International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

# Dual Arbiter PUF with Shift Register Based TRNG on Basys-3 FPGA Board and its Performance Analysis on Uniqueness, Reliability and Randomness

[1]Lukram Dhanachandra Singh, [2]Preetisudha Meher, [3]Amit Kumar Panda

**Submitted**: 25/06/2023 **Revised**: 07/08/2023 **Accepted**: 28/08/2023

**Abstract:** Hardware security has become a crucial issue as a result of the semiconductor industry's explosive growth and globalization. Various attacking and tampering methods are continuously devised to intrude the current methods of hardware security and protection. Physical Unclonable Functions (PUFs) is a well-known solution for these threats. Hence, many researchers are in deep search for advancing the sequence generator based on different Physical Unclonable Functions (PUFs), trying to improve their reliability, uniqueness & randomness. In this paper, Shift Register based Dual arbiter PUF (SR-APUF) is designed and implemented on Basys-3 Field Programmable Gate Array (FPGA) board and the drawbacks of existing PUF designs in certain areas are improvised finding the reason behind it. In addition to the architecture suggested in this paper, a complication method is also used to increase security. In this proposed SR-APUF, shift register will be controlled by a random number generated by delay lines of the PUF and one-time programmable memory device, which cannot be reconfigure later by attackers and so enhance the security. The responses generated are analyzed by implementing the proposed design of SR-APUF and the experimental results demonstrate a better uniqueness of 47.3%, compare to other FPGA-based APUF. Additionally, a reliability rate of 95.7% is attained. And its randomness is also good as it passes the NIST test.

***Keywords:*** *Physically Unclonable Function, FPGA, Hardware security, Arbiter PUF.*

## 1. Introduction

Nowadays, Internet of things (commonly known as IoTs) are dominating on almost all our daily applications i.e., a lot of the items we use every day are connected to the Internet and run by computers. And therefore, communications between the computers or machines over the internet is increasingly common procedures. For these procedures, a high level of authentication security is absolutely necessary. While, there are many security challenges such as chip authentication, IC Piracy and IC overbuilding, Side Channel Attacks, etc.[1,2]. PUFs [3,4,5] which are physical functions that produce a distinct response to a challenge-input, have been suggested as a solution for most of these challenges [2]. PUF-based authentication can safeguard the chip available for authentication on a device and the intellectual property core of FPGAs [6]. PUF can be used as cryptographic primitives [7]. PUFs are relatively easy to construct as circuits and provide for lightweight authentication. PUFs can be built on FPGA in addition to an application-specific integrated circuit (ASIC) [8]. FPGAs are used in a variety of goods [9,11] because its programmable synthesizing allows for both customization and security. It is somewhat cheaper to build a system with an FPGA than an ASIC for low-volume manufacturing.

There are many types of PUF and they can be classified in various ways, in [10] the author divided into two classes as memory-based and delay-based PUFs. Delay-based PUFs, which employ delay-time information from data transmission in the circuit, include arbiter PUFs. An Arbiter PUF's fundamental idea was introduced in 2002 [12]. PUFs are classified into strong and weak PUFs as well. Weak PUFs have hardly few numbers of challenges while strong PUFs contain large number of CRPs i.e., they are suitable for device identification and authentication due to their ability to apply a variety of challenges. In strong PUF, it is too complex for the attacker to predict the relativity of challenges and responses. Strong PUFs, especially arbiter-PUFs (among the most notable of these PUFs) and its variations, have been discovered to be sensitive to Machine Learning based modelling attacks in the past few years [14,15,16]. Even blockchain security solution can also enhance with FPGA as mentioned earlier [13].

In this paper, we address a variety of PUFs, particularly the FPGA-based Arbiter PUF, and to overcome the limitations mentioned earlier, a novel shift register based arbiter PUF model (SR-APUF) design is proposed and determine the robustness of our new design using the data acquired from our new design's FPGA implementation. Further precisely, the contributions of our research work are as follows:

[1,2] *Electronics & Communication Engineering Department, National Institute of Technology Arunachal Pradesh, India*
[3]*Department of Electrical and Electronics Engineering, Birla Institute of Technology and Science Pilani Hyderabad Campus, India*
** Corresponding author's Email: dhana.lukram0@gmail.com*

- We propose a novel Shift register based dual Arbiter PUF architecture that has two layer of delay line constructed using multiplexers and multiple flip flops along with a shift register.
- SR-APUF cell is implemented on a Xilinx Basys 3 (Artix-7 based) FPGA board as a hard-macro to confirm well-adjusted routing.
- The efficiency of the suggested PUF design's FPGA implementation and its low hardware resource consumption are both demonstrated. The suggested SR-APUF architecture only needs fourty four slices of a Xilinx Basys 3 (Artix-7 based) FPGA to provide a 1-bit response. It uses fewer hardware resources than the preceding APUF architecture.
- Challenge obfuscation is also used for enhancing the security.
- Using the data retrieved from the FPGA implementation, we evaluate the proposed SR-APUF.
- According to the authors' knowledge, the experimental results demonstrate a better uniqueness of 47.3%, which is an FPGA-based APUF's finest outcome till date. Additionally, a reliability rate of 95.7% is attained. And its randomness is also good as it passes the NIST test.

The rest of this paper is prepared as: Section 2 presents associated work: structures of various conventional PUFs design, their vulnerability and countermeasures. In section 3, we introduce our SR-APUF as an alternate countermeasures and FPGA implementation. Experimental results and performance analysis are deliberated in section 4. Lastly, section 5 provides the conclusion.

## 2. Related Work

Despite the fact that various PUFs may work as random number generators, this paper primarily considers the arbiter PUF and its modifications as examples. So, here, we'll go through the fundamentals of the arbiter PUF. Fig. 1 illustrates the design of a conventional arbiter, which consists of N stages of pairs of multiplexers also known as delay lines with an arbiter. It consists of two MUX arrays, with two MUXs at each stage of the array linking to the two MUXs in the next stage. In accordance with the value of the selection bit to the MUX in each stage known as the challenge bit, two signals simultaneously start from the two MUXs in the first stage. Its genetic basis is a race between the delays between the two signals. The two signals travel via a variety of routes based on the challenges. These routes are chosen by the selector pairs based on an n-bit challenge which is provided as the selection input. Depending on these challenges and delay of signals, it will determine whether logic 0 or 1 will be latched as response. Here logic-1 will be latched if the step signal reached the input port of D flip-flop first or logic-0 will be latched if it reaches input port of the clock.

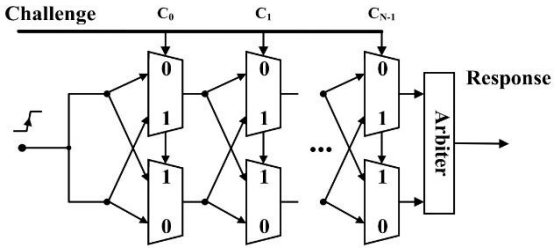


**Fig 1**. Conventional Arbiter PUF

To reduce the predictability of the results, the N-XOR Arbiter PUF was suggested, which utilizes the N Arbiter PUFs constructed on the identical chip's XOR N responses.

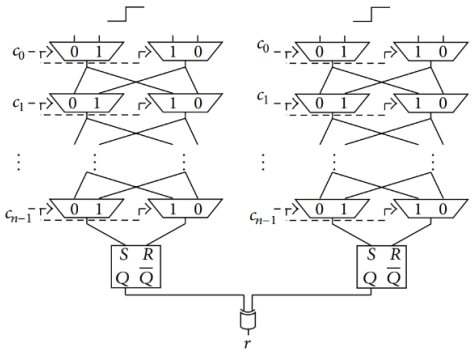

**Fig 2**. 2-1 Dual XOR Arbiter PUF

And 2-1 Double Arbiter PUF [17] was suggested, whose response was generated by XORing the responses generated by two Double Arbiter PUF to broaden the range of symmetric routes. Although this PUF reduces the accurateness of support vector machine's (SVM) prediction by 36.3% and its reliability also declines by 0.06. Fig. 2 shows the design of 2-1 Dual XOR Arbiter PUF.

As an alternative, the transfer function block's linearity can be destroyed by employing nonlinear tables to make cascaded blocks nonlinear [18]. Feed forward loops are another method that can boost the A-PUF circuit's nonlinearity [19]. All of the changes to this approach have shown a rise of randomness from 10% to 20% and a reduction in uniqueness and dependability of 5% to 10%. The resistance to machine learning assaults is thus effectively increased by enhancing the randomness by partly breaking up the linear addition at a reduced hardware cost.

Additionally, a 3-1 Double Arbiter PUF was suggested in [20] to increase unpredictability even on FPGAs with wiring issues and it claims that the 3-1 Double Arbiter PUF performs well in terms of tolerance, uniqueness, and stability and has a substantially better lenience to machine-learning attacks than the traditional N-XOR Arbiter PUF. It is again possible to anticipate that XORing replies from several PUFs on the same chip will raise the difference in interchip across chips that have stayed assessed for uniqueness [21].

A controlled PUF design was also proposed in [22] to protect the PUF from ML Attacks by decreasing challenge-response mapping leakage. This method's main goal is to use a hash function to randomize the challenge and/or response values. It can increase the resistance on modelling attack of Arbiter PUF by using challenge obfuscation. In [23], a challenge processing algorithm based on patterns is suggested. Applying a portion of the challenge to several A-PUF instances is the core idea, and selecting a response using a random number is the secondary notion. This approach appeals because it maintains the reliability of the A-PUF.

In reality, the majority of strategies for improving reliability and security effectively explore the reliability-security cost - benefit space of PUFs. The PUF reliability will suffer if we add additional randomness to strengthen the security against modelling attacks. It will therefore make the PUF useless. However, increasing PUF reliability would have a negative impact on security. The modelling approach can theoretically attain a modelling accuracy of about 100% with a surplus of CRPs with minimal noise from the PUFs with improved reliability. Finding a new level of granularity that offers better reliability and strong resilience to modelling assaults is therefore essential. Therefore, better reliability for a small set of CRPs which will be utilized for authentication and strong security with an overall uncertain PUF are both feasible.

## 3. Design and implementation of PUFs

Various PUF designs have been introduced for facing the modelling attacks. Out of which Arbiter PUF as one of the strong PUF is being used commonly. But as mentioned earlier, when we try to improve randomness, we need to compromise reliability, so here we proposed to add challenge obfuscation using a shift register which is controlled by selection bits generated by stages of mux pairs and A1 which will be generated by OTP (one time programmable) devices [24,25,26]. The security upgrade of OTP technology is also studied in [27].

### 3.1 Proposed Design of SR-APUF & its methodology

In this Shift register based Dual Arbiter PUF, we proposed to add a shift register for challenge obfuscation which will be triggered or control by a random number which is generated in delay line of the designed PUF itself. This random binary sequence and A1 will control the selection line of the shift register, that either it will be left shift or right shift and for how many bits it will be shifted. Thus, this shifting itself will be random and therefore providing more unpredictable challenges. So, it may increase the randomness and uniqueness of the sequence generated by proposed design of shift register based dual arbiter PUF.

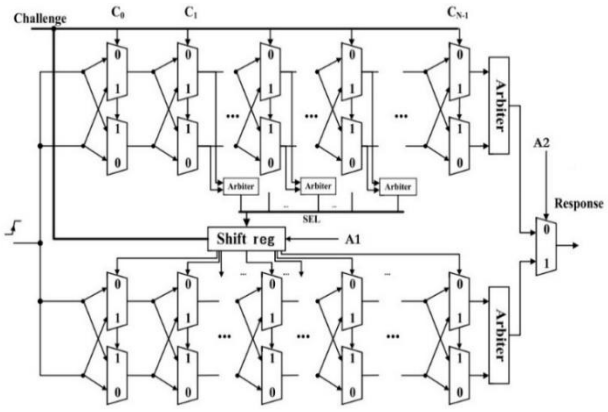


**Fig 3.** Proposed Model SR-APUF

The sequence generated by dual arbiter PUF is fed to control logic and it will check for even or odd parity and accordingly we programmed the control logic to generate a triggering bit for shift register. Thus, if the generated binary sequence has odd parity, then it will be shifted left and if it is even parity then it will be shifted right. And the control logic will also compare the numbers of 1s and 0s in generated sequence and the difference will be the bit by which the shift will be done. Thus, the attacker will counter difficulties as the shifting direction and bits both will be random and hard to predict, providing increase in randomness.

### 3.2 FPGA Implementation

As mentioned before, most of Arbiter PUF designs are insignificant to FPGA implementation because of the difficulties when accomplishment of routing, which may greatly affect the outcomes. In this part, we focus on implementation of PUFs in FPGA. It was also suggested to implement a 64-stage earlier APUF design on the Nexys4 board [20].

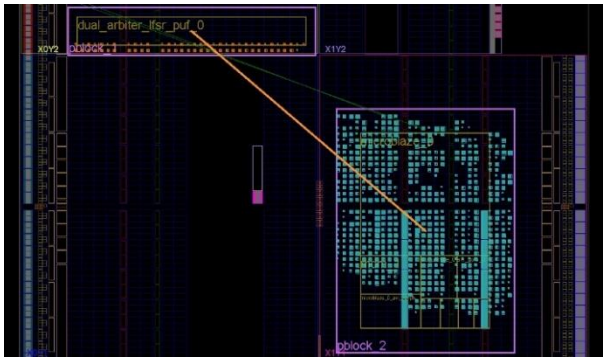


**Fig 4.** The floorplan of proposed SR-APUF design

The Artix-7 XC7A35T has 33,280 logic cells in 5200 slices. Fig. 4 illustrates the planned 64-bit SR-APUF design floor plan as well as the optimized routing map for a single slice. Each slice's optimized routing makes sure that the response is strong. The suggested SR-APUF design only needs 44 slices to produce an 1-bit response; 42 slices are occupied to construct 4 F/F and 3 MUX on one slice, and the final 2 slices for the additional cross-coupled NAND gates. Look up tables (LUTs) or specific internal slice MUXes can be used to implement both MUX and cross-coupled NAND gates. The LUT technique is used for the following experimental results. The Basys 3 FPGA features 4 flip-flops per slice, allowing for the placement of the 4 FFs of the proposed SR-APUF cell in a sole slice. The 44 slices are employed as a hard macro for the construction of each 1-bit response; therefore, to obtain a 64-bit response, 64 hard macros and 64 by 44 slices are utilized. The suggested SR-APUF design and the previously reported APUF design from [17] are implemented on the same technology, and Table 1 gives the comparison of the FPGA hardware resource usage of these designs. The suggested SF-APUF architecture uses less FPGA resources than the prior APUFs for equal bit security.

## 4. Experimental Results & Performance Analysis

The experimental set up for hardware implementation of Basys 3 FPGA Board is as shown in Fig. 5. The integrated logic analyzer added while designing the PUF provide the waveform of the challenge response pair generated as shown in the figure below.

**Fig 5**. Hardware implementation

**Table 1.** Comparison of Resource Consumption of FPGA hardware

| | Previous APUF [15], [16] | FF-APUF [17] | SR-APUF |
|---|---|---|---|
| Strong/Weak PUF | Strong | | |
| Response | 64 bit | | |
| FPGA | Artix -7 | | Basys3 (Artix 7) |
| Type of slice | SLICE L | | |
| Usage (in slices) | 129 X 64 | 44 X 64 | 44 X 64 |
| Usage (in %age) | 52% | 17.77% | 17.77% |

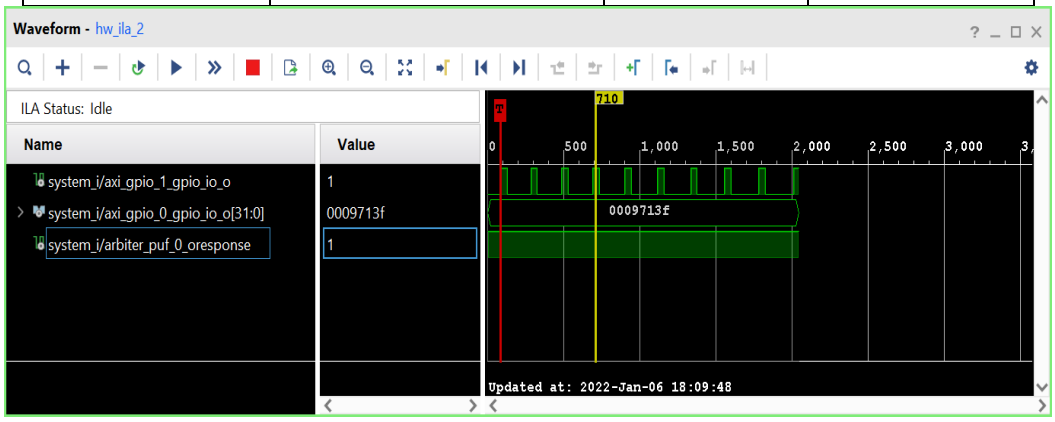


**Fig. 6** Waveform obtained from ILA while running or debugging the application code developed in Xilinx SDK

After designing this dual arbiter PUF with shift register, we implemented the proposed 64-bit SR-APUF design on six hardware platform of Basys-3 FPGA board. Then we create an application using Xilinx SDK to run and generate the challenge response pair and from which we save the generated random binary sequences. We need to collect 100 different $10^6$ binary sequences for further randomness analysis known as NIST. We collect six 128-bit (n = 128) responses from six (k = 6) devices to compute the hamming distances. We used hair dryer for heating the board so that we can measure the outcomes in changed temperature to check the reliability of the proposed SR APUF.

### 4.1 Uniqueness

In this research, the hamming distance (HD) is utilized to assess the uniqueness of our SR-APUF. It is a measure of variation in inter-chip by analysing how readily a specific PUF circuit designs may differ in k distinct devices. When two devices are given the same challenge, a PUF circuit

should ideally yield an typical inter-chip hamming distance (HD) close to 50%, meaning that even though the same challenge has been used, the two devices differs with half of the response bits. In light of this, a typical inter-chip HD-based % figure of merit for uniqueness may be established. Suppose there are a pair of chips, x and y, implementing the identical PUF circuit and when given with the same challenge, n-bit responses Rx and Ry, respectively, then the uniqueness which is used to define the typical inter-chip HD between k devices can be expressed as:

$$Uniqueness = \frac{2}{k(k-1)} \sum_{x=1}^{k-1} \sum_{y=x+1}^{k} \frac{HD(R_x, R_y)}{N} \times 100 \quad (1)$$

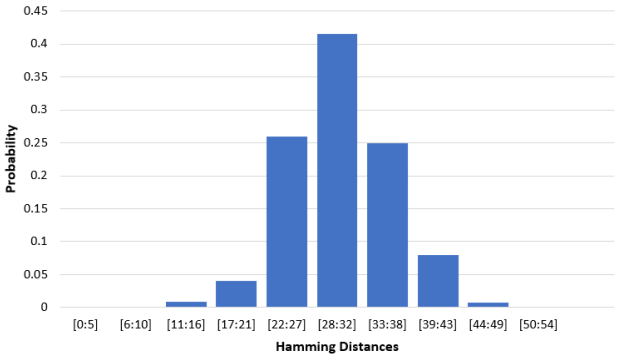


**Fig 7.** Distribution of the Uniqueness for SR-APUF

The experiment to obtain uniqueness is run under typical settings, such as standard room temperature and Basys3's (Artix-7 FPGA) core supply voltage of l.0 v. Six 128-bit (n = 128) responses from six (k = 6) devices are used to calculate the values of hamming distances. The suggested SR-APUF design's uniqueness result is shown in Fig. 7, and it is roughly 47.3% with an 8% variance. The suggested SR-APUF design exhibits a substantial increase in its capacity to discriminate between distinct devices when compared with the outcome of 9.42 percent for the APUF in [17].

### 4.2 Reliability

The steadiness of PUF signatures produced by the identical challenge in several tests is assessed using reliability. PUF signatures ought to be consistent across different observations and the same challenges.

Actually, a number of environmental factors, including temperature, voltage, and device ageing, can bring in variations in delay of the circuit and alter PUF signatures. Any 2 signatures produced with the identical challenge in subsequent trials by a PUF with good reliability must only differ slightly from one another. Calculating the typical intra-chip HD of n samples of responses, $R'_x$, obtained at various operating settings in comparison to n-bit reference response, Rx, recorded at normal operational circumstances establishes reliability for a device x as an individual value.

The average intra-chip HD is assessed as follows:

$$HD_{INTRA} = \frac{1}{n} \sum_{t=1}^{n} \frac{HD(R_x, R'_{x,t})}{N} \times 100 \quad (2)$$

where $R'_{x,t}$ is the t[th] sample of $R'_x$. The %age figure of merit for reliability is characterised as:

$$Reliability = 100 - HD_{INTRA} \quad (3)$$

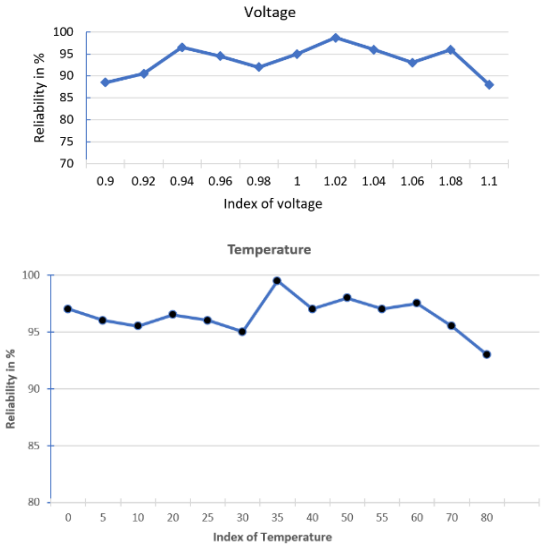


**Fig 8.** Reliability evaluation

Naturally, 100 percent is the optimal reliability value. The suggested SR-APUF design's dependability findings are shown in Fig.  as 95.7% across a temperature range of 0°C to 75°C and 94.10% with a 10% change in supply voltage. As a consequence, the suggested SR-APUF design shows better reliability results.

### 4.3 Randomness

Randomness is one of the most important properties to check the number or sequence generated by any generator or PUF design if the generated number or sequence is random or not. That means either the generated number can be easily evaluate by attacker. The National Institute of Standard & Technology's NIST test is a statistical test set for testing random and pseudorandom number generators for use in cryptography applications. So, this test is taken for analysis the designed generator. NIST test involve 15 different statistical test to check the randomness of the sequence produced.

Table 2 demonstrates the comparative analysis of effectiveness of the suggested SR-APUF design along with that of earlier existing PUF designs. All PUF designs' reliability evaluations are conducted in trials with variable voltage and/or temperature. The proposed SR-APUF design produces highly reliable results using FPGA, which is noteworthy given that the suggested SR-APUF architecture provides reliability results that are comparable to those of the preceding APUF design.

Comparing the findings of the suggested SR-APUF architecture to those of the earlier APUF designs in both FPGA and ASIC, the proposed design likewise displays a improved uniqueness in the response. The suggested SR-APUF uses the least amount of FPGA slices compared to existing APUFs and also achieves efficient area utilization in terms of the use of hardware resources. Complexity gains need a trade-off between area utilization and complexity.

**Table 2.** Comparison on Uniqueness, Reliability & resource consumption of various PUFs

| PUF Architecture | Type | U | R | Hardware | Response bit size | Consumption of Resources |
|---|---|---|---|---|---|---|
| **Latch PUF** | Weak | 46% | >87% | Spartan 3 | 128 | 2 x 128 slices |
| **BR PUF** | Weak | 14.30% | 99.2% | SPICE simulation | 64 | 64 x $COMB^2$ |
| **Flip-flop PUF** | Weak | ~ 50% | >90% | Vertex 2 | 4096 | 4096 F/Fs |
| **SRAM PUF** | Weak | 49.97% | >88% | FPGA | 128 | 4600 SRAM mem bits |
| **RO PUF** | Weak | 46.15% | 99.5% | Virtex 4 | 128 | 16 x 64 $array^2$ |
| **Arbiter PUF** | Strong | 23% | 96.3% | TSMC 110nm | 64 | 1212nm x 1212nm |
| **CRO PUF** | Weak | 43.5% | >96% | Spartan 3 | 127 | 64 slices for Ros |
| **Arbiter PUF** | Strong | ~ 50% | 98.1% | 45nm | 64 | 36nm x 50nm |
| **Butterfly PUF** | Weak | ~ 50% | 94% | Virtex 5 | 64 | 130 slices |
| **Arbiter PUF** | Strong | 9.42% | - | Artix7, Spartan6 | - | 129 x 64 slices |
| **FF- APUF** | Strong | 40% | 97.1% | Artix 7 | 64 | 44 x 64 slices |
| **Proposed SR-APUF** | Strong | 47.3% | 95.7% | Basys3-Artix7 | 64 | 42 x 64 slices |

The NIST test results of our proposed design is as given below in table 3.

**Table 3.** NIST test result of SR-APUF Design

| Statistical Tests | Pass/ Fail | Statistical Tests | Pass/ Fail |
|---|---|---|---|
| Frequency | Pass | Non-overlapping Templates | Pass |
| Block Frequency (m = 64) | Pass | Random Excursions (x = +1) | Pass |
| Cumulative sum-Forward | Pass | Approximate Entropy | Pass |
| Cumulative sum-Reverse | Pass | Overlapping Templates | Pass |
| Runs | Pass | Universal | Pass |
| Long Runs of Ones | Pass | Random Excursions Variant (x = -1) | Pass |
| Rank | Pass | Serial | Pass |
| DFT | Pass | Linear Complexity (M = 100) | Pass |

## 5 Conclusion

Thus, the proposed methodology of adding a shift register controlled by a random number from the stages of mux pairs known as delay lines and OTP devices for challenge obfuscation will enhance the overall performance of the PUF. With FPGA technology reducing the hardware complexity and enhance the hardware security level, the sequence generated by the proposed SR-APUF design is analysed. And it was observed that the experimental results demonstrate a good uniqueness of 47.3%, which is an FPGA-based APUF's finest outcome to date. Additionally, a reliability rate of 95.7% is attained. And its randomness is also good as it passes the NIST test. This can be further implied with IOTs or blockchain technology for various applications. If Built In Self NIST is provided, then the requirement of multiple platform & time consumption will be reduced.

**Conflicts of Interest**

The authors declare no conflict of interest.

**Author Contributions**

Conceptualization, Lukram Dhanachandra Singh; methodology, Lukram Dhanachandra Singh; hardware, NIT Manipur; validation, Lukram Dhanachandra Singh, Amit Kumar Panda, and Preetisudha Meher; formal analysis, Lukram Dhanachandra Singh, Amit Kumar Panda, and Preetisudha Meher; investigation, Lukram Dhanachandra Singh; writing—original draft preparation, Lukram Dhanachandra Singh; writing—review and editing, Lukram Dhanachandra Singh; supervision, Preetisudha Meher and Amit Kumar Panda.

**Acknowledgments**

This work was supported by the National Institute of Technology Arunachal Pradesh and National Institute of Technology Manipur.

## References


[1] Mark Tehranipoor, Nitin Pundir, Nidish Vashistha, Farimah Farahmandi, "Hardware Security Primitives Based on Emerging Technologies", *Hardware Security Primitives*, pp.145, 2023.

[2] Kin Fun Li, Narges Attarmoghaddam, "Challenges and Methodologies of Hardware Security", *2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA)*, pp.928-933, 2018.

[3] I. Papakonstantinou and N. Sklavos, "Physical unclonable functions (pufs) design technologies: Advantages and trade offs" in Computer and Network Security Essentials, Springer, pp. 427-442, 2018.

[4] R. S. Pappu, Physical one-way functions [Ph.D. thesis], Massachusetts Institute of Technology, Cambridge, Mass, USA, 2001.

[5] R. Pappu, B. Recht, J. Taylor, and N. Gershenfeld, "Physical one-way functions," Science, vol. 297, no. 5589, pp. 2026–2030, 2002.

[6] J. Guajardo, S. S. Kumar, G. J. Schrijen, and P. Tuyls, "FPGA intrinsic PUFs and their use for IP protection," in Proceedings of the 9th International Workshop on Cryptographic Hardware and Embedded Systems (CHES '07), pp. 63–80, Vienna, Austria, 2007.

[7] Vikash Kumar Rai, Somanath Tripathy, Jimson Mathew, "Design and Analysis of Reconfigurable Cryptographic Primitives: TRNG and PUF", *Journal of Hardware and Systems Security*, vol.5, no.3-4, pp.247, 2021.

[8] Y. Lu et al., "FPGA-based RO PUF with low overhead and high stability", *Electron. Lett.*, vol. 55, no. 9, pp. 510-513, May 2019.

[9] J.-L. Zhang et al., "Techniques for design and implementation of an FPGA-specific physical unclonable function", *J. Comput. Sci. Technol.*, vol. 31, no. 1, pp. 124-136, 2016.

[10] T. McGrath, I. B. Bagci, Z. M. Wang, U. Roedig and R. J. Young, "A PUF taxonomy", *Appl. Phys. Rev.*, vol. 6, Feb. 2019.

[11] L.D. Singh, Meher, P., A FPGA-Based PUF Integrated Blockchain to Overcome the Challenges of Internet of Everything (IoE), *Lecture Notes in Electrical Engineering*, 2020, 686, pp. 49–61

[12] B. Gassend et al, "Silicon physical random functions," in Proceedings of the 9th ACM Conference on Computer and Communications Security, pp. 148– 160, Washington, DC, USA, Nov 2002.

[13] L. D. Singh, P. Meher, Advancement of Blockchain Security Solution with FPGA, International Journal of Advanced Science and Technology, Elsevier, Vol. 28, No. 3, 276-283, 2019.

[14] U. Ruhrmair et al, "PUF modeling attacks: An introduction and overview," in DATE, 2014.

[15] Wei Liu, et al., "Multiclass Classification-Based Side-Channel Hybrid Attacks on Strong PUFs", *IEEE Transactions on Information Forensics and Security*, vol.17, pp.924-937, 2022.

[16] Sean Donnelly, Liam Meany, "A Taxonomy of Machine Learning Methodologies Used Against Physical Unclonable Functions", *2021 IEEE 4th 5G World Forum (5GWF)*, pp.206-211, 2021.

[17] T. Machida, D. Yamamoto, M. Iwamoto, and K. Sakiyama, "A new mode of operation for arbiter PUF to improve uniqueness on FPGA," in Proceedings of the Federated Conference on Computer Science and Information Systems (FedCSIS '14), pp. 871–878, Warsaw, Poland, September 2014.

[18] A. Vijayakumar, V. C. Patil, C. B. Prado, and S. Kundu, "Machine learning resistant strong PUF: Possible or a pipe dream?" in Proc. Hardw. Orient. Secur. and Trust (HOST'16), McLean, USA, May 2016, pp. 19–24

[19] Y. Lao and K. K. Parhi, "Statistical analysis of MUX-based physical unclonable functions," IEEE Trans. on Comp.-Aided Des. of Int. Circ. and Syst., vol. 33, no. 5, pp. 649–662, 2014.

[20] Takanori Machida, Dai Yamamoto, Mitsugu Iwamoto, and Kazuo Sakiyama, "A New Arbiter PUF for Enhancing Unpredictability on FPGA", The Scientific World Journal, Volume 2015.

[21] Jing Wen, Minghui Huang, Ziheng Chen, Lijia Zhu, Shuai Chen, Bing Li, "A Multi-line Arbiter PUF with Improved Reliability and Uniqueness", *2019 IEEE 4th International Conference on Signal and Image Processing (ICSIP)*, pp.641-648, 2019.

[22] Y. Gao, H. Ma, S. F. Al-Sarawi, D. Abbott and D. C. Ranasinghe, "PUF-FSM: A controlled strong PUF", *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 37, no. 5, pp. 1104-1108, May 2018.

[23] Bo Chen, Pengjun Wang, Gang Li, "An Obfuscated Challenge Design for APUF to Resist Machine Learning Attacks", *2019 IEEE 13th International Conference on ASIC (ASICON)*, pp.1-4, 2019.

[24] E. Terzioglu, G. I. Winograd, and M. C. Afghahi, “One-time programmable memory,” U.S. Patent 7 508 694, Mar. 24, 2009.

[25] A. Lele, S. Sadana, A. Singh, H. S. Jatana and U. Ganguly, " A simple PECVD SiO 2 OTP memory based PUF for 180 nm node for IoT ", *Proc. IEEE DRC*, pp. 1-2, Jun. 2017.

[26] S. Sadana, A. Lele, S. Tsundus, P. Kumbhare and U. Ganguly, "A highly reliable and unbiased PUF based on differential OTP memory", *IEEE Electron Device Lett.*, vol. 39, no. 8, pp. 1159-1162, Aug. 2018.

[27] Shao-Yu Shaun Chou, Shawn Chen, Jun-Hao Chang, Wan-Hsueh Cheng, Yu-Der Chih, Philex Fan, Chia-En Huang, Cher-Ming Hung, Gu-Huan Li, Yih Wang, Shao-Ding Wu, Tsung-Yung Jonathan Chang, "A 16-kb Antifuse One-Time-Programmable Memory in 5-nm High-K Metal-Gate FinFET CMOS Featuring Bootstrap High-Voltage Scheme, Read Endpoint Detection, and Pseudodifferential Sensing", *IEEE Solid-State Circuits Letters*, vol.4, pp.170-173, 2021.

[28] Krishna, N. ., R., A. ., John, N. M. ., & Kurian, S. M. . (2023). Training and Classification of PCA with LRM model for Diabetes Prediction . International Journal on Recent and Innovation Trends in Computing and Communication, 11(4s), 08–15. https://doi.org/10.17762/ijritcc.v11i4s.6302

[29] Li Wei, Machine Learning in Fraudulent E-commerce Review Detection , Machine Learning Applications Conference Proceedings, Vol 2 2022.

[30] Dhabliya, D., & Dhabliya, R. (2019). Key characteristics and components of cloud computing. International Journal of Control and Automation, 12(6 Special Issue), 12-18. Retrieved from www.scopus.com