ISSN: 2454-9940

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

E-Mail :
editor.ijasem@gmail.com
editor@ijasem.org

www.ijasem.org

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN2454-9940

www.ijasem.org

Vol 18, Issue.2 April 2024

A LIGHTWEIGHT TRUE RANDOM NUMBER GENERATION FOR ROOT OF TRUST APPLICATION

M. ANJANEYULU *, DR. E KRISHNAHARI**

PG SCHOLAR*, ASSOCIATE PROFESSOR**

DEPARTMENT OF ECE, HOLYMARY INSTITUTE OF TECHNOLOGY AND SCIENCE (APPROVED BY AICTE NEW DELHI, AFFILIATED TO JNTUHYDERABAD) BOGARAM (V), KEESARA (M), MEDCHAL DISTRICT -501 301

ABSTRACT: With the rapid development of communication technology and the popularization of network, information security has been highly valued by all walks of life. Random numbers are used in many cryptographic protocols, key management, identity authentication, image encryption, and so on. True random numbers (TRNs) have better randomness and unpredictability in encryption and key than pseudorandom numbers (PRNs). Chaos has good features of sensitive dependence on initial conditions, randomness, periodicity, and reproduction. These demands coincide with the rise of TRNs generating approaches in chaos field. This survey paper intends to provide a systematic review of true random number generators (TRNGs) based on chaos. Firstly, the two kinds of popular chaotic systems for generating TRNs based on chaos, including continuous time chaotic system and discrete time chaotic system are introduced. The main approaches and challenges are exposed to help researchers decide which are the ones that best suit their needs and goals. Then, existing methods are reviewed, highlighting their contributions and their significance in the field. We also devote a part of the paper to review TRNGs based on current-mode chaos for this problem. Finally, quantitative results are given for the described methods in which they were evaluated, following up with a discussion of the results. At last, we point out a set of promising future works and draw our own conclusions about the state of the art of TRNGs based on chaos.

INTRODUCTION: In recent years, with the rapid development of the Internet, the requirements for information security in various fields are getting higher and higher, and the security issues are getting more and more attention [1–5]. In the field of information security, encryption algorithm, and key generation are important factors of encryption system; they must be unpredictable [6–9]. In most cryptographic algorithms, random number is an

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT
ISSN2454-9940
www.ijasem.org
Vol 18, Issue.2 April 2024

indispensable element, and random number generator (RNG) has important applications in the field of information security, such as generating parameters of public key cryptosystems (such as ECC, RSA) or image encryption [10–12]. According to the different random sequence generated, random numbers can be divided into two categories, namely pseudo-random numbers (PRNs) and true random numbers (TRNs), as shown in Figure 1. PRNs [13, 14] refer to the extension of one seed into another long output sequence by a determined algorithm, which are generally repeatable, so they are widely used in the field of simulation and testing. Unlike PRNs, TRNs [15, 16] cannot be generated by pure mathematical random algorithms, but only by random physical processes. Compared with PRNs, TRNs not only have good statistical characteristics but also have good unpredictability. They could be used in systems with high security requirements.

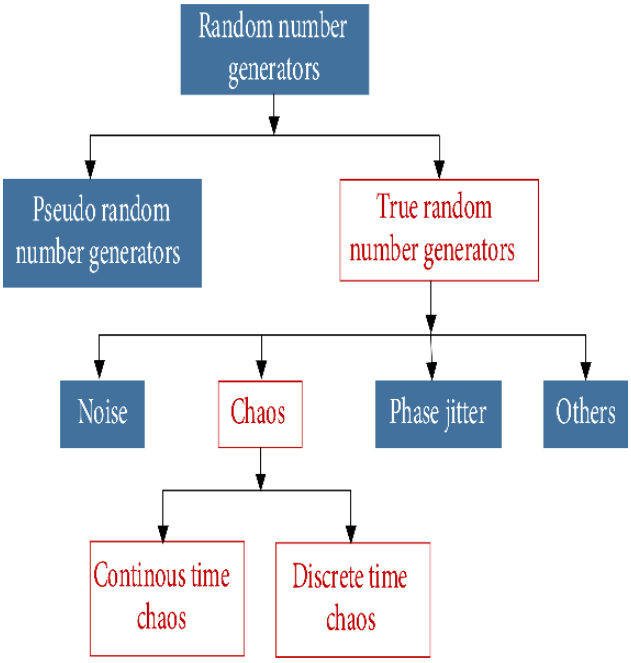

Fig 1: The architecture of random numbers generator.

The typical TRNG structure can be divided into five modules: (1) analog random signal is obtained from the entropy source; (2) sampling and quantifying the random signal; (3) analog-to-digital conversion of the analog signal to output the random number sequence; (4) the sequence obtained at this time does not necessarily satisfy the uniform distribution, and it needs to be processed; and (5) through random number test suite, as shown in Figure 2.

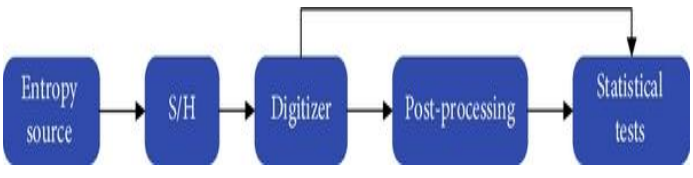

Fig 2: The typical TRNG structure

In true random number generators (TRNGs), there are three main types of entropy sources: thermal noise on resistors and capacitors [17, 18], phase jitter of oscillating signals [19–21], chaos [22–24] and others, as shown in Figure 1. For the TRNGs based on thermal noise, the resistance noise is amplified to a suitable range by an ideal amplifier, and then processed by a comparator to compare the amplified noise voltage with the reference level to obtain a digital random signal [17]. In practice, due to the influence of some nonideal factors, such as the limited bandwidth of the amplifier, misalignment, periodic noise of the power supply coupled to the system, the randomness of the random number sequence generated by the system will be affected [18]. For the oscillator-based TRNGs, the random source is the phase jitter noise in the ring oscillator in Complementary Metal Oxide Semiconductor (CMOS) circuit [20]. The quality of the random sequence generated by the true random number generator is largely determined by the root mean square (RMS) value of the phase jitter of the low frequency oscillator [21]. But the disadvantage is that it is not suitable for full custom integrated circuit (IC), and the randomness of circuit implementation is low. Compared with the former two methods, the characteristics of chaos, such as nonperiodicity, wide spectrum, unpredictability, and sensitivity to initial conditions [25–28], are in good agreement with the properties of random numbers. Therefore, chaotic theory opens up broad prospects for the design and implementation of TRNGs. The main contributions of our work are as follows: (1) we provide a broad survey of generating methods that might be useful for TRNGs with chaos; (2) an in-depth and organized review of the most significant methods that use chaos for TRNGs, their origins, and their contributions; (3) we have conducted a comprehensive performance evaluation, which collects quantitative indicators. For example, power, output bit rate, energy and technology, etc.; and (4) a discussion about the above results, and a list of possible future works that may determine the course of upcoming advances, as well as a conclusion summarizing the state of the art of the field.

LITERATURE REVIEW

Over the past decade field programmable gate arrays (FPGAs) have become invaluable components in many facets of digital design. As a result of increased integration, FPGA devices are now used across a wide assortment of fault tolerant

and mission critical digital platforms. This application-level diversity has necessitated increased interest in FPGA test so that faulty components can be quickly identified and recovered. Given the large range of applications and programmable configurations each FPGA device may support, FPGA test can be substantially more complex than ASIC test, providing motivation for new, efficient testing techniques. Information regarding defect location is particularly important in today's test environment since new techniques [1] have been developed that can reconfigure FPGAs to avoid faults. To operate effectively, these approaches require that the specific location of the fault be clearly identified. The reconfigurability of FPGAs plays an important role in reducing on-chip testing hardware relative to ASICs. While ASIC DFT approaches require the modification of circuit functionality to perform test, FPGA test hardware can be swapped out of the device once verification is complete. Reconfigurability does incur other test costs, including increased test generation complexity and increased test application time. Unlike ASICs, which require a single configuration for fault detection, FPGAs require multiple configurations to test an assortment of switch settings. In general, fault coverage is directly related to the number and scope of test configurations that are created. A trend of multiprocessor system-on-chip (MPSoC) design being interconnected with onchip networks is currently emerging for applications of parallel processing, scientific computing, and so on. Permutation traffic, a traffic pattern in which each input sends traffic to exactly one output and each output receives traffic from exactly one input, is one of the important traffic classes exhibited from on-chip multiprocessing applications. Many of the MPSoC applications compute in real-time, therefore, guaranteeing throughput is critical for such permutation traffics. Most on-chip networks in practice are general-purpose and use can be implemented as source routing or distributed routing. However, such application aware routings cannot efficiently handle the dynamic changes of a permutation pattern, which is exhibited in many of the application phases. The difficulty lies in the design effort to compute the routing to support the permutation changes in runtime, as well as to guarantee the permutated traffics. This becomes a great challenge when these permutation networks need to be implemented under very limited on-chip power and area overhead. Reviewing on-chip permutation networks (supporting

either full or partial permutation) with regard to their implementation shows that most the networks employ a packet-switching mechanism to deal with the conflict of permuted data. Their implementations either use first-input first-output (FIFO) queues for the conflicting data or time-slot allocation in the overall system with the cost of more routing stages, or a complex routing with a deflection technique that avoids buffering of the conflicting data. The choices of network design factors, i.e., topology, switching technique and the routing algorithm, have different impacts on the on-chip implementation. System on- Chip (SoC), composed of heterogeneous cores on a single chip, has entered billion-transistor era. As the microprocessor industry is moving from single-core to multicore and eventually to many-core architectures, containing tens to hundreds of identical cores arranged as chip multiprocessors, which also require efficient communications among processors. Both SoC and microprocessor call for a high-performance, flexible, scalable, and design-friendly interconnection. How to provide efficient communication poses a challenge to researchers. Before the advent of network-onchip, interconnection architectures are usually based on dedicated wires or shared buses. Dedicated wires provide point-to-point connection between every pair of nodes, effective for small systems of a few cores. But as the number of cores increases, the number of wires in the point-to-point architecture grows quadratically, making it unable to scale. Compared to dedicated wires, a shared bus which is a set of wires shared by multiple cores is more scalable and reusable. However, due to the inherent disadvantage of buses, only one communication transaction is allowed at a time, blocking communication for all other cores. The disadvantages of shared bus architectures include long data delay, high energy consumption, increasing complexity in decoding arbitration, low bandwidth. It would be daunting inefficient if hundreds of nodes are connected by shared buses. Thus, the usage of shared buses is limited to a few dozens of IP cores. To deal with the problems in shared buses, a hierarchical architecture, which segments bus into shorter ones, is introduced. Hierarchical bus architectures may relax some of constraints faced by dedicated wires and shared buses, since different buses may account for different bandwidth needs, protocols and also increase communication parallelism. Nonetheless, scalability remains a problem for hierarchical bus architectures.

OVERVIEW OF THE PROPOSAL

On-chip network design is used a pipelined circuit-switching approach with a dynamic path-setup scheme supporting runtime path arrangement A dynamic path-setup scheme is the key point of the proposed design to support a runtime path arrangement when the permutation is changed. Each path setup, which starts from an input to find a path leading to its corresponding output, is based on a dynamic probing mechanism. The concept of probing is introduced in works in which a probe (or setup flit) is dynamically sent under a routing algorithm in order to establish a path towards the destination. Exhausted profitable backtracking (EPB) is proposed to use to route the probe in the network work. A path arrangement with full permutation consists of sixteen path setups. To meet the growing computation-intensive applications and the needs of low-power, high-performance systems, the number of computing resources in single-chip has enormously increased, because current VLSI technology can support such an extensive integration of transistors. By adding many computing resources such as CPU, DSP, specific IPs, etc to build a system in System-on-Chip, its interconnection between each other becomes another challenging issue. In most System-on-Chip applications, a shared bus interconnection which needs arbitration logic to serialize several bus access requests, is adopted to communicate with each integrated processing unit because of its low-cost and simple control characteristics. However, such shared bus interconnection has some limitation in its scalability because only one master at a time can utilize the bus which means all the bus accesses should be serialized by the arbitrator. Therefore, in such an environment where the number of bus requesters is large and their required bandwidth for interconnection is more than the current bus, some other interconnection methods should be considered. This network has a rearrange able property that can realize all possible permutations between its input and outputs. The choice of the three-stage Close network with a modest number of middle-stage switches is to minimize implementation cost, whereas it still enables a re-arrange able property for the network. Where as a path arrangement with partial permutation may consist of a subset of sixteen path setups. As designed in this network, each input sends a probe containing a 4-bit output address to find an available path leading to the target output. During the search, the probe moves forwards when it finds a free link and

moves backwards when it faces a blocked link. By means of non-repetitive movement, the probe finds an available path between the input and its corresponding idle output. The EBP-based path-setup scheme is designed with a set of probe routing algorithms The following example describes how the path setup works to find an available path by using the set of path diversity .It is assumed that a probe from a source (e.g., an input of switch 01) is trying to set up a path to a target destination (e.g., an available output of switch 22. First, the probe will non-repetitively try paths through the second-stage switches in the order of 10-11-12-13

CONCLUSION AND FUTURE WORK

To identify the state-of-the-art in the area of TRN and to find out what we know about TRNGs based on chaos, we conducted and presented in this article a systematic literature mapping. The purpose of this article is to help readers (including practitioners and researchers) conduct the most comprehensive survey in the field of TRNG based on chaos. In the end, the research results are discussed, which provides useful insights for future research directions and open issues in this field. From this study, we can draw a general conclusion that TRNGs based on chaos has obtained many successful cases, but it is still an open problem, and its solution will prove very useful for wide application. By classifying the entire body of knowledge, this survey paper “mapped” the body of knowledge on TRNGs based on chaos. We systematically classified a large set of 85 papers and investigated several review structures under three groups. The first group investigated the contribution as well as the TRNGs based on continuous time chaotic systems. The second group investigated the mappings for TRNGs based on discrete time chaotic systems. The third group investigated the TRNGs based on current-mode chaos. In recent years, it has been proved that continuous time chaotic systems can be used in the design of TRNGs. Because the number of positive Lyapunov exponents of entropy sources is limited, so hyperchaotic systems used in TRNGs is one of the important development directions in the future. It can be seen that PRNGs based on discrete-time chaos have developed from one-dimensional to two-dimensional and multi-dimensional, so the design of TRNGs using multi-dimensional discrete-time chaotic map is also the future research direction. The current-mode devices have good frequency gain characteristics and the bandwidth of these kind of devices are almost independent of gain, so there are no

ISSN2454-9940

www.ijasem.org

Vol 18, Issue.2 April 2024

need to weigh the gain and bandwidth in the design circuit, which can improve the working frequency of the circuit. Therefore, using current mode devices to realize TRNGs have gradually become a new research direction. Recently, a TRBG based on a memristive chaotic circuit was proposed in [85]. The proposed TRBG structure used a memristive canonical Chua's oscillator and a logistic mapping as the entropy source, while the XOR function was used for post-processing. It can be seen that TRBGs based on memristive chaotic system and multi-entropy sources will be an important development direction in the future. As future work, we are committed to improving the out bit rate, randomness and development cost of TRNG solutions and applications. There are three very important research groups, many of which are under development based on chaos of current mode devices or memristive chaotic system or multi-entropy sources, like combination of continuous-time chaotic system and discrete-time chaotic system. We are currently analysing how to study different solutions and other suggestions in these approaches.

REFERENCES

[1] M. Majzoobi et al., "FPGA-Based True Random Number Generation Using Circuit Metastability with Adaptive Feedback Control," Cryptographic Hardware and Embedded Systems, pp. 17-31, 2011.

[2] S. Srinivasan, et. al, "2.4GHz 7mW all-digital PVT-variation tolerant true random number generator in 45nm CMOS," VLSI Circuit, pp. 203-204, 2010.

[3] V. Fischer et al., "True random number generator embedded in reconfigurable hardware," Cryptographic Hardware and Embedded Systems, pp. 415-430, 2002.

[4] B. Sunar et al., "A Provably Secure True Random Number Generator with Built-In Tolerance to Active Attacks," IEEE Trans. Comp., vol. 58, pp. 109-119, 2007.

[5] C. Tokunaga et al.,"A True Random Number Generator with a Metastability-Based Quality Control," Proc. IEEE Int. Solid-State Circuits Conf., Digest of Technical Papers, 2007.

[6] M. Bucci et al., "A high-speed oscillator-based truly random number source for cryptographic applications on smart card IC," IEEE Trans. Comput., vol. 52, pp. 403-409, 2003.

[7] D. Schellekens, et al., "FPGA vendor agnostic TRNG," in Proc. 16th Int. IEEE Conf. Field Programmable Logic and Applications, pp. 139-144, 2006.

ISSN2454-9940

www.ijasem.org

Vol 18, Issue.2 April 2024

[8] B. Sunar, "True Random Number Generators for Cryptography," In: Cryptographic Engineering. Springer, Heidelberg, 2009.

[9] A. Rukhin et al., "Improving the Robustness of Ring Oscillator TRNGs," J. ACM Trans. Reconfigurable Technol. Syst., pp. 1-30, 2010.

[10] V. Fischer, "A Closer Look at Security in Random Number Generators Design," Third International Workshop, COSADE, pp. 167-182, 2012.

[11] D. Holcomb, W. Burleson, and K. Fu, "Power-Up SRAM State as an Identifying Fingerprint and Source of True Random Numbers," IEEE Transactions on Computers, vol. 58, pp. 1198-1210, 2009.

[12] S. Srinivasan et al., "A 4 Gbps 0.57 pJ/bit process-voltage-temperature variation tolerant all digital true random number generator in 45 nm CMOS," in 22nd IEEE Int. Conf. VLSI Design, 2009.

[13] A. Rukhin et al., "A statistical test suite for random and pseudorandom number generators for statistical applications," NIST Special Publication in Computer Security, pp. 800-822, 2001.

[14] S. Borkar, "Designing Reliable Systems from Unreliable Components: The Challenges of Transistor Variability Degradation," IEEE Micro, vol. 25, no. 6, pp. 10-16, 2005.

[15] K. Kuhn et al., "Process technology variation," IEEE Trans. Elec. Devices, vol. 58, no. 8, pp.2197 -2208, 2011.

[16] S. Kwok et al., "A Comparison of Post-Processing Techniques for Biased Random Number Generators," International Workshop, WISTP, pp. 175-190, 2011.

[17] C. Krishna et al., "Achieving High Encoding Efficiency With Partial Dynamic LFSR Reseeding," ACM Trans. Des. Auto. Elec. Syst., vol. 9, no. 4, pp. 500-516, 2004.

[18] X. Zhang et al., "Detection of trojans using a combined ring oscillator network and off-chip transient power analysis," J. Emerg. Technol. Comp. Sys., pp. 1-20, 2013.

[19] A. Strak et al., "Analysis of timing jitter in inverters induced by power-supply noise," Proc. IEEE Int. Conf. Design Test Integr. Syst. Nano. Tech., pp. 52-56, 2006.

[20] S. Krishnappa et al., "Incorporating Effects of Process, Voltage, and Temperature Variation in BTI Model for Circuit Design," IEEE Latin American Symp. on Circuits and Syst., pp. 236-239, 2010.