International Journal of Gender, Science and Technology
ISSN: 2040-0748
UGC Care Group I Journal.
Vol-13 Issue-02 Dec 2024
https://ijgst.com.2024.v13.i2.pp946-956

Design and Implementation of a Compact True Random Number Generator for Applications Relating to Root of Trust on FPGA

Mr.A.Prasad [(1)] , Mr.K.Ch.Malla Reddy [(2)] , Ubbarapu Venkata Nasaraiah [(3),] Panchakatla Siva [(4),] Cheedella Venkata Chethan [(5)]
[1,2] *Krishna Chaithanya Institute Of Technology & Sciences, Ece Department, Markapur, Andhra Pradesh.*
[3,4,5,] *Krishna Chaithanya Institute Of Technology & Sciences, UG Student-ECE, Markapur, Andhra Pradesh.*

Abstract. In this paper,various approaches have been proposed to safeguard integrated circuits (ICs) from unauthorized access and usage, or to reduce security risks. They create the groundwork for the hardware roots of trust, which rely on generators of random numbers as vital security primitives. These generators are specifically utilized to produce one-time challenges, or once's, that assist the authentication processes used by ICs to prevent potential dangers like unauthorized users accessing them. IC vendors, however, express several reservations about these solutions' complexity, citing issues with testability, design flow impact, and area overhead. These issues served as the driving force for this study, which presents a lightweight, all-digital, self-testable random number generator that is straightforward yet produces time being. It is based on a generic ring generator architecture, which is a multiple-output ring oscillator-driven variant of a linear feedback shift register that is optimized for both area and time. The suggested scheme's viability and efficiency are demonstrated by a thorough valuation that is based on three statistical test suits from the National Institute of Standards and Technology and BSI.

Keywords: TRNG with RO, FPGA, Verilog HDL, ICs.

I.INTRODUCTION

Introduction to True Random Number Generators (TRNG):
In the realm of computing and cryptography, the generation of random numbers plays a pivotal role in numerous applications, ranging from secure communications to simulation and gaming. True Random Number Generators (TRNGs) represent a class of devices or algorithms designed to produce random numbers that possess certain essential properties of randomness, such as unpredictability and uniform distribution.
Unlike Pseudo-Random Number Generators (PRNGs), which generate sequences of numbers deterministically based on initial seed values, TRNGs harness physical processes or phenomena to generate randomness inherently. This reliance on physical sources distinguishes TRNGs by providing a higher degree of randomness, often referred to as "true randomness."
TRNGs exploit various physical phenomena to generate random bits, including:
Thermal Noise: Thermal noise, also known as Johnson-Nyquist noise, arises from the random motion of electrons within a conductor at non-zero temperatures. TRNGs can sample this noise to extract random bits, as the fluctuations are unpredictable and exhibit a uniform distribution over time.

https://ijgst.com.2024.v13.i2.pp946-956

Radioactive Decay: The decay of radioactive isotopes occurs randomly and unpredictably over time. TRNGs can utilize the timing of radioactive decay events as a source of randomness, ensuring that the generated numbers are truly random.

Electronic Component Variability: Variations in electronic components, such as resistor values or semiconductor properties, can introduce randomness into electrical signals. TRNGs exploit these variations to generate random numbers by measuring analog signals or digital circuit characteristics.

Photon Arrival Times: The arrival times of photons in a photodetector can be inherently random, especially in low-intensity light conditions. TRNGs can sample these arrival times to generate random bits with high entropy.

Chaos Theory: Chaotic systems exhibit sensitive dependence on initial conditions, leading to seemingly random behavior over time. TRNGs can exploit chaotic systems, such as chaotic oscillators or chaotic maps, to generate random sequences.

The output of a TRNG typically undergoes post-processing techniques, such as whitening or conditioning, to enhance the statistical properties of the generated random numbers. Additionally, TRNGs are often subjected to rigorous testing and certification processes to ensure compliance with recognized standards for randomness, such as the National Institute of Standards and Technology (NIST) Special Publication 800-22.

TRNGs find applications in various domains where high-quality randomness is crucial, including cryptographic key generation, secure communications, numerical simulations, and gambling industries. Their ability to provide true randomness is indispensable for ensuring the security and integrity of sensitive data and systems in today's digital age.

True Random Number Generation (TRNG) with Ring Oscillators

Ring oscillators are widely used in electronic circuits for generating clock signals or oscillating waveforms. However, they can also be leveraged to create true random number generators (TRNGs) due to the inherent noise and instability present in their operation. Here's how TRNGs can be constructed using ring oscillators:

Basic Principle:

A ring oscillator consists of an odd number of inverting stages connected in a loop. The delay through each stage causes the output signal to oscillate.The frequency of oscillation is determined by the propagation delay of each stage, which can be affected by various factors such as temperature, voltage, and process variations.

Noise and Instability:

Due to manufacturing variations and environmental factors, the propagation delay of each stage in the ring oscillator fluctuates randomly.These fluctuations lead to variations in the oscillation frequency and phase, resulting in noise and instability in the output signal.

Randomness Extraction:

The variations in the oscillation frequency can be sampled and digitized to generate random bits.By measuring the time intervals between oscillation cycles or comparing the frequencies of multiple ring oscillators, random bits can be extracted from the inherent noise and instability.

Post-Processing:

The raw random bits generated by the ring oscillator TRNG may undergo post-processing techniques to enhance their statistical properties and remove biases.

Techniques such as whitening, entropy estimation, and conditioning algorithms may be applied to ensure that the output random numbers exhibit uniform distribution and high entropy.

Testing and Validation:

The performance of the ring oscillator TRNG should be thoroughly tested and validated to ensure randomness and security. Standardized statistical tests, such as the NIST Statistical Test Suite, may be applied to assess the quality of the generated random numbers and verify compliance with recognized randomness standards.

Integration and Application:

The ring oscillator TRNG can be integrated into electronic devices and systems where secure and high-quality random number generation is required.Applications include cryptographic key generation, secure communication protocols, random number seeding for simulations, and gaming applications.

Considerations:

Careful design considerations must be taken into account to mitigate potential vulnerabilities and sources of bias in the TRNG. Environmental factors, such as temperature variations and electromagnetic interference, may impact the performance and reliability of the ring oscillator TRNG and should be carefully managed.

In summary, ring oscillators offer a simple yet effective approach to generating true random numbers by exploiting the inherent noise and instability in their operation. When properly designed and implemented, ring oscillator TRNGs can provide a reliable source of randomness for a wide range of applications in security, cryptography, and data integrity.

2.LITERATURE

Foundations on which all secure operations of an integrated circuit (IC) depend is typically defined as a hardware root of trust [55]. High-end roots of trust are usually integrated into silicon as separate, custom-designed security modules—immune from malware attacks—that handle chip and device identities, cryptographic keys and functions, secure boot processes, attestation, authentication, firmware updates,etc. As a security vehicle, the hardware root of trust must be capable of detecting an intrusion, disabling access pending further actions, and/or obfuscating (camouflaging) logic operations of the IC. What lays foundation for a silicon-based fixed-function root of trust is its authentication protocol. It performs a specific set of functions, such as true random number generation, data hashing and encryption, keys validation, and logic locking [59]. As an initial part of the actual challenge-response procedure, an IC creates a random token, commonly known as a challenge or a nonce, and sends it to a secure server. The nonce is typically produced by an on-chip true random number generator (TRNG) that should yield different combinations of 0 and 1 s every time it is activated. It may also contain some individual data from the IC such as its electronic identification number.

In light of the above it is clear that TRNGs became key hardware security primitives capable of producing random sequences by harvesting the randomness present in physical processes, such as the thermal instability and noise [7], [8], [29], [39], [46], metastability [20], [25], [31], [37], [60], [61], [65], edge racing in digital designs [70], chaotic behavior of cellular automata [21], [27], [34], [45], power supply variations [58], stochastic nature of magnetic tunnel junction [64], quantum effects [56], or phase jitters in ring

oscillators (ROs) [54]. The latter approach has gained noticeable popularity because it provides a simple yet effective method to build random number generators just by chaining an odd number of inverters into a ring structure. As a result, a wide range of solutions using this principle and its derivatives have been proposed in the contemporary technical literature and industrial practice. For the sake of illustration, let us recall a few exemplary solutions.

The most straightforward mechanism to extract randomness from a jitter is to sample the output of an RO using the output signal of another RO [Fig. 1(a)]. Such coupled oscillators are presented in [2], [8], [18], [51], and [68]. In [18], two ROs are coupled by a nonlinear circuit, whereas in [68] the first RO feeds a programmable delay chain that is sampled by a bit extractor driven by the other RO. If periods of both signals are very close to each other, they form a basis for coherent sampling [32], [44], [69]. Alternatively, one can combine the output signals of several ROs by means of XOR trees [Fig. 1(b)], as shown in [3], [33], [53], [57], [66], or [67]. In particular, the work of [56] provides a thorough mathematical treatment of an approach where combined jitter signals form a source of entropy. A low power scheme where two identical ROs enable, through an XOR gate, a third RO clocking a counter is described in [10]. A reconfigurable TRNG based on transient effect ROs with two different sampling methods is introduced in [1]. In [14], four ROs drive associated LFRSs whose outputs are sampled through a multiplexer driven by yet another RO. ROs with a multistage feedback structure can be XOR-ed to produce true random numbers, as detailed in [15]. Somehow different approach is discussed in [17], [19], [23], and [42]; it implements an RO by replacing a simple circular feedback with a more complex network comprising XOR gates in a way corresponding to conventional Fibonacci or Galois LFSRs. Here, inverters replace memory elements. To enhance the performance of RO-based TRNG, one can also deploy Muller C-gates instead of inverters. These elements are then interleaved to form an asynchronous pipeline that is capable of propagating several simultaneous voltage events sampled by an XOR tree [12], [13]. Finally, different RO-based TRNGs are compared in [47] to demonstrate how they are amenable to FPGA-based implementations. We also refer the interested readers to other relevant papers, such as [6], [9], [11], [16], [36], [48], [60], and [63].

Besides the schemes recalled above, there are other techniques deployed to produce truly random sequences of bits. Those schemes include the use of chaotic maps [5] with the von Neumann correction algorithm [43], sampling a jitter in a phase-locked loop circuitry [22], extracting a design fingerprint during the power-up of SRAMs [26], or detecting a beat frequency in FPGAs [28].

In addition to its unpredictability, a modern TRNG design is expected to be compact, fast, self-testable, and easily synthesizable by using exclusively digital components [57], i.e., no amplifiers or other analog devices are allowed. Furthermore, additional post-processing steps and the corresponding circuitry to adjust the sampling frequency or to increase the per-bit entropy [53] should be avoided. Consequently, this article proposes a high-performance device that may assume the role of a lightweight all-digital TRNG. Although it was originally designed as a hardware generator of one-time challenges produced for the sake of IC authentication protocols [49], many tests

have confirmed that it can be considered as a reliable source of truly random numbers used in a variety of cryptographic or security-related applications.

II. PROPOSED METHOD AND ITS METHODOLOGY

The proposed design rests on a ring generator architecture are harvesting a source of entropy implemented by a conventional free running RO, and further processing the captured data due to its feedback network.

2.Existing System Block diagram

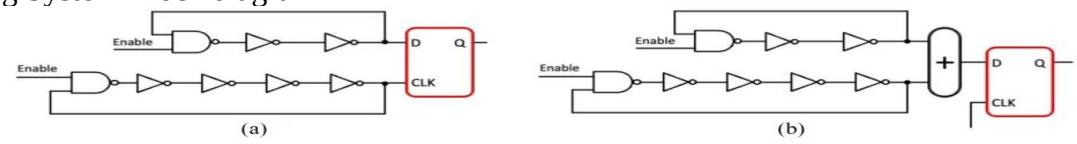


Fig. 1. Conventional RO-based TRNG architectures: (a) sampling RO by another RO and (b) combining ROs to form a single sequence.

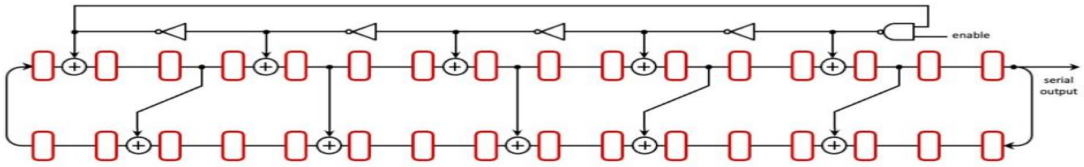


Fig. 2. Ring-generator-based TRNG with the characteristic polynomial $x^{32} + x^{27} + x^{21} + x^{16} + x^{10} + x^5 + 1$.

Explanation

A True Random Number Generator (TRNG) for applications relating to a Root of Trust is a specialized hardware or software component designed to generate truly random numbers in a secure and reliable manner. The Root of Trust is a critical component in computer security that establishes a secure foundation for various security functions. Here are the key components and considerations:

1. True Random Number Generator (TRNG):

Definition:

A TRNG is a device or algorithm that generates random numbers from unpredictable physical processes or sources, ensuring true randomness that cannot be easily predicted.

Importance in Security:

TRNGs are fundamental in cryptographic applications, ensuring the generation of unpredictable keys, initialization vectors, or nonces, which are crucial for maintaining the security of systems.

2. Root of Trust (RoT):

Definition:

The Root of Trust is the foundation of a security system. It is a trusted entity or component that is securely initialized and from which trust in the security of the entire system is derived.

Functions:

The Root of Trust is responsible for tasks such as secure bootstrapping, key management, secure storage, and the generation of cryptographic keys.

3. Applications Relating to Root of Trust:

Cryptographic Key Generation:

TRNGs are often used in Root of Trust modules to generate cryptographic keys with a high level of entropy and unpredictability, ensuring the security of cryptographic operations.

Initialization Vectors and Nonces:

TRNGs are employed to generate unpredictable initialization vectors and nonces, which are essential for secure communication protocols and cryptographic algorithms.

Secure Boot:

In the context of secure boot, the Root of Trust relies on secure and random values generated by a TRNG to establish a secure and trusted boot process.

4. Considerations and Importance:

Unpredictability:

The key feature of a TRNG is its ability to produce truly unpredictable and unbiased random numbers. This is crucial for the effectiveness of cryptographic operations.

Entropy Source:

The entropy source of the TRNG, which could be based on physical processes like electronic noise or radioactive decay, is a critical consideration. The higher the entropy, the more secure the generated random numbers.

Continuous Operation:

In applications relating to the Root of Trust, the TRNG should be designed for continuous and reliable operation, providing a steady stream of random numbers as needed.

Testing and Validation:

Rigorous testing and validation of the TRNG, including statistical analysis and entropy testing, are necessary to ensure its reliability and security.

Protection Against Attacks:

The TRNG should be designed with countermeasures against potential attacks, including those attempting to manipulate or bias the random number generation process.

Creating a Ring Oscillator-based True Random Number Generator (TRNG) involves constructing a ring oscillator circuit and extracting random bits from the fluctuations in its output. Here's a simplified example of how you might implement such a TRNG:

2.2 Ring Oscillator TRNG Circuit: Ring Oscillator Circuit:

Construct a ring oscillator using an odd number of inverting stages (typically inverters or buffers) connected in a loop.

Each stage introduces a delay, causing the signal to oscillate at a frequency determined by the cumulative delay around the loop.

The number of stages and the properties of the individual inverters/buffers can influence the oscillation frequency and randomness of the output.

Noise Injection:

Introduce sources of noise into the ring oscillator circuit to induce fluctuations in the oscillation frequency. This can be achieved by adding passive components like resistors or capacitors to the feedback path, which can exhibit random variations due to thermal noise or process variations.

Clock Signal Output:

Tap the output of the ring oscillator to obtain the clock signal waveform.

This waveform will exhibit variations in frequency and phase due to the inherent noise and instability in the oscillator circuit.

Sampling and Digitization:

Sample the clock signal at regular intervals to capture the variations in its frequency.

Use a comparator circuit or a threshold detector to convert the analog waveform into digital signals. The timing of the transitions

in the digital signal can be considered as random bits.

Post-Processing:

Apply post-processing techniques to the raw digital output to enhance its randomness and remove biases.

Techniques such as entropy estimation, whitening, and conditioning algorithms can be employed to improve the quality of the generated random numbers.

Testing and Validation:

Perform statistical tests and validation procedures to assess the randomness and quality of the generated random numbers.

Ensure that the TRNG meets recognized standards for randomness and security.

Considerations:

Oscillator Stability: Ensure that the oscillator operates within a stable range to maintain reliable oscillation while still exhibiting enough noise for randomness.

Noise Sources: Experiment with different noise sources and circuit configurations to optimize randomness while minimizing bias and non-uniformity in the generated random numbers.

Environmental Factors: Consider environmental factors such as temperature, voltage fluctuations, and electromagnetic interference that may affect the performance and reliability of the TRNG.

Security Considerations: Implement appropriate measures to protect the TRNG against potential attacks and vulnerabilities, such as physical tampering or side-channel attacks.

By carefully designing and implementing a ring oscillator TRNG with appropriate noise sources and post-processing techniques, you can create a reliable and secure source of true random numbers for various applications in cryptography, security, and data integrity.

III. RESULTS AND ANALYSIS DISCUSSION

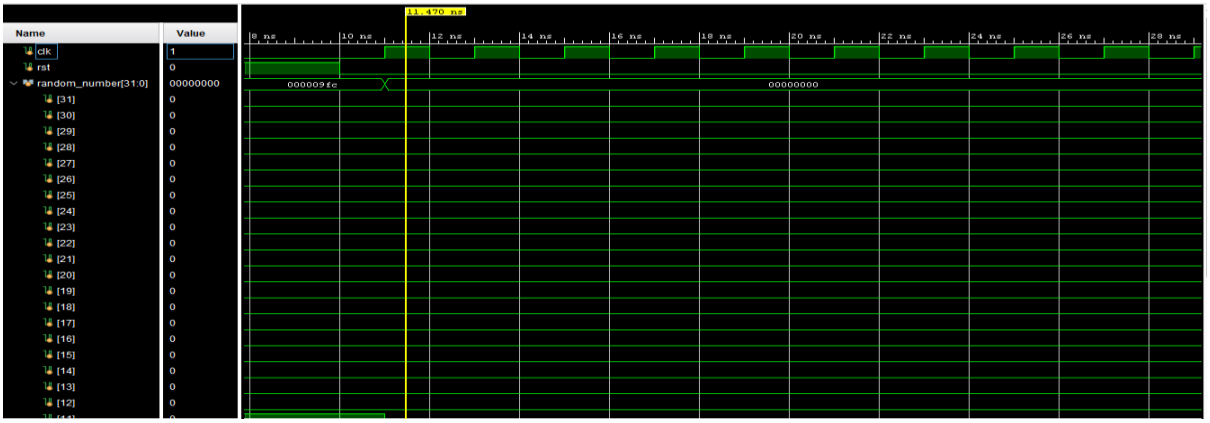


Fig6.1 TRNG with RO simulation output

International Journal of Gender, Science and Technology
ISSN: 2040-0748
UGC Care Group I Journal.
Vol-13 Issue-02 Dec 2024
https://ijgst.com.2024.v13.i2.pp946-956

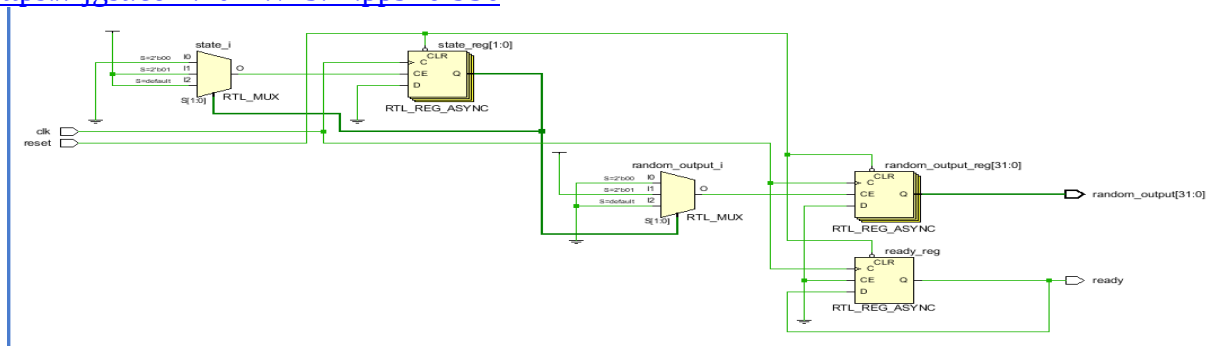


Fig.6.2.RTL schematic diagram

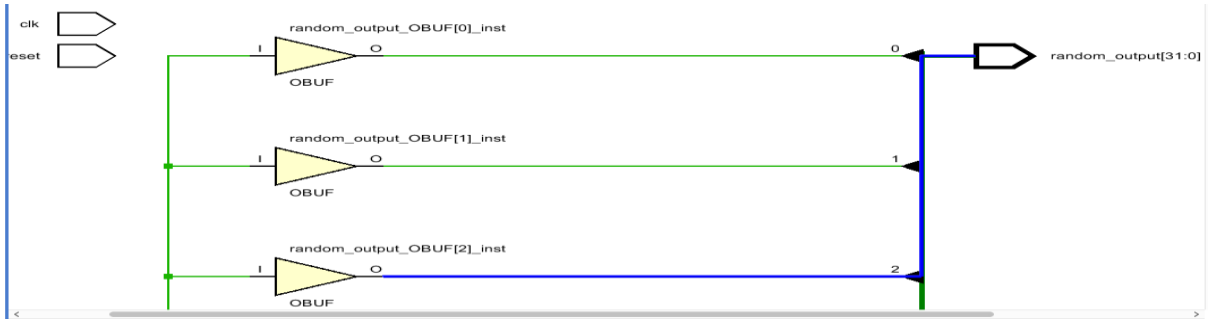


Fig.6.3.TRNG with RO synthesis design

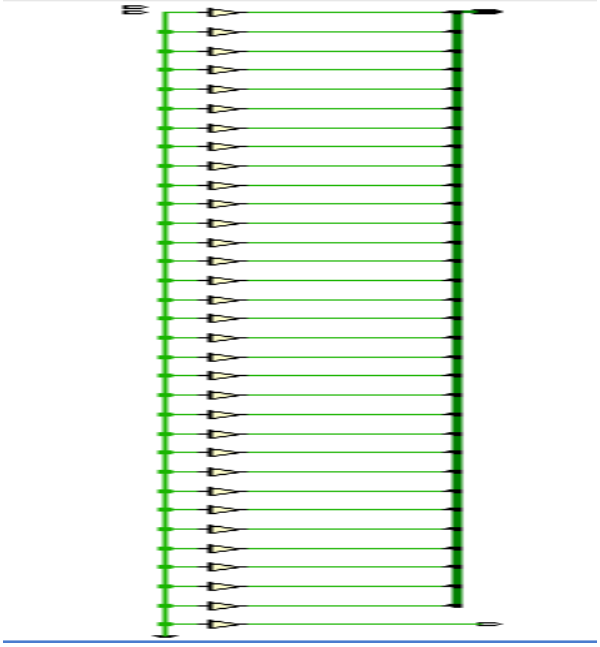
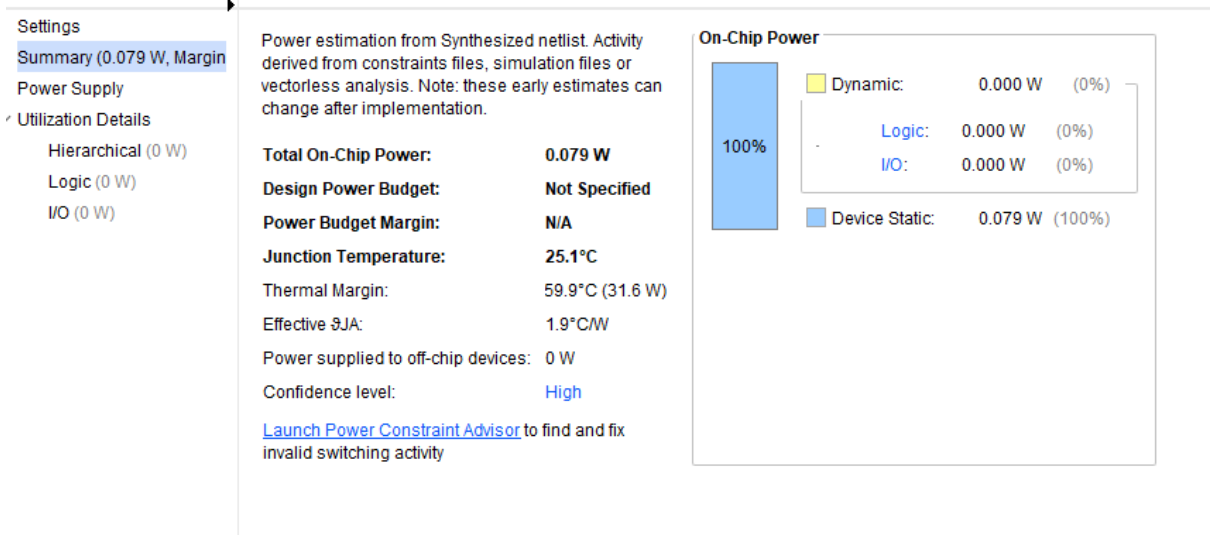


Fig 6.4.32 bit output synthesis design of TRNG.

International Journal of Gender, Science and Technology
ISSN: 2040-0748
UGC Care Group I Journal.
Vol-13 Issue-02 Dec 2024
https://ijgst.com.2024.v13.i2.pp946-956



Fig,6.5.Power report

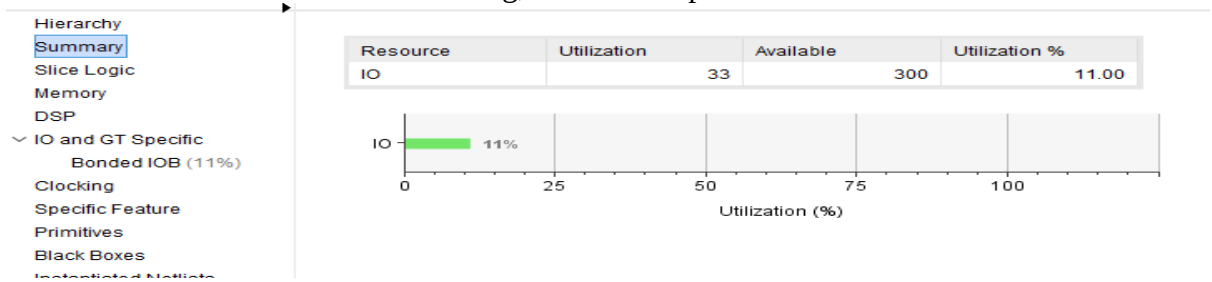


Fig 6.6. Area of LUTs Utilization

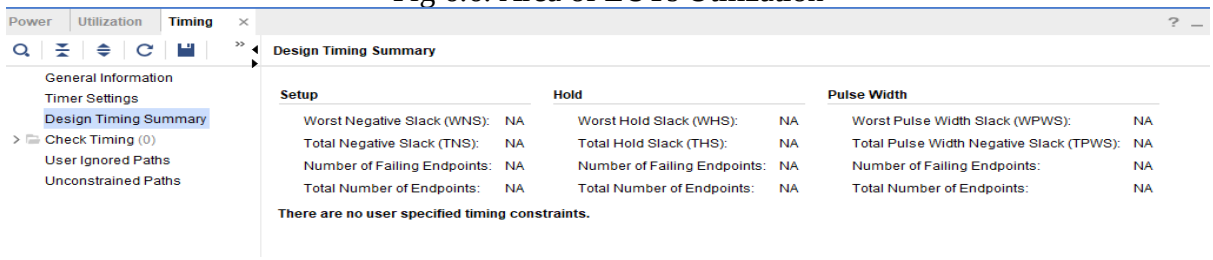


Fig 6.7. Timing Report.

CONCLUSION

This project presents a lightweight design of a nonce generator for the root of trust applications. This very compact design leverages the benefits of both the timing jitter of a single multiple-output RO and a high-speed ring generator architecture. It appears that the scheme can fulfil requirements for an efficient and secure TRNG. It is suitable for the implementation in an all-digital, standard cell-synthesis design flow which makes it a very versatile of its kind. The performance of our scheme has been studied with the help of hardware and simulation platforms. The randomness of the raw binary sequences without any postprocessing has been comprehensively tested with NIST and AIS-31 test suites showing that this ring-generator-based approach fulfils all test requirements and is able to work across a wide range of TRNG

sizes, thus, making the proposed design a scalable solution. Moreover, we have demonstrated that the hardware's circuitry is easily self-testable with respect to single stuck-at faults. Finally, in order to compare the performance of the new scheme with the existing state-of-the-art solutions, we consider six recent techniques, as they already compare favourably with other schemes introduced earlier in the technical literature.

REFERENCES

[1] B. Acar and S. Ergun, "A reconfigurable random number generator based on the transient effects of ring oscillators," IEEE Trans. Circuits Syst.II, Exp. Briefs, vol. 67, no. 9, pp. 1609–1613, Sep. 2020.
[2] T. Amaki, M. Hashimoto, and T. Onoye, "An oscillator-based true random number generator with jitter amplifier," in Proc. IEEE Int. Symp. Circuits Syst., 2011, pp. 725–728.
[3] N. N. Anandakumar, S. K. Sanadhya, and M. S. Hashmi, "FPGAbased true random number generation using programmable delays in oscillator-rings," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 67, no. 3, pp. 570–574, Mar. 2020.
[4] L. Bassham et al., "A statistical test suite for random and pseudorandom number generators for cryptographic applications," NIST, Gaithersburg, MD, USA, Rep. 800–22 Rev 1a, 2010.
[5]A. Beirami and H. Nejati, "A framework for investigating the performance of chaotic-map truly random number generators," IEEETrans. Circuits Syst. II, Exp. Briefs, vol. 60, no. 7, pp. 446–450,Jul. 2013.
[6] N. Bochard, F. Bernard, V. Fischer, and B. Valtchanov, "True randomness and pseudo-randomness in ring oscillator-based true random number generators," Int. J. Reconfig. Comput., vol. 2010, Dec. 2010,Art. no. 879281.
[7] R. Brederlow, R. Prakash, C. Paulus, and R. Thewes, "A low-power true random number generator using random telegraph noise of single oxide-traps," in IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers,2006, pp. 1666–1675.
[8] M. Bucci, L. Germani, R. Luzzi, A. Trifiletti, and M. Varanonuovo,"A high-speed oscillator-based truly random number source for cryptographic applications on a smart card IC," IEEE Trans. Comput., vol. 52,no. 4, pp. 403–409, Apr. 2003.
[9] M. Bucci and R. Luzzi, "Design of testable random bit generators," in Proc. Cryptogr. Hardw. Embedded Syst., 2005, pp. 147–156.
[10] Y. Cao, X. Zhao, W. Zheng, Y. Zheng, and C.-H. Chang, "A new energy-efficient and high throughput two-phase multi-bit per cycle ring oscillator-based true random number generator," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 69, no. 1, pp. 272–283,Jan. 2022.
[11] T. Chen, Y. Ma, J. Lin, Y. Cao, N. Lv, and J. Jing, "A lightweight full entropy TRNG with on-chip entropy assurance," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 40, no. 12, pp. 2431–2444, Dec. 2021.
[12] A. Cherkaoui, V. Fischer, A. Aubert, and L. Fesquet, "A self-timed ring based true random number generator," in Proc. Int. Symp. Asynchronous Circuits Syst., 2013, pp. 99–106.
[13] A. Cherkaoui, V. Fischer, L. Fesquet, and A. Aubert, "A very high speed true random number generator with entropy assessment," in Proc. Int. Workshop Cryptogr. Hardw. Embedded Syst., 2013, pp. 179–196.

[14] C. J. Clark, “Anti-tamper JTAG TAP design enables DRM to JTAG registers and P1687 on-chip instruments,” in Proc. Int. Symp. Hardw.- Oriented Security Trust, 2010, pp. 19–24.
[15] J. Cui et al., “Design of true random number generator based on multistage feedback ring oscillator,” IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 69, no. 3, pp. 1752–1756, Mar. 2022.
[16] R. D. Sala, D. Bellizia, and G. Scotti, “A novel ultra-compact FPGAcompatible TRNG architecture exploiting latched ring oscillators,” IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 69, no. 3, pp. 1672–1676, Mar. 2022.
[17] K. Demir and S. Ergun, “Random number generators based on irregular sampling and Fibonacci–Galois ring oscillators,” IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 66, no. 10, pp. 1718–1722, Oct. 2019.
[18] S. N. Dhanuskodi, A. Vijayakumar, and S. Kundu, “A chaotic ring oscillator based random number generator,” in Proc. Int. Symp. Hardw.- Oriented Security Trust, 2014, pp. 160–165.
[19] M. Dichtl and J. D. Golic, “High-speed true random number generation ´ with logic gates only,” in Proc. Cryptogr. Hardw. Embedded Syst., 2007,pp. 45–62.
[20] M. Epstein, L. Hars, R. Krasinski, M. Rosner, and H. Zheng, “Design and implementation of a true random number generator based on digital circuit artifacts,” in Proc. Cryptogr. Hardw. Embedded Syst., 2003,pp. 152–165.
[21] E. Farcot, S. Best, R. Edwards, I. Belgacem, X. Xu, and P. Gill, “Chaos in a ring circuit,” Chaos Interdiscipl. J. Nonlinear Sci., vol. 29, no. 4, Apr. 2019, Art. no. 043103.
[22] V. Fischer and M. Drutarovsky, “True random number generator embedded in econfigurable hardware,” in Proc. Cryptogr. Hardw. Embedded Syst., 2002, pp. 415–430.
[23] J. D. Golic, “New methods for digital generation and postprocessing ´ of random data,” IEEE Trans. Comput., vol. 55, no. 10, pp. 1217–1229,Oct. 2006.
[24] P. Hagerty and T. Draper, “Entropy bounds and statistical tests,” in Proc.NIST Random Bit Gener. Workshop, 2012, pp. 1–28.
[25] H. Hata and S. Ichikawa, “FPGA implementation of metastability-based true random number generator,” IEICE Trans. Inf. Syst., vol. E95.D,no. 2, pp. 426–436, 2012.
[26] D. E. Holcomb, W. P. Burleson, and K. Fu, “Power-up SRAM state as an identifying fingerprint and source of true random numbers,” IEEE Trans. Comput., vol. 58, no. 9, pp. 1198–1210,Sep. 2009.