Ročník 2014 electroscope Číslo III

Enhanced Generic Architecture for Safety Increase of True Random Number Generators

V. Kotě[1, 2], V. Molata[1, 2], J. Jakovenko[1]
[1] Department of Microelectronics, Faculty of Electrical Engineering, Czech Technical University in Prague, Technická 2, Prague
[2] STMicroelectronics Design and Application, s.r.o., Pobřežní 620/3, Prague
E-mail: kotevlas@fel.cvut.cz, molatvla@fel.cvut.cz, jakovenk@fel.cvut.cz

Abstract:
Conventionally used generic architecture of true random number generators does not allow testing of random numbers during their generation. This paper introduces an extension of the conventionally used generic architecture and describes mechanisms that can be implemented in new blocks and ensures safety increase of true random number generators. Objective of new architecture extension is detection of deliberate malicious attacks that are directed against noise source and revelation of a significant decrease of the approximate entropy in the subsequences of generated random number sequences. The enhanced generic architecture has been implemented into known software model. Obtained results show that described mechanisms allow increasing quality of the generated random numbers sequences.

INTRODUCTION

Random number generators (RNGs) are used for a variety of purposes. RNGs help to ensure the security of cryptographic and communication systems by generating different session or encryption keys. They are commonly used in CAD (computer aided design) programs for modeling and simulating of various physical processes. For example, simulators of electrical circuits and elements, semiconductor structures or simulators of physical fields that need sequences of independent random numbers for some types of simulations. Specifically, the Monte Carlo simulation method is based on generating high-quality random number sequences. RNGs are used during a selection of random samples from larger data sets and also appear in commercial applications such as various lottery games and gambling slot machines. RNGs are also very often used for measurement in acoustics as a source of random quantities. And not for measurements only, sequences of random numbers also appear in the arts, in music or in literature. Functions of many electronics devices and systems depend on generating of high-quality random number sequences.

For higher safety, devices based on a physical source of randomness are incorporated into structures of modern systems instead of pseudo random number generators (PRNGs) that are based on computational algorithms. Generated number sequences by PRNGs only approximate properties of sequences of true random numbers. These sequences are generated by devices that are commonly called true random number generators (TRNGs) and use physical phenomena in which appear a random behavior, for example, thermal noise generated by resistors, noises generated in semiconductor structures [1], especially flicker noise and thermal noise generated by complementary metal oxide semiconductor (CMOS) structures or the jitter that arises in ring oscillators. However, magnitudes of these sources are very small. Therefore, designed TRNGs are very sensitive to disturbance. The basic function of TRNGs can be affected by sources of undesired deterministic phenomena.

Any failure of the TRNG leads to total failure of the whole system. In the past, different variations of true random number generators were only presented in specialized cryptographic devices such as hardware security modules, cryptographic accelerators or smart carts. With the advent of modern fast computers and more sophisticated programs, the need for available high-quality random number sequences has increased. Gradually, TRNGs have been incorporated into more complex systems such as motherboard chipsets [2], platforms or processors [3].

Generally, true random number generators generate sequences of random numbers based on three different techniques. The first technique is the direct amplification of an analog noise [4], [5]. The second uses the phenomenon commonly called the jitter [6], [7] that arises in all digital clocked circuits and it is an uncertainty in the exact timing of the rising edge or the falling edge in a square wave signal. The third technique is based on metastable states that arise in digital circuits [8], [9]. Majority of the true random number generators follows a generic architecture that was presented in [6] and [10].

Safety of contemporary systems is dependent on the unpredictability of random numbers. Safety can be violated, if there is possibility that the value of generated numbers can be detected or affected by a deliberate attack. Therefore, it is important that the random number sequences are generated by TRNGs. However, these generated sequences are not tested and qualified by statistical test suites during

generation. It can be an obstacle for some applications of TRNGs. Above mentioned disadvantage is eliminated by new design of TRNG that allows testing of random numbers during their generation.

For testing of most TRNGs and PRNGs, two well-known statistical test suites are used. The first test suite is FIPS PUB 140-2, Security Requirements for Cryptographic Modules [11] (the FIPS test suite) and the second is A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications [12] (the NIST test suite) that is very strict and is suitable for applications, where the strongest requirements are needed. A necessary condition for an implementation of a TRNG into modern systems is positive results of these test suites.

ENHANCED GENERIC ARCHITECTURE

Each true random number generator can be attacked by deliberate malicious attacks for a purpose to affect the output random number sequences. Then, these sequences lose random properties. Consequently, such attack causes a failure of security of the whole system. Activities that the system performs are threatened. Therefore, a mechanism that is able to detect deliberated malicious attacks must be incorporated into the architecture of the designed true random number generators.

Except of the deliberate attacks, the loss of randomness is usually caused by deterministic noises, for example by deterministic distortion of power supply or of chip substrate, by regular fluctuations in temperature, by defects that are created during manufacturing of the chip of by other nonrandom processes. Therefore, it is also necessary to incorporate a block that is able to detect these deviations and to stop number generation.

Partial extension of architecture of TRNGs was published in [10] but this paper present new complex extension of the generic architecture with description of used mechanisms.

Conventional generic architecture

The previously published true random number generators follow conventional architecture (Fig. 1) that was published in [6], [10] and does not allow testing of random numbers during their generation. A noise source and a digitizer form a digitized noise source. The noise source produces the analog noise signal $n(t)$ with non-deterministic features. The digitized noise signal $s[i]$ arises by sampling in the digitizer.

Discrepancies in the digitized noise signal $s[i]$ that arise in the noise source and in the digitizer are processed and corrected in the post-processing block. Probability distribution of the digitized noise signal $s[i]$ may not be purely uniform. Therefore, statistical defect in generated sequences are compensated in this block. In the post-processing block, some types of lossless compression algorithms are often implemented due to increase randomness of generated sequences. The post-processing block produces the internal random numbers $r[i]$. Then in the output interface, the internal random numbers $r[i]$ are transformed into required data format of the external random numbers $r_{out}[i]$.

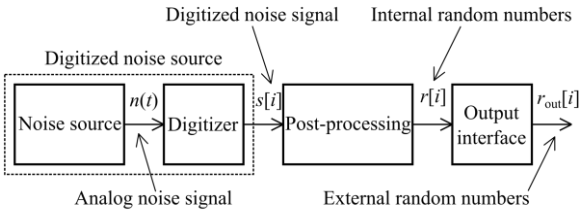

Fig. 1: The conventional generic architecture [6], [10]

Extension of conventional architecture

After detection of the attack, an implemented mechanism creates a notification that stops the output stream of random numbers. The notification is sent to the internal blocks – to the post-processing block and to the output interface. The designed TRNGs are able to notify the master control block by a message about the deliberate attack. Thus, the notification is available at output pins of the designed TRNGs.

The attack detector works with the analog noise signal $n(t)$. This is connected between the noise source and the digitizer and detects failures of the noise source. It is possible to consider a case that the attack affects the analog noise signal $n(t)$. This disturbance of the analog noise signal $n(t)$ cannot be detected after digitalization. Therefore, the attack detector is incorporated behind the noise source. The analog noise signal $n(t)$ has known parameters but they will be affected during the attack. Thus, the detector is able to detect the malicious attack and creates a notification about the attack.

The generated external random numbers $r_{out}[i]$ must be really unpredictable. In other words, the generated external random numbers $r_{out}[i]$ must meet the requirements of the statistical test suites in all cases. It means that they must have sufficient entropy per bit. The entropy is an important notion in the information theory. It expresses the degree of unpredictability. For example, the set of symbols is assumed. Symbols come independently and are consecutive. Maximum of the entropy occurs, when there is no way to predict the following symbol. The entropy is zero, if it is possible to determine with certainty the following symbol. The bases of the entropy were described by Claude E. Shannon in [13]. The Shannon entropy is defined as information which is produced by a process.

As already mentioned, the entropy per bit of the external random numbers $r_{out}[i]$ can decrease in some cases. This can result in loss of randomness in the generated number sequences. Therefore, it is appropriate to map the level of the entropy per bit and in a significant drop of the entropy per bit to create a message. For the design of new block which is able to detect low entropy per bit, the following consideration is used. If the entropy per bit of the digitized noise signal $s[i]$ is sufficient, then the entropy per bit of the internal random numbers $r[i]$ is also sufficient. Unpredictability of the output number sequences is guaranteed when the internal random numbers $r[i]$ have the higher entropy per bit then a specified minimum entropy limit. It means that if the entropy per bit of the digitized noise signal $s[i]$ is sufficient, then the generated external random numbers $r_{out}[i]$ are really unpredictable.

A low entropy detector is device that is able to estimate the entropy per bit of the relevant generated signal. In this case, the low entropy detector is incorporated between the digitizer and the post-processing block and processes the digitized noise signal $s[i]$. If the entropy per bit decreases under a certain level, then a notification is created that draw attention to the low level of the entropy per bit. This notification is sent to the post-processing block and to the output interface and is also available at output pins of the designed TRNGs for the master control block. The certain level of the sufficient entropy per bit – the low entropy threshold – is determined on the basis of results of the successful completion statistical test suites.

The enhanced generic architecture with the tunable noise source is depicted in Fig. 2. The considered noise sources can contain tunable circuits. Thus, it allows changing parameters on the basis of requirement that is created by the low entropy detector. The low entropy detector must be able to provide information about the state of the digitized noise signal $s[i]$.

PROTECTIVE MECHANISMS

Currently, modern communication and cryptographic systems face attacks every day that threaten security and cause problems in life. Therefore, the new enhanced generic architecture is proposed and contains mechanisms that are able to detect the deliberate malicious attacks and the low entropy per bit of the generated random numbers. This chapter describes implementation of defined protective mechanisms.

Attack detector

The attack detector is based on a model when an offensive signal $a(t)$ is superimposed to the analog noise signal $n(t)$. This model is shown in Fig. 3. In normal conditions, the system produces a probability distribution that is changed during deliberate malicious attacks. In order to detect attacks, the probability distribution of the analog noise signal $n(t)$ must be considered because any changes in the probability distribution of the digitized noise signal $s[i]$ are unobservable. This type of attack may be considered the side channel attack.

A change of the probability distribution of the analog noise signal $n(t)$ is caused by the offensive signal $a(t)$. But, one crucial condition must be valid. The standard deviation of the analog noise signal probability distribution must be smaller than an amplitude of the offensive signal $a(t)$. This condition allows detecting deliberate malicious attacks. A similar approach is used in [10] where a detector is based on counting the number of samples falling beyond two fixed thresholds.

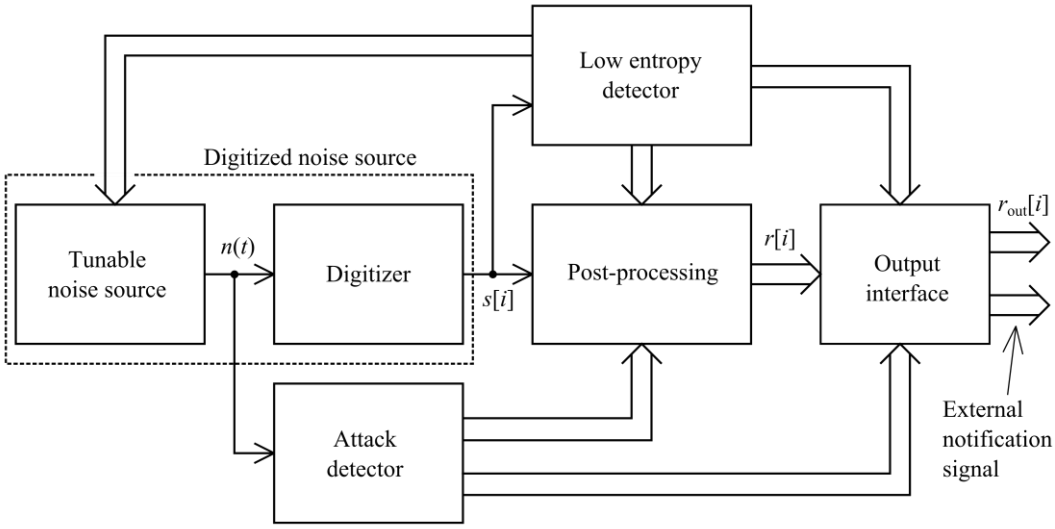

Fig. 2: The enhanced generic architecture with the tunable noise source

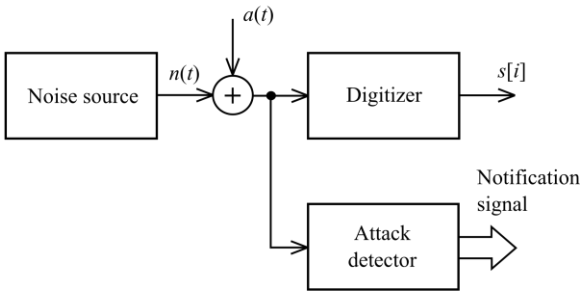

Fig. 3: Model of the malicious attack

Low entropy detector

The low entropy detector must determine the entropy of the number sequences during their generation. But, it is not an easy task. The definition of the Shannon entropy is based on the knowledge of the whole number sequence. Therefore, it is not possible to determine the value of the entropy directly from the Shannon definition but a way must be found how to estimate the entropy of generated sequences.

The new block – the low entropy detector – estimates the approximate entropy of binary sequences that the digitized noise source produces. Thus, the low entropy detector processes the digitized noise signal $s[i]$ that is digital and is only composed of logic ones and logic zeros. Under this assumption, an alphabet $\boldsymbol{A}$ can be introduced and $\boldsymbol{A}$ = {0, 1}. Then, the probability distributions are $\Pr(s[i]=1) = p_1$ and $\Pr(s[i]=0) = p_0 = (1-p_1)$. According to the usual convention, it is defined that $0 \cdot \log_2 0 = 0$. At this time, it is possible to use the definition of the Shannon entropy and obtain a relationship between the entropy H and the probability distribution of the digitized noise signal $s[i]$ when $s[i]=1$. So

$$\begin{aligned} H(p_1) &= -\sum_{\upsilon \in \boldsymbol{A}} p_\upsilon \log_2 p_\upsilon = \\ &- p_1 \log_2 p_1 - p_0 \log_2 p_0 = \\ &- p_1 \log_2 p_1 - (1-p_1)\log_2 (1-p_1). \end{aligned} \tag{1}$$

As can be seen from the formula 1, it is not possible to determine the exact value of the entropy of random number sequences that are generating. This is due to the fact that the probability distribution p_1 can be varied during generation. Therefore in this case, an approximation of the entropy is determined from consecutive subsequences with fixed length. The subsequence is defined as $g[j] = s[i+j]$ where $0 \leq j \leq m-1$ and m is number of bits of the subsequence. According to formula 1, periodically repeated sequences of ones and zeros have the maximal value of the entropy. However, generation of these sequences is absolutely impermissible. Thus, the approximate entropy of subsequences must be computed more complexly. For this reason, a method of the approximately entropy estimation that was published in [14] is adopted. Firstly, it is necessary to define a notion the binary derivate of the binary subsequence $d[j]$. This notion was established in [15] and is defined as

$$d[j] = \begin{cases} 1 & \text{if } g[j] \neq g[j+1], \\ 0 & \text{if } g[j] = g[j+1]. \end{cases} \tag{2}$$

Marking $d^l[j]$ represents an application of $d[j]$ l times. In other words, l is order of the binary derivate. By the application of the binary derivate, it is possible to detect periodically recurring patterns in the subsequence $g[j]$.

Using formula 1 it is possible to compute the entropy of each binary derivate of the binary subsequence $g[j]$ and mark as $H(p_{1l})$ where p_{1l} is the probability distribution and $p_{1l} = \Pr(d^l[j]=1)$. For estimation of the approximate entropy H_{approx}, mechanism using the weighting method was especially developed and was presented in [14]. Thus the approximate entropy H_{approx} can be calculated as

$$H_{\text{approx}} = \frac{1}{\sum_{k=0}^{n-1} w[k]} \left(\sum_{k=0}^{n-1} H(p_{1k}) \cdot w[k] \right) \tag{3}$$

where $w[k]$ is a weighting function and n is number of binary derivatives.

For application in the true random number generators, the suitable weighting function is

$$w[k] = 2^k \tag{4}$$

and this method is called as power weighting. Every order of the derivate has own weight and every binary derivative has certain value of the Shannon entropy according to formula 1. Higher order of the binary derivate has higher weight. Therefore it is possible to eliminate subsequences with periodic pattern.

The highest binary derivate is not used for computing of the approximate entropy H_{approx} because its contribution is not essential. Therefore the first n-2 binary derivatives of the subsequence $g[j]$ are used for computation of a weighted average of the entropy as

$$H_{\text{approx}} = \frac{1}{\sum_{k=0}^{n-2} 2^k} \left(\sum_{k=0}^{n-2} H(p_{1k}) \cdot 2^k \right) \tag{5}$$

where p_{1k} is the probability distribution and $p_{1k} = \Pr(d^k[j]=1)$.

The formula that is suitable for computation of the approximate entropy H_{approx} is obtained by substituting formulas 1 into equation 5. Then the partial sum of the geometrical series is applied and

the approximate entropy H_{approx} of the binary subsequence is

$$H_{approx} = \frac{1}{2^{n-1}-1}\sum_{k=0}^{n-2}\left(-p_{1k}\log_2 p_{1k} - \right. \\ \left. -\left(1-p_{1k}\right)\log_2\left(1-p_{1k}\right)\right)\cdot 2^k. \quad (6)$$

As written above, the thresholding method is used in the low entropy detector. Therefore, a threshold of the minimal approximate entropy of the binary subsequence is set. On this basis, the low entropy detector decides whether the subsequence is valid or, conversely, will be discarded. Disadvantage of this method of the approximate entropy computation is quadratic time requirements.

ACHIEVED RESULTS

The new blocks that are described above have been implemented into the software model of the TRNG with direct amplification of an analog noise that was published in [5]. Using the extended software model, new random number sequences have been generated and tested by the described statistical test suites (FIPS [11] and NIST [12]).

The FIPS test suite is composed of 4 statistical tests and each test checks a property of the tested generated binary sequence. Achieved values are compared with the expected values for a random sequence that are specified by this standard. The tested sequence must have the length of 20000 bits.

The NIST test suite is a package consisting of 15 statistical tests. A variety of the different types of non-randomness are searched in an arbitrarily long sequence of generated values. This test suite is intended for testing TRNGs or pseudo-random number generators.

New obtained results were compared with the published results. Let us consider the results obtained using the Von Neumann corrector. Clear comparison of the NIST tests results is shown in Table 1. All FIPS tests were passed. Therefore in Table 1, only the NIST tests are shown with an important parameter *P-value*. This important parameter indicates success of the relevant test and must be higher than 0.01. The results with symbol * are an arithmetic average of result set because some tests from the NIST test suite are composed of some subtests. Result of each type is successful only if all subtests are successful. For testing, the random sequences with the length of 325000 bits were used. Some test could not be performed due of the insufficient number of random numbers. Results of these tests are marked as *NA*.

By the Approximate Entropy Test, it is possible to determine the entropy of the generated sequence. The entropy of the published results is very high and its value is 0.9976 bits. However, the entropy of the new random sequence is 0.9998 bits. This value is higher because the subsequences with the low approximate entropy were discarded. For clarity, achieved values are listed in Table 2.

Table 2: Achieved values of the entropy

Architecture	**Value of the entropy (bit)**
Conventional	0.9976
Enhanced	0.9998

The obtained results show that the digitized noise signal $s[i]$ can contain binary subsequences with the low entropy. But addition of the new described blocks, it is possible to prevent the spread into the output random number sequences and improve the quality of the random numbers. In other words, addition of the new blocks helps to increase safety of modern electronics systems.

Table 1: Results obtained by the NIST tests

	Conventional architecture		***Enhanced architecture***	
	P-value	***Result***	***P-value***	***Result***
Monobit	0.0631	SUCCESS	0.9368	SUCCESS
Frequency within block	0.4143	SUCCESS	0.4565	SUCCESS
Runs	0.0000	FAILED	0.6344	SUCCESS
Longest runs	0.0823	SUCCESS	0.7031	SUCCESS
Binary matrix rank	0.5772	SUCCESS	0.2850	SUCCESS
DFT	0.1002	SUCCESS	0.8078	SUCCESS
Non-overlapping template	0.4593*	FAILED	0.4902*	FAILED
Overlapping template	0.0105	SUCCESS	0.7012	SUCCESS
Universal statistical	*NA*	*NA*	*NA*	*NA*
Linear complexity	0.4732	SUCCESS	0.1564	SUCCESS
Serial	0.4417*	SUCCESS	0.9310*	SUCCESS
Approximate entropy	0.0788	SUCCESS	0.8473	SUCCESS
Cumulative sums	0.0907*	SUCCESS	0.9441*	SUCCESS
Random excursions	*NA*	*NA*	*NA*	*NA*
Random excursions variant	*NA*	*NA*	*NA*	*NA*

CONCLUSIONS

In this paper, the new structure of the true random number generators that is extended with the protective mechanism has been shown and described. Implementation of the above mentioned mechanisms helps to increase safety of modern electronics and communication systems. The conventional generic architecture has been enhanced by the attack and the low entropy detectors.

The attack detector is able to detect the deliberate malicious attack on the base of the forcibly modified probability distribution of the noise source. The low entropy detector is able to discarded binary subsequences whose the approximate entropy is too low. Thus, the entropy of the output random numbers increases. For the entropy detector the approximate entropy algorithm is used and is able to detect not only sequences with low entropy but also sequences with periodically recurring patterns for that the Shannon entropy is maximal.

For comparison of achieved results, the NIST test suite has been used. The achieved results show that quality of the generated random numbers increases after extension of the conventional architecture by the attack detector and the low entropy detector.

ACKNOWLEDGMENTS

This work is part of the CTU SGS grant No. SGS14/195/OHK3/3T/13 (MiNa).

REFERENCES

[1] Kotě, V., Nápravník, T., Molata, V. and Jakovenko, J. Structure, Modeling and Realization of True Random Number Generator with Analog Noise Amplification. In: *Proceedings of Electronic Devices and Systems EDS 2012*. Brno: VUT v Brně, FEKT, 2012, vol. 1, p. 145-150. ISBN 978-80-214-4539-0.

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

[3] *Evaluation of VIA C3 Nehemiah Random Number Generator*. Cryptography Research, Inc. White Paper Prepared For VIA Technologies, 2003. Available from WWW: <http://www.cryptography.com/public/pdf/VIA rng.pdf>.

[4] Eberlein, M. and Abu Bakar R. An Integrated Channel Noise-based True Random Number Generator. *IEEE*. 2007, p. 391–134.

[5] Kotě, V., Molata, V. and Jakovenko, J. Improved Structure of True Random Number Generator with Direct Amplification of Analog Noise. *ElectroScope* [online]. 2012, vol. 2012, no. VI, ISSN 1802-4564. Available from WWW: <http://147.228.94.30/images/PDF/Rocnik2012/Cislo6_2012/r6c6c1.pdf>.

[6] Schellekens, D., Preneel, B. and Verbauwhede I. FPGA Vendor Agnostic True Random Number Generator. *IEEE*, 2006.

[7] Guler, U. and Ergun, S. A High Speed IC Random Number Generator Based on Phase Noise in Ring Oscillators. *IEEE*, 2010.

[8] Tokunaga, C., Blaauw, D. and Mudge, T. True Random Number Generator With a Metastability-Based Quality Control. *IEEE, Journal of Solid-State Circuits*. Vol. 43, no. 1, 2008, p. 78–85.

[9] Srinivasan, S., Mathew, S., Erraguntla, V. and Krishnamurthy, R. A 4Gbps 0.57pJ/bit Process-Voltage-Temperature Variation Tolerant All-Digital True Random Number Generator in 45nm CMOS. In: *22nd International Conference on VLSI Design*. 2009, p. 301–306.

[10] Bucci, G. and Luzzi, R. *Design of Testable Random Bit Generators.* International Association for Cryptologic Research, 2005, p. 147–156.

[11] *Federal Information Processing Standards Publication 140-2. Security Requirements for Cryptographic Modules.* National Institute of Standards and Technology, 2001. Available from WWW: <http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf>.

[12] Rukhin, A. et al. *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*. National Institute of Standards and Technology, 2010. Available from WWW: <http://csrc.nist.gov/groups/ST/toolkit/rng/documents/SP800-22rev1a.pdf>.

[13] Shannon, C. E. A Mathematical Theory of Communication. *The Bell System Technical Journal*. Vol. 27, 1948, p. 379–423, 623–656. Available from WWW: <http://cm.bell-labs.com/cm/ms/what/shannonday/shannon1948.pdf>.

[14] Croll, G. J. *BiEntropy – The Approximate Entropy of a Finite Binary String*. Cornell University Library. Available from WWW: <http://arxiv.org/ftp/arxiv/papers/1305/1305.0954.pdf>.

[15] Goka, T. An Operator on Binary Sequences. *SIAM Review.* Vol. 12, no. 2, 1970, p. 264–266. ISSN 0036-1445.