Design of a Lightweight Reconfigurable PRNG Using Three Transistor Chaotic Map

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

IEEE 62nd International Midwest Symposium on Circuits and Systems (MWSCAS), Dallas, TX, USA, August 2019.

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

Citation Information (BibTex):

```
@INPROCEEDINGS{MWSCAS: shanta2019design,
  author=" A. S. Shanta and M. S. Hasan and M. B. Majumder
       and G. S. Rose",
  title="Design of a Lightweight Reconfigurable PRNG Using
Three Transistor Chaotic Map",
  booktitle="{IEEE} International Symposium on Circuits
      and Systems {MWSCAS}",
  month="August",
  year="2019",
  address="Dallas, TX, USA"
}
```

Design of a Lightweight Reconfigurable PRNG Using Three Transistor Chaotic Map

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

***Abstract*—In this paper, we implemented a lightweight reconfigurable pseudo-random number generator (PRNG). The map generates discrete-time chaotic signals. The chaotic oscillator requires two map circuits to generate chaotic signals. The design is reconfigurable because two map circuits have two bifurcation parameters which can be leveraged to generate multiple random sequences. The design of the PRNG utilizes two chaotic oscillators, analog mux, 10 bit ADC, 2 bit shift register and an XOR gate. The circuit has been implemented in 65 nm CMOS technology with a supply voltage of 1.2 V. The estimated power consumption of the circuit is 2.12 mW and the area overhead is 0.132 mm^2. The throughput of the proposed PRNG is 100 MS/s. The proposed PRNG can be used for security applications in IoT devices which requires circuits with less area and power overhead. The PRNG design passes all the tests in NIST SP 800-22 test suite.**

***Index Terms*—Chaotic map, PRNG, CMOS, VLSI, hardware security, IoT**

I. Introduction

True random numbers can be generated from various sources present in nature. The sequence can be extracted from thermal noise, electronic noise, shot noise, movement of mouse on the computer screen, lava lamps, etc. Data extraction from these sources is not feasible because the bit rate of the generated binary data is low. Moreover, the design becomes complex and the data is vulnerable to external environment. The rising challenges of designing true random number generators led researchers to develop pseudo-random number generators (PRNGs).

Linear RNGs are breakable after a few iterations which is why nonlinear RNGs are becoming popular. Pseudo-random sequences are expected to have long periods. If linear shift registers are used then it becomes costly since the circuit complexity increases and large memory is required to store the bits. Recently a lot of research is conducted on chaos theory to develop nonlinear pseudo-random number generators for cryptographic applications. The theory of chaos is considered to be one of the most monumental discoveries of the 20^{th} century. Chaotic dynamical systems produce deterministic and unpredictable phenomenon which is sensitive to small changes in the initial conditions. Chaotic sequences inherently have long periods which is a desirable property for PRNGs. The security performance of the system depends on the randomness of the sequence. Popular chaotic maps are Gauss map, logistic map, tent map, etc.

Nonlinear pseudo-random number generators (PRNGs) have applications in the implementation of strong authentication protocols, generation of keys for block cipher, generation of a stream of keys for stream cipher, etc. In many cryptographic protocols or algorithms, random numbers play a vital role in the computation. PRNGs are useful in generating challenge-response pairs for authenticating systems, public and private key pairs for asymmetric algorithms and for commercial applications such as lottery machines. Smart cards also require random numbers to protect against side-channel attacks [1].

Researchers have worked on implementing popular chaotic maps to develop PRNGs. Linear generators can be coupled with nonlinear chaos generators in order to make the PRNG attack resistant [2]. Nonlinear PRNGs can be easily developed from nonlinear recurring equations. Blum's nonlinear generator utilizes a square function and modulus operation but its process time and hardware cost is higher [3]. Nonlinear feedback shift registers have long-period and high throughput [4]. One of the important characteristics of a good PRNG is long period length which means the number of outputs until the the sequence repeats itself. Several techniques have been explored by researchers to increase the period length of a chaotic gate such as self reseeding [5], modifying state transitions [6], exploiting nonlinear effects of truncation [7], adding noise [8], etc.

We introduce a novel scheme to generate random numbers using a three transistor map circuit [9]–[11]. In this paper, we connected two one-dimensional (1-D) chaotic maps in cascade mode in order to develop the chaotic oscillator. One-dimensional maps are mathematical systems where a single variable evolves with continuing iterations. The raw sequences extracted from the chaos gate have innate properties that are significant for random sequences such as high sensitivity towards small perturbations in the initial input. A standard approach of generating random sequences is using two discrete-time chaotic map and compare the generated sequences. Common discrete-time chaotic maps such as logistic map, piecewise linear map, etc. gives only one bit per iteration. In this paper, the design of the PRNG requires two chaotic oscillators, 10 bit analog-to-digital converter (ADC), analog mux, a 2 bit shift register (SR) and an XOR gate. It is possible to extract two bits per iteration from each chaotic oscillator.

The remainder of the paper is organized as follows. Section

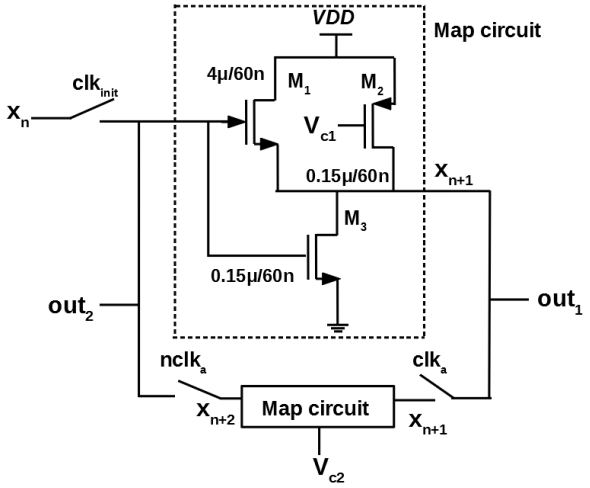

Fig. 1: Chaotic oscillator [9]. The dotted lines represent the three transistor chaotic map circuit.

II describes the circuit operation, bifurcation parameter and Lyapunov exponent. The usage of chaos generator as a PRNG is described in Section III. Section IV talks about the overhead analysis of the proposed PRNG. Section V describes the possible applications of the chaos gate as a PRNG. Finally, concluding remarks are given in Section VI.

II. Circuit Design

A three transistor map circuit has been implemented in 65 nm CMOS process. The size of the transistors in the map circuit are: M1, W = 4 µm, L = 60 nm, M2 and M3, W = 150 nm, L = 60 nm. T The chaotic oscillator along with the clocking circuits consumes 0.556 μm^2 of area and 18.4 µW of power. In conventional designs, capacitors are used in the sample-and-hold circuit to store the signals. In this topology, the parasitic capacitance of the pass transistors can be leveraged to store the data. This reduces overhead since capacitors consume a huge amount of area on chip. The chaos generator is shown in Fig. 1. Two map circuits are used in the generator, one in the forward path and the other in the feedback path. Non-overlapping clocks, clk_a and $nclk_a$ are used to sample-and-hold the outputs of the chaotic oscillator. Initially, the input, also known as the seed, is applied from outside and clk_{init} passes it into the input of the first chaotic map. The first output is sampled from out_1. In the second iteration, out_1 is transferred to the input of the second chaotic map using clk_a. The feedback map generates an output and it is sampled as out_2. When $nclk_a$ is high, out_2 is sampled onto the input of the forward chaotic map. The cycle continues and two analog voltages are generated at each iteration.

Three control parameters, input voltage, V_{in} and control voltages V_{c1} and V_{c2}, are available in the chaotic oscillator. The two control voltages can be shorted together or different voltages can be applied to generate a series of different sequences. We call the proposed PRNG reconfigurable because slight change in V_c will generate a new sequence. Different

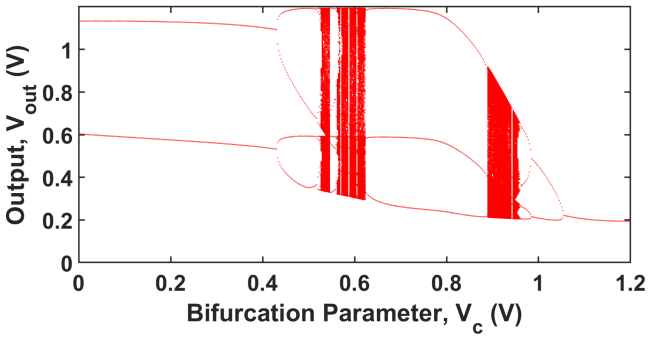

Fig. 2: Bifurcation diagram of the chaotic oscillator.

combinations of the two control voltages can be explored to generate random numbers. In earlier work, we demonstrated a way to increase the design space by applying varying V_c at each iteration [12]. In order to ensure that the chaotic oscillator acts as a PRNG, the control voltage should be chosen from the chaotic region. The chaotic region is estimated from the bifurcation diagram and Lyapunov exponent.

Bifurcation diagram displays the output sequences for changes in bifurcation parameter, V_c. As can be seen from Fig. 2, two chaotic regions are generated from the chaotic oscillator. The first region starts from 527.5 mV and ends at 622.5 mV. The second chaotic region lies between 890 mV and 952.5 mV. The control parameter, V_c is chosen from the first region because the range of the output voltage is higher. The PRNG's output will span over a range of 294 mV to 1.194 V in the first region. The seed value is also chosen from this output range. Moreover, the delay of the circuit increases as V_c increases. In order to make the chaotic oscillator faster, it is wise to choose control voltages from the first region.

Lyapunov exponent is a measure of the chaotic behavior of a nonlinear system. Positive Lyapunov exponent means that two trajectories generated using close initial conditions diverges faster from one another with each iteration. Fig. 3 represents the Lyapunov exponent of the chaotic oscillator. There are two regions where Lyapunov exponent is greater than zero. The chaotic regions derived from the Lyapunov exponent are consistent with the results obtained from the bifurcation diagram. In this paper, a single control voltage is used for random number generation. V_c is chosen where the Lyapunov exponent has the most positive value, i.e. 592.5 mV.

III. Pseudo Random Number Generator and Results

Fig. 4 displays the proposed pseudo-random number generator using two chaotic oscillators. The chaotic oscillator alone is not sufficient for random number generation. In order to enhance diffusion in the generated binary sequence, we generate 2 million analog data from chaotic oscillator, CO_A and 1 million data from CO_B. Clk_1 has double the frequency of clk_2 so that Sel pin of the MUX (same frequency as clk_1)

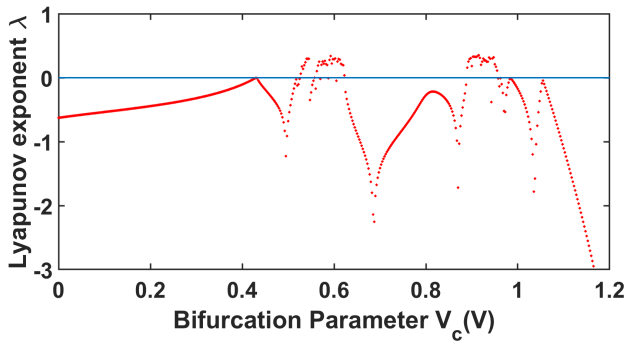

Fig. 3: Lyapunov exponent of the chaotic oscillator.

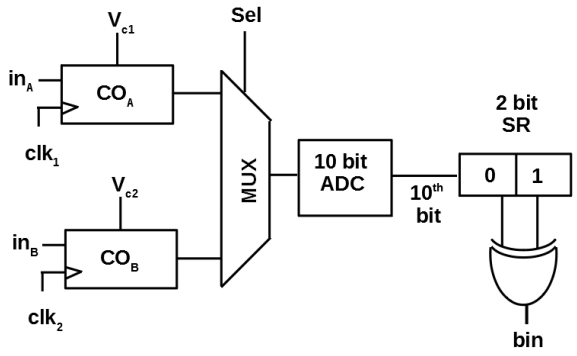

Fig. 4: The proposed PRNG design using chaotic oscillators.

can choose which analog voltage to transfer to the input of the ADC. This mechanism uses a single ADC and reduces area and power overhead. From CO_A, every other analog data is sampled starting at second position and from CO_B the first 1 million sequences are sampled. The ADC converts the analog voltages into binary sequence and stores only the 10^{th} bit in the 2 bit shift register. We only used the least significant bit (LSB) of the ADC because it provides the highest entropy required for developing the pseudo-random number sequence. As soon as two binary data is available from two chaotic oscillators, they are XORed to enhance the randomness and the resulting output (bin) is sampled as the final bit sequence.

An initial analog voltage is applied as a seed to the proposed PRNG, in_A = in_B. The same control voltage has been used which means same V_c has been applied to the two chaotic oscillators, V_{c1} = V_{c2}. V_c has been chosen to be 592.5 mV with the help of bifurcation diagram and Lyapunov exponent. The seed should be chosen from the output range of the selected V_c, 310 mV to 1.194 V. After the sequences are generated, tests are performed on the binary sequences to establish whether the sequences are applicable for security purposes.

NIST test suite measures the randomness of the generated binary sequence by applying different statistical tests. The tests are validated by comparing the probability statistics ($p-value$) calculated from the data and the significance value, α which is usually equal to 0.01. If there exists more than one $p-value$ for the sequence then the average value is calculated. The pass rate of the tests has to be at least 0.96. From Table I, it can be seen that the generated binary data from the proposed PRNG passes 15 out of 15 NIST tests. We have validated the randomness of the generated data using 100 one million sequences. The 100 different sequences have been generated for 100 different seeds.

TABLE I: NIST test results of the proposed PRNG.

NIST test	P-value	Pass-rate
Frequency	0.0966	0.99
Block frequency (m = 128)	0.3838	0.98
Cumulative sums*	0.4356	0.99
Runs	0.7792	1
Longest runs of ones (M = 10000)	0.8832	1
Rank	0.4190	1
DFT	0.8514	0.99
Non-overlapping template* (m = 9)	0.4994	0.98
Overlapping template (m = 9)	0.0329	1
Universal (L = 7, Q = 1280)	0.6371	0.99
Approximate entropy (m = 10)	0.2368	0.99
Random excursion*	0.2558	0.99
Random excursion variant*	0.2213	0.99
Serial* (m = 16)	0.52155	0.99
Linear complexity (M = 500)	0.3191	0.98

*average result of mutiple tests is shown

IV. Overhead Analysis

PRNG designs based on established chaotic maps such as, logistic map, piecewise linear map, tent map, etc., can be found in literature. Circuit implementation of these maps in CMOS technology can be area consuming. In order to reduce the area, only one bifurcation parameter is chosen which reduces the available control parameters of the chaotic circuit [5]. The circuit used in this paper is reconfigurable because it retains all the control parameters which is important for generating different chaotic sequences. Moreover, different control voltage combinations should generate different PRNG sequences. We can also leverage the body bias voltage to add more control parameters to the design [15].

The generated pseudo-random sequences need to pass the NIST tests so that it is applicable for security applications. There are several post processing schemes in the literature but they add more complexity to the existing design. Moreover, post-processing techniques add to the delay of the circuit and becomes more area and power hungry. In this paper, we used a simple scheme to add more randomness to the generated data. After the two sequences are generated from the chaotic oscillators, the two analog voltages are applied to the ADC using an analog MUX. The LSB of the binary values are stored in the shift register and then XORing is performed on the incoming two bits. This post-processing adds negligible area and power overhead.

The proposed lightweight PRNG has been designed in 65 nm CMOS technology with a supply voltage of 1.2 V. The area of the chaotic oscillator is only 0.556 μm^2. A low power and less area consuming SAR ADC [16] can be implemented

TABLE II: Overhead comparison of established PRNGs with the proposed design.

	Technology (nm)	**Supply Voltage** (V)	**Area** (mm^2)	**Power** (mW)	**Throughput** (MS/s)
[13]	350	3.3	0.752	56	40
[13]	180	1.8/3.3	0.126	22	100
[14]	180	1.8	0.275	13.9	6400
[8]	180	1.8	0.767	37	120
This work	**65**	**1.2**	**0.132**	**2.12**	**100**

in order to complete the design. The ADC circuit only consumes 1.6 mW of power and the area overhead is 0.132 mm^2. The total overhead of the PRNG design including the chaotic oscillators, ADC, shift register, analog MUX and XOR gate is approximately 0.132 mm^2 and the power consumption is 2.12 mW. As can be seen from Table. II, this system consumes significantly less power compared to prior work.

The throughput of the chaotic oscillator is 330 MS/s but the throughput of the ADC is only 100 MS/s. Therefore, the throughput of the entire system is defined by ADC's throughput. If we use the entire 10 bits from the ADC, then it is possible to increase the throughput to 1000 MS/s but we will have to adopt a more complex post-processing scheme to pass all the NIST tests. A Flash ADC can also be used if only the 10^{th} bit from the ADC is used, but the trade off is high power consumption and more area overhead. A bigger three transistor map circuit will be faster but will have larger overhead in terms of power and area.

V. Application

IoT devices are becoming increasingly popular with the advancement in technology. The security of these devices is a major concern since a lot of information is exchanged over the internet. PRNGs are an integral part of security domain and is required for generation of keys, challenge-response pairs, authentication protocols, etc. IoT devices are resource-constrained so circuits with less area overhead is required for security purposes. The proposed PRNG topology is ideal for security purposes in IoT devices since it consumes low power and area overhead. The PRNG passes all fifteen tests in the NIST SP 800-22 suite. The randomness provided by this PRNG should be sufficient for the security of small devices.

VI. Conclusion

A lightweight reconfigurable PRNG is designed in 65 nm CMOS process. The core of the design is a three transistor map circuit. The entire system of the random number generator consumes only 2.12 mW of power and 0.132 mm^2 of area. The throughput of the PRNG is 100 MS/s. The generated pseudo-random sequence passes all the NIST tests with minimal post-processing overhead. The proposed design is suitable for the security of IoT devices since it consumes very little area and power compared to the established chaotic map based PRNGs. The circuit design is reconfigurable since it contains multiple control voltages.

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] W. Rankl and W. Effing, *Smart card handbook*. John Wiley & Sons, 2004.

[2] C.-Y. Li, H.-P. Chou, L.-Y. Deng, J.-J. H. Shiau, and H. H.-S. Lu, "Nonlinear pseudo-random number generators via coupling dx generators with the logistic map," in *Anti-counterfeiting, Security, and Identification*. IEEE, 2012, pp. 1–5.

[3] L. Blum, M. Blum, and M. Shub, "A simple unpredictable pseudo-random number generator," *SIAM Journal on computing*, vol. 15, no. 2, pp. 364–383, 1986.

[4] B. M. Gammel, R. Gottfert, and O. Kniffler, "An nlfsr-based stream cipher," in *2006 IEEE International Symposium on Circuits and Systems*. IEEE, 2006, pp. 4–pp.

[5] C.-Y. Li, T.-Y. Chang, and C.-C. Huang, "A nonlinear prng using digitized logistic map with self-reseeding method," in *Proceedings of 2010 International Symposium on VLSI Design, Automation and Test*. IEEE, 2010, pp. 108–111.

[6] J. Keller and G. Spenger, "Tweaking cryptographic primitives with moderate state space by direct manipulation," in *2017 IEEE International Conference on Communications (ICC)*. IEEE, 2017, pp. 1–7.

[7] T. Addabbo, M. Alioto, A. Fort, A. Pasini, S. Rocchi, and V. Vignoli, "A class of maximum-period nonlinear congruential generators derived from the rényi chaotic map," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 54, no. 4, pp. 816–828, 2007.

[8] H.-T. Yang, J.-R. Huang, and T.-Y. Chang, "A chaos-based fully digital 120 mhz pseudo random number generator," in *The 2004 IEEE Asia-Pacific Conference on Circuits and Systems, 2004. Proceedings.*, vol. 1. IEEE, 2004, pp. 357–360.

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

[10] M. B. Majumder, M. S. Hasan, M. Uddin, and G. S. Rose, "Chaos computing for mitigating side channel attack," in *2018 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*. IEEE, 2018, pp. 143–146.

[11] M. B. Majumder, M. S. Hasan, A. Shanta, M. Uddin, and G. Rose, "Design for eliminating operation specific power signatures from digital logic," in *Proceedings of the 2019 on Great Lakes Symposium on VLSI*. ACM, 2019, pp. 111–116.

[12] A. S. Shanta, M. B. Majumder, M. S. Hasan, M. Uddin, and G. S. Rose, "Design of a reconfigurable chaos gate with enhanced functionality space in 65nm cmos," in *2018 IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS)*, Aug 2018, pp. 1016–1019.

[13] F. Pareschi, G. Setti, and R. Rovatti, "Implementation and testing of high-speed cmos true random number generators based on chaotic systems," *IEEE transactions on circuits and systems I: regular papers*, vol. 57, no. 12, pp. 3124–3137, 2010.

[14] C.-Y. Li, Y.-H. Chen, T.-Y. Chang, L.-Y. Deng, and K. To, "Period extension and randomness enhancement using high-throughput reseeding-mixing prng," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 20, no. 2, pp. 385–389, 2012.

[15] M. S. Hasan, M. B. Majumder, A. S. Shanta, M. Uddin, and G. S. Rose, "Evaluation, optimization, and enhancement of chaos based reconfigurable logic design," in *2019 IEEE SoutheastCon*, Apr 2019.

[16] J. Ma, Y. Guo, L. Li, Y. Wu, X. Cheng, and X. Zeng, "A low power 10-bit 100-ms/s sar adc in 65nm cmos," in *2011 9th IEEE International Conference on ASIC*. IEEE, 2011, pp. 484–487.