Ročník 2012 electroscope Číslo VI

Improved Structure of True Random Number Generator with Direct Amplification of Analog Noise

V. Kote[1, 2], V. Molata[1, 2], J. Jakovenko[1]
[1] Department of Microelectronics, Faculty of Electrical Engineering, CTU in Prague,
Technicka 2, Prague
[2] ST-Ericsson s.r.o., Pobrezni 620/3, Prague
E-mail: kotevlas@fel.cvut.cz, molatvla@fel.cvut.cz, jakovenk@fel.cvut.cz

Abstract:
Research in the sphere of random number generators is a dynamically evolving segment. The sequences of non-deterministic numbers with some specific probability distribution are especially required for the need of greater safety. This paper deals with the true random number generator (TRNG) that use direct amplification of analog noise for generation random number sequences. The true random number generators help to ensure the security of cryptographic and communication systems by the generation of different session keys. TRNGs can be also used in simulations of various physical and non-physical problems as so-called Monte Carlo technique. The proposed generator exploits a physical phenomenon – noise in semiconductor devices – that is purely random. Its structure is based on generic architecture. Noise source is created by noise amplifiers and core with bipolar transistors. For reduction of potential bias, the XOR corrector and the von Neumann corrector are implemented in the post-processing block. The proposed true random number generator is controlled by the microcontroller that also mediates communication TRNG with personal computer. Proposed structure was simulated and debugged using software model that was created in Verilog-A HDL language. Quality of output random number sequences was evaluated on the base of the Federal Information Processing Standards (FIPS) and the National Institute of Standards and Technologies (NIST) test suites. This work will serve as a base of following research of the reliable true random number generator that will be designed in CMOS technology and will be implemented in cryptographic applications for ensuring secure data transfers.

INTRODUCTION

In contemporary communication systems, unpredictable and nonrecurring numbers – random numbers – ensure the security of a variety of electronic transactions. Therefore, it is very important to generate a high-quality random numbers with specific features, for example, uniform distribution and high entropy. There exist two basic types of random number generators for random numbers generation. The first type – the pseudo-random number generators (PRNGs) – is based on computational algorithms and produces sequences of numbers that seem to be random. In other words, sequences of numbers generated by PRNGs approximate properties of random numbers sequences. These sequences are determined by the value of the initial state. There exists possibility that numbers in some sequence could be predicted. Therefore, the pseudo-random number generators are not suitable for use in cryptographic applications.

Devices called the true random number generators (TRNGs) are the other type of random number generators. These generators use randomness that appears in physical phenomena, for example, thermal noise generated by resistors, noise generated by semiconductors, jitter in ring oscillators or randomness that results from nuclear decay. There exists the third derived type of random number generators – the hybrid generators. This type allows combine the true random number generator with the pseudo-random number generator. TRNG generates random number that is used as the value of the PRNG initial state. Then, PRNG creates sequence of random numbers using computational algorithms. This idea is basic concept of the hybrid generators.

This work describes the true random number generator that uses direct amplification of analog noise for generation of random numbers sequences. Basic principle is shown in Fig. 1. This proposal modifies and improves structure of hardware-based true random number generator based on the presence Johnson noise in resistors that was presented in [1]. For evaluation quality of output random numbers sequences, the National Institute of Standards and Technologies (NIST) and the Federal Information Processing Standards (FIPS) test suites are used [2], [3].

The second part of this work deals with architecture of various types of TRNGs that are constructed from similar specific parts. Properties and function of each part are analysed and described in details. Structure and functionality of the proposed true random number generator are investigated in the third part. Software model of the proposed TRNG was created in Verilog-A HDL language for debugging of proposed structure. The fourth part is devoted to this topic. Software model and proposed true random number generator are evaluated by the NIST and the FIPS test suites in the next part. Last part not only summarizes this work, but also compares results of software model and the proposed TRNG reached during research of this topic.

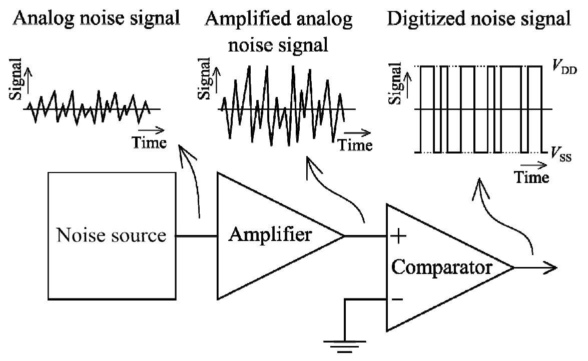

Fig. 1: Basic principle of TRNGs with direct amplification of analog nosie

GENERIC ARCHITECTURE

Every true random number generator follows generic architecture that was presented in [4] and in [5]. Generic architecture consists of four basic blocks – noise source, digitizer, post-processing block and output interface. The block diagram of generic architecture is depicted in Fig. 2.

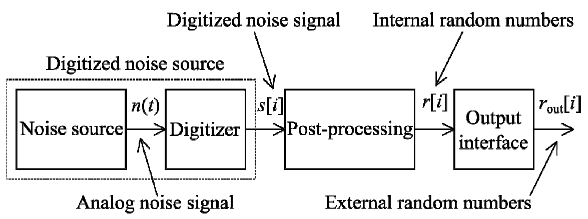

Fig. 2: The block diagram of generic architecture

The first block – noise source – is the unit that creates analog noise signal $n(t)$ on the base of some physical process with non-deterministic features. The ideal noise source generates signal with uniform probability distribution on a finite range. However, this source of randomness does not exist in the real world. Some correction techniques are usually used for approach real noise source properties to ideal noise source properties. TRNGs are implemented into some miniaturized devices and therefore it is important to use noise source which is possible to implement on chip for analog non-deterministic signal generating. Many noise sources of known TRNGs are based on jitter utilization [4] that arises in oscillators. Metastable state emerging in digital systems is also possible to use as noise source [6]. A considerable part of noise sources of published TRNG uses direct noise amplification that arises in semiconductor devices on p-n junction, on resistors immediately or in similar noise source [1], [7].

The most of contemporary systems needs for their work digital signals. Hence, the analog noise signal $n(t)$ have to be converted into digital form. Therefore, the second block – digitizer – is incorporated into the structure of the most of TRNGs. This block produces so-called digitized noise signal $s[i]$ and together with noise source creates digitized noise source.

Table 1: The XOR truth-table

X	*Y*	Output bit
0	0	0
0	1	1
1	0	1
1	1	0

The third block – post-processing – produces internal random numbers $r[i]$ and solves two problems. First, probability distribution of the raw random bits $s[i]$ is not purely uniform. There exists possibility that the statistical defects appear in purposed noise source or in digitizer. Therefore, it is necessary to compensate these defects. The other problem is that digitized noise signal $s[i]$ can have low entropy. The entropy per bit of internal random numbers $r[i]$ can be increased by the use of suitable post-processing algorithm. For example, the compression functions can be applied to digitized noise signal $s[i]$ as suitable post-processing algorithm. Consequence is that the rate of output stream that is lower but entropy of output signal increases. The inclusion of post-processing algorithms allows using the noise source with lower entropy per bit and also increasing the resistance against the tampering and the environmental changes. In post-processing block, cryptographic hash function or cyclic codes can be used but the most popular algorithms are a XOR corrector and a von Neumann corrector. The XOR corrector is the method that uses logic exclusive or (XOR). This operation can be described by the truth-table (Table 1), where X and Y denote random bits that can take values 0 or 1. The mean value of digitized noise signal $s[i]$ is equal to 1/2 in ideal TRNG. But the mean value of the digitized noise signal is not exactly 1/2 in real TRNG due to correlation between adjacent bits. Difference from the ideal mean value can be reduced using the XOR operation in post-processing block. If X and Y are independent random bits then the mean value $E(X \otimes Y)$ of the XOR operation between X and Y is calculated as

$$E(X \otimes Y) = \frac{1}{2} - 2\left[E(X) - \frac{1}{2}\right]\left[E(Y) - \frac{1}{2}\right], \qquad (1)$$

where E(X) and E(Y) are the mean values of independent random bits. But in the case of some dependence between adjacent bits, the XOR corrector also has its limitations. If correlation cor(X,Y)

between X and Y is not equal to 0 then the mean value of the XOR operation is expressed as

$$\begin{aligned} E(X \otimes Y) &= \frac{1}{2} - 2\left[E(X) - \frac{1}{2}\right] \\ &\left[E(X) - \frac{1}{2}\right] - 2 \cdot \mathrm{cor}(X,Y) \\ &\sqrt{E(X)[1-E(X)]E(Y)[1-E(Y)]}. \end{aligned} \tag{2}$$

Nonzero correlation between X and Y adds bias to internal random numbers $r[i]$. This property can be seen from the formula 2. The use of the XOR operation allows the simple mechanism implementing to improve properties of internal random numbers $r[i]$. In 1951, John von Neumann proposed often used post-processing technique for removing bias from digitized noise signal. Corrector based on von Neumann technique converts two input bits on one or none output bit. If input bit X is equal to 0 and input bit Y is equal to 1 then output bit is equal to one. The similar situation occurs when input bit X is 1 and input bit Y is 0. Then output bit is equal to zero. In case of equal input bits X and Y, corrector generates no bit. Function of the von Neumann corrector can be described using Table 2. The von Neumann corrector offers a simple way how to remove bias from digitized noise signal and to produce a balanced distribution of ones and zeros. Disadvantage of this corrector is a variable bitrate on output that can complicate some use in practice. The corrector generates an average of one bit for every six bits of digitized noise signal [1]. It means the significant reduction of the output bitrate and it can cause problems with fulfillment of the requirements.

Table 2: Function of the von Neumann corrector

X	Y	**Output bit**
0	0	none
0	1	1
1	0	0
1	1	none

The true random number generators are usually part of complex systems. Random numbers have to be converted into a suitable format for specific system. Therefore, output interface is incorporated into generic architecture. This block modifies format of internal random numbers $r[i]$ on required format of external random numbers $r_{\mathrm{out}}[i]$ that can be created according to communications protocols USB, standard RS232, etc.

GENERATOR DESCRIPTION

For research of the true random number generators with analog noise sources behavior, the TRNG with direct amplification of analog noise arising in semiconductor structures in p-n junction was developed. In this phase of research, proposed generator was created on printed circuit board (PCB) using of commercially available components. The structure of this proposal is shown in Fig. 3. This proposal is possible to divide into four blocks according to the generic architecture.

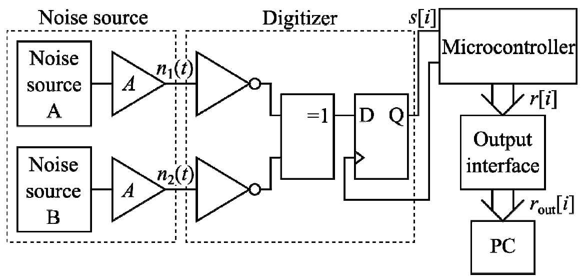

Fig. 3: Structure of proposed generator

Analog noise signals $n_1(t)$ and $n_2(t)$ of proposed generator are generated in noise sources block that is composed of two independent noise sources and of two amplifiers. Schematic diagram of whole block is depicted in Fig. 4. During development and testing, a lot of noise source types were used, for example, generally known connections with various operational amplifiers or instrumentation amplifier to amplify very weak signals. But, the best noise source for this application is simple connection with bipolar transistors. NPN-type of bipolar transistors with marking BC546B was selected. The reverse polarized p-n junction emitter-base of transistor Q_1 works as source of the avalanche noise. It means that random spikes arise in current that flows through the reverse polarized p-n junction. This current with random spikes flows into the base of the second bipolar transistor Q_2 that has unstabilized operating point. It causes that noise signal appears on the collector of bipolar transistor Q_2 and it is amplified in next part of circuit – in the amplifiers of the noise. The output analog noise signals $n_1(t)$ and $n_2(t)$ must have sufficient amplitude to be able to drive the input circuits of the digitizer. Therefore, the amplifiers are incorporated. These amplifiers are created from the same bipolar transistors as noise sources. Modification of connection the common emitter was used. The decoupling capacitors were added between the amplifiers inputs and noise sources outputs due to DC components of the signals separation.

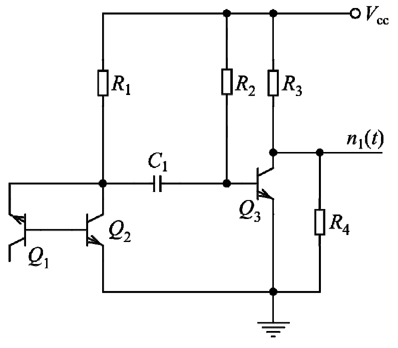

Fig. 4: Schematic diagram of one independent noise source with amplifier of analog noise

Digitizer processes analog noise signal $n_1(t)$ and $n_2(t)$ and creates digitized noise signal $s[i]$. The input part of digitizer is composed of two inverters. The analog noise signals $n_1(t)$ and $n_2(t)$ are applied to the input of these inverters and make random flipping of the inverter. Thus, the inverter generates random signals with two voltage levels that can be marked as digital random signals. The next part of digitizer is the XOR gate that combines both digital random signals. Output part of the digitizer is created by D-type flip-flop. This circuit takes samples of the signal produced by the XOR gate and creates digitized noise signal $s[i]$. The clock signal for D-type flip-flop is generated in the post-processing block.

The base of post-processing block is the microcontroller. This block controls the function of the whole device. The task of post-processing block is to reduce potential bias in digitized noise signal $s[i]$ and produces internal random numbers $r[i]$. The microcontroller reads digitized noise signal $s[i]$ and processes it by implemented algorithms. Therefore, the XOR corrector, the von Neumann corrector and direct reading of digitized noise signal $s[i]$ are implemented in the microcontroller. One of three methods can be selected by the encoding switch. The first two methods are described in previous section. The third method – direct reading of digitized noise signal $s[i]$ – reads samples of digitized noise signal $s[i]$ and send them direct on output of post-processing block. This method is implemented due to possibility of comparison of previous two methods. The very well-equipped 8-bit microcontroller PIC182550-I/SP with high performance, enhanced flash and other accessories from Microchip Technology, Inc. was used. Quartz crystal resonator with frequency 4 MHz was selected for this application. Communication with the output interface is one of tasks of the post-processing. The serial type of communication was selected. Therefore, the Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART) module in the microcontroller was used. So, pins RX and TX are directly connected to the output interface.

The output interface mediates communication between the post-processing block and a personal computer. Communication standard RS-232 with the asynchronous transmission was chosen. External random numbers $r_{\text{out}}[i]$ are sent as a series of the consecutive bits. Tera Term software was used for the reading of external random numbers $r_{\text{out}}[i]$ that can be saved to the text file. And then, this text file can be tested by the statistical test suites. The base of the output interface is the integrated circuit MAX232I from Texas Instruments Incorporated. It contains two drivers, two receivers and capacitive voltage generator creating voltage levels described in standard RS-232. Receivers convert RS-232 input voltage levels to TTL output voltage levels. The drivers convert TTL input voltage levels to RS-232 output voltage levels.

SOFTWARE MODEL

Testing of hardware prototypes is very expensive and lengthy because each prototype must be manufactured and each of its modification means the manufacturing of the new prototype. Therefore, software model of the proposed true random number generator was created and was used during design of the real device. This software model allows functions and behavior simulations of this device. Based on software model, it was possible to optimize the proposed device. It meant that the number of manufactured prototypes was dramatically reduced.

Software model of the proposed true random number generator with direct amplification of analog noise was created in Verilog-A HDL for debugging of real device. This type of TRNG is suitable for realization on printed circuit board (PCB) or for realization in integrated form. Software model was created strictly on the base of the proposed structure that is described in previous section Generator description and is depicted in Fig. 3. Individual parts were created as modules in Verilog-A language. For generation of analog noise signal $n(t)$, the pseudo-random number generator was used. This PRNG is function that is implemented in Verilog-A. All modules were connected in Cadence Schematic XL and the whole model was simulated in Cadence Spectre Circuit Simulator. In the post-processing block, the same algorithms as in the real device are implemented. Every generated external random signals $r_{\text{out}}[i]$ were read and stored in the appropriate text files. Outputs of this model were tested by statistical test suites NIST and FIPS.

RESULTS

For evaluation of output random numbers sequences – external random numbers $r_{\text{out}}[i]$, two types of statistical test suites were used. The task was to test the output random numbers sequences using the NIST test suite and using the FIPS test suite. Details of this statistical test suites are in [2] and in [3]. The FIPS test suite is package consisting of 4 statistical tests. Each test checks a property of the tested sequence. Achieved value is compared with the expected values for a random sequence. The tested sequence must have the length of 20000 bits to be able to be tested. First, software model of the proposed true random number generator was tested using FIPS test suite. Three output random numbers sequences were generated. The first sequence was generated without the use of any correctors, the second by the use of the von Neumann corrector and the third by the use of the XOR corrector. All generated random numbers sequences fulfilled all four tests (Table 3). Any bias was not detected by the FIPS test suite.

Table 3: FIPS test suite results of software model

Test	Without corrector	XOR	Von Neumann
Monobit	PASSED	PASSED	PASSED
Poker	PASSED	PASSED	PASSED
Runs	PASSED	PASSED	PASSED
Long runs	PASSED	PASSED	PASSED

By the realized true random numbers generator, three output random numbers sequences were generated. All generated random numbers sequences fulfilled all four tests (Table 4) and any bias was not detected by the FIPS test suite.

Table 4: FIPS test suite results of the realized true random numbers generator

Test	Without corrector	XOR	Von Neumann
Monobit	PASSED	PASSED	PASSED
Poker	PASSED	PASSED	PASSED
Runs	PASSED	PASSED	PASSED
Long runs	PASSED	PASSED	PASSED

The NIST test suite is package that is composed of 15 statistical tests. These tests are designed for the testing of randomness. So, a variety of the different types of non-randomness are searched in an arbitrarily long sequence of bits. This sequence can be generated by some TRNGs or by some pseudo-random number generators. The output random number sequence generated by software model without any corrector was tested by the NIST test suite. Some tests failed because bias was found. The bias was detected by the Frequency test within a block, the Runs test, the Overlapping test and by the Approximate entropy test. The output random numbers sequence generated by software model by the use of the XOR corrector was tested and bias was also found. Some tests failed. In this case, the XOR corrector did not remove bias and thus it is not appropriate to use this corrector. The output random number sequence generated by software model by the use of the von Neumann corrector was tested and the most tests were fulfilled. List of results is in Table 5. Some tests were not applicable, these tests are marked *NA*. It is possible to say that bias was removed to a large extent in this case. For this software model, it is suitable to use the von Neumann corrector for the reduction of bias.

Three sequences of random number sequences were generated by the realized generator. The first sequence was generated without the corrector. Some tests from NIST test suite failed because bias was found in this sequence. As written above, the NIST test suite is very strict. Therefore, if there is small bias there, some statistical tests from the NIST test suite can fail. The noise source works with very small signals. So, there is a possibility of interference of these signals. Therefore, the noise source must be designed to minimize the possibility of interference. The second random number sequence generated by the use of the XOR corrector was tested by NIST test suite and bias was also found and some test failed. List of results is in Table 6. The XOR corrector did not remove bias and thus it is not appropriate to use this corrector. This situation is similar to the case of the software model. The third sequence generated by the use of the von Neumann corrector was also tested by NIST test suite and most statistical tests were fulfilled. Therefore, the von Neumann corrector is suitable for this realized generator because bias was removed to a large extent.

Table 5: NIST test suite results of software model

Test	Without corrector	XOR	Von Neumann
Monobit	PASSED	FAILED	PASSED
Frequency	FAILED	FAILED	PASSED
Runs	FAILED	PASSED	FAILED
Long runs	PASSED	FAILED	PASSED
Matrix	PASSED	PASSED	PASSED
DFT	PASSED	PASSED	PASSED
Non-overlap.	FAILED	FAILED	FAILED
Overlap.	FAILED	FAILED	PASSED
Universal	PASSED	PASSED	*NA*
Linear complexity	PASSED	PASSED	PASSED
Serial	FAILED	PASSED	PASSED
Entropy	FAILED	FAILED	PASSED
Cum. sums	PASSED	FAILED	PASSED
Rand. excursion	PASSED	*NA*	*NA*
Rand. exc. variant	PASSED	*NA*	*NA*

Table 6: NIST test suite results of the realized true random numbers generator

Test	Without corrector	XOR	Von Neumann
Monobit	FAILED	FAILED	PASSED
Frequency	PASSED	PASSED	PASSED
Runs	FAILED	FAILED	FAILED
Long runs	PASSED	PASSED	PASSED
Matrix	PASSED	PASSED	PASSED
DFT	PASSED	PASSED	PASSED
Non-overlap.	FAILED	FAILED	PASSED
Overlap.	PASSED	PASSED	PASSED
Universal	PASSED	PASSED	PASSED
Linear complexity	PASSED	PASSED	PASSED
Serial	PASSED	PASSED	PASSED
Entropy	PASSED	PASSED	PASSED
Cum. sums	FAILED	FAILED	PASSED
Rand. excursion	*NA*	*NA*	PASSED
Rand. exc. variant	*NA*	*NA*	PASSED

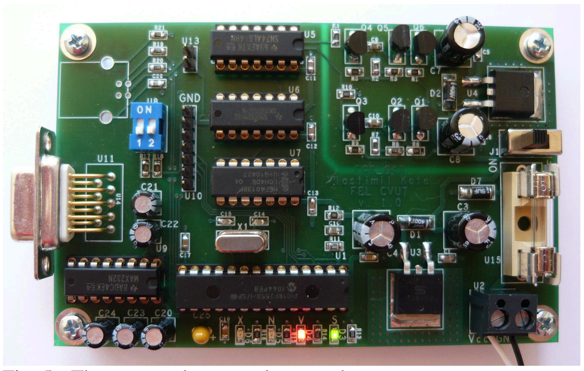

Fig. 5: The proposed true random number generator

CONCLUSIONS

Improved structure of the true random number generator with direct amplification of analog noise was presented in this paper. The aim of this work was proposal and realization of the random number generator with direct amplification of analog noise that creates stable base for future research in the sphere of the true random number generators. Proposal of this structure builds on published works [1], [4], [6]. An integral part of this proposal was creation of software model for debugging and optimization of generator structure. The proposed generator (Fig. 5.) was realized on the printed circuit board and it consists of four blocks according to generic architecture. The first block is the noise source block that contains two independent noise sources with amplifiers of analog noise. The second block is the digitizer that is composed of two inverters, one XOR gate and one D-type flip-flop. As the main component of the post-processing block the microcontroller was used. The XOR corrector, the von Neumann corrector and the method without the use of any correctors were implemented in the microcontroller that was programmed in C programming language. Modifications of the created system without the need for production of new prototype are allowed by inclusion of the microcontroller that is possible to reprogram. The fourth block is the output interface that modifies the signals between the generator and the personal computer to the desired format. The realized generator and its software model were tested by the FIPS test suite and the NIST test suite. The results are consistent with the experience gained through the modeling. Bias was observed in the sequences that were generated without the corrector and with the XOR corrector. The von Neumann corrector eliminated the bias and is suitable for future use.

ACKNOWLEDGMENTS

This work is part of the GAČR project No. 02/09/160, CTU SGS grant No. SGS11/156/OHK3/3T/13 and Ministry of the Interior grant No. VG20102015015.

REFERENCES

[1] B. Jun and P. Kocher, The Intel Random Number Generator [online]. Cryptography Research, Inc. White Paper Prepared For Intel Corporation, 1999 [cit. 2012-09-10]. Available from WWW: <http://www. cryptography.com/ resources/whitepapers/IntelRNG.pdf>

[2] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray, and S. Vo. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications [online]. NIST Special Publication 800-22 [cit. 2012-10-10]. Available from WWW: <http://csrc.nist.gov/ rng/SP800-22b.pdf>

[3] Federal Information Processing Standards Publication 140-2. Security Requirements for Cryptographic Modules [online]. National Institute of Standards and Technology, 2001[cit. 2012-10-10]. Avalible from WWW: <http:// csrc.nist.gov/publications/ps/ps140-2/ps1402.pdf >.

[4] D. Schellekens, B. Preneel, I. Verbauwhede, "FPGA Vendor Agnostic True Random Number Generator," IEEE, pp. 1 – 6, 2006.

[5] G. Bucci and R. Luzzi, "Design of Testable Random Bit Generators," in Cryptographic Hardware and Embedded Systems – CHES 2005, 7th International Workshop, Edinburgh, Scotland, August 29 – September 1, 2005, Proceedings, ser. Lecture Notes in Computer Science, J. R. Rao and B. Sunar, Eds., vol. 3659. pp. 147 – 156, Springer, 2005.

[6] V. B. Suresh and W. P. Burleson, "Entropy extraction in metastability-based TRNG," Hardware-Oriented Security and Trust (HOST), 2010 IEEE International Symposium, pp. 135 – 140, IEEE, 2010.

[7] Y. Yamanashi and N. Yoshikawa, "Superconductive Random Number Generator Using Thermal Noises in SFQ Circuits," IEEE Transactions on Applied Superconductivity, Vol. 19, No. 3, pp. 630 – 633, 2009.