Implementation of Stable RO PUF on FPGA

Maitri Patel
Electronics and Communication Engineering Departmnent
Institute of Technology, Nirma University
Ahmedabad, India
20bec085@nirmauni.ac.in

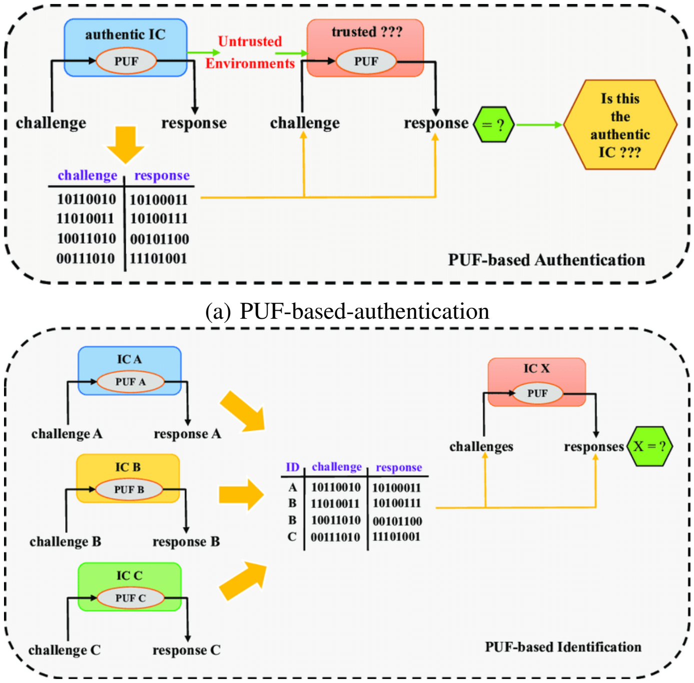
***Abstract*—A Physical Unclonable function (PUF) serves as a physical security measure that enables the extraction of intrinsic digital identifiers from electronic devices. It shows great promise in enhancing security for lightweight devices designed for IoT applications, mainly due to its cost-effectiveness. One of the most well-known PUFs that may generate the volatile key by comparing the frequencies of ROs is the ring oscillator (RO) PUF. RO-PUF have the main drawback of being susceptible to environmental changes making them unstable. Reliability is reduced in RO pairs with small frequency differences because bit flips are more frequent in these pairs. Studies on boosting reliability have been carried out over time. The uniqueness has decreased and the amount of space used has increased, among other problems. To balance reliability with uniqueness, this research suggests a stable RO PUF on FPGA that uses less space than a typical design. The results (uniqueness=49.16%,reliablity=98.7%,uniformity=49.37%) show better performance compared to previously suggested methods for a similar platform (Altera) and the reliability is as good as the latest method. All of the presented measurements were performed on Altera DE2 Cyclone II.**

***Keywords*—FPGA, Physical Unclonable function, Reliability, Stable RO PUF, Uniqueness.**

I. Introduction

Device authentication poses a significant challenge in cryptography as it requires each device to store a unique secret key in non-volatile memory on the chip. However, this approach becomes difficult, expensive, and vulnerable to hacking attacks in devices such as RFIDs and smart cards. To address this issue, it is necessary to employ secure FPGA-based designs and applications to prevent tampering attempts. Alternatively, tamper-proof memory can be used, but it is costlier, more complex, and consumes more energy. Ensuring the security of cryptographic keys on computing systems is crucial for maintaining overall security, privacy protection, and reliable computing. To achieve this, various advanced countermeasures and certification procedures have been developed and implemented[1].

Physical unclonable function (PUF) has recently been used to describe key creation and storage. PUF delivers a greater level of protection without the need of persistent power by keeping secrets in its special intrinsic physical properties that are arbitrarily defined by fabrication variances, such as the subtle difference in the delays of two wires with equal lengths at the design phase [2]. It has the ability to extract unique signatures, from the intrinsic process variability of silicon devices, using a challenge and response mechanism.PUF has applications in several security-related fields, including IC authentication(Fig.1a), IC identification(Fig.1b), hardware/software intellectual property protection, device remote activation, and trusted computing[3,4,5]. The two major groups of PUFs that are suitable for FPGAs according to their sources of randomness are delay-based and memory-based PUFs. A memory-based PUF makes use of the variations in the relative threshold voltages of the transistors. Uncontrollable random initial SRAM cell value can be generated during power-up [7]. Similar to SRAM PUFs, Maes et al. [8] suggested Flip-Flop PUF in 2008 to extract the initial values of regular flip-flops (FFs) on FPGAs. Just like SRAM cells, uninitialized flip-flops have an uncertain state that is influenced by manufacturing variations and can be utilized as a PUF response. One another example of a memory-based PUF is the Butterfly PUF, which was introduced by Kumar et al. [9] in 2008. It consists of a pair of cross-coupled latches which tries to mimic the startup behavior of cross-coupled inverters in an SRAM cell. Su et al. [10] proposed the Latch PUF,



challenge	response
10110010	10100011
11010011	10100111
10011010	00101100
00111010	11101001

(a) PUF-based-authentication

ID	challenge	response
A	10110010	10100011
B	11010011	10100111
B	10011010	00101100
C	00111010	11101001

(b) PUF-based-identification

Fig. 1: Application of PUF [6]

which is quite similar to the Butterfly PUF and produces each response by utilizing an unstable value of a latch made up of interconnected logic gates like NOR/NAND gates. Delay-based PUFs make use of the unpredictable fluctuations in the logic gate and connection delays. Some examples of these types of PUFs are Arbiter PUFs (exploit race conditions in ICs) [11,12], RO PUFs(frequency variations found in ICs), Glitch PUFs [13], etc.

In this research, a uniform Ring Oscillator Physical Unclonable Function (RO PUF) architecture is proposed to address the trade-off between area utilization, reliability, and uniqueness. The objective is to design a RO PUF that occupies a smaller area compared to conventional designs while maintaining high reliability and uniqueness.

The remaining portions of the paper are structured as follows: The related research on RO PUFs is discussed in Section II. The proposed RO PUF design is introduced in Section III. The implementation and experimental findings of PUFs are displayed in Section IV. Section V concludes the paper.

II. The RO PUF and Related Work

Suh and Devadas [14] created the RO PUF in 2007. Fig2 demonstrates the conventional RO PUF architecture, which includes an array of ROs, two multiplexers, two counters, and a comparator. The wire delay and inverter delay in each RO are not programmable due to manufacturing variances, which results in varying frequencies of the RO output. We may use the counters to measure the pulses in a specified unit of time by choosing two ROs based on the PUF input. The PUF output is '1', for instance, if the first counter has a higher value than the second. The PUF output is '0' in all other cases. For given a certain input, the RO PUF should always output the same bit value, but in reality, bit errors and bias are involved in the output. The bit errors of FPGA-based PUFs are directly generated by a chosen pair of ROs with close frequencies, which results in the unstable measurement in the counters and the flipped output in the comparator. Despite the systematic or correlated process variation and the environmental noise caused by the voltage and temperature variations degrading the output stability, these bit errors are still caused by the environmental noise.

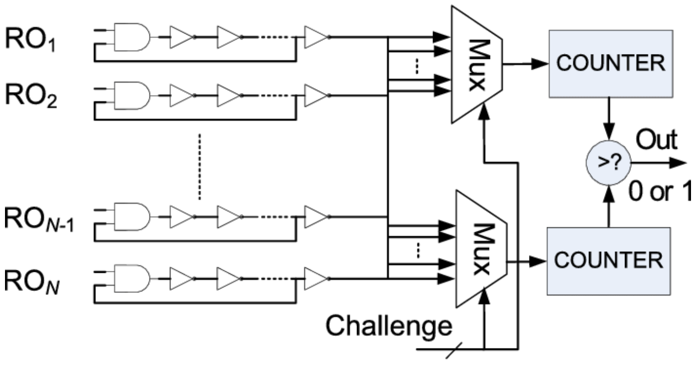


Fig. 2: Architecture-of-Conventional-RO-PUF[15]

Since then, the development of RO-PUF has involved several researchers. Chi and Gang [16] suggested a temperature-aware cooperative (TAC) approach that transforms faulty bit-producing RO pairings into dependable bit-producing RO pairs. Merli et al [17] evaluated the RO's frequency on the Xilinx field programmable gate arrays (FPGA) to improve its performance. Multi-stage ROs were tested by the authors throughout a range of runtimes. Kodytek et al.[18,19] subsequently presented a novel approach where they proposed using ring oscillators (ROs) to generate more output bits from each RO chain pair. Unlike previous designs, these ROs were not reliant on symmetry and did not impose any limitations on the placement of the ROs. This approach allowed for increased flexibility and improved performance in terms of the number of output bits produced by the RO PUF. In 2016, Rahman et al. [20] introduced a novel design for aging-resistant RO PUFs, which aimed to mitigate the impact of negative bias temperature instability (NBTI) and hot carrier injection (HCI) stress. Their approach involved incorporating additional ring oscillators (ROs) into the design, effectively replacing the conventional error correction scheme. By doing so, the degradation of RO frequency was significantly slowed down, resulting in fewer bit flips occurring in the PUF over time. In 2017 Chen et al [21] propose a statistical modeling approach using Gaussian Mixture Models (GMMs) to accurately capture the behavior of RO PUFs. The study demonstrates the effectiveness of GMM-based modeling in predicting RO PUF responses and discusses its implications for security and reliability. In 2018 Lee et al [22] investigates the vulnerability of RO PUFs to machine learning-based attacks. The authors explore the feasibility of using supervised learning algorithms to predict RO PUF responses based on a limited number of challenge-response pairs. The study highlights the importance of considering machine learning attacks in the design and evaluation of RO PUFs. Further investigations have highlighted that interference from other circuits can affect the frequency of ring oscillators (ROs). Gag et al [23] conducted a study on the crosstalk effects in field-programmable gate arrays (FPGAs) and demonstrated the impact of coupling capacitance on interconnections through digital conversion measurements. In many cases, electronic design automation (EDA) tools automatically establish connections within the FPGA chip, which can introduce routing imbalances at specific locations. This imbalance can have a detrimental effect on RO performance, particularly in terms of reducing uniqueness, as discussed by Ikeda et al [24]. This issue was further examined by Giechaskiel et al. [25], who showed that the pulses generated by the ROs can be interfered with by other signals.

The primary challenge associated with RO PUFs is the presence of unstable ring oscillators (ROs) that exhibit sensitivity to environmental variations. Specifically, RO pairs with narrow frequency differences often introduce bit flips, which significantly diminishes the reliability of the PUF. Researchers have made continuous efforts to address this issue over the years. However, these attempts have given rise to other challenges, including a decrease in uniqueness and an increase in the utilized area of the PUF design.

Hence, there is a need for an RO PUF design that achieves a balance between area utilization, reliability, and uniqueness. This research introduces a uniform RO PUF architecture that utilizes a smaller area compared to conventional designs. The main focus of this study is to enhance PUF stability while utilizing limited hardware resources. By manipulating the enable signal, multiple output bits of responses are generated. The research delves into investigating the instability characteristics of PUF responses and presents a method to transform unstable RO pairs into stable ones, thereby improving overall PUF stability.

III. Architecture of Proposed RO PUF

While significant research has been conducted on RO PUFs, there is still room for improvement, particularly in the context of FPGA implementations. In this section, we will discuss several valuable hardware enhancements for RO PUFs in FPGA designs.

A. The Stable RO-PUF Design

The proposed stable RO-PUF architecture is demonstrated in Fig3.It comprises a controller, ROs, multiplexers, counters, and a shifter. The controller handles the management of RO runtimes, while the shifter is responsible for generating a multiple-bit response. Activation of an RO is achieved through an input enable mechanism. As a result, the proposed RO PUF design offers advantages such as increased utilization of ROs, improved stability of the ROs, generation of multiple output bits, and reduced hardware requirements.

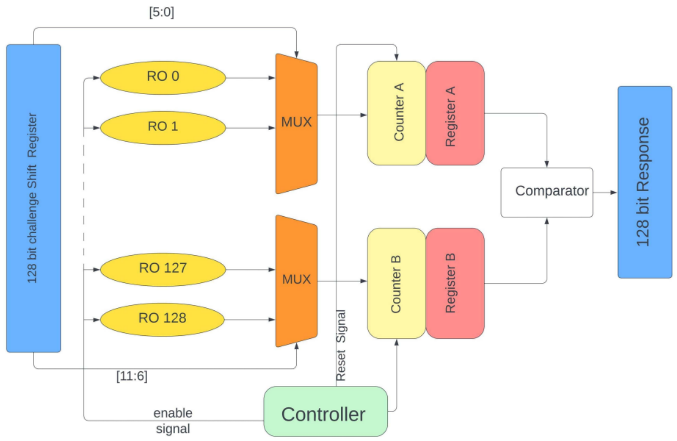


Fig. 3: Architecture of Proposed RO PUF

B. Stable RO Design

Through experimentation on the Altera Cyclone II FPGA, it was determined that a five-stage inverter chain Ring Oscillator (RO) operates at a frequency of around 400 MHz. However, this high frequency causes a violation of the counter block's setup time, resulting in measurement instability. As with all oscillators, the rate of oscillation is determined by the length of a delay implemented in a loop. Thus, to reduce the frequency, multiplexer-based ROs is used instead of inverters, and the output frequency is divided by two. According to research [26] mux-based RO has good symmetry, stability, and uniqueness. Fig4 shows the structure of our RO.

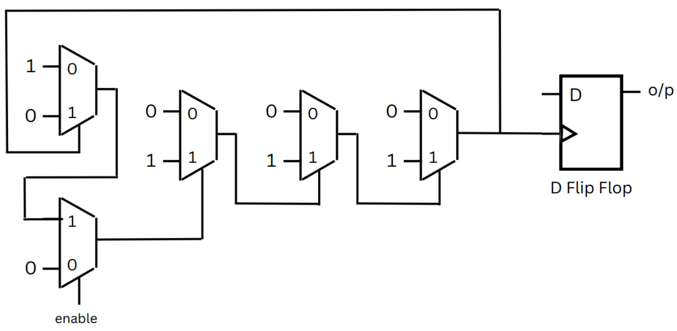


Fig. 4: RO Structure

C. CRP Generation Circuit

In order to generate variable-length responses and utilized all ROs the design is shown in Fig3. This example uses 128-bit ROs to generate a 128-bit response which is very less hardware compared to conventional ROPUF. Once a challenge is received it is stored in the shift register. The first six bits from the register are chosen as the selection line for the upper multiplexer (MUX), while the subsequent six bits serve as the selection line for the lower MUX. These selection lines determine the chosen ring oscillator (RO) from a RO array. A comparison is then made between the frequencies of the RO pair, resulting in a 1-bit response that is stored in an array. When the second enable pulse is received, the shift register undergoes a rightward shift by one bit, generating new selection lines for the two MUXes. This enables another frequency comparison between two ROs, and the second response is stored in the array. By repeating this process for a 128-bit challenge, a 128-bit response can be generated through 127 shift operations.

D. Counters and Registers block

Metastability can arise at the counter's input in the conventional architecture. When the counters receive a clock rising edge for comparison, this happens. The outcome of the comparison will not be greater, smaller, or equal, but rather a metastable state if the RO output flips as the counter value changes. With the help of our FPGA implementations, we have tested this situation. The bit errors in the answers rise as a result of the metastability. It should be noted that the counters receive RO output in an asynchronous timing, while the system clock regulates their enable and disable. As a result, if the measurement window closes at the input edge, the counter value is no longer valid. Thus, the comparison outcome goes into a metastable state. To solve this problem The counter values are stored in the register. The counter can always output an accurate value prior to comparison by using a register that is in the same clock domain as the comparator. As a result, the metastability can be safely removed.

E. Controller

We use a controller block to generate internal enable and reset signals. Enable signal use for ROs and shift register. While reset signal uses for counter reset in every shift. The design of the controller is shown in Fig 5. As we can see here, the counter is running on its own output. So when the counter value is full the not gate output is 0000 and the counter stops. Enable signals are generated in some instances during the counter running time. Reset signals are also generated after every low enable signal.

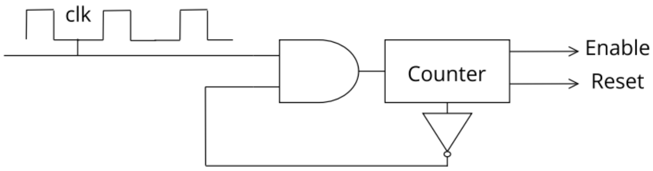


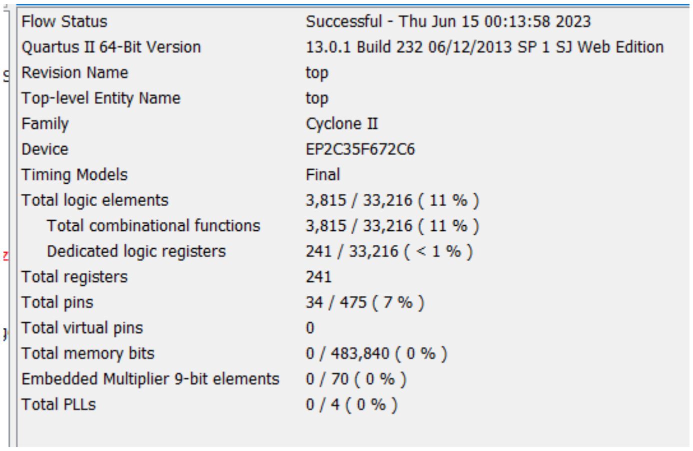
Fig. 5: Controller design

IV. Implementation and Results

The experimental work was conducted using Altera Cyclone II (EP2C35F672C6) FPGAs as the hardware platform. The system design was implemented and evaluated using the Quartus II 13.0spi edition as the software development environment.

A. Implementation Results

Resource utilization and RTL view of implemented RO PUF is shown in Fig 6 and 7.



Flow Status	Successful - Thu Jun 15 00:13:58 2023
Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition
Revision Name	top
Top-level Entity Name	top
Family	Cyclone II
Device	EP2C35F672C6
Timing Models	Final
Total logic elements	3,815 / 33,216 (11 %)
Total combinational functions	3,815 / 33,216 (11 %)
Dedicated logic registers	241 / 33,216 (< 1 %)
Total registers	241
Total pins	34 / 475 (7 %)
Total virtual pins	0
Total memory bits	0 / 483,840 (0 %)
Embedded Multiplier 9-bit elements	0 / 70 (0 %)
Total PLLs	0 / 4 (0 %)

Fig. 6: Implementation result on DE2 board

B. Simulation Results

Simulation of the designed architecture has been carried out with the help of the simulation tool ModelSim Altera and the obtained results are presented in Fig 8 and 9:

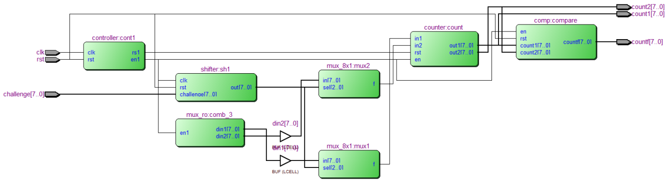
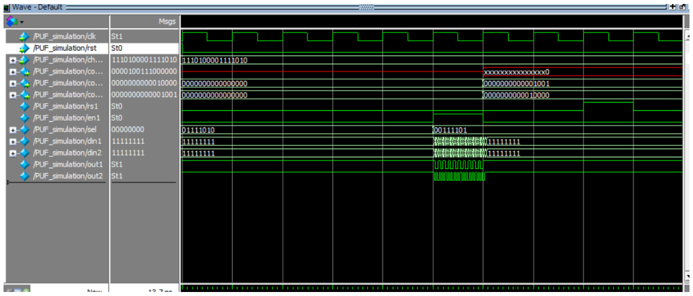
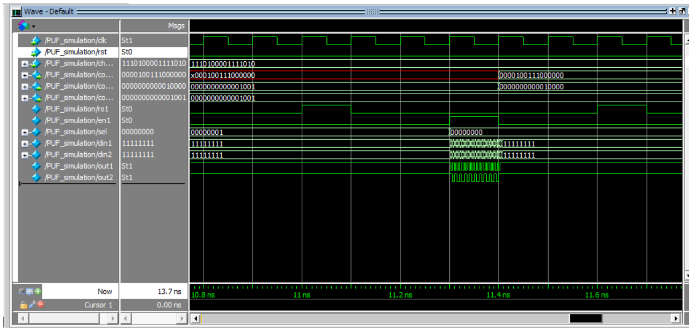


Fig. 7: RTL View of Implemented RO PUF



```
/PUF_simulation/clk      St1
/PUF_simulation/rst      St0
/PUF_simulation/ch...    1110100001111010   1110100001111010
/PUF_simulation/co...    0000100111000000                       xxxxxxxxxxxxxxx0
/PUF_simulation/co...    0000000000010000   0000000000000000    0000000000001001
/PUF_simulation/co...    0000000000001001   0000000000000000    0000000000010000
/PUF_simulation/rs1      St0
/PUF_simulation/en1      St0
/PUF_simulation/sel      00000000           01111010            00111101
/PUF_simulation/din1     11111111           11111111            11111111
/PUF_simulation/din2     11111111           11111111            11111111
/PUF_simulation/out1     St1
/PUF_simulation/out2     St1
```

Fig. 8: Simulation: Challenge Received



```
/PUF_simulation/clk      St1
/PUF_simulation/rst      St0
/PUF_simulation/ch...    1110100001111010   1110100001111010
/PUF_simulation/co...    0000100111000000   x000100111000000    0000100111000000
/PUF_simulation/co...    0000000000010000   0000000000001001    0000000000010000
/PUF_simulation/co...    0000000000001001   0000000000001001
/PUF_simulation/rs1      St0
/PUF_simulation/en1      St0
/PUF_simulation/sel      00000000           00000001            00000000
/PUF_simulation/din1     11111111           11111111            11111111
/PUF_simulation/din2     11111111           11111111            11111111
/PUF_simulation/out1     St1
/PUF_simulation/out2     St1
Now        13.7 ns    10.8 ns    11 ns    11.2 ns    11.4 ns    11.6 ns
Cursor 1   0.00 ns
```

Fig. 9: Simulation: Response Generated

C. Experimental Results

The evaluation process of the ring oscillator PUF developed in this study uses three performance metrics commonly used in almost all recent related work: uniqueness, uniformity, and reliability[27,28]. The work was implemented and tested on the Six FPGAs (F1, F2, F3, F4, F5, and F6). Therefore, the final experimental results represent an average value obtained from the 6-FPGAs, (average of 6 measurements).Table I provides the experimental parameters used to determine the PUFs' performance metrics.

TABLE I: EXPERIMENTAL PARAMETERS

Parameters	Definitions
n	Number of response bits
$r_{i,1}$	The l^{th} bit of n-bit response
p	The number of ROPUF instances
$HD\left(R_i, R_j\right)$	Hamming distance for the response of different PUFs
${R_i}^t$	The t^{th} sample of the R_i response
$HD\left(R_i, R_i^t\right)$ same PUF	Hamming distance for two response samples of the

1) Uniqueness: The replies of each chip should be unexpected since the output response of a PUF will be utilized for security applications, such as device authentication and key creation. When the same challenge input is presented to many PUF implementations, uniqueness measures how easily those replies may be distinguished. The average intra-chip Hamming Distance (HD) can be used to define uniqueness. The ideal uniqueness value is 50%, which can be determined in (1). The uniqueness of the proposed RO PUF is 49.16% which is closer to its ideal value.

$$\text{Uniqueness } = \frac{2}{p(p-1)} \sum_{i=1}^{p-1} \sum_{j=i+1}^{p} \frac{HD\left(R_i, R_j\right)}{n} \times 100\% \tag{1}$$

2) Reliablity: A PUF design should always result in the same response to a given challenge. The response can, however, be impacted by changes in the supply voltage and temperature. The resilience of the PUF design under various environmental conditions is evaluated by reliability. By computing the average intra-chip HD, it is possible to determine the PUF reliability, which is measured by the bit error rate. The ideal uniqueness value is 100%, which can be determined as in (2). The reliability of the proposed RO PUF is approx 98.7% which is approx 4% more than other RO PUF architectures.

$$\text{Reliability } = \left(1 - \frac{1}{p} \sum_{i=1}^{p} \frac{HD\left(R_i, R_i^t\right)}{n}\right) \times 100\% \tag{2}$$

3) Uniformity: The ratio of 1s or 0s to the total number of PUF response bits is determined by uniformity. The PUF answer must have an equal amount of 1s and 0s in order for this measure to be perfect, which means that the ideal uniformity value should be 50%. Uniformity can be calculated using equation (3). The uniformity of the proposed RO PUF is 49.37% which is closer to its ideal value.

$$\text{Uniformity } y_i = \frac{1}{n} \sum_{l=1}^{n} r_{i,l} \times 100\% \tag{3}$$

Table II compares the proposed RO PUF work with related work in terms of uniformity, uniqueness, and reliability.

TABLE II: PERFORMANCE COMPARISON OF THE RELATED WORK

ROPUF structure	Uniqueness %	Uniformity %	Reliability %
Sahoo DP, et al. [29]	47.57	47	90.70
Gao et al. [30]	47.31	-	95.95
Zhang et al [31]	49.33	49.33	95.45
Liu et al[32]	48.76	-	97.72
Proposed RO PUF	49.16	49.37	98.7

The Findings in Table II indicate that the ring oscillator PUF, which was introduced in this study, exhibits better performance in terms of uniformity, reliability, and uniqueness compared to the existing related research.

V. Conclusion

In this paper, we presented a practical design and implementation of RO PUFs on FPGAs. Our primary objective was to enhance the reliability of FPGA-based RO PUFs by addressing the issue of metastability. We accomplished this by utilizing stable RO pairs instead of unstable ROs, which significantly improved the stability and security of the RO PUFs.To increase the number of Challenge-Response Pairs (CRPs), we incorporated a controller and Shifter logic into our design. This allowed us to generate variable-length responses by shifting the shift register, resulting in a higher number of CRPs for enhanced security and versatility.

In terms of performance, our implemented design surpassed several existing FPGA-based ring oscillator PUFs in terms of uniqueness, uniformity, and reliability. Through careful considerations and hardware improvements, we achieved better overall performance and robustness compared to state-of-the-art approaches. Overall, our research demonstrates the practicality and effectiveness of our RO PUF design on FPGAs, offering improved reliability(approx 4%), security, and performance. These findings contribute to the advancement of PUF technology and provide a solid foundation for further research and development in the field of hardware-based security.

In Future work, investigating the integration of this RO PUF architecture with cryptographic systems which involves exploring techniques for key generation, secure key storage, and secure communication protocols that leverage the unique properties of RO PUFs to enhance overall system security.

References

[1] W. Hu, C. -H. Chang, A. Sengupta, S. Bhunia, R. Kastner and H. Li, ”An Overview of Hardware Security and Trust: Threats, Countermeasures, and Design Tools,” in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 40, no. 6, pp. 1010-1038, June 2021, doi: 10.1109/TCAD.2020.3047976.

[2] Gassend, Blaise, et al. ”Silicon physical random functions.” Proceedings of the 9th ACM Conference on Computer and Communications Security. 2002.

[3] B. Kim, S. Yoon, Y. Kang, and D. Choi, ”PUF based IoT Device Authentication Scheme,” 2019 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, Korea (South), 2019, pp. 1460-1462, doi:10.1109/ICTC46691.2019.8939751.

[4] W. Che, F. Saqib and J. Plusquellic, ”PUF-based authentication,”2015 IEEE/ACM International Conference on Computer-Aided Design(ICCAD), Austin, TX, USA, 2015, pp. 337-344, doi: 10.1109/ICCAD.2015.7372589.

[5] Maes, R. (2013). PUF-Based Entity Identification and Authentication. In: Physically Unclonable Functions. Springer, Berlin, Heidelberg.https://doi.org/10.1007/978-3-642-41395-75

[6] Anagnostopoulos, Nikolaos Athanasios, et al. ”An overview of DRAM-based security primitives.” Cryptography 2.2 (2018): 7.

[7] A. R. Korenda, F. Afghah, B. Cambou and C. Philabaum, ”A Proof of Concept SRAM-based Physically Unclonable Function (PUF) Key Generation Mechanism for IoT Devices,” 2019 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), Boston, MA, USA, 2019, pp. 1-8.

[8] R. Maes, P. Tuyls, I. Verbauwhede, Intrinsic PUFs from Flipflops on Reconfigurable Devices, in: 3rd Benelux Workshop Information and System Security, 2008, p. 17.

[9] S. S. Kumar, J. Guajardo, R. Maes, G. J. Schrijen, P. Tuyls, Extended abstract: The butterfly PUF protecting IP on every FPGA, in: IEEE International Workshop on HardwareOriented Security and Trust, 2008,pp. 67–70

[10] Y. Su, J. Holleman, B. Otis, A 1.6pJ/bit 96% Stable Chip-ID Generating Circuit using Process Variations, in: 2007 IEEE International Solid-State Circuits Conference. Digest of Technical Papers, 2007, pp. 406–611.

[11] . Gu, W. Liu, Y. Cui, N. Hanley, M. O'Neill, F. Lombardi, A Flip-Flop Based Arbiter Physical Unclonable Function (APUF) Design with High Entropy and Uniqueness for FPGA Implementation, IEEE Transactions on Emerging Topics in Computing (TETC) (2019).

[12] D. P. Sahoo, D. Mukhopadhyay, R. S. Chakraborty, P. H. Nguyen, A Multiplexer-Based Arbiter PUF Composition with Enhanced Reliability and Security, IEEE Transactions on Computers 67 (3) (2018) 403–417.

[13] D. Suzuki, K. Shimizu, The Glitch PUF: A New Delay-PUF Architecture Exploiting Glitch Shapes, in: S. Mangard, F.- X. Standaert (Eds.),Cryptographic Hardware and Embedded Systems, CHES 2010, Springer Berlin Heidelberg, 2010, pp. 366–382

[14] Suh, G.E.; Devadas, S. Physical unclonable functions for device authentication and secret key generation. In Proceedings of the 2007 44th ACM/IEEE Design Automation Conference, San Diego, CA, USA, 4–8 June 2007; pp. 9–14.

[15] Gao, Yansong, et al. ”mrPUF: A novel memristive device based physical unclonable function.” Applied Cryptography and Network Security: 13th International Conference, ACNS 2015, New York, NY, USA, June 2-5, 2015, Revised Selected Papers 13. Springer International Publishing, 2015.

[16] Yin, C.-E.; Qu, G. Temperature-aware cooperative ring oscillator PUF. In Proceedings of the 2009 IEEE International Workshop on Hardware-Oriented Security and Trust, San Francisco, CA, USA, 27 July 2009; pp. 36–42.

[17] Merli, D.; Stumpf, F.; Eckert, C. Improving the quality of ring oscillator PUFs on FPGAs. In Proceedings of the 5th Workshop on Embedded Systems Security, Scottsdale, AZ, USA, 24 October 2010; pp.1-9.

[18] Kodýtek, F.; Lórencz, R. A design of ring oscillator based PUF on FPGA. In Proceedings of the 2015 IEEE 18th International Symposium on Design and Diagnostics of Electronic Circuits and Systems, Belgrade, Serbia, 22–24 April 2015; pp. 37–42.

[19] Kodýtek, F.; Lórencz, R.; Buˇcek, J. Improved ring oscillator PUF on FPGA and its properties. Microprocessor. Microsyst.2016,47, 55–63.

[20] Rahman, M.T.; Rahman, F.; Forte, D.; Tehranipoor, M. An aging-resistant RO-PUF for reliable key generation. IEEE Trans. Emerg.Top. Comput. 2016, 4, 335–348.

[21] Chen, B., et al. (2017). RO PUF Modeling Using Gaussian Mixture Models. IEEE Transactions on Information Forensics and Security, 12(6), 1294-1306.

[22] Lee, C., et al. (2018). Machine Learning-Based Attacks on RO PUFs: A Case Study. Proceedings of the ACM/IEEE International Conference on Computer-Aided Design (ICCAD).

[23] Gag, M.; Wegner, T.; Waschki, A.; Timmermann, D. Temperature and on-chip crosstalk measurement using ring oscillators in FPGA. In Proceedings of the 2012 IEEE 15th International Symposium on Design and Diagnostics of Electronic Circuits and Systems (DDECS), Tallinn, Estonia, 18–20 April 2012; pp. 201–204.

[24] Ikeda, M.; Kang, H.; Iwamura, K. Direct challenge ring oscillator PUF (DC-ROPUF) with novel response selection. In Proceedings of the 2017 IEEE 6th Global Conference on Consumer Electronics (GCCE), Nagoya, Japan, 24–27 October 2017; pp. 1–2.

[25] Giechaskiel, I.; Eguro, K.; Rasmussen, K.B. Leakier wires: Exploiting FPGA long wires for covert-and side-channel attacks. ACM Trans. Reconfig. Technol. Syst. (TRETS) 2019, 12, 1–29.

[26] Yao, Liang, et al. ”M-RO PUF: a portable pure digital RO PUF based on MUX Unit.” Microelectronics Journal 119 (2022): 105314.

[27] Zulfikar, Zulfikar, et al. ”Runtime analysis of area-efficient uniform RO-PUF for uniqueness and reliability balancing.” Electronics 10.20 (2021):2504

[28] G. Komurcu and G. Dundar, ”Determining the quality metrics for PUFs and performance evaluation of Two RO-PUFs,” 10th IEEE International NEWCAS Conference, Montreal, QC, Canada, 2012, pp. 73-76, doi:10.1109/NEWCAS.2012.6328959.

[29] Sahoo, Durga Prasad, Debdeep Mukhopadhyay, and Rajat Subhra Chakraborty. ”Design of low area-overhead ring oscillator PUF with large challenge space.” In 2013 International Conference on Reconfigurable Computing and FPGAs (ReConFig), pp. 1-6, 2013.

[30] Mingze Gao, Khai Lai, Gang Qu, A highly flexible ring oscillator PUF, in: 2014 51st ACM/EDAC/IEEE Design Automation Conference (DAC), 2014, pp. 1–6.

[31] J. Zhang, X. Tan, Y. Zhang, W. Wang, Z. Qin, Frequency Offset-Based Ring Oscillator Physical Unclonable Function, IEEE Transactions on Multi-Scale Computing Systems 4 (2018) 711–721

[32] W. Liu, L. Zhang, Z. Zhang, C. Gu, C. Wang, M. O'Neill,F. Lombardi, XOR-Based Low-Cost Reconfigurable PUFs for IoT Security, ACM Trans. Embed. Comput. Syst. 18 (3) (2019) 25:1–25:21