International Journal of Engineering & Technology, 7 (3.8) (2018) 69-73

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper



Design And Analysis of Analog TRNG Using Sample and Hold Circuit

Aksshaya B[1], G. Madhura L. V.[2], Nivethashri S[3], Vishnuvarthini T[4], Mohankumar N[*]

Student, Dept. Of Electronics And Communication Engineering., Amrita School Of Engineering, Coimbatore, India
**Corresponding Author E-Mail: N_Mohankumar@Cb.Amrita.Edu*

Abstract

Implementation of analog True random number generators is inevitable in almost all the security applications and encryption protocols nowadays. Although many digital True Random Number Generators are available, we proposed a method of random number generation using analog module of mixed signals. In actual fact generation of True Random Numbers is by utilizing the sample and hold circuit which is controlled by another random clock source, and a post processing circuit for generation of unpredictable binary sequence of numbers. The primary input source is an analog signal, essentially highly random noise from the external environment. The high unpredictability, less resource and simple circuit design are some highlights of the proposed work. Finally, the randomness is evaluated using NIST test suites and results are plotted and analyzed.

Keywords: *Hardware Security; NIST Tests; Random noise; Sample and hold; TRNG.*

1. Introduction

Information has value, especially in today's world. Keeping information protected is crucial. To avoid disclosing the information to unauthorized parties we go for cryptographic techniques. The basic function of cryptography is encryption, decryption of the message and cryptographic hashing. Of these cryptographic techniques, many cryptographic protocols require random number generators. And it is also important that these are strong random numbers that are unpredictable and non-reproducible. There are two basic random number generators [1]: True Random Number Generator (TRNG) and Pseudo Random Number Generators (PRNG). True random number generator, otherwise called hardware random number generator generates random numbers from physical sources, like the intensity of light in an image at a random pixel, or the speed of cooling fan in a processor, or random external noises of environment from a microphone. The generated random numbers have many applications such as simulation, decision making, numerical analysis, sampling etc. As we are concerned for the security of information and data transfer, we identify that in security aspect many cryptosystem algorithms are based on random number generators. The true random number generator we proposed in this paper is mainly for the purpose of encryption in cryptographic techniques. We also have pseudo random number generators which are computer generated that use mathematical algorithms. These algorithms use several variables to generate numbers making it difficult to predict the sequence of the numbers. Many algorithms were developed from different sources like visible spectrum, electronic components which use circuits like Phase Lock Loop (PLL), Discrete-time chaos method that limit A/D resolution to generate truly random sequence of numbers, strings of bits in which the next bits should be unpredictable. But writing an algorithm itself means that the numbers generated are following a sequence that will be repeated at some point of time. This explains, though PRNG is used for many applications it cannot be applied for high priority data encryption. The efficiency of random number generator has a significant influence on the security levels of the cryptographic systems. Only when the generated numbers are extremely and truly random, the information we are transmitting will be difficult to be decoded by unauthorized users. There are some statistical methods to test the randomness of the numbers like frequency test, chi-square test, collision test etc. NIST (National Institute of Standards and Technology) [1] will be one of the most used and significant method we use to check the randomness and it will be based on the range of some parameters P-value (it should be >0.01), threshold value (is 0.01 for 99% acceptance) etc. are calculated in the test.

2. Literature Survey

With communication and technology advancing in a fast pace and data transfer becoming an essential element, the need to secure vital information from attackers is inevitable. The concept of Random number generation **[1]** has been employed almost everywhere in cryptography and other areas where unpredictability is desired. The two principal methods to generate random numbers, pseudo random number generation whose generator is deterministic whereas hardware random number generation ensures complete unpredictability of the binary output sequence. TRNG through full custom chip design was carried out and tests using statistical test suites were performed.

The choice of chaotic systems **[2]** for TRNGs has been proposed, possible and the randomness performance of discrete time chaotic systems is studied. Implementation of 1D Discrete time chaotic maps in TRNG applications was evaluated and a measure of information called T-entropy was used to find the maximum amount of threshold and parameter controlling the chaos to determine hardware design parameter variation limits.

Copyright © 2018 Authors. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The advantage of TRNG over PRNG in high security applications is preferred as the physical source is naturally occurring entropy or any other phenomenon such as thermal or atmospheric noise. The selection of entropy source **[3]** is directly related to the amount of randomness. Digital circuits that exhibit metastable behavior was chosen and suitable entropy source was designed and tested for randomness. LUT based RNGs can be used to generate metastability rather than FPGA based ones as they can be easily reconstructed using simple algorithms.

One of the main applications of hardware random number generator is in the field of data encryption, to create random cryptographic keys. Rather than storing keys in the memory, using Physically Unclonable Function (for asymmetric cryptography) they are generated when needed. Ring oscillators is the source of entropy here making it advantageous to have TRNGs and PUFs in one single device **[4]**. The tests mentioned here should be implemented to test both functionalities at once as TRNGs and PUFs are on the same circuit. Our proposed technique provides asymmetrical key generation even more efficient and secured with comparatively high sampling rate and resolution.

FPGA optimized RNGs is a simple method that is more resource efficient, meeting the needs of the engineer. The implementation of modified LUT-SR generators is a midway between LUT-FIFO RNGs and LUT-OPT RNGs, providing high quality with a lesser resource utilization. The number of states determines the randomness of the generators. Also, LUT based Shift registers shows reduced delays when compared to the traditional ones **[5]**.

Technology has now developed to securely convert biometric data into cryptographic key for encryption and decryption purposes. As biometric data is unique, randomly fluctuating binary digits were used to generate sequences of random bits. **[6]** Neuro physiological brain response and galvanic skin response were the data sets for a simple data acquisition providing a sample of readings that were transformed into binary sequences and XOR-ed to increase the confidentiality. The randomness of these sequences was then verified using FIPS 140-2 suite recommended by the National Institute of Standards and Technology (NIST). The excellent results exhibited proved that there is a new technology to generate random bits in real time in a very simple way using biological systems. Features such as sampling rate, accuracy of measurement and variability of data for sampling to be methodically chosen for a biometric random number generator **[6]**. Other biometric candidates such as face, EEG, blood volume pulse could be considered that are obtained through relatively simple techniques and contain randomness in its internal structure.

The author has proposed a promising method of random number generation using fingerprints **[7]**, another possible approach using biometrics for its ease of use and low computational complexity.

A novel approach of iris data acquisition along with the chaotic function gave unique randomness and unpredictability leading to new kind of random number generator called Iris and Chaos-based random number generator (ICRNG) **[8]**, combining cryptography with biometrics. However the process of data collection from biometric templates is a challenging task because of the noise inherency and natural variability of the biometric credentials. Resistance to impersonation attack and irrevocability of the biometric templates can be overcome by our proposed method of generation.

Increasing the dependability of the true random number generators to guarantee the security and few challenges, also guidelines that can be applied to TRNGs have been proposed in **[9]**

Various statistical tests are developed to study and evaluate the quality of data for randomness **[10].** Real-time analysis of binary sequence produced by the generator and randomness testing based on NIST tests as an application in LabVIEW was proposed **[11].**

An extensive classification of hardware random number generators (linear and non-linear) on FPGA is presented and statistical comparison of the test results produced **[12]**, also the resistance to various forms of cryptanalysis like correlation attacks and algebraic ones are improved. A three-tier security scheme by Embedding unique signatures with minimalistic hardware or area overhead using a PRNG is proposed by Manoj Reddy et.al. [**15**].

3. Proposed Technique

A technique is intended to generate binary true random numbers. The generalized block diagram of a True Random Number generation circuit includes a source, which outputs a random signal and a post processing circuit.

The general block diagram for generation of TRNG sequence is as shown in figure 1.



Fig. 1: General Block Diagram of TRNG circuit.

***A*:** Input Block:

The Excellency of the generated True Random Number sequence lies in the degree of randomness of the sequence. A higher degree of randomness is essential. The randomness in the sequence depends on the randomness in the signal generated by the source. Henceforth, the choice of source is important and essential requirement. A random signal is essentially a noise signal. The proposed technique uses the noise from the external environment. This noise signal is fed into the audio input pin of the desktop as the input noise source. The circuit uses SIMULINK to acquire this noise in real time.

The output from this signal is highly random with wide and random values. Thus the real time audio input signal is successfully acquired and can be displayed using the time scope block in Simulink as shown in figure 2.

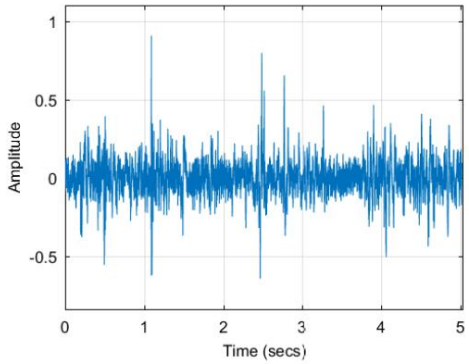


Fig. 2: Acquired noise signal plotted using the time scope.

B: Post Processing Block:

After the acquisition of the input signal, the noise signal is processed to generate the required TRNG sequence. The circuit used a sample and hold circuit in the post processing block.

As implied by its name, sample and hold circuit is used to sample the given input signal for a small interval of time and hold this sampled valued until it is sampled again. The simple circuit for sample and hold circuit is as shown in figure 3.

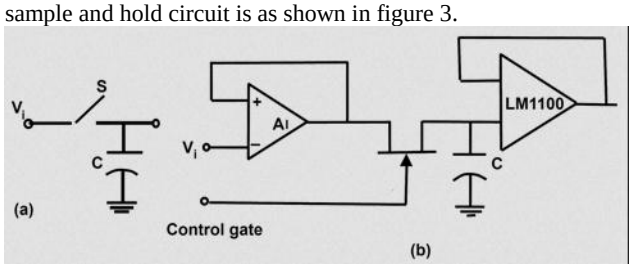


Fig. 3: Block Diagram of Sample and Hold circuit.

The input signal is an analog signal. The control input is a logical signal or a clock pulse. In the above circuit it is applied to the gate input of the MOSFET device .The signal decides when the input

has to be sampled or be held. When the control input is high, the MOSFET turns on and the capacitor charges. Thus input analog signal is sampled. When the control is low, the input is held since the capacitor has no path for discharge.

The proposed circuit uses the sample and hold circuit .To increase the randomness of the output sequence the control input signal is given a random binary signal. Thus, the input is sampled and held in a random manner. For this, a random number generator block is used. The output of this block is converted to a binary signal by modulo-2 division. Hence, the output is either zero (low) or one (High). This can be visualized by displaying using the time scope block as shown in figure 4.

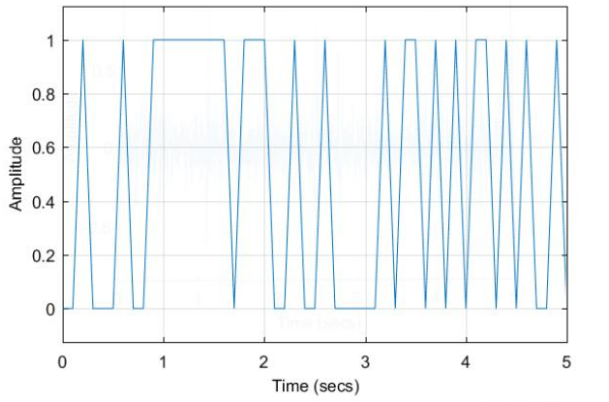


Fig. 4: Control Input signal plotted using time scope

Accordingly, the audio input is sampled or held. The output of the sample and hold circuit is as shown in figure 5.
For different audio inputs, different random number can be generated by changing the mean, variance and the seed value of the random number generator block.
The output of the sample and hold circuit is a digital signal whose amplitude is finite but may be an integer or a decimal value as shown in figure 5.

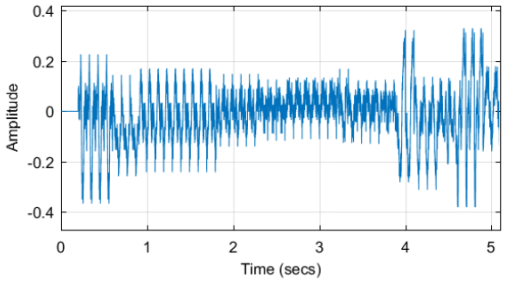


Fig. 5: Output waveform of sample and hold circuit using Time Scope.

In order to generate a binary sequence, the output of the sample and hold circuit is converted to 32-bit integer. The resulting output is then converted to a binary value by modulo-2 division on each output value. The output can be displayed using the time scope block as shown in figure 6.

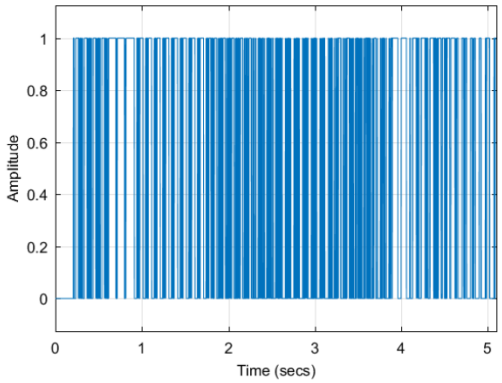


Fig. 6: Generated Output TRNG sequence plotted using Time scope

The circuit is designed using SIMULINK as shown in figure 7.

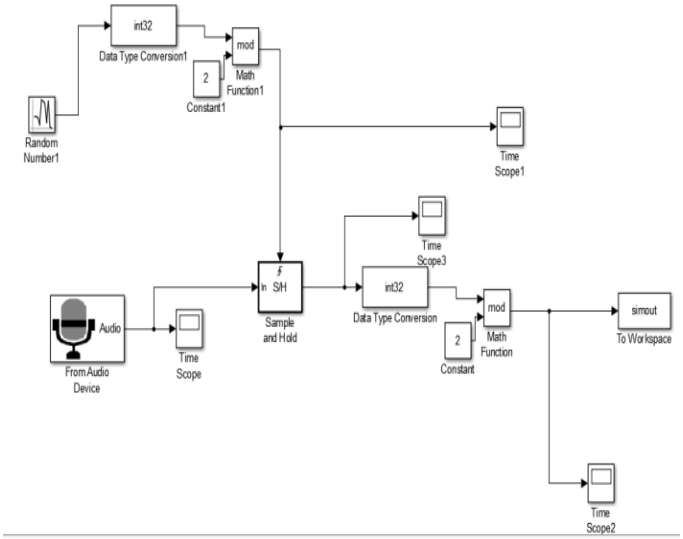


Fig . 7: Circuit Proposed using SIMULINK.

C: Block Diagram:

The Block diagram of the circuit is as shown in figure 8.

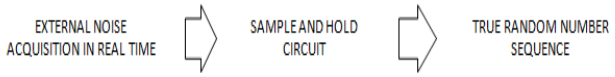


Fig. 8: Block Diagram of the proposed circuit.

4. Result and Analysis

Ten sequences were generated using 10 different noise sources and 10 different values of mean, variance and seed values. The out sequence is a 32 bit integer and is 52224 long sequence.

The primary input is the real time noise in the external environment. Different environmental noise like sound of a fan, music etc. were acquired using Simulink as inputs and are fed into the sample and hold circuit.

The random number generator block at the control input has three input parameters namely mean, variance and seed. The seed value is an integer value and for every unique seed, value generates the same sequence. Thus with the same mean, variance and seed value the clock can be regenerated. Thus the output clock generated is a binary pseudo random number sequence.
The input parameters given to the random number generator block is as shown in table 1.

Table 1: Input given to the random number generator block

Sequence Number	Mean	Variance	Seed
1	0	1	1
2	0	1	2
3	0	1	5
4	0	1	10
5	3	5	1
6	3	5	5
7	3	5	7
8	6	7	1
9	0	7	1
10	0	5	1

The input noise is sampled using the clock generated and the output true random sequence is thus generated. A true random number sequence ideally is highly random in nature with infinite period. The generated True random number sequence is tested for its randomness.

The National Institute of Standards and Technology (NIST) developed 15 statistical test suits which exhaustively tests the different aspects of randomness in the sequences. 11 test cases as developed by the NIST statistical test was written in MATLAB environment to test the randomness in the sequence generated using the proposed circuit. The results obtained for 8 different randomness tests for the sequence generated is as shown in figure 9.

Table 2: Results obtained for the randomness tests.

Test Performed	Freq. Monobit Test	Freq. Test Within the Block	Run Test	Longest Run of Ones in a Block Test	Overlapping Template Matching Test	Non Overlapping Template Matching Test	Cumulative Sums (Cusums) Test	Spectral Test
Sequence 1	0.0000	1.0000	Need Not be Performed	0.0000	NaN	1.0000	1.0000	0.0556
Sequence 2	0.4676	1.0000	0.0000	0.0000	NaN	1.0000	1.0997	0.4038
Sequence 3	0.0063	1.0000	0.0000	0.0000	NaN	1.0000	1.0000	0.1974
Sequence 4	0.0000	1.0000	Need Not be Performed	0.0000	NaN	1.0000	1.9933	0.0718
Sequence 5	0.0004	1.0000	0.0000	0.0000	NaN	1.0000	1.0000	0.7419
Sequence 6	0.0000	1.0000	Need Not be Performed	0.0000	NaN	1.0000	1.0000	0.0000
Sequence 7	0.0000	1.0000	Need Not be Performed	0.0000	NaN	1.0000	1.0000	0.0000
Sequence 8	0.0929	1.0000	0.0000	0.0000	NaN	1.0000	1.0000	0.0000
Sequence 9	0.0000	1.0000	Need Not be Performed	0.0000	NaN	1.0000	1.0000	0.2157
Sequence 10	0.8611	1.0000	0.0000	0.0000	NaN	1.0000	1.0000	0.2608

Frequency Mono bit Test is performed to identify the proportion of ones and zeroes in the entire sequence. A highly random sequence should cross a threshold of 0.01 to be concluded as a random sequence.
If a sequence does not cross the threshold it can be concluded that, the sequence has frequent occurrence of ones when the result is a large positive and large negative result indicates the frequent occurrence of 0.

It can be observed that, in the sequence generated, the results for sequence 2, 8 and 10 have crossed the threshold value.
All other sequences generated have greater number of ones compared to zeroes.

Frequency Test within the Block is same as the above described except that the test is performed by splitting the sequence into blocks of desired size. According to this test, sequence having values greater than 0.01 can be measured to be random.
It is observed that all the sequences have crossed the threshold value and according to this test all the sequences are considered random.

Runs Test helps in identifying the number of blocks of uninterrupted sequence of a single bit.
If a sequence passes the frequency test, the run test value shall not be performed. Since all the sequences have passed the frequency test 2, the run tests need not be performed. Also, if the run test is not applicable the value is set to 0. Hence it is observed that the values for sequence 2, 8, and 10 is set to 0.

Overlapping and Non-Overlapping Template Test evaluates the number of occurrences of pre-determined bit strings so that it is possible to detect if any specific aperiodic sequence often occurs. Non overlapping template identifies the required sequence to cross a threshold of 0.01 that are considered to be random. It is seen that all the sequences have crossed the threshold.

Cumulative Sums Test is to verify if the sum of partial sequence is too large or small. The results of this test presume the given sequence to cross a threshold of 0.01to be considered random. It is inferred that all the sequences are random.

Spectral test is used to identify the periodic patterns that appear close to each other. It can be concluded that all the sequences are random as the occurrence of periodic patterns are not considerably close.

Thus, it is seen that all the generated sequence have passed 5 out of 8 tests. Sequences 2, 8 and 10 have passed all the tests.

5. Conclusion

In this paper, a simplified method of generating true random numbers is implemented. The circuit is able to generate a truly random sequence, passing almost all the tests. A highly noisy environment results in a highly irregular and a truly random sequence. The circuit is quite simple with minimum components. When the generated sequences are tested and the results are analyzed, it is seen that the sequence has passed almost all the tests that checks for non-periodicity. It can be concluded that this simple circuit can generate a highly non-periodic sequence and hence be used in applications were non-periodicity is needed, particularly in the field of data security and cryptography.

References

[1] Avantika Yadav, Design and Analysis of Digital True Random Number Generator, Virginia Commonwealth University Richmond, Virginia (2013).
[2] IhsanCicek, Ali EmrePusane, GunhanDundar, A novel design method for discrete time chaos based true random number generators, Elsevier Ltd. INTEGRATION ,the VLSI journal 47 (2014) 38-47.
[3] Lakshmi Sreekumar, Dr. Ramesh P, Selection of an Optimum Entropy Source Design for a True Random Number Generator, Elsevier Ltd. Procedia Technology 25 (2016) 598 – 605.
[4] Simona Buchovecka, RobertLórencz, Filip Kodýtek, JiríBucek, True random number generator based on ring oscillator PUF circuit, Elsevier Ltd. Microprocessors and Microsystems 53 (2017) 33-41.

[5] Remya Justin, Binu K Mathew, Susan Abe, FPGA Implementation of High Quality Random Number Generator using LUT based Shift Registers, Elsevier Ltd. Procedia Technology 24 (2016) 1155 – 1162.
[6] J. Szczepanski, E. Wajnryb, J.M. Amigo, Maria V. Sanchez-Vives, M. Slater, Biometric random number generators, Elsevier-Computers and Security (2004) 23, 77-84.
[7] ShkodranGerguri, Vaclav Matyas, ZdenekRiha, LudekSmolik, Random Number Generation Based on Fingerprints, P. Samarati et al. (Eds.): WISTP 2010, LNCS 6033, pp. 170–182, 2010.
[8] Hegui Zhu, Cheng Zhao, Xiangde Zhang, Lianping Yang, A novel iris and chaos-based random number generator, Computers and Security 36 (2013) 40-48.
[9] Honorio Martin, Giorgio Di Natale, Luis Entrena, Towards a Dependable True Random Number Generator with Self-Repair Capabilities, IEEE Transactions on Circuits and Systems–I:, VOL. 65, NO. 1, January 2018.
[10] Filip Veljkovic, Vladimir Rozic, Ingrid Verbauwhede, Low-Cost Implementations of On-the-Fly Tests for Random Number Generators, 978-3-9810801-8-6/DATE12/ conference@2012 EDAA.
[11] JoãoDionísio, Tiago Mota, Iola Pinto, Manfred Niehus, Real Time Random Number Generator Testing, Elsevier Ltd. Procedia Technology 17 (2014) 534 – 541.
[12] Mohammed Bakiri, Christophe Guyeux, Jean-François Couchot, AbdelkrimKamelOudjida, Survey on hardware implementation of random number generators on FPGA: Theory and experimental analyses, Elsevier Ltd. Computer Science Review 27 (2018) 135-153.
[13] Shiva Prasad R., AnirudhSiripagada, SantthoshSelvaraj, Mohankumar N. "Random Seeding LFSR based TRNG for Hardware Security Applications", Proc. of 2nd Intl. Conf. on Integrated Intelligent Computing, Communication & Security (ICIIC-2018), 2018.
[14] Ashok Kumar Mohan; Nirmala Devi M.; M. Sethumadhavan; Santhya R.,"A Selective Generation of Hybrid Random Numbers via Android Smart Phones
[15] Manoj Reddy, Akshay K P, Giridhar R, Kharan SD, Mohankumar N, "BHARKS: Built-in hardware authentication using random key sequence, Proc of 4th IEEE Conference on Signal Processing Computing and Control (ISPCC), pp 200-204, Salon, 2017. DOI: 10.1109/ISPCC.2017.8269675