A Two-Dimensional Chaotic Logic Gate for Improved Computer Security

James Bohl, Lok-Kwong Yan, and Garrett S. Rose

IEEE International Midwest Symposium on Circuits and Systems (MWSCAS), Fort Collins, CO, August 2015.

Citation information (BibTex):

```
@INPROCEEDINGS{Bohl:2015,
  author=”James Bohl and Lok-Kwong Yan and Garrett S.
      Rose”,
  title=”A Two-Dimensional Chaotic Logic Gate for
      Improved Computer Security”,
  booktitle=”Proceedings of {IEEE} International
      Midwest Symposium on Circuits and Systems
      ({MWSCAS})”,
  month=”August”,
  year=”2015”,
  location=”Fort Collins, CO”
}
```

A Two-Dimensional Chaotic Logic Gate for Improved Computer Security

James Bohl[*], Lok-Kwong Yan[†], and Garrett S. Rose[‡]
Email: jbohl@androcs.com, lok.yan@us.af.mil, garose@utk.edu
[*]Andro Computational Solutions, Rome, NY, USA
[†]Information Directorate, Air Force Research Laboratory, Rome, NY, USA
[‡]Department of Electrical Engineering and Computer Science, The University of Tennessee, Knoxville, TN, USA

***Abstract*—In recent years the concept of chaos-based computing has emerged as a way to harness the rich state space of chaotic systems for robust computation. Potential advantages of such chaotic computational elements include improved security in the form of logic obfuscation and power analysis mitigation. For example, the chaotic nature of computation leads to a chaotic power profile that is difficult to use for side-channel attacks. In this paper, we explore the construction of chaotic logic gates based on Chua's circuit. We propose a two-dimensional chaotic logic gate that utilizes the two state variables of Chua's circuit to implement all possible two-input logic functions. Further, the likelihood of any logic function is shown to approach 1/16 as the evolution time of the gate is increased. Equally likely functions are beneficial from a security perspective in that the power profile and potentially the logic itself can be obfuscated from potential attackers. It is difficult to determine the effective logic function without knowledge of past states.**

Keywords—Chaos; Chua's circuit; chaos logic; nonlinear circuits; integrated circuit design

I. INTRODUCTION

Limits in semiconductor device scaling as well as power and thermal constraints have led to the need for novel approaches to improving the performance of computer architectures. In recent years, a common approach has been to implement multi-core and even many-core processors that leverage parallelism. However, not all applications benefit from parallelism. A few novel approaches such as neuromorphic computing also continue to draw attention as possible ways to move beyond the limits of traditional von Neumann architectures. In this paper we explore another non-traditional computing paradigm: chaos based computing, that leverages the rich state space of chaotic oscillators to quickly converge on a solution.

The use of chaotic dynamical systems for computation was first proposed in [7]. The early approaches focused on using chains of coupled chaotic elements to perform arithmetic operations. Later approaches focused more on the implementation of logic gates. A logic gate capable of implementing the $\overline{AB}$, $\overline{A+B}$, $\overline{A}$, and $A \oplus B$ functions using threshold controlled chaotic elements was proposed in [6]. This implementation used a single iteration of the logistic map. In the same paper, a gate using Chua's circuit is described that can implement the functions $\overline{AB}$ and $\overline{A+B}$. A logic gate capable of implementing eight logic functions, AB, $A+B$, $\overline{AB}$, $\overline{A+B}$, $A \oplus B$, $\overline{A \oplus B}$, ON and OFF, was proposed in [5]. This last chaotic gate design leverages multiple iterations of the logistic map to allow for the implementation of any of the eight functions at different points in time and for different control conditions.

The gate proposed by Ditto et al. [5] is a one-dimensional gate as it depends on one state variable of the chaotic oscillator. The example gate described in [5] has two digital inputs, one output and, as mentioned above, is able to implement 8 of the 16 possible two-input logic functions. This limitation in the possible functions stems from the fact that the initial state is a function of the sum of the inputs. Thus, inputs {0,1} and {1,0} are indistinguishable from each other. In this paper we propose a two-dimensional gate using Chua's circuit [1,2]. By assigning each of the two inputs to its own state variable in Chau's circuit, the gate is able to implement all 16 2-input logic functions.

Implementing a chaotic logic gate requires more hardware than standard CMOS gates. As a result, it is desirable to increase the functionality of a single gate as much as possible. The two-dimensional gate has the additional benefit of having two outputs, making it capable of performing two different logic operations simultaneously, effectively doubling its functionality.

Another potential advantage of the two-dimensional chaotic gate proposed here is a more normal distribution of possible functions (frequency of functions) centered at 1/16. More specifically, the likelihoods for most of the possible functions over the state space of the system are equalized as the circuit evolves in time. A normal distribution for the frequency of functions is desirable from a security perspective in that the underlying operations of a circuit constructed with such gates can be obfuscated from potential attacks.

The remainder of this paper is outlined as follows. The one-dimensional chaotic logic gate is described in section 2 along with the Chua's circuit. In section 3, a two-dimensional chaotic logic gate is described. Experimental results for the two-dimensional gate are provided in section 4. Concluding remarks are provided in section 5.

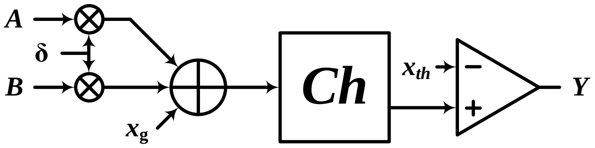

Fig. 1. Schematic illustration of a single state variable chaotic logic gate as described by Ditto et al. [5].

II. Background

A. The One-Dimensional Chaotic Logic Gate

A one-dimensional chaotic logic gate was first described by Ditto et al. [5]. The gate uses a chaotic oscillator (e.g. Chua's function or logistic map) to determine the functionality of the logic gate at any given time. Consider the schematic of a one-dimensional chaotic logic gate as shown in Fig. 1. A generic chaotic oscillator is represented by the box labeled '*Ch*'. The signal going into the box from the left is the initial condition and the output signal coming from the right side of the box is the state variable. The initial condition is formed by the sum of the analog control input x_g and two digital inputs *A* and *B*, each of which are converted to an analog value by multiplication with a constant weighting factor δ. The digital output *Y* is generated using the comparator with a threshold voltage x_{th}.

The logic functionality of the gate can be altered by varying the control input x_g, weighting factor δ, or the threshold x_{th}. If the chaotic oscillator is seen as a pseudorandom number generator, then x_g and δ determine the seed and x_{th} converts the analog output into a digital 0 or 1.

For a generic two-input reconfigurable logic gate, there are sixteen possible logic functions. However, for the one-dimensional chaotic logic gate [5] the input sets {0,1} and {1,0} will produce the same initial condition so the output for these two input sets will be identical. This restriction on the outputs reduces the number of possible logic operations the gate can perform to eight. These logic functions are as follows: OFF, *AB*, $A+B$, $\overline{AB}$, $\overline{A+B}$, $A\oplus B$, $\overline{A\oplus B}$, ON.

B. Chua's Circuit

At the heart of the chaotic logic gate shown in Fig. 1 is the chaotic oscillator *Ch*. One possible implementation is Chua's circuit (Fig. 2), an electronic oscillator capable of exhibiting chaotic behavior [1, 2]. Components *R*, C_1, C_2 and *L* are all standard linear passive components. The key element responsible for producing chaotic behavior is the Chua diode, G_N. This Chua diode is a nonlinear (piecewise linear in practice) negative resistor. When the absolute value of the voltage across the Chua diode is below some threshold, the resistance will be G_{NL}. For voltages higher than the threshold, the resistance will be G_{NH}. The Chua's circuit will exhibit double scroll chaotic behavior as shown in Fig. 3 with the proper component values. The effective values used for the circuit in Fig. 2 to produce the results of Fig. 3 are as follows: L=14.3mH, C_1=10nF, C_2=100nF, R=1.43KΩ, G_{NL}=-1.25KΩ, and G_{NH}=-2KΩ.

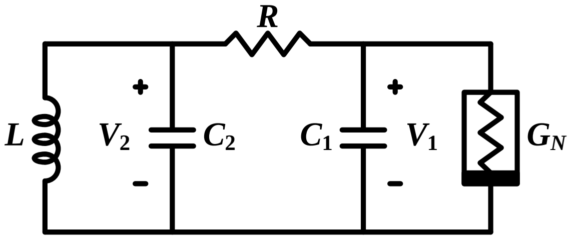

Fig. 2. Chua's circuit including a Chua diode exhibiting a negative impedance (G_N).

C. Experimental Implementation of Chua's Circuit

The double scroll in Fig. 3 is achieved experimentally via the circuit shown in Fig. 4. For experimental purposes, the Chua diode was implemented using a negative impedance converter and the inductor was implemented using a gyrator circuit. These respective components are highlighted in the schematic shown in Fig. 4. The negative impedance converter is composed of an op-amp and resistors R_{N1} to R_{N3} [3]. The resistance of this circuit as seen from the non-inverting input of the op-amp is equal to $R=-\frac{R_{N1}R_{N3}}{R_{N2}}$. This resistance is equal to the low voltage resistance G_{NL} of the Chua diode. As was done in [2], two diodes in series with R_{N4} provide the nonlinearity of the Chua diode. When the voltage across the Chua diode exceeds the threshold voltage of the diodes, they conduct, adding the resistor $\mathrm{R_{N4}}$ in parallel with G_{NL}. R_{N4} is selected so the parallel combination of R_{N4} and G_{NL} equals G_{NH}. The resistor values used in the implementation of the effective Chua diode circuit are as follows: R_{N1}= 220Ω, R_{N2}=15KΩ, R_{N3}=82KΩ, R_{N4}=3.3KΩ. For experimental purposes, the op-amp is a TL072.

The gyrator circuit [4] in Fig. 4 is composed of the second op-amp, R_{L1}, R_{L2}, and C_{L1}. This circuit will provide an inductance $L=R_{L1}R_{L2}C_{L1}$ in series with a parasitic resistance R_{L1} and in parallel with parasitic impedance formed by the series combination of R_{L2} and C_{L1}. To prevent these parasitic impedances from affecting the circuit significantly, R_{L1} is made small while R_{L2} and C_{L1} are selected to have high impedances. The component values used in the gyrator circuit are as follows: R_{L1}=10Ω, R_{L2}=150KΩ, C_{L1}=10nF, and the op-amp is a TL072.

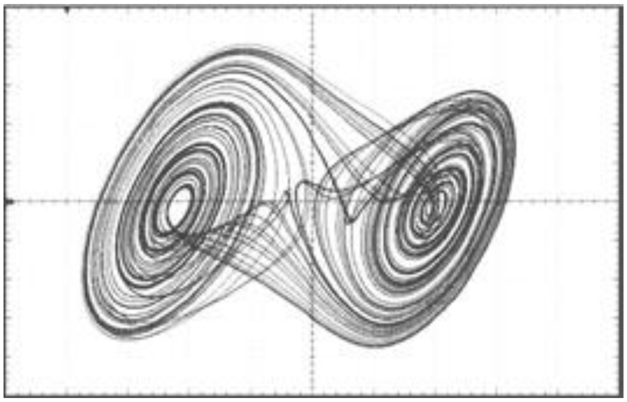

Fig. 3. State diagram of prototyped Chua's circuit showing V_2 (x-axis) vs. V_1 (y-axis), voltages across the capacitors. Generated as an XY plot on an oscilloscope for the circuit shown in Fig. 4.

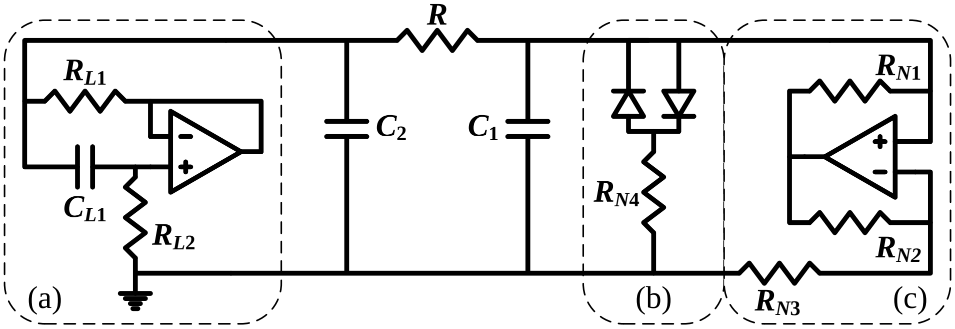

Fig. 4. Implementation of Chua's circuit for experimental analysis. Shown in dashed lines are (a) the gyrator circuit used to implement the inductor, (b) two diode circuit for nonlinearity, and (c) the negative impedance converter used to implement the Chua diode.

III. A Two-Dimensional Chaotic Logic Gate

Though some logic functions are achievable with the one-dimensional chaotic gate, it may be desirable to generate all functions via a purely chaotic gate. Such an implementation may also be of value in that little additional logic circuitry is required. The two-dimensional gate solves the problem of implementing all 16 logic functions and also provides a second output resulting in a gate that is able to implement two logic functions simultaneously. Since a single chaotic oscillator, such as Chua's circuit, can be used to implement two gates, the use of the two-dimensional gate can result in a reduction in hardware and power as compared to the one-dimensional gate.

The two-dimensional chaotic logic gate sets two state variables to initial conditions based on the two digital inputs and two control inputs. Two digital outputs are generated based on the two state variables. Since different initial conditions are generated for the {0,1} and {1,0} input states, the gate is not restricted to only eight logic functions as in the one-dimensional case.

A schematic of the two-dimensional chaotic logic gate is shown in Fig. 5. As can be seen in the figure, two state variables of the chaotic oscillator *Ch* can be initialized. Each state variable has an associated control input, $x_{g,A}$ and $x_{g,B}$. The digital inputs, *A* and *B*, are multiplied by weighting factors δ_A and δ_B, respectively, and added to the respective control inputs. The state variables associated with each input are initialized to $x_A(0) = x_{o,A} = \delta_A A + x_{g,A}$ and $x_B(0) = x_{o,B} = \delta_B B + x_{g,B}$. The state variables x_A and x_B are then allowed to evolve for some specified amount of time. The digital outputs are obtained in the same way as the one-dimensional gate. Comparators are used to test if a state variable is above or below the threshold x_{th} producing digital outputs Y_A and Y_B.

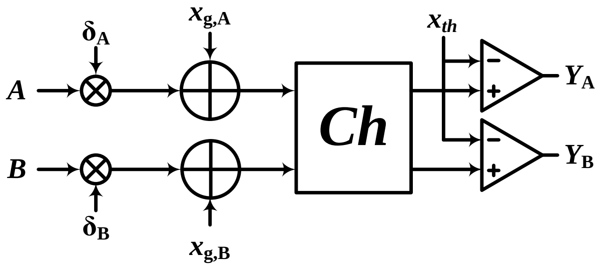

Fig. 5. A two-dimensional (leverages two state variables) chaotic gate. Allows implementation of all possible two-input logic functions.

The operation of the two-dimensional chaotic logic gate can be understood using the output response graph in Fig. 6. This graph plots the outputs of the two comparators after 2.5ms versus the two initial conditions, $x_{o,A}$ and $x_{o,B}$. The digital outputs for any combination of control inputs and digital inputs can be found using these diagrams. Each set of digital inputs, {0,0}, {0,1}, {1,0}, and {1,1}, will map to a different point in the state diagram, as illustrated by the orange box in Fig. 6. Since each digital input controls a separate state variable, the points corresponding to the digital input sets can be arranged in a box on the output response graph. The weighting of the digital inputs based on factors δ_A and δ_B will determine the size of the box. The two control inputs $x_{g,A}$ and $x_{g,B}$ will determine the location of the box on the output response. By changing the control inputs, the box can be moved and any logic function can be obtained.

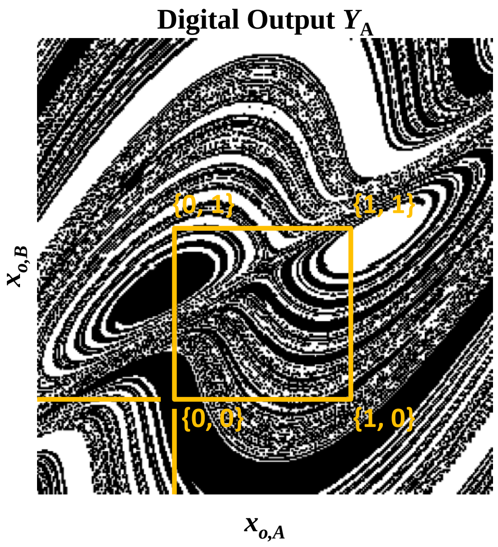

Fig. 6. Output response for the digitized state variable Y_A for the prototyped 2D chaotic gate based on Chua's circuit. 8-bit digital to analog converters (DACs) were used to generate the two initial conditions. Counters were used to cycle through all possible initial conditions. The digital outputs were read after a set period of time (2.5ms) by a Tektronix TLA7012 logic analyzer.

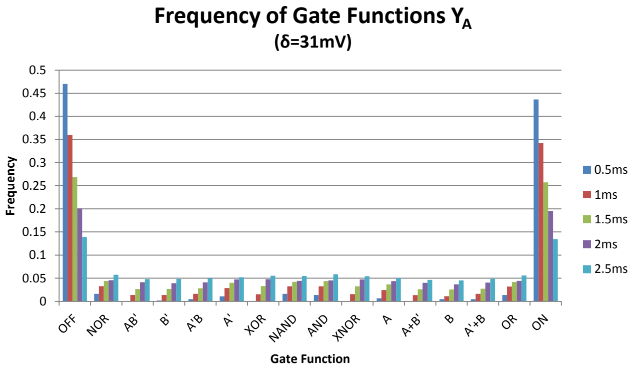

Fig. 7. Probability of functions for output Y_A of the two-dimensional chaotic gate.

IV. Experimental Results

Fig. 7 shows results for output Y_A of the two-dimensional chaotic gate. The frequency of functions is plotted for each gate function and five different evolution times. As can be seen in the figure, each function is capable of being implemented. The frequency of functions is much greater for the "ON" and "OFF" functions as a result of the large overlapping patches of one and zero output shown in the output response in Fig. 6. As the evolution time of the gate is increased, the patches become smaller resulting in more equal distributions of gate functions as shown in Fig. 7. The results also indicate that the likelihood of achieving any function slowly increases (or decreases in terms of "ON" and "OFF") towards .0625 (1/16) which is desirable.

It is not necessarily the case that the likelihoods will equalize as the circuit is allowed to evolve or if we have simply chosen the incorrect threshold and is thus biasing a 0 or 1 in the output. If the output threshold $x_{th,}$ of the gate is changed, the distribution of the output values will also change which would result in a different gate function distribution.

V. Conclusion

In this paper, a two-dimensional chaotic logic gate based on the Chua's circuit has been proposed and implemented. This gate is able to implement all 16 possible two-input logic functions and has two outputs such that it can perform two separate logic functions simultaneously. Further, the two outputs of the two-dimensional gate effectively doubles its functionality over the one-dimensional gate. In other words, the two-dimensional gate acts like two one-dimensional gates that each use only a single Chua's circuit.

The work in this paper has only explored a two-dimensional gate. The Chua's circuit has three state variables. In theory it is possible to make a three-dimensional gate that is capable of three digital inputs and three outputs. In practice, setting and reading the third state variable, the inductor current, is complicated by the fact that the capacitor C_{L1}, whose voltage is proportional to the inductor current, is not grounded. Using the third state variable would require more hardware, but this may be justified based on increased functionality.

Acknowledgment

We would like to thank Nathan McDonald and Bryant Wysocki of the Air Force Research Laboratory as well as Cory Merkel of the Rochester Institute of Technology for interesting and useful discussions on the topic.

This work was supported by AFOSR, LRIR 13RI05COR.

The material and results presented in this paper have been CLEARED (Distribution A) for public release, unlimited distribution by AFRL, case number 88ABW-2013-4663. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of AFRL or its contractors.

References

[1] L. O. Chua and G. Lin, "Canonical realization of Chua's circuit family," *IEEE Trans. Circuits Syst.*, vol. 37, no. 7, pp. 885–902, Jul. 1990.

[2] T. Matsumoto, L. O. Chua, and M. Komuro, "The double scroll," *IEEE Trans. Circuits Syst.*, vol. CAS-32, no. 8, pp. 797-818, Aug. 1985.

[3] B. Carter and L. P. Huelsman, "Handbook of operational amplifier active RC networks," Texas Instruments, Dallas, TX, Application Report SBOA093A, Oct. 2001.

[4] S. C. D. Roy and V. Nagarajan, "On inductor simulation using a unity-gain amplifier," *IEEE J. of Solid-State Circuits,* vol. 5, no. 3, pp. 95–98, Jun. 1970.

[5] W. L. Ditto, A. Miliotis, K. Murali, S. Sinha, and M. L. Spano, "Chaogates: Morphing logic gates that exploit dynamical patterns," *American Institute of Physics*, vol. 20, no. 037107, Sep. 2010.

[6] W. L. Ditto, K. Murali, and S. Sinha, "Chaos computing: ideas and implementations," *Phil. Trans. R. Soc. A*, vol. 366, no. 1865, pp. 653–664, Feb. 2008.

[7] S. Sinha, and W. L. Ditto, "Dynamics Based Computation," *Phys. Rev. Lett.*, vol. 81, no. 10, pp. 2156–2159, Sep. 1998.