JOURNAL OF SEMICONDUCTOR TECHNOLOGY AND SCIENCE, VOL. 24, NO. 3, JUNE, 2024
https://doi.org/10.5573/JSTS.2024.24.3.240

ISSN(Print) 1598-1657
ISSN(Online) 2233-4866

TRNG-PUF Integration Utilizing Programmable Delay Logics on FPGAs

Heehun Yang, Jiho Park, Jooseung Lee, Hui-Myoung Oh, Soonwoo Lee, and Hoyoung Yoo

***Abstract*—This paper introduces a novel TRNG-PUF structure using Programmable Delay Logic (PDL)-based Ring Oscillators (ROs), offering enhanced performance for both True Random Number Generators (TRNGs) and Physical Unclonable Functions (PUFs). Unlike previous approaches utilizing standard ROs, our design employs PDL to fine-tune the ROs, enabling effective harnessing of entropy for TRNGs and providing unique identification for PUFs. The proposed TRNG-PUF structure is implemented and tested on a Xilinx Artix-7 100T FPGA, demonstrating superior area efficiency and performance. In terms of hardware complexity, it showed the highest hardware efficiency among various designs. Particularly, compared to the conventional structure without shared sources, the proposed TRNG-PUF structure reduces the area of LUT and flip-flops by 41% and 24%, respectively. Moreover, the TRNG component of the structure is evaluated using the NIST SP 800-22 test, and it successfully passed all 15 tests. In contrast, previous TRNG-PUF designs only achieved partial success. Finally, the performance of the PUF is assessed through Hamming distance measurements, which showed excellent HD_{inter} and comparable HD_{intra} values. According to experimental results, the proposed TRNG-PUF structure is not only more area-efficient but also provides improved TRNG and PUF performance compared to previous TRNG-PUF designs.**

***Index Terms*—Field programmable gate array, ring oscillator, true random number generator, physically unclonable function**

I. INTRODUCTION

Secure digital systems require the development of robust and reliable hardware-based security mechanisms to counter the increasingly complex cyber security threats. True Random Number Generators (TRNGs) and Physical Unclonable Functions (PUFs) have been recognized as crucial components of such mechanisms [1, 2]. TRNGs are essential for generating unpredictable random numbers for cryptographic applications, while PUFs provide each physical hardware device with a unique and immutable identifier, ensuring the trustworthiness for hardware. More precisely, a TRNG generates unpredictable numbers by harnessing the intrinsic physical properties as entropy sources. For example, representative TRNGs provide security utilizing entropy sources like the metastable states of Flip-Flops [3, 4], SRAM [5, 6], and the clock jitter of Ring-Oscillators (ROs) [7, 8]. On the other hand, a PUF offers unique identification by capitalizing on the minute, uncontrollable variations inherent in the manufacturing process of hardware components. PUFs generate secure cryptographic keys through distinct challenge-response pairs, with various implementations such as RO PUFs [9, 10], arbiter PUFs [11], and SRAM PUF [12].

Recently, there has been a focus on combining True Random Number Generators (TRNGs) with Physical Unclonable Functions (PUFs) in order to develop a more efficient and compact architecture. The integrated structures provide advantages in terms of both area and

Manuscript received Nov. 15, 2023; reviewed Feb. 29, 2024; accepted Mar. 12, 2024

Electronics Engineering, Chungnam National University, Daejeon 305-764, Korea
E-mail : hyyoo@cnu.ac.kr

power efficiency as compared to individual implementations [13-18]. These advantages are further enhanced when the hardware source used for TRNG and PUF is same. [16] was the first to integrate a TRNG and PUF. [16] achieved this by using a Ring Oscillator as the hardware source to create the TRNG and then successively producing the PUF. It was noticeable for proposing a unified structure based on RO for the first time, but there are difficulties in operating both TRNG and PUF simultaneously due to their sequential operation. Following on, [17] and [19] introduced structures that could provide outputs for both TRNG and PUF at the same time. Like [16], these systems rely on RO, where a part of the produced bits is used as a TRNG and another piece as a PUF. This resolved the difficulty of generating bits at the same time, but also introduced a deterioration in the performance by dividing the generated bits for use.

Therefore, this paper presents a compact and versatile TRNG-PUF structure that achieves superior performance for both TRNG and PUF functionalities. In contrast to previous structures that utilized standard Ring Oscillators (RO), we have deployed a RO based on Programmable Delay Logic (PDL) to exploit the entropy source of the True Random Number Generator (TRNG) and the identification for the Physically Unclonable Function (PUF). Furthermore, PDL-based RO improves TRNG and PUF performance by accurately adjusting the PDLs. The subsequent sections of this work are structured as follows: Section 2 provides a comprehensive overview of RO-based TRNG and PUF structures, and Section 3 presents the proposed dual-mode TRNG-PUF structure using PDL-based ROs. Section 4 assesses the effectiveness of the proposed structure, while Section 5 provides the last remarks of the paper.

II. Background

1. Ring Oscillator(RO)

This section will begin by providing an overview of the fundamental functioning of a ring oscillator (RO). Subsequently, it will outline the configurations of typical RO-based TRNG and RO-based PUF. As shown in Fig. 1, a ring oscillator is a circuit consisting of an odd number of inverters, where the output of the last inverter is connected back to the input of the first inverter, forming a closed-loop configuration. In a standard ring oscillator, each inverter inverts the incoming signal, and therefore an odd number of inverters in a closed loop can create oscillatory behavior. The oscillation frequency or period is dictated by the internal delay of the inverters and the number of inverters in place. Ideally, all ROs would exhibit identical periods, but in practice, various factors such as the circuit's propagation delay, noise, thermal variations, and power supply variations induce slight deviations from the ideal period, known as jitter [20]. It is noteworthy that this jitter inherent in ROs is harnessed as an entropy source for RO-based TRNGs. Furthermore, while it may be expected that multiple identical implementations of a RO would inherently operate at the same frequency, variations in hardware implementation lead to differences, and each independently operating RO will possess unique frequency characteristics. Due to the independent nature of the hardware, the properties of each inverter may vary depending on its own physical characteristics. Fig. 1 highlights the fact that a set of *n* ring oscillators have a unique frequency in practical applications. It is essential to employ the physically

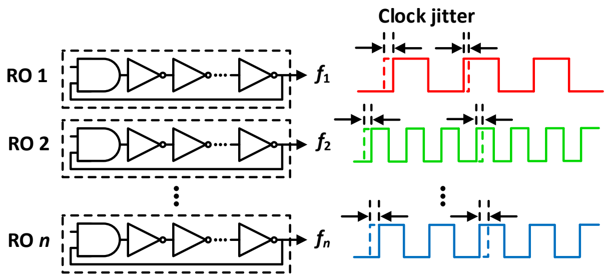


Fig. 1. Ring-Oscillator (RO).

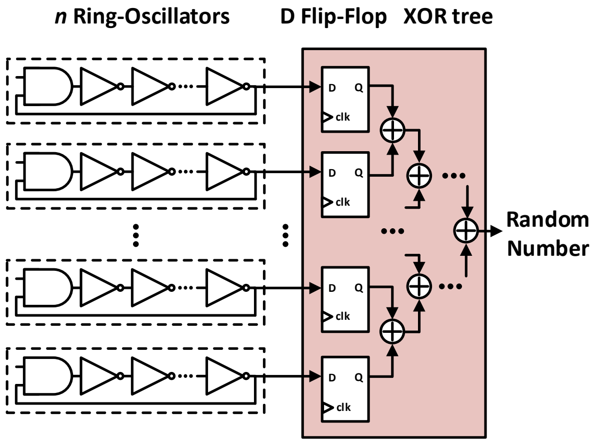


Fig. 2. Typical RO-based TRNG.

unclonable distinctive attribute of RO-PUFs.

2. RO-based TRNG

Fig. 2 depicts a typical RO-based TRNG as described in [21], where it consists of ring oscillators, D-FFs and XOR tree. First, the fundamental element of RO-based TRNG is the ring oscillator, gives rise to an oscillating signal with a certain frequency. The uniqueness of this setup lies in the jitter, which is the minute fluctuations in the oscillation frequency and period caused by a multitude of factors, such as thermal noise, supply voltage variations, and process imperfections during manufacturing. This inherent jitter is unpredictable and non-repeatable, making it an excellent source of entropy for random number generation. Secondly, the oscillating signal produced by the RO is sampled at discrete intervals using a D-Flip Flop, which is controlled by a stable reference clock. By sampling this jittery signal, we capture its randomness and convert it into a stream of random bits. However, the randomness extracted from a single RO may not be sufficient to achieve the desired level of entropy required for cryptographic applications. To address this, multiple ROs are employed, each functioning independently and contributing its own unique jitter characteristics to the overall entropy pool. Lastly, the outputs from these various ROs are then fed into an XOR tree that combines the sampled bits. This process of aggregation not only increases the overall entropy but also helps in mitigating any potential biases or patterns that might be present in the output of a single RO, thus enhancing the randomness of the final output.

3. RO-based PUF

Fig. 3 depicts a structure of RO-based PUF, where it produces a physically unclonable response when presented with a challenge input. A typical RO-based PUF is implemented using ring oscillators, counters, and a compactor. Each RO in this configuration has the role of producing a unique frequency. Each RO in this configuration has the function of producing a unique frequency. The uniqueness of each RO arises from the disparity between ideal condition and realistic implementation. Under ideal condition, it can be expected that the outputs of all the *n* ROs would be identical. However, the outputs of ROs in practical situations differ due to the independent physical nature of the circuitry. In a situation where *n* ROs possess independent frequencies, the challenge input is responsible for selecting two out of these *n* ROs. The frequencies of the selected RO pairs are counted using pairs of counters, and the resulting values are compared by the compactor. Finally, this comparison results in a binary response of either 0 or 1, depending on the outcome of the comparison. It is highly noteworthy that the response varies for each hardware due to the minute differences in hardware characteristics, similar to the individuality of a human fingerprint, which represents the distinct nature of each hardware device.

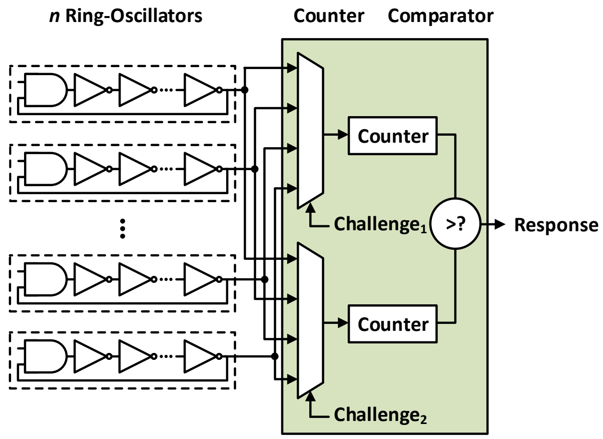


Fig. 3. Block diagram of the proposed transmitter.

III. PROPOSED TRNG-PUF

In this section, we introduce a compact and versatile TRNG-PUF structure that demonstrates improved performance for both TRNG and PUF functionalities. Diverging from conventional designs that employ standard RO, we utilize a RO that has been developed with Programmable Delay Logic (PDL). It enables us to effectively harness the entropy source essential for TRNG and facilitates the unique identification capabilities for PUF. Initially, we delve into the principles of PDL, followed by an in-depth explanation of the intricate architecture of our proposed TRNG-PUF system based on PDL. First, we explore the fundamental principles of PDL. Then, we provide a detailed explanation of the proposed TRNG-PUF based on PDL

1. Programmable Delay Logic (PDL)

Recently, Programmable Delay Logic (PDL) has been widely applied in various security implementations due to its ability to adjust the delay of logic gates. For example, when implementing a RO as depicted in Fig. 1, we can use a PDL-based NOT gate instead of a conventional NOT gate, which enables fine-tuning of the delay for each individual gate, allowing for greater optimization in hardware configuration. Typically, PDL logic is predominantly implemented in Field Programmable Gate Arrays (FPGAs). When constructing logic elements in FPGAs, Look-Up Tables (LUTs) are mainly used, as illustrated in Fig. 4. A LUT in a FPGA consists of SRAM, which holds logic configuration information, and a MUX tree that determines the output. Depending on the values stored in the SRAM, the LUT can perform various logical operations such as AND, OR, NOT, etc. Fig. 4 demonstrates an example of implementing a NOT gate, where the configuration of a 6-input LUT performs the operation $O = \overline{I_0}$. In this configuration, when the input I_0 is set to 0, the output O is consistently 1. Conversely, when I_0 is set to 1, the output O is consistently 0. This behavior is determined by the SRAM and MUX. Furthermore, inputs I_1 to I_5 are utilized to program a delay of the NOT gate. In this 6-input LUT, selections from $\{I_1, I_2, I_3, I_4, I_5\}$ = 5'b00000 to 5'b11111 are possible. The values chosen by the PDL do not alter the logical value of the output, but they can adjust the paths within the LUT, as depicted in Fig. 4. When PDL is applied to the conventional RO in Fig. 1, standard NOT gates can be substituted with PDL-based NOT gates. By utilizing these path variations, the characteristics of PDL allow for the fine-tuning of the RO.

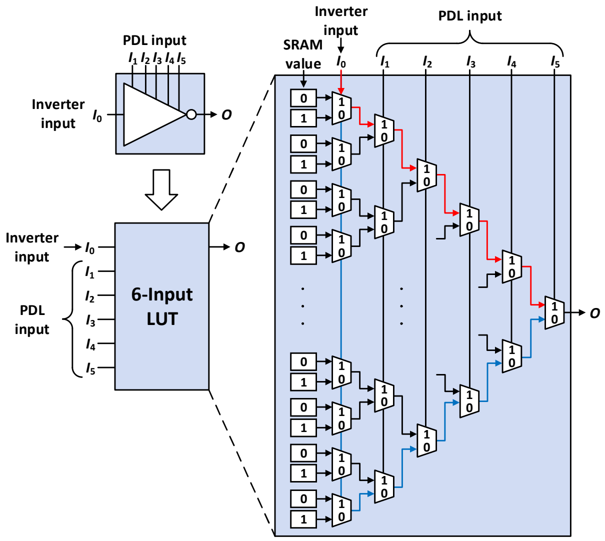


Fig. 4. Programmable Delay Logic (PDL).

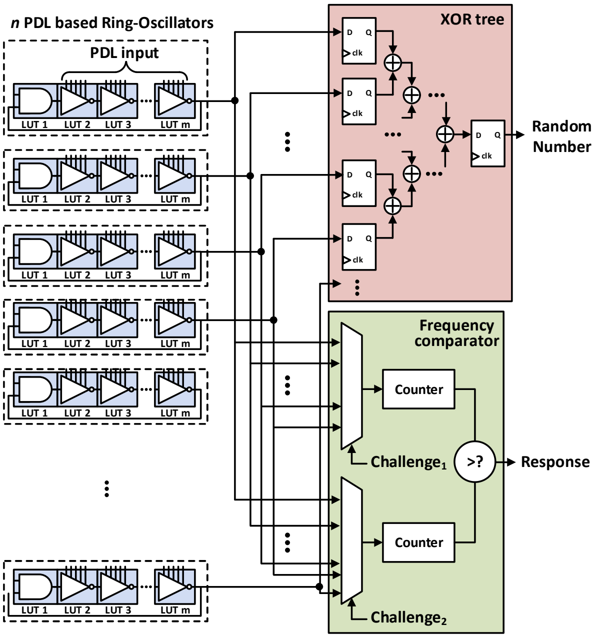


Fig. 5. Proposed PDL-RO-based TRNG-PUF design.

2. Detailed Proposed Design

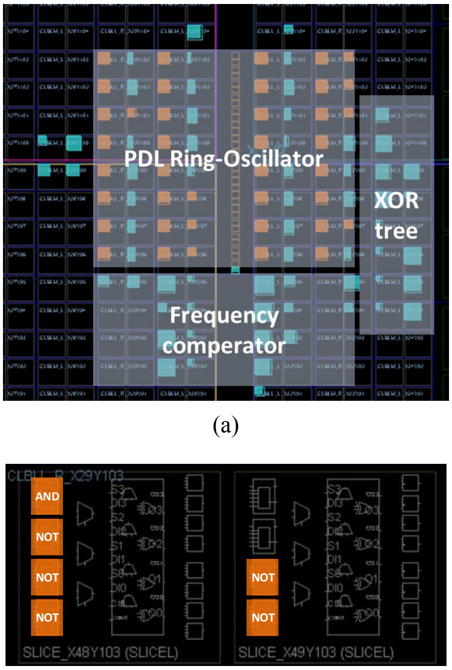
Fig. 5 illustrates the PDL-RO-based TRNG-PUF structure proposed in this paper. Unlike previous TRNG-PUF structures [16, 17] that used conventional ROs, we employ PDL-based ROs to achieve superior performance and provide a parallel channel to enable the simultaneous operation of TRNG and PUF. The proposed design consists of n sets of ROs and circuits for TRNG and PUF with the use of shared ROs. The shared PDL-based ROs serve as a source of entropy based on jitter in TRNG mode, and offer distinct frequencies in PUF mode.

To elaborate, in our proposed TRNG-PUF structure, TRNG generation operates as follows: At first, the PDL-based RO produces an oscillating signal with unique frequency variations, known as jitter, due to factors like thermal noise and manufacturing process. The proposed TRNG-PUF offers superior jitter optimization by using PDL-based ROs compared to the previous TRNG-PUFs. The subsequent steps are similar to the standard TRNG procedures as shown in Fig. 2. The oscillating signal is first sampled using a D-Flip Flop and then merged using an XOR tree, generating random bits. Furthermore, in our proposed TRNG-PUF structure, PUF generation operates as follows: At first, the PDL-based ROs are used to generate distinct frequencies. Since the PDL-based RO can adjust each delay, the PDL-based RO has the capability to regulate the generation of different frequencies in order to maximize the performance of PUF, which is not possible with a regular RO. The subsequent process is similar to typical PUF operations as shown in Fig. 3, where a challenge selects ROs, followed by counting and comparison to generate the PUF response.

Consequently, our proposed structure utilizes shared PDL-based ROs for both TRNG and PUF, employing them as an entropy source and for their independent hardware characteristics, leading to an area-efficient design. Additionally, by implementing PDL-based ROs instead of standard ROs, we provide better quality sources for TRNG and PUF, resulting in enhanced performance. It is logical that PDL-based ROs, being capable of fine-tuning in response to the various variations in FPGA, are expected to outperform standard ROs, which are set arbitrarily during the place and route process. Finally, by implementing TRNG and PUF structures in parallel, it becomes feasible to achieve simultaneous operation through a single PDL-based RO, contributing to overall system performance enhancement.

IV. Experimental Results

In order to verify the advantages of the proposed structure, the proposed TRNG-PUF structure is implemented on a Xilinx Artix-7 100T FPGA using the Xilinx Vivado 2020.2 EDA tool. Fig. 6(a) displays the results of implementing the proposed TRNG-PUF, and Fig. 6(b) illustrates the detailed implementation of the PDL-based RO. As shown in Fig. 6(b), the proposed TRNG-PUF design utilizes 32 ROs, each consisting of 5 inverters and 1 AND gate.



(a)

(b)

Fig. 6. (a) Proposed TRNG-PUF implementation; (b) Proposed PDL-RO implementation.

Table 1. Hardware complexity comparison

Metric	Conventional	Proposed	Improvement*
FPGA	Artix-7	Artix-7	-
LUT	547(LUT6)	323(LUT6)	40.95%
FF	133	101	24.06%
Slice	171	111	35.09%

***Improvement = (Conv. - Proposed) / Conv.**

The PDL-based ring oscillator offers the flexibility to finely adjust to a variety of environmental factors, such as temperature shifts, voltage changes, and noise, thereby optimizing or maximizing the performance of TRNGs and PUFs. Hence, setting the inputs for the PDL is a critical factor for the proposed TRNG-PUF design. For experiments, we first screened potential candidates based on TRNG performance from all combinations of PDL settings. Then, from those satisfying our TRNG performance criteria, we selected the final PDL input configuration that provided the highest PUF performance. As a result, we were able to select a PDL input that ensures superior performance for both TRNG and PUF.

1. Hardware Complexity

Table 1 compares the hardware complexity of various TRNG-PUF structures, including conventional, previous [16], previous [17, 18] and the proposed structure. The conventional structure refers to an arrangement where TRNG as shown in Fig. 2 and PUF as shown in Fig. 3 are implemented separately without hardware sharing. Previous [16] represents the first RO-based TRNG-PUF

Table 2. TRNG performance comparison (NIST SP 800-22)

Statistical test	**Previous [18]**		**Proposed**	
	P-Value	**Proportion**	**P-Value**	**Proportion**
Frequency	-	0.53	0.68	**1.00**
BlockFrequency	-	0.85	0.13	**0.96**
CumulativeSums	-	0.54	0.71	**0.99**
Runs	-	0.81	0.40	**0.99**
LongestRun	-	**0.97**	0.44	**0.97**
Rank	-	**0.99**	0.30	**0.99**
FFT	-	**0.98**	0.99	**0.99**
NonOverlappingTemplate	-	0.85	0.51	**0.99**
OverlappingTemplate	-	**0.91**	0.37	**1.00**
Universal	-	**0.98**	0.35	**0.99**
ApproximateEntropy	-	**0.94**	0.68	**0.97**
RandomExcursions	-	**0.97**	0.39	**0.99**
RandomExcursionVariant	-	**0.97**	0.36	**0.99**
Serial	-	**0.97**	0.24	**0.98**
LinearComplexity	-	**1.00**	0.98	**0.99**
TRNG Performance	**Partial Pass(10/15)**		**Complete Pass(15/15)**	

structure, while previous [17, 18] signifies a structure that achieves TRNG-PUF by splitting counting bits. The proposed TRNG-PUF shown in Fig. 5, unlike previous structures that use standard RO, utilizes and shares PDL-based ROs. Note that we implemented both the conventional and proposed structures on Xilinx Artix7, and the results for previous structures were extracted from [16-18]. Experimental results indicate that the proposed TRNG-PUF is the most area-efficient structure by sharing PDL-based ROs. The proposed TRNG-PUF structure shares PDL-based ROs, exploiting them as an entropy source for TRNG and as unique hardware nature for PUF, respectively. Compared to the conventional structure without shared sources, the proposed design reduced the area of LUT and flip-flops by 41% and 24%, respectively.

2. TRNG Performance

To assess the performance of the TRNG, we carried out the NIST SP 800-22 test [22]. We conducted the NIST SP 800-22 test by performing 100 iterations using 1,000,000-bit bitstreams, resulting in a total of 100,000,000 random number bits created. Table 2 shows the results of the NIST SP 800-22 test. The NIST SP 800-22 test consist of 15 comprehensive tests designed to assess the level of randomness. The P value is a statistical measure that quantifies the likelihood of obtaining results, and the proportion indicates the ratio of passing the test. If the P-value exceeds 0.01 and the percentage is more than 0.96, the NIST SP 800-22 classifies the bitstream as compatible with TRNG standards. According to experimental results, while the previous TRNG-PUF [18] failed to pass all the tests, the proposed TRNG-PUF successfully passed all 15 tests. This can be attributed to the to the optimization of randomness by adjusting the parameters of PDL and and using parallel processing of TRNG and PUF structures, resulting in no influence on the performance of either TRNG or PUF.

3. PUF Performance

PUFs are characterized by their uniqueness, reproducibility, and randomness, and this paper assesses uniqueness through inter Hamming distance (HD_{inter},) reproducibility through intra Hamming distance (HD_{intra},) and randomness through the autocorrelation function (ACF). Note that the Hamming Distance quantifies the disparity in bits between two bitstreams. We computed HD_{inter}, which measures the performance of the PUF across several FPGAs, and HD_{intra}, which measures the performance of the PUF within the same FPGA. The values of HD_{inter} and HD_{intra} can be calculated as

$$HD_{inter} = \frac{2}{N(N-1)} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \frac{HD(R_i, R_j)}{k}, \quad (1)$$

$$HD_{intra} = \frac{1}{I} \sum_{i=1}^{I} \frac{HD(R_i, R_F)}{k}, \quad (2)$$

where N and I denote the quantity of devices and the number of iterations, respectively. Meanwhile, k represents the overall length of the response bits. R_i, R_j, and R_F refer to the i-th and j-th response bits, as well as the reference response bit, respectively. It is important to note that HD_{inter} value of 50% indicates the optimal performance, meaning that separate PUFs are generated independently regardless of the FPGA, while HD_{intra} value of 0% indicates that independent PUF performance is achieved within the same FPGA over several iterations. Table 3 presents a comparison of the performance of PUFs among different designs. A total of 200,000,000 responses were generated for all challenges in order to calculate the Hamming distance. According to

Table 3. PUF performance comparison

Metric	Previous [16]	Previous [17]	Proposed
HD_{Inter}	42.80 %	49.15 %	49.93 %
HD_{Intra}	19.40 %	2.05 %	4.17 %
Performance*	0.01	0.57	3.43

***Performance** = 1 / ($|HD_{inter}$-50$|\times HD_{intra}$)

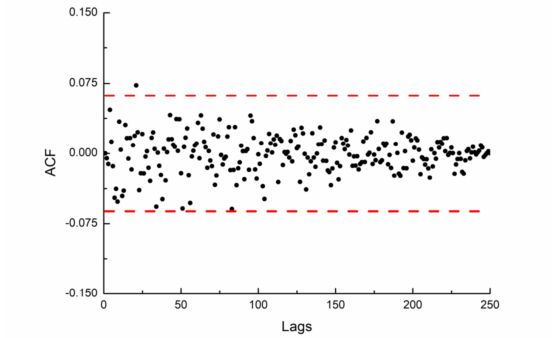


Fig. 7. Autocorrelation test result.

experimental results, the proposed TRNG-PUF achieved an HD_{inter} of 49.93% and an HD_{intra} of 4.17%. Compared to previous TRNG-PUF structures, it demonstrates superior performance in HD_{inter} and nearly similar levels in HD_{intra}.

Furthermore, to verify the randomness of the extracted PUF bits, we calculate the ACF as (3) similar to [15].

$$r_x = \frac{1}{T \times \sigma}\sum_{t=1}^{T-k}(y_t - \mu)(y_{t+k} - \mu)\,, \tag{3}$$

where T is the length of the bits, μ is the mean, σ represents the variance, y_t is the bit at position t, and k denotes the lag. Fig. 7 shows the results of the ACF test for the extracted PUF bits. The experimental results demonstrate that the extracted bits are distributed within the 95% confidence interval (CI) value of 0.0620, indicating no significant autocorrelation. As a result, the proposed design is capable of generating a PUF bitstream with high levels of uniqueness, reproducibility, and randomness.

V. Conclusions

In this paper, we introduce a new TRNG-PUF structure using PDL-based ROs. This novel design distinguishes itself from the previous designs by using PDL for precise adjustment of ROs. It enables us to effectively harness the entropy source essential for TRNG and facilitates the unique identification capabilities for PUF. Our structure is successfully implemented and evaluated on a Xilinx Artix-7 100T FPGA. According to experimental results, it shows significant reduction in area by sharing the PDL-based ROs. In addition, The proposed TRNG-PUF structure successfully pass all 15 tests of the NIST SP 800-22 standard, outperforming previous designs that only achieved partial success. This highlights its robustness in generating truly random numbers. Lastly, the performance of the PUF is evaluated using Hamming distance measures, demonstrating its excellent performance. It emphasize the reliability and uniqueness of our PUF responses. As a result, the proposed TRNG-PUF design can be a potential candidate as an efficient and secure solution in the realm of hardware-based cryptography.

Acknowledgments

This research was supported by Korea Electrotechnology Research Institute (KERI) Primary research program through the National Research Council of Science & Technology (NST) funded by the Ministry of Science and ICT (MSIT) (No. 24A01045)

References

[1] Sergei P. Skorobogatov, “Semi-invasive attacks-A new approach to hardware security analysis,” No. UCAM-CL-TR-630. University of Cambridge, Computer Laboratory, Apr. 2005.

[2] Y. Yang, L. Wu, G. Yin, L. Li and H. Zhao, "A Survey on Security and Privacy Issues in Internet-of-Things," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1250-1258, Oct. 2017.

[3] F. Frustaci, F. Spagnolo, S. Perri and P. Corsonello, "A High-Speed FPGA-Based True Random Number Generator Using Metastability With Clock Managers," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 70, no. 2, pp. 756-760, Feb. 2023.

[4] Lee, Donggeon, Hwajeong Seo, and Howon Kim. "Metastability-based feedback method for enhancing fpga-based trng," *International Journal*

of Multimedia and Ubiquitous Engineering 9.3, pp. 235-248, 2014.

[5] D. E. Holcomb, W. P. Burleson and K. Fu, "Power-Up SRAM State as an Identifying Fingerprint and Source of True Random Numbers," *IEEE Transactions on Computers*, vol. 58, no. 9, pp. 1198-1210, Sept. 2009.

[6] Van der Leest, Vincent, et al. "Efficient implementation of true random number generator based on sram pufs," *Cryptography and Security: From Theory to Applications,* Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 300-318, 2012.

[7] B. Sunar, W. J. Martin and D. R. Stinson, "A Provably Secure True Random Number Generator with Built-In Tolerance to Active Attacks," *IEEE Transactions on Computers*, vol. 56, no. 1, pp. 109-119, Jan. 2007.

[8] N. Nalla Anandakumar, S. K. Sanadhya and M. S. Hashmi, "FPGA-Based True Random Number Generation Using Programmable Delays in Oscillator-Rings," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 3, pp. 570-574, Mar. 2020.

[9] Suh, G. Edward, and Srinivas Devadas. "Physical unclonable functions for device authentication and secret key generation." *Proceedings of the 44th annual design automation conference*, 2007.

[10] C. Marchand, L. Bossuet, U. Mureddu, N. Bochard, A. Cherkaoui and V. Fischer, "Implementation and Characterization of a Physical Unclonable Function for IoT: A Case Study With the TERO-PUF," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 1, pp. 97-109, Jan. 2018.

[11] Daihyun Lim, J. W. Lee, B. Gassend, G. E. Suh, M. van Dijk and S. Devadas, "Extracting secret keys from integrated circuits," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 13, no. 10, pp. 1200-1205, Oct. 2005

[12] Guajardo, Jorge, et al. "FPGA intrinsic PUFs and their use for IP protection." *Cryptographic Hardware and Embedded Systems-CHES 2007,* 9th International Workshop, Vienna, Austria, Sep. 2007.

[13] S. Taneja, V. K. Rajanna and M. Alioto, "In-Memory Unified TRNG and Multi-Bit PUF for Ubiquitous Hardware Security," *IEEE Journal of Solid-State Circuits*, vol. 57, no. 1, pp. 153-166, Jan. 2022.

[14] S. K. Satpathy et al., "An All-Digital Unified Physically Unclonable Function and True Random Number Generator Featuring Self-Calibrating Hierarchical Von Neumann Extraction in 14-nm Tri-gate CMOS," *IEEE Journal of Solid-State Circuits*, vol. 54, no. 4, pp. 1074-1085, Apr. 2019.

[15] B. Gao, B. Lin, X. Li, J. Tang, H. Qian and H. Wu, "A Unified PUF and TRNG Design Based on 40-nm RRAM With High Entropy and Robustness for IoT Security," *IEEE Transactions on Electron Devices*, vol. 69, no. 2, pp. 536-542, Feb. 2022.

[16] Maiti, Abhranil, et al. "Physical unclonable function and true random number generator: a compact and scalable implementation." *Proceedings of the 19th ACM Great Lakes symposium on VLSI,* 2009.

[17] F. Kodýtek, R. Lórencz, and J. Buček. "Improved ring oscillator PUF on FPGA and its properties," *Microprocessors and Microsystems*, vol. 47, pp. 55-63, 2016.

[18] S. Buchovecká, et al. "True random number generator based on ring oscillator PUF circuit," *Microprocessors and Microsystems*, vol. 53, pp. 33-41, 2017.

[19] I. Baturone, R. Román and Á. Corbacho, "A Unified Multibit PUF and TRNG Based on Ring Oscillators for Secure IoT Devices," *IEEE Internet of Things Journal*, vol. 10, no. 7, pp. 6182-6192, 1 Apr. 2023.

[20] J. A. McNeill, "Jitter in ring oscillators," *IEEE Journal of Solid-State Circuits*, vol. 32, no. 6, pp. 870-879, Jun. 1997.

[21] K. Wold, and CH. Tan, "Analysis and enhancement of random number generator in FPGA based on oscillator rings," *International Journal of Reconfigurable Computing*, pp. 1-8, Jan. 2009.

[22] A. Rukhin, et al., “A statistical test suite for random and pseudorandom number generators for cryptographic applications,” *US Department of Commerce, Technology Administration, National Institute of Standards and Technology*, vol. 22, 2001.

Heehun Yang received the B.S. degree in electronics engineering from Chungnam National University, Daejeon, South Korea, in 2020, where he is currently working toward the M.S. degree. His current research interests include embedded systems hardware security, evaluation of true random number generators and physical unclonable functions aimed at cryptographic applications, FPGA platform, VLSI for DSP.

Jiho Park received the B.S. degree in electronics engineering from Chungnam National University, Daejeon, South Korea, in 2023, where he is currently working toward the M.S. degrees. His current interests include embedded systems hardware security, evaluation of true random number generators and physical unclonable functions aimed at cryptographic applications.

Jooseung Lee received the B.S. degree in electronics engineering from Sogang University, Seoul, South Korea, in 2014, and the M.S. degree in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2016. Since 2019, he has been with the Power Grid Research Division, Korea Electrotechnology Research Institute (KERI). Prior to joining KERI, he was with Samsung Electronics, Hwasung, South Korea, where he was involved in the research of NAND flash memories controllers. His current research interests include algorithms and implementations for smart grids and electric vehicle charging systems.

Hui-Myoung Oh received the B.S. degree in electrical engineering from the Yonsei University, Seoul, South Korea in 1998, and the M.S. and the Ph.D. degrees in electrical and electronic engineering from the same university in 2000 and 2009, respectively. He was a researcher in the Korea Electrotechnology Research Institute (KERI) from 2001 to 2005 and a senior researcher from 2006 to 2015, and has been working as a principal researcher in the same institute since 2016. His research interests include digital communication systems, digital twin systems, EV communication protocols, and smart grids based on renewable energy.

Soonwoo Lee received his Ph.D degrees in mechatronics engineering from Korea University, Seoul, South Korea in 2018. He has been with the Korea Electrotechnology Research Institute (KERI) since 2005 and is currently a principal researcher in the Power ICT Center. His research interests include signal processing, digital control, and digital circuit design for power utility and smart grid applications.

Hoyoung Yoo received the B.S. degree in electrical and electronics engineering from Yonsei University, Seoul, South Korea, in 2010, and the M.S., and Ph.D. degrees in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2012 and 2016, respectively. Since 2016, he has been with the Department of Electronics Engineering, Chungnam National University (CNU), Daejeon, where he is currently an Associate Professor. Prior to joining CNU, in 2016, he was with Samsung Electronics, Hwasung, South Korea, where he was involved in the research of nonbinary LDPC decoders for NAND flash memories. His current research interests include algorithms and architectures for errorcorrecting codes, FPGA reverse engineering, GNSS communication, and 5G communication systems.