A Comprehensive Analysis of Chaos-based Secure Systems

Ava Hedayatipour*, Ravi Monani, Amin Rezaei, Mehrdad Aliasgari, and Hossein Sayadi

College of Engineering, California State University Long Beach
*ava.hedayatipour@csulb.edu

Abstract. Chaos is a deterministic phenomenon that emerges under certain conditions in a nonlinear dynamic system when the trajectories of the state variables become periodic and highly sensitive to the initial conditions. Chaotic systems are flexible, and it has been shown that communication is possible using parametric feedback control. Chaos synchronization is the basis of using chaos in communication. Chaos synchronization refers to the characteristic that the trajectories of two identical chaotic systems, each with its own unique initial conditions, converge over time.

In this paper, data extraction is performed on different chaotic equations implemented as circuits. Lorenz is the base system implemented in this paper, followed by Modified Lorenz, Chua's, L*ü*'s, and R*ö*ssler systems. Additionally, more recent systems (e.g., SprottD Attractor) are included in the data extraction process. The robust system implementations provide an alternative to software chaos and architectures, and will further reduce the required power and area. These chaotic systems serve as alternatives for quantum era computing, which will cause synchronous and asynchronous techniques to fail. The data extracted organize different modes of chaos implementation based on the ease of their fabrication in integrated circuits. Performance metrics including power consumption, area, design load, noise, and robustness to process and temperature variant are extracted for each system to demonstrate a figure of merit. The figure of merit showcases chaos equations fitting to be implemented as a transmitter/receiver with a mode of chaotic ciphering in communication.

Keywords: Chaos · Synchronization · Lorenz · CMOS · Gm-C filter.

1 Introduction

Chaos is a deterministic phenomenon that emerges under certain conditions in a nonlinear dynamic system when the trajectories of the state variable/variables become aperiodic and highly sensitive to the initial conditions. In 1963, Lorenz presented the first well known chaotic system, marking the beginning of chaos theory, a branch of non-linear system theory which has been studied intensively in recent years [1].

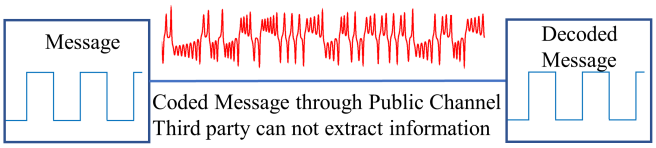


Fig. 1: Data is encrypted and transmitted over a public channel to the receiver, where it is decrypted before further processing.

Chaos can be defined as the unpredictability of a deterministic system which is highly dependent on its initial conditions. Chaos synchronization refers to the characteristic that the trajectories of two identical chaotic systems, each with its own unique initial condition, converge over time. Fig. 1 shows an overview of a chaotic encryption system. The input signal is the raw unencrypted data that is scrambled by the chaotic transmitter before being transmitted over the public channel. The public channel can be wireless as in body sensor networks or wired as in a power grid.

Chaotic systems are flexible and can be utilized for communication using parametric feedback control [2]. Lorenz-based chaotic circuits can be synchronized for communication [1]. Instead of conventional frequency synthesizers, chaos generators can be used as communication carriers. Here, the digital information modulates the chaotic signal causing the digital signal to be transmitted as a chaotic spectral signal that looks like noise to a third party.

In this paper, different modes of chaotic equations suitable for communication are implemented and simulated. The robustness of system implementations are examined and their reduction of power and area compared to software implementation is explored. The data extracted organizes these different modes of chaos implementation based on the ease of fabrication in integrated circuits. For the purpose of comprehensive analysis, various performance metrics including power consumption, area, design load, frequency range, noise, and robustness to process and temperature variant are extracted and compared for each system to demonstrate a figure of merit.

2 Chaotic Ciphering of Communication

Though chaotic communication has been known for decades, with the commercialization of computers, asymmetric and symmetric key encryption Cryptography has become a fundamental part of communication between devices such as cars, implanted medical devices, and internet of things devices (IoTs). However, commonly used cryptosystems that are used in our everyday devices are expected to fail once large quantum computers exist. Quantum computing, first proposed based on a model of the Turing machine [3], originated in the 1980s based on complex phenomenons relating to quantum-mechanical physics such as superposition, the uncertainty principle, wave particle duality, and entanglement to

perform computation. Later, it was suggested that computers performing quantum computation should be known as quantum computers. In October 2019, Google in partnership with NASA, claimed they achieved quantum computing [4]; though this claim raised some dispute [5, 6], it is still one of the most impressive milestones in quantum computing. Quantum computers can break symmetric and asymmetric cryptography keys quickly by exhaustively trying long bits of all secret keys. Therefore, methods of encryption other than symmetric and asymmetric security are gaining more importance and quickly becoming necessary.

Chaotic secure communication is advantageous in terms of having a strong real time performance, but real world implementation of these systems is still scarce. Research studies illustrate that the implementation of chaos theory in numerical simulations do not always perform well or expected for real world implementations. Therefore, in this paper, we focus on systems that can be implemented in hardware for various applications. Hardware based chaotic platforms can minimize area and power for a more efficient system implementation.

The Lorenz attractor, with its butterfly-like projection, is one of the first and most well-known chaotic attractors, though it has a complex equation. The main disadvantage of this system, however, lies in the necessity of using multipliers in realization of Lorenz equations, which is hard to implement. The modified Lorenz system proposed by Elwakil et al. [7] captures the essential behavior of Lorenz attractor with three differential equations and no multipliers. Notably, the Modified Lorenz system projects the "butterfly effect" and unsymmetrical Lorenz systems. As an example, to improve the stability or predictability of the Lorenz system, Stenflo and Leonov derived the following four-dimensional Lorenz–Stenflo system with four parameters. Another well-known chaos generator is Chua's chaotic system, which consists of multi-scroll chaotic oscillators derived from Chua's double-scroll equation.

Chua's equation has been implemented on a CMOS chip using $g_m - C$ modulators and non-linear resistors for the third order non linear differential equations. [8, 9]. The area and power consumption were very large and the design is complex. A double scroll like chaotic oscillator was implemented using the non-linearity of CMOS inverters [10].

Current conveyor based oscillators using commercially available devices implemented Chua's equations for a master-slave communication system have been used [11]. 3-, 5-, and n-scroll attractors parameters have been approximated using real devices and integrated circuits [12, 13]. Multi-scroll chaotic designs implemented using discrete components have the significant drawback of needing many external bias currents; however, V to I inverter cells which take advantage of the gate capacitance sizing can be used to address this. [14].

Other than Chua's based attractors, the Lorenz equations are an alternative method of signal ciphering for communication security. The voltage equivalant of the chaotic equations, a Lorenz chaotic oscillator was fabricated back in 1999 [15], and more recently a modified Lorenz-Stnflow with reduced power consumption was implemented as an encryption system [16]. Active control methods may be

used in the system to reduce synchronization error and can be implemented using multipliers, opamps and passive components [17].

Analog circuits exhibit process, temperature, and age variations and thus present some challenges when used to implement chaotic systems. In particular, the components must have significant degrees of matching in the transmitter and receiver. It may be possible to mitigate matching issues through feedback and techniques such as using floating gates. In recent years, neural networks have been used to eliminate unwanted noise and error and train the receiver to generate the expected outputs of chaotic systems [18, 19].

Chaotic generators can also be implemented using digital systems which eliminate the matching issues found in analog circuits. However, channel noise is still a significant issues in these implementations. FPGAs have also been widely used for the implementation of chaotic systems such as Chua's system, L$\ddot{u}$'s system, Rössler's system, Chen's system, etc [20]. However, the area and power consumption are high when using FPGA's and the implementations of these designs on integrated chips is rather rare. As the security we are targeting here mostly target portable and wearable and generally resource limited devices, use of FPGA to implement security will not be viable.

3 Design of Chaos for Integrated Circuits

Chaotic equations have been an appealing area of research for many mathematicians for more than three decades. They tried to simplify the chaotic behaviors as simple equations in order to analyze and study these behaviors. These equations aim to answer this basic question: What is the necessary and sufficient conditions for the differential equations to become chaotic?

3.1 Continuous Time

Continuous-time chaos generators are systems that can be described by nonlinear differential equations. These equations can be either differential equations (ODEs) or delay-differential equations. The positive entropy in these chaotic dynamical systems leads to continuous instability and the output being unpredictable at all times.

Chaos can be implemented using various equations. In case of all these equations, the most important block is the nonlinear element that has multiple equilibrium points, hence though the system output is unpredictable, it is bounded to "attractive regions". Integrators, sinusoidal waveform generators, delay based systems, and polynomial forms, and piecewise-linear (PWL) functions are among these non linear elements. In Table.1, different equations produce continuous chaos, along with references implementing them and their implementation based on attracting or type and function.

Table 1: Examples of continues time chaotic systems

Name	References	Equation	Scroll Type	Function
Lorenz[a]	[21], [22], [23]	$x' = \lambda\sigma(y - x)$ $y' = \lambda((\beta - z)x - y)$ $z' = \lambda(xy - \rho z)$	Double Scroll Multi Scroll	OTA, Multiplier
Modified Lorenz[b]	[7], [24], [25]	$x' = \sigma(y - x)$ $y' = K(\beta - z) + m$ $z' = (\|x\| - \rho z)$	Double Scroll Multi Scroll	OTA
Lorenz– Stenflo[c]	[26], [27]	$x' = \sigma(y - x) + \lambda\omega$ $y' = (\beta - z)x - \theta y)$ $z' = xy - \epsilon z$ $\omega' = -x - \rho\omega$	Double Scroll	OTA, Product
Chua[d]	[8], [9]	$x' = \sigma(y - x - f(x))$ $y' = x - y + z$ $z' = -\beta y$	Multi Scroll	PWL
Rössler[e]	[28]	$x' = -y - z$ $y' = x + \sigma y$ $z' = \beta + z(x - \rho)$	Double Scroll	OTA, Product
Lü [f]	[29]	$x' = \sigma(y - x)$ $y' = \beta y - xz$ $z' = -\rho z + xy$	Multi Scroll	OTA, Product
SprottD	[30]	$x' = -y$ $y' = x + z$ $z' = 2y^2 + xz - a$	Multi Scroll	OTA, Product

a: λ, σ, β, and ρ are parameters whose choice of value results in a chaotic system.

b: σ, β, and ρ are parameters whose choice of value results in a chaotic system.
K is a bipolar switching constant which is 1 for x $\geq$ 0 and -1 for x $<$ 0.

c: λ, σ, β, ϵ, θ and ρ are parameters whose choice of value results in a chaotic system.

d: σ, and β are parameters whose choice of value results in a chaotic system and
f(x) is a nonlinear element.

e: σ, β, and ρ are parameters whose choice of value results in a chaotic system.

f: σ, β, and ρ are parameters whose choice of value results in a chaotic system.

3.2 Discrete Time

Discrete systems can also be used to generate chaos. A discrete system is expressed as $x_{n+1} = f(C, x_n)$ that shows the next state of the system, x_{n+1} is a function of the present state, x_n, and the control parameter, C. Same as continuous time chaos, nonlinear functions are also essential here to create a chaotic map. Depending on the number of the state variables, chaotic maps are of two kinds: 1) One-dimensional maps, where deterministic equations are the only element responsible for the evolution of a single state variable, with functions such as sine map, tent map, and logistic map and 2) Multi-dimensional chaotic maps where more than one deterministic equation is needed to define the evolution of multiple state variables. In particular, Hénon map is a good example of this second category. These common functions are showcased in Table. 2 as commonly used mathematical chaotic maps. Their simple mathematical expressions can be suitable for applications like FPGA-based image encryption [31]. However, it is reported that the CMOS-based compact implementation of classic chaotic maps including logistic map [32], sine map [33], and tent map [34], becomes highly hardware-hungry. As a solution to this issue, researchers have been exploring to leverage the built-in non-linearity in transistors to design simple, hardware-effective discrete maps with good chaotic properties [35], [36], [37]. Though discrete time chaos has a great potential as a base of a chaotic communication system, we will not be discussing them, as continuous time chaos is a better fit for using chaos ciphering in wearable and other resource limited systems.

Table 2: Examples of some familiar mathematical chaotic maps

Name	Mathematical expression	Parameter bounds
Logistic map [38]	$x_{n+1} = Cx_n(1 - x_n)$	$x_n = [0, 1]$ $C = [0, 4]$
Hénon map [39]	$\begin{cases} x_{n+1} = 1 - C_a x_n^2 + y_n \\ y_{n+1} = C_b x_n \end{cases}$	$x_n = [0, 1.4]$ $C_a = [0, 1.4]$ $C_b = [0, 0.3]$
Sine map [40]	$x_{n+1} = Csin(\pi x_n)$	$x_n = [0, 1]$ $C = [0, 1]$
Tent map [38]	$x_{n+1} = \begin{cases} Cx_n & , x_n < 0.5 \\ C(1 - x_n) & , x_n \geq 0.5 \end{cases}$	$x_n = [0, 1]$ $C = [0, 2]$

4 On-Chip Chaos Implementation and Simulation

Here we discuss continuous-time chaotic equations that can be described by non-linear differential equations. The steps to implement chaos as an integrated

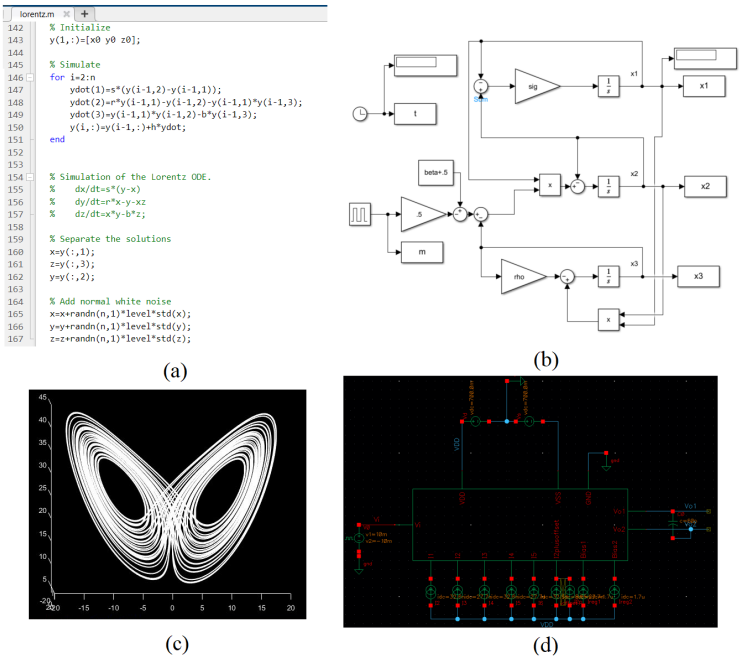


Fig. 2: The flow of implementing chaotic synchronization on chip (a) Implementing the chaotic equations in MATLAB and extracting the initial conditions and parameters needed to achieve chaos (b) Implementing the equations as MATLAB simulink block diagram (c) Simulating the chaotic equations to confirm the bounded chaos needed for ciphering of signals using chaotic synchronization (d) Implementing Simulink blocks as integrated circuits using 65nm MOSFET technology, the block showed in the picture is an n-W powered integrator.

circuit are shown in Fig.2. The first step is implementation of these equations in MATLAB, after simulations and confirmation of chaos, parameters and initial conditions that satisfy chaos are extracted. The equations are then translated to a block diagram implemented in MATLAB Simulink. The transformation from MATLAB Simulink to circuit block diagram requires implementing each block in the diagram to a low power circuit. Simulation results for each type of our equation is seen in Fig.3. The building blocks of these systems as seen from the equations listed in Table. 3 are multipliers, integrators, amplifiers, and PieceWise-Linear (PWL) functions. To extract performance parameters of each chaotic equation as an integrated chip, each block used in the chaotic system is implemented using 65nm CMOS technology and simulated.

The first building block implemented and simulated is a low power integrator. The integrator is based on Rieger et. al [41]. This integrator, consuming power in range of nano-watts, has a very large tunable time constant without using area consuming resistors or a big capacitor. The nominal value of the time con-

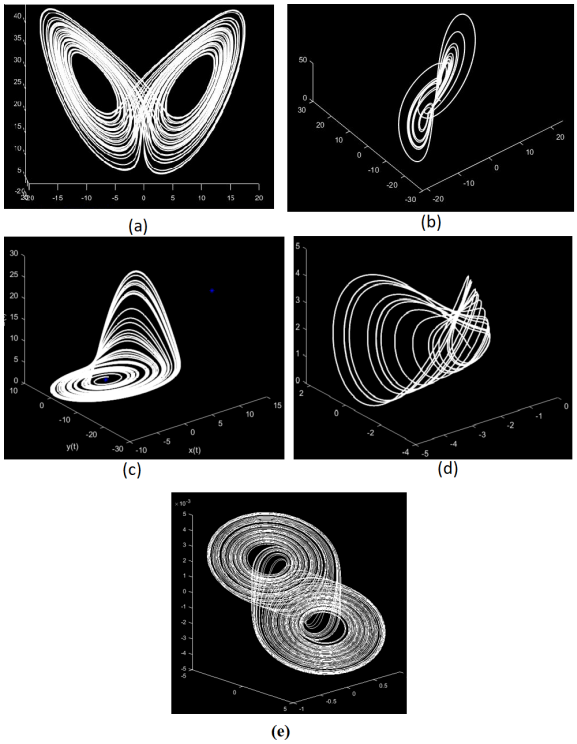


Fig. 3: Simulation results for (a) Lorenz System (b) L*ü*'s system (c) R*ö*ssler system (d) SprottD system and (e) Chua system, The axes in the pictures are x, y and z showing the 3 states of the system.

stant with a 2pF capacitor is 5s which can also be useful for slower signals. The integrator is based on cascading of basic transconductance and transimpedance (gm- 1/gm chain). The gm blocks implemented here are 4 operational transconductance amplifiers (OTAs) with a bias current injected to them from VDD. The 1/g blocks are grounded transistors acting as voltage attenuator resistances. A chain of two gm and two 1/gm blocks, alternating, are used as an attenuator to drive the OTA and the capacitor (OTA-C). The OTA-C section consists of a current source biasing a PMOS OTA, which regulates the NMOS mirror transistors (M5-M10). This integrator is a good fit for resource limited applications that are required to consume low power and low area. Since there will be process, temperature, and voltage variations leading to an output offset on the capacitor nodes, a current source to eliminate the offset can be implemented on the last 1/gm block. The diode connected M11 is used to achieve better-balanced dc conditions. The circuit implementation of the integrator along with the simulation is seen in Fig. 4. The integrator output voltage (Vo) in this picture is simulated by extracting the response versus an square pulse as slow as 4Hz for the input.

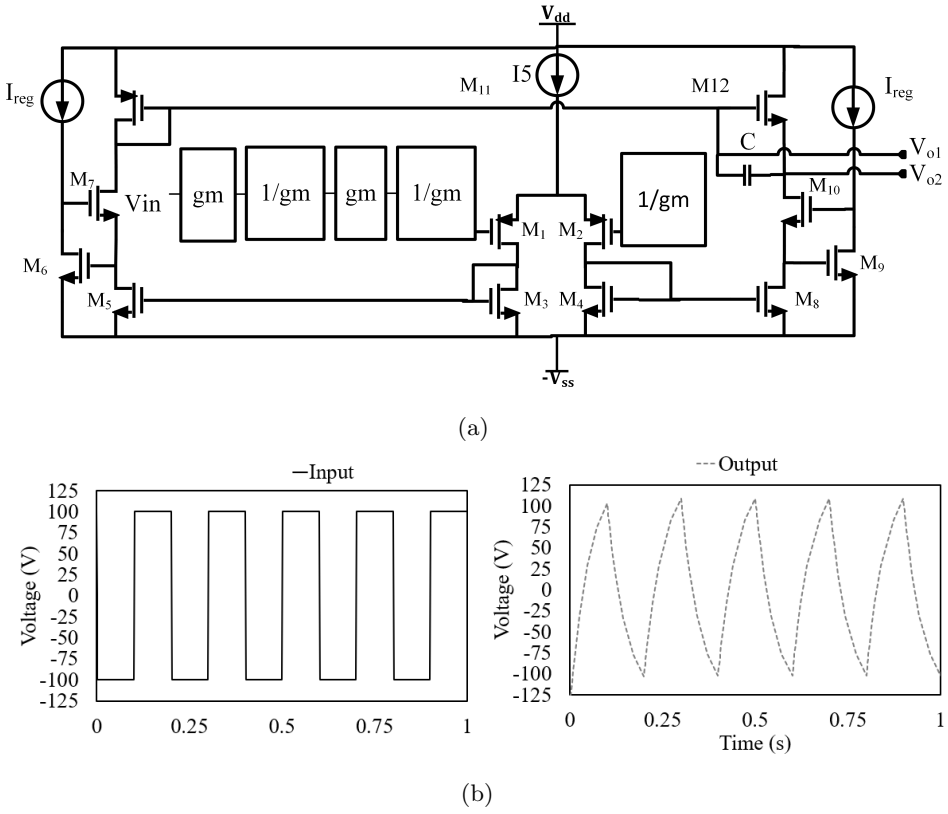


Fig. 4: (a) Circuit diagram of an integrator (b) Simulation results shown for an input pulse.

Gilbert cell typologies are good topologies to be used as multipliers. Gilbert cells are mixers with output signals that are proportional to the product of two input signals. This cell, depicted in Fig.5 uses eight NMOS transistors and two active loads, all are working in saturation region. There are 2 sets of differential input fed to the circuit and the top 4 transistors work as a switch that source the current in the lower part of the circuit. In the lower circuit, the signal is multiplied by the signal fed into M1-M4, and the output obtained is a differential output. To simulate this block, a sinusoidal wave and a square pulse are given as the two sets of input and as shown in the Gilbert cell simulation, the output is the multiplication of two signals. The power consumption of Gilbert cells is two orders of magnitude higher than the integrator and around 200-500μW based on the gain implemented.

Chua's or L$\ddot{u}$'s circuit can use a CMOS implementation of the PWL as the nonlinear element of chaos. This function is constructed of various straight line segments connecting points creating custom wave-forms. PWLs are integral parts

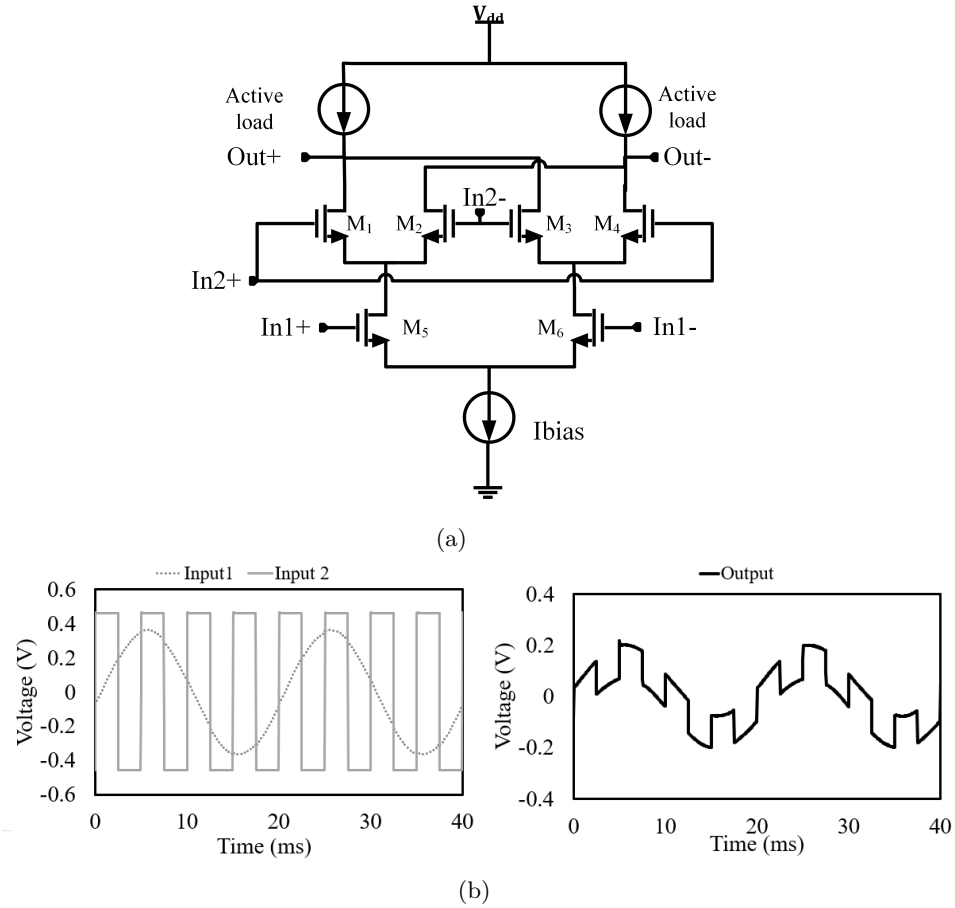
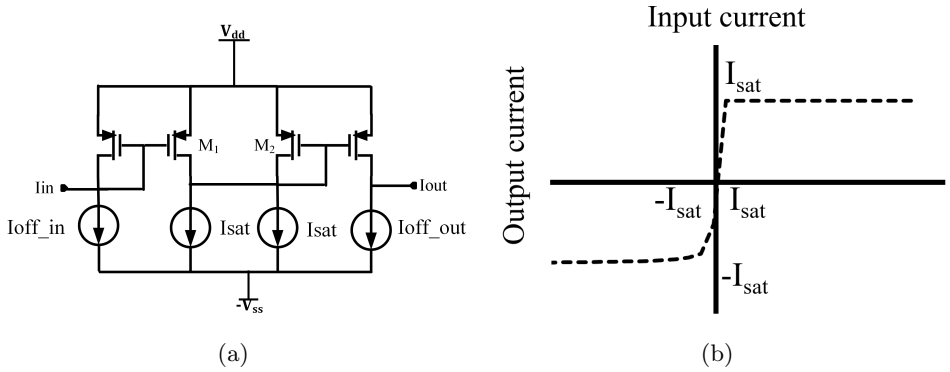


Fig. 5: (a) Circuit diagram of a multiplexer (b) Simulation results shown for multiplication of a pulse and a sinusoidal wave.

in achieving chaos since they are the limiting components when it comes to frequency. A convenient way to implement these line segments is to have a summation of simple functions. The topology shown in Fig.6, based on Carbajal-Gomez et. al [42], can be programmed to have a break-point set by $Ioff_{in}$ and $Ioff_{out}$ and a slope that can be set by $Isat$. This implementation is also advantageous in terms of stability since it has a current mode open loop configuration. The circuit, implemented in current mode shows a better frequency response than voltage mode and is easy to implement with only few transistors.

Apart from these introduced blocks, amplifiers to determine the coefficients of the equations through gain blocks, and passive elements like resistors are common in the chaos generator circuits. It must be noted that not all the chaotic

circuits discussed in literature use these blocks exclusively, but use of these blocks in this paper is to extract performance parameters using common blocks.



(a) (b)

Fig. 6: (a) Circuit diagram of implementation of a piece wise-linear circuit (b) Simulation results showing the current output of a piece wise-linear circuit.

5 Discussion

To implement chaotic synchronization, different blocks introduced in the previous sections are used. In this section, performance of these equations based on their area and power consumption, sensitivity, and robustness is discussed. Chaotic synchronization is very sensitive to initial conditions, in this sense, making a small change in the initial condition of this complex, nonlinear system, produces a huge change in the behavior of the system. With slightly different initial conditions, we start with a slight difference in the results, then beyond a certain time, the system would no longer be predictable. The sensitivity of Rössler, L*ü* and SprottD systems can be seen in Fig.7. Though having a big change in output with a slight change in initial conditions seems desirable as it provides better ciphering, there is indeed a trade off. Systems more sensitive toward initial conditions are also more sensitive toward process, voltage, and temperature (PVT) variations. The small changes formed in the fabrication process of the integrated chip, makes the more sensitive implementations almost impossible to synchronize as two identical systems implemented on chip will still be slightly different and posing an extremely different output. To eliminate these PVT variations, considerations for tuning circuit and post-fabrication processing are needed to contribute towards power and area consumption. L*ü*'s system is seen to be more sensitive toward initial conditions as seen in the simulation.

Lorenz was the main equation to implement chaos for decades. This set of equations, however, needs two multipliers which are power and area consuming to be implemented on chip and can contribute to a significant DC offset.

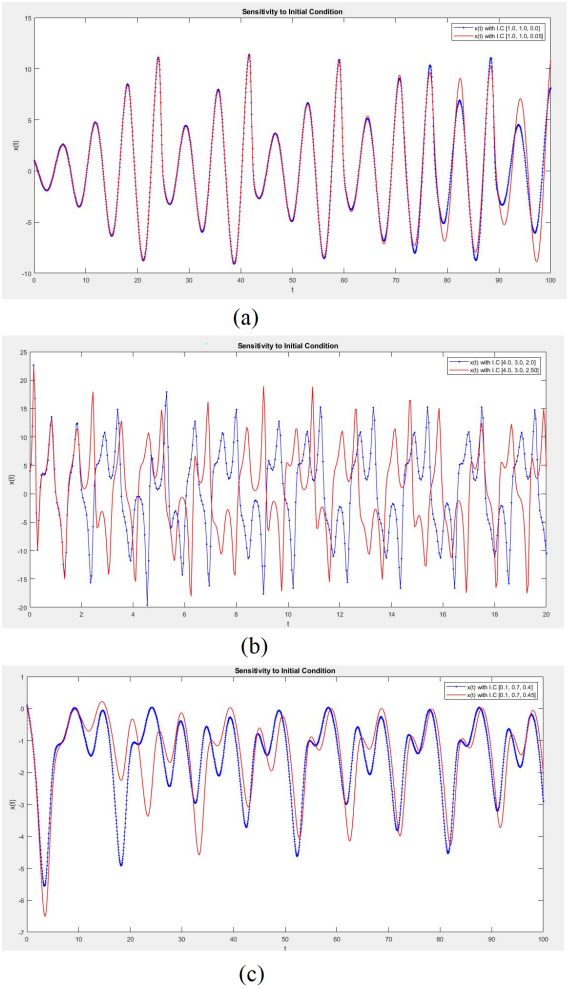


Fig. 7: Sensitivity of (a) R*ö*ssler, (b) L*ü* and (c) SprottD systems to initial condition.

Modified Lorenz eliminates the multipliers which will reduce the power and area consumption. Chua's design also eliminates the multipliers, replacing them with a PWL circuit. The power consumption is hence proportional to the number of multipliers in the circuit as the power consummation of the Gilbert cells is 2 order of magnitude higher than that of amplifiers, PWLs and integrators.

These data are utilized to come up with a Figure of Merit (FOM), that is smaller for a better design as shown in equation 1. The design performance improves as we use less power and area consumption (proportional to the number of multipliers), fewer blocks (using only primitive blocks, such as amplifiers,

integrators, multipliers, etc. for design purposes), and reduced noise sensitivity (the FOM is designed to be proportional with these parameters). The design will improve if we are robust to PVT variation (or if the design is easily tunable after fabrication). For the design load of Lorenz, L*ü*, R*ö*ssler, and SprottD, they are estimated as 1 since they can be implemented using the integrators and multipliers and the design load of Modified Lorenz and Chua's are estimated as 2 because of need to design specialty blocks. Lorenz– Stenflo requires 4 equations to be implemented and it's design load is estimated as 2. Chua's robustness to PVT is estimated as 2 because of ease of tuning the PWL currents after the process. The detailed comparison of these systems in terms of performance is shown in Table. 3.

$$FOM = \frac{Power\&AreaConsumption \times \#ofblocks \times DesignLoad \times Noise}{Robustness} \tag{1}$$

Table 3: Comparison table with state-of-the-art chaotic communication

Name	Based on	# of main blocks	# of multipliers	FOM
Lorenz	[23]	5	2	15
Modified Lorenz	[25]	4	0	8
Lorenz– Stenflo [a]	[26]	6	2	18
Chua	[8], [9]	4	0	4
R*ö*ssler	[28]	4	1	8
L*ü* [b]	[29]	5	0-2	10
SprottD [c]	[30]	5	2	7.5

a: This design has more output states leading to a more robust ciphering.

b: Various alternatives exist to implement L*ü*'s system with one or no multiplier.

c: This design has no equilibria leading to a more robust ciphering.

6 Conclusion

In this paper Multiple Lorenz, Chua's, L*ü*'s, and R*ö*ssler, and sprottD systems are implemented and simulated. The system implementations are considered as alternatives to the software chaos and architectures which can further reduce the power and the area overheads. Reducing power and area of these systems pave the way for the effective utilization of security at chip level for resource limited applications such as wearables, implantable devices, and Internet-of-Things

(IoT) devices where security has been overlooked even in regularly used devices. Various performance metrics including power consumption, area, design load, noise, and and robustness are extracted and compared for each system to demonstrate a figure of merit. The figure of merit shows the importance of reducing the use of multipliers by introducing chaotic equations that avoid using multipliers.

7 Acknowledgement

The simulations on this paper are done using Cadence virtuoso, supplied by Cadence university program to California State University Long Beach. This work is supported by the National Science Foundation under award No. 2131156.

References

1. K. M. Cuomo, A. V. Oppenheim, and S. H. Strogatz, "Synchronization of lorenz-based chaotic circuits with applications to communications," *IEEE Transactions on Circuits and Systems II: Analog and digital signal processing*, vol. 40, no. 10, pp. 626–633, 1993.
2. E. Ott, C. Grebogi, and J. A. Yorke, "Controlling chaotic dynamical systems," in *Chaos: Soviet-American Perspective on Nonlinear Science*. American Institute of Physics, 1990, pp. 153–172.
3. P. Benioff, "The computer as a physical system: A microscopic quantum mechanical hamiltonian model of computers as represented by turing machines," *Journal of Statistical Physics*, vol. 22, no. 5, pp. 563–591, 1980.
4. F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. Brandao, D. A. Buell *et al.*, "Quantum supremacy using a programmable superconducting processor," *Nature*, vol. 574, no. 7779, pp. 505–510, 2019.
5. G. Kalai, "The argument against quantum computers," *Quantum, Probability, Logic*, pp. 399–422, 2020.
6. F. L. Smith III, "Quantum technology hype and national security," *Security Dialogue*, p. 0967010620904922, 2020.
7. A. S. Elwakil and M. P. Kennedy, "Construction of classes of circuit-independent chaotic oscillators using passive-only nonlinear devices," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 48, no. 3, pp. 289–307, 2001.
8. M. Delgado-Restituto and A. Rodriguez-Vazquez, "A cmos analog chaotic oscillator for signal encryption," in *ESSCIRC'93: Nineteenth European Solid-State Circuits Conference*, vol. 1. IEEE, 1993, pp. 110–113.
9. M. Delgado-Restituto, A. Rodriguez-Vazquez, and M. Linan, "A modulator/demodulator cmos ic for chaotic encryption of audio," in *ESSCIRC'95: Twenty-first European Solid-State Circuits Conference*. IEEE, 1995, pp. 170–173.
10. A. S. Elwakil, K. N. Salama, and M. P. Kennedy, "An equation for generating chaos and its monolithic implementation," *International Journal of Bifurcation and Chaos*, vol. 12, no. 12, pp. 2885–2895, 2002.

11. R. Trejo-Guerra, E. Tlelo-Cuautle, C. Cruz-Hernández, and C. SÁNCHEZ-LÓPEZ, "Chaotic communication system using chua's oscillators realized with ccii+ s," *International Journal of Bifurcation and Chaos*, vol. 19, no. 12, pp. 4217–4226, 2009.
12. C. Sánchez-López, R. Trejo-Guerra, J. Munoz-Pacheco, and E. Tlelo-Cuautle, "N-scroll chaotic attractors from saturated function series employing ccii+ s," *Nonlinear Dynamics*, vol. 61, no. 1-2, pp. 331–341, 2010.
13. R. Trejo-Guerra, E. Tlelo-Cuautle, J. Jimenez-Fuentes, C. Sánchez-López, J. Munoz-Pacheco, G. Espinosa-Flores-Verdad, and J. Rocha-Pérez, "Integrated circuit generating 3-and 5-scroll attractors," *Communications in Nonlinear Science and Numerical Simulation*, vol. 17, no. 11, pp. 4328–4335, 2012.
14. R. Trejo-Guerra, E. Tlelo-Cuautle, M. Jiménez-Fuentes, J. M. Muñoz-Pacheco, and C. Sánchez-López, "Multiscroll floating gate–based integrated chaotic oscillator," *International Journal of Circuit Theory and Applications*, vol. 41, no. 8, pp. 831–843, 2013.
15. O. A. Gonzalez, G. Han, J. P. De Gyvez *et al.*, "Cmos cryptosystem using a lorenz chaotic oscillator," in *ISCAS'99. Proceedings of the 1999 IEEE International Symposium on Circuits and Systems VLSI (Cat. No. 99CH36349)*, vol. 5. IEEE, 1999, pp. 442–445.
16. Y.-L. Wu, C.-H. Yang, Y.-S. Li, and C.-H. Wu, "Nonlinear dynamic analysis and chip implementation of a new chaotic oscillator," in *2015 IEEE 12th International Conference on Networking, Sensing and Control*. IEEE, 2015, pp. 554–559.
17. L. Xiong, Y.-J. Lu, Y.-F. Zhang, X.-G. Zhang, and P. Gupta, "Design and hardware implementation of a new chaotic secure communication technique," *PloS one*, vol. 11, no. 8, p. e0158348, 2016.
18. C. Liang, Q. Zhang, J. Ma, and K. Li, "Research on neural network chaotic encryption algorithm in wireless network security communication," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, p. 151, 2019.
19. L. Zhang, "Artificial neural network model design and topology analysis for fpga implementation of lorenz chaotic generator," in *2017 IEEE 30th Canadian conference on electrical and computer engineering (CCECE)*. IEEE, 2017, pp. 1–4.
20. M. Tuna, M. Alçın, İ. Koyuncu, C. B. Fidan, and İ. Pehlivan, "High speed fpga-based chaotic oscillator design," *Microprocessors and Microsystems*, vol. 66, pp. 72–80, 2019.
21. O. A. Gonzales, G. Han, J. P. De Gyvez, and E. Sánchez-Sinencio, "Lorenz-based chaotic cryptosystem: a monolithic implementation," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 47, no. 8, pp. 1243–1247, 2000.
22. S. Yu, J. Lü, W. K. Tang, and G. Chen, "A general multiscroll lorenz system family and its realization via digital signal processors," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 16, no. 3, p. 033126, 2006.
23. D. Brown, A. Hedayatipour, M. B. Majumder, G. S. Rose, N. McFarlane, and D. Materassi, "Practical realisation of a return map immune lorenz-based chaotic stream cipher in circuitry," *IET Computers & Digital Techniques*, vol. 12, no. 6, pp. 297–305, 2018.
24. S. Özoğuz, A. S. Elwakil, and M. P. Kennedy, "Experimental verification of the butterfly attractor in a modified lorenz system," *International Journal of Bifurcation and Chaos*, vol. 12, no. 07, pp. 1627–1632, 2002.
25. A. Radwan, A. Soliman, and A. El-Sedeek, "Mos realization of the modified lorenz chaotic system," *Chaos, Solitons & Fractals*, vol. 21, no. 3, pp. 553–561, 2004.

26. Y.-L. Wu, C.-H. Yang, and C.-H. Wu, "Design of initial value control for modified lorenz-stenflo system," *Mathematical Problems in Engineering*, vol. 2017, 2017.
27. F. Zhang, R. Chen, and X. Chen, "Analysis of a generalized lorenz–stenflo equation," *Complexity*, vol. 2017, 2017.
28. D. N. Butusov, T. I. Karimov, I. A. Lizunova, A. A. Soldatkina, and E. N. Popova, "Synchronization of analog and discrete rössler chaotic systems," in *2017 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*. IEEE, 2017, pp. 265–270.
29. T.-L. Liao, H.-C. Chen, C.-Y. Peng, and Y.-Y. Hou, "Chaos-based secure communications in biomedical information application," *Electronics*, vol. 10, no. 3, p. 359, 2021.
30. Z. Wei, "Dynamical behaviors of a chaotic system with no equilibria," *Physics Letters A*, vol. 376, no. 2, pp. 102–108, 2011.
31. B. Baruah and M. Saikia, "An fpga implementation of chaos based image encryption and its performance analysis," *IJCSN-International Journal of Computer Science and Network*, vol. 5, no. 5, 2016.
32. J. Lopez-Hernandez, A. Diaz-Mendez, R. Vazquez-Medina, and R. Alejos-Palomares, "Analog current-mode implementation of a logistic-map based chaos generator," in *2009 52nd IEEE International Midwest Symposium on Circuits and Systems*. IEEE, 2009, pp. 812–814.
33. A. Farfan-Pelaez, E. Del-Moral-Hernández, J. Navarro, and W. Van Noije, "A cmos implementation of the sine-circle map," in *48th Midwest Symposium on Circuits and Systems, 2005.* IEEE, 2005, pp. 1502–1505.
34. S. Callegari, G. Setti, and P. J. Langlois, "A cmos tailed tent map for the generation of uniformly distributed chaotic sequences," in *1997 IEEE International Symposium on Circuits and Systems (ISCAS)*, vol. 2. IEEE, 1997, pp. 781–784.
35. P. Dudek and V. Juncu, "Compact discrete-time chaos generator circuit," *Electronics Letters*, vol. 39, no. 20, pp. 1431–1432, 2003.
36. V. Juncu, M. Rafiei-Naeini, and P. Dudek, "Integrated circuit implementation of a compact discrete-time chaos generator," *Analog Integrated Circuits and Signal Processing*, vol. 46, no. 3, pp. 275–280, 2006.
37. B. Kia, K. Mobley, and W. L. Ditto, "An integrated circuit design for a dynamics-based reconfigurable logic block," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 64, no. 6, pp. 715–719, 2017.
38. Y. Zhou, Z. Hua, C.-M. Pun, and C. P. Chen, "Cascade chaotic system with applications," *IEEE Transactions on Cybernetics*, vol. 45, no. 9, pp. 2001–2012, 2014.
39. W. F. H. Al-Shameri, "Dynamical properties of the hénon mapping," *Int. Journal of Math. Analysis*, vol. 6, no. 49, pp. 2419–2430, 2012.
40. Z. Hua and Y. Zhou, "Dynamic parameter-control chaotic system," *IEEE Transactions on Cybernetics*, vol. 46, no. 12, pp. 3330–3341, 2015.
41. R. Rieger, A. Demosthenous, and J. Taylor, "A 230-nw 10-s time constant cmos integrator for an adaptive nerve signal amplifier," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 11, pp. 1968–1975, 2004.
42. V. H. Carbajal-Gomez, E. Tlelo-Cuautle, J. M. Muñoz-Pacheco, L. G. de la Fraga, C. Sanchez-Lopez, and F. V. Fernandez-Fernandez, "Optimization and cmos design of chaotic oscillators robust to pvt variations," *Integration*, vol. 65, pp. 32–42, 2019.