Fig. 22. For all cases, the state space presents a complex shape due to the perturbation method applied. Moreover, Fig. 22 shows that the autocorrelation approximates a Dirac delta function in all approaches, showing that the sequence generated does not have an apparent correlation between its values. Ultimately, the histogram was obtained, as shown in Fig. 22, and the distribution for the cases (A) and (B) presented is close to a uniform distribution. However, in (C), it is seen a non-uniform distribution, reaffirms the results of the NIST SP 800-22 test suite, on which the system failed for the block frequency test.

Furthermore, Fig. 23-A exhibits the time series of the system evaluating the Bi-HUB and Bi-Standard approaches. Clearly, Fig. 23-A shows that the numerical format used could influence directly the output of the system since the pseudo-orbits of the

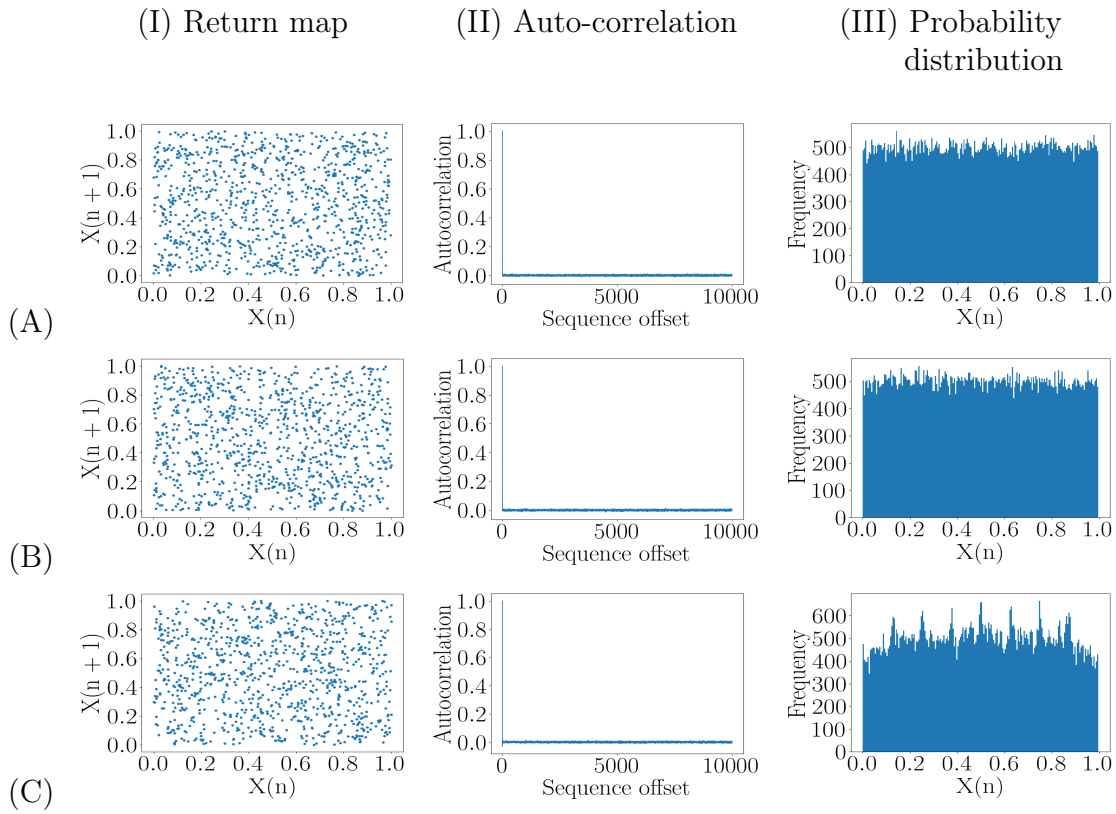

Figure 22 – Results for the proposed hardware architectures. Column (I) corresponds to the chaotic state space, column (II) shows the autocorrelation obtained, and column (III) exhibits the histograms of the systems. Cases (A), (B), and (C) correspond, respectively, to the following architectures: Bi-HUB, Bi-Standard, and Bi LFSR-HUB.

systems diverge after the first iteration. Additionally, Fig. 23-B reveals that using the HUB format has a positive effect of avoiding the chaos annulling condition when all 32 explicit bits of the initial condition are set to zero. Since this numerical representation has a half unit in the last place, such a condition makes the output of the chaotic maps different than zero, not degenerating the chaotic behavior. In contrast, for the same hardware using the standard fixed-point representation, if the input is set to be a bit vector with all bits set to zero, it will completely degenerate chaos as the output of the system will always be zero. Additionally, it is valid to mention that empirical tests were made to find more chaos annulling conditions for the architecture of Fig. 17-C, but no such values were found.

Table 6 summarises the hardware resources achieved for different representations. Notice that the proposed system, employing the HUB format (Bi-HUB), uses slightly fewer logical resources than the standard fixed-point representation (Bi-Standard). This is because, when adopting the HUB format in a given representation, some operations are simpler to be accomplished, such as the two's complement and rounding to the nearest. Moreover, the Bi-Standard architecture performs rounding by truncation, which corresponds to an effective rounding down. However, if we use the standard fixed-point

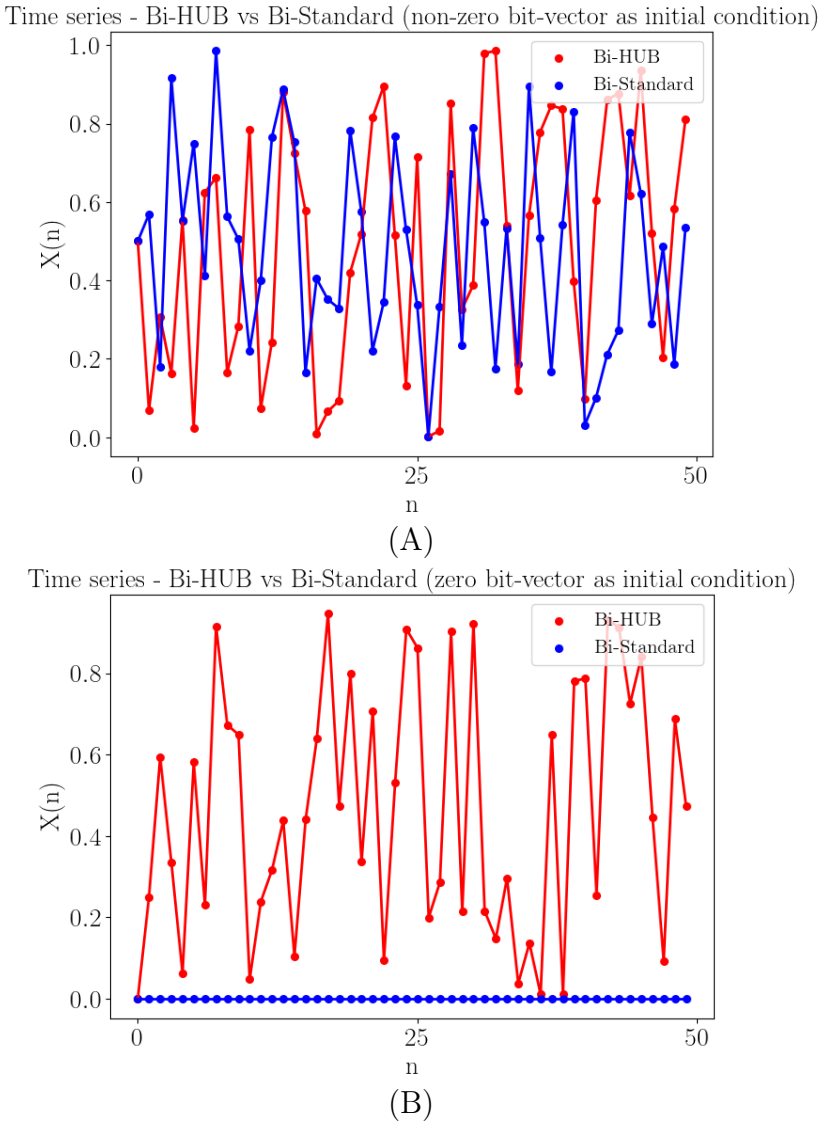

Figure 23 – Time series of the Bi-HUB architecture (in red) and the Bi-Standard architecture (in blue). In (A), both systems had the same bit vector as the initial condition. In (B), an initial condition of a bit vector full of zeros was used.

Table 6 – Comparison analysis in terms of hardware resources of different approaches. The following terms were used: LUT (look-up table), FF (flip-flop), DSP (digital signal processor), IO (input/output).

Chaotic system	**Reference**	**LUT**	**FF**	**DSP**	**IO**
Bi-HUB	**This work**	**242**	**65**	**8**	**66**
Bi-Standard	**This work**	**258**	**65**	**8**	**66**
Bi LFSR-HUB	**This work**	**226**	**65**	**8**	**66**
PRNG - Exponential map (32 bits)	Cardoso et al. [2021]	915	1101	6	66
PRNG - Exponential map (64 bits)	Cardoso et al. [2021]	1466	1256	48	130
PRNG - Henon Map	Hobincu and Datcu [2018]	856	521	32	20
CSS-2	Gugapriya et al. [2019]	549	192	72	97
PRNG - LCM Modified	Saber and Eid [2021]	5806	-	-	-
PWLCM	Kopparthi et al. [2022]	4215	578	0	-

Table 7 – Comparison analysis in terms of performance.

Chaotic system	Reference	Power (mW)	FMax (MHz)	Throughput (Mbps)	Latency (clock cycles)	Lyapunov Exponent
Bi-HUB	**This work**	**114**	**61.94**	**991.040**	**2**	**5.15388**
Bi-Standard	**This work**	**114**	**63.90**	**1022.400**	**2**	**5.15332**
Bi LFSR-HUB	**This work**	**113**	**64.81**	**2073.920**	**1**	**0.8514**
PRNG - Exponential map (32 bits)	Cardoso et al. [2021]	96	121.95	23.508	166	0.7576
PRNG - Exponential map (64 bits)	Cardoso et al. [2021]	90	43.47	111.283	25	2.2733
PRNG - Henon Map	Hobincu and Datcu [2018]	128	-	470.592	-	-
CSS-2	Gugapriya et al. [2019]	-	-	-	-	0.00154
PRNG - LCM Modified	Saber and Eid [2021]	300	80.592	-	-	-
PWLCM	Kopparthi et al. [2022]	10	106	1296.000	31	1.16

representation performing rounding to the nearest, it would be necessary to use even more hardware resources, as additional logic circuits would be required.

Furthermore, Table 6 compares the state-of-the-art chaotic system implementations regarding resource consumption, and Table 7 compares the approaches in terms of performance. These tables show that the proposed hardware architectures are competitive with the state-of-the-art papers in terms of hardware resources and performance. The power consumption calculated for the hardware architectures of this paper, in Table 7, corresponds to the total on-chip power indicated by the Xilinx Vivado 2019.1 synthesis tool, being equivalent to the sum of static and dynamic power obtained. Besides, for this work, the throughput calculated in Table 7 was based on equation 4.1:

$$T = \frac{Fmax \cdot B}{Latency} \tag{4.1}$$

Where, $Fmax$ is the maximum operating frequency, B is the number of output bits of the architecture, and $Latency$ corresponds to the number of clock cycles necessary to produce an iteration of the system.

Summarising, the hardware architecture that presents the best trade-off between chaotic properties, pseudorandom features, and consumption of logical resources is the Bi-HUB architecture. The main advantages of using the HUB format identified are:

- It prevents the system from falling into a chaos-annulling condition when having an input with a bit vector full of zeros.
- It had a positive effect on lowering the use of hardware resources. This can make a considerable impact in platforms that have tight constraints, such as embedded systems, on which resources available are a key factor.

4.2 Sine System

The sine map implemented without any perturbation was submitted to the NIST SP 800-22 test suite Bassham et al. [2010]. Thus, Table 8 shows the results obtained for the sine map without any perturbation method using the IEEE 754 floating-point representation with single precision, as well as the posit<32,2> format. It is clear that in both cases, the sine map failed all tests, proving the necessity of applying a perturbation method to mitigate dynamical degradation and increase the randomness. Table 8 also shows the P-values obtained when applying the proposed perturbation method. It can be seen that the complete scheme passed all the tests of the NIST SP 800-22 test suite, demonstrating that it possesses pseudorandom properties.

Table 8 – The results obtained using the SP 800-22 test suite for all architectures using the sine map. In all cases, $\eta = 0.962$ was used, and 100 sequences of one million bits each were generated.

	Sine Map - Float		**Sine Map - Posit**		**Sine Map - Posit with Perturbation**	
Test	**P-value**	**Proportion**	**P-value**	**Proportion**	**P-value**	**Proportion**
Frequency	0.000000	0/100	0.000000	0/100	0.437274	100/100
Block Frequency ($m = 128$)	0.000000	0/100	0.000000	0/100	0.171867	100/100
Cusum-Forward	0.000000	0/100	0.000000	0/100	0.383827	99/100
Cusum-Reverse	0.000000	0/100	0.000000	0/100	0.236810	100/100
Runs	0.000000	0/100	0.000000	0/100	0.474986	100/100
Longest Runs of Ones	0.000000	0/100	0.000000	0/100	0.798139	100/100
Rank	0.000000	0/100	0.000000	0/100	0.719747	100/100
FFT	0.000000	0/100	0.000000	0/100	0.249284	100/100
Non-overlapping Templates	0.000000	0/100	0.000000	0/100	0.350485	98/100
Overlapping Templates ($m = 9$)	0.000000	0/100	0.000000	0/100	0.554420	98/100
Universal	0.000000	0/100	0.000000	0/100	0.946308	99/100
Approximate Entropy	0.000000	0/100	0.000000	0/100	0.574903	98/100
Random Excursions ($x = +1$)	-	-	-	-	0.494392	58/58
Random Excursions Variant ($x = -1$)	-	-	-	-	0.739918	57/58
Linear Complexity ($M = 500$)	0.000000	0/100	0.000000	0/100	0.085587	97/100
Serial ($m = 16$, $\nabla\Psi^2_m$)	0.000000	0/100	0.000000	0/100	0.978072	98/100

Furthermore, we submitted the sine map implemented without perturbation to the ENT test suite Walker [2008], another tool commonly used in the literature to validate pseudorandom sequences. As shown in Table 9, the sequence generated using posit numbers had higher entropy per bit and a serial correlation closer to zero than the sequence generated employing floating-point numbers. This is an advantage of using the posit over floating-point representation because more bits are being used in the fractional part in the interval [0, 1], which is the domain of the sine map. In contrast, when using the standard floating-point representation, the 9 most significant bits (sign bit and exponent bits) have a lower variation through the map iterations.

Additionally, the Lyapunov Exponent is an important factor in determining if a map presents chaotic behavior. A system is considered chaotic if it has at least one positive Lyapunov Exponent Dingwell [2006]. Table 9 shows the Lyapunov Exponent calculated for the schemes proposed in this work when $\eta = 0.98$ and using the Kantz's method Hegger

Table 9 – Results of the ENT test suite and the estimated Lyapunov Exponent for the architectures presented in this second system.

Test	**Expected value**	**Sine Map - Float**	**Sine Map - Posit**	**Sine Map - Posit with Perturbation**
Entropy (bits per bit)	1.0	0.993711	0.999688	1.000000
Chi-square (abs)	-	27859.75	1383.28	0.51
Chi-square (percentage)	$10 \sim 90$	0.01	0.01	47.50
Arithmetic Mean	0.5	0.5467	0.5104	0.4998
Monte Carlo Value for π	3.14159265	3.635196352	3.499955000	3.128371284
Serial Correlation	0.0	0.122755	0.041227	0.002370
Lyapunov Exponent	> 0	0.5488	0.49807	6.0158

et al.. For all cases, the systems exhibit a positive Lyapunov Exponent.

Furthermore, Fig. 24 presents, in lines (A) and (B) of column (I), the return map of the sine map using the Sugeno fuzzy inference approximation with posit numbers, and with floating-point numbers. For the complete scheme with perturbation, line (C) of column (I), it is noticeable that the shape of the attractor is more complex than in cases (A) and (B), in accordance with the specified objectives, and it has been completely modified due to the perturbation method applied. The auto-correlation function is presented in column (II), in which, for the complete scheme, the auto-correlation obtained approximates a delta Dirac function, indicating that there is no apparent linear correlation between the numbers generated of each sequence.

Figure 24 also shows, in column III, the histogram obtained of each scheme. For cases (A), (B), and (C), the histogram has non-uniform distribution. For the complete scheme, in line (C), this happens mostly because, as posit representation has tapered precision, its values will mass around zero, and it will have a higher precision for numbers in the interval $[-1, 1]$, in contrast to extremely high or extremely low numbers Murillo et al. [2020]. Therefore, we conclude that using the posit format will result consequently in a non-uniform distribution. However, since most personal computers and hardware architectures do not support posit numbers natively, one might need to do a representation number conversion for the schemes that use posit numbers in most scenarios. As an example, line (D), it is presented the results after taking all 32 bits of the output of the complete posit-base scheme and converting the sequence of bits into integer numbers. For instance, if the output of the complete scheme using posit numbers is $00000000000000000000000000001001_2$, this value will be converted into the integer number 9_{10}. For this case, in line (D), we have obtained a histogram distribution close to a uniform shape because, for integer numbers, we have the same precision for all of their domain values.

Moreover, in column (IV) of Fig. 24, it is presented the bifurcation diagram for the approaches. For the sine map without a perturbation method being applied, it can be seen that, as the control parameter is increasing, there are double period events until it gets close to the value of 1, where there is a continuum. This is expected, once the sine map presents a chaotic behavior only for control parameter values inside the interval [0.87, 1],

as described in Mansouri and Wang [2020]. Nonetheless, when applying the perturbation method proposed, the bifurcation diagram approximates a continuum in a considerable part of the control parameter domain, evidencing that the perturbation approach mitigates dynamical degradation and increases the chaotic range of the map.

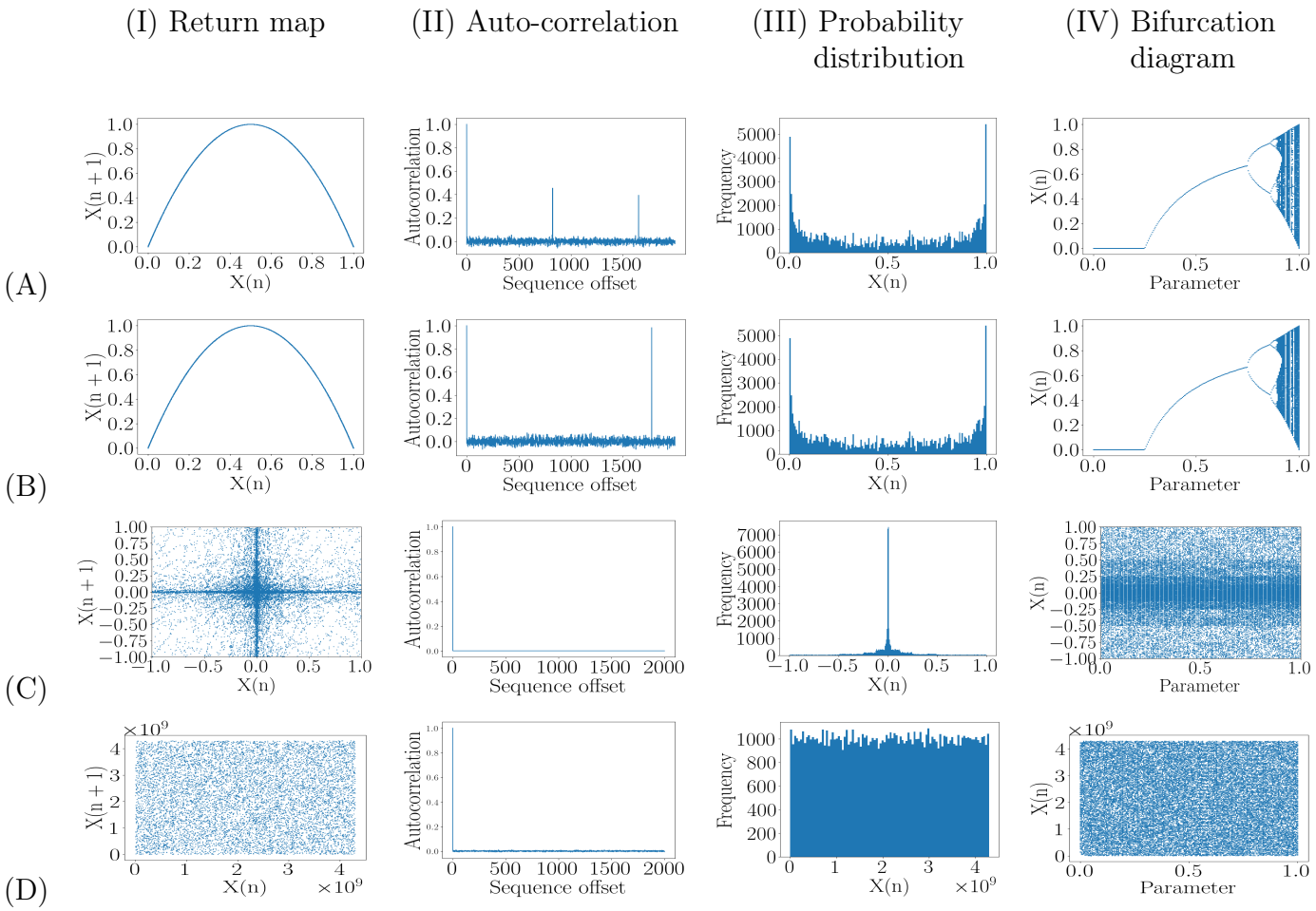

Figure 24 – Results for the proposed scheme. Column (I) corresponds to the return map, column (II) shows the autocorrelation obtained, and column (III) exhibits the histograms of the systems. Cases (A), (B), (C), and (D) correspond, respectively, to the following: sine map using floating-point numbers; sine map using posit numbers; sine map with perturbation using posit numbers; sine map with perturbation taking all the 32 bits of the output of the complete posit-base scheme and converting the sequence of bits into integer numbers. For case (C), it was plotted only values between -1 and 1, for better visualization.

Hence, in contemplation of the results obtained, the main advantages of using the posit representation in the algorithm proposed, in summary, are:

- The posit number representation used has a higher entropy per bit than floating-point numbers, mitigating dynamical degradation.

- Although other methods could be employed to obtain similar results in floating-point, the posit representation allowed perturbing more bits without making the sine map diverge, thereby allowing for more possibilities to mitigate the process of chaos degradation. Specifically in this paper, the least 27 significant bits of the sine map were perturbed.

As a trade-off, the disadvantage identified when using posit numbers for this chaos-based scheme is:

- As most hardware architectures and personal computers do not support natively posit numbers, one might need to do a conversion of posit numbers into another representation, depending on the application.

Chapter 5

Conclusions

In this work, we described two systems in order to analyze the influence of using different numerical representations on the process of chaos degradation. In the first system, a new chaos-based PRNG employing the HUB format that bi-couples the tent map in conjunction with the Bernoulli map is proposed. We effectively implemented the hardware architecture in the low-cost FPGA *XC7A100TCSG324*. Results show the proposed system has chaotic behavior as well as pseudorandom properties, with a positive Lyapunov Exponent, a uniform histogram, an auto-correlation function similar to a Dirac delta function, and passing in statistical test suites for random sequences. These characteristics validate the use of the proposed chaotic system as a PRNG, being suitable for use in some applications involving, for instance, instrumentation and measurements, such as the Monte Carlo simulation. Moreover, the adoption of the HUB format has a positive impact since it avoids the system from having chaos annulling conditions when using an input vector with all zeros as an initial condition, and it also makes the architecture employ fewer hardware resources. Besides, results show that the proposed Bi-coupled system is competitive with other state-of-the-art systems, as it demands low resources and power consumption.

It was proposed, in the second system, a novel chaos-based PRNG that uses the sine map and employs the Sugeno fuzzy inference to approximate the sine function with simple arithmetic operations. In this chaotic map, it was employed both the IEEE standard for floating-point arithmetic and the posit<32,2> format. The results indicate that the novel algorithm presents better performance when using posit numbers than the floating-point representation. For instance, the algorithm shows higher entropy and no serial correlation when using posit arithmetic. Additionally, we found that the proposed perturbation method could only be applied to posit numbers due to the potential risk of the algorithm leaving the domain of the sine map when using floating-point numbers. It was also observed that the proposed algorithm scheme passed all NIST SP 800-22 tests, demonstrating its suitability as a PRNG. However, it is important to note that most hardware/software platforms as well as personal computers do not natively support posit

numerical format, which could limit its wider application.

We conclude that this work has made contributions to evaluating the influence of using non-conventional formats on digital chaotic systems, such as using the posit representation. Nevertheless, the sine system was evaluated on software even though it demands only a few basic arithmetic operations. Therefore, future works could produce a hardware implementation of the proposed sine system using reconfigurable hardware, such as the FPGA *XC7A100TCSG324*. Furthermore, future works could explore the dynamics of the systems when using 64 bits for the word length. Thus, in this way, a comparison analysis could also be performed for different numerical precisions.

Finally, future works could also apply the systems in image encryption schemes to demonstrate and evaluate their usage in more applications. Thus, as the proposed systems can produce sequences that appear random, they could be used as a component to contribute to the confusion and diffusion properties of encryption schemes. This could, therefore, potentially produce cipher images that have no valuable information for an intruder, contributing to security systems.

References

O. A. Aboulseoud and S. M. Ismail. FPGA floating point fractional-order chaotic map image encryption. In *2019 31st International Conference on Microelectronics (ICM)*. IEEE, dec 2019. doi: 10.1109/icm48031.2019.9021500. URL https://doi.org/10.1109/icm48031.2019.9021500.

L. Aguirre. *Sistemas Dinâmicos Não Lineares: Conceitos e Análise de Dados*. 2023.

H. A. S. Ahmed, M. F. Zolkipli, S. M. Ismail, and Y. A. Alsariera. Pseudo random bits' generator based on tent chaotic map and linear feedback shift register. *Advanced Science Letters*, 24(10):7383–7387, oct 2018. doi: 10.1166/asl.2018.12946. URL https://doi.org/10.1166/asl.2018.12946.

K. T. Alligood, T. D. Sauer, and J. A. Yorke. *CHAOS An Introduction to Dynamical Systems*. Springer, 1997.

J. Arif, M. A. Khan, B. Ghaleb, J. Ahmad, A. Munir, U. Rashid, and A. Y. Al-Dubai. A novel chaotic permutation-substitution image encryption scheme based on logistic map and random substitution. *IEEE Access*, 10:12966–12982, 2022. doi: 10.1109/access.2022.3146792. URL https://doi.org/10.1109/access.2022.3146792.

M. Baptista. Cryptography with chaos. *Physics Letters A*, 240(1-2):50–54, Mar. 1998. doi: 10.1016/s0375-9601(98)00086-3. URL https://doi.org/10.1016/s0375-9601(98)00086-3.

L. E. Bassham, A. L. Rukhin, J. Soto, J. R. Nechvatal, M. E. Smid, E. B. Barker, S. D. Leigh, M. Levenson, M. Vangel, D. L. Banks, N. A. Heckert, J. F. Dray, and S. Vo. A statistical test suite for random and pseudorandom number generators for cryptographic applications. Technical report, 2010. URL https://doi.org/10.6028/nist.sp.800-22r1a.

M. B. R. Cardoso, S. S. da Silva, L. G. Nardo, R. M. Passos, E. G. Nepomuceno, and J. Arias-Garcia. A new PRNG hardware architecture based on an exponential chaotic map. In *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, may 2021. doi: 10.1109/iscas51556.2021.9401653. URL https://doi.org/10.1109/iscas51556.2021.9401653.

J.-H. Chen, H.-T. Yau, and J.-H. Lu. Implementation of FPGA-based charge control for a self-sufficient solar tracking power supply system. *Applied Sciences*, 6(2):41, Feb. 2016. doi: 10.3390/app6020041. URL https://doi.org/10.3390/app6020041.

J. D. Cook. Posits: tapered precision real numbers. https://www.johndcook.com/blog/2018/04/11/anatomy-of-a-posit-number/, 2018. (Accessed on 04/27/2023).

R. M. Corless. What good are numerical simulations of chaotic dynamical systems? *Computers & Mathematics with Applications*, 28(10-12):107–121, nov 1994. doi: 10.1016/0898-1221(94)00188-x. URL https://doi.org/10.1016/0898-1221(94)00188-x.

P. Cvitanović, R. Artuso, R. Mainieri, G. Tanner, and G. Vattay. *Chaos: Classical and Quantum*. Niels Bohr Inst., Copenhagen, 2016. URL http://ChaosBook.org/.

Y. Deng, H. Hu, W. Xiong, N. N. Xiong, and L. Liu. Analysis and design of digital chaotic systems with desirable performance via feedback control. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(8):1187–1200, Aug. 2015. doi: 10.1109/tsmc.2015.2398836. URL https://doi.org/10.1109/tsmc.2015.2398836.

R. L. Devaney. *An Introduction to Chaotic Dynamical Systems*. Addison-Wesley Publishing Company, New York, 1987.

J. B. Dingwell. Lyapunov exponents, Apr. 2006. URL https://doi.org/10.1002/9780471740360.ebs0702.

D. J. Driebe. *Fully Chaotic Maps and Broken Time Symmetry*, volume 4 of *Nonlinear Phenomena and Complex Systems*. Springer Netherlands, 1999. ISBN 978-90-481-5168-4.

L. Fortuna, M. Frasca, and A. Rizzo. Chaotic pulse position modulation to improve the efficiency of sonar sensors. *IEEE Transactions on Instrumentation and Measurement*, 52(6):1809–1814, dec 2003. doi: 10.1109/tim.2003.820452. URL https://doi.org/10.1109/tim.2003.820452.

M. François, D. Defour, and P. Berthomé. A pseudo-random bit generator based on three chaotic logistic maps and IEEE 754-2008 floating-point arithmetic. In *Lecture Notes in Computer Science*, pages 229–247. Springer International Publishing, 2014. doi: 10.1007/978-3-319-06089-7_16. URL https://doi.org/10.1007/978-3-319-06089-7_16.

M. Garcia-Bosque, C. Sanchez-Azqueta, G. Royo, and S. Celma. Lightweight ciphers based on chaotic map - LFSR architectures. In *2016 12th Conference on Ph.D. Research in Microelectronics and Electronics (PRIME)*. IEEE, jun 2016. doi: 10.1109/prime.2016.7519519. URL https://doi.org/10.1109/prime.2016.7519519.

M. Garcia-Bosque, A. Perez-Resa, C. Sanchez-Azqueta, C. Aldea, and S. Celma. Chaos-based bitwise dynamical pseudorandom number generator on FPGA. *IEEE Transactions*

on Instrumentation and Measurement, 68(1):291–293, jan 2019. doi: 10.1109/tim.2018.2877859. URL https://doi.org/10.1109/tim.2018.2877859.

J.-M. Ginoux and C. Letellier. Van der pol and the history of relaxation oscillations: Toward the emergence of a concept. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 22(2):023120, June 2012. doi: 10.1063/1.3670008. URL https://doi.org/10.1063/1.3670008.

G. Gugapriya, K. Rajagopal, A. Karthikeyan, and B. Lakshmi. A family of conservative chaotic systems with cyclic symmetry. *Pramana*, 92(4):48, apr 2019. doi: 10.1007/s12043-019-1719-1. URL https://doi.org/10.1007/s12043-019-1719-1.

J. L. Gustafson and I. T. Yonemoto. Beating floating point at its own game: Posit arithmetic. *Supercomputing Frontiers and Innovations*, 4(2), June 2017. doi: 10.14529/jsfi170206. URL https://doi.org/10.14529/jsfi170206.

H. S. Hassan and S. M. Ismail. CLA based floating-point adder suitable for chaotic generators on FPGA. In *2018 30th International Conference on Microelectronics (ICM)*. IEEE, dec 2018. doi: 10.1109/icm.2018.8704074. URL https://doi.org/10.1109/icm.2018.8704074.

R. Hegger, H. Kantz, and T. Schreiber. Nonlinear time series routines. https://www.pks.mpg.de/tisean/Tisean_3.0.1/index.html. (Accessed on 04/21/2022).

R. Hobincu and O. Datcu. FPGA implementation of a chaos based PRNG targetting secret communication. In *2018 International Symposium on Electronics and Telecommunications (ISETC)*. IEEE, nov 2018. doi: 10.1109/isetc.2018.8583863. URL https://doi.org/10.1109/isetc.2018.8583863.

J. Hormigo and J. Villalba. Optimizing DSP circuits by a new family of arithmetic operators. In *2014 48th Asilomar Conference on Signals, Systems and Computers*. IEEE, nov 2014. doi: 10.1109/acssc.2014.7094576. URL https://doi.org/10.1109/acssc.2014.7094576.

J. Hormigo and J. Villalba. Measuring improvement when using HUB formats to implement floating-point systems under round-to-nearest. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 24(6):2369–2377, jun 2016. doi: 10.1109/tvlsi.2015.2502318. URL https://doi.org/10.1109/tvlsi.2015.2502318.

C. Jiang and S. Wu. A valid algorithm of converting chaos sequences to uniformity pseudo-random ones. In *2009 International Symposium on Information Engineering and Electronic Commerce*. IEEE, may 2009. doi: 10.1109/ieec.2009.67. URL https://doi.org/10.1109/ieec.2009.67.

M. Jiménez, R. Palomera, and I. Couvertier. *Introduction to Embedded Systems*. Springer New York, 2014. doi: 10.1007/978-1-4614-3143-5. URL https://doi.org/10.1007/978-1-4614-3143-5.

H. Kantz. A robust method to estimate the maximal Lyapunov exponent of a time series. *Physics Letters A*, 185(1):77–87, jan 1994. doi: 10.1016/0375-9601(94)90991-1. URL https://doi.org/10.1016/0375-9601(94)90991-1.

V. R. Kopparthi, A. Kali, S. L. Sabat, K. K. Anumandla, R. Peesapati, and J. A. E. Fouda. Hardware architecture of a digital piecewise linear chaotic map with perturbation for pseudorandom number generation. *AEU - International Journal of Electronics and Communications*, 147:154138, apr 2022. doi: 10.1016/j.aeue.2022.154138. URL https://doi.org/10.1016/j.aeue.2022.154138.

L. Liu, B. Liu, H. Hu, and S. Miao. Reducing the dynamical degradation by bi-coupling digital chaotic maps. *International Journal of Bifurcation and Chaos*, 28 (05):1850059, may 2018. doi: 10.1142/s0218127418500591. URL https://doi.org/10.1142/s0218127418500591.

A. Mansouri and X. Wang. A novel one-dimensional sine powered chaotic map and its application in a new image encryption scheme. *Information Sciences*, 520:46–62, May 2020. doi: 10.1016/j.ins.2020.02.008. URL https://doi.org/10.1016/j.ins.2020.02.008.

R. M. May. Simple mathematical models with very complicated dynamics. *Nature*, 261 (5560):459–467, June 1976. doi: 10.1038/261459a0. URL https://doi.org/10.1038/261459a0.

S. Mishra, R. R. Tripathi, and D. K. Tripathi. Implementation of configurable linear feedback shift register in VHDL. In *2016 International Conference on Emerging Trends in Electrical Electronics &amp Sustainable Energy Systems (ICETEESES)*. IEEE, Mar. 2016. doi: 10.1109/iceteeses.2016.7581406. URL https://doi.org/10.1109/iceteeses.2016.7581406.

R. Murillo, A. A. D. Barrio, and G. Botella. Deep PeNSieve: A deep learning framework based on the posit number system. *Digital Signal Processing*, 102:102762, July 2020. doi: 10.1016/j.dsp.2020.102762. URL https://doi.org/10.1016/j.dsp.2020.102762.

B. Muthuswamy and S. Banerjee. *A Route to Chaos Using FPGAs*. Springer International Publishing, 2015. doi: 10.1007/978-3-319-18105-9. URL https://doi.org/10.1007/978-3-319-18105-9.

İ. Öztürk and R. Kılıç. Digitally generating true orbits of binary shift chaotic maps and their conjugates. *Communications in Nonlinear Science and Numerical Simulation*,

62:395–408, Sept. 2018. doi: 10.1016/j.cnsns.2018.02.039. URL https://doi.org/10.1016/j.cnsns.2018.02.039.

N. Pareek, V. Patidar, and K. Sud. Image encryption using chaotic logistic map. *Image and Vision Computing*, 24(9):926–934, Sept. 2006. doi: 10.1016/j.imavis.2006.02.021. URL https://doi.org/10.1016/j.imavis.2006.02.021.

A. Perez-Resa, M. Garcia-Bosque, C. Sanchez-Azqueta, and S. Celma. Chaotic encryption applied to optical Ethernet in industrial control systems. *IEEE Transactions on Instrumentation and Measurement*, 68(12):4876–4886, dec 2019. doi: 10.1109/tim.2019.2896550. URL https://doi.org/10.1109/tim.2019.2896550.

S. C. Phatak and S. S. Rao. Logistic map: A possible random-number generator. *Physical Review E*, 51(4):3670–3678, Apr. 1995. doi: 10.1103/physreve.51.3670. URL https://doi.org/10.1103/physreve.51.3670.

A. Pisarchik and M. Zanin. Image encryption with chaotically coupled chaotic maps. *Physica D: Nonlinear Phenomena*, 237(20):2638–2648, Oct. 2008. doi: 10.1016/j.physd.2008.03.049. URL https://doi.org/10.1016/j.physd.2008.03.049.

G. S. Rani, S. Jayan, and B. Alatas. Analysis of chaotic maps for global optimization and a hybrid chaotic pattern search algorithm for optimizing the reliability of a bank. *IEEE Access*, 11:24497–24510, 2023. doi: 10.1109/access.2023.3253512. URL https://doi.org/10.1109/access.2023.3253512.

G. Robinson. Statistics, overview. In *International Encyclopedia of Human Geography*, pages 436–451. Elsevier, 2009. doi: 10.1016/b978-008044910-4.00537-x. URL https://doi.org/10.1016/b978-008044910-4.00537-x.

M. Rüdisüli, T. Schildhauer, S. Biollaz, and J. V. Ommen. Measurement, monitoring and control of fluidized bed combustion and gasification. In *Fluidized Bed Technologies for Near-Zero Emission Combustion and Gasification*, pages 813–864. Elsevier, 2013.

M. Saber and M. M. Eid. Low power pseudo-random number generator based on lemniscate chaotic map. *International Journal of Electrical and Computer Engineering (IJECE)*, 11(1):863, feb 2021. doi: 10.11591/ijece.v11i1.pp863-871. URL https://doi.org/10.11591/ijece.v11i1.pp863-871.

M. Sharma and A. Bhargava. Chaos based image encryption using two step iterated logistic map. In *2016 International Conference on Recent Advances and Innovations in Engineering (ICRAIE)*. IEEE, Dec. 2016. doi: 10.1109/icraie.2016.7939535. URL https://doi.org/10.1109/icraie.2016.7939535.

SoftPosit. Cerlane Leong / SoftPosit · GitLab — gitlab.com. https://gitlab.com/cerlane/SoftPosit. [Accessed 21-Jan-2023].

K. Stroethoff. Bhaskara's approximation for the sine. *The Mathematics Enthusiast*, 11(3): 485–494, Dec. 2014. doi: 10.54870/1551-3440.1313. URL https://doi.org/10.54870/1551-3440.1313.

M. Sugeno. Industrial applications of fuzzy control. *Supercomputing Frontiers and Innovations Amsterdam*, 1985.

J. S. Teh, K. Tan, and M. Alawida. A chaos-based keyed hash function based on fixed point representation. *Cluster Computing*, 22(2):649–660, nov 2018. doi: 10.1007/s10586-018-2870-z. URL https://doi.org/10.1007/s10586-018-2870-z.

J. Walker. Pseudorandom number sequence test program. https://www.fourmilab.ch/random/, 2008. (Accessed on 01/28/2023).

T. Yoshida, H. Mori, and H. Shigematsu. Analytic study of chaos of the tent map: Band structures, power spectra, and critical behaviors. *Journal of Statistical Physics*, 31(2):279–308, may 1983. doi: 10.1007/bf01011583. URL https://doi.org/10.1007/bf01011583.

H. Zhang and S.-B. Ko. Efficient multiple-precision posit multiplier. In *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, May 2021. doi: 10.1109/iscas51556.2021.9401213. URL https://doi.org/10.1109/iscas51556.2021.9401213.

J. Zheng and T. Bao. An image encryption algorithm using cascade chaotic map and s-box. *Entropy*, 24(12):1827, Dec. 2022. doi: 10.3390/e24121827. URL https://doi.org/10.3390/e24121827.