A Robust Chaos-Based True Random Number Generator Embedded in Reconfigurable Switched-Capacitor Hardware

Miloš DRUTAROVSKÝ, Pavol GALAJDA

Dept. of Electronics and Multimedia Communications, Technical University of Košice, Park Komenského 13, 04120 Košice, Slovak Republic

milos.drutarovsky@tuke.sk, pavol.galajda@tuke.sk

Abstract. *This paper presents a new chaos-based True Random Number Generator (TRNG) with a decreased voltage supply sensitivity. Contrary to the traditionally used sources of randomness it uses a well-defined deterministic switched-capacitor circuit that exhibits chaos. The whole design is embedded into a commercially available mixed-signal Cypress PSoC reconfigurable device without any external components. The proposed design is optimized for a reduction of influence of the supply voltage to the quality of the generated random bit stream. The influence of circuit non-idealities is significantly reduced by the proposed XOR corrector and optimized circuit topology. The ultimate output bit rate of the proposed TRNG is 60 kbit/s and the quality of generated bit-streams is confirmed by passing standard FIPS and correlation statistical tests performed in the full range of PSoC device supply voltages.*

Keywords

Cryptography, chaos, Markov chains, TRNG, PSoC mixed-signal hardware, FIPS tests.

1. Introduction

With the rapid development of computer networks and wireless communications, information security becomes one of the major problems for effective use of information technology. These problems can be solved by using cryptography. In many practical applications (e.g. large sensor networks) cryptographic primitives have to be implemented in a very cost effective way by using standard hardware components. In a cryptographic application where ultimate security is necessary, a True Random Number Generator (TRNG) is required. TRNGs are widely used for example as confidential key generators for symmetric key crypto-systems (e.g. AES) and public-key ones (e.g. RSA, ECC). In some algorithms (e.g. digital signatures) or protocols (e.g. zero-knowledge), random numbers are intrinsic to the computation [1].

Classical TRNG uses some random physical phenomenon [2]. Currently the most frequently used phenomenon in embedded TRNGs is direct noise amplification of a resistor [3] and jitter noise of digital clock signals [4]. Embedded TRNGs frequently use some external devices or rather heuristic source of randomness (e.g. metastability [5]). Although these limitations can be overcome by a design of proper custom circuits, randomness extraction is still a big challenge in the designs based on off the shelf devices as FPGA [6], [7] or general microprocessors [8].

Chaotic circuits [9] represent an efficient alternative to the classical TRNGs. Contrary to traditionally used sources of randomness they use a well-defined analog deterministic circuit that exhibits chaos. Chaotic systems are characterized by a "sensitive dependence on initial conditions", i.e. a small perturbation eventually causes a large change in the state of the system. However, the slightest uncertainty about the initial state (which is unavoidable in all analog implementations of chaos systems) leads to a very large uncertainty after very short time. With such initial uncertainties, the system's behavior can be predicted only for a short time period. Additionally as it has been recently shown [10], if the state variable of chaos system is not available to the observer, and the chaos-based system map is well designed, the output of the system cannot be predicted at all. Such implementation is a source of infinite entropy that is absolutely required for a good TRNG.

Many random number generators based on analog and deterministic chaotic phenomena have been proposed, see e.g. [11], [12] and references in [10]. Some of them have been simulated only. Others have not been sufficiently optimized for cryptographic applications as they provide certain bias or other deviations. The design [14] describes practical implementation of recently proposed deterministic chaos circuit based on Markov map [9], [13]. In contrary with Field Programmable Analog Array (FPAA) implementation [13], chaos based TRNG implementation [14] uses mixed-signal Cypress PSoC reconfigurable hardware [15], [16]. It was shown that after a suitable modification of originally proposed Markov map [10] it can be easily embedded in the selected PSoC hardware [14], [17]. Al-

though this TRNG provides much lower output speeds than the pure FPAA implementation [13], used mixed-signal PSoC hardware includes also embedded microcontroller. Such hardware can be in principle used for other cryptographic tasks (e.g. in sensor networks). Moreover, special attention was devoted to the identification of analog circuit non-idealities. The original design [17] is optimized for reduction of their influence and is fully functional for the supply voltage V_{CC}=5 V±10%. This paper describes a new robust implementation of TRNG in PSoC device that is functional in the full range of supply voltages V_{CC}=3.1 V÷5.5 V.

The paper is organized as follows. A brief overview of the theory of Markov chaotic sources is given in Section 2. In Section 3, a new robust method of mapping chaos based TRNG into PSoC devices is presented. The experimental TRNG hardware used to test the proposed method is described in Section 4. In Section 5, statistical evaluations of internal and output TRNG signals are made. Finally, concluding remarks are presented in Section 6.

2. An Overview of the Theory of Markov Chaotic Sources

An ideal Random Number Generator (RNG) is a discrete memory-less information source that generates equiprobable symbols and its state chain is shown in Fig. 1a.

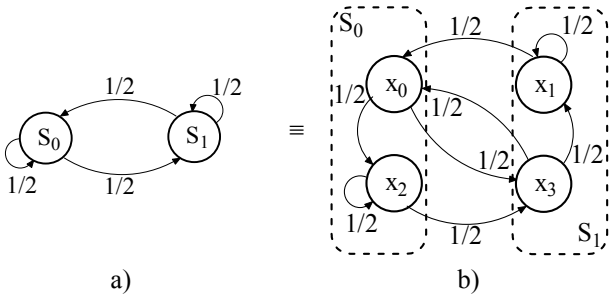

Fig. 1. State chains a) of an ideal RNG, b) Markov chain of the proposed RNG implementation.

The system is in the state S_0 when an output of the RNG has been "0" and in the state S_1 when an output has been "1". A RNG output has always the same probability ½ to get the system in either state, S_0 or S_1. It is known that these sources can be build up from Piece-Wise Affine (PWA) Markov chaotic maps [10]. These maps are one dimensional, discrete-time ones, in which the state variable $x(n)$ is computed as

$$x(n+1) = M[x(n)] \tag{1}$$

where M: [-1,1] → [-1,1], and $x(0)$ is the initial condition. We focus on the recently proposed PWA chaotic map [9], [13]

$$M(x) = (2x+1) \bmod 2 - 1 \tag{2}$$

that is plotted in Fig. 2. The map $M[(x)]$ can be expressed as

$$x(n+1) = \begin{cases} 2x(n)-2 & \text{for } x > 1/2 \\ 2x(n) & \text{for } -1/2 < x < 1/2 \\ 2x(n)+2 & \text{for } x < -1/2 \end{cases} \tag{3}$$

The main advantage of this chaotic map is its increased robustness in the case of an analog implementation [9], [10]. The dynamics of equation (1) can be represented by the Markov chain and its evolution can be studied through a square matrix (often referred to as the kneading matrix) defined for the map (3) in Fig. 2 [9], [10].

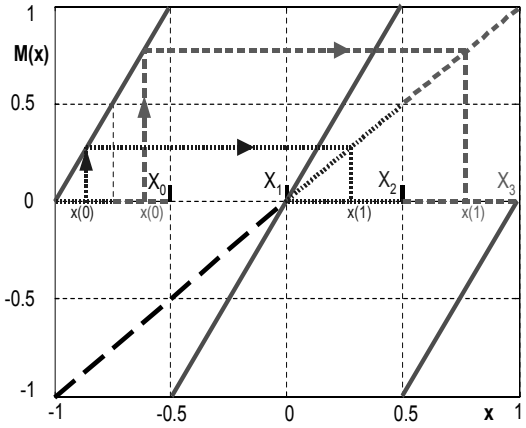

Fig. 2. Proposed piece-wise chaotic map based on the eq. (3) and its kneading matrix.

This iterative process (1), (2) can be interpreted as a Markov state chain with 4 states (Fig. 1b). The state machine is in its discrete state x_i when the chaotic system has its continuous state variable x in the partition interval X_i. The weights assigned to the graph arrows represent the probabilities by which the state machine changes from one state to another.

The chain in Fig. 1b is not suitable for a direct realization of the ideal RNG since it is sequential (it has a memory). However, it is possible to easily build a rigorously independent binary sequence from $x(n)$. In fact, it is sufficient to aggregate the Markov states into two macrostates S_0 and S_1 shown with dotted lines in Fig. 1b [17]. Note, that this aggregation is different from that introduced in [10] in order to allow simpler implementation in PSoC devices.

3. PSoC Based TRNG

3.1 An Overview of PSoC Architecture

The Cypress PSoC™ device family [15] consists of a mixed-signal array with an on-chip CPU. A PSoC device includes configurable Analog Blocks (AB) [16] and Digital Blocks (DB), as well as programmable interconnections as is shown in Fig. 3.

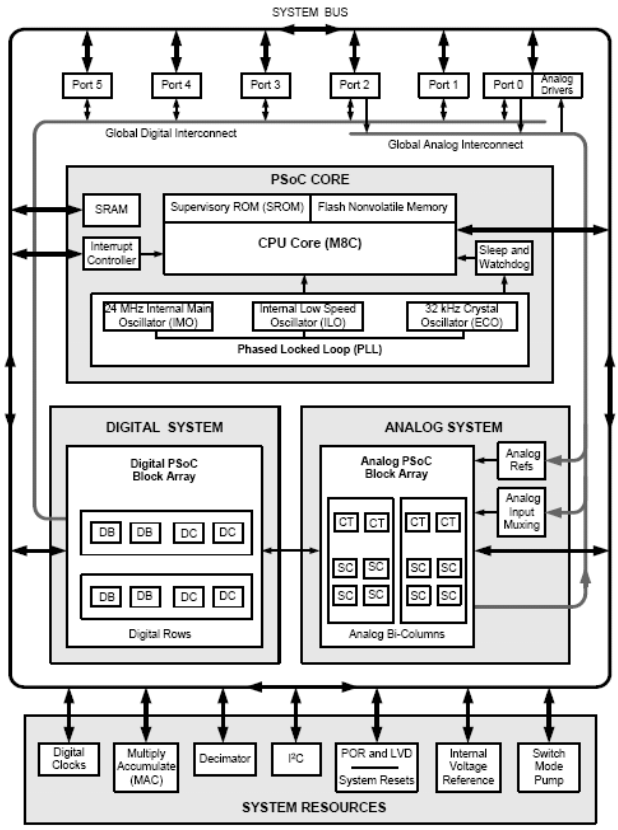

Fig. 3. Block diagram of PSoC device architecture.

The basic features of the PSoC device are as follows:

- powerful Harvard CPU architecture,
- advanced peripherals (PSoC Blocks):
  - 4 rail-to-rail continuous analog PSoC blocks,
  - 8 Switched Capacitor (SC) analog blocks,
  - 8 digital PSoC blocks,
- programmable references voltage $V_{ref} = 1.3$ V (set by the internal bandgap reference) or $V_{ref} = V_{CC}/2$,
- internal 24 MHz oscillator (allows to build a real single chip application),
- precision, programmable clocking (provides two phase clocks- Φ_1, Φ_2 for SC),
- flexible on-chip memory,
- programmable pin configurations.

The PSoC architecture allows implementation of many different functions merely by altering the internal circuit's switches. The on-chip CPU controls the functionality of the digital and analog blocks and it can dynamically change the parameters (e.g. gain of SC block) and topology of these blocks.

3.2 Four SC Blocks TRNG Architecture

As shown in [10], the chaotic map in Fig. 2 can be implemented by a pipeline-ADC with 1.5-bit/stage architecture. In [17] we introduced a different implementation of the chaotic map (3) by using four SC (4-SC) analog PSoC blocks. The equation (3) can be rewritten according to Tab. 1 where $b(n+1)$ is a binary output of the TRNG at the time step $(n+1)$.

The steps given by equations in Tab. 1 can be directly mapped into four PSoC SC blocks shown in Fig. 4. Addition/subtraction of the reference voltage and comparison operations from steps one and two are mapped to SC1 and SC3, respectively. Multiplication by 2.0 from step three is realized in SC3 and SC4. Comparison from step three is implemented in SC3. We have chosen a common SC implementation operating on two-phase clocks (Φ_1 and Φ_2 driven by the on-chip oscillator). During the first phase, the block SC1 adds the reference voltage $\pm V_{ref}$ (the actual polarity is controlled by on-chip CPU) to the input voltage V_{in} according to the polarity of the analog voltage at the output of the block SC3. If this voltage is negative, the

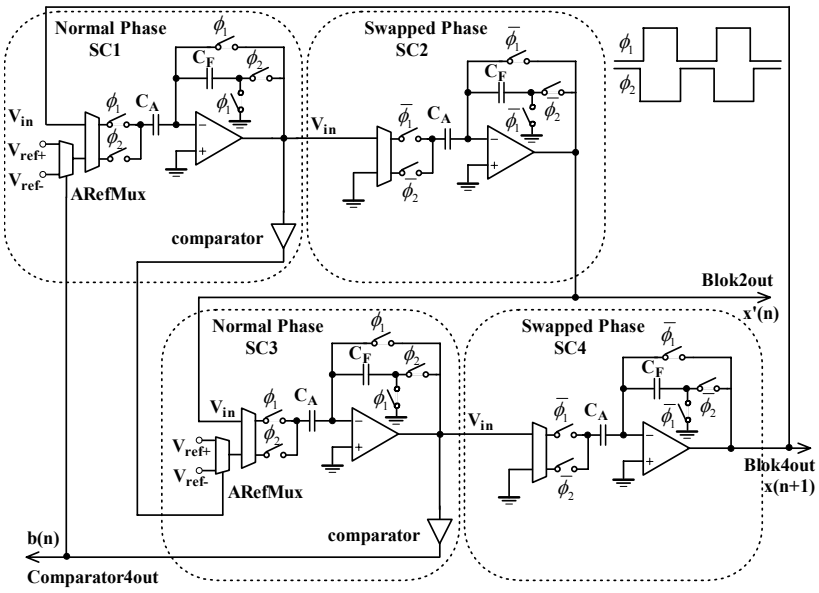

Fig. 4. PSoC implementation of the 4-SC blocks TRNG.

reference voltage $V_{ref+} = +V_{ref} = 1.3$ V will be added in the next clock phase to the input voltage, otherwise the voltage $V_{ref-} = -V_{ref} = -1.3$ V must be added. During the second clock phase, the block SC1 tests the polarity of the output voltage for the block SC3. The operation of the block SC3 is similar to that of the SC1.

1st step	$x'(n) = \begin{cases} x(n) - V_{ref} \text{ for } x(n) > 0 \\ x(n) + V_{ref} \text{ for } x(n) < 0 \end{cases}$
2nd step	$x''(n) = \begin{cases} x'(n) - V_{ref} \text{ for } x'(n) > 0 \\ x'(n) + V_{ref} \text{ for } x'(n) < 0 \end{cases}$
3rd step	$x(n+1) = 2x''(n) \quad \text{for all } x(n)$ $b(n+1) = \begin{cases} \log 1 \text{ for } x(n+1) > 0 \\ \log 0 \text{ for } x(n+1) < 0 \end{cases}$

Tab. 1. TRNG chaotic map equations optimized for PSoC 4-SC blocks.

The blocks SC2 and SC4 realize the Sample and Hold function between the blocks SC1 and SC3. This is needed for a correct operation of SC blocks connected as a "ring oscillator". Additionally, the blocks SC3 and SC4 define the gain of the circuit, which has to be as close as possible to the ideal gain equal to 2.0. In our case the gain is limited by discrete values of capacitors C_F and C_A to the value

$$(27\times19)/(16\times16)=513/256\approx 2.004. \tag{4}$$

It can be shown that the structure in Fig. 4 processes two independent bit streams, $b_1(n)$ and $b_2(n)$ in a pipeline processing structure. The bit streams are read out sequentially from this structure and they are overlapped according to the equations

$$\begin{aligned} b_1(n) &= b(2n), \ n = 0,1,\ldots \\ b_2(n) &= b(2n+1), \ n = 0,1,\ldots \end{aligned} \tag{5}$$

Properties of 4-SC TRNG:

- it uses a voltage range from *AGND* to $\pm 2V_{ref}$ ($x(n)$ is actually between 0 V and 5 V, with a low distortion in power supply boundary voltages),
- there are two independent bitstreams (can be used for effective postprocessing, as was shown in [17]),
- it requires a relatively simple control that must be performed by the processor core (only one analog comparator interrupt is used),
- the gain is split and can be set more precisely to the optimal value (equal to 2),
- it uses no external devices.

Disadvantage of 4-SC TRNG:

- the proposed design requires the supply voltage equal to 5 V. This drawback can be resolved by the new design with 5-SC blocks TRNG implementation, as described in the next section,
- the sensitivity to the V_{CC} can be removed only by using an external V_{ref}.

3.3 New Robust PSoC TRNG Architecture

Robust PSoC TRNG is similar to that realized by our 4-SC blocks TRNG with the difference that SC1 block in 4-SC TRNG is replaced by a pair of SC11 and SC12 ones, as is shown in Fig. 5. The main practical disadvantage of 4-SC TRNG implementation is the above mentioned requirement of the supply voltage $V_{CC} = 5$ V. This is a consequence of mapping equations given in Tab. 1 that are

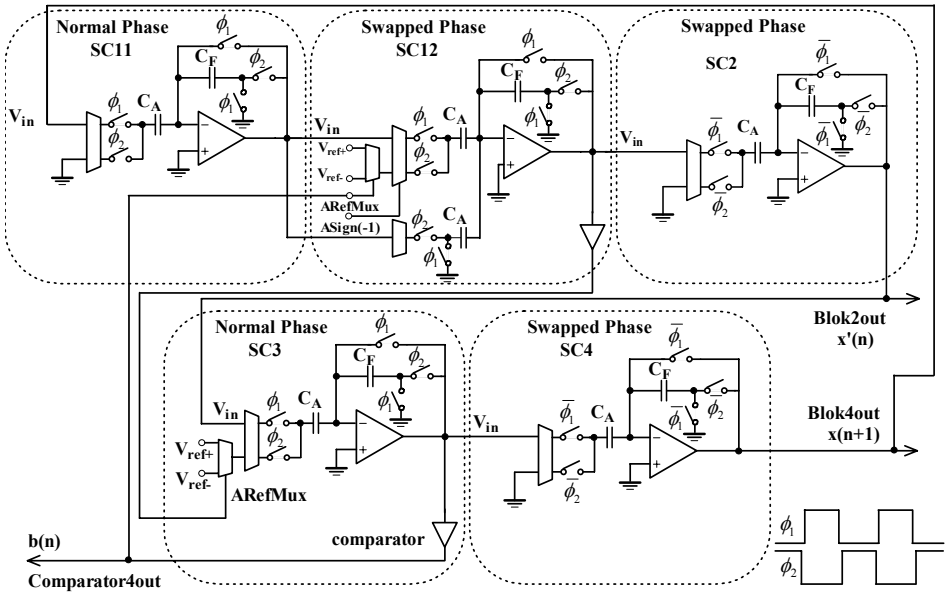

Fig. 5. PSoC implementation of the new 5-SC blocks TRNG.

optimized for use with the internal $V_{\text{ref}} = 1.3$ V. This drawback can be resolved by the design referred as 5-SC blocks TRNG implementation, as described in this section.

1st step	$x'(n) = \begin{cases} V_{ref} - 2x(n) & \text{for } x(n) > 0 \\ -V_{ref} - 2x(n) & \text{for } x(n) < 0 \end{cases}$
2nd step	$x(n+1) = \begin{cases} x'(n) - V_{ref} & \text{for } x'(n) > 0 \\ x'(n) + V_{ref} & \text{for } x'(n) < 0 \end{cases}$
3rd step	$b(n+1) = \begin{cases} \log 1 & \text{for } x(n+1) > 0 \\ \log 0 & \text{for } x(n+1) < 0 \end{cases}$

Tab. 2. TRNG chaotic map equations optimized for PSoC 5-SC blocks.

The 5-SC TRNG implementation realizes the equation (3) in a different way than in the case with 4-SC blocks. The steps are given according to Tab. 2, where $b(n+1)$ is a binary output of the TRNG at the time step $(n+1)$. The steps given by equations in Tab. 2 can be directly mapped into five PSoC SC blocks as shown in Fig. 5 and its hardware implementation is depicted in Fig. 6. The user module placement of 5-SC TRNG in PSoC reconfigurable hardware is shown in Fig. 7. The equations in the 1st and the 2nd steps generate a modified map depicted in Fig. 8. Although this is a mirrored version of the original map (3), it has no influence on TRNG function. The voltage range of values $x(n)$ are from $AGND$ to $\pm V_{\text{ref}}$, where $V_{\text{ref}} = V_{CC}/2$, whereby 5-SC TRNG has become independent on the supply voltage V_{CC}. The detailed operation of individual steps realized by SC blocks is similar as those described for 4-SC TRNG.

It can be shown that the switched-capacitor structure in Fig. 5 processes two independent bit streams, $b_1(n)$ and $b_2(n)$ in a pipeline processing structure analogous to 4-SC TRNG. This is a direct consequence of 2-phase clocks-Φ_1, Φ_2 used for SC. The bit streams $b_1(n)$ and $b_2(n)$ are read out sequentially from this structure and they are overlapped according to the equation (5).

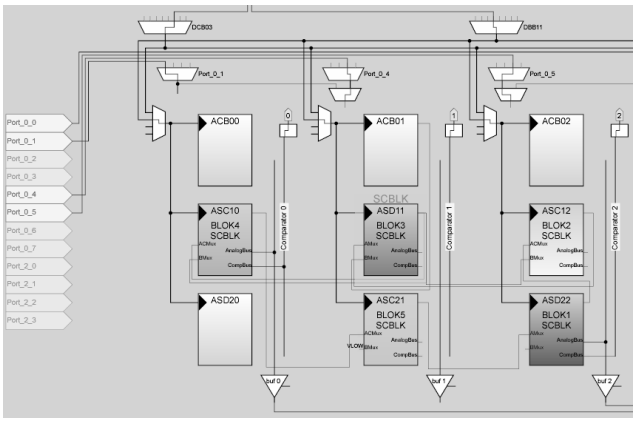

Fig. 7. User module placement of the 5-SC blocks TRNG implemented in PSoC reconfigurable hardware.

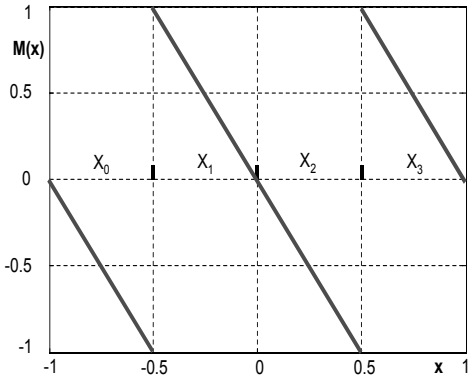

Fig. 8. The modified piece-wise chaotic map of 5-SC TRNG implementation.

4. Experimental TRNG Hardware

The experimental TRNG bit streams were acquired from a PSoC based TRNG implementation depicted in Fig. 6 [14]. In Fig. 9, particular testing signals acquired from the oscilloscope are shown, as well. These signals are connected to the output pins just for testing purposes. During normal TRNG operations their outputs are disabled. Presented results have been obtained using 28-pin CY8C27443 based PSoC module used in PSoC Cypress contest [14]. All tested bit streams analyzed in the next section were generated with 60 kbit/s.

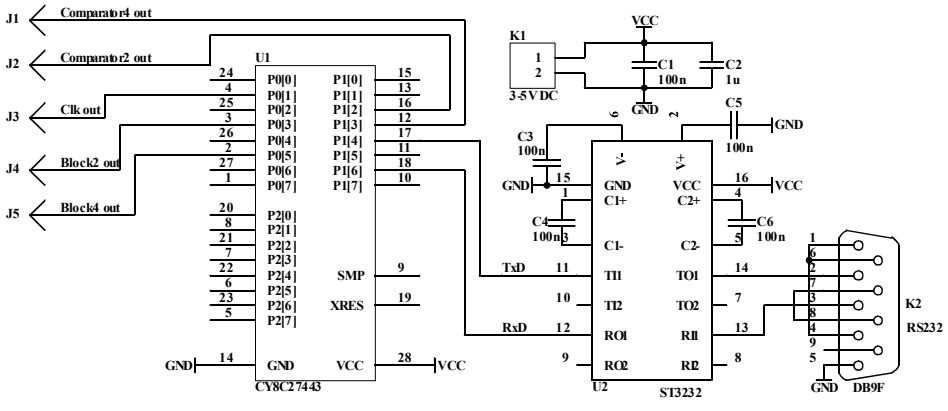

Fig. 6. Schematic diagram of the TRNG testing hardware. Outputs $J1$, $J2$, …, $J5$ are for testing purposes only.

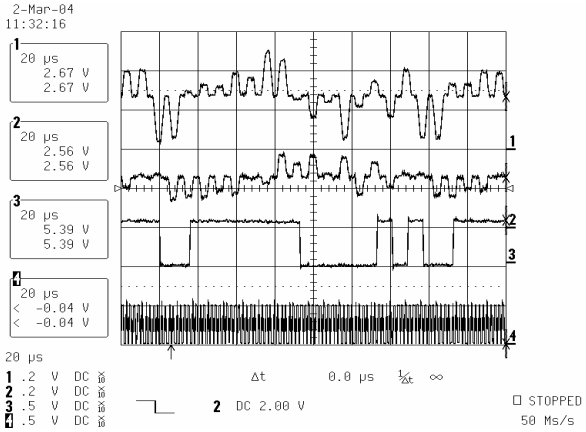

Fig. 9. Waveforms 1, 2, 3, and 4 represent $x'(n)$ (main analog output), $x(n+1)$ (sub analog output), $b(n)$ (main compare output), and SC clock (clk source), respectively.

5. Testing of Generated TRNG Data

Any practical TRNG implementation behaves as an information source with a memory and generates non-equiprobable bits. This is caused by circuit nonidealities (e.g. SC circuit offsets, gain errors, temperature and supply voltage variations, etc.) and generated bits exhibit a certain redundancy. In the following section we will use some well-defined basic statistical tests [1], to measure the deviation of the proposed TRNG from the ideal RNG.

5.1 Correlation Tests

When two random variables b_1 and b_2 are statistically independent, their correlation is zero (note that the opposite statement is not generally valid). Correlations are always between -1 and 1 and are defined as

$$\mathrm{corr}(b_1, b_2) = \frac{E\left[\left(b_1 - E[b_1]\right)\left(b_2 - E[b_2]\right)\right]}{\sqrt{\mathrm{var}(b_1)\,\mathrm{var}(b_2)}}. \tag{6}$$

$E[b]$ denotes the expected value or the mean value of b, that is, the average value of a large number of repeated trials. In the case of random bits, $E[b]=\Pr(b=1)$ where Pr denotes probability. The expression

$$\mathrm{var}(b) = E[(b - E[b])^2] \tag{7}$$

denotes the variance of b.

From the previous analysis we know that the proposed TRNG generates two interleaved data streams, $b_1(n)$ and $b_2(n)$ that should be completely independent (and hence uncorrelated). Crosscorrelation values of 1,000,000 bit long sequences $b_1(n)$ and $b_2(n)$ are shown in Fig. 10. They are within the expected random fluctuations

$$\approx \pm 3/\sqrt{1{,}000{,}000} = \pm 0.003 \tag{8}$$

expected for the ideal RNG [1] and 1,000,000 bit records. This result confirms that deviation from independency of b_1 and b_2 cannot be detected for 1,000,000 bit records.

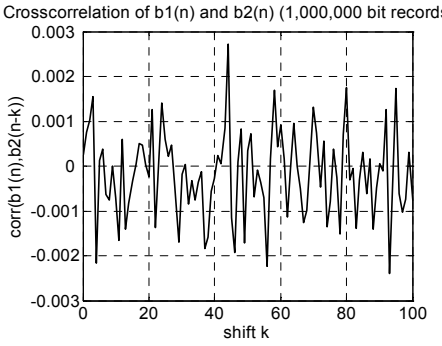

Fig. 10. Crosscorrelation of 1,000,000 bit records $b_1(n)$ and $b_2(n)$ for different shift values k.

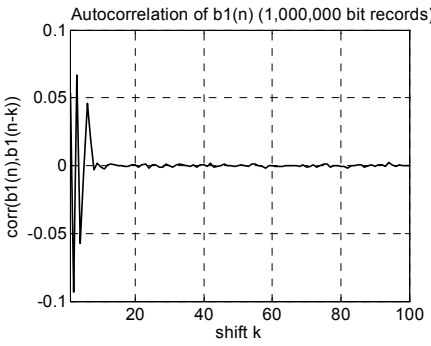

Fig. 11. Autocorrelation of 1,000,000 bit record $b_1(n)$ for different shift values k.

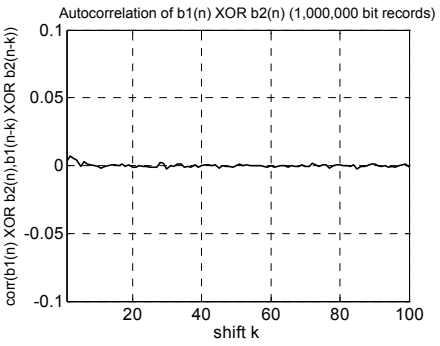

Fig. 12 Autocorrelation of XOR decimated 1,000,000 bit record $b_1(n)$ XOR $b_2(n)$ for different shift values k.

Autocorrelation values for the sequence $b_1(n)$ are shown in Fig. 11 (the values for $b_2(n)$ have the same character and are not included due to a space limitation). It is clearly visible that the autocorrelation values for small values ($k = 1, 2, \ldots 7$) are higher than random fluctuations given by limit (7) for 1,000,000 bit records. These deviations are caused e. g. by the circuit saturation, V_{ref} asymmetry and a gain error of SC circuits as it was shown in [17]. Although these deviations are clearly visible, they are relatively small and can be further decreased by a suitable additional postprocessing.

5.2 Postprocessing of Raw TRNG Data

Redundancy in an information source can be caused by two sources, namely a difference in the probabilities of the two binary symbols, and a memory of an information source. The simplest redundancy reduction technique that affects both sources of randomness is XOR correction [19]. In XOR correction, nonoverlapped bits from the original binary sequence are grouped in a block of p bits and summed up modulo 2 to produce one output bit per block. An example of XOR corrector for p=2 is shown in Fig. 13.

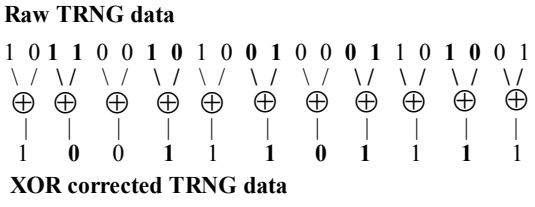

Fig. 13. Example of XOR correction of TRNG output for p=2 that is used for reduction of the output bitstream bias.

It must be stressed that XOR correction can improve statistical properties of the input bitstream only if the input bits are statistically independent. It was shown in [19] that statistical properties of two weakly correlated and slightly biased sequences can be improved, as well.

XOR corrector can be easily implemented in the proposed TRNG as there are two independent bitstreams $b_1(n)$ and $b_2(n)$. Each of these bitstreams is only slightly autocorrelated, as shown in Fig. 11. The bias of $b_1(n)$ and $b_2(n)$. are also very small and XOR corrector with p=2 improves the statistical properties of the corrected TRNG bitstream. Fig. 12 shows autocorrelation of the XOR corrected sequence. It is clearly visible that autocorrelation properties of the XOR corrected sequence are significantly better and consistent with the results [19], [17].

5.3 FIPS Statistical Tests

Testing of the quality of random data is a very important issue, especially in cryptography. There are several ready to use test packages and recommendations. We have chosen FIPS140-2 tests that analyze 20,000 bit records and define thresholds to assess TRNG randomness. Note that FIPS140-2 tests are really only a very basic statistical tests [1] and they cannot reveal all possible TRNG deviations. If a deeper analysis is required it is possible to use e.g. the NIST statistical test suite [17] that can analyze much longer bit records. Longer records are necessary for detection of very small statistical deviations.

FIPS 140-2 consists of standard and easily understood statistical tests (if one of the tests fails, the generator fails the test) [18]:

5.3.1 Monobit Test

The number of 1's (T_1) in the tested bit stream should satisfy 9726< T_1<10274.

5.3.2 Poker Test

Divide the 20,000 bit stream into 5,000 contiguous 4 bit segments. Count and store the number of occurrences of the 16 possible 4 bit values. Denote *f(i)* as the number of each 4-bit value where 0≤ i ≤15. Evaluate the following:

$$T_2 = \frac{16}{5000}\left(\sum_{i=0}^{15} f^2(i)\right) - 5000 \tag{9}$$

The test is passed if 2.16<T_2<46.17.

5.3.3 Run Test

A run is defined as a maximal sequence of consecutive bits of either all ones or all zeros that is a part of the 20,000 bit sample stream. The incidences of runs (for both consecutive zeros G_i and consecutive ones B_i) of all lengths (≥1) in the sample stream should be counted and stored. The test is passed if the runs that occur (of lengths 1 through 6) are each within the corresponding interval specified in [18]. This must hold for both the zeros and ones (i.e., all 12 counts must lie in specified interval). For the purposes of this test, runs greater than 6 are considered to be of length 6.

5.3.4 Long Runs Test

A long run is defined to be a run of length 26 or more (of either zeros or ones). On the sample of 20,000 bits, the test is passed if there are no long runs.

Monobit Test	Poker Test	Runs Test		Long Run Test
		B1=2587	G1=2569	
		B2=1286	G2=1183	
T1=10,107	T2=22.1312	B3=588	G3=662	No long run
		B4=290	G4=311	
		B5=155	G5=177	
		B6+=147	G6+=151	
Passed	Passed	Passed		Passed

Tab. 3. Results of FIPS tests for one 20,000 bit record (B_i, G_i is the number of consecutive zeros and ones, respectively, of length i=1,2,3,4,5,6 in the tested bitstream).

We had split 1,000,000 bit XOR corrected TRNG records (for 4-SC TRNG@5V and 5-SC TGNG@5V) into 100 non- overlapped 20,000 bit records and tested by all FIPS 140-2 tests. All XOR corrected records passed it without problems for both TRNG structures. Typical outputs of FIPS tests for one 20,000 bit record are given in Tab. 3. The results of FIPS140-2 tests for 5-SC and 4-SC TRNGs performed for different supply voltages V_{CC} are shown in Tab. 4 and Tab. 5.

Voltage / Tests	3.0* [V]	3.1 [V]	3.2 [V]	3.5 [V]	4.0 [V]	4.5 [V]	5.0 [V]	5.5 [V]
Monobit	F	P	P	P	P	P	P	P
Poker	F	P	P	P	P	P	P	P
Run	F	P	P	P	P	P	P	P
Log runs	F	P	P	P	P	P	P	P

Tab. 4. FIPS140-2 test results for 5-SC TRNG (**F**-fail, P - pass, *-tested hardware is not working for this supply voltage).

Voltage / Tests	3.0* [V]	3.5 [V]	4.0 [V]	4.5 [V]	4.55 [V]	4.6 [V]	5.0 [V]	5.5 [V]
Monobit	F	P	F	F	P	P	P	P
Poker	F	F	F	F	P	P	P	P
Run	F	F	F	F	P	P	P	P
Log runs	F	P	P	P	P	P	P	P

Tab. 5. FIPS140 test results for 4-SC TRNG (**F** - fail, P - pass, *tested hardware is not working for this supply voltage).

6. Conclusions

In this paper we have described and evaluated the new robust chaos-based TRNG embedded in a switched-capacitor reconfigurable hardware. The circuit topology was optimized in order to extend the range of supply voltages. Experimental results confirmed that the new 5-SC TRNG works under a significantly larger supply voltage range than the previously proposed 4-SC TRNG. The proposed TRNG provides FIPS compliant quality of random data at up to 60 Kbit/s data rates.

Acknowledgements

This work has been done in the frame of the Slovak scientific projects VEGA 1/4054/07, VEGA 1/4088/07 of the Slovak Ministry of Education, and under COST 297 grant. This is an extended and improved version of the paper published in the Proceedings of 17th International Conference Radioelektronika 2007, April 24- 25, 2007, Brno, Czech Republic.

References

[1] MENEZES, J. A., OORSCHOT, P. C., VANSTONE, S. A. *Handbook of Applied Cryptography*. New York: CRC Press, 1997.

[2] EASTLAKE, D., CROCKER, S., SCHILLER, J. Randomness recommendations for security. *Request for Comments 1750*, December 1994, www.ietf.org/rfc/rfc1750.txt.

[3] PETRIE, C. S., CONNELLY, J. A. A noise-based IC random number generator for applications in cryptography. *IEEE Trans. Circuits and Systems I*, 2000, vol. 47, no. 5, pp.615-621.

[4] BUCCI, M., et al. A high-speed oscillator-based truly random number source for cryptographic applications on a smart card IC. *IEEE Transactions on Computers*, 2003, vol.52, no.4, pp.403-409.

[5] EPSTEIN, M., HARS, L., KRASINSKI, R., ROSNER, M., ZHENG, H. Design and implementation of a True Random Number Generator based on digital circuit artifacts. In *C.D. Walter, C. K. Koc, Ch. Paar (Eds.): CHES 2003, LNCS 2779*, Springer, Berlin, 2003, pp.152-165.

[6] FISCHER, V., DRUTAROVSKY, M. True Random Number Generator embedded in reconfigurable hardware. In *B.S. Kaliski, Jr., C.K. Koc, C. Paar (Eds.): Cryptographic Hardware and Embedded Systems, 4th International Workshop, CHES 2002.* Redwood Shores (California, USA), August 13-15, 2003, LNCS 2523, Springer, Berlin, 2003, pp.415-430.

[7] DRUTAROVSKY, M., FISCHER, V., SIMKA, M., CELLE, F. A Simple PLL-based True Random Number Generator for embedded digital systems. *Computing and Informatics*, 2004, vol. 23, no.5-6, pp. 501-516.

[8] Open Random Bit Generator, available at: *http://mywebpages.comcast.net/orb/index.html*.

[9] KENNEDY, M. P., ROVATTI, R., SETTI, G. (Eds.) *Chaotic Electronics in Telecommunications*. Boca Raton: CRC International Press, 2000.

[10] CALLEGARI, S., ROVATTI, R., SETTI, G. Embeddable ADC-Based True Random Number Generator for cryptographic applications exploiting nonlinear signal processing and chaos. *IEEE Trans. on Signal Processing*, 2005, vol. 53, no. 2, pp.793-805.

[11] BERNSTEIN, G. M., LIEBERMAN, M. A. Secure random number generation using chaotic circuits. *IEEE Transactions on Circuits and Systems*, 1990, vol.37, no.9, pp.1157-1164.

[12] STOJANOVSKI, T., PIHL, J., KONCAREV, L. Chaos-based Random Number Generators-Part II: Practical realization. *IEEE Transactions on Circuit and Systems-I: Fundamental Theory and Applications*, 2001, vol. 48, no.3, pp. 382-385.

[13] CALLEGARI, S., ROVATTI, R., SETTI, G. First direct implementation of a true random source on programmable hardware, *Int. J. Circ. Theor. Appl.,* 2005; 33:1-16.

[14] DRUTAROVSKY, M., BACA, M., GALAJDA, P. Chaos Based True Random Number Generator, *design entry to the International PSoC Design Contest, Cypress MicroSystems Inc.*, February 2004.

[15] *PSoC Mixed Signal Array Final Data Sheet, Document No. 38-12012 Rev.C,* August 28, 2003, pp.1-332, available at: http://www.cypressmicro.com.

[16] ESS, D. V. Understanding switched capacitor analog blocks. *Cypress Microsystems Application Note AN2041*, 2002, pp.1-16.

[17] DRUTAROVSKY, M., GALAJDA, P. Chaos-based True Random Number Generator embedded in a reconfigurable hardware. *Journal of Electrical Engineering*, 2006, vol. 57, no.4, pp.218-225.

[18] *NIST FIPS PUB 140-2: Security Requirements for Cryptographic Modules (2001),* available at: csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf.

[19] DAVIES, R. B. *Exclusive OR (XOR) and hardware random number generators*. February 28, 2002, pp.1-11, available at: http://www.robertnz.net/pdf/xor2.pdf.

About Authors...

Miloš DRUTAROVSKÝ was born in Prešov, Slovak Republic, in 1965. He received the Ing. (M.Sc.) degree in Radioelectronics and CSc. (Ph.D.) degree in Electronics from the Faculty of Electrical Engineering, Technical University of Košice, Slovak Republic, in 1988 and 1995, respectively. He defended his habilitation work Digital Signal Processors in Digital Signal Processing in 2000. He is currently working as an Associated Professor at the Department of Electronics and Multimedia Communications, Technical University of Košice. His current research interests include applied cryptography, digital signal processing, DSP and FPGA devices and algorithms for embedded cryptographic architectures.

Pavol GALAJDA was born in Košice, Slovak Republic, in 1963. He received the Ing. (M.Sc.) degree in Radioelectronics and CSc. (Ph.D.) degree in Electronics from the Faculty of Electrical Engineering, Technical University of Košice, Slovak Republic, in 1986 and 1995, respectively. He is currently working as an Associated Professor at the Department of Electronics and Multimedia Communications, Technical University of Košice. His research interest is in nonlinear circuits theory, spread-spectrum communication via chaotic synchronizations and modulations, software defined radio and electronics systems based on FPGA devices.