Journal of Electronic Testing (2022) 38:653–666
https://doi.org/10.1007/s10836-022-06034-7

Design and Evaluation of XOR Arbiter Physical Unclonable Function and its Implementation on FPGA in Hardware Security Applications

R. Naveenkumar[1] · N. M. Sivamangai[2] · A. Napolean[1] · S. Sridevi Sathya Priya[2]

Received: 31 May 2022 / Accepted: 3 November 2022 / Published online: 17 December 2022

Abstract
Hardware security has become most prevalent challenging concept of improving the Internet of Things (IoT) in human routine as well as in future engineering processes. IoT systems face a wide range of problems, including a dearth of resources, a requirement for equipment protection from cyber-attacks, and lower power consumption. Especially, the methods are constrained by power consumption and an insufficient of computing capacity. Moreover, the customary method of keeping secret keys in non-volatile memory is susceptible to assaults like side-channelling and reverse engineering. Physical Unclonable Functions (PUFs) are a technique for improving security of physical device and resolving difficulties with current cryptographic algorithms. PUFs are simple operations that force each terminal to have a unique personality based on physical characteristics imposed during production that are unpredictable and impossible to replicate. The focus of this work is on XOR arbiter PUF (XORAPUF) architecture with the three factors: reliability, uniqueness, and uniformity. Experiments show that the proposed XORAPUF implemented on field programmable gate array (FPGA) achieves inter-chip hamming distance (HD) closer to 50% with excellent uniqueness and uniformity of 49.88% and 48.74%, respectively. The reliability of the designed PUF is also optimized to 99.20%. On comparing the designed PUF metrics results with conventional PUF, the XORAPUF circuit generated better results.

Keywords Hardware security · Internet of things (IoT) · Physical Unclonable Functions (PUFs) · XOR arbiter PUF (XORAPUF) · Field programmable gate array (FPGA)

1 Introduction

The Internet of Things (IoT) is a vast network of networked objects with embedded sensors, software, and electrical components. These devices can acquire information, communicate it, make assessments, and possibly even make decisions without human involvement thanks to their internet connections [1]. Digital towns, digital homes, cutting-edge systems for driver assistance, chemicals, defence, and agriculture are just a few of the businesses using IoT [2]. It suffered from hardware security issues like hardware trojan [3] and authentication..Authentication is among the most significant Concerns about security. The wide-ranging identification issue in the literature has led to the development of a broad variety of cryptographic algorithms. Threats include attacks, side-channel analysis, minimally invasive attacks, and reverse engineering. Due to these risks, it is no longer possible to employ this traditional method as IoT device security measures. Devices and equipment have become increasingly widely utilized in people's everyday

Responsible Editor: S. Bhunia

✉ R. Naveenkumar
naveentamil256@gmail.com

N. M. Sivamangai
nmsivam@gmail.com

A. Napolean
nepojustin@gmail.com

S. Sridevi Sathya Priya
s.d.s.priya@gmail.com

[1] Research Scholar, Dept. of ECE, Karunya Institute of Technology and Sciences, Tamilnadu, Coimbatore 641114, India

[2] Dept. of ECE, Karunya Institute of Technology and Sciences, Tamilnadu, Coimbatore 641114, India

lives in recent years, and the security issue has gotten a lot of attention. Counterfeiting, cloning, reverse engineering, and vicious component addition are just a few of the security issues. As a result of the insecure and identity information leakages, attackers have several possibilities to gain access to steal one's private details. Traditional software encryption algorithms are typically sophisticated, they are therefore inappropriate to IoT devices with minimal resources. Furthermore, they are demonstrated to be invulnerable to side-channel assaults. Primitives like side-channel countermeasures, PUFs, hardware obfuscation [4], and true random number generators (TRNG) are introduced in the last decade to resist hardware security threats. The approaches are utilized to improve system security in a variety of applications. Nowadays, PUFs are widely used in hardware security-based applications to allow devices to be identified and authenticated. PUF is lightweight hardware security primitive for the purpose of authentication and recognition of devices. PUFs are secure one-way functions that utilize intrinsic physical variances during the production process to provide specific output to a given input. As a result, challenge-response pairs (CRP) generated by PUF circuits made using identical manufacturing procedures are different. PUFs are unclonable and resistant to reverse engineering assaults as a result of this. Applications such as security, E-health, authentication of Wi-Fi, key setup for patient monitoring, authentication for IoT connectivity, and various others evolve PUF for security key generation. Intruder extract a fingerprint unique to the device using the uniqueness created during the device's fabrication. When an external stimulus is provided, one or more specified device parameters are monitored. When a device's parameter is taken for the first time, it is referred to as the "original response" for a certain stimulus or memory location, which is referred to as a "challenge," and both are maintained in the server. A response occurs if the same parameter is assessed with the same environmental stimulation. The Challenge-Response Pair (CRP) consists of such challenges and responses, which are used to confirm the device's identification. The CRP error is the difference between a PUF's CRP during the registration and authentication processes (CRP error).

Depending on their circuit configuration, PUFs can be classed as memory PUFs or delay-based PUFs, whereas, based on their character, Weak and strong PUFs are the two types of PUFs. Weak PUFs take use of manufacturing heterogeneity and enable the digitization of a hardware device's "fingerprint." When the PUF is weak, the number of responses and the number of components in the CRP-generating device is proportional [5]. As a result, there are just a few CRPs with steady responses that are typically resistant to environmental changes. Weak PUF responses are commonly utilized for secret key creation due to their great stability and consistency. A device with a significant number of CRPs is referred to as a "strong PUF." Generally, if there are enormously high CRPs, The attacker will be unable to use all of the responses and gain access to the system. For authenticating, strong PUFs are widely used [5]. PUF responses in large numbers, on the other hand, may provide more cryptographic strength by resulting in longer cryptographic keys [6]. One cannot predict the CRP in the PUF even if one CRP is known. This is known as independent CRP, implying that no common information can be shared. Based on the achievement of Uniqueness, PUF can be categorized. PUFs that have had their variation generated by externally adding extra procedures, such as coating PUFs are known as Explicit PUFs. Implicit PUFs [7] are unpredictability that occurs naturally as a result of variances in the production process. The PUF key metrics of the intra-distance and inter-distance statistical factors for determining PUF usability are specified as follows in [8]: "Intra-distance: the Hamming or fractional Hamming distance between two separate PUF challenge replies". "Inter-distance: the Hamming or fractional Hamming distance between two replies to a given challenge by two separate PUFs". The reproducibility and uniqueness of the PUFs are determined by these metrics. The following is an outline of the work: Section II deals with traditional APUF and the suggested implementation of the XOR APUF design is discussed in detail in section III. Results and PUF metrics are assessed in sections IV and V.

2 Existing Arbiter PUF

Delay-based PUFs create various delays in a circuit as a result of manufacturing variances in its components, even though the layout is the same. APUFs and Ring Oscillator (RO) PUFs are the most well-known delay-based PUFs [9]. The silicon PUFs are the first of their kind, proficient of foreseeing delay discrepancies among parallel of two delay lines and generating distinct CRPs. The intrinsic timing variations of 2 symmetrically constructed paths are used by arbiter PUFs to create a single bit of output response [10]. The Arbitrator PUF (APUF) structure is made up of numerous interconnected phases and at the end of the PUF chain, an arbitrator. There are also challenges on the stage, which are used to create selected signals. The input is an identical enable signal during the first stage, although the last stage is coupled with a unique arbiter it ultimately decides which signal obtained initially. Due to this structure, the arbiter produces bits as a response to the APUF. Although the crossed paths are the same, the small-time difference between the two paths can still generate different responses.

One of the key features of APUF is its support for CRP: 2^N, the exponential number of the total number of multiplexer blocks N (switches). The circuit is challenging to model because it takes an exponentially large number of

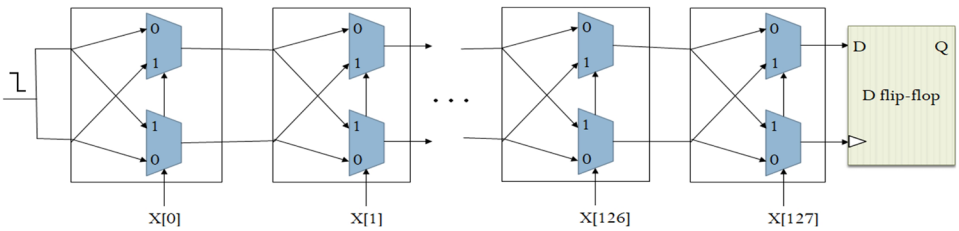


Fig. 1 Arbiter PUF Schematic[4]

attempts to achieve a certain level of uniqueness and dependability. In addition, the number of latency variables that have an impact on the response created is only linearity in N, indicating that 2^N demanding does not give free responses. By gathering the delay characteristics and interacting with the difficult bits, it is possible to model or clone the Arbiter PUF without having physical access rights to the PUF and predict the result with a little amount of extra latency. Figure 1 depicts the Arbiter PUF schematic.

The test pulse generator (TPG) block, path swapping switches (switch I, I = 0, 1,..., N), and the arbiter circuit are used to make an A-PUF. Figure 2 depicts how APUF is fundamentally organized. Two symmetrical paths are selected by an N-bit input challenge C for the test pulse propagation. Each switch (switch I has two operational modes: straight and cross. Figure 2 displays both switch I setups when Ci = 0: If Ci = 1, then I1 to O2 and I2 to O1 for the cross mode, and I2 to O2 and I1 to O1 for the straight mode. If the lowest signal (O2) is quicker than the higher signal (O1) produced through the arbiter, which is frequently a DFF implementation, Switch N1 produces 0. The arbiter is frequently a DFF implementation while the lowest signal (O2) is quicker than the higher signal (O1). One of the main issues with the programme is DFF's metastability, which accounts for this implementation's low reliability.

A Challenges Associated With APUF

(i) The APUF circuit has already been identified to be vulnerable to risks from incremental delay software modelling [11]. The upper and lower routes of the APUF on the very last level are taken into consideration the level's total latency in the suggested model. It is possible for determine the discrepancy in delay that is utilized to estimate the response by estimating the delays at certain paths. Obtaining the delay discrepancy for each CRP while doing a modeling assault is difficult. By the use of Linear Programming techniques, depending on the sign of the latency difference, the response can be predicted.

(ii) Due to physical layout constraints, APUF has problems such as weak uniqueness and dependability, especially while employed on FPGAs. A new element called FFAPUF developed for overcome these difficulties; it has a compact design, high uniqueness, and reliability features, and is appropriate for FPGA implementation. The conventional APUF design and the suggested FFAPUF design are far high versatile in path selection options than the traditional design of APUF. With LR and CMAES, the FFAPUF was proven to be extra robust to modeling attacks than traditional APUF design. The resistance of the FF-APUF architecture is assessed using the two most popular machine learning-based modelling assaults, linear regression (LR) and covariance matrix adap-

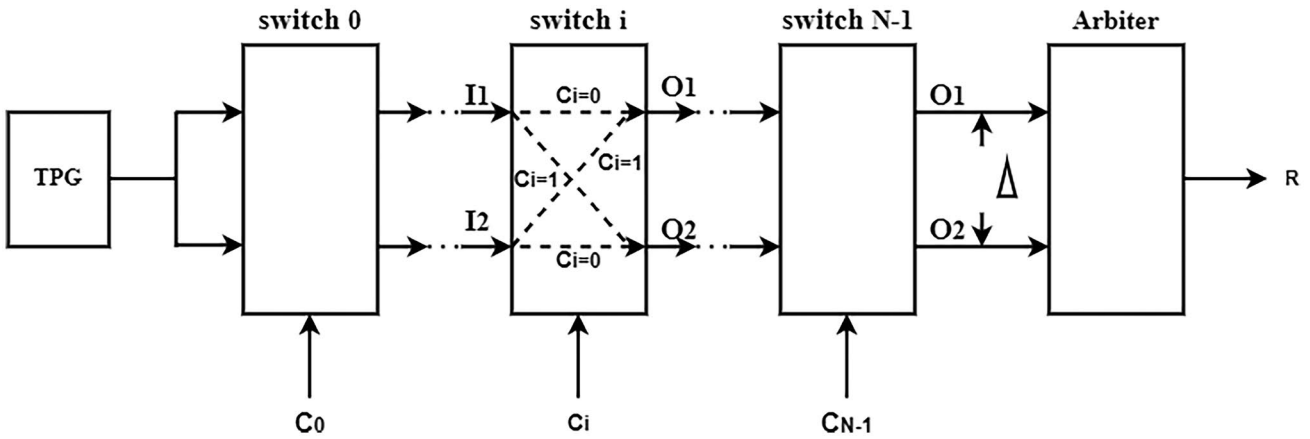


Fig. 2 An A-PUF structure with N stages [6]

tation evolution strategies (CMA-ES). One of the most effective modelling assaults to undermine the APUF design is LR. While the results of the same configuration for the FF-APUF design are around 34 and 79 s, respectively, the computational attack time when utilising LR for the APUF design is less than 1 s with response sizes of 64-bit and 256-bit [12]. Another effective modelling attack technique that is frequently used for APUF assaults is CMA-ES. For XORed APUF designs, it has been demonstrated that the reliability-based CMA-ES assault outperforms the LR attack. Utilizing several modelling attack methodologies, the FF-APUF is harder to attack than the APUF [13]. Beyond LR and CMA-ES, other two modelling assaults name such as Multi-layer Perceptron (MLP) and Hybrid LR-reliability Assaults. The MLP has been used to simulate powerful PUFs since 2012. Hospodar et al. [14] employed an MLP architecture with a hidden layer of four neurons to simulate a 2-XORAPUF with 64 stages, with a prediction accuracy of nearly 90%. In 2017, Alkatheiri et al. [15] suggested a novel MLP design with three hidden layers, each with two $k+1$ neurons, where k is the number of loops in an FF-PUF. This MLP design can break FF-APUF with 64 and 128 stages. Then, multiple MLP designs were employed to represent l-XOR-APUFs. Mursi et al. [16]. Suggested MLP design is the most effective. The MLP, like LR, takes as input the feature vector associated with the problem. In hybrid LR-reliability assaults is to prevent reliability-based attacks from constantly focusing on the same APUF. There are two main ways to include weight constraints in reliability-based attacks: the first is to maintain the original CMA-ES attack and add a new APUF that is distinct from the other APUFs in the set to the weight set in each iteration; the second is to simultaneously learn all APUF weights and keep them distinct. The LR-reliability hybrid assault used the second approach.

FPGA-oriented the strong FFAPUF concept beats existing APUF designs and it has the ability to be used as the foundation for IoT CRP-oriented authenticating process. The designed FFAPUF design has less predictive rates than the traditional APUF design.

(iii) Due to the fact that PUFs' attributes depend on tiny, irregular process changes, a low supply voltage may provide a PUF with maximum variation sensitivity and consuming low power. A PUF sub-threshold arbiter for 45 nm CMOS is created [17]. The sub-threshold arbiter PUF achieves lower power, higher uniqueness, and appropriate dependability and security. The greater gate widths make the 45 nm sub-threshold PUF arbiter lower efficiency than the future version, despite the fact that it is ideal for low-power applications.

(iv) Two key concerns with APUF developed on FPGAs are susceptibility for machine learning (ML) attacks and lower uniqueness. To overcome the issues, a double APUF (DAPUF) is implemented to duplicate the standard APUF. On the Xilinx Virtex5, According to experimental findings, DAPUF's uniqueness is nearly perfect, and the ML assaults detection rate lowers from range of 86% to 57%. [18]. The 21 DAPUF predicting rate is 69%, which is low but enhanced by APUF. Although the predicting rate for 31 DAPUF is 57%, which can be described as certain to a certain limit, despite the limited process tolerance.

(v) Using three-digit quadruple (trit) responses, [19] discusses a reliable proper authentication approach that relies on a Hardware implementation of an APUF. A two-flip arbiter has been used to generate a trit for metastability detection. On assessing the ordered response to the 4 permutations of the first and last control bits, each and every quadruple response could be compressed into a quadruple representing one of the five most reproducible trit quadruple responses classes. This allows you to create an accurate APUF template on the host without having to store the full authentication CRP. Although the quadruple challenge-response classification of the method minimizes the amount of mistake correction required, this one has issues regarding lot of hardware resource needs.

(vi) The APUF with a Programmable Delay Line (PDL) is implemented using the FPGA-based APUF architecture. This paper [5] describes a scalable design process for building a nearly ideal APUF on a Xilinx FPGA using the typical Xilinx CAD tool flow. The essential concept is to develop bias-free symmetrical delay pathways using the Hard Macro functionality of the Xilinx design flow. Environmental changes can affect PDL delays in a variety of operational environments.

(vii) The CAD tool's hardware macro functionality is necessary to implement the PDL-based APUF, which limits design freedom. Additionally, for attain features of PUF, such systems necessitate fine adjustment. Path Change Switch (PCS) [6] is a new switching mechanism that can be simply implemented on FPGAs. The findings demonstrate that the recommended APUF design has excellent thermal stability and enhances PDL-based APUFs without any further fine-tuning changes. However, without a hard macro on an FPGA, APUF has serious restrictions. The delays caused by the tools' automated routing greatly

outweigh the little delays caused by process variability. Because the PUF is based on the unpredictability inherent in process variability, this design will not work properly. The difference in routing dominates the randomness, not the variations in process variation. However, by employing hard macros, the routing may be made quite symmetric [20].

The Xilinx FPGA Editor's Hard Macro function is a useful tool for PUF designers. It enables the designer to describe a module with defined routing and placement that can be instantiated later on-demand without having to go through the time-consuming synthesis, placement, and routing processes. Using pdlCore.ncd, we created a hard macro pdlCoreHm.nmc. This hard macro may be used to define an APUF, which is a collection of PDLs. Our hard macro-based approach enables the designer to put any XORAPUF with an arbitrarily high number of challenge bits. Furthermore, unlimited PDL chain deployment utilising hard macro minimizes routing congestion by removing the necessity for vertical or horizontal placements [5]. Making a hard macro from a lengthy PDL string, yet, requires effort and reduces design flexibility.

(viii) The arbiter PUF (APUF) is an example of a well-known PUF circuit. Moreover, its FPGA implementation is unreliable, and to minimize the noise in answers, error-correcting codes (ECCs) are frequently necessary, resulting in significant additional hardware complexity [8]. Discusses a high accuracy arbiter PUF with enhanced uniqueness utilizing the Binary Testing Strategy (BST). In a traditional PUF arbiter, a latency tracking circuit is included to evaluate the latency variation which creates every bit of the PUF response in real-time, and a confidence flag is used to indicate the response as reliable. In BSTA-PUF, confidence flags will reveal sensitive response information, this makes the modelling attack more vulnerable and lowers the number of CRPs required for a successful modelling attack. Data for CRP and help are encrypted before being transmitted across the network, finding it difficult for attackers to get them. These protocols use lightweight encryption like XOR to provide a good blend of protection and low overhead.

(ix) To reduce its vulnerability to the ML threats while maintaining a high level of reliability and uniqueness, to address entry-related issues, APUF uses a Multi-Entry Signature Register (MISR) [9]. Using vector machine support and enhanced gradient learning methods with a training sample of 100,000 sets of challengers, the A MISR's 128-stage arbiter PUF's predictive power was improved in the FPGA implementation from 98%. Because MISR is used in this method, No new hardware resources are needed because the registered Built in Logic Block Observation (BILBO) can be used instead, hence this module has to be constructed separately.

(x) The structure of APUF with multi-line function selection was devised using the linear relationship between both reliability and signal latency difference [21]. In order to reduce ambient noise distortion and improve APUF uniqueness, it additionally duplicates numerous pairs of APUF switches. Also, it shows how this APUF design is very stable. This design has high resource consumption.

(xi) The Arbiter PUF's challenges and responses have a linear correlation, and as a result, the intruder could analyze the APUF using ML algorithms. By raising the amount of challenge inputs discarded, the PUF arbiter challenge data pre—processing framework (CPPAPUF) [22] increases APUF's ability to withstand ML attacks. The outcomes of the experiment reveal that the designed CPPAPUF seems to have an identical uniqueness and uniformity to the ideal PUF, but it has significantly less stability than the classic APUF design.

3 Implementation of Proposed XOR Arbiter

Since a regular XOR PUF is composed of many regular arbiter PUFs as components, it is more dependable than feed-forward (FF) PUFs or traditional arbiter PUFs. The XOR arbiter PUF uniquely identifies each IoT device. This approach is particularly effective against ML attacks. It is thus shown that it solves the drawbacks of different existing PUF-based authentication techniques. Methods of machine learning are used to attack PUFs. The architecture incorporates a non-linear characteristic to increase resistance against APUF attacks using machine learning. The XOR-APUF [13] is a prevalent approach for giving the PUF design nonlinearity. PUF in the XOR arbiter structure, to create the final response bit from a PUF response, an XOR operation is used. Figure 3 depicts schematic for the XOR PUF.

More secure results are obtained in XOR PUF by XORing parallel APUF outputs. To overcome the APUF's lack of modeling attack resistance, XOR PUFs were developed. In Fig. 3, the basic premise, PUFs are given the identical challenge and create N different response outputs, which are then XORed simultaneously to make a one-bit final response. The assumption behind Calculating individual PUF responses from the XOR output is extremely complex and time-consuming with XOR PUFs. Since of this, XOR PUFs are generally better secure than MUX PUFs. As previously stated, the number

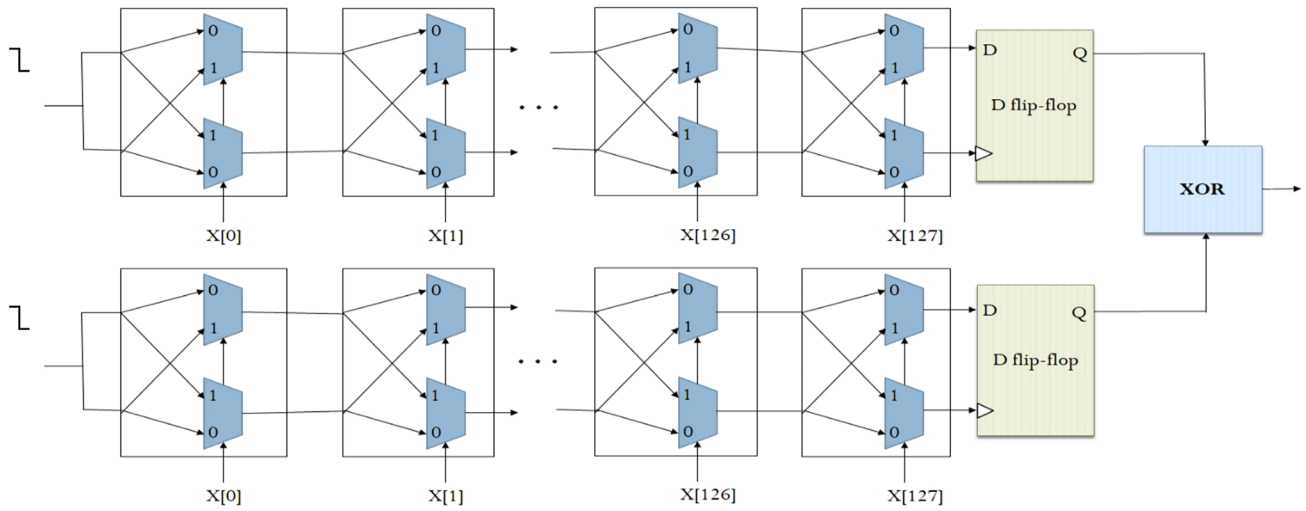


Fig. 3 Schematic for the XOR PUF

of PUFs used for XOR operations has a considerable effect on a PUF's security and stability. The outputs of simultaneous APUFs are XORed to form a single-bit response in this system. Non-linearity in the design is improved by XORing the outputs, making it more immune to modeling attacks. As a result, attackers will be unable to imitate the Challenge-Response Pairs (CRPs).

The two pathways in the typical PUF Arbiter configuration are built using 64 2-to-1 MUX pairs [17], It is a D flip-flop, the Arbiter. When the clock rises, designers can check to see if the D input has reached because the clock is connected to two routes and D is the D flip-flop input. The outputs of two D flip-flops are connected to the XOR gate in our XORAPUF by doing the same thing. It is difficult to design symmetric signal routing in PUFs and FPGAs, particularly in delay-based PUFs. This issue can be effectively solved by the hand-drawn design of ASICs, whereas FPGAs are physically constrained by their interconnect arrangement [18]. The module containing the physical details for a 2to1 MUX pair is implemented in [19] using hard macros. By copying the hard macros 64 times, it is able to construct a routing that is largely symmetrical. The whole layout can be produced in a matter of seconds by merging hard macros, which are precompiled modules. Designers still struggle to establish a perfect symmetry between the top and bottom routes even though the asymmetry issue can be solved using hard macros. A modelling assault is made possible by enhancing the training data, and the resulting prediction accuracy is 99%.

4 Architecture Design of XOR Arbiter PUF

This section introduces and discusses the XOR Arbiter circuit. XOR gates play an important role in the proposed delay PUFs. Multiple arbiter PUFs make up an XOR APUF or XOR PUF. APUF with n stages consists of n pairs of 2-to-1 multiplexers, each obtaining the identical input challenge bit at the same stage. At all levels of the routes, both signals pass via gates, resulting in somewhat varied delays as they pass over different gates. The ultimate output is determined by an arbiter, which is commonly a D-latch, based on which signal comes initially. In case the top path is selected initially, for example, the result is 1, or it is 0. At all stages, the pathways are determined by the challenge bit values and, as a result, the signal latency, resulting in a total of 2^N possibilities. If the challenge and the response fulfill the linear model of cumulative latencies, machine learning attacks can quickly break arbiter PUFs [23].

In Fig. 4, N is the number of constituent APUFs, which is three. The components of a K-Stage N-XOR APUF, which is depicted in Fig. 4, are k APUFs. To generate the final response for the associated challenge, Constituent PUF responses are XORed together. XOR PUFs have a greater circuit implementation cost than arbiter PUFs due to the additional gates, However, the XOR gate raises the response's complexity as a function of challenge bits, making ML-based assaults extremely hard [25].

This study investigates XOR PUFs for circuit design parameter ranges. Plan to process XOR PUFs with 2 to 8 component arbiter PUFs and stages of 16, 32, 48, and 64. While XOR PUFs have been widely explored in the context of Hardware implementation and improvisation [15] and security risk assessments, there has never been a study of XOR PUFs that covers such a wide variety of parameter values. ML assaults PUFs with fewer than 10 elements cannot be defeated by the 64-stage XOR PUFs [26], as these methods of attacks need a higher amount of CRPs, K-Stage N-XOR APUF (Fig. 4). (In this diagram, N represents the number of element APUFs, which is 3). These can only be accessed

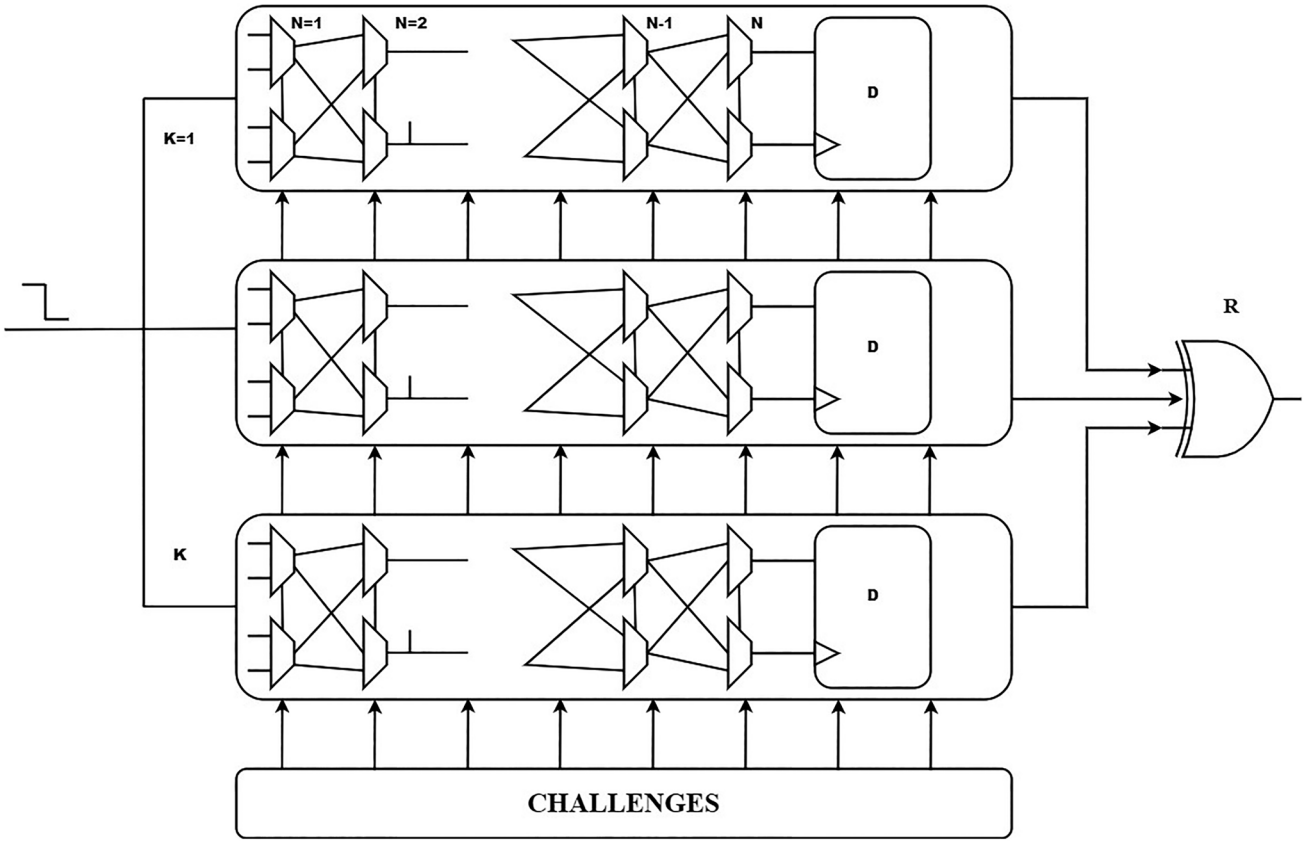


Fig. 4 K-Stage N-XOR APUF [24]

when the PUF provides an interface with no restrictions. Whenever a PUF-embedded device has a mutual authentication process, CRPs can only be retrieved via eavesdropping on conversations among the PUF as well as its communication partner, resulting in a significant reduction in the number of CRPs available. Because mutual authentication is used, lightweight (tiny) It's worth looking into XOR PUFs with fewer than ten elements, as they use fewer resources which may still be extremely secure for certain applications. This idea comes from a comprehensive view of both PUF resource needs and security in opposition to ML attacks.

Almost the same comprehensive concerns lead one to believe that compact PUFs with stages below 64, such as 48 and 32, could be able to provide sufficient protection for devices with mutual authentication, an assumption that motivates us to investigate stages 48, 32, and 16 to see whether 16 achieves the usefulness limit, at which point XOR PUFs will perform poorly almost in all properties. Some other reason is that response-obscuring techniques [27] allow lightweight PUFs, such as smaller XOR PUFs with few components and stages, to maintain strong security. When examining their possible use in devices that are using response-obscuring treatments, XOR PUFs become meaningful for a variety of circuit architectural model parameters, especially small values. ML assaults patterns can consistently predict PUF responses in some cases, with a prediction accuracy rate exceeding 98% in certain instances. Before an attack method can accurately predict PUF reactions, it should be trained on a set of CRPs of the PUF to be assaulted. According to research, ML attacks are vulnerable to 64-bit XOR PUFs having fewer than ten components. However, the attacking methods [28] must be tested on a large number of CRPs. PUF-enabled/incorporated devices with mutual authentication or response obfuscation mechanisms may be able to prevent intruders from collecting a great amount of CRPs.

The amount of information generated for training machine learning systems has a significant impact on their prediction power. As a result, the pairing of CRPs in use in training, as well as the prediction accuracy whenever the trained ML technique is evaluated on a pair of new challenges mostly for prediction power, are used to assess a PUF's security as opposed to ML assaults.

5 Simulation and FPGA Implementation

There are actually several studies that have already examined APUF and XOR PUF, respectively. While some articles highlight the vulnerability of XOR arbiter PUFs, some papers

highlight the physical properties of arbitrator PUFs. Few of them, nevertheless, compare the similarities and contrasts amongst them. A 64-bit Arbiter PUF and an XOR Arbiter PUF that accepts 64-bit challenges were our design goals. Both of these PUFs were implemented on the ZYBO FPGA board. The unclonable chip signature produced by process variances is utilised in the PUF design. The same concept underlies both Arbiter PUF and XOR Arbiter PUF: create a circuit that creates logic-0 or logic-1 depending on process variances, and then run the circuit n times to get replies in n-bit binary chips. In a typical Arbiter PUF, both routes are triggered by a single input. An arbitrator (D flip-flop) is used to translate the analogue delay difference between pathways into a digital value at the intersection of two parallel (racing) paths. The two pathways can be split up into more manageable sub-paths by adding a path switch. Each set of inputs to the switch functions as a challenge set, defined by Ci, generating a new pair of race routes whose durations may be compared to produce a bit-response. Arbiter PUF and XOR Arbiter PUF both need to determine which path causes the arbiter to reach a decision with the least amount of delay. However, XOR Arbiter PUF twice the process and employs an XOR gate to finish the output. By placing the two last pickers before the arbiter, the goal is to produce a result that is more random. We assess the dependability, homogeneity, and originality of these two PUFs in this work.

The XORAPUF is designed in Xilinx Vivado Software and the output schematic is taken is shown in Fig. 5.

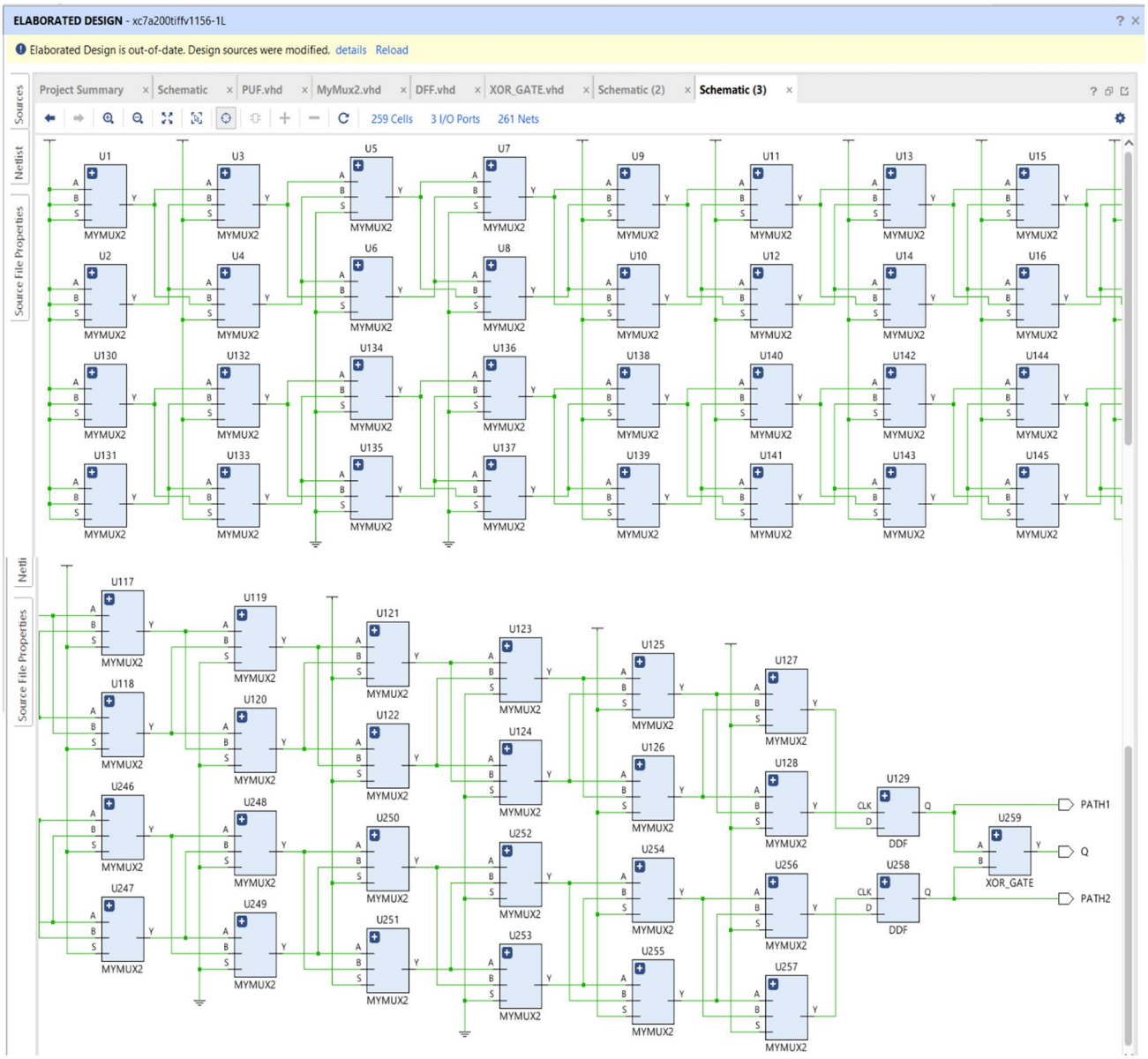


Fig. 5 The XOR APUF's output diagram

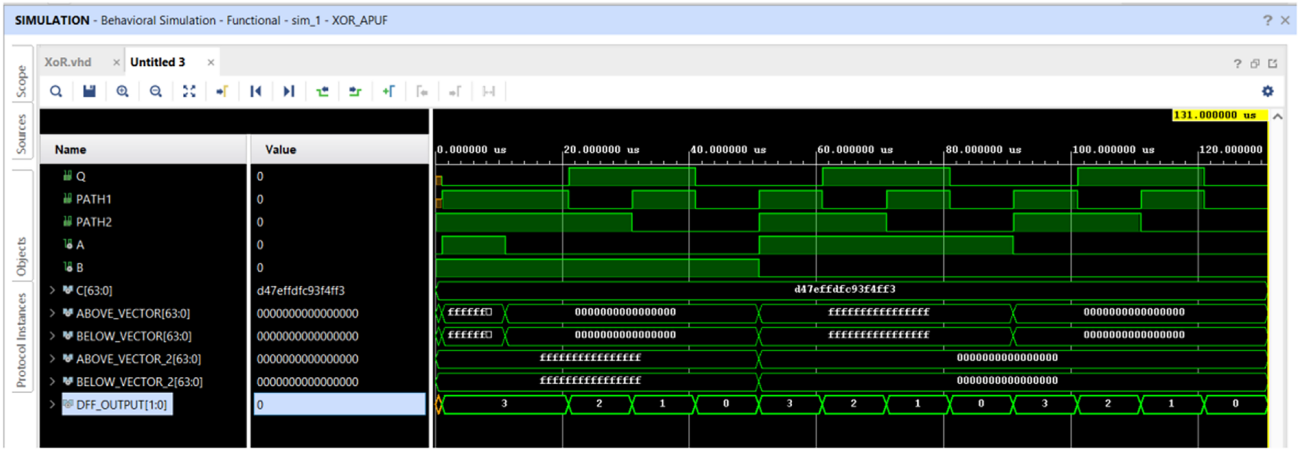


Fig. 6 Output waveform when the input (A = 0 and B = 0)

A. Output waveforms of xor arbiter puf

These output waveforms shown in Figures 6, 7, 8, and 9 are generated with different input challenges to obtain random responses.

These are the four conditions taken into consideration and this is implemented in the FPGA (ZYBO) kit for collecting the data.

B. Output data analysis

The output data is collected from different FPGA (ZYBO) kits, it shows the ability to predict the correct responses for the specific CRPs produced. Its input and output parameters are shown in Table 1, and information is collected as 0's and 1's. Finally, 100 sets of data are collected by different FPGA kits. For every dataset, the first 64 are inputs and the last one is output.

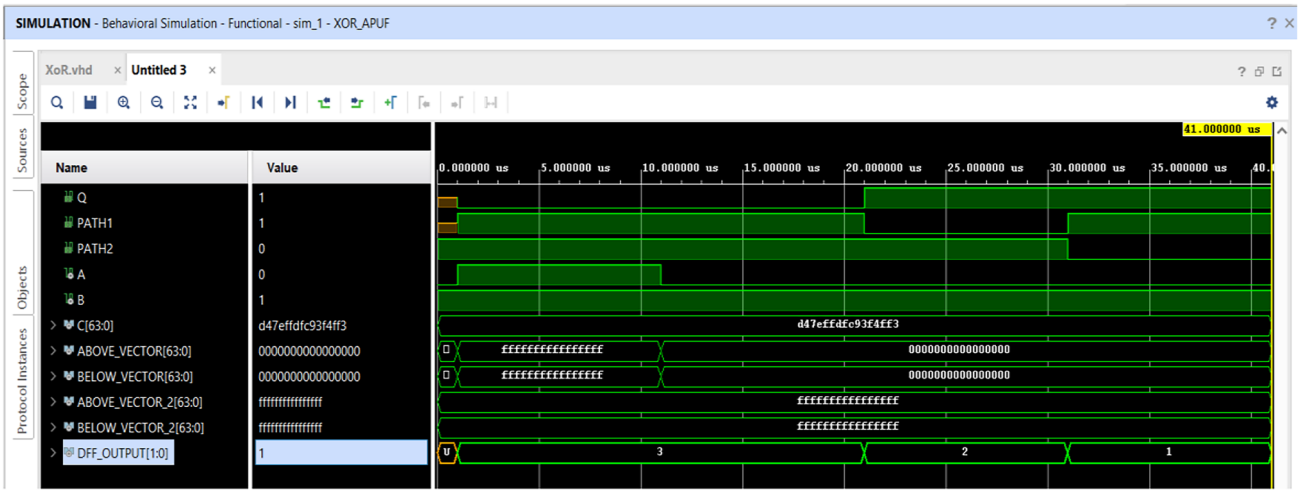


Fig. 7 Output waveform when the input (A = 0 and B = 1)

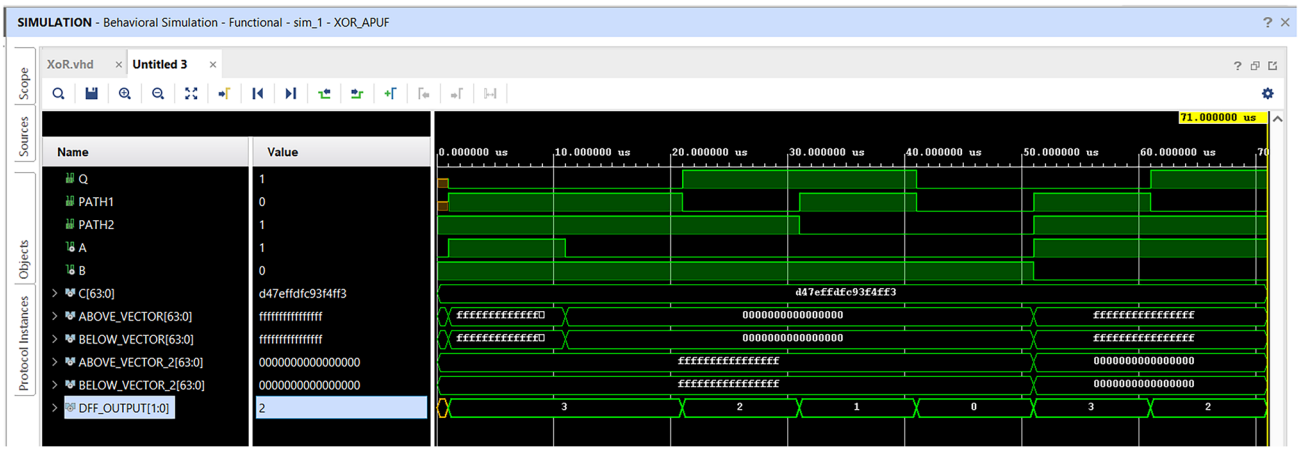


Fig. 8 Output waveform when the input (A = 1 and B = 0)

III. FPGA ZYBO Kit Outputs:

Figure 10 shows various responses to the FPGA kit. The output bit is taken as a response from the FPGA kits and the output is taken in 3 different criteria as shown in table 3. First, the identical design was used on two distinct boards (inter) to the same challenges that unique and arbitrary responses observed. It demonstrates the PUF's uniqueness. Second, relatively similar design, similar board, but different clock area (intra), same challenges, and distinct and random responses are recorded. Third, observed the same response bits again and over with the same design, board, and clocking region, but on various timings. It demonstrates the PUF's reliability.

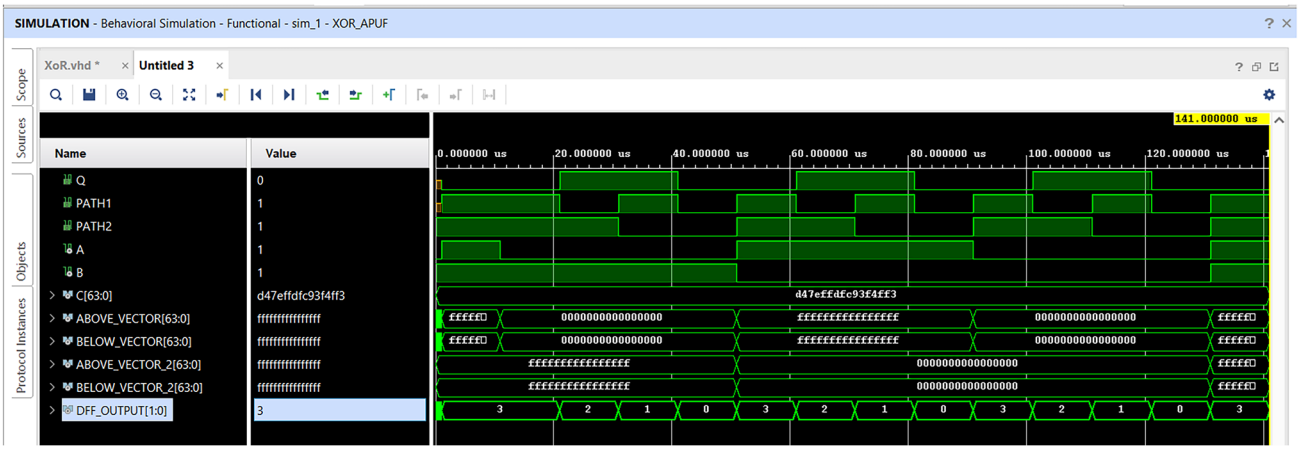


Fig. 9 Output waveform when the input (A = 1 and B = 1)

Table 1 Collected output data using FPGA Kit

A	B	S(64-bit)	Path 1	Path 2	Q
0	0	d47effdfc93f4ff3	0	0	0
			1	0	1
			0	1	1
			1	1	0
0	1	d47effdfc93f4ff3	0	0	0
			1	0	1
			0	1	1
			1	1	0
1	0	d47effdfc93f4ff3	0	0	0
			1	0	1
			0	1	1
			1	1	0
1	1	d47effdfc93f4ff3	0	0	0
			1	0	1
			0	1	1
			1	1	0

6 Analyzing the PUF Metrics

Various metrics were specified to measure the performance of the suggested PUFs, however, each implementation's parameter or evaluation criterion differed. To measure the performance of these PUFs, a common set of parameters is required. To analyze and examine the effectiveness of PUFs, the quality metrics were refined systematically. Uniqueness, uniformity, and randomness are the three parameters. The metrics of PUF are analyzed in the MATLAB tool. The MATLAB code was created to calculate uniqueness, uniformity, and randomness, and results are obtained.

A. Uniqueness:

After several readings, reliability assesses how consistently the PUF replicates a response for the same challenge. Reliability is used to calculate the average intra-class score. The Hamming distance, HD (R, R'), is determined over x samples in the following way: The typical intra-chip HD over k samples/chips on a per chip, denoted as i, has an n-bit standard response Ri(n) as it is from the chip i under normal operational conditions and the similar n-bit response R′i (n) for the challenge C, accordingly.

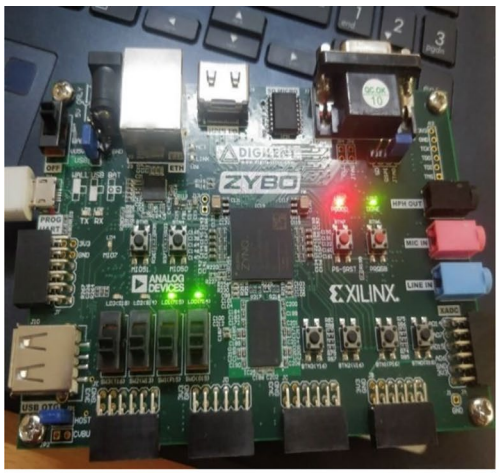
$$U = \frac{2}{m(m-1)} \sum_{i=1}^{m-1} \sum_{j=j+1}^{m} \frac{HD(Ri, Rj)}{n} * 100$$

The inter-chip HD distribution is used to determine uniqueness, as shown in Fig. 11. This quality can be defined by the value of inter-HD distance. From the MATLAB code, we calculated the uniqueness in terms of inter-HD distance is 49.88%, which refers to this percentage being quite near the ideal figure is 50%. Hence, the performance of uniqueness for XOR Arbiter PUF is good.

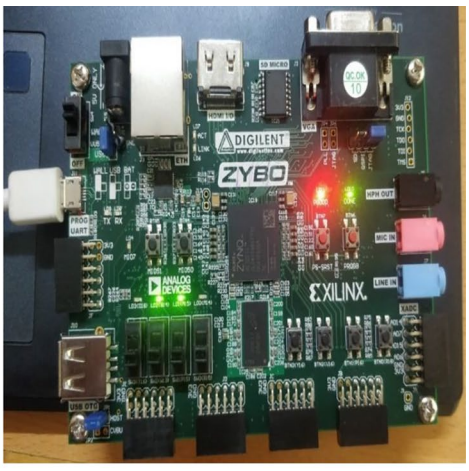
B. **Reliability:**

To determine Reliability, we utilize the Hamming distance (HD) among two PUF identifiers. Unless two devices,

Fig. 10 FPGA kit outputs



When A=1, B=1, path1, and path2 are high, the output response (Q) will be '0'



When A=1, B=0, path1=0, and path2=1, the output response (Q) will be '1'

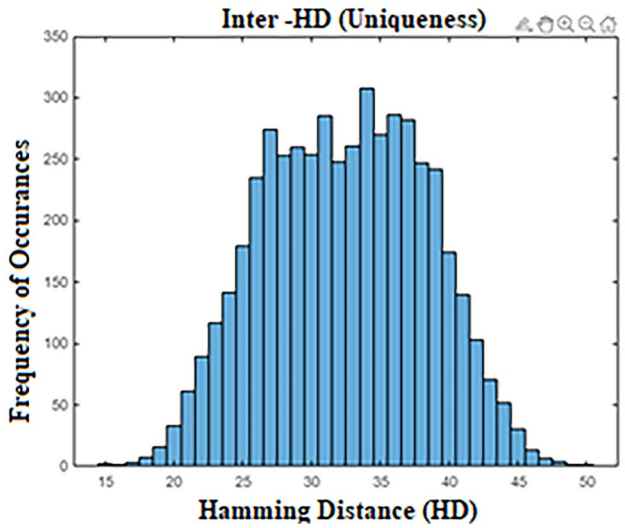


Fig. 11 Uniqueness graph of XOR Arbiter PUF

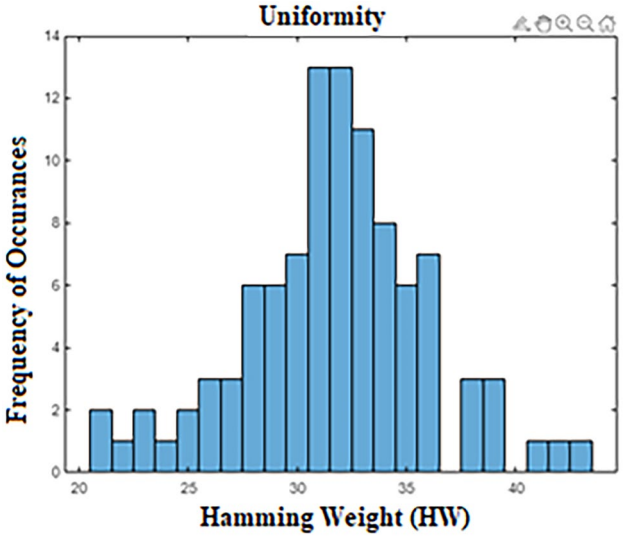


Fig. 13 Uniformity graph of XOR Arbiter PUF

i and j (i! = j), for challenge C, have n-bit responses, Ri and Rj, essentially. PUF's uniqueness is determined by how well it can identify the device on which it is used. The average inter-class Hamming distance is computed as follows:

$$\boldsymbol{R} = \frac{1}{\boldsymbol{x}} \sum_{y=1}^{x} \frac{\boldsymbol{HD(Ri, Riy)}}{\boldsymbol{n}} * 100$$

The number of devices is M, and the n-bit response of i th and 3j th PUF occurrences, approximately, are Ri and Rj. PUF responses lose their uniqueness if all of the chips shave the same bit value ('0' or '1'). The reliability graph is shown in Fig. 12. We calculated the reliability in terms of HD distance is 99.20%, which refers to this percentage being quite near the ideal figure of 100%.

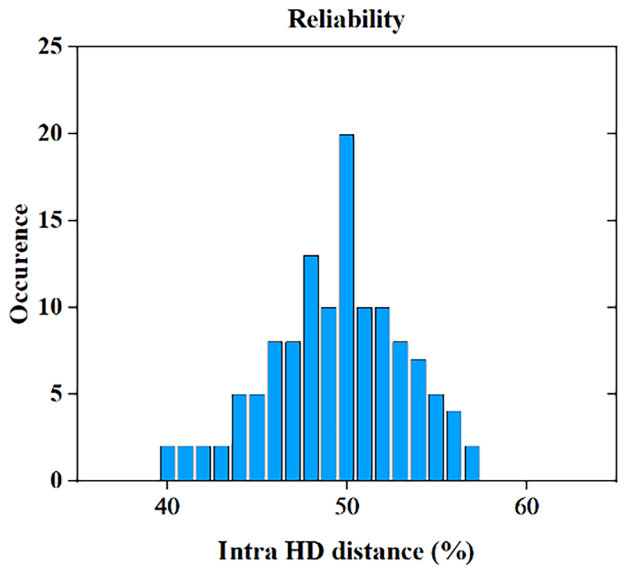


Fig. 12 Reliability graph of XOR Arbiter PUF

C. Uniformity:

The ratio of 0 s and 1 s in a PUF's response bits is measured by its uniformity. For completely arbitrary PUF responses, this percentage must equal 50%. An n-bit PUF identification's homogeneity is measured using the percentage of the Hamming Weight (HW).

$$U = \frac{2}{m(m-1)} \sum_{l=0}^{n} ri, l*100$$

It can find that the probability for '0' and '1' is random. Due to the XOR Arbiter PUFs performances of uniformity, the frequency of 0 and 1 for this FPGA is 48.74%, which means they are distributed equally likely between '0' and'1'. Hence, XOR Arbiter PUF shows good performance in uniformity, uniformity is shown in Fig. 13.

Also, the obtained values are compared to the existing PUF metrics in Table 2. The results were much better than the existing values. The designed XOR Arbiter PUF has produced good results in terms of uniqueness, uniformity and reliability.

Table 2 Comparison of Proposed PUF metrics with traditional PUF

METRICS	EXISTING PUF [13] (%)	EXISTING PUF [9] (%)	EXISTING PUF [29] (%)	PROPOSED XOR PUF CIRCUIT (%)
Uniqueness	41.53%	49.30%	44	49.88%
Reliability	97.10%	98.80%	-	99.20%
Uniformity	-	-	-	48.74%

7 Conclusion

We examined the newest Arbiter PUF advancements as well as the difficulties in creating such security hardware. When defending against machine learning-based attacks, a greater advantage belongs to the XOR Arbiter PUF. The XORAPUF provides better uniformity performance due to its distinct XOR gate design. The reliability of the responses increases with the size of the delay difference. The experimental assessment of the enhanced XORAPUF design implemented on FPGA reveals a uniqueness,Uniformity and reliability of 49.88%, 48.74% and 99.20%, respectively, greater than those of the work [9, 13], and [29]. This is because of its unique XORAPUF design.

Author Contributions Statement The first author wrote the manuscript, and the second author supervised, third and fourth author- done the proof reading.

Data Availability Not Applicable.

Declarations

Research Involving Human and Animal Participants Not Applicable.

Competing Interests Nil.

References

1. El-hajj M, Chamoun M, Fadlallah A, Serhrouchni A (2017) Taxonomy of authentication techniques in Internet of Things (IoT). IEEE 15th Student Conference on Research and Development (SCOReD) 67–71. https://doi.org/10.1109/SCORED.2017.8305419
2. El-hajj M, Fadlallah A, Chamoun M, Serhrouchni A (2019) A Survey of Internet of Things (IoT) Authentication Schemes. Sensors (Basel, Switzerland) 19(5):1141. https://doi.org/10.3390/s19051141
3. Naveenkumar R, Sivamangai N M, Napolean A, Janani V (2021) A Survey on Recent Detection Methods of the Hardware Trojans. 3rd International Conference on Signal Processing and Communication (ICPSC) 139–143. https://doi.org/10.1109/ICSPC51351.2021.9451682
4. Naveenkumar R, Sivamangai N M, Napolean A, Nissi G A (2022) Hardware Obfuscation for IP Protection of DSP Applications. J. Electron. Test 38: 9–20. https://doi.org/10.1007/s10836-022-05984-2
5. Sahoo DP, Chakraborty RS, Mukhopadhyay D (2015) Towards Ideal Arbiter PUF Design on Xilinx FPGA: A Practitioner's Perspective. Euromicro Conference on Digital System Design. https://doi.org/10.1109/dsd.2015.51
6. Mahalat M H, Mandal S, Mondal A, Sen B (2019) An Efficient Implementation of Arbiter PUF on FPGA for IoT Application. 32nd IEEE International System-on-Chip Conference (SOCC) 324–329. https://doi.org/10.1109/socc46988.2019.157054
7. Gope P, Lee J, Quek TQ (2018) Lightweight and Practical Anonymous Authentication Protocol for RFID Systems Using Physically Unclonable Functions. IEEE Trans Inf Forensics Secur 13:2831–2843. https://doi.org/10.1109/TIFS.2018.2832849
8. He Z, Chen W, Zhang L, Chi G, Gao Q, Harn L (2020) A Highly Reliable Arbiter PUF With Improved Uniqueness in FPGA Implementation Using Bit-Self-Test. IEEE Access 8:181751–181762. https://doi.org/10.1109/access.2020.3028514
9. Zalivaka SS, Ivaniuk AA, Chang C (2017) Low-cost fortification of arbiter PUF against modeling attack. IEEE International Symposium on Circuits and Systems (ISCAS). https://doi.org/10.1109/iscas.2017.8050671
10. Tajik S, Dietz E, Frohmann S, Seifert J, Nedospasov D, Helfmeier C, Boit C, Dittrich H (2014) Physical Characterization of Arbiter PUFs. IACR Cryptol ePrint Arch. https://doi.org/10.1007/978-3-662-44709-3_27
11. Lim D, Lee J W, Gassend B, Suh G E, Dijk M V, Devadas S (2005) Extracting secret keys from integrated circuits. IEEE Transactions on Very Large Scale Integration (VLSI) Systems 13:1200–1205. https://doi.org/10.1109/TVLSI.2005.859470
12. Yao J, Pang L, Zhang Z, Yang W, Fu A, Gao Y (2022) Design and Evaluate Recomposited OR-AND-XOR-PUF. IEEE Trans. Emerg. Top. Comput. 10: 662–677. https://doi.org/10.1109/tetc.2022.3170320
13. Gu C, Liu W, Cui Y, Hanley N, O'Neill M, Lombardi F (2021) A Flip-Flop Based Arbiter Physical Unclonable Function (APUF) Design with High Entropy and Uniqueness for FPGA Implementation. IEEE Trans Emerg Top Comput 9(4):1853–1866. https://doi.org/10.1109/TETC.2019.2935465
14. Hospodar G, Maes R, Verbauwhede I M (2012) Machine learning attacks on 65nm Arbiter PUFs: Accurate modeling poses strict bounds on usability. 2012 IEEE International Workshop on Information Forensics and Security (WIFS) 37–42. https://doi.org/10.1109/WIFS.2012.6412622
15. Alkatheiri MS, Zhuang Y (2017) Towards fast and accurate machine learning attacks of feed-forward arbiter PUFs. IEEE Conference on Dependable and Secure Computing. https://doi.org/10.1109/DESEC.2017.8073845
16. Mursi KT, Thapaliya B, Zhuang Y, Aseeri AO, Alkatheiri MS (2020) A Fast Deep Learning Method for Security Vulnerability Study of XOR PUFs. Electronics 9(10):1715. https://doi.org/10.3390/electronics9101715
17. Lin L, Holcomb DE, Krishnappa DK, Shabadi P, Burleson WP (2010) Low-power sub-threshold design of secure physical unclonable functions. ACM/IEEE International Symposium on Low-Power Electronics and Design (ISLPED). https://doi.org/10.1145/1840845.1840855
18. Machida T, Yamamoto D, Iwamoto M, Sakiyama K (2015) Implementation of double arbiter PUF and its performance evaluation on FPGA. The 20th Asia and South Pacific Design Automation Conference 6–7. https://doi.org/10.1109/ASPDAC.2015.7058919
19. Zalivaka SS, Ivaniuk AA, Chang C (2019) Reliable and Modeling Attack Resistant Authentication of Arbiter PUF in FPGA Implementation With Trinary Quadruple Response. IEEE Trans Inf Forensics Secur 14:1109–1123. https://doi.org/10.1109/tifs.2018.2870835
20. Patterson M, Zambreno J, Sabotta C, Vyas S, Mills A (2011) Ring Oscillator PUF Design and Results
21. Wen J, Huang M, Chen Z, Zhu L, Chen S, Li B (2019) A Multiline Arbiter PUF with Improved Reliability and Uniqueness. IEEE 4th International Conference on Signal and Image Processing (ICSIP) 641–648. https://doi.org/10.1109/siprocess.2019.886888
22. Ge W, Hu S, Huang J, Liu B, Zhu M (2020) FPGA implementation of a challenge pre-processing structure arbiter PUF designed for machine learning attack resistance. IEICE Electron Express 17(2):1–6. https://doi.org/10.1587/elex.16.20190670

23. Ruhrmair U, Sehnke F, Solter, J, Dror G, Devadas S, Schmidhuber J (2010) Modeling attacks on physical unclonable functions. Proceedings of the 17th ACM conference on Computer and communications security 237–249. https://doi.org/10.1145/1866307.1866335
24. Mursi K T, Zhuang Y, Alkatheiri M S, Aseeri A O (2019) Extensive Examination of XOR Arbiter PUFs as Security Primitives for Resource-Constrained IoT Devices. 17th International Conference on Privacy, Security and Trust (PST) 1–9. https://doi.org/10.1109/pst47121.2019.8949070
25. Aseeri AO, Zhuang Y, Alkatheiri MS (2018) A Machine Learning-Based Security Vulnerability Study on XOR PUFs for Resource-Constraint Internet of Things. IEEE International Congress on Internet of Things (ICIOT). https://doi.org/10.1109/ICIOT.2018.00014
26. Becker G T (2015) The Gap Between Promise and Reality: On the Insecurity of XOR Arbiter PUFs. Cryptographic Hardware and Embedded Systems (CHES), Lecture Notes in Computer Science (9293):535–555. https://doi.org/10.1007/978-3-662-48324-4_27
27. Majzoobi M, Rostami M, Koushanfar F, Wallach DS, Devadas S (2012) Slender PUF Protocol: A Lightweight, Robust, and Secure Authentication by Substring Matching. IEEE Symposium on Security and Privacy Workshops. https://doi.org/10.1109/SPW.2012.30
28. Ruhrmair U, Solter J, Sehnke F, Xu X, Mahmoud A, Stoyanova V, Dror G, Schmidhuber J, Burleson WP, Devadas S (2013) PUF Modeling Attacks on Simulated and Silicon Data. IEEE Trans Inf Forensics Secur 8(11):1876–1891. https://doi.org/10.1109/TIFS.2013.2279798
29. Sushma R, Murty N S (2018) Feedback Oriented XORed Flip-Flop Based Arbiter PUF. International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT) 1444–1448. https://doi.org/10.1109/iceeccot43722.2018.9001605

R. Naveenkumar Karunya Institute of Technology and Sciences. Email: naveentamil256@gmail.com. R. Naveenkumar, is a research scholar at the department of Electronics and Communication Engineering of Karunya Institute of Technology and Sciences, Tamilnadu, India. He got his M.E. degree from Sri Shakthi Institute of Science and Technology, India in 2014. He has 6 years of teaching and 1 year of industry experience. His research interest is hardware security in microelectronics. He is also a member of MISTE, MIEANG, MIRED, and MSDIWC committees.

N. M. Sivamangai is an Associate Professor, Department of ECE, Karunya Institute of Technology and Sciences, India. She received her Ph.D. degree from Anna University, Chennai, India in 2011. She has 13 years of teaching experience. She was instrumental in the fabrication of IC jointly with Indian Institute of Science - Bangalore, in the year 2008. Her research interests are to design and test high performance semiconductor memories and to design VLSI based systems.

A. Napolean is working as a researcher in Electronics and Communication Engineering at Karunya University Coimbatore. He worked at VSB Engineering College from 2008 to 2017. He obtained M.Tech. degree in Sensor System Technology from Vellore Institute of Technology University in the year 2002. He completed his M.Sc. in Physics from Bharathiar University in the year 2006. He completed his Bachelor of Science in Physics in the year of 2004 from Manonmaniam Sundaranar University-Tirunelveli, India. He has research experience in the fields of nano materials, Electronic devices and Nano-scale memory devices. He is a member of MISTE, MIEANG, MIRED, and MSDIWC committees.

S. Sridevi Sathya Priya is an Assistant Professor, Department of ECE, Karunya Institute of Technology and Sciences, India. She received her Bachelor of Technology degree in Electronics and Communication Engineering form Madras University, Madras, India in 2001. She got her M.E degree from Karunya Institute of Technology and Sciences, India in 2006. She received her Ph.D. degree from Karunya Institute of Technology and Sciences, Coimbatore, India in 2017. Her research interests includes Hardware security, Cryptographic systems, Internet of things, Artificial Intelligence and parallel processing architecture.