IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 19 July 2023, accepted 3 August 2023, date of publication 14 August 2023, date of current version 17 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3304901

RESEARCH ARTICLE

Exploiting the DD-Cell as an Ultra-Compact Entropy Source for an FPGA-Based Re-Configurable PUF-TRNG Architecture

RICCARDO DELLA SALA AND GIUSEPPE SCOTTI, (Senior Member, IEEE)
DIET Department, Sapienza University of Rome, 00184 Rome, Italy
Corresponding author: Giuseppe Scotti (giuseppe.scotti@uniroma1.it)

ABSTRACT Physical Unclonable Functions (PUFs) and True Random Number Generators (TRNGs) are both needed in the Privacy Preserving Mutual Autentication (PPMA) protocol, often used in IoT Applications to generate and secure cryptographic keys. Since to guarantee security of IoT nodes in an untrusted setting, the PPMA key and encrypted data must be located on the same chip, the concept of integrating both a PUF and a TRNG on the same device has emerged as a new security paradigm. Up to now only a few designs for achieving PUF and TRNG simultaneously on field programmable gate array (FPGA) platforms have been presented in the technical literature, and most of them show sub-optimal performance for one of the two cryptographic primitives. This paper presents a re-configurable design that is able to operate as an FPGA-compatible PUF+TRNG primitive, and relies on the Delay-Difference-Cell (DD-Cell) as the basic entropy source. A theoretical model of the DD-Cell explaining the PUF and the TRNG behaviour of the DD-Cell which highlights the effects of the routing connections on the FPGA on the performances is presented. The proposed solution has been implemented on the Artix-7 FPGA platform, and an extensive measurement campaign involving 32 FPGA boards has been carried out. Measured performances of the proposed PUF and TRNG primitives have been compared against state of the art PUFs and TRNGs, showing performances in line with the state of the art. The comparison against the PUF+TRNG designs available in the literature has shown that the proposed solution exhibits the best trade-off among PUF and TRNG performance, providing the most compact PUF and the highest throughput TRNG.

INDEX TERMS Physical unclonable function (PUF), true random number generator (TRNG), metastability, field programmable gate array (FPGA), hardware-security.

I. INTRODUCTION

In today's digital age, security has become a crucial aspect of our lives. With the increasing amount of sensitive information being stored and transmitted over networks, there is a growing need for robust security mechanisms. Cryptographic algorithms needed to safeguard user data privacy rely on cryptographic keys to ensure data protection. The current standard practice for securing mobile systems involves storing a confidential key in nonvolatile memory. However, over the past two decades, several tampering techniques, such as micro-probing, focused ion beam, glitch attacks, and side-channel attacks, have exposed vulnerabilities in the traditional approach of storing keys in cryptographic hardware devices [1], [2]. These attacks raise concerns about the security of stored keys and their susceptibility to exploitation by adversaries. To enhance security, hardware cryptographic operations like digital signatures and encryption are employed. Nevertheless, this approach has its drawbacks, including increased design area and power consumption, leading to higher costs [1]. Additionally, the vulnerability of nonvolatile memory to invasive attacks necessitates the continuous use of active tamper detection and prevention circuitry, consuming power [2], [3].

The associate editor coordinating the review of this manuscript and approving it for publication was Renato Ferrero.

Over the past decade, Physical Unclonable Functions (PUFs) have emerged as a crucial hardware-based protection mechanism for generating identification strings and cryptographic keys. PUFs leverage the unique physical properties of devices to generate unique and unclonable keys. These PUFs have been harnessed in various security protocols, such as key establishment protocols [4], [5], authentication protocols [3], [6], [7], [8], anti-counterfeiting protocols [9], [10], [11], and tamper detection protocols [12], [13], to enhance hardware security. Another essential hardware cryptographic primitive used in these protocols is the True Random Number Generator (TRNG). TRNGs generate truly random numbers, crucial for cryptographic applications. Furthermore, TRNGs find widespread usage in password generation [14], gaming [15], [16], [17], cloud computing [18], and SSL/TLS encryption, where random numbers serve as keys for encrypting application data through a symmetric cipher [19]. Due to their necessity in various security mechanisms, TRNGs have gained significant attention in recent years.

The Privacy Preserving Mutual Autentication (PPMA) protocol utilizes both PUF and TRNG as security primitives to safeguard privacy, resulting in a simple solution for generating and securing keys [20], [21]. Since to ensure security of IoT nodes in an untrusted setting, the PPMA key and encrypted data must be located on the same chip, the concept of integrating both PUF and TRNG on the same device has emerged as a new security paradigm.

PUF and TRNG primitives can be integrated on Application Specific Integrated Circuits (ASICs) or implemented in Field Programmable Gate Array (FPGA) platforms. Both PUF and TRNG primitives with high performance and low silicon area footprint have been successfully integrated on ASICs [22], [23], [24], [25], [26], exploiting a full-custom or a semi-custom design flow in which the adopted devices and/or the place and route of the standard cells can be optimized by the designer [27], [28]. A very simple approach to implement both a PUF and a TRNG in a single device (PUF+TRNG) is based on the usage of SRAM blocks [29], [30], [31].

Referring to the FPGA design flow, the elementary blocks are pre-placed and routing connections are pre-defined. Therefore, the FPGA implementation of PUFs and TRNGs faces a major challenge in terms of hardware resources usage and statistical performances, also because the designer has less degrees of freedom both in terms of devices selection and routing connections. Indeed TRNG and PUF performances can still be optimized by appropriately selecting the hardware resources and the routing strategy, and several FPGA-based PUF and TRNG designs have been proposed in the literature [32], [33], [34], [35], [36], [37].

Due to the low non-recurrent costs, the short time-to-market, and the availability of third party IPs, FPGAs are the chosen platform for many applications such as medical devices, remote sensing, military and aerospace, government systems, automotive and consumer electronics, industrial control systems and so on [38].

Up to now only a few designs for achieving PUF and TRNG simultaneously on field programmable gate array (FPGA) platforms have been presented in the technical literature, and most of them show sub-optimal performance for one of the two cryptographic primitives [21], [39], [40].

This paper presents a re-configurable design that is able to operate as an FPGA-compatible PUF+TRNG primitive. The proposed re-configurable design exploits the Delay-Difference-Cell (DD-Cell) as the basic entropy source. The DD-Cell has been already proposed to implement a PUF primitive in [41], and the idea to exploit the DD-Cell also as a TRNG so to implement a PUF+TRNG circuit has been outlined in [42]. This work is an extension of the conference paper [42], and several contributions are added to the previous study, as listed in the following:

- a theoretical model of the DD-Cell explaining the PUF and the TRNG behaviour, which allows to gain insight into the effects of the routing connections on the FPGA, is presented for the first time;
- an optimized excitation sequence which allows to drastically improve the throughput of the TRNG, while guaranteeing high reliability for the PUF is exploited in this work;
- the results of NIST tests and AIS tests, considering the typical condition and also temperature and supply voltage variations, are presented for the first time;
- the characterization of the TRNG performances over 32 FPGA devices is presented for the first time;
- the performances of the proposed re-configurable design are evaluated and compared with respect to the state of the art of PUFs and TRNGs.

The paper is structured as follows. Section II reviews the state of the art of PUF, TRNG and PUF+TRNG architectures. Section III provides a detailed analysis of the DD-Cell as an entropy source suitable to implement both PUF and TRNG primitives. Section IV describes the FPGA implementation of the DD-Cells, and in particular of the 4-bits macro. Section V summarizes the evaluation metrics and statistical tests adopted for the characterization of PUF and TRNG circuits. Section VI reports the results of the measurements carried out on 32 FPGA boards. Finally, a comparison against the state of the art and some conclusions are reported in Section VII and in Section VIII respectively.

II. BACKGROUND

A. REVIEW OF PREVIOUS TRNG ARCHITECTURES

The randomness performance and the resource usage of a TRNG are strongly affected by the physical phenomenon exploited as entropy source. There are several sources of entropy which can be exploited to develop a TRNG, and one of the most used is the Jitter accumulation [43], [44], [45]. Another approach to extract entropy relies on the usage of the Coherent Sampling (CS) technique, which involves measuring the Jitter from two mutually prime

oscillators [46]. Other popular TRNG architectures, are those based on Ring-Oscillator (RO) configurations [39], [44], [47]. The main drawback of such architectures is that they often suffer from low throughput, which has become a more and more hard to guarantee requirement. To improve the throughput of RO-based TRNGs, several techniques have been proposed in the recent literature. The first one is based on modulating the oscillation frequency of ROs by exploiting delay-lines control [43] or the effect of different frequencies in beat frequency detection (BFD) operation for digital clock manager (DCM)-based TRNGs [48], [49]. Another approach relies on a Multi-Stage-Feedback Ring Oscillator (MSFRO) TRNG which employs feedback to implement high-frequency ROs [50]. Other approaches take into account the fast carry logic, the DCM and PLLs of Xilinx's FPGAs to enhance the frequencies of ring oscillators, thus improving the throughput. It has to be pointed out that, even if these approaches can provide high throughput, they may require high power consumption, and it is good to keep in mind that a trade-off between speed, power consumption and resources usage has always to be satisfied. Indeed, high throughput TRNGs typically require a significant amount of power, which can be a limiting factor in certain applications. Furthermore, the quality of bitstreams generated by high-throughput architectures can be sensitive to environmental variations, and in many cases, particular care must be taken to ensure sufficient robustness of the TRNG primitives [48], [49].

Lightweight architectures, such as those based on Latched Ring Oscillators (LROs), have been proposed to drastically reduce the usage of hardware resources [45], but they may generate bitstreams that are not able to meet randomness requirements. In these cases, design strategies aimed at improving bitstream quality, such as post-processing techniques [51], or feedback strategies for changing the excitation time of metastable architectures [45], must be considered. Consequently, intricate computations are necessary to identify the optimal parameters for these types of TRNGs [49]. Also the MSFRO TRNG may present limitations in terms of entropy, and parity filters should be considered to improve the bias of the bitstream [50]. Metastable cells are a valuable alternative to ROs for the implementation of TRNGs with limited hardware resources and power consumption, and allow the implementation of TRNGs with higher throughput than RO-based ones [52], [53]. A metastable cell is a circuit that can produce unpredictable output when stimulated with an input that violates its setup or hold time requirements. By intentionally violating these requirements, the circuit can produce random output bits that can be used as a source of entropy. Therefore, TRNGs based on metastable cells exhibit several advantages over traditional TRNGs. Requiring less hardware resources and consuming less power, they are excellent candidates to be implemented on heavily constrained components, such as sensors and IoT devices. Additionally, being faster than other TRNGs, are suitable also for applications that require large amounts of random data. Indeed, despite their advantages, metastable TRNGs exhibit some limitations. In fact, they are more susceptible than other types of TRNGs to environmental noise and external factors that can affect the stability of the stimuli signals. In addition, in some cases, tuning and/or calibration could be required to ensure good statistical performances [54].

Thermal, shot and flicker noise can also be used as a source of entropy by measuring the random fluctuations in voltage or current in a resistor or transistor [52], [53], [55]. This approach is commonly used in hardware implementations of TRNGs, where the noise is amplified and filtered to extract the randomness [56].

B. REVIEW OF PREVIOUS PUF ARCHITECTURES

In the existing literature, there is a distinction between two important subtypes of PUFs: strong PUFs and weak PUFs [8]. Strong PUFs are characterized by a large number of possible challenges, making it practically impossible to determine or measure all challenge-response pairs within a limited time frame. On the other hand, weak PUFs have a limited number of challenges, sometimes even just one fixed challenge [57], [58]. The responses generated by weak PUFs are used to derive a classical binary secret key, which is then processed by the embedding system in a standard manner, serving as a secret input for classical cryptosystems. This similarity to non-volatile key storage makes weak PUFs more difficult to read out invasively compared to common non-volatile memory technologies like EEPROM. Moreover, weak PUFs leverage inherent manufacturing variations to individualize hardware without requiring costly, dedicated individualization steps during production. This work focuses on weak PUFs in order to generate keys which can be further exploited in Cryptography protocols such as key establishment protocols [4], [5] and authentication protocols [3], [6], [7], [8].

In [59] the butterfly PUF was introduced as an FPGA-amenable architecture to generate keys. The butterfly PUF exploits a metastable architecture to generate an unique and unpredictable response to a stimulus, relying on physical properties of the silicon and on mismatch variations. A plenty of physical phenomena have been exploited in literature to extract a unique key from physical variations. For example, in [60] a key is generated by comparing the oscillation frequency of two nominally identical ROs. However, since the performance of the PUF strongly relies on the physical implementation, RO-based PUFs typically suffer of poor Uniqueness [61], [62]. Indeed, long delay lines and inverter inter-connections can systematically affect the delay differences between the two compared ROs, so that the bits extracted from the comparison can result deterministic [62], [63].

The SRAM-based PUF reported in [64] is much more compact and exhibits better Uniqueness than RO-based implementations [32], [62], [65]. This is due to the fact that differences in a metastable SRAM cell come just from

two main sources: the delay of the interconnections and the propagation delay of the logic gates. SRAM-based PUFs generate the key leveraging on small asymmetries between the two symmetrically and nominally identical branches that compose the cross-coupled cell. Since the sign of the delay difference, which determines the value of the output bit, can change under environmental variations, these kind of PUFs are sensitive to supply voltage and temperature variations and the reliability of the key is poor with respect to RO-based PUFs.

In the last decade, several PUF solutions exploiting metastability have been proposed in the literature, such as the PUF based on the NAND latch [66], [67], the PICO-PUF [34], [62], [68], the Meta-XOR PUF [65] and so on, nevertheless resulting in lower reliability but also lower resources usage than those based on ROs, such as [60] and [62]. Indeed, in the FPGA implementation of the NAND latch or of other metastable cells, the typical values of the logic cells delay and of the intra-connections delay, give rise to a transient effect [69], [70], [71], similar to the one reported in the transient effect ring oscillator (TERO) PUF [72], which mainly consists in transient oscillations whose duty cycle progressively decreases (increases) till the output reaches a stable logic '0' ('1'). It is worth noting that these metastability-based PUFs are different from the TERO-PUF, both for the FPGA implementation (the TERO-PUF require many resources for one bit extraction), and for the working principle. Indeed, TERO-PUF extracts the bit from the comparison of the number of oscillation which two nominally identical circuit branches have performed before reaching the steady state, whereas the other metastability-based PUFs extract the final state information [41].

Recently, a technique to enhance the reliability of metastability-based PUFs has been presented in [32], [41], and [42], which is based on a novel excitation sequence to collect a stable and reliable PUF response after an optimal number of clock cycles.

C. REVIEW OF PREVIOUS PUF-TRNG ARCHITECTURES

A very simple approach to implement both a PUF and a TRNG in a single device (PUF+TRNG) is based on the usage of SRAM blocks. In this solution, the SRAM bit cells with a stable startup value are used for the PUF, whereas the cells with unstable value are used for the TRNG. Previous research has explored this unified approach using different gate-level implementations, including cross-coupled inverters commonly adopted to implement SRAM cells [29], [30], [31].

On the other hand, some PUF+TRNG solutions based on ROs and suitable to be implemented on FPGA platforms have been presented in [21] and [39]. In [39], the PUF key is extracted by comparing the oscillation frequency of pair-wised ROs excited with the same challenge, whereas the extraction of the random sequence relies on the Jitter of ROs. More in detail, the outputs of several ROs are XOR-ed together, and the output of the XOR is sampled by a D-Flip-Flop. The sampled values are then collected in a FIFO and then a post-processing technique (e.g. the Von Neumann encoding) is applied to obtain the output bitstream. With respect to the study of [39], in [21] a separation method between unique key extraction and random bit collection is introduced. The main idea is to exploit a conventional RO PUF which can be programmed to behave as a TRNG. With this aim, a Von Neumann corrector and a XOR operation are applied at the output of each pair of ROs. In addition, a real-time control which discards ROs which doesn't exhibit enough entropy is introduced, thus improving the statistical performances at the price of limiting the overall throughput. This solution is effective and the obtained random sequence is able to pass National Institute of Standards and Technology (NIST) [73], [74] tests also under environmental variations. An Universal TERO-based (UTERO) structure consisting of a TERO loop and a PUF/TRNG bit extraction logic has been presented in [75]. In detail, the number of oscillations of the metastable waveform [76], due to the delay difference among the two symmetrical paths [77] is exploited to generate a TRNG response. For what concerns the PUF behavior, the stable output of the oscillation is used to generate the PUF response, such as in other metastable-based PUFs. All the experiments were performed on a Microsemi (former Actel) FPGA. Also in this case, being the output bit of the TRNG collected after a large number of clock cycles, the throughput is limited to about 0.9 Mbit/s which, if compared with other TRNGs, is very low. Recently, another approach to extract random bits from two pairs of ring oscillators has been presented in [40]. This technique relies on the usage of one of the two ring oscillators (RO_1) as a reference circuit. The oscillation is started by a common signal called *init* which is leaved high until RO_1 reaches a selected number (N_{ref}) of oscillation periods. When RO_1 reaches the given number of oscillations, the *init* signal goes low, and both RO_1 and RO_2 are stopped. The number of oscillation periods of RO_2 (counted by a N bit counter) is exploited to extract a PUF key and a random sequence. In detail, the less significant bits (LSBs) of the RO_2 counter are used to extract Jitter noise from the RO, whereas, a post-processing on the count value is employed to extract the bits of the PUF. Each pair of oscillators is enabled sequentially: this improves the randomness and reduces the power consumption, but strongly impacts the throughput which is limited to 24.1 kb/s.

III. THE DD-CELL AS A METASTABILITY-BASED ENTROPY SOURCE

A. THE OPERATING PRINCIPLES

The block scheme of DD-Cell exploited in this work as a metastability-based entropy source is reported in Fig. 1. It is made up of two D-type latches L_1 and L_2, with an asyncronous reset input (R), and two inverter gates IV_1 and IV_2. In this metastable cell, the delay t_{p1} from the input of IV_1 to the output of L_1, is nominally identical to the delay

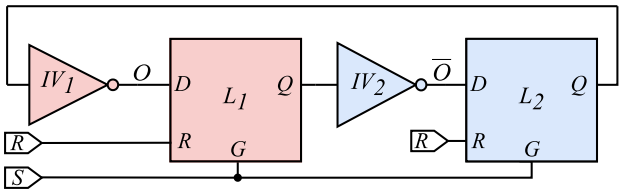

FIGURE 1. Block scheme of the DD-Cell exploited as a metastability-based entropy source.

t_{p2} from the input of IV_2 to the output of L_2. If mismatches between the propagation delay of the logic cells L_1, L_2, IV_1 and IV_2, and between the two routing paths are taken into account, the delay difference $\Delta t_p = t_{p2} - t_{p1}$, has to be considered a random variable.

Referring to Fig. 1, the signal R allows to force the D-type latches L_1 and L_2 in a reset condition, whereas the signal S, which controls the gate (G) pin of the latches, is exploited to start a race condition and to sample a metastable or stable value depending on the time instant in which the sampling occurs. The excitation sequence needed to properly stimulate the DD-Cell is described as follows:

- $R = 1$ and $S = 0$, *Reset Condition*: the outputs Q_1 and Q_2 of the latches L_1 and L_2 are forced to logic $'0'$ thus setting the inputs D_1 and D_2 of L_1 and L_2 to logic $'1'$, due to the inverters IV_1 and IV_2;
- $R = 0$ and $S = 1$, *Start Condition*: the latches L_1 and L_2 become transparent and follow the variations on their respective D inputs, hence a race condition is triggered and, depending on the delays' mismatch of the two branches composed by L_1, IV_1 and L_2, IV_2, a final state is reached after an amount of time which is also dependent on the value of the delay difference Δt_p, as will be better detailed in the following.
- $R = 0$ and $S = 0$, *Sampling Condition*: L_1 and L_2 are forced in the hold state and the outputs Q_1 and Q_2 are latched.

Considering the FPGA implementation of the block scheme reported in Fig. 1, the delay of the routing interconnections is not negligible and plays a key role in determining the behaviour of the circuit. Indeed, considering the typical values of logic cells delay and of interconnections delay for the FPGA implementation, the DD-Cell exhibits an oscillating behaviour similar to the one obtained in the TERO PUF, (i. e., oscillations with progressively decreasing or increasing duty cycle) [69], [70], [71]. The final state reached by the DD-Cell depends on delay asymmetries of the architecture [41], [42]. Fig. 2 shows the typical waveforms obtained from a DD-Cell when the above excitation sequence is used. The PUF zone and the TRNG zone are highlighted in the figure.

Experimental results have confirmed that the number of oscillations required to reach the steady state after the triggering condition increases if the $\Delta t_p = t_{p2} - t_{p1}$ tends to 0. If the oscillating waveform at the output of the DD-Cell is sampled after a long time (i.e., after a large number of clock cycles), a stable state is always collected, and the final bit value depends on the sign of the delay difference $sign(\Delta_{t_p})$. Thus, the unpredictability of the PUF key depends on the overall mismatch between the propagation delays and on the number of clock cycles that the cell lets itself ring.

On the other hand, if the oscillating waveform at the output of the DD-Cell is sampled during the oscillating phase (i.e., a few clock cycles after the triggering condition), a steady state has not yet been reached and a metastable state is collected. This meta-stable state can be exploited to collect random output bits that, due to Jitter, process variations and mismatches can randomly be '0' or '1'.

B. THEORETICAL MODEL OF THE DD-CELL CONFIGURED AS PUF

The propagation delay of a generic logic gate is affected by process (i.e., die to die) variations and follows a Normal distribution with mean value μ_{prc} and standard deviation σ_{prc}:

$$T_{prc} \sim \mathcal{N}(\mu_{prc}, \sigma^2_{prc}) \tag{1}$$

The propagation delay is also affected by mismatch (i.e., whithin die) variations which follow a Normal distribution with mean value μ_{mm} and standard deviation σ_{mm}:

$$T_{mm} \sim \mathcal{N}(\mu_{mm}, \sigma^2_{mm}) \tag{2}$$

The generic delay can be therefore written as:

$$T_{Di} = T_{prc,i} + T_{mm,i} \tag{3}$$

and follows a multivariate Normal distribution. If a single realization of the manufacturing process is considered, T_{prc} is a constant whose value is denoted as τ_D. Then, the generic propagation delay can be written as:

$$T_{Di} = T_{mm,i} + \tau_{Di} \tag{4}$$

In order to develop a theoretical model of the architecture in Fig. 1, the block scheme of the DD-Cell highlighting the different delay contributions reported in Fig. 3 is considered. In Fig. 3, T_{D1} and T_{D3} denote the propagation delay of IV_1 and L_1 respectively, and T_{D5} represents the routing delay between the output of L_1 and the input of IV_2, whereas T_{D2} and T_{D4} denote the propagation delay of IV_2 and L_2 respectively, whereas T_{D6} represents the routing delay between the output of L_2 and the input of IV_1.

By using the above notation, the delay difference between the two paths can be expressed as:

$$\Delta T = -\sum_{i=1}^{6}(-1)^i(T_{mm,i} + \tau_{Di}) \tag{5}$$

which, if the different delay contributions are assumed as uncorrelated between each other, follows a Normal distribution with mean value

$$\mathbb{E}[\Delta T] = -\sum_{i=1}^{6}(-1)^i\tau_{Di} \tag{6}$$

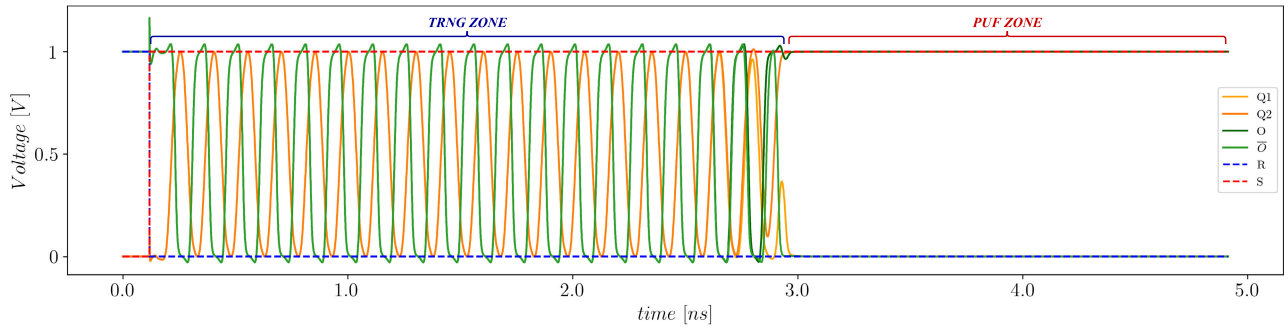

FIGURE 2. Initialization sequence of the DD-Cell.

and variance:

$$\mathrm{Var}[\Delta T] = \sum_{i=1}^{6} \sigma_{mm,i}^2 \tag{7}$$

It has to be remarked that, since nominally the two inverters IV_1 and IV_2 and the two D-latches L_1 and L_2 are identical to each other, if the delay of the routing paths T_{D5} and T_{D6} are also nominally identical to each other (or if their value can be neglected with respect to the other propagation delays), it can be concluded that $\mathbb{E}[\Delta T] = 0$. Obviously the perfect matching between the delays T_{D5} and T_{D6} of the two routing paths is not easy to achieve on FPGA implementations, as will be better discussed in the following.

Focusing on the PUF zone in Fig. 2, the value of the output bit, can be expressed as:

$$X = \frac{1 + sign\{\Delta T\}}{2} \tag{8}$$

and its expected value is:

$$\mathbb{E}[X] = \mathbb{E}\left[\frac{1 + sign\{\Delta T\}}{2}\right] = \frac{1}{2} + \frac{\mathbb{E}\left[sign\{\Delta T\}\right]}{2} \tag{9}$$

Thus, according to Eq. 9 the output bit of the PUF follows a Bernoulli distribution whose mean value is related to the expected value of the sign of ΔT.

The expected value of the $sign\{\Delta T\}$ can be written as:

$$\mathbb{E}[sign\{\Delta T\}] = 1 \cdot \mathrm{P}(\Delta T > 0) - 1 \cdot \mathrm{P}(\Delta T < 0) \tag{10}$$

where $P(A)$ denotes the probability of the event A.

Under the hypothesis that $\mathrm{P}(\Delta T > 0) = \mathrm{P}(\Delta T < 0)$ (i.e.perfectly matched routing paths) it follows that:

$$\mathbb{E}[X] = 0.5 \tag{11}$$

which is the optimum value for an entropy source.

Considering the architecture of FPGA devices, the delay of routing interconnections can not be neglected and the matching between the two routing paths is not easy to achieve. Indeed, referring to AMD/Xilinx FPGA platforms, connections among Slice elements pass through the Switch Matrix, which manages intra-Slice and inter-Slice connections. Furthermore, it can be assumed that LUTs on the same

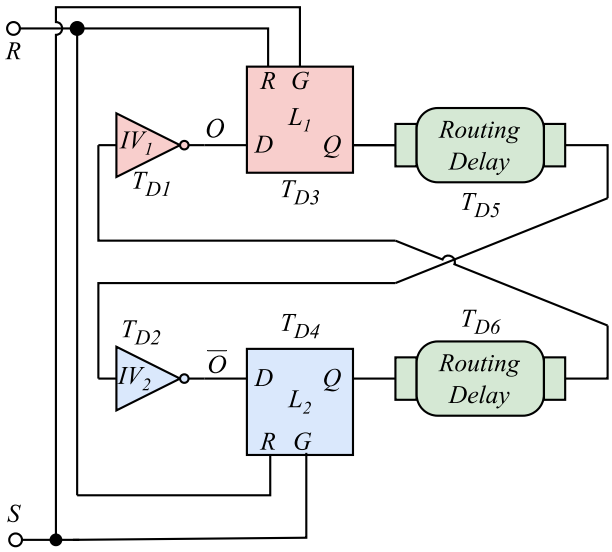

FIGURE 3. Block scheme of the DD-Cell highlighting the different delay contributions.

Slice perform identically with respect to the propagation delay (this is true if one consider LUTs with same input port (e.g. the input 3) and same output port (i.e. O_5 or O_6)). Thus, considering a realistic FPGA implementation, the mean value of the sign of ΔT can be expressed as:

$$\begin{aligned}&\mathbb{E}[sign\{\Delta T\}]\\ &= 1 \cdot \mathrm{P}(\Delta T > 0) - 1 \cdot \mathrm{P}(\Delta T < 0)\\ &= 2 \cdot P(\Delta T > 0) - 1\end{aligned} \tag{12}$$

Then, assuming that ΔT follows a Normal distribution with mean value $\mu_{\Delta T} = \tau_{6D} - \tau_{5D}$ and variance as reported in Eq. 7, it can be derived that:

$$\mathbb{E}[sign\{\Delta T\}] = -\mathrm{erf}\left(\frac{\tau_{6D} - \tau_{5D}}{\sqrt{2 \cdot \sum_{i=1}^{6} \sigma_{mm,i}^2}}\right) \tag{13}$$

and thus:

$$\mathbb{E}[X] = \frac{1}{2} - \frac{1}{2}\mathrm{erf}\left(\frac{\tau_{6D} - \tau_{5D}}{\sqrt{2 \cdot \sum_{i=1}^{6} \sigma_{mm,i}^2}}\right) \tag{14}$$

Eq. 14 clearly shows that if the designer is able to guarantee a good matching among the two routing delays, enough entropy can be extracted and no bias given by the deterministic intra-Slice connection difference will be extracted. Thus the routing strategy adopted in the FPGA implementation is crucial to achieve good performances and it is strictly recommended to guarantee that the intra-Slice connections are as symmetric as possible in order to extract the maximum entropy.

C. THEORETICAL MODEL OF THE DD-CELL CONFIGURED AS TRNG

In order to derive a theoretical model of the DD-Cell configured as TRNG, the block scheme of the DD-Cell reported in Fig. 3, and the same notation adopted in section III-B are considered. According to the excitation sequence of the DD-Cell described in section III-A, the oscillating waveform is sampled (latched) in a time instant belonging to the TRNG zone in Fig. 2, and the probability to sample a logic '1' ('1' event probability in the following) is equal to the duty cycle (DC) of the transient-effect oscillating waveform [78].

Since in TRNG configuration the most important contribution to delay variations comes from Jitter variations, a single realization of the manufacturing process and of the mismatch process can be considered, and the generic delay contribution T_{Di} in Fig. 3 can be expressed as:

$$T_{Di} \sim \mathcal{N}(0, \sigma_{t_i}^2) + t_{D,i} \tag{15}$$

where $\sigma_{t_i}^2$ denotes the variance of the time Jitter, and $t_{D,i}$ is the value of the delay for the considered process and mismatch realizations. Under these hypotheses, the period T of the oscillating waveform produced by the DD-Cell (see Fig. 2) can be expressed as:

$$T \sim \mathcal{N}(0, \sigma_T^2) + \sum_{i=1}^{6} t_{D,i} \tag{16}$$

where:

$$\sigma_T^2 = \sum_{i=1}^{6} \sigma_{t_i}^2 \tag{17}$$

The duty cycle of the oscillating waveform decreases at each period of a quantity related to the delay difference ΔT as shown in Fig. 4 [78]. The Duty Cycle of the waveform in the first oscillation period is depicted in Fig. 4a: ΔT is affected by Jitter, and its standard deviation is denoted as $\sigma_{\Delta T}^2$. The $\sigma_{\Delta T}^2$ can be computed as follows:

$$\sigma_{\Delta T}^2 = \sum_{i=1}^{6} \sigma_{t_i}^2 \tag{18}$$

and thus it is equal to σ_T^2.

The Duty Cycle of the waveform at the M-th oscillation period is depicted in Fig. 4b and can be written as:

$$DC = \frac{1}{2} - M \cdot \frac{\Delta T}{T} \tag{19}$$

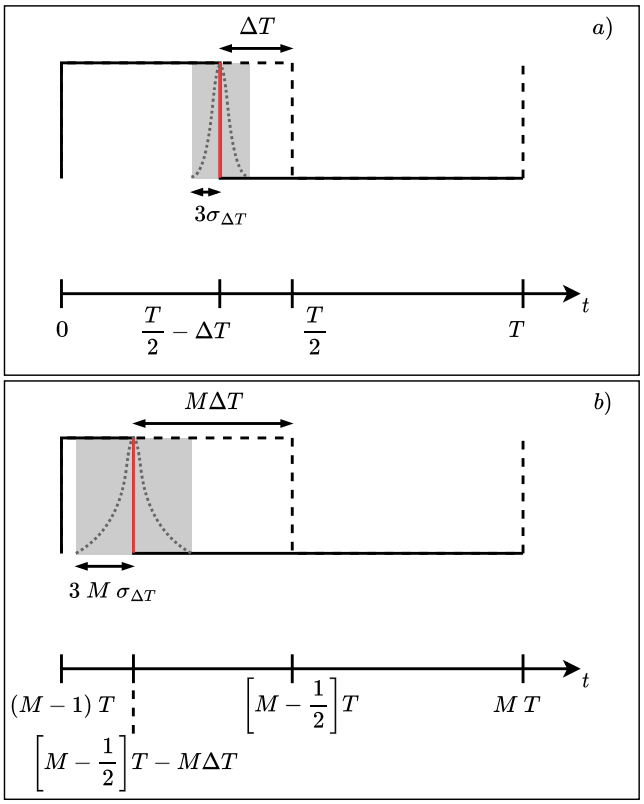

FIGURE 4. Graphical representation of the DC of the proposed TRNG at the first clock cycle a) and at the *M*-th clock cycle.

showing how the Jitter is accumulated over M oscillation periods.

However, being T and ΔT two correlated random variables, also DC is a random variable whose expected value can be computed as:

$$E[DC] = \frac{1}{2} - M \cdot E\left[\frac{\Delta_T}{T}\right] \tag{20}$$

And the variance can be written as:

$$VAR[DC] = M^2 \cdot VAR\left[\frac{\Delta_T}{T}\right] \tag{21}$$

From equations 20 and 21, exploiting the mathematical tools reported in [79], the expected value and the variance of the ratio Δ_T/T can be derived as:

$$E\left[\frac{\Delta_T}{T}\right] \approxeq \mu_{\Delta_T}/\mu_T + \sigma_T^2\mu_{\Delta_T}/\mu_T^3 - \rho\sigma_T\sigma_{\Delta_T}/\mu_T^2 \tag{22}$$

$$VAR\left[\frac{\Delta_T}{T}\right] \approxeq \sigma_T^2\mu_{\Delta_T}^2/\mu_T^4 + \sigma_{\Delta_T}^2/\mu_T^2 - 2\rho\sigma_{\Delta_T}\sigma_T\mu_{\Delta_T}/\mu_T^3 \tag{23}$$

where ρ denotes the correlation coefficient between Δ_T and T, μ_T is the mean value of the oscillation period T, and μ_{Δ_T} denotes the mean value of the delay difference Δ_T. Finally,

it can be concluded that the DC follows a normal distribution that can be as a first approximation written as:

$$DC \sim \mathcal{N}\Bigg(\frac{1}{2} - M \cdot (\mu_{\Delta_T}/\mu_T + \sigma_T^2 \mu_{\Delta_T}/\mu_T^3 - \rho\sigma_T\sigma_{\Delta_T}/\mu_T^2), \\ M^2 \cdot (\sigma_T^2\mu_{\Delta_T}^2/\mu_T^4 + \sigma_{\Delta_T}^2/\mu_T^2 - 2\rho\sigma_{\Delta_T}\sigma_T\mu_{\Delta_T}/\mu_T^3)\Bigg) \tag{24}$$

From the above expressions it is evident that achieving a stable condition (i.e., mean value of DC equal to 0 or 1) requires larger values of M if μ_{Δ_T} is low (i.e., $t_{D,6}$ is close to $t_{D,5}$), showing again the importance to match the routing paths within the FPGA device. It is also evident that the standard deviation of DC increases linearly with M due to Jitter accumulation, and therefore a good matching between the two paths leads to a greater variance, which in turns decreases the correlation between multiple instances. To maximize jitter accumulation, the sampling instant should be placed towards the end of the TRNG zone in Fig. 2, by keeping an adequate margin to sample an oscillating waveform also considering PVT variations. It has to be noted however that, even if a sampling instant placed close to the end of the TRNG zone maximizes the variance of the DC of a single DD-Cell, this choice is not the optimal one, when the outputs of several DD-Cells have to combined. In this last case the sampling instant has to be chosen as a tradeoff between the variance of DC and the number of unstable cells in a given array, as will be better discussed in the following.

IV. FPGA IMPLEMENTATION OF THE DD-CELL

The proposed DD-Cell has been integrated on the Xilinx Artix-7 XC7A100T FPGA, using Vivado 2021.2. An ad-hoc strategy to improve paths' symmetries has been adopted, trying to achieve the best balanced design, thus increasing entropy for TRNG and PUF applications and at the same time to improve the Uniqueness of the PUF. With this aim, the FPGA architecture has been carefully managed according to [27], [32], [41], and [45]. Xilinx FPGAs are organized as Configurable Logic Blocks (CLBs), each CLB has two Slices and each Slice has eight Flip-Flops, four of which can be configured as Latches. Each Slice contains also four Look Up Tables (LUTs) which can be programmed through a 64'bits init value to perform five inputs two outputs functions or six inputs one output functions. Further details about the Xilinx architecture and Slice elements can be found in reference Xilinx manuals [80], [81], [82]. As outlined in [41], the DD-cell can be mapped in just half-Slice, by exploiting two Flip-Flops programmed as Latches and by configuring LUTs as *NOT* gates. By using this approach, a single CLB can generate 4 bits, thus attaining a very lightweight entropic macro.

It is worth noting that, as pointed out in the previous section, it is crucial to assure that the FPGA implementation of the DD-Cells results as symmetric as possible. With this aim, in the design of the 4 bits macro, we spent a great effort to minimize nominal paths' delays, thus reducing mismatches at design routing level. The Vivado View of the 4 bits macro is reported in Fig. 5. The 4 instantiated DD-Cells have been highlighted, and, as it can be observed, LUT_A and LUT_D have been coupled with FF_A and FF_D to generate the first two bits of the first two Slices (red and cyan colors). We denote as $\tau_{Di,A}$ the routing delays of intra-Slice connections ($i = 5, 6$ refers to the upper or lower Slice of the CLB), from the output of FF_A to the input of LUT_D, whereas we denoted as $\tau_{Di,D}$ the interconnection delay from the output of FF_D to the input of LUT_A. The same kind of notation is used for $\tau_{Di,B}$ and $\tau_{Di,C}$. The routing delay for each Slice intra-connection and the relative delay difference for the 4 DD-Cell instances are reported in Table 1. As it can be observed, the worst case routing delay difference results to be about 5 *ps*, and this confirms that a very well balanced design has been achieved. It has to be pointed out that, even if we used the Vivado tool to match the routing delays in the Artix-7 device, a similar approach can be used also on different FPGA types such as for example Intel FPGAs in which the Quartus Prime tool can be exploited. The DD-Cell macro has also been designed on the Spartan-6 platform by using the ISE design tool and achieving a good matching among routing delays [41].

TABLE 1. Delay of routed interconnections.

Upper Slice			Lower Slice		
$i = 5$	$[ps]$	$\lvert\Delta_\tau\rvert$ $[ps]$	$i = 6$	$[ps]$	$\lvert\Delta_\tau\rvert$ $[ps]$
$\tau_{D5,A}$	457	1	$\tau_{D6,A}$	686	5
$\tau_{D5,D}$	456		$\tau_{D6,D}$	691	
$\tau_{D5,C}$	485	1	$\tau_{D6,C}$	494	2
$\tau_{D5,B}$	486		$\tau_{D6,B}$	492	

V. EVALUATION METRICS AND FIGURES OF MERIT

A. PUF METRICS

Since PUFs are utilized as a secure solution in several authentication protocols, such as Challenge Response authentication, it is imperative to evaluate their performances using standard and well assessed metrics. These metrics also facilitate the comparison between different PUF architectures and implementations. The most frequently employed metrics for PUF performance evaluation are *Uniformity*, *Randomness*, *Uniqueness*, and *Reliability*. These metrics are briefly reviewed in the following of this Section.

To ensure that PUF-generated keys are suitable for cryptographic purposes, it is essential to guarantee high entropy. The entropy is maximized (Shannon entropy equal to 1), if the number of 0s and the number of 1s, in a PUF-generated key, are equal to each other. In the general case, the number of 0s over the total number of bits of the key, is referred to as the *Uniformity* or the *Bias* of the response, and is ideally equal to 0.5.

The *Uniqueness* of a PUF is based on the inherent randomness generated by manufacturing variability in the physical

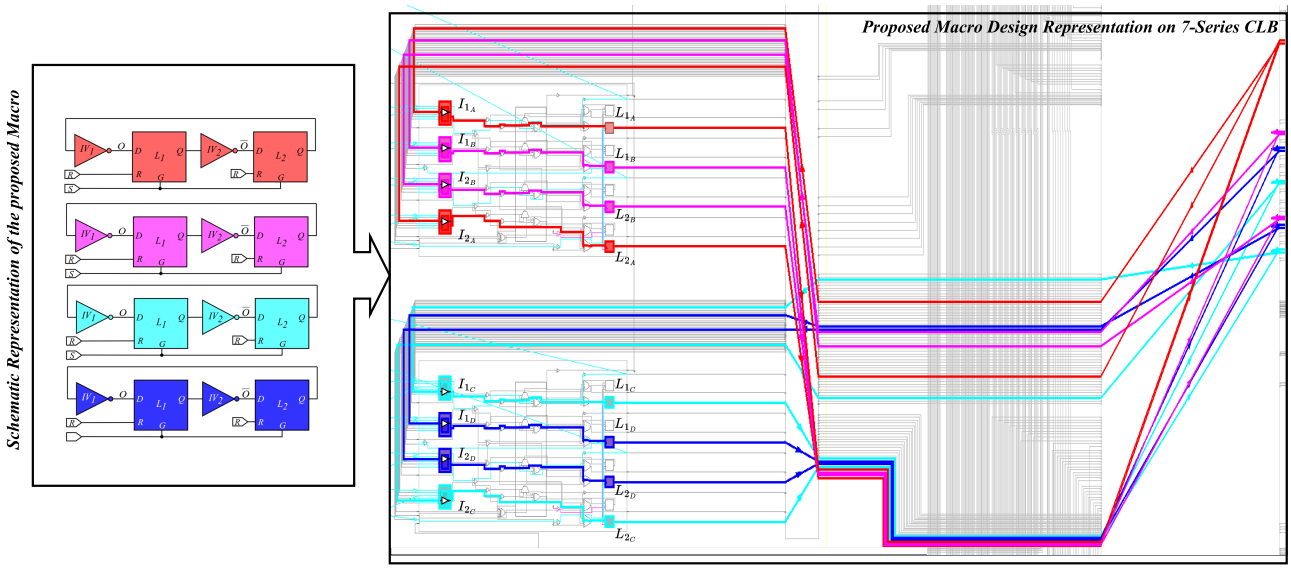

FIGURE 5. Vivado View of the 4 bits macro of the DD-Cell.

structure. Each PUF instance on silicon is unique to the device due to variations associated with the manufacturing process of the integrated circuit. Therefore, *Uniqueness* must be quantified across different implementations of the same PUF circuit on various devices (FPGAs or chips). To do this, the same design must be physically implemented on different devices and excited with the same stimuli under the same environmental conditions. Next, the unique identifier (i.e., the collected response to a given challenge) has to be extracted from each device. Then, the average value of the sum of inter-class Hamming Distance (HD_{inter}) between each possible pair-wise response couple has to be computed. The inter-class HD is defined as:

$$Uniqueness = HD_{inter} = \frac{2}{k(k-1)} \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \frac{HD(R_i, R_j)}{n} \tag{25}$$

where k denotes the number of devices under test, n is the number of response bits, and (R_i) refers to the i-th response obtained from the i-th PUF implementation under test. To ensure the unpredictability of each PUF fingerprint, applying the same challenge to n PUF implementations should result in different responses. Consequently, the optimal inter-class Hamming Distance between n chips should be 50%.

The *Reliability* of a PUF is determined by its ability to consistently produce the same response for a given stimulus across various sessions and environmental conditions (i.e., different temperatures and supply voltages). In some cases, certain bit-cells within the PUF array may exhibit variations in their output when subjected to noise, voltage fluctuations, or temperature changes. These cells are referred to as *Unstable Cells* and directly impact the *Reliability* of the PUF. It is important to note that a PUF bit-cell is considered unstable if, in a set of measurements, generates at least one output which results different from the others. The *Reliability* is assessed by selecting a reference challenge-response pair, known as the Golden-Key (GK), extracted under nominal conditions. This reference pair is then compared to new challenge-response pairs generated using stimuli in different operating conditions. Specifically, the intra-class Hamming Distance (HD_{intra}) is calculated between the GK and k responses collected under their respective power supply voltages and working temperatures, typically within $\pm 10\%$ of the nominal V_{DD} and T ranging from 0°C to 80°C. In details, the *Reliability* is defined as follows:

$$Reliability(V, T) = 1 - HD_{intra} = \frac{1}{k} \sum_{i=0}^{k-1} \frac{HD(R_{ref}, R_i^{V,T})}{n} \tag{26}$$

where R_i denotes the i-th response generated under a specific power supply voltage and temperature, while R_{ref} represents the reference GK response.

The Bit-Error-Rate (*BER*), defined as:

$$BER = HD_{intra} \tag{27}$$

is another important metric used to evaluate PUF performance, especially when voltage and temperature variations are taken into account. However, it is worth noting that the value of the *Reliability* can be inferred from the information of the Bit Error Rate (*BER*) under a specific condition, as indicated by Equation 26. Consequently, in PUF evaluation, the *BER* is commonly utilized instead of directly assessing the *Reliability*. Additionally, it should be emphasized that each PUF has a nominal *BER* (BER_{Typ}) associated with transient noise variations that introduce noise into the excitation sequence of the PUF.

B. PUF FIGURES OF MERIT

Several Figures of Merit (FOMs) which allow the comparison between different PUF architectures and implementations have been introduced in [32]:

$$FOM_{HD} = \frac{1}{\sqrt{HD_{intra}^2 + (0.5 - HD_{inter})^2}}$$

$$FOM_{BER_{V,T}} = \frac{1}{\sqrt{\left(\frac{BER_{wc_V}}{\Delta_V / V_{typ}}\right)^2 + \left(\frac{BER_{wc_T}}{\Delta_T / T_{typ}}\right)^2 + BER_{Typ}^2}} \quad (28)$$

$$\hat{FOM}_{HD} = \frac{\left(\frac{bits}{Slice}\right)}{\sqrt{HD_{intra}^2 + (0.5 - HD_{inter})^2}}$$

$$\hat{FOM}_{BER_{V,T}} = \frac{\left(\frac{bits}{Slice}\right)}{\sqrt{\left(\frac{BER_{wc_V}}{\Delta_V / V_{typ}}\right)^2 + \left(\frac{BER_{wc_T}}{\Delta_T / T_{typ}}\right)^2 + BER_{Typ}^2}} \quad (29)$$

The first FOM, FOM_{HD}, evaluates the Reliability and Uniqueness of a PUF design based on the intra-Hamming distance (HD_{intra}) and inter-Hamming distance (HD_{inter}). The second FOM, $FOM_{BER_{V,T}}$, evaluates the Reliability of a PUF design with respect to transient noise, voltage variations and temperature variations, based on the worst-case bit error rate (BER) measured under different operating conditions. In particular, BER_{typ} is the BER measured in typical conditions, while Δ_V and Δ_T denote the maximum range of voltage and temperature variations assumed in the measurements. The $BER_{wc_{V,T}}$ is the worst-case BER measured under voltage and temperature variations, respectively.

Both FOMs have a normalized version, $\hat{FOM}_{HD}$ and $\hat{FOM}_{BER_{V,T}}$, which takes into account the resources usage in terms of bits per FPGA Slice ($bits/Slice$ denotes the number of PUF-bits that can be implemented in a single FPGA Slice).

Overall, higher values of FOM_{HD} and $\hat{FOM}_{HD}$ indicate better performance in terms of Uniqueness and Reliability, whereas higher values of $FOM_{BER_{V,T}}$ and $\hat{FOM}_{BER_{V,T}}$ indicate better Reliability in the face of environmental variations. Lower BER values lead to higher FOM values.

C. TRNG METRICS AND TEST SUITES

To quantify the speed performance of a TRNG the most important metrics are throughput (TP) and operating frequency (OF), whereas the resources usage is quantified in terms of FPGA Slices or silicon area footprint for ASIC implementations. To assess the statistical quality of TRNG bitstreams, test suites such as NIST SP 800–22 [73], NIST SP 800-90B suite [74] and AIS-31 (Algorithmic Test Suite for Random Numbers) tests [45], have to be performed both in typical conditions and for different values of operating temperature and supply voltage.

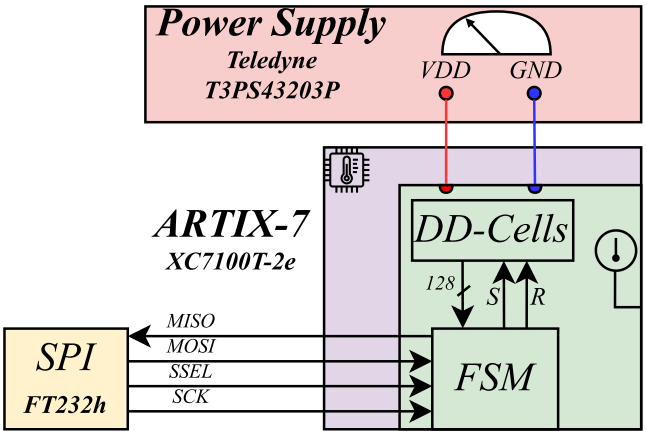

FIGURE 6. Testbench for the validation of the DD-Cells as PUF+TRNG.

D. TRNG FIGURES OF MERIT

A FOM which combines throughput (TP) performance, operating frequency (OF) and hardware resources usage (in terms of FPGA Slices) has been defined in [45] as:

$$FOM_T = \frac{TP}{OF \cdot Slices}\left[\frac{bits}{Slice}\right] \quad (30)$$

A more complete FOM (FOM_E) which considers also the bistream quality in terms of entropy has been recently introduced in [27] as:

$$FOM_E = \frac{FOM}{8 - H_8} = \frac{TP}{O.F.\ Slice} \cdot \frac{1}{8 - H_8} \quad (31)$$

where H_8 is the byte entropy estimated through the T8 test of the AIS-31.

It has to be pointed out that cryptographic primitives tested only in typical conditions may underestimate the resources usage, because the extracted bitstream may not be sufficiently random to pass NIST and AIS-31 tests when considering PVT variations. In fact, PVT variations can affect the bitstream behavior in a deterministic way, requiring additional resources to mask this effect.

VI. MEASUREMENTS

A. TEST-BENCH

In order to characterize the DD-Cell as an entropy source, 128 DD-Cell instances have been placed on the Artix-7 XC7A100T by consuming only 32 CLBs and 64 Slices. The testbench used to validate the proposed PUF+TRNG primitive is depicted in Fig. 6. An *FT-232H* SPI interface has been engaged to provide the excitation sequence and the excitation time was modulated through custom *Python* Scripts. Control signals S and R are sent by a finite state machine (FSM) which also allows to collect the outputs of the 128 DD-Cells through the SPI interface. In order to test the performance of the DD-Cell under voltage variations, a reference board has been modified so that the FPGA's core supply voltage can be externally set through a Teledyne T3PS43203P programmable power supply unit.

B. CONFIGURABLE PERFORMANCE

The characterization of the DD-Cells exploits the approach previously presented in [42] to investigate the main PUF performances under stimuli variations. Differently than what done in [42], two reference clocks at two different frequencies have been used in this work. The high frequency clock clk_H has a frequency $fclk_H$ equal to 450 MHz, whereas the low frequency one clk_L has a clock frequency $fclk_L$ equal to 50MHz. The performance of the DD-Cell has been investigated when the output signal is sampled at different times, defined here in terms of the number of clock cycles ($NCLK$) of the high frequency clock clk_H. The availbility of the high frequency clock allowed us to characterize the DD-Cell with a much lower time resolution with respect to the previous study in [42], and a time range from from 2.22 ns to about 5.1 μs with a time resolution of only $1/fclk_H$ (i. e., about 2.22 ns) has been analyzed in this work. The results of the characterization in terms of unstable cells (denoted as UCs in percentage), HD_{intra} in percentage, average bias trend (on a bitstream extracted from 1000 iterations and 128 sites), and the Shannon Entropy, are reported in Fig. 7 as a function of $NCLK$.

Results in Fig. 7 have been exploited for the design of the TRNG. In particular the optimum value of $NCLK$ has been chosen as the one which guarantees the maximum number of UCs and thus the maximum value of HD_{intra}. In this way, the uncorrelation of different bit-cells is maximized and the bitstreams extracted from several different bit-cells can be XOR-combined as in [27], in order to extract a single slower bitstream with good entropy and good statistical performances, as will be better detailed in the following.

C. DD-PUF CHARACTERIZATION

The PUF characterization was carried out by means of the test-bench of Fig. 6 and, according to [41], the stable bits are sampled after 128 clock cycles of the low frequency clock clk_L (i.e., $NCLK = \frac{450}{50} \cdot 128 = 1152$). With referring to the proposed DD-Cell PUF, we have reported the measurement results for the inter-HD and intra-HD in Fig. 8. As it can be observed, the results are in accordance with those reported in [41], which however validate the PUF performance over fewer boards. It was found that the Intra-HD has a mean value $\mu \approx 1.67$ % and a standard deviation of about $\sigma \approx 1.19$ %, whereas the mean value of the inter-HD was found $\mu \approx 49.48$ % with a $\sigma \approx 2.98$ %.

The *Reliability* of the DD-PUF was tested on the reference FPGA board whose core supply voltage was varied through the Teledyne T3PS43203P. Measurement results for different core supply voltages have been depicted in Fig. 9. As it can be observed, the device reaches a *Reliability* that, in worst case, is always greater than 90% whereas, in nominal condition an average *Reliability* of about 98.33 % is assured.

Reliability as a function of temperature, has been computed in a similar way as done for supply voltage, and results are depicted in Fig. 10, showing very good values in the whole considered temperature range (i.e., from 0°C to 80°C).

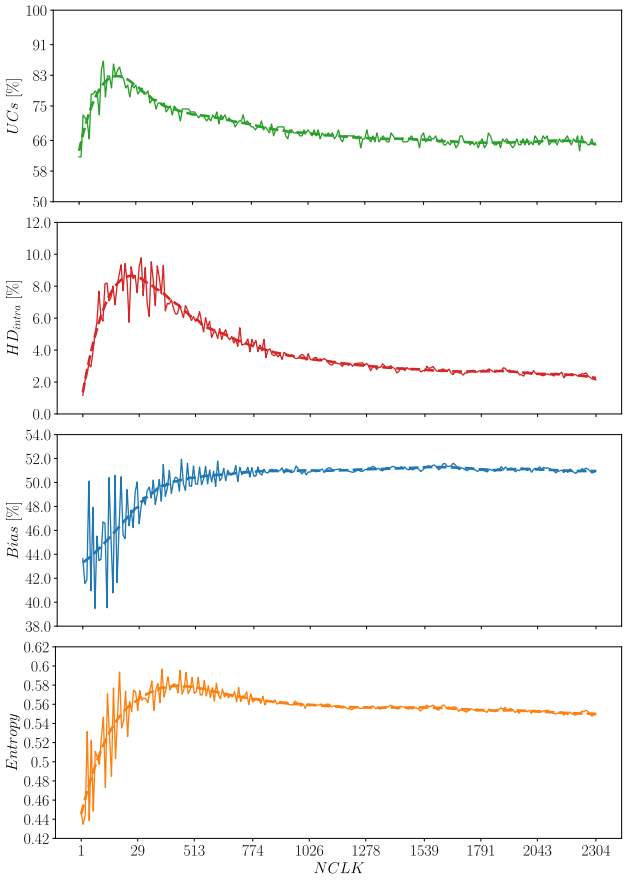

FIGURE 7. Characterization of the DD-Cell in terms of UCs in percentage, HD_{intra} in percentage, average bias, and Shannon Entropy, as a function of $NCLK$ (i.e., the time interval expressed in number of clk_H cycles in which the signal S is high).

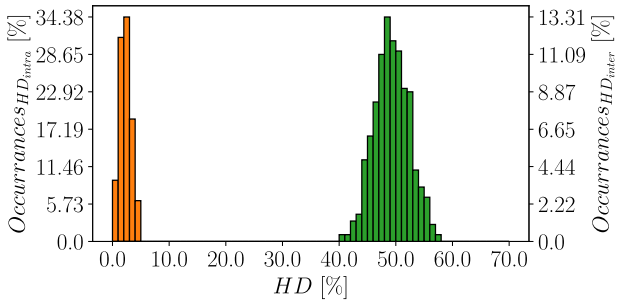

FIGURE 8. Intra-HD and Inter-HD on 32 different Artix-7 FPGAs.

D. TRNG DESIGN AND XOR COMBINING TO IMPROVE ENTROPY

In order to optimize the design of the proposed TRNG, a preliminary study to find the value of $NCLK$ which maximizes the number of UCs has been carried out over 32 boards. The results of this study are reported in Fig. 11, and show that the mean value of the optimum $NCLK$ is 17, whereas its standard deviation is about 3. These preliminary results have shown that, even considering the optimum $NCLK$, the single DD-Cell is not able to pass NIST or AIS-31 tests. Therefore,

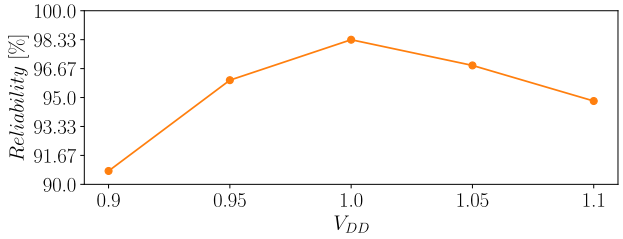

FIGURE 9. Reliability of the DD-PUF on the reference board when supply voltage varies in the range of $V_{DD} \pm 10\%$.

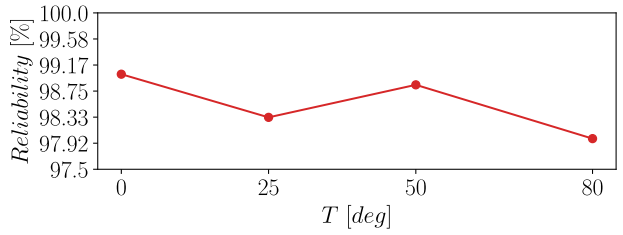

FIGURE 10. Reliability of the DD-PUF on the reference board when temperature varies in the range from 0°C to 80°C.

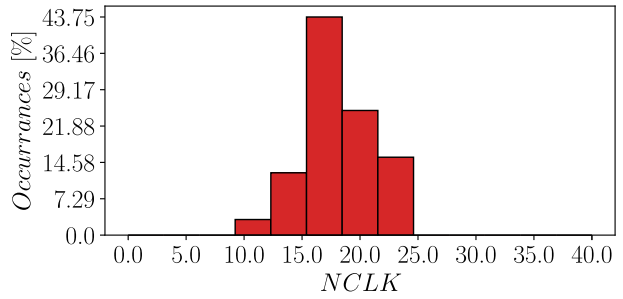

FIGURE 11. Distribution of the optimum value of *NCLK* across 32 different FPGA boards.

with the aim of improving the randomness performance of the TRNG and increase uncorrelation between multiple identical DD-cells, we have exploited the XOR combining approach presented in [27]. The XOR combining of multiple identical DD-cells resulted in an unbiased (i.e. bias ≈ 0.5) and uncorrelated bitstream. As a further optimization, the minimum number of XOR-combining rounds needed to pass NIST tests in typical conditions, while guaranteeing an high entropy value set as margin to account for PVT variations, has been investigated by several measurements. The results of this characterization across the 32 FPGA boards are summarized in Fig. 12, showing that the trend is in the order of 5 XOR rounds, with a minimum of 4, and a maximum of 7 XOR rounds.

For the rest of this characterization we consider the reference FPGA board with the adjustable power supply voltage. For this reference FPGA, the optimum *NCLK* has resulted to be 16 (very close to the mean value) and the

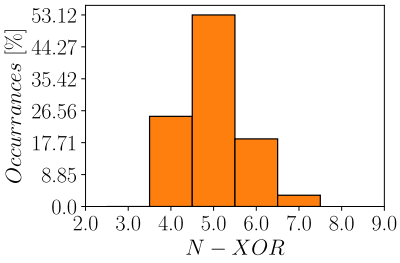

FIGURE 12. Distribution of the minimum number of XOR rounds needed to pass NIST tests in typical conditions across 32 different FPGAs.

TABLE 2. Estimated (Est.) byte-entropy by T8-test of AIS-31 vs Environmental (Env.) conditions.

Env. Condition	0.9V , 25°C	1.0V, 25°C	1.1V, 25°C	1.0V, 0°C	1.0V, 50°C	1.0V, 75°C
Est. Byte-Entropy	7.99631	7.99617	7.99992	7.99634	7.99715	7.99913

minimum number of XOR rounds needed to pass NIST tests in typical conditions has been found to be 4 (the minimum value). Therefore the proposed TRNG on the reference FPGA exhibits a throughput of:

$$TP = N_{bits} \cdot \frac{f_{clk_H}}{NCLK \cdot (2^{Xor_{rounds}})} = 225Mbit/s \qquad (32)$$

where N_{bits} is the number of instantiated DD-cells (equal to 128 in these experiments). Obviously, according to eq. 32, the TP will be lower on FPGA devices which require an higher value of *NCLK* and/or an higher value of XOR rounds.

E. TRNG CHARACTERIZATION

Performances of the DD-Cells configured as TRNG have been characterized without considering any off-chip post-processing (e.g. Von Neumann). To assess randomness of the extracted bitstream the NIST SP 800–22, NIST SP 800-90B and AIS-31 tests mentioned in section V-C have been used. Restart experiment verification has also been performed to assess that at each start-up the DD-Cell generates a bitstream which results uncorrelated with the ones generated before. In addition, bitstream randomness behavior has been extensively validated on different supply voltage and temperature conditions, and across 32 different FPGA devices.

F. RESTART EXPERIMENT TEST

In order to verify the Uniqueness of the bitstream at each extraction, the correlation between multiple sequences of 1024 bits extracted the FIFO has been measured. More in detail, the correlation has been computed among 1000 sequences of 1024 bits extracted from 1000 identical restart experiments, according to [27] and [45], and it has been found that the correlation follows a Normal Distribution with mean value $\mu \approx 1.834 \cdot 10^{-5}$ and standard deviation $\sigma \approx 0.0013$.

TABLE 3. Results of the NIST SP800-90B tests at V_{DD}=1V.

Test			Result			
			C[i][0]	C[i][1]	C[i][2]	Pass
Excursion			21	0	6	✓
NumDirectionalRuns			6	0	53	✓
LenDirectionalRuns			0	6	1	✓
NumIncreasesDecreases			6	0	7	✓
NumRunsMedian			6	0	12	✓
LenRunsMedian			2	4	5	✓
AvgCollision			6	0	205	✓
MaxCollision			46	4	2	✓
Permutation tests	Periodicity	(1)	210	1	5	✓
		(2)	6	0	12	✓
		(8)	57	0	6	✓
		(16)	6	0	27	✓
		(32)	6	0	7	✓
	Covariance	(1)	16	0	6	✓
		(2)	6	0	12	✓
		(8)	6	0	390	✓
		(16)	6	0	52	✓
		(32)	6	0	14	✓
	Compression		6	0	153	✓

Test		Result			
		Score	degrees of freedom	p-value	Pass
χ^2 tests	Independence	64988.402497	65280	0.789985	✓
	Goodness of fit	2340.630787	2295	0.248590	✓

Test		Result
Entropy	H-bitstring	0.999198
	H-original	7.946989
Longest Repeated Substring test		✓

G. RESULTS OF THE AIS-31 TESTS FOR DIFFERENT VALUES OF THE SUPPLY VOLTAGE AND TEMPERATURE

The AIS-31 tests have been executed on the 32 bitstreams extracted from the 32 devices and each bitstream has resulted to satisfy tests requirements. Tests results on the reference FPGA board have been reported in Tab. 2. AIS-31 tests have been also performed on bitstreams extracted from the reference FPGA board in different supply voltage and temperature conditions. As it can be observed, when voltage or temperature increases the overall entropy of the sequence increases. This behavior is due to the fact that the propagation delay of the logic is lower when voltage or temperature increases, hence for the same value of *NCLK* (which is set by an environmental independent reference circuit (i.e. a PLL)), the accumulated Jitter (which is related to the propagation delay difference of the two paths) is greater. It is worth noting that a reliable TRNG has to satisfy randomness performance in each corner condition. Our design approach ensures a robust random behavior, even when facing PVT variations. Unlike other methods that require additional post-processing techniques or extra XOR rounds, our approach does not compromise performance. It inherently maintains a high level of randomness without the need for any subsequent modifications or additions.

H. NIST TESTS VALIDATION

NIST SP800-90B tests have been executed considering a bitstream length of 100 Mbits. Also these tests have been executed on 32 different FPGA boards, and results have confirmed that bitstreams extracted from the proposed TRNG on each FPGA are able to pass tests requirements. Results of the NIST SP800-90B tests suite on TRNG implemented on the reference FPGA are reported in Tab. 3. As it can be observed, the bitstream extracted from the DD-Cells configured as TRNG satisfies random requirement needed to pass these tests, achieving a byte entropy of 7.946989 and thus a bit entropy of 0.999198.

To provide further validation of the proposed TRNG, the NIST SP 800–22 tests have been executed considering also supply voltage variations, and results are reported in Tab. 4, confirming the capability of the proposed TRNG to pass

TABLE 4. Results of the NIST SP800-22 tests for different values of V_{DD}.

	0.9 V			1.0 V			1.1 V		
Test Name	**p-value**	**pass-rate**	**result**	**p-value**	**pass-rate**	**result**	**p-value**	**pass-rate**	**result**
Frequency	0.1626	98/100	✓	0.3669	100/100	✓	0.0911	100/100	✓
Block Frequency	0.4019	100/100	✓	0.9963	99/100	✓	0.4372	100/100	✓
Cumulative Sums	0.6786/0.3669	99/100	✓	0.0590/0.9642	100/100	✓	0.2368/0.9643	100/100	✓
Runs Test	0.3838	100/100	✓	0.5544	0.5544	✓	0.2368	100/100	✓
Longest Run	0.2022	98/100	✓	0.1538	97/100	✓	0.1296	99/100	✓
Rank	0.8343	99/100	✓	0.2493	97/100	✓	0.1626	96/100	✓
FFT	0.5141	98/100	✓	0.5955	99/100	✓	0.5341	99/100	✓
Non Overlapping	[0.0191,0.9915]	[96,100]/100	✓	[0.0119,0.9963]	[96,100]/100	✓	[0.0668,0.9978]	[96,100]/100	✓
Overlapping	0.6579	98/100	✓	0.7197	98/100	✓	0.6371	100/100	✓
Universal	0.7399	100/100	✓	0.3505	100/100	✓	0.7399	99/100	✓
Approximate Entropy	0.0661	97/100	✓	0.4944	97/100	✓	0.0428	98/100	✓
Random Excursion	[0.1223,0.7399]	12/12	✓	[0.0909,0.1626]	[16/17]/17	✓	[0.1023,0.8132]	[9/9]	✓
Random Excursion Variant	[0.0668,0.7399]	12/12	✓	[0.0126,0.4373]	[16/17]/17	✓	[0.0268,0.7512]	[9/9]	✓
Serial	[0.3838,0.8677]	[97/98]/100	✓	[0.0669,0.4373]	[99,100]/100	✓	0.0669/0.1917	[98/100]/100	✓
Linear Complexity	0.1816	100/100	✓	0.1373	99/100	✓	0.7792	96/100	✓

TABLE 5. Comparison Table of TRNG performances.

		Area$^{\diamond}$				O.F.	TP	Byte	Bit	T	V_{DD}	FOM_T	FOM_E	Platform
		LUTs	**FFs**	**CC**	**Slices**	**[MHz]**	**[Mbit/s]**	**Entropy**‡	**Entropy***	**range [°C]**	**range [%]**$^{+}$	$\left[\frac{\text{bits}}{\text{Slice}}\right]$	$\left[\frac{\text{bits}}{\text{Slice}}\right]$	
PUF+TRNG	DD-This Work	256	256	0	64	450	225	7.9470	0.9992	[25,75]	[-10,+10]	$3.9 \cdot 10^{-3\,\lambda}$	1.044^{λ}	Artix-7
	UTERO [75]	-	-	-	-	18	0.9		-	[-13,100]	[-10,+10]	-	-	Actel Fusion
	RO [39]	1288	945	0	322	-	3.2	-	-	TYP	TYP	-	-	Spartan-3
	RO [40]	4325	122	0	1083	395.33	0.024	-	-	TYP	TYP	$0.056 \cdot 10^{-3}$	-	Artix-7
	RO [21]	83	27	0	21	100	100	-	-	[0,80]	[-10,+10]	$45.45 \cdot 10^{-3}$	-	Virtex-6
TRNG	LX [27]	32	0	0	8	50	12.5	7.9979	0.9997	[25,75]	[-10,+10]	$27.77 \cdot 10^{-3\,\lambda}$	**13.228**$^{\lambda}$	Spartan-6
	TROT [83]	32	55	17	33	125	12.5	7.996	0.9995	[-20,70]	[-10,TYP]	$3.03 \cdot 10^{-3}$	0.757	Zynq-7000
	ES [84]	7	6	1	4	100	10	-	-	TYP	TYP	$2.5 \cdot 10^{-3}$	-	Spartan-6
	LRO [45]	4	3	0	**1**	50	0.76	**7.9983**	**0.9998**	[25,75]	[-10,+10]	$15.15 \cdot 10^{-3}$	8.992	Spartan-6
	RO [43]	528	177	0	270	24	6	7.9946	0.9993	[25,75]	TYP	$0.26 \cdot 10^{-3}$	0.048	Spartan-3A
	RO [85]	712	753	0	-	-	3.2	-	-	TYP	TYP	-	-	Spartan-3E
	RO [44]	10	5	1	$3^{\dagger}$	100	1.15	7.9760$^{\intercal}$	0.9970$^{\intercal}$	TYP	TYP	$3.83 \cdot 10^{-3\,\dagger}$	0.160	Spartan-6
	RO [47]	521	131	0	-	-	2.57	7.9920	0.9990	TYP	TYP	-	-	Spartan-6
	DNO [86]	15	0	0	$4^{\dagger}$	100	100	-	-	TYP	TYP	$\mathbf{250 \cdot 10^{-3}}$	-	Artix-7
	Meta-stability [87]	56	19	0	14	400	100	-	-	[0,80]	[-10,+10]	$17.86 \cdot 10^{-3}$	-	Virtex-6
	Meta-stability [88]	-	-	-	797	100	79.18	7.9952	0.9994	TYP	TYP	$0.99 \cdot 10^{-3}$	0.206	Artix-7
	Meta-stability [53]	-	256	-	580	100	12.5	-	-	TYP	TYP	$0.22 \cdot 10^{-3}$	-	Virtex-4
	Meta-stability [52]	128	-	-	-	-	2	-	-	TYP	TYP	-	-	Virtex-5
	STR [89]	616	616	0	$154^{\dagger}$	400	100	7.9760$^{\intercal}$	0.9970$^{\intercal}$	TYP	TYP	$1.623 \cdot 10^{-3}$	0.068	Virtex-5
	MSFRO [50]	25	2	0	7	-	**290**	-	-	[0,80]	[-20,+20]	-	-	Virtex-6
	Jitter [90]	-	-	-	32	$63^{<}$	190	-	-	[0,80]	[-20,+20]	$94 \cdot 10^{-3}$	-	Virtex-6

*Bit Entropy computed from the H_8; $^{\lambda}$Extra resources considered for the XOR combining function; *Flip-Flops used as latches; $^{\diamond}$Area of the TRNG cell without considering control logic; †Estimated considering the Xilinx Spartan-6 or Artix-7 CLBs [91]; $^{\intercal}$Estimated ; ‡Estimated with T8 test of the AIS-31; $^{+}$ Tested supply voltage relative variations in percentage; $^{<}$ Estimated from the Operating Period.

NIST tests also for a ±10% supply voltage variations without any post-processing technique.

VII. COMPARISON WITH STATE OF THE ART

A. COMPARISON WITH OTHER TRNG

An overall comparison of FPGA-compatible TRNGs has been carried out and results are shown in Tab. 5. Regarding throughput, the proposed PUF+TRNG architecture has the fastest TRNG with respect to others PUF+TRNG primitives in the literature, whereas the proposed TRNG results among the fastest TRNG-only designs. Only [50] achieves an higher throughput, but its operating frequency is not declared, making it difficult to compare it with respect to the TP and operating frequency tradeoff. It has also to be remarked that the TRNG in [21] achieves a better tradeoff with respect to FOM, despite having a lower throughput than the proposed design. The resources usage spread is due to the high entropy extracted to ensure a random bitstream with respect to voltage and temperature variations, as evidenced by the high FOM_E. Compared to other TRNGs in the literature, the proposed design has a significantly higher FOM_E, except for the TRNGs in [27] and [45], which, however exhibit a very low throughput.

TABLE 6. Comparison Table of PUF performances.

			Nominal				PVT					FOMs·10^{-2}				FOMs·10^{-2}			
		Platform	Uniqueness	Slice/bit	BER_{typ}	Reliability	Δ_V [V]	V_{Typ} [V]	BER_{wc_V}	Δ_T [deg]	BER_{wc_T}	FOM_{HD}	$FOM_{BER_V^\bullet}$	$FOM_{BER_T^\circ}$	$FOM_{BER_{V,T}^\dagger}$	$\hat{FOM}_{HD}$	$\hat{FOM}_{BER_V^\bullet}$	$\hat{FOM}_{BER_T^\circ}$	$\hat{FOM}_{BER_{V,T}^\dagger}$
PUF+TRNG	**DD** (This Work)	Artix-7	49.48	0.50	1.67	98.33	0.20	1.00	9.23	75.00	3.50	57.17	2.17	47.80	2.16	114.35	4.33	95.60	4.33
	UTERO [75]	Actel Fusion	48.82	-	2.37	97.63	0.30	1.50	3.40	113.00	3.00	37.77	5.83	40.39	5.82	-	-	-	-
	RO [39]	Spartan-3	39.90	11.00	22.10	77.90	-	-	-	-	-	4.12	-	-	-	0.37	-	-	-
	RO [40]	Artix-7	48.47	181.00	1.30	98.70	0.48	1.20	10.00	40.00	7.00	49.81	3.99	20.41	3.93	0.28	0.022	0.11	0.022
	RO [21]	Virtex-6	49.79	21	3.02	96.98	-	-	-	-	-	33.03	-	-	-	1.573	-	-	-
PUF	**NAND** [32]	Artix-7	49.50	0.50	1.38	98.62	0.18	0.90	6.70	75.00	3.50	68.13	2.98	53.51	2.98	136.26	5.97	107.03	5.96
	Meta-XOR [65]	Artix-7	49.47	0.50	1.06	98.94	0.20	1.00	6.25	-	-	84.38	3.20	-	-	168.76	6.40	-	-
	SS-RO$^{\ddagger\times}$ [62]	Artix-7	48.05	1.00	0.70	99.30	-	-	-	-	-	48.27	-	-	-	48.27	-	-	-
	PICO [34]	Artix-7	49.90	1.00	5.47	94.53	0.20	1.00	8.60	75.00	3.54	18.28	2.31	17.81	2.31	18.28	2.31	17.81	2.31
	NAND$^{\times}$ [67]	Spartan-6	49.24	2.00	0.82	99.18	0.12	1.20	2.46	85.00	4.06	89.44	4.06	65.43	4.06	44.72	2.03	32.72	2.03
	NAND [66]	Spartan-6	49.00	2.00	0.86	99.14	0.12	1.20	5.30	-	-	75.82	1.89	-	-	37.91	0.94	-	-
	Meta [65]	Spartan-6	49.03	0.50	2.46	97.54	0.24	1.20	10.89	-	-	37.82	1.84	-	-	75.63	3.67	-	-
	DD [41]	Spartan-6	49.28	0.50	1.63	98.37	0.24	1.20	9.00	75.00	2.00	56.12	2.22	56.12	2.22	112.24	4.44	112.24	4.44
	PICO [68]	Spartan-6	49.93	1.00	6.04	93.96	0.24	1.20	9.13	75.00	6.46	16.56	2.17	15.45	2.16	16.56	2.17	15.45	2.17
	NAND [66]	Spartan-3	46.00	2.00	2.40	97.60	-	1.20	-	85.00	5.00	21.44	-	34.75	-	10.72	-	17.37	-
	Butterfly [59]	Virtex-5	-	-	6.00	94.00	-	-	-	-	-	-	-	-	-	-	-	-	-

‡ Single Slice Ring Oscillator,$^{\bullet}$ with considering only voltage variations;$^{\circ}$ with considering only temperature variations; † with considering voltage and temperature variations; $^{\times}$ post-processing required.

TABLE 7. Comparison Table of PUF+TRNG performances.

		PUF				TRNG				PUF+TRNG
	Platform	Slice/bit	Uniqueness	Reliability	$\hat{FOM}_{HD}$	Slice	TP [Mbit/s]	O.F. [MHz]	FOM_T	FOM_{uni}
This Work	Artix-7	0.500	49.480	98.330	114.345	64.000	225	450	0.008	0.893
[75]	Actel Fusion	-	48.82	97.63	-	-	0.9	18	-	-
[39]	Spartan-3	11	39.9	77.9	0.374	322	3.2	-	-	-
[40]	Artix-7	181	48.47	98.7	0.275	1083	0.024	395	0.06μ	0.02μ
[21]	Virtex-6	21	49.79	96.98	1.573	21	100	100	0.048	0.075

B. COMPARISON WITH OTHER PUF

The proposed DD-Cell configured as PUF has then been compared with other FPGA-integrated PUF architectures and main metrics are summarized in Table 6. For what concerns the *Uniqueness* and the *Reliability* in nominal conditions, the proposed PUF results comparable with the state-of-the-art.

With respect to the resources usage, the proposed DD-PUF is among the most compact PUF-only designs in the literature and results the most compact PUF+TRNG primitive. It can include up to 2 bits/Slice which is 42 times better than [21], which was the most compact PUF+TRNG in the literature. In terms of resources usage, it is comparable to other PUF-primitives employed in [34], [41], [62], and [68] which occupy two LUTs and two flip-flops. With respect to the FOM, $\hat{FOM}_{HD}$ it outperforms all the other PUF+TRNG designs in the literature. Also with respect to the $\hat{FOM}_{BER_{V,T}}$ it outperform other works and is second to only [32].

C. COMPARISON WITH OTHER PUF+TRNG

In order to compare PUF+TRNG designs, we have focused on some performance metrics which have been evaluated in all the recent papers dealing with FPGA-compatible PUF+TRNG primitives, such as *Uniqueness*, *Reliability*, resources usage and throughput, and we have combined them in the following Figure Of Merit:

$$FOM_{uni} = \frac{TP\left(\frac{bits}{Slice_{PUF}}\right)}{OF \cdot Slice_{TRNG} \cdot \sqrt{HD_{intra}^2 + (0.5 - HD_{inter})^2}} = FOM_T \cdot FOM_{HD} \quad (33)$$

in which $\frac{bits}{Slice_{PUF}}$ denotes the resources usage in terms of bits/Slice of the PUF and with $Slice_{TRNG}$ the number of Slices used by the TRNG.

Results of the comparison reported in Tab. 7 show that the proposed PUF+TRNG primitive reaches the best trade-off among PUF and TRNG performance. This is due to the fact that our design has been derived from the DD-PUF, which was already a state-of-the-art design and then has been optimized to perform as a TRNG with state-of-the-art performance in terms of TP and resources usage. For what concerns [75], it can not be compared with other works since the resources usage on the Actel Fusion has not been declared. For what concerns the PUF+TRNG in [39], the FOM_{uni} can not be derived since the operating frequency has not been reported. In addition, we want to remark that further FOMs for PUF and TRNG (e.g. the FOM_E and FOM_{BER}) can not be

included in the comparison table because most PUF+TRNG in the literature have not been characterized with respect to all these parameters, or in some cases some important tests (e.g. NIST tests or Reliability under PVT variations) are missing. In this work a full PUF and TRNG characterization has been carried out and hence all the FOMs respectively for PUF and TRNG performance can be computed.

VIII. CONCLUSION

In this paper we have proposed an FPGA-compatible PUF+TRNG primitive based on the DD-Cell as the basic entropy source. A theoretical model of the DD-Cell explaining the PUF and the TRNG behaviour of the DD-Cell has been presented, and the relations between main performance figures and design parameters have been evaluated. The proposed PUF+TRNG architecture has been implemented on the Artix-7 FPGA platform, and an extensive measurement campaign involving 32 FPGA boards has been carried out. It has to be noted that performance of the DD-Cell architecture is strongly dependent on the design strategy. This is because the performance of metastable cells, such as the DD-Cell, depends on symmetries of two nominally identical branches [32], [34], [41], [62], [68]. The analytical study carried out in this work has highlighted this limitation, as the bias of the response is strongly related to the matching of the two routing delays of the intra-slice connections. However, it is also important to note that state-of-the-art performance can be achieved on the different FPGA platforms if the design guidelines reported in this work are followed.

Measured performances of the proposed PUF have been compared against state of the art PUFs showing Uniqueness and Reliability comparable to the state of the art. In terms of resources usage, it includes 2 bits/Slice which is 42 times better than the most compact PUF+TRNG in the literature. The proposed PUF+TRNG design exhibits the fastest TRNG with respect to others PUF+TRNG primitives in the literature, whereas it results among the fastest TRNG-only designs. The comparison against the PUF+TRNG designs available in the literature has shown that the proposed solution exhibits the best trade-off among PUF and TRNG performance, providing the most compact PUF and the highest throughput TRNG, with overall good performances for both cryptographic primitives.

REFERENCES

[1] B. Halak, M. Zwolinski, and M. S. Mispan, "Overview of PUF-based hardware security solutions for the Internet of Things," in *Proc. IEEE 59th Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Oct. 2016, pp. 1–4.

[2] 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.

[3] Y. Yilmaz, S. R. Gunn, and B. Halak, "Lightweight PUF-based authentication protocol for IoT devices," in *Proc. IEEE 3rd Int. Verification Secur. Workshop (IVSW)*, Jul. 2018, pp. 38–43.

[4] M. A. Qureshi and A. Munir, "PUF-RAKE: A PUF-based robust and lightweight authentication and key establishment protocol," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 4, pp. 2457–2475, Jul. 2022.

[5] R. E. Navas, M. Lagos, L. Toutain, and K. Vijayasankar, "Nonce-based authenticated key establishment over OAuth 2.0 IoT proof-of-possession architecture," in *Proc. IEEE 3rd World Forum Internet Things (WF-IoT)*, Dec. 2016, pp. 317–322.

[6] A. Aysu, E. Gulcan, D. Moriyama, P. Schaumont, and M. Yung, "End-to-end design of a PUF-based privacy preserving authentication protocol," in *Cryptographic Hardware and Embedded Systems—CHES 2015*. Berlin, Germany: Springer, Sep. 2015, pp. 556–576.

[7] Y. Yilmaz, L. Aniello, and B. Halak, "ASSURE: A hardware-based security protocol for resource-constrained IoT systems," *J. Hardw. Syst. Secur.*, vol. 5, no. 1, pp. 1–18, Mar. 2021.

[8] Y. Yilmaz, V.-H. Do, and B. Halak, "ARMOR: An anti-counterfeit security mechanism for low cost radio frequency identification systems," *IEEE Trans. Emerg. Topics Comput.*, vol. 9, no. 4, pp. 2125–2138, Oct. 2021.

[9] P. Tuyls and L. Batina, "RFID-tags for anti-counterfeiting," in *Topics in Cryptology—CT-RSA 2006*. Berlin, Germany: Springer, 2006, pp. 115–131.

[10] S. Devadas, E. Suh, S. Paral, R. Sowell, T. Ziola, and V. Khandelwal, "Design and implementation of PUF-based 'unclonable' RFID ICs for anti-counterfeiting and security applications," in *Proc. IEEE Int. Conf. RFID*, Apr. 2008, pp. 16–17.

[11] Y. S. Lee, T. Y. Kim, and H. J. Lee, "Mutual authentication protocol for enhanced RFID security and anti-counterfeiting," in *Proc. 26th Int. Conf. Adv. Inf. Netw. Appl. Workshops*, Mar. 2012, pp. 558–563.

[12] V. Immler, J. Obermaier, M. König, M. Hiller, and G. Sig, "B-TREPID: Batteryless tamper-resistant envelope with a PUF and integrity detection," in *Proc. IEEE Int. Symp. Hardw. Oriented Secur. Trust (HOST)*, Apr. 2018, pp. 49–56.

[13] Y. Zheng, Y. Cao, and C.-H. Chang, "A PUF-based data-device hash for tampered image detection and source camera identification," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 620–634, 2020.

[14] R. Gupta, A. Pandey, and R. K. Baghel, "FPGA implementation of chaos-based high-speed true random number generator," *Int. J. Numer. Model., Electron. Netw., Devices Fields*, vol. 10, p. e2604, Apr. 2019.

[15] M. Stipcevic, "Quantum random number generators and their applications in cryptography," in *Proc. SPIE*, vol. 8375, pp. 20–34, May 2012.

[16] D. Stucki, S. Burri, E. Charbon, C. Chunnilall, A. Meneghetti, and F. Regazzoni, "Towards a high-speed quantum random number generator," *Proc. SPIE*, vol. 8899, pp. 129–134, Oct. 2013.

[17] F. Acerbi, N. Massari, L. Gasparini, A. Tomasi, N. Zorzi, G. Fontana, L. Pavesi, and A. Gola, "Structures and methods for fully-integrated quantum random number generators," *IEEE J. Sel. Topics Quantum Electron.*, vol. 26, no. 3, pp. 1–8, May 2020.

[18] S. Takahashi and K. Iwamura, "Secret sharing scheme suitable for cloud computing," in *Proc. IEEE 27th Int. Conf. Adv. Inf. Netw. Appl. (AINA)*, Mar. 2013, pp. 530–537.

[19] H. K. Lee, T. Malkin, and E. Nahum, "Cryptographic strength of SSL/TLS servers: Current and recent practices," in *Proc. 7th ACM SIGCOMM Conf. Internet Meas.*, New York, NY, USA, Oct. 2007, pp. 83–92.

[20] Y. Cao, W. Liu, L. Qin, B. Liu, S. Chen, J. Ye, X. Xia, and C. Wang, "Entropy sources based on silicon chips: True random number generator and physical unclonable function," *Entropy*, vol. 24, no. 11, p. 1566, Oct. 2022.

[21] Y. Wang, H. Liang, Y. Wang, L. Yao, M. Yi, Z. Huang, and Y. Lu, "A reconfigurable PUF structure with dual working modes based on entropy separation model," *Microelectron. J.*, vol. 124, Jun. 2022, Art. no. 105445.

[22] R. Della Sala, D. Bellizia, F. Centurelli, and G. Scotti, "A monostable physically unclonable function based on improved RCCMs with 0–1.56% native bit instability at 0.6–1.2 v and 0–75 °C," *Electronics*, vol. 12, no. 3, p. 755, Feb. 2023.

[23] X. Zhao, P. Gan, Q. Zhao, D. Liang, Y. Cao, X. Pan, and A. Bermak, "A 124 fJ/bit cascode current mirror array based PUF with 1.50% native unstable bit ratio," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 66, no. 9, pp. 3494–3503, Sep. 2019.

[24] S. Satpathy, S. K. Mathew, V. Suresh, M. A. Anders, H. Kaul, A. Agarwal, S. K. Hsu, G. Chen, R. K. Krishnamurthy, and V. K. De, "A 4-fJ/b delay-hardened physically unclonable function circuit with selective bit destabilization in 14-nm trigate CMOS," *IEEE J. Solid-State Circuits*, vol. 52, no. 4, pp. 940–949, Apr. 2017.

[25] K. Yang, Q. Dong, D. Blaauw, and D. Sylvester, "A 553F^2 2-transistor amplifier-based physically unclonable function (PUF) with 1.67% native instability," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2017, pp. 146–147.

[26] S. K. Mathew, S. K. Satpathy, M. A. Anders, H. Kaul, S. K. Hsu, A. Agarwal, G. K. Chen, R. J. Parker, R. K. Krishnamurthy, and V. De, "1A 0.19pJ/b PVT-variation-tolerant hybrid physically unclonable function circuit for 100% stable secure key generation in 22 nm CMOS," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2014, pp. 278–279.

[27] R. D. Sala, D. Bellizia, and G. Scotti, "High-throughput FPGA-compatible TRNG architecture exploiting multistimuli metastable cells," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 69, no. 12, pp. 4886–4897, Dec. 2022, doi: 10.1109/TCSI.2022.3199218.

[28] R. D. Sala, F. Centurelli, and G. Scotti, "A novel differential to single-ended converter for ultra-low-voltage inverter-based OTAs," *IEEE Access*, vol. 10, pp. 98179–98190, 2022.

[29] I. Baturone, M. A. Prada-Delgado, and S. Eiroa, "Improved generation of identifiers, secret keys, and random numbers from SRAMs," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 12, pp. 2653–2668, Dec. 2015.

[30] J.-L. Danger, R. Yashiro, T. Graba, Y. Mathieu, A. Si-Merabet, K. Sakiyama, N. Miura, and M. Nagata, "Analysis of mixed PUF-TRNG circuit based on SR-latches in FD-SOI technology," in *Proc. 21st Euromicro Conf. Digit. Syst. Design (DSD)*, Aug. 2018, pp. 508–515.

[31] S. K. Satpathy, S. K. Mathew, R. Kumar, V. Suresh, M. A. Anders, H. Kaul, A. Agarwal, S. Hsu, R. K. Krishnamurthy, and V. De, "An all-digital unified physically unclonable function and true random number generator featuring self-calibrating hierarchical von Neumann extraction in 14-nm tri-gate CMOS," *IEEE J. Solid-State Circuits*, vol. 54, no. 4, pp. 1074–1085, Apr. 2019.

[32] R. Della Sala and G. Scotti, "A novel FPGA implementation of the NAND-PUF with minimal resource usage and high reliability," *Cryptography*, vol. 7, no. 2, p. 18, Apr. 2023.

[33] N. N. Anandakumar, M. S. Hashmi, and S. K. Sanadhya, "Design and analysis of FPGA-based PUFs with enhanced performance for hardware-oriented security," *ACM J. Emerg. Technol. Comput. Syst.*, vol. 18, no. 4, pp. 1–26, Oct. 2022.

[34] C. Gu, N. Hanley, and M. O'Neill, "Improved reliability of FPGA-based PUF identification generator design," *ACM Trans. Reconfigurable Technol. Syst.*, vol. 10, pp. 1–23, May 2017.

[35] A. P. Johnson, R. S. Chakraborty, and D. Mukhopadhyay, "A PUF-enabled secure architecture for FPGA-based IoT applications," *IEEE Trans. Multi-Scale Comput. Syst.*, vol. 1, no. 2, pp. 110–122, Apr. 2015.

[36] N. N. Anandakumar, M. S. Hashmi, and S. K. Sanadhya, "Compact implementations of FPGA-based PUFs with enhanced performance," in *Proc. 30th Int. Conf. VLSI Design 16th Int. Conf. Embedded Syst. (VLSID)*, Jan. 2017, pp. 161–166.

[37] N. N. Anandakumar, M. S. Hashmi, and S. K. Sanadhya, "Efficient and lightweight FPGA-based hybrid PUFs with improved performance," *Microprocessors Microsyst.*, vol. 77, Sep. 2020, Art. no. 103180.

[38] N. N. Anandakumar, M. S. Hashmi, and M. Tehranipoor, "FPGA-based physical unclonable functions: A comprehensive overview of theory and architectures," *Integration*, vol. 81, pp. 175–194, Nov. 2021.

[39] A. Maiti, R. Nagesh, A. Reddy, and P. Schaumont, "Physical unclonable function and true random number generator: A compact and scalable implementation," in *Proc. 19th ACM Great Lakes Symp. VLSI*, New York, NY, USA, May 2009, pp. 425–428.

[40] I. Baturone, R. Román, and Á. Corbacho, "A unified multibit PUF and TRNG based on ring oscillators for secure IoT devices," *IEEE Internet Things J.*, vol. 10, no. 7, pp. 6182–6192, Apr. 2023.

[41] R. Della Sala, D. Bellizia, and G. Scotti, "A novel ultra-compact FPGA PUF: The DD-PUF," *Cryptography*, vol. 5, no. 3, p. 23, Sep. 2021.

[42] R. D. Sala and G. Scotti, "The DD-cell: A double side entropic source exploitable as PUF and TRNG," in *Proc. 17th Conf. Ph.D Res. Microelectron. Electron. (PRIME)*, Jun. 2022, pp. 353–356.

[43] N. N. Anandakumar, S. K. Sanadhya, and M. S. Hashmi, "FPGA-based true random number generation using programmable delays in oscillator-rings," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 67, no. 3, pp. 570–574, Mar. 2020.

[44] B. Yang, V. Rožic, M. Grujic, N. Mentens, and I. Verbauwhede, "ES-TRNG: A high-throughput, low-area true random number generator based on edge sampling," *IACR Trans. Cryptograph. Hardw. Embedded Syst.*, vol. 10, pp. 267–292, Aug. 2018.

[45] R. D. Sala, D. Bellizia, and G. Scotti, "A novel ultra-compact FPGA-compatible TRNG architecture exploiting latched ring oscillators," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 69, no. 3, pp. 1672–1676, Mar. 2022.

[46] V. Fischer and M. Drutarovský, "True random number generator embedded in reconfigurable hardware," in *Cryptographic Hardware and Embedded Systems—CHES 2002*. Berlin, Germany: Springer, Feb. 2003, pp. 415–430.

[47] O. Petura, U. Mureddu, N. Bochard, V. Fischer, and L. Bossuet, "A survey of AIS-20/31 compliant TRNG cores suitable for FPGA devices," in *Proc. 26th Int. Conf. Field Program. Log. Appl. (FPL)*, Aug. 2016, pp. 1–10.

[48] Q. Tang, B. Kim, Y. Lao, K. K. Parhi, and C. H. Kim, "True random number generator circuits based on single- and multi-phase beat frequency detection," in *Proc. IEEE Custom Integr. Circuits Conf.*, Sep. 2014, pp. 1–4.

[49] N. Fujieda, M. Takeda, and S. Ichikawa, "An analysis of DCM-based true random number generator," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 67, no. 6, pp. 1109–1113, Jun. 2020.

[50] J. Cui, M. Yi, D. Cao, L. Yao, X. Wang, H. Liang, Z. Huang, H. Qi, T. Ni, and Y. Lu, "Design of true random number generator based on multi-stage feedback ring oscillator," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 69, no. 3, pp. 1752–1756, Mar. 2022.

[51] B. Sunar, W. Martin, and D. Stinson, "A provably secure true random number generator with built-in tolerance to active attacks," *IEEE Trans. Comput.*, vol. 56, no. 1, pp. 109–119, Jan. 2007.

[52] M. Majzoobi, F. Koushanfar, and S. Devadas, "FPGA-based true random number generation using circuit metastability with adaptive feedback control," in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst.* Cham, Switzerland: Springer, 2011, pp. 17–32.

[53] H. Hata and S. Ichikawa, "FPGA implementation of metastability-based true random number generator," *IEICE Trans. Inf. Syst.*, vol. E95.D, no. 2, pp. 426–436, 2012.

[54] F. Frustaci, F. Spagnolo, S. Perri, and P. Corsonello, "A high-speed FPGA-based true random number generator using metastability with clock managers," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 70, no. 2, pp. 756–760, Feb. 2023.

[55] C. Li, Q. Wang, J. Jiang, and N. Guan, "A metastability-based true random number generator on FPGA," in *Proc. IEEE 12th Int. Conf. ASIC (ASICON)*, Oct. 2017, pp. 738–741.

[56] L. Gong, J. Zhang, H. Liu, L. Sang, and Y. Wang, "True random number generators using electrical noise," *IEEE Access*, vol. 7, pp. 125796–125805, 2019.

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

[58] *Towards Hardware-Intrinsic Security*. Berlin, Germany: Springer, 2010.

[59] E. S. Kumar, J. Guajardo, R. Maes, G.-J. Schrijen, and P. Tuyls, "Extended abstract: The butterfly PUF protecting IP on every FPGA," in *Proc. IEEE Int. Workshop Hardw.-Oriented Secur. Trust*, Jul. 2008, pp. 67–70.

[60] 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.

[61] S. Eiroa and I. Baturone, "An analysis of ring oscillator PUF behavior on FPGAs," in *Proc. Int. Conf. Field-Program. Technol.*, Dec. 2011, pp. 1–4.

[62] C. Gu, C.-H. Chang, W. Liu, N. Hanley, J. Miskelly, and M. O'Neill, "A large-scale comprehensive evaluation of single-slice ring oscillator and PicoPUF bit cells on 28-nm Xilinx FPGAs," *J. Cryptograph. Eng.*, vol. 11, no. 3, pp. 227–238, Sep. 2021.

[63] S. Morozov, A. Maiti, and P. Schaumont, "An analysis of delay based PUF implementations on FPGA," in *Reconfigurable Computing: Architectures, Tools and Applications*. Berlin, Germany: Springer, 2010, pp. 382–387.

[64] J. Guajardo, S. S. Kumar, G.-J. Schrijen, and P. Tuyls, "FPGA intrinsic PUFs and their use for IP protection," in *Cryptographic Hardware and Embedded Systems—CHES 2007*. Berlin, Germany: Springer, 2007, pp. 63–80.

[65] R. D. Sala, D. Bellizia, and G. Scotti, "A lightweight FPGA compatible weak-PUF primitive based on XOR gates," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 69, no. 6, pp. 2972–2976, Jun. 2022.

[66] 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.

[67] B. Habib, J.-P. Kaps, and K. Gaj, "Efficient SR-latch PUF," in *Applied Reconfigurable Computing* Cham, Switzerland: Springer, Mar. 2015, pp. 205–216.
[68] C. Gu and M. O'Neill, "Ultra-compact and robust FPGA-based PUF identification generator," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2015, pp. 934–937.
[69] A. Cherkaoui, L. Bossuet, and C. Marchand, "Design, evaluation, and optimization of physical unclonable functions based on transient effect ring oscillators," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 6, pp. 1291–1305, Jun. 2016.
[70] U. Mureddu, B. Colombier, N. Bochard, L. Bossuet, and V. Fischer, "Transient effect ring oscillators leak too," in *Proc. IEEE Comput. Soc. Annu. Symp. VLSI (ISVLSI)*, Jul. 2019, pp. 37–42.
[71] F. Bernard, P. Haddad, V. Fischer, and J. Nicolai, "From physical to stochastic modeling of a TERO-based TRNG," *J. Cryptol.*, vol. 32, no. 2, pp. 435–458, Apr. 2019.
[72] L. Bossuet, X. T. Ngo, Z. Cherif, and V. Fischer, "A PUF based on a transient effect ring oscillator and insensitive to locking phenomenon," *IEEE Trans. Emerg. Topics Comput.*, vol. 2, no. 1, pp. 30–36, Mar. 2014.
[73] L. Bassham, A. Rukhin, J. Soto, J. Nechvatal, M. Smid, S. Leigh, M. Levenson, M. Vangel, N. Heckert, and D. Banks, "A statistical test suite for random and pseudorandom number generators for cryptographic applications," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep. Special Publication 800-22 revision 1a, Apr. 2010, p. 131.
[74] M. S. Turan, E. Barker, J. Kelsey, K. McKay, M. Baish, and M. Boyle, "Recommendation for the entropy sources used for random bit generation," in *Proc. CSRC NIST*, Jan. 2018, pp. 1–12.
[75] M. Varchola, M. Drutarovsky, and V. Fischer, "New universal element with integrated PUF and TRNG capability," in *Proc. Int. Conf. Reconfigurable Comput. FPGAs (ReConFig)*, Dec. 2013, pp. 1–6.
[76] T. Kacprzak, "Analysis of oscillatory metastable operation of an RS flip-flop," *IEEE J. Solid-State Circuits*, vol. JSSC-23, no. 1, pp. 260–266, Feb. 1988.
[77] M. Varchola and M. Drutarovsky, "New high entropy element for FPGA based true random number generators," in *Cryptographic Hardware and Embedded Systems—CHES 2010*. Berlin, Germany: Springer, 2010, pp. 351–365.
[78] M. Varchola and M. Drutarovsky, "New high entropy element for FPGA based true random number generators," in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst.* Cham, Switzerland: Springer, 2010, pp. 351–365.
[79] J. Hayya, D. Armstrong, and N. Gressis, "A note on the ratio of two normally distributed variables," *Manage. Sci.*, vol. 21, no. 11, pp. 1338–1341, Jul. 1975.
[80] *7 Series FPGAs Configurable Logic Block User Guide (UG474)*, Xilinx, San Jose, CA, USA, 2016.
[81] *Vivado Design Suite 7 Series FPGA and ZYNQ-7000 All Programmable SOC Libraries Guide (UG953)*, Xilinx, San Jose, CA, USA, 2019.
[82] *Artix-7 FPGAs Data Sheet: DC and AC Switching Characteristics (DS181)*, Xilinx, San Jose, CA, USA, 2022.
[83] M. Grujic and I. Verbauwhede, "TROT: A three-edge ring oscillator based true random number generator with time-to-digital conversion," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 69, no. 6, pp. 2435–2448, Jun. 2022.
[84] N. Klein, E. Harel, and I. Levi, "The cost of a true random bit—On the electronic cost gain of ASIC time-domain-based TRNGs," *Cryptography*, vol. 5, no. 3, p. 25, Sep. 2021.
[85] T. Addabbo, A. Fort, R. Moretti, M. Mugnaini, H. Takaloo, and V. Vignoli, "A new class of digital circuits for the design of entropy sources in programmable logic," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 7, pp. 2419–2430, Jul. 2020.
[86] X. Wang, H. Liang, Y. Wang, L. Yao, Y. Guo, M. Yi, Z. Huang, H. Qi, and Y. Lu, "High-throughput portable true random number generator based on jitter-latch structure," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 68, no. 2, pp. 741–750, Feb. 2021.
[87] N. N. Anandakumar, M. S. Hashmi, and S. K. Sanadhya, "Field programmable gate array based elliptic curve Menezes-Qu-Vanstone key agreement protocol realization using physical unclonable function and true random number generator primitives," *IET Circuits, Devices Syst.*, vol. 16, no. 5, pp. 382–398, Aug. 2022.
[88] A. Cherkaoui, V. Fischer, L. Fesquet, and A. Aubert, "A very high speed true random number generator with entropy assessment," in *Proc. 15th Int. Workshop Cryptogr. Hardw. Embedded Syst. (CHES)*. Berlin, Germany: Springer-Verlag, Aug. 2013, pp. 179–196.
[89] Y. Lu, H. Liang, L. Yao, X. Wang, H. Qi, M. Yi, C. Jiang, and Z. Huang, "Jitter-quantizing-based TRNG robust against PVT variations," *IEEE Access*, vol. 8, pp. 108482–108490, 2020.
[90] *Spartan-6 FPGA Configurable Logic Block UG384*, Xilinx, Xilinx, San Jose, CA, USA, 2010.

RICCARDO DELLA SALA was born in April 1996. He received the bachelor's and M.S. degrees (summa cum laude) in electronics engineering from the Sapienza University of Rome, Italy, in 2018 and 2020, respectively. His current research interests include the design and development of PUFs and TRNGs for hardware security. Furthermore, in the context of analog design, his research activity is focused on ultra-low voltage ultra-low power topology for the IoT and biomedical applications.

GIUSEPPE SCOTTI (Senior Member, IEEE) received the M.S. and Ph.D. degrees in electronic engineering from the Sapienza University of Rome, Italy, in 1999 and 2003, respectively. In 2010, he became a Researcher (an Assistant Professor) with the DIET Department, Sapienza University of Rome. In 2015, he was an Associate Professor with the DIET Department. He has coauthored more than 80 publications in international journals, about 80 contributions to conference proceedings, and is the co-inventor of two international patents. His research interests include integrated circuit design and focused on design methodologies able to guarantee robustness with respect to parameter variations in both analog circuits and digital VLSI circuits. In the context of cryptographic hardware, his focus in the past has been on novel PAAs methodologies and countermeasures, whereas recently he has been involved in research activities modeling with PUFs and TRNGs.

Open Access funding provided by 'Università degli Studi di Roma "La Sapienza" 2' within the CRUI CARE Agreement