Randomness Assessments for PDL-based TRNG on FPGA

Heehun Yang, Jiho Park, and Hoyoung Yoo
Dept. of Electronics Engineering
Chungnam National University
Daejeon, Korea
hhyang.cas@gmail.com, jhpark.cas@gmail.com, hyyoo@cnu.ac.kr

Abstract

This paper investigates the enhancement of True Random Number Generators (TRNGs) in Field-Programmable Gate Arrays (FPGAs) through Programmable Delay Logic (PDL). We examine how PDL influences the entropy quality and performance of TRNGs by controlling Ring Oscillators (ROs). Our study, conducted using Xilinx Artix-7 7A100T and Kintex Ultrascale KU040 devices, utilizes NIST SP 800-22 tests to evaluate different PDL settings. Findings demonstrate that precise PDL configurations are crucial for optimizing TRNG performance and security. This research underlines the importance of PDL in developing robust FPGA-based security systems, offering significant insights into effective digital system security enhancements.

Keywords: FPGA, Xilinx, TRNG, Programmable Delay Logic.

1. Introduction

FPGA (Field Programmable Gate Array) is a technology that has been widely used in various industrial sectors. This is due to the programmable flexibility that FPGAs offer compared to ASICs (Application-Specific Integrated Circuits). FPGAs provide the ability to modify hardware functions in the same manner as software, which significantly reduces the time and expense required for product design and testing. Particularly, FPGAs are ideal for small to medium scale production and offer the advantage of rapid prototyping and reduced time to market with low initial investment. On the other hand, ASICs are designed for specific applications and are suitable for mass production but come with the disadvantages of high initial costs and long development cycles. Due to these characteristics of FPGAs, their use is increasing in various application areas such as high-speed operations in communication systems, real-time processing in the automotive industry, and sectors like defense, aerospace, medical imaging, and financial systems. This emphasizes the importance of research and development in digital circuit designs.

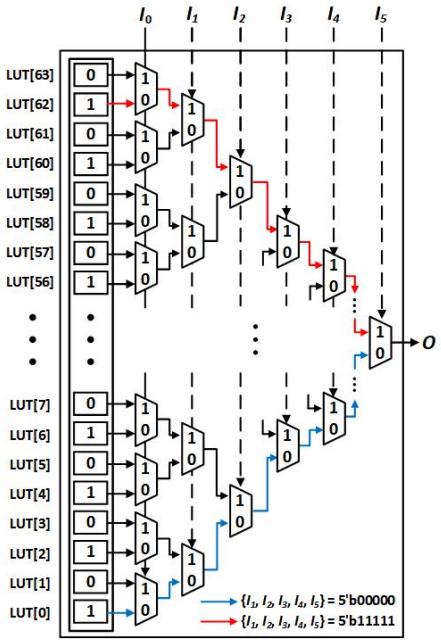

Figure 1. Implementation of NOT gate using PDL

Furthermore, in modern digital system design, ensuring security has become an essential aspect beyond fundamental operations. Secure digital systems necessitate the development of robust and reliable hardware-based security mechanisms to address the growing complexities of cybersecurity threats. True Random Number Generators (TRNGs) have emerged as vital components within these security frameworks [1, 2]. TRNGs are crucial for generating random numbers that are unpredictable and necessary for cryptographic applications. Specifically, TRNGs create these unpredictable numbers by utilizing the natural physical properties as sources of entropy. For instance, TRNGs are widely implemented by exploiting entropy sources such as the metastable states of Flip-Flops [3, 4], SRAM [5, 6], and the clock jitter of Ring-Oscillators

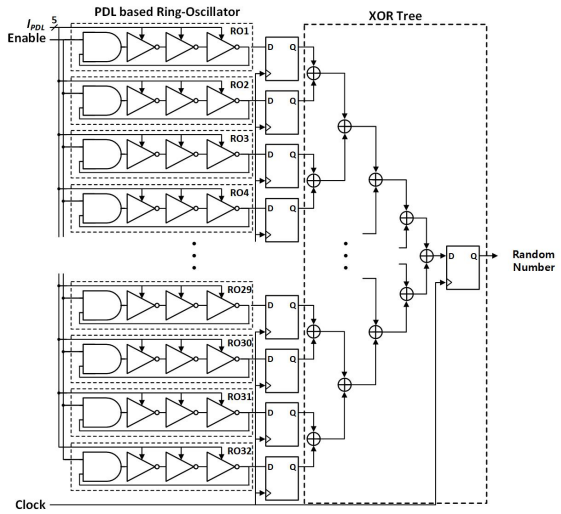

Figure 2. Proposed PDL-based TRNG

(ROs) [7, 8]. In this paper we aim to analyze the performance of TRNGs according to different PDL inputs and evaluate the randomness of TRNG outputs across various PDL inputs. This will establish the significance and enhance the validity of the PDL-based TRNG structure. The paper is organized as follows: Section 2 discusses the FPGA-based PDL addressed in this paper. Section 3 describes the structure of PDL-based TRNGs, and Section 4 evaluates the randomness of TRNGs across different PDLs using NIST 800-22 across Xilinx Artix and Kintex FPGAs. Finally, Section 5 presents conclusion of this paper.

2. Background

In general, within FPGAs, logic gates are implemented using Look-up tables (LUTs) and MUX trees as shown in Figure 1. The values stored in the LUT and the control inputs of the MUX determine the formation of various logic gates such as AND, OR, and NOT. This is the key structure that provides FPGA's reconfigurability unlike ASICs. For instance, to implement a given LUT as $O = \overline{I_0}$, one would input 64h'5555_5555_5555_5555 into the LUT and sequentially connect MUX signals to I_0, I_1, I_2, I_3, I_4, and I_5. With such an implementation, when I_0 is input as 0, the LUT value 1 is always selected, and when I_0 is input as 1, the LUT value 0 is always selected, allowing it to function as a NOT gate. Delving deeper into the LUT inputs, I_0 determines the inverting value, while the remaining inputs I_1 to I_5 decide the path through the LUT. According to Figure 1, when $\{I_5, I_4, I_3, I_2, I_1\}$ = 5'b00000 and I_0 = 1'b0, LUT[0]'s value 1 is chosen, setting the path along the blue line, and when $\{I_5, I_4, I_3, I_2, I_1\}$ = 5'b11111, LUT[62]'s value 1 is chosen, setting the path along the red line. Hence, changes in the path do not affect the final output of Boolean equation but introduce variations in the output delay. This circuit design method utilizing such variations is referred to as a PDL-based circuit.

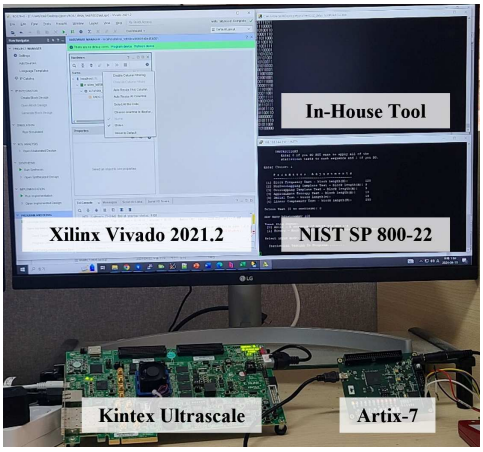

Figure 3. Experimental environments

3. PDL-based TRNG

Figure 2 illustrates a PDL-based TRNG consisting of ring oscillators, D-FFs, and an XOR tree. The core component, the ring oscillator, generates an oscillating signal with inherent frequency jitter due to factors like thermal noise, supply voltage variations, and manufacturing imperfections [9]. This jitter, being unpredictable and non-repeatable, serves as a robust source of entropy for random number generation. The signal from the RO is periodically sampled by a D-Flip Flop controlled by a stable clock, capturing its randomness in the form of a binary sequence. To enhance the entropy level suitable for cryptographic uses, multiple ROs are used, each adding its unique jitter. The outputs from these ROs are combined in an XOR tree, which increases the overall entropy and reduces biases, thus improving the randomness of the final output. Note that the primary distinction between conventional TRNGs and PDL-based TRNGs lies in the use of PDL logic to control the Ring Oscillators (ROs) that determine the quality of randomness, instead of using standard methods. By controlling the PDL logic to provide optimal PDL inputs in PDL-based TRNGs, it is possible to maximize system performance. According to previous works [8], PDL-based TRNGs can offer better quality entropy sources compared to traditional TRNGs, resulting in enhanced performance by implementing PDL-controlled ROs instead of standard ROs. Consequently, previous researches have successfully proposed structures for PDL-based TRNGs that enhance TRNG performance within FPGAs. However, the previous papers failed to presents the importance of selecting the appropriate PDL inputs for PDL-based TRNGs.

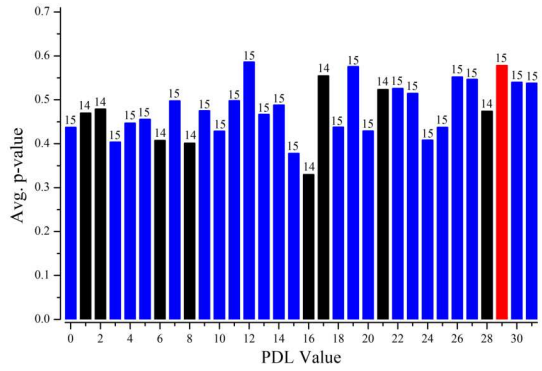

Figure 4. NIST SP800-22 results for Artix-7 7A100T device

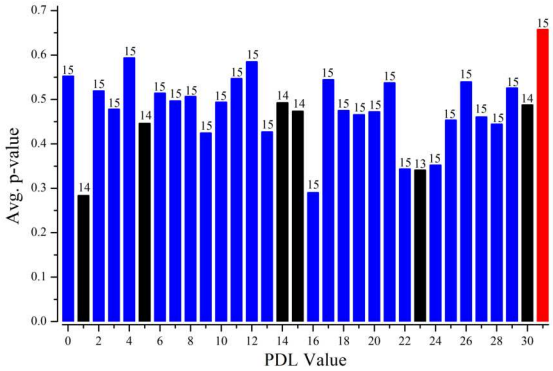

Figure 5. NIST SP800-22 results for Kintex Ultrascale KU040 device

Therefore, we aim to analyze the performance of TRNGs according to different PDL inputs and evaluate the randomness of TRNG outputs across various PDL inputs. This will establish the significance and enhance the validity of the PDL-based TRNG structure.

4. Experimental Results

As shown in Figure 3, to measure the performance changes of the TRNG based on PDL inputs, the PDL-based TRNG from Figure 2 was implemented on Xilinx Artix-7 7A100T device and Xilinx Kintex Ultrascale KU040 device using Xilinx's Vivado 2021.1 EDA tool. The sampling clocks were set at 50MHz and 200MHz, respectively. For evaluating the performance of the TRNG, the NIST SP 800-22 test suite was conducted. Each test generated a 1,000,000-bit stream for 100 rounds, thus producing a total of 1,000,000,000 bits per PDL setting. The PDL settings were changed after generating 1,000,000,000 bits.

Figures 4 and 5 illustrate the performance of the TRNG on the Xilinx Artix-7 7A100T and Xilinx Kintex Ultrascale KU040 devices according to different PDL inputs. The horizontal axis shows the possible values of PDL inputs from 0 to 31, while the vertical axis displays the average p-value for the 15 categories of the NIST SP 800-22 test, where a higher average p-value indicates better performance. Additionally, the numbers on each graph bar represent the number of categories out of the 15 in the NIST SP 800-22 test that were passed. As Figures 4 and 5 demonstrate, the performance of the PDL-based TRNG varies with different PDL inputs. For instance, even using the same PDL-based TRNG structure, the Xilinx Artix-7 7A100T device does not achieve full pass at PDL inputs 1, 2, 6, 8, 16, 17, 21 and 28, and similarly, the Xilinx Kintex Ultrascale KU040 device fails at PDL inputs 1, 5, 14, 15, 23 and 30. Table 1 and Table 2 present the results of

Table 1. Test results of NIST test for Artix-7 7A100T device(PDL = 5'b11101)

	p-value	Proportion
Frequency	0.91	1.00
Block Frequency	0.96	1.00
Cumulative Sums	0.22	0.99
Runs	0.60	0.99
Longest Run	0.87	0.97
Rank	0.95	0.99
FFT	0.07	0.99
Non-Overlapping	0.49	0.99
Overlapping	0.37	0.99
Universal	0.28	1.00
Approximate Entropy	0.82	1.00
Random-Excursions	0.55	0.98
Random-Variant	0.56	0.99
Serial	0.67	0.99
Linear Complexity	0.38	0.98
Average p-value	0.58	-

Table 2. Test results of NIST test for Kintex Ultrascale KU040 device(PDL = 5'b11111)

	p-value	Proportion
Frequency	0.88	0.99
Block Frequency	0.66	1.00
Cumulative Sums	0.48	1.00
Runs	0.99	1.00
Longest Run	0.72	0.98
Rank	0.68	1.00
FFT	0.85	0.98
Non-Overlapping	0.51	0.99
Overlapping	0.37	1.00
Universal	0.46	0.99
Approximate Entropy	0.87	1.00
Random-Excursions	0.62	0.99
Random-Variant	0.47	0.99
Serial	0.62	1.00
Linear Complexity	0.70	0.97
Average p-value	0.66	-

NIST tests for various test categories conducted at the highest average p-value among all PDLs. For each of the 15 categories, a p-value below 0.01 indicates failure to pass the test. The closer the p-value is to 1, the more random the data is considered. For the Artix-7 7A100T device with PDL = 5'b11101 exhibited an average p-value of 0.58, and for the Kintex Ultrascale KU040 device with PDL = 5'b111111, it measured 0.66. The experimental results suggest that for the 7A100T device, selecting PDL = 5'b11101, and for the KU040 device, selecting PDL = 5'b11111 could optimize TRNG performance. Consequently, selecting the optimal PDL when implementing a PDL-based TRNG is critically important.

5. Conclusion

In conclusion, this study has demonstrated the importance and impact of Programmable Delay Logic (PDL) on the performance and security of True Random Number Generators (TRNGs) within FPGA environments. By leveraging the capabilities of PDL to fine-tune the behavior of Ring Oscillators (ROs), we have been able to significantly enhance the entropy and randomness quality essential for cryptographic applications. The experimental results from implementing the PDL-based TRNG on Xilinx Artix-7 and Kintex Ultrascale devices underline the variability in performance based on different PDL settings, revealing that certain configurations can optimize TRNG performance more effectively than others.

Furthermore, our findings indicate that selecting appropriate PDL inputs is crucial for achieving the desired level of security and efficiency. The NIST SP 800-22 test results not only validated the robustness of the TRNG outputs but also highlighted the critical role of PDL settings in maximizing the effectiveness of security mechanisms in digital systems. Moving forward, it will be essential for future research to focus on developing methodologies for optimizing PDL configurations to consistently achieve higher performance across various devices. This will ensure that FPGA-based security systems can reliably meet the increasing demands of modern digital applications.

Acknowledgement

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (No. 2021R1I1A3055806), the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2022R1A5A8026986), Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (2022-0-01170), and the EDA tool was supported by the IC Design Education Center(IDEC), Korea.

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