Design and Analysis of FPGA-based PUFs with Enhanced Performance for Hardware-oriented Security

N. NALLA ANANDAKUMAR, Florida Institute for Cybersecurity (FICS) Research, University of Florida, Gainesville, FL
MOHAMMAD S. HASHMI, School of Engineering and Digital Sciences, Nazarbayev University, Kazakhstan
SOMITRA KUMAR SANADHYA, School of Artificial Intelligence and Data Science, IIT-Jodhpur, India

This article presents a thorough analysis of two distinct Physically Unclonable Functions (PUF), namely RO-PUF (Ring oscillator-based PUF) and RS-LPUF (RS Latch-based PUF), prototyped on FPGA. It is shown that the implemented PUFs possess significantly enhanced performance when compared to the state of the art. It is also identified that the enhancements are achieved through the incorporation of Programmable Delay Lines of FPGA Lookup Tables, the Temporal Majority Voting (TMV) scheme, and placed macro techniques for routing and placements of PUF units. The prototypes developed on Xilinx Artix-7 FPGAs are used for validation over the rated temperature range of $0-85^{\circ}C$ with ±5% variation in the supply voltage. The proposed schemes when evaluated experimentally also achieve good uniformity, bit-aliasing, uniqueness, and reliability. Finally, it is shown that the proposed designs outperform the existing conventional PUFs in the area and speed tradeoff.

CCS Concepts: • **Security and privacy → Security in hardware**; *Embedded systems security*; • **Hardware security implementation** → Hardware-based security protocols;

Additional Key Words and Phrases: PUF, internet of things (IoT), PDL, FPGA, RO-PUF, RS-LPUF, TMV

ACM Reference format:
N. Nalla Anandakumar, Mohammad S. Hashmi, and Somitra Kumar Sanadhya. 2022. Design and Analysis of FPGA-based PUFs with Enhanced Performance for Hardware-oriented Security. *J. Emerg. Technol. Comput. Syst.* 18, 4, Article 72 (October 2022), 26 pages.
https://doi.org/10.1145/3517813

1 INTRODUCTION

The proliferation of the **Internet of Things (IoT)** to encompass a broad spectrum of devices and applications from diverse domains such as biomedical, vehicular, home automation, sensing, and so forth is bringing monumental security-related challenges [47]. It is envisaged that **Physically**

The work was supported by the Collaborative Research Grant (CRP) Number 021220CRP0222 at Nazarbayev University.
Authors' addresses: N. N. Anandakumar, Florida Institute for Cybersecurity (FICS) Research, University of Florida, Gainesville, FL, USA, 32603; email: nallananth@gmail.com; M. S. Hashmi, School of Engineering and Digital Sciences, Nazarbayev University, Kazakhstan, 010000; email: mohammad.hashmi@nu.edu.kz; S. K. Sanadhya, School of Artificial Intelligence and Data Science, IIT-Jodhpur, India, 342030; email: somitra@iitj.ac.in.

1550-4832/2022/10-ART72 $15.00
https://doi.org/10.1145/3517813

Unclonable Function (PUF)-based security solutions could provide the requisite economic viability along with the necessary security features to the interconnected devices within the framework of IoT [14, 20, 44, 47, 70]. This can be inferred from the fact that PUFs possess the ability to derive a unique digital signature from the manufacturing process variation of ICs by using a challenge-and-response mechanism. For example, PUFs are used for secure key generation in silicon [14, 38, 54, 57], which eliminates the need for storing keys in non-volatile memory. Furthermore, PUFs can also be employed for online device authentication [54].

There has been increased usage of PUFs in various hardware-entangled cryptographic schemes and protocols, such as a PUF-based key exchange protocol [5] and PUF-based block cipher [8]. A PUF can be realized in IoT products by integrating a special PUF circuit, e.g., as a stand-alone ASIC or as part of a system on chip (SoC), or **Field-Programmable Gate Array (FPGA)**. The design of PUF primitives on FPGAs is preferred owing to their flexibility, quick turn-around, and reconfigurability. Moreover, the PUFs can be seamlessly integrated with other IPs in FPGA-based products owing to their inherent flexibility and ease of integration. Two major FPGA manufacturers, namely Intel (formerly Altera) [23] and Xilinx [49], have announced PUF implementations in their respective products for security purposes. Applications of FPGA-based PUFs are diverse and can be found in secure key generation [54], IP protection [30], IC counterfeit detection [67], and IoT security [37, 44, 70]. The literature is replete with two distinct PUFs, namely Memory-based and the Delay-based PUFs, realized on FPGAs. The PUFs making use of digital race conditions or arbitrary variations in frequency in the integrated circuits, such as Arbiter PUF [54] and **Ring oscillator-based PUF (RO-PUF)** [54], are called delay-based PUFs. On the other hand, Memory-based PUFs, such as SRAM-PUF [30] and **RS Latch-based PUF (RS-LPUF)** [65], are developed around the instability of volatile memory cells. According to the number of **challenge-response pair (CRP)** spaces, PUF architectures can be broadly divided into two categories [6]: strong PUF and weak PUF. The former has a large set of CRPs that increase exponentially with PUF size. The strong PUFs can be used directly for authentication without additional cryptographic hardware. Examples of Strong PUFs are Arbiter PUF, Lightweight Secure PUF, and Bistable Ring PUF [6]. The alternative weak PUF architectures have a limited number of CRPs, ideally increasing linearly with PUF size, and they are more suited to applications such as **Pseudo-Random Number Generators (PRNGs)** and key generation. Examples of weak PUFs are RO-PUF, SRAM-PUF, and RS-LPUF [6]. This article reports implementations of RO-PUF and RS-PUF on FPGAs suitable for lightweight security applications such as secure key generation or for seeding a PRNG.

It is well accepted that the common metrics to benchmark the PUFs, namely *uniformity*, *bit-aliasing*, *uniqueness*, and *reliability*, are often dependent on several external and internal factors. This could be understood by taking the case of variations in the environmental factors that adversely affect the *uniqueness* and *reliability* of the PUFs. Moreover, even a slight variation in the environmental factors may lead to weak safeguards against external attacks [40]. The performance of PUFs is enhanced through the TMV scheme [8], hard/placed macro techniques [40], **Programmable Delay Lines (PDLs)** [42], and combining PUF outputs [54, 64]. For example, the combination of PUF responses from multiple PDLs requires an aggregate function (XOR operation or crypto-algorithms) of PUFs existing in the system to enhance uniformity and security [4, 54, 64]. Furthermore, high-resolution PDLs implemented by a single **lookup table (LUT)** on the FPGA can significantly improve the number of independent response bits by partially alleviating the problem of systematic design bias [42]. The TMV concept aids in mitigating the variability issues and in achieving more stable results by averaging N sequential measurements [8]. The hard/placed macro technique, provided by standard design tools, is commonly used to enhance uniqueness and bit-aliasing of RO-PUF [40] and RS-LPUF [32]. However, the existing solutions, although they enhance PUF performance, are still inferior when compared to the ideal desired metrics.

1.1 Our Contributions

This article advances the state of the art in the domain of FPGA-based PUF primitives. This has been achieved by incorporating the TMV scheme, placed macro techniques, and coarse, or fine, programmable delay lines in conjunction with conventional PUF modules concurrently. A preliminary investigation was reported earlier [3] and now a fairly detailed design, analysis, and investigation are reported in this article. The objective in this work is to achieve area-efficient and speed-enhanced, distinct PUF primitives that substantially improve performance in terms of *uniqueness*, *reliability*, *uniformity*, and *bit-aliasing*. The main features of this work include (1) significantly modified RS-LPUF and RO-PUF through incorporation of the TMV scheme and coarse PDL techniques; (2) validation of all these PUF primitives over a wide range of temperatures ($0 - 85^{\circ}C$) with ±5% variation in the supply voltage, which is very exciting considering that earlier designs [3] were validated only at a normal operating temperature and core voltage of 1.2V; (3) entropy estimation, auto-correlation test, and effect of aging on the two proposed PUF designs; and (4) proposal of distinct statistical analysis metrics to assess the performance of all the PUFs. The main contributions of this article are summarized below:

- Area-efficient RO-PUF and RS-LPUF designs on Xilinx Artix-7 FPGAs. The required respective slices for the proposed RO-PUF and RS-LPUF are 107 and 101, respectively.
- Demonstration of increase in the number of independent responses through the use of fine and coarse programmable delay lines of FPGA LUTs.
- Proposal of more stable (i.e., more reliable) PUFs by incorporating of TMV scheme in the conventional PUFs.
- Achievement of better PUF performance in terms of uniformity, bit-aliasing, and uniqueness by XORing PUF responses together from multiple PDL configurations and utilizing the placed macro design technique (i.e., placement strategy) to make all the PDLs identical in terms of placement and routing.
- Detailed analysis of the proposed RS-LPUF and RO-PUF designs in terms of correlation resistance, attack resilience, entropy, and aging effects.

The organization of the article is as follows. Section 2 briefly discusses the Xilinx Artix-7 FPGAs, PDLs, and common PUF performance metrics. The implementation details of the proposed PUFs are presented in Section 3. Possible attacks on the proposed architectures and the countermeasures are discussed in Section 4. Experimental validation of the proposed design is given in Section 5, and discussion on implementation results is given in Section 6. Finally, conclusions are presented in Section 7.

2 PRELIMINARIES

2.1 Xilinx Artix-7 FPGA Structure

In this work, the Artix-7 FPGA (28 nm CMOS) from Xilinx has been used to prototype the two proposed PUF designs. An FPGA consists of an array of **Configurable Logic Blocks (CLBs),** which are made of slices. The Artix-7 FPGA has two distinct slices called Slice L and Slice M. Each CLB has two slices; i.e., both are Slice L or one Slice L and one Slice M. Furthermore, each slice contains four LUTs, which have six inputs, and eight **flip-flops (FFs)**. The work proposed in this article requires instantiation of the six-input primitives [62] for realization of logic gates.

2.2 Programmable Delay Lines (PDLs)

The inputs of LUTs can be regulated to manipulate the propagation delays, and this essentially aids in the generation of the internal variations of the FPGA LUTs [42, 43]. For example, as shown

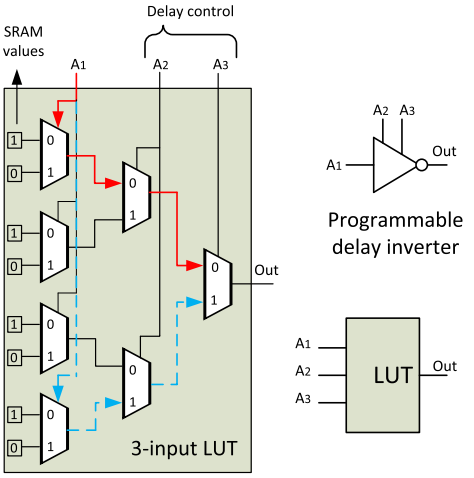

Fig. 1. PDL using a three-input LUT [43].

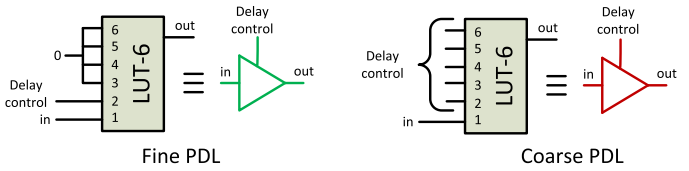

Fig. 2. Fine and coarse PDLs implemented by a single six-input LUT [43].

in Figure 1 [43], the LUT is programmed in such a way that an inversion of its first input (A_1) gives the LUT output (O). Subsequently, the inputs A_2 and A_3, even though they are "don't-care" bits, still aid in the generation of the desired propagation delay from A_1 to the output (O). The earlier work reported that the shortest input to output propagation path in such a three-input LUT is for $A_2A_3 = 00$ (solid red arrowed lines), whereas the longest path is for $A_2A_3 = 11$ (dashed blue arrowed lines) [43]. It is pertinent to mention that the Artix-7 FPGA, with six-input LUTs, used in this work facilitates fine and coarse PDLs. The generic configurations of fine and coarse PDLs on the six-input LUTs (i.e., five control inputs) are depicted in Figure 2. For the fine PDL, the first LUT input A_1 is the inverter input and the LUT inputs A_3 to A_6 are fixed to zero, whereas the only input that controls the delay is A_2 [43]. For the coarse PDL, A_1, the first LUT input, is the input of the inverter, whereas the other LUT inputs (A_2 to A_6) are regulated by 32 (=2^5) discrete levels.

2.3 PUF Performance Metrics

The important performance metrics are *uniqueness*, *reliability*, *uniformity*, and *bit-aliasing* [39]. These metrics are used to quantify a PUF's performance across multiple chip dimensions. Next, we provide a brief description of the quality metrics.

(1) *Uniqueness (UQ):* This metric measures the variation of responses generated from different PUF chips for the same set of challenges. If S_i and S_j are the n-bit responses of the ith and jth chips, respectively, for the same challenge C, then the *uniqueness* (HD_{INTER}) is given by the average inter-chip hamming distance (HD) among the k devices and is calculated by Equation (1):

$$Uniqueness = \frac{2}{k(k-1)} \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \frac{\mathrm{HD}(S_i, S_j)}{n} \times 100\%, \tag{1}$$

where $HD(S_i, S_j)$ is the Hamming distance between n bit strings S_i and S_j, and k is the number of PUF instances (tested devices). The ideal value of *uniqueness* is 50%.

(2) *Reliability (RE):* This metric determines how the PUF device can perfectly reproduce its output whenever it is queried with a challenge at different operating conditions (i.e., temperature and supply voltage variations) over a period of time. For the i^{th} device, reliability is represented as a single value by finding the average intra-chip HD of m response samples, $S_{i,t}$; this is taken at different operating conditions compared to a baseline n bit reference response, S_i, taken at nominal operating conditions. The average intra-chip HD is estimated using Equation (2). Then the *reliability* of a PUF chip is defined in Equation (3):

$$HD_{INTRAi} = \frac{1}{m}\sum_{t=1}^{m}\frac{HD(S_i, S_{i,t})}{n} \times 100\%, \tag{2}$$

$$Reliability_i = 100\% - HD_{INTRAi}. \tag{3}$$

The ideal value of HD_{INTRA} is 0% (i.e., ideal value for *reliability* is 100%) and the *average reliability* of k chips can be calculated using Equation (4):

$$Average\ Reliability = \frac{1}{k}\sum_{i=1}^{k} Reliability_i. \tag{4}$$

(3) *Uniformity (UF):* It is a measure of the proportion of zeros and ones across the whole of the response set of the PUF and uses Equation (5):

$$Uniformity_i = \frac{1}{n}\sum_{j=1}^{n} u_{i,j} \times 100\%, \tag{5}$$

where $u_{i,j}$ is the jth bit of the n-bit response of the i^{th} chip. The ideal value of *uniformity* is 50%.

(4) *Bit-aliasing (BA)* happens when different chips may produce nearly identical PUF responses for certain challenges. The *bit-aliasing* is calculated using Equation (6):

$$Bit\text{-}aliasing_j = \frac{1}{k}\sum_{i=1}^{k} b_{i,j} \times 100\%, \tag{6}$$

where k is the number of chips/devices and $b_{i,j}$ is the jth bit of the n-bit response of the i^{th} chip. The ideal value of *bit-aliasing* is 50%.

3 DESIGN AND IMPLEMENTATIONS

The optimized implementations of RO-PUF and RS-LPUF along with their performance assessments on Xilinx Artix-7 FPGAs. These PUFs make use of Xilinx Vivado design tool 20.1 and the coding is carried out using Verilog HDL. The PUFs communicate with the user PC (MATLAB) through the universal asynchronous receiver-transmitter (UART) 8-bit interface.

3.1 RO-PUF

In this PUF, the response is derived from the difference in oscillator frequencies of selected pairs of ROs [40]. The RO-PUF, in Figure 4, on Artix-7 devices incorporates PDLs and realizes two ROs inside a single CLB, but each RO is placed in a single slice, each with a different color scheme, as shown in Figure 3. Each RO is realized using three inverters and one AND gate, as can be seen in Figure 5 (black dashed lines). The designs of inverters and the AND gate use three LUTs and one LUT, respectively. It is pertinent to note that placement and routing of ROs in FPGA are extremely important; otherwise, the quality of the PUFs is affected in addition to an undesired introduction

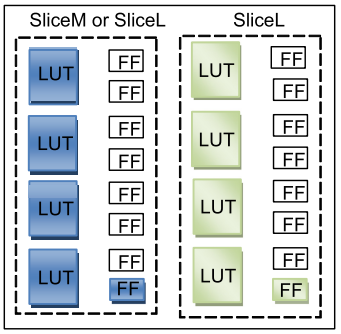

Fig. 3. Implementation of two ROs per CLB.

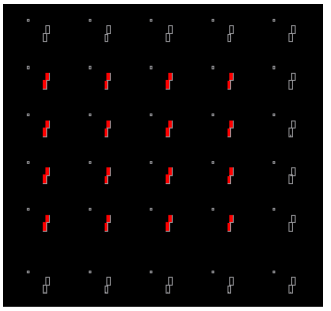

Fig. 4. PUF array configuration of 32 ROs.

of bias in the PUF responses [27, 34, 40, 53]. Moreover, a full-chip characterization of Xilinx FPGAs based on a RO-PUF in-depth analysis was presented in [27, 34], and they show that the locations of slices can impact the quality of the PUF implemented on FPGAs. It has been earlier reported that digital circuits and other IP cores' activity in SoC significantly influence the reproducibility of PUF responses [53]. Furthermore, the ring oscillator frequencies also depend on their location on an FPGA die; e.g., frequencies of ROs at the center of an FPGA are higher than edge ROs [40]. Moreover, the authors in the above works [27, 34, 40, 53] have pointed out some potential techniques to reduce design bias in the PUF placement and improve the quality of the PUFs, such as placing and routing PUF circuits on the unused hardware resources, near the complex logic that implements the entire system; placing the macro design technique; avoiding unnecessary nearby switching logic, placing each RO in the single same slice of CLB; adding additional feedback to the ring oscillators in order to avoid glitches; employing error correction circuitry; and so forth. For further details, one may refer to [27, 34, 40]. In this context, we employed directed placement constraints of FPGA LUT to fix the internal routing path of each RO [63]. However, the Xilinx Vivado tool automatically routes the other parts. Our proposed design in this article consists of 32 ROs that are not placed exactly at the center of the chip (i.e., configured in between the right edges and the center of FPGA) in a 4×4 matrix of CLBs, as can be seen in Figure 4 (identified by red mark). To overcome the bias induced by design, we make use of the placed macro technique [63] to make all the ROs identical by placing them at the selected locations. Furthermore, in this work, we incorporated PDLs to increase the randomness and uniqueness, and the TMV technique to reduce instability of the response and improve reliability.

For the generation of programmable delays inside the six-input LUT to achieve fine PDLs, one of the inputs to the LUT is used for ring connection and the other input is configurable. The rest of the LUT inputs are fixed to zero. The coarse PDL is realized by using of the inputs of LUT for ring connection, using one LUT input as the challenge bit, and then using all other LUT inputs for configuring as $2^4 = 16$ discrete levels (from 0000 to 1111). In addition, one of the inputs to the LUT of the AND gate is used for enabling the RO, as evident in Figure 5.

Algorithm 1 describes the process of generating response bits by the proposed design of RO-PUF. As shown in Figure 5, for the realization of RO-PUF, first the 8-bit master challenge from a personal computer is initiated through a UART for the 8-bit Galois LFSR [22] (having maximum cycle length). As a consequence, 256 subsequent challenges are generated. Then from each of these subchallenges two different ROs are chosen for comparison. The sub-challenges consist of two parts: the 4 **least significant bits (LSBs)** select 1 of the 16 ROs in group 1 (gray dashed lines) through the multiplexer 1, while the 4 **most significant bits (MSBs)** of the subchallenge select 1 of the 16 ROs in group 2 (green dashed lines) through the multiplexer 2. The frequency of the selected

ALGORITHM 1: Pseudocode for response generation from the proposed RO-PUF design using coarse PDLs

```
Input: 8-bit master challenge
Output: 256-bit final response
/* Pseudocode: Steps for Response bits generation                                   */
Generate 256 sub-challenges from the 8-bit master challenge
/* applied to the delay control inputs for each sub-challenge                      */
sub-challenge ← 1
for sub-challenge = 1 to 256 do
    Delay control inputs ← 0
    for Delay control inputs = 0 to 15 do
        /* each delay control inputs is applied 15 times                           */
        apply each delay control inputs ← 0
        reference counter ← 0
        while apply each delay control inputs < 15 do
            while reference counter < maximum value do
                Counter 1 ← frequency of selected one of the 16 ROs in group 1
                Counter 2 ← frequency of selected one of the 16 ROs in group 2
                reference counter ← reference counter + 1
            end
            if reference counter = maximum value then
                if Counter 1 > Counter 2 then
                    Raw response bit 1
                end
                else
                    Raw response bit 0
                end
                reference counter ← 0
            end
            15-bit shift register ← Raw response bit
            apply each delay control inputs ← apply each delay control inputs + 1
        end
        /* TMV concept is applied on 15-bit shift register                         */
        if more than 50 percent of the bits in the raw response are 1s then
            Golden response bit 1
        end
        else
            Golden response bit 0
        end
        16-bit shift register ← Golden response bit
        Delay control inputs ← Delay control inputs + 1
    end
    /* Final response generation                                                   */
    Final response bit ← XORing of the sixteen 1-bit golden responses
    /* Final responses are stored in a dedicated 256-bit shift register        */
    256-bit shift register ← Final response bit
    sub-challenge ← sub-challenge + 1
end
return 256 bit shift register value (i.e., 256 bit final response)
```

ROs in group 1 and group 2 are then obtained and fed into the 32-bit respective counters. A crystal 100-MHz clock signal generated by an on-board oscillator drives the 8-bit reference counter. The counter 1, counter 2, and reference counter start counting at the same time and are forced to stop when the reference counter hits its maximum value. Then, the comparison of counter 1 and counter 2 values generates a response bit 0 or 1 for this RO pair depending on which counter had the higher value. For the fine PDLs, each sub-challenge requires application of 0 and 1 to the delay control inputs. On the other hand, for the coarse PDLs, the delay control inputs get 2^4 (= 16) discrete values for each sub-challenge. A similar strategy is also adopted in the RS-LPUF design later in the article. This article therefore makes use of TMV prior to the XOR operation as this helps in increased attack complexity and reduced response instability [61]. As a first step, the generated raw responses,

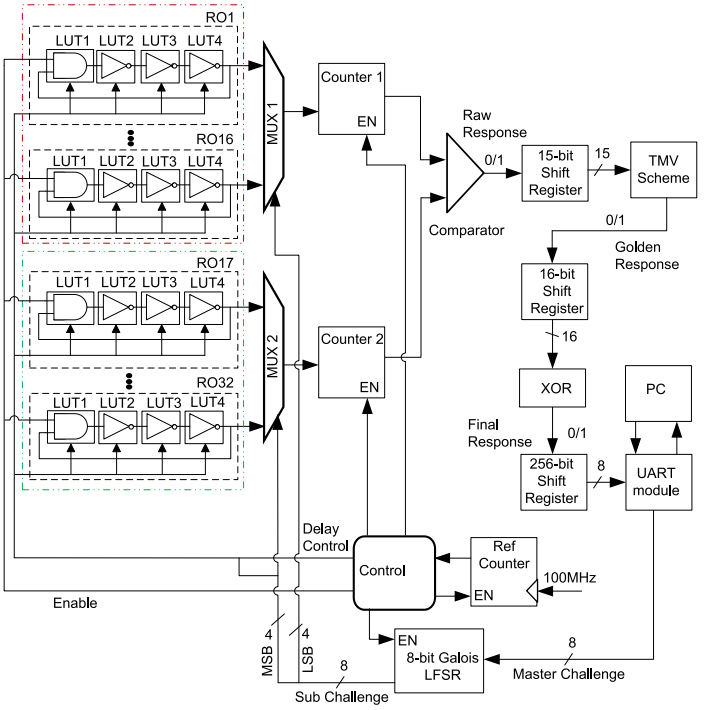

Fig. 5. Proposed design of RO-PUF.

which are the outcome of the application of delay control inputs 15 times, are stored in a dedicated 15-bit shift register. In the second step, the golden response is generated by applying the TMV to these stored bits. The golden response is either 1, if more than 50% of the bits in the raw response are ones, or 0, for the case when more than 50% of bits in the raw response are zeros, and stored in a 16-bit shift register. The generated golden responses are stored in a 2-bit shift register for the case of fine PDLs. Finally, the final response that is of 1 bit is achieved by the XORing of the sixteen 1-bit golden responses in the case of coarse PDLs. Alternatively, for the fine PDLs, XORing is done for the two 1-bit golden responses for the eventual creation of a final response. As a consequence, each 8-bit master challenge creates 256 final response bits. These final responses are stored in a dedicated 256-bit shift register. Finally, the generated response bits are sent to the personal computer using a UART for the PUF quality analysis. The controller circuitry is responsible for starting and stopping the ROs using the enable input, selecting the ROs, counting the oscillations, and returning the responses based on their comparisons.

3.2 RS-LPUF

The proposed PUF response is derived from the difference in counting the numbers of ones of the selected pairs of RS-LPUFs by applying consecutive rising edges. For the RS latch cell, in Figure 6, when the input is zero in a stable state, then the output is one. As the input of the RS latch changes from zero to one, the RS latch cell first enters a metastable state. Then it settles down to either output zero or one at the end of metastable state.

In our work, two NAND gates, one NOT gate, and one FF (for each SR-latch) are implemented in a single slice on one CLB as shown in Figure 7. We use three LUTs and one FF to create one SR-latch. The LUTs are used to create NOT and NAND gates, while the FF, which precedes the two NAND gates in Figure 6, reduces the clock skew. Furthermore, two latches are implemented inside a single CLB, whereas each RS latch is placed in a single slice. These are shown in Figure 7

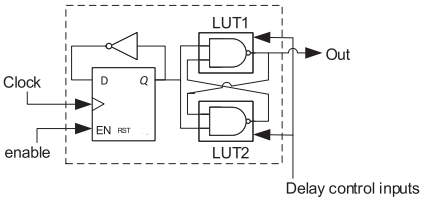

Fig. 6. Configurable RS latch cell.

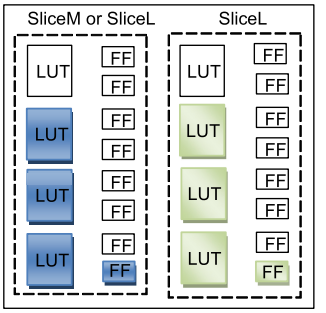

Fig. 7. Implementation of two SR latches per CLB.

with different colors. In summary, our work uses only 32 RS latches for generating 256 response bits, whereas [65] and [32] require 128 RS latches and 512 latches, respectively. Subsequently, we employ the PDL concept in the PUF design for improving the random PUF response. In the design of fine PDL, we use two inputs of each six-input LUT for connection, i.e., the NAND gate output and the flip-flop output. One of the LUT inputs is a configurable bit, while the rest of the LUT inputs are fixed to zero. The coarse PDL requires two inputs of each of the LUTs for connection. Furthermore, this work configures the rest of the inputs of LUTs in 16 discrete levels (from 0000 to 1111).

The proposed RS-LPUF architecture is shown in Figure 8. A 100-MHz clock signal (which is generated by an on-board oscillator) is applied to each FF, which divides it into a 50-MHz clock signal that is applied to the corresponding RS latches. The D-type FF acts as a frequency divider by feeding back the output from Q to the input terminal D; the output pulses at Q have frequencies that are exactly one-half that of the input clock frequency, as can be seen in Figure 8. The RS latches are stopped using the enable signal. The FF is reset before application of the enable signal to ensure that the latches always start with the same initial state.

Once again, similar to the RO-PUF approach, the generation of 256 challenges and selection of two SR latches are done in two groups based on the subsequent challenge inputs. In every clock cycle, the outputs of selected RS latches in group 1 and group 2 are fed into the 8-bit counters 1 and 2, respectively. Then both counter values are incremented every time selected RS latches output ones. The counter 1, counter 2, and 8-bit reference counter start counting at the same time and are forced to stop when the reference counter hits its maximum value. The comparison of counter 1 and counter 2 values generates a response bit 0 or 1 for this RS latch pair, depending on which counter had the higher value. Once again, the final responses are generated by employing the PDL and TMV scheme and stored in a 256-bit shift register before PUF quality analysis.

4 SECURITY ANALYSIS

The silicon PUFs have received a lot of attention and they have been adopted by industry for many hardware-oriented cryptographic applications [28]. However, several attacks have been reported to break the PUF security successfully [6]. In this article, we will discuss **machine learning (ML)**- and side-channel-based attacks and the countermeasures in detail.

4.1 Machine Learning (ML)-based Modeling Attacks

A literature survey reveals that strong PUFs are more commonly subjected to ML-based modeling attack nowadays [6, 24, 50]. In such a scheme, the attacker relies on the PUF model considering that it is trained using a subset of the PUF CRPs. This allows the attacker to figure out the responses coming out of unknown challenges. In general, the ML-based model has been used to

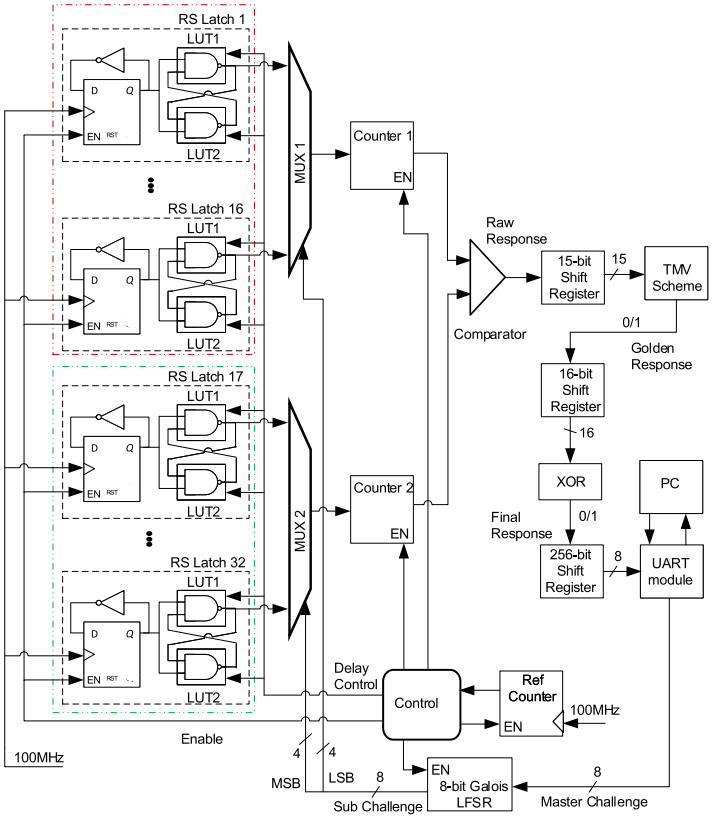

Fig. 8. The proposed design of RS-LPUF.

attack PUF circuits that were based on the physical variations of delay elements by using **Logistic Regression (LR)**, and **Support Vector Machine (SVM)** [50]. A number of countermeasures exist to overcome ML-based modeling attacks. These include incorporation of non-linearity to the PUF structures, XORing of multiple individual responses, random selection of a substring of the response, and randomization of the challenges [17, 69]. These countermeasures obviously, although they have been proven to be successful, either increase the complexity of the PUFs or make the protocol-level protections complex. In the current work, we have incorporated appropriate defense mechanisms to increase the cost of ML-based attacks before deriving the final response in the proposed PUF designs. Some of these are concept of majority voting before the XOR operation [61], generation of internal challenges from the external challenge, discrete PDL configurations, and obfuscation of the response by XORing multiple individual responses by varying the PDL inputs to form a single-bit response [6]. However, the ML-based modeling attacks need a huge amount of PUF CRPs during the learning phase. Therefore, this attack will not be effective for weak PUFs such as the SRAM PUF, our two proposed PUF designs, and similar architectures [6, 70]. The two proposed designs in the current work fall into the classification of weak PUFs since they use a small number of CRPs and use two independent ROs (or RS latches) for single-bit response generation for each challenge. As shown later in Section 5.6, the generated response bits of the proposed PUF designs are uncorrelated. This will make it very hard for an adversary to predict response to an unknown challenge even when given a set of known CRPs. Further, the proposed PUFs are aimed at applications such as key generation. In such applications, there is no access interface to read the response inside the chip for an attacker and thus he or she does not have very large CRP space. As mentioned in [72], the CRP access interface is implemented by fuses that are destroyed

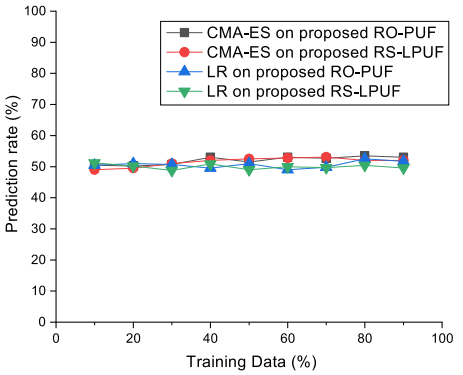

Fig. 9. Prediction rates for proposed PUF designs by LR and CMA-ES attacks.

after designers obtain the CRPs. Therefore, it is difficult to conduct ML-based modeling attacks on our proposed PUFs designs. However, for the sake of completeness, we have still carried out analysis and investigation for the expected performance in case the proposed PUF designs in this article are subjected to ML attack.

In this work, we adopted two common machine learning algorithms, LR and **covariance matrix adaptation evolution strategies (CMA-ES)**, to measure the modeling attack resilience of the proposed PUFs. We have used Python's packages, scikit-learn [19] and pycma [33], for LR and CMA-ES, respectively. To carry out the attack, we first obtained all the CRPs (i.e., 256 CRPs) from the proposed PUF designs on FPGAs. Then, we selected a certain percentage of CRPs as the training set to train these algorithms. The remaining CRPs constituted a test set to test the prediction ability of the algorithms. Each of these experiments was repeated 10 times and the prediction rate of the ML attack was calculated as the average of these 10 trials. Figure 9 shows the LR and CMA-ES attack results for different percentages of the training dataset. The y-axis represents the learning prediction rate. The maximum value on the y-axis is 100% which means that the PUF responses can be completely predicted. For ideal learning resistance, the prediction rate should be around 50%, meaning that the machine learning prediction is no better than a random guess. The maximum prediction rate of the proposed RO-PUF and RS-LPUF implementations under the LR attack for coarse PDLs are 52.5% and 51.1%. The maximum prediction rate of the proposed RO-PUF and RS-LPUF implementations under the CMA-ES attack for coarse PDLs are 53.5% and 53.1%.

The results indicate that the proposed PUFs provide good resistance (i.e., around 53% prediction rate) to the LR and CMA-ES machine learning attacks. In [59], experimental analysis of some **configurable RO (CRO)** PUFs [16, 37, 40] against ML-based attacks (i.e., LR and CMA-ES attacks) was reported. They evaluated the earlier designs with a range of CRPs. The prediction rates against earlier PUF designs under the LR attack increased from 50% to 90% as the number of training CRPs increased from 125 to 2,000 in their attack. For a small sample set of CRPs ($\approx$ 200 CRPs), they obtained prediction rates of around 55% and above 60% against the earlier improved CRO PUF designs [16, 37, 59] and the traditional CRO PUF design [40], respectively. For a similar number of CRPs (i.e., 200), the prediction rate of our proposed PUF designs is $\approx$ 53%. It is thus safe to convey that the proposed PUFs are more resistant to LR and and CMA-ES machine learning attacks with small CRPs when compared with previous CRO PUF designs [16, 37, 40, 59].

4.2 Side-Channel Attacks (SCAs)

Side-channel attacks (SCAs) statistically analyze the execution time, power consumption, or **electromagnetic (EM)** radiation of a cryptographic device to gain knowledge about integrated secrets. In 2013, Merli et al. [46] successfully attacked a RO-PUF using an EM attack that directly

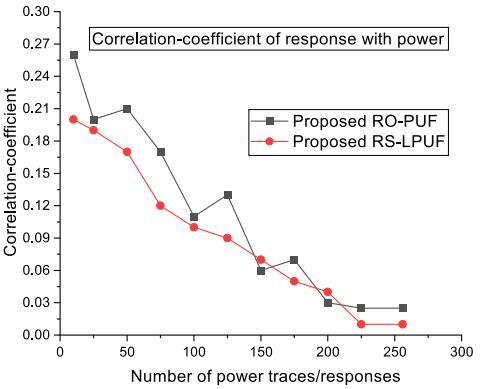

Fig. 10. Correlation coefficient with number of CRPs.

targeted the PUF and not the error correction. More recently, Tebelmann et al. [55] successfully attacked the unprotected Loop PUF by exploiting non-invasive power and EM side channels. Some of the potential countermeasures were suggested in the works presenting the attacks such as temporal masking, measurement path randomization, and interleaved placement.

In the current work, the SCA is found by correlating the dynamic power consumed by the respective PUFs and the corresponding CRPs [26]. This knowledge of correlation enables identification of the vulnerability of the proposed PUFs to power analysis attacks [2, 9, 48]. The correlation coefficient is computed by measuring the consumed power by each of the 256 CRPs for the two designed PUFs. Then Equation (7) is used to compute the correlation between the response bits (Y) and power (X). In this equation, E is the expectation, μ_X and μ_Y are the mean values of X and Y, and σ_X and σ_Y are the standard deviations of X and R, respectively:

$$\textit{Correlation-coefficient}\ (X, Y) = \frac{E[(Y - \mu_Y)(X - \mu_X)]}{\sigma_Y * \sigma_X}. \tag{7}$$

Among the 256 CRPs of the proposed RO-PUF, the maximum and minimum powers using the coarse PDLs are 6.6mW and 4.3mW, respectively. The corresponding values for the proposed RS-LPUF are 3.6mW and 2.8mW. Figure 10 depicts the achieved relationship of the correlation coefficient between power and CRPs. It has been observed that there is a sharp decline in the correlation coefficient with the increase in investigated CRPs, for both the proposed PUFs. It can be inferred that a weak correlation exists between the power and the response bits for both the PUF circuits. It is therefore safe to conclude that the proposed implementations are resilient against basic correlation power analysis attacks. Note that these experiments do not prove immunity of our design from side-channel attacks since there may be other attack schemes, such as use of Hamming weight or Hamming distance power models [9, 51] or EM-based attack [55]. Furthermore, it is well known that any PUF-based security mechanism would be vulnerable to SCA unless appropriate countermeasures are taken [38]. This article focuses on the two FPGA-based PUF designs that involve different strategies to achieve better characteristics with lower area consumption from PUFs. It is expected that the area as well as other characteristics will be severely affected when we try to make these designs SCA resistant.

5 PERFORMANCE ANALYSIS AND DISCUSSION

The number of FPGA testbeds used for performance evaluation of PUFs varies significantly in the literature. It has been shown that excellent performance is achieved for 5 to 10 [28, 52, 64], and for 10 to 50 [44, 54, 70], and even for more than 100 [27, 40, 60] testbeds. In this work, the

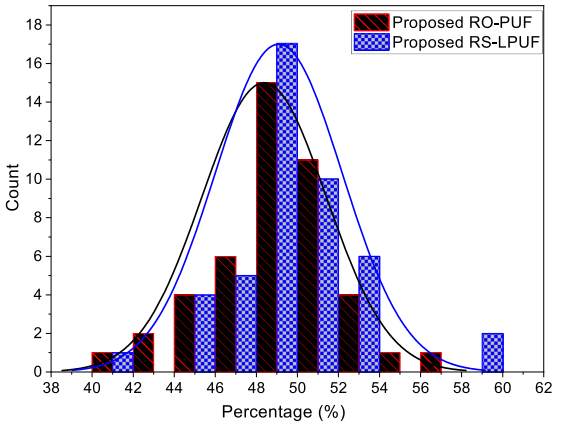

Fig. 11. Uniqueness: Inter-chip HD distribution for fine PDL. For a 256-bit response, the ideal (50%) average HD is 128 bits (μ = 50%) and the expected standard deviation is 8 bits (σ = 3.125%).

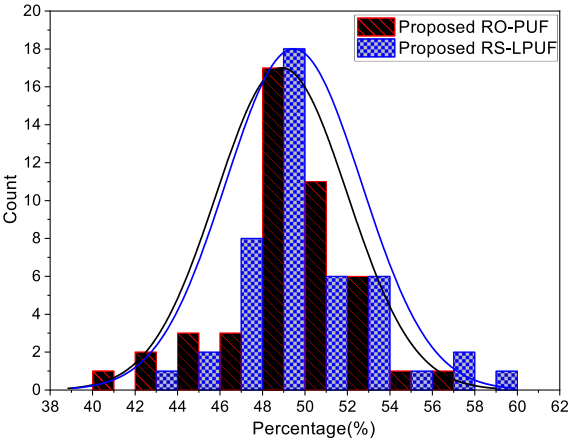

Fig. 12. Uniqueness: Inter-chip HD distribution for coarse PDL. Ideal value: μ = 50%, σ = 3.125%.

evaluation of PUF performance in terms of uniqueness, bit-aliasing, uniformity, and reliability for the two proposed PUF designs have been carried out through implementations on 10 Artix-7 FPGAs (XC7A100T).

5.1 Uniqueness

It is measured by calculating the inter-chip Hamming distance (HD) between different PUF devices using Equation (1). To investigate uniqueness in the generated response of the two proposed PUF designs, $k = 10$ (10 FPGAs) and $n = 256$ (response bit length) is used. This provides a total of 10 responses from 10 FPGAs, one response per FPGA, at standard temperature of 25°C and core supply voltage of 1.0V for each proposed PUF design. We evaluated the proposed designs using the concept of fine and coarse PDLs. It is done to figure out the effectiveness of the XORing responses while the PDL inputs are varied. The histograms of normalized inter-chip Hamming distance between two arbitrary responses among the 10 responses, i.e., $\binom{10}{2} = 45$ combinations, of the proposed designs using fine and coarse PDL concepts are shown in Figures 11 and 12, respectively. The horizontal axis represents the percentage HD and the vertical axis represents the number of occurrences of a specific HD between any two PUF responses. The histogram diagrams in Figures 11 and 12 also include the best-fit ideal binomial curves. The mean (μ) and the standard deviation (σ) of the proposed RO-PUF and RS-LPUF designs using both the fine and the coarse PDLs are given in Table 1.

95% confidence interval (CI): We then estimated the interval for the uniqueness metric for 95% CI proposed in [35]. The CI estimates an interval within which the estimated value of a population lies with some confidence. If μ and σ are the respective mean and standard deviation of the uniqueness, then the lower value for 95% confidence interval is $\mu - t_{0.025;(N-1)} \times (\frac{\sigma}{\sqrt{N}})$ and the upper value is $\mu + t_{0.025;(N-1)} \times (\frac{\sigma}{\sqrt{N}})$, where $t_{0.025;(N-1)}$ denotes the 2.5 percentile in the t-distribution with $N - 1$ degrees of freedom [35]. Here, $N = 45$; the total number of response combinations, $(\frac{\sigma}{\sqrt{N}})$, is the **standard error of the mean (SEM)**; and $t_{0.025;(N-1)} = 2.0154$. The SEM and 95% CI of μ for the proposed implementations using the fine and coarse PDLs are given in Table 1.

It is apparent from Table 1 that the designs presented in this article exhibit enhanced uniqueness. This can be attributed to the incorporation of fine and coarse PDLs, XORing responses together

Table 1. Mean (μ), Standard Deviation (σ), SEM, and 95% Confidence Interval (CI) of μ for the Two Proposed Designs' Uniqueness Metric Using the Fine and Coarse PDLs

Design	Metric	PDL	μ (%)	σ (%)	SEM (%)	95% *CI* of Mean [Lower, Upper]
Our RO-PUF	Uniqueness	Fine	48.38	3.02	0.45	[47.48, 49.29]
		Coarse	48.91	3.10	0.46	[47.97, 49.86]
Our RS-LPUF	Uniqueness	Fine	49.08	3.09	0.46	[48.15, 50.01]
		Coarse	49.47	3.20	0.48	[48.50, 50.42]

Table 2. Experimental Results of the Proposed Designs with/without TMV

Design	Metric	Only PDL (without TMV) (%)	Both PDL and TMV (Including TMV) (%)
Our RO-PUF	UQ	48.91	48.91
	UF	49.55	49.62
	BA	49.55	49.62
	RE	97.91	99.39
Our RS-LPUF	UQ	49.47	49.47
	UF	51.02	50.68
	BA	51.02	50.68
	RE	98.29	99.46

from the PDL configurations, and the placement strategy in the presented designs. In essence, the XORing and placement strategies are used to eliminate the influence of the biased responses. Moreover, it is also pertinent to mention that the incorporation of coarse PDLs in the PUFs seems better when compared to the incorporation of fine PDLs when considered for improving the uniqueness of the PUF responses. This is due to the fact that only 1-bit control is employed for fine PDLs, whereas all the configurations of LUT control input lines are used in coarse PDLs. Thus, the effect of XORing responses with varying the PDL inputs is clearly identified from the difference of uniqueness results of the designs incorporating the fine and coarse PDLs. Furthermore, another investigation to assess the impact of variation in the input of PDLs is carried out. In this case, the experiment has been repeated by varying the PDL input considering both the cases with and without the TMV. For the experiment without TMV, the 15-bit shift register and TMV modules are not included in the proposed RO-PUF and RS-LPUF designs (i.e., the raw response bits are directly stored in a 16-bit shift register in Figure 5 and Figure 8). For example, the uniqueness results of PUFs incorporating the coarse PDLs with or without TMV are tabulated in Table 2. It is apparent that the uniqueness is almost independent of TMV and therefore as a consequence it is safe to conclude that the incorporation of coarse PDLs marginally improves the uniqueness as it eliminates the bias of the response bits more effectively.

It can be seen in Table 1 that uniqueness (close to 50%) obtained from the proposed PUFs is excellent. Moreover, the spread of the 95% CI of μ is at most ±2%. Apparently, these results give a clear indication of the ability of the reported PUFs to produce highly unique responses.

Table 3. NIST Randomness Test Results

Test			Frequency	Block Frequency	Runs	Longest Run	Cumulative Sums	Approx. Entropy
p-value	Our RO-PUF	Fine	0.3425	0.1834	0.5672	0.4280	0.3578	0.2563
		Coarse	0.3412	0.5163	0.2410	0.1125	0.4745	0.5435
	Our RS-LPUF	Fine	0.4921	0.3672	0.2189	0.3581	0.5462	0.4218
		Coarse	0.5527	0.4175	0.4791	0.4363	0.2194	0.6416
Pass Rates	Our RO-PUF	Fine	9/10	10/10	9/10	9/10	9/10	9/10
		Coarse	10/10	9/10	10/10	10/10	10/10	10/10
	Our RS-LPUF	Fine	9/10	9/10	9/10	10/10	10/10	9/10
		Coarse	10/10	10/10	9/10	10/10	9/10	10/10
Result	Our RO-PUF	Fine	Pass	Pass	Pass	Pass	Pass	Pass
		Coarse	Pass	Pass	Pass	Pass	Pass	Pass
	Our RS-LPUF	Fine	Pass	Pass	Pass	Pass	Pass	Pass
		Coarse	Pass	Pass	Pass	Pass	Pass	Pass

5.2 NIST Statistical Test

The measured PUF responses are also tested using the NIST 800-22 suite [18] to evaluate randomness. Most of the statistical tests from NIST SP 800-22 require long input bit sequences. However, some tests can be adapted to test a small amount of data. These tests are frequency test, runs test, block frequency test, longest run test, cumulative sums test, and approximate entropy test. In this work, we have used the six tests (which were adopted in [44]) to check the randomness of the proposed PUFs. The selected subset of the NIST test suite requires a string with a minimum length of 128 bits. The main objective of this evaluation is to rapidly eliminate PUF responses that are not random. To generate input sequences for these six tests, we concatenated all 10 responses (i.e., from 10 FPGAs) to form one long string of 2,560 bits. The bit string is then split into 10 sequences of the 256 bits that are used as the input sequences for the NIST tool. The output of the NIST tool is the distribution of p-values and pass rates for each of the tests. The significance level on which we test the distribution of p-values is 0.1. The tool determines the minimum pass rate of these tests, and for our case the pass rate is 8/10 (because we chose our number of sequences as N = 10). The statistical results of p-values and pass rates presented in Table 3 show that the proposed PUF implementations pass all the six tests and hence the generated responses cannot be distinguished statistically from a true-random source. This impressive NIST test outcome can be attributed to the use of all the PDL configurations and XORing responses together from the PDL configurations to determine the final responses and the placement strategy in the presented designs. In essence, the XORing and placement strategies are used to eliminate the influence of the biased response. It should be noted that passing these tests does not guarantee that the PUF responses are random. Rather, it can be seen as a sanity check, showing there is no obvious problem found by the statistical results.

5.3 Entropy Estimation

It has become a standard practice to assess the unpredictability of PUF responses using Entropy. In this context, the upper bound, i.e., the best case, of the entropy is widely determined using the **context-tree weighting (CTW)** algorithm [58]. On the other hand, the min-entropy concept has become the de facto metric for the determination of the lower bound of the entropy [29, 58]. This work, therefore, also makes use of these well-known techniques.

Table 4. CTW Ratio and Min-Entropy Results

Design	Devices	PDL	Original Size (Bits)	Compressed Size (Bits)	CTW Ratio	Min-entropy per Bit
Our RO-PUF	10	Fine	2560	2512	98.13%	0.621
		Coarse	2560	2518	98.36%	0.634
Our RS-LPUF	10	Fine	2560	2531	98.86%	0.641
		Coarse	2560	2544	99.37%	0.667

5.3.1 CTW Algorithm. The CTW algorithm [58] is used to find out whether PUF responses can be compressed. If compression is possible, the PUF responses do not have full entropy. We first concatenate all 10 responses (i.e., $256 \times 10 = 2{,}560$ bits) in order to perform the CTW compression test. The proposed PUF designs in this work exhibit good compression resistance too and are almost independent of CTW as can be inferred from Table 4.

5.3.2 Min-Entropy. Besides the compression factor, it is also possible to estimate the min-entropy of PUFs. The min-entropy measures the worst-case scenario (i.e., lower bound of entropy) so as to identify the unpredictability in the random data. The commonly used method in the literature to calculate this employs the method outlined in the NIST-SP800-90B [56] specification for evaluating the min-entropy (i.e., lower bound of entropy) of a binary source. For the k devices and their n-bit responses, the bits p_1 and p_0 can be 1 and 0, respectively, and this is identified in terms of occurrence probability. For $p_{i\,max} = \max(p_1, p_0)$, Equation (8) gives the min-entropy of each bit, whereas Equation (10) gives the total min-entropy as explained in [29]:

$$H_{min,i} = -\log_2(p_{i\,max}), \tag{8}$$

where

$$p_{i\,max} = \begin{cases} \frac{HW_i}{k} & \text{if } HW_i > \frac{k}{2} \\ 1 - \frac{HW_i}{k} & \text{otherwise,} \end{cases} \tag{9}$$

where $\frac{HW_i}{k}$ is the number of ones in k devices:

$$(H_{min})_{average} = \frac{1}{n} \sum_{i=1}^{n} H_{min,i}. \tag{10}$$

We apply Equations (8) and (10) to evaluate the minimum entropy for the designed PUFs by carrying out the experiments using 10 FPGAs and list the outcomes in Table 4. It can be readily inferred that the proposed designed PUFs achieve entropy close to 0.65 by employing fine and coarse PDLs. It is imperative to mention that the use of more FPGAs will increase this entropy value [29]. In brief, it is safe to conclude that the proposed PUF implementations achieve the entropy per bit between 0.99 (based on the compression test) and 0.62 (from min-entropy). A very large-scale analysis with many boards is needed to get a meaningful entropy value of PUF designs. Even though the largest conducted experiment consisted of more than 100 boards [27, 40, 60], it is not really enough to accurately determine the entropy. In general, how to determine the exact entropy of the proposed PUF responses is another important open research problem.

5.4 Uniformity

The uniformity metric is used for the estimation of the proportion of zeros and ones in the PUF response. It is calculated using Equation (5). In [21], Sahoo et al. have theoretically shown that

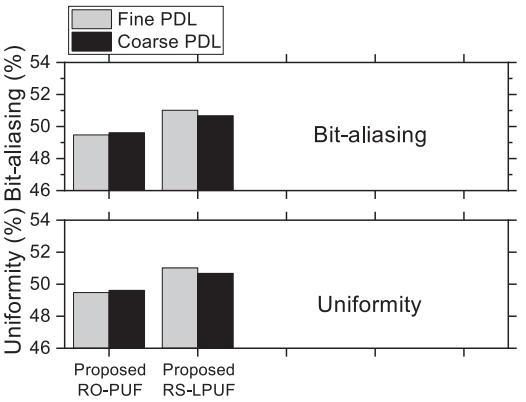

Fig. 13. Uniformity and bit-aliasing of the proposed designs using the fine and coarse PDLs.

a large bias reduces the uniformity of PDL-based Arbiter PUF responses. Further, uniformity is important for security as it prevents an attacker from guessing the response of a particular device. For PUF uniformity, the experiments have been performed on 10 Artix-7 FPGAs (i.e., one response per FPGA). The uniformity of the proposed RO-PUF and RS-LPUF implementations using the fine PDLs are 49.48% and 51.02% (see Figure 13), while the standard deviations are 0.87% and 0.59%, respectively. The corresponding uniformity by using the coarse PDLs are 49.62% and 50.68%, while the standard deviations are 0.56% and 0.47%, respectively. Moreover, the PUFs with the coarse PDLs possess better uniformity when compared to the designs utilizing fine PDLs. Once again this can be attributed to the use of all the configurations to the LUT control input lines and XORing more single golden responses to determine the final responses. This eliminates the influence of the biased responses and produces better uniformity.

5.5 Bit-Aliasing

It may give us information about any systematic spatial effect across devices. The presence of bit-aliasing leads to the situation when different chips will produce similar responses. As a consequence, it becomes relatively easier for an attacker to guess the response. In principle, any effective PUF response generator should be independent of FPGA used and should exhibit no bit-aliasing. The bit-aliasing calculated using Equation (6) is measured by estimating the bias of a particular response bit across several chips. In order to calculate the bit-aliasing, we have implemented the proposed designs on 10 ($k = 10$) Artix-7 FPGAs. It can be deduced from the results depicted in Figure 13 that the proposed designs with fine and coarse PDLs achieve close to the ideal bit-aliasing value of 50%. This excellent outcome can be attributed to the incorporation of placed macros (i.e., the placement strategy) during the design of PUFs.

5.6 Correlation between Bits

If the bits are highly correlated, then an attacker might be able to predict response to an unknown challenge from a set of known CRPs. The auto-correlation test can be used to detect correlation between bits of a response. Systematic aspects to process variation may show up as significant correlation at particular intervals. The responses are extracted from a common fabric and hence it is possible for spatial correlation to appear. The auto-correlation function AC_{xx} test is used to estimate dependency between the bits of a given PUF response, where x is the PUF response being observed, and AC_{xx} is evaluated at lag j. The values of this metric tend toward 0.5 for uncorrelated

bit-strings and toward 0 or 1 for correlated bit-strings.

$$AC_{xx}(j) = \frac{1}{n}\sum_{i=1}^{n} x_i \oplus x_{i-j}. \tag{11}$$

In this case, the least values of AC_{xx} out of the obtained 10 distinct 256-bit responses from the 10 FPGAs (one 256-bit response per FPGA) for the RO-PUF and RS-LPUF implementations with fine PDLs are 0.42 and 0.43, while the maximum values of AC_{xx} are 0.59 and 0.59. The corresponding minimum AC_{xx} with the coarse PDLs are 0.42 and 0.43, while the maximum AC_{xx} are 0.58 and 0.57. The average AC_{xx} of the proposed RO-PUF and RS-LPUF implementations using fine PDLs are 0.493% and 0.505%. The corresponding numbers utilizing coarse PDLs are 0.505% and 0.503%. It is interesting to mention that our proposed PUFs show excellent promise for substantially increasing the unpredictability of their responses considering that they achieve a correlation of almost 0.5 between the bits.

5.7 Reliability

The PUF responses are affected by external factors such as temperature variation, supply voltage fluctuation, and thermal noise, and these lead to issues in their reproducibility. Reliability measures the PUF device can perfectly reproduce its output whenever it is queried with a same challenge in different operating conditions and is calculated using Equation (3) for one chip, whereas Equation (4) is used for the calculation of the average reliability (RE) of k chips. For the assessment of reliability in the the output responses, the two proposed PUF designs with fine and coarse PDLs schemes are now implemented on 10 Artix-7 FPGAs and the core supply voltage is varied within the rated voltage of $0.95 - 1.05$V. The temperature for the evaluation of reliability is varied according to the rated values from $0°C$ to $85°C$ using a temperature-controlled chamber [4]. In this task, at the outset, the first reference response of 256-bit for each FPGA is generated at the supply voltage and temperature of 1.0V and $25°C$. Then arbitrary responses are generated at different temperature and supply voltage variations and compared with the reference and analyzed.

The results of reliability with respect to supply voltage bias at $25°C$ for one FPGA are given in Figure 14. On the other hand, Figure 15 depicts the average reliability for the responses obtained from 10 FPGAs with respect to temperature bias at 1.0V and voltage bias at $25°C$. Table 5 lists the mean reliability on voltage and temperature variations for the 10 FPGAs used in the realization of the proposed designs. Table 5 also contains the standard deviation (σ), SEM, and 95% CI of μ for the proposed designs with fine and coarse PDLs.

In addition, Table 2 shows that the proposed designs achieve significantly better reliability after incorporation of the TMV scheme. Similarly, the reliability results with fine and coarse PDLs in Table 5 are above 99% and the spread of the 95% CI is at most ±0.5%. It means that the proposed PUF designs possess reliability that is almost insensitive to the variations in temperature and supply voltage. Slight anomaly can always be corrected by employing circuitry for error correction [6].

5.8 Aging Effect

Silicon devices can be affected by various aging mechanisms including **Hot Carrier Injection (HCI)**, **Time-Dependent Dielectric Breakdown (TDDB)**, oxide breakdown, **Negative Bias Temperature Instability (NBTI)**, and EM, resulting in performance degradation and eventually design failures [1, 25, 41]. In [1], path delays in an LUT are exhaustively characterized using RO frequencies to better differentiate between brand-new and recycled FPGAs (i.e., aging). As the PUF hardware ages, the number of unreliable bits present in the responses is expected to increase. For example, in the RO-PUF, the aging process influences the frequency of the RO-PUF, which can

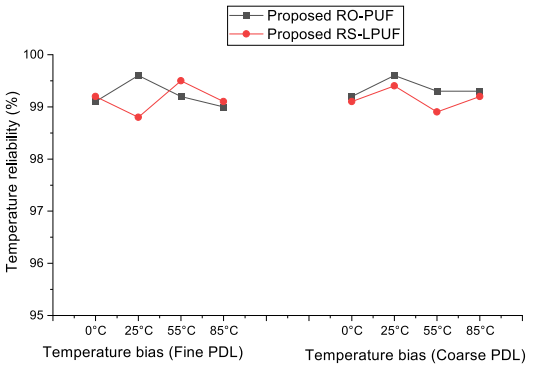

Fig. 14. Reliability (single FPGA) with respect to supply voltage bias at 25°C (Fine PDL and coarse PDL).

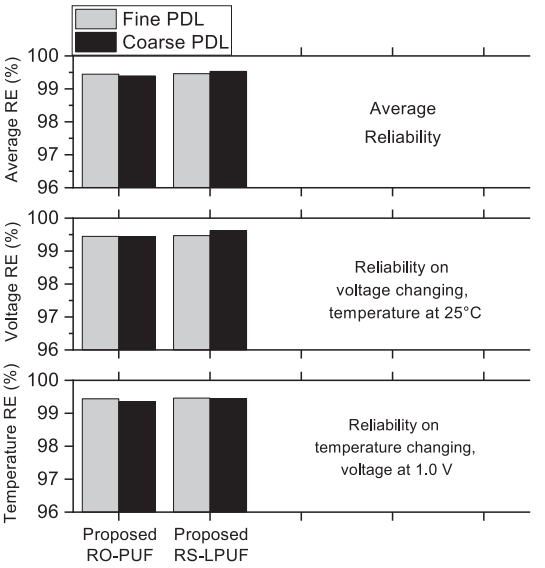

Fig. 15. Temperature Reliability (RE), Voltage Reliability, and Average Reliability of the proposed designs using the fine and coarse PDLs.

Table 5. Mean (μ), Standard Deviation (σ), SEM, and 95% Confidence Interval (CI) of μ for the Two Proposed Designs' Reliability Metric Using the Fine and Coarse PDLs

Design	Metric	PDL	μ (%)	σ (%)	SEM (%)	95% *CI* of Mean [Lower, Upper]
Our RO-PUF	Reliability	Fine	99.44	0.28	0.03	[99.36, 99.51]
		Coarse	99.39	0.38	0.05	[99.28, 99.48]
Our RS-LPUF	Reliability	Fine	99.49	0.32	0.03	[99.37, 99.61]
		Coarse	99.46	0.36	0.04	[99.39, 99.58]

affect the PUF responses [25, 41]. A possible effect that can occur due to the aging of a chip is shown in Figure 16. In the beginning, the RO_1 is faster than the RO_2, thus producing an output 1. But as the RO_1 is aging faster, it becomes slower than the RO_2 throughout the lifetime of the chip. This leads to a bit flip (i.e., output 1) and the intra-chip HD rises. The occurrence of this effect is mostly inevitable but needs to be estimated because any negative effect on the PUF-based key generation application has to be prevented. To estimate the aging effect on the proposed PUFs, first we measure the frequency of 32 ROs in our RO-PUF at the reference supply voltage and temperature of 1.0V and 25°C, respectively. These ROs oscillated at frequencies between 265 MHz and 343 MHz on the Xilinx Artix-7 in the fine PDL configurations. The mean frequency of 32 ROs was 296 MHz. Then, we measured the absolute and relative frequency of ROs [25] at the respective elevated temperature and supply voltage of 85°C and 1.05V (stressed conditions) for 7 days. During this period, we noted that the 32 ROs oscillated at frequencies between 240 MHz and 329 MHz on the Xilinx Artix-7 for the fine PDL configurations. The absolute mean frequency value over 32 tested ROs for the fine PDL configurations was observed to be 282 MHz. We noted that the maximum absolute mean frequency difference was only 12 MHz. Furthermore, the maximum and minimum relative

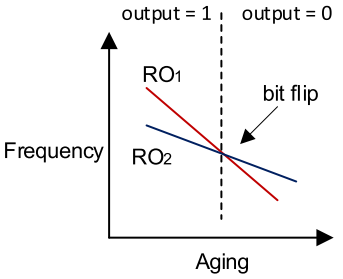

Fig. 16. Bit flip due to the aging of two ROs.

Table 6. Aging Impact Comparison with Previous FPGA-based PUFs

Design	PDL	Measurement Condition	Aging Duration	Res. Bit Length	Bit Error Rate (BER) (%)	Target FPGA
Our RO-PUF	Fine	85°C, 1.05V	10 years	256	2.5	Artix-7
	Coarse	85°C, 1.05V	10 years	256	2.9	
Our RS-LPUF	Fine	85°C, 1.05V	10 years	256	2.7	
	Coarse	85°C, 1.05V	10 years	256	2.2	
RO-PUF	Chowdhury et al. [13]	100°C, 1.45V	10 years	64	2.2	Spartan-3A
	Maiti and Schaumont [41]	80°C, 1.8V	10 years	511	1	Spartan 3E
	Zhang et al. [71]	15°C, 1.0V	30 days	64	6	Zynq-7000
	Gehrer et al. [25]	80°C, 1.0V	31 years	2664	4	Zynq-7000

frequency degradation over 32 tested ROs with fine PDLs were 0.29% and 0.18%. It has to be noted that our RO-PUF showed only small absolute and relative frequency change.

Moreover, Maiti and Schaumont [41] present the effects of simulated aging on PUF hardware by purposefully stressing the devices (i.e., accelerated testing) beyond normal operating conditions. By varying both voltage and temperature, they were able to show a drift in the intra-PUF variation over time that will lead to false negatives. In this work, we have also carried out the accelerated testing [41] for the proposed PUF designs. It is done at the respective elevated temperature and supply voltage of 85°C and 1.05V (stressing the devices) and measuring the PUF output at 4 hours' duration over a period of 4 weeks. This gives enough information to assess the aging effect for 10 years [41]. Subsequently, these outcomes at the elevated supply voltage and temperatures are compared with the responses at the reference values of 1.0V and 25°C, respectively. The BERs, computed using Intra-chip HD (2), of the proposed RO-PUF and RS-LPUF designs by employing fine PDLs are 2.5% and 2.7%, respectively. The corresponding numbers with coarse PDLs are 2.9% and 2.2%. As can be seen in Table 6, the proposed RO-PUF and RS-LPUF designs achieve lower BER when compared to existing RO-PUF designs [25, 71] but slightly higher BER than other RO-PUF designs [13, 41]. However, our results show that the proposed RO-PUF and RS-LPUF implementations are still usable as the BER is below 3%, and this can be corrected using error-correcting codes [6].

5.9 Performance Comparisons with Previous PUFs

The evaluation metric of PUF structures may vary in different application scenarios [11]. For example, the metrics of uniqueness, reliability, and uniformity may have different importance in different PUF usages such as identification, authentication, or encryption as explained ahead.

- Identification: It is well accepted that PUFs can be relied upon for the generation of unique serial numbers for identification and tracking of parts at the time of manufacturing.

Table 7. Performance Comparisons with Previous FPGA-based Weak PUF Designs (Experimental Results)

Design		Uniqueness	Reliability	Uniformity	Bit-alias	Target FPGA	Comment of Each Architecture
	Ideal Value	50%	100%	50%	50%		
RO-PUF	Zhang et al. [70]	49.33	95.45	49.5	—	Virtex-5	Environmental sensitivity
	Marchand et al. [44]	55.00	94.50	—	—	Spartan-6	
	Cui et al. [16]	49.97	98.41	—	—	Spartan-6	Spatial correlation exists
	Cui et al. [15]	48.30	95.27	—	—	Virtex-II	
	Previous work [3]	47.13	99.16	50.61	—	Spartan-6	
	Chauhan et al. [10]	49.83	99.35	—	—	Artix-7	
	Gu et al. [27]	48.05	99.30	50.19	50.19	Artix-7	
	Yin and Qu [68]	—	99	—	—	Virtex-4	Design requires more slices
	Choudhury et al. [12]	47.40	—	49.2	49.1	Artix-7	
	Yan et al. [66]	—	99.33	50.05	—	Kintex-7	
Our RO-PUF (Section 3.1)	Fine PDL	48.38	99.44	49.48	49.48	Artix-7 (XC7A100)	Area-efficient design with good statistical properties
	Coarse PDL	48.91	99.39	49.62	49.62		
SRAM PUF	Guajardo et al. [30]	49.97	88.00	—	—	—	Vulnerable to SCA
PicoPUF	Gu et al. [28]	45.60	98.74	—	—	Artix-7	Degrade the uniqueness
Butterfly PUF	Kumar et al. [36]	43.16	96.20	—	—	Virtex-5	
RS-LPUF	Previous work [3]	48.10	99.19	50.20	—	Spartan-6	
	Ardakani et al. [7]	49.32	98.80	44.65	44.65	Spartan-3	Large design bias
	Yamamoto et al. [65]	49.00	96.34	—	—	Spartan-6	Environmental sensitivity
	Habib et al. [32]	49.24	98.87	—	—	Spartan-6	
	Stanciu et al. [53]	34.73	92.00	—	—	Spartan-6	Poor uniqueness and reliability
Our RS-LPUF (Section 3.2)	Fine PDL	49.08	99.49	51.02	51.02	Artix-7 (XC7A100)	Area-efficient design with good statistical properties
	Coarse PDL	49.47	99.46	50.68	50.68		

Uniqueness is the most important metric in this situation. Further, reliability is not a major concern in the scenario of identification as long as the BER is relatively small. A large BER may lead to unacceptably high probability of bit-aliasing, and this has a detrimental effect as it significantly reduces the ability to generate unique IDs.

- Authentication: It is also known that PUFs are widely employed to securely identify the chip in which they are embedded. In this scenario, randomness and reliability are the most critical metrics. In addition, the PUF should also have good uniqueness properties. Moreover, a very large CRP space is necessary to prevent attackers from reading all the responses and creating a clone. The large CRP space also prevents ML attacks against the SCA adversaries.
- Encryption: For asymmetric encryption, the PUFs can easily create random nonce. On the other hand, PUFs have the ability to create keys for symmetric encryption algorithms. The randomness, reliability, and uniqueness are critical in such applications. However, a large CRP space is not necessary in cases where only a small number of key generations are required over the entire chip life. In this case, BER must be zero, which may require error correction.

The obtained uniqueness, bit-aliasing, uniformity, and reliability for the proposed architectures, implemented on Xilinx Artix-7 FPGAs, are compared with a number of earlier reported designs in Table 7. We summarize the primary observations below:

- The proposed RO-PUF design with fine and coarse PDLs outperforms existing RO-PUF designs [3, 12, 15, 27, 44] in terms of uniqueness. However, the obtained uniqueness is slightly inferior when compared to other RO-PUF designs [10, 16, 70]. Moreover, the proposed RO-PUF design using fine and coarse PDLs outperforms all the previous RO-PUF designs in

terms of reliability. Additionally, the proposed RO-PUF design using coarse PDL improves the bit-aliasing and uniformity metrics when compared to other similar designs [12, 70].

- The results also convey that the proposed RS-LPUF design using coarse PDLs exhibits better uniqueness when compared to existing RS-LPUF designs [3, 7, 32, 53, 65], Butterfly PUF [36], and PicoPUF [28] but performs slightly more poorly when compared to SRAM PUF [30]. The design with fine PDLs leads to enhanced uniqueness when compared to PUF ID [28], RS-LPUF design [53, 65], and Butterfly PUF [36] but poorly compares to SRAM PUF [30] and the existing RS-LPUF designs [7, 32]. Moreover, the proposed RS-LPUF design improves the bit-aliasing and uniformity metrics when compared to other RS-LPUF designs [7]. Furthermore, the proposed RS-LPUF design using fine and coarse PDLs outperforms all the previous designs of RS-LPUF in terms of reliability.

In summary, the experimental results clearly show that the proposed PUF architectures achieve superior uniformity, uniqueness, bit-aliasing, and reliability. Therefore, our proposed PUF designs can be used for chip identification and encryption applications [11]. However, using a PUF itself may not be suitable for the chip authentication scenario because our two proposed PUF designs are not having very large CRP space. On the other hand, chip authentication can be achieved by using a cryptographic function and a secret key that can be generated from the proposed PUF designs (having a limited number of CRPs).

6 HARDWARE OVERHEAD ANALYSIS

The Xilinx Artix-7 FPGAs are used for the realization of the proposed PUFs and the achieved metrics are compared with the existing state-of-the-art specially developed on Xilinx FPGAs in Table 8. In this table, the Virtex-5, Spartan-6, and Zynq-7000 FPGAs also have four LUTs, eight registers, and three MUXes in each slice, the same as the Artix-7 FPGA used in this work. As can be seen in Table 8, the proposed RO-PUF and RS-LPUF designs are area efficient when compared to the existing RO-PUF [31, 38, 70], RS-LPUF [32, 65], PUF ID [28], and Butterfly PUF [36]. Compared to our previous work [3], the area requirements for the proposed PUF designs are slightly more but achieve better performance in terms of uniqueness and reliability (see Table 7). Furthermore, the proposed PUF implementations **without TMV (WOTMV)** outperform the previous conventional RO-PUF and RS-LPUF in terms of processing time (throughput). However, the proposed implementations **with TMV (WTMV)** have poor throughput compared with the conventional RO-PUF [31, 38, 70] but achieve better throughput than the RS-LPUF [32]. It is noted that the timing overhead of WOTMV in this work is around 15 times more when compared to the time delay of WTMV. However, TMV on the native PUF bits prior to error correction circuitry that can minimize the area overhead of the error correction circuitry achieves significantly superior performance as compared to the WOTMV [45] (i.e., WTMV achieves significantly better reliability when compared to the WOTMV, as outlined in Table 2). Moreover, our two proposed PUF implementations outperform all the previous works in terms of the resources consumed per response bit (hardware efficiency).

We also report the estimated power usage of the proposed PUF designs. The Xilinx XPower Analyzer tool is used to analyze the power estimate, which includes static power and dynamic power overheads. Compared to an earlier design [70], our designs consumed more energy but consumed less power. Additionally, it's worth mentioning that the PUF will only be used when there is a need to generate a secret response in security applications. Therefore, the energy overhead (in nanojoules) can be negligible because the PUF unit can be powered off to save power/energy when it doesn't need to be used anymore.

Table 8. FPGA Implementation Results (after Place and Route) of Proposed PUFs and Comparisons

PUF Design		Number of Cells	Area		Process Time (msec)	Res. Bit Length	CLBs/ Res. Bit	Total Power (mW)	Energy (Power × Time) (nJ)	Energy/ Res. Bit (nJ)	Target FPGA Device
			CLBs†	Slices†							
RO-PUF	Zhang et al. [70]	512 ROs	93	186	0.005	136	0.68	1050	5.25	0.03	Virtex-5
	Chauhan et al. [10]	32 ROs	—	—	—	255	—	—	—	—	Artix-7
	Yan et al. [66]	128 ROs	323.5	647	—	128	—	—	—	—	
	Günlü et al. [31]	256 ROs	424.5	849	1.68	1275	0.33				Zynq-7000
	Maes et al. [38]	1024 ROs	476	952	4.59	2226	0.22	—	—	—	Spartan-6
	Marchand et al. [44]	128 ROs	—	—	—	128	—	—	—	—	
	Previous work [3]	32 ROs	41	82	—	256	0.16	—	—	—	
Our RO-PUF (Section 3.1)	WOTMV	32 ROs	46.5	93	1.05*	256	0.18	16.7	17.53	0.069	Artix-7 (XC7A100)
	WTMV	32 ROs	53.5	107	15.83‡	256	0.21	19.3	305.5	1.19	
Butterfly PUF	Kumar et al. [36]	64 RS latches	65	130	—	50	1.3	—	—	—	Virtex-5
PUF ID	Gu et al. [28]	128 IDs	64	128	—	128	0.5	—	—	—	Spartan-6
RS-LPUF	Stanciu et al. [53]	77 RS latches	—		—	128	—	—	—	—	
	Habib et al. [32]	512 RS latches	162	324	5120	256	0.63	—	—	—	
	Yamamoto et al. [65]	128 RS latches	128	256	—	256	0.5	—	—	—	
	Previous work [3]	32 RS latches	38	76	—	256	0.15	—	—	—	
Our RS-LPUF (Section 3.2)	WOTMV	32 RS latches	42	84	1.05	256	0.16	10.6	11.13	0.043	Artix-7 (XC7A100)
	WTMV	32 RS latches	50.5	101	15.83	256	0.20	13.8	218.4	0.85	

†The number of CLBs/slices for the PUF implementations with control logic (without UART).
*Without BRAM.
‡Clock cycles required to generate a response with TMV = Number of sub challenges × (Ref. counter counts × (delay eval. × TMV)) + control logic.

7 CONCLUSIONS

This article reported implementations of RO-PUF and RS-PUF on FPGA. The reported designs exhibit significant performance enhancement. It has been shown statistically that the incorporation of fine and coarse PDLs of FPGA LUTs has the potential to substantially enhance the randomness and uniqueness in the RO-PUF and RS-PUF responses. Furthermore, the TMV scheme demonstrates the effectiveness by improving the reliability to a great extent. The proposed designs achieve better hardware efficiency compared to the existing RO-PUF and RS-PUF designs. The reported designs are also insensitive to variations in supply voltage, temperature, aging effects, and process. Overall, the interesting exciting features in the proposed designs potentially make this a potential candidate for lightweight security applications such as secure key generation or for seeding a PRNG.

REFERENCES

[1] Md Mahbub Alam, Mark Tehranipoor, and Domenic Forte. 2016. Recycled FPGA detection using exhaustive LUT path delay characterization. In *2016 IEEE International Test Conference (ITC'16)*. IEEE, 1–10.

[2] N. Nalla Anandakumar, M. Prem Laxman Das, Somitra K. Sanadhya, and Mohammad S. Hashmi. 2018. Reconfigurable hardware architecture for authenticated key agreement protocol over binary edwards curve. *ACM Transactions on Reconfigurable Technology and Systems* 11, 2, Article 12 (Nov. 2018), 19 pages.

[3] N. Nalla Anandakumar, Mohammad S. Hashmi, and Somitra Kumar Sanadhya. 2017. Compact implementations of FPGA-based PUFs with enhanced performance. In *2017 30th International Conference on VLSI Design and 2017 16th International Conference on Embedded Systems (VLSID'17)*. IEEE, 161–166.

[4] N. Nalla Anandakumar, Mohammad S. Hashmi, and Somitra Kumar Sanadhya. 2020. Efficient and lightweight FPGA-based hybrid PUFs with improved performance. *Microprocessors and Microsystems* 77 (2020), 103180.

[5] N. Nalla Anandakumar, Mohammad S. Hashmi, and Somitra Kumar Sanadhya. 2022. Field programmable gate array based elliptic curve Menezes-Qu-Vanstone key agreement protocol realization using physical unclonable function and true random number generator primitives. *IET Circuits, Devices & Systems* 16 (2022), 1–17.

[6] N. Nalla Anandakumar, Mohammad S. Hashmi, and Mark Tehranipoor. 2021. FPGA-based physical unclonable functions: A comprehensive overview of theory and architectures. *Integration* 81 (2021), 175–194.

[7] Amir Ardakani, Shahriar B. Shokouhi, and Arash Reyhani-Masoleh. 2018. Improving performance of FPGA-based SR-latch PUF using transient effect ring oscillator and programmable delay lines. *Integration* 62 (2018), 371–381.

[8] Frederik Armknecht, Roel Maes, Ahmad-Reza Sadeghi, Berk Sunar, and Pim Tuyls. 2009. Memory leakage-resilient encryption based on physically unclonable functions. In *Advances in Cryptology (ASIACRYPT'09)*. 685–702.

[9] Georg T. Becker and Raghavan Kumar. 2014. Active and passive side-channel attacks on delay based PUF designs. *IACR Cryptology ePrint Archive* 2014 (2014), 287.

[10] A. S. Chauhan, V. Sahula, and A. S. Mandal. 2019. Novel randomized biased placement for FPGA based robust random number generator with enhanced uniqueness. In *IEEE 32nd International Conference on VLSI Design (VLSID'19)*. 353–358.

[11] W. Che, M. Martinez-Ramon, F. Saqib, and J. Plusquellic. 2018. Delay model and machine learning exploration of a hardware-embedded delay PUF. In *IEEE International Symposium on Hardware Oriented Security and Trust (HOST'18)*. 153–158.

[12] M. Choudhury, N. Pundir, M. Niamat, and M. Mustapa. 2017. Analysis of a novel stage configurable ROPUF design. In *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS'17)*. IEEE, 942–945.

[13] S. Chowdhury, X. Xu, M. Tehranipoor, and D. Forte. 2017. Aging resilient RO PUF with increased reliability in FPGA. In *2017 International Conference on ReConFigurable Computing and FPGAs (ReConFig'17)*. 1–7.

[14] Yijun Cui, Yunpeng Chen, Chenghua Wang, Chongyan Gu, Máire O'Neill, and Weiqiang Liu. 2020. Programmable ring oscillator PUF based on switch matrix. In *2020 IEEE International Symposium on Circuits and Systems (ISCAS'20)*. 1–4.

[15] Yijun Cui, Chongyan Gu, Chenghua Wang, Máire O'Neill, and Weiqiang Liu. 2018. Ultra-lightweight and reconfigurable tristate inverter based physical unclonable function design. *IEEE Access* 6 (2018), 28478–28487.

[16] Y. Cui, C. Wang, W. Liu, Y. Yu, M. O'Neill, and F. Lombardi. 2016. Low-cost configurable ring oscillator PUF with improved uniqueness. In *2016 IEEE International Symposium on Circuits and Systems (ISCAS'16)*. 558–561.

[17] Jeroen Delvaux, Roel Peeters, Dawu Gu, and Ingrid Verbauwhede. 2015. A survey on lightweight entity authentication with strong PUFs. *ACM Computing Surveys* 48, 2, Article 26 (Oct. 2015), 42 pages.

[18] Bassham et al. 2010. *Sp 800-22 rev. 1a. a statistical test suite for random and pseudorandom number generators for cryptographic applications*. National Institute of Standards & Technology.

[19] Pedregosa et al. 2011. Scikit-Learn: Machine learning in python. *Journal of Machine Learning Research* 12 (Nov. 2011), 2825–2830.

[20] Ram et al. 2021. Eternal-Thing: A secure aging-aware solar-energy harvester thing for sustainable IoT. *IEEE Transactions on Sustainable Computing* 6, 2 (2021), 320–333.

[21] Sahoo et al. 2016. On the Architectural Analysis of Arbiter Delay PUF Variants. Cryptology ePrint Archive, Report 2016/057.

[22] Fpga4fun. 2008. LFSR counters: LFSR reaches all possible states. https://www.fpga4fun.com/Counters3.html.

[23] Steve Gabriel. 2015. Altera Partners with Intrinsic-ID to Develop World's Most Secure High-End FPGA. https://newsroom.intel.com/news-releases/altera-partners-intrinsic-id-develop-worlds-secure-high-end-fpga.

[24] Fatemeh Ganji, Shahin Tajik, Fabian Fäßler, and Jean-Pierre Seifert. 2017. Having no mathematical model may not secure PUFs. *Journal of Cryptographic Engineering* 7, 2 (June 2017), 113–128.

[25] S. Gehrer, S. Leger, and G. Sigl. 2015. Aging effects on ring-oscillator-based physical unclonable functions on FPGAs. In *2015 International Conference on ReConFigurable Computing and FPGAs (ReConFig'15)*. IEEE, 1–6.

[26] R. Govindaraj, S. Ghosh, and S. Katkoori. 2018. Design, analysis and application of embedded resistive RAM based strong arbiter PUF. *IEEE Transactions on Dependable and Secure Computing* 17, 6 (2018), 1232–1242. DOI: 10.1109/TDSC.2018.2866425

[27] Chongyan Gu, Chip-Hong Chang, Weiqiang Liu, Neil Hanley, Jack Miskelly, and Máire O'Neill. 2021. A large-scale comprehensive evaluation of single-slice ring oscillator and PicoPUF bit cells on 28-nm Xilinx FPGAs. *Journal of Cryptographic Engineering* 11, 3 (2021), 227–238.

[28] Chongyan Gu, Neil Hanley, and Máire O'Neill. 2017. Improved reliability of FPGA-based PUF identification generator design. *ACM Transactions on Reconfigurable Technology and Systems* 10, 3, Article 20 (2017), 23 pages.

[29] C. Gu, W. Liu, N. Hanley, R. Hesselbarth, and M. O'Neill. 2019. A theoretical model to link uniqueness and min-entropy for PUF evaluations. *IEEE Transactions on Computers* 68, 2 (2019), 287–293.

[30] Jorge Guajardo, Sandeep S. Kumar, Geert-Jan Schrijen, and Pim Tuyls. 2007. FPGA intrinsic PUFs and their use for IP protection. In *Cryptographic Hardware and Embedded Systems (CHES'07)*, Vol. 4727. Springer, 63–80.

[31] Onur Günlü, Tasnad Kernetzky, Onurcan Iscan, Vladimir Sidorenko, Gerhard Kramer, and Rafael F. Schaefer. 2018. Secure and reliable key agreement with physical unclonable functions. *Entropy* 20, 5 (2018), 340.

[32] Bilal Habib, Jens-Peter Kaps, and Kris Gaj. 2015. Efficient SR-latch PUF. In *Proceedings of the Applied Reconfigurable Computing - 11th International Symposium (ARC'15)*, Vol. 9040. Springer, 205–216.

[33] Nikolaus Hansen. 2006. *The CMA Evolution Strategy: A Comparing Review*. Springer, Berlin, 75–102. https://github.com/CMA-ES/pycma.

[34] Andreas Herkle, Holger Mandry, Joachim Becker, and Maurits Ortmanns. 2019. In-depth analysis and enhancements of RO-PUFs with a partial reconfiguration framework on xilinx Zynq-7000 SoC FPGAs. In *2019 IEEE International Symposium on Hardware Oriented Security and Trust (HOST'19)*. IEEE, 238–247.

[35] Y. Hori, T. Yoshida, T. Katashita, and A. Satoh. 2010. Quantitative and statistical performance evaluation of arbiter physical unclonable functions on FPGAs. In *International Conference on Reconfigurable Computing and FPGAs*. 298–303.

[36] S. S. Kumar, J. Guajardo, R. Maes, G. J. Schrijen, and P. Tuyls. 2008. Extended abstract: The butterfly PUF protecting IP on every FPGA. In *IEEE International Workshop on Hardware-Oriented Security and Trust*. IEEE, 67–70.

[37] Weiqiang Liu, Lei Zhang, Zhengran Zhang, Chongyan Gu, Chenghua Wang, Maire O'Neill, and Fabrizio Lombardi. 2019. XOR-based low-cost reconfigurable PUFs for IoT security. *ACM Transactions on Embedded Computing Systems* 18, 3, Article 25 (April 2019), 21 pages.

[38] Roel Maes, Anthony Van Herrewege, and Ingrid Verbauwhede. CHES 2012. *PUFKY: A Fully Functional PUF-Based Cryptographic Key Generator*. Springer, Berlin, 302–319.

[39] Abhranil Maiti, Vikash Gunreddy, and Patrick Schaumont. 2011. A systematic method to evaluate and compare the performance of physical unclonable functions. *IACR Cryptology ePrint Archive* (2011). http://eprint.iacr.org/2011/657.

[40] Abhranil Maiti and Patrick Schaumont. 2011. Improved ring oscillator PUF: An FPGA-friendly secure primitive. *Journal of Cryptology* 24 (2011), 375–397.

[41] A. Maiti and P. Schaumont. 2014. The impact of aging on a physical unclonable function. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 22, 9 (Sept. 2014), 1854–1864.

[42] Mehrdad Majzoobi, Farinaz Koushanfar, and Srinivas Devadas. 2010. FPGA PUF using programmable delay lines. In *IEEE International Workshop on Information Forensics and Security (WIFS'10)*. IEEE, 1–6.

[43] Mehrdad Majzoobi, Farinaz Koushanfar, and Srinivas Devadas. 2011. FPGA-based true random number generation using circuit metastability with adaptive feedback control. In *Cryptographic Hardware and Embedded Systems (CHES'11)*, Vol. 6917. Springer, 17–32.

[44] C. Marchand, L. Bossuet, U. Mureddu, N. Bochard, A. Cherkaoui, and V. Fischer. 2018. 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* 37, 1 (Jan. 2018), 97–109.

[45] S. K. Mathew, S. K. Satpathy, M. A. Anders, H. Kaul, S. K. Hsu, A. Agarwal, G. K. Chen, R. J. Parker, R. K. Krishnamurthy, and V. De. 2014. 16.2 A 0.19pJ/b PVT-variation-tolerant hybrid physically unclonable function circuit for 100% stable secure key generation in 22nm CMOS. In *IEEE International Solid-State Circuits Conference (ISSCC'14)*. IEEE, 278–279.

[46] D. Merli, J. Heyszl, B. Heinz, D. Schuster, F. Stumpf, and G. Sigl. 2013. Localized electromagnetic analysis of RO PUFs. In *IEEE International Symposium on Hardware-Oriented Security and Trust (HOST'13)*. IEEE, 19–24.

[47] A. Mosenia and N. K. Jha. 2017. A comprehensive study of security of internet-of-things. *IEEE Transactions on Emerging Topics in Computing* 5, 4 (Oct. 2017), 586–602.

[48] N. Nalla Anandakumar. 2015. SCA resistance analysis on FPGA implementations of sponge based MAC PHOTON. In *Innovative Security Solutions for Information Technology and Communications*. Springer, Cham, 69–86.

[49] Graham Prophet. 2016. Xilinx to Add PUF Security to Zynq Devices. https://www.eenewseurope.com/news/xilinx-add-puf-security-zynq-devices-0.

[50] Ulrich Rührmair, Frank Sehnke, Jan Sölter, Gideon Dror, Srinivas Devadas, and Jürgen Schmidhuber. 2010. Modeling attacks on physical unclonable functions. In *Proceedings of the 17th ACM Conference on Computer and Communications Security (CCS'10)*. ACM, 237–249.

[51] S. Zeitouni, Y. Oren, C. Wachsmann, P. Koeberl, and A-Raza Sadeghi. 2015. Remanence decay side-channel: The PUF case. *IEEE Transactions on Information Forensics and Security* 11, 6 (2015), 1106–1116.

[52] D. P. Sahoo, R. S. Chakraborty, and D. Mukhopadhyay. 2015. Towards ideal arbiter PUF design on xilinx FPGA: A practitioner's perspective. In *Euromicro Conference on Digital System Design*. IEEE, 559–562.

[53] A. Stanciu, M. N. Cirstea, and F. D. Moldoveanu. 2016. Analysis and evaluation of PUF-based SoC designs for security applications. *IEEE Transactions on Industrial Electronics* 63, 9 (Sept. 2016), 5699–5708.

[54] G. Edward Suh and Srinivas Devadas. 2007. Physical unclonable functions for device authentication and secret key generation. In *Proceedings of the 44th Design Automation Conference (DAC'07)*. IEEE, 9–14.

[55] Lars Tebelmann, Jean-Luc Danger, and Michael Pehl. 2021. Self-secured PUF: Protecting the Loop PUF by masking. In *Constructive Side-Channel Analysis and Secure Design*. 293–314.

[56] Meltem Sönmez Turan, Elaine Barker, John Kelsey, Kerry A. McKay, Mary L. Baish, and Mike Boyle. 2018. Recommendation for the entropy sources used for random bit generation. *NIST Special Publication* 800, 90B (2018). https://www.nist.gov/publications/recommendation-entropy-sources-used-random-bit-generation.

[57] M. A. Usmani, S. Keshavarz, E. Matthews, L. Shannon, R. Tessier, and D. E. Holcomb. 2019. Efficient PUF-based key generation in FPGAs using per-device configuration. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 27, 2 (Feb. 2019), 364–375.

[58] Vincent van der Leest, Geert-Jan Schrijen, Helena Handschuh, and Pim Tuyls. 2010. Hardware intrinsic security from D Flip-flops. In *Proceedings of the 5th ACM Workshop on Scalable Trusted Computing*. ACM, 53–62.

[59] Z. Wei, Y. Cui, Y. Chen, C. Wang, C. Gu, and W. Liu. 2020. Transformer PUF : A highly flexible configurable RO PUF based on FPGA. In *2020 IEEE Workshop on Signal Processing Systems (SiPS'20)*. IEEE, 1–6.

[60] A. Wild, G. T. Becker, and T. Güneysu. 2017. A fair and comprehensive large-scale analysis of oscillation-based PUFs for FPGAs. In *27th International Conference on Field Programmable Logic and Applications (FPL'17)*. IEEE, 1–7.

[61] Nils Wisiol, Christoph Graebnitz, Marian Margraf, Manuel Oswald, Tudor A. A. Soroceanu, and Benjamin Zengin. 2017. Why attackers lose: Design and security analysis of arbitrarily large XOR arbiter PUFs. In *PROOFS@CHES 2017 (EPiC Series in Computing)*, Vol. 49. Springer, 68–83.

[62] Xilinx. 2013. Xilinx 7 Series FPGA and Zynq-7000 All Programmable SoC Libraries Guide for HDL Designs. https://www.xilinx.com/support/documentation/sw_manuals/xilinx14_7/7series_hdl.pdf.

[63] Xilinx. 2018. Vivado Design Suite User Guide Using Constraints. https://www.xilinx.com/support/documentation/sw_manuals/xilinx2018_3/ug903-vivado-using-constraints.pdf.

[64] Xiaolin Xu, Ulrich Rührmair, Daniel E. Holcomb, and Wayne Burleson. 2015. Security evaluation and enhancement of bistable ring PUFs. In *Radio Frequency Identification*. Springer, 3–16.

[65] Dai Yamamoto, Kazuo Sakiyama, Mitsugu Iwamoto, Kazuo Ohta, Masahiko Takenaka, and Kouichi Itoh. 2013. Variety enhancement of PUF responses using the locations of random outputting RS latches. *Journal of Cryptographic Engineering* 3, 4 (Nov. 2013), 197–211.

[66] W. Yan, C. Jin, F. Tehranipoor, and J. A. Chandy. 2017. Phase calibrated ring oscillator PUF design and implementation on FPGAs. In *27th International Conference on Field Programmable Logic and Applications (FPL'17)*. IEEE, 1–8.

[67] K. Yang, D. Forte, and M. M. Tehranipoor. 2017. CDTA: A comprehensive solution for counterfeit detection, traceability, and authentication in the IoT supply chain. *ACM Transactions on Design Automation and Electronic Systems* 22, 3, Article 42 (April 2017), 31 pages.

[68] Chi-En Daniel Yin and Gang Qu. 2010. LISA: Maximizing RO PUF's secret extraction. In *2010 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST'10)*. IEEE, 100–105.

[69] J. Zhang and C. Shen. 2021. Set-based obfuscation for strong PUFs against machine learning attacks. *IEEE Transactions on Circuits and Systems I: Regular Papers* 68, 1 (2021), 288–300. DOI: 10.1109/TCSI.2020.3028508

[70] J. Zhang, X. Tan, Y. Zhang, W. Wang, and Z. Qin. 2018. Frequency offset-based ring oscillator physical unclonable function. *IEEE Transactions on Multi-Scale Computing Systems* 4 (2018), 711–721.

[71] J. Zhang, Q. Wu, Y. Ding, Y-Qiang Lv, Q. Zhou, Z. Xia, X. Sun, and X-Wei Wang. 2016. Techniques for design and implementation of an FPGA-specific physical unclonable function. *Journal of Computer Sciences and Technology* 31, 1 (2016), 124–136.

[72] C. Zhou, K. K. Parhi, and C. H. Kim. 2017. Secure and reliable XOR arbiter PUF design: An experimental study based on 1 trillion challenge response pair measurements. In *2017 54th ACM/IEEE Design Automation Conference (DAC'17)*. 1–6.

Received September 2021; revised December 2021; accepted February 2022