LDCPUF: A Novel FPGA-based Physical Unclonable Function with Ultra-low Hardware Cost

Luo Zufeng\textsuperscript{1,2}, and Yuan Guoshun\textsuperscript{1,a)

Abstract The physical unclonable function based on ring oscillator (RO PUF) is a traditional design suitable for FPGA implementation, but such designs have the disadvantage of low hardware efficiency. This article proposes a new FPGA-based ring oscillator PUF, called loop delay configurable (LDC) PUF. The construction of LDC PUF relies on configurable delay units (CDUs). An LDC PUF configured with \( n \) CDUs can generate \( 2^n \times (2^n - 1) \) response bits. Additionally, we apply the programmable delay lines technology to enhance the reliability of LDC PUF. Compared with the traditional RO PUFs, the LDC PUF has good uniqueness (48.52\%) and reliability (96.91\%), and most importantly, it has ultra-low hardware cost.

key words: RO PUF, FPGA, PDL, hardware cost
Classification: Integrated circuits

1. Introduction

With technology developing, the iteration of the digital chip is getting faster and faster. Therefore, researchers increasingly favor Field programmable gate array (FPGA) because of their flexible and reconfigurable characteristics. Nevertheless, FPGA-based designs face severe security threats such as counterfeiting devices and hardware Trojans\textsuperscript{[1, 2]}. In order to deal with these threats, researchers propose to apply physical unclonable function (PUF) to provide adequate protection for FPGAs\textsuperscript{[3, 4, 5]}. PUF is a hardware cryptographic primitive that exploits device mismatches in the manufacturing process to generate unpredictable and reliable identifiers. Many PUF structures have been proposed, among which the digital PUF is the most popular for researchers. The arbiter PUF (APUF)\textsuperscript{[6, 7, 8]} is one of the first digital PUFs to be studied, generating a PUF response by comparing the delays on two identical paths. Since it is extremely challenging to implement the APUF on FPGAs to ensure the high symmetry between delay paths, the uniqueness of the APUF implemented on FPGAs is unsatisfying\textsuperscript{[9, 10]}. After the APUF, many other PUFs appeared one after another, such as SRAM PUF\textsuperscript{[11, 12]}, Butterfly PUF\textsuperscript{[3]}, DFF PUF\textsuperscript{[13, 14]}, Glitch PUF\textsuperscript{[15, 16, 17]}. However, most of them are not adaptive to FPGAs. Butterfly PUF has extremely high requirements on wiring symmetry just like APUF, and SRAM PUF is difficult to achieve due to the power-on reset of FPGAs. DFF PUF and Glitch PUF can be implemented on FPGA, but the sampling process consumes too many resources.

Suh proposed the earliest PUF design based on ring oscillators, named RO PUF\textsuperscript{[18]}. RO PUF is simple, and the hard macro technology makes it very easy to implement on FPGAs\textsuperscript{[19]}. In [18], to improve the reliability of RO PUF, the RO pairs selection method of 8-1 mask is used, which makes the RO PUF with high hardware cost occupy resources more. In order to decrease hardware cost and improve efficiency, researchers began to study configurable RO PUF (CRO PUF)\textsuperscript{[20, 21, 22, 23, 24, 25, 26]}. The design idea of CRO PUF is similar to the 8-1 mask method of RO PUF, but it screens frequency by modifying the delay inside ring oscillator pairs. Due to the analogous mechanism, normal CRO PUFs can only slightly improve hardware efficiency, which means resources occupation on FPGA is still high.

This article aims at the above problems and proposes an FPGA-based RO PUF with ultra-low hardware cost. Specifically, the main contributions are the following three points:

1) A configurable delay unit (CDU) is proposed, and a novel RO PUF with ultra-low hardware cost is designed based on the CDU, which is named Loop Delay Configurable PUF (LDC PUF). Compared to the traditional CRO PUF where challenge-response pairs (CRPs) grow linearly with resources, the LDC PUF can generate more CRPs with exponential efficiency.

2) Employing programmable delay line (PDL) technology based on look up table (LUT) to optimize the reliability of the LDC PUF, experimental data show that it improves the reliability by 0.73\% without post-processing and without adding any resources.

3) The proposed PUF is implemented on Xilinx Spartan-6 FPGA, and the experimental results show that the method has good uniqueness (48.52\%, ideal 50\%) and good reliability (96.91\%, ideal 100\%), and achieves hugely high hardware efficiency (generating \( 1.37 \times 10^{11} \) CRPs using 20 CLBs).

\textsuperscript{1}Institute of Microelectronics of Chinese Academy of Sciences, Beijing 100029, China
\textsuperscript{2}University of Chinese Academy of Sciences, Beijing 100049, China
\textsuperscript{a) yuanguoshun@ime.ac.cn

DOI: 10.1587/elex.19.20220246
Received May 24, 2022
Accepted July 13, 2022
Publicized July 21, 2022
The rest of this article is organized as follows: Section 2 introduces the structure of the LDC PUF, including the design of the CDU unit, the structure and the model of LDC PUF, and the LUT-based PDL technology; section 3 presents the experimental design method and experimental results; section 4 summarizes the research work of this article.

2. Proposed PUF

2.1 The structure of LDC PUF

Although the traditional RO&CR0 PUFs are easy to implement due to their simple structure, the implementation requires too many resources, and their efficiency in generating CRPs is limited. APUF has extremely high requirements on the symmetry of the layout, which inconveniences the implementation. So we propose to combine the characteristics of RO&CRO PUF and APUF. First, we extract the basic structure of LDC PUF that we call configurable delay unit(CDU) from Eq.(1). Furthermore, how the delay unit is configurable will be explained in Section 2.3.

\[
Y = CX + \overline{C}X. \tag{1}
\]

Eq.(1) is a boolean function. Y always equals X no matter what the value of C is. Nonetheless, the control factor C affects how quickly X transfers its value to Y. Fig. 1 can be a good help in comprehending this. We can see that when \( C = 0 \), the value of X passes through Path 1, and when \( C = 1 \), it passes through Path 2. Due to manufacturing variation, the delays of Path 1 and Path 2 cannot be exactly the same.

![CDU implemented with 6-input LUT](image)

Configurable logic blocks(CLBs) are the primary resources on Xilinx FPGAs. Each CLB contains several SLICEs, and each SLICE contains 4 LUTs. To reduce the uncertainty of the medium and long wire routed in FPGAs, we implement each logic gate of CDU with a 6-input LUT so that each delay unit is composed of 4 LUTs, which can just be put into one SLICE. This operation can better ensure the similarity of the two delay paths.

Inspired by [27, 28], we decided not to use the simultaneous comparison of the same structural unit like other RO PUFs but adopt the strategy of time-division measurement. As shown in Fig. 2, for a set of challenges input to N CDUs, we first measure the oscillation frequency of the ring oscillator under the influence of a series of challenges and store it in the register of the Frequency Measuring block. Then we repeat measurement in another series of challenges for contrasting. After two measurements, the two frequency values are sent to the Response Generation module, where they are compared to generate the response output.

2.2 The model of LDC PUF

The LDC PUF proposed in this article is a delay-based PUF whose entropy is extracted from the uncertainty during LUT fabrication, discussed in detail below. The generation of the LDC PUF response comes from the different frequencies generated by the two series of challenges, which is actually the delay difference of the ring oscillator caused by diverse challenges. The delay composition of the ring oscillator in LDC PUF can be described by Eq.(2):

\[
d_{\text{loop}} = d_{\text{wire}} + d_{\text{and}} + d_{\text{inv}} + \sum_{i=1}^{n} d_{\text{CDU},i}. \tag{2}
\]

Where \( d_{\text{and}} \) is the delay of the AND gate, \( d_{\text{inv}} \) is the delay of the inverter, \( d_{\text{wire}} \) is the delay of the routing wires, and \( d_{\text{CDU},i} \) is the delay of the \( i \)th CDU. We can get the PUF response by simply making a subtraction as Eq.(3) and Eq.(4) show:

\[
\Delta_d = d_{\text{loop}} - d_{\text{loop}}' = \sum_{i=1}^{n} d_{\text{CDU},i} - \sum_{i=1}^{n} d_{\text{CDU},i}'. \tag{3}
\]

\[
\text{Response} = \begin{cases} 0 & \Delta_d > 0, \\ 1 & \text{otherwise}. \end{cases} \tag{4}
\]

Maiti proposed a delay model of the ring oscillator in [20], which divides the loop delay into three parts: the average delay of a specific structure, the random delay caused by the manufacturing discrepancy, and the system variance delay caused by different positions of ring oscillators on the same chip. As shown in Eq.(3), credit to the self-comparison design method, the only dependency left for response generation is the difference of \( d_{\text{CDU}} \) under diverse challenges, which is the manufacturing discrepancy between LUTs. Therefore, in the LDC PUF, the system variance is effectively reduced or even eliminated.

2.3 Programmable delay lines in CDU

LUTs are not only one of the basic units on FPGAs implementing the boolean function, but also the main delay configurable units. A LUT consists of SRAM cells and tree-structured multiplexers. The SRAM cells store preset function values, and the multiplexers select specific SRAM cells
to the output ports. A LUT can be instantiated as an arbitrary logic function by configuring the SRAM cells. When the implemented function input is less than the largest input port of the LUT, the unoccupied input terminals will not change the configured logic function. However, they will affect the output latency. Fig. 3 shows how PDL affects the delay of LUT using a schematic diagram of a 3-input LUT implementing an AND gate.

We can see that the value of A3 does not affect the expression $O = A1 \cdot A2$, but does determine which path the data is output from (solid or dashed line). Therefore, we propose to improve the reliability of the LDC PUF by configuring the LUT delay. The generation of the response bits depends on the difference between the delays of two LUTs. When the difference is slight, it is easy to be influenced by the environment and change the responses, thus weakening the reliability of the PUF. If we make the difference more conspicuous by the PDL technology, the reliability of the LDC PUF can be effectively improved. We will demonstrate this experimentally in the next section.

3. Experimental results and analysis

To check whether the performance of the LDC PUF is acceptable, we implemented the design on a Xilinx Spartan-6 XC6SLX150 FPGA (abbreviated as LX150). Due to limited conditions, we can only conduct experiments on one piece of LX150 chip. Fortunately, the LDC PUF occupies very few resources, and the resources on the LX150 are sufficient. So we partition the chip into 24 partitions through Xilinx’s PlanAhead tool and implement 24 LDC PUFs. Each LDC PUF is configured with 20 CDUs, which can generate $2^{20}$, or 1,048,576 frequencies. The number of responses produced using these frequencies is enormous, and by taking any two frequencies to produce a response, we have $2^{2N-1} - 2^{N-1} \approx 5.497 \times 10^{11}$ choices taking $N = 20$. In our design, the response width of the LDC PUF is configured as 128 bits. We evaluate LDC PUF by three metrics: unit response cost, uniqueness, and reliability. After that, a comparison with other related works based on these metrics will be displayed.

3.1 Unit response cost

Different PUF designs can generate different numbers of CRPs under the same resource constraints. If a design yields more CRPs while occupying fewer resources, we will consider the design a high hardware efficiency. The CLBs are Xilinx FPGAs’ primary hardware resources, and the ring oscillators’ implementation generally takes CLB as the basic unit. However, the placement and routing (P&R) deviation between ring oscillators will affect the quality of the PUF response. Fortunately, the hard macro replication technology can make the placement and routing the same within the CLBs. Therefore, to ensure the consistency of P&R, each ring oscillator should better be implemented in a single CLB. Inspired by Zhang et al.[23, 30], we introduce the criteria of unit response cost (URC) to describe the hardware efficiency of PUF designs, which considers the number of CLBs required to form ring oscillators as Eq.(5) shows.

$$URC = \frac{N_{CLB}}{N_{Rbit}} \times 100\%.$$  

Where $N_{CLB}$ is the number of CLBs occupied by the ring oscillators, and $N_{Rbit}$ is the number of available responses produced by these CLBs. Each CLB in Xilinx Spartan-6 FPGAs contains three multiplexers, eight LUTs, sixteen flip-flops, and miscellaneous logic. Let us first analyze how these CLBs construct ring oscillators, using Suh’s design as an example. In Suh’s PUF design[18], the ring oscillator consists of an odd number of inverters and an AND gate, so we can implement one ring oscillator using a single CLB. Then in the case of applying n CLBs, the design of Suh can generate $\binom{n}{2}$ CRPs, from which we conclude that its URC is $n/\binom{n}{2}$. Based on the analysis of Suh’s PUF, we can easily deduce the URCS of other works.

Maiti implemented the CRO PUF design by inserting multiplexers into the inverter chain in Suh’s ring oscillator[20]. Limited to the resources in a CLB, the inverters of CRO are cascaded up to three stages: eight configurable frequencies. It is worth noting that the structure of CRO could have yielded far more CRPs. However, in pursuit of the ultimate reliability, Maiti applied a frequency screening method that would result in a substantial decrease in available CRPs, making it only 8 times more efficient than Suh’s design. So the URC of CROPUF is $n/(8\binom{n}{2})$. Xin utilized the flip-flops in CLBs to extend Maiti’s CRÖ structure[21], enabling the new CRO to generate 256 different configurations, so the URC of Xin’s PUF is $n/(256 \cdot \binom{n}{2})$.

Gao proposed a highly flexible RO PUF[22]. Unlike Maiti’s work, Gao connected one output of a multiplexer to an inverter and the other to a buffer, and he called the structure delay unit. We implement the delay unit within a single CLB; then, for a ring oscillator composed of $k$ CLBs, it has $\sum_{i=0}^{k} \binom{k}{i}$ kinds of configurable frequencies. Gao’s design requires paired ROs, so for a design that occupies $n = 2k$ CLBs, its URC is $n/(\sum_{i=0}^{k} \binom{k}{i})^2$. The XCRO PUF[23] proposed by Zhang has certain similarities with Gao’s work. Zhang chose XOR gates instead of inverters to achieve the same function.
as Gao. We can implement up to seven XOR gates and one AND gate in each CLB, so each XCRO has \( \sum_{i=0}^{7} \binom{7}{2i+1} = 64 \) configurable frequencies. Since Zhang’s design adopted a response extraction strategy similar to Maiti’s, its URC is \( n/(64 - \binom{n}{2}) \).

For the LDC PUF proposed in this article, the CDU is the basis for composing the ring oscillator. The implementation of each CDU needs four LUTs so that we can implement each CDU with one CLB; then, with \( n \) CLBs occupied, our ring oscillator can generate \( 2^n \) frequencies. Besides, the inverter and AND gate also need one CLB, so the number of CLBs occupied should be \( n + 1 \). The response bits generated with \( n + 1 \) CLB is \( \binom{n}{2} \). To sum up, the unit response cost of LDC PUF is shown as Eq.(6):

\[
URC_{LDCPUF} = \frac{n + 1}{2^{n-1}(2^n - 1)} \times 100\%.
\]  

Eq.(6) shows the number of different bits between two responses, as Eq.(7) depends on the Hamming Distance, which represents the capability of any two PUFs to generate different CRPs under the same challenges. It indicates the ability of any two PUFs to generate different CRPs under the same challenges. It depends on the Hamming Distance, which represents the number of different bits between two responses, as Eq.(7) shows:

\[
HD(R_i, R_j) = \sum_{l=1}^{m} R_i[l] \oplus R_j[l].
\]

The uniqueness metric indicates the ability of any two PUFs to generate different CRPs under the same challenges. It depends on the Hamming Distance, which represents the number of different bits between two responses. We call \( HD(R_i, R_j) \) as Inter-chip Hamming Distance. Now the uniqueness of LDC PUF can be described as Eq.(8):

\[
Uniq. = \frac{2}{k(k-1)} \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \frac{HD(R_i, R_j)}{m} \times 100\%.
\]  

Here, the blue bars reveal the experimental results, and the solid curve fits the experimental results according to the binomial distribution probability density function with parameters \( n = 128 \) and \( p = 0.4852 \). The dotted curve (purple) represents the binomial distribution with parameters \( n = 128 \) and \( p = 0.5 \), which is the ideal distribution. We can see that there is only a slight difference between the fitted distribution and the ideal distribution, indicating that the LDC PUF has good uniqueness.

3.3 Reliability

The construction of the reliability metric depends on the Hamming Distance between the responses of the same chip to the same challenges in different working environments, which is defined as Eq.(9):

\[
Rel. = \left(1 - \frac{1}{n} \sum_{i=2}^{n} \frac{HD(R_{i,1}, R_{i,i})}{n}\right) \times 100\%.
\]  

Where \( m \) is the bit width of response \( R_i \) and \( R_j \). We call \( HD(R_{i,1}, R_{i,i}) \) as Intra-chip Hamming Distance.
To analyze the effect of PDL technology on the reliability of the LDC PUF, we count the responses generated under different configurations of irrelevant bits of 6-input LUTs at room temperature (about 20°C). According to [29], the latency of LUTs is minimal if we set all irrelevant bits to 0. In contrast, we get the highest latency if all the bits are set to 1. We need to find out how the irrelevant bits affect the latency difference of the LUTs implemented as AND gates. Therefore, we uniformly change LUT irrelevant bits in all DCUs. Mark the number of irrelevant bit 1 of 6-input LUT B (see Fig. 1) as $i$, and the number of irrelevant bit 1 of 6-input LUT D as $j$. Then classify according to the value of $i - j$, and define that the value of $i - j$ is ”-0” when $i = j = 0$ while ”+0” when $i = j = 4$. Fig. 6 shows the experimental results. Compared with not using PDL technology at all, the reliability is improved by up to 0.73%.

![Fig. 6. Reliability varies with irrelevant bits configuration](image)

Referring to the results of the PDL experiments, we select the most stable irrelevant bit configuration to conduct the reliability experiment. We placed the FPGA in a constant temperature experimental box set at 25°C to obtain the reference responses and gradually increased the temperature to 85°C in a gradient of 10°C. As Fig. 7 shows, the LDC PUF has strong temperature resistance. When the temperature is up to 85°C, the reliability slightly decreases 0.3%. Now we can say that the reliability of LDC PUF is 96.91% in the worst case (85°C) as shown in Fig. 8. The experimental results conform to the normal distribution with parameters $\mu = 3.96$ and $\sigma = 1.9118$. Such a distribution means a very low probability (less than 0.1%) of more than 10 erroneous bits in a 128-bit response, which is completely enough for a chip to be recognized or authenticated.

In summary, we have discussed three metrics of LDC PUF. To comprehensively compare the performance with other related works, Table I is made. Note that we uniformly set the number of CLBs at 20 in order to show better the difference in hardware efficiency of PUFs with different structures. As shown in Table I, our LDC PUF is similar to other PUFs regarding uniqueness. It is slightly weaker than others in terms of reliability but still within the acceptable range for practical applications. Regarding unit response cost, LDC PUF has a considerable improvement compared to other PUFs, which means we can well improve the disadvantage of traditional RO PUFs that requires enormous resources.

4. Conclusion

To solve the problem of low hardware efficiency of traditional RO PUFs, this article proposes the LDC PUF. The LDC PUF is easy to implement in FPGAs without particular placement and routing, and has incredibly high hardware efficiency. To improve the reliability of the LDC PUF, we introduce PDL technology, which improves reliability by 0.73% without any post-processing and additional resources.
Final results show that the LDC PUF achieves ultra-low unit response cost (1.45 × 10⁻¹⁰ using 20 CLBs) and has satisfactory uniqueness (48.52%) and reliability (96.91%).

References

[1] E. Simpson and P. Schaumont: "Offline Hardware/Software Authentication for Reconfigurable Platforms," Cryptographic Hardware and Embedded Systems 4249 (2006) 311-323 (DOI: 10.1007/11894063_25).
[2] S. Sunkavilli, et al.: "New Security Threats on FPGAs: From FPGA Design Tools Perspective," IEEE Computer Society Annual Symposium on VLSI (ISVLSI) (2021) (DOI: 10.1109/ISVLSI1109.2021.00058).
[3] S. S. Kumar, et al.: "Extended abstract: The butterfly PUF protecting IP on every FPGA," IEEE International Workshop on Hardware-Oriented Security and Trust (2008) 67-70 (DOI: 10.1109/HOST.2008.459053).
[4] S. M. Trimberger and J. J. Moore: "FPGA Security: Motivations, Features, and Applications," Proceedings of the IEEE 102 (2014) 1248-1265 (DOI: 10.1109/JPROC.2012.2231672).
[5] S. Elgendy and E. Y. Tawfiq: "Impact of Physical Design on PUF Behavior: A Statistical Study," IEEE International Symposium on Circuits and Systems (ISCAS) (2021) (DOI: 10.1109/ISCAS5156.2021.9401140).
[6] J. W. Lee, et al.: "A technique to build a secret key in integrated circuits for identification and authentication applications," Symposium on VLSI Circuits (2004) Digest of Technical Papers (IEEE Cat. No.04CH37525) 176-179 (DOI: 10.1109/VLSIC.2004.1346548).
[7] J. Delvaux and I. Verbauwhede: "Fault Injection Modeling Attacks on 65 nm Arbiter and RO Sum PUFs via Environmental Changes," IEEE Transactions on Circuits and Systems 61 (2013) 1701-1713 (DOI: 10.1109/TCSI.2013.2290845).
[8] D. P. Sahoo, et al.: "A Multiplexer-Based Arbiter PUF Composition with Enhanced Reliability and Security," IEEE Transactions on Computers ) 67 (2018) 403-417 (DOI: 10.1109/TC.2017.2749226).
[9] C. Gu, et al.: "Novel lightweight FF-APUF design for FPGA." IEEE International System-on-Chip Conference (SOCC) (2016) 75-80 (DOI: 10.1109/SoCC.2016.7095439).
[10] Y. Cao, et al.: "An Energy-Efficient Current-Starved Inverter Based Strong Physical Unclonable Function With Enhanced Temperature Stability," IEEE Access 7 (2019) 105287-105292 (DOI: 10.1109/ACCESS.2019.2932022).
[11] J. Guajardo, et al.: "FPGA Intrinsic PUFs and Their Use for IP Protection," Cryptographic Hardware and Embedded Systems (CHES) 4727 (2007) (DOI: 10.1007/978-3-540-74735-2_5).
[12] A. Garg and T. T. Kim: "Design of SRAM PUF with improved uniformity and reliability utilizing device aging effect," IEEE International Symposium on Circuits and Systems (ISCAS) (2014) 1941-1944 (DOI: 10.1109/ISCAS.2014.6865541).
[13] R. Maes, et al.: "Intrinsic PUFs from flip-flops on reconfigurable devices," workshop on information and system security (WISSec) (2008).
[14] S. Hemavathy and V. S. Kanchana Bhaaskaran: "Double Edge-Triggered Tristate Flip-Flop Physical Unclonable Function for Secure IoT Ecosystem," IEEE International Symposium on Smart Electronic Systems (iSES) (2021) 44-47 (DOI: 10.1109/iSES5264.2021.00022).
[15] Shimizu, et al.: "Glitch PUF: extracting information from usually unwanted glitches," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences E95-A (2012) 223-233 (DOI: 10.1587/transfun.E95.A.223).
[16] Y. Nozaki, et al.: "Performance Evaluation of Unrolled Cipher based Glitch PUF Implemented on Virtex-7," International Symposium on Devices, Circuits and Systems (ISDCS) (2021) 1-4 (DOI: 10.1109/ISDCS5206.2021.9397981).
[17] Y. Dong, et al.: "An Anti-interference Design based on Glitch PUF," International Conference on Electronics Information and Emergency Communication (ICEIEC) (2019) 217-221 (DOI: 10.1109/ICEIEC.2019.8784512).
[18] G. E. Suh and S. Devadas: "Physical Unclonable Functions for Device Authentication and Secret Key Generation," ACM/IEEE Design Automation Conference (2007) 9-14 (DOI: 10.1145/1278480.1278484).
[19] C. Gu and M. O’Neill: "Ultra-compact and robust FPGA-based PUF identification generator," IEEE International Symposium on Circuits and Systems (ISCAS) (2015) 934-937 (DOI: 10.1109/ISCAS.2015.7168788).
[20] A. Maiti, and P. Schaumont: "Improved Ring Oscillator PUF: An FPGA-friendly Secure Primitive," J. Cryptol 24 (2011) 375-397 (DOI: https://doi.org/10.1007/s00145-010-9088-4).
[21] X. Xin, et al.: "A configurable ring-oscillator-based PUF for Xilinx FPGAs," Euromicro conference on digital system design (2011) 651-657 (DOI: 10.1109/dsd.2011.88).
[22] M. Z. Gao, et al.: "A highly flexible ring oscillator PUF," ACM/EDAC/IEEE Design Automation Conference (DAC) (2014) 1-6 (DOI: 10.1145/2593069.2593072).
[23] L. Zhang, et al.: "XOR gate based low-cost configurable RO PUF," IEEE International Symposium on Circuits and Systems (ISCAS) (2017) 1-4 (DOI: 10.1109/ISCAS.2017.8050628).
[24] Y. Cui, et al.: "Ultra-Lightweight and Reconfigurable Tristate Inverter Based Physical Unclonable Function Design," IEEE Access 6 (2018) 28478-28487 (DOI: 10.1109/ACCESS.2018.2839363).
[25] Z. Wei, et al.: "Transformer PUF: A Highly Flexible Configurable RO PUF Based on FPGA," IEEE Workshop on Signal Processing Systems (SiPS) (2020) 1-6 (DOI: 10.1109/SiPS50750.20195259).
[26] D. Deng, et al.: "Configurable Ring Oscillator PUF Using Hybrid Logic Gates," IEEE Access 8 (2020) 161427-161437 (DOI: 10.1109/ACCESS.2020.3021205).
[27] Z. Cherif, et al.: "An Easy-to-Design PUF Based on a Single Oscillator: The Loop PUF," Euromicro Conference on Digital System Design (2012) 156-162 (DOI: 10.1109/DSD.2012.22).
[28] J. Gan, et al.: "A FPGA-based RO PUF with LUT-Based Self-Compare Structure and Adaptive Counter Time Period Tuning," IEEE International Symposium on Circuits and Systems (ISCAS) (2018) (DOI: 10.1109/ISCAS.2018.8351014).
[29] M. Majzoobi, et al.: "FPGA-Based True Random Number Generation Using Circuit Metastability with Adaptive Feedback Control," Cryptographic Hardware and Embedded Systems (CHES) (2011) (DOI: https://doi.org/10.1007/978-3-642-23951-9_2).
[30] C. Herder, et al.: "Physical Unclonable Functions and Applications: A Tutorial," Proceedings of the IEEE 102 (2014) 1126-1141 (DOI: 10.1109/JPROC.2014.2320516).