2008 International Symposium on Nonlinear Theory and its Applications
NOLTA'08, Budapest, Hungary, September 7-10, 2008

Stochastic Modeling and Verification of a 0.35 μm CMOS Chaos-based True Random Number Generator

Fabio Pareschi[†], Riccardo Rovatti[‡], and Gianluca Setti[†]

[†] ENDIF, University of Ferarra – Via Saragat, 1 - 44100 Ferrara, ITALY
[‡] DEIS, University of Bologna – Viale risorgimento, 2 - 40133 Bologna, ITALY
All authors are also with ARCES - University of Bologna – Via Toffano, 2 - 40125 Bologna, ITALY
Email: fabio.pareschi@unife.it, rrovatti@arces.unibo.it, gianluca.setti@unife.it

Abstract—In this paper we present a high-level stochastic model for a true random number generator designed in 0.35 μm CMOS technology, which internally exploits a pipeline analog-to-digital converter modified to operate as a set of chaotic maps. The model is tuned on Monte Carlo circuit-level simulations to include the non-idealities of the designed circuit in the chaotic map model. The parameters of the model are then verified with a comparison between results of NIST statistical tests of the output streams given both by the stochastic model and by the implemented circuit. Once properly tuned, such model actually defines a macro-block which can be exploited for fast generation of a random bit stream necessary in the simulation of others circuits/systems.

1. Introduction

By definition, a Random Number Generator (RNG) is a circuit capable of producing perfectly *unpredictable* bits. RNGs represent a fundamental issue in many engineering tasks; for instance they are are used in all cryptographic applications, where they are fundamental in the synthesis of confidential keys for both symmetric and public-key crypto-systems [1].

It is generally recognized that ideal random sources can only be approximated. An ideal random source is capable of producing infinitely long sequences made of perfectly *independent* bits, with the property that, when restarted, the source never reproduces a previously delivered sequence (*non-repeatability*).

Traditionally the most used sources of randomness rely on pseudo-random number generators (PRNGs). These are algorithms capable of "expanding" short seeds into long, irregular bit sequences. The procedure is completely repeatable and, in fact, non random at all; the only source of entropy in the process is actually represented by the seed.

The recent, huge growth of data-security applications is questioning the general application of PRNGs, since no cryptographic algorithm can be stronger than its underlying RNG [2]. For this reason, many true random number generators (TRNGs) as opposed to PRNGs are now being developed. Typically TRNGs are based on processes like Johnson thermal noise [3], shot noise, jitter in PLL or free oscillator [4], but can also be based on quantum effects like single photon reflection [5]. In particular with the latter implementation it is possible to achieve very good results in terms both of quality and speed; however this often do not represent a solution currently embeddable in silicon integrated technology.

The RNG we discuss in this paper belong to the class of True-random number generators, and it is based on a set of simple one-dimensional chaotic maps [6], following the approach recently proposed in [7]. There, a 1.5 bit/stage pipeline analog-to-digital converter (ADC) [8] has been reconfigured to implement a chaotic circuit that has been theoretically proved to generate *independent and identically distributed* – i.e. *random* – symbols. A prototype of this TRNG has been designed and fabricated in 0.35 μm 3.3 V CMOS technology [9].

In this paper we take into account a behavioral model developed for the designed TRNG and based on Monte Carlo simulations of the circuit, and compares it with results achieved by measurements on the chip prototype. Through this comparison, we can adjust the model parameters to fit the implemented circuit. With this adjustment, we can provide a realistic model including all the non-idealities of the designed circuit. This model can be used, for instance, in high-level simulations of a cryptographic system to predict the effect of non-idealities of the RNG on system security.

The paper is organized as follows. In section 2 we provide a brief description of the designed circuit. In section 3 we provide a behavioral, high-level model of the circuit developed on Monte Carlo simulation; while in section 4 we tune model parameters on measurements on circuit prototype. Finally, we draw some conclusions.

2. Circuit Description

The designed circuit is based on pipeline ADCs technology: the architecture of a standard pipeline ADC, including the modifications introduced in order to make the circuit working as a TRNG, is shown in Fig. 1. When working as an A/D converter, each i-th stage (excluding the final k-th that is usually incomplete) computes a *coarse* m-bit representation $D^{(i)}$ of its input $v^{(i)}$ (sampled at the beginning of the time step) and then calculates and rescales an analog *error conversion* $e^{(i)}$ to be passed (at the following time step) to the next stage $(i+1)$-th. Intriguing, in the architecture known as $1 + 1/2$ bit/stage the (normalized) relation between the input $v_n^{(i)}$ of the i-th stage at time step n and the input $v_{n+1}^{(i+1)}$ of the following $(i+1)$-th stage at the next time step $n+1$ is

$$v_{n+1}^{(i+1)} = M\left(v_n^{(i)}\right) \tag{1}$$

where the $M(\cdot)$ function is depicted in Fig. 2-(a) and it can be written as:

$$M(x) = \begin{cases} 2x+2 & \text{if } x \le -1/2 \\ 2x & \text{if } -1/2 \le x < 1/2 \\ 2x-2 & \text{if } x > 1/2 \end{cases} \tag{2}$$

If we take the generic i-th stage including the Sample/Hold, and close it into a loopback, i.e. if we set $v^{(i)} = v^{(i+1)}$, we get a system whose evolution is described by:

$$v_{n+1}^{(i)} = M\left(v_n^{(i)}\right) \tag{3}$$

i.e. we have a discrete-time 1D autonomous system usually known as *chaotic map*. Furthermore, the particular structure of $M(\cdot)$ makes this map a Piece-Wise Affine Markov (PWAM) map.

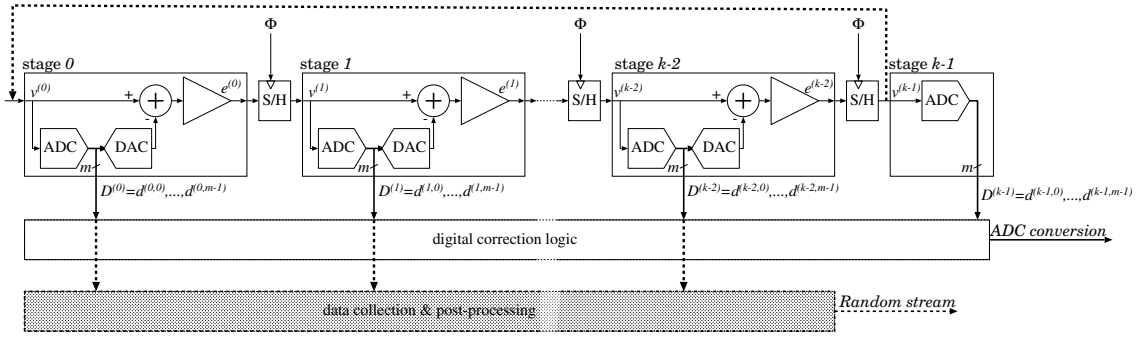


Figure 1: Basic structure of a pipeline ADC and modifications required *(dotted elements)* to obtain a TRNG.

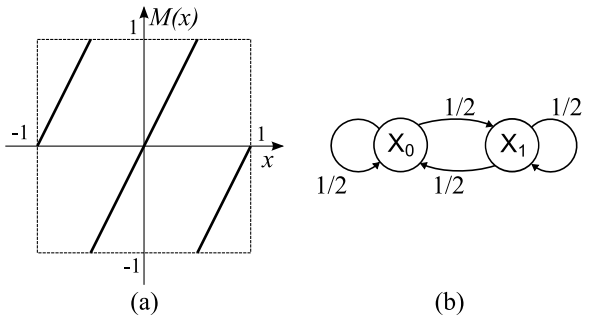


Figure 2: (a) PWAM map found in 1.5 bit/stage ADC pipeline converters; and (b) Markov chain describing the system evolution.

A deep and exhaustive analysis on chaotic maps and of PWAM maps can be found in [6]. Here it is enough to recall that the evolution of PWAM map systems can be studied with a Markov chain: when $M(\cdot)$ is the function in (2), and assuming the two states $\mathsf{X}_0 : v^{(i)} \in [-1, -1/2) \cup [1/2, 1]$ and $\mathsf{X}_1 : v^{(i)} \in (-1/2, 1/2]$ the Markov chain associated to system (3) is the chain of Fig. 2-B and it is exactly the chain describing a random number generator, more specifically a random *bit* generator. Note that the particular structure of the $1\frac{1}{2}$ bit stage allows us to evaluate if the system is in the state X_0 or X_1 simply looking at the digital conversion $D^{(i)}$ of the stage. This means that closing a single stage of a $1\frac{1}{2}$ A/D converter into a loopback and adding a simple digital logic it is enough to get a chaos-based TRNG.

Instead, if we close the loopback after the $k-1$ of complete stages (that is the modification proposed in Fig. 1), we have the a *pipeline* of chaotic map:

$$\begin{cases} v_{n+1}^{(0)} = M\left(v_n^{(k-2)}\right) \\ v_{n+1}^{(1)} = M\left(v_n^{(0)}\right) \\ \cdots \\ v_{n+1}^{(k-2)} = M\left(v_n^{(k-3)}\right) \end{cases} \tag{4}$$

It is easy to recognize that system (4) is equivalent to $k-1$ chaotic maps as (3) running in parallel, i.e. $k-1$ random bit generators running simultaneously. So, closing into a loopback the whole pipeline (except the last, incomplete stage) changes the A/D converter into a pipeline of chaotic maps, that also works as a random number generator when substituting the digital correction logic ot the A/D converter with a proper digital logic.

Following this, we designed a RNG in 0.35 μm technology. The chip, whose microphotograph can be found in Fig. 3, includes two pipelines, the first one composed by two stages, the second one by eight stages. The maximum working speed is 5 Mbit/s per stage; however the RNG achieves the best output stream quality at the speed of 1.5 Mbit/s per stage. A more detailed description of the circuit can be found in [9].

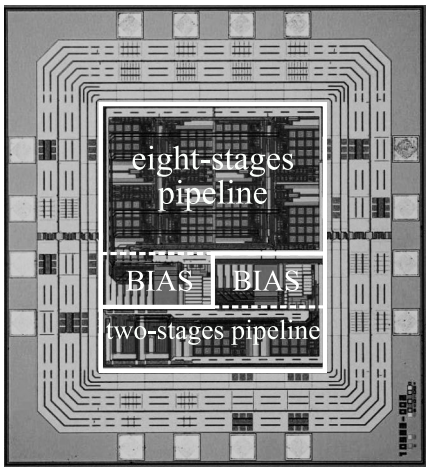


Figure 3: Microphotograph of the designed ADC-based RNG.

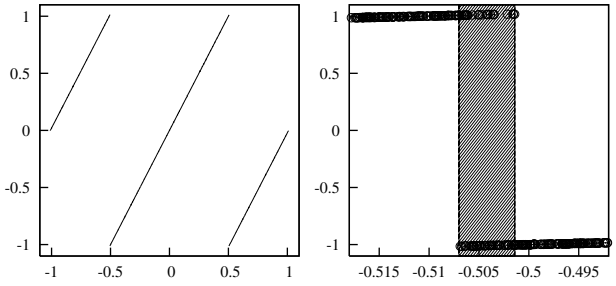
3. Behavioral model

From many Monte Carlo simulation runs of about 25×10^3 clock periods of the two-stages pipeline circuit a behavioral model of the circuit has been extracted and presented in [10]. The model has been designed upon the two sets of points (x_{k+1}, x_k), one for every stage, extracted from the Monte Carlo runs. With these sets, we tried to rebuild the *actual* $M(\cdot)$ function, i.e. the function including all non-idealities due to implementation errors. Furthermore, a different function has been rebuild to each stage, thus including into the model both the differences which may exist between stages of different pipelines or between stages of the same pipeline.

The switched capacitor implementation and the fully differential architecture ensure a very high linearity and a high precision. An example of the M function extracted by simulation is depicted in Fig. 4-(a) ; we modeled $M(\cdot)$ as

$$M(x) = \begin{cases} 1.9995x + \beta & \text{if condition } \lambda_1(x) \text{ is true} \\ 1.9995x & \text{if condition } \lambda_2(x) \text{ is true} \\ 1.9995x - \beta & \text{if condition } \lambda_3(x) \text{ is true} \end{cases} \tag{5}$$

The determination of the three condition λ_1, λ_2 and λ_3 is non-trivial. Ideally, two points α^- and α^+ exist, with $\lambda_1(x) \equiv x < \alpha^-$,



(a) (b)

Figure 4: (a) A collection of (x_{k+1},x_k) points for a single map; and (b) zoom around point α^-.

parameter(s)	mean value	standard deviation
β	2.0376	0.09806
$\Delta\alpha$	0	0.004551
s^+,s^-	124.7316	60.177

Table 1: Expected value and standard deviation for model parameters.

$\lambda_2(x) \equiv \alpha^- \leq x < \alpha^+$ and $\lambda_3(x) \equiv x \geq \alpha^+$. Yet, the real behavior of the system around α^- is depicted in Fig. 4-(b) (the behavior near α^+ is very similar). While at a certain distance from the point the behavior is fully deterministic, a point very close to the α^- could sometimes verify condition λ_1 and sometimes λ_2 (the gray area in the figure). This behavior could be explained considering interferences (e.g spikes on the power supply voltage) from the other parts of the circuit. To include this, we have developed a *stochastic model*, i.e. standing the $M(\cdot)$ function in (5),the the three condition λ_1, λ_2 and λ_3 are described as a probability function of x:

$$M(x) = \begin{cases} 1.9995x + \beta & x < 0, \text{with probability } 1 - p(x) \\ 1.9995x & \text{with probability } p(x) \\ 1.9995x - \beta & x > 0, \text{with probability } 1 - p(x) \end{cases} \quad (6)$$

with

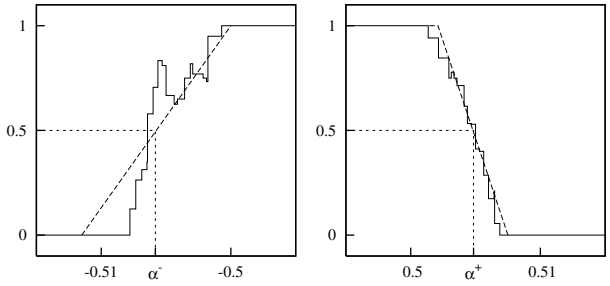
$$p(x) = \begin{cases} 0 & x < \alpha^- - 1/2s^- \\ 1/2 + (x - \alpha^-)\, s^- & \alpha^- - 1/2s^- \leq x < \alpha^- + 1/2s^- \\ 1 & \alpha^- + 1/2s^- \leq x < \alpha^- - 1/2s^+ \\ 1/2 - (x - \alpha^+)\, s^+ & \alpha^+ - 1/2s^+ \leq x < \alpha^+ + 1/2s^+ \\ 0 & x \geq \alpha^+ + 1/2s^+ \end{cases} \quad (7)$$

i.e. $p(x)$ is a trapezoidal function that can be seen in Fig. 5), along with the observed density of point satisfying condition λ_2.

In the light of numerical analysis of the Monte Carlo simulations, the parameters of the model are the following:

- β is assumed as a global parameter for the entire pipeline
- α^+ and α^- are parameter which different for each stage in the pipeline, and they are computed as $\alpha^+ = -\alpha^- = \beta/4 + \Delta\alpha$
- s^- and s^+ are independent parameters assumed different for each stage of the pipeline.

The four basic parameters β, $\Delta\alpha$, s^- and s^- have been assumed to be Gaussian random variables, whose mean value and statistical deviation are indicated in Table 1.



(a) (b)

Figure 5: (a) Density of points satisfying condition λ_2 (solid line) and its trapezoidal approximation $p(x)$ (dotted line) around α^-; and (b) around α^+.

4. Comparison between circuit model and circuit measurements

In this paper we compare results of the model introduced in the previous section with results of the eight-stages pipeline included in the designed prototype, and running at the optimum speed of 1.5 Mbit/s per stage, i.e. 12 Mbit/s. The comparison has been made in terms of comparison of the quality of the random bitstream generated both by the model and by the prototype. To allow a correct interpretation of the results, let us clarify two points. (a.) The developed model is a stochastic model; this means that it requires an internal random source to be applied. Results presented here are obtained using the Mersenne-Twister algorithm; switching to any other random source gives similar results. (b.) The comparison is only statistical, i.e. we compared the ratio of generated sequences capable to pass a random test. However, while each sequence has been generated by a difference instance of the model (i.e. with different, randomly drawn parameters), only a few prototypes have been tested. This limitation is due to the limited number of samples delivered by the foundry.

To evaluated the quality of the generated random bits we used the SP800-22 test suite from NIST [11]. This suite is composed by 14 independent tests to be applied to a stream of 10^6 bits. We have generated several 10^6 bits sequences (more precisely, 20,000 sequences) both with the developed model and with the prototype, and compared the ratio of sequences passing the NIST tests. Results are shown in Table 2, and show that the quality of the generated bitstream (that is very high, since a simple post processing like the XOR-2 is enough to let near all sequences pass all the tests in the suite) is slightly overestimated by the model. This is, of course, expected.

In order to match the model with the measurements from the prototype, we have to find new set of parameters, such that the quality of the stream generated by the model with them can be compared with the quality stream generated by the prototype. We can assume the new parameters with the same mean values as in Table 1, introducing a *degrading factor* Θ as a multiplicative factor for the standard deviation for all of them. In this way, the determination of the correct set of parameters is reduced to the computation of the degrading factor. Furthermore, to compare the quality of the two random streams, we have considered the two vectors A and B, whose i-th components are the ratio of sequences passing the i-th test in the suite, and the two quantities

$$d(A,B) = \frac{1}{n}\sqrt{\sum_{i=1}^{n}(A_i - B_i)^2} \quad (8)$$

SP800-22 test	XOR-2 post-proc model	XOR-2 post-proc results	No post-proc model	No post-proc results
Frequency	0.963800	0.948650	0.447000	0.000000
Block Frequency	0.966900	0.987850	0.885400	0.478800
Cumulative Sums	0.965000	0.951300	0.456100	0.000000
Runs	0.967000	0.987800	0.626000	0.000000
Longest Run of 1s	0.968600	0.989750	0.947300	0.988900
Matrix Rank	0.967800	0.988950	0.969300	0.987300
Spectral (DFT)	0.965900	0.988850	0.944300	0.936500
NOT Matching	0.968200	0.988550	0.949000	0.981300
OT Matching	0.968100	0.984900	0.910700	0.989900
Universal	0.966400	0.989000	0.942700	0.986000
Approx. Entropy	0.965600	0.987950	0.730000	0.014100
Random Excursion	0.987809	0.988390	0.991262	0.973585
Random Exc. Var.	0.988468	0.988756	0.988815	0.992453
Serial	0.967800	0.989350	0.961200	0.980900
Linear Complexity	0.968900	0.990150	0.967100	0.990400

Table 2: Ratio of NIST test passed for sequences generated both by the prototype and by the model, with a very simple XOR-2 post-processing and without any postprocessing.

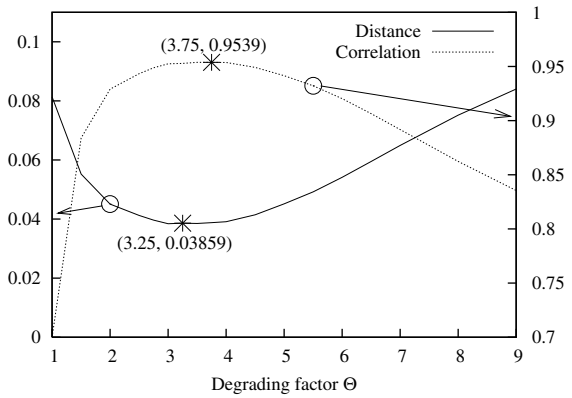


Figure 6: Comparison between the quality of the bitstream of the prototype and the model assuming a degrading factor Θ using the distance function (8) and the correlation function (9).

that represents the distance between the two vectors, and

$$c(A,B) = \frac{\sum_{i=1}^{n}(A_i - \mathrm{E}[A_i])(B_i - \mathrm{E}[B_i])}{\frac{1}{2}\sum_{i=1}^{n} A_i^2 + \frac{1}{2}\sum_{i=1}^{n} B_i^2} \qquad (9)$$

that represent the correlation between A and B. In (8) we have assumed $\mathrm{E}[A_i] = \mathrm{E}[B_i] = 0.99$ that is the expected ratio of sequences passing the test in the ideal case.

Results are shown in Fig. 6, where bot distance function (8) and correlation function (9) are shown for different values of Θ. The comparison has been made between unprocessed data (i.e. without any post=processing). The value of Θ for which the two bitstreams have the most similar quality is around $\Theta = 3.5$. This can be assumed as the value of the degrading ratio for which the model match the prototype.

5. Conclusions

In this paper we have compared results of randomness tests achieved by a prototype of a designed RNG with results achieved by a stochastic, high-level model of the same circuit obtained though circuital simulation. With this comparison we are able to fit the model parameters to the true-implemented circuit. In this way we can provide a more realistic model that can be used to simulate how the non-idealities of the proposed RNG can affect the system where it is employed.

References

[1] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1996.

[2] D. E. Eastlake, S. D. Crocker, and J. I. Shiller, "RFC 1750: Randomness recommendation for security" in *Internet Society Request for Comments*, Internet Engineering Task Force, December 1994.

[3] B. Jun and P. Kocher, "The Intel Random Number Generator", Crypt. Reasearch, Inc. white paper prepared by Cryptography Research, Inc. for Intel Corp., April 1999. Available at `http://www.cryptography.com/resources/whitepapers/IntelRNG.pdf`

[4] Cryptography Research, "Evaluation of VIA C3 Nehemiah Random Number Generator", white paper prepared by Cryptography Research, Inc., San Farncisco (USA). February 27, 2003. Available at `http://www.cryptography.com/resources/whitepapers/VIA_rng.pdf`

[5] idQuantique, "Random Numbers Generation using Quantum Physics" White paper, 2004. Available at `http://www.idquantique.com/products/files/quantis-whitepaper.pdf`

[6] G. Setti, G. Mazzini, R. Rovatti, and S. Callegari, "Statistical modeling of discrete time chaotic processes: Basic finite dimensional tools and applications", in *Proceedings of the IEEE*, special issue on "Applications of Nonlinear Dynamics to Electronic and Information Engineering", vol. 90, no. 5, pp. 662-690, May 2002.

[7] S. Callegari, R. Rovatti, and G. Setti, "Embeddable ADC-Based True Random Number Generator for Cryptographic Applications Exploiting Nonlinear Signal Processing and Chaos", in *IEEE Transaction on Signal Processing*, vol. 53, no. 2, pp. 793–805, February 2005.

[8] T. B. Cho, and P. R. Gray, "A 10 b, 20 Msample/s, 35mW Pipeline A/D Converter", in *IEEE Journal of Solid-State Circuits*, vol. 30, no. 3, pp. 166–172, March 1995.

[9] F. Pareschi, G. Setti and R. Rovatti, "A Fast Chaos-based True Random Number Generator for Cryptographic Applications," in *Proceedings of 26th European Solid-State circuit Conference (ESSCIRC2006)*, pp 130-133. Montreux, Switzerland, 19-21 September 2006.

[10] F. Pareschi, G. Setti, and R. Rovatti, "A macro-model for the efficient simulation of an ADC-based RNG", in *Proceedings of 2005 IEEE International Symposium on Circuits and Systems* (ISCAS2005), pp. 4349–4353. Kobe (Japan), May 23–26, 2005.

[11] National Institute for Standards and Technology (NIST), "A statistical test suite for random and pseudorandom number generators for cryptographic applications", Special publication 800-22, May 2001.