A macro-model for the efficient simulation of an ADC-based RNG

This is the author's version of the article that has been presented at IEEE ISCAS2005
The editorial version of the paper is available at http://dx.doi.org/10.1109/ISCAS.2005.1465594

A Macro-Model for the Efficient Simulation of an ADC-based RNG

Fabio Pareschi*, Gianluca Setti* and Riccardo Rovatti†
*DI - University of Ferrara, via Saragat 1, 44100 Ferrara - ITALY
†DEIS - University of Bologna, viale risorgimento 2, 40136 Bologna - ITALY
All authors are also with ARCES - University of Bologna, via Toffano 2/2, 40125 Bologna - ITALY
Email: {fpareschi,gsetti}@ing.unife.it, rrovatti@arces.unibo.it

***Abstract*— In this paper we present a macro-model for a true random number generator which internally exploits a pipeline analog-to-digital converter modified to operate as an interleaved chaotic map. The model is tuned to reproduce the non-idealities of a 0.35μm CMOS double-poly triple-metal technology. It is based on circuit-level simulations but is extremely more efficient and can be used to run the statistical tests to assure the quality of the output stream.**

I. Introduction

Random Number Generators (RNGs) represent a fundamental issue in many engineering tasks. For instance RNGs are inherent in many communication protocols and in the synthesis of confidential keys for symmetric and public-key crypto-systems in the computation of algorithms such as the DSA [1], [2]. The properties a RNG must satisfy depend on the specific application where the RNG is employed. Generators suitable for security related applications must meet much stronger requirements than for other applications [2]; the fundamental property of all cryptographic techniques to foil pattern analysis is strongly dependent on the unpredictability of the random generators they employ. In the general case, unpredictability is the most important property of a RNG. We can say a random number generator is *ideal* when, basing on the observation of the previous symbols, we get no information at all about the next symbol.

It is generally recognized that ideal random sources can only be approximated. An ideal 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*).

The RNG we discuss here belongs to the class of *true-random* number generators, i.e. generators that rely on micro-cosmic processes which result in macroscopic observables that can be regarded as random noise. They are the best approximation of ideal random generators and often, in common perception, completely identified with them. This category of generators is widely deployed in security related applications such as data security and cryptography, where the deterministic (and thus predictable) behavior of, for example, *pseudo-random* generators can hardly be desirable.

Our RNG (section II) is based on a chaotic circuit source [3]. The capability of non-linear systems to exhibit non-classical, irregular (i.e. *chaotic*) behaviors has longly been appealing for RNGs [4], [5]. A chaotic random source is based on a deterministic model where entropy enters only as system initial condition. However, the initial condition is set by the system noise floor, so that the information comes from a physical process and the associated entropy is infinite. The main advantages of the proposed circuit is that it is very similar to the implementation of recent pipeline analog-to-digital converters (ADCs) based on 1.5 bit/stage cells; this permits a vast re-use of design competences and macro-blocks developed in this field and also ensure embeddability in all mixed signal integrated circuit.

A circuit implementing this RNG has been designed (section III); to validate the design i.e. to verify the quality of the output stream, statistical tests are available (see, for example, [6], [7] and [8]). However these tests require sequences of millions of bits, which cannot be generated with a circuit-level simulation since, due to the circuit complexity, they achieve a throughput of few hundreds of bits per hours. So we developed an efficient macro-model of the circuit (section IV) based on Monte-Carlo simulations of the circuit; though simple, this model is realistic since it includes possible errors coming from implementation inaccuracies. With this model we can test (section V) the quality of the output stream, and and also check what kind of post-processing circuit is the most appropriate for the circuit.

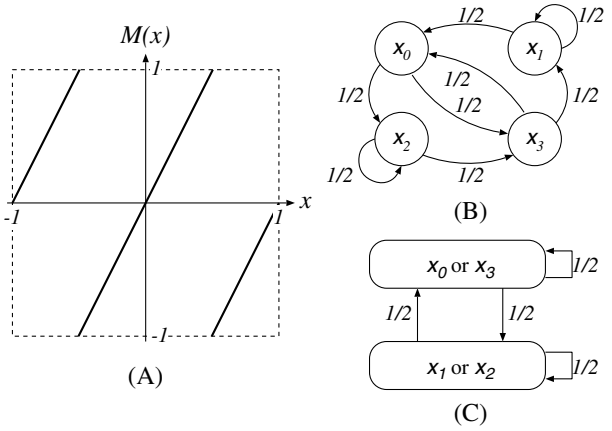


Fig. 1. PWAM map found in ADC pipeline converters (A); Embedded Markov chain (B); and simplified chain (C).

II. Circuit description

The proposed circuit relays on a simple chaotic map as chaotic source. Chaotic maps [9] are 1D discrete-time autonomous systems, whose evolution is described as $x_{n+1} = M(x_n)$ where M is a non-linear function that maps an interval I (usually $I = [-1, 1]$) into itself.

Supposing M is *exact* [9], from any initial condition $x_0 \in I$ these systems generate trajectories whose symbols x_k appears as completely random and also which are distributed according to a symbol density which is independent of the initial condition and called *invariant density*. Also, these systems exhibit a strong dependence on the initial condition: two trajectories given by the same system with two very close initial conditions appear as completely uncorrelated. This property is fundamental since it guarantees non repeatability in true-implemented systems, where initial condition is set by circuit noise. The key idea is to achieve a RNG from a quantization of the system state x_k of a chaotic map. Since quantization is non-reversible operation, the analog state of the system cannot be obtained through the observation of all the quantized quantities and so the evolution of the system cannot be predicted.

The map M used as chaotic source is the map depicted in figure 1-A. This is a variant of the well-known *Bernoulli shift* and has very good robustness properties [10], [11] against both noise perturbations and implementation errors, which make this map very suitable as true-implemented chaotic source. Also this map is a Piece-Wise Affine Markov (PWAM) map [12]; if we set the partition $\{X_0, X_1, X_2, X_3\} = \{[-1, -1/2[, [-1/2, 0[, [0, 1/2[, [1/2, 1]\}$ and consider only the interval where, at every time step, the map state x_k is, the dynamics of the system evolution can be studied through the Markov chain in figure 1-B. The chain is clearly not suitable for the direct generation of independent and identically distributed (*iid*) – loosely speaking random – symbols, but its particularly regular

This is the author's version of the article that has been presented at IEEE ISCAS2005
The editorial version of the paper is available at http://dx.doi.org/10.1109/ISCAS.2005.1465594

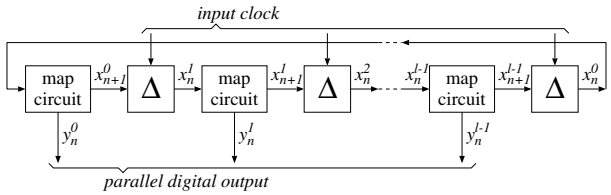


Fig. 2. Basic structure of a pipeline Chaotic Map.

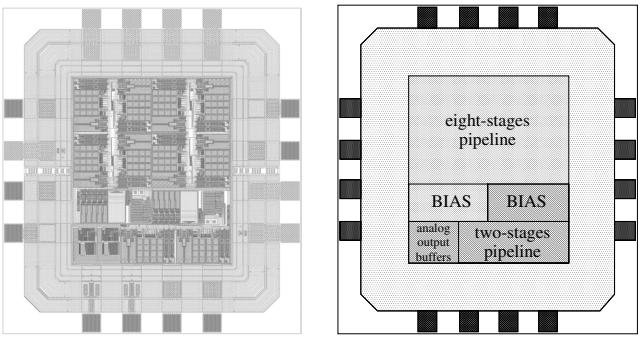


Fig. 3. Layout of the designed ADC-based RNG.

Max. working frequency:	5 MHz
Max. data throughput:	5 Mbit/s per stage
(two-stages pipeline):	10 Mbit/s
(eight-stages pipeline):	40 Mbit/s
Area (with pads):	2.400 mm^2
	(1480 μm x 1620 μm)
Area (without pads):	0.752 mm^2
(two-stages pipeline):	0.234 Mbit/s
(eight-stages pipeline):	0.518 Mbit/s
Power supply voltage:	3.3 V
Power consumption (estimated):	56 mW
(two-stages pipeline):	27 mW
(eight-stages pipeline):	29 mW

TABLE I
CIRCUIT CHARACTERISTIC FOR THE DESIGNED ADC-BASED RNG.

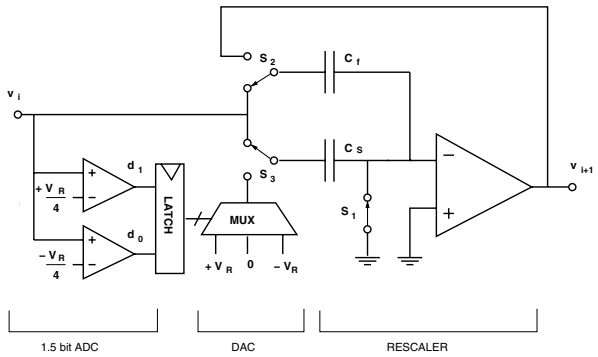


Fig. 4. Switched capacitor 1.5bit A/D converter used as $M(x)$ block.

structures allows us to aggregate its states obtaining a simpler chain, equivalent to that of the unbiased coin toss (figure 1-C).

To minimize the effect of isolated non-idealities and increase throughput it is sensible to unroll the time-updating relationship $x_k = M(x)$ in a l-stages pipeline structure as in figure 2 If the map M of figure 2-A is employed at each stages, the architecture is very similar to that of a pipeline A/D converter, i.e. particular A/D architecture in which the conversion is obtained through a series of l simpler A/D intermediate conversions [13]. In fact, is shown in [3] that ADCs with one-and-a-half bit per stages architecture exploit an analog processing unit whose in/out relationship is precisely that in figure 1-A. The only difference between a 1.5 bit/stage pipeline ADC and the proposed pipeline chaotic map is that the ADC is an open-loop system, while the chaotic map is a closed-loop system.

This structure can be proven [3] to be equivalent to l independent chaotic systems operating in parallel. The model is *interleaved* in the sense that the evolution function of every chaotic system at time n does not depend on its state at time $n-1$, but on the state, at time $n-1$, of another system. With this configuration we get at each time step an iid bit for each of these systems; hence the whole pipeline generates l iid bits per time step.

III. CIRCUIT IMPLEMENTATION

The proposed ADC-based RNG has been implemented with AMS CMOS 0.35μm technology. The layout of the chip is shown in figure 3 while the circuit characteristics are reported in table I. This chip contains two different pipelines, the first with only two stages and the second with eight stages. The two pipeline have two different bias section to reduce interferences, and are designed to operate with a working frequency up to 5 MHz, so the circuit maximum output data rate is 40Mbit/s for the eight-stages pipeline.

The core of the circuit is the 1.5 bit A/D cell whose schematic is shown in figure 4 and is taken from [13]. While a single-ended configuration is shown for simplicity, the actual implementation is fully-differential.

This stage operates on a two-phase clock. In a first phase, the input signal v_i, which ranges from $-V_{\rm ref}/2$ to $+V_{\rm ref}/2$, is applied either to the coarse 1.5 bit ADC (with thresholds $-V_{\rm ref}/4$ and $+V_{\rm ref}/4$) and to the sampling capacitors C_s and C_f. The output of the ADC is also latched at the end of the clock phase. In the second phase, C_f closes the negative feedback loop around the op-amp while C_s is switched to the output of the DAC (a simple three-input multiplexer), thus subtracting the output of the multiplexer from the output signal v_{i+1}. Analytically:

$$V_{i+1} = \begin{cases} \left(1+\frac{C_s}{C_f}\right) V_i - V_{\rm ref}, & V_i < -\frac{V_{\rm ref}}{4} \\ \left(1+\frac{C_s}{C_f}\right) V_i, & -\frac{V_{\rm ref}}{4} < V_i < \frac{V_{\rm ref}}{4} \\ \left(1+\frac{C_s}{C_f}\right) V_i + V_{\rm ref}, & V_i > \frac{V_{\rm ref}}{4} \end{cases} \quad (1)$$

Setting $C_s = C_f$ the resulting input/output characteristic is the desired one.

Due to the switched capacitor implementation this scheme has many advantages. First, S/H stages are not necessary in the pipeline if the cells are driven alternatively with two non overlapping clocks; this however implies that the number of stages must be an even number. Also the switched capacitor implementation ensures a very high accuracy.

All the necessary voltages (0, $\pm V_R$, $\pm V_R/4$) are generated with a resistive matched ladder biased with a current. This does not guarantee extreme accuracy since both bias current and ladder resistance depend on technology parameters which may have large variations; however this is not a problem since the circuit performance depends only on voltage ratios, which conversely can be fixed with high accuracy.

To validate the design, a netlist extracted from layout and affected by parameter variations reproducing fabrication imperfections must be simulated and results matched against test for randomness [6], [7], [8]. Due to the switched capacitor nature of the circuit, time-domain simulations are necessary. These simulations are extremely expensive in terms of computing power. With a state-of-the-art CPU and a commercial *spectre* simulator, we get a simulation speed of about 600bit/hour (i.e. about 0.15bit/s) for the two stages-pipeline circuit. This is of course unacceptable, since many tests require millions of bits to run. For this reason we developed a macro-model capable of a throughput of several order of magnitude higher than the full circuit simulation. The macro-model has been developed from the circuit implementing a two-stages pipeline, but is capable to describe a circuit with any number of stages.

This is the author's version of the article that has been presented at IEEE ISCAS2005
The editorial version of the paper is available at http://dx.doi.org/10.1109/ISCAS.2005.1465594

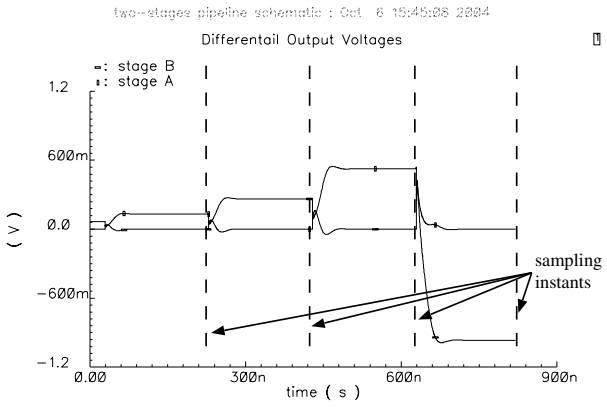
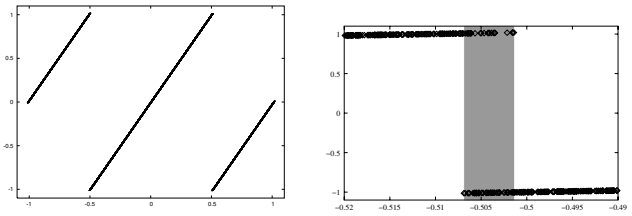


Fig. 5. Short time transient analysis for the two-stages pipeline.



(A) (B)

Fig. 6. A collection of (x_{k+1},x_k) points for a single 1.5 bit ADC stage (A); and zoom around breakpoint α^- (B)

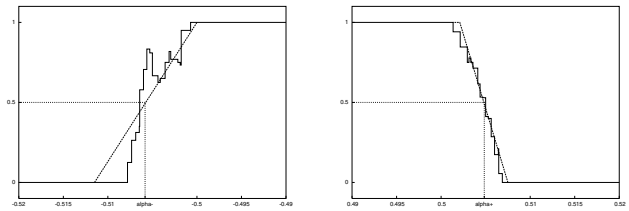
Since the circuit is, ideally, 1D discrete-time and time independent, a 1D discrete-time and time independent model has been selected, i.e. we focused on modeling the profile of the implemented M and describe how its varies depend on implementation inaccuracies. Due to the discrete-time nature of the circuit, the M function can be analyzed only with a collection of (x_{k+1}, x_k) points obtained from simulations. This could be done with a parametric simulation of a single time step with different initial values of x_k; however there are no guarantees that the computed initial solution is the actual one. A better way is to extract the values of the points (x_{k+1}, x_k) from a single, long transient simulation, and down-sample the output stream with a sampling instant few ns before the front of the clock as shown in figure 5, where only the differential values of the circuit outputs are drawn for simplicity. Since it can be proved that points x_k are ideally uniformly distributed in I, from this sampling we obtain a good discretization of M. This simulation has also been performed with different values of the actual process parameters to obtain a model which includes all implementation inaccuracies.

IV. Macro-Model of the Circuit

Many Monte-Carlo runs of about 25×10^3 clock periods have been simulated for the two-stages pipeline circuit. From each of these runs, two sets of about 25×10^3 points (x_{k+1},x_k), one for every stage, have been extracted. From these sets, a version of function M is computed for every stage of every Monte-Carlo run; these functions have been analyzed to obtain a simple but realistic map description including an evaluation of the differences which may exist between two stages of two different pipelines or between two stages of the same pipeline.

The model used for the M function is a piece-wise linear model. The switched capacitor implementation ensures (figure 6-A) a very high linearity, and also a very good precision on the multiplying factor. Also the fully differential architecture ensures a high symmetry; so the M can be described by

$$M(x) = \begin{cases} 2x+\beta & \text{if condition } \lambda_1(x) \text{ is true} \\ 2x & \text{if condition } \lambda_2(x) \text{ is true} \\ 2x-\beta & \text{if condition } \lambda_3(x) \text{ is true} \end{cases} \quad (2)$$



(A) (B)

Fig. 7. Density of points satisfying condition λ_2 (solid line) and linear approximation $p(x)$ (dotted line) around α^- (A) and around α^+ (B).

The determination of the three condition λ_1, λ_2 and λ_3 is non-trivial. Ideally, two breakpoints α^- and α^+ exist, with $\lambda_1(x) \equiv x < \alpha^-$, $\lambda_2(x) \equiv \alpha^- \le x < \alpha^+$ and $\lambda_3(x) \equiv x \ge \alpha^+$. Yet, the real behavior of the system can be seen in figure 6-B, which represents a zoom of figure 6-A around the ideal breakpoint α^-. While at a certain distance from the breakpoint the behavior is fully deterministic, a point very close to the breakpoint could sometimes verify condition λ_1 and sometimes λ_2 (the gray area in the figure). This could be explained considering interferences (e.g spikes on the power supply voltage) coupling from the other parts of the circuit which may alter the behavior of the two comparators. Due to the static nature of the macro-model, these interferences cannot be modeled in any way but as noise perturbation. So a stochastic transition model has been implemented; in this model a probability function decides which linear piece of M is used.

The solid line of figure 7 shows the density of points around the breakpoints verifying condition λ_2 in a Monte-Carlo run; the figure has been obtained with an histogram analysis. Assuming the system is ergodic, this function has been taken as the probability function $p(x)$ that condition λ_2 is verified for a point x. With this we set

$$M(x) = \begin{cases} 2x+\beta & x<0, \text{with probability } 1-p(x) \\ 2x & \text{with probability } p(x) \\ 2x-\beta & x>0, \text{with probability } 1-p(x) \end{cases} \quad (3)$$

Ideally $p(x) = \chi_{[\alpha^-,\alpha^+]}$, where χ is the classical indicator function of an interval. In this model, $p(x)$ has been considered a trapezoidal function (the dotted line in figure 7):

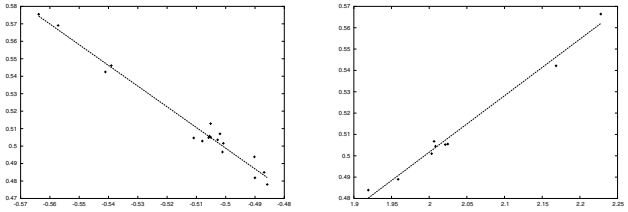
$$p(x) = \begin{cases} 0 & x < \alpha^- - \frac{1}{2s^-} \\ 1/2 + \left(x-\alpha^-\right)s^- & \alpha^- - \frac{1}{2s^-} \le x < \alpha^- + \frac{1}{2s^-} \\ 1 & \alpha^- + \frac{1}{2s^-} \le x < \alpha^- - \frac{1}{2s^+} \\ 1/2 - \left(x-\alpha^+\right)s^+ & \alpha^+ - \frac{1}{2s^+} \le x < \alpha^+ + \frac{1}{2s^+} \\ 0 & x \ge \alpha^+ + \frac{1}{2s^+} \end{cases} \quad (4)$$

The breakpoints α^- and α^+ are the points where the (fitted) probability to be in one or another of the two linear pieces of M is equal. These parameters, as well as s^- and s^+, are computed through two separate linear regression, including all points around α^- (or α^+ respectively) that verify condition λ_2 in each Monte-Carlo run. All points far enough from the breakpoints to ensure a deterministic decision (i.e. when the density is 1 or 0) have not been considered. Even if some slightly differences can be observed in the value of α^- and α^+, they are strongly related (see figure 8-A) so we can assume $\alpha = \alpha^+ = -\alpha^-$.

Numerical analysis shows that there can be large differences in these parameters for different pipeline implementations. However the differences between different stages in a single pipeline are very small. This reflects the fact that the latter differences depend only on inaccuracies such as matching errors, which are typically limited.

For example, a variation up to $\pm20\%$ from its nominal value can be observed in β in different Monte-Carlo runs since it depends on reference voltages which may strongly vary; however no sensible variation can be observed in β for the two stages of a single Monte-Carlo run, since in a single simulation (i.e. in a single pipeline) the

This is the author's version of the article that has been presented at IEEE ISCAS2005
The editorial version of the paper is available at http://dx.doi.org/10.1109/ISCAS.2005.1465594



(A) (B)

Fig. 8. Scatter plots for α^- vs α^+ (A); and β vs α (B)

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

TABLE II
EXPECTED VALUE AND DEVIATION FOR MODEL PARAMETERS.

reference voltages are the sames. So β in the macro-model is assumed to be a global parameter for the entire pipeline. From the analysis of β in all Monte-Carlo runs, its value is modeled as a Gaussian random variable with mean value and variation shown in table II. On the contrary, a fluctuation on the value of α can be observed even between the two stages of a single pipeline. The value of α in the macro-model so is computed for every stage of a pipeline as $\alpha = \alpha^0 + \Delta\alpha$ where α^0 is a global parameter for the entire pipeline, and $\Delta\alpha$ is computed for every stage as in table II. Actually, α^0 is not an independent parameter, and it is strongly related to β (see figure 8-B). In the model it is taken $\alpha^0 = \beta/4$, since this is the expected relation between these two parameters. However, no relations between the values of s^-, s^+ and the other parameters has been found; also since no link with the other parameters can be anticipated based on circuit design, they are taken as independent for each pipeline stage.

In the light of this, the proposed model has four parameters:

- β, which is assumed as a global random variable for the entire pipeline.
- $\Delta\alpha$, s^- and s^+, which are random variables computed for each stage of the pipeline.

All these parameters have been assumed to be Gaussian random variables.

	no post–processing	XOR post–processing XOR-2	XOR-4	NLSR
SP800-22 test	fraction of P-value greater than 0.01			
F	0.057478	1.000000	0.988839	0.988839
BF	0.717076	0.914062	1.000000	0.997768
RN	0.062500	0.594866	1.000000	0.992746
LROO	0.723214	0.984375	0.986607	0.991071
RK	0.991629	0.988839	0.988839	0.988281
DFT	0.289621	0.991071	0.997768	0.992746
NOTM	0.292411	0.981027	0.993304	0.984933
OTM	0.407366	0.966518	0.997768	0.986607
U	0.460938	0.992188	0.991071	0.989955
LC	0.987165	0.989955	0.984375	0.988839
LZ	0.286830	0.982143	0.988839	0.991629
S1	0.037946	0.727679	0.986607	0.992188
S2	0.724330	0.983259	0.986607	0.992188
AE	0.042969	0.667411	0.986607	0.993304
CS1	0.056920	0.997768	0.986607	0.991629
CS2	0.060268	0.998884	0.988839	0.992746
RE	N/A	0.986254	0.986111	0.984862
VRE	N/A	0.986254	0.993056	0.991986

TABLE III
RESULTS OF RANDOMNESS TESTS ON THE CHAOTIC-ADC RNG WITH DIFFERENT POST-PROCESSING STAGES.

V. TESTS

The proposed model has been used to perform extensive tests among several possible choice in terms of number of stages and post-processing architectures. As an example, table III reports the results for the SP800-22 test suite [7]. Test are performed generating instances of an eight-stages ADC. Each instance is simulated to assess its quality as a converter, and only instances actually producing a conversion error not larger than 1lsb are passed onto the following step. For all ADCs passing we generate 10^6 bits. We stop when about 2000 random number generators have been tested.

Results are presented for the system without any post-processing stage, with a simple XOR post-processing stage with depth-level of two and four and a NLSR post-processing [14] Table III reports the fraction of instances that produce a sufficiently high P-value which is a real number in $[0, 1]$ estimating the probability that a finite realization of an ideally random binary process deviates from the ideal statistic more than the given string. Obviously, the higher the P-value the more random the string. We consider that a test is passed if the P-value is greater than $\alpha = 0.01$, as suggested in [7].

As can be seen, our circuit ensures a very good randomness even with simplest post-processing stages.

VI. CONCLUSIONS

In this paper an implementation of a 1.5 bit ADC based RNG is presented. The circuit has been designed and a macro-model of the circuit from layout simulations has been extracted. This macro-model has been developed with the analysis of several Monte-Carlo simulation runs, and it is intended to include errors due both to coupling perturbations, observed in the comparators behavior, and to circuit implementation inaccuracies. Simulations carried over by means of the macro-model produce few Mbits/s to be compared with the 0.15bit/s of the full transient simulation of the simplest (unrealistic) circuit. Hence, its availability is essential to perform the very long simulation needed to assure the quality of the output stream with respect to the stringent statistical tests currently adopted.

REFERENCES

[1] W. Schindler and W. Killmann, "Evaluation criteria for true random number generators used in cryptographic applications," in CHES 2002, vol. 2523 of *Lecture Notes in Computer Science*, pp. 431–449. Springer-Verlag, Feb. 2003.
[2] D. E. Eastlake, S. D. Crocker, and J. I. Shiller, "RFC 1750: Randomness recommendations for security," in *Internet Society Request for Comments*. Internet Engineering Task Force, 1994.
[3] S. Callegari, R. Rovatti, and G. Setti, "Efficient chaos-based secret key generation method for secure communications," in *Proceedings of NOLTA*, Xian, Oct. 2002.
[4] M. Delgado-Restituto, F. Medeiro, and A. Rodríguez-Vázquez, "Non-linear, switched current CMOS IC for random signal generation," in *Electronics Letters*, no. 25, pp. 2190–2191, 1993.
[5] T. Stojanovski, L. Kocarev "Chaos-based random number generators – part II: Practical realization," in *IEEE Transactions on Circuits and Systems,*, Part I, pp. 48(3):382-385 2001.
[6] "Security requirements for cryptographic modules," Federal Information Processing Standards 140-2, National Institute for Standards and Technology, (NIST) May 2001.
[7] "A statistical test suite for random and pseudorandom number generators for cryptographic applications," Special publication 800-22, National Institute for Standards and Technology (NIST), May 2001.
[8] *The Marsaglia Random Number CD-ROM, with the Diehard Battery of Test of Randomness*, Florida State University, 1985.
[9] A. Lasota and M. C. Mackey, *Chaos, Fractals and Noise. Stochastic Aspects of Dynamics*, Springer-Verlag, 2nd edition, 1995.
[10] S. Callegari, R. Rovatti and G. Setti, *Robustness of Chaos in Analog Implementations*, ch. 12, pp. 397–442. In [12], 2000.
[11] F. Pareschi, G. Setti and R. Rovatti, "Noise Robustness condition for chaotic maps with Piecewise constant invariant density," in *Proceedings of ISCAS*, Vancouver, May 2004.
[12] M. P. Kennedy, R. Rovatti, and G. Setti, eds., *Chaotic Electronics in Telecommunications*. Boca Raton: CRC International Press, 2000.
[13] A. M. Abo and P. R. Gray, "A 1.5-V, 10-bit, 14.3-MS/s CMOS pipeline analog-to-digital converter," in *IEEE Journal of Solid State Circuits*, vol. 34, no. 5, pp. 599–606, May 1999.
[14] S. Poli, S. Callegari, R. Rovatti, G. Setti, "Post-Processing of data generated by a chaotic pipelined ADC for the robust generation of perfectly random bitstreams," in *Proceedings of ISCAS*, Vancouver, May 2004.