Analog Integr Circ Sig Process (2012) 73:363–374
DOI 10.1007/s10470-012-9893-9

Discrete-time chaotic-map truly random number generators: design, implementation, and variability analysis of the zigzag map

Hamid Nejati · Ahmad Beirami · Warsame H. Ali

Received: 6 March 2012 / Accepted: 4 June 2012 / Published online: 27 June 2012

Abstract In this paper, we introduce a novel discrete chaotic map named zigzag map that demonstrates excellent chaotic behaviors and can be utilized in truly random number generators (TRNGs). We comprehensively investigate the map and explore its critical chaotic characteristics and parameters. We further present two circuit implementations for the zigzag map based on the switched current technique as well as the current-mode affine interpolation of the breakpoints. In practice, implementation variations can deteriorate the quality of the output sequence as a result of variation of the chaotic map parameters. In order to quantify the impact of variations on the map performance, we model the variations using a combination of theoretical analysis and Monte-Carlo simulations on the circuits. We demonstrate that even in the presence of the map variations, a TRNG based on the zigzag map passes all of the NIST 800-22 statistical randomness tests using simple post processing of the output data.

H. Nejati (✉)
Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109, USA
e-mail: hnejati@umich.edu

A. Beirami
School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA
e-mail: beirami@ece.gatech.edu

W. H. Ali
Electrical and Computer Engineering Department, Prairie View A&M University, Prairie view, TX 77446, USA
e-mail: whali@pvamu.edu

Keywords Truly random number generator · Current-mode circuits · Switched-current technique · Variability · Discrete-time chaotic maps · Tent map

1 Introduction

Chaos is a non-periodic, long-term non-predictive behavior that can be generated by certain nonlinear dynamical systems [1]. The chaotic systems are inherently deterministic given the initial state of the system. The chaotic behavior is a result of the exponential sensitivity of the system to the initial state that can not be exactly determined in practice. Chaotic waveforms have been extensively used in various research areas including biology for the modeling of the behavior of human organs [2], telecommunications for the modulation of the signals, chaotic oscillators, and chaotic encryption of the data [1, 3, 4], and cryptography for the security of the system [5].

Cryptography serves as the security block for the transmission of a message in the presence of eavesdroppers [5]. The security of a cryptographic system relies on the security of the truly random number generator (TRNG). A TRNG is capable of producing uncorrelated and unbiased binary sequences through a nondeterministic and irreproducible process [6]. Many practical cryptographic systems use a pseudo-random number generator (PRNG) as their random bit generator. PRNG is a deterministic system that produces a periodic limited-length binary sequences using a much shorter seed [5]. The period of the generated sequence grows exponentially with the length of the seed. The output of the PRNG with length much less than one period demonstrates random behavior in terms of bias and correlation between the generated bits. However, the entropy of any sequence generated using a PRNG is limited to the entropy of the seed sequence by the data processing theorem [7]. Consequently, a truly random seed maximizes the entropy of the sequence. Furthermore, the seed has to be periodically changed to keep the system secure. The large period of PRNG requires a large amount of storage capacity and redundant circuitry that makes PRNGs ineffective for high speed low cost cryptosystems [5].

Software programs have been used for the implementation of TRNGs, such as time interval between keystrokes or mouse movements, system clock, content of input–output (I/O) buffers, and operating system values such as network statistics [5, 8]. Similar to most PRNGs, software based TRNGs generate output data using computer, which is in a large but finite number of states resulting in a pseudo-random output sequence [5]. In order to enhance the quality of these TRNGs, a complex combination of different sources of randomness should be used. This combination is practically inefficient, slow, and non-integrable.

On the other hand, hardware based TRNGs utilize inherent randomness of circuitry. The high entropy TRNG can be generated by the inherent randomness of the natural phenomena such as, the thermal and shot noise of the analog circuitry and the frequency instability (phase noise) of oscillators [5]. Noise is presented in electronic circuits and it is also an important parameter in designing amplifiers and RF/analog systems [9–16]. Direct amplification of the thermal noise of a resistor followed by a sampling stage seems to generate random sequences. However, this structure can not be used in integrated circuits due to the sensitivity to deterministic environmental variations such as power supply signals [17]. Most of the existing TRNGs are based on oscillator jitter, which is the variation of low-to-high and high-to-low transition points of the oscillator. If a low frequency oscillator with high jitter is used to sample the output of a high frequency oscillator, a random sequence can be obtained. Intel and Via use this structure in their security platforms [18, 19]. However, the bit rate of a jitter-based random bit generator is limited to about 100 kbps [20], which is not suitable for high speed cryptosystems. The use of continuous-time chaotic systems has been recently proposed for the generation of random numbers [21]. However, these methods suffer from the high correlation between the generated bits resulting in complicated post-processing of the generated bits. Therefore, these methods are not efficient in practice due to the sensitivity to the deterministic environmental variations that deteriorates the quality of the output random bit streams. In conclusion, these methods are not efficient in practice due to the sensitivity to deterministic environmental variations that deteriorates the quality of the output random bit streams.

A discrete-time chaotic map, formed by the iteration of the output value in a transformation function, can be used for the generation of random numbers. Simple piecewise affine I/O characteristics have been extensively used for the generation of random bits, e.g., the Bernoulli map [6] and the tent map [22]. The entropy source of a chaotic map is the inherent noise of the system, which is amplified in the positive gain feedback loop by the iteration of the output signal in the map function [6]. The output of the system will be unpredictable after a few first output samples. Pipelining multiple stages of the chaotic map circuit can increase the overall open loop gain of the system. The increased open loop gain can reduce the number of bits that need to be discarded in the beginning of the sequence due to the small inherent noise of the system. Pipelining multiple stages of the chaotic map circuit can also increase the speed of the whole system while contributing to the randomness and therefore improving the quality of the generated bits. The output of each stage is connected to the input of the successive stage. High Speed, capability of integration, and the high quality of the generated bits make the discrete-time chaotic maps excellent candidates for high speed embeddable random number generators. Discrete-time chaotic TRNGs are much faster than the jitter based TRNGs. The high speed of these systems mainly owes to the circuit implementation and the pipelined structure, which can increase the output bit rate. The capability of being integrated is another important positive point for this kind of TRNGs.

Imperfections in practical implementation of TRNGs, so-called non-idealities, result in bias and correlation in the generated binary sequence of a truly random bit sequence. Hence, the post processing of the output data is required to improve the statistical characteristics of the bit sequence. Neumann [23], SHA-1 [24], and SHA-2 [25] are the most widely used post-processing algorithms. All of these methods suffer from the trade off between bit-rate and the entropy of the generated binary sequence. Theory suggests that the knowledge of the non-ideality of the chaotic map can improve the efficiency of the post processing unit [26–28]. NIST 800-22 statistical test suite is a standard evaluation tool for the examination of the quality of the generated bit stream [29]. This test can only prove that a large binary sequence is not random if it fails the tests. This test suite can not be used for the evaluation of the true randomness of the binary sequence. It is impossible to analytically prove the truly randomness of a sequence of data.

The practical application of both the tent map and the Bernoulli map can be hindered by noise and implementation errors, where they are unable to maintain the state of the system confined in the map [6]. In the preliminary version of this work [30] , we proposed zigzag map as a modification to the tent map that would make it robust to implementation variations and presented a very basic implementation. The zigzag map is interchangeable with the tent map in practice while solving the confinement problem. In this paper, we present two implementations for the zigzag map based on the switched current technique as well as the current-mode affine interpolation of the breakpoints for high speed applications. The current mode circuits are capable of producing high gain, less distortion, and high speed compared to voltage mode circuits. Less sensitivity to switching noise and process variations are other benefits of the current mode circuits. The effect of process variation on the parameters of the chaotic map is explored and a set of Monte Carlo simulations is used to quantize the map non-idealities. The second circuit demonstrated less sensitivity to process variations. The post-processing of the output data of a non-ideal map passes the randomness test suite.

This paper is organized as follows. In Sect. 2, the practical problems of the Bernoulli and tent maps are pointed out. Then, we present the zigzag map and investigate its chaotic characteristics. In Sect. 3, we explain the feasibility of implementation of the presented zigzag map. The deviation of the circuit I/O characteristic from the desired map deteriorates the quality of the output random bit sequences. This effect is modeled and analyzed in Sect. 4. The generated sequence is fed to a post-processing unit and validated against NIST 800-22 randomness test suite. Section 6 concludes the paper.

2 Discrete-time chaotic map TRNGs

2.1 Common discrete-time chaotic maps

Discrete-time chaotic maps are a subclass of discrete-time nonlinear dynamical systems that can exhibit chaotic behaviors that are formed by the iteration of the output of the system into the system in every time step. A chaotic map is deterministic. If the initial state of a deterministic system is exactly known, the output behavior can be predicted exactly. Impossibility of exact knowledge of the initial state in practice is the reason of the chaotic behavior.

Bernoulli and tent maps are two most commonly used discrete-time chaotic maps. The Bernoulli shift map, shown in Fig. 1(a), is a piecewise affine map that exhibits chaotic behavior. A 0 is generated if $x_n < 1/2$ and a 1 is generated if $x_n > 1/2$. The probability of all transitions are equal to 1/2 regardless of the current state of the system. A pipelined ADC is used for practical implementation of the Bernoulli shift map. The I/O characteristic of a pipelined ADC is shown in Fig. 1(b). In practice, this map suffers from not being able to confine the output sequence in the chaotic region of $(-1, 1)$. If for any reason the input value becomes slightly greater than 1 or less than -1, the output stream gets out of the map and will never return back to the map. Decreasing the slope from 2 and using $1\frac{1}{2}$-bit ADC are the most widely solutions used to overcome this problem [6, 31], which results in very complicated structures.

The tent map, shown in Fig. 1(c), also exhibits chaotic behavior [22, 32]. This system has been proved to be asymptotically stable with a uniform density distribution. 0 is defined as (0, 1/2) and 1 is defined as (1/2, 1). The proposed structures are sensitive to implementation variations [33]. A tailed tent map was presented in [33], which has a uniform density distribution. However, the process of generating bits from the tailed tent map is not straightforward. However, the post-processing required to compensate for non-idealities can result in complicated circuitry or reduced output bit-rate.

2.2 Presented zigzag map

We introduce a novel discrete chaotic map and call it the zigzag map, shown in Fig. 1(d). It is notable that the output values of the zigzag map will alternate between positive and negative values. We will show later that the alternation between positive and negative values will be helpful in the presence of implementation non-idealities. As shown in Fig. 1(d), the tent map and the zigzag map have outputs that are equal in absolute value due to the symmetry but the output values alternate between positive and negative values in the zigzag map. Thus, the absolute value of the

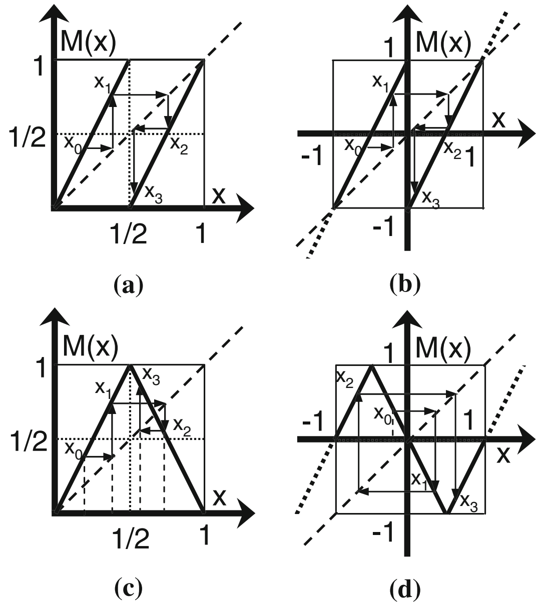

Fig. 1 **a** The Bernoulli shift map. **b** Practical implementation of the Bernoulli shift map using a pipelined ADC. **c** The tent map. **d** Our presented discrete chaotic map, called zigzag map

generated output stream is the same for both maps. In other words, if x_1, x_2 and x_3 are three successive outputs of the tent map, the outputs of the zigzag map will be given by x_1, $-x_2$ and x_3. Generation of bits from this map is straightforward. $|x_n| < 1/2$ represents 0 and $|x_n| > 1/2$ represents 1. The generalized zigzag map can be defined as

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

where m is a real number and $m \in (-3, 3)$. In this equation, $m = -2$ represents the zigzag map. This formulation can be used to find the bifurcation diagram of the presented chaotic map with respect to the bifurcation parameter m (Fig. 2). A bifurcation diagram can be used to identify the chaotic behavior of a system. It can represent the possible steady state values of a dynamical system as a function of the bifurcation parameter.

Without loss of generality, the initial state of the system has been assumed to be a slightly positive number, i.e., $x_0 = 0^+$. We start analyzing the diagram from larger values of $m{\cdot}m = 3$ is a critical point in the diagram. This case is at the verge of instability, since the system can be driven out of the map. In $2 < m < 3$, the system is chaotic and stable. Chaos can be observed for $1 < m < 2$. Since the initial state of the system has been assumed to be positive, the output value is confined to positive values. If the system gets out of the positive region and goes into the negative region it will get trapped in the negative region. For $|m| < 1$, no chaotic behavior is observed. The output of the system will ultimately settle in zero. $-2 < m < -1$ is also a chaotic region for the system. The output in this region alternates between positive and negative values. Note that there is no asymptotic density distribution since for odd and even time steps, the output values are in the positive and negative regions, respectively. In other words $\lim f_n(x)$ does not exist as $n \to \infty$. The bifurcation diagram of the tent map is the same as given here for $0 < m < 2$.

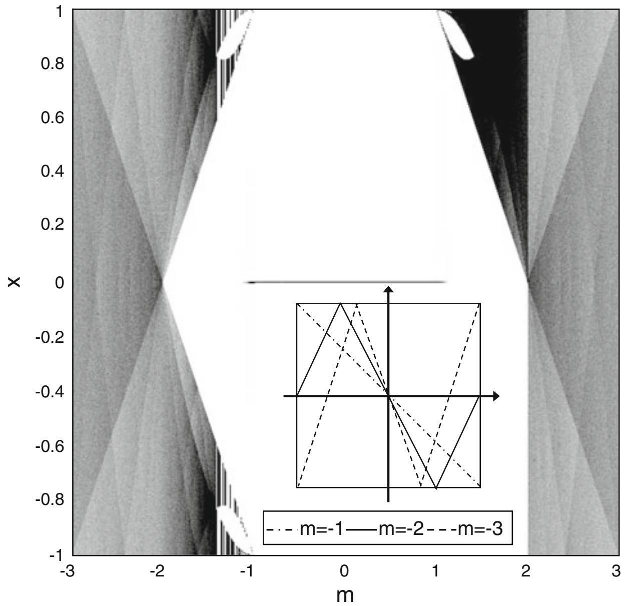

Fig. 2 Bifurcation diagram for the zigzag map with bifurcation parameter m, used in (1). The generalized zigzag map with three different values of m (-1, -2, and -3) is shown in the inset

3 Switched-current circuit implementation for the zigzag map

Current mode circuits have recently attracted great attention [34]. The signal is represented by current of branches instead of voltage of the nodes in the traditional voltage-mode circuits. Reduced distortion, high gain, low input and high output impedances, high speed, less sensitivity to switching noise and better ESD immunity are other excellent characteristics of current mode circuits that makes them one of the best candidates in the future telecommunications, analog signal processing, multiprocessors, and high speed computer interfaces [34].

In order to consider the effect of inter-die (different devices in one chip) or intra-die (among devices on various chips) variations, we design a current mode circuit with low sensitivity to the process variations. The symmetric double break point current mirrors are less sensitive to the process variations compared to other implementation of piecewise linear I/O current characteristics [35]. If we combine two different double break point circuits together, we can expect the I/O characteristic curve, shown in Fig. 1(d). One of the current mirrors is implemented by NMOS transistors, while for the sake of symmetry the other one is implemented by PMOS transistors. The double break point current mirrors do not accumulate errors originated from process variations. It is straightforward to design a desired piecewise linear I/O curve with this circuit. The preprocessing part of the circuit consists of two track and hold stages that enable the iteration of the output into the system.

The upper side of the circuit, shown in Fig. 3(a), can generate the double break point current mirror with the gain of 4, while the gain of the lower part is 2. Adding the output current of this two structures, we can generate the zigzag map by the I/O characteristic of the presented circuit. I_1, I_2, and I_3 are the current sources connected to the upper and lower side, discerning by an extra p and n index for upper (PMOS) and lower (NMOS) circuits, respectively. The values of these current sources are as follows:

$$I_{1p} = x,\ I_{2p} = 2x,\ I_{3p} = 0,$$
$$I_{1n} = 2\alpha,\ I_{2n} = 2x + \alpha,\ I_{3n} = 2\alpha. \tag{2}$$

The I/O characteristics for $x = 3.6$ µA and $\alpha = 0.5$ µA as an example is shown in Fig. 3(b).

The reference current sources are implemented with resistors whose thermal noise can increase the randomness and uncertainty of the system.

Another common method to generate discrete chaotic maps is based on the current-mode affine interpolation of the breakpoints [36]. This can be implemented by the affine interpolation of the value at the breakpoints [36]. In this method, any affine partition of the map is implemented using one elementary block based on the detection of the breakpoints. Therefore, it is straight-forward to find out whether the output value is within a certain affine partition of the map. In order to obtain the open-loop characteristic function of the circuit, the output of the circuit was connected to a matched load with the same impedance as the input impedance of the circuit. The simulations were carried out with CADENCE in IBM 0.13 µm technology. In Fig. 3(b), the I/O characteristic of the first circuit obtained by simulation is shown. x and α are two parameters in the map, highlighted in Fig. 3(b). ±2x is the required distance for the map. α is the extra length in the positive and negative side of the map. $\alpha = 0$ can cause a trap point at zero,

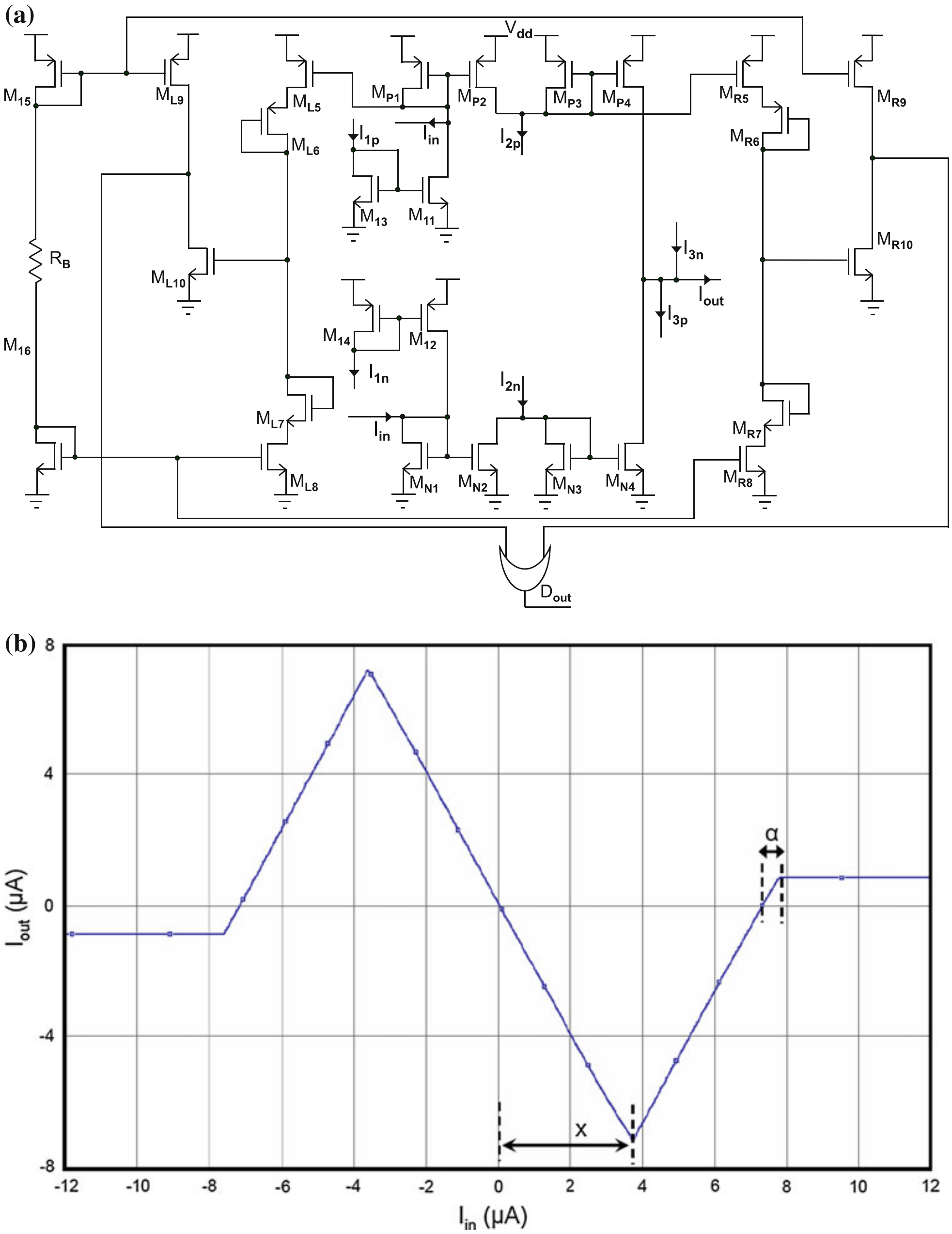

Fig. 3 **a** The double breakpoint current mirrors for the generation of the I/O current characteristic of the zigzag map. **b** The I/O current characteristic of the implemented circuit with the parameters ($x = 3.6$ µA and $\alpha = 0.5$ µA)

while considering non zero α can remove this state from the system and the possibility of trapping in the zero can be zero. The causes of slight changes in the simulation results will be taken into account in the next section. We performed HSPICE simulations in IBM 0.13 μm technology for the second circuit. The resulting I/O current characteristics of the circuit is shown in Fig. 4. The small variations of the map compared to the ideal map will be explained in the next section.

Four of the above maps are pipelined in a feedback loop. The noise floor of the analog circuits will set the initial state of the system that results in the non-predictive behavior of the system after a few first outputs. The inherent noise of the system is really small. However, first m runs of the system amplify the noise to a detectable range. If the gain of the open loop system is A, we should discard first m bit of the output stream given by $m = \log_A(P_0/N)$, where P_0 and N are the detecting power and noise power, respectively. A single stage of the system has a gain 2 and the whole system has a gain of 16. The number of bits that need to be discarded from the beginning of the output sequence will be about 20.

By investigating the transient response of both circuits with an input current pulse, we demonstrated that at the worst case the output would settle within 1 % of its final value in less than 12.5 and 15 ns for switched current circuit and interpolation of the values at breakpoints circuit, respectively.

4 Modeling the impact of implementation variations

This section is divided into two parts. In Subsect. 4.1, we derive the impact of the variations of the physical parameters of the presented circuit as non-idealities of the chaotic map. In Sect. 4.2, we investigate the transition matrix of the underlying Markov process in the presence of variations in the parameters of the chaotic map. We also quantify the bias and correlation in the output bit sequence.

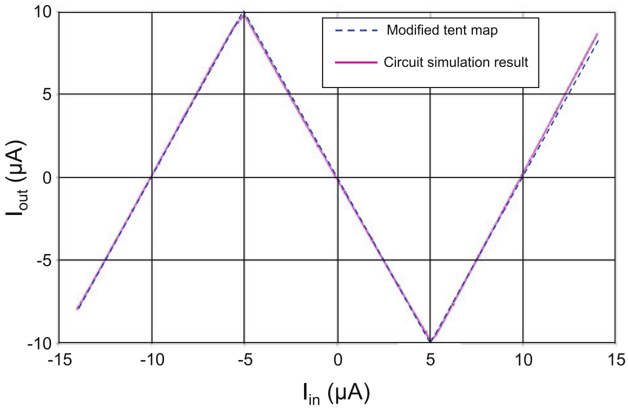

Fig. 4 The expected zigzag map (*dashed line*) and resulting I/O current characteristic of the zigzag map of the implemented circuit (*solid line*) based on the current-mode affine interpolation of the breakpoints with the parameters ($x = 3.6$ μA and $\alpha = 0.5$ μA)

4.1 The main causes of non-ideality in the presented circuit

The circuit is based on the switched current technique. Hence, the process, voltage and temperature (PVT) variations mainly impact two different parts of the circuit, namely the sources of currents that are the current mirrors and the current limiter stages.

In a current mirror, the current I that is supposed to mirror I_s, source current, is designed using the ratio of the width of transistors W_2/W_1. The current in a MOS transistor in saturation region is given by

$$I = \frac{1}{2}\mu C_{ox}\frac{W}{L}(V_{GS} - V_{th})^2(1 + \lambda V_{DS}). \tag{3}$$

This suggests that

$$\frac{I_2}{I_1} = \frac{W_2}{W_1}\frac{L_1}{L_2}\frac{1 + \lambda_2 V_{DS2}}{1 + \lambda_1 V_{DS1}}. \tag{4}$$

Since we are not designing in deep sub micron processes, we can neglect the effect of variations in λ. We can approximate the variation in different process and V_{DS} provided that the changes compared to the parameters are negligible. Hence,

$$\frac{I_2}{I_1} \approx \frac{W_2}{W_1}\left(1 + 2|\Delta W| + 2|\Delta L| + 4\frac{|\Delta V_{th}|}{V_{gs} - V_{th}} + 2\frac{\lambda|\Delta V_{DS}|}{1 + \lambda V_{DS}}\right), \tag{5}$$

where $\Delta W, \Delta L, \Delta V_{th}$, and ΔV_{DS} are the variation in width, length, threshold voltage, and the drain-source voltage of the transistors, respectively. If the V_{DS} changes are large, we should compensate that by further tuning of the width of the transistors. We need to investigate the worst-case variations. The impact of variations in V_{DS} and V_{th} is in the range of 1% which can be ignored compared to other parameters [37, 38]. Thus, the inner die variations for width of two different transistors in the current mirror will play the dominant role in altering the gain of the current mirror topology. The standard deviation of the variations are around 5 % [38–41].

The current limiter can not exactly provide the desired fixed current due to the output resistance of the transistors. This non-ideality results in a small variation in the slope in the I/O characteristic of the circuit, which is proportional to λ and the bias current of the circuit. While we can reduce this effect by further tuning of the width of transistors, the variation is inevitable.

4.2 Investigating the effect of map non-idealities on bias and correlation of output data sequence

We first consider a map with symmetric slope variation as in Fig. 5(a, b). We later extend to the non-symmetric case when we study the impact of post-processing. As discussed earlier, the ideal zigzag map has the same functionality as the tent map except for the alternations of the output between positive and negative values. Due to the symmetry in the map function and the bit generation process, the non-ideal maps can be analyzed using the maps shown in Fig. 5(c, d). Since the latter are easier to work with, we use them for the rest of the derivations. Note that without loss of generality, every chaotic map can be scaled to $(-1, 1)$. We assume the slope (gain) in the rising part of the map to be $2(1 + \delta_{g1})$ and the gain in the falling part to be $-2(1 + \delta_{g2})$. The non-ideal map function $N(x) : (0, 1) \to (0, 1)$ is thus given by

$$N(x) = \begin{cases} 2(1 + \delta_{g1})x & x < x_b \\ 1 - 2(1 + \delta_{g2})(x - x_b) & x > x_b \end{cases}. \tag{6}$$

where x_b is abscissa of the boundary between rising and falling sections and is

$$x_b = \frac{1}{2(1 + \delta_{g1})}. \tag{7}$$

Note that our analysis of the symmetric non-ideal map is also valid for the non-ideal tent map. In this map, bit 0 or 1 is generated if $x \in (0, x_b)$ or $x \in (x_b, 1)$, respectively. Throughout our analysis, we assume that variations are small, and we use first-order approximations for δ_{g1} and δ_{g2}. This assumption is valid according to our analysis of the process variations in the current mirrors. Accordingly, x_b can be easily shown to be

$$x_b \approx \frac{1}{2}(1 - \delta_{g1}). \tag{8}$$

Let δ_O be the ordinate of the end point of the map with abscissa $x = 1$, which is given by

$$\delta_0 = -(\delta_{g1} + \delta_{g2}). \tag{9}$$

The impact of x_b and δ_0 is illustrated in Fig. 6(a). The analysis of the two cases of $\delta_O > 0$ and $\delta_O < 0$ is slightly different. First, consider Fig. 5(c), i.e., $\delta_O > 0$. We want to find a good approximation for the asymptotic probability distribution $f(x)$ in the map. We approximate $f(x)$ using a four step distribution, as shown in Fig. 5(e). Let f_u denotes the approximate probability density at $x = 1$. According to the Frobenius–Perron operator, the distribution in $(0, \delta_O)$ can be given by

$$f_0 = \frac{1}{2} f_0 = 0. \tag{10}$$

We can calculate the distribution in $(\delta_O, 2\delta_O)$, f_1, as

$$f_1 = \frac{1}{2} f_0 + \frac{1}{2} f_u = \frac{1}{2} f_u. \tag{11}$$

Finally, the probability distribution for $(2\delta_O, 4\delta_O)$, f_2, can also be calculated using the same approach to be

$$f_2 = \frac{1}{2} f_1 + \frac{1}{2} f_u = \frac{3}{4} f_u. \tag{12}$$

The probability distribution in $(4\delta_O, 1)$ is approximated using f_O. Now, f_O can be calculated by noting that the probability distribution integrates to unity, i.e., $\int f(x)dx = 1$. Thus,

$$f_u = \frac{1}{1 - 2\delta_0} \approx 1 + 2\delta_0. \tag{13}$$

The results of simulation are provided in Fig. 7(a–c) for different values of δ_O. We can see that our approximate model captures the impact of the variations even for fairly large values of δ_O.

The case of Fig. 5(d), i.e., $\delta_O < 0$, can be analyzed similarly. The four step density distribution is shown in Fig. 5(f). We derive the distribution in $(0, |\delta_O|)$ as

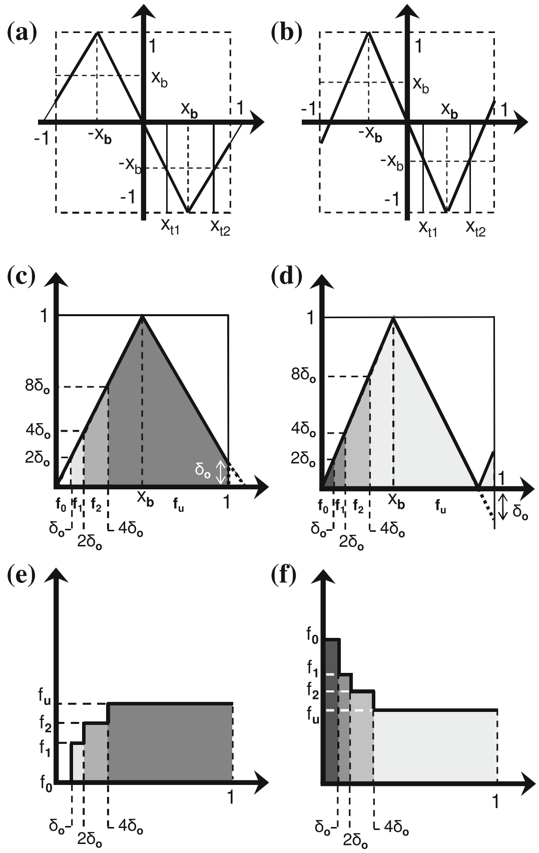

Fig. 5 The zigzag map with a fixed offset. The slope variations for **a** negative and **b** positive changes. The equivalent tent map for the (**a**) and (**b**) are shown in (**c**) and (**d**), respectively. The probability density function as a function of the chaotic map parameter for (**c**) and (**d**) is depicted in (**e**) and (**f**), respectively

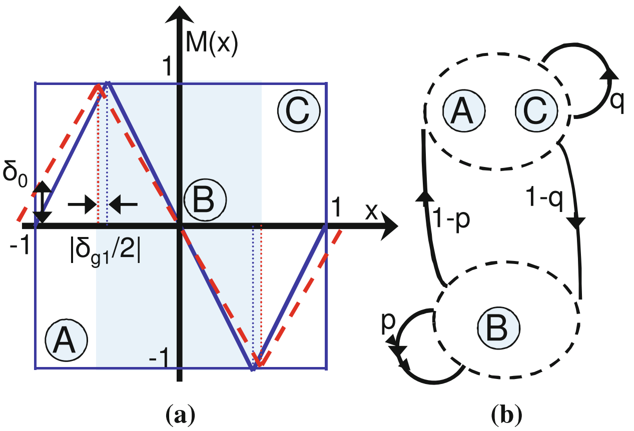

Fig. 6 **a** The impact of variations on the zigzag map characteristics. **b** The states of the Markov chain for truly random number generation

$$f_0 = \frac{1}{2} f_0 + \frac{1}{2} f_u + \frac{1}{2} f_u = 2 f_u. \tag{14}$$

The probability distribution in (|δ_0|, 2|δ_0|), i.e., f_0, the probability distribution in (2|δ_0|, 4|δ_0|), i.e., f_1, and the distribution in (|4δ_0|, 1), i.e., f_2, can be calculated in a similar way. Therefore, $f_1 = \frac{3}{2} f_u$ and $f_2 = \frac{5}{4} f_u$. Finally, we have

$$f_u = \frac{1}{2(1-\delta_0)} \approx 1 + 2\delta_0. \tag{15}$$

In the case of $\delta_0 < 0$, we have $f_u < 1$, i.e., the value of $f(x)$ for $x \approx 1$ is less than unity. Again, the simulation results presented in Fig. 7(d–f) show good agreement between the presented model and simulation results.

We model the bit generation process using a Markov chain as demonstrated in Fig. 6(b). The transition probabilities of the map can be derived by integration of the probability distribution of the map to form the transition matrix of the Markov chain as

$$\begin{aligned} P(0|0) &= \frac{\int_0^{x_{t1}} f(x)dx}{\int_0^{x_b} f(x)dx}, P(1|0) = \frac{\int_{x_{t1}}^{x_b} f(x)dx}{\int_0^{x_b} f(x)dx}, \\ P(0|1) &= \frac{\int_{x_{t2}}^{1} f(x)dx}{\int_{x_b}^{1} f(x)dx}, P(1|1) = \frac{\int_{x_b}^{x_{t2}} f(x)dx}{\int_{x_b}^{1} f(x)dx}, \end{aligned} \tag{16}$$

where x_{t1} and x_{t2} are the points whose ordinates are equal to x_b and are shown in Fig. 5(a, b).

We use our approximate model for the evaluation of the integrals. First, we derive p and q. i.e., $P(0|0)$ and $P(1|1)$. For fairly small variations (|δ_0| < 1/16), the three regions f_0, f_1 and f_2 lie in (0, x_{t1}). This is indeed the region of interest for our purposes. We derive the transition probabilities as functions of δ_{g1} and δ_{g2}. p and q are given by

$$\begin{aligned} p &= P(0|0) = \frac{1}{2} + \frac{3}{2}\delta_{g1} + 2\delta_{g2}, \\ q &= P(1|1) = \frac{1}{2} - \frac{1}{2}\delta_{g2}. \end{aligned} \tag{17}$$

Thus far, we derived the transition probabilities of the underlying Markov chain in (16). Next, we demonstrate that these can actually be used to derive the bias and correlation in the output bit sequence of the truly random number generator [42]. It is straightforward to show that the bias is

$$b = \left|\frac{1}{2} - P(0)\right| = \left|\frac{1}{2} - P(1)\right| = \frac{|p-q|}{2-p-q}. \tag{18}$$

Further, $\lambda_1 = |p + q - 1|$ is the second eigenvalue of the transition Matrix. It can be shown that the correlation

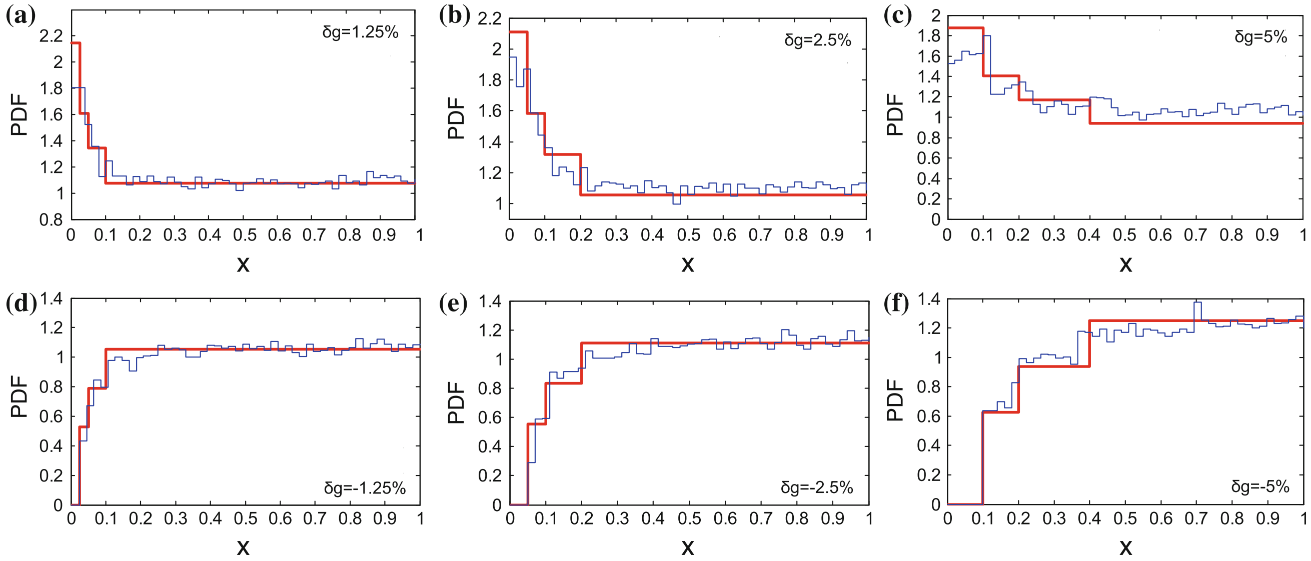

Fig. 7 The probability distribution function of the non-ideal system for different slope variations of **a** +1.25 %, **b** +2.5 %, **c** +5 %, **d** −1.25 %, **e** −2.5 %, and **f** −5 %. The *thick solid line* demonstrates the presented model and the *thin solid line* represents the simulation results

between the bits in the output sequence decays exponentially with their distance as 2^{-cn}, where $c = -\log_2(\lambda_1)$. Note that if $\delta_{g1} = \delta_{g2} = 0$, we will have $p = q = \frac{1}{2}$ and thus there is no bias ($b = 0$) and no correlation ($c = \infty$) in the output sequence.

5 Post processing

We performed an extensive set of Monte Carlo simulations with HSPICE to determine the impact of process variations on the slope of I/O characteristic curve. The absolute value of the slope, characterized by m, should be 2 for the presented chaotic map. Monte Carlo simulation technique uses different randomly chosen values for the transistor parameters with a defined Gaussian distribution. We assume that the transistor parameter values are normally distributed and uncorrelated with a standard deviation of 2 %, which is within the range of predicted values for current and future process technologies. Monte Carlo simulations demonstrate that the second circuit is less sensitive to process variations; therefore we use this circuit for the rest of this paper. By inspecting the variation of different linear pieces of the I/O characteristic curve as the objective of this Monte Carlo simulation, we investigate that the variation in slopes are less than 4 %. To be more accurate, the standard deviation of the slopes are less than 2, 4, and 1 % for the first, second, and third linear pieces, respectively.

Due to imperfections in practice, the existence of correlation and bias is inevitable that can be removed with post-processing unit. Von-Neumann, SHA-1, and SHA-2 post-processing algorithms are the most widely algorithms in practice and have proved to be effective in removal of the bias and correlation [21, 43, 44]. In Von-Neumann algorithm, the sequence is divided into the blocks of length two. Every 01 and 10 are considered 0 and 1, respectively, while 00 and 11 are discarded. This results in a variable bit-rate that is not desired. SHA-1 and SHA-2 algorithms are proper for removing the correlation and bias from the bit streams. Although these methods are effective, the implementation requires FPGA or complex hardware configurations. The sequences of data should pass through the post-processing stage mainly because of inherent non-idealitiy factors in process or circuit.

Our post-processing module is based on the method proposed by Addabbo et. al. [45]. The method relies on two main facts. First, bits that are located further apart from each other have less correlation. The correlation approaches 0 as the distance of the bits approaches infinity. The other fact is that XOR of two *independent* biased sequence will result in a sequence with less bias. See [46] for details.

The debiasing block, shown in Fig. 8, consists of a shift register of length l and an XOR gate. $z_{n\text{-}l}$, which is the output of time $n - l$ is XORed with $d_{in}(n)$ which is the input of time n. l must be large enough so that z_n and z_{n+l} are almost uncorrelated. In practice, l can be efficiently chosen by using the estimations made in the last sections. This has been proved to remove the bias of the output sequence in steady state. The decorrelation block is used to remove the correlation of the bits with a distance of kl. Note that we take the number of stages of shift register l and the number of stages of the map m to be prime with respect to each other, $(m, l) = 1$. This will reduce greatly the correlation between bits of distance kl, where k is not a multiplicand of l. The presented TRNG is robust to process and circuit variations, which provides simple post-processing stage compared to other chaotic maps used in TRNGs.

The output bit streams of the ideal and non-ideal I/O characteristics are tested in NIST 800-22 test suite and the results are presented in Table 1. As listed in the table, before post-processing, the generated bits fail some of the tests but after passing through post-processing unit the generated bits pass all the tests.

In the generation of the random bits, we consider the worst case scenario, which consists of four stages with similar I_{out}/I_{in} characteristics. In practice, these four stages are not exactly the same, therefore, the mismatch can enhance the quality of random output sequence. Utilization of four stages can also increase the total bit rate by

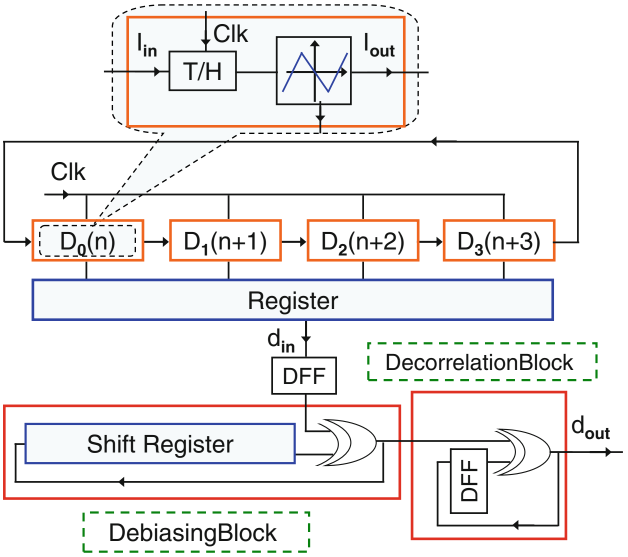

Fig. 8 The overall implementation of the presented TRNG with four pipelined chaotic maps. The post processing unit can remove the correlation and bias of the bit shuffled streams with decorrelation and debiasing blocks, respectively

Table 1 NIST-800-22 test suite results for the non-ideal zigzag map before and after applying the post-processing unit

NIST-800-22 test suite	Nonideal map w post-processing	Nonideal map w/o post-processing
Apen	0.513673	0.431503
Block frequency	0.410214	0.94278
Cumulative sums	0.80785	0.095407
Cumulative sums	0.975089	0.03606
FFT	0.871131	0.721176
Frequency	0.575479	0.047704
Linear complexity	0.082259	0.017841
Longest run	0.024845	0.553323
Non-periodic templates	100 % success	99.32 % success
Overlapping templates	0.736301	0.510453
Rank	0.98771	0.445713
Runs	0.044596	0.689421
Serial	0.443073	Failure
Serial	0.548205	Failure
Bias	0.04 %	0.50 %

multiplying the internal clock (track and hold clock) of the system by the number of cascading stages. The output bits of all stages go to a 4 bit register, known as bit shuffling [47]. The uniformity of the output bit stream is more than the inputs. The output of the register feed to the post processing unit.

The presented chaotic map can generate random bits, which have lower sensitivity to the circuit parameter fluctuations caused by process variation or circuit element mismatches. In addition to low power consumption (1.42 mW), the presented circuit can operate in clock frequencies as high as 10 MHz with the overall bit rate of 40 Mbps. To calculate the worst case power consumption of the circuit, we calculate the average power consumption of the circuit with different input currents. Due to randomness and chaotic behavior of the presented chaotic map, the probability of the system to be in any point in the steady state is the same.

6 Conclusion

In this paper, we presented a TRNG based on a novel discrete-time chaotic map, namely the zigzag map, that is capable of being used in high speed embedded cryptographic systems. The proposed map proved to be chaotic and has been implemented by switched current technology. The random bits are generated due to the transition among different points of the chaotic map. Pipelining four stages in a positive feedback loop, we can amplify the inherent noise of the system within a detectable range, which results in random number generation. A modeling methodology has been developed to predict the performance of the system in the presence of non-idealities. Non-ideality sources are modeled and their effect on the performance of the output binary sequence (correlation and bias) has been studied. The non-ideal binary stream has been fed to a post processing module to increase the quality of the random digits. The presented system has been demonstrated to generate random bits that pass the NIST 800-22 test suite.

Acknowledgments This work was done in part when the authors were affiliated with Rice University. The authors would like to thank Professor Yehia Massoud for the valuable discussions, comments and suggestions that helped to improve the quality of this paper.

References

1. Wiggins, S. (1990). *Introduction to applied nonlinear dynamical systems and chaos*. Berlin: Springer-Verlag.
2. Kaneko, K., & Tsuda, I. (2001). *Complex systems: Chaos and beyond: A constructive approach with applications in life science*. Berlin: Springer.
3. Nejati, H., Ragheb, T., Hosseini, A., & Massoud, Y. (2007). A programmable input-pulse dependent chaotic oscillator. In *Proceedings of IEEE MWSCAS* (pp. 173–176).
4. Nejati, H., Ragheb, T., & Massoud, Y. (2008) On the feasibility of bandwidth tuning in cascaded non-autonomous chaotic oscillators. In *Proceedings of IEEE MWSCAS* (pp. 902–905).
5. Menezes, A., Oorschot, P., & Vanstone, S. (1997). *Handbook of applied cryptography*. Boca Raton, FL : CRC Press Series on Discrete Mathematics and its Applications.
6. Callegari, S., Rovatti, R., & Setti, G. (2005). Embeddable adc-based true random number generator for cryptographic applications exploiting nonlinear signal processing and chaos. *IEEE Transaction on Signal Processing, 53*(2), 793–805.
7. Cover, T. M., & Thomas, J. A. (2006). *Elements of information theory*. New York: Wiley.
8. Hu, Y., Liao, X., Wong, K. -W., & Zhou, Q. (2009). A true random number generator based on mouse movement and chaotic cryptography. *Chaos, Solitons and Fractals, 40*(5), 2286–2293.
9. Ragheb, T., Laska, J., Nejati, H., Kirolos, S., Baraniuk, R., & Massoud, Y. (2008). A prototype hardware for random demodulation based compressive analog-to-digital conversion. In *Proceedings of IEEE MWSCAS* (pp. 37–40).
10. Pfetsch S., Ragheb T., Laska J., Nejati H., Gilbert A., Strauss M., Baraniuk R., & Massoud Y. (2008). On the feasibility of hardware implementation of sub-nyquist random-sampling based analog-to-information conversion. In *Proceedings of the IEEE ISCAS* (pp. 1480–1483).
11. Ragheb T., Nejati H., Nieuwoudt A., & Massoud Y. (2006). Parasitic-aware analytical modeling of fully integrated switchable narrow-band cmos low noise amplifiers. In *Proceedings of IEEE WAMICON* (pp. 1–5).
12. Nejati, H., Ragheb, T., Nieuwoudt, A., & Massoud, Y. (2007). Analytical modeling methodology for ultra wideband low noise amplifiers with generalized filter-based impedance matching. *ALOG, 51*(2), 121–127.
13. Nejati, H., Ragheb, T., Nieuwoudt, A., & Massoud, Y. (2007). Modeling and design of ultrawideband low noise amplifiers with generalized impedance matching networks. In *Proceedings of the IEEE ISCAS* (pp. 2622–2625).
14. Nejati, H., Ragheb, T., & Massoud, Y. (2008). On the design of customizable low-voltage common-gate lna-mixer pair using

current and charge reusing techniques. In *Proceedings of the ACM GLSVLSI* (pp. 195–200).
15. Nejati, H., Ragheb, T., & Massoud, Y. (2008). Analytical modeling of common-gate low noise amplifiers. In *Proceedings of the IEEE ISCAS* (pp. 888–891).
16. Nieuwoudt, A., Ragheb, T., Nejati, H., & Massoud, Y. (2009). Numerical design optimization methodology for wideband and multi-band inductively degenerated cascode cmos low noise amplifiers. *IEEE Transactions on Circuits and Systems I, 56*(6), 1088–1101.
17. Petrie, C. S., & Connelly, J. A. (2000). A noise-based ic random number generator for applications in cryptography. *IEEE Transactions on Circuits and Systems I, 47*(5), 615–621.
18. Jun, B., & Kocher, P. (1999) The intel random number generator. *Technical Report Cryptography Research Inc. [online]* http://www.cryptography.com/resources/whitepapers.
19. Evaluation of VIA C3 Nehemiah random number generator. *Technical Report Cryptography Research Inc. [online]* http://www.cryptography.com/resources/whitepapers. (2003).
20. Liu, C., & McNeill, J. A. (2005). A digital-PLL-based true random number generator. In *IEEE PhD research in microelectronics and electronics* (pp. 113–116).
21. Yalcin, M. E., Suykens, J. A. K., & Vandewalle, J. (2004). True random bit generation from a double-scroll attractor. *IEEE Transactions on Circuits and Systems I, 51*(7), 1395–1404.
22. Bean, J., & Langlois P. J. (1994). A current mode analog circuit for tent maps using piecewise linear functions. In *IEEE ISCAS* (pp. 125–128).
23. Neumann, J. V. (1951). Various techniques used in connection with random digits. *Applied Mathematics Series, 12*, 36–38.
24. SHA-1 Standard, National Institute of Standards and Technology (NIST), Secure Hash Standard. (1995). FIPS PUB 180-1. http://www.itl.nist.gov/fipspubs/fip180-1.htm.
25. SHA-2 Standard, National Institute of Standards and Technology (NIST), Secure Hash Standard. (2002). FIPS PUB 180-2. http://csrc.nist.gov/publications/fips/fips180-2/fips180-2.pdf.
26. Beirami, A., & Fekri F. (2011) Results on the redundancy of universal compression for finite-length sequences. In *Proceedings of IEEE International Symposium on Information Theory* (pp. 1604–1608).
27. Beirami, A., Sardari, M., & Fekri, F. (2012). Results on the fundamental gain of memory-assisted universal source coding. In *Proceedings of IEEE International Symposium on Information Theory* (pp. 1092–1096).
28. Beirami, A., & Fekri, F. (2012). On lossless universal compression of distributed identical sources. In *Proceedings of IEEE International Symposium on Information Theory* (pp. 566–570).
29. A statistical test suite for random and pseudorandom number generators for cryptographic applications. *Natioanal Institute for Standards and Technology,* Special publication 800-22 (2001).
30. Nejati H., Beirami A., & Massoud Y. (2008). A realizable modified tent map for true random number generation. In *Circuits and systems, 2008. MWSCAS 2008. 51st midwest symposium* (pp. 621–624).
31. Li, D., Jin, Y., Shen, H., & Yan, X. (2006). Design of random number generation algorithm. In *IEEE ICCIAS* (pp. 1287–1290).
32. Morie, T., Sakabayashi, S., Nagata, M., & Iwata, A. (2000). Cmos circuits generating arbitrary chaos by using pulsewidth modulation techniques. *IEEE Transactions on Circuits and Systems I, 47*(11), 1652–1657.
33. Callegari, S., Setti, G., & Langlois, P. J. (1997). A CMOS tailed tent map for the generation of uniformly distributed chaotic sequences. In *IEEE ISCAS* (pp. 781–784).
34. Yuan, F. (2007). *CMOS current-mode circuits for data communications*. New York: Springer.
35. Wilamowski, B., Ferre-Pikal, E., & Kaynak, O. (2000). Low power, current mode cmos circuits for synthesis of arbitrary nonlinear functions. In *9th NASA symposium on VLSI design* (pp. 7.3.1–7.3.8).
36. Rovatti, R., Manaresi, N., Setti, G., & Franchi, E. (1999). A current-mode circuit implementing chaotic continuous piecewise-affine markov maps. In *Proceedings of the seventh international conference on microelectronics for neural, fuzzy and bio-inspired systems* (pp. 275–282).
37. Fiez, T., Liang, G., & Allstot, D. (1991). Switched-current circuit design issues. *IEEE Journal on Solid State Circuits, 26*(3), 192–202.
38. International Technology Roadmap for Semiconductors (ITRS). http://public.itrs.net. (2009).
39. Nassif S. (2001) Modeling and analysis of manufacturing variations. In *IEEE CICC* (pp. 223–228).
40. Nieuwoudt, A., Ragheb, T., Nejati, H., & Massoud, Y. (2007). Increasing manufacturing yield for wideband rf cmos lnas in the presence of process variations. In *Proceedings of the IEEE ISQED* (pp. 801–806).
41. Nieuwoudt, A., Ragheb, T., Nejati, H., & Massoud, Y. (2009). Variation tolerant design methods for wideband low noise amplifiers. *ALOG MSL, 58*, 49–54.
42. Beirami, A., Nejati, H., & Massoud, Y. (2008). A performance metric for discrete-time chaos-based truly random number generators. In *Proceedings of IEEE MWSCAS* (pp. 37–40).
43. Zhou, T., Yu, M., & Ye, Y. (2006). A robust high-speed chaos-based truly random number generator for embedded cryptosystems. In *IEEE MWSCAS* (pp. 536–540).
44. Wang, Y. -H., Zhang, H. -G., Shen, Z. -D., & Li, K. -S. (2005). Thermal noise random number generator based on SHA-2 (512). In *IEEE ICMLC* (pp. 3970–3974).
45. Addabbo, T., Alioto, M., Fort, A., Rocchi, S., & Vignoli, V. (2006). Efficient post-processing module for a chaos-based random bit generator. In *IEEE ICECS* (pp. 1224–1227).
46. Sunar, B., Martin, W. J., & Stinson, D. R. (2007). A provably secure true random number generator with built-in tolerance to active attacks. *IIEEE Transactions on Computers, 56*(1), 109–119.
47. Gerosa, A., Bernardini, R., & Pietri, S. (2002). A fully integrated chaotic system for the generation of truly random numbers. *IEEE Transactions on Circuits and Systems I, 49*(7), 993–1000.

Hamid Nejati is currently a Ph.D. student in radiation laboratory at the Electrical Engineering and Computer Science department, University of Michigan, Ann Arbor, MI. He received his B.Sc. and M.Sc. (with honors) in electrical engineering from Sharif University of Technology, Tehran, Iran in 2004 and 2006, respectively. Hamid's research interests include radar remote sensing, analog synthesis, modeling and design of RF/mixed signal circuits and systems, and nanophotonic devices.

Ahmad Beirami received his B.Sc. in Electrical Engineering from Sharif University of Technology in 2007. Since 2009, he has been pursuing his Ph.D. in Electrical and Computer Engineering at Georgia Institute of Technology. Ahmad's research interests broadly include information theory and coding, theoretical statistics, plasmonic waveguides, and probabilistic modeling and analysis.

Warsame H. Ali received the B.Sc. degree in Electrical Engineering from King Saud University of Riyadh, Saudi Arabia, in 1986, M.Sc. from Prairie View A&M University, Prairie View, Texas, in 1988, and Ph.D. degree in Electrical Engineering from University of Houston, Houston, Texas, in 2004. Presently, Dr. Ali is an associate professor of Electrical and Computer Engineering (ECE) Department, Roy G. Perry, College of Engineering at Prairie View A&M University (PVAMU). Dr. Ali teaches graduate and undergraduate courses in the Department of ECE. He is the Program Coordinator of the Electrical Engineering Undergraduate program. He also provides services to different agencies such as National Science Foundation (NSF) to evaluate proposals. Dr. Ali's research field includes the Renewable Energy, Power Systems, Power Electronics and Motor Drives, Controls, Application of Digital PID Controllers, Digital Methods to Electrical Measurements, and Mixed Signals Testing Techniques.