Design and Implementation of Chaos-based Random Number Generators for IoT Platforms

by

Ngoc NGUYEN THI THU

MANUSCRIPT-BASED THESIS PRESENTED TO ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
IN PARTIAL FULFILLMENT FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
Ph.D.

MONTREAL, 03 FEBRUARY 2022

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

BOARD OF EXAMINERS

THIS THESIS HAS BEEN EVALUATED

BY THE FOLLOWING BOARD OF EXAMINERS

M. Georges Kaddoum, Thesis Supervisor
Departement of Electrical Engineering, École de technologie supérieure

M. Pascal Giard, Thesis Co-Supervisor
Departement of Electrical Engineering, École de technologie supérieure

M. Chamseddine Talhi, Chair, Board of Examiners
Department of Software Engineering and Information Technologies, École de technologie supérieure

M. Ghyslain Gagnon, Member of the Jury
Departement of Electrical Engineering, École de technologie supérieure

M. Christophe Guyeux, External Examiner
Femto-ST Institute, Université de Bourgogne Franche-Comté, France

THIS THESIS WAS PRESENTED AND DEFENDED

IN THE PRESENCE OF A BOARD OF EXAMINERS AND THE PUBLIC

ON 20 JANUARY 2022

AT ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

FOREWORD

This dissertation is mainly based on research outcomes, which have been accomplished under the supervision of Dr. Georges Kaddoum and Dr. Pascal Giard between May 2017 and May 2021. This work has been partly supported by the FRQNT and NSERC Ph.D. fellowships, and the Schlumberger Foundation for the Faculty for the Future (FFTF). This dissertation is subjective to address the hardware design of random number generator for multiple platforms. Resultantly, my Ph.D. study successfully ended with 3 journal papers published, and 1 journal paper submitted as the first author, and 2 conference papers. Apart from the first two chapters, where the background of random number generators is intensively introduced, the remaining chapters are based on my journal papers. For those chapters, I did a comprehensive literature review, proposed a chaotic system design, mathematically analyzed and simulated the dynamical and chaotic characteristics. Then, the hardware design for the proposed system was provided. After the presentation of those chapters, chapter 7 concludes the thesis and lists several future research directions.

ACKNOWLEDGEMENTS

First and foremost, I would like to gratefully acknowledge and express my sincere thanks to my supervisor Dr. Georges Kaddoum for his considerate guidance, valuable inspiration, constructive suggestions, and consistent encouragement throughout my four years research study. This thesis would not have come to the completion without his dedicated mentorship and scholarly inputs.

Besides my supervisor, I am also appreciative to Dr. Pascal Giard, who is my co-supervisor, for his immense knowledge, technical meetings and discussions. Similar profound gratitude also goes to Dr. François Gagnon, who is my formal co-supervisor, for his encouragement and guidance from the beginning of my Ph.D. journey. I also gratefully thank Professor Chamsedine Talhi, Professor Ghyslain Gagnon, and Professor Christophe Guyeux, for their agreement to serve as my jury members. Each of the members of my Dissertation Committee has provided me extensive personal and professional guidance and taught me a great deal about both scientific research and life in general. Also, a special mention and thanks goes to the Schlumberger Foundation for their financial support during my Ph.D. study.

I am grateful to all of those with whom I have had the pleasure to work during this and other related projects. Many thanks also goes to my friends for their help and encouragement to embrace my research problems with high enthusiasm. I do hereby acknowledge all my colleagues from LACIME group, including Vu, Dat, Khaled, Bassant, Victor, Dawa, Zeeshan, Elli, Hamza, Long, Jung, Michael, Marwan, etc., and my ETS friends including Sahar, Nakisha, Benaf She, Nam, Ha, Dung, Dai.

Nobody has been more patient in the pursuit of this journey than the members of my family. I would like to thank my loving and supportive husband, Toan Bui who always provides me encouragement, unconditional support, and love, and my two beautiful daughters Thao Nguyen and Alyssa, who sacrificed a lot and provided unending inspiration.

Last but not the least, I would like to wholeheartedly thank my mother and my mother-in-law for their continued patience, unconditional support, and warm love. They are sacrifying their freedom and giving up the life of ease to take care of my daughters. I owe them an appology since I gave them a burden instead of being beside and taking care of them during the hardest time of life. I feel sorry for my departed father and father-in-law who cannot share this moment with us, although I believe they would have been proud of me and given me their total support. Also, my special thanks and appreciation goes to my brothers and sisters for their spiritually invaluable love and support to my studies and life.

Conception et mise en œuvre de générateurs de nombres aléatoires basés sur le chaos pour plates-formes IoT

Ngoc NGUYEN THI THU

RÉSUMÉ

Pour les communications futures, la sécurité est l'un des principaux défis. Dans ce contexte, le générateur de nombres aléatoires, qui est chargé de générer des clés publiques, des clés privées et d'autres types de nombres aléatoires, est un moteur essentiel des algorithmes cryptographiques et du secret des données. Les algorithmes cryptographiques nécessitent des capacités de calcul élevées et une puissance élevée, ce qui est un défi dans les appareils limité en ressources. Par conséquent, le générateur de nombres aléatoires matériel, qui fournit un débit élevé à faible puissance, est un composant essentiel pour les futurs appareils. De nombreuses approches ont été développées pour améliorer le caractère aléatoire et la sécurité des générateurs de nombres aléatoires utilisés sur des dispositifs limité en ressources. Cependant, les générateurs de nombres aléatoires matériel actuels ont été confrontés à des défis de conception, tels que l'efficacité énergétique (qui est l'énergie consommée pour générer un seul bit aléatoire), la sécurité et la flexibilité. Pour relever ces défis, des générateurs de nombres aléatoires basés sur le chaos sont apparus dans lesquels les systèmes chaotiques non linéaires jouent un rôle clé.

Pour les générateurs de nombres aléatoires basés sur le chaos, les caractéristiques chaotiques et dynamiques sont cruciales. Plus la complexité et la dynamique du système chaotique sont élevées, plus le caractère aléatoire des bits de sortie est élevé; cependant, le caractère aléatoire des bits de sortie dépend également d'autres facteurs, tels que l'implémentation et le post-traitement des données. Récemment, la communauté des chercheurs s'est particulièrement intéressée aux systèmes chaotiques continus. Par conséquent, cette recherche se concentre sur le développement de nouveaux systèmes chaotiques continus, hautement dynamiques et hautement dimensionnels des données, afin d'améliorer la sécurité et le débit des générateurs de nombres aléatoires proposés.

Les contributions de cette thèse sont quadruples: (i) le développement de systèmes chaotiques robustes et l'analyse des caractéristiques dynamiques pour trouver les meilleurs paramètres (ii) l'implémentation matérielle en tenant compte des applications cibles et des plates-formes d'appareils; (iii) la proposition d'applications techniques utilisant les générateurs de nombres aléatoires proposés; (iv) la conception et la fabrication de véritables générateurs de nombres aléatoires et de générateurs de nombres pseudo-aléatoires pour fournir des dispositifs de sécurité prêts à l'emploi pouvant être utilisés comme produits commerciaux.

Tout d'abord, nous développons plusieurs systèmes chaotiques qui se concentrent sur l'extension des dimensions (ce qui peut améliorer le débit global des générateurs de nombres aléatoires), et sur l'amélioration du niveau de sensibilité du système en masquant les informations des points d'équilibre et en faisant dépendre les caractéristiques du système des conditions initiales difficiles à prévoir. L'analyse mathématique montre les avantages des systèmes chaotiques proposés par rapport aux systèmes précédents. Dans la mise en œuvre matérielle, la consommation d'énergie, les ressources de l'appareil et le niveau de sécurité sont des compromis. Par conséquent, les systèmes chaotiques avec une conception matérielle peu complexe sont prioritaires.

En ce qui concerne la mise en œuvre matérielle, nous présentons deux stratégies différentes pour les véritables générateurs de nombres aléatoires et les générateurs de nombres pseudo-aléatoires. La conception de circuits analogiques est utilisée pour mettre en œuvre un véritable générateur de nombres aléatoires, dans lequel le bruit du circuit et l'imperfection du circuit affectent les caractéristiques chaotiques. Les générateurs de nombres pseudo-aléatoires sont implémentés dans des dispositifs FPGA qui peuvent être intégrés dans plusieurs plates-formes matérielles.

Avec les plates-formes IoT, les développeurs peuvent créer une large gamme d'applications spécifiquement à des fins IoT, telles que la surveillance et la surveillance en temps réel, dans lesquelles le secret des données est important. Pour prouver les avantages de l'utilisation des générateurs de nombres aléatoires désignés dans les applications, nous fournissons un cryptage / décryptage d'image à l'aide d'un encodage de données de pad à usage unique. Par conséquent, un cryptosystème de pas à usage unique basé sur le chaos est développé, les données (images) sont codées à l'aide du générateur de nombres aléatoires basé sur le chaos proposé. Le récepteur et l'émetteur partagent des informations sur l'état initial du système chaotique pour récupérer la clé et décoder les données.

Enfin, après près de quatre ans de recherche sur les générateurs de nombres aléatoires avec les réalisations que nous avons obtenues jusqu'à présent, il est possible de commencer à commercialiser notre projet de recherche. Reconnaissant la demande croissante de sécurité personnelle, notre produit vise à donner aux utilisateurs le contrôle le plus élevé sur leurs données en leur fournissant le générateur de clés. Par conséquent, les données stockées ne peuvent pas être décodées même si l'hôte (où les données sont stockées / réservées) est attaqué. De plus, notre produit vise à fournir des solutions d'encodage / décodage de données à haute vitesse pour des applications en temps réel telles que la diffusion en continu de vidéo, en particulier pendant la quarantaine de la pandémie mondiale COVID-19.

Mots-clés: générateur de nombres aléatoires, système chaotique, carte chaotique, hyperchaos, système chaotique continu, sécurité, cryptographie, prochaine génération de réseau.

Design and Implementation of Chaos-based Random Number Generators for IoT Platforms

Ngoc NGUYEN THI THU

ABSTRACT

For future network communication, security is one of the main challenges. In this context, the random number generator, which is responsible for generating public keys, private keys, and other kinds of random numbers, is a critical engine in cryptographic algorithms and data secrecy. Cryptographic algorithms require high computational capabilities and high power, which can be challenging in hardware-constrained devices. Therefore, the hardware-based random number generator, which provides high throughput at low power, is an essential component for future devices. Many approaches have been developed to enhance the randomness and security of random number generators used in constrained devices. However, current hardware-based random number generators have been facing design challenges, such as energy efficiency (which is the energy consumed to generate one single random bit), security, and flexibility. To address these challenges, chaos-based random number generators, in which non-linear chaotic systems are playing a key role, have emerged .

For chaos-based random number generators, the chaotic and dynamical characteristics are crucial. The higher the complexity and dynamics of the chaotic system, the higher the randomness of the output bits; however, the randomness of the output bits also depends on other factors, such as the implementation and data post-processing. Recently, considerable attention has been drawn by the research community to continuous chaotic systems. Therefore, this research focuses on developing novel continuous chaotic systems, which are highly dynamic and highly data dimensional, to improve the security and throughput of the proposed random number generators.

The contributions of this dissertation are fourfold: (i) development of robust chaotic systems and dynamic characteristics analysis to find the best parameter set (ii) hardware implementation in consideration of the target applications and device platforms; (iii) proposition of engineering applications using the proposed random number generators; and (iv) design and fabrication of true random number generators and pseudo-random number generators to provide ready-to-use security devices that can be used as commercial products.

First of all, we develop several chaotic systems which focus on extending dimensions (which can improve the overall throughput of random number generators), improving system sensitivity level by hiding information of equilibrium points and making the system's characteristics depend on the initial conditions which are difficult to predict. Mathematical analysis shows the advantages of the proposed chaotic systems compared to previous systems. In hardware implementation, the power consumption, device resources, and security level are tradeoffs. In chaotic system, non-linear function is the main component. Therefore, the non-linear functions, which have low complexity hardware design, are prioritized.

Regarding hardware implementation, we present two different strategies for true random number generators and pseudo-random number generators. Analog circuit design is employed to implement true random number generator, in which the circuit noise and circuit imperfection affect the chaotic characteristics. The pseudo-random number generators are implemented in FPGA devices which can be integrated into multiple hardware platforms.

With IoT platforms, developers can build a wide range of applications, such as real-time monitoring, surveillance, in which data secrecy matters. To prove the benefits of using the designated random number generators in these applications, we provide image encryption/decryption using one-time pad data encoding. Therefore, a chaos-based one-time pad cryptosystem is developed, the data (images) is encoded using the proposed chaos-based random number generator. The receiver and transmitter share information on the initial condition of the chaotic system to recover the key and decode the data.

Finally, after nearly four years of research on random number generators with the achievements we have been getting so far, it is possible to start commercializing our research project. Acknowledging the increasing demand for personal security, our product aims to give users the highest control over their data by providing them with the key generator. Therefore, stored data cannot be decoded even if the host (where the data is stored/reserved) is attacked. Moreover, our product aims to provide high-speed data encoding/decoding solutions for real-time applications such as video streaming, especially during the quarantine of the worldwide pandemic COVID-19.

Keywords: random number generator, chaotic system, chaotic map, hyperchaos, continuous chaotic system, security, cryptography, next generation network.

TABLE OF CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

LIST OF ABBREVIATIONS

GSM	Global System for Mobile communications
SIM	Subscriber Identification Module
RNG	Random Number Generator
TRNG	True-Random Number Generator
PRNG	Pseudo-Random Number Generator
FFRK	Fourth-Folding Runge-Kutta
LPWA	Low-Power Wide Area
IoT	Internet-of-Thing
PMOD	Peripherial module interface
JTAG	the Joint Test Action Group
DSP	Digital Signal Processing
LFSR	Linear-Feedback Shift Register
DRC	Design Rule Checks
LVS	Layout Versus Schematic
ESD	Electrostatic discharge
I^2C	Inter-Integrated Circuit Protocol
USB	Universal Serial Bus
UART	Universal asynchronous receiver-transmitter
TSMC	Taiwan Semiconductor Manufacturing Company

PDK	Process Design Kit
PCB	Printed-Circuit Board
LUT	Look-up Table
FF	Flip-flop
TP	Throughput

INTRODUCTION

The downscale of semiconductor technology has been opening an era of the Internet of Things (IoT), causing an unprecedented impact on society. IoT services are likely to be a key driver for future growth in cellular networks. According to the report of Machina Research in May 2015, 30 billion devices are expected to be deployed by 2025, of which cellular IoT and Low-Power Wide-Area (LPWA) modules are forecasted to account for 7 billion units (Nokia (2017a)). IoT networks, which allow people to connect with multiple devices, where some of them contain private information, such as heartbeat records and credit card information, are subjects to continuous attacks. Therefore, among several significant obstacles remaining to fulfill the IoT vision, security is the chief. Although $550M have been spent on IoT security, it was reported that more than 30% attacks involve IoT devices by March 2020(Bonvarlet (2017)). Researcher from Unit 42, a threat intelligence team at Palo Alto Networks, also discovered 98% of IoT devices are unencrypted, exposing personal and confidential data on the network, which are vulnerable to cybersecurity attacks UNIT42 (2020).

25 years ago, when GSM was developed and standardized, the first security functions were introduced partly because of emerging threats. The invention of a tamper-resistant subscriber identification module (SIM) provided strong authentication of the subscriber and strong binding to fraud (Nokia (2017b)). Moving to 3G and 4G standards, additional improvements were made including mutual authentication and data encryption at the base station. Moreover, true key management was employed to protect against potential physical attacks.

Reflecting on the development of 2G-4G security, it can be said that security was developed to protect a basic connectivity service (voice and packet data) (Roman, Najera & Lopez (2011)). The drivers for next generation networks are continuously being developed to provide trustworthy basic connectivity services. However, additional key driving factors will be considered because the upcoming 5G networks are not only designed to serve mobile users but also connecting industries such as manufacturing, transport, smart grids, healthcare, etc.

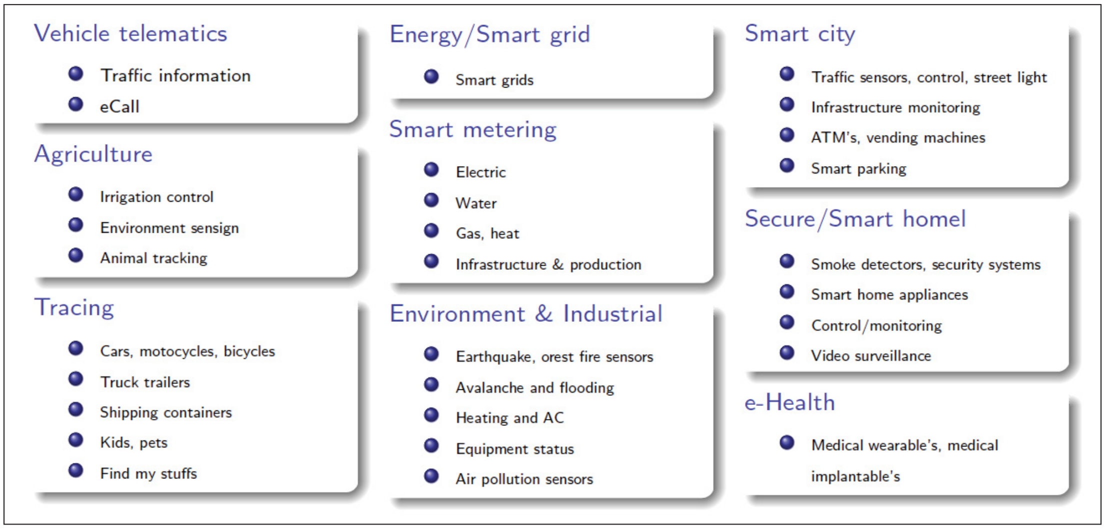


Figure 0.1 Cellular IoT use cases

Figure 0.1 shows multiple applications and IoT connections that will be required in future networks. IoT applications and services vary widely based on their service requirements, data throughput, latency, connectivity, and reliability. Security is required at many levels in communications, from the physical layers to cloud services; data needs to be protected for safe operations. Most of the existing security mechanisms are based on software implementation, which has limitations when it comes to providing high-level security in cryptographic applications deployed in low-power consumption devices (Bae, Kim, Park & Kim (2017)). Software-based cryptographic applications require high device resources which make them not applicable to embedded systems in low power consumption devices.

In software implementations, linear methods have been used to generate pseudo-random bits such as LFSRs, which are very popular due to their efficiency, easy embedding, and programming. However, being deterministic and periodic, their key streams are predictable and reproducible.

In this context, hardware-based RNGs using non-linear methods have been emerging as a natural choice for high-security applications.

Research motivation

In this section, we address two main factors motivating our research toward designing new RNGs. First, the high security requirement of IoT devices is presented. Second, we demonstrate that a hardware implementation of RNGs is required for future networks.

- **Information security**

  Figure 0.2 shows the data flow from IoT devices to the central cloud which allows devices to connect to the Internet. The IoT security framework is composed of three elements: the IoT secure device, the IoT cloud service providing device and security management, and the IoT central cloud. In ubiquitous network environments, every physical device has its device identification, which is locatable, addressable, and readable/writable, on the Internet. This device identification produces and consumes services and collaborates with each other toward a common goal.

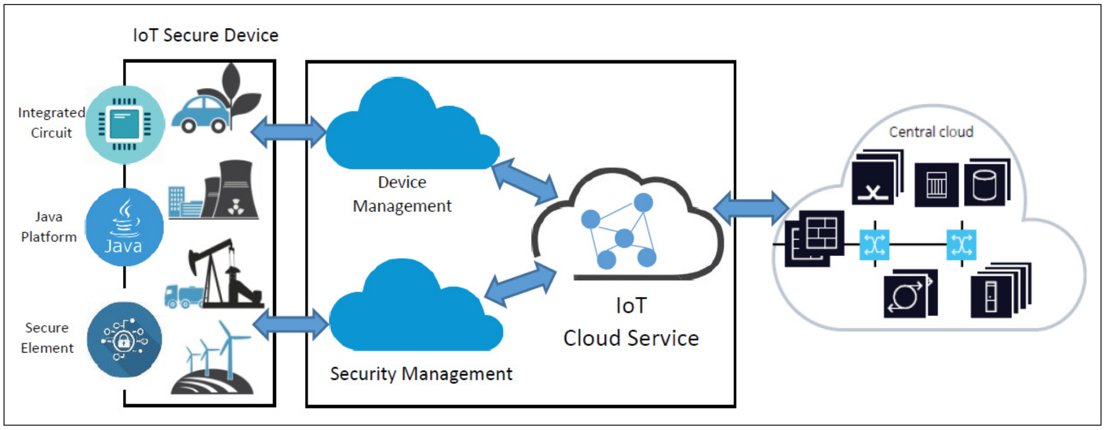


Figure 0.2 Elements of IoT security architecture

The first element, IoT secure device is a microprocessor that includes a secure element and a high-level application platform within an integrated circuit. This device operates in real-time and supports standard IoT protocols enabling it to be connected to the IoT cloud service (IoT

CS). The IoT device's ID in the IoT network is a feature that distinguishes the IoT devices within the Internet. The IoT devices collect data and monitor the system state through the sensors. Then, the data travels through IoT network protocols to the IoT CS. Some of the existing protocols that are used and developed currently by different companies or groups are listed in Table 0.1. In data cryptography, random number generator provides random keys such as private key, public key, authenticate key for any kind of cryptographic operations. The random number generators in the IoT secure devices should provide random key bitstreams at proper throughputs to be compatible with the network protocols they are serving. The designs for the RNG at the device side have to satisfy the limited power consumption and device size requirements. The second element is the IoT CS which provides device and security management. In secured communications, devices need the provision the device's IDs and the activation keys. The device manager manages the device's ID and enables the IoT CS to recognize authorized devices and prevent unauthorized devices from attacking its services. The security management module provides an activation key to establish a secure channel between devices and services. Finally, the IoT CS connects to the central cloud which provides many applications for users. The IoT CS employs trusted key management providing high-security keys and high throughput to support a large number of devices. The throughput and security requirements of the random number generators at the IoT CS are more critical than that of the IoT devices. The last element is the IoT cloud center (IoT CC) which provides multiple services for users. The connection between the IoT CS and IoT CC is standardized. It is out of the scope of this report.

There are many attacks that can occur during the data transfer process, such as interference, shielding, theft, and side-channel attacks, which cause data leakage and other security issues if insufficient data protection is adopted. As such, any IoT design that places a high priority on security needs a secure element to encrypt or decrypt data and provide an integrity checking mechanism to ensure information security.

Table 0.1 List of existing network protocols

Protocol	Standard	Frequency (Hz)	Range	Data rate (bps)
Bluetooth	Bluetooth 4.2	2.4 G	50-100 m	1 M
ZigBee	ZigBee 3.0	2.4 G	10-100 m	250 k
Z-wave	Z-wave Alliance	900 M	30 m	< 10-100 k
6LowLan	RFC6282	2.4G / <1G	N/A	N/A
Thread	Thread	2.4 G	N/A	N/A
Wifi	802.11n	2.4 G / 5 G	50 m	600 M-1 G
Cellular	2G,3G,4G,LTE	900/1800/8900/2100 M	35 Km	10 M
NFC	IOS/IEC	13.56 M	10 cm	100-420 k
Neul	Neul	900,458,470-490 M	10 Km	< 100 k
SigFox	SigFox	900 M	< 50 Km	10 bps-1 k
LoRaWan	LoRaWan	Various	10 Km	0.3-50 k

- **Hardware design requirement**

  As the key engine of the secure element, the random number generator plays an important role in information security applications. With the development of network information technologies, especially the booming of IoT devices, the random number generator's requirements are no longer limited to soft computing. The traditional random number generator in a secure element is implemented in the instruction set architecture processor (ISAP) that uses a software implementation mode, for example 32-bit ARM Cortex-M4F processors in the Tiva^{TM} C Series from Texas Instrument (TI). However, the generic form of the instructions facilitates the modeling of physical attacks. As a result, ISAP-based RNGs' resistance to attacks is generally lower than hardware-based RNGs. Hence, ISAP-based devices are not applicable to scenarios where high security is required.

  Physical attacks can be divided into invasive, semi-invasive, and non-invasive attacks according to the way the attacker acts on the circuits. Invasive attacks damage the package and remove the passivation layer of the chip. Here, attackers obtain the circuit function or the key information by imaging the circuit structure. These attacks are very powerful and cause permanent and irreversible damage to the targets. However, they require a long time and high cost. Semi-invasive attacks need to only remove the package of the chip only. Non-invasive attacks do not damage the original circuit or its package. They usually attack by analyzing information such as power consumption and electromagnetic radiation which are leaked during the chip's operation (Liu, Wang & Wei (2018)). The software-based random generator, which is a deterministic process in software implementation, such as xorshift, linear congruential generator, is non-cryptographically generator. Additionally, some cryptographically software-based PRNGs, such as SHA256PRNG, Fortuna, consume high power (for example $0.5 \mu J$ for each integer). Compared to hardware implementation; they have drawback of either low security level or high power consumption. Therefore, the hardware-based RNGs are in demand for IoT devices. Figure 0.3 shows the typical architecture of an IoT secure device that performs several cryptographic algorithms for secured transactions. The true random number generator, which is implemented in hardware design, is a core engine of the secure element.

- **Challenges of random number generator design**

  There has been numerous approaches to design random number generators in hardware. The random number generator in a secure element should be designed considering a trade-off between four factors: energy efficiency, security, reliability, and flexibility. We need to consider the balance between these factors in the design of a random number generator.

  - **Energy efficiency**

    The power consumed to generate a single random bit is denoted as energy efficiency, which is evaluated by dividing the power consumption by the throughput. It is one of the most important design metrics to evaluate the system design. Specific applications and IoT communication protocols require different throughputs of random bit generators. For example, applications using real-time secure transmission through Zigbee protocol

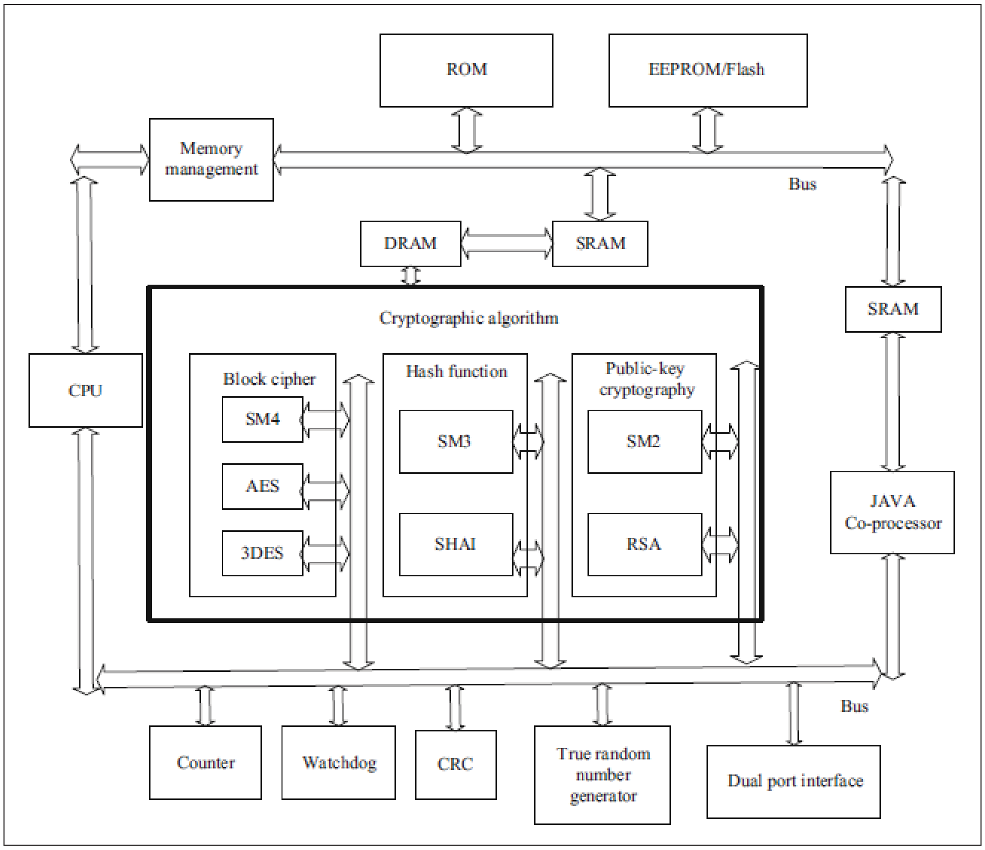


Figure 0.3 Block diagram of secure element for secured transactions
Taken from Liu et al. (2018, p.5)

require random bits at a throughput of at least 100 kbps, while the throughput requirement for 5G cellular communications is 100 Mbps. In stream cipher, which provides higher security of data encryption than the block cipher, requires high security, unpredictable random bitstream at the same speed as the plaintext stream. Therefore, the random number generators in the cipher stream need to meet the network's requirements in terms of data rate or throughput. Software implementations of RNGs are associated with a low energy efficiency due to their highly computational requirements and the fact that they are not optimized for specific applications. Hardware designs are necessary to provide high speed and high efficiency.

- **Security**

  Random number generators for high-security communications should provide high resistance to physical attacks. Most of the existing works on RNG designs do not analyze their security ability. The invasive attack resistance could be improved by masking the circuit with many layers of metal shields and sensors on top of the chip. Non-invasive attacks should be considered while designing TRNG circuits. Thermal noise-based TRNGs are affected by supply noise injection. Attackers can use the side-channel information to change the ground potential. As a result, the randomness of the signal generated from this circuit could be destroyed. Multiple ring oscillator-frequencies were added to supply noise, which locked the oscillation with smaller jitter and greatly reduced the entropy harvested (Yang *et al.* (2014)). The metastability-based method using cross-coupled inverters provides excellent operating frequency and power efficiency. However, this approach is sensitive to environmental variations that degrade the robustness against attacks. The resistance to physical attacks is a key element for RNG designs in IoT devices due to two main reasons.

  - The environment conditions vary during the chip's operations; the circuit should be verified in a wide range of working conditions. All the circuit components are affected by the voltage supply variation. TRNGs, which are based on the circuit noise and metastability, are highly affected by voltage variations, device mismatches and working temperature.
  - Recently, many powerful non-invasive attacks have emerged. A low-frequency attack method that is based on sound or electromagnetic wave has been successful in jeopardizing many secure chips. This method uses specific MHz or kHz signals leaked by the attack target, even if the circuit works at the GHz range Liu *et al.* (2018). Therefore, using high sampling operations to resist the low-frequency noise attacks is

no longer valid Bae *et al.* (2017). Therefore, the secure devices have to be designed considering novel attack methodologies and potential future attacks.

Therefore, the designated RNGs have to pass the randomness tests for cryptography purposes such as standard statistical tests.

- **Reliability**

  There are many RNG designs that rely on physical entropy sources which can be based on different effects. Some researchers proposed to harvest thermal noise of a resistor in an analog circuit. Jitter noise in a ring oscillator can be added as a random source to fully digital circuits. Entropy sources, such as metastability state or jitter measurement, can be implemented in digital primitives or digital standard cells. Due to the high speed and adaptability to different technology processes, they were recently widely developed. However, these solutions are highly sensitive to systematic noises. Additionally, these entropy sources are assumed to have a Gaussian distribution, but in reality, they experienced unknown statistical distributions due to their limited dynamic range. This is caused by external factors, such as supply and temperature variations. To make them suitable for high-security applications, more reliable designs are necessary.

- **Flexibility**

  There are many IoT communication protocols for multiple purposes, but designing an RNG compatible with all these applications is neither practical nor reliable. The current RNG methods are optimized for a specific application. However, an IoT secure device is mainly used to process various tasks as encrypt/decrypt data or real-time authentication which have different requirements to RNGs. It is better to integrate multiple operation modes in a processor.

Research objectives

RNGs are required in many applications, such as wireless networking, gaming, military communications, online payments, etc. RNGs are used to generate keys, initialize vectors, and other random numbers used in many security standards and applications. As an example, the IoT is a fast-growing market where data can easily be intercepted and devices can be hacked, especially if weak RNGs are selected. Our main long-term objective is to design robust low-cost RNGs based on chaos systems for secure communication applications. Thus, we aim to achieve the following specific objectives:

- Robust RNG designs: Based on our investigation, the physical entropy sources like thermal noise, jitter noise, and metastability have low reliability. Therefore, we aim to design robust RNGs using chaotic systems. Chaos-based RNGs have been developed in many works. However, the existing chaos-based solutions are limited by their large occupation areas, high power consumption, or their use of 1D piecewise chaotic maps which degrade the dynamic properties. Our robust chaos-based RNG designs are expected to overcome these limitations.
- To deploy the designated RNGs in different IoT platforms, we will optimize the proposed approaches to meet the requirements in data rate, operating frequency, power consumption, etc. Different optimization strategies are investigated and performed. For example, the device resource optimization would be applied for constrained devices, with less consideration of data throughput. We can also optimize the proposed RNGs in which security and data rate are priorities.

Contributions and Outline

The organization of this dissertation, which includes 6 chapters, is structured and detailed as follows. Chapter 1 presents the background knowledge in chaos theory and random number generators, and the comprehensive literature review of conventional RNGs and chaos-based RNGs. In this vein, a discussion on the trends of future research in RNGs is presented.

Chapter 2 presents the research methodology for both the TRNG and PRNG designs, which are used in this project from system development to design. Although this is a research-based project, the methodology adopted the scheme and processes of high-tech commercial product development. However, for industrial products, more product constrains could be applied during design and verification process. For example, industrial products should be verified in a wider range of temperature, voltage variations.

Chapter 3 presents the first article on designing a TRNG based on a novel 3D continuous chaotic system and an application of the designated chaos-based RNG in data encryption. In this chapter, a novel 3D continuous chaotic system is presented and analyzed mathematically in terms of dynamic characteristics, stability, and chaotic trajectories. The chaotic system and data post-processing circuit are implemented in a 130 nm CMOS technology with a low voltage supply of 1.2 V. The system design enables the utilization of multiple chaotic signals to generate random bits. The core of the chaotic circuit consumes $630\mu W$ in static mode and a maximum of $660\mu W$ in running mode which is lower than previous continuous chaos-based TRNGs. Moreover, the random bits are successfully tested by the NIST test suite and other kinds of randomness tests, such as entropy and correlation tests. Then, the generated random bits are used for image encryption.

Chapter 4 presents the second article discussing another TRNG based on a hyperchaotic system. The novel hyperchaotic system has four dimensions that provide higher throughputs for the RNG. The advantages of the proposed hyperchaotic system in generating random bits are discussed and compared to the state-of-the-art. The analog circuit design of the proposed 4D hyperchaotic system and the data post-processing provides truly random bits. Due to analog circuit imperfections and circuit noises at the initial conditions, the unpredictable trajectories of chaotics signals provide high randomness for the output bits. The circuit is implemented and fabricated using 130 nm CMOS technology. The proposed novel system design and its circuit implementation provide the best energy efficiency of 4.37 pJ/b at a maximum sampling frequency of 100 MHz which is much better than previous chaos-based TRNGs.

Chapter 5 presents the third article proposing a novel 5D hyperchaotic system that provides improved dynamic characteristics. Moreover, a digital implementation of the proposed 5D chaotic system and data post-processing provides a high throughput PRNG. A novel method to implement the fourth-order Runge-Kutta algorithm for chaotic systems in FPGAs which reduces the implementation cost compared to traditional implementations, is provided. The generated random bits are tested with the NIST test suite and the excessive random test TestU01. The proposed random bit generator can achieve a maximum throughput of 6.78 Gbps which is higher in compared to state-of-the-art designs.

Chapter 6 presents the fourth article proposing a novel 4D hyperchaotic system that provides high dynamic characteristics. Digital implementations of the chaotic system and data post-processing are performed using the Xilinx System Generator, which reduces the time-to-market. The advantage of the proposed 4D hyperchaotic system is that the non-linear function is implemented in a simple manner without multiplexers. Thus, the implementation cost is reduced while the high dimensional output signal provides a high throughput bitstream at a maximum of 8.56 Gbps which is higher than previous continous-based PRNGs's.

Chapter 7 provides the conclusion of the dissertation and recommendations for future works.

For convenience, a big picture of the thesis is provided in Figure 0.4.

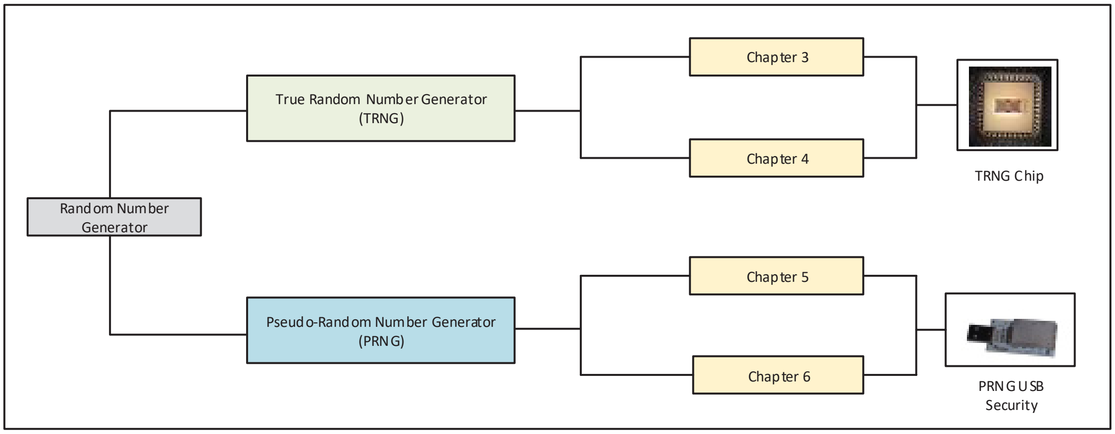


Figure 0.4   Paradigm of the thesis's contributions

CHAPTER 1

BACKGROUND AND LITERATURE REVIEW

As mentioned in the Introduction, security is required at many layers of a communication system: device, network, and cloud to ensure efficient and safe operations by protecting data in motion and at rest. In this manner, the RNG is the key engine of any security method. In this chapter, a review of conventional RNGs is presented. Recent research on chaos-based RNGs and the prerequisite background in chaos theory are provided. Lastly, randomness evaluation metrics are presented.

1.1 Architecture of RNGs

1.1.1 Architecture of conventional RNGs

As previously stated, this research focuses on the hardware implementation of RNGs. The conceptual architectures of conventional hardware PRNG and TRNG are presented in Figure 1.1. The conventional hardware PRNG architecture is similar to that of software-based PRNG; however, it is implemented in a hardware device such as FPGA Addabbo *et al.* (2007). The periodicity and deterministic algorithms used in the transition functions are the main engines of PRNGs Hazwani *et al.* (2014). The initial state $x(0)$ is the seed. The next state is determined by applying the transition function $x(t+1) = f(x(t))$. The function f is an arithmetic linear or non-linear function in PRNGs and a physical phenomenon in TRNG Kim, Ha & Lee (2017b). Then, the output function $g(x(t))$ is applied to the current state. This process, presented in Figure 1.1-b), is considered as the data post-processing in TRNG.

1.1.1.1 Physical entropy source

In the last few decades, various RNGs that can employ physical sources, such as thermal noise, jitter noise, telegraph noise, and metastability, have been proposed.

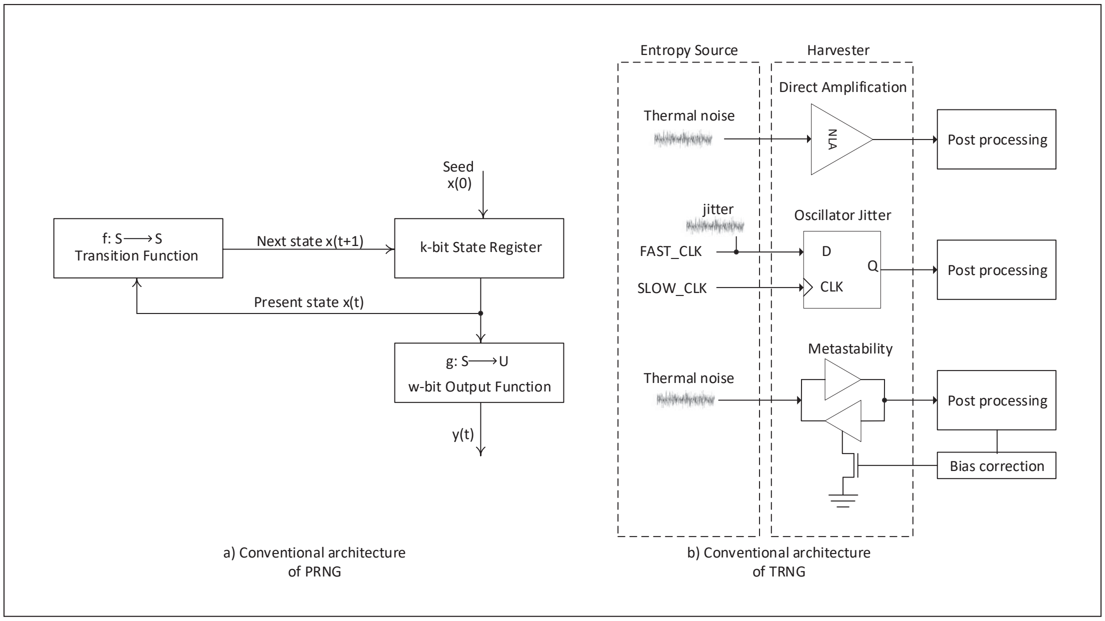


Figure 1.1 Conventional architecture for the hardware implementation of a) PRNG and b) TRNG
Taken from Kim et al. (2017, p.2)

- **Thermal noise:** Thermal circuit noise, which is very easy to harvest, is utilized in some TRNGs. However, the Flicker noise ($1/f$ noise) affects the quality of the randomness at low frequency operation. Moreover, the relative low noise power available needs to be amplified before harvesting, which increases the circuit bias. The thermal noise power harvested from a resistor R at absolute temperature T is extracted as:

$$P_{AN} = kT\Delta f \rightarrow (\frac{v_n^2}{2})\frac{1}{R} = kT\Delta f, \tag{1.1}$$

$$\rightarrow \overline{v_n^2} = 4kTR\Delta f. \tag{1.2}$$

The voltage spectral density derived from the above equations is:

$$S_v(f) = \frac{\overline{v_n^2}}{\Delta f} = 4kTR \quad (V^2/Hz), \tag{1.3}$$

where Δf is the bandwidth of the signal. It is noticed that the power of the noise extracted from a resistor fabricated in CMOS technology is insufficient for direct use, it should be amplified before post-processing. However, amplifiers suffer from other deterministic noise sources, such as power variations and process variations, that degrade the randomness by varying the bias points. Some researchers utilized external metal layers which are high-temperature sensitive materials, such as SiN embedded in Metal-X layers used in CMOS technologies, resulting in much higher noise powers, without amplifiers Matsumoto *et al.* (2008).

- **Jitter noise:** Jitter noise, which has random distribution as the white noise, is a common noise in any clock signal. Therefore, in recent research, it has been used as an entropy source for generating random bits. Jitter-noise-based RNGs use two clock sources with one being faster than the other. The jitter noise that occurs in the faster clock signal is sampled by the slower clock. The output bit is random according to the level of jitter noise. Kim, Lee & Kim (2017a) accumulated jitter noise from differential ring oscillators. The chip fabricated in 65 nm CMOS technology showed attractive performances: up to approximately 10Mbps throughput with low power consumption (<0.5 mW). The researchers from the University of Michigan (Yang *et al.* (2014)) also proposed a TRNG based on jitter noise, which was implemented in both 28 nm and 65 nm CMOS technologies. The results showed the highest throughput was 23.16 Mbps while consuming 0.54 mW. Mathew *et al.* (2016) presented an all-digital TRNG based on the collapse time of two racing edges in even-stage ring oscillators.
- **Metastability:** Metastability takes advantage of the uncertainty of either a logical state or its resolve time. The main drawback of these methods is that they require very specific and stable initial and/or operating conditions. Therefore, the entire system is extremely sensitive to deterministic system noise sources, such as power variations, bias voltage variations, and device mismatches Kim *et al.* (2017a); Yang *et al.* (2014); Bae *et al.* (2017); Petrie & Connelly (2000); Chen *et al.* (2009); Yang, Blaauw & Sylvester (2016); Chen, Li, Wang, Liu & Yang (2016).

1.1.1.2 Harvester

Each kind of physical entropy source requires a specific harvesting circuit. Due to the low power characteristic of thermal noise, a low-noise amplifier is needed in the harvester circuit. Figure 1.4

Table 1.1 List of existing physical entropy true random number generators

Design	**Entropy**	**Technology**	**TP(bps)**	**Power(W)**
Petrie & Connelly (2000)	Thermal	2 μm	1 M	3.9m
Mathew *et al.* (2012)	Meta.	45 nm	2.4 G	7 m
Yang *et al.* (2014)	Jitter	28 nm/65 nm	23.16 M/2.8 M	0.54 m
Laurenciu C. (2015)	Thermal	65 nm	1 G	–
Liu, Liu, Li & Zou (2016)	Jitter	130 nm	0.1 M	40μ
Mathew *et al.* (2016)	Jitter	40 nm/180 nm	2 M/1 M	46μ/109μ
Bae *et al.* (2017)	Meta.&Jitter	65 nm	3 G	5 m
Kim *et al.* (2017a)	Jitter	65 nm	8.2 M	0.289 m

shows a simple implementation of a thermal noise harvesting circuit and the simulation results. The circuit is implemented in 65 nm CMOS technology. Laurenciu C. (2015) compared two identity resistor noise sources to generate the random sequence. Using comparison eliminates the common-mode and power supply interferences. The circuit, implemented in 65 nm CMOS technology, can provide bit samples at 1GHz. The work of Bae *et al.* (2017) proposed a combination of thermal noise and metastability, which allows to use a high sampling frequency to dominate the Flicker noise. The implementation using 65 nm CMOS technology presented high throughputs (3 Gbps) with a relatively low power consumption of 5 mW.

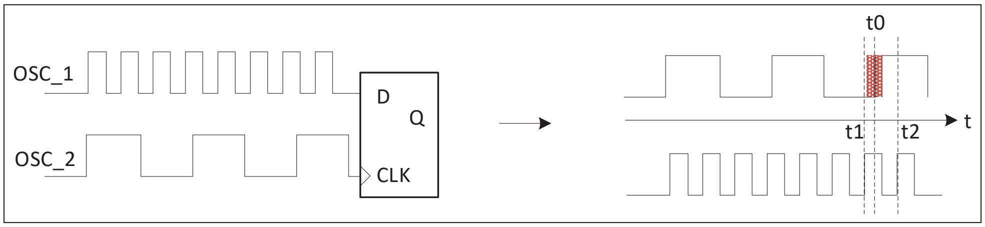


Figure 1.2 Jitter noise harvesting circuit implementation

Figure 1.2 shows an example of implementation of the jitter noise harvesting circuit. However, physical noise, such as thermal noise and jitter noise, present several disadvantages, such as limitted dynamic range and unknown statistical distributions, that hinders their implementations Pamula *et al.* (2018).

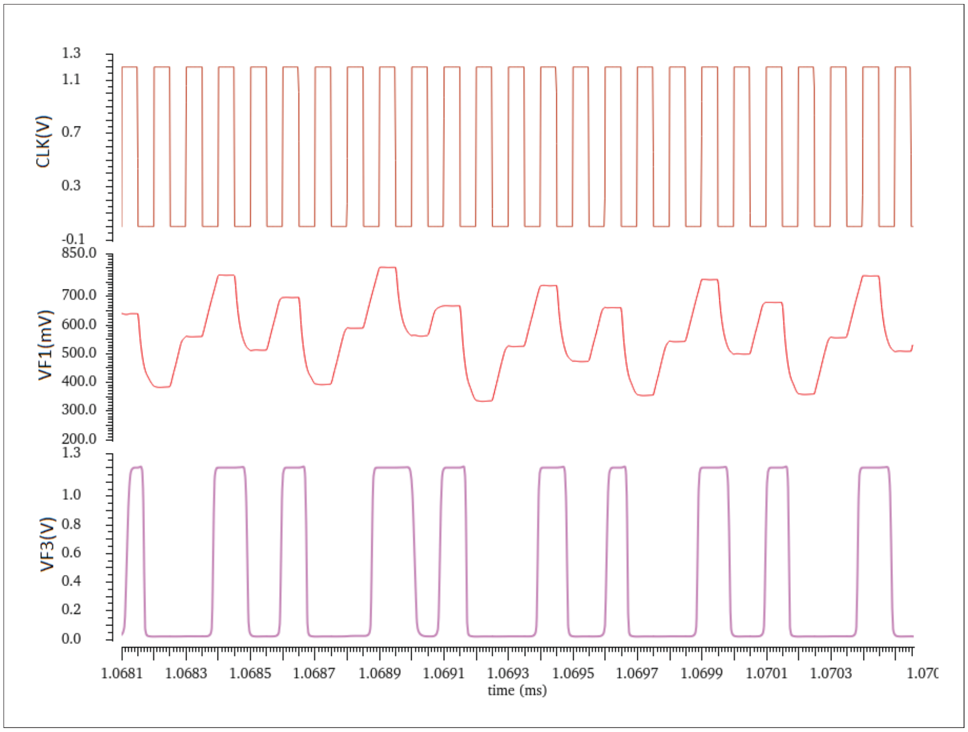


Figure 1.3 Simulation results of the thermal noise circuit

On the other hand, metastability is difficult to implement because it requires sophisticated circuit design, the elimination of other noise sources, and high accuracy calibration circuits. Figure 1.5 represents the concept of metastability source, which can be used in TRNG. Tokunaga, Blaauw & Mudge (2008) proposed a scheme for the use of metastability of circuit noise with a complicated control circuit based on statistical analysis. A time-to-digital converter was implemented to provide input for the statistical analysis block which was performed in a digital circuit. Charge injection was utilized to drive the system state to the metastability region. Mathew *et al.* (2012) designed delay cells in a digital circuit to control circuits in metastability. This design achieved a throughput of 2.4 Gbps. The following table compares existing researches on physical entropy TRNGs.

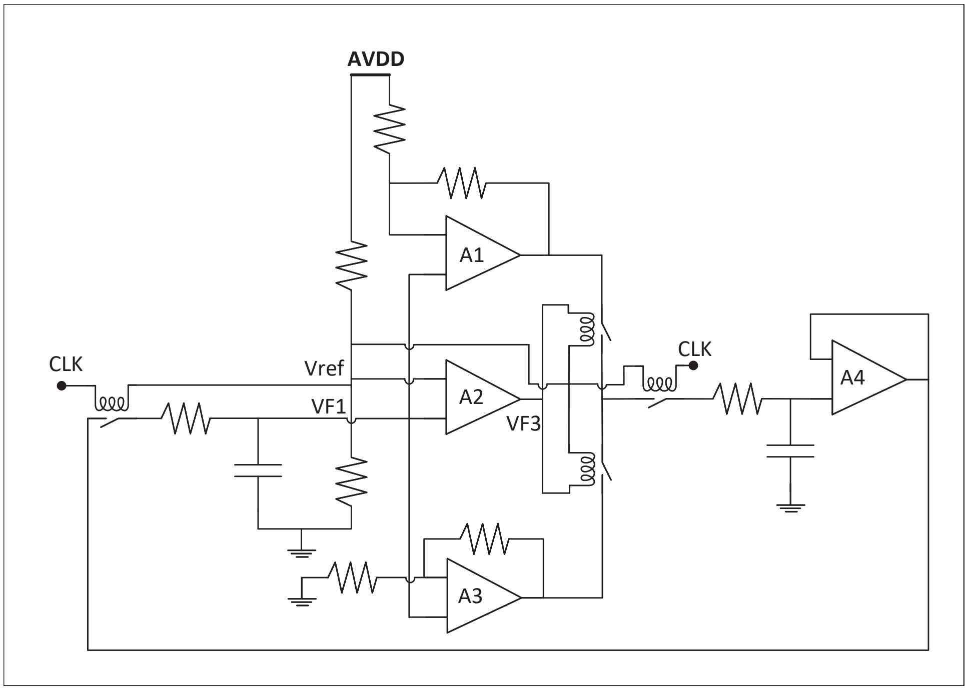


Figure 1.4 Thermal noise harvesting circuit

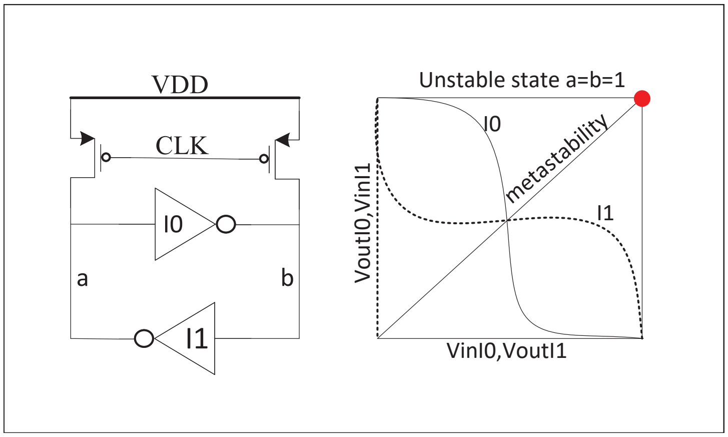


Figure 1.5 Metastability to generating random bits
Taken from Mathew et al. (2012, p.2-3)

1.1.1.3 Data post-processing

The output of an RNG is expected to be unpredictable in the information-theoretic sense. Therefore, it should have good statistical properties. In other words, it contains no recognizable patterns or regularities. Ideally, it should have uniform distribution. However, the output of physical entropy sources usually doesn't have ideal properties due to circuit imperfections, and uncontrollable circuit noise, such as device mismatches, and power variations. Although the technical design eliminates major problems, we still need post-processing to ensure the properties of the output random bits meet the expectations. The random extractor, which needs to be designed to suite with the random sources, is the main part of the post-processing Bouda, Krhovjak, Matyas & Svenda (2009). There are multiple algorithms and techniques that can be used in software-based data correction. However, in hardware designs, especially for light-weight devices, the post-processing circuit should be simple and effective in terms of statistical analysis. Therefore, three commonly used techniques in hardware implementation are listed in what follows.

- **XOR-based correction:** XOR-based correction is superior in implementation cost since it uses only XOR-operations Kwok *et al.* (2011). Two bits from the random source are XOR-ed together to construct a one-bit output. If the input bit bias is supposed to be ϵ, the correction rate is $1/2$ and the output bias is reduced to $2 \times \epsilon^2$.
- **Von-Neumann correction:** The Von-Neumann method, which can produce perfect unbiased outputs, is usually used for a biased random stream. The correction technique is applied to the non-overlapping pairs of bitstreams such that (1) two identical adjacent bits are discarded, and (2) only the first bit is reserved for two different adjacent bits Rozic, Yang, Dehaene & Verbauwhede (2016). Therefore, the conversion rate, which is defined as the length of the output bit stream over the length of the input bit stream, is fairly small. Assume that the input bits have a bias of ϵ, the correction rate is $(1/4 - \epsilon^2)$.
- **Linear code correction:** The construction of linear code correction is given by a transform $L : GF(2)^8 \times GF(2)^8 \to GF(2)^8$:

$$L(X, Y) = X \oplus (X << 1) \oplus (X << 2) \oplus (X << 4) \oplus Y. \tag{1.4}$$

  This correction needs two independent random sources with bias of ϵ, in which each source contributes 8 bits, then compresses to an 8-bit output. Therefore, the compression rate is $1/2$

while this function achieves an output bias of $2^4 \times \epsilon^5$. Linear code correction is generalized by Kwok *et al.* (2011) as follows.

Theorem 1 *Let G be a linear corrector mapping n bits to m bits. Then the bias of any non zero linear combination of the output bits is less than or equal to* $2^{d-1}\epsilon^d$*, where d is the minimal distance of the linear code constructed by the generator matrix G.*

According to these above data post-processing methods, we can choose the right data post-processing for random generators which is suitable with the input data bias and the requirement of random output bias.

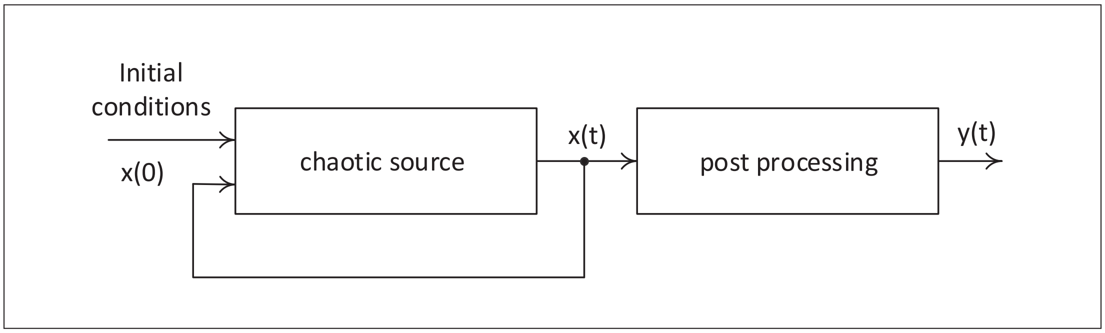


Figure 1.6   Conceptual architecture of chaos-based RNG

1.1.2 Architecture of chaos-based RNG

In this dissertation, we deal with chaos as an entropy source. The conceptual architecture of chaos-based RNGs is presented in Figure 1.6. A non-linear system exhibits chaotic behavior if its features inherent characteristics include (i) high sensitivity to initial conditions – a slight change in initial conditions yields significantly different future trajectories, and (ii) irregular motion in the phase space – phase space trajectories do not converge to a point or a periodic orbit Stojanovski, Pihl & Kocarev (2001). Thanks to these properties, any small unavoidable uncertainty in the system's initial conditions will make a chaotic system, at a certain point of observation, an actual unpredictable random-like process. The advantage with respect to the previously considered architectures is that, by using chaos as an entropy source, it is possible to have a precise knowledge of the process statistics that are set by the chaotic circuit.

1.1.2.1 Random chaotic source

Random chaotic sources can be classified into discrete-time and continuous-time where both approaches can be effectively used to produce random numbers. In discrete chaotic systems, a mapping function in the form of $x_{k+1} = F(x_k)$ is used iteratively. Examples of such systems include the logistic map, the Renyi map, and piecewise affine Markov maps Kocarev, Szczepanski, Amigo & Tomovski (2006); Nguyen, Kumar & Song (2018); Wieczorek & Golofit (2018); Palacios-Luengas & Duchen-Sainchez (2012). As demonstrated in García-Guerrero, Inzunza-González, López-Bonilla, Cárdenas-Valdez & Tlelo-Cuautle (2020), many researchers have been improving the complexity of chaotic maps, where higher dimensional chaotic maps, such as Logistic 2D map, Chen hyperchaotic map, and Rossler hyperchaotic map, have been used. Conversely, continuous chaotic systems are presented by differential equations $X' = F(X)$ Nguyen, Pham-Nguyen, Nguyen & Kaddoum (2020).

1.1.2.2 Data post-processing

Similar to data post-processing used in conventional RNGs, the post-processing process in chaos-based RNGs is also based on three kinds of data correction: XOR, Von-Neumann technique, and linear code correction. The XOR-based technique is the most commonly used due to its simple hardware design Wannaboon, Tachibana & San-Um (2018); Wieczorek & Golofit (2018); Chandrasekaran, Karnam & Sanyal (2020).

1.2 Background on dynamical system

Since chaos is a mathematical branch of dynamical systems, it is important to review background information on dynamical systems and their characteristics. Chaos is a special region of dynamical systems that has random states of disorder and irregularities, that are highly sensitive to initial conditions. Small differences in initial conditions can yield divergent outcomes that are difficult to predict. Chaotic behaviors also exist in many natural systems, such as weather, climate, and solar systems, and are applied in many scientific research and disciplines, including sociology, physics, computer science, economics, and engineering. The concept and characteristics of chaotic systems are very suitable for random number generators where randomness and non-periodicity are important criteria. Chaotic systems have been used as sources of randomness for decades Stojanovski *et al.* (2001). Different from physical noise sources, it is proved in mathematics that it behaves randomly in a coarse-grained state-space. In this section, we present several crucial characteristics that are used to evaluate the strength of a chaotic entropy source in generating random numbers. A dynamical system consists of a set of possible states, together with a rule that determines the present states based on the previous states Alligood, Sauer & Yorke (2000). If the rule is applied at discrete time, the system is called iterated map. Otherwise, if the time is thought of as being continuous, the future state is defined by the change in the previous states and called a differential equation. We previously showed that there are two main chaotic entropy sources in chaos-based RNGs, namely continuous and discrete-time chaotic sources, which correspond to two main types of dynamical systems, i.e. differential equations and iterated maps, respectively.

1.2.1 Classification of dynamical systems

1.2.1.1 Iterated maps

Iterated maps (also known as difference equations), are dynamical systems in which a set of possible states are determined based on the previous state using a rule. The general form of iterated maps is given as

$$X_{n+1} = F(X_n), \tag{1.5}$$

where the system state vector at discrete time step n is $X_n = [x_1, x_2, ...x_m] \in \mathbb{R}^m$, which has m dimensions, and the function $F(\cdot)$ is applied to previous states on some state spaces (or so-called the phase space) $\mathbb{U}$. The continuous iterations from the initial point $x_0 \in \mathbb{U}$ generate the phase space.

1.2.1.2 Differential equations

In continuous time, dynamical systems are given by a set of differential equations. A differential equation that involves only one independent variable is called an ordinary differential equation (ODE). In this thesis, we deal with differential equations where the independent variable is time. The general form of a differential equation is defined as

$$X(t)' = \frac{\partial X}{\partial t} = F(X(t)), \tag{1.6}$$

where the system state vector $X(t) = [x_1, x_2, ...x_m] \in \mathbb{R}^m$, and m indicates the number of dimensions of the differential equation.

1.2.2 Equilibrium points and system stability

The stability of dynamic systems is evaluated with equilibrium points. Dynamic systems should be unstable to exhibit chaotic characteristics. If the equilibrium points are stable, dynamic systems have fixed-point solutions, and then the system is stable or steady. Conversely, if the equilibrium points are unstable, the system is unstable and can have chaotic characteristics Strogatz (2015). An equilibrium point is the simplest possible solution of a dynamical system, where the state variable is a constant. In iterated maps, it is simple to find the equilibrium points by solving the equation $F(X_n) = X_n$. In differential equations, the state variables are not changed at the equilibrium points, therefore the equilibria satisfy the equation $\frac{\partial X}{\partial t} = 0$. To evaluate the stability of equilibrium points, the Lyapunov criterion is evaluated.

- **Lyapunov criterion:** The stability analysis of an equilibrium point X_0 is verified by evaluating the stability of the corresponding linearized system in the vicinity of the equilibrium point Datta (2004). Considering the differential equation in (1.6) [the iterated map in (1.5)], the corresponding linearized system is:

$$X'(t) = J.X(t) + B \qquad [X(n+1) = J.X(n) + B], \tag{1.7}$$

where J is called Jacobian matrix. For both dynamical system types, the following statements are applied:

- If all the eigenvalues of the matrix J have "negative real parts", the equilibrium point X_0 and the dynamical systesm are asymptotically stable.
- If at least one of the eigenvalues of the matrix J has a "positive real part" then the equilibrium point and the dynamical system are unstable .
- If at least one eigenvalue of the matrix J is located "on the imaginary axis" while all the other eigenvalues have a "negative real part" ["modulus less than 1"], it is not possible to conclude anything about the stability of the equilibrium point (X_0).

The eigenvalues of the Jacobian matrix at the equilibrium point satisfy the condition ($J - \lambda I = 0$), which is called the "characteristic equation". The Routh-Hurwitz criterion is used to evaluate the eigenvalues of the Jacobian matrix J as follows Ivanescu (2001):

- **Routh-Hurwitz criterion:** Routh-Hurwitz criterion is applied on the characteristic polynomial ($P(\lambda)$) to evaluate the eigenvalues of the matrix J at the equilibrium point.
  - The second-degree polynomial, $P(\lambda) = \lambda^2 + a_1\lambda + a_0$ has both roots in the open left half plane, and consequently the system with characteristic equation $P(\lambda) = 0$ is stable, if and only if both coefficients satisfy $a_i > 0$.
  - The third-order polynomial $P(\lambda) = \lambda^3 + a_2\lambda^2 + a_1\lambda + a_0$ has all roots in the open left half plane if and only if a_2, and a_0 are positive and $a_2a_1 > a_0$.
  - In general the Routh stability criterion states that a polynomial has all roots in the open left half plane if and only if all first-column elements of the Routh array have the same sign.

1.2.3 Lyapunov exponents and dynamic characteristics

The Lyapunov exponents are used to evaluate the dynamic characteristics of a dynamic system. To exhibit chaotic characteristics, the necessary condition is that the system has at least one positive Lyapunov exponent Strogatz (2015). The Lyapunov exponent, which is calculated for each dimension of the system state $X(t) = [x_1, x_2, ..., x_n]$, is defined as:

$$L_i = \lim_{t\to\infty} \frac{1}{t} \log \frac{\|\partial x_i(t)\|}{\|\partial x_i(0)\|}. \tag{1.8}$$

The Lyapunov dimension, which is an effective metric to evaluate the complexity of a dynamical system, is calculated as:

$$D_L = j + \frac{1}{|L_{j+1}|} \sum_{i=1}^{j} L_i, \tag{1.9}$$

where j is the largest index of non-negative Lyapunov exponents.

1.2.4 Phase space orbits

Phase space or phase portrait is a diagram showing all the qualitatively different trajectories of the system. By observing the phase space, the dynamic characteristics of the system are visualized, and the range of data space is observed. If the system is stable or steady, the phase space is converged to equilibrium points. Additionally, the system can have periodic orbits of phase space (the system has limitted solutions) or chaotic orbits (the system has chaos) Strogatz (2015).

1.2.4.1 Periodic orbits

If there exists a $T > 0$ such that $F(t + T, X_0) = F(t, X_0), \forall t$, and X_0 is not an equilibrium, the system state $F(t, X_0)$ is called a periodic orbit or cycle.

1.2.4.2 Chaotic orbits

Dynamic systems have chaotic orbits when they are unstable and the system's states are bounded. This means that the system's state travels stochastically in a limitted, bounded space. In this region, they are very sensitive to initial conditions, therefore, they do not apprear to be either periodic or converged to equilibrium solutions.

1.2.5 Bifurcation diagram

The bifurcation diagram describes the data space of the dynamic system according to the variation of one or more parameters. Then, a bifurcation occurs when there is a significant change in the structure of the solutions of the dynamic system as a parameter varies. For example, the system changes from having fixed-point solutions to having chaos Morris W. Hirsch,Stephen Smale (2013).

1.2.6 Poincare map

A Poincare map is the intersection of a periodic orbit in the state space of a continuous dynamic system with a certain lower-dimensional subspace, called the Poincare section, transversal to the flow of the system. It is a method to study phenomena commonly occurring in a chaotic system by taking an N-dimensional chaotic system and mapping it in a coordinate of $N - 1$ dimensions by taking slices of the phase space of time-space at specific values of one of the parameters. One such example would be to study a chaotic system with a periodic forcing function by taking slices through the phase space at every integer multiple of the period. This gives us an idea of where we expect to catch the system at every cycle. For a periodic system, this section will be a cluster of points at the same location whereas, for a chaotic system, this section provides us with some fascinating pictures that give us a more in-depth look than a phase space plot alone into how the system works. In other cases, we can choose to make a Poincare plot by selecting the inflection points of one of the coordinates and mapping the others at those points.

1.2.7 Solving differential equations on computer

There are several numerical methods to solve differential equations which are used in chaotic systems. In this dissertation, I use two nummerical algorithms to solve chaotic systems and also evaluate dynamic characteristics: the Euler method and the fourth-order Runge-Kutta algorithm.

1.2.7.1 Euler method

The Euler method (also called forward Euler method) is a simple first-order nummerical procedure for solving ordinary differential equations (ODEs). It can be considerred as the simplest Runge-Kutta method. Recall the continuous chaotic system equation that is $X'(t) = F(t, X(t))$, choose a value h for the size of every step and set $t_n = t_0 + nh$, then one step of the Euler method from t_n to $t_{n+1} = t_n + h$ is:

$$X_{n+1} = X(n) + hF(t_n, X_n). \tag{1.10}$$

The value of X_n is an approximation of the ODE's solution at time $t_n : X_n \approx X(t_n)$. The Euler method is explicit, i.e. the solution X_{n+1} is an explicit function of X_i for $i \leq n$.

1.2.7.2 Fourth-order Runge Kutta algorithm

The initial ODE problem is specified as follows:

$$X'(t) = F(X(t)), X(t_0) = X_0. \tag{1.11}$$

Now we pick a step-size $h > 0$ and define:

$$X_{n+1} = X_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4), t_{n+1} = t_n + h. \tag{1.12}$$

For $n = 0, 1, 2, 3, \ldots$, we use four primitive parameters:

$$\begin{aligned} k_1 &= F(t_n, X_n) \\ k_2 &= F(t_n + \frac{h}{2}, X_n + h\frac{k_1}{2}) \\ k_3 &= F(t_n + \frac{h}{2}, X_n + h\frac{k_2}{2}) \\ k_4 &= F(t_n + h, X_n + hk_3) \end{aligned} \tag{1.13}$$

1.3 Chaos-based RNGs in the literature

In the previous section, it was shown that thermal noise based TRNGs present a simple circuit design, but the noise harvested is small and needs to be amplified. Moreover, thermal noise circuits provide low-throughput and low-speed output bit sequences. Meanwhile, jitter noise-based TRNGs provide higher throughputs compared to thermal-noise-based methods. However, jitter noise circuits also do not provide the required high-speed signal output, because they use a slow-speed clock signals. Both thermal noise and jitter noise-based TRNGs are highly affected by deterministic system noises. Metastability based TRNGs generate random bit sequences at high speed and high throughput with a low-power consumption, which provides high-energy efficiency. However, this method requires a complex control circuit for calibration and driving the system state into the metastability region. Also, the metastability method is very sensitive to the environment; hence, attack resistance is a challenge for this method. It is very challenging to use metastability-based TRNGs in high-security applications.

Chaos-based TRNGs are emerging as the natural choice for high security and high-efficiency applications. In the past, chaotic circuit designs were implemented using high power consumption off-the-shelf devices. Nowadays, with the development of microelectronic devices and the down scaling of technology, many chaotic circuits have been implemented in CMOS. Moreover, there are more and more chaotic systems being developed and verified mathematically. If the chaotic system and its parameters are chosen properly, the system could be implemented at the circuit level to exhibit chaos despite the variation of parameters. A chaotic system is expressed by a deterministic expression. Therefore, it is not sensitive to the system's noises such as voltage and temperature variations.

1.3.1 RNGs based on chaotic maps

Most truly chaos-based RNGs are induced from linear chaotic maps because of their simplified implementation (Fatemi-Behbahani, Ansari-Asl & Farshidi (2016); Wieczorek & Golofit (2018); Zhou, Zhou, Yu & Ye (2006)). Among several approaches for implementing linear chaotic maps, Markov maps are widely used in very large scale integration designs due to their low implementation cost. Generally, the Markov chaotic map of a dynamic system with state variable x is updated as follows Callegari, Rovatti & Setti (2005).

$$x_{n+1} = M(x_n), \tag{1.14}$$

where $M : S \rightarrow S$ is a map of an interval partition $S = \bigcup X_i, i = 0, 1, ..., p-1$; here:

- M is affine on each X_i
- Either $X_i \cap M(X_j) = \emptyset$ or $X_i \subseteq M(X_j)$ for any i, j (which is equivalent to saying the partition points are mapped into partition points).

Some derivations of Markov chaotic maps are N-Bernoulli maps, tent maps, zigzag maps, and piecewise affine Markov (PWAM) maps. The following section addresses each kind of map with practical implementations.

1.3.1.1 Logistic map

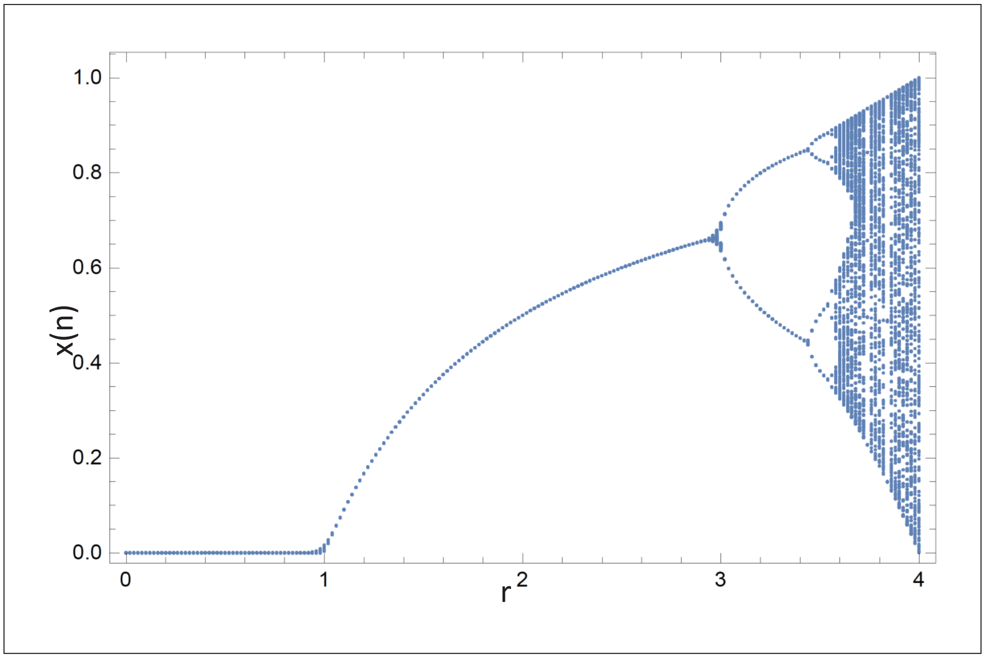


Figure 1.7 Logistic bifurcation diagram with $r \rightarrow [0, 4]$

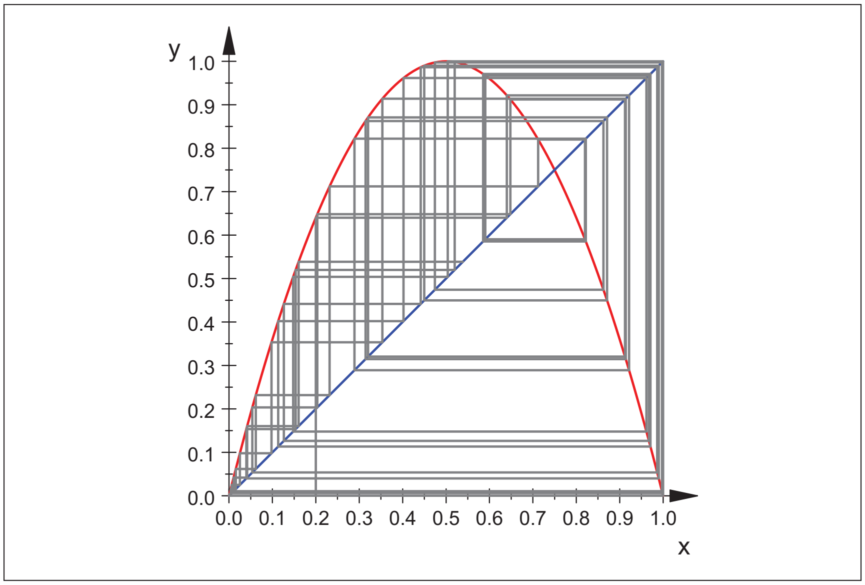


Figure 1.8   Chaotic behavior of logistic map for $r = 4$

The logistic map is expressed by the equation $x_{n+1} = rx(1 - x)$ where r is a system parameter. Figure 1.7 shows the logistic bifurcation diagram. As can be seen from the bifurcation diagram, the system behavior changes with the variations in r. With r below 3.6, the system fails to exhibit chaos. Most of the values with r above 3.6 cause chaotic behavior, but there are still certain isolated ranges of r that show non-chaotic behavior. Figure 1.8 shows the trajectory of the logistic map with $r = 4$. N Nguyen, Kaddoum & Gagnon (2018) proposed a scheme using both thermal noise and a logistic map which is based on fuzzy modeling. The circuit design, implemented in 65 nm CMOS technology, showed a better energy efficiency in compared to the previous chaotic maps implemented in CMOS technology.

1.3.1.2 N-Bernouli map

The N-Bernoulli map can be expressed as

$$M : [-1, 1] \to [-1, 1], \quad M(x) = (Nx \quad mod \quad 2) - 1. \tag{1.15}$$

Figure 1.9 shows N-Bernoulli maps for different values of N.

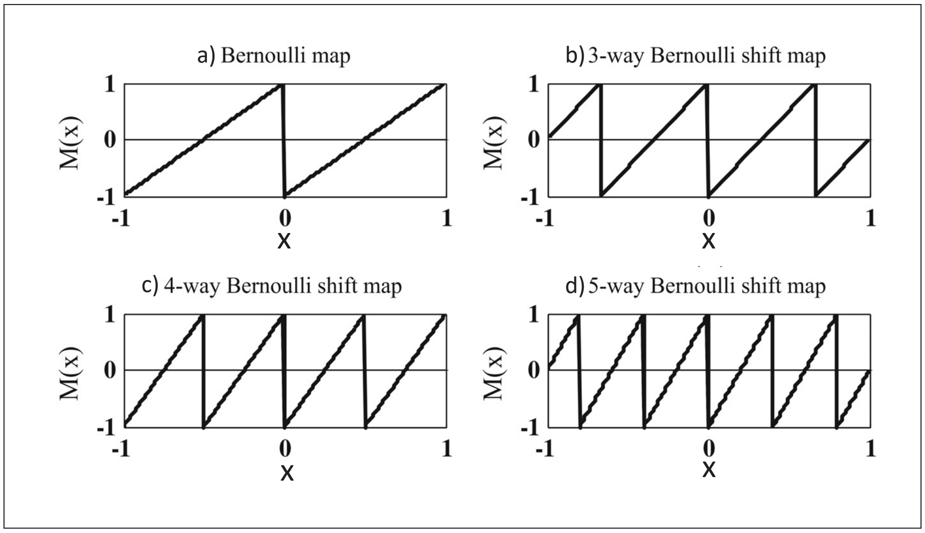


Figure 1.9 N-Bernoulli shift map for different values of N
Taken from Fatemi-Behbahani et al. (2016, p4)

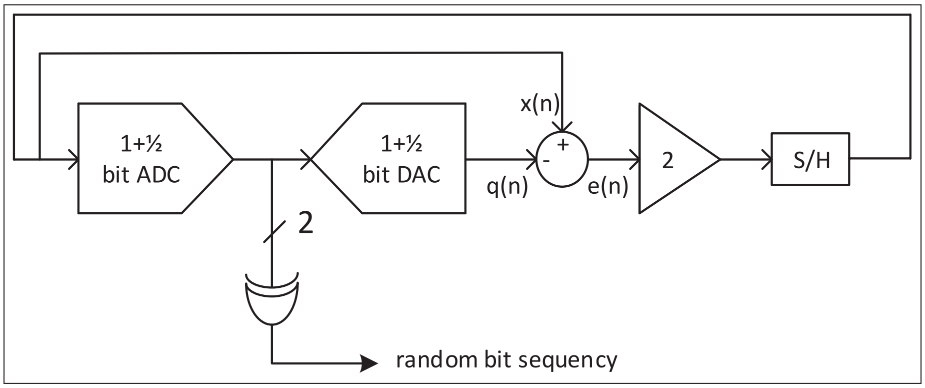


Figure 1.10 Basic block diagram of 1.5-bit ADC using as true random source. (S/H is sample and hold block)

Considering the 1-bit stage of the algorithmic analog-to-digital conversion (ADC), the input-output relationship is expressed as:

$$x_{n+1} = y_i = \begin{cases} 2(x_i + 1/2) & -1 < x_i < 0(D_i = 0) \\ 2(x_i - 1/2) & 0 < x_i < 1(D_i = 1) \end{cases} , \tag{1.16}$$

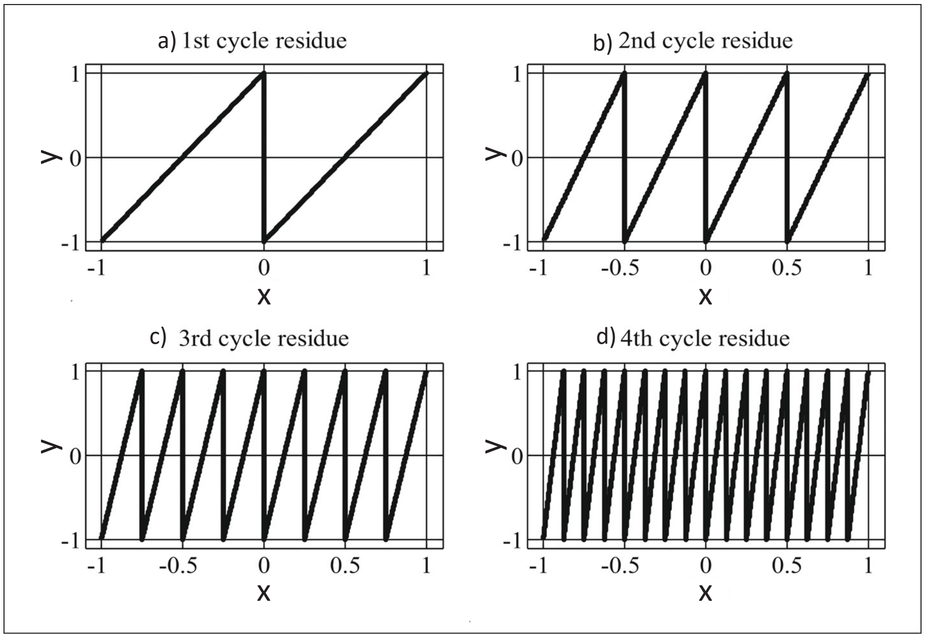


Figure 1.11 The different cycle residue of 1-bit ADC
Taken from Fatemi-Behbahani et al. (2016, p8)

where D_i is digital output value.

This equation is the Bernoulli map with $N = 2$. Therefore, the algorithmic ADC structure is employed to implement Bernoulli chaotic maps. The block diagram of a 1.5-bit ADC used in a TRNG is shown in Figure 1.10. Figure 1.11 shows the full-bit/cycle residue with different number of cycles used in Fatemi-Behbahani *et al.* (2016). Kim *et al.* (2017a) indicated that $M-$cycles of the algorithmic ADC is exactly identical to $N-$Bernoulli shift maps where $N = 2^M$. Based on the fundamental theorem of random variables, the M^{th} cycle output's probability density function (pdf) (f_{y_M}) is given by:

$$f_{y_M} = \begin{cases} \frac{1}{2^M} \sum\limits_{n=1}^{2^M} f_x(\frac{y-(2n+1-N)}{N}) & -1 < y < 1 \\ 0 & otherwise \end{cases} , \tag{1.17}$$

where f_x is the input's probability density function (pdf). Assuming that the input signal has a uniform pdf in the interval $[-1, 1]$, then the output pdf is also uniform. In reality, f_x and f_{y_M} are nonzero, therefore they can be extended using Fourier series for periodic signals as follows.

$$\tilde{f_x} = \sum_{k=-\propto}^{+\propto} a_k e^{j\pi kx} \rightarrow a_k = \frac{1}{2}\int_{-1}^{+1} f_x(x) e^{-j\pi kx} dx, \tag{1.18}$$

$$\tilde{f_{y_M}} = \sum_{k=-\propto}^{+\propto} b_k e^{j\pi ky} \rightarrow b_k = \frac{1}{2}\int_{-1}^{+1} f_{y_M}(y) e^{-j\pi ky} dy. \tag{1.19}$$

From these above equations, we get the relation between a_k and b_k as: $b_k = (-1)^k a_{Nk}$. Therefore, if the number of stages is sufficient, only the DC components are preserved and all the harmonics are weeded out. Then, after numerical iterations, the pdf converges to the uniform distribution that enables these maps to be used in RNGs Fatemi-Behbahani *et al.* (2016). The benefit of these approaches in comparison with the conventional Bernoulli map comes from the reuse of basic building blocks (1.5-bit or 1-bit blocks) of pipeline ADCs or arithmetic ADCs, which are very common designs in mixed-signal circuits. Other work on designing Bernoulli maps in programmable analog building blocks is presented in (Callegari (2016)). The map parameters are obtained by programming analog blocks to form comparators, voltage references, and amplifiers. In fact, this implementation suffers from circuit variations. Moreover, it is well known that in Bernoulli maps, a small variation in the map parameters may change the mapping behavior abruptly, possibly causing parasitic equilibrium points.

1.3.1.3 Tent and zigzag maps

Palacios-Luengas & Duchen-Sainchez (2012) digitized a chaotic map called T-1D (Tent map) and its inverse IT-1D (Inverse Tent map) chaotic map which can be expressed by the equation:

$$x_{n+1} = f_\mu(x_n) = \begin{cases} 2\mu x_n + \frac{1-\mu}{2} & a \le x_n \le \frac{1}{2} \\ -2\mu x_n + \frac{1-\mu}{2} & \frac{1}{2} \le x_n \le b \end{cases}. \tag{1.20}$$

This iterated function generates orbits of system state $\{x_0, x_1, x_2, ...\}$, which depend on initial condition x_0, and control parameter μ. The system state is in the range $[a, b] \in \Re$, with

$a < b \in [0,1]$. This map was implemented in the FPGA platform to generate digital noise for secure communications and was shown to yield a flat distribution for the output noise. The zigzag map is a chaotic map derived from the tent map where the next state is updated according to the following equation:

$$x_{n+1} = \begin{cases} -m(x_n + \frac{2}{|m|}) & -1 < x_n < -\frac{1}{|m|} \\ mx_n & -\frac{1}{|m|} < x_n < \frac{1}{|m|} \\ -m(x_n - \frac{2}{|m|}) & \frac{1}{|m|} < x_n \leq 1 \end{cases}, \tag{1.21}$$

where $m \in \mathfrak{R}$ is control parameter, and $m \in (-3, 3)$. Nejati, Beirami & Ali (2012) demonstrated a switched-circuit implementation for the above zigzag map. The proposed circuit was simple and easy to implement. However, switched-circuits have high harmonic frequencies at switching points. The more switched circuits are used, the worse the randomness of the output stream.

1.3.1.4 Recent research in extended chaotic maps

Recently, numerous research works have focused on increasing the dynamic characteristics of chaotic maps. This can be achieved by 1) extending one-dimensional chaotic maps to higher-dimensional generations to achieve a tradeoff between the implementation cost and chaotic performance Ozturk & Kilic (2019) and 2) enhancing the chaos complexity of chaotic maps Hua, Zhou & Zhou (2019); Hua & Zhou (2020). Ozturk & Kilic (2019) proposed a generalization of the higher-dimensional Baker map, in which the Lyapunov exponents and the attractors are analyzed mathematically. Moreover, a hardware implementation of PRNG based on the high-dimensional Baker map was presented. Hua *et al.* (2019); Hua & Zhou (2020) presented an improvement to existing 2D chaotic maps by using sinusoidal functions. The dynamic characteristics were demonstrated by analyzing the Lyapunov exponents and bifurcation diagrams.

1.3.2 RNG based on continuous chaotic systems

1.3.2.1 3D chaotic systems

Continuous chaotic systems have been developed in hardware design, only a few were designed in a highly integrated circuit, such as CMOS technologies, due to their high power consumption, low frequency operation, and inability to operate at low voltage levels. Among the numerous continuous chaotic systems available, Chua's circuit seems to pioneer in the implementation of the continuous chaotic system in integrated circuits. Chua's circuit is implemented using four linear elements (two capacitors (C_1, C_2), an inductor (L), and a resistor (R)) and a nonlinear resistor, called Chua's diode, as represented in Figure 1.12 (Elwakil & Kennedy (2000a); Muthuswamy (2009)).
The system state is expressed by the following differential equations:

$$\begin{cases} C_1 \frac{d_{v_{C_1}}}{dt} = \frac{1}{R}(v_{C_2} - v_{C_1}) - i_R \\ C_2 \frac{d_{v_{C_2}}}{dt} = \frac{1}{R}(v_{C_1} - v_{C_2}) + i_L \\ L \frac{d_{i_L}}{dt} = -v_{C_2} \end{cases}, \tag{1.22}$$

$$i_R = g(v_{C_1}) = m_0 v_{C_1} + \frac{1}{2}(m_1 - m_0)(|v_{C_1} + E_1| - |v_{C_1} - E_1|), \tag{1.23}$$

where m_0 and m_1 are slopes of two fragments of non-linear function.

Chua's circuit requires a local oscillator LC and a non-linear element that is designed from off-the-shell OPAMP devices. Moreover, there are works on Chua's circuit that use an RC oscillator or other kinds of oscillators to replace the LC-oscillator which consumes high power and a large circuit area making it incompatible with semiconductor designs. Other chaos systems can be considered, such as stroboscopic Poincare maps and Jerk equations which have been employed in RNGs due to their simple circuit design (Ergun & Ozoguz (2005); Mendoza, Araque-Lameda & Colina-Morles (2016); Gandhi & Roska (2009)). The circuit level of one of the transformations of stroboscopic Poincare equations used in RNGs is shown in circuit level in

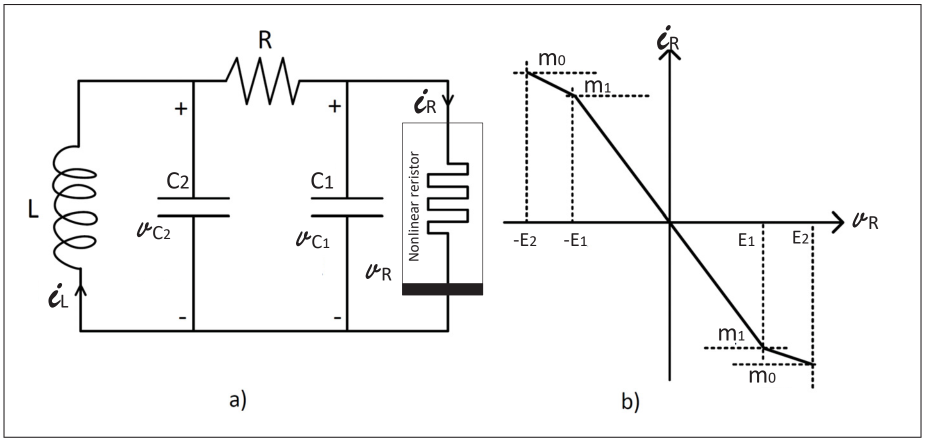


Figure 1.12 a) Chua’s circuit b) Chua’s diode

Figure 1.13. The following equation expresses the chaotic oscillator:

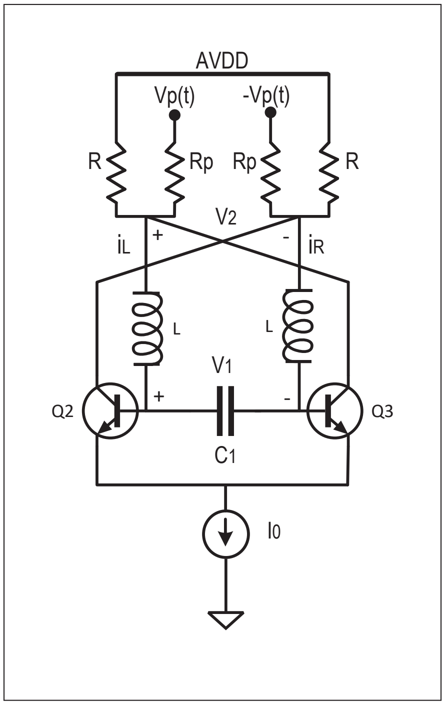


Figure 1.13 Stroboscopic map chaos-based Oscillator

$$\begin{cases} C_1\dot{v_1} = -i_3 \\ L\dot{i_3} = (v_1 - v_2) \\ C_p\dot{(v_2)} = i_3 - (\frac{1}{R} + \frac{1}{R_p})v_2 + \frac{2}{R_p}V_P\mathrm{sign}(\sin(\omega t)) + I_0\tanh(\frac{v_1}{2V_T}) \end{cases} \tag{1.24}$$

where $i_3 = i_R - i_L$, $v_p(t) = V_P\mathrm{sign}(\sin(\omega t))$ is the external periodical pulse, and V_T is the thermal noise. With proper values for passive components, the above equation exhibits a double-scroll chaotic characteristic. The circuit was targeted for implementation in 0.35μm BiCMOS technology. However, the works of Ergun & Ozoguz (2005) provided simulation results only, without an experimental circuit design. Therefore, to realize this kind of approach, more works are needed to verify the robustness of the system in terms of circuit variations.

Other 3D chaotic systems are used for random bit generation including: Lorenz system Azzaz, Tanougast, Sadoudi & Dandache (2009); Zhang (2017b), Chen's system Hu, Liu & Ding (2013), Sprott's system Bonny, Al Debsi, Majzoub & Elwakil (2019), and Jerk's system Patidar & Sud (2005). The Lorenz system is commonly used in engineering applications due to its high dynamic characteristics and perfect symmetry.

1.3.2.2 Recent research on 3D chaotic systems

Recently, numerous works have been focusing on developing and proposing new 3D chaotic systems that have more complicated dynamic characteristics, in order to increase the randomness of the generated output bits. Zhang & Wang Zhonglin (2013) proposed a new 3D chaotic system based on Lu's system by using an exponential function instead of the nonlinear term in the origin. The Lyapunov exponents were evaluated to prove the dynamic characteristics of the proposed system. Although chaotic systems are unpredictable and have random-like state trajectories, they can be studied and recovered using computational tools. A cyber attack has a high possibility of success if the target system uses well-known and self-excited oscillators for its random bit generator. After the transient process, a trajectory starting from the point of an unstable manifold in a small neighborhood of unstable equilibrium can be revealed. Therefore, the system's parameters can be computed for recovering the target system. From this point of view, the development of modern computers enables the numerical simulation of complex nonlinear dynamical systems and therefore the structure of their trajectories can be deduced. However, this approach showed very limited success with regards to hidden attractor chaotic systems Jafari & Sprott (2013). Therefore, a new branch of chaotic systems, which has no equilibrium or unlimited equilibria as suggested in Pham *et al.* (2016a), was developed . Additionally, chaotic systems having a memory element, such as a memristive devices, are also attracting significant attention Bao, Zou, Liu & Hu (2013).

1.3.2.3 Hyperchaotic-based RNG

High-order continuous chaotic systems – chaotic systems with at least four dimensions and two positive Lyapunov exponents (LE) have been recently attracting significant attention. In a chaotic system, Lyapunov exponents are important criteria to evaluate the system's dynamics. The sensitivity to initial conditions of a dynamic system is represented by a positive LE. An n-dimensional dynamical system has a spectrum of n-Lyapunov exponents. In order to exhibit chaos, a dynamical system needs to be at least three dimensional (3D) with a positive LE. A hyperchaotic system exhibits rich dynamics since the system states are simultaneously expanded exponentially in several directions. Due to this property, hyperchaotic systems are interesting candidates for the generation of random keys used in miscellaneous applications in engineering, such as secure communications, cryptosystems, and encryptions Teh, Teng & Samsudin (2016). Some hyperchaotic systems with hidden attractors are presented in Bao *et al.* (2018); Prousalis, Volos, Stouboulos & Kyprianidis (2017); Pham, Vaidyanathan, Volos, Jafari & Wang (2016b).

1.3.3 RNG based on chaos laser semiconductor

In the last few decades, laser diodes have been used for the generation of random number sequences, known as quantum-based RNGs (QTRNG). The theoretical background is based on a quantum physics mechanism measuring the photon arrivals emitted from a laser diode (LD). The arrival time of successive photons are independent and identically distributed according to the exponential distribution with parameter λ, representing the average emission rate of a photon from a source. Then, the number of events k in the time interval $[t, t+T]$ is distributed according to the Poisson distribution:

$$P[N(t+T) - N(t) = k] = \frac{e^{-\lambda t}(\lambda t)^k}{k!}. \tag{1.25}$$

In the QTRNG scheme, depicted in Figure 1.14, a counter is utilized to measure time bins

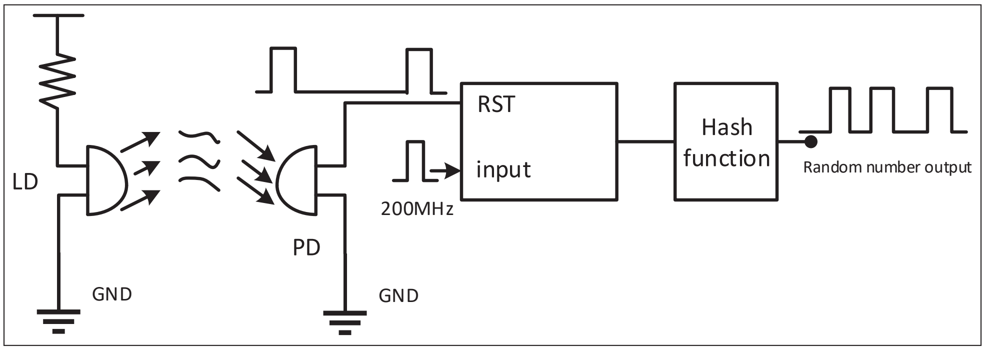


Figure 1.14 Quantum true random number generator

between two consecutive successful photon arrivals (De Jesus Lopes Soares, Alencar Mendonca & Viana Ramos (2014)). Postprocessing is required to ensure high quality randomness for the extracted bit sequence. This process is done with a so-called randomness extractor using a hash function. Different algorithms can be applied to perform the hash function. Three categories of randomness extractors can be distinguished: linear correction, non-linear correction, and Von-Newman correction (Lacharme (2008); Dichtl (2007)). To reduce the amount of post-processing in QTRNG, a method was presented in Tadić, Goll & Zimmermann (2017) which shapes the photon flux emitted from the laser diode sources to have time dependence in the form of $(1 - t/T)^{-}1$, where T is a controllable time constant, and t is the independent time variable. Thus the current controlling laser diode should have time dependence in the form of $(1 - t/T)^{-}1$.

Many researchers attempted to enhance the throughput of random bit sequences by using multiple detectors instead of using a single-photon detector. For example, the research of Patel *et al.* (2012) proposed an avalanche photodetector (APD) with a driving frequency of 1 GHz providing a highest throughput of 340 MHz. Using multiple APD arrays of $[128, 512]$, Samuel *et al.* (2013) achieved the maximum throughput of 5 Gbps with a consumption of 125 mW, while Massari *et al.* (2016) used a matrix of 16×16 single-photon avalanche diodes (SPADs), and achieved a throughput of 128 Mbps. All of these attempts showed their advantages in specific applications. However, besides their high power dissipation, they are incompatible with the current semiconductor technologies. It is noted that if laser diodes have chaos characteristics that follow uniform distribution, then the final output current also represents chaos characteristics Sunada *et al.* (2012). This is the main motivation of recent research on the development of chaos laser diodes.

The first implementation of chaos laser was proposed in 2008. Since then, chaotic LDs have been regarded as a highly promising source of randomness for ultrafast physical RNGs, with many significant works being reported over the past ten years. Reidler, Aviad, Rosenbluh & Kanter (2009) demonstrated that 12.5 Gbps random bits could be extracted from the first-order derivative of a digitized chaotic signal from a single optical feedback LD by a virtual 8 bit ADC, with respect to its time-shifted version. Further, Wang, Wang & Wang (2016a) enhanced this rate to 400 Gbps using optical-feedback semiconductor lasers as an entropy source. Very recently, the generation rates of RNGs have reached a level of a terabit per second (Tbps) using similar multi-bit extraction (Zhang *et al.* (2017)). The work of Li *et al.* (2018) provided ultrafast photonic RNGs using chaos LDs which demonstrated a 320 Gbps output bit stream.

Despite the potential ultra-high speed advantages, these approaches encounter two main technical obstacles that prevent them from dominating other chaos-based RNGs. The first one is the limited bandwidth of current ADCs compared to the requirements of ultra-high bandwidth chaos laser RNGs. The ADC is a critical component in any mixed-signal processing. The analog bandwidth of most recent ADCs embedded in digital oscilloscopes are limited to the GHz range. Thus, the super high-bandwidth ADC requirement in signal processing limits the popularization of chaos laser RNGs. Secondly, driving the circuit to work at a very high speed gives rise to many challenges, especially in the synchronization phase, such as power dissipation, clock design, slew-rate limitation, and time-shift at pico-scale. To give a sense of the frequency scale to the readers, the RF signal's frequency in 5G cellular networks reaches up to 12 GHz, while the baseband frequency is about 400 MHz.

1.3.4 Discussion

Analog implementations of continuous chaotic systems, especially hyperchaotic systems perform better in terms of dynamic behaviors. Despite these advantages, it is complicated to implement hyperchaotic systems in analog circuit designs. Most of the existing approaches are based on 3D continuous chaotic systems, such as Chua's circuit, Lu's system, and Lorent equations.

Since in cryptographic applications, nonlinear methods for the generation of random keys are highly recommended, digitizing pseudo-chaos-based RNGs using nonlinear chaotic maps, high dimensional chaotic systems, and expanded chaotic spatial systems is practically utilized in high-security applications instead of 1-dimensional linear chaotic maps. Other approaches in pseudo-chaos-based RNGs are digitizing continuous chaotic systems, which will also be covered in this part. The theoretical approach of digitizing a chaotic system is detailed in Kocarev *et al.* (2006). The most important thing in chaotic digitizing systems is that the chaotic characteristics should be preserved as in the original chaotic systems, regardless of the system errors in the finite data representation. The intensive research of Addabbo *et al.* (2007) provided an analysis of Renyi maps in software implementation. Other attempts as (Tang, Man & Chen (2001); Vazquez-Medina, Del-Río-Correa, Rojas-López & Díaz-Méndez (2012); Palacios-Luengas & Duchen-Sainchez (2012)) digitized the simple, linear chaotic maps. However, they lack the mathematical analysis of system errors to show that they can properly be used for cryptographic applications.

In this chapter, we have reviewed the existing RNG solutions, such as chaotic maps, continuous systems, chaos laser semiconductors, and physical entropy sources. Each methodology has its advantages and disadvantages in terms of power consumption, data rate, noise sensitivity, etc. As demonstrated in previous investigations, and analysis, physical entropy sources, like thermal noise, jitter noise, and metastability, are not reliable random sources for high-security applications due to their unknown statistical distributions and limited dynamic ranges. To achieve the high-security requirements of next-generation IoT communication networks, chaos-based RNGs have been developed to overcome these aforementioned limitations. A chaotic system has a deterministic mathematical expression, which is constructed from multiple functional blocks. Therefore, it is less sensitive to system noises. Moreover, a chaotic system exhibits irregular, aperiodic, and noise-like trajectories due to extreme sensitivity to initial conditions. Multiple practical approaches have been reported for generating random bit sequences using chaotic maps; however, they are limited to 1D piecewise chaotic maps, such as Bernoulli maps, tent maps, and zigzag maps, in which the data should be precise within the map region. It is well known that in Bernoulli maps, a small variation in the map parameters causes drastic changes to stable equilibrium points. Moreover, 1D chaotic maps exhibit limited dynamic characteristics as as the signal is stretched exponentially in one dimension. In this context, high dimensional chaotic systems could be applied to extend the dynamic range. Therefore, continuous chaotic systems, which have at least three dimensions, and exhibit rich dynamics, are highly recommended in security and cryptography. The state-of-the-art research on continuous chaotic systems has been limited to some common continuous chaotic systems like Chua's circuit, Lu's system, Lorent's equation, Jerk's equation, etc. Implementations of conventional continuous systems use inductors as energy storage elements, which occupy large spaces in integrated circuits; or use external inductors Ergun & Ozoguz (2005); Toker, Ozoguz, Demirkol, Zeki & Tavas (2009); Gandhi & Roska (2009); Galajda, Guzan & Petrzela (2016); Wannaboon *et al.* (2018). The implementation of continuous chaotic systems using off-the-shelf devices, like the work of Mendoza *et al.* (2016), are totally out of our interest because of their high power consumption and large size which make them unfit for IoT secure devices.

Another limitation of current continuous chaotic system designs is the implementation of non-linear functions. There are commonly used non-linear functions in continuous chaotic systems, like the sign, modules, multiplication, tanh, piecewise, or transconductor, which are either too simple or power-hungry circuit designs.

In conclusion, we found that the design and implementation of chaos-based random number generators have many potentials that are worthy of attention due to the following reasons:

- The chaotic system selection is a critical step in the design of chaos-based RNGs. Currently, many chaotic systems are verified and analyzed mathematically due to the development of simulation tools. We can easily develop and verify novel chaotic systems that are suitable to implement in hardware design.
- The development of microelectronics and downscale in transistor size enable the implementation of complex architectures in integrated circuits. Many complex computing architectures have been developed based on the development of microelectronics redefining the device variability and interconnection Saxena (2018). Therefore, we can define and implement complicated non-linear functions for continuous chaotic systems based on emerging technologies.

1.4 Randomness evaluation

In order to be used in engineering applications, output bits from the RNGs should be tested to verify the level of randomness. The following design metrics should be addressed in evaluating randomness, including the signal entropy (applied to the raw signal or the signal after data post-processing), the signal correlation, the signal histogram and distribution, and the statistical random test such as NIST tests, TestU01, etc.

1.4.1 Entropy

The signal entropy is a commonly used criterion to evaluate signal randomness. There are many kinds of entropy that can be used. In this thesis, we mainly applied Shannon's entropy and the min-entropy. Shannon's entropy of an output random bitstream is defined as:

$$H(X_n) = -\sum_{i=0}^{N-1} p_i \log_2 p_i, \tag{1.26}$$

where N is the number of symbols, and p_i is probability of symbol i. The average entropy per bit is calculated as:

$$\bar{E} = \frac{H(X_n)}{N_b}, \tag{1.27}$$

in which, X_n denotes the chaotic signal and $N_b = \log_2 N$ is the number of bits in the signal. The min-entropy is defined as

$$H_{min}(B_x) = -\log_2[\max_{B_x \subset \Lambda} P_\Lambda(B_x)](\text{bit/symbol}), \tag{1.28}$$

where B_x is a raw binary random variable with probability $P_\Lambda(B_x)$.

1.4.2 Correlation

The correlation, a measure of similarity between two series as a function of the displacement of one relative to the other, is used to measure the mutation of two bitstreams Demir & Ergun (2018-11). The cross-correlation is calculated as:

$$r_{x_1x_2}(k) = \frac{c_{x_1x_2}(k)}{s_{x_1}s_{x_2}} \qquad k = 0, \pm1, \pm2, ... \tag{1.29}$$

where k is the number of time shifts (lag) and $c_{x_1x_2}$ is the cross-covariance coefficient of the time series $x_{1,t}$ and $x_{2,t}$, calculated as

$$c_{x_1x_2}(k) = \begin{cases} \frac{1}{T}\sum_{t=1}^{T-k}(x_{1,t} - \bar{x_1})(x_{2,t+k} - \bar{x_2}) & k = 0, 1, ... \\ \frac{1}{T}\sum_{t=1}^{T+k}(x_{2,t} - \bar{x_2})(x_{1,t-k} - \bar{x_1}) & k = 0, -1, ... \end{cases} \tag{1.30}$$

where s_{x_1} and s_{x_2} are standard deviations of the series $\sqrt{c_{x_1x_1}(0)}$, and $\sqrt{c_{x_2x_2}(0)}$, respectively.

1.4.3 Histogram and distribution

The data histogram and distribution in data space provide visualizable randomness evaluation. The distribution of a data set provides all possible values of a pair of two consecutive data. For a set of samples, if we have similar frequencies of occurrence for the different values, the observed data will have high randomness in statistics. The histogram organizes a group of data points into sub-ranges which are presented in a bar graph. The histogram condenses a data series into an easily interpretable visual representation by taking many data points and grouping them into logical ranges or bins. Highly random data is presented by a flat histogram or a uniform distribution. However, a flat histogram is not a sufficient condition to guarantee randomness, it is only one criterion to compare visually random bitstreams. Therefore, the following standard statistical tests are presented.

1.4.4 Randomness test suites

1.4.4.1 NIST tests

The final binary output bitstreams are evaluated using statistical tests to verify the randomness, according to the well-known test suite NIST SP 800-22 Rukhin *et al.* (2010); Pareschi, Rovatti & Setti (2012). This statistical test works under a tentative assumption of randomness ($\mathcal{H}_0$). The output of each test (the P-value) is computed by comparing some features of the stream to that of an effectively random stream. Tests are designed in a way that, if $\mathcal{H}_0$ is true, the P-value is a random variable uniformly distributed in the interval $[0, 1]$; conversely, if $\mathcal{H}_0$ is false, the P-values collapse to zero. When a single sequence is available, the test interpretation is achieved by comparing the achieved P-value with a small but non-zero threshold value (a typically considered value is 0.01). The sequence is considered random if the P-value is larger than the threshold value, and non-random if smaller. When multiple sequences are available from the same generator, it is also possible to compute all the associated P-values and check for the uniformity of their distribution Pareschi *et al.* (2012).

1.4.4.2 TestU01 tests

TestU01 is another commonly used test-suite providing a collection of utilities for empirical randomness testing of RNGs. TestU01 provides both general and extensive set of software tool for statistical testing of RNGs. It is more flexible in compared to other previous testsuites due to providing software implementation of commonly used RNGs and also allowing to use sequences of real numbers in the interval (0, 1) L'Ecuyer & Simard (2007). TestU01 also provides excessive statistical tests including "Small Crush" (which consists of 10 tests), "Crush" (which consists of 96 tests), and "Big Crush" (which consists of 160 tests). Because TestU01 requires a large number of samples, RNGs based on analog implementation usually do not to provide these tests as it is a time-consuming process.

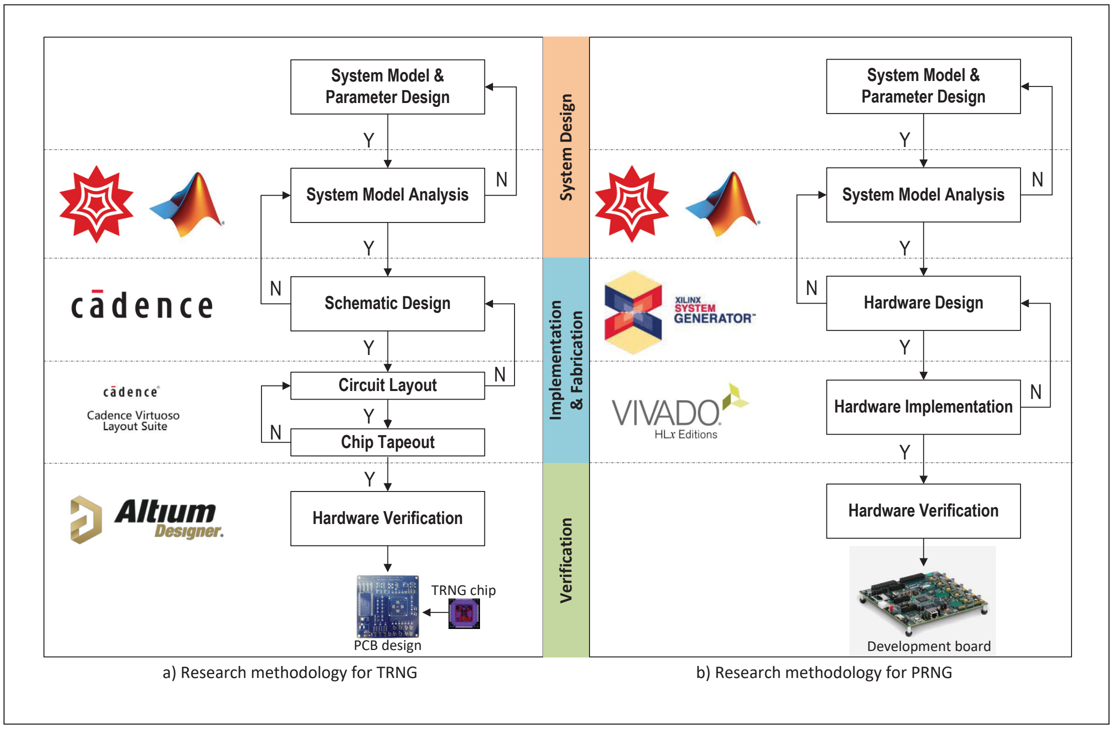
CHAPTER 2

RESEARCH APPROACH METHODOLOGY

This chapter details the research methodology for the research problems. It explains the methods and objectives at each step to achieve the main objectives.

2.1 Hierarchy of research methodology

In this thesis, we aim to design both TRNGs and PRNGs based on chaotic systems. Therefore, there are two different research methodologies adopted, as described in Figure 2.1. This methodology follows the top-down design from system model to circuit implementation. The arrow shows the direction to the next step. If the requirements at one step are not met (N), it is possible to come back to the previous step to optimize the design/ parameters. Otherwise, we continue to the following step (Y).



a) Research methodology for TRNG

b) Research methodology for PRNG

Figure 2.1 Hierarchy of research methodology for a) TRNG and b) PRNG

In the system design, both methodologies employ Matlab and Mathematical tools to simulate and analyze the chaotic systems. The dynamical characteristics are also evaluated in the system model analysis.

To design a TRNG that is implemented in analog circuits, the Virtuoso Analog Design Environment in Cadence Design Toolkit, which provides analog and mixed-signal implementation, simulation, and verification, is used. The circuit schematic and layout are performed in the implementation phase. The circuit design includes several blocks such as integrator, amplifier, multiplier, buffer, etc. Each block design is simulated and verified carefully to meet the design requirements. The whole circuit is also simulated in both schematic and layout implementation steps.

To design a PRNG, the chaotic system is implemented in FPGA by using the Xilinx Design Suite which includes the Xilinx System Generator and the Xilinx Vivado Design Suite. At each step, simulation is carried out to verify the design. The Xilinx System Generator provides visualizable tools which employ the Matlab Simulink to capture the simulation results.

Finally, the fabrication phase is performed. For the TRNGs, the fabrication phase includes chip tape out and design exporting, which is sent to the manufacturer. After receiving the chip, the PCB for validating the chip using the Altium Design Suite is designed and fabricated. For the PRNGs, design synthesis and DRC check are performed in this phase and during the early design stages. Then, the PRNGs are validated in evaluation boards where the outputs are observed in an oscilloscope. Moreover, the validation on the board assist the evaluation of the chip area, the power consumption, the timings.

2.2 System design

First, the chaotic system is chosen or developed from the existing chaotic systems. The system model is analyzed and verified mathematically using some computation software/tools. The system parameters are optimized to suitable with circuit design where the parameter's variation exist (in TRNGs), or to reduce implementation cost (in PRNGs) before going to the next step.

Depending on the system analysis, the energy-boosting techniques and post-processing processes are determined. The phase space, in which all of the possible states of a system are presented, is one of the terms used to evaluate a chaotic system. Each possible state corresponds to a unique point in the phase space. Chaotic systems form distinct shapes in the phase space, which means that the phase plot of a system is a useful tool for identifying chaos. An attractor is a set towards which a variable, moving according to the dictates of a dynamic system, evolves. That is, points that get close enough to the attractor remain close even if slightly disturbed. The attractor is a region in an $N-$dimensional space.

In the system design step, dynamic characteristics are evaluated in term of Lyapunov exponents, Lyapunov dimension, bifurcation diagram, Poincare map, and data trajectories.

2.3 Implementation

In this dissertation, we present hardware implementations for the TRNG and PRNG. For the TRNG, the implementation phase was done in analog circuit design using the 130 nm CMOS TSMC technology. For the PRNG, the implementation phase was done in FPGA chips using design tools from Xilinx. In what follows, we present the introduction and overview of these implementation methods and the reasons why we chose these tools for our designs.

2.3.1 Cadence Design

Cadence Design is a very powerful tool for analog and mixed-signal design. The design flow is centered around Cadence Virtuoso Schematic Editor, Cadence Virtuoso ADE Product Suite, and Cadence Virtuoso Layout Suite. Cadence Design tools working with PDKs from the chip's manufacturers (which has an NDA with the academics) provide a fully design-to-manufacture environment for designers. Cadence Virtuosos ADE Product Suite provides a combination of a GUI-based and script-based environment with multiple circuit simulation modes such as: DC, AC, and transient simulations. The Cadence Virtuoso Layout Suite is used to make the layout of the circuit in compatible with design rule. The capacitance from extracted layout is added to simulation in order to get the post-payout simulation results. The chaotic systems for TRNG are implemented at the circuit level using the 130 nm CMOS TSMC technology. The power consumption, chaotic internal frequency, circuit noises, chip area are some of the design metrics. These design metrics are evaluated in simulation. Moreover, if the chip is fabricated, it is possible to have measurement results.

2.3.2 Xilinx Design Suite

Xilinx Design Suite is suitable for academic projects at universities due to its high supports from design to device programming with multiple tools using Xilinx chips. In this dissertation, due to limited time and high computation requirements for system design, we chose Xilinx System Generator, which provides co-simulation using MATLAB, Simulink, and Xilinx library.

2.3.2.1 Xilinx System Generator

Xilinx System Generator is a design tool in the Vivado Design Suite that uses the MathWorks model-based Simulink design environment for FPGA design. This tool provides standard block designs for Xilinx FPGA chips which are integrated into MATLAB Simulink. The powerful simulation tool in MATLAB Simulink provides visualizable simulation results and facilitates design optimization. Therefore, the design-to-device time is reduced.

2.3.2.2 Xilinx Vivado Design Suite

After synthesizing the design from System Generator to RTL design, the full system, which includes clock signal generator, custom design in the previous step, and inputs/outputs, is integrated and synthesized using Xilinx Vivado Design Suite. The design is exported as an IP core, then it is embedded to block design in Xilinx Vivado Design Suite. A block-design-based project is created in the Xilinx Vivado Design Suite which includes Zynq microprocessor, PRNG, clock and memory control. The Vivado High-Level Synthesis provides power analysis, timing report, and ultilization report.

2.4 Fabrication

The TRNG fabrication using 130 nm CMOS technology follows the semiconductor device fabrication process for typical CMOS devices used in integrated circuit chips that are present in most electrical and electronic devices. Depending on the CMOS technology, the design in schematic and layout must pass the design verification, which includes two phases:

- **Design Rule Check (DRC):** This step is to verify if the designed layout can be manufactured by the fabrication lab with a good yield.
- **Layout Versus Schematic (LVS):** This step is to verify if the layout of the design is functionally equivalent to the schematic of the design.

The entire manufacturing process from start to packaged chip ready for shipment takes approximately 16 weeks. Post-layout simulation and whole chip test before sending to the manufacturer are necessary to ensure the chip is working after fabrication. Following the design flow in CMOS technology, the following final chip tape out checklist should be verified.

- **Clocks:** The TRNG design includes analog and mixed-signals, the clock is used for the comparator and the digital post-processing circuit. The design can use an internal or external clock. The fan-out of clock networks are checked.
- **Power supplies:** The TRNG design includes an analog circuit for a chaotic system and a digital post-processing circuit. Two separate power supplies are used for the analog and digital parts. Moreover, the DC bias points are tested.
- **Layout:** After passing DRC and LVS as required, the metal integrity also needs to satisfy the requirement.
- **I/O and ESD:** All the inputs and outputs have ESD paths to protect from over-voltages. We also have to check if all the inputs designed can handle the expected voltage/current ranges, and all the output designed can drive the anticipated load.

The post-layout simulation is carried to verify the chip before fabrication. The final tapeout procedure follows the following steps:

- Stream out the layout design to GDS file.

- Stream in the output GDS into a new library.
- Check DRC, LVS of the streamed-in library.
- Compress the GDS file and send it to the foundry.

2.5 Verification

2.5.1 Testboard for TRNG chip

PCB design is required for TRNGs in this dissertation. The test-board for the TRNG chip is designed in the Altium Designer Tool. Figure 2.2 shows the 2D view of the test board for the TRNG chip presented in Chapter 4. In this test senarior, we provide external DC voltages for the initial conditions of the chaotic system. A micro-controller is utilized to provide control signals such as reset and enable the TRNG chip.

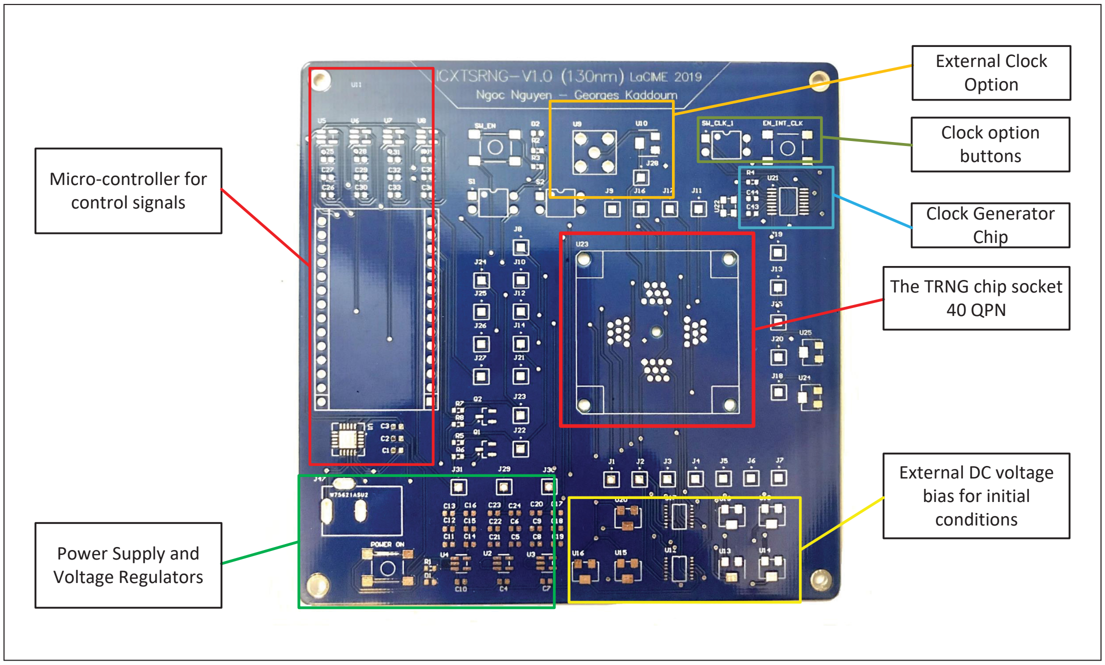

External Clock Option
Clock option buttons
Clock Generator Chip
The TRNG chip socket 40 QPN
External DC voltage bias for initial conditions
Micro-controller for control signals
Power Supply and Voltage Regulators
ICXTSRNG–V1.0 (130nm) LaCIME 2019
Ngoc Nguyen – Georges Kaddoum

Figure 2.2 Testboard PCB design of the TRNG chip

2.5.2 Hardware verification for chaos-based PRNG

The hardware implementation for the PRNG is synthesized in the Xilinx FPGA chip. In this project, we have implemented the system in the Xilinx XC7Z020 chip on the ZedBoard Zynq-7000 Development board and the Xilinx ZC702 Evaluation board. The boards contain the Xilinx XC7Z020 FPGA chip combining a dual-core ARM Cortex-A9. Thus, hardware verification on the FPGA chip and software development on the ARM processor is facilitated on this board. The onboard JTAG protocol is used to program and debug the design via origial JTAG connector or JTAG-USB connector. Several PMOD compatible headers are used to collect and observe the signals on the oscilloscope.

CHAPTER 3

A LOW POWER CIRCUIT DESIGN FOR CHAOS-KEY BASED DATA ENCRYPTION

Ngoc Nguyen[1] , Loan Pham-Nguyen[2] , Minh Nguyen[2] , Georges Kaddoum[1]

[1] Department of Electrical Engineering, École de Technologie Supérieure,
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3
[2] School of Electronics and Telecommunication, Hanoi University of Science and Technology,
1 Dai Co Viet, Ha Noi, Viet Nam

Paper published in « IEEE Access » in June 2020.

3.1 Abstract

Dynamic and non-linear systems have been used to generate random bits in high-security applications for decades. In this perspective, due to their stochastic characteristic, chaotic systems have been emerging as the natural choice for the generation of random bits. This paper presents the design and the implementation of a chaos-based true random number generator and a chaos-key based data encryption scheme for secure communications. The mathematical expression of the dynamic system is presented and analyzed to evaluate the possibility of chaos occurrence. Then, the chaotic system is realized at the circuit level using 130 nm CMOS technology to generate random bit sequences, which are utilized in data encryption. Chaotic signal outputs of the chaos-based random number generator circuit are sampled at a maximum frequency of 50 MHz, enabling a high throughput of random bits. The core of the chaotic circuit consumes 630μW in static mode and a maximum of 660μW in running mode. The chaos-based one-time pad encryption scheme using the chaos-key generator shows the advantages of using this random number generator in secure communications. In this context, the data secrecy is compared to the advanced encryption standard AES128. Moreover, the design is simulated in different working conditions such as voltage supply and temperature variations, where it is shown that the random bit output benefits from a high entropy per bit and passes the standard statistical test suite (NIST) for cryptographic applications.

3.2 Introduction

Random number generators are critical components that are responsible of generating public keys, private keys, and other kinds of random numbers that are utilized in cryptographic applications and data security Lee, Lee, Seo & Yim (2018); Giakoumis *et al.* (2018); Çavuşoğlu, Panahi, Akgül, Jafari & Kacar (2019); Li, Liao & Jiang (2019); Park *et al.* (2019). The explosive growth of Internet-of-Things (IoTs) devices places new challenges on random number generators including energy efficiency, hardware security, and flexibility. Nowadays, true random number generators, taking advantage of the nonlinear and dynamic characteristics of chaotic systems, are attracting substantial research interest Demir & Ergun (2018-11); Tchitnga, Nguazon, Louodop Fotso & Gallas (2016-03); Prousalis *et al.* (2017); Ergun (2018-10); Gong, Zhang, Liu, Sang & Wang (2019). A chaotic system, which is represented by a deterministic expression, is nonlinear and dynamic Demir & Ergun (2018-11). Therefore, it is not sensitive to the system's noise, such as voltage and temperature variations. Despite being deterministic, chaotic systems have been emerging as the natural choice for random bit generators in high-security applications, due to their sensitivity to initial conditions and irregular motion in the phase space. Therefore, at a certain point of observation, the chaotic system behaves as a random-like process Gong *et al.* (2019). Chaotic systems used in generating random bits are categorized into discrete and continuous systems. The discrete chaotic maps, in which the next state is calculated depending on some previous states and is represented by a map $x_{k+1} = M(x_k)$, include logistic maps, piece-wise affine Markov maps, Renyi maps. Whereas continuous chaotic systems, in which the next state is an integration of the previous states, are regulated by a set of differential equations $X' = F(X)$. Some well-known continuous chaotic systems are Lorenz's equations, Chua's circuit, and Rossler's oscillators. Both discrete chaotic maps and continuous chaotic systems can be digitized to produce pseudo-random numbers. The "pseudo" term refers to random number generators which use deterministic algorithms where the data is represented by a digital word Cho & Miyano (2017); Murillo-Escobar, Cruz-Hernández, Cardoza-Avendaño & Méndez-Ramírez (2017); Rezk, Madian, Radwan & Soliman (2020). Whereas true random number generators are generated from physical entropy sources such as thermal noise, jitter noise, or analog implemented chaotic systems. Although chaotic systems have deterministic expressions, they are considered as true random number generators if the chaotic system is implemented in analog circuit design. Many chaos-based true random number generators have been designed in the state-of-the-art Modeste Nguimdo, Tchitnga & Woafo (2013); Hsueh & Chen (2019); Tsafack *et al.* (2020); Bae *et al.* (2017); Wannaboon *et al.* (2018); Park, Rodgers & Lathrop (2015). Due to quantization errors in the digital domain, the dynamic characteristics of chaotic pseudo-random number generators are limited. Therefore, we focus on the design of a true random number generator based on a chaotic system in the analog circuit.

Many true chaos-based RNGs are induced from linear maps due to their simplified implementation. Among the numerous approaches used to implement linear chaotic maps, Markov chaotic maps are widely used in very-large-scale integration designs for their low implementation cost. Some transforms of Markov chaotic maps such as N-Bernoulli maps, tent maps, zigzag maps, and piecewise affine Markov maps (PWAM) have been utilized to generate random bit sequences Callegari *et al.* (2005); Palacios-Luengas & Duchen-Sainchez (2012); Nejati *et al.* (2012); Callegari, Setti & Langlois (1997); Wieczorek & Golofit (2018). However, in cryptographic applications, linear methods are not recommended for generating random keys. Ngoc et al. proposed an analog circuit to implement a nonlinear logistic map based on fuzzy modeling N Nguyen *et al.* (2018). However, the accuracy of nonlinear modeling maps depends on experience and is limited by circuit parameters.

Although many continuous chaotic systems have been developed and verified by mathematics, due to their high power consumption, low-frequency operation, and low capability of operating at low voltage levels, only a few of them were designed in a highly integrated circuit . Most of them use off-the-shelf electronic devices which are power-hungry circuits Deniz, Cam Taskiran & Sedef (2018-07); Vaidyanathan, Kingni, Sambas, Mohamed & Mamat (2018); Xu, Wang, Iu, Yu & Yuan (2019). Chua's circuit is a well-known implementation of a continuous chaotic system in integrated circuits Elwakil & Kennedy (2000b); Elwakil, Salama & Kennedy (2002-12); Xu & Yu (2009). Moreover, implementations in integrated circuit using external components including inductors, and capacitors which occupy a large area have been conducted in Guo & Zhou (2018-05); Abzhanova, Dolzhikova & James (2018-08). The research in Wannaboon *et al.* (2018) presented a fully CMOS circuit design for random bit generator using Jerk equations. The circuit was implemented in 180 nm CMOS technology. However, the system analysis is not fully addressed and the large capacitors limit the internal oscillator frequency which reduces the random signal throughput.

This work presents a low power circuit design for a chaotic system that fills the gap between theoretical analysis and implementation of chaotic systems for the generation of random bit sequences. A novel 3D continuous chaotic system is proposed with mathematical analysis. The chaotic circuit is implemented in 130 nm CMOS technology with a supply voltage of 1.2V which significantly reduces the power consumption. Moreover, post-processing is implemented to eliminate the bias effect and provide ready-to-use random bits. Therefore, according to the evaluation methodology of random number generators proposed by the German Federal Office for Information Security (AIS-20/31), the proposed random bit generator falls into the PTG.3 class, which can be used in cryptography Petura, Mureddu, Bochard, Fischer & Bossuet (2016). As an application, chaos-based one-time pad cryptography is developed using the proposed chaos-key generator. In one-time pad (OTP) cryptography, the security relies on the randomness of the keys Miyano & Cho (2016); Argyris, Pikasis & Syvridis (2016). The OTP cryptosystem based on the XOR operation uses chaos-based random bit sequences as the OTP codebook. The contributions of this paper include:

- The design of a novel 3D continuous chaotic system which is rich of dynamic characteristics.
- The implementation of a chaos-based true random number generator in 130 nm CMOS technology providing ready-to-use random bits.
- The application of an one-time pad cryptography for image encryption/decryption based on the chaos-key generator.

The rest of this paper is organized as follows. To assess the sytem's robustness, Section 3.3 presents the mathematical analysis of the chaotic system. Section 3.4 details the circuit design using 130 nm CMOS technology to be used in integrated devices. The randomness performance is evaluated in Section 3.5. The OTP image encryption based on chaos-key generation is presented in Section 3.6. Finally, Section 3.7 concludes the paper.

3.3 Proposed 3D continuous chaotic system

This section presents the 3D continuous chaotic system in mathematical expressions. The system dynamics and chaotic complexity are analyzed using Lyapunov spectrums, bifurcation diagrams, wavelet analysis, and stability evaluation.

3.3.1 Continuous chaotic system

The proposed 3D dynamic system is expressed in the canonical form $X' = F(X)$, $X = x_1, x_2, x_3 \in \mathbb{R}^3$, with

$$\begin{cases} x_1' = -(x_2 - x_3) \\ x_2' = -x_3 \\ x_3' = -a \times (x_1 - x_2 + x_3) + f(x_1) \end{cases}, \tag{3.1}$$

where $f(x_1) = b_1 \times \tanh(b_2 \times x_1 - b_3)$ is a non-linear function. The following subsections present the mathematical analysis in terms of chaotic and dynamic characteristics by evaluating the Lyapunov exponents and Kaplan-York dimension. Moreover, the stability analysis of equilibrium points is also an important aspect that we address in this section.

3.3.2 Mathematical analysis

In this part, we present the dynamic characteristics of the dynamic system mentioned above. Based on mathematical analysis, we can evaluate if a dynamic system has a chaotic characteristic and how strong it is. The dynamic system was simulated in MATLAB using the 4^{th}-order Runge-Kutta integration algorithm with a step size of 10^{-4}. The Lyapunov exponents of a differential system are defined as:

$$L_i = \lim_{t \to \infty} \frac{1}{t} \log \frac{\|\partial x_i(t)\|}{\|\partial x_i(0)\|}. \tag{3.2}$$

For $a = 0.3, b_1 = 0.1, b_2 = 80$, and $b_3 = 0.7$, the Lyapunov exponents of the differential system are: $L_1 = 0.0841, L_2 = 0$, and $L_3 = -0.3844$. The initial conditions are choosen as $x_1(0) = 0.01, x_2(0) = 0.01$, and $x_3(0) = 0$. The presence of a positive Lyapunov exponent verifies the unstable characteristic of a dynamic system. Therefore, the dynamic system possibly exhibits chaotic characteristic and thus can be used to generate random bits. The divergence of (3.1) is evaluated based on the following conditions:

$$\begin{aligned} \sum_{i=1}^{3} \frac{\partial \dot{x}_i}{\partial x_i} &= -0.3 < 0, \\ L = L_1 + L_2 + L_3 &= -0.3 < 0. \end{aligned} \tag{3.3}$$

A negative L indicates that the differential chaotic system does not converge. Moreover, the Lyapunov dimension proposed by Kaplan and York Kaplan & Yorke (1979), which represents the chaotic complexity, is defined as:

$$D_L = 2 + \frac{L_1}{|L_3|} = 2.21. \tag{3.4}$$

Table 3.1 Lyapunov exponents comparison

Chaotic system	**L_1**	**L_2**	**L_3**	**D_L**
Zhou & Ke (2017)	0.5698	0	-4.2189	2.1347
Sambas (2018)	0.1704	0	-2.5921	2.0657
Yang & Qiao (2019)	0.2263	0	-1.2263	2.1845
Lü, Chen & Cheng (2004)	0.2532	0	-11.3944	2.0221
This work	0.0841	0	-0.3844	2.2188

Table 3.1 compares the Lyapunov exponents of the proposed 3D chaotic system to previous studies. As evidenced in the table, the proposed 3D chaotic system has the highest Lyapunov dimension which indicates highly dynamic complexity.

We evaluate the robustness of the system parameters by observing the bifurcation diagrams and the Lyapunov spectrums according to the primary parameters b_1 and b_3 as shown in Figure 3.1 and Figure 3.2, respectively. As evidenced in these figures, the chaotic characteristics of the chaotic system are preserved over a wide range of values for b_1, and b_3. The largest Lyapunov exponent remains positive and stable. The bifurcation diagrams also indicate that the proposed chaotic system is sustained in chaos state. In other words, the parameter set $a = 0.3$, $b_1 = 0.1$, and $b_3 = 0.7$ ensures that the dynamic characteristics of the system are preserved, even in the presence of circuit parameter variations.

Wavelet analysis has been proved to be a valuable tool in the study of chaotic systems. Therefore, to study the non-periodicity of the proposed 3D chaotic system, the scale index is calculated based on the inner scalogram of the continuous chaotic signals Benítez, Bolós & Ramírez (2010). The scale indexes are compared to Lyapunov spectrums and bifurcation diagrams as seen in Figure 3.1 and Figure 3.2. The scale index is such that $0 \leq i_{scale} \leq 1$ and it can be used to measure the degree of non-periodicity of the chaotic signal. By definition, the scale index is close to zero for periodic signals and close to one for highly non-periodic signals. According to the spreading values of the scale indexes, the non-periodic behavior of the chaotic system is visualized within a wide range of parameters b_1 and b_3.

3.3.3 Stability of equilibium points

The stability evaluation of the equilibrium is an important step in practical designs supporting the choice of system parameters. The Jacobian matrix of the chaotic system is calculated to evaluate the stability of equilibrium points.

$$J = \begin{bmatrix} 0 & -1 & 1 \\ 0 & 0 & -1 \\ A & a & -a \end{bmatrix} \tag{3.5}$$

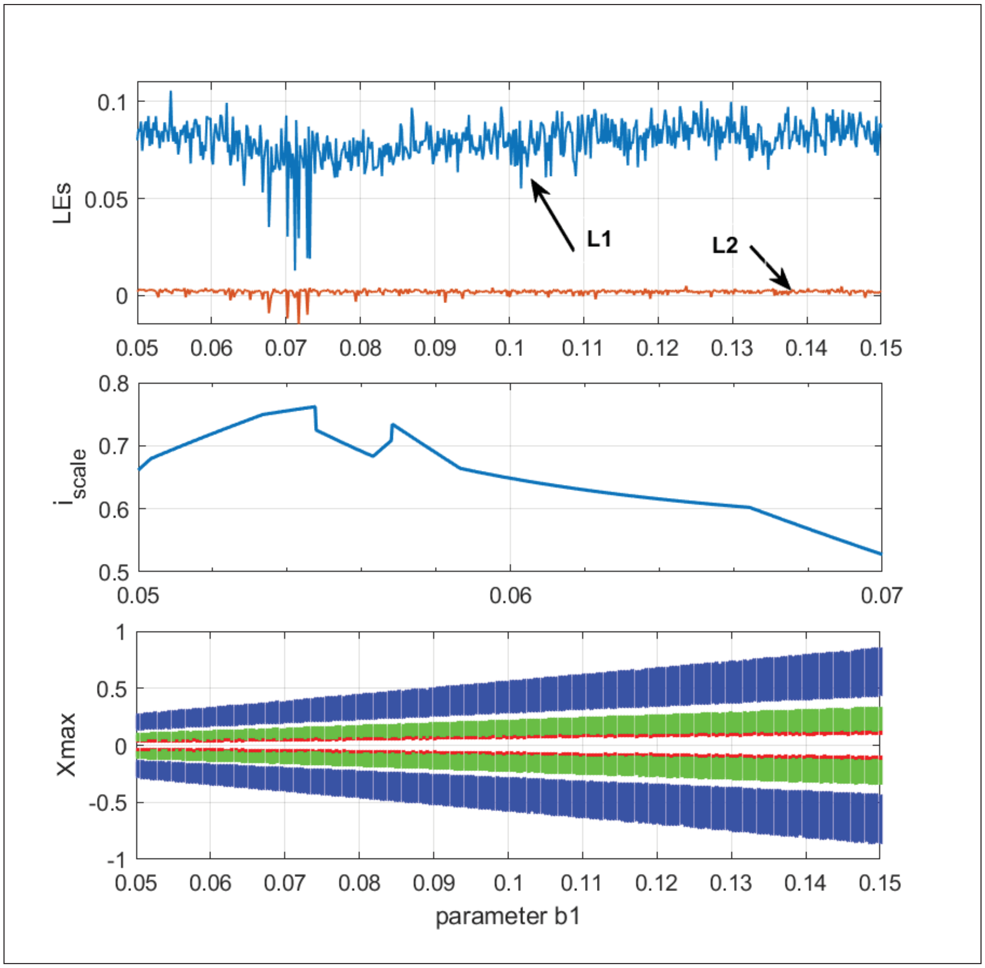


Figure 3.1 Lyapunov spectrums, scale index, and bifurcation diagram according to parameter b_1

where $A = -a + b_1 b_2 (1 - \tanh^2(b_2 x - b_3)))$. The eigenvalues of the Jacobian matrix satisfy the condition:

$$J - \lambda I = 0 \leftrightarrow \lambda^3 + a\lambda^2 + (a + A)\lambda - A = 0. \tag{3.6}$$

The chaotic system has equilibrium at $E = [x_1, 0, 0]$, in which x_1 satisfies:

$$\begin{aligned} -ax_1 + b_1 \tanh(b_2 x_1 - b_3)) &= 0 \\ \leftrightarrow \tanh(b_2 x_1 - b_3) &= \frac{ax_1}{b_1}. \end{aligned} \tag{3.7}$$

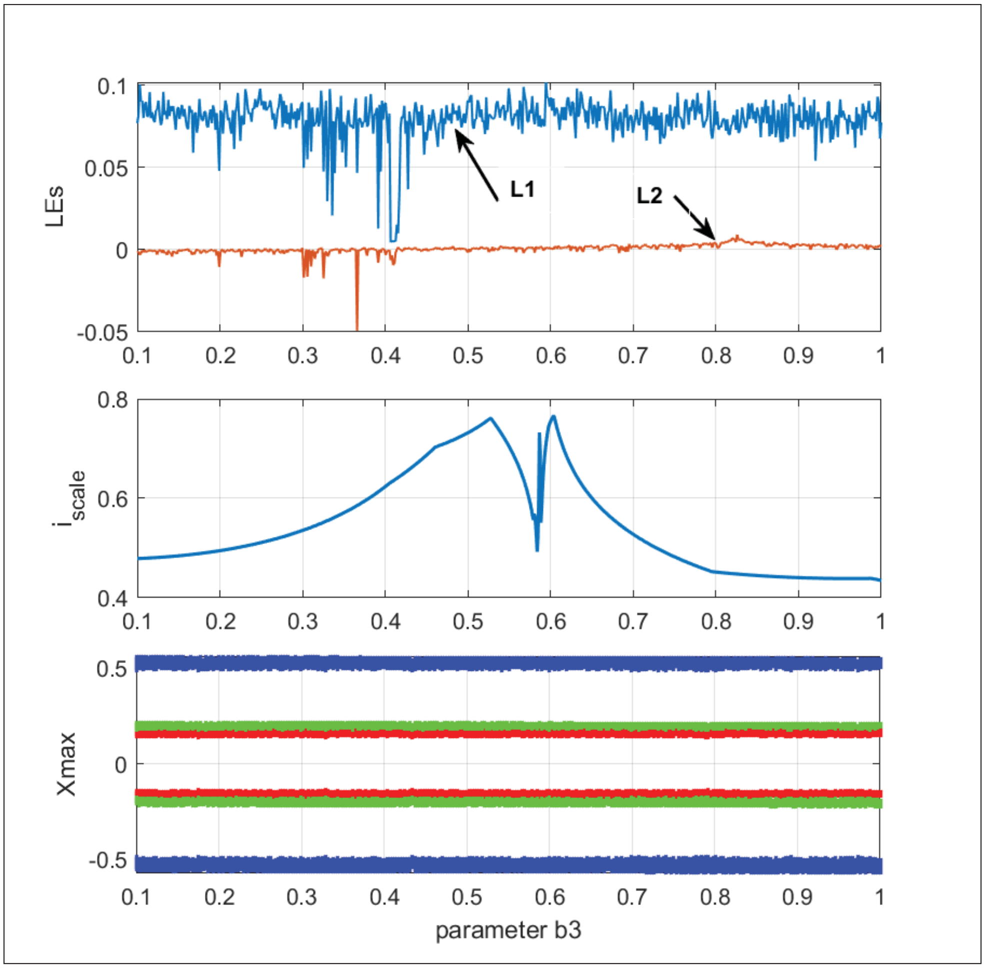


Figure 3.2 Lyapunov spectrums, scale index, and bifurcation diagram according to parameter b_3

For a small perturbation from the fixed points $X(t) = X(0) + \Delta X$. If $\Delta x \approx e^{\lambda}t$, the characteristic polynomial equation is derived as in (3.6). According to the Routh-Hurwitz criterion, the real parts of the roots of (3.6) are negative if and only if $A < 0$, and $a(a + A) + A > 0$. The system at the equilibrium set E must be unstable, thereby the chaotic phenomenon is enabled. The parameters must satisfy one of the following conditions:

$$\begin{aligned} -a + b_1 b_2(1 - (\frac{ax_1}{b_1})^2) &> 0, \\ a^2 + (a+1)(-a + b_1 b_2(1 - (\frac{ax_1}{b_1})^2) &< 0. \end{aligned} \tag{3.8}$$

From (3.7), we determined that the equilibrium of the system is at $E = [0.03167, 0, 0]$. Therefore, the first condition in (3.8) is satisfied. In other words, we prove that the proposed chaotic system has an unstable and saddle focus equilibrium which enables chaotic characteristics. Overall, the above presented chaotic system has non-periodicity in which the data space is limited in the range [-0.5-0.5] as shown in Figure 3.3. This enables the deployment of the proposed chaotic system using CMOS devices with a supply voltage of 1.2 V. In this context, the common voltage is set to 0.6 V.

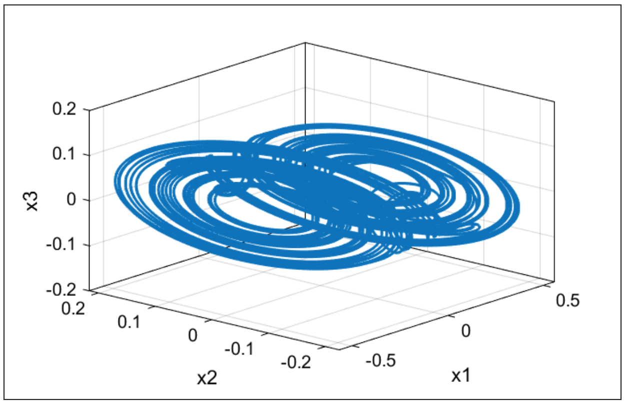


Figure 3.3 3D trajectories of chaotic signals simulated in MATLAB

3.4 Circuit Implementation

In this section, we present the chaotic circuit design for the above differential chaos and the post-processing circuit using a 130 nm CMOS technology.

3.4.1 Chaotic circuit design

The main components of continuous chaotic systems are integrators. In this section, we present the Gm-C integrator circuit design and the non-linear function based on operation amplifier. The chaotic system is formaluated using Kirchhoff's law, and the results reveal the following system of Ordinary Differential Equations (ODEs):

$$\begin{cases} v_x' = -\frac{g_m}{C_x} v_y + \frac{g_m}{C_x} v_z \\ v_y' = -\frac{g_m}{C_y} v_z \\ v_z' = -\frac{g_m}{C_z} v_x + \frac{g_m}{C_z} v_y - \frac{g_m}{C_z} v_z + \frac{f(v_x)}{C_z} \end{cases} \tag{3.9}$$

where C_x, C_y, C_z are integrating capacitors with $C_x = C_y$, and $f(v_x)$ is the nonlinear tanh(.) function. The circuit realization in (3.9) is normalized by the time constant $\tau = \frac{C_x}{g_m}$, then it is undimensioned by an arbitrary voltage V_r as:

$$\begin{bmatrix} \partial V_x & V_x \\ \partial V_y & V_y \\ \partial V_z & V_z \end{bmatrix} = \begin{bmatrix} \frac{\partial V_x}{\partial \tau} & \frac{V_x}{V_r} \\ \frac{\partial V_y}{\partial \tau} & \frac{V_y}{V_r} \\ \frac{\partial V_z}{\partial \tau} & \frac{V_z}{V_r} \end{bmatrix} \tag{3.10}$$

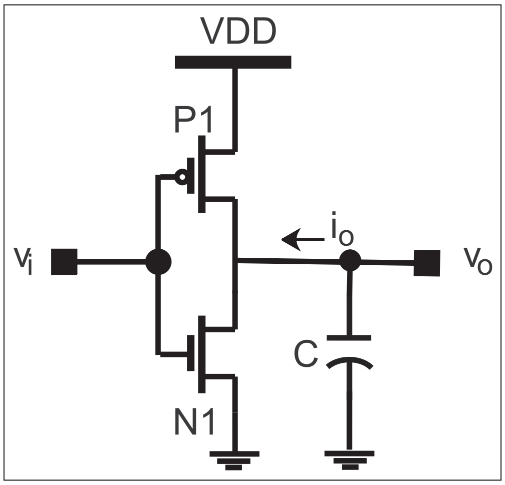


Figure 3.4 Inverted-based Gm-C integrator

The integrator is implemented using an inverted-based Gm-C integrator as depicted in Figure 3.4 Radwan, Soliman & El-Sedeek (2003). The relationship between the output current i_o and the input voltage v_i follows:

$$\begin{aligned} i_o &= -g_m v_i \\ i_o = \frac{1}{C}\frac{\partial v_o}{\partial t} &= -g_m v_i, \\ \rightarrow \frac{\partial v_o}{\partial t} &= -\frac{g_m}{C} v_i. \end{aligned} \tag{3.11}$$

The transconductance gain g_m is comprised of a pair of nMOS and pMOS devices. The total transconductance gain is calculated as:

$$g_m = \mu_n C_{0x}\frac{W_n}{L_n}(V_{GSn} - V_{THn}) + \mu_p C_{0x}\frac{W_p}{L_p}(V_{GSp} - V_{THp}). \tag{3.12}$$

The nonlinear function $i_{out} = f(v_x)$ is a hyperbolic tangent function. The circuit implementation of the non-linear function $f(v_x)$ using a differential amplifier is shown in Figure 3.5. The saturated drain current I_{sat} of a mosfet device is exponential to the gate and source voltages as:

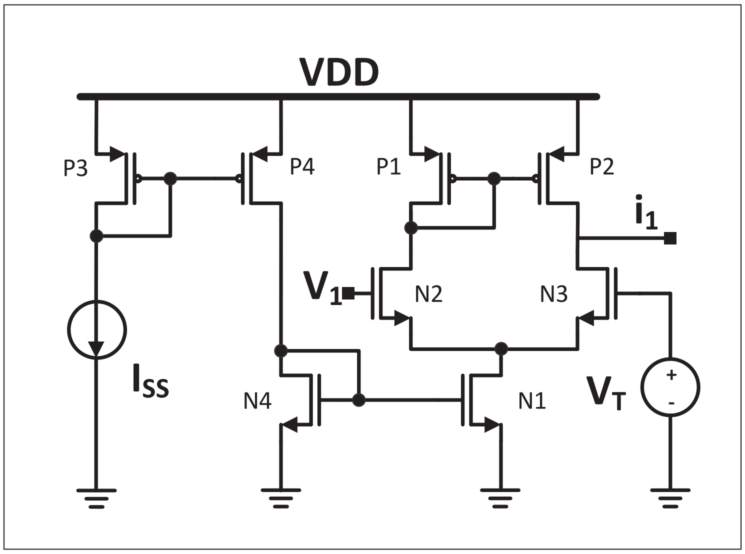


Figure 3.5 Hyperbolic function implementation

$$I_{sat} = I_D e^{\kappa V_G - V_S}. \tag{3.13}$$

Assuming that a differential input pair is saturated, the voltage-to-current transfer characteristic $i_{out} = i_{D1} - i_{D2} = f(v_x)$, is proportional to the difference between two drain currents as follow:

$$\begin{aligned} i_{out} = i_{D1} - i_{D2} &= I_{SS}\frac{e^{\kappa v_x} - e^{\kappa V_T}}{e^{\kappa v_x} + e^{\kappa V_T}}, \\ &= I_{SS}\tanh\frac{\kappa(v_x - V_T)}{2}. \end{aligned} \tag{3.14}$$

The current bias I_{SS} was designed to be resilient against process-voltage-temperature (PVT) variations based on the research in Wang, Tan & Chan (2017-04). Here we match the parameters in the first equation and the circuit parameters as:

$$a = \frac{C_x}{C_z}; b_1 = \frac{I_{SS}C_x}{g_m C_z}; b_2 = \frac{\kappa}{2}; b_3 = \frac{\kappa}{2}V_T. \tag{3.15}$$

The whole circuit design of the chaotic system is elaborated in Figure 3.6. The integrating capacitors set to $C_x = 1.2\,\text{pF}$, $C_y = 1.2\,\text{pF}$, and $C_z = 4\,\text{pF}$. The transconductance g_m in (3.12) is $240\,\mu S$, and the current source for $\tanh(.)$ $I_{SS} = 80\,\mu A$. The circuit design of continuous chaotic systems has a great impact on the intrinsic oscillator frequency. Low-frequency chaotic oscillators limit the final random bit throughput. In our chaotic circuit design, the optimization and tradeoff between circuit parameters including transconductance and capacitors are carefully taken into account. The intrinsic oscillator frequency is $f = \frac{g_m}{2\pi C} = 31.8\,\text{MHz}$, which is indeed a very high frequency with respect to other discrete or integrated solutions.

Continuous chaotic systems implemented using off-the-shelf devices are power-hungry circuits that, due to the large values of passive components , limit the oscillator frequency in the "kHz" range. In Ergun & Ozoguz (2005), the authors used inductor $L = 10\,\text{mH}$ and capacitor $C = 10\,\text{nF}$ in the chaotic circuit, where the maximum oscillator frequency is 830 kHz and the sampling frequency is 19 MHz. In the full CMOS implementation of a continuous chaotic system presented in Wannaboon *et al.* (2018), the chaotic signal is post-processed at a frequency of 50 MHz, but the oscillator frequency is undeclared and is expected to be much lower than in this work due to the large capacitors used (between 10 nF and 20 nF, instead of 1.2 pF).

Trajectories of the chaotic outputs of the chaotic system are shown in Figure 3.7 with control voltage $V_T = 0.7\,\text{V}$. The attractor of the chaotic system is observed and compatible with the simulation results. The layout diagram of the chaotic circuit is shown in Figure 3.9. Continuous chaotic systems have high dynamic characteristics (compared to chaotic maps and chaotic iterations) due to higher dimensional signals and multiple parameters. However, they are also associated to more complicated circuit designs, especially when the existing physical noises can degrade the dynamic characteristics if the chaotic system is too sensitive to its parameters. Our continuous chaotic system is shown to be robust since the dynamic characteristics are preserved with existence of circuit noise and device mismatch.

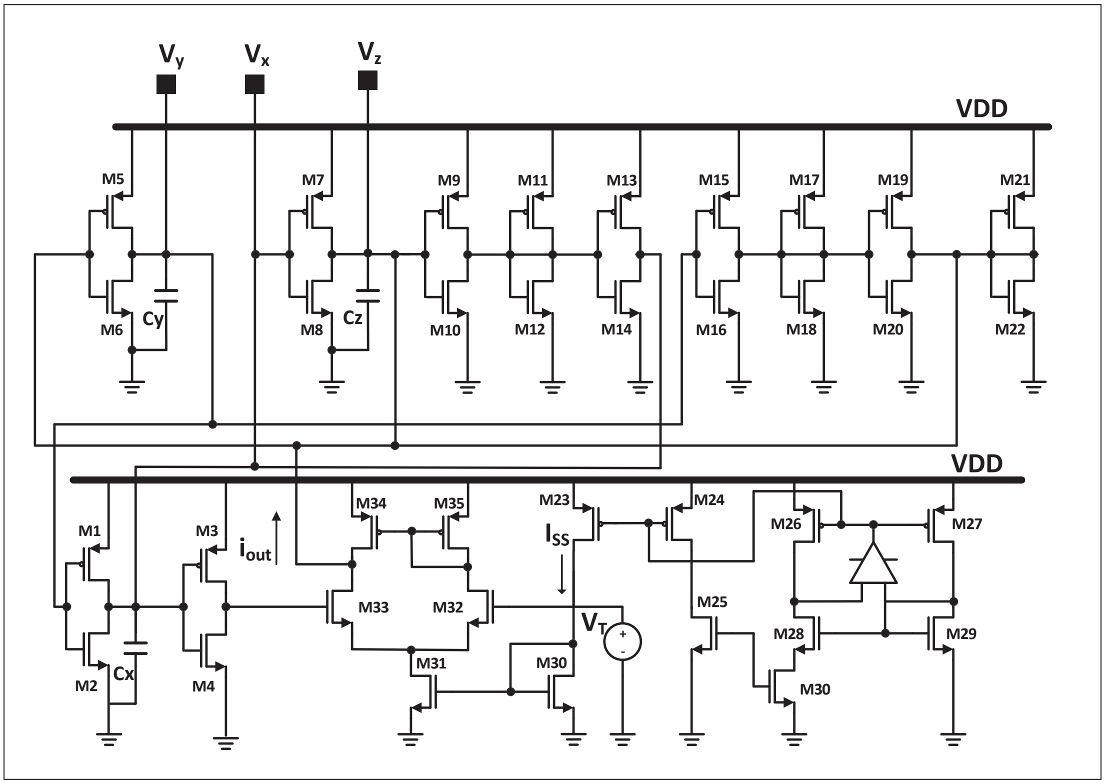


Figure 3.6   Fully CMOS cicuit design of the proposed continuous chaotic system

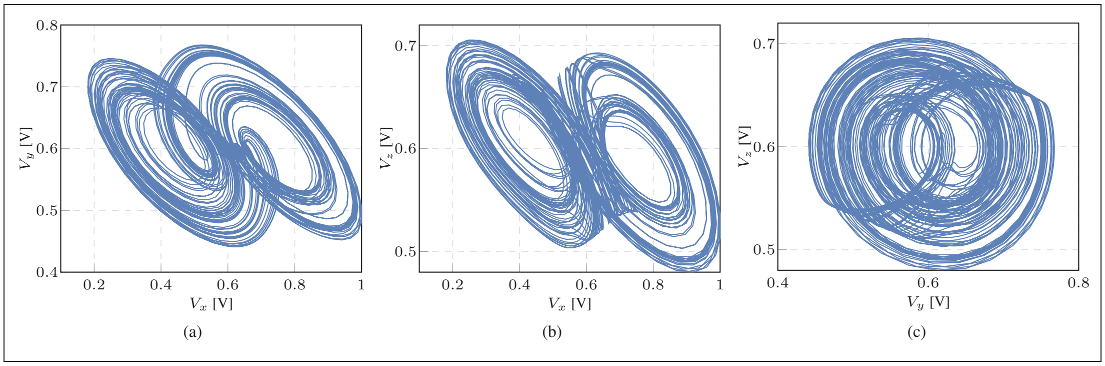


Figure 3.7 2D trajectories with initial state of $V = [0.62, 0.61, 0.6]$

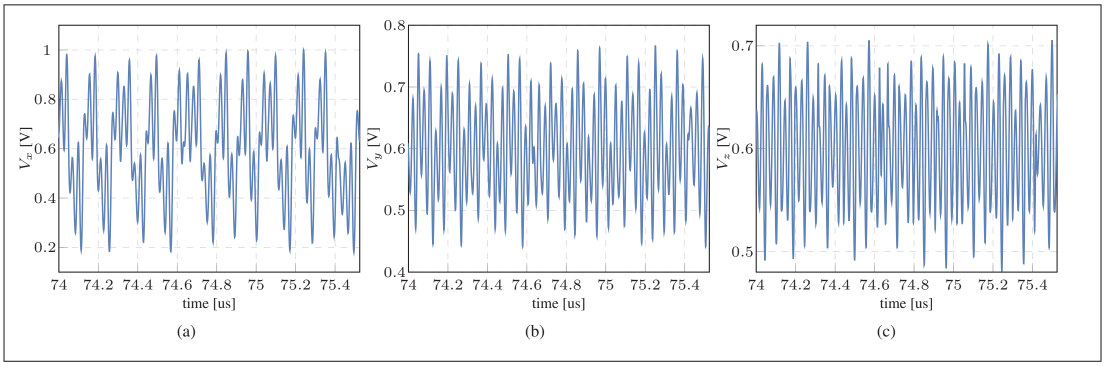


Figure 3.8 Chaotic signal waveform outputs

3.4.2 Post-processing circuit design

3.4.2.1 Comparator

The comparator includes a preamplifier (PREAMP) and a latch circuit (LATCH). The main function of the preamplifier is to provide sufficient gain to overcome the offset of the subsequent comparator without introducing significant offset of its own. The comparator is designed to work at a frequency of 50 MHz; the aperture time is 10 ns with a 50%-duty cycle clock. Random offsets due to transistor mismatches, which may be introduced to the second-stage of the comparator (LATCH circuit), will be eliminated by the following post-processing. Therefore, the comparator circuit does not require a highly critical design. Transistor sizes are chosen

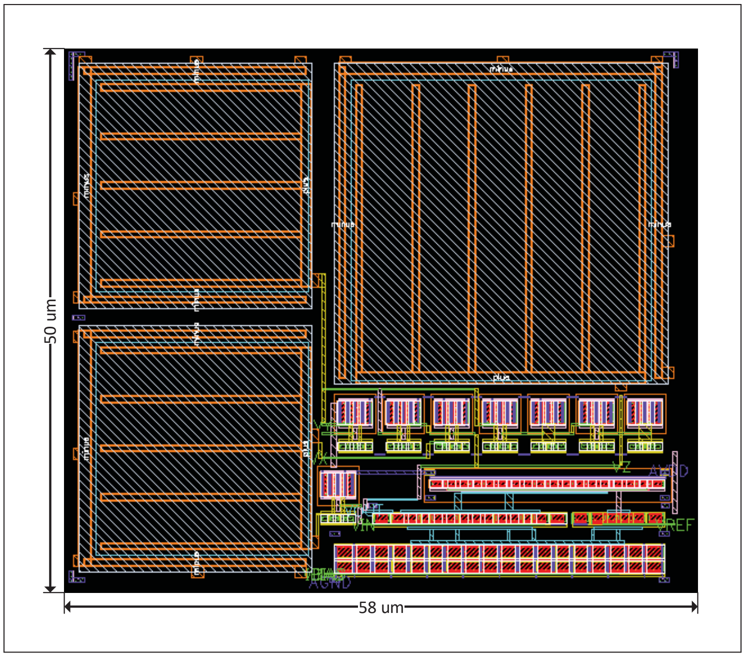


Figure 3.9 Layout diagram of the chaotic circuit design without padding

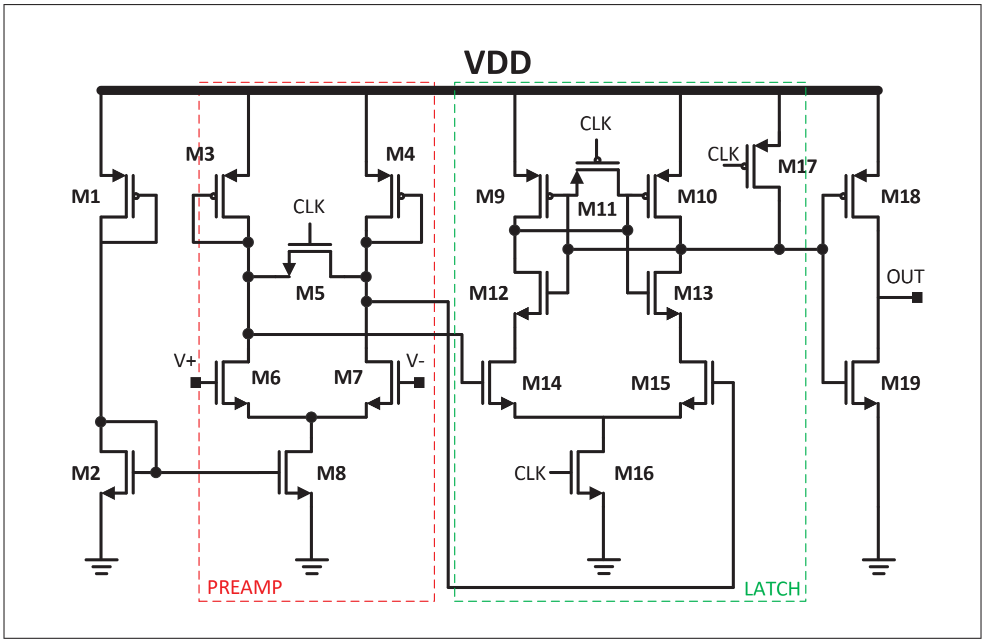


Figure 3.10 Comparator circuit design

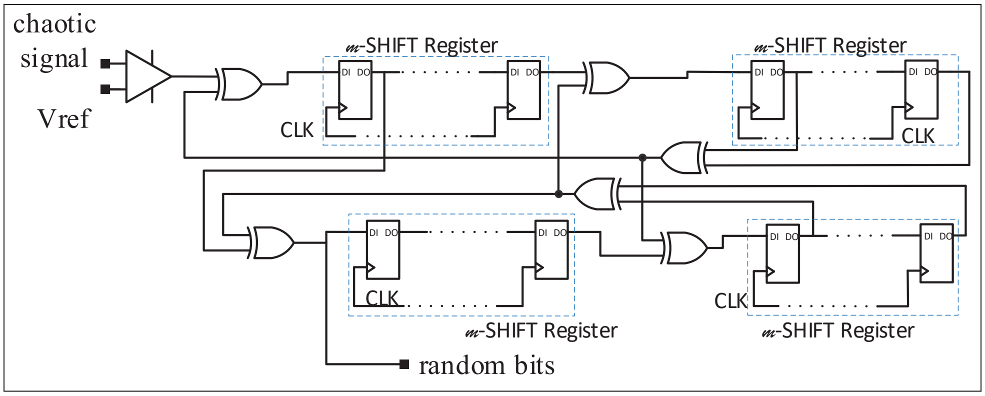


Figure 3.11 SHIFT-XOR based post processing circuit

properly to reduce mismatch and satisfy the gain requirement. The PREAMP circuit, shown in Figure 3.10, uses an output reset switch (M_5), to prevent regeneration during the comparing phase. When the clock signal CLK is at a low level, the dioded-connected PMOS transistors M_3 and M_4 generate the output. The offset of PREAMP is amplified with a high gain. Meanwhile, when CLK is at a high level, the output of PREAMP is fed into an edge-triggered latch (LATCH) that also amplifies its inputs.

3.4.2.2 SHIFT-XOR based post-processing circuit

The circuit design is based on shift and exclusive-OR operations, in which the eliminated bits are re-used by the feedback. Therefore, the bit-rate between input and output is preserved. This post-processing is composed of four shift registers Rozic *et al.* (2016); Pareschi, Rovatti & Setti (2006). The working principle is to evaluate the incoming bits from comparators and reuse these bits by XORing them with the same bit-stream after a few step shifting. In this context, we use 8-bit length registers. The circuit design for a one-bit shift register uses a positive-edge trigger dynamic flip-flop while 8-bit shift registers are composed of eight one-bit shifters. The circuit design is shown in Figure 3.11.

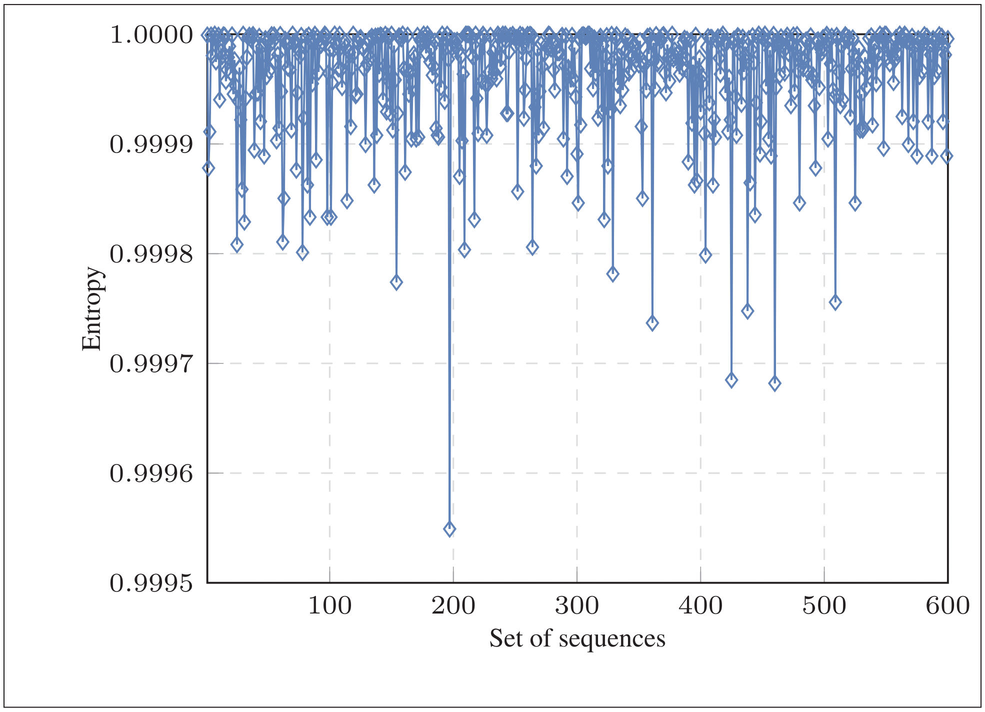


Figure 3.12 Entropy test result of the chaotic circuit design

Table 3.2 Summary of comparison with previous designs

Design	**Entropy Source**	**Techno**	**VDD [V]**	**TP [Mbps]**	**Power [mW]**	**Energy Eff. [pJ/b]**
Kim *et al.* (2017b)**	DT-Chaos	180 nm	0.6	0.27	82(nW)	0.3
N Nguyen *et al.* (2018)*	DT-Chaos	65 nm	2.5	5	0.15	33.33
Wannaboon *et al.* (2018)*	CT-Chaos	180 nm	1.8	50	1.32	26.4
Kim *et al.* (2017a)**	Jitter acc.	65 nm	1.2	9.9	0.418	23.71
Bae *et al.* (2017)**	Meta.	65 nm	1.2	3000	5	1.6
This work*	CT-Chaos	130 nm	1.2	50	0.78	15.6

(*) post-layout simulation results, (**) measurement results.

3.5 Randomness evaluation

The following section presents the system's performance in terms of power consumption and randomness evaluation including signal entropy, signal correlation, and random-test suite. The chaotic circuit consumes 630 μW in static mode and a maximum of 660 μW in running mode. The comparator consumes 120 μW at a 50 MHz sampling frequency. Therefore, the low-power

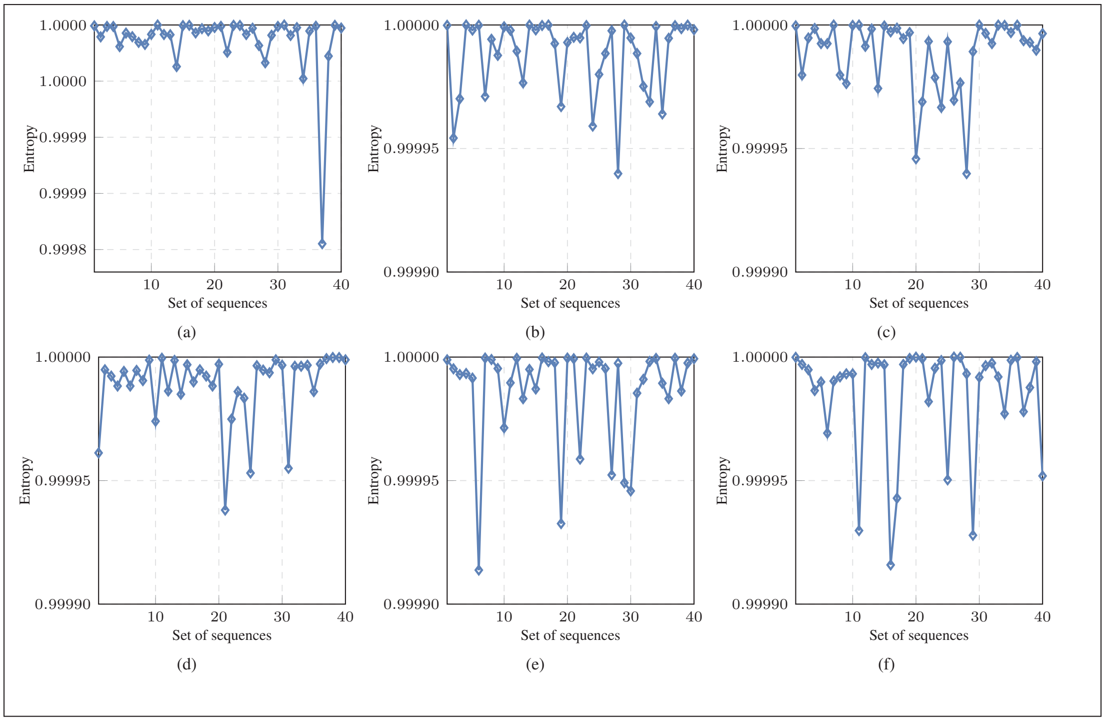


Figure 3.13 Average entropy with multiple corners in Monte Carlo simulations: (a) TT(-20°C,VDD=1.2 V) (b) TT(60°C,VDD=1.3 V) (c) TT(60°C,VDD=1.1 V) (d) SS(60°C,VDD=1.2 V) (e) SF(60°C,VDD=1.2 V) (f) FS(60°C,VDD=1.2 V)

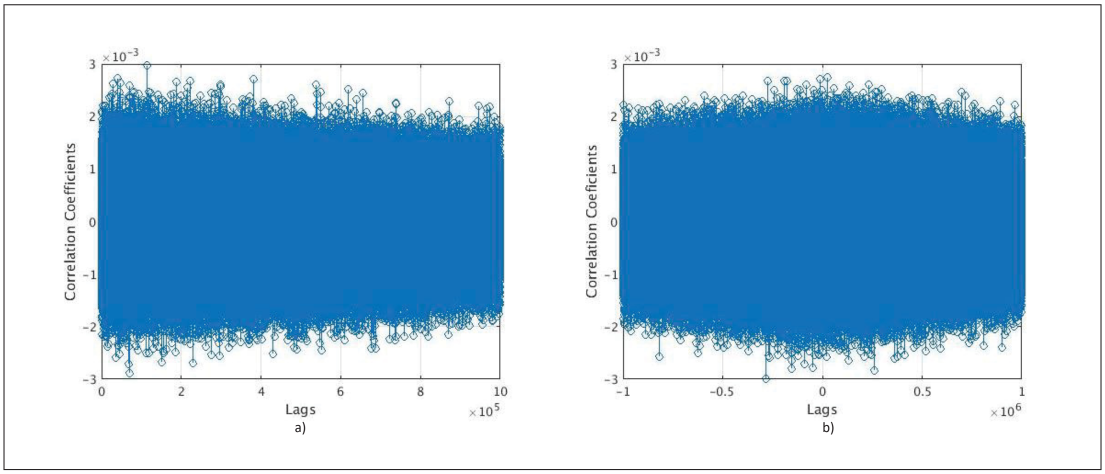


Figure 3.14 Correlation measurement of two different time series a) auto-correlation and b) cross-correlation

circuit design achieves an energy efficiency of 15.6 pJ/b for random binary outputs. The proposed system's performance is compared to other previous designs in Table 3.2. The comparison indicates that our proposed design benefits from a low power consumption with high throughput compared to other chaos-based RNGs using different continuous chaotic systems or discrete time chaotic maps in N Nguyen *et al.* (2018); Wannaboon *et al.* (2018). Although the design in Kim *et al.* (2017b) consumes less power, the data rate is limited at "kHz". Besides, we compare our work with previous generators using physical noises Bae *et al.* (2017); Kim *et al.* (2017a). The work in Bae *et al.* (2017), which uses metastability and jitter noise to generate random bits, has the highest throughput. However, it consumes much more power than our design due to circuits for calibration and elimination of systematic noises. Therefore, it is complicated to reduce the power consumption to suit tiny devices. Thus, the proposed system design is more suitable for applications that require a relatively high throughput (Mbps) and low power.

To evaluate the randomness, Shannon's entropy of the output bitstream is calculated as follows:

$$H(X) = -\sum_{i=0}^{N-1} p_i \log_2 p_i, \tag{3.16}$$

where p_i is the probability of a given symbol and N is the number of symbols. The ideal entropy of a binary bitstream is unity. The entropy test for 600 sets of 60 Kb length binary sequences is illustrated in Figure 3.12. It is observed that the entropy of all the sets is near unity, and 98% of the samples have an entropy higher than 0.9998, which indicates a highly random performance. Moreover, the design was evaluated in different working conditions such as power supply and temparature variations. As illustrated in Figure 3.13, the average random bit entropy does not show significant changes with power variation at different corners in Monte Carlo simulations.

The correlation, which measures the similarity between two time-serial sequences, is another important standard function to evaluate the randomness. The correlation is calculated as:

$$r_{y_1 y_2}(k) = \frac{c_{y_1 y_2}(k)}{s_{y_1} s_{y_2}} \qquad k = 0, \pm 1, \pm 2, ..., \tag{3.17}$$

where $c_{y_1y_2}$ denotes the cross-covariance of the time series $y_{1,t}$ and $y_{2,t}$, which is calculated as:

$$c_{y_1y_2}(k) = \begin{cases} \frac{1}{T}\sum_{t=1}^{T-k}(y_{1,t} - \bar{y_1})(y_{2,t+k} - \bar{y_2}) & k = 1, 2, ... \\ \frac{1}{T}\sum_{t=1}^{T+k}(y_{2,t} - \bar{y_2})(y_{1,t-k} - \bar{y_1}) & k = -1, -2, ... \end{cases} \tag{3.18}$$

where s_{y_1} and s_{y_2} are standard deviations of the series $\sqrt{c_{y_1y_1}(0)}$ and $\sqrt{c_{y_2y_2}(0)}$, respectively, and k is the number of lags (time delays). Likewise, the auto-correlation measures the similarity between the time series and its k-lags delay. Figure 3.14 shows the auto-correlation and cross-correlation measurements of the proposed output bitstreams. The average similarity is of 0.13% which demonstrates the uncorrelated relationship between two different time series and the non-periodic random outputs.

Moreover, the ready-to-use random bits after the data post-processing must satisfy the randomness criteria, measured by statistical tests, to determine that the proposed chaotic system can be used as a random source. The highly-acceptable statistical test suite NIST sp800-22, with numerical tests developed by the National Institute of Standards and Technology, is utilized to evaluate the randomness of binary sequences. This statistical test works under a tentative assumption of randomness ($\mathcal{H}_0$). The output of each test (the P-value) is computed by comparing features of the stream to those of an effectively random stream. Tests are designed in a way that, if $\mathcal{H}_0$ is true, the P-value is a uniformly distributed random variable in the interval $[0, 1]$; conversely, if $\mathcal{H}_0$ is false, P-values collapse to zero. When a single sequence is available, the test interpretation is achieved by comparing the achieved P-value with a small but non-zero threshold value (a typical considered value is 0.01). The sequence is considered random if the P-value is larger than the threshold value, and non-random when smaller. When multiple sequences are available from the same generator, it is also possible to compute all the associated P-values and check the uniformity (χ^2) of their distribution which is known as the second-level test Rukhin *et al.* (2010).

Eighty streams of 1-Mb length are directly used for the sub-tests in NIST including overlapping test, Maurer's universal statistical test, linear complexity test, serial test, run excursion test, and random excursion variant test which require at least 1-Mb length data. The other tests including the mono-bit test, frequency test within a block, run test, the longest run of one in a block, binary matrix rank test, approximate entropy test, and cumulative sums test use 500 streams of 160-kbit length. The results of the NIST test are presented in Table 3.3. All the tests are passed with a reasonable proportion. The minimum pass rate for the first ten tests is 488 for a sample size of 500 bitstreams. The minimum pass rate for the last five tests is 76/80. Moreover, the random bit streams passed the second-level tests of randomness.

Table 3.3 NIST test result of final output bit stream

NIST SP-800.22 Test	**P-value(**)**	**Result**	**Pass rate**
Monobit test	0.506194	Pass	498/500
Frequency within block test	0.841226	Pass	499/500
Runs test	0.390721	Pass	499/500
Longest run 1's test	0.783019	Pass	496/500
Rank test	0.632955	Pass	498/500
DFT test	0.159022	Pass	496/500
Cumulative sum	0.719747	Pass	498/500
Overlapping template	0.185555	Pass	498/500
Non-overlapping template	0.502011	Pass	Pass(*)
Linear complexity test	0.419021	Pass	489/500
Maurers universal test	0.582678	Pass	79/80
Approximate entropy	0.549978	Pass	80/80
Serial	0.672292	Pass	79/80
Random excursions	0.524457	Pass(*)	80/80
Random excursion variant	0.487512	Pass(*)	80/80

(*): all the sub-tests pass the minimum requirement.
(**): the average value.

3.6 Chaos-Based One-Time Pad Cryptosystem

In this section, we present an OTP image encryption application using chaos-key generation. Security and encryption performance analysis are performed to demonstrate the advantages of using chaos-key in data encryption. Standard images are utilized to test the image encryption. Figure 3.15 shows the OTP image encryption, in which a chaos-key generator is used as an OTP codebook. The chaos-key generator block is implemented using 130 nm CMOS technology to provide an analog chaotic signal $[V_x, V_y, V_z]$. Then, the post-processing circuit, which is

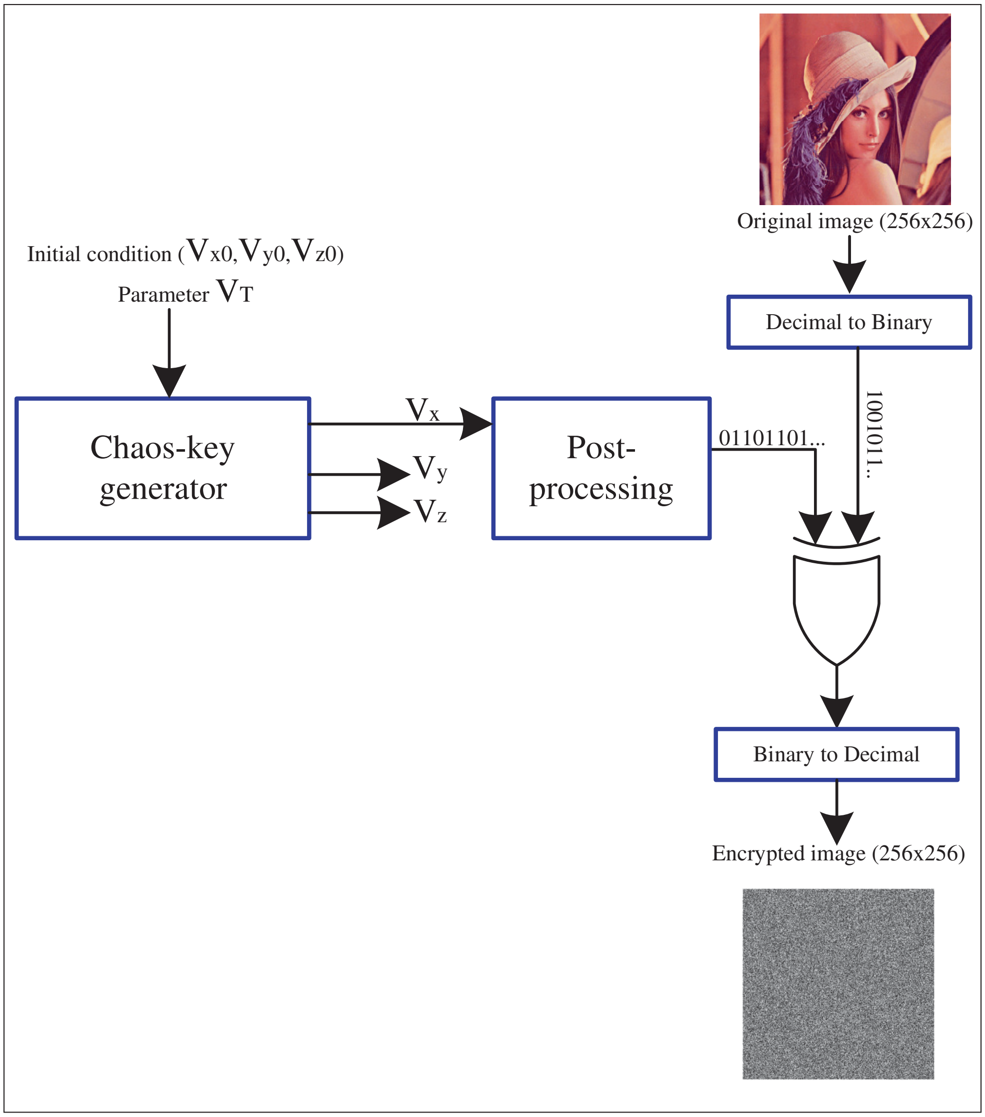


Figure 3.15 Chaos-key generator based image encryption algorithm scheme

also implemented using 130 nm CMOS technology, is used to generate ready-to-use random bits. These random bits are collected and used for the OTP image encryption and decryption in MATLAB. The plaintext then applies an XOR operation with the chaos-based generated random bits. The OTP encryption algorithm is the simplest and least expensive in terms of device resource (uses an XOR operations between the plain image pixel values and generated random bits)Wang, Wang, Zhu & Luo. This application is presented as a proof of concept that the ready-to-use random bits generated from the proposed chaos-based generator can be

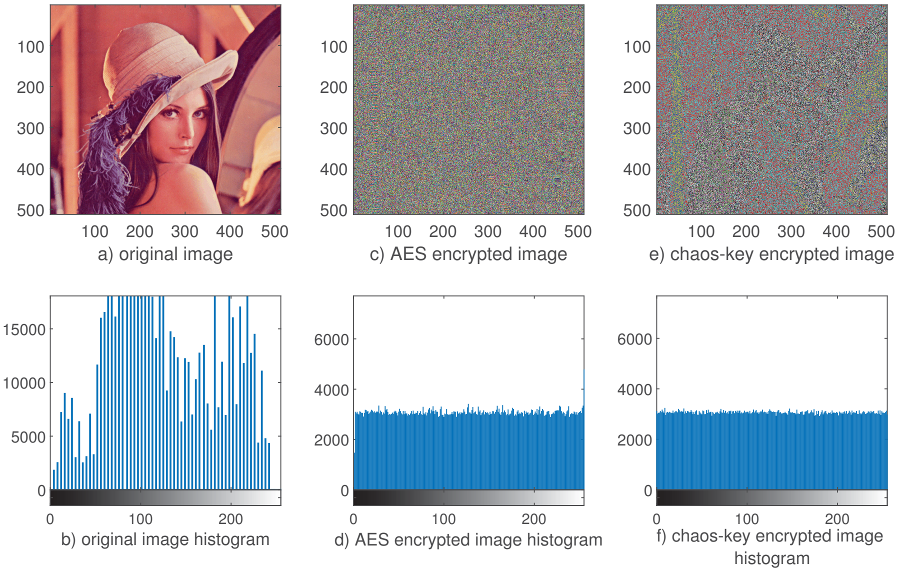


Figure 3.16 Comparison between AES and Chaos-key based image encryption

used directly for secure communications. In this context, the OTP cryptography can use a very-long generated chaos-based random key, which is securely sent to the receiver, for multiple messages until the length of key is reached. The initial conditions and control voltage used to generate chaotic signals define the key space. Different initial conditions and control voltages generate different codebooks for the encryption process. In this context, we assume that the key is securely sent to the receiver over a secure channel which with no transmission errors.

3.6.1 Statistical analysis

3.6.1.1 Histogram analysis

The Advanced Encryption Standard (AES) and chaos-key based OTP cryptography were implemented using MATLAB on an 8th-generation Intel core i7 and 8G RAM computer. Figure 3.16 provides comparison between AES and the proposed algorithm using chaos-key generation for image encryption. As described in these figures, the encrypted images using the AES algorithm and the proposed encryption scheme have a flat histogram. The chaos-key based algorithm finished within 13 s. The encryption time excludes the key generation. For AES encryption, the keys are static and stored in the memory. The proposed chaos-key generation has a throughput of 50 Mbps, therefore, it takes approximately 0.042 s to generate 262144 8-bit chaotic numbers for the sample color image size of $[W \times H] = 512 \times 512$, which is insignificant compared to the encryption time.

3.6.1.2 Image entropy, UACI and NPCR evaluation

Entropy is a statistical measure of randomness that can be used to characterize the texture of an image. In image encryption, the image entropy is used to evaluate the randomness of encrypted image pixel values. The entropy test results, shown in Table 3.4, indicate that the encrypted images have good randomness pixel values. To test the influence of an one-pixel change on the whole image encrypted using the proposed chaos-based algorithm, two common measures are used: Number of Pixels Change Rate (NPCR) and Unified Average Changing Intensity (UACI). Let the grey-scale value of the pixel at grid (i, j) be denoted as $C_1(i, j)$ in the plain image and $C_2(i, j)$ in the encrypted image, the NPCP is defined as follows:

$$\begin{aligned} NPCR &= \frac{\sum_{i,j} D(i,j)}{W \times H} \times 100\% \\ D(i,j) &= \begin{cases} 1 & C_1(i,j) \neq C_2(i,j) \\ 0 & C_1(i,j) = C_2(i,j) \end{cases} \end{aligned} \tag{3.19}$$

The UACI, which measures the average intensity of differences between the plain image and the encrypted image, is defined as follows.

$$UACI = \frac{\sum_{i,j} \|C_1(i,j) - C_2(i,j)\|}{[W \times H] \times 255} \times 100\%. \tag{3.20}$$

Table 3.4 shows the NPCR and UACI test results for image encryption with different image sizes.

Table 3.4 Image entropy, *NPCR*, and *UACI* test results for image encryption

Image	Size	NPCR(%)	UACI(%)	Entropy
cameraman	256 × 256	99.6	31.17	7.997
pool	510 × 383	99.61	39.83	7.999
airplane	512 × 512	99.61	32.66	7.999

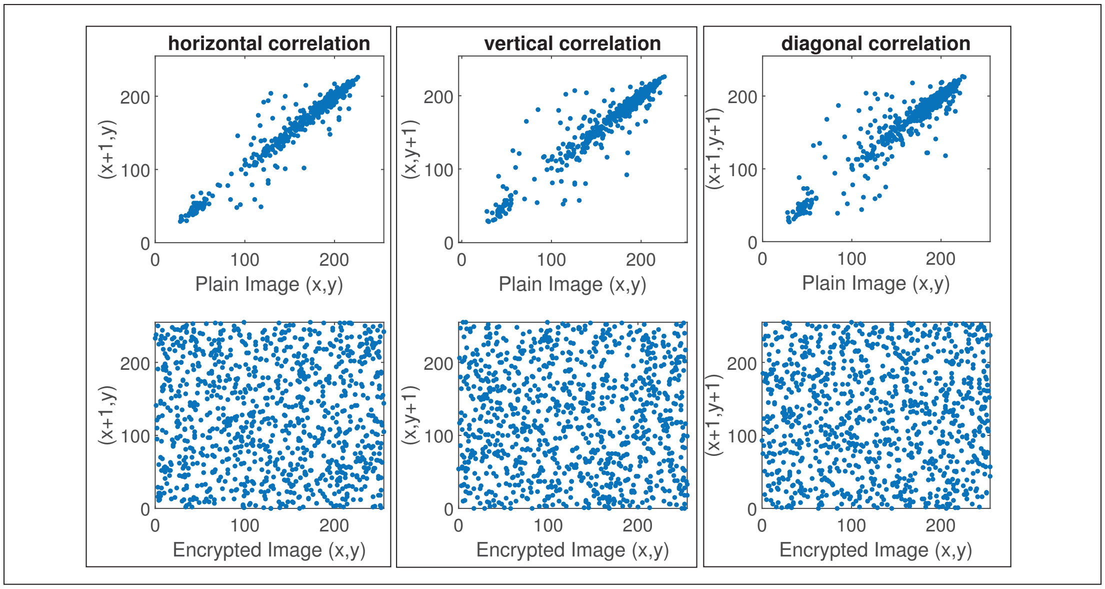


Figure 3.17 Correlation tests of two adjacent pixels in the plain image and chaos-based encrypted image

3.6.1.3 Correlation test

The correlation tests of two horizontally adjacent pixels, two vertically adjacent pixels, and two diagonally adjacent pixels in the plain image and encrypted image using the chaos-key generator

Table 3.5 Correlation coefficients of two adjacent pixels in two images

Dimension	Plain Image	Encrypted Image
horizontal	0.974	0.025
vertical	0.957	0.002
diagonal	0.947	0.006

are presented in Figure 3.17. According to the correlation coefficients listed in Table. 3.5, it can be inferred that the adjacent pixels in the encrypted image are uncorrelated.

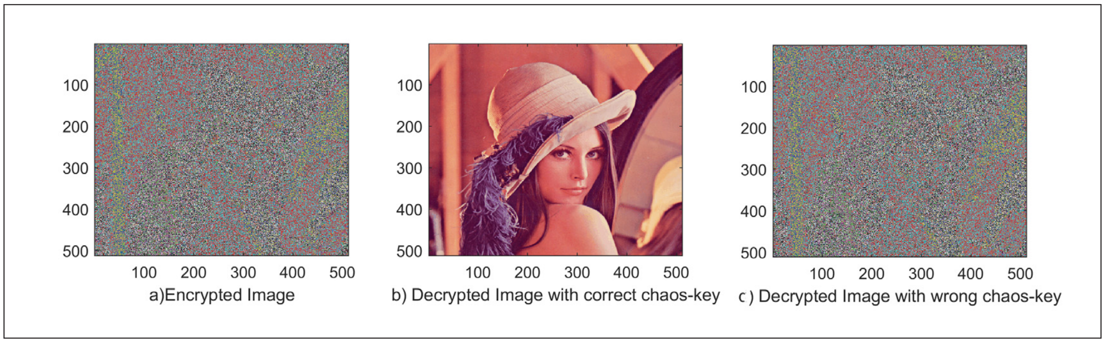

100
200
300
400
500
100 200 300 400 500
a)Encrypted Image
b) Decrypted Image with correct chaos-key
c) Decrypted Image with wrong chaos-key

Figure 3.18 Comparison of image decryption with correct key k_1 and a wrong chaos-key k_2

3.6.2 Key space

Data security should have a big enough keyspace to withstand brute-force attacks. In chaos-key generation, the size of the chaotic system and its parameters define the keyspace. In this application, each parameter has 2^{32} different values. The 3D continuous chaotic system has a key length of 2^{96} because it has a different value set (V_{x0}, V_{y0}, V_{z0}) for initialization. Moreover, the system parameter V_T is also controlled by a 32-bit key length. In total, the chaotic system has a keyspace of 2^{128}.

3.6.3 Key sensitivity

The different generated chaos-keys with different sets of input $(V_{x0}, V_{y0}, V_{z0}, V_T)$ are evaluated. The chaos-key k_1 is generated by the input set $(0.615, 0.6, 0.6, 0.7)$ and the chaos-key k_2 is generated by the input set $(0.61500001, 0.6, 0.6, 0.7)$. Chaos-key k_1 is used to encrypt the original image. Figure 3.18 shows the image decryption with correct chaos-key k_1 and image decryption using wrong chaos-key k_2. A tiny change in the input set caused a huge difference in the decryption results. Therefore, the proposed scheme with chaos-key generation has a high sensitivity to secret keys in the encryption and decryption process.

3.7 Conclusion

We have presented a novel continuous chaotic system implementation in highly integrated analog circuit design and its engineering application to image encryption. The chaotic dynamics were analyzed and studied. The circuit realization in 130 nm CMOS technology enables our design to be used in constrained devices. The generated random numbers passed all the statistical tests of the NIST testsuite at a throughput of 50 Mbps. Moreover, an image encryption scheme using the chaos-key generator was presented. The encryption and decryption performances of the chaos-key based image encryption scheme were compared to the standard AES128 algorithm in terms of data secrecy and accomplishing time. In the future, the chaotic system implementation could be integrated into a microprocessor as a standalone cryptographic processor. Moreover, power and speed optimization will be attempted to increase system performance.

CHAPTER 4

A FULLY CMOS TRUE RANDOM NUMBER GENERATOR BASED ON HIDDEN ATTRACTOR HYPERCHAOTIC SYSTEM

Ngoc Nguyen[1] , Georges Kaddoum[1] , Fabio Pareschi[2,3] , Riccardo Rovatti[3,4] , Gianluca Setti[2,3]

[1] Department of Electrical Engineering, École de Technologie Supérieure,
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3
[2] Department of Electronics and Telecommunications, Politecnico di Torino,
10129, Turin, Italy
[3] Advanced Research Center on Electronic Systems(ARCES), University of Bologna,
40125 Bologna, Italy
[4] Department of Electrical, Electronic, and Information Engineering, University of Bologna,
40136 Bologna, Italy

Paper published in « Nonlinear Dynamics » in October 2020.

4.1 Abstract

Low power devices used in Internet-of-things networks have been short of security due to the high power consumption of random number generators. This paper presents a low-power hyperchaos based true random number generator, which is highly recommended for secure communications. The proposed system, which is based on a fourth-dimensional chaotic system with hidden attractors and oscillators, exhibits rich dynamics. Numerical analysis is provided to verify the dynamic characteristics of the proposed system. A fully customized circuit is deployed using 130nm CMOS technology to enable integration into low-power devices. Four output signals are used to seed a SHIFT-XOR based chaotic data post-processing to generate random bit output. The chip prototype was simulated and tested at 100 MHz sampling frequency. The hyperchaotic circuit consumes a maximum of 980 μW in generating chaotic signals while dissipates a static current of 623 μA. Moreover, the proposed system provides ready-to-use binary random bit sequences which have passed the well-known statistical randomness test suite NIST SP800-22. The proposed novel system design and its circuit implementation provide a best energy efficiency of 4.37 pJ/b at a maximum sampling frequency of 100 MHz.

4.2 Introduction

The security of cryptography algorithms highly depends on the randomness of the keys generated from random number generators (RNGs). Most random number generators available today are software-based, which are commonly referred to as pseudo-random number generators (PRNGs). The term “pseudo-random” refers to the random bits generated from a deterministic algorithm in digital computing software. In this context, the generator knows exactly the next state/the next number while these numbers appear random to the other side. In a true random number generator (TRNG), conversely, the computation of the next state/the next number relies typically on a physical process (entropy source) and is unknown until it is revealed. Therefore, these numbers are “random” for both generator and observer. Some TRNG commercial chips have been utilized in high-performance microprocessors Willie (2010) where an unpredictable entropy source generates a random seed to a pseudo-random generator. Entropy sources include thermal noise, jitter noise, metastability, and chaotic systems. However, these methods have a critical drawback; they are inherited from unknown statistical entropy sources due to the limited dynamic range of the entropy sources and the affection of deterministic system noise sources such as power variations, bias voltage variations, and device mismatches Kim *et al.* (2017a); Yang *et al.* (2014); Bae *et al.* (2017); Petrie & Connelly (2000); Chen *et al.* (2009); Yang *et al.* (2016); Chen *et al.* (2016).

In this paper, we deal with chaos as an entropy source. A non-linear system exhibits chaotic behavior if it features inherent characteristics including (i) high sensitivity to initial conditions – a slight change in initial conditions yields significantly different future trajectories, and (ii) irregular motion in the phase space – phase space trajectories do not converge to a point or a periodic orbit Stojanovski *et al.* (2001). Thanks to these properties, and despite the deterministic evolution, even a small unavoidable uncertainty on the system’s initial condition will make a chaotic system, at a certain point of observation, an actual unpredictable random-like process. Chaotic systems can be classified into discrete-time and continuous-time where both approaches can be effectively used to produce random numbers. In discrete chaotic systems, the iterated functions are used in the form of $x_{k+1} = F(x_k)$. Examples of such systems include the logistic map, the Renyi map, and piecewise affine Markov maps Kocarev *et al.* (2006); N Nguyen *et al.* (2018); Nguyen *et al.* (2018); Wieczorek & Golofit (2018); Vazquez-Medina *et al.* (2012); Palacios-Luengas & Duchen-Sainchez (2012). As demonstrated in García-Guerrero *et al.* (2020), many researchers have been improving the complexity of chaotic maps, where higher dimensional chaotic maps, such as Logistic 2D map, Chen hyperchaotic map, and Rossler hyperchaotic map, have been used. Conversely, continuous chaotic systems are presented by differential equations $X' = F(X)$ Nguyen *et al.* (2020).

Nowadays, continuous chaotic systems achieve higher complexity by conveying integer-order systems into the fractional-order domain Tlelo-Cuautle, Dalia Pano-Azucena, Guillén-Fernández & Silva-Juárez (2020a). However, fractional-order systems are complicated to implement in hardware design due to their memory dependency. The hardware implementation of fractional-order differentiators and integrators requires careful considerations Tolba *et al.* (2017). Here we focus on high-order continuous chaotic systems – chaotic systems with at least four dimensions and two positive Lyapunov exponents (LE) which are implemented in analog integrated circuit design. In a chaotic system, Lyapunov exponents are important criteria to evaluate the system's dynamics. Sensitivity to initial conditions of a dynamic system is represented by a positive LE. An n-dimensional dynamical system has a spectrum of n-Lyapunov exponents. In order to exhibit chaos, a system requires to be at least three dimensional (3D) with a positive LE. A hyperchaotic system exhibits rich dynamics since system states are expanded exponentially in several directions simultaneously. Due to this property, the hyperchaotic system is an interesting candidate for the generation of random keys used in miscellaneous applications in engineering such as secure communications, cryptosystems, and encryptions Teh *et al.* (2016). Moreover, continuous chaotic systems are further classified into two sub-categories according to their dynamic characteristics: self-excited and hidden attractors Tlelo-Cuautle *et al.* (2020a). A nonlinear chaotic system is considered as self-excited if it has a basin of an attractor from an unstable equilibrium point. Lorenz, Rössler, Chen, Lü, or Sprott systems are well-known self-excited systems. Recently, the second group of hidden attractors, which has been developed theoretically and practically, is attracting great attention. The aim of this paper is to introduce a novel hyperchaotic system with hidden attractors suitable for the generation of high-quality random numbers.

4.2.1 Motivations and contributions

As far as continuous chaotic circuits implementation is concerned, numerous contributions have been reported in the literature, all unfortunately presenting drawbacks and limitation, such as high power consumption, low operation frequency, and inability to operate at low voltage levels, which hinders their capabilities to be adopted in practical engineering applications. As an example, Chua's circuit, the first continuous chaotic system implemented in integrated form, requires the use of complicated non-linear functions Elwakil & Kennedy (2000b); Xu & Yu (2009). Limiting ourselves to more recent works, in Trejo-Guerra *et al.* (2012) the authors introduced the first integrated versions of a multi-scroll continuous chaotic oscillator showing 3- and 5-scroll attractors in a $0.5\,\mu$m CMOS technology. They include a very interesting process, voltage, and temperature (PVT) analysis, showing that the desired chaotic behavior is maintained even in presence of consistent parameters variation; unfortunately the circuit topology is still rather complex and the overall circuit is not low-power. In Trejo-Guerra, Tlelo-Cuautle, Carbajal-Gómez & Rodriguez-Gómez (2013), authors compare various integrated circuit design techniques for chaotic oscillators based on with various nonlinear functions (i.e. piecewise linear (PWL), sinusoidal, sawtooth, hysteresis, complex, and $\tanh(\cdot)$ functions). In all these implementations, complexity of the circuit implementing the non-linearity in a issue, as it is the overall charateristic operating frequency of the 5 chaotic oscillators, which is quite low, since one of them works at a 7 MHz frequency using switching currents floating-gate FGMOS transistors, while the others operate between 118 kHZ and 3.5 MHz. The contribution in Carbajal-Gomez *et al.* (2019) is interesting since presents guidelines for the CMOS circuit design of basic building blocks (such as current follower, current mirror, and voltage follower) which are used for obtaining particularly simple saturated nonlinear functions (SNLFs). Finally, in Nguyen *et al.* (2020), some of the authors of this manuscript presented the design of a 3D continuous chaotic system in CMOS technology, and its engineering application in image encryption. The main point in common between the *design* in Nguyen *et al.* (2020) and the chaotic system *implementation* presented in this manuscript is that they both rely on the analog realization of a $\tanh(\cdot)$ nonlinear function. Yet, the work presented here offers several improvements. First, the chaotic circuit is now 4-dimentional, which is a fundamental fact for implementing a TRNG: in fact, a 3D autonomous chaotic system only possess a self-excited attractor whose basin can be revealed by a computational tool Jafari & Sprott (2013), and this may spoil its capability to work effectively as an entropy source. Furthermore, with respect to Nguyen *et al.* (2020), we provide a thorough characterization in terms of robustness with respect to PVT variations, and of the performances of the system as TRNG by including tests on the entropy of raw (i.e., unaltered by the prost-processing stage) generated data.

Although chaotic systems are unpredictable and have random-like state trajectories, they can be studied and recovered by using computational tools. However, this approach showed very limited success with regards to hidden attractor chaotic systems Jafari & Sprott (2013). The hyperchaotic systems with hidden attractors in Bao *et al.* (2018); Prousalis *et al.* (2017); Pham *et al.* (2016b) were proposed to overcome these attacks. However, they are deployed using off-the-shelf analog electronic devices that consume high power and require high voltage operation. Therefore, their implementation is inappropriate in highly integrated circuit designs. In conclusion, based on our investigation, hyperchaotic systems with hidden attractors have many advantages when used in generating random bits for highly secure applications. However, the hardware implementation of these systems still has many limitations that need to be addressed. Therefore, our research targets the shortcomings of practical circuit realizations of hyperchaotic systems. We propose a novel hyperchaotic system with four dimensions and hidden attractors which provides high dynamic characteristics. The proposed hyperchaotic system is presented and analyzed in terms of Lyapunov exponents and stability analysis. Comparison against the state-of-the-arts establishes the advantages of the proposed system. Moreover, the proposed system is implemented in a low power integrated circuit using 130 nm CMOS technology. To generate the ready-to-use binary bitstreams, the proposed chaotic signals are utilized to feed a SHIFT-XOR based post-processing circuit. Multiple configurations are evaluated to find the best frequency operation while the randomness is guaranteed. Statistical tests prove the reliability of using the proposed random number generator in information security. The paper contribution can be summarized as follow: (i) design of a novel hyperchaotic system with hidden attractors, which is highly recommended in security and (ii) circuit implementation of the proposed system using 130nm CMOS technology.

The rest of this paper is organized as follows. Section 4.3 presents the mathematical model of the proposed hyperchaotic system with numerical analysis to verify the robustness of the system. The circuit implementation using 130 nm CMOS technology is elaborated in Section 4.4. The system performances such as randomness measurement, signal entropy and correlations, power consumption, and throughput are evaluated in Section 4.5. Finally, Section 4.6 concludes this paper.

4.3 System design and mathematical analysis

This section presents the proposed hidden attractor hyperchaotic system, which is expressed by four differential equations, depicted in 4.1. The theoretical analysis is divided into two parts. The first part presents the proposed chaotic system and the theoretical study of its chaotic characteristics while the second part addresses the stability of its equilibrium points.

4.3.1 The proposed hyper-chaotic system

The hyperchaotic system is presented in the canonical form $X' = F(X)$, in which the vector $X = [x_1, x_2, x_3, x_4] \in \mathbb{R}^4$, with

$$\begin{cases} x_1' = x_2, \\ x_2' = x_3, \\ x_3' = x_4, \\ x_4' = -a_1 x_3 - a_2 x_4 + b_1 \tanh(b_2 x_1 - b_3) x_2. \end{cases} \tag{4.1}$$

where $\tanh(\cdot)$ is the standard hyperbolic tangent function. The above system can be described as a hyperjerk system which sastifies

$$x'''' = -a_1 x'' - a_2 x''' + b_1 \tanh(b_2 x - b_3) x'. \tag{4.2}$$

In Section 4.4 we discuss the analog implementation of this system. Here we propose its analysis by means of MATLAB numerical integration with the aim of highlighting many properties. As summarized in Tlelo-Cuautle *et al.* (2020a), there are many numerical methods that have been applied to solve differential equations such as Forward-Euler, 4th-order Runge-Kutta algorithm, Adams-Bashfort2, and Adams-Bashfort3. As indicated in Tlelo-Cuautle *et al.* (2020a), the 4th-order Runge-Kutta algorithm provides the lowest error. Therefore, the 4^{th}-order Runge-Kutta algorithm is utilized to simulate the proposed design in MATLAB with a step size of 10^{-4}. The proposed system has equilibrium points only located on the line $E = [x_1, 0, 0, 0]$. Therefore, the proposed system is a dynamical system with hidden attractors. According to Barati, Jafari, Sprott & Pham (2016); Pham *et al.* (2016b), it is impossible to locate the chaotic attractor by choosing an arbitrary initial condition. In other words, from a computational point of view, these attractors are hidden and knowledge about equilibria does not help in their localization. To study the dynamical behavior of the proposed system, we resort to numerical mathematics such as the Lyapunov exponents and bifurcation diagram. The Lyapunov exponents of the system are defined as

$$L_i = \lim_{t\to\infty} \frac{1}{t} \log \frac{\|\partial x_i(t)\|}{\|\partial x_i(0)\|}. \tag{4.3}$$

For $a_1 = 1, a_2 = 0.5, b_1 = 2, b_2 = 2.5$, and $b_3 = 0.62$, the Lyapunov exponents of the novel 4D chaotic system are: $L_1 = 0.088, L_2 = 0.01, L_3 = 0$, and $L_4 = -0.598$, respectively. The initial conditions of the proposed chaotic system are chosen as $x_1(0) = 0.02, x_2(0) = 0.005, x_3(0) = 0$, and $x_4(0) = 0$. There are two criteria to evaluate the divergence of the dynamic system presented in (4.1) as follow:

$$\sum_{i=1}^{4} \frac{\partial \dot{x}_i}{\partial x_i} = -a_2 < 0, \\ L = L_1 + L_2 + L_3 + L_4 < 0. \tag{4.4}$$

Moreover, the Kaplan-York dimension, an effective metric to evaluate the complexity of a chaotic oscillator, is calculated as:

$$D_{KY} = j + \frac{1}{|L_{j+1}|} \sum_{i=1}^{j} L_i, \tag{4.5}$$

where j is the largest index of the positive Lyapunox exponent. In the proposed system, $j = 3$, and the Kaplan-York dimension is therefore:

$$D_{KY} = 3 + \frac{L_1 + L_2 + L_3}{|L_4|} = 3.164. \tag{4.6}$$

Table 4.1 compares the Lyapunov exponents of the proposed 4D hyperchaotic system with hidden attractors to previous studies. Moreover, the proposed hyperchaotic system obtain a higher Kaplan-York dimension than previous systems. The dynamic characteristis of a chaotic system highly depend on the complexity of the non-linear function. The non-linear functions in current state-of-the-art chaotic systems based on a single common function such as multiplication, sign, piece-wise linear function, and tanh function. However, in our proposed hyperchaotic system, the non-linear function includes both multiplication and tanh functions.

Table 4.1 Lyapunov exponents comparison

Chaotic system	**L_1**	**L_2**	**L_3**	**L_4**	**D_{KY}**
Liu & Tong (2016)	2.1990	0.071	0	-14.362	3.160
Pham *et al.* (2016b)	0.0730	0.0018	0	-0.5755	3.130
Prousalis *et al.* (2017)	0.0895	0	0	-0.8997	3.099
Yujun, Xingyuan & Mingjun (2010)	1.416	0.5318	0	-39.101	3.0498
Bao *et al.* (2018)	0.0397	0.0001	0	-0.6395	3.0622
This work	0.1528	0.0661	-0.1723	-0.5465	3.164

4.3.2 Stability analysis of line equilibria

Stability analysis of the equilibrium points helps evaluate the practical design of the system such as the circuit stability and linearity. To evaluate the stability of equilibria, the Jacobian matrix of the proposed hyperchaotic system is calculated as

$$J = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ A & B & -a_1 & -a_2, \end{bmatrix} \tag{4.7}$$

where $A = b_1 b_2 x_2 (1 - \tanh^2(b_2 x_1 - b_3))$, and $B = b_1 \tanh(b_2 x_1 - b_3)$. The eigenvalues of the Jacobian matrix satisfy the condition

$$J - \lambda I = 0 \leftrightarrow \lambda^4 + a_2 \lambda^3 + a_1 \lambda^2 - B\lambda + A = 0. \tag{4.8}$$

The proposed system has equilibrium points only located on the line $E = [x_1, 0, 0, 0]$ where the Jacobian matrix at these equilibria is obtained as

$$J = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & b_1 \tanh(b_2 x_1 - b_3) & -a_1 & -a_2. \end{bmatrix} \tag{4.9}$$

Let $X_0 = [x_1(0), 0, 0, 0]$ be a fixed point, and ΔX be a small perturbation such that $X = X_0 + \Delta X$. If $\Delta x \approx e^{\lambda} t$, the characteristic polynomial equation is derived as

$$\lambda(\lambda^3 + a_2 \lambda^2 + a_1 \lambda - B) = 0. \tag{4.10}$$

Thus, the Jacobian matrix has four eigenvalues where one of them is zero. Let $g(\lambda) = \lambda^3 + a_2 \lambda^2 + a_1 \lambda - B$ be a polynomial function of three non-zero eigenvalues, the real parts of the roots of $g(\lambda) = 0$ are negative if and only if $a_2 > 0, B < 0, a_1 a_2 + B > 0$. For the typical parameter set ($a_1 = 1, a_2 = 0.52, b_1 = 2, b_2 = 2.5$, and $b_3 = 0.55$), to make the equilibrium set E unstable, thereby enabling the possibility of chaos occurance, the initial condition $x_1(0) = c$ must satisfy $c < 0.117834$ or $c > 0.22$. The three nonzero eigenvalues $\lambda_i (i = 1, 2, 3)$ of the equilibrium set E for several typical values of c, are listed in Table 4.2. Depending on the initial value, the proposed system has stable or unstable saddle-focus points. Thus, the dynamical behavior of the equilibrium line chaotic system are heavily dependent on the initial state of the variable x_1, in addition to the system parameters. When $a_1 = 1, a_2 = 0.52, b_1 = 2, b_2 = 2.5, b_3 = 0.6$ are kept constant, the parameter c in the initial conditions $[x_1(0) = c, x_2(0), x_3(0), x_4(0)]$ varies in the range $[0, 0.26]$. The bifurcation diagram of the state variable $x_1(t)$ of the proposed system is shown in Figure 4.1, where it is indicated that the characteristics of the system vary with c. Moreover, the stable region is clearly observed for $0.1178 < c < 0.22$; while for $c > 0.22$ the system is unstable, diverged, and unfolded, therefore, these regions are not interesting. Meanwhile, $0.05 < c < 0.1178$ is a chaotic region with a limited number of periods, and the data space in this region is small. Finally, in the range $0.01 < c < 0.05$, the system exhibits rich dynamic characteristics.

Table 4.2 Stability of equilibria with different initial conditions

c	λ_1	$\lambda_{2,3}$	**Description**
−0.1	−1.3546	$0.1773 \pm 0.9741i$	Unstable state
0	−1.2455	$0.1223 \pm 0.8884i$	Unstable state
0.05	−1.1631	$0.0815 \pm 0.8265i$	Unstable state
0.1	−1.0514	$0.0257 \pm 0.7439i$	Unstable state
0.2	−0.3768	$-0.3116 \pm 0.4099i$	Stable state
0.24	0.1489	$-0.5744 \pm 0.5840i$	Unstable state

The robustness of the system is illustrated by the choice of system parameters. By evaluating the parameter bifurcations and the corresponding Lyapunov spectrum, we choose the parameter ranges in which the characteristics of the proposed system are preserved. Figure 4.1-b,c shows the bifurcation diagrams of state variable x_1 according to the system parameters a_1, and a_2. The variations of b_1, b_2, and b_3 affect the chaotic characteristics of the proposed system as depicted in Figure 4.2. Therefore, we select the parameter set as $a_1 = 1, a_2 = 0.52, b_1 = 2, b_2 = 2.5$, and $b_3 = 0.6$ to determine the chaotic characteristics of the proposed system. The circuit design imperfection and device mismatches contributing to the parameter variations will be evaluated in the next section.

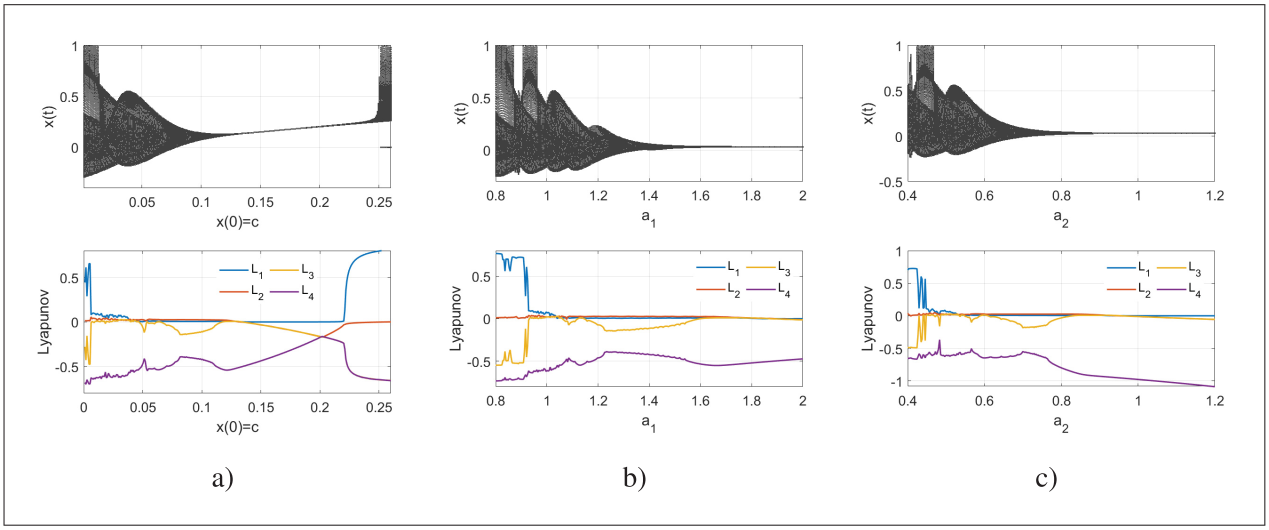


Figure 4.1 Bifurcation diagram and Lyapunov spectrum of signal x_1 according to parameters (a) initial condition $x(0) = c$ (b) a_1 and (c) a_2

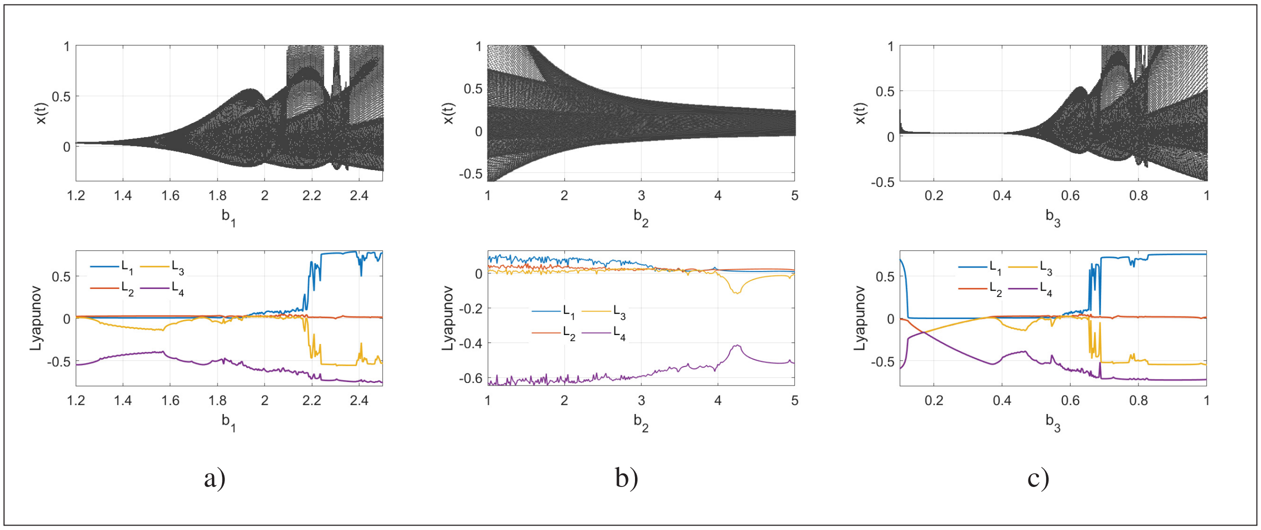


Figure 4.2 Bifurcation diagram and Lyapunov spectrum of the signal x_1 according to parameters (a) b_1, (b) b_2, and (c) b_3

4.4 Circuit implementation

4.4.1 Hyperchaotic circuit design

In circuit realization, the proposed system is formulated using Kirchhoff's law, and the results reveal the following system of Ordinary Differential Equations (ODEs):

$$\begin{cases} v_1' = \dfrac{g_{m1}}{C_1} v_2 \\ v_2' = \dfrac{g_{m2}}{C_2} v_3 \\ v_3' = \dfrac{g_{m3}}{C_3} v_4 \\ v_4' = -\dfrac{g_{m4} f}{C_4} v_3 - \dfrac{g_{m5}}{C_4} v_4 + \dfrac{i_{out}}{C_4}. \end{cases} \tag{4.11}$$

in which, $g_m = g_{m1} = g_{m2} = g_{m3} = -g_{m4} = 110\,\mu\mathrm{S}$ and $C = C_1 = C_2 = C_3 = C_4$ for circuit simplicity. Inverted-based Gm-C configuration is chosen to design the integrator in this circuit due to its low power consumption, high linearity and high input dynamic as depicted in Figure 4.3 Radwan *et al.* (2003). In this figure, the current output i_o is the inverse of the current i_1, where

$$i_1 = -g_m v_i. \tag{4.12}$$

A pair of NMOS and PMOS devices are utilized to provide the total transconductance gain

$$g_m = \mu_n C_{ox} \frac{W_n}{L_n}(V_{GSn} - V_{THn}) + \mu_p C_{ox} \frac{W_p}{L_p}(V_{GSp} - V_{THp}). \tag{4.13}$$

Two couples of devices (N2, N3) and (P2, P3) are used to form bi-directional current mirrors,

$$\begin{aligned} i_o &= -i_1 = g_m v_i, \\ i_o &= \frac{1}{C}\frac{\partial v_o}{\partial t} = g_m v_i, \\ \rightarrow \frac{\partial v_o}{\partial t} &= \frac{g_m}{C} v_i. \end{aligned} \tag{4.14}$$

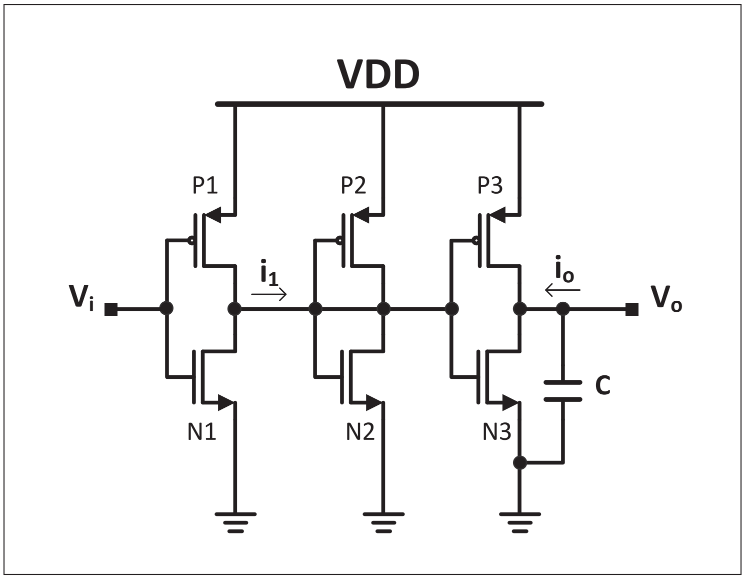


Figure 4.3 Integrator circuit design

The transistor size of P_1 and N_1 are chosen to satisfy $K = \mu_p C_{ox} \frac{W_p}{L_p} = \mu_n C_{ox} \frac{W_n}{L_n}$. Then, the transconductance g_m is approximated as $g_m = K(V_{DD} - V_{THn} - V_{THp})$. The simple circuit of an inverted-based transconductance integrator has a limited dynamic range of input voltage. To drive all the transistors to the saturation region, the voltage headroom is $[V_{THn} \rightarrow V_{DD} - V_{THp}]$. Therefore, low voltage threshold devices are used in the circuit design to increase voltage headroom. The output swing is limited by the overdride voltages of NMOS and PMOS devices from $V_{OD}(NMOS)$ to $(V_{DD} - V_{OD}(PMOS))$. The nonlinearity and variability in the transconductance and the intrinsic capacitor contribute to variations in the DC-transfer function. The transconductance in (4.14) is assumed to be independent of the gate voltages when the transistors are in the saturation region. However, the drain-to-source voltages of the MOSFETs V_{DS} or the output voltage may drive the devices to linear regions when $V_{DS} \leq (V_{GS} - V_{TH})$. Moreover, transistor mismatches in threshold voltages and the transistor parameter K affect the variability of transconductance. A typical mismatch between two physically adjacent transistors is 20%, corresponding to a difference in gate voltage of 10mV which should be taken into account. A circuit of constant Gm can help increase the linearity of an integrator.

The non-linear function $i_{out} = f(v_1, v_2) = i_1 \times i_2$ in (4.11) was implemented using a low-power multiplier Gunhee Han & Sanchez-Sinencio (1998); Satansup & Tangsrirat (2018). The current multiplier was implemented based on current square cells as shown in Figure 4.4. Assuming all transistors are working in the saturation region, the relation between the drain to source current of a transistor (I_D) and the gate-to-source voltage (V_{GS}) is expressed as

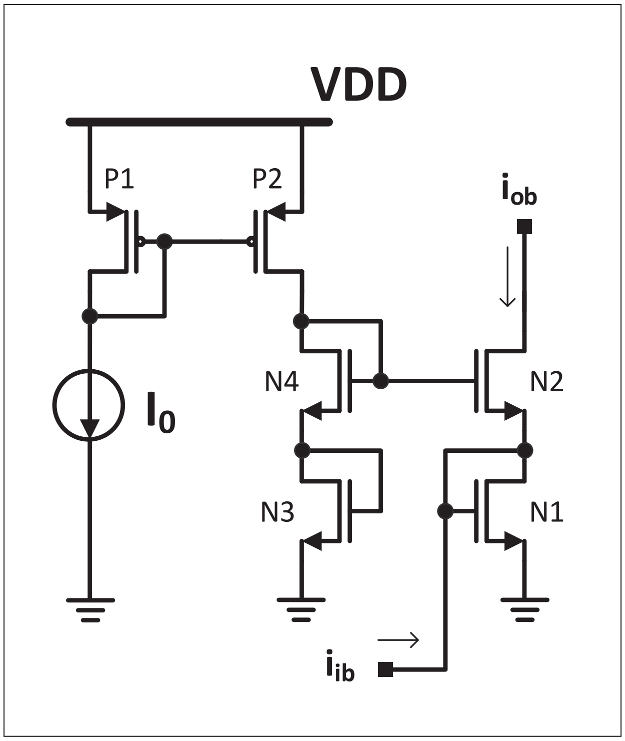


Figure 4.4 Circuit design of current square

$$I_D = K(V_{GS} - V_{TH})^2. \tag{4.15}$$

In Figure 4.4, N3 and N4 are identical diode-connected NMOS devices, therefore the voltage V_B is calculated as

$$V_B = 2V_{GS3} = 2\left(\sqrt{\frac{I_0}{K}} + V_{TH}\right). \tag{4.16}$$

The drain current of N1 is $(i_{ib} + i_{ob})$, therefore the gate voltage of N1 is calculated as

$$V_{GS1} = \sqrt{\frac{i_{ib} + i_{ob}}{K}} + V_{TH}, \tag{4.17}$$

and the gate-to-source voltage of N2 is obtained as

$$V_{GS2} = \sqrt{\frac{i_{ob}}{K}} + V_{TH}. \tag{4.18}$$

Calculating the voltage V_B based on the gate-to-source voltages of N1 and N2, we get

$$\begin{aligned}
&V_B = V_{GS1} + V_{GS2},\\
&\leftrightarrow \sqrt{i_{ob}} + \sqrt{i_{ib} + i_{ob}} = 2\sqrt{I_0},\\
&\leftrightarrow \sqrt{i_{ob} + i_{ib}} = 2\sqrt{I_0} - \sqrt{i_{ob}},\\
&\leftrightarrow i_{ob} + i_{ib} = 4I_0 + i_{ob} - 4\sqrt{I_0 \times i_{ob}},\\
&\leftrightarrow 16I_0 \times i_{ob} = (4I_0 - i_{ib})^2,\\
&\leftrightarrow i_{ob} = \frac{(4I_0 - i_{ib})^2}{16I_0}.
\end{aligned} \tag{4.19}$$

In the current square circuit, various mismatches including channel length modulation, input current mismatch caused by devices mismatch in current mirrors, and transistor mismatches in the circuit in Figure 4.4 introduce current offsets at the output. The current error caused by the input current mismatch decreases with the increase of the input current and depends on the mismatch percentage of the input current which relates to the current mirror mismatch. It can be reduced by choosing large devices in the current mirror. According to Satansup & Tangsrirat (2018), the output current error is half of the current mirror mismatch percentage of the input current. The DC-transfer function of current conveying relies on the assumption that (i) N_3 and N_4 are identical and (ii) N_1, N_2, N_3, and N_4 have identical transistor parameter K. The transistor mismatches caused the input current error, and the threshold voltage mismatches lead to a current offset at the output. Therefore, the transistor sizing should take them into account to reduce the current offset. Large devices are preferred to reduce current mirror mismatches.

The non-linear function $i_1 = f(V_1)$ is a hyperbolic tangent function which is based on a differential amplifier circuit as shown in Figure 4.5. Assuming the MOSFET devices are working in saturation regions, the drain to source current is calculated as:

$$I_{sat} = I_D e^{\kappa V_G - V_S}. \tag{4.20}$$

In this circuit, the current output charges the integrated capacitors. Thus, the current mode is preferred. Driving the differential input pair to saturation, the transfer function of this circuit is proportional to the input offset, $i_1 = i_{D1} - i_{D2} = f(V_{in})$ as

$$\begin{aligned} i_1 = i_{D1} - i_{D2} &= I_{SS} \frac{e^{\kappa V_1} - e^{\kappa V_T}}{e^{\kappa V_1} + e^{\kappa V_T}}, \\ &= I_{SS} \tanh \frac{\kappa(V_1 - V_T)}{2}. \end{aligned} \tag{4.21}$$

The deviations from the ideal behavior of the hyperbolic $\tanh(\cdot)$ circuit derive from the following: transistors mismatch, voltage limitations due to transistors coming out of saturation, and finite slope of the drain curves in saturation. In Figure 4.5, the PMOS devices P_1 and P_2 in the current mirror are not 100% identical which leads to a shift and a difference between the negative and positive asymptotes of the tanh curve. This contributes to the asymmetric geography of the chaotic attractor. The voltage headroom at the output depends on the saturation properties of P_2. Therefore, the drain-source voltage at saturation $V_{OD_{sat}}$ of PMOS P_2 below V_{DD} sets the upper limit, while drain-source voltages at saturation of NMOS N_1 and N_3 above GND constrain the minimum output voltage.

An inverted-based transconductance amplifier was utilized to obtain current $i_2 = g_{m6} v_2$ and three current square cells were employed to construct the multiplier as shown in Figure 4.6, in which $i_3 = i_1 + i_2$, and the current output from multiplier i_{out} is calculated as

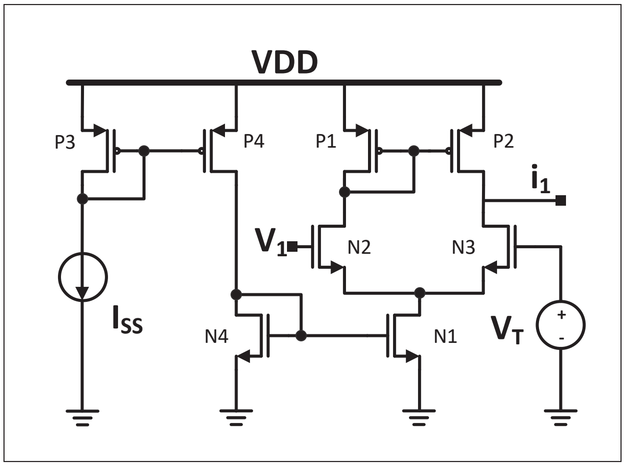


Figure 4.5 Hyperbolic function implementation

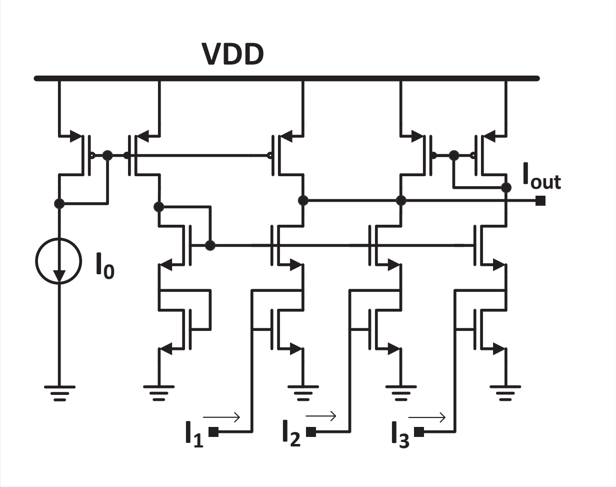


Figure 4.6 Multiplier circuit design

$$
\begin{aligned}
i_{out} &= I_0 + i_{o3} - i_{o1} - i_{o2}, \\
&= I_0 + \frac{(i_1 + i_2 - 4I_0)^2}{16I_0} - \frac{(i_1 - 4I_0)^2}{16I_0} - \frac{(i_2 - 4I_0)^2}{16I_0}, \\
&= \frac{i_1 \times i_2}{8I_0} = \frac{I_{SS} \tanh \frac{\kappa(V_1 - V_T)}{2} g_{m6} v_2}{8I_0}.
\end{aligned}
\tag{4.22}
$$

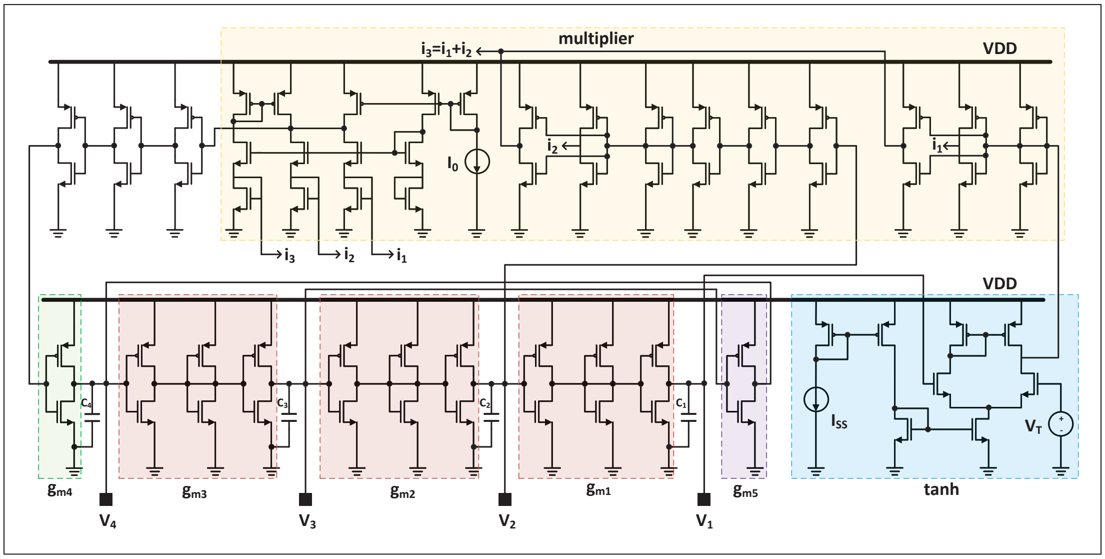


Figure 4.7 Circuit design of the proposed 4D hyperchaotic system

The proposed circuit realization in (4.11) is reformed to the original formulation of the chaotic system in (4.1) by normalizing by the time constant $\tau = \frac{C}{g_m}$, and dimensionless by an arbitrary voltage V_r, respectively as

$$\begin{bmatrix} \partial V_1 & V_1 \\ \partial V_2 & V_2 \\ \partial V_3 & V_3 \\ \partial V_4 & V_4 \end{bmatrix} = \begin{bmatrix} \frac{\partial V_1}{\partial \tau} & \frac{V_1}{V_r} \\ \frac{\partial V_2}{\partial \tau} & \frac{V_2}{V_r} \\ \frac{\partial V_3}{\partial \tau} & \frac{V_3}{V_r} \\ \frac{\partial V_4}{\partial \tau} & \frac{V_4}{V_r} \end{bmatrix} \tag{4.23}$$

The circuit components are chosen to be compatible with the chaotic system parameters as

$$a_1 = \frac{C_1}{C_4}; a_2 = \frac{g_{m5}}{g_m}; b_1 = \frac{g_{m6} I_{SS}}{8 g_m I_0}; b_2 = \frac{\kappa}{2}; b_3 = \frac{\kappa}{2} V_T. \tag{4.24}$$

The hyperchaotic core circuit design is presented in Figure 4.7 using low-voltage devices in 130 nm CMOS technology with a supply voltage of 1.2 V. All the capacitors were specifically chosen as $C = C_1 = C_2 = C_3 = C_4 = 3.2\,\text{pF}$. The bias current in the differential circuit to conduct the tanh(·) function is set to $I_{SS} = 40\,\mu\text{A}$, the voltage input $V_T = 0.65\,\text{V}$, and the current source to $I_0 = 10\,\mu\text{A}$. The intrinsic capacitors of the MOSFETs introduce a variation in the system's parameters of the chaotic circuit implementation in (4.11) g_m/C. According to Figure 4.7, the parasitic capacitors at the MOSFET gates introduce the variations in the integrated capacitors. Therefore, to minimize the effect of intrinsic parasitic capacitors, the devices' sizes are minimized to reduce gate capacitors, which are proportional to $W \times L \times C_{ox}$ (W, L, and C_{ox} denote the device width, device length, and the gate-oxide capacitor per unit area, respectively). The effect of parasitic capacitors is also investigated according to the PVT variations as depicted in Figure 4.8. As seen from this figure, the PVT variations may contribute up to 6% of the integrated capacitor. Despite these effects, the system successfully generates chaotic signals. The initial condition of the chaotic circuit is controlled by the initial voltages of integrated capacitors. External voltages are used to charge the integrated capacitors to provide initial values. Then, the circuit is switched to an autonomous process.

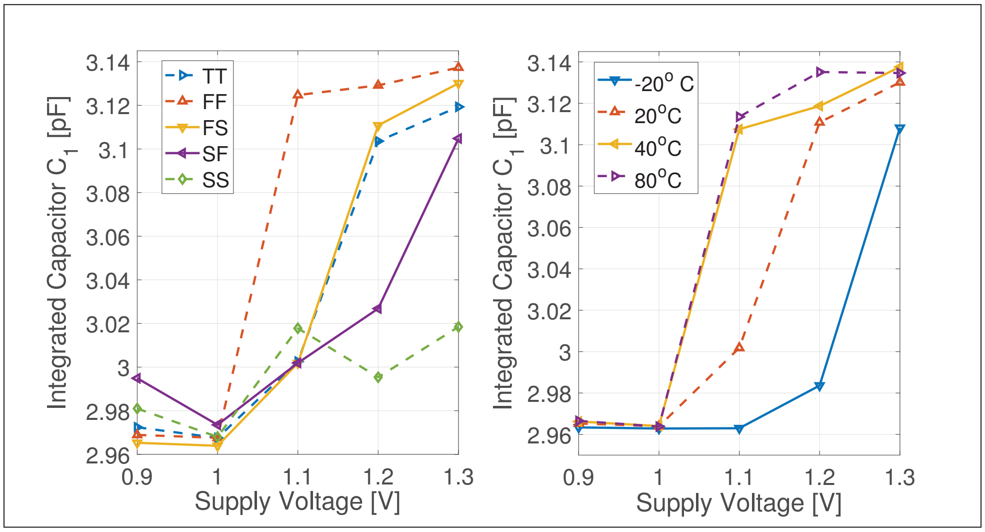


Figure 4.8 Integrated capacitor C_1 according to PVT variations

The active and passive components in the circuit design affect the intrinsic oscillator frequency of continuous chaotic systems. The sampling frequency of the comparators is expected to be as high as possible at the price of the randomness of the output bit-streams. In our hyperchaotic circuit design, the circuit topologies are considered in a trade-off between the power consumption, the circuit stability, and linearity. Indeed, the proposed chaotic system implemented in a fully CMOS circuit design has a self-oscillator frequency of $f = \frac{g_m}{2\pi C} = 5.473\,\text{MHz}$ which is compatible with the state-of-the-art. Therefore, the throughputs of the binary outputs can be increased by using proper post-processing and a high sampling frequency for the comparator. Compared to using off-the-shelf devices as in Ergun & Ozoguz (2005), where the oscillator frequency is limited to a maximum of 830 kHz and the sampling frequency is 19 MHz, the proposed system uses a sampling frequency between 12 MHz and 100 MHz with different configurations of the post-processing circuit.

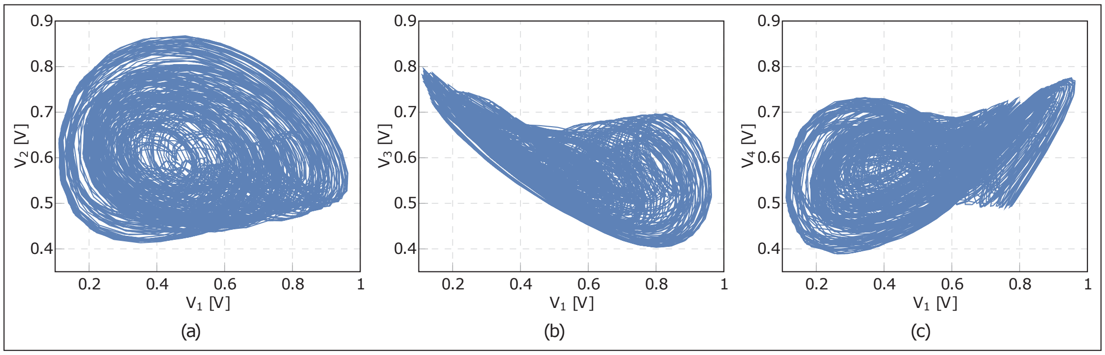


Figure 4.9 Trajectories of the chaotic outputs (a) V_1–V_2, (b) V_1–V_3, and (c) V_1–V_4

The chaotic output phase spaces shown in Figure 4.9 are compatible with the simulation results in MATLAB. Moreover, the practical circuit design provides higher dynamic characteristics than simulation results. For example, we can still observe hidden attractors with arbitrary trajectories of chaotic signals while it is in the limited periodic region in the system simulations in MATLAB. The power spectrum density in Figure 4.10 shows the chaotic signals from the proposed 4D chaotic system circuit design in the frequency domain. As can be observed from the figure, the peak of the power spectrum is concentrated around the intrinsic oscillator frequency; however, it is possible to find spectral components with a non-negligible power for a wide band of frequencies. This allows us to use a sampling frequency much higher than the intrinsic oscillation frequency, while still expecting good results in terms of randomness.

As a final comment, we can notice that using the proposed continuous hyperchaotic system to generate random bits has two advantages compared to chaotic maps. The first advantage comes from superior dynamic characteristics. The second advantage is its four-dimensional chaotic outputs. Although the proposed chaotic system has two positive Lyapunov exponents, corresponding to V_1 and V_2 voltage outputs, all four chaotic signal outputs could be used to generate random bits in parallel. Moreover, in constrast to other continuous chaotic systems, our proposed circuit design uses small embedded CMOS capacitors (3.2 pF) allowing a non-negligible increase in the intrinsic frequency oscillator.

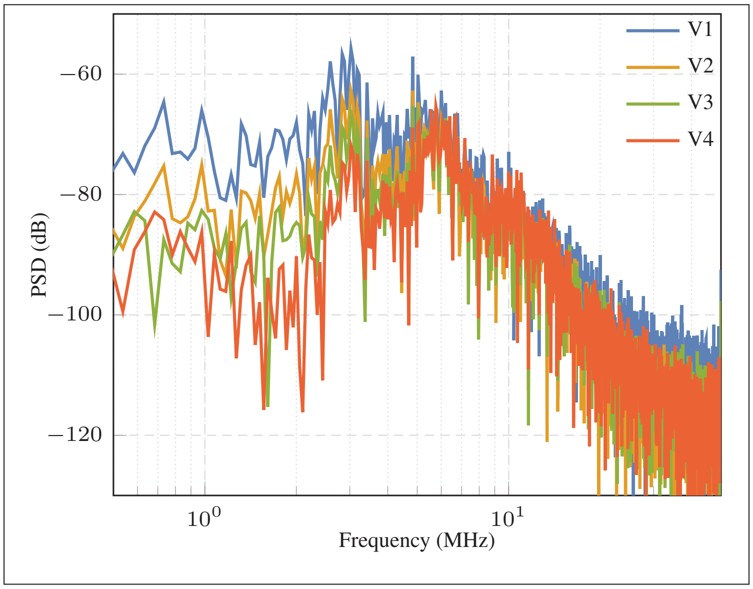


Figure 4.10 Chaotic signal output frequency spectrum

4.4.2 Comparator

Comparators with a maximum sampling frequency of 100 MHz are deployed. The comparator circuit design is detailed in what follows and elaborated in Figure 4.11, which includes two stages. The first stage is a preamplifier which is expected to have a small gain with a high input dynamic. The output reset switch using NMOS N_4 in Figure 4.11 is employed to reduce regeneration in the comparison phase. The second stage is a latch circuit which provides sufficient gain for the comparison phase at the rising edge of the clock signal. The clock frequency is operating at the maximum of 100 MHz with 50%-duty cycle. The comparator amplifies the input offset at the first stage by the cross-coupled PMOS transistors P_1 and P_2, then the offset output is amplified with a high gain at the second stage when the clock signal CLK is at a high level. The PREAMP block is desirable to track the sampled input which is expected to have a large enough input bandwidth

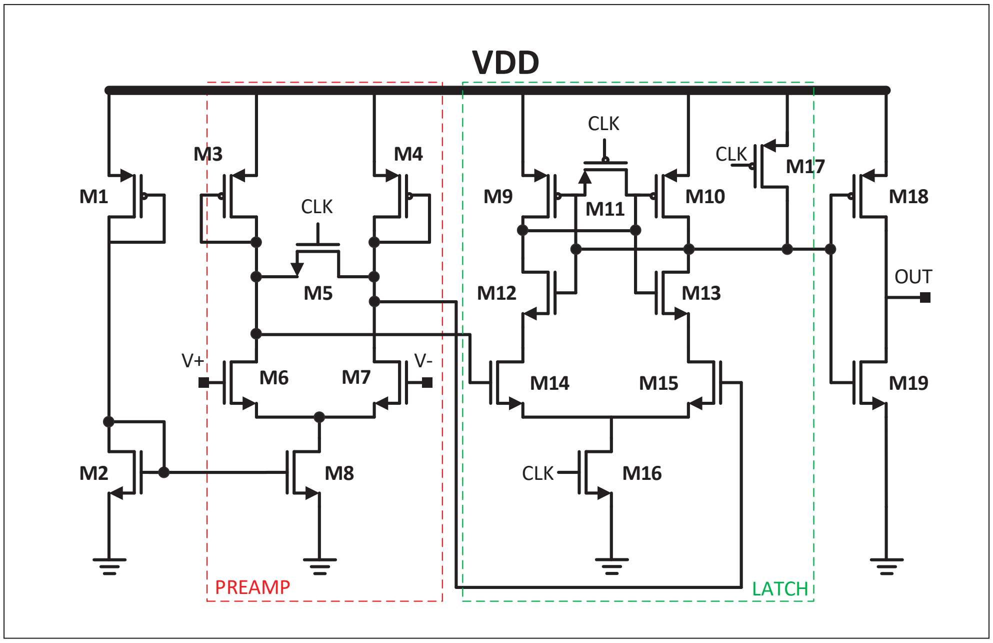


Figure 4.11 Comparator circuit design

and low gain $g_m(N1)/g_m(P1)$. Relatively small NMOS input devices are used to meet the low input capacitance requirement. However, the random offsets due to transistor mismatches, which is the main source of nonlinearity, may be improved by increasing the device's length at the expense of higher input capacitance. The random offset caused by transistor mismatch (in both voltage threshold mismatch and transistor parameter mismatch) introduces the referred input offset V_{OS}. Two partitions of the attractor are considered $\Lambda_1 = [V_{min}, V_{ref} - V_{OS}]$ and $\Lambda_2 = [V_{ref} + V_{OS}, V_{max}]$, the output bit from the comparator is deduced as

$$B_x = \begin{cases} 0 & V_x \subset \Lambda_1 \\ 1 & V_x \subset \Lambda_2 \end{cases} \quad (x = 1, 2, 3, 4). \tag{4.25}$$

The chaotic circuit nonlinearity and mismatches contribute to the imperfection of two partitions Λ_1 and Λ_2 observed compared to the MATLAB simulations. These effects are minimized at each previous block circuit design in trade-off with its circuit requirements in both schematic and layout. The distribution of analog chaotic signals, and the statistical analysis of the mean value and standard deviation are used to setup the reference voltages V_{ref} of the comparators.

4.4.3 SHIFT-XOR based post-processing

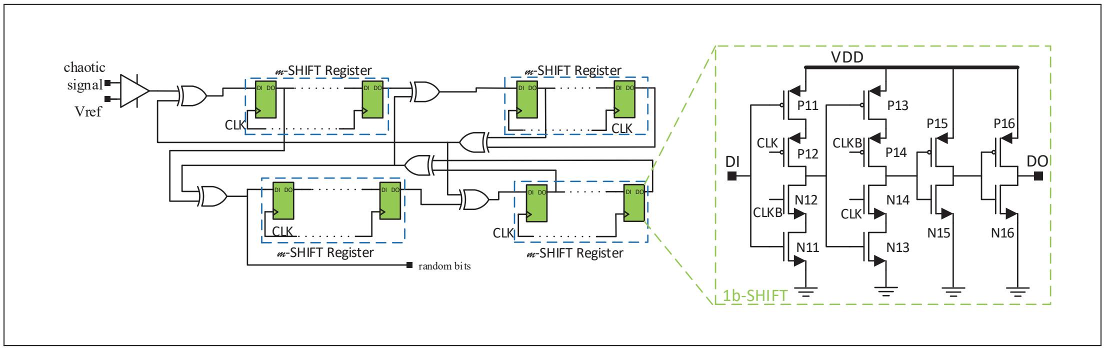


Figure 4.12 Circuit design for post-processing based on dynamic D Flip-Flop

In this paper, we used a SHIFT-XOR based PRNG with multiple values for the length of SHIFT registers. This circuit consists of four shift registers m-SHIFT registers and exclusive-ORs Rozic *et al.* (2016); Pareschi *et al.* (2006). The binary output bits from the comparator are evaluated and reused to XOR-operators with the same bit-stream after a few time steps. With such an approach, as observed in Rozic *et al.* (2016), it is possible to have a much higher bit-rate preservation efficiency compared to canonical approaches such as a simple Von Neumann post-processing. Three values of length of shift registers $m = 2$, $m = 6$, and $m = 8$ will be evaluated in the statistical test.

The circuit design for a one-bit shift register (1b-SHIFT) is elaborated in Figure 4.12 using a positive-edge trigger dynamic flip-flop. The first period, when CLK is low, and CLKB is high, is the sampling period where the input signal is stored. In the second phase, when CLK is changed to a high and CLKB is low, the signal is transferred to the output. Finally, the input signal is shifted one clock period.

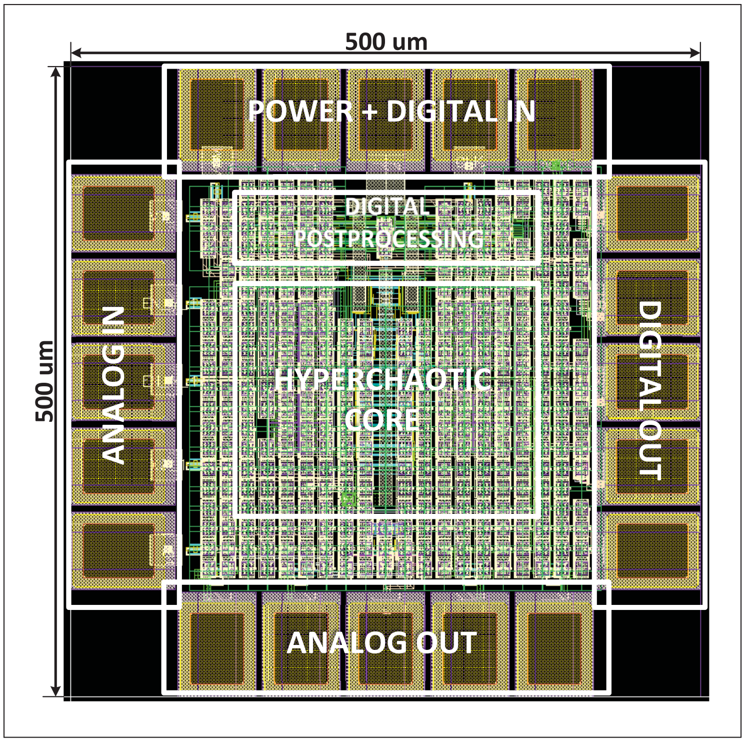


Figure 4.13 Layout diagram of the chip

4.5 Performance evaluation

The proposed circuit was designed and simulated using 130 nm CMOS technology with a 1.2 V voltage supply (VDD). In this section, we present the random bit generator performance including the power consumption, the randomness evaluation by the statistical tests, and the inter-signal correlation test. Moreover, a comparison to state-of-the-art designs is provided to emphasize the work's contribution to engineering applications.

4.5.1 Power consumption

The hyperchaotic circuit consumes a maximum of 980 μW in generating chaotic signals while dissipates a staic current of 623 μA. The comparator utilizes 192 μW for data sampling at 100 MHz. The total power consumption without post-processing is 1240 μW at the normal sampling frequency of 12 MHz which provides a throughput of 48 Mbps by four chaotic output signals. The proposed hyperchaos-based RNG has a high energy efficiency of 25.83 pJ/b in normal operation. The power consumption is summarized in Table 4.3. We also tested the proposed TRNG at high-speed operation mode of 100 MHz for each chaotic output signal. In this case, a high order polynomial feedback function is used in post-processing circuit. In high-speed operation mode, the total power consumption is 1748 μW at a throughput of 400 Mbps (each chaotic output signal provides a throughput of 100 Mbps after its post-processing), which yields an energy efficiency of 4.37 pJ/b. The circuit layout is illustrated in Figure 4.13, in which 8-bit SHIFT registers are used in the post-processing circuit. The total size includes the hyperchaotic core circuit and the digital post-processing circuit, in which the digital power is separated from the analog power to reduce noise effects.

Table 4.3 Power consumption summary

Power (μW)	**Hyperchaos**	**Comparator**	**Total**
Static	748	48	843
Dynamic@12 MHz	980(*)	65	1240
Dynamic@50 MHz	980(*)	120	1459
Dynamic@100 MHz	980(*)	192	1748

(*): the maximum power.

4.5.2 Randomness evaluation

One hundred sixty million bits were collected for the numerical evaluation. Each chaotic dimensional signal contributed forty million bits. The standard operation tests were conducted with a normal supply (VDD=1.2 V) and at a temperature of 20°C. The environment testing included measurements of the influence of temperature and power supply variations. The operation of the proposed TRNG was tested on a wide range of temperature (0°C, 20°C, and 60°C) and a 10% voltage variation (0.9 V, 1.1 V, 1.2 V, and 1.3 V).

4.5.2.1 Min-entropy estimation

To estimate the number of random bits extracted from chaotic signals, the min-entropy, which provides a lower-bound of the raw binary sequences extracted from chaotic signals before post-processing process, is evaluated as

$$H_{min}(B_x) = -\log_2[\max_{B_x \subset \Lambda} P_\Lambda(B_x)](\text{bit/symbol}), \tag{4.26}$$

where B_x is the raw binary random variable which is the binary bit output from the comparator, with probability $P_\Lambda(B_x)$. Chaotic signals are converted into binary streams by the comparator. The conversion rate of binary sequences should be more than $H_{min}(B_x)$ to obtain maximum entropy. The chaos-based random number generator is determined as a non-IID (non-independent and identically distributed) entropy source as described by NIST SP 800-90B Turan *et al.* (2018). Since the chaotic signals are digitalized into binary bits by the comparators, four estimation strategies including most common value, collision estimation, Markov estimation, and compression estimation are applied to the raw binary bits. Three raw binary sequences are collected with a sampling frequency $F_s = 3\,\text{MHz}$. Table 4.4 shows the results of entropy estimation on the raw binary sequences from the chaotic circuit. Since the minimum-entropy estimation is not high due to the asymmetrical geography of chaotic signals and circuit design imperfections, the post-processing circuit is needed to remove bias and increase randomness.

Table 4.4 Entropy per bit estimation of the raw binary sequences

Raw data (binary)	**sequence 1**	**sequence 2**	**sequence 3**
Most Common Value	0.9976	0.9980	0.9984
Collision Estimation	0.7541	0.7045	0.7487
Markov Estimation	0.9729	0.9582	0.9312
Compression Estimation	0.6086	0.6631	0.6437

4.5.2.2 Correlation tests

The correlation, a measure of similarity between two series as a function of the displacement of one relative to the other, is used to measure the mutation of two bitstreams Demir & Ergun (2018-11). The cross-correlation is calculated as:

$$r_{x_1x_2}(k) = \frac{c_{x_1x_2}(k)}{s_{x_1}s_{x_2}} \qquad k = 0, \pm1, \pm2, ..., \tag{4.27}$$

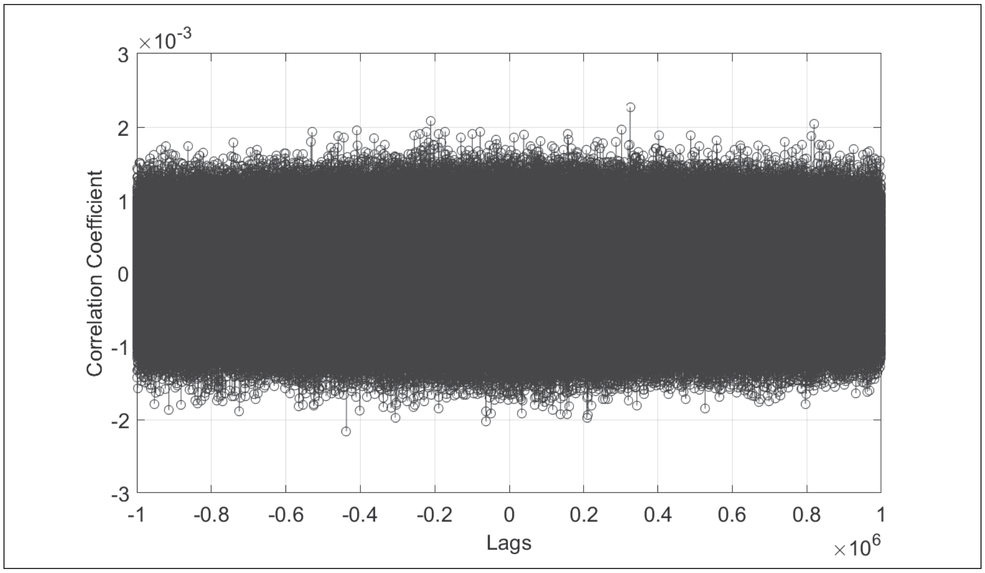


Figure 4.14 Cross-correlation measurement of random bitstreams generated from different chaotic signals V_1 and V_2 at the same time

where k is the number of time shifts (lag) and $c_{x_1x_2}$ is the cross-covariance coefficient of the time series $x_{1,t}$ and $x_{2,t}$, calculated as

$$c_{x_1x_2}(k) = \begin{cases} \frac{1}{T}\sum_{t=1}^{T-k}(x_{1,t}-\bar{x_1})(x_{2,t+k}-\bar{x_2}) & k = 0, 1, ..., \\ \frac{1}{T}\sum_{t=1}^{T+k}(x_{2,t}-\bar{x_2})(x_{1,t-k}-\bar{x_1}) & k = 0, -1, ..., \end{cases} \tag{4.28}$$

where s_{x_1} and s_{x_2} are standard deviations of the series $\sqrt{c_{x_1x_1}(0)}$, and $\sqrt{c_{x_2x_2}(0)}$, respectively. To enable the use of four chaotic signals as entropy sources for random bit generators independently, the cross-correlation between these output ports is measured as depicted in Figure 4.14. This figure shows the un-correlated relationship between the random bitstreams generated by the chaotic signal V_1 and V_2 after post-processing with a sampling frequency of 100 MHz.

4.5.2.3 NIST's test results

The final binary output bitstreams are evaluated using statistical tests to verify the randomness, according to the well-known test suite NIST SP 800-22 Rukhin *et al.* (2010); Pareschi *et al.* (2012). This statistical test works under a tentative assumption of randomness (H0). Therefore, if the randomness assumption is true for the data, the resulting calculated test statistic value on the data will have a very low probability of exceeding the critical value. If the P-value, which is calculated based on the critical value for each test, is larger than 0.01, there is a 99.9% possibility that the data is random. Then the data could be used for cryptographic purposes Pareschi *et al.* (2012). In total, fifteen statistical tests were separated into two parts. 160 M binary bits collected were divided into 1000 streams of 160 Kb length for the first ten tests. The second part used 160 bitstreams of 1 Mb length. Fifteen statistical test results presented in Table 4.5 show the average P-value (PV) for each test and their proportional pass rates (PP). At a normal operation frequency (F_s=12 MHz), the m-SHIFT-XOR passed these tests with high P-values, and high pass proportions with $m = 2$. To evaluate the relation between the length of the shift registers and the possible sampling frequency, we increased the sampling frequency from 12 MHz to 100 MHz. The m-SHIFT-XOR ($m = 2$) does not pass all the tests at 20 MHz. However, it can pass NIST tests with higher value of m, in other words, a higher order of polynomial feedback function. However, due to the trade-off between security and randomness, we could not increase the

Table 4.5 Statistical test results of the output bitstreams after post-processing

NIST	**m=2 12 MHz**		**m=2 20 MHz**		**m=6 50 MHz**		**m=6 80 MHz**		**m=8 100 MHz**	
	PV	**PP**	**PV**	**PP**	**PV**	**PP**	**PV**	**PP**	**PV**	**PP**
Monobit	0.572	0.991	0.618	0.985	0.090	0.984	0.218	0.975	0.324	0.995
Frequency	0.316	0.990	0.038	0.989	0.863	0.986	0.017	0.974	0.764	0.989
Runs	0.428	0.993	0.039	0.995	0.492	0.986	0.670	0.972	0.164	0.988
Longest 1's	0.630	0.994	**0.001**	**0.798**	0.138	0.990	0.920	0.983	0.630	0.994
Rank	0.575	0.988	0.058	0.987	0.542	0.990	0.043	0.974	0.575	0.988
Sum	0.360	0.995	0.883	0.988	0.723	0.988	0.186	0.980	0.924	0.993
Fourier	0.012	0.982	**0.001**	0.985	0.404	0.990	**0.001**	0.975	0.655	0.986
Overlap (*)	0.130	0.989	**0.001**	**0.825**	0.978	0.982	0.408	0.987	0.130	0.989
Non-overlap	0.823	0.990	0.426	0.994	0.837	0.995	0.182	0.991	0.999	1
Linear	0.251	0.994	0.510	0.979	0.035	0.981	**0.001**	**0.957**	0.663	0.986
Universal	0.020	1	0.530	1	0.453	0.966	**0.000**	**0.121**	0.441	1
Entropy	0.459	0.992	0.477	0.991	0.031	0.987	0.666	0.975	0.956	0.987
Serial	0.937	0.988	0.426	0.989	0.411	0.986	0.401	0.979	0.740	0.987
Excur.	0.666	1	0.628	1	0.387	0.969	0.922	0.981	0.304	0.993
Excur. var.	0.808	1	0.967	1	0.249	0.990	0.899	0.990	0.049	1

PV: P-value - PP: Proportion (the fail values are in bold) ratio between the sampling frequency and the intrinsic frequency excessively. The randomness is guaranteed in the high-frequency operation mode of 50 MHz with m-SHIFT-XOR when $m \geq 6$. The maximum operating frequency is tested at 100 MHz, in which the 8-SHIFT-XOR post-processing passed these statistical tests. The first ten tests require minimum proportional pass of 980 samples (98%), while the minimum requirement for the second part is 95% or 152 samples passed (the fail values are in bold).

Table 4.6 Comparison of modern TRNGs implemented in various entropy sources

Design	**Source**	**Techno.**	**VDD [V]**	**TP [Mbps]**	**Power [mW]**	**Ener. Effi. [pJ/b]**
Hsueh & Chen (2019)*	Dis.Chaos	65 nm	0.4	0.01	0.142	14.2
N Nguyen *et al.* (2018)*	Dis.Chaos	65 nm	1.8	50	1.32	26.4
Wannaboon *et al.* (2018)*	Cont.Chaos	180 nm	0.6	0.27	0.000082	35.5
Satpathy *et al.* (2019)**	Meta.	14 nm	0.65	1480	3.7	2.5
Danesh *et al.* (2020)**	Thermal	65 nm	1	100	0.036	0.36
This work*	Cont. Hyperchaos	130 nm	1.2 1.2	48 400	1.24 1.748	25.83 4.37

(*) post-layout simulation results, (**) measurement results.

Table 4.6 shows a comparison between the proposed system and previous chaos-based RNGs in terms of supply voltage, bit throughput, power consumption, and energy efficiency. Our design is comparable to other chaos-based random number generators. Due to the high dimensional chaotic signals and the effectiveness of the post-processing, all four chaotic signal outputs can be used to generate random bits, and therefore the maximum throughput of the generator is increased radically. Moreover, our work is comparable to other kinds of generators which are based on physical entropy such as jitter noise and metastability Bae *et al.* (2017); Kim *et al.* (2017a). The work in Bae *et al.* (2017), which shows a highest random bit throughput, consumes much higher power in compare to our design. Thus, the proposed random bit generator benefits from a low power consumption and a relatively high throughput.

4.6 Conclusion

In this paper, we presented a fully customized CMOS true random number generator including a new hyperchaotic system with hidden attractors and m-SHIFT-XOR post-processing to provide random binary bits for cryptographic applications. The standalone generator is fabricated in 130 nm-CMOS technology. The novelty of the proposed 4D chaotic system was described using theoretical and mathematical analysis. Moreover, the circuit design was simulated in various working conditions against physical attacks such as power variations and noise attacks. The proposed true random number generator provides a high energy efficiency of 4.37 pJ/b for a throughput of 400 Mbps.

CHAPTER 5

PSEUDO-RANDOM BIT GENERATOR BASED ON A NOVEL 5D-HYPERCHAOTIC SYSTEM

Ngoc Nguyen[1] , Toan Bui Q.T[1] , Ghyslain Gagnon[1] , Pascal Giard[1] , Georges Kaddoum[1]

[1] Department of Electrical Engineering, École de Technologie Supérieure,
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3

Paper published in « IEEE Transactions on Industrial Electronics » in June 2021.

5.1 Abstract

Dynamic and non-linear systems are emerging as potential candidates for random bit generation. In this context, chaotic systems, which are both dynamic and stochastic, are particularly suitable. This paper introduces a new continuous chaotic system along with its corresponding implementation, which targets FPGA. This chaotic system has five dimensions, which exhibit complex chaotic dynamics, thus enabling the utilization of chaotic signals in cryptography. A mathematical analysis is presented to demonstrate the dynamic characteristics of the proposed hyperchaotic system. A novel digital implementation of the proposed system is presented. Moreover, a data scrambling circuit is implemented to eliminate the bias effect and increase the randomness of the bitstream generated from the chaotic signals. We show that the proposed random bit generator has high randomness. The generated bits successfully pass two well-known statistical randomness test-suites, i.e., NIST SP800-22, Diehard and TestU01. The ready-to-use random bit generator is deployed on a Xilinx Zynq-7000 SoC ZC702 Evaluation Kit. Experimental results show that the proposed random bit generator can achieve a maximum throughput of 6.78 Gbps, which is over 3.6 times greater than state-of-the-art designs, while requiring under 4% of the resources available on the targeted FPGA.

5.2 Introduction

Random number generators are critical components that are responsible for generating public keys, private keys, and other kinds of random numbers that are utilized in cryptographic applications and security Li *et al.* (2019); Liu, Qin, Liao & Wu (2020). Differential chaotic systems, which present high dynamic characteristics and multi-dimensional signals, are superior in terms of generating random bits due to their ability to achieve a high level of randomness. Therefore, in this work, we focus on the design of such systems, targeted at FPGA.

A hyperchaotic system exhibits rich dynamics since the system states it hosts are expanded exponentially in several directions simultaneously. This property makes the hyperchaotic system an interesting candidate for the generation of random keys used in miscellaneous applications in engineering, such as secure communications, cryptosystems, and encryptions Liu & Tong (2016); Irfan *et al.* (2020). Therefore, in the present work, we develop a 5D hyperchaotic system with three positive Lyapunov exponents to provide better dynamic characteristics than the state of the art. Moreover, high-dimensional chaotic systems provide multiple outputs which improve the throughput of the overall random bit generator.

The implementation methodology adopted has a great impact on the digitalization of differential chaotic systems. The research in Zidan, Radwan & Salama (2011) presented an implementation of numerical techniques, including the Euler, mid-point, and fourth-order Runge-Kutta (RK) methods on FPGA. Among these, the Euler method has the shortest data path, but the least accuracy. The fourth-order RK method shows the highest accuracy, but also has the longest time frame.

In state-of-the-art RK algorithm implementations, mapping functions are used four times and implemented separately Akgul, Calgan, Koyuncu, Pehlivan & Istanbullu (2016); Koyuncu,Ismail Turan (2017). We propose the FFRK method to allow reusing the mapping functions (the latter are implemented only once and reused four times), and as a result, the device resources required to implement the mapping functions are reduced by 75‰. The iterations are controlled by adding MUXs and control signals, which use insignificant amounts of device resources. Moreover, taking advantage of the multiple dimensional signals in hyperchaotic systems, we implement an effective post-processing, in which five chaotic outputs are used to generate ready-to-use random bitstreams. In summary, the contributions of this work include i) a new 5D hyperchaotic system to generate stochastic signals, ii) a novel implementation of the fourth-order RK algorithm, and iii) a simple and effective data post-processing circuit.

The remainder of this paper is organized as follows. The related works are reviewed in Section 5.3. The mathematical model of the hyperchaotic system is clarified in Section 5.4. The implementation of the proposed hyperchaotic system in a FPGA hardware device using the new FFRK method is detailed in Section 5.5. The experimental results are presented in Section 5.6. Then, the randomness evaluation of the random bit generator is presented in Section 5.7. Finally, Section 5.8 summarizes the work in this paper.

5.3 Related works

In this section, we review the differential chaotic systems and their implementations in the digital world to emphasize our research contribution. There are several commonly used differential chaotic systems in secure communications, including the Lorenz system, Chua's system, Liu's system and Lu's system Zhang (2017b); Tolba *et al.* (2017). The Lorenz system has been used in different applications in many research works Zhang (2017b). The research in Zhang (2017b) presents the implementation of a Lorenz system in FPGA hardware devices and co-simulation with Matlab. Liu's system is implemented in Tolba *et al.* (2017) using the Grunward-Letniknov algorithm. However, the above chaotic systems utilize multiple multiplexers, which are resource-demanding in hardware implementations. Moreover, the authors in Koyuncu, Ozcerit & Pehlivan (2013) provide both analog and digital implementations of the Bruke-Shaw chaotic system in a Virtex-6 FPGA chip, which consumes a lot of device resources. The research in Tlelo-Cuautle, Rangel-Magdaleno, Pano-Azucena, Obeso-Rodelo & Nunez-Perez (2015) presents an FPGA realization of Chua's system with multiple scrolls by using different saturated functions. The work in Akgul *et al.* (2016) implements a 3D chaotic system with no equilibrium points. A high-speed FPGA implementation of a 3D continuous chaotic system, which achieves a maximum operating frequency of 293 MHz, is presented in Koyuncu (2017). Although instructive, these chaotic systems are 3-dimensional systems which has only one positive Lyapunov exponent. This means that thay have limited dynamic characteristics as the signal expands exponentially in only one direction. Speed optimization is applied to a 3D chaotic system and then expanded to a 4D chaotic system in Bonny *et al.* (2019), achieving a maximum throughput of 1882 Mbps for the random bit generator.

Nowadays, differential chaotic systems achieve higher complexity by conveying integer-order systems into the fractional-order domain. However, fractional-order systems are complicated to implement in hardware design due to their memory dependency. The hardware implementation of fractional-order differentiators and integrators requires careful consideration Tolba *et al.* (2017).

Although chaotic systems are unpredictable, and have random-like state trajectories, they can be studied and recovered by using computational tools. A cyber-attack has a high possibility of success if the target system uses a well-known and self-excited oscillator for its random bit generator. After the transient process, a trajectory, starting from the point of an unstable manifold in a small neighborhood of unstable equilibrium, can be revealed. Therefore, the system's parameters can be computed to recover the target system. From this point of perspective, the development of modern computers enables the numerical simulation of complex nonlinear dynamical systems, and therefore the structure of their trajectories can be deduced. However, this approach shows very limited success when it comes to hidden attractor chaotic systems Barati *et al.* (2016); Fozin Fonzin, Srinivasan, Kengne & Pelap (2018). The hyperchaotic systems with hidden attractors in Bao *et al.* (2018); Çavuşoğlu *et al.* (2019) were proposed to overcome such attacks. As compared to previous systems, a mathematical analysis of our hyperchaotic system with hidden attractors shows rich dynamic characteristics that are suitable for use in security and cryptographic applications.

5.4 5D differential chaotic system model

The following section presents the proposed hidden attractor hyperchaotic system, which is expressed by five differential equations as given in (1). The theoretical analysis is divided into three parts. The first part presents the proposed chaotic system and a theoretical study of chaotic characteristics, while the second part addresses the stability of the equilibrium. The simulation of chaotic transitions is investigated in the third part.

5.4.1 Mathematical analysis

The proposed 5D dynamic system, which is developed from the 4D hyperjerk chaotic system presented in Pham *et al.* (2016a), is expressed as $S' = F(S), = (x, y, z, u, v) \in \mathbb{R}^5$, with

$$\begin{cases} x' = y \\ y' = z \\ z' = u \\ u' = -z - 0.5 \times u + (x - 1) \times y \\ v' = -u - 0.5 \times v + (x - 1) \times z. \end{cases} . \tag{5.1}$$

The ordinary differential equations (ODEs) are solved and simulated in MATLAB, based on the fourth-order RK integration algorithm with a step size of 10^{-2}. Here, a small step size is chosen to provide a higher resolution and better accuracy. The equilibrium points of the proposed system are located on the line $E(x, 0, 0, 0, 0)$. Therefore, the proposed system is a dynamical system with hidden attractors. According to Çavuşoğlu *et al.* (2019), it is difficult to expose a chaotic attractor, and then reveal the chaotic system architecture by choosing an arbitrary initial condition. In other words, the chaotic attractors are invisible to attackers and their basin localization does not dissolve the chaotic oscillator. In what follows, we present a mathematical analysis of the novel chaotic system in terms of Lyapunov exponents and bifurcation diagrams.

- **Lyapunov exponents:** The Lyapunov exponents of the system are defined as follows.

$$\begin{aligned} L_1 &= \lim_{t\to\infty} \frac{1}{t} \log \frac{\|\partial x(t)\|}{\|\partial x(0)\|} & L_2 &= \lim_{t\to\infty} \frac{1}{t} \log \frac{\|\partial y(t)\|}{\|\partial y(0)\|} \\ L_3 &= \lim_{t\to\infty} \frac{1}{t} \log \frac{\|\partial z(t)\|}{\|\partial z(0)\|} & L_4 &= \lim_{t\to\infty} \frac{1}{t} \log \frac{\|\partial u(t)\|}{\|\partial u(0)\|}. \\ L_5 &= \lim_{t\to\infty} \frac{1}{t} \log \frac{\|\partial v(t)\|}{\|\partial v(0)\|} \end{aligned} \tag{5.2}$$

The Lyapunov exponents of the novel 5D chaotic system are: $L_1 = 0.093790$,$L_2 = 0.001101$, $L_3 = 0.000107$, $L_4 = -0.500100$, and $L_5 = -0.594898$. The initial conditions of the proposed chaotic system are chosen as $x_0 = 0.0002$,$y_0 = 0.0005$,$z_0 = 0.00005$,$v = 0.001$, and $v_0 = 0$. The divergence of (5.1) is evaluated based on the following conditions:

$$\begin{gathered} \frac{\partial \dot{x}}{\partial x} + \frac{\partial \dot{y}}{\partial y} + \frac{\partial \dot{z}}{\partial z} + \frac{\partial \dot{u}}{\partial u} + \frac{\partial \dot{v}}{\partial v} = -1 < 0, \\ L = L_1 + L_2 + L_3 + L_4 + L_5 < 0. \end{gathered} \tag{5.3}$$

- **Lyapunov dimension:** The Lyapunov dimension (D_L), which is closely related to the correlation dimension, is commonly utilized to evaluate the chaotic complexity Chlouverakis & Sprott (2005). A higher value of the Lyapunov dimension represents a higher level of complexity of the chaotic system.

$$D_L = j + \frac{1}{|L_{j+1}|} \sum_{i=1}^{j} L_i, \tag{5.4}$$

where j is the largest index of the positive Lyapunox exponent. In the proposed system, $j = 3$, and the Lyapunov dimension is therefore:

$$D_L = 3 + \frac{L_1 + L_2 + L_3}{|L_4|} = 3.1899. \tag{5.5}$$

Figure 5.1 shows the Poincaré section of the phase-space in the plane $x = 1$. Distinct set of points in the Poincaré section indicates the chaotic region of dynamic system.

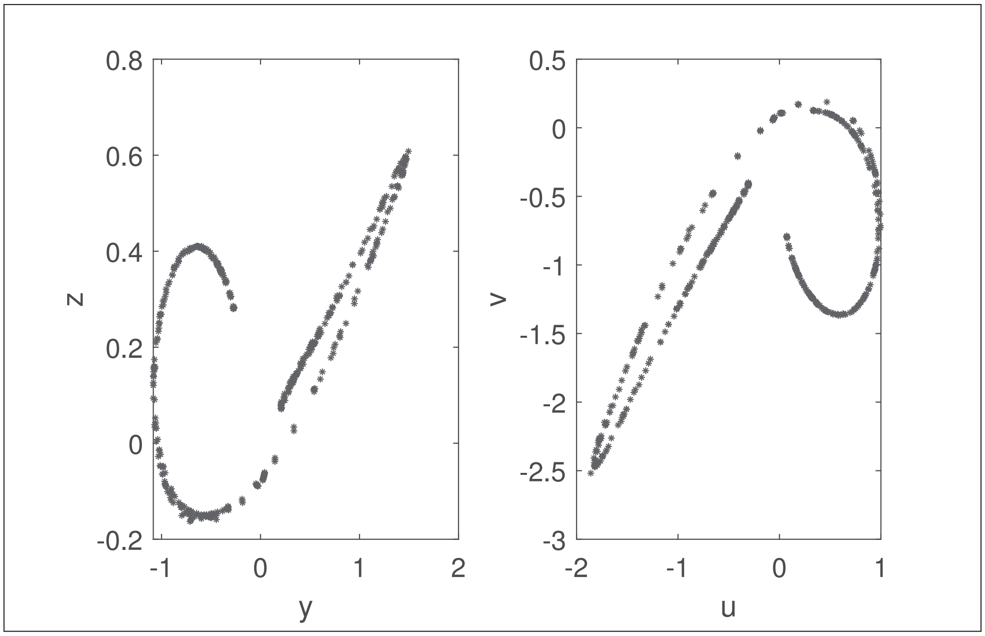


Figure 5.1 Poincaré section of the phase-space in the plain ($x = 1$)

5.4.2 Stability of equilibria

The stability evaluation of the equilibria is an important step to find the chaotic region of a dynamic system. To evaluate the stability of equilibria, the Jacobian matrix at the equilibria $E = [c, 0, 0, 0]$ of the proposed hyperchaotic system is calculated. The eigenvalues of the Jacobian matrix satisfy the condition $J_E - \lambda I = 0$.

$$J_E = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & (c-1) & -1 & -0.5 & 0 \\ 0 & 0 & (c-1) & -1 & -0.5 \end{bmatrix}. \tag{5.6}$$

Assuming a small perturbation from the fixed points $x(t) = x(0) + \Delta x$. If $\Delta x \approx e^{\lambda}t$, the characteristic polynomial equation is derived as:

$$\lambda(\lambda + 0.5)(\lambda^2 + 1.5\lambda + (c - 0.5)) = 0. \tag{5.7}$$

This equation indicates that the Jacobian matrix at equilibrium points has one zero value and three non-zero values. According to the Routh-Hurwitz criterion, apart from the zero eigenvalues, the real parts of the roots of (5.7) are negative if and only if $(c - 0.5) > 0$. To make the equilibrium set E unstable, thereby enabling the possibility of chaos occurrence, the initial condition $x(0) = c$ must satisfy $c < 0.5$. In other words, depending on the initial value, the proposed system has stable or unstable saddle-focus points. Thus, the dynamical behaviors of the line equilibrium chaotic system are heavily dependent on the initial state of the variable $x(t)$.

5.4.3 Transitions to chaotic region

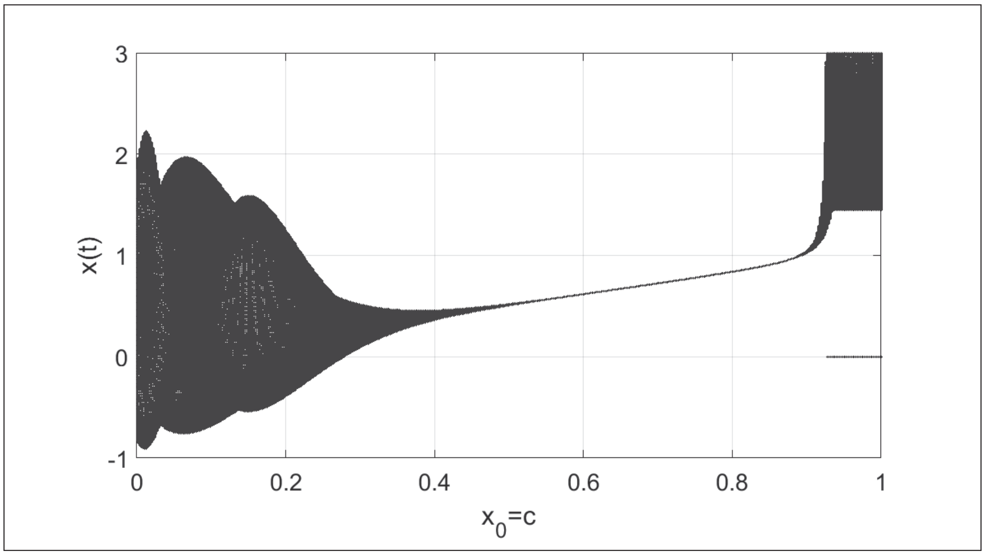


Figure 5.2 Bifurcation diagram according the initial condition

The bifurcation diagram and the Lyapunov spectrum of the state variable $x(t)$ of the proposed system are shown in Figure 5.2 and Figure 5.3, respectively. Both figures illustrate that the value of the initial condition c has an impact on the characteristics of the system. Moreover,

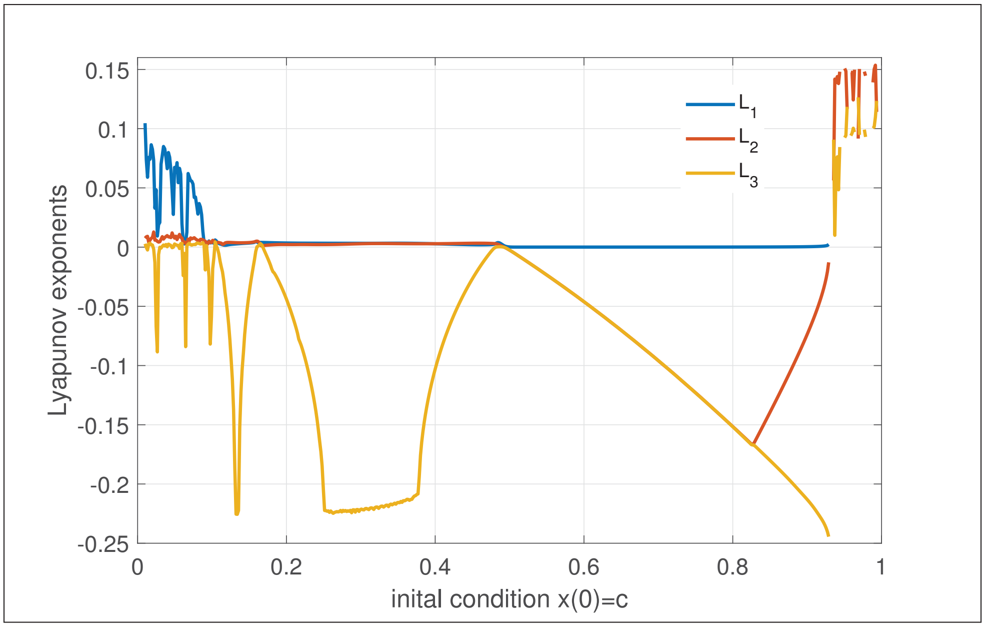


Figure 5.3 Lyapunov spectrum according to initial condition x(0)=c

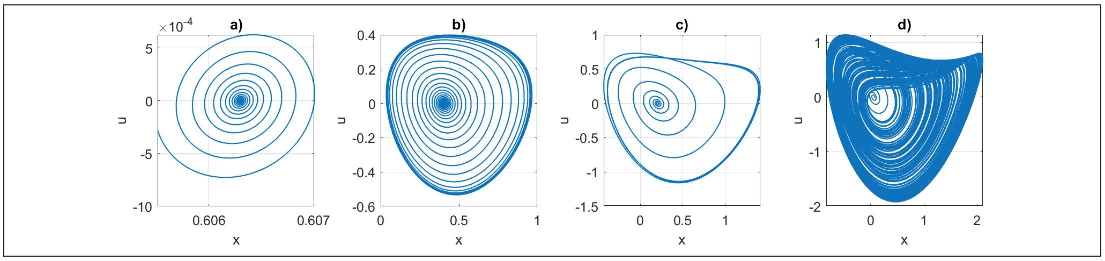


Figure 5.4 Transition to chaos regions depending to initial condition $x(0) = c$: a) stable region for $c = 0.6$, b) limited dynamic region for $c = 0.4$, c) chaotic region with limited periods for $c = 0.2$, and d) rich dynamic and chaotic region for $c = 0.05$

a stable region is observed for $0.5 < c < 0.92$ in Figure 5.2, where the data space converges. This corresponds to the negative Lyapunov exponents in Figure 5.3. A dynamic and unbounded region is observed for $c > 0.92$, a region that corresponds to positive high-value Lyapunov exponents in Figure 5.3. Finally, in the $0 < c < 0.5$ range, the system is dynamic and bounded in a limited data space, as illustrated in the bifurcation diagram. This corresponds to the presence of positive Lyapunov exponents L_1 and/or L_2, as shown by the Lyapunov spectrum. However, the system only exhibits rich and chaotic dynamics for $c < 0.05$, where a high positive value of

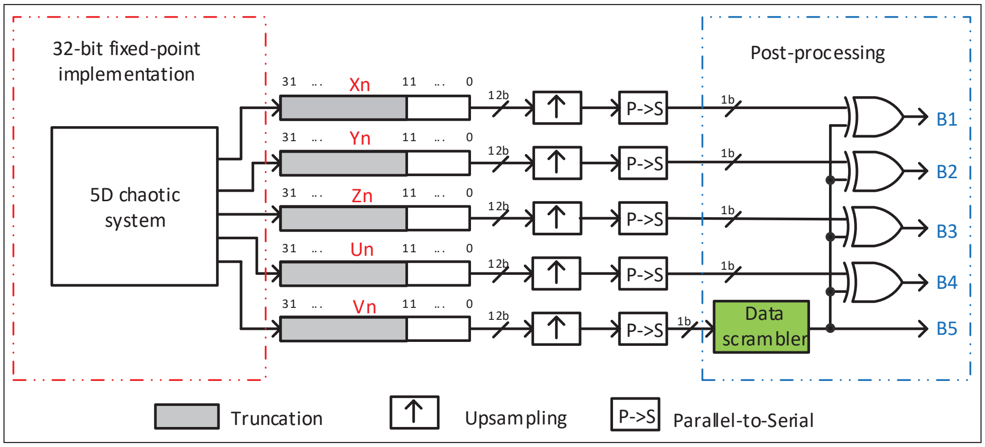


Figure 5.5 The proposed pseudo-random number generator scheme

L_1 is presented, as shown in Figure 5.3. The gradual transition from periodic orbits to chaotic regions based on various values of the initial condition c is observed in Figure 5.4. The chaotic region with rich dynamics illustrated under Figure 5.4-d) is obtained with $c = 0.05$ which is compatible to the stability analysis of equilibria in the previous section.

5.5 Pseudo-random number generator scheme

In this section, we present a pseudo-random number generator scheme based on the chaotic output signals. We use fixed-point 32-bits to represent data, with 1-bit for the sign and 27 bits for the fractions. Figure 5.5 shows the scheme of the proposed random bit generator which includes the implementation of 5D chaotic system block using FFRK with 32-bit fixed-point data, the truncation block is followed. Then, the data is up-sampled to transfer from parallel to serial before the post-processing. In the post-processing, the data scrambler is proposed using a simple linear feedback-shift register (LFSR) of one of the chaotic outputs, while the random bits are taken by XOR operation the output from the data scrambler and the other chaotic signals.

Due to the long data path of the RK algorithm and low self-oscillator frequency of the chaotic system, signal truncation is applied in order to have better randomness qualification in the following block. Therefore, 32-bit chaotic outputs are truncated, where only 12 least significant bits (LSB) are used in the following post-processing. 12-LSBs are up-sampled and serialized before the post-processing step. In what follows, we present a novel FPGA-based implementation of the fourth-order RK algorithm in a new manner to solve the differential chaotic equations. Moreover, we present the post-processing process with a data scrambler to increase the randomness of the generated bitstreams and eliminate bias effects.

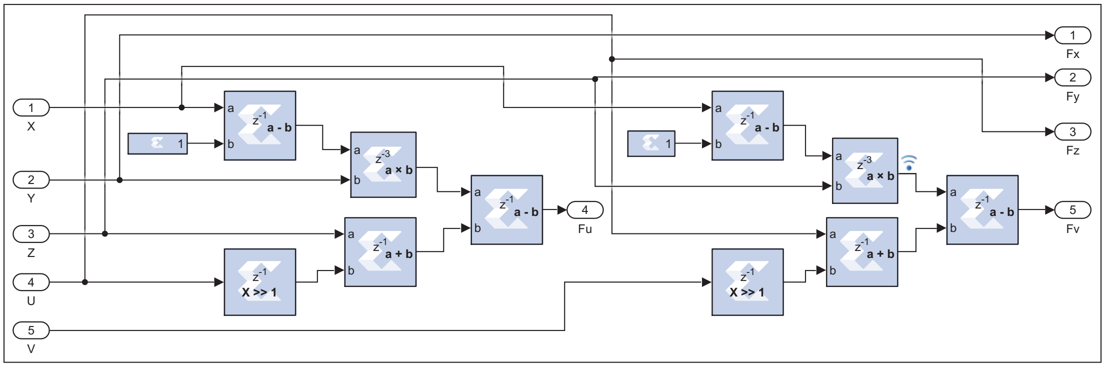


Figure 5.6 Hyperchaos function $F(S_k)$ implementation in XSG

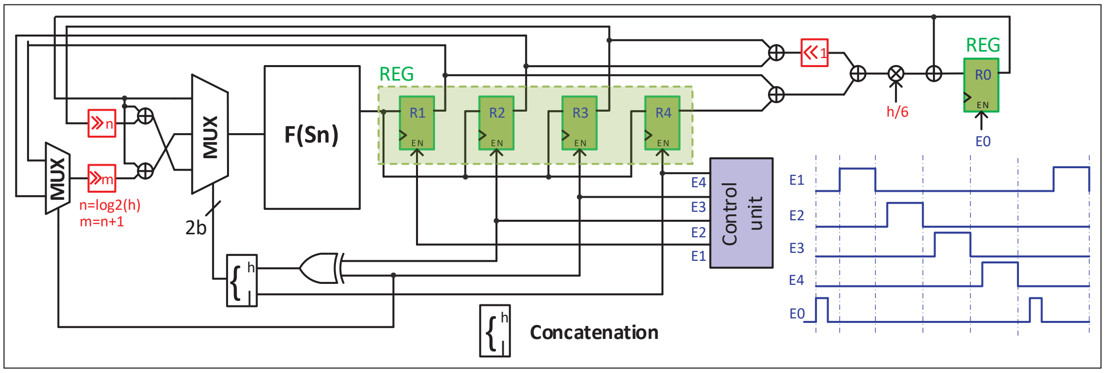


Figure 5.7 Block diagram of the proposed fourth-folding Runge-Kutta algorithm

5.5.1 Implementation of the 5D hyperchaotic system

To solve differential equations, there are three well-known numerical methods, with different complexity and accuracy, namely the Euler method, Midpoint method, and RK method. Compared to the other two candidates, the RK method has the longest calculation path and

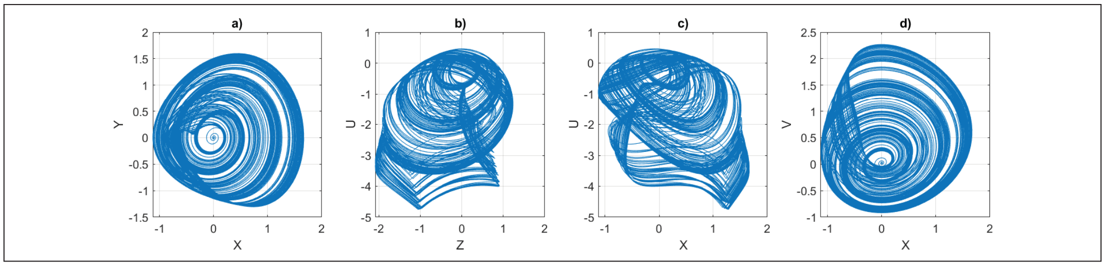


Figure 5.8 Trajectories of the chaotic outputs

requires larger areas Zidan *et al.* (2011). However, it produces the most accurate results. The proposed FFRK implementation optimizes the resources of the 5D hyperchaotic system. Compared to previous RK implementations, the proposed FFRK provides the most accurate solution for differential equations and consumes less device resource. The system state at the n-th time scale is denoted by the vector $S_n = [x_n, y_n, z_n, u_n, v_n]$. The proposed hyperchaotic system in discrete-time implementation is calculated as follows:

$$F(S_n) = \begin{cases} F_x = y_n \\ F_y = z_n \\ F_z = u_n \\ F_u = -z_n - 0.5 \times u_n + (x_n - 1) \times y_n \\ F_v = -u_n - 0.5 \times v_n + (x_n - 1) \times z_n \end{cases} . \tag{5.8}$$

The 4th-order RK algorithm is obtained by defining these primitive parameters k_1, k_2, k_3, and k_4 which are calculated as:

$$\begin{aligned} k_1 &= F(S_n), \\ k_2 &= F(S_n + \frac{1}{2}h \times k_1), \\ k_3 &= F(S_n + \frac{1}{2}h \times k_2), \\ k_4 &= F(S_n + h \times k_3). \end{aligned} \tag{5.9}$$

where h is the discrete step size. The next system state vector S_{n+1} is then evaluated using the previous system state S_n and these above primitive parameters.

$$S_{n+1} = \begin{bmatrix} x_{n+1} \\ y_{n+1} \\ z_{n+1} \\ u_{n+1} \\ v_{n+1} \end{bmatrix} = S_n + \frac{1}{6}h \times (k_1 + 2k_2 + 2k_3 + k_4). \tag{5.10}$$

Figure 5.6 presents the block diagram implementation of $F(S_n)$ in (5.8), which is reused to implement the k_i parameters with different input variables. The block diagram presented in Figure 5.7 shows the implementation of the FFRK algorithm. The discrete step size $h = 0.01$ provides enough resolution for chaotic signals. First, the parameter k_1 is calculated while the register enable control signal E_1 is high. The output k_1 from the block $F(S_n)$ will be temporary stored in register R_1. Then, the parameters k_2, k_3, and k_4 are produced at the high level of the control signals E_2, E_3 and E_4, respectively. Therefore, four primitive parameters k_1, k_2, k_3, and k_4 are stored in four registers. Finally, the next state of the system is produced when all these parameters are ready at the high period of the enable signal E_0 of register R_0. The enable signals are generated by the control unit which uses delay blocks. The phase space of the chaotic outputs of the proposed implementation shown in Figure 5.8 are compatible with the simulation results and mathematical analysis.

5.5.2 Data post-processing implementation

Before addressing the post-processing process, 32-bit chaotic signals are truncated as indicated in Figure 5.5. In previous works, the truncation was mentioned and applied Hua *et al.* (2019); Mao, Cao & Liu (2006). In Hua *et al.* (2019), random bits are generated from the 33^{rd} bit to the 40^{th} bit in each 52-bit data. The authors in Mao *et al.* (2006) used 16-bit data out of a 32-bit signal for the post-processing. Although it is obvious that not the full data-width in chaotic signals are used to generate random bits due to the low self-oscillators, to obtain sufficient randomness quality, there is no clear presented reason to choose a certain truncated position in data-width. The least significant bits (LSBs) are preferred due to their higher fluctuation. Therefore, it is understood that the bits from the LSB to the truncated position will be used in the post-processing. Our research shows that the signal entropy is strongly related to the truncated position. To evaluate the truncated chaotic signal's entropy, Shannon's entropy of a bitstream is calculated. For a given random variable $\tilde{X_n}$, with a possible outcome x_i, $i \subset [0, N-1]$, each with probability p_i, the signal entropy is followed as:

$$H(\tilde{X_n}) = -\sum_{i=0}^{N-1} p_i \log_2 p_i, \tag{5.11}$$

where N is the number of symbols. The average entropy per bit is calculated as:

$$\bar{E} = \frac{H(\tilde{X_n})}{N_b}, \tag{5.12}$$

in which, $\tilde{X_n}$ denotes the truncated chaotic signal and $N_b = \log_2 N$ is the truncated position. Figure 5.9 provides the average entropy per bit with different length of the truncated signals. The average entropy per bit decreases significantly when the number of remaining bits larger than 12. Therefore, in this work, a 12-bit truncated chaotic vector $\tilde{S_n} = [\tilde{X_n}, \tilde{Y_n}, \tilde{Z_n}, \tilde{U_n}, \tilde{V_n}]$ is used in the data post-processing.

The proposed data post-processing hardware implementation based on shift registers and XOR operations is presented in Figure 5.10. The data scrambler is composed of four shift registers (m-shift registers) and XOR operations Pareschi *et al.* (2006); Rozic *et al.* (2016). The working principle is to evaluate the incoming bits from the truncated chaotic signals and to reuse these bits. One of the chaotic signal outputs of the proposed 5D hyper-chaotic system ($\tilde{V_n}$) is used to generate the seed for the data scrambler. Then, the other serialized truncated chaotic signals are XOR-ed with the output bit from the data scrambler. Based on the experimental result, $m = 6$ shift registers, which are implemented by delay units (flip-flop) in the Xilinx System Generator, are utilized to obtain high success rates in the statistical tests, . Five random bitstreams ($B_1 \rightarrow B_5$) are collected for the evaluation phase, which will be presented below.

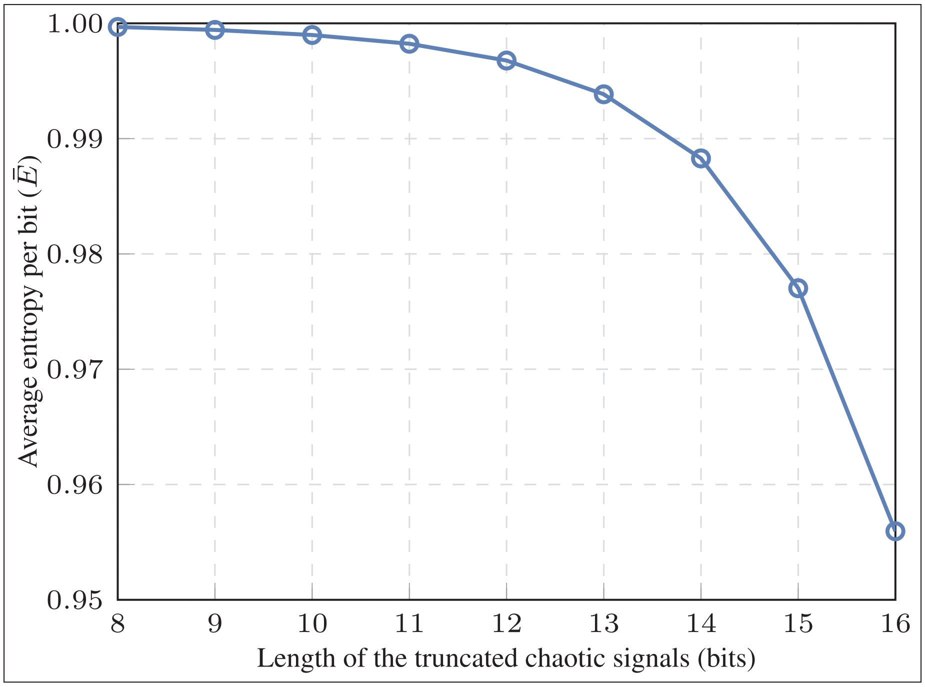


Figure 5.9 Average entropy per bit with different length of truncated chaotic signals

5.6 Experimental results

In this section, we present the deployment of the random bit generator in the Xilinx Zynq-7000 SoC ZC702 Evaluation Kit with the chip xc7-z020-1clg484. The random number generator design is generated as IP-core and communicates to the main processor.

5.6.1 Experimental setup

In this section, we present the deployment of the random bit generator on the Xilinx Zynq-7000 SoC ZC702 Evaluation Kit. The implementation is performed using the Xilinx System Generator tool. The random number generator is generated as an IP-core and evaluated on Xilinx Vivado tools with the default configuration.

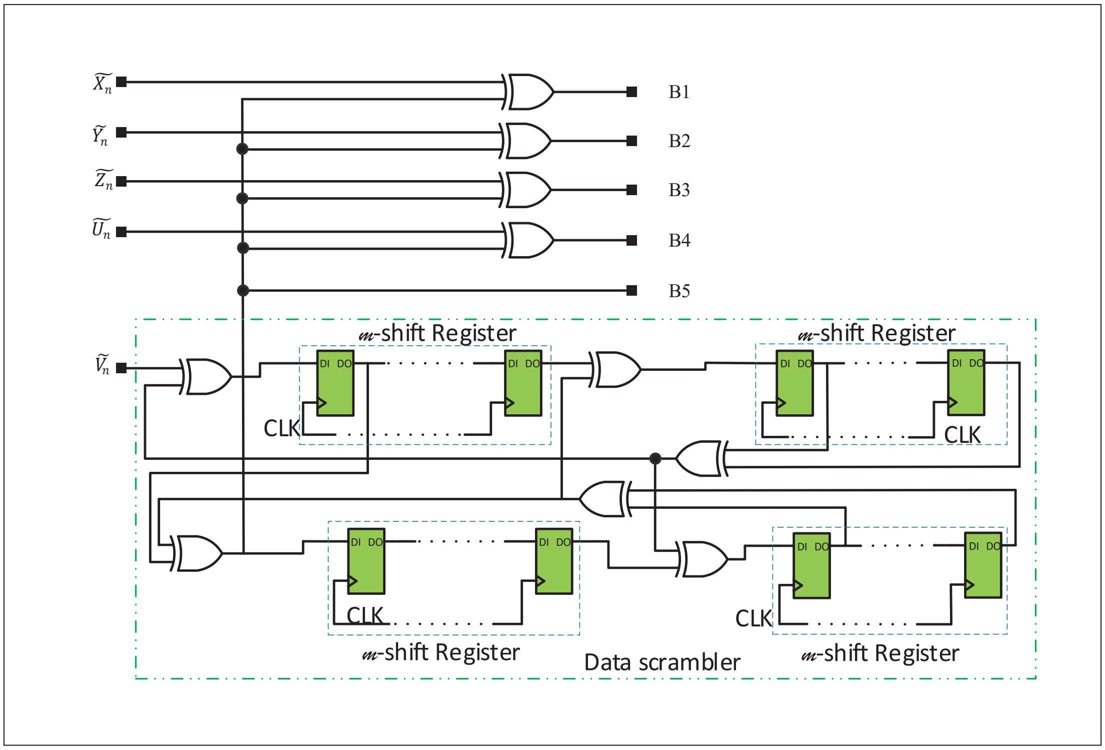


Figure 5.10 Post-processing design

5.6.2 Comparison and discussion

We start this section with a comparison with the state-of-the-art chaotic systems, i.e., a Lorenz system as well as both Elwaki's systems of Elwakil & Kennedy (2001). To illustrate the advantage of our FFRK implementation, we carried out our own implementation of the systems, in which we produced versions that either integrated the traditional RK implementation or our proposed FFRK method. The implementation is produced on the evaluation kit mentioned above. Four iterations of the mapping functions are used to generate k_i in the traditional fourth-order RK method.

Figure 5.11 shows the amount of FPGA resources required for the various chaotic systems. The Lorenz systems are denoted LRZ-RK and LRZ-FFRK, while the Elwaki's systems are denoted EWK1-RK, EWK2-RK, EWK1-FFRK, and EWK2-FFRK; the RK and FFRK suffixes respectively indicate whether the system uses either the traditional RK implementation or our proposed FFRK implementation. From Figure 5.11, it can be seen that compared to traditional RK methods Azzaz *et al.* (2009); Zidan *et al.* (2011), the proposed FFRK requires 4 times less DSPs, under half the number of LUT, and approximately 15% less FF. Figure 5.11 also includes

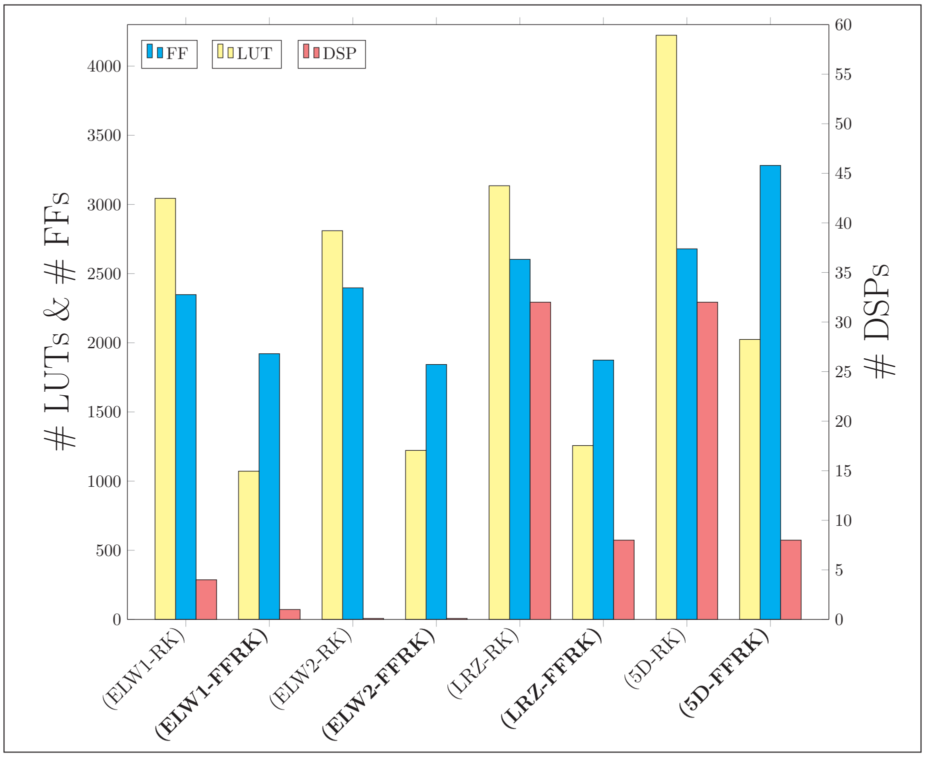


Figure 5.11 Amount of FPGA resources in terms of LUT, FF, and DSPs required by various continuous chaotic systems, using the traditional RK implementation and the proposed FFRK

results for the proposed 5D chaotic system, both with the traditional RK method (denoted 5D-RK) and the proposed FFRK method (denoted 5D-FFRK). Using the proposed FFRK leads to a significant reduction in the amount of LUT and DSPs required, approximately the same as that seen for the other systems. There is a small increase, (~ 25%), in the amount of FF required.

In terms of resource requirements, it can be observed that the 3D chaotic system using the fourth-order RK algorithm in Koyuncu (2017), which utilized up to 32% of the available resources from the Virtex-6 XC6VLX240T-1-FF1156 FPGA chip, requires more resources than the other systems. Compared to our work, the Grünwald-Letnikov (GL)-based algorithms of Tolba *et al.* (2017) require a similar number of FF and from 3× to 6.4× the number of slices. While the implementation of the 3D-fractional-order chaotic system using the Adomian decomposition method (ADM) from Rajagopal, Akgul, Jafari & Aricioglu (2018) requires a smaller number of slices, the random number generator is only performed in software design. The 3D-chaos based PRNG, which is itself based on Lorenz and Lu chaotic systems from Rezk, Madian, Radwan & Soliman (2019) using the Euler method, uses the least number of resources among all systems, but offers an achievable throughput of 1872 Mbps, which is much lower than that of our proposed system. The 4D-chaos based system from Bonny *et al.* (2019), which is optimized for throughput, is the one that comes the closest to our system in terms of the number of LUTs, where our implementation achieves 3.6× greater throughput. The system in Koyuncu, Tuna, Pehlivan, Fidan & Alcin (2020) uses the highest number of resources among all the 3D-chaos based PRNG implementations using the Euler method. We recall that the hardware implementation of the Euler method requires fewer device resources, but achieves less accuracy as compared to the fourth-order Runge-Kutta algorithm Zidan *et al.* (2011).

Our design seeks to reach a high throughput, and a high level of randomness while requiring a modest amount of resources. The Euler method implementation provides less accuracy, and therefore the average entropy per bit is decreased and so is the randomness level for the output random bits. We also note that for the targeted FPGA, the proposed system requires under 4% of the available LUTs, FFs, or DSP blocks. The design works at a maximum frequency of 113 MHz, which enables a maximum throughput of 6.78 Gbps for the generator. The estimated power consumption of the random number generator is 73mW@6.78Gbps. Therefore, the estimated energy efficiency is 10.8 pJ/b. The FFRK implementation for the chaotic system has a delay of 65 clock cycles while there is only one latency at the random outputs.

5.7 Randomness performance evaluation

In this section, we present multiple tests performed on the harvested binary bitstreams before and after applying post processing. These bitstreams are collected by hardware co-simulation of Vivado System Generator and Matlab. The PRNG core runs on the FPGA board connected

Table 5.1 FPGA resource comparison for various implementations of continuous chaos-based PRNGs

PRNG	Chaotic	Method	Resource			Max. Freq.	TP
	System		LUT	FF	DSP	[MHz]	[Mbps]
Koyuncu (2017)	3D	RK	43732	42092	–	293.8	58.7
Tolba *et al.* (2017)	3D	GL	5688	4962	99	38	1554
Rajagopal *et al.* (2018)	3D	ADM	1220	192	8	87	–
Rezk *et al.* (2019)	3D	Euler	494	118	8	78	1872
Bonny *et al.* (2019)	3D	Euler	1169	416	–	107	1178
Bonny *et al.* (2019)	4D	Euler	1882	480	–	112	1869
Koyuncu *et al.* (2020)	3D	Euler	1355	1318	–	464	464
This work	5D	FFRK	2017	3458	8	113	6780

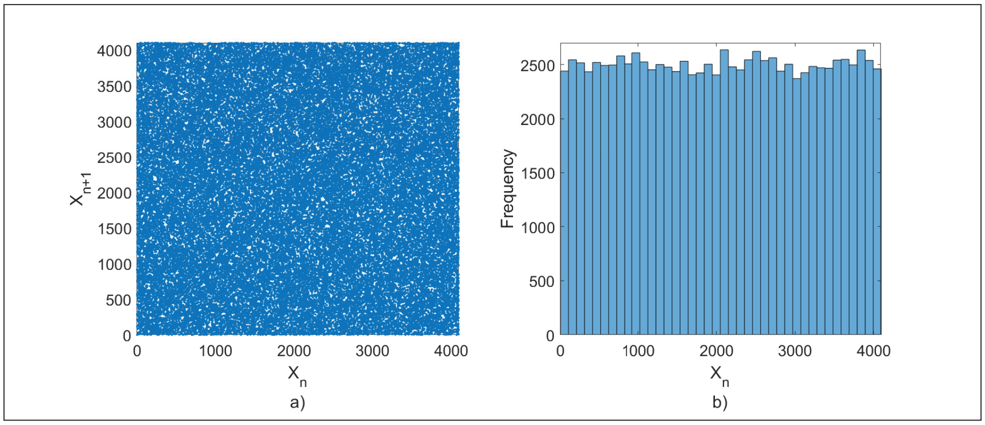


Figure 5.12 12b-truncated chaotic signals in a) X-Y distribution points and b) Histogram diagram. (before post-processing)

to the computer and the output data are collected through JTAG. The experimental setup is illustrated in Figure 5.14. The random binary outputs are observed using an oscilloscope.

5.7.1 Histogram

First, 100K length of 12b-truncated chaotic signals are collected and used in distribution and histogram analysis. As shown in Figure 5.12-a), since there are no obvious patterns in their distribution, 12b-truncated chaotic signals have good randomness characteristics. Figure 5.12-b) shows the histogram of the signal $\tilde{X}_n$ before post-processing where several peaks that can be

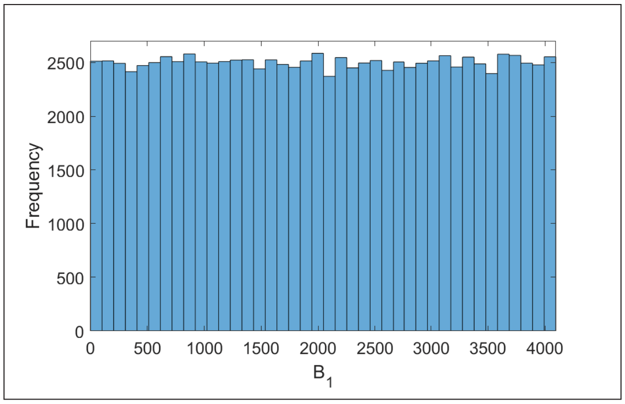


Figure 5.13 Histogram of output bitstreams.(after post-processing)

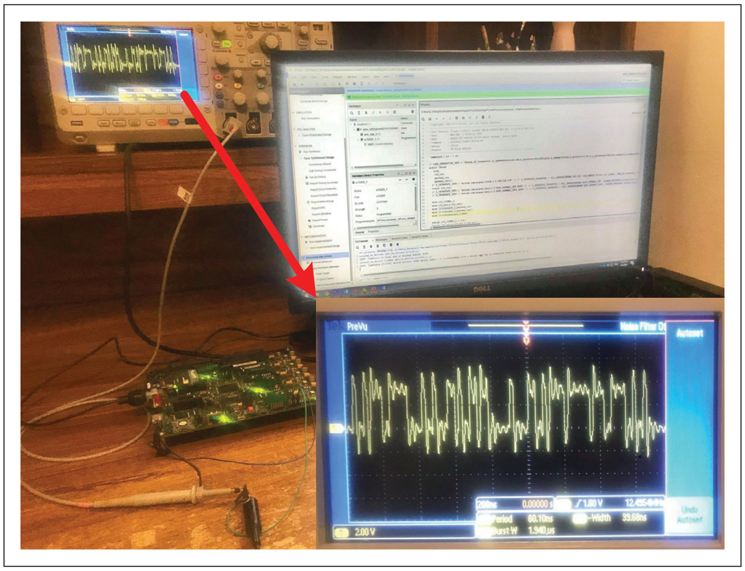


Figure 5.14 The experimental setup for the proposed chaos-based PRNG

observed. The histogram of the output bitstream B_1 after post-processing is illustrated in Figure 5.13, which indicates a random distribution, where no apparent pattern can be discerned. Similar results are observed for B_2 to B_5.

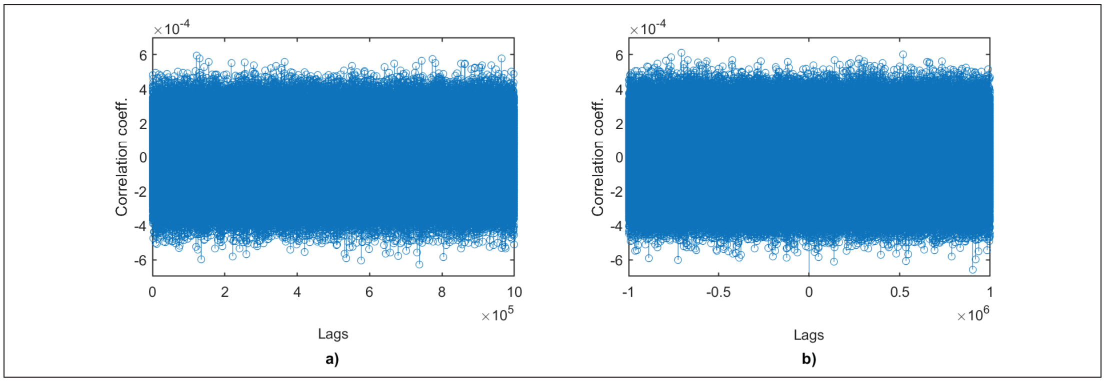


Figure 5.15 a) autocorrelation of intra-signal random bitstream B_1 and b) crosscorrelation of inter-signal random bitstreams B_1 vs. B_2

5.7.2 Standard statistical random testsuites

In order to be used in cryptography applications, the random binary bits should be tested using statistical tests which require long bitstreams. The NIST SP800-22 test-suite, which is developed and introduced by The National Institute of Standards and Technology (NIST), is commonly used Rukhin *et al.* (2010). The NIST test results are presented in Table 5.2. Moreover, Table 5.3 provides the results for the Diehard tests which consist of 12 sub-tests. TestU01 is another commonly used test-suite providing a variety of statistical tests for random bit generators. We applied two battery tests including the Rabbit and the Alphabit tests, to the binary sequences from the proposed random bit generator. The Rabbit test includes 39 subtests and the Alphabit test includes 17 subtests for a bitstream of 2^{25} bit length. The battery test results are presented in Table 5.4. Most of the tests are passed, except for the multinomial test of bitstream B_3, with a P-value = 0.00097, which is very close to the threshold (0.001). It should not be understood that if a RNG fails in some statistical tests, it cannot be used in practical problems; even the best commercial RNGs experience failures in very complicated tests L'Ecuyer & Simard (2007).

5.7.3 TestU01 test

TestU01 is another commonly used test-suite providing a variety of statistical tests for random bit generators. We applied two battery tests including Rabbit and Alphabit to the binary sequences

Table 5.2 Results of NIST SP800-22 for the proposed PRNG, $N = 10^3$ sequences of the length of 1Mb

NIST SP-800.22	P-value(*)	Proportion
Monobit test	0.323668	998/1000
Frequency within block test	0.763677	997/1000
Runs test	0.164425	992/1000
Longest run 1's test	0.630872	1000/1000
Rank test	0.575608	990/1000
DFT test	0.655608	996/1000
Cumulative sum	0.924844	997/1000
Overlapping template	0.130366	990/1000
Non-overlapping template	0.999999	997/1000
Linear complexity test	0.663130	998/1000
Maurers universal test	0.999142	998/1000
Approximate entropy	0.956970	1000/1000
Serial	0.740523	1000/1000
Random excursions	0.304039	997/1000
Random excursion variant	0.049499	988/1000

(*): the average value.

Table 5.3 Diehard tests for the proposed PRNG, sequence of 5M bits.

Dieharder	P-value	Result
Birthdays	0.0528	PASSED
OPERM5	0.0000	FAILED
Rank 6x8	0.0042	WEAK
Bitstream	0.9566	PASSED
OPSO	0.0109	PASSED
DNA	0.1829	PASSED
Count 1s string	0.9989	PASSED
Parking lot	0.2452	PASSED
2d sphere	0.1491	PASSED
3d sphere	0.1581	PASSED
Sums	0.0222	PASSED
Runs	0.9288	PASSED
Craps	0.4094	PASSED

from the proposed random bit generator. The Rabbit test includes 39 subtests and the Alphabit test includes 17 subtests for a bitstream of 2^{25} bit length. The battery test results are presented

Table 5.4 Battery test results for 2^{25} output binary bits

TestU01	**Rabbit**	**Alphabit**
B_1	39/39	17/17
B_2	39/39	17/17
B_3	38/39	17/17
B_4	39/39	17/17
B_5	39/39	17/17

in Table 5.4. Most of the tests are passed, except the multinomial test of bitstream B_3, with a P-value = 0.00097, which is very close to the threshold (0.001). It should not be understood that if a RNG fails in some statistical tests, it cannot be used in practical problems; even the best commercial RNGs have failure in very complicated tests which are impractical to break through L'Ecuyer & Simard (2007).

5.8 Conclusion

We have presented a novel 5D hyperchaotic system along with its implementation on FPGA. In this work, we proposed a novel method, which reduces device resource usage, to implement the fourth-order Runge-Kutta numerical technique to solve differential equations in chaotic systems. The PRNG, which is designed to take advantage of the high dimension and the high accuracy of the hyperchaotic system's hardware implementation, reaches a maximum throughput of 6.78 Gbps. The design, which is deployed in the Xilinx FPGA chip, occupies under 4% of the available FPGA hardware resources. The random generated bits have been verified in modern statistical tests confirming that our proposed system is suitable for industrial applications. Future work includes integrating the proposed PRNG in real-time communication applications such as the secure streaming of images and videos.

CHAPTER 6

CONTINUOUS-CHAOS-BASED PSEUDO RANDOM NUMBER GENERATOR FOR HARDWARE CONSTRAINED DEVICES

Ngoc Nguyen[1] , Toan Bui Q.T[1] , Ghyslain Gagnon[1] , Pascal Giard[1] , Georges Kaddoum[1]

[1] Department of Electrical Engineering, École de Technologie Supérieure,
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3

Paper submitted for publication in « IEEE Transactions on Very Large Scale Integration (VLSI) Systems » in February 2022.

6.1 Abstract

In this chapter, we introduce a high-throughput and low-complexity pseudo-random number generator that employs a continuous chaotic system as an entropy source. More specifically, a four-dimensional chaotic system that takes advantage of high-dimensional data, whereas low-complexity of non-linear absolute function is proposed to increase the throughput of the generated random bitstream, while utilizing lesser hardware resources. The dynamic characteristics of the chaotic system are presented to show the non-periodicity and stochastic trajectories. A hardware implementation is performed in a field-programmable gate array (FPGA), using the Xilinx System Generator. The proposed pseudo-random number generator, which is deployed on a Xilinx XC7Z020-1CLG484C chip, achieves a throughput of 8.56 Gbps at the maximum frequency of 214 MHz. The implementation requires under 1% of the available resources. Generated random bits successfully passed all the subtests in two standard statisticaltests: the NIST test and TestU01. Moreover, the image encryption algorithm, which uses the proposed random number generator, achieves uniform distribution of encrypted images.

6.2 Introduction

Pseudo-random number generators (PRNGs) have been used in many cryptographic applications and data security. Sequences of random numbers are expected to be unpredictable, stochastic, and have a like-wise uniform distribution. Recently, PRNGs based on non-linear dynamic systems have been attracting noteworthy research attention. In dynamic systems, chaos is an interesting phenomenon that is characterized by irregular motion in data space and sensitivity to initialize conditions. A dynamic system is presented by a set of differential equations, in which the present sates is calculated based on the previous state and its change over time. If the rule is applied at discrete times, the system is called iterated map. Iterated maps, with chaotic characteristics, are called chaotic maps. Otherwise, we assume the time as continuous, the dynamic system, with chaotic characteristics, is called a continuous chaotic system.

Chaos has been used as the source of randomness for decades. Chaotic maps (Logistic map, Tent map, Bernoulli map) are based on polynomial function using a non-linear dynamic transformation Wang, Song, Liu, Pan & Ding (2014); Zhang (2017a). The logistic map has been implemented by different methodologies Dabal & Pelka (2014). In Giard, Kaddoum, Gagnon & Thibeault, four different chaotic maps are implemented and compared on FPGA, namely, Bernoulli, Chebychev, Tent, and Cubic chaotic maps. In Hua *et al.* (2019), a sine chaotic model is applied to enhance the performance of the logistic, sine, and tent maps. However, the sine function, which consumes a large number of device resources, hinders the application of this approach on constrained devices.

The main drawbacks of PRNG based on chaotic maps are their limitation of non-linearity and complexity, which lower their randomness levels. Moreover, most chaotic maps have only one dimension, which limits the throughput of the random bitstreams.

In chaotic systems, Lyapunov exponents (LEs), that characterize the rate of data separation in time evolution, are used to evaluate dynamic characteristics. The rate of data separation is different for different orientations. Thus, there is a spectrum of $n-$LEs corresponding to the $n-$dimensional system or the $n-$dimensional phase space. A continuous dynamic system has at least three dimensions (3D) with a positive LE to indicate chaotic characteristics (with some other properties that we will discuss later in this paper). It is proved that higher-dimensional chaotic systems - chaotic systems with at least four dimensions - have a higher possibility of having two positive Lyapunov exponents. Thus, the system state expands exponentially in several directions simultaneously. Thanks to this property, hyperchaotic systems are valuable candidates to be used as entropy sources, which are to generate random keys in high-security applications such as data encryption, e-transfer, and secure communication Liu & Tong (2016); Pham *et al.* (2016b); Wang *et al.* (2016b). In this paper, we propose a four-dimensional(4D) chaotic system that has two positive Lyapunov exponents to provide high dynamic characteristics. Four chaotic signals are processed parallelly in the post-processing part to improve the random bit generator's throughput.

The remainder of the paper is laid out as follows. The related works are reviewed in Section 6.3. The system model and dynamic analysis of the proposed hyperchaotic system are provided in Section 6.4. Section 6.5 presents the implementation of the proposed PRNG based on the Euler method. The implementation is designed in FPGA using the model-based design. Then, two different statistical tests applied to the random bitstreams to evaluate the randomness are presented in Section 6.6. Finally, the conclusion and future of this work are provided in Section 6.7.

6.3 Related works

In this section, we review related works and emphasize our contributions. There are several recently developed continuous chaotic systems that are used in secure communications. A 3D memristive-based chaotic system is presented and implemented in FPGA in Haliuk, Krulikovskyi, Vovchuk & Corinto (2022). The PRNG is implemented by using both the Euler method and the Runge-Kutta method to solve the chaotic system. Although statistical tests applied to the output random bitstream are passed, the device utilization is not evaluated in this work. A chaotic system with a hidden attractor is proposed in Çavuşoğlu *et al.* (2019). Circuit implementation in the analog circuit is simulated in OrCAD-PSpice, then the chaotic signals are used to generate random bits for a data encryption application. The authors in Koyuncu (2017) presented a high-speed TRNG based on a 3D chaotic system. The TRNG achieves output bit rate of 58.7 Mbps at the maximum clock frequency of 293 MHz. The authors in Koyuncu *et al.* (2020) presented the FPGA implementation of a continuous chaotic system which is proposed in Pehlivan & Uyaroglu (2012). The authors also presented a dual-entropy TRNG based on continuous chaos and ring-oscillator. The chaotic system designed in FPGA works at the maximum frequency of 465 MHz, providing the maximum throughput of 465 Mbps for the TRNG

Recently, the following high-dimensional chaotic systems are developed. The authors in Prakash *et al.* (2020) proposed a novel simple 4D chaotic system and the implementation of chaos-based TRNG in FPGA. This design achieves the bit rate of 185 Mbps. The work presented in Bonny *et al.* (2019) implemented both 3D and 4D chaotic systems in FPGA. Two optimization strategies are presented: throughput-optimized and resource-optimized architectures. However, the dynamic characteristics of the proposed 4D chaotic system in that work haven't been analyzed. Four-wing hyperchaotic system is implemented in FPGA and the PRNG based on this chaotic system is proposed in Yu, Wan, Jin *et al.*. The maximum frequency has been achieved as 135 MHz, which results in 62.5 Mbps.

Recently, fractional-order implementation of chaotic systems has been taking research's attention due to its high complexity and higher dynamic characteristics, in comparison to the integer-order representation. Liu system and V-shape multi-scroll chaotic system are implemented in FPGA by using the Grunwald-Letnikov fractional operator in Tolba *et al.* (2017). The maximum Lyapunov exponents are increased significantly by using fractional-order representation. However, using fractional-order utilizes multipliers which consume lots of device resources.

The aim of this work is to propose a simple hyper-dimensional chaotic system in which the dynamic characteristics are comparative to the previous hyper-chaotic systems. Additionally, the simple chaotic system (which has simple non-linear functions), resulting in the device resource redundancy, allows utilization of the design in hardware-constrained devices, such as the FPGA chip without DSPs. Moreover, the higher-dimensional chaotic signal improves significantly the random bit throughput of the proposed chaos-based PRNG.

6.4 Four-dimensional hyperchaotic system

In this section, we present our proposed 4D dynamic system which is developed and extended from 3D dynamic systems Patidar & Sud (2005); Liu & Chen (2003). Absolute function is an interesting candidate to use as non-linear function to reduce implementation cost. However, existing chaotic systems either use high power of absolute function or absolute function with multiplier, which both consume a large number of device resources. Our system contains one absolute function, which is also reused in the extended dimension.

The 4D system is formed of fourth differential equations which is expressed by $S = x, y, z, u$, with

$$\begin{cases} x' = y \\ y' = z \\ z' = a \cdot (|x| - 1) - (y + b \cdot z) \\ u' = (|x| - 1) - z \end{cases} \tag{6.1}$$

In order to analyze the dynamic characteristics, the Euler method is used to solve the differential problem. Therefore, the system is digitized, and the next state is calculated based on the previous state and the time step-size Δh as follows.

$$\begin{cases} x(k+1) = x(k) + y(k) \cdot \Delta h \\ y(k+1) = y(k) + z(k) \cdot \Delta h \\ z(k+1) = z(k) + (a \cdot (|x(k)| - 1) - (y(k) + b \cdot z(k))) \cdot \Delta h \\ u(k+1) = u(k) + ((|x(k)| - 1) - z(k)) \cdot \Delta h \end{cases} \tag{6.2}$$

The equilibrium point, where the system is stable, is the solution of this equation:

$$\begin{cases} x(k) = 0 \\ z(k) = 0 \\ a \cdot (|x(k)| - 1) - (y(k) + b \cdot z(k)) = 0 \\ (|x(k)| - 1) - z(k) = 0 \end{cases} . \tag{6.3}$$

Therefore, the system has two equilibria which are located at $(\pm 1, 0, 0, 0)$. To exhibit chaotic characteristics, a dynamic system requires necessary conditions: i) has a positive LE - the trajectories are sensitive to initial conditions, and ii) has a negative sum of LEs - the phase space is bounded. The dynamic system presents different characteristics according to system parameters. A bifurcation diagram is an effective diagram to find the region of chaos in a dynamic system's solution space. Moreover, to evaluate the level of dynamic, the Lyapunov exponents, Lyapunov dimensions are calculated and compared to previous systems. Additionally, the wavelet analysis, which is a powerful tool to evaluate the periodicity of chaotic time series, is discussed.

6.4.1 Bifurcation diagram

The bifurcation diagram of the signal $x(t)$ according to parameter a is shown in Figure 6.1. The system is dynamic and unbounded for $a > 0.945$, and therefore, it is a non-chaotic region. For $a < 0.945$, the system is in the dynamic and bounded region. However, the level of dynamics varies according to a. Figure 6.2 shows the bifurcation diagram of the chaotic signal $x(t)$ according to the parameter b. For $b > 1.2$, the solution of the dynamic system starts converging to a point. Therefore, the parameter set is chosen as $a = 0.91$, and $b = 0.5$ (which can easily implemented in digital domain by a bit-right-shift operator).

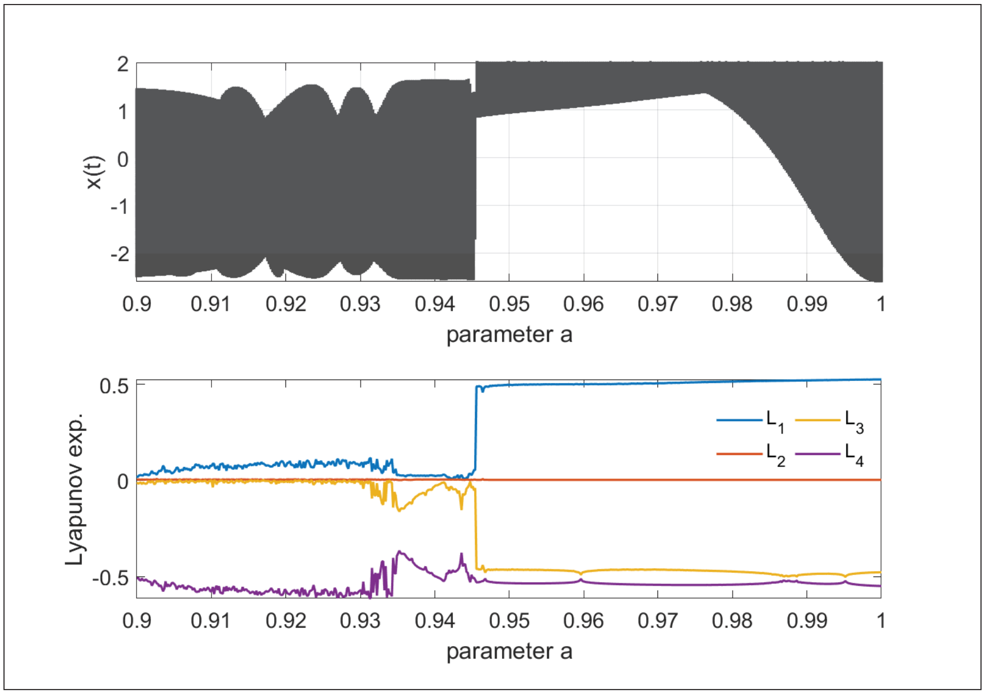


Figure 6.1 Bifurcation diagram of x(t) according to parameter *a*

6.4.2 Dynamic characteristics

A positive Lyapunov exponent is a signature of chaos in dynamic systems. Higher-dimensional dynamic systems are highly possible to have more than one positive Lyapunov exponents where the data stretch out exponentially in more than one dimension. However, comparison of positive Lyapunov exponents between different dynamic systems with different data space is un-fair and cannot conclude about the dynamic characteristics. Therefore, the Lyapunov dimension, which is calculated based on the ratio of Lyapunov exponents, is used to compare and evaluate the dynamic level.

- **Lyapunov exponents:** The Lyapunov exponents of the system are defined as follows.

$$\begin{aligned} L_1 L_2 &= \lim_{t\to\infty} \frac{1}{t} \log \frac{\|\partial y(t)\|}{\|\partial y(0)\|} \\ L_3 L_4 &= \lim_{t\to\infty} \frac{1}{t} \log \frac{\|\partial u(t)\|}{\|\partial u(0)\|}. \end{aligned} \tag{6.4}$$

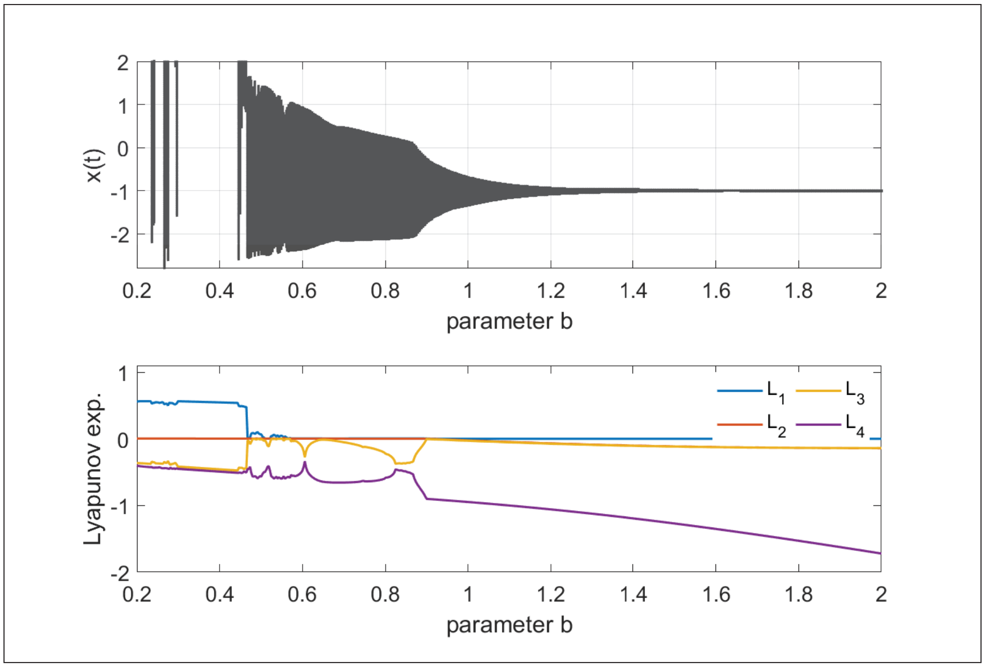


Figure 6.2 Bifurcation diagram of x(t) according to parameter *b*

The Lyapunov exponents of the chaotic system in (6.1) are: $L_1 = 0.060527$, $L_2 = 0.001127$, $L_3 = 0.000248$, and $L_4 = -0.5619$ for a set of parameter $[a, b] = [0.91, 0.5]$, while the initial conditions are chosen as, $x_0 = 0.02$, $y_0 = 0.02$, $z_0 = 0.01$, and $u_0 = 0.05$. The system has two positive Lyapunov exponents which enable chaotic characteristics for a dynamic system. The divergence of (6.1) is evaluated based on the following conditions:

$$\begin{gathered}\frac{\partial \dot{x}}{\partial x} + \frac{\partial \dot{y}}{\partial y} + \frac{\partial \dot{z}}{\partial z} + \frac{\partial \dot{u}}{\partial u} + \frac{\partial \dot{v}}{\partial v} = -1 < 0, \\ L = L_1 + L_2 + L_3 + L_4 < 0.\end{gathered} \tag{6.5}$$

- **Lyapunov dimension:** The rate of entropy production or the level of stochastic is evaluated by the knowledge of the Lyapunov spectrum, which is possible to obtain a parameter called the Lyapunov dimension (D_L) Chlouverakis & Sprott (2005). The level of complexity is proportional to the value of D_L.

$$D_L = j + \frac{1}{|L_{j+1}|} \sum_{i=1}^{j} L_i, \tag{6.6}$$

where k is the maximum integer such that the sum of k largest LEs is positive. According to the previous calculation, the given system has $k = 3$. Thus, the Lyapunov dimension is:

$$D_L = 3 + \frac{L_1 + L_2 + L_3}{|L_4|} = 3.11 \tag{6.7}$$

Table 6.1 shows the complexity and dynamic level of 4D hyperchaotic systems in previous works and this work. The system in Bao *et al.* (2018) has a total of 9 terms with 3 nonlinear terms which use multipliers and absolute functions. The authors in Prousalis *et al.* (2017) proposed a chaotic system with a total of 12 terms, in which 3 terms are nonlinear functions using multipliers. The above systems have a lower Lyapunov dimension than our proposed system. The system in Prakash *et al.* (2020) has a total of 9 terms with 2 nonlinear terms as ours, while it has a higher Lyapunov dimension than us. However, it is noted that the complexity of the multiplier is higher than the absolute function in terms of hardware implementation. The proposed 4D chaotic system in Chen *et al.* (2018) provides highest Lyapunov dimension. However, this system has a total of 13 terms with 5 nonlinear terms of multipliers, which requires high computational cost.

Table 6.1 Comparison of different 4D hyperchaotic systems

4D chaos	**Total No. of terms**	**No. of nonlinear terms**	**Name of nonlinear terms**	**Lyapunov Dimension**
Prakash *et al.* (2020)	9	2	Multipliers	3.14
Bao *et al.* (2018)	9	3	Multipliers Absolute	3.0622
Chen, Tang, Li & Zhong (2018)	13	5	Multipliers	3.32
Prousalis *et al.* (2017)	12	3	Multipliers	3.099
This work	9	2	Absolute	3.11

- **Wavelet analysis:** Wavelet transformation is a powerful tool to evaluate the periodicity of chaotic time series. The scaled index is calculated based on the inner scalogram of the chaotic signals Benítez *et al.* (2010). The scale indexes, which are in the range $[0; 1]$, are extracted according to the change of parameter a, corresponding to the Lyapunov spectrum as seen in Figure 6.3. The degree of periodicity is proportional to the scale index. A higher value of the scale index indicates that the signal has higher non-periodicity.

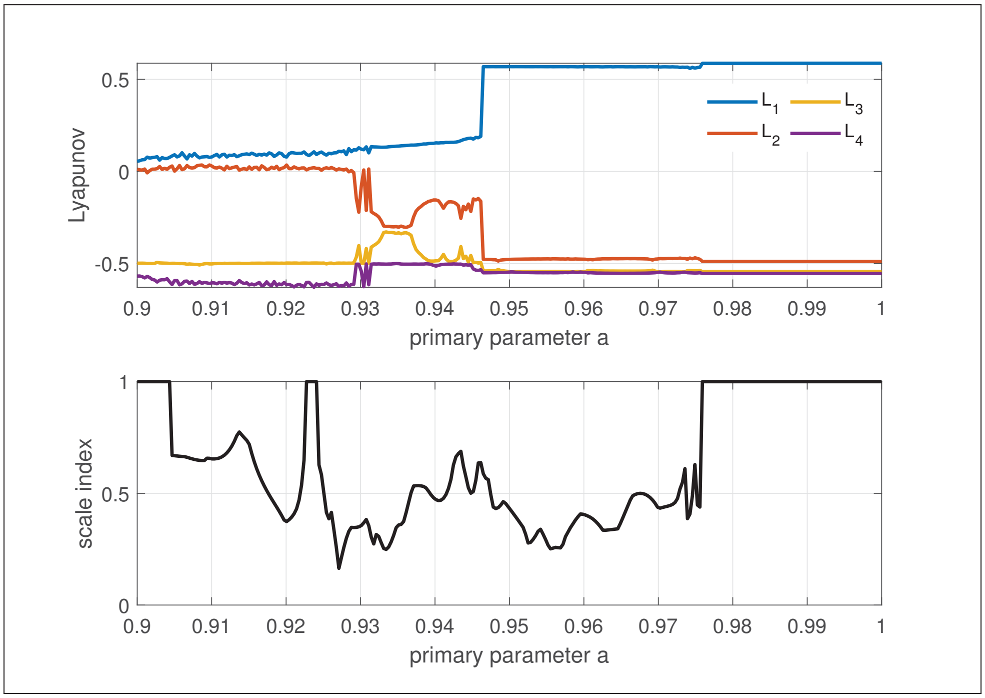


Figure 6.3 Lyapunov spectrum according to the parameter *a*

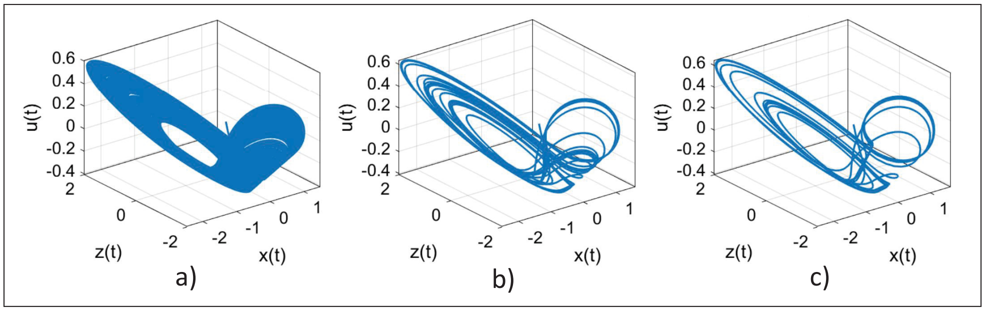


Figure 6.4 Transition from chaotic region to periodic region with parameter *a* - a) $a = 0.91$, b) $b = 0.93$, and c) $a = 0.94$.

6.4.3 Transitions to chaotic region

Dynamic systems present chaotic characteristics in some regions with specific values of the system parameters. Figure 6.4 shows the transition from a chaotic region with, $a = 0.91$ to periodic regions with $a = 0.93$ or $a = 0.94$. This result is compatible with the wavelet scale index extracted previously. The scale index i_{scale} is 0.69 with $a = 0.91$, while it is lower for $a = 0.93$ and $a = 0.94$. A slight variation in parameter a can drive the system out of the chaotic region. However, given the implementation in digital domain, the chaotic region of the system is preserved.

6.5 Hardware implementation

In what follows, we present the implementation of the 4D hyperchaotic system in hardware design and the proposed PRNG scheme. The block design of the proposed PRNG is presentd in Figure 6.5. The 4D hyperchaotic system is implemented using 16-bit fixed-point data representation (1-bit for the sign, and 12-bit for the fraction). To reduce the implementation cost, the Euler numerical technique is used to implement the chaotic system in the digital domain. Then, n-bit data are truncated from chaotic signals to use in data post-processing block. n-bit random numbers are collected for statistical tests. Post-processing block is designed based on parameter n, which is defined at synthesis and will be evaluated to find the best truncated position. In this context, the hardware Co-Simulation is chosen to optimize the time to design and verification. The 4D hyperchaotic system and data post-processing blocks are implemented and run in the Zynq-7000 board, then the random bits are collected for random tests using Matlab.

6.5.1 Four-dimensional chaotic system

To implement a continuous chaotic system, first, we need to represent the system state in discrete time by the vector $S_k = [x_k, y_k, z_k, u_k]$. Then, the differential equations, which indicate the

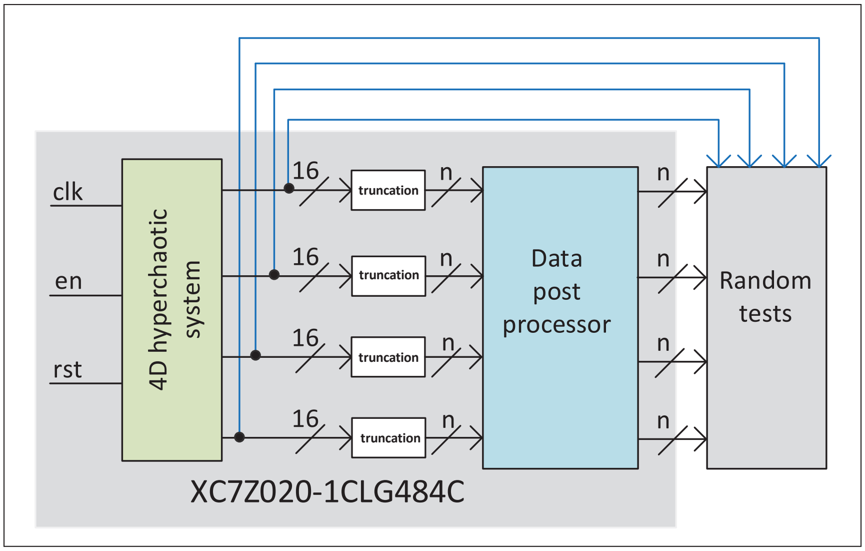


Figure 6.5 System overview of the proposed PRNG with hardware implementation in FPGA. The width of data in post-processing block (n) is defined at synthesis

system state-changing by time, is calculated as

$$F(S_k) = \begin{cases} F_x = y_k \\ F_y = z_k \\ F_z = a \cdot (|x_k| - 1) - (y_k + b \cdot z_k) \\ F_u = (|x_k| - 1) - z_k. \end{cases} \tag{6.8}$$

Numerical techniques are used to approximate the next system state in discrete time based on the previous state and its changing over time. Among several numerical techniques, the Euler method, which requires one calculation per numerical iteration, and in return produces less accurate results compared to other more complex techniques, is used in this work. The upcoming system state vector S_{k+1} is calculated as

$$S_{k+1} = S_k + \Delta h \cdot F(S_k), \tag{6.9}$$

where $\Delta h = 2^{-8}$ is the discrete step size. Thus, the time step is $t_{k+1} = t_k + \Delta h$.

6.5.2 Data post-processing

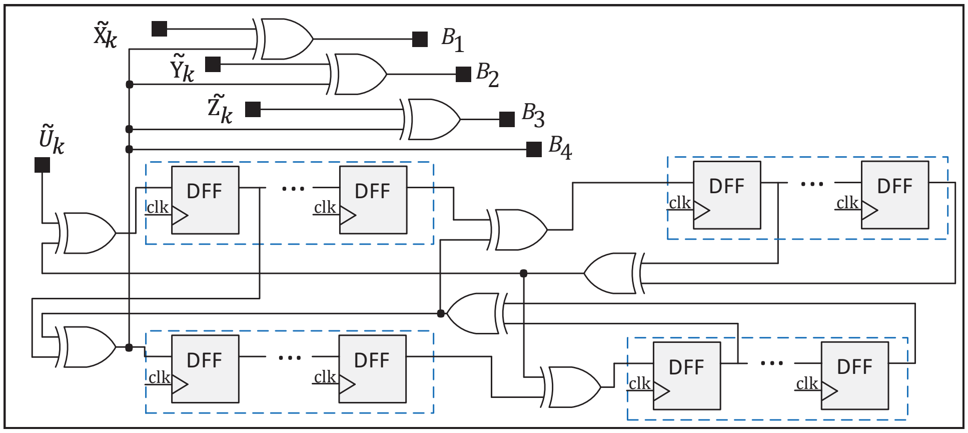


Figure 6.6 The post-processing design in the Xilinx System Generator

A post-processing data block, which is applied in truncated data from four chaotic signals. The hardware implementation uses four SHIFT registers and XOR-operations as proposed in Rozic *et al.* (2016); Pareschi *et al.* (2006). The truncated data $\tilde{U}_k$ is utilized to feed a data scrambler which is as depicted in Figure 6.6. Then the result is XOR-ed with other truncated data to generate random bitstreams. Due to the low self-oscillator frequency of the chaotic system (the internal frequency oscillator) compared to the calculating frequency (the clock cycle of one execution), only a number of least significant bits (LSBs) are used in the post-processing part to generate random bits. To choose the proper truncated position in data width, we evaluate the entropy of the truncated data according to the truncation position. The Shannon entropy of truncated data $\tilde{X}_k$ is defined as

$$H(\tilde{X}_k) = -\sum_{i=0}^{N-1} p_i \log_2 p_i, \tag{6.10}$$

where p_i is probability of data at the symbol $i \subset [0, N-1]$. The total number of possible outcomes of data is $N = 2^{N_b}$. Thus, the average entropy per bit is calculated as:

$$\bar{E} = \frac{H(\tilde{X}_k)}{N_b}. \tag{6.11}$$

The average entropy, which can be observed in Figure 6.7, decreases according to increasing

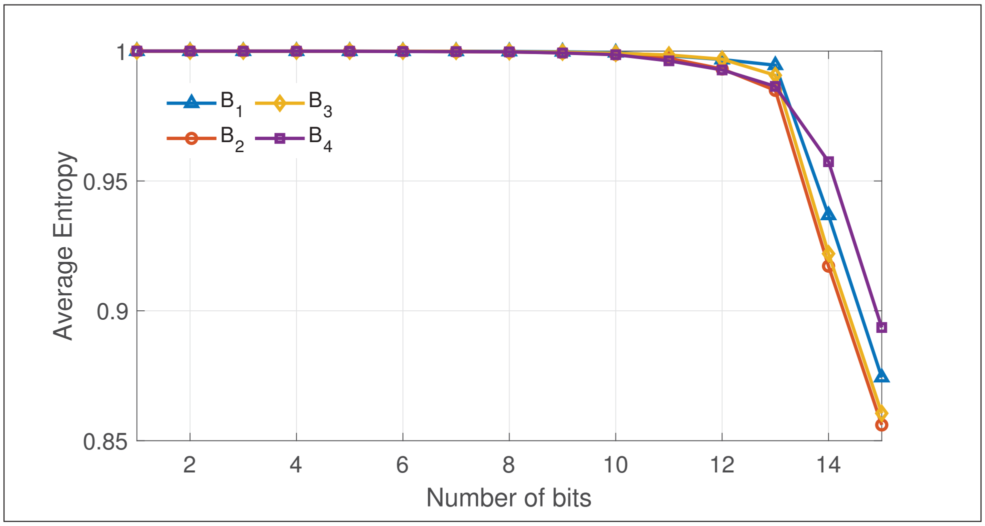


Figure 6.7 The variation of average entropy of random data depending on the truncated position

the number of bits that remained in truncated data or the truncated position. If the truncated data has more than 12 bits, the average entropy drops dramatically. Thus, in this work, 10-bit truncated data $[\tilde{X_k}, \tilde{Y_k}, \tilde{Z_k}, \tilde{U_k}]$ are used in the post-processing part. The enable signal en and the reset signal rst are single-bit signals that control the timing of the whole block design and synchronize the block with the connected processing system. The clock signal clk is generated from the processing system. In this design, the parameter $a = 0.91$ is represented by a 5 bit fractional. To reduce the implementation cost, all the system parameters are fixed and defined before synthesizing. The initial conditions, which are $[0.01, 0.01, 0.02, 0.02]$, are defined by the initial values of four registers embedded in the design. Initialization for registers is supported by most Xilinx FPGA chips. The chaos-based pseudo-random bit generator is deployed on the XC7Z020-1CLG484C chip of the Xilinx Zynq-7000 board. The experimental setup is illustrated in Figure 6.8. The random binary outputs are observed using an oscilloscope with a system clock of 50 MHz.

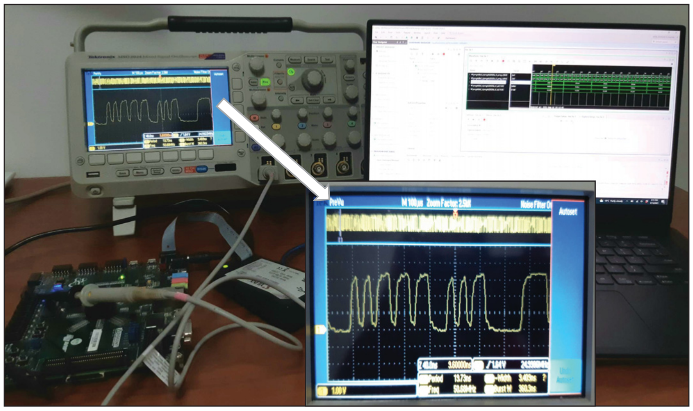


Figure 6.8 Hardware validation of the proposed PRNG. Only the LSB of the random data B_1 is observed in oscilloscope. The system is running at 50 MHz

6.5.3 Implementation Results

In this section, we compare our work with recent developed linear PRNG for lightweight devices and chaos-based PRNG as presented in Table 6.2. Linear PRNGs have advantages of low device resource requirement and high throughput. LUT-SR requires the lowest amount of area resource while providing the highest throughput (19.5 Gbps) Thomas & Luk (2013). The recent proposed linear feedback shift register (LFSR) PRNG in Syafalni, Jonatan, Sutisna, Mulyawan & Adiono (2022), which is implemented in a low-cost Xilinx FPGA PYNQ Z1 board, also provides high throughput at 10.7 Gbps while requiring 241 LUTs and 91 FFs. However, the above PRNGs are not passed statistical tests such as NIST and TestU01. The authors in Bakiri, Couchot & Guyeux (2018a) improve the performance of linear PRNGs such as multiple recursive Taus88, and LFSR113 by sing chaotic iteration as post-processing data. In such a way, these PRNGs are able to pass the statistical test NIST and TestU01. Two above linear PRNGs requires DSPs and BRAM.

Additionally, we compare our work with chaotic PRNGs which aim to constrained devices as in Table 6.2. The logistic (LG) map is a famous chaotic map that is used in many applications including generating random numbers. The LG-PRNG in Dabal & Pelka (2014) utilizes a number of device resources that are similar to our design but requires 16 DSPs. The maximum throughput of 7.5 Gbps also is very close to our design. However, this PRNG does not pass TestU01 Bakiri *et al.* (2018a). Recently, an exponential chaotic map (EM) is used for PRNG as presented in Cardoso *et al.* (2021). This work achieves the maximum of 2.4 Gbps which is far from our design while requiring more than 2× number of device resources. The authors in Bakiri, Guyeux, Couchot, Marangio & Galatolo (2018b) proposed chaotic iteration (CI)-PRNG using a linear PRNG as input. In table 6.2, we compare their best performance by using the CIPRNG for LFRS113 input linear PRNG with our work. The maximum throughput of that work is 6.95 Gbps which is lower than our design while utilizing a slightly lower number of device resources. Our previous work in Nguyen, Bui, Gagnon, Giard & Kaddoum (2021b) proposed a 5D chaotic-based PRNG and the fourth-folding Runge-Kutta method is used to implement a chaotic system that requires more device resources than our current design. However, the maximum throughput is much lower than the current design. We also note that by using the absolution function, the current design does not require DSPs in FPGA chip. The implementation presented in this paper requires under 1% of the available LUTs, FFs, and no DSP blocks in the XC7Z020-1CLG484C chip. The maximum frequency of the implementation is of 214 MHz which corresponds to a throughput of 8.56 Gbps for a random bitstream (four 10-bit length random data at the frequency of 214 MHz). The improvement we achieved by using the Euler method to solve chaotic systems and using simple implementation non-linear function (absolute function). Additionally, different from other chaotic systems, an absolute function is used twice in two non-linear terms without adding more implementation cost.

6.6 Randomness performance evaluation

This section presents several statistical tests to evaluate the randomness of the harvested binary bitstreams including the data distribution, the correlation test, and two standard tests. Figure 6.9

Table 6.2 Comparison in device utilization of existing chaos-based PRNGs and the proposed PRNG

PRNG	**LUT**	**FF**	**RAM**	**DSP**	**TP** Gbps	**NIST**	**TestU01**
LUT-SR Thomas & Luk (2013)	64	64	0	0	19.5	PASS	NO
LFSR Syafalni *et al.* (2022)	241	91	0	0	10.7	NO	NO
Taus88 Bakiri *et al.* (2018a)	358	830	2	6	5.2	PASS	PASS
LFSR113 Bakiri *et al.* (2018a)	357	853	2	6	5.3	PASS	PASS
LG Dabal & Pelka (2014)	313	842	–	16	7.5	PASS	NO
EM Cardoso *et al.* (2021)	915	1101	23	6	2.4	PASS	–
CIPRNG-LFRS113 Bakiri *et al.* (2018b)	273	344	–	–	6.95	PASS	PASS
5D-PRNG Nguyen *et al.* (2021b)	2017	3458	0	8	6.78	PASS	PASS
This work	314	642	0	0	8.56	PASS	PASS

shows the chaotic signal $x(n)$ in time domain with different initial conditions. Here, $LSB = 2^{-11}$ is approximately 0.0005. This figure compares the chaotic signal $x(n)$ at the initial value $S(0)$=[0.0010, 0.01, 0.02, 0.01] and at the other initial value $S'(0)$ = [0.0015, 0.01, 0.02, 0.01]. After several time steps, different initial conditions yield divergent outputs where the difference is a random signal.

6.6.1 Distribution

First, 400-k 10-b-truncated chaotic signals and generated random data after post-processing are collected and used in the distribution analysis. Figure 6.10 shows the distributions of the truncated chaotic signal $X\tilde(k)$ before post-processing and random sequence B_1, after post-processing. A uniform distribution is expected in random data. That means it is an equal possibility of any given output value. This figure shows the non-uniform distribution of the truncated data and the like-wise uniform distribution of the generated random sequence B_1. This also shows the

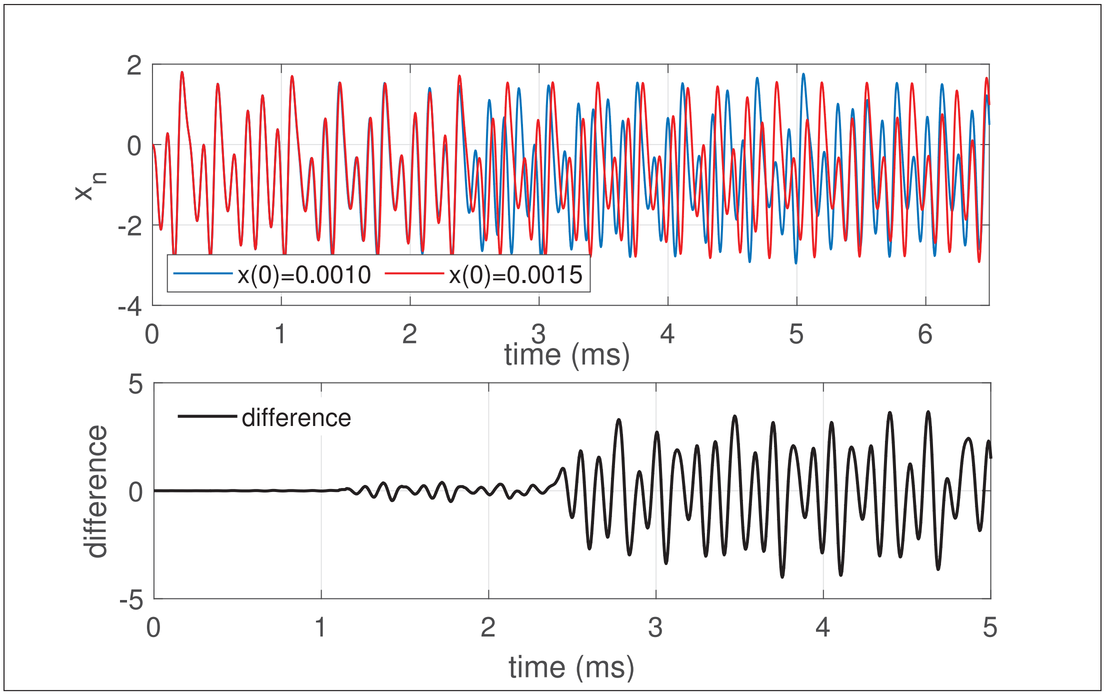


Figure 6.9 The system's senitivity to the initial condition

effect of the post-processing part on the truncated data in removing the bias. Similar results are observed for B_2 to B_4.

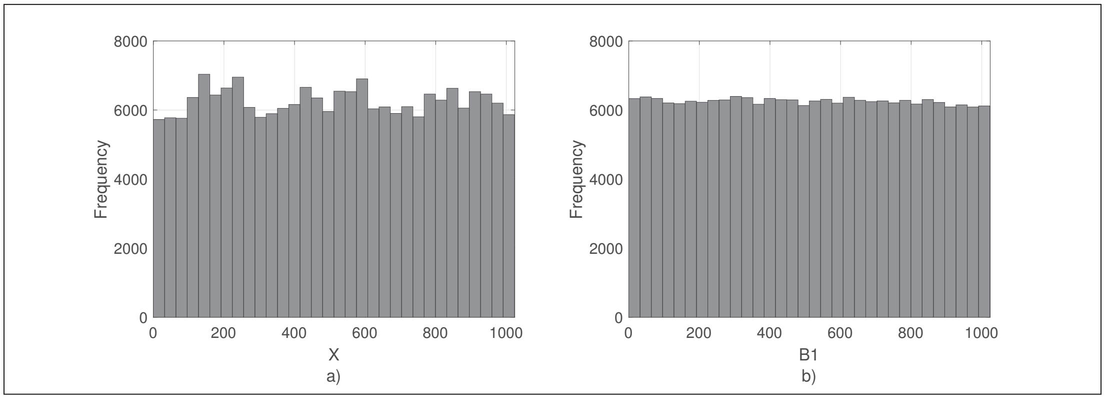


Figure 6.10 Histogram of a) the 10-bit raw truncated chaotic signal X - before post-processing, and b) the 10-bit random signal B_1 - after post-processing

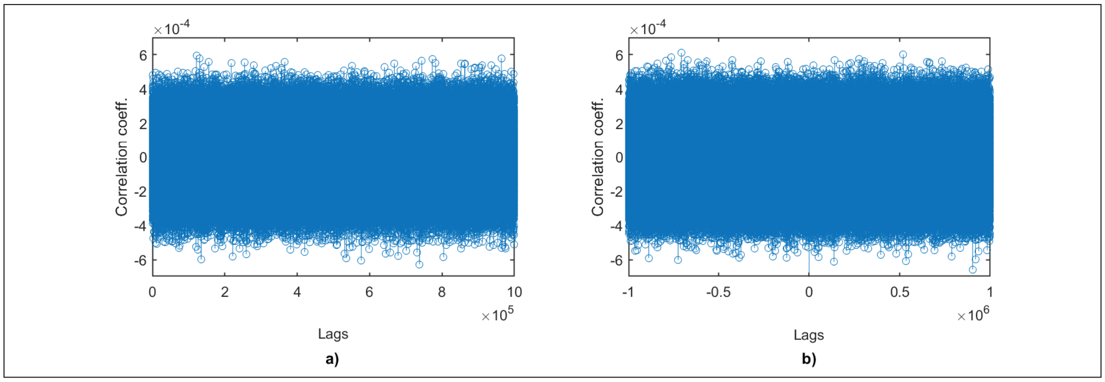


Figure 6.11 a) auto-correlation of intra-signal random bitstream B_1 and b) crosscorrelation of inter-signal random bitstreams B_1 vs. B_2

6.6.2 Correlation test

To enable the utilization of five truncated chaotic signal outputs in the generation of random bits, we evaluate the intra-bitstreams and inter-bitstreams correlations after post-processing. Pearson correlation is commonly used to measure the degree of similarity between two linearly related variables Demir & Ergun (2018-11). The correlation coefficient between two serial variables at its k^{th} time delay is calculated as:

$$r_{b_1b_2}(k) = \frac{c_{b_1b_2}(k)}{s_{b_1}s_{b_2}} \qquad k = 0, \pm 1, \pm 2, ..., \tag{6.12}$$

where $c_{b_1b_2}(k)$ expresses the cross-covariance between two variables $b_{1,t}$ and $b_k 2, t$ at the k^{th}-lags, which is calculated as:

$$c_{b_1b_2}(k) = \begin{cases} \frac{1}{T}\sum_{t=1}^{T-k}(b_{1,t} - \bar{b_1})(b_{2,t+k} - \bar{b_2}) & k = 1, 2, ..., \\ \frac{1}{T}\sum_{t=1}^{T+k}(b_{2,t} - \bar{b_2})(b_{1,t-k} - \bar{b_1}) & k = -1, -2, ..., \end{cases} \tag{6.13}$$

where s_{b_1} and s_{b_2} are the standard deviations of the series $\sqrt{c_{b_1b_1}(0)}$ and $\sqrt{c_{b_2b_2}(0)}$, respectively. The auto-correlation is evaluated within a binary stream to detect any repeated sequences. The correlation test results are illustrated in Figure 6.11, which expresses the auto-correlation of intra-signal random bitstreams B_1 and the cross-correlation of inter-signal random bitstreams B_1

and B_2. The average similarity of 0.05% demonstrates an uncorrelated relationship between the time series and the non-periodic random outputs. Finally, these correlation results of inter-signal random bitstreams support the parallel utilization of chaotic signals in hyperchaotic systems.

6.6.2.1 NIST SP800-22 test

The National Institute of Standards and Technology (NIST) developed a standard statistical test called the NIST SP800-22 providing multiple statistical tests to evaluate the randomness of a given data including 15 sub-tests. This test-suite is widely accepted among the scientist community and industry Rukhin *et al.* (2010). A null hypothesis is made if the sequence is random. A critical value (α) is used to determine the reference point for the decision to accept or reject the null hypothesis. If the output of each test (P-value) exceeds the critical value α, the evaluated sequence is considered as random with the confidence of 99%, and the null hypothesis is accepted Pareschi *et al.* (2012). 10-Mb streams are collected from the 4 output ports for NIST-tests. Table 6.3 shows the test results with the P-values for each test at each output port. The table shows that our PRNG has high average P-values, compared to the critical value ($\alpha = 0.01$) which indicates a high level of statistical randomness.

6.6.3 TestU01 test

TestU01 is a software library providing intensive empirical tests for an independent random variable. In this work, we applied two battery tests including Rabbit and Alphabit to 2^{25} generated random bits. Thus, there is a total of 17 subtests in the Alphabit battery test, and 39 subtests in the Rabbit battery test. Table 6.4 presents the test results in which all these subtests are passed for all the bitstreams B_1, B_2, B_3, and B_4. We also noted that many existing commercial PRNGs have gone through failure in complicated tests L'Ecuyer & Simard (2007).

Table 6.3 Results of NIST SP800-22 for the proposed PRNG for streams of 10Mb

NIST SP-800.22	$\mathbf{B_1}$	$\mathbf{B_2}$	$\mathbf{B_3}$	$\mathbf{B_4}$
Monobit	0.7398	0.7677	0.6289	0.8993
Frequency	0.5823	0.4477	0.2140	0.9230
Runs	0.6384	0.6208	0.7360	0.4658
Longest run 1's	0.7946	0.4948	0.8903	0.5591
Rank	0.0186	0.6082	0.0853	0.1398
DFT	0.8640	0.5499	0.9884	0.8435
Cumulative sum(*)	0.9030	0.5861	0.7278	0.6021
Overlapping	0.3812	0.2455	0.9069	0.2979
Non-overlapping(*)	0.4887	0.4877	0.4777	0.4960
Linear complexity	0.9455	0.2420	0.6473	0.5476
Maurers universal	0.2871	0.6074	0.9172	0.3481
Entropy	0.7313	0.8551	0.8826	0.7212
Serial(*)	0.3017	0.7704	0.9266	0.5441
Rand. Excursions(*)	0.3607	0.6918	0.4770	0.6623
Rand. Variant	0.3879	0.4975	0.4256	0.5365

(*): the average value.

Table 6.4 TestU01 statistical test results on the sample of 2^{25} generated random bits

TestU01	$\mathbf{B_1}$	$\mathbf{B_2}$	$\mathbf{B_3}$	$\mathbf{B_4}$
Rabbit	39/39	39/39	39/39	39/39
Alphabit	17/17	17/17	17/17	17/17

6.6.4 Image encryption test

To test the proposed PRNG in a security application, an application, which utilizes the generated bits to encrypt images, is developed on the Linux operating system in an ARM-Cortex A9 processor. The hardware architecture is detailed in Figure 6.12, where the design-under-test (DUT), being the controller for the PRNG, communicates with the application developed in the Zynq-7000 SoC via the AXI-Lite communication protocol. The AXI-Lite controller provides standard communication channels to the application through the AXI interconnection bridge. This bridge provides multi-channel for high-speed and high-performance data transferring. This bridge also has channels for reading/writing to/from memory via AXI-DMA to store the random

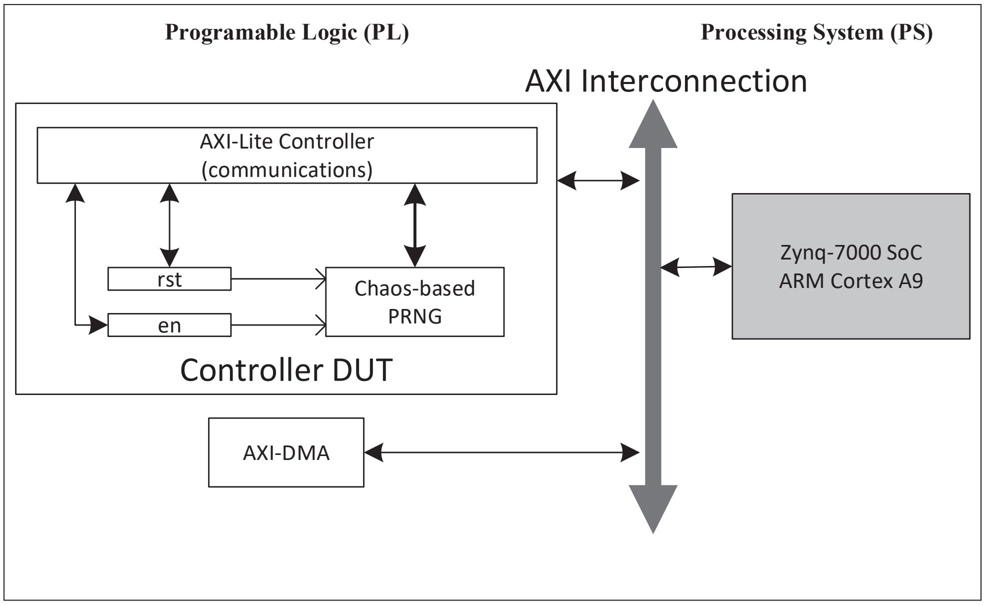


Figure 6.12 Hardware architecture of data encryption based on the chaos-based PRNG

numbers. The application developed on the processing system side controls memory access to read random data from Chaos-based PRNG and write to memory. Then, it reads random data from the memory to encrypt images. In addition, the application provides control signals (rst and en) to reset and enable chaos-based PRNG. The image encryption algorithm is presented in Algorithm 6.1.

Figure 6.13 compares the histogram of the real image and encrypted image. The encrypted image has an uniform distribution which shows the high randomness and security of the encryption. Table 6.5 shows image encryption evaluation, including the number of pixels change rate (NPCR), the unified average changing intensity (UACI), and the image entropy, using the proposed PRNG for different image sizes .

6.7 Conclusion

In this paper, we have developed and extended a 3D chaotic system to a 4D hyperchaotic system that uses a low-complexity non-linear function: the absolute function. The aims of these developments are to increase the random bit throughput whereas decrease the device utilization

Algorithm 6.1 Image encryption

```
Input: 24-bit image data $I(R, G, B)$ whose size $[W \times H]$ and random sequence $R$
Output: Encrypted image $E(R, G, B)$
long $L = [W \times H]$;
Read 24 bit image data $I[3, L]$;
Separate 8 bit RGB data $IR = I(1; 1 : L)$, $IG = I(2, 1 : L)$, $IB = L(3, 1 : L)$;
for $i = 1; i \leq L; i++$ do
    Read random number $R(i)$;
    Reform random number to 8-bit random data $P(i) = (R(i) \ \& \ 0XFF$ ;
end for
for $i = 1; i \leq L; i++$ do
    Encrypted Red data $ER(i) = IR(i) \oplus P(i)$;
    Encrypted Green data $EG(i) = IG(i) \oplus P(i)$;
    Encrypted Blue data $EB(i) = IB(i) \oplus P(i)$;
end for
Write encrypted data $E = (E_R, E_G, E_B)$ to encrypted image.
```

Table 6.5 Image entropy, *NPCR*, and *UACI* test results for image encryption

Image	**Size**	**NPCR(%)**	**UACI(%)**	**Entropy**
cameraman	256× 256	99.8	32.79	7.9999
pool	510× 383	99.9	38.87	7.9997
airplane	512× 512	99.8	33.26	7.9998

of the PRNG based on the proposed chaotic system. The PRNG is designed in FPGA and deployed on the XC7Z020-1CLG484C chip. The implementation results show that our PRNG requires under 1% of the available FPGA hardware resources while it achieves the maximum frequency of 214 MHz. Thus, the maximum throughput can be obtained at the output random data is 8.56 Gbps. The randomness of the generated binary bits is verified using the NIST SP800-22 and TestU01 modern statistical tests. An image encryption application, which is developed using the chaos-based PRNG as a random key generator, was provided to demonstrate the utility of the proposed PRNG in security applications. Future work will employ the proposed PRNG in high-throughput and high-speed data encryption such as video streaming.

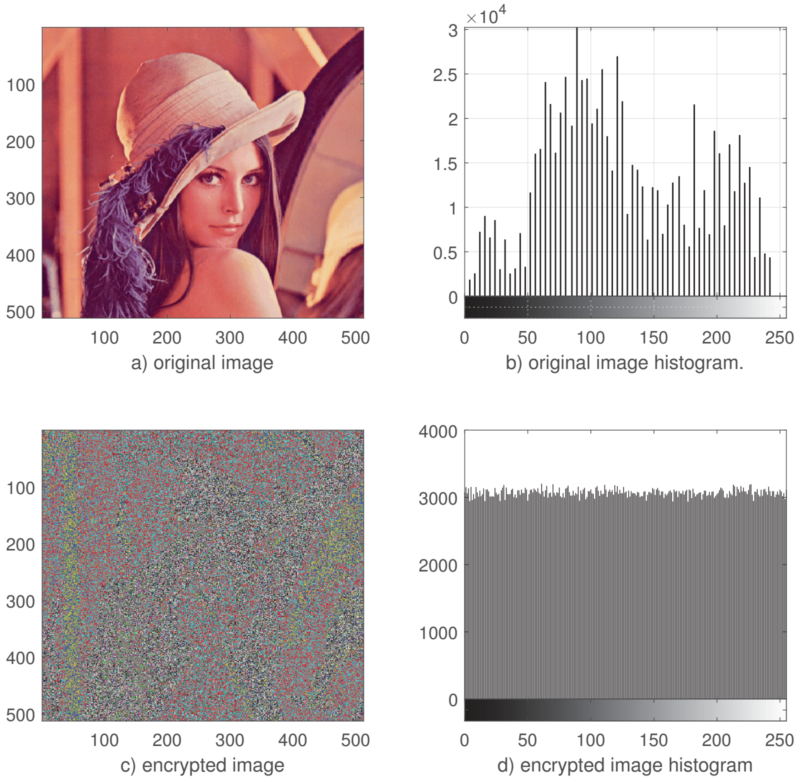


Figure 6.13 Comparison between original image and encrypted image

CONCLUSION AND RECOMMENDATIONS

7.1 Conclusion

The random number generator is a critical component that is responsible for generating public keys, private keys, and other kinds of random numbers. It serves in devices and applications from daily life activities, such as games, lotteries, to scientific applications, such as data secrecy, statistics, cryptography, and security. The ever-increasing number of devices places new challenges and requirements on random number generators. Software-based RNGs, which are advantageous in implementation and development (by coding), are not suitable for constrained devices in IoT networks. Hardware-based RNGs were developed to achieve high quality randomness for randomly generated bits while consuming low device resources, including the number of transistors (in analog designs), number of arithmetic components, and flipflops (in digital designs).

The existing solutions for RNG designs include physical entropy sources, chaotic maps, continuous chaotic systems, and chaos laser semiconductors. As per previous investigation and analysis, physical entropy sources like thermal noise, jitter noise, and metastability are not reliable random sources for high-security applications due to their unknown statistical distributions and limited dynamic ranges. For a reliable random number generator, non-linear, dynamic systems are suitable to use as random sources due to two main reasons: (i) they yeild deterministic mathematical expressions and (ii) the dynamic systems's characteristics cover randomness and stochastic. Therefore, dynamic systems have been used in RNGs for decades, where chaotic systems are commonly used due to their high sensitivity to the initial conditions, and unpredictable and stochastic motions.

Therefore, the main goals of this dissertation are to design and implement chaos-based RNGs that provide the following characteristics: good energy efficiency (consume less power to generate one random bit), compact hardware design with lower device resources, and high quality of randomness for data secrecy and security applications. With these aims in mind, there are three main aspects of this dissertation: (i) designing and analyzing novel chaotic systems that benefit the research goals, (ii) implementing the systems considering the power consumption and device's resources, and (iii) verifying the randomness and applying the RNGs in engineering applications. In this context, Chapter 2 provides our research methodologies for TRNG and PRNG designs. Multiple design tools in both software design and hardware implementation are utilized in this dissertation. Moreover, the aforementioned aspects are further detailed as follows.

- Chapter 3 highlights the novel 3D continuous chaotic system, which is used for the TRNG, designed in analog circuits. A non-linear element based on the tanh function, which is easily designed in an analog circuit using a differential amplifier, is utilized. From this research, we extend the previous 3D chaotic system to a novel 4D chaotic system which contributes to increasing the RNG throughput. Moreover, a four-dimensional system provides higher dynamics which increases the randomness. This work is presented in Chapter 4. In Chapters 5 and 6, we propose hyperchaotic systems of five dimensions and four dimensions, respectively. The motivation of using hyperchaos is to achieve higher throughputs at the outputs. In those Chapters, due to the digital implementation of the PRNG, a non-linear element based on arithmetic absolute function is utilized in these chaotic systems.
- In this dissertation, we provide both TRNG and PRNG design and implementation. The chaotic systems for TRNGs presented in Chapter 3 and Chapter 4 are implemented at the transistor level using the 130nm CMOS TSMC technology. The difficulties in designing analog circuits for chaotic systems lie in controlling the initial conditions and driving circuit parameters to enable the chaos phenomenon of the circuit. The digital implementation of hyperchaotic systems in Chapters 5 and 6 is performed in Xilinx FPGA chips to provide high-performance PRNGs.

- The designated TRNGs and PRNGs are verified and tested in the sense of randomness. Multiple tests are performed on the generated random bits. Moreover, the designs are applied in engineering applications. In Chapter 3, a one-time pad cryptosystem is designed using the proposed TRNG which achieves high-security performance.

In addition, a TRNG chip is fabricated using 130nm CMOS technology. It provides four chaotic signal outputs and a post-processing which provides ready-to-use random bits. Moreover, a secure data storage and PRNG is developed using a Xilinx FPGA chip. A standard high speed USB interface is used to provide communication between the device and the high-level software applications.

7.2 Future works

Following the works presented in this dissertation, the following research directions and development could be extended in the future.

7.2.1 Fractional-order chaotic systems

In system design, fractional-order chaotic systems are potential directions to investigate due to their higher dynamic characteristics. The chaotic systems presented in this dissertation are integer chaotic systems that are formed by ordinary differential equations. Although chaos phenomenon has been discovered and chaos theory has been evolving for many years, scientists still do not fully understand it. Most mathematical analysis developed and used to define chaotic regions and conditions of chaotic occurrence, are fragmentary and only help us understand one-side of the problem. Fractional-order representations of a chaotic system allow us to describe and model a real object more accurately than the classical "integer" methods. Therefore, the dynamic characteristics of chaotic systems could be analyzed more accurately. Currently, the implementation of fractional-order chaotic systems has many obstacles at the circuit level, both in analog and digital circuits, due to their reliance on memory elements. However, the development of memristor devices at the transistor level opens an opportunity to implement fractional-order dynamic systems at a lower cost.

7.2.2 Development of IoT secure devices

The products presented in this dissertation are in the early stage of development. Many works and research could be done to make them reach maturity. The TRNG chip should provide standard communication interfaces such as I2C, UART, etc. The secure data storage device utilizes the chaos-based PRNG, which provides high throuhgput random bitstreams in many security applications for developers and end-users, could be developed in future. This device aims to provide developers a reliable random device that can be used in software applications or software platforms, such as secure video streaming. In addition, this device could be used by end-users for secure data storage and authentication, where the keys would belong to the end-users.

Author's Contributions

The outcomes of the author's Ph.D. research are the articles listed below, published and submitted to Q1 journals from IEEE and Springer journals.

Nguyen, N., Pham-Nguyen, L., Nguyen, M.B. & Kaddoum, G. (2020). A low power circuit design for chaos-key based data encryption. *IEEE Access*, 8, 104432-104444

Nguyen, N., Kaddoum, G., Pareschi, F. et al. (2020). A fully CMOS true random number generator based on hidden attractor hyperchaotic system. *Nonlinear Dynamics*, 102, 2887–2904

Nguyen, N., Bui, T., Gagnon, G., Giard, P., & Kaddoum, G. (2021). Designing a pseudo-randombit Generator with a novel 5D-hyperchaotic system. *IEEE Transactions on Industrial Electronics*, early access

Nguyen, N., Bui, T., Gagnon, G., Giard, P., & Kaddoum, G. (2021). Continuous-Chaos-Based Pseudo RandomNumber Generator for Hardware-Constrained Devices. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*. Manuscript submitted for publication

Beside the above articles that contribute to the main contents of this dissertation, other publications that the author was involved in and which are not included in this dissertation are follow.

Nguyen, N., Kaddoum, G., & Gagnon, F. (2018). Implementation of a chaotic true random number generator based on fuzzy modeling. *16th IEEE International New Circuits and Systems Conference (NEWCAS)*, 238-242

Nguyen, N., Kaddoum, K., & Giard, P. (2020). A fully CMOS circuit design for an autonomous-continuous chaotic system. *IEEE Eighth International Conference on Communications and Electronics*, 3, 189-192

In addition to the research articles, the author has contributions to industrial applications with a standalone random number generator chip, which is fabricated in the TSMC 130nm CMOS technology.

BIBLIOGRAPHY

Abzhanova, T., Dolzhikova, I. & James, A. P. (2018-08). Implementation of True Random Number Generator based on Double-Scroll Attractor circuit with GST memristor emulator. *International Conference on Computing and Network Communications(CoCoNet)*, pp. 95–102.

Addabbo, T., Alioto, M., Fort, A., Pasini, A., Rocchi, S. & Vignoli, V. (2007). A Class of Maximum-Period Nonlinear Congruential Generators Derived From the R&eacute;nyi Chaotic Map. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 54(4), 816–828.

Addabbo, T., Alioto, M., Fort, A., Rocchi, S. & Vignoli, V. (2006). Efficient Post-Processing Module for a Chaos-based Random Bit Generator. *2006 13th IEEE International Conference on Electronics, Circuits and Systems*, pp. 1224–1227.

Akgul, A., Calgan, H., Koyuncu, I., Pehlivan, I. & Istanbullu, A. (2016). Chaos-based engineering applications with a 3D chaotic system without equilibrium points. *Nonlinear Dynamics*, 84(2), 481–495.

Akgul, A., Arslan, C. & Aricioglu, B. (2019). Design of an Interface for Random Number Generators based on Integer and Fractional Order Chaotic Systems. *Chaos Theory and Applications*, 1(1), 1–18. Number: 1.

Alligood, K. T., Sauer, T. D. & Yorke, J. A. (2000). *Chaos: an introduction to dynamical systems* (ed. Corr. 3. print). Springer.

Arellano-Cardenas, O., Molina-Lozano, H., Moreno-Cadenas, J., Gomez-Castaneda, F. & Flores-Nava, L. (2000). CMOS analog neurofuzzy prototype based on ANFIS. *Proceedings - IEEE International Symposium on Circuits and Systems (ISCAS)*, 3, 726–729 vol.3.

Argyris, A., Pikasis, E. & Syvridis, D. (2016). Gb/s One-Time-Pad Data Encryption With Synchronized Chaos-Based True Random Bit Generators. *Journal of Lightwave Technology*, 34(22), 5325–5331.

Avaroglu, E., Koyuncu, I., Ozer, A. B. & Turk, M. (2015). Hybrid pseudo-random number generator for cryptographic systems. *Nonlinear Dyn.*, 82(1-2), 239–248.

Azzaz, M. S., Tanougast, C., Sadoudi, S. & Dandache, A. (2009). Real-time FPGA implementation of Lorenz's chaotic generator for ciphering telecommunications. *2009 Joint IEEE North-East Workshop on Circuits and Systems and TAISA Conference*, pp. 1–4.

Bae, S.-G., Kim, Y., Park, Y. & Kim, C. (2017). 3-Gb/s High-Speed True Random Number Generator Using Common-Mode Operating Comparator and Sampling Uncertainty of D Flip-Flop. *IEEE Journal of Solid-State Circuits*, 52(2), 605–610.

Bakiri, M., Couchot, J.-F. & Guyeux, C. (2018a). CIPRNG: A VLSI Family of Chaotic Iterations Post-Processings for $\mathbf{F}_2$ -Linear Pseudorandom Number Generation Based on Zynq MPSoC. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 65(5), 1628–1641.

Bakiri, M., Guyeux, C., Couchot, J.-F., Marangio, L. & Galatolo, S. (2018b). A Hardware and Secure Pseudorandom Generator for Constrained Devices. *IEEE Transactions on Industrial Informatics*, 14(8), 3754–3765.

Bao, B., Zou, X., Liu, Z. & Hu, F. (2013). Generalized memory element and chaotic memory system. *International Journal of Bifurcation and Chaos*, 23(08), 1350135.

Bao, H., Wang, N., Bao, B., Chen, M., Jin, P. & Wang, G. (2018). Initial condition-dependent dynamics and transient period in memristor-based hypogenetic jerk system with four line equilibria. *Communications in Nonlinear Science and Numerical Simulation*, 57, 264–275.

Barakat, M. L. (2013). Generalized Hardware Post-processing Technique for Chaos-Based Pseudorandom Number Generators. *ETRI Journal*, 35(3), 448–458.

Barati, K., Jafari, S., Sprott, J. C. & Pham, V.-T. (2016). Simple Chaotic Flows with a Curve of Equilibria. *International Journal of Bifurcation and Chaos*, 26(12), 1630034.

Benítez, R., Bolós, V. & Ramírez, M. (2010). A wavelet-based tool for studying non-periodicity. *Computers & Mathematics with Applications*, 60(3), 634–641.

Bonny, T., Al Debsi, R., Majzoub, S. & Elwakil, A. S. (2019). Hardware Optimized FPGA Implementations of High-Speed True Random Bit Generators Based on Switching-Type Chaotic Oscillators. *Circuits, Systems, and Systems Processing*, 38(3), 1342–1359.

Bonvarlet, L. (2017). *Gemalto and Oracle IoT CS Secure and Industrialize your IoT deployment*. Paris: Gemalto.

Bosco, A. K. D., Sato, N., Terashima, Y., Ohara, S., Uchida, A., Harayama, T. & Inubushi, M. (2017). Random Number Generation From Intermittent Optical Chaos. *IEEE Journal of Selected Topics in Quantum Electronics*, 23(6), 1–8.

Bouda, J., Krhovjak, J., Matyas, V. & Svenda, P. (2009). Towards True Random Number Generation in Mobile Environments. In Jøsang, A., Maseng, T. & Knapskog, S. J. (Eds.), *Identity and Privacy in the Internet Age* (vol. 5838, pp. 179–189). Berlin, Heidelberg: Springer Berlin Heidelberg.

Callegari, S. (2016, May). True random number generators as configware for mixed mode programmable systems on chip. *IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1850-1853.

Callegari, S., Setti, G. & Langlois, P. (1997). A CMOS tailed tent map for the generation of uniformly distributed chaotic sequences. *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2, 781-784 vol.2.

Callegari, S., Rovatti, R. & Setti, G. (2005). Embeddable ADC-based true random number generator for cryptographic applications exploiting nonlinear signal processing and chaos. *IEEE Transactions on Signal Processing*, 53(2), 793–805.

Carbajal-Gomez, V. H., Tlelo-Cuautle, E., Muñoz-Pacheco, J. M., de la Fraga, L. G., Sanchez-Lopez, C. & Fernandez-Fernandez, F. V. (2019). Optimization and CMOS design of chaotic oscillators robust to PVT variations: INVITED. *Integration*, 65, 32–42.

Cardoso, M. B. R., da Silva, S. S., Nardo, L. G., Passos, R. M., Nepomuceno, E. G. & Arias-Garcia, J. (2021). A New PRNG Hardware Architecture Based on an Exponential Chaotic Map. *2021 IEEE International Symposium on Circuit and System (ISCAS)*, pp. 1–5.

Çavuşoğlu, Ü., Panahi, S., Akgül, A., Jafari, S. & Kacar, S. (2019). A new chaotic system with hidden attractor and its engineering applications: analog circuit realization and image encryption. *Analog Integrated Circuits and Signal Processing*, 98(1), 85–99.

Chandrasekaran, S. T., Karnam, V. E. G. & Sanyal, A. (2020). 0.36-mW, 52-Mbps True Random Number Generator Based on a Stochastic Delta–Sigma Modulator. *IEEE Solid-State Circuits Letters*, 3, 190–193.

Chen, G., Mao, Y. & Chui, C. K. (2004). A symmetric image encryption scheme based on 3D chaotic cat maps. *Chaos, Solitons & Fractals*, 21(3), 749–761.

Chen, L., Tang, S., Li, Q. & Zhong, S. (2018). A new 4D hyperchaotic system with high complexity. *Mathematics and Computers in Simulation*, 146, 44–56.

Chen, W., Che, W., Bi, Z., Wang, J., Yan, N., Tan, X., Wang, J., Min, H. & Tan, J. (2009). A 1.04 μW Truly Random Number Generator for Gen2 RFID tag. *2009 IEEE Asian Solid-State Circuits Conference*, pp. 117–120.

Chen, X., Li, B., Wang, Y., Liu, Y. & Yang, H. (2016). A Unified Methodology for Designing Hardware Random Number Generators Based on Any Probability Distribution. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 63(8), 783–787.

Chlouverakis, K. E. & Sprott, J. (2005). A comparison of correlation and Lyapunov dimensions. *Physica D: Nonlinear Phenomena*, 200(1-2), 156–164.

Cho, K. & Miyano, T. (2017). Design and Test of Pseudorandom Number Generator Using a Star Network of Lorenz Oscillators. *International Journal of Bifurcation & Chaos*, 27(12), 1750184.

Cicek, I., Pusane, A. E. & Dundar, G. (2014). A new dual entropy core true random number generator. *Analog Integrated Circuits and Signal Processing*, 81(1), 61–70.

Corinto, F., Krulikovskyi, O. V. & Haliuk, S. D. (2016). Memristor-based chaotic circuit for pseudo-random sequence generators. *2016 18th Mediterranean Electrotechnical Conference (MELECON)*, pp. 1–3.

Dabal, P. & Pelka, R. (2014). A study on fast pipelined pseudo-random number generator based on chaotic logistic map. *17th International Symposium on Design and Diagnostics of Electronic Circuits & Systems*, pp. 195–200.

Danesh, M., Venkatasubramaniyan, A. B., Kapoor, G., Ramesh, N., Sadasivuni, S., Chandrasekaran, S. T. & Sanyal, A. (2020). Unified Analog PUF and TRNG Based on Current-Steering DAC and VCO. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 28(11), 2280-2289.

Datta, B. N. (2004). Chapter 7 - Stability, Inertia, And Robust Stability. In Datta, B. N. (Ed.), *Numerical Methods for Linear Control Systems* (pp. 201-243). San Diego: Academic Press.

De Jesus Lopes Soares, E., Alencar Mendonca, F. & Viana Ramos, R. (2014). Quantum Random Number Generator Using Only One Single-Photon Detector. *IEEE Photonics Technology Letters*, 26(9), 851–853.

Demir, K. & Ergun, S. (2018-11). Analytical Modeling of Continuous-Time Chaos Based Random Number Generators. *2018 New Generation of CAS (NGCAS)*, pp. 106–109.

Deniz, H. I., Cam Taskiran, Z. G. & Sedef, H. (2018-07). An Analog Chaotic Lorenz Circuit Based on CCII+ and Multiplier. *41st IEEE International Conference on Telecommunications and Signal Processing (TSP)*, pp. 1–5.

Dichtl, M. (2007). Bad and Good Ways of Post-processing Biased Physical Random Numbers. In Biryukov, A. (Ed.), *Fast Software Encryption* (vol. 4593, pp. 137–152). Berlin, Heidelberg: Springer Berlin Heidelberg.

Dmitrieva, L. A., Kuperin, Y. A., Smetanin, N. M. & Chernykh, G. A. (2016). Method of calculating Lyapunov exponents for time series using artificial neural networks committees. *2016 Days on Diffraction (DD)*, pp. 127–132.

Elwakil, A. S., Salama, K. N. & Kennedy, M. P. (2002-12). An Equation For Generating Chaos And Its Monolithic Implementation. *International Journal of Bifurcation and Chaos*, 12(12), 2885–2895.

Elwakil, A. & Kennedy, M. (2000a). Generic RC Realizations of CHUA's Circuit. *International Journal of Bifurcation and Chaos*, 10, 1981–1985.

Elwakil, A. & Kennedy, M. (2000b). Chua's circuit decomposition: a systematic design approach for chaotic oscillators. *Journal of the Franklin Institute*, 337(2-3), 251–265.

Elwakil, A. & Kennedy, M. (2001). Construction of classes of circuit-independent chaotic oscillators using passive-only nonlinear devices. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 48(3), 289–307.

Ergun, S. (2018-10). Vulnerability Analysis of a Chaos-Based Random Number Generator. *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 3331–3334.

Ergun, S. & Ozoguz, S. (2005). A Truly Random Number Generator Based on a Continuous-Time Chaotic Oscillator for Applications in Cryptography. In *Computer and Information Sciences - ISCIS 2005* (vol. 3733, pp. 205–214). Berlin, Heidelberg: Springer Berlin Heidelberg.

Ericsson. (2017). *5G Security scenarios and solutions*. Ericsson White paper.

Fatemi-Behbahani, E., Ansari-Asl, K. & Farshidi, E. (2016). A New Approach to Analysis and Design of Chaos-Based Random Number Generators Using Algorithmic Converter. *Circuits, Systems, and Signal Processing*, 35(11), 3830–3846.

Figliolia, T., Julian, P., Tognetti, G. & Andreou, A. (2016). A true Random Number Generator using RTN noise and a sigma delta converter. *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 17–20.

Fozin Fonzin, T., Srinivasan, K., Kengne, J. & Pelap, F. (2018). Coexisting bifurcations in a memristive hyperchaotic oscillator. *AEU - International Journal of Electronics and Communications*, 90, 110–122.

Gabriel, C., Wittmann, C., Hacker, B., Mauerer, W., Huntington, E., Sabuncu, M., Marquardt, C. & Leuchs, G. (2012). A high-speed secure quantum random number generator based on vacuum states. *2012 Conference on Lasers and Electro-Optics (CLEO)*, pp. 1-2.

Galajda, P., Guzan, M. & Petrzela, J. (2016). Implementation of a custom Chua's diode for chaos generating applications. *2016 26th International Conference Radioelektronika*, pp. 66–70.

Gandhi, G. & Roska, T. (2009). MOS-integrable circuitry for multi-scroll chaotic grid realization: A SPICE-assisted proof. *International Journal of Circuit Theory and Applications*, 37(3), 473–483.

García-Guerrero, E., Inzunza-González, E., López-Bonilla, O., Cárdenas-Valdez, J. & Tlelo-Cuautle, E. (2020). Randomness improvement of chaotic maps for image encryption in a wireless communication scheme using PIC-microcontroller via Zigbee channels. *Chaos, Solitons & Fractals*, 133, 109646.

Gerosa, A., Bernardini, R. & Pietri, S. (2001). A fully integrated 8-bit, 20 MHz, truly random numbers generator, based on a chaotic system. *2001 Southwest Symposium on Mixed-Signal Design (Cat. No.01EX475)*, pp. 87-92.

Giakoumis, A., Volos, C. K., Munoz-Pacheco, J. M., del Carmen Gomez-Pavon, L., Stouboulos, I. N. & Kyprianidis, I. M. (2018). Text encryption device based on a chaotic random bit generator. *2018 IEEE 9th Latin American Symposium on Circuits & Systems (LASCAS)*, pp. 1–5.

Giard, P., Kaddoum, G., Gagnon, F. & Thibeault, C. FPGA implementation and evaluation of discrete-time chaotic generators circuits. *IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society*, pp. 3221–3224.

Gong, L., Zhang, J., Liu, H., Sang, L. & Wang, Y. (2019). True Random Number Generators Using Electrical Noise. *IEEE Access*, 7, 125796–125805.

Goos, G., Hartmanis, J., van Leeuwen, J., Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J. M., Kobsa, A., Mattern, F., Mitchell, J. C., Naor, M., Nierstrasz, O., Rangan, C. P. & Steffen, B. *Lecture Notes in Computer Science*. Springer.

Guanyu Wang, Dajun Chen, Jianya Lin & Xing Chen. (1999). The application of chaotic oscillators to weak signal detection. *IEEE Transactions on Industrial Electronics*, 46(2), 440–444.

Gunhee Han & Sanchez-Sinencio, E. (1998). CMOS transconductance multipliers: a tutorial. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 45(12), 1550–1563.

Guo, C. & Zhou, Y. (2018-05). A True Random Number Generator Based on the Multiple-Scrolls Chaotic System. *2018 3rd International Conference on Information Systems Engineering (ICISE)*, pp. 98–103.

Haliuk, S., Krulikovskyi, O., Vovchuk, D. & Corinto, F. (2022). Memristive Structure-Based Chaotic System for PRNG. *Symmetry*, 14(1), 68.

Hamano, K. (2005). The Distribution of the Spectrum for the Discrete Fourier Transform Test Included in SP800-22. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E88-A(1), 67–73.

Hasimoto-Beltran, R., Al-Masalha, F. & Khokhar, A. (2011). *Performance evaluation of chaotic and conventional encryption on portable and mobile platforms*. Springer Berlin Heidelberg.

Hazwani, S., Khan, S., Siddiqi, M. U., Al-Khateeb, K. A., Habaebi, M. H. & Shahid, Z. (2014). Randomness Analysis of Pseudo Random Noise Generator Using 24-Bits LFSR. *2014 5th International Conference on Intelligent Systems, Modelling and Simulation*, pp. 772-774. doi: 10.1109/ISMS.2014.141.

Herrero-Collantes, M. & Garcia-Escartin, J. C. (2017). Quantum random number generators. *Reviews of Modern Physics*, 89(1).

Holman, W., Connelly, J. & Dowlatabadi, A. (1997). An integrated analog/digital random noise source. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 44(6), 521–528.

Hsueh, J.-C. & Chen, V. H.-C. (2019). An ultra-low voltage chaos-based true random number generator for IoT applications. *Microelectronics Journal*, 87, 55–64.

Hu, H., Liu, L. & Ding, N. (2013). Pseudorandom sequence generator based on the Chen chaotic system. *Computer Physics Communications*, 184(3), 765–768.

Hua, Z. & Zhou, Y. (2020). Two-Dimensional Sine Chaotification System With Hardware Implementation. *IEEE Transactions on Industrial Informatics*, 16(2), 11.

Hua, Z., Zhou, B. & Zhou, Y. (2019). Sine Chaotification Model for Enhancing Chaos and Its Hardware Implementation. *IEEE Transactions on Industrial Electronics*, 66(2), 1273–1284.

Irfan, M., Ali, A., Khan, M. A., Ehatisham-ul Haq, M., Mehmood Shah, S. N., Saboor, A. & Ahmad, W. (2020). Pseudorandom Number Generator (PRNG) Design Using Hyper-Chaotic Modified Robust Logistic Map (HC-MRLM). *Electronics*, 9(1), 104.

Ivanescu, M. (2001). Chapter 9 - Control. In Marghitu, D. B. (Ed.), *Mechanical Engineer's Handbook* (pp. 611-714). San Diego: Academic Press.

Jafari, S. & Sprott, J. (2013). Simple chaotic flows with a line equilibrium. *Chaos, Solitons & Fractals*, 57, 79–84.

K P, A. & Azeem, M. (2013). CMOS Based Current Mode Defuzzifier Circuit for Analog Fuzzy Inference System. *Journal of Innovation in Electronics and Communication*, 3, 1–5.

Kaplan, J. L. & Yorke, J. A. (1979). Chaotic behavior of multidimensional difference equations. *Functional Differential Equations and Approximation of Fixed Points*, pp. 204–227.

Khan Mohammadi, A., Enne, R., Hofbauer, M. & Zimmermanna, H. (2015). A Monolithic Silicon Quantum Random Number Generator Based on Measurement of Photon Detection Time. *IEEE Photonics Journal*, 7(5), 1–13.

Kim, E., Lee, M. & Kim, J.-J. (2017a). 8.2 8Mb/s 28Mb/mJ robust true-random-number generator in 65nm CMOS based on differential ring oscillator with feedback resistors. *2017 IEEE International Solid-State Circuits Conference (ISSCC)*, pp. 144–145.

Kim, M., Ha, U. & Lee, K. J. (2017b). A 82-nW Chaotic Map True Random Number Generator Based on a Sub-Ranging SAR ADC. *IEEE Journal of Solid-State Circuits*, 52(7), 1953–1965.

Kocarev, L., Szczepanski, J., Amigo, J. & Tomovski, I. (2006). Discrete Chaos-I: Theory. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 53(6), 1300–1309.

Kote, V., Molata, V. & Jakovenko, J. (2012). Improved Structure of True Random Number Generator with Direct Amplification of Analog Noise. *Electroscope*, 1–6.

Koyuncu,Ismail Turan, a. O. A. (2017). The design and realization of a new high speed FPGA-based chaotic true random number generator. *Computers & Electrical Engineering*, 58, 203–214.

Koyuncu, I., Ozcerit, A. T. & Pehlivan, I. (2013). An analog circuit design and FPGA-based implementation of the Burke-Shaw chaotic system. *Optoelectronics and Advanced Materials-Rapid Communications*, 1–5.

Koyuncu, s., Tuna, M., Pehlivan, h., Fidan, C. B. & Alcin, M. (2020). Design, FPGA implementation and statistical analysis of chaos-ring based dual entropy core true random number generator. *Analog Integrated Circuits and Signal Processing*, 102(2), 445–456.

Kwok, S.-H., Ee, Y.-L., Chew, G., Zheng, K., Khoo, K. & Tan, C.-H. (2011). A Comparison of Post-Processing Techniques for Biased Random Number Generators. In *Information Security Theory and Practice. Security and Privacy of Mobile Devices in Wireless Communication* (vol. 6633, pp. 175–190). Berlin, Heidelberg: Springer Berlin Heidelberg.

Lü, J., Chen, G. & Cheng, D. (2004). A New Chaotic System and Beyond: The Generalized Lorenz-Like System. *International Journal of Bifurcation and Chaos*, 14(05), 1507–1537.

Lacharme, P. (2008). Post-Processing Functions for a Biased Physical Random Number Generator. In Nyberg, K. (Ed.), *Fast Software Encryption: 15th International Workshop, FSE 2008, Lausanne, Switzerland, February 10-13, 2008, Revised Selected Papers* (pp. 334–342). Berlin, Heidelberg: Springer Berlin Heidelberg.

Laurenciu C., S. D. (2015). Low cost and energy, thermal noise driven, probability modulated random number generator. *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 2724–2727.

L'Ecuyer, P. & Simard, R. (2007). TestU01: A C Library for Empirical Testing of Random Number Generators. *ACM Transactions on Mathematical Software*, 33(4).

Lee, K., Lee, S.-Y., Seo, C. & Yim, K. (2018). TRNG (True Random Number Generator) Method Using Visible Spectrum for Secure Communication on 5G Network. *IEEE Access*, 6, 12838–12847.

Li, B., Liao, X. & Jiang, Y. (2019). A novel image encryption scheme based on improved random number generator and its implementation. *Nonlinear Dynamics*, 95(3), 1781–1805.

Li, P., Guo, Y., Guo, Y., Fan, Y., Guo, X., Liu, X., Li, K., Shore, K. A., Wang, Y. & Wang, A. (2018). Ultrafast Fully Photonic Random Bit Generator. *Journal of Lightwave Technology*, 36(12), 2531–2540.

Ling Cong & Wu Xiaofu. (2001). Design and realization of an FPGA-based generator for chaotic frequency hopping sequences. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 48(5), 521–532.

Liu, D., Liu, Z., Li, L. & Zou, X. (2016). A Low-Cost Low-Power Ring Oscillator-Based Truly Random Number Generator for Encryption on Smart Cards. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 63(6), 608–612.

Liu, L., Wang, B. & Wei, S. (2018). *Reconfigurable Cryptographic Processor*. Singapore: Springer Singapore.

Liu, W. & Chen, G. (2003). A New Chaotic System and Its Generation. *International Journal of Bifurcation and Chaos*, 13(01), 261–267.

Liu, Y. & Tong, X. (2016). Hyperchaotic system-based pseudorandom number generator. *IET Information Security*, 10(6), 433–441.

Liu, Y., Qin, Z., Liao, X. & Wu, J. (2020). Cryptanalysis and enhancement of an image encryption scheme based on a 1-D coupled Sine map. *Nonlinear Dyn.*, 100(3), 2917–2931.

Ma, X. (2016). Quantum random number generation. *NPJ Quantum Information*, 1–9.

Mao, Y., Chen, G. & Lian, S. (2004). A Novel Fast Image Encryption Scheme Based on 3D Chaotic Baker Maps. *International Journal of Bifurcation and Chaos*, 14(10), 3613–3624.

Mao, Y., Cao, L. & Liu, W. (2006). Design and FPGA Implementation of a Pseudo-Random Bit Sequence Generator Using Spatiotemporal Chaos. *2006 International Conference on Communications, Circuits and Systems*, pp. 2114–2118.

Massari, N., Gasparini, L., Tomasi, A., Meneghetti, A., Xu, H., Perenzoni, D., Morgari, G. & Stoppa, D. (2016). 16.3 A 16x16 pixels SPAD-based 128-Mb/s quantum random number generator with -74dB light rejection ratio and 6.7ppm/C bias sensitivity on temperature. *2016 IEEE International Solid-State Circuits Conference (ISSCC)*, pp. 292–293.

Mathew, S. K., Srinivasan, S., Anders, M. A., Kaul, H., Hsu, S. K., Sheikh, F., Agarwal, A., Satpathy, S. & Krishnamurthy, R. K. (2012). 2.4 Gbps, 7 mW All-Digital PVT-Variation Tolerant True Random Number Generator for 45 nm CMOS High-Performance Microprocessors. *IEEE Journal of Solid-State Circuits*, 47(11), 2807–2821.

Mathew, S. K., Suresh, S., Anders, M. A., Kaul, H., Hsu, S. K., Farhana, S., Amit, A., Sudhir, S. & RamK. Krishnamurthy, F. (2016). An All-Digital Edge Racing True Random Number Generator Robust Against PVT Variations. *IEEE Journal of Solid-State Circuits*, 51(4), 1022–1031.

Matsumoto, M., Yasuda, S., Ohba, R., Ikegami, K., Tanamoto, T. & Fujita, S. (2008). 1200um2 Physical Random-Number Generators Based on SiN MOSFET for Secure Smart-Card Application. *2008 IEEE International Solid-State Circuits Conference - Digest of Technical Papers*, pp. 414-624.

Mendoza, J., Araque-Lameda, L. & Colina-Morles, E. (2016). Understanding chaos through a Jerk circuit. *2016 Technologies Applied to Electronics Teaching (TAEE)*, pp. 1–5.

Miyano, T. & Cho, K. (2016). Chaos-based one-time pad cryptography. *2016 International Symposium on Information Theory and Its Applications (ISITA)*, 156-160.

Modeste Nguimdo, R., Tchitnga, R. & Woafo, P. (2013). Dynamics of coupled simplest chaotic two-component electronic circuits and its potential application to random bit generation. *Chaos*, 23(4), 043122.

Morris W. Hirsch,Stephen Smale, R. L. D. (2013). Chapter 8 - Equilibria in Nonlinear Systems. In *Differential Equations, Dynamical Systems, and an Introduction to Chaos* (pp. 159-187). Academic Press.

Moysis, L., Tutueva, A., Volos, C. & Butusov, D. (2020). A Chaos Based Pseudo-Random Bit Generator Using Multiple Digits Comparison. *Chaos Theory and Applications*, 2(2), 11.

Murillo-Escobar, M. A., Cruz-Hernández, C., Cardoza-Avendaño, L. & Méndez-Ramírez, R. (2017). A novel pseudorandom number generator based on pseudorandomly enhanced logistic map. *Nonlinear Dynamics*, 87(1), 407–425.

Muthuswamy, B. (2009). *Memristor Based Chaotic Circuits*. USA: UC Berkeley.

N Nguyen, T. T., Kaddoum, G. & Gagnon, F. (2018). Implementation of a Chaotic True Random Number Generator Based on Fuzzy Modeling. *2018 16th IEEE International New Circuits and Systems Conference (NEWCAS)*, pp. 238–242.

Nejati, H., Beirami, A. & Ali, W. H. (2012). Discrete-Time Chaotic-Map Truly Random Number Generators: Design, Implementation, and Variability Analysis of the Zigzag Map. *Analog Integrated Circuits and Signal Processing*, 73, 363–374.

Nguyen, N., Pham-Nguyen, L., Nguyen, M. B. & Kaddoum, G. (2020). A Low Power Circuit Design for Chaos-Key Based Data Encryption. *IEEE Access*, 8, 104432-104444.

Nguyen, N., Kaddoum, G., Pareschi, F., Rovatti, R. & Setti, G. (2020). A fully CMOS true random number generator based on hidden attractor hyperchaotic system. *Nonlinear Dyn.*, 102(4), 2887–2904.

Nguyen, N. T. T., Bui, T. Q. T., Gagnon, G., Giard, P. & Kaddoum, G. (2021a). Designing a Pseudo-Random Bit Generator with a Novel 5D-Hyperchaotic System. *IEEE Transactions on Industrial Electronics*, 1-1.

Nguyen, N. T. T., Bui, T. Q. T., Gagnon, G., Giard, P. & Kaddoum, G. (2021b). Designing a Pseudo-Random Bit Generator with a Novel 5D-Hyperchaotic System. *IEEE Transactions on Industrial Electronics*, 1-1.

Nguyen, V. H., Kumar, S. & Song, H. (2018). A family of fully integrated CMOS chaos generators with strictly 1-D linear-piecewise chaos maps. *Journal of Computational Electronics*, 17(3), 1343–1355.

Nokia. (2017a). *Security challenges and oppertunities for 5G mobile networks*. Filand: Nokia White paper.

Nokia. (2017b). *LTE evolution for IoT connectivity*. Filand: Nokia White paper.

Ozturk, I. & Kilic, R. (2019). Higher Dimensional Baker Map and its Digital Implementation With LSB-Extension Method. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 66(12), 4780–4792.

Palacios-Luengas, L. & Duchen-Sainchez, G. I. (2012). Digital Noise Generator Design Using Inverted 1D Tent Chaotic Map. *VLSI Design*, 2012, 1–10.

Pamula, V. R., Sun, X., Kim, S. M., Rahman, F. u., Zhang, B. & Sathe, V. S. (2018). A 65-nm CMOS 3.2-to-86 Mb/s 2.58 pJ/bit Highly Digital True-Random-Number Generator With Integrated De-Correlation and Bias Correction. *IEEE Solid-State Circuits Letters*, 1(12), 237–240.

Pareschi, F., Rovatti, R. & Setti, G. (2012). On Statistical Tests for Randomness Included in the NIST SP800-22 Test Suite and Based on the Binomial Distribution. *IEEE Transactions on Information Forensics and Security*, 7(2), 491–505.

Pareschi, F., Rovatti, R. & Setti, G. (2006). Simple and effective post-processing stage for random stream generated by a chaos-based rng. *The 2006 International Symposium on Nonlinear Theory and its Applications (NOLTA2006)*, pp. 5.

Pareschi, F., Setti, G. & Rovatti, R. (2010). Implementation and Testing of High-Speed CMOS True Random Number Generators Based on Chaotic Systems. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 57(12), 3124–3137.

Park, B. K., Park, H., Kim, Y.-S., Kang, J.-S., Yeom, Y., Ye, C., Moon, S. & Han, S.-W. (2019). Practical True Random Number Generator Using CMOS Image Sensor Dark Noise. *IEEE Access*, 7, 91407–91413.

Park, M., Rodgers, J. C. & Lathrop, D. P. (2015). True random number generation using CMOS Boolean chaotic oscillator. *Microelectronics Journal*, 46(12), 1364–1370.

Patel, K., Dynes, J., Sharpe, A., Yuan, Z., Penty, R. & Shields, A. (2012). Gigacount/second photon detection with InGaAs avalanche photodiodes. *Electronics Letters*, 48(2), 111.

Patidar, V. & Sud, K. K. (2005). Bifurcation and chaos in simple jerk dynamical systems. *Pramana*, 64(1), 75–93.

Pehlivan, I. & Uyaroglu, Y. (2012). A new 3D chaotic system with golden proportion equilibria: Analysis and electronic circuit realization. *Computers and Electrical Engineering*, 38(6), 1777-1784.

Petrie, C. & Connelly, J. (2000). A noise-based IC random number generator for applications in cryptography. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 47(5), 615–621.

Petura, O., Mureddu, U., Bochard, N., Fischer, V. & Bossuet, L. (2016). A survey of AIS-20/31 compliant TRNG cores suitable for FPGA devices. *2016 26th International Conference on Field Programmable Logic and Applications (FPL)*, pp. 1-10.

Pham, V.-T., Jafari, S., Volos, C., Giakoumis, A., Vaidyanathan, S. & Kapitaniak, T. (2016a). A Chaotic System With Equilibria Located on the Rounded Square Loop and Its Circuit Implementation. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 63(9), 878–882.

Pham, V.-T., Vaidyanathan, S., Volos, C. K., Jafari, S. & Wang, X. (2016b). A Chaotic Hyperjerk System Based on Memristive Device. In *Advances and Applications in Chaotic Systems* (vol. 636, pp. 39–58). Springer International Publishing.

Prakash, P., Rajagopal, K., Koyuncu, I., Singh, J. P., Alcin, M., Roy, B. K. & Tuna, M. (2020). A Novel Simple 4-D Hyperchaotic System with a Saddle-Point Index-2 Equilibrium Point and Multistability: Design and FPGA-Based Applications. *Circuits, Systems, and Signal Processing*, 39(9), 4259–4280.

Prousalis, D. A., Volos, C. K., Stouboulos, I. N. & Kyprianidis, I. M. (2017). A hyperjerk memristive system with infinite equilibrium points. *Mathemathical methods and computational techniques in science and engineering*, pp. 020024.

Radwan, A. G., Soliman, A. M. & El-Sedeek, A. L. (2003). MOS realization of the double-scroll-like chaotic equation. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 50(2), 285–288.

Rajagopal, K., Akgul, A., Jafari, S. & Aricioglu, B. (2018). A chaotic memcapacitor oscillator with two unstable equilibriums and its fractional form with engineering applications. *Nonlinear Dynamics*, 91(2), 957–974.

Reidler, I., Aviad, Y., Rosenbluh, M. & Kanter, I. (2009). Ultrahigh-Speed Random Number Generation Based on a Chaotic Semiconductor Laser. *Physical Review Letters*, 103(2).

Rezk, A. A., Madian, A. H., Radwan, A. G. & Soliman, A. M. (2019). Reconfigurable chaotic pseudo random number generator based on FPGA. *AEU - International Journal of Electronics and Communications*, 98, 174–180.

Rezk, A. A., Madian, A. H., Radwan, A. G. & Soliman, A. M. (2020). Multiplierless chaotic Pseudo random number generators. *AEU - International Journal of Electronics and Communications*, 113, 152947.

Roman, R., Najera, P. & Lopez, J. (2011). Securing the Internet of Things. *Computer*, 44(9), 51-58.

Rovatti, R., Mazzini, G., Setti, G. & Giovanardi, A. (2003). Statistical modeling and design of discrete-time chaotic processes: advanced finite-dimensional tools and applications. *The IEEE International Symposium on Circuits and Systems, Tutorial Guide: ISCAS 2003.*, 2, 93-114.

Rozic, V., Yang, B., Dehaene, W. & Verbauwhede, I. (2016). Iterating Von Neumann's post-processing under hardware constraints. *2016 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pp. 37–42.

Rukhin, A. et al. (2010). A statistical test suite for random and pseudorandom number generators for cryptographic applications [Special Publication 800-22]. National Institute of Standards and Technology (NIST). Retrieved from: SpecialPublication800-22.

Sambas, A. (2018). A New Jerk Chaotic System with Three Nonlinearities and Its Circuit Implementation. *Proceedings of the International Conference on Industrial Engineering and Operations Management*, pp. 8.

Samuel, B., Stucky, D., Maruyama, Y., Bruschini, C., Charbon, E. & Regazzoni, F. (2013). Jailbreak Imagers: Transforming a Single-Photon Image Sensor into a True Random Number Generator. *2013 International Image Sensor Workshop*.

Satansup, J. & Tangsrirat, W. (2018). 1.5-V CMOS Current Multiplier/Divider. *International Journal of Electrical and Computer Engineering (IJECE)*, 8(3), 1478.

Satpathy, S. K., Mathew, S. K., Kumar, R., Suresh, V., Anders, M. A., Kaul, H., Agarwal, A., Hsu, S., Krishnamurthy, R. K. & De, V. (2019). An All-Digital Unified Physically Unclonable Function and True Random Number Generator Featuring Self-Calibrating Hierarchical Von Neumann Extraction in 14-nm Tri-gate CMOS. *IEEE Journal of Solid-State Circuits*, 54(4), 1074-1085.

Saxena, V. (2018). A Compact CMOS Memristor Emulator Circuit and its Applications. *2018 IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS)*, pp. 190-193.

Sciamanna, M. & Shore, K. A. (2015). Physics and applications of laser diode chaos. *Nature Photonics*, 9(3), 151–162.

Silva-Juarez, A., Rodriguez-Gomez, G., Fraga, L. G. d. l., Guillen-Fernandez, O. & Tlelo-Cuautle, E. (2019). Optimizing the Kaplan–Yorke Dimension of Chaotic Oscillators Applying DE and PSO. *Technologies*, 7(2), 38.

Stojanovski, T., Pihl, J. & Kocarev, L. (2001). Chaos-based random number generators. Part II: practical realization. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 48(3), 382–385.

Strogatz, S. H. (2015). In Strogatz, S. H. (Ed.), *Nonlinear dynamics and chaos : with applications to physics, biology, chemistry, and engineering*. Westview Press, Perseus Books Group.

Stucki, D., Burri, S., Charbon, E., Chunnilall, C., Meneghetti, A. & Regazzoni, F. (2013). Towards a high-speed quantum random number generator. *Emerging Technologies in Security and Defence; and Quantum Security II; and Unmanned Sensor Systems X*, 8899, 129 – 134.

Sunada, S., Harayama, T., Arai, K., Muramatsu, J., Yoshimura, K., Tsuzuki, K., Davis, P. & Uchida, A. (2012). Theory and Experiments of Fast Non-Deterministic Random Bit Generation Using On-Chip Chaos Lasers. *Procedia IUTAM*, 5, 190–194.

Sunar, B., Martin, W. & Stinson, D. (2007). A Provably Secure True Random Number Generator with Built-In Tolerance to Active Attacks. *IEEE Transactions on Computers*, 56(1), 109–119.

Suresh, V. B., Antonioli, D. & Burleson, W. P. (2013). On-chip lightweight implementation of reduced NIST randomness test suite. *2013 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, pp. 93–98.

Syafalni, I., Jonatan, G., Sutisna, N., Mulyawan, R. & Adiono, T. (2022). Efficient Homomorphic Encryption Accelerator With Integrated PRNG Using Low-Cost FPGA. *IEEE Access*, 10, 7753–7771.

Tadić, N., Goll, B. & Zimmermann, H. (2017). Laser Diode Current Driver With $(1 - t/T)^{-1}$ Time Dependence in 0.35- μm BiCMOS Technology for Quantum Random Number Generators. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 64(5), 510-514.

Tang, K., Man, K. & Chen, G. (2001). Digitized n-scroll attractor model for secure communications. *ISCAS 2001 IEEE International Symposium on Circuits and Systems (Cat. No.01CH37196)*, 2, 787–790.

Tchitnga, R., Nguazon, T., Louodop Fotso, P. H. & Gallas, J. A. C. (2016-03). Chaos in a Single Op-Amp–Based Jerk Circuit: Experiments and Simulations. *IEEE Transactions on Circuits and Systems II, Express Briefs*, 63(3), 239–243.

Teh, J. S., Teng, W. & Samsudin, A. (2016). A true random number generator based on hyperchaos and digital sound. *2016 3rd International Conference on Computer and Information Sciences (ICCOINS)*, pp. 264–269.

Tehranipoor, F., Wortman, P., Karimian, N., Yan, W. & Chandy, J. A. (2018). DVFT: A Lightweight Solution for Power-Supply Noise-Based TRNG Using Dynamic Voltage Feedback Tuning System. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 26(6), 1084-1097.

Thomas, D. B. & Luk, W. (2013). The LUT-SR Family of Uniform Random Number Generators for FPGA Architectures. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 21(4), 761–770.

Tlelo-Cuautle, E., Rangel-Magdaleno, J., Pano-Azucena, A., Obeso-Rodelo, P. & Nunez-Perez, J. (2015). FPGA realization of multi-scroll chaotic oscillators. *Communications in Nonlinear Science and Numerical Simulation*, 27(1-3), 66–80.

Tlelo-Cuautle, E., de la Fraga, L. G., Pham, V.-T., Volos, C., Jafari, S. & Quintas-Valles, A. d. J. (2017). Dynamics, FPGA realization and application of a chaotic system with an infinite number of equilibrium points. *Nonlinear Dynamics*, 89(2), 1129–1139.

Tlelo-Cuautle, E., Dalia Pano-Azucena, A., Guillén-Fernández, O. & Silva-Juárez, A. (2020a). *Analog/Digital Implementation of Fractional Order Chaotic Circuits and Applications*. Cham: Springer International Publishing.

Tlelo-Cuautle, E., Díaz-Muñoz, J. D., González-Zapata, A. M., Li, R., León-Salas, W. D., Fernández, F. V., Guillén-Fernández, O. & Cruz-Vega, I. (2020b). Chaotic Image Encryption Using Hopfield and Hindmarsh–Rose Neurons Implemented on FPGA. *Sensors*, 20(5), 1326.

Toker, A., Ozoguz, S., Demirkol, A., Zeki, A. & Tavas, V. (2009). Integrated cross-coupled chaos oscillator applied to random number generation. *IET Circuits, Devices & Systems*, 3(1), 1–11.

Tokunaga, C., Blaauw, D. & Mudge, T. (2008). True Random Number Generator With a Metastability-Based Quality Control. *IEEE Journal of Solid-State Circuits*, 43(1), 78–85.

Tolba, M. F., AbdelAty, A. M., Soliman, N. S., Said, L. A., Madian, A. H., Azar, A. T. & Radwan, A. G. (2017). FPGA implementation of two fractional order chaotic systems. *AEU - International Journal of Electronics and Communications*, 78, 162–172.

Trejo-Guerra, R., Tlelo-Cuautle, E., Jiménez-Fuentes, J., Sánchez-López, C., Muñoz-Pacheco, J., Espinosa-Flores-Verdad, G. & Rocha-Pérez, J. (2012). Integrated circuit generating 3- and 5-scroll attractors. *Communications in Nonlinear Science and Numerical Simulation*, 17(11), 4328–4335.

Trejo-Guerra, R., Tlelo-Cuautle, E., Carbajal-Gómez, V. & Rodriguez-Gómez, G. (2013). A survey on the integrated design of chaotic oscillators. *Applied Mathematics and Computation*, 219(10), 5113–5122.

Tsafack, N., Kengne, J., Abd-El-Atty, B., Iliyasu, A. M., Hirota, K. & Abd EL-Latif, A. A. (2020). Design and implementation of a simple dynamical 4-D chaotic circuit with applications in image encryption. *Information Sciences*, 515, 191–217.

Turan, M. S. et al. (2018). Recommendation for the Entropy Sources Used for Random Bit Generation [Special Publication 800-90B]. National Institute of Standards and Technology (NIST). Retrieved from: SpecialPublication800-90B.

UNIT42. (2020). *2020 Unit 42 IoT Threat Report - Key findings on how to reduce IoT risks*. US: Palo Alto Networks.

Vaidyanathan, S., Kingni, S. T., Sambas, A., Mohamed, M. A. & Mamat, M. (2018). A New Chaotic Jerk System with Three Nonlinearities and Synchronization via Adaptive Backstepping Control. *International Journal of Engineering & Technology*, 7(3), 1936.

Vazquez-Medina, R., Del-Río-Correa, J. L., Rojas-López, C. E. & Díaz-Méndez, J. A. (2012). Digital Chaotic Noise Using Tent Map without Scaling and Discretization Process. In *Hybrid Artificial Intelligent Systems* (vol. 7209, pp. 105–115). Berlin, Heidelberg: Springer Berlin Heidelberg.

Wang, A., Li, P., Zhang, J., Zhang, J., Li, L. & Wang, Y. (2013). 45 Gbps high-speed real-time physical random bit generator. *Optics Express*, 21(17), 20452.

Wang, A., Wang, L. & Wang, Y. (2016a). Post-processing-free 400 Gb/s true random number generation using optical heterodyne chaos. *2016 25th Wireless and Optical Communication Conference (WOCC)*, pp. 1-4.

Wang, D., Tan, X. L. & Chan, P. K. (2017-04). A 65-nm CMOS Constant Current Source With Reduced PVT Variation. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 25(4), 1373–1385.

Wang, F.-X., Wang, C., Chen, W., Wang, S., Lv, F.-S., He, D.-Y., Yin, Z.-Q., Li, H.-W., Guo, G.-C. & Han, Z.-F. (2015a). Robust Quantum Random Number Generator Based on Avalanche Photodiodes. *Journal of Lightwave Technology*, 33(15), 3319–3326.

Wang, H., Song, B., Liu, Q., Pan, J. & Ding, Q. (2014). FPGA Design and Applicable Analysis of Discrete Chaotic Maps. *International Journal of Bifurcation and Chaos*, 24(04), 1450054.

Wang, J.-m., Xie, T.-y., Zhang, H.-f., Yang, D.-x., Xie, C. & Wang, J. (2015b). A Bias-Free Quantum Random Number Generation Using Photon Arrival Time Selectively. *IEEE Photonics Journal*, 7(2), 1–8.

Wang, Q., Yu, S., Li, C., Lu, J., Fang, X., Guyeux, C. & Bahi, J. M. (2016b). Theoretical Design and FPGA-Based Implementation of Higher-Dimensional Digital Chaotic Systems. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 63(3), 401–412.

Wang, X., Wang, Y., Zhu, X. & Luo, C. A novel chaotic algorithm for image encryption utilizing one-time pad based on pixel level and DNA level. *Optics and Lasers in Engineering*, 125, 105851.

Wannaboon, C., Tachibana, M. & San-Um, W. (2018). A 0.18- μ m CMOS high-data-rate true random bit generator through $\Delta\Sigma$ modulation of chaotic jerk circuit signals. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 28(6), 063126.

Wayne, M. A. & Kwiat, P. G. (2010). Low-bias high-speed quantum random number generator via shaped optical pulses. *Optics Express*, 18(9), 9351.

Wieczorek, P. Z. & Golofit, K. (2018). True Random Number Generator Based on Flip-Flop Resolve Time Instability Boosted by Random Chaotic Source. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 65(4), 1279–1292.

Willie, J. (2010). Intel Makes a Digital Coin Tosser for Future Processors. *IEEE Spectrum*.

Xu, B., Wang, G., Iu, H. H.-C., Yu, S. & Yuan, F. (2019). A memristor–meminductor-based chaotic system with abundant dynamical behaviors. *Nonlinear Dynamics*, 96(1), 765–788.

Xu, F. & Yu, P. (2009). Global stabilization and synchronization of n-scroll chaotic attractors in a modified chua's circuit with hyperbolic tangent function. *International Journal of Bifurcation and Chaos*, 19(8), 2563–2572.

Yang, J. J., Strukov, D. B. & Stewart, D. R. (2013). Memristive devices for computing. *Nature Nanotechnology*, 8(1), 13–24.

Yang, K., Blaauw, D. & Sylvester, D. (2016). An All-Digital Edge Racing True Random Number Generator Robust Against PVT Variations. *IEEE Journal of Solid-State Circuits*, 51(4), 1022–1031.

Yang, K., Fick, D., Henry, M. B., Lee, Y., Blaauw, D. & Sylvester, D. (2014). 16.3 A 23Mb/s 23pJ/b fully synthesized true-random-number generator in 28nm and 65nm CMOS. *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pp. 280–281.

Yang, Q. & Qiao, X. (2019). Constructing a New 3D Chaotic System with Any Number of Equilibria. *International Journal of Bifurcation& Chaos*, 29(05), 1950060.

Yu, F., Wan, Q., Jin et al. Design and FPGA Implementation of a Pseudorandom Number Generator Based on a Four-Wing Memristive Hyperchaotic System and Bernoulli Map. *IEEE Access*, 7, 181884–181898.

Yujun, N., Xingyuan, W. & Mingjun, W. (2010). A new hyperchaotic system and its circuit implementation. *Communications in Nonlinear Science and Numerical Simulation*, 15(11), 3518–3524.

Zhang, C. & Wang Zhonglin. (2013). Design and realization of a new chaotic system. *Proceedings of 2013 International Conference on Sensor Network Security Technology and Privacy Communication System*, pp. 101–104.

Zhang, L. (2017a). Fixed-point FPGA model-based design and optimization for Henon map chaotic generator. *2017 IEEE 8th Latin American Symposium on Circuits & Systems (LASCAS)*, pp. 1–4.

Zhang, L. (2017b). System generator model-based FPGA design optimization and hardware co-simulation for Lorenz chaotic generator. *2017 2nd Asia-Pacific Conference on Intelligent Robot Systems (ACIRS)*, pp. 170–174.

Zhang, L., Pan, B., Chen, G., Guo, L., Lu, D., Zhao, L. & Wang, W. (2017). 640-Gbit/s fast physical random number generation using a broadband chaotic semiconductor laser. *Scientific Reports*, 7(1).

Zhou, P. & Ke, M. (2017). A New 3D Autonomous Continuous System with Two Isolated Chaotic Attractors and Its Topological Horseshoes. *Complexity*, 2017, 1–7.

Zhou, T., Zhou, Z., Yu, M. & Ye, Y. (2006). Design of A Low Power High Entropy Chaos-Based Truly Random Number Generator. *IEEE Asia Pacific Conference on Circuits and Systems APCCAS 2006*, pp. 1955–1958.

Zidan, M. A., Radwan, A. G. & Salama, K. N. (2011). The effect of numerical techniques on differential equation based chaotic generators. *Proceedings of the International Congress of Mathematicians(ICM) 2011*, pp. 1–4.