Design of a Reconfigurable Chaos Gate with Enhanced Functionality Space in 65nm CMOS

Aysha S. Shanta, Md. Badruddoja Majumder, Md Sakib Hasan, Mesbah Uddin and Garrett S. Rose

IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS), Windsor, ON, Canada, August 2018.

The online home for this paper may be found at: http://web.eecs.utk.edu/ grose4/

Citation Information (BibTex):

```
@INPROCEEDINGS{MWSCAS: shanta2018design,
  author=" A. S. Shanta and M. B. Majumder and M. S. Hasan
       and M. Uddin and G. S. Rose",
  title="Design of a Reconfigurable Chaos Gate with Enhanced
      Functionality Space in 65nm CMOS",
  booktitle="{IEEE} International Symposium on Circuits
      and Systems {MWSCAS}",
  month="August",
  year="2018",
  pages="1016-1019",
  address="Windsor, ON, Canada"
}
```

Design of a Reconfigurable Chaos Gate with Enhanced Functionality Space in 65nm CMOS

Aysha S. Shanta, Md. Badruddoja Majumder, Md Sakib Hasan, Mesbah Uddin, and Garrett S. Rose
Department of Electrical Engineering and Computer Science
University of Tennessee, Knoxville
Knoxville, Tennessee 37996 USA
Email: {ashanta1, mmajumde, mhasan4, muddin6, garose}@vols.utk.edu

***Abstract*—In this paper, a three transistor circuit has been implemented in a standard 65 nm CMOS technology. The circuit has been used as a map to generate discrete-time chaotic signals. The map circuit has been combined with another map circuit in order to build a chaotic generator. The circuit is compact since it uses only twelve transistors in total to generate different Boolean functions as part of a chaotic computer. The total power consumption of the chaotic generator is only 18.4 μW and the area is 0.556 μm^2. The circuit can be used as a logic gate and has been designed to generate various functions using multiple configurations. The number of functionalities of the chaos gate has been increased by altering the bias and threshold voltages.**

***Index Terms*—Chaos computing, logic obfuscation, CMOS, VLSI, hardware security**

I. Introduction

Chaos is found in many fields of study such as, medicine, biology, chemistry, physics and engineering. Chaotic behavior has been found in brains, hearts, lasers and electronic circuits [1]. Researchers have studied integrated circuit implementations of non-linear deterministic systems which are capable of generating chaotic signals for more than two decades. Chaos based computation uses chaotic circuits for implementing logic functions [2]. The conventional Boolean circuit consists of a set of switching transistors which operates based on the incoming input. In order to implement different Boolean functions, different logic gates are required. Chaotic circuits, on the other hand, are dynamical systems which map current state to future states. A single dynamical system can be used to implement different functions [3], meaning the area will be significantly less. Each chaos based system can be programmed and reconfigured to adapt to different conditions in order to implement different functions. This adaptability provides the computing system with variability and flexibility. Moreover, the operational cost of chaos computing is not expensive since the system is intrinsically a chaotic physical system.

The inputs to the circuit are usually provided from a digital-to-analog (DAC) converter. The output of the chaotic circuit is converted to a digital value using a comparator at the end. The analog value is compared to a threshold voltage. Different schemes have been developed by researchers to implement different logical functions by changing different parameters of the circuit topologies. Biasing the initial input to the chaotic circuit in a different manner can be used as a technique

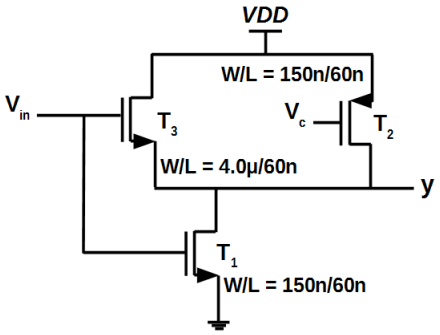

Fig. 1: A three transistor map circuit [4].

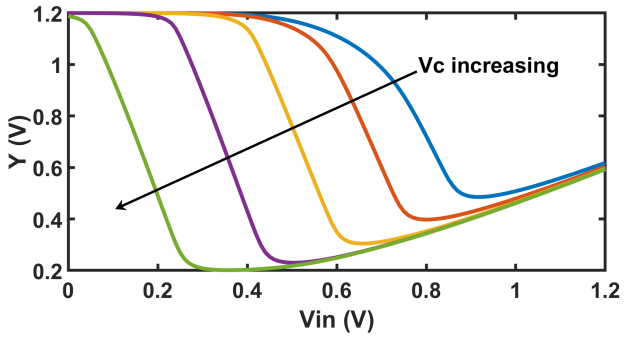

Fig. 2: DC characteristics of the map circuit.

to implement different functions [3]. Researchers have used Chua's circuit to implement all logic functions (AND, OR, NAND, NOR and XOR) where the chaos gate can switch between different logic functions by changing the threshold parameter [5]. Murali *et al.* have implemented a fundamental NOR gate using Chua's chaos circuit where the basic logic operations can be constructed [6]. Ditto *et al.* developed a gate that uses a chaotic oscillator to determine the functionality of the logic gate. The logic functionality can be altered by changing the control voltage, weighting factor or threshold [7]. Sinha *et al.* used chaotic maps to design systems that can be used as logic gates and can perform arithmetic operations such as addition and multiplication. The system is also capable of calculating the least common multiplier (LCM) of a sequence of numbers [8].

A standard CMOS gate requires less hardware compared to a given chaotic logic implementation. One way to overcome

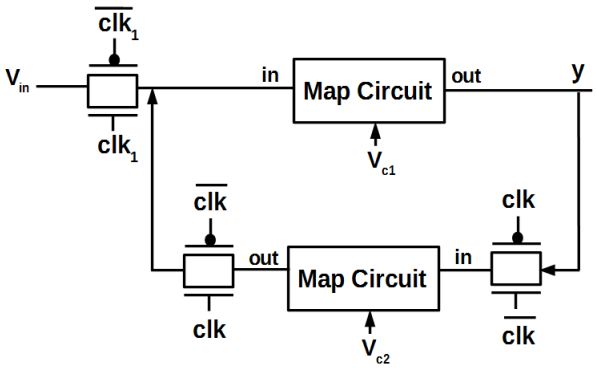

Fig. 3: Chaos generator with two map circuits, modeled at the transistor level [4].

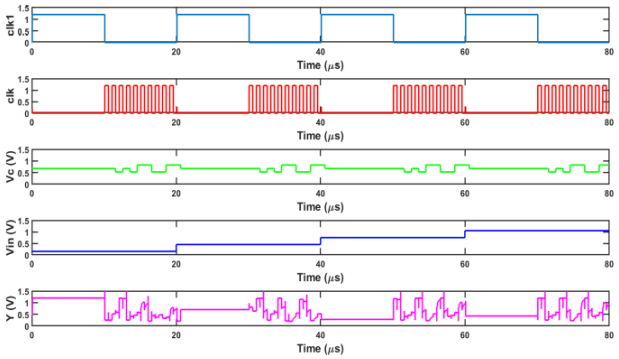

Fig. 4: Transient response of the chaos generator.

the area overhead of chaotic gates is to increase the functionality of a single gate. One of the main contributions of this paper is to increase the design space (number of functionalities) by altering the threshold voltage at the output along with varying the bias voltage in every iteration. The total number of implementable functions exponentially increases compared to a linear increase in previous work [9]. The number of individual functions (AND, NOR, XOR etc.) also increases linearly with the increase in functionality space. Moreover, this work focuses on decreasing the area and power of an already established three map circuit [4]. The area of the two map circuits is only 0.556 μm^2 and the total power consumption is 18.4 μW.

The remainder of the paper is organized as follows. Section II describes the circuit operation and dc characterization of the map circuit. The usage of chaos generator as a logic gate and different techniques to increase the design space are discussed in Section III. Section IV describes the possible applications of the chaos gate. Finally, concluding remarks are given in Section V.

II. Circuit Design

In this paper, a three transistor CMOS circuit is used to implement a nonlinear map in order to generate chaotic signals. The map circuit is shown in Fig. 1. When the threshold voltage of T_1 is greater than the input voltage, V_{in}, the output voltage, V_{out} is at V_{DD}. The transistor T_3 is in cutoff region. Transistors T_1 and T_2 act like an inverter forcing V_{out} to decrease as V_{in} increases. At a certain point, the voltage V_{in} becomes larger than V_{out} making the gate-source voltage of transistor T_3 larger than its threshold voltage. Transistor T_3 starts conducting and starts behaving like a source follower since an increase in input voltage, V_{in} causes V_{out} to increase.

Chaotic circuits are generally designed to imitate one of the popular chaos maps, e.g. a logistic map or a tent map. The chaos map of the circuit has been designed carefully to achieve higher functionality. If the length of the transistors in the map circuit are increased, the part of the chaos map with the negative slope becomes steeper, resulting in less implementable Boolean functions. Another design consideration was to make the width of the transistor T_3 much higher than T_1 in order to make the average positive slope close to unity. The characteristic of the map circuit can be altered by changing the bias voltage, V_c as shown in Fig. 2.

In a chaos generator, two sample and hold circuits are required in addition to a map circuit to generate discrete-time chaotic signals. Two capacitors C_1 and C_2 and an analog buffer are required to hold the output of the map circuit. In this paper, instead of using a buffer and capacitors, two map circuits are used as displayed in Fig. 3. One clock, clk_1 is used to apply the input and two complementary clocks, clk and $\overline{clk}$ are generated to sample the outputs of the map circuit. When clk_1 is high, a new input is applied to the chaos generator. When clk is high, the first map circuit generates an output signal which corresponds to $x_{n+1} = f(x_n)$. When $\overline{clk}$ is high, the second map circuit in the feedback generates an output signal which corresponds to $x_{n+2} = f(x_{n+1})$. Due to component mismatch and process variation, the physical implementation of the two circuits will never be identical. Using two map circuits in the chaos generator is efficient since we can use two bias voltages from the two map circuits to configure Boolean functions.

TABLE I: Comparison of proposed circuit with prior work

	Prior work [10]	**This work**
Technology	0.6 μm	65 nm
Supply Voltage	5 V	1.2 V
Area	32 × 19 μm	0.556 μm^2
Power	7.85 mW (experimental)	8.14 μW (simulated)

The circuits have been simulated in a standard 65 nm CMOS technology. The power supply voltage used for simulation is 1.2 V. The chaotic generator in this paper has much smaller area compared to the conventional chaotic generators which have been implemented in a 0.6 μm and 5 V process. The transistors in the map circuit are sized as: T_1 = 150 nm/60 nm, T_2 = 150 nm/60 nm and T_3 = 4 μm/60 nm. The three switches in the chaos generator are implemented using pass-gate transistors where the size of the p-MOS transistor is 8 μm/100 nm and that of the n-MOS transistor is 4 μm/100 nm. The capacitors in the sample-and-hold mechanism are implemented using the internal input capacitances of the map circuits so the entire circuit requires only twelve transistors.

The transistors can be used as switches instead of using sample-and-hold circuits which consume huge amounts of area and power. The total power consumption in 1.2 V process is 18.4 μW (simulated) which is much less than 7.85 mW (experimental) in a 5 V process [10]. The power consumption is higher in the 5 V process because huge amount of current flows from V_{DD} to ground as the input voltage, V_{in} increases. The area of the two map circuits implemented in 65 nm CMOS technology requires only 0.556 μm^2 whereas the area in 0.6 μm process is 32×19 μm. This area does not include the area of the pass transistors. Table I shows the comparison between previous work and this work.

III. Gate Implementation And Design Space Enhancement

It is possible to achieve any infinite number of functionalities from a single nonlinear circuit. The term functionality implies not only particular functions but also various ways of realizing them. Although, all the functions generated from the circuit might not be usable due to instability of the inputs or presence of noise. Kia *et al.* used the same three transistor map circuit to implement a large number of functions. In [9], the authors mapped the digital data and control input bits to the initial condition using a DAC and extracted the digital data at the output using a comparator.

The output from the map circuit is compared to a threshold voltage in order to determine the digital value. The equation used for the threshold partitioning is:

$$y = \begin{cases} 1, & \text{if } f(x) \geq V_{th} \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

In [9], the functionality space of the map circuit has been increased by implementing 16 control bits along with the 4 input bits. Different functions can be implemented with the number of loop iterations since the output at each iteration can be different from the previous one. Let b be the number of bias voltages considered in the chaos generator, the control input bits be denoted by c and number of iterations by n. The equation for representing the functionality space is as follows:

$$f(n) = b \times 2^c \times n. \tag{2}$$

From equation 2, it can be inferred that the functionality space of the chaos gate proposed in [9] increases linearly with the number of iterations. For a particular number of iterations, the space is limited by the number of bias voltage levels considered and number of control bits used in the DAC. According to the bifurcation diagram of the chaos generator circuit demonstrated in [4], not all bias voltages lead to a chaotic behavior. Rather, only a short range of bias voltages generate chaotic behavior in the three transistor non-linear circuit. This is why the number of possible bias voltages, b which can be used in equation 2 is limited.

The size (bits) of the DAC for the chaos gate implementation depends on the total number of input and control bits. However, the maximum achievable resolution of a DAC is limited by the signal to noise ratio. Therefore, the number of control bits, c used in equation 2 is limited too. For an optimal choice of b and c, the functionality space of the chaos gate implemented in [9] can be increased by letting the chaos generator circuit run for more iterations. Again, this increase is only linear and therefore, requires a very large number of iterations for larger space.

In this paper, a few techniques have been introduced to increase the functionality space of the chaos gate significantly. In order to increase the functionality space exponentially with the number of iterations, the bias voltage is randomly varied cycle to cycle, instead of applying a fixed bias voltage on every iteration. In this implementation, for simplicity we consider the bias voltages of both map circuits to be the same. However, considering both bias voltages independent of one another would increase the functionality space even more. Furthermore, multiple threshold levels are considered for the comparator circuit at the output instead of using a fixed one. We have chosen four threshold voltages and three bias voltages in this implementation. Bias voltages have been determined by partitioning 1.2 V into eight equally spaced intervals. Three bias voltages out of eight, 0.525 V, 0.675 V and 0.825 V have displayed the most variation in functionality from iteration to iteration. The threshold voltage levels are chosen to be 0.15 V, 0.45 V, 0.75 V, and 1.05 V. The four input voltages chosen for representing (0,0), (0,1), (1,0) and (1,1) are 0.15 V, 0.45 V, 0.75 V, and 1.05 V respectively.

The transient simulation of the chaos generator can be seen in Fig.4. When clk_1 is high, a new V_{in} is applied to the input of the chaos generator. When clk_1 is low, clk goes through ten iterations. V_c changes randomly on every iteration of clk. The output, Y demonstrates a chaotic behavior.

The equation for representing the functionality space in the proposed method is given by the following equation:

$$f(n) = b^n \times n \times N_{v_{th}}, \tag{3}$$

where b is the number of bias voltage levels, n is the number of iteration and $N_{v_{th}}$ is the number of threshold voltage levels considered in this implementation.

Fig. 5 displays the increase in functionality space with the number of iterations for the chaos gate implemented in a previous work [9] and in this work. As we already explained, the increase in functionality space in the previous work is almost linear as shown by the red curve. In contrast, for the design method proposed in this paper, the increase in functionality is exponential as represented by the blue curve. Over the first 6 iterations, the functionality space is higher for the previous work's design. After 6 iterations, the functionality increases drastically for the design proposed in this paper and at the 8^{th} iteration it is almost $3\times$. The functionality space will display a significant increase if a few more iterations are considered.

Fig. 5 shows the total functionality space over runtime of the chaos generator. However, it is also necessary to find out whether the number of individual functions achieved from

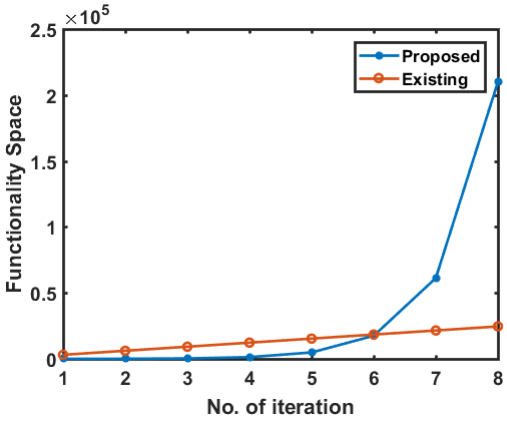

Fig. 5: Comparison between the functionality space of the proposed design with an existing one with respect to number of iterations considered.

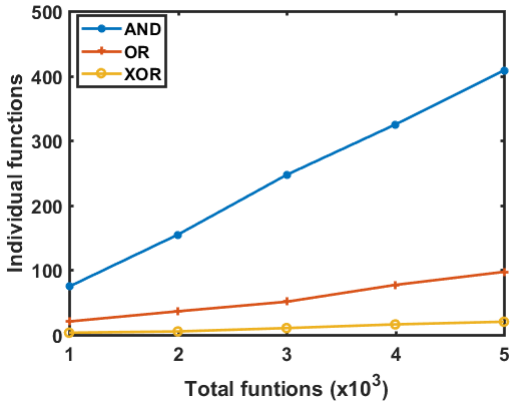

Fig. 6: Demonstration of number of individual functions increasing linearly with the increase of total functionality space.

the chaos gate increases proportionately with the increase in total functionality space. A total of 5000 functions have been generated in order to evaluate whether the individual functions are increasing in number with the increase in design space. As can be seen in Fig. 6, the number of individual functions, AND, OR and XOR are also increasing with functionality space. This observation ensures that increase of individual function space is also exponential with the number of iterations of the chaos generator.

IV. Application

Security is an important design constraint in modern computing systems. A side channel attack represents a security vulnerability where an attacker studies the power profiles of computation in a processor to gain information about the processed data [11]. Researchers have studied possible obfuscation techniques in order to mitigate power analysis based side channel attack. Chaos based computing can help in side channel attack mitigation since it has an enhanced space of functionality and chaotic computation generates a random power profile that is difficult to use for side-channel attacks [12]. A chaos based arithmetic logic unit has been designed in [13]. It states that any two operations are unique in terms of power profiles, control parameters, etc. and this very quality of chaotic circuits can be utilized for code obfuscation and side channel attack mitigation.

V. Conclusion

In this paper, a simple nonlinear chaotic map circuit has been implemented in 65 nm CMOS technology in order to generate different Boolean functions. A total of twelve transistors are required in order to realize the chaos generator. The area and power are significantly low compared to the previous works. The purpose of this paper is to increase the functionality space by changing the bias and threshold voltages on every iteration. It has been demonstrated that the number of functionalities increase exponentially with number of iterations. The individual functions also increase with the increase in design space.

Acknowledgment

This work is based upon work supported by the Air Force Office of Scientific Research under award number FA9550-16-1-0301.

References

[1] T. Munakata, S. Sinha, and W. L. Ditto, "Chaos computing: implementation of fundamental logical gates by chaotic elements," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 49, no. 11, pp. 1629–1633, 2002.

[2] W. L. Ditto, K. Murali, and S. Sinha, "Chaos computing: ideas and implementations," *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 366, no. 1865, pp. 653–664, 2008.

[3] B. Kia, J. F. Lindner, and W. L. Ditto, "Nonlinear dynamics based digital logic and circuits," *Frontiers in Computational Neuroscience*, vol. 9, p. 49, 2015.

[4] P. Dudek and V. Juncu, "Compact discrete-time chaos generator circuit," *Electronics Letters*, vol. 39, no. 20, pp. 1431–1432, 2003.

[5] M. R. Rizk, A.-M. A. Nasser, E.-S. A. El-Badawy, and E. Abou-Bakr, "A new approach for obtaining all logic gates using chua's circuit: Advantages and disadvantages," in *Computer and Communication Engineering (ICCCE), 2012 International Conference on*. IEEE, 2012, pp. 730–733.

[6] K. Murali, S. Sinha, and W. L. Ditto, "Implementation of nor gate by a chaotic chua's circuit," *International Journal of Bifurcation and Chaos*, vol. 13, no. 09, pp. 2669–2672, 2003.

[7] W. L. Ditto, A. Miliotis, K. Murali, S. Sinha, and M. L. Spano, "Chaogates: Morphing logic gates that exploit dynamical patterns," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 20, no. 3, p. 037107, 2010.

[8] S. Sinha and W. L. Ditto, "Computing with distributed chaos," *Physical Review E*, vol. 60, no. 1, p. 363, 1999.

[9] B. Kia, J. F. Lindner, and W. L. Ditto, "A simple nonlinear circuit contains an infinite number of functions," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 63, no. 10, pp. 944–948, 2016.

[10] 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.

[11] P. Kocher, J. Jaffe, B. Jun, and P. Rohatgi, "Introduction to differential power analysis," *Journal of Cryptographic Engineering*, vol. 1, no. 1, pp. 5–27, 2011.

[12] J. Bohl, L.-K. Yan, and G. S. Rose, "A two-dimensional chaotic logic gate for improved computer security," in *Circuits and Systems (MWSCAS), 2015 IEEE 58th International Midwest Symposium on*. IEEE, 2015, pp. 1–4.

[13] G. S. Rose, "A chaos-based arithmetic logic unit and implications for obfuscation," in *VLSI (ISVLSI), 2014 IEEE Computer Society Annual Symposium on*. IEEE, 2014, pp. 54–58.