IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received January 11, 2021, accepted January 24, 2021, date of publication February 1, 2021, date of current version February 9, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3056291

An Efficient Design of Anderson PUF by Utilization of the Xilinx Primitives in the SLICEM

ARMIN LOTFY[1], MASOUD KAVEH[2], DIEGO MARTÍN[1], AND MOHAMMAD REZA MOSAVI[2]

[1]ETSI Telecomunicación, Universidad Politécnica de Madrid, 28040 Madrid, Spain
[2]Department of Electrical Engineering, Iran University of Science and Technology, Tehran 16846-13114, Iran

Corresponding author: Diego Martín (diego.martin.de.andres@upm.es)

ABSTRACT Physical unclonable functions (PUFs) are known as one of the most recent promising technologies for cryptographic key generation. A PUF circuit is designed in such a way to produce random digits based on true-random and uncontrollable variations during the integrated circuits (IC) manufacturing process. The response of PUF can be used as a unique identity for the device where the PUF is embedded in it. Field-programmable gate arrays (FPGAs) are usually considered as one of the first choices for implementing PUFs. This paper proposes a novel FPGA-derived Anderson PUF by optimizing all elements located in one configurable logic blocks (CLBs). The experimental results on Spartan-6 family Xilinx XC6SLX9 FPGAs show that the proposed architecture improves the PUF's uniformity, uniqueness, and reliability to 49.41%, 50.89%, and 91.25%, respectively. Furthermore, the proposed structure increases the complexity and unpredictability of the PUF while decreases the hardware area overhead.

INDEX TERMS Anderson PUF, FPGAs, low-cost design.

I. INTRODUCTION

Physical security has become a significant concern in security applications related to cyber-physical systems in recent years. The concept of physical security can meet various efforts aimed at securing the cryptosystem against the physical attacks, e.g. secure key generation, secure key storage, secure cryptographic implementation, etc., [1]–[3]. Physical unclonable functions (PUFs) have been introduced as a promising technology to ensure physical security in recent years. A PUF is a unique identity that entirely depends on the uncontrollable variations of its manufacturing process. It is practically hard (or even impossible) to make a copy of PUF even with have the exact manufacturing process. Since the manufacturing process variations are entirely random, the PUFs responses are usually hard to predict [4]. These unique features of PUF make it very suitable for security applications such as key generation, authentication, and identification [5]–[9].

Assuming PUF as a function that maps a set of challenges (input) to a set of responses (output), then it can only be evaluated by its own physical system. There are various physical systems where PUFs can be implemented on them, such as delay-based intrinsic PUFs, memory-based intrinsic PUFs, non-electronic PUFs, and analogue electric PUFs. The first two kinds of PUFs are typical ones which are implemented in integrated circuits (ICs) [10]–[12]. Among all existing PUF types, delay-based intrinsic PUFs are known as the most famous ones like Arbiter PUF and ring oscillator PUF (RO PUF) [13], [14].

The utilization of field-programmable gate arrays (FPGAs) is rapidly grown in security applications. The excellent delay-based infrastructure and hard-macro property make FPGAs as one of the common choices for implementing delay-based intrinsic PUFs [15]–[18]. One of the most famous FPGA-based PUF named as Anderson PUF was proposed by Anderson in 2010 [19]. Anderson directly utilized components of modern FPGA devices, i.e. configurable logic blocks (CLBs), each of which consists of a few lookup tables (LUTs), multiplexers, and flip-flops. The key point of Anderson PUF's design relies on the difference of the

The associate editor coordinating the review of this manuscript and approving it for publication was Kuo-Hui Yeh.

switching latencies of two multiplexers chained together. The critical point to generating random digits through Anderson PUF is to generate a glitch of sufficient length to be created by many interlocking multiplexers called carry chains located in the CLB. This glitch inputs to a flip-flop and enables its preset. All of Anderson PUF elements can be implemented on two CLBs that makes it very hardware-efficient PUF. In this paper, we aim to improve the original architecture of Anderson PUF. Hence, the contributions of this paper are listed as follows:

- We proposed a novel structure for Anderson PUF that overcomes the original one's limitations and significantly improves its uniformity, uniqueness, and reliability.
- The proposed structure only needs one SLICEM to generate the desirable glitch, which leads to optimizing the used hardware in FPGA platform.
- By utilizing the exclusive-OR (XOR) gate in the SLICEM in its correct location, we improve the complexity and unpredictability of Anderson PUF without using any extra hardware.
- We implement our proposed PUF on a low-price and very public FPGA from Spartan-6 family named Xilinx XC6SLX9, which shows our scheme's scalability for low-cost applications.

The rest of this paper is organized as follows. A background of PUF and FPGA with the related works are presented in section II. The proposed PUF design is detailed in section III. The security and performance analyses are provided in section IV. Finally, section V concludes the conclusion of this paper.

II. BACKGROUND AND RELATED WORKS

In this section, we first review the structure of used FPGA, secondly the prior works about the PUF, particularly the Anderson PUF. In the next step, we go in detail of the methodology of the proposed Anderson PUF that is introduced in this paper.

A. SPARTAN-6 FPGA

The XILINX Company produces several categories of FPGAs that meet broad different needs. One of these categories is the Spartan series that give a reasonable price and adequate hardware resources simultaneously. Concerning acceptable performance and reasonable price, the Spartan 6 series become one of the most used series in different applications. The used FPGA in this paper is the XC6SLX9 series that is known as a low-price FPGA of this category and uses the 45 nm manufacturing technology.

According to the Xilinx documents, Spartan 6 encompass CLBs that each of them consists of two units named SLICE. Three distinct types of SLICEs used in Spartan 6 families are SLICEL, SLICEX, and SLICEM. Each of these types has especial characteristics and design. In the following, we go in detail of the structure, design, and the order of these SLICEs in CLBs. First of all, we review the structure of the SLICMs in Spartan 6, which is used in Anderson PUF. The structure of SLICEM is shown in Figure 1. There are two main characteristics of SLICEM. First of all, the carry chain that is implemented in this SLICE. The other fundamental feature is that the LUTs that are in these SLICEs can be used as shift registers. Although the SLICELs have carry chains such as SLICEM, the LUTs in this SLICEs are not able to use as shift registers. The last type of the SLICEs is SLICEX that has the minimum features in comparison with the other types of SLICEs. The implemented sources in each of the mentioned SLICEs are shown in Table 1.

TABLE 1. Comparison between hardware sources of three types of slides.

Features	SLICEX	SLICEL	SLICEM
6-input LUTs	✓	✓	✓
8 flip-flops	✓	✓	✓
Wide multiplexers		✓	✓
Carry logic		✓	✓
Distributed RAM			✓
Shift registers			✓

Finally, the order of these SLICEs is so critical because this structure is different in each series of FPGAs. Concretely, in Spartan 6 LX9, the order of these SLICEs is shown in Figure 2. Given this Figure, each CLB encompasses two distinct SLICEs; the right side is always the type of X, and the left side changes between slices of type M and L.

B. PHYSICAL UNCLONABLE FUNCTION

PUFs are defined into two main categories weak and strong [17]. Occasionally, weak PUFs are used to provide digital keys in the cryptographic algorithm. The central positive aspect of PUFs compared with the other methods is to generate these digital keys without storing them in the device [18]. The most known types of weak PUFs are SRAM PUF, butterfly PUF, and Anderson PUF [19]. In this paper, the Anderson PUF is selected, and several features make it an appropriate choice to implement on the FPGA series with the minimum required hardware sources. The fundamental feature that has a vital role in choosing Anderson is the required hardware sources that are impressively lower than the other types. Figure 3 shows the structure of the Anderson PUF. The other critical characteristic of Anderson PUF is about the speed of generating the intrinsic responses. The intrinsic feature causes a glitch through the lower multiplexer's path up to the upper multiplexer [20]. This time difference causes generating a glitch, and in the next step, this glitch will appear on the preset port of the flip-flop. Concerning changing the status of this signal, a response of 1 or 0 will occur. The LUTs play a role as a shift registers to generate a string of numbers that complement each other.

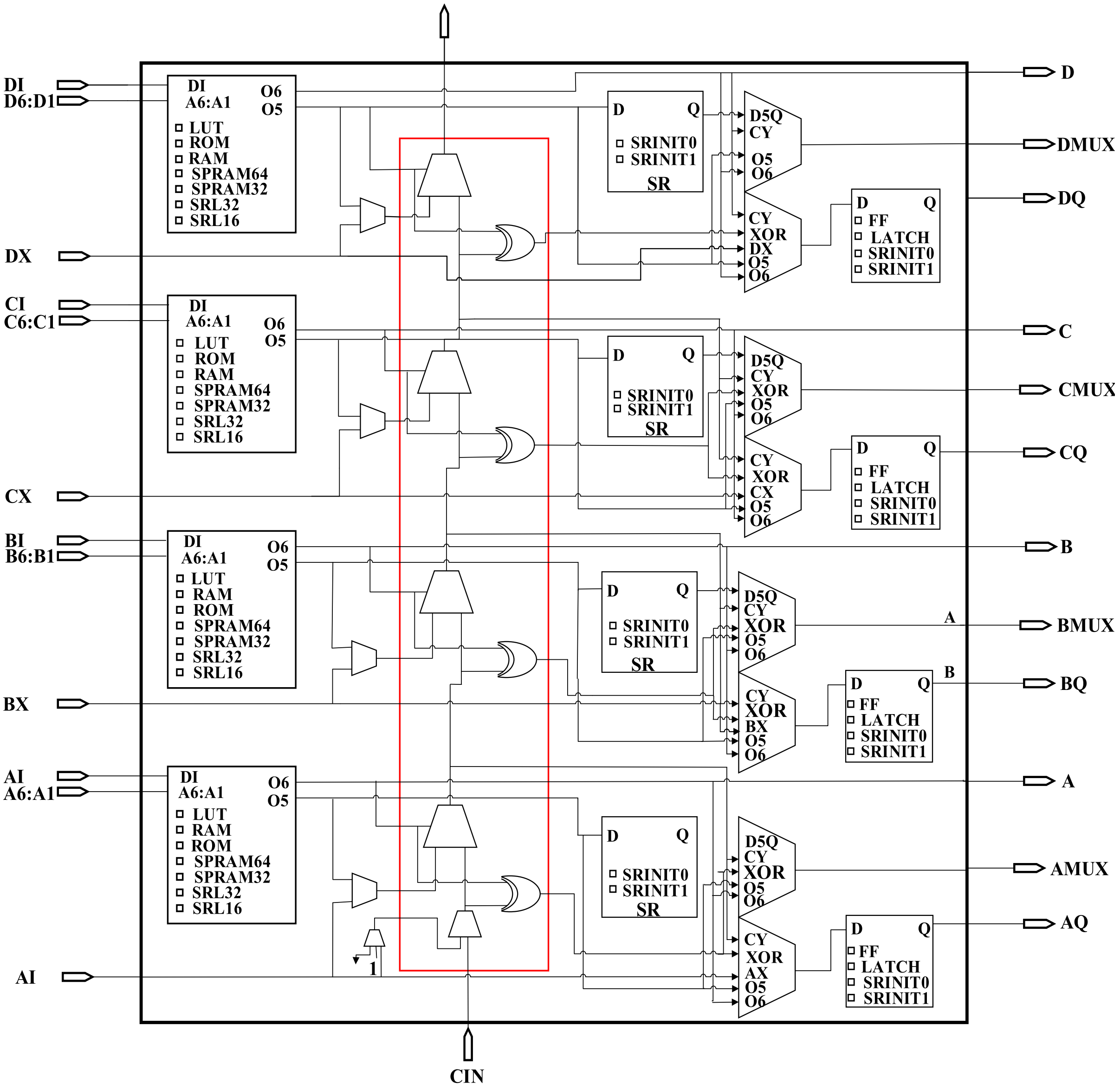

FIGURE 1. The structure and hardware resources of SLICEM.

C. RELATED WORKS

The concept of Anderson PUF was first proposed by Anderson [19]. The Anderson PUF is classified as a weak PUF. The structure of generating the responses is based on the glitch generated from the delay of two shift registers and a delay line. Owing to this point, finding the optimal values of each element's delay is a crucial point. The primary element in Anderson PUF is the shift registers that generate the two string of complement numbers. These numbers are connected to the select port of the two multiplexers. The port number one of multiplexers is connected to the value of one, and the other port is connected to the value of zero. Thereby, by changing the numbers on the select port of multiplexers, the output value of them will change from response of 1 to 0. The delay of this changing value in two levels generates a time difference. Through the delay line between two multiplexers, the difference time is added by t_{CHAIN}. In more detail, the delay of delay chain diminishes the time difference between two multiplexers. The output of the last level multiplexer is connected to the preset port of a flip-flop. The width of this signal is a vital factor to be visible by this flip-flop. If this signal's length is too narrow, the flip-flop will not distinguish this signal on its port.

Several types of modification followed it to improve the drawbacks of this PUF. The main negative point of Anderson PUF is to generate the complement string of numbers needing to use the LUTs implementing in SLICEMs in shift registers mode. Unfortunately, the distribution of the SLICEMs is

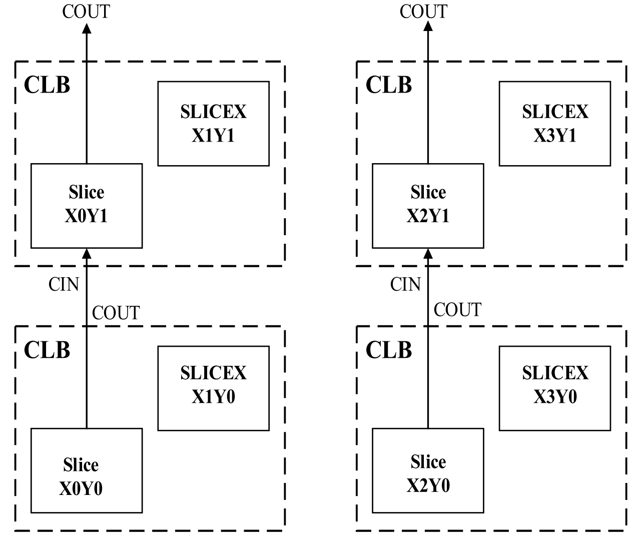

FIGURE 2. The order of SLICEs in CLBs and the way of connecting the inputs and outputs.

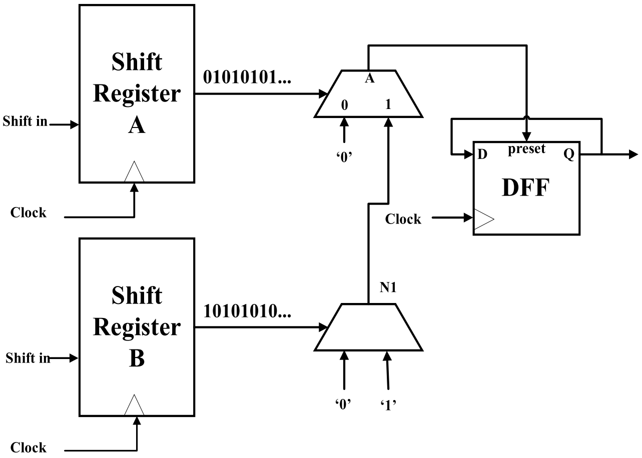

FIGURE 3. The Anderson PUF design [19].

not the same as the other SLICEs; in this way, the number of SLICEMs and SLICELs are half of SLICEXs. Concerning to this feature, the designers will face the restriction to implement the Anderson PUFs on the FPGAs. Literature suggests several ways to address this drawback. Usmani *et al.* 2018 replaced the shift registers with a couple of LUT and flip-flop performing as a NOT gate [21]. By implementing a couple of a flip-flop and a LUT instead of a shift register that can only be implemented on SLICEM, the number of SLICEMs is not a restriction factor in the PUF design. They could improve the uniformity and uniqueness of Anderson PUF by changing its structure, such as adding some multiplexers at the output of the last level multiplexer and putting filter stages to control the glitch width. These extra multiplexers act as a node to manipulate the width of glitch, thereby the unstable responses can be detected by changing the numbers of these multiplexers, and the detected responses will be omitted from the final responses. Concerning adding the different values of delay helps with improving reliability.

Barbareschi *et al.* [22] search on the resistance of PUF to voltage variations in 2016. They showed that changing supply voltage values leads to change the Anderson PUF responses dramatically. They found that the response of each 1-bit PUFs will change if the supplied value exceeds a threshold, and each of 1-bit PUFs has a spatial threshold.

The other improved Anderson PUF introduced by Hou *et al.* [23] in 2019. They added a linear-feedback shift register (LFSR) to the Anderson PUF to propose a new strong PUF that can be reconfigured. The proposed method in [23] used an LFSR and an Anderson PUF in parallel. The Anderson PUF provides the required seeds of LFSR; as a consequence, the implemented LFSR that is a especial LFSR for each FPGA provides unique unclonable responses after running a fixed number of cycles. The proposed method improves the uniformity and uniqueness of Anderson PUF.

The other design proposed by Huang and Li [24] is to integrate a pair of multiplexers with the Anderson PUF to improve the randomness of PUF. This method used these multiplexers to combine the challenge ability to Anderson PUF. These extra multiplexers act as the select switches to choose a pair of multiplexers to change the delay line's length. The main drawback of Anderson PUF still exists in this PUF; concerning this point, the limited number of SLICEM is a restriction factor in design.

III. PUF DESIGN METHODOLOGY

This section proposes a novel structure of Anderson PUF while addressing the main negative point of the original Anderson PUF and improving its security and performance features. The novel architecture tries to achieve two primary goals. First of all, decreasing the use of SLICEM negatively impacts the PUF design, and the second goal is to implement Anderson PUF on a low-cost FPGA such as Spartan 6. Concerning the different physical characteristics of this kind of FPGAs that are 45 nm FPGAs compared with the Virtex-5 that is used in the original Anderson PUF is different, regard to the point we proposed a new design that can address the required needs. Anderson PUF is considered as a delay-based PUF in which modifying/improving the delay path that is emerged in Anderson PUF has a fundamental role to generate better responses. Equation (1) represents the relation between the emerged delay times in each part, where t_A is the required time that the generated numbers from shift register A detect on the select port of multiplexer and the value of the output port changes, and also t_B is the required time to transfer the generated numbers of shift register B to the corresponding multiplexer's select port and to change the value of its output port. The final element of this equation is t_{CHAIN} that is the required time for a signal to transfer throughout the path between these two multiplexers.

$$Glitch = \Delta t = t_A - (t_{CHAIN} + t_B) \tag{1}$$

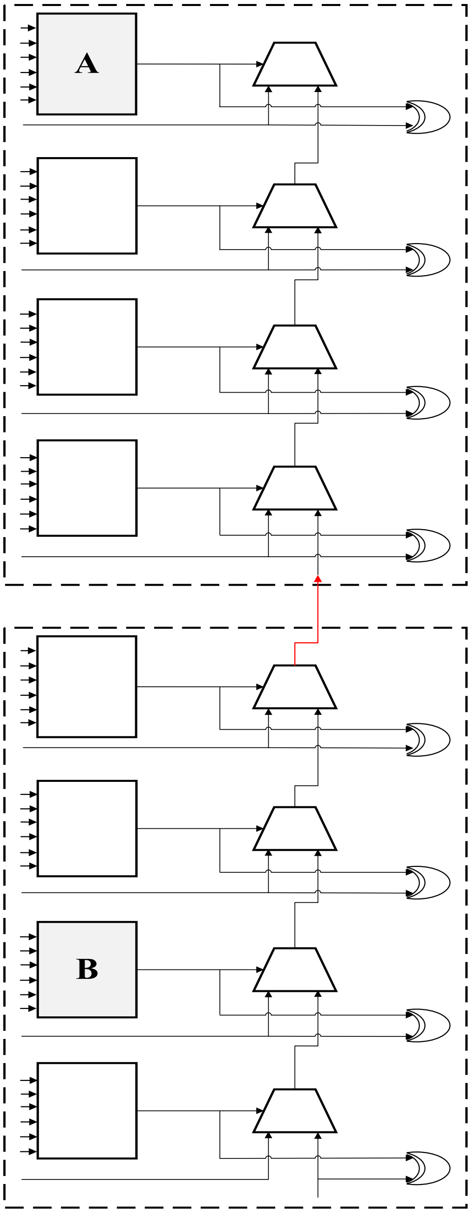

FIGURE 4. The order of the carry chain in the Anderson PUF.

A. DELAY CHAIN

As mentioned before, there are two primary primitives in Anderson PUF that one of them is the carry chain. The carry chain encompasses four multiplexers and four XOR gates. Original Anderson PUF uses these carry chains as a delay chain, so using them in cascade mode to provide the required delay (t_{CHAIN}) needed, as shown in Equation (1) to generate the glitch. The structure of the implemented carry chain in the original Anderson PUF is shown in Figure 4. This figure reveals that the original Anderson PUF ignores the XOR gates while designing its delay lines through the PUF structure. In this paper, these gates are the underlying feature in the proposed design. Unlike the Anderson PUF, the proposed method uses these gates as an alternative way to produce the required delay to generate the proper glitches

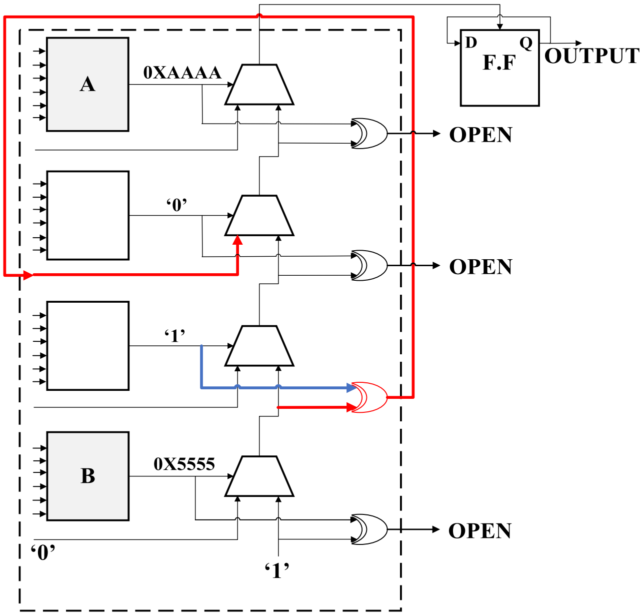

FIGURE 5. The proposed PUF structure.

in its structure. Implementing this design diminishes the required resources, or more precisely, the required SLICEM. This feature provides a golden opportunity to implement the PUF design with less restriction in the number required SLICEM, being a limiting factor.

The proposed structure requires one SLICEM that encompasses one SLICEM and one SLICEX. Figure 5 demonstrates the proposed PUF structure. The goal of this paper is to minimize the number of SLICEMs. Concerning this point, we limit ourself into the available primitives that are implemented in one SLICEM. Fig.1 shows the available hardware units in SLICEM. One of these primitives is XOR gate. After trying and testing the available primitives, the only primitive that meet our required delay was the XOR gate. In the proposed design, the XOR gate connects the multiplexer output in the previous stage to the upper multiplexer input, and by sending the signal through this suggested path, the required delay can be generated. The multiplexers location plays a role as adjusting nodes; by changing their position, the value of t_{CHAIN} will change. In the experimental test, to estimate the optimal length of the delay chain and the position of each element, the value of t_A and t_B are assumed to be equal. Figure 5 reveals the best position for each element. The vital point is to select the correct XOR gate. In practice, we have one choice to select an appropriate couple of XOR gate and multiplexer because selecting the XOR gate and multiplexer in different stages lead to change the Hamming weight in output responses.

Figure 5 shows a carry chain; it reveals that the first XOR gate is before the first multiplexer that is connected to the LUT, and the last multiplexer is connected to the shift register. Regarding these conditions; the only selected pair is the second and third XOR gates and the second and third multiplexers. In this paper, using the second XOR gate and

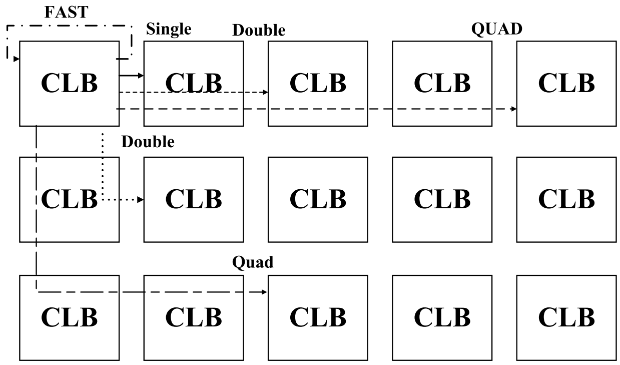

FIGURE 6. The different types of paths in Spartan-6.

the third multiplexer is suggested. The only couple can meet the delay condition by trying, and testing method that leads to approximately generate 50% Hamming weight is this couple.

The interconnection connects the XOR output of the carry chain to the multiplexer's input is called fast interconnects by Xilinx. Figure 6 shows several types of existing interconnects in Spartan-6 family FPGA. The different categories of interconnects are fast-interconnects, single interconnects, double interconnects, and quad interconnects.

Although this interconnect is the fastest interconnect among the others, it is still slower than the connection in the carry chain. Therefore, it can be considered a suitable candidate to replace it with another carry-chain. The connection is connected by the ISE software between the two selected units, the second XOR gate output and the third multiplexer input. This path can meet our condition to add incalculable delay to our circuit that leads to approximately generate 50 % of Hamming weight. The other fundamental advantage of using the XOR gates, which is as vital as other cases, is to increase PUF responses' entropy. This feature can also increase the unpredectibility of the PUF responses and its resistance against modeling attacks by increasing of PUF design complexity.

B. POSITIONING OF THE ELEMENTS

As mentioned before, the proposed design elements play a fundamental role in providing the required delays. The location of the XOR gate and the multiplexer is discussed in the previous section. The other elements that have to be considered are shift registers and flip-flop at the last level.

The location and the distance between the shift register drastically impact on the final result. In fact, by increasing the distance between these shift registers, the value of t_{CHAIN} will grow. Consequently, the change in t_{CHAIN} affect the value of Hamming weight and leads to the balance of 1s and 0s getting away from the ideal value. Moreover, the optimal location of the shift registers places at the first and last LUTs in the same SLICE according to experimental tests.

The last (but not least) term is the flip-flop in the final level generating the response of 0 or 1. This flip-flop and its route are couples generating the final responses. The first part of this couple is a flip-flop. Flip-flop is initialized logic 0, so the default output answer is 0. It has its output Q feedback to its input D. The preset port of this flip-flop is connected to the output of the carry chain. When the width of the output signal of the carry chain has adequate width to be recognizable to the flip-flop, the output Q will be the response of 1. Moreover, the feedback will save this response as the final answer until the PUF is reset. The second part is the routing path acting as a low pass filter. In other words, if the width of the signal is too broad, the routing path damping out the high-frequency pulse, so the final answer will save the previous status. This feature is caused by the resistive and capacitive loading on the routing path, thereby routing path acts as a low pass filter. According to the description provided, the length of the routing path and location of the flip-flop affect the probability of the final answer.

TABLE 2. The role of initial values on hamming weight.

Register A	Register B	Number of 1s	Percentage of 1s
AAAA	5555	62	48.43
5555	AAAA	61	47.65
AAAA	1111	128	100
1111	AAAA	53	41.4
2222	5555	56	43.75
5555	2222	128	100
5555	8888	128	100
8888	5555	50	39.06
4444	AAAA	53	41.4
AAAA	4444	128	100

C. INITIAL VALUES

The final modification impacting on the final output of PUF is to set appropriate initial values to the shift registers. The shift registers are used in the Anderson PUF, and the proposed PUF has 16 bits length. Different FPGA types require different initial values. The original Anderson PUF used 0 × 5555 and 0xAAAA as the shift registers initial values, while these values generate inappropriate answers in other FPGAs [24]. In our experimental test, the optimal initial values that address the requirements are 0XAAAA for the shift register A, and 0 × 5555 for shift register B. Table 2 shows the average number of 1s in the 128-bit responses for different initial values of the shift registers. This table reveals that the order of the initial values plays a significant role in establishing a good uniformity of the PUF responses.

IV. SECURITY AND PERFORMANCE ANALYSIS

In this paper, the proposed 128-bit PUFs are implemented on FPGA. The experiments are carried out on ten Xilinx Spartan-6 XC6SLX9 FPGA devices, supplied with 45 nm manufacturing technology. The boards are used in this paper

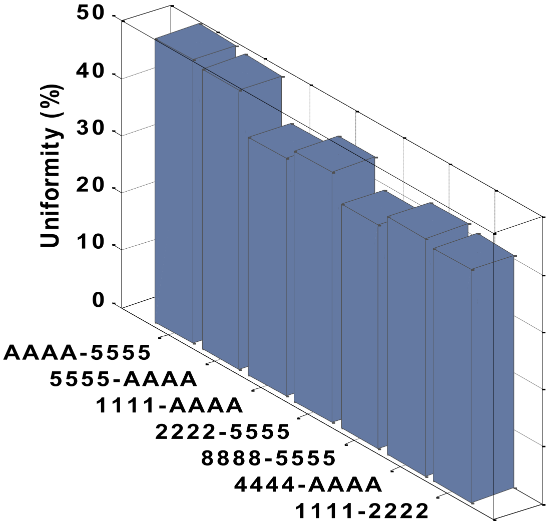

FIGURE 7. Uniformity of the proposed PUF for the different initial values.

have a 50 MHz clock signal. The provided responses are read by ChipScope in ISE 14.7. The experimental tests are done on ten FPGA boards with the same features, and each of FPGAs is divide into two disjoint sections for each PUF. Our 128-bit PUF uses 348 slices that are 8% of all slices that are in the Spartan-6 XC6SLX9 FPGA device. The proposed 1-bit PUF uses two 16-bit shift registers implementing into one SLICEM, and needs one carry chain to implement the required delay line; and finally, the final level flip-flop should be implemented into adjacent SLICEX that is in the upper CLB. The other factor is the initial values of the shift registers that should be considered as a factor that is as crucial as other factors. These values might be different from type to type of FPGAs; this difference is due to the manufacturing technology of the chips. In what follows, the results of various tests are presented to show how the proposed PUF design meets the quality requirements.

A. UNIFORMITY

In the first experiment, twenty 128-bits proposed PUFs are implemented on the ten same FPGA boards and in two different locations from each. This test is performed with twelve sets of initial values for each of the PUFs. This test is done at a normal temperature around 25°C. The percentage of the average number of 1s in the 128-bits PUF responses per each initial value is shown in Figure 7. According to this figure, the best value of hamming weight (uniformity) is obtained through AAAA and 5555 initial values for shift register A and shift register B, respectively. As we consider this initial value for our PUF, the average uniformity of the proposed PUF structure is 49.14%, which is very close to the ideal value (50%).

B. UNIQUENESS

In the uniqueness test, the proposed structure with the same initial values of shift registers (AAAA-5555) for all PUFs is implemented on ten different boards where each board consists of two disjoint PUFs. The uniqueness of the proposed PUF is calculated by Equation (2) where g is the number of PUF instances, and a is the length of responses (128-bit). The average of the absolute uniqueness of the proposed PUF is 50.89% that is close to the ideal value (50%). Fig. 8 shows the distribution of Hamming distances among the all proposed PUF instances (between different chips or different locations on the same chip).

$$100 * (1 - \frac{2}{g*(g-1)} \sum_{i=1}^{g-1} \sum_{j=i+1}^{g} \frac{HD(r_i, r_j)}{a}) \tag{2}$$

TABLE 3. Average value of uniformity, uniqueness, and reliability of the proposed PUF In normal temperature (25°C).

Uniformity	Uniqueness	Reliability
49.14%	50.89%	91.25%

TABLE 4. Utilization of improved PUF on Xilinx XC6SLX9 FPGA.

Slice logic utilization	Used	Available	Utilization
Number of slice registers	400	11,440	3%
Number of slice LUTs	502	5,720	8%
Number used as logic	400	5,720	6%
Number used as memory	100	1,440	6%
Number of occupied slices	348	1,430	24%

C. RELIABILITY

In this experiment, we evaluate the proposed PUF instances in twelve operating temperatures to measure reliability. To that end, we collect the PUF responses at temperatures of 0°C to 80°C with a 10°C gap. The reference temperature is considered as 25°C. The average bit errors in the 128-bit response of PUF is 8.75, i.e. the reliability of the proposed Anderson PUF is 91.25%. Figure 9 exhibits the average percentage of the Hamming weight and reliability for the proposed PUF responses in each temperature. Although by increasing of temperature the Hamming weight and reliability deviate from their ideal value (50% and 100%, respectively), it has still been able to perform reasonably well. Table 3 lists the average total value of uniformity, uniqueness, and reliability of the proposed PUF in average temperature (25°C).

D. HARDWARE COST

One of the primary features of Anderson PUF that is maintained in this paper is that it needs low resources of FPGA. Improving the features of the original Anderson PUF and diminishing the required resources are the goals being achieved in this paper simultaneously. The proposed PUF structure eliminates the need for one more SLICEM, and

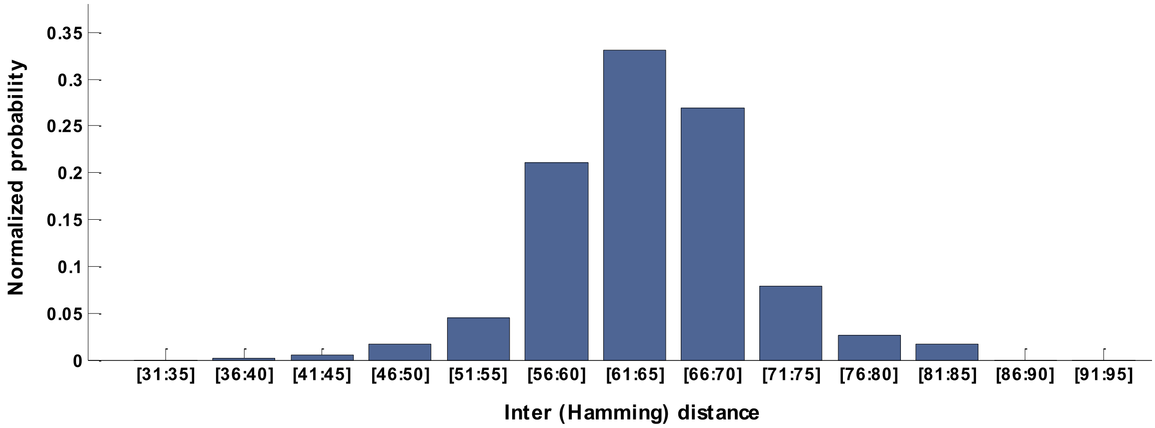

FIGURE 8. The distribution of Hamming distances among 20 128-bit proposed PUF responses.

TABLE 5. Comparison between different PUFs in required hardware resources.

Slice logic utilization	Our PUF	Scheme [21] A	Scheme [21] B	Scheme [23]	Scheme [19]	Scheme [11]	Scheme [25]
Number of slice registers	400	NA	NA	NA	268	NA	1157
Number of slice LUTs	502	1020	510	130	1076	NA	1327
Number used as logic	400	NA	NA	NA	819	NA	1152
Number used as memory	100	NA	NA	128	256	NA	136
Number of occupied slices	348	NA	NA	256	426	532	672
Number of flip-flop	300	1530	765	NA	140	NA	NA

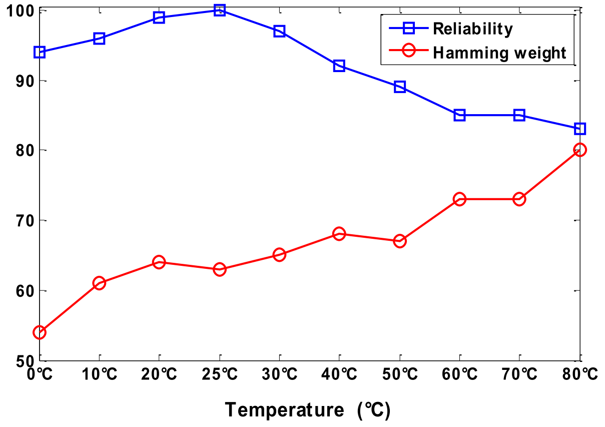

FIGURE 9. Average percentage of the Hamming weight and reliability in different temperatures.

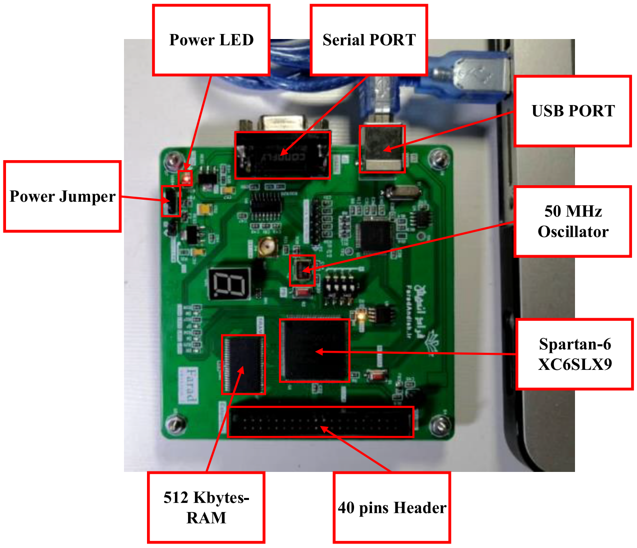

FIGURE 10. The experimental setup of the proposed scheme.

utilizes the XOR gate, thereby decreases the restriction about the lack of number of SLICEM. In what follows, the used resources of proposed PUF architecture are shown in Table 4. Furthermore, Table 5 presents a hardware efficiency comparison between the proposed PUF in this paper, three other versions of Anderson PUF, and two other weak PUFs including RO PUF [11] and SR Latch PUF [25]. Figure 10 shows the experimental setup of the proposed scheme.

V. CONCLUSION

One of the primary goals of PUFs is to provide cryptographic keys when they are embedded in the corresponded devices. Anderson PUF is an appropriate choice to be implemented on the hardware with the minimum required resources. The primary drawback of Anderson PUF is highly dependent on the SLICEM in FPGAs. SLICEM contains a smaller share of hardware resources compared to other types of SLICEs. Concerning this negative point, this dependency makes a restriction for the designers. The proposed structure in this paper improves the characteristic of Anderson PUF and diminishing the number of required SLICEM simultaneously. Using the XOR gates in the carry chain is the underlying solution to address the drawbacks of Anderson PUF. The XOR gates in the carry chain provide a situation to decrease the number of required carry chains to provide the required delay time (glitch) and increase the entropy of PUF. In this paper, several nodes are introduced to adjust the delay of each element to achieve the required delay. Several tests are performed to evaluate the features of the proposed Anderson PUF. In the experimental tests, the 49.14% uniformity of responses has been obtained in the uniformity test. The result of the uniqueness tests shows 50.89 % inter distances among the responses in the normal situation (25°C) that it reveals the uniqueness of the proposed PUF is entirely near the ideal value. The reliability test under 0°C-80°C temperature shows the 91.25% reliability. Improving the reliability of the proposed PUF can be considered as future works. Finally, the required SLICEM in the proposed design is half of the number of SLICEM in the original Anderson PUF; thereby, the required resources to implement the Anderson PUF are reduced, and also this PUF can be implemented on the cheap FPGAs with minimum resources.

REFERENCES

[1] J. Obermaier and V. Immler, "The past, present, and future of physical security enclosures: From battery-backed monitoring to PUF-based inherent security and beyond," *J. Hardw. Syst. Secur.*, vol. 2, no. 4, pp. 289–296, Dec. 2018.

[2] O. A. Ibrahim and S. Nair, "Cyber-physical security using system-level PUFs," in *Proc. 7th Int. Wireless Commun. Mobile Comput. Conf.*, Jul. 2011, pp. 1672–1676.

[3] M. Kaveh, D. Martín, and M. R. Mosavi, "A lightweight authentication scheme for V2G communications: A PUF-based approach ensuring cyber/physical security and identity/location privacy," *Electronics*, vol. 9, no. 9, p. 1479, Sep. 2020.

[4] Y. Gao, S. F. Al-Sarawi, and D. Abbott, "Physical unclonable functions," *Nature Electron.*, vol. 3, no. 2, pp. 81–91, 2020.

[5] M.-D. Yu, M. Hiller, J. Delvaux, R. Sowell, S. Devadas, and I. Verbauwhede, "A lockdown technique to prevent machine learning on PUFs for lightweight authentication," *IEEE Trans. Multi-Scale Comput. Syst.*, vol. 2, no. 3, pp. 146–159, Jul. 2016.

[6] Y. Liu, Y. Xie, C. Bao, and A. Srivastava, "A combined optimization-theoretic and side-channel approach for attacking strong physical unclonable functions," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 26, no. 1, pp. 73–81, Jan. 2018.

[7] M. Kaveh, S. Aghapour, D. Martin, and M. R. Mosavi, "A secure lightweight signcryption scheme for smart grid communications using reliable physically unclonable function," in *Proc. IEEE Int. Conf. Environ. Elect. Eng. Ind. Commercial Power Syst. Eur.*, Jun. 2020, pp. 1–6.

[8] P. Gope and B. Sikdar, "Privacy-aware authenticated key agreement scheme for secure smart grid communication," *IEEE Trans. Smart Grid*, vol. 10, no. 4, pp. 3953–3962, Jul. 2019.

[9] M. Kaveh and M. R. Mosavi, "A lightweight mutual authentication for smart grid neighborhood area network communications based on physically unclonable function," *IEEE Syst. J.*, vol. 14, no. 3, pp. 4535–4544, Sep. 2020.

[10] J. Delvaux, "Machine-learning attacks on PolyPUFs, OB-PUFs, RPUFs, LHS-PUFs, and PUF–FSMs," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 8, pp. 2043–2058, Aug. 2019.

[11] R. Maes, *Physically Unclonable Functions: Constructions, Properties and Applications*. Berlin, Germany: Springer-Verlag, 2013.

[12] C. Böhm and M. Hofer, *Physical Unclonable Functions in Theory and Practice*. New York, NY, USA: Springer-Verlag, 2012.

[13] S. S. Zalivaka, A. A. Ivaniuk, and C.-H. Chang, "Reliable and modeling attack resistant authentication of arbiter PUF in FPGA implementation with trinary quadruple response," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 4, pp. 1109–1123, Apr. 2019.

[14] M. Delavar, S. Mirzakuchaki, and J. Mohajeri, "A ring oscillator-based PUF with enhanced challenge-response pairs," *Can. J. Electr. Comput. Eng.*, vol. 39, no. 2, pp. 174–180, 2016.

[15] M. Huang and S. Li, "A delay-based PUF design using multiplexers on FPGA," in *Proc. IEEE 21st Annu. Int. Symp. Field-Program. Custom Comput. Mach.*, Apr. 2013, p. 226.

[16] J. Zhang, Y. Lin, Y. Lyu, and G. Qu, "A PUF-FSM binding scheme for FPGA IP protection and pay-per-device licensing," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 6, pp. 1137–1150, Jun. 2015.

[17] C. Herder, M.-D. Yu, F. Koushanfar, and S. Devadas, "Physical unclonable functions and applications: A tutorial," *Proc. IEEE*, vol. 102, no. 8, pp. 1126–1141, Aug. 2014.

[18] G. E. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in *Proc. 44th ACM/IEEE Design Autom. Conf.*, Jun. 2007, pp. 9–14.

[19] J. H. Anderson, "A PUF design for secure FPGA-based embedded systems," in *Proc. 15th Asia South Pacific Design Automat. Conf. (ASP-DAC)*, Jan. 2010, pp. 1–6.

[20] D. Suzuki and K. Shimizu, "The glitch PUF: A new delay-PUF architecture exploiting glitch shapes," in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst.*, 2010, pp. 366–382.

[21] M. A. Usmani, S. Keshavarz, E. Matthews, L. Shannon, R. Tessier, and D. E. Holcomb, "Efficient PUF-based key generation in FPGAs using per-device configuration," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 27, no. 2, pp. 364–375, Feb. 2019.

[22] M. Barbareschi, P. Bagnasco, and A. Mazzeo, "Supply voltage variation impact on anderson PUF quality," in *Proc. 10th Int. Conf. Design Technol. Integr. Syst. Nanosc. Era (DTIS)*, Apr. 2015, pp. 1–6.

[23] S. Hou, Y. Guo, and S. Li, "A lightweight LFSR-based strong physical unclonable function design on FPGA," *IEEE Access*, vol. 7, pp. 64778–64787, 2019.

[24] M. Huang and S. Li, "A delay-based PUF design using multiplexer chains," in *Proc. Int. Conf. Reconfigurable Comput. FPGAs (ReConFig)*, Dec. 2013, pp. 1–6.

[25] D. Yamamoto, K. Sakiyama, M. Iwamoto, K. Ohta, M. Takenaka, and K. Itoh, "Variety enhancement of PUF responses using the locations of random outputting RS latches," *J. Cryptograph. Eng.*, vol. 3, no. 4, pp. 197–211, Nov. 2013.

ARMIN LOTFY received the B.S. degree from the Babol Noshirvani University of Technology (NIT), Babol, Iran, in 2015, and the M.S. degree in electronic engineering from the Iran University of Science and Technology (IUST), Tehran, Iran, in 2018. His research interests include signal processing, FPGA design, and physical unclonable functions (PUFs).

MASOUD KAVEH received the B.Sc. degree in electrical engineering from the Babol Noshirvani University of Technology, Iran, in 2014, and the M.Sc. degree in electrical engineering from the Marine Science University of Nowshahr established in collaboration with the Iran University of Science and Technology (IUST), Iran, in 2016. He is currently pursuing the Ph.D. degree with IUST. His research interests include physically unclonable functions (PUFs), ASIC and FPGA design, cryptographic protocols, and machine learning.

DIEGO MARTÍN received the B.Sc. degree in computer engineering, the M.Sc. degree in computer science, and the Ph.D. degree in 2012 from the Department of Informatics, Carlos III University of Madrid, Spain. He is currently a Lecturer with the Department of Telematics, Technical University of Madrid (UPM). His main research interests, within the GISAI Group, UPM, include the Internet of Things, cyber physical systems, physically unclonable functions, blockchain, knowledge management, information retrieval, and research methods.

MOHAMMAD REZA MOSAVI received the B.S., M.S., and Ph.D. degrees in electronic engineering from the Iran University of Science and Technology (IUST), Tehran, Iran, in 1997, 1998, and 2004, respectively. He is currently a Faculty Member (Full Professor) with the Department of Electrical Engineering, IUST. He is the author of more than 400 scientific publications in journals and international conferences in addition to 11 academic books. His research interest includes circuits and systems design. He is also the Editor-in-Chief of *Iranian Journal of Marine Technology* and an Editorial Board Member of *Iranian Journal of Electrical and Electronic Engineering*.