Hardware Random Number Generator Using FPGA

D. Indhumathi Devi*, S. Chithra and M. Sethumadhavan

TIFAC-CORE in Cyber Security, Amrita School of Engineering,
Coimbatore, Amrita Vishwa Vidyapeetham, India
E-mail: indhudevaraj@zoho.com; chithrasnarayan@gmail.com;
m_sethu@cb.amrita.edu
**Corresponding Author*

Received 04 January 2018; Accepted 04 June 2018;
Publication 20 June 2019

Abstract

Random numbers are employed in wide range of cryptographic applications. Output of an asynchronous sampling of ring oscillators can be used as the source of randomness and Linear Hybrid Cellular Automata is used to improve the quality of random data. FPGA is an ideal platform for the implementation of random number generator for cryptographic applications. The circuit described in this paper has been implemented on a highly efficient FPGA board which generated a 32-bit random number at a frequency of 125 MHz. The generated sequence of random numbers were subjected to Diehard test and NIST test for testing randomness and found to pass these tests. These tests are a battery of statistical tests for measuring the quality of a random number generator.

Keywords: Ring oscillator, Linear Hybrid Cellular Automata, Field Programmable Gate Array, Diehard Test.

1 Introduction

Random number generator (RNG) has become a most important component in many cryptographic applications such as PIN/password generation,

Journal of Cyber Security and Mobility, Vol. 8_4, 409–418.
doi: 10.13052/jcsm2245-1439.841

authentication protocol, key generation, random padding and nonce generation. Hardware random number generator (HRNG) is also known as True random number generator (TRNG) which is used for many cryptographic applications which must meet stringent specification since all security protocol demands on unpredictable keys or initialization vector used. Thus an attacker having entire knowledge about the design of HRNG will not be able to predict the future bits. A hardware random number generator or true random number generator is one in which the probability of bits generated is statistically independent and unbiased.

Random number generators are classified into Pseudo random number generator (PRNG) and True random number generator. In case of PRNG the generated random numbers are based on deterministic algorithm. In PRNG the initial seed value will be given to the random number generator and based on the seed value the other sequence of random numbers are generated. The security of the PRNG depends on the initial value of the seed. If the initial seed value is known to the adversary, the entire random sequence that are generated by PRNG can be predicted and then sequence of key stream generated using the particular seed value becomes unsecured. In this case in order to increase the security of the key generated the initial key size should be increased to reduce the predictability.

Hardware random number generator or TRNG accomplishes randomness based on physical phenomenon where the source of randomness to produce a random number becomes faster, higher in quality. TRNG output is entirely based on random physical process. Unlike PRNG there is no internal state kept in the generator since the outputs are based on physical phenomenon and not on any previously produced bits. Thus the output sequence of hardware random number generator is assured to have good statistical property which can be verified using NIST and Diehard test suite. Hardware random number generators produce randomness based on non-deterministic phenomenon.

True random number generator uses physical phenomenon such as noise produced in the electronic device which generate bits at very low bit rate. True random number generators are purely based on digital constructions, simply integrated in a single chip. There are techniques to find the digital circuit's behavior that will give the possibility to generate random bit sequence on demand with higher bit rate, without any possibility to have access to element of the sequence. Hardware random number generator in the proposed method uses time delay and jitters to generate random bits which are constructed by using re-programmable digital circuits.

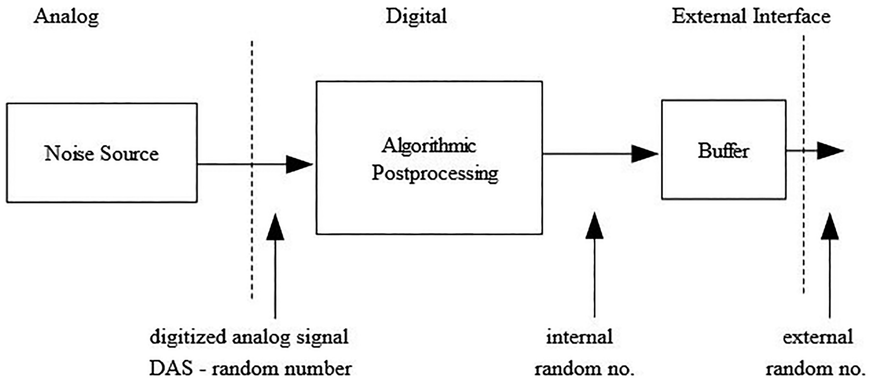

Figure 1 Generic Design of TRNG.

The Ring oscillators which use jitters as the source of signal generation are sampled at low frequency to obtain the random bits. In the proposed method Ring oscillators (RO) are used as the basic unit of HRNG. The TRNG is implemented on Xilinx FPGA board and the generated random numbers were subjected to Diehard and NIST statistical test. The generic design of TRNG [6] is shown in Figure 1.

2 Ring Oscillator

A ring oscillator is a device composed of odd number of NOT gate in a closed loop whose output oscillates between two voltage level, representing TRUE or FALSE [3]. Jitter in oscillators are used as the source of randomness [5]. Due to ease and simplicity of design ring oscillator based HRNG has been implemented. In case of RO more than one logic gates are connected in a loop to oscillate if total numbers of logic inversions are odd. The inverter output is connected back to its input using a feedback which helps the ring oscillator to generate its output at very high frequency. The Figure 2 shows the simple ring oscillator built using XOR gate. The gate propagation delay and routing delay determines the output of the ring oscillator. The frequency of the ring oscillator is not very stable because it is affected by temperature and voltage variation and noise in the system. Asynchronous sampling of ring oscillator will produce truly random binary bits. The Figure 3 shows the sampled construction of ring oscillator. There are few drawbacks in this method. Sampling the flip flop for the first time meta-stability condition is obtained so its output is not reliable in more than one place. The output of ring oscillator should be double

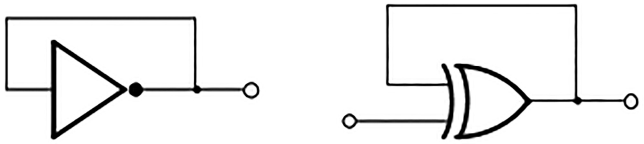

Figure 2 Simple Ring Oscillator Using XOR and NOT Gate.

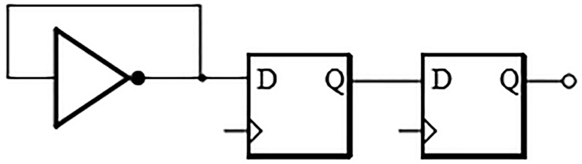

Figure 3 Sampled Ring Oscillator.

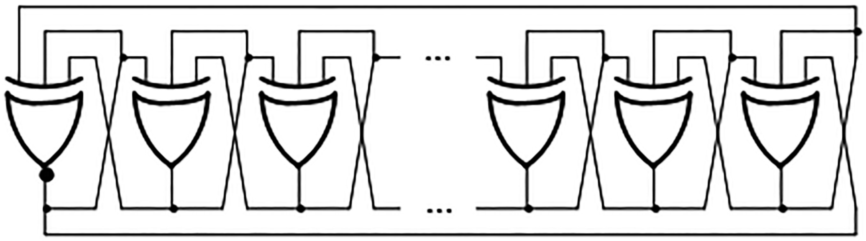

Figure 4 Generalized n-bit Ring Oscillator.

sampled in order to avoid meta-stability condition. The double sampled ring oscillator [9, 10] will produce a stable logic output. The second drawback is the correlation between consecutive pixels. The output of ring oscillator should be a perfect symmetry in order to obtain the equal probability of random bits 0 and 1. In this paper the experiment was performed to produce a 32-bit random sequence per clock. The above two issues can be solved by using generalized ring oscillator and Linear hybrid cellular automata. The Figure 4 is the generalized n-bit ring oscillator.

The ring oscillators [4] designed with cascaded chain of delay is used in numerous stages. The generalized ring oscillator [2] is nothing but a interconnection of multiple XOR gate which is used to generate n-bit output. In the proposed method interconnection of gates is in such a way that each XOR gate connected to its two neighborhoods XOR gates. This interconnection makes the circuit scalable by achieving highest oscillating frequency. The nth XOR gate is inverted in order to avoid the circuit stability condition. The ring oscillator will provide a chaotic behavior due to the logical delay and noise present in the circuit.

3 Linear Hybrid Cellular Automata

Linear Hybrid cellular automata (LHCA) is also known Linear finite state machine (LHSM) in which each bit comprised of a one-dimensional array of cells. Each cell is known as buffer, these cells are allowed to communicate only with their neighbors. LHCA produces a pseudo random bit every clock instead of just one bit per clock; the complexity of LHCA is similar to that of LFSR. The parallel implementation of LHCA utilizes only smaller circuit design and faster in speed. Linear hybrid cellular automata and linear feedback shift register are finite state machines. The future state of each bit depends on itself and the two neighbors. The Figure 5 shows the generalized architecture of LHCA.

The properties of both LHCA and LFSR are same. There is a one to one relation between maximum length LFSR and LHCA and there exist a certain set of coefficients for each primitive characteristic polynomial.

4 Hardware Random Number Generation Using Generalized RO and LHCA

The combination of generalized Ring oscillators and LHCA [7, 8] are used to achieve high quality 32-bit random number. The generalized RO produces

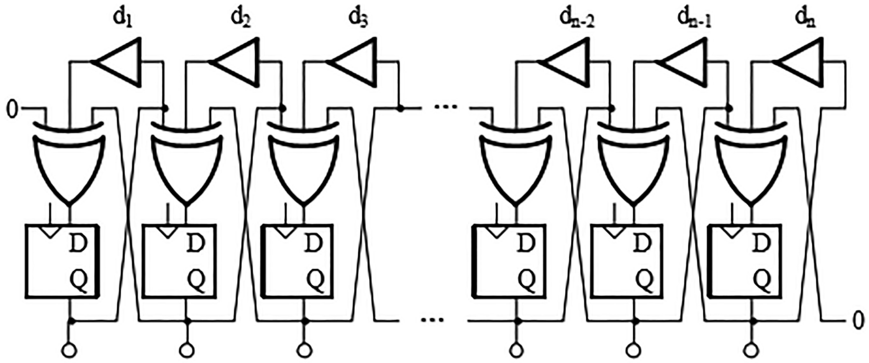

Figure 5 Generalized Linear Hybrid Cellular Automata.

the random number with uniform probability distribution but the sampling rate is too high which increases the correlation between the random numbers generated. Thus the output of generalized RO is given to generalize LHCA to scramble the output of generalized RO which helps in producing the number at a very high quality. In the proposed system random number generator uses only two flip-flops and two 4 input look-up table for every random sequence that is generated. This design is implemented on a Xilinx FPGA [1, 3] board so this system is assured to be a more secure because there is no need of external components. In this paper the 32-bit true random number is generated at a frequency of 125 MHz with a bit rate of 4 Gbps. The circuit design of hardware random number generator using generalized RO and LHCA is shown in Figure 6.

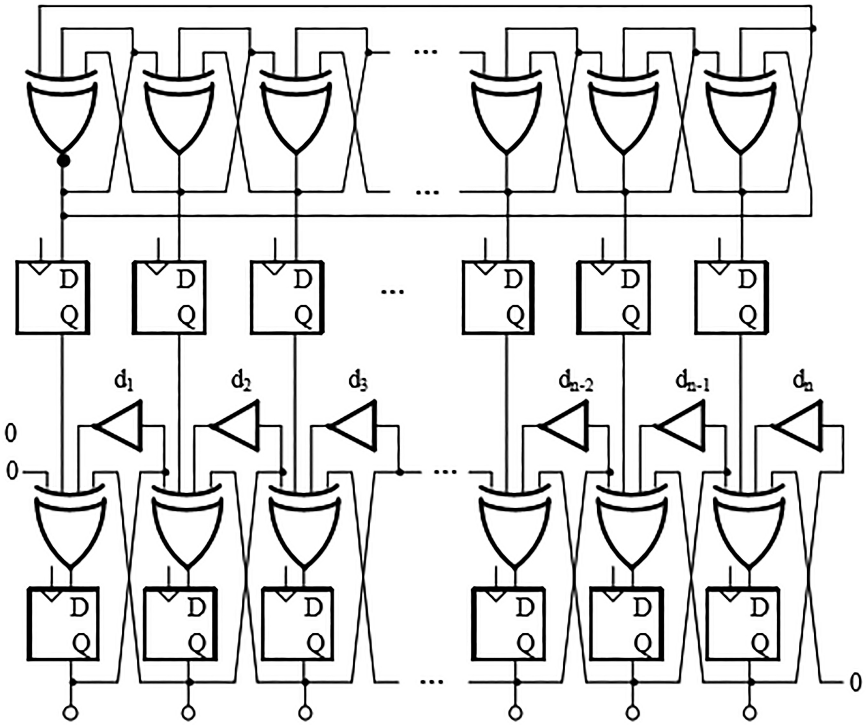

Figure 6 Hardware random number generator using generalized RO and LHCA.

5 Statistical Test

The Diehard tests are a battery of tests for measuring the quality of a random number generator [11]. Diehard was built to test large sets of random positive integers, read in from binary files.

In our experiment, the implementation of random number generator was done on the Artix-7 FPGA board and the UART is used to communicate with the PC for storing the generated random numbers. The Diehard test is applied for an 11 MB of minimum data from collected data set of random sequence in order to test its statistical property.

The Diehard Test consists of 15 types of randomness tests. Result of Diehard Test is interpreted based on a p value which should be uniform on (0, 1) then only we can say the generated data set is truly random. The p-value should be in the range of 0.025–0.975 to pass the randomness test.

The proposed hardware random number generator when subjected to Diehard test resulted with the values between 0.0052–0.925 for the entire 15 test. The output of Diehard test implies that the hardware random number generator designed was proved to be a secured random number generator.

6 Conclusion

True random numbers of each 32-bit at 125 MHz clock frequency are generated using a Artix-7 FPGA board and Vivado Design Suite HLx. The generated random numbers had been proved to be a high quality random number since it passed the entire 15 Diehard test. The future work is to test the randomness of the HRNG at different frequencies in order to obtain the different set of random data. The generated random data set can be used as keys for many cryptographic applications to ensure higher security for data communication.

References

[1] Saichand, V., Arumugam, S. and Mohankumar, N., 2008, November. FPGA realization of activation function for artificial neural networks. In Intelligent Systems Design and Applications, 2008. ISDA'08. Eighth International Conference on (Vol. 3, pp. 159–164). IEEE.

[2] Baetoniu, C., High speed true random number generators in xilinx fpgas. Online]. Dosegljivo:http://forums.xilinx.com/xlnx/attachments/xlnx/EDK/27322/1/HighSpeedTrueRandomNumberGenerators inXilinxFPGAs.pdf. [Dostopano 12. 8. 2016].

[3] Johnson, A. P., Chakraborty, R. S. and Mukhopadyay, D., 2017. An Improved DCM- Based Tunable True Random Number Generator for Xilinx FPGA. IEEE Transactions on Circuits and Systems II: Express Briefs, 64(4), pp. 452–456.

[4] Mandal, M. K. and Sarkar, B. C., 2010. Ring oscillators: Characteristics and applications. Vancouver.

[5] Hajimiri, A., Limotyrakis, S. and Lee, T. H., 1999. Jitter and phase noise in ring oscillators. IEEE Journal of Solid-state circuits, 34(6), pp. 790–804.

[6] Shanmuga Sundaram, P., 2010. Development of a FPGA-based True Random Number Generator for Space Applications.

[7] Zhang, S., Byrne, R., Muzio, J. C. and Miller, D. M., 1995. Quantitative analysis for linear hybrid cellular automata and LFSR as built-in self-test generators for sequential faults. Journal of Electronic Testing, 7(3), pp. 209–221.

[8] Stipevi, M. and Ko, K., 2014. True random number generators. In Open Problems in Mathematics and Computational Science (pp. 275–315). Springer International Publishing.

[9] Toza, S. and Matuszewski, 2014, September. A true random number generator using ring oscillators and SHA-256 as post-processing. In Signals and Electronic Systems (ICSES), 2014 International Conference on (pp. 1–4). IEEE.

[10] Schellekens, D., Preneel, B. and Verbauwhede, I., 2006, August. FPGA vendor agnostic true random number generator. In Field Programmable Logic and Applications, 2006. FPL'06. International Conference on (pp. 1–6). IEEE.

[11] Brown, Robert G., Dirk Eddelbuettel and David Bauer. "dieharder: A Random Number Test Suite, 2007." URL http://www. phy.duke.edu/rgb/General/dieharder.php. C program archive dieharder, version 2.3.

Biographies

D. Indhumathi Devi received M.Tech. (Cyber Security) from Amrita Vishwa Vidyapeetham, Coimbatore currently working as a Junior Research Fellow at Amrita Vishwa Vidyapeetham. Her areas of interest are Cryptography and Cyber Forensics.

S. Chithra received M.Tech. in Cyber Security from Amrita Vishwa Vidyapeetham, Coimbatore and currently working as Information security Analyst at Paladion Networks Pvt Ltd. Her areas of interests include information security, cryptography and Endpoint security.

M. Sethumadhavan received his PhD (Number Theory) from Calicut Regional Engineering College. Currently, he is working as a Professor in the Centre for Cyber Security, Amrita Vishwa Vidyapeetham, Coimbatore. His current research interests include: Cryptography and Boolean functions.