Journal of Cryptographic Engineering (2021) 11:227–238
https://doi.org/10.1007/s13389-020-00244-5

REGULAR PAPER

A large-scale comprehensive evaluation of single-slice ring oscillator and PicoPUF bit cells on 28-nm Xilinx FPGAs

Chongyan Gu[1] · Chip-Hong Chang[2] · Weiqiang Liu[3] · Neil Hanley[1] · Jack Miskelly[1] · Máire O'Neill[1]

Received: 10 April 2020 / Accepted: 7 October 2020 / Published online: 24 December 2020

Abstract

Lightweight implementation of security primitives, e.g., physical unclonable functions (PUFs) and true random number generator, in field programmable gate array (FPGA) is crucial replacement of the conventional decryption key stored in battery-backed random access memory or E-Fuses for the protection of field reconfigurable assets. A slice is the smallest reconfigurable logic block in an Xilinx FPGA. The entropy exploitable from each slice of an FPGA is an important factor for the design of security primitives. Previous research has shown that the locations of slices can impact the quality of delay-based PUF designs implemented on FPGAs. To investigate the effect of the placement of each single-bit PUF cell free from the routing resource constraint between slices, single-bit ring oscillator (RO) and identity-based PUF design (Pi-coPUF) cells that can each be fully fitted into a single slice are evaluated. To accurately evaluate their statistical performance, data from a large number of devices are required. To this end, 217 Xilinx Artix-7 FPGAs has been employed to provide a large-scale comprehensive analysis for the two designs. This is the first time single-slice disorder-based security entities have been investigated and compared on 28-nm Xilinx FPGA. Uniqueness, uniformity, correlation, reliability, bit-aliasing and min-entropy of each type of cell are evaluated for four different types of cell placement. Our experimental results corroborate that the location of both cell types in the FPGA affects their performances. For both cell types, the lower the correlation between devices, the higher the min-entropy and uniqueness. Overall, the min-entropy, correlation and uniqueness of PicoPUF are slightly higher than those of RO. Otherwise, the uniformity, bit-aliasing and reliability of the PicoPUF are slightly lower than those of the RO. Comparing the resource usage and metrics of the PicoPUF, ring oscillator PUF and some existing memory-based PUF implementations, PicoPUF stands out as a lightweight FPGA-based weak PUF design. The raw data for the PicoPUF design are made publicly available to enable the research community to use them for benchmarking and/or validation.

Keywords FPGA · Entropy · Single slice · PUF

1 Introduction

Due to its reconfigurability and fast design turnaround time, FPGA has become an attractive target platform for developing hardware security primitives such as PUF and TRNG. A PUF is a security primitive that exploits the imperfect manufacturing process variations to generate a unique digital fingerprint for a monolithically integrated electronic device or system. Since the physical disorder properties introduced by process variations among different nanoscale devices on

✉ Chongyan Gu
cgu01@qub.ac.uk

✉ Weiqiang Liu
liuweiqiang@nuaa.edu.cn

Chip-Hong Chang
echchang@ntu.edu.sg

Neil Hanley
n.hanley@qub.ac.uk

Jack Miskelly
jmiskelly08@qub.ac.uk

Máire O'Neill
m.oneill@ecit.qub.ac.uk

[1] Centre for Secure Information Technologies (CSIT), Queen's University Belfast, Belfast, UK

[2] School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, Singapore

[3] College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, China

the same die and across dies are outside the control of the manufacturer, PUFs are inherently difficult to clone. Accordingly, a PUF circuit has a number of desirable features for security applications, such as the ability to provide low-cost unclonable identity of an (IC) or to return a device-specific response to an input challenge for chip authentication. These unique device intrinsic properties can be utilized in a number of different use cases, such as key generation, lightweight authentication protocols, anticounterfeiting and supply chain security. Some PUFs can also be used as TRNGs. A TRNG is another widely used hardware security primitive that makes use of noise and non-systematic variations in physical processes [1,2] to support security-critical tasks such as secret or public key generation, seeds for cryptographic primitives and nonces.

The major difference between (ASIC) and FPGA-based PUFs is that individual devices of an ASIC PUF are not manufactured until the design has been physically placed and routed, whereas the hardware resources of an FPGA PUF have already been manufactured prior to physical design. Consequently, the maximum and minimum entropies that can be extracted from an FPGA chip for PUF become more dependent on the size and locality of its bit cells even though every bit cell is identically designed. Specifically, the entropy of a logic slice, which is the minimum reconfigurable unit of an FPGA, is an essential factor that contributes to the quality of these security primitives. Investigating and evaluating the entropy contributed by each slice of an FPGA will therefore provide invaluable insight into single bit cell response of FPGA-based PUF and TRNG designs independent of the routing delay between slices. Unfortunately, the bit cell of most known PUFs that are suitable for FPGA implementation cannot be configured into a single slice of an FPGA, which introduces inaccuracy and inconsistency in evaluation due to the routing constraint between slices at different localities. In this paper, we consider two PUF designs, namely RO and identity-based PUF (referred to in this paper as PicoPUF) [3,4], whose repetitively used core elements can be implemented on a single FPGA slice. The oscillation frequency generated by the same smallest 3-stage ROs is different from slice to slice and from device to device. ROs are the fundamental component of glitter-based TRNGs which has been widely considered for FPGA implementation. The frequencies generated by two ROs are also usually compared to produce an output bit of a PUF. A number of different PUF structures based on ROs have been proposed in the literature, such as the original RO PUF design [5] and the SUM-PUF [6]. In contrast, PicoPUF generates a random bit based on the difference in timing between two delay paths on the same slice of an FPGA. Previous research [7] has shown that PUF metrics are affected by the number of devices used to evaluate the PUF designs. The larger the number of devices, the more accurate the inference about the evaluated metrics. Hence, a testbed is built to provide a large-scale analysis of the core bit cells of these two designs.

This paper is an invited extension of our preliminary work reported in [8] for this special issue. A comprehensive evaluation of two single-slice-based designs is investigated. More specifically, our research contributions are summarized as follows.

- The testbed comprising 217 FPGA devices with 8000 PicoPUF instances and 6592 RO instances is, to the best of the authors' knowledge, the largest reported to date.
- A comprehensive large-scale experimental analysis of uniqueness, reliability, uniformity, bit-aliasing, correlation and min-entropy for both RO and PicoPUF.
- The impact of floorplan on min-entropy of ROs and quality metrics of PicoPUF evaluated over a large-scale testbed.
- A detailed analysis and comparison of the two single-slice entropy sources for the design and application consideration of security primitives made of these components on FPGA platform.
- A comparison of both single-slice designs with other related works is presented.
- The raw data are made publicly available to the research community as a reference for further research into the design and implementation of security primitives on FPGA.

The rest of the paper is organized as follows: Sect. 2 provides a literature review of previous works on FPGAs. Two single-slice-based designs are utilized and introduced in Sect. 3. Section 4 presents the FPGA implementations of the two designs. A comprehensive experimental evaluation and comparison are provided in Sect. 5. Conclusions are given in Sect. 6. Section 7 provides the links to access the raw data.

2 Related works

A number of previous works have examined ROs on FPGA in the context of PUFs [5,9–11], as well as process variations [12,13]. However, there are only two existing large-scale RO PUF datasets on FPGAs. In [14], the testbed comprises 193 FPGA devices and 512 ROs, which is smaller in size than this work. Moreover, their data are generated from Xilinx Spartan-3 FPGAs, which are somewhat outdated. Recently, another dataset based on 100 Xilinx 28-nm Artix-7 FPGAs is provided by [15] for evaluating three oscillation-based PUFs, including RO PUF, transient effect RO PUF (TERO) and Loop PUF. The number of RO PUF instances per device, $16 \times 80 = 1280$, is less than this research work. To provide a large dataset, a strategy of instantiating multiple copies of the PUF design on each FPGA device was employed [16]. This is

Table 1 Uniqueness, min-entropy and CTW ratio reported for SRAM, FF and buskeeper PUFs

Type	Uniqueness	Min-entropy	CTW ratio	Sample size
SRAM-NXP [22]	0.49	0.75	99.1	20
SRAM-TSMC [22]	0.50	0.76	100	20
DFF [22]	0.50	0.77	100	20
Buskeeper [25]	0.50	0.82	99	194

acceptable for some types of analysis, but it is also restrictive for inter-device variation analysis. The UNIQUE project also investigated RO PUFs on a large number of devices (96) [17], but it targeted 65 nm ASICs instead of FPGAs. The impact of floorplan location for RO PUFs was also investigated [18] based on a small-scale testbed of FPGAs. A large-scale RO PUF analysis in terms of slice type, evaluation time and temperature on 28-nm Xilinx FPGAs was recently performed in [19]. In this paper, the same dataset has been employed to further evaluate the entropy of the RO design and investigate the relationship between different metrics. This is the first evaluation of a single-slice delay-based PUF design [3] on a large-scale testbed. The investigation leads to the first comparison of two known single-slice-based PUF cell designs.

PUF metrics are affected by the number of devices used to evaluate the PUF designs, as revealed in [7]. For greater accuracy, it is essential to evaluate a PUF design on large-scale testbeds. *Correlation* is a statistical relationship between two random variables. In the context of PUFs, it relates the likelihood of predicting the bit response of one device from the response of another device. Correlation reduces the effort for an adversary to predict the secret generated by or protected with the PUF. Several works have been published to evaluate the spatial correlation of PUF designs [20,21]. A number of other methods have also been proposed to assess the unpredictability of a PUF. In CTW, lossless compression algorithm is utilized to predict the *upper bound* of entropy (i.e., the best case) of a response [17,18,22–24]. Min-entropy is the most conservative estimate of the response unpredictability, and it represents the *lowest bound* of entropy (i.e. the worst case) of a set of PUF responses [16,22,24–26]. The actual entropy is expected to lie somewhere between this and the estimate described in the (NIST) specification 800-90 [27].

Table 1 presents some previously reported results on uniqueness and randomness of various PUF designs [28]. It is reasonable to believe that as the randomness of the PUF response increases, the inter-die HD between responses tends to get nearer to the ideal value of 0.5. However, the reverse may not be true for min-entropy. Although the uniqueness results are very close to or equal to 0.5, the min-entropy results are not as close to their ideal value of 1. The ideal CTW is 100%, implying no data correlation or redundancy can be exploited for compression. Except for the buskeeper PUF evaluated in [25], the results of CTW in Table 1 are only evaluated over the responses of a small number of physical devices.

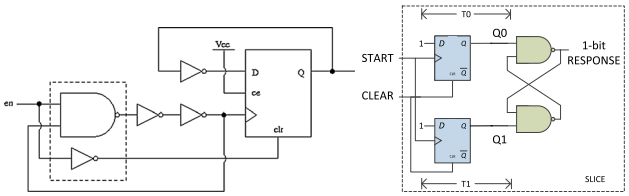


Fig. 1 Single-slice **a** RO cell and **b** PicoPUF cell, respectively

3 FPGA-based design entities under tests

3.1 RO design

The design utilized in the evaluation is a three-stage RO, as shown in Fig. 1a. An *enable* input activates or deactivates the oscillator, and the response is output by a toggle flip flop. It can be compactly fitted in a single Xilinx Artix-7 slice. The ROs are placed and routed consistently over all the FPGAs.

3.2 PicoPUF design

The PicoPUF design [3,4] is shown in Fig. 1b. It is based on a cross-coupled NAND construction, with the input signals provided by two D flip flops (DFFs). Each DFF has a synchronous enable and an asynchronous clear signal, and the D input is connected to 1. To evaluate the PicoPUF, the DFFs are first cleared to reset the outputs of Q0 and Q1 to 0 simultaneously. This turns both outputs of the crossed-couple NAND gates to 1. Then, the clear signal is disabled and the clock enable signal is set. This will create a race condition between the two NAND gates. The output of the faster NAND gate will switch from 1 to 0 while the output of the slower NAND gate will be kept at 1. The NAND gate transitions is determined by the random manufacturing process variations, which can be used to generate a single PUF bit of unpredictable response.

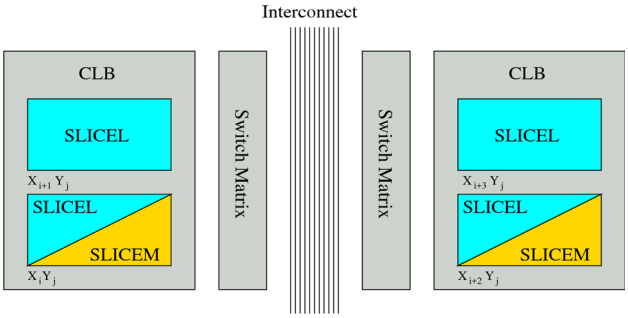


Fig. 2 Locations (left or right) of CLB relative to the switch matrix and positions (upper or lower) of single slice within the CLB for RO placement

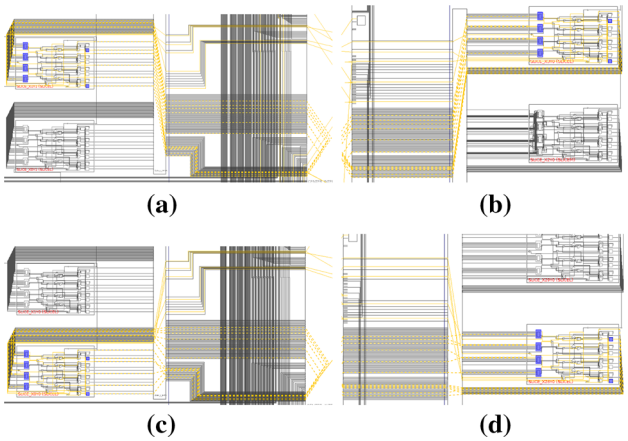


Fig. 3 Consistent routings for the four different placement types of PicoPUF: **a** LEFT-UPPER, **b** RIGHT-UPPER, **c** LEFT-LOWER and **d** RIGHT-LOWER

Table 2 Numbers of ROs and PicoPUFs cells for each type of CLB placement in an FPGA

Location	#ROs	#PicoPUFs
LEFT-UPPER	1600	1952
LEFT-LOWER	1600	1952
RIGHT-UPPER	1696	2048
RIGHT-LOWER	1696	2048
Total	6592	8000

4 FPGA implementation

There are two types of slices on an Xilinx Artix-7 FPGA: SLICEL and SLICEM. All logic components required for a single cell RO implementation are available on both types of slice. Therefore, there is no restriction on the placement of the single cell RO on the FPGA.

As shown in Fig. 2, the slices are paired up in the configurable logic blocks (CLBs) of the Artix-7 FPGA. The RO type is identified as either *upper* or *lower* according to its position within a CLB, and as *left* or *right* according to the location of its CLB routing channel to the switch box. The numbers of each of the four possible placements of RO and PicoPUF cells implemented on one Artix-7 chip are listed in Table 2. The total numbers of ROs and PicoPUF implemented on one chip are 6592 and 8000, respectively.

The PicoPUF implementations for the four different locations of CLB placement of a Xilinx Artix-7 FPGA are shown in Fig. 3. The routing for any path that contributes directly to the race condition was fixed using the hook script in Vivado design flow. The detailed implementations of both the RO and PicoPUF designs can be found in [3,19], respectively.

5 Experimental results

5.1 Experimental setup

The experimental platform consists of four modules in total, each of which holds 60 Basys-3 boards, 10 7-port USB hubs, a Raspberry PI-2 and power supply. The USB connection between the PI-2 and Basys3 boards powers the FPGA as well as provides a JTAG interface to configure the bitstream of the design into the FPGA. A UART interface is used to communicate with the configured design and receive the measurement results. The Raspberry-Pi communicates over a (LAN) with a global experiment control server, which also stores the measured data.

The RO frequency was measured indirectly by counting the positive edges of the toggle flip-flop as shown in Fig. 1a during an evaluation time D, with different evaluation times ranging from 0.50 μs to 10.00 ms.

5.2 Overall metrics

A number of metrics have been suggested [29] for the evaluation of different PUF designs. Table 3 shows the experimental results on uniqueness, reliability, uniformity, bit-aliasing, correlation and min-entropy of RO and PicoPUF designs in four different types of CLB placement. Details of the analysis and a comparison are provided in the following subsections.

5.3 Uniqueness

Uniqueness represents the ability to distinguish between different devices based on its response to the same challenge. As the instantiations are identical, the difference between the responses is based completely on the process variations. In order to use the designs as an intrinsic identifier, no two

Table 3 Experimental results of PicoPUF and RO based on slice locations

	LEFT-LOWER		LEFT-UPPER		RIGHT-LOWER		RIGHT-UPPER		ALL		Ideal
PUF type	PicoPUF	RO	PicoPUF	RO	PicoPUF	RO	PicoPUF	RO	PicoPUF	RO	–
# Bits (n)	1952	1600	1952	1600	2048	1696	2048	1696	8000	6592	–
Uniqueness											
μ_{inter}	0.4796	0.4717	*0.4968*	0.4895	0.4816	**0.4714**	0.4962	0.4895	0.4886	0.4805	0.50
σ_{inter}	0.0158	0.0174	*0.0124*	**0.0178**	0.0151	0.0169	*0.0124*	0.0173	0.0094	0.0087	0.00
Uniformity											
0_{frac}	**0.4103**	0.5127	0.4790	0.5045	0.4180	0.5172	0.4726	*0.5019*	0.4450	0.5091	0.50
1_{frac}	**0.5897**	0.4873	0.5210	0.4955	0.5820	0.4828	0.5274	*0.4981*	0.5550	0.4909	0.50
Reliability											
0_{stable}	*48.09%*	46.70%	**41.37%**	47.48%	47.26%	46.22%	42.75%	47.75%	44.87%	47.75%	50%
1_{stable}	**31.65%**	49.22%	37.61%	48.39%	32.16%	*49.69%*	37.70%	48.11%	34.78%	48.11%	50%
μ_{intra}	0.0229	0.0068	**0.0243**	0.0068	0.0237	*0.0067*	0.0225	0.0069	0.0233	0.0069	0.00
σ_{intra}	**0.0037**	0.0030	0.0036	0.0030	0.0036	*0.0029*	**0.0037**	*0.0029*	0.0020	0.0029	0.00
Bit-Aliasing											
μ_{bit}	**0.4103**	0.5127	0.4790	0.5045	0.4180	0.5172	0.4726	*0.5019*	0.4450	0.5091	0.50
σ_{bit}	0.0593	**0.1229**	0.0505	0.0797	0.0616	0.1228	*0.0501*	0.0798	0.0636	0.1038	0.00
Correlation											
μ_{bit}	0.0146	**0.0605**	0.0103	0.0254	0.0157	0.0604	*0.0102*	0.0255	0.0165	0.0431	0.00
σ_{bit}	0.0770	0.0728	**0.0774**	0.0751	0.0769	*0.0722*	*0.0772*	0.0746	0.0744	0.0674	0.00
Min-entropy											
$\mu_{H_{\text{min}}}$	**0.7585**	0.7734	*0.8826*	0.7945	0.7706	0.7701	0.8781	0.7928	0.8225	0.7825	1.00
$\sigma_{H_{\text{min}}}$	**0.1278**	*0.0768*	0.0861	0.0813	0.1261	0.0786	0.0897	0.0794	0.1236	0.0790	0.00

Italic represents the best result in a row, bold represents the worst result in a row, μ is the mean value, σ is the standard deviation value. Note, all the values for RO PUF are the best results over all the evaluations

devices should have the same response, and the responses learnt from a (large) number of devices should not allow an adversary to infer any information about the response from a different device. Uniqueness can be measured by the average fractional HD between the responses generated from different pairs of devices. A fractional HD of 0 indicates all bits between two strings are different, and 1 means that all the bits are identical. Ideally, the expected fractional HD between any pair of responses is 0.5. The uniqueness experiment is carried out on 217 FPGA devices. A total of 217 responses are generated. Each response has 6592 bits generated from 6592 independent single-slice bit cells of a device.

In Table 3, the PicoPUF at the *LEFT-UPPER* location has the best uniqueness of 0.4968 with a small standard deviation (STD) of 0.0124. Therefore, PicoPUF is best implemented on the *LEFT-UPPER* location of Xilinx Artix-7 for uniqueness. Although the uniqueness of the RO PUF is not as good, it still has reasonably good uniqueness of 0.4895 when it is placed at the *RIGHT-UPPER* location. Additionally, the *RIGHT-LOWER* location of Xilinx Artix-7 should be avoided for the RO PUF implemented with the single-slice ROs due to its worst uniqueness at this location.

The histogram of the fractional HD for the RO PUF responses over 217 devices is shown in Fig. 4. The mean and (STD) of the distribution are 0.4805 and 0.0087, respectively. As shown in Fig. 5, the uniqueness of the PicoPUF obtained from the mean of the fractional HD distribution is ≈ 0.4886, which is closer to the ideal value of 0.5 than that of the RO. The STD of its distribution is 0.0094. It is interesting to note from Figs. 4 and 5 that the distribution of the fractional Hamming distances of the RO PUF is closer to Gaussian than that of the PicoPUF. This is probably attributed to the distribution of the delay deviation of wiring among slices more uniform than the distribution of delay deviations of different active elements (e.g., DFFs and NAND gates)

5.4 Correlation

Table 3 shows the spatial correlation scores computed based on the method in [20,21]. The best correlation result (0.0102) is from the PicoPUF in the *RIGHT-UPPER* location, and the worst (0.0605) is from the RO in the *LEFT-LOWER* location. The PicoPUF has a lower correlation between devices than the RO. Based on the uniqueness and correlation results, it is recommended to place the PicoPUF and single-slice RO at the **-UPPER* locations instead of the **-LOWER* locations.

Figure 6a shows the correlation results of the RO frequencies at the four different RO locations for 15 different

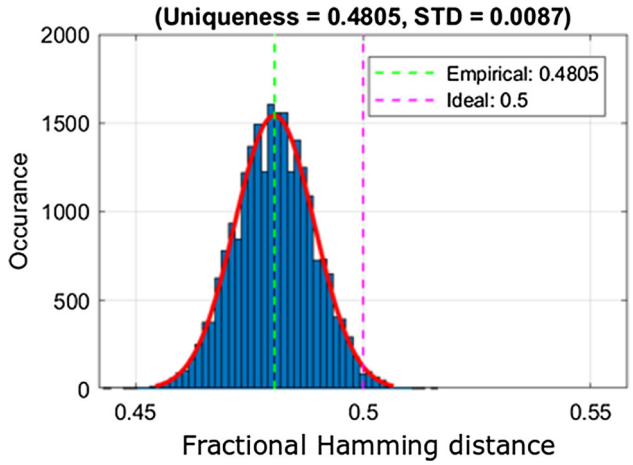


Fig. 4 Distribution of fractional HDs of ROs

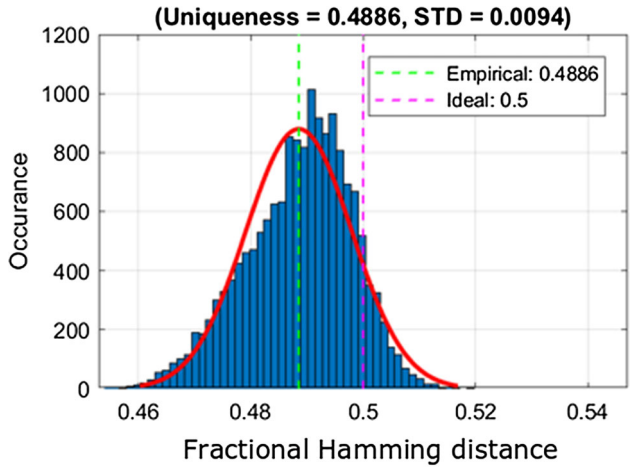
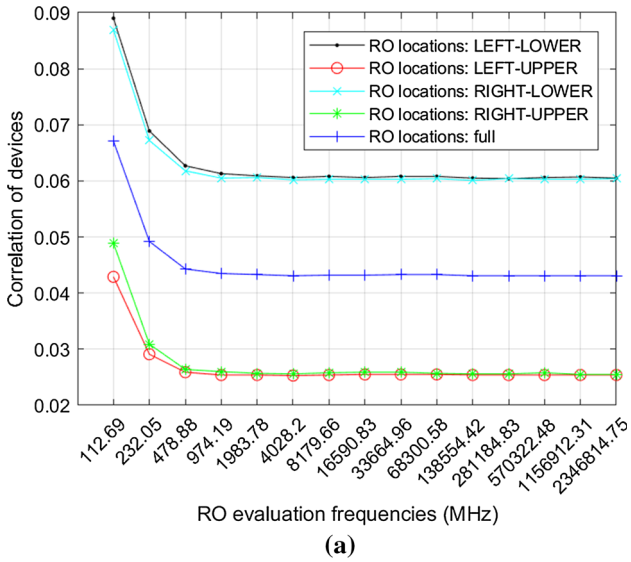


Fig. 5 Distribution of fractional HDs of PicoPUFs

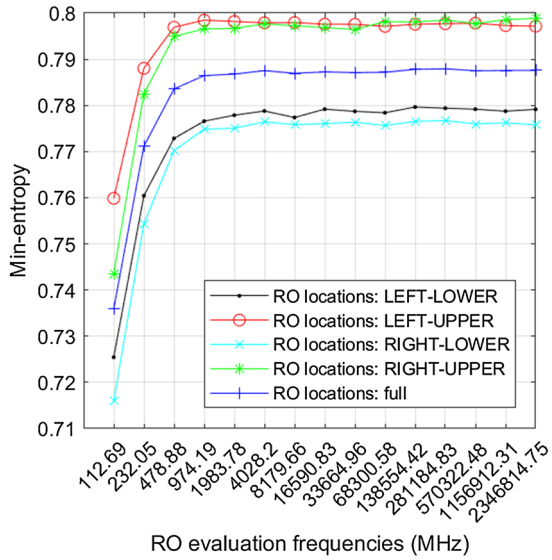
evaluation frequencies. The evaluation frequency is the reciprocal of the evaluation time. The longer the evaluation time for the ROs, the lower the correlation between the devices. The RO placements in the *LEFT-UPPER* and *RIGHT-UPPER* locations have lower correlation than those in the *LEFT-LOWER* and *RIGHT-LOWER* locations.

5.5 Min-entropy

Min-entropy is commonly used as a worst-case analysis for describing the unpredictability and randomness of the outcome of a non-uniform distribution of secret [7,22]. The occurrence probability of 1 and 0 in the n-bit responses generated from m devices is denoted by $p1$ and $p0$, respectively. $p1$ can be calculated by the fractional HD of each bit b of m devices, $\frac{\mathrm{HW}_b}{m}$, and $p0$ by $1-\frac{\mathrm{HW}_b}{m}$. The maximum probability, $p_{b\,\max} = \max(p0; p1)$, is used to estimate the min-entropy per bit in 6 ("Appendix A").



(a)



(b)

Fig. 6 **a** Spatial correlation of RO frequencies; and **b** min-entropy of RO frequencies with varying evaluation time

Table 3 presents the min-entropies of the RO and PicoPUF at four different placement locations. The best and worst min-entropy results of both designs are observed at the *LEFT-UPPER* and *LEFT-LOWER* locations, respectively. In particular, the PicoPUF at the *LEFT-LOWER* location has the worst STD of 0.1278. The results match well with the relative uniqueness at different locations, which confirm the correlation between uniqueness and min-entropy. Figure 7 shows both the average min-entropy for different numbers of devices and the bit entropy distribution over all locations for the PicoPUF. Its average min-entropy of 0.8225 is higher than that of the RO, which is 0.7825. Previous research [7] indicated that with inadequate sampling, a small number of

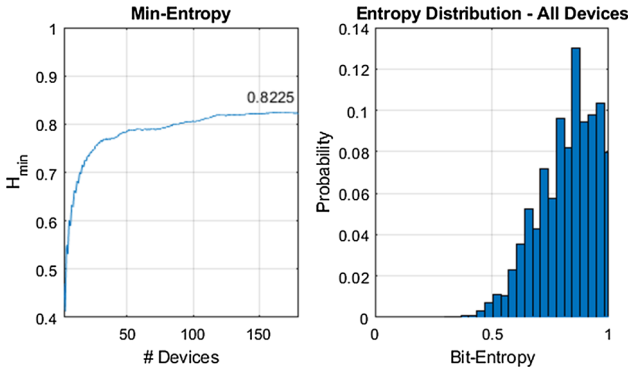


Fig. 7 Min-entropy of PicoPUF responses

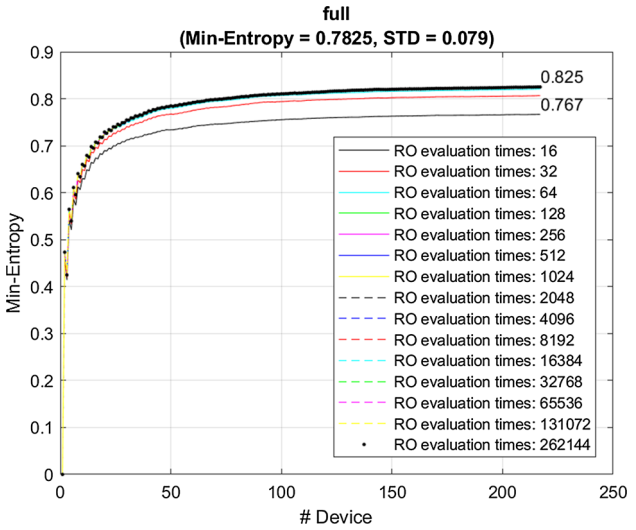


Fig. 8 Min-entropy of RO frequencies for 15 different RO evaluations and different numbers of devices

outliers can bias the evaluation of PUF quality metrics. The result in Fig. 7 confirms this observation. The min-entropy converges only with measurements taken from more than 150 devices.

5.5.1 Effect of locations and evaluation time for the ROs

Figure 8 presents the min-entropy results of the RO frequencies at different RO locations for 15 RO evaluations. The longer the RO evaluation time, the greater the min-entropy, but the increase is insignificant when the RO evaluation time is larger than 974.19 MHz. The ROs at the *LEFT-UPPER* and *RIGHT-UPPER* locations have higher min-entropy than at other locations. The results again suggest that the **-UPPER* locations are the best locations for the placement of these two types of PUF cell.

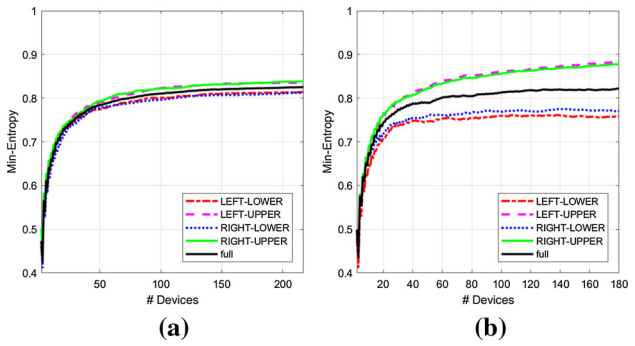


Fig. 9 Min-entropy of **a** the PicoPUF and **b** RO frequencies, respectively, for different numbers of devices

5.5.2 Effect of the number of devices

Figure 9 shows the min-entropy results of RO and PicoPUF at the four different locations and over different numbers of devices. It indicates that the larger the number of devices, the higher the min-entropy. It can be seen that approximately 140 devices ($m \geq 140$) are required in order to minimize the estimation error of the average min-entropy of the design. The min-entropy ranges from 0.7585 to 0.8826 for the PicoPUF as shown in Fig. 9b and from 0.7701 to 0.7945 for the RO as shown in Fig. 9a. Hence, PicoPUF has a broader spread of min-entropy than RO over different placement locations.

5.6 Reliability

It is important to be able to repeat the response of each bit cell of the PUF under test at all times. The greater the reliability of the raw response, the less costly the error correction. Intra-HD is a popular metric for investigating the reliability of a PUF response. It measures the fractional HD between a reference response and the measured response.

To test the reliability, $r = 10,001$ and $r = 1000$ repeated measurements were taken for every response bit of each PicoPUF and RO cell, respectively, on each FPGA. The results in the rows pertaining to reliability of Table 3 are obtained from $m \times n = 180 \times 8000 = 1.44M$ response bits of PicoPUF and $m \times n = 217 \times 6592 = 1.43M$ response bits of RO. For PicoPUF, a significant portion of the response bits are reliable of which 44.87% (or 646, 128 of 1440, 000) are stable 0's and 34.78% (or 500, 832 of 1440, 000) are stable 1's for each of the r acquisitions. For RO, 47.75% of the stable response bits (or 687, 600 of 1440, 000) are 0's and 48.11% of the reproducible bits (or 692, 782 of 1440, 000) are 1's for each of the r acquisitions. Hence, PicoPUF has approximately 10% difference between the number of stable 0's and 1's. RO bit cells are more reliable than PicoPUF bit cells, with a smaller difference of approximately 1-3% between the stable 0's and 1's.

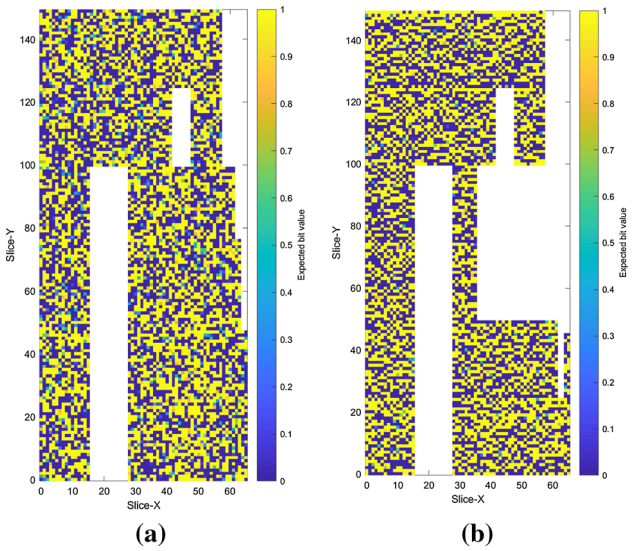


Fig. 10 Reliability heatmaps of **a** PicoPUF responses and **b** RO frequencies, respectively

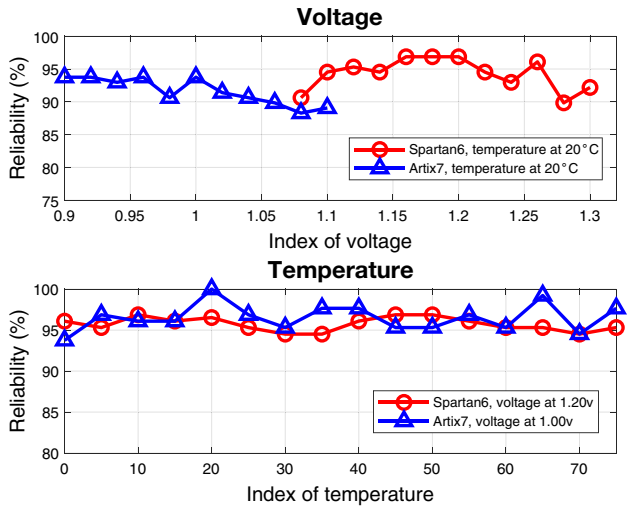


Fig. 11 Reliability results of two FPGA device implementations of PicoPUF over voltage and temperature variations

The heatmap in Fig. 10a shows the mapping of the reliability of each PicoPUF bit from a randomly selected device of the testbed to the corresponding location in the FPGA floorplan. Each box presents the probability of occurrence of 1 of each response bit in r repeated measurements. It can be seen that the '1' or '0' bits are evenly and randomly distributed. A small number of bits are unreliable. They are also randomly distributed. The heatmap in Fig. 10b presents similar results for the RO. The missing cells in the middle right of the image are due to the slices utilized for Miroblaze, with the remaining blank spaces not containing slices due to block RAM (BRAM) or digital signal processing (DSP) blocks. The reliabilities of both PicoPUF and RO show no significant dependence on the surrounding paths.

We also evaluated the response reliability against temperature and supply voltage variations for the PicoPUF implemented on 65-nm technology Xilinx Spartan-6 and 28-nm technology Artix-7 FPGAs. The results are plotted in Fig. 11. In this evaluation, the core supply voltage is varied by $\pm$ 10% from its nominal voltage. The nominal core voltages of Xilinx Spartan-6 and Artix-7 FPGAs are 1.2 V and 1.0 V, respectively. The average reliability of the PicoPUF against voltage variation is 94.27% on Xilinx Spartan-6 and 91.62% on Artix-7. The specified operating temperature range of both FPGA boards is 0–75°C. The average reliability over this working temperature range of the PicoPUF is 95.73% on Spartan-6 and 96.53% on Artix-7. The results show that the PicoPUF responses are more sensitive to voltage than temperature variation, particularly for advanced technology node with lower nominal supply voltage and reduced on–off current ratio.

5.7 Uniformity

The uniformity metric depicts how the response from each device is split between [0,1]. It is the expected 'weight' or 'bias' of a response bit for a randomly chosen device calculated by taking the average of all the response bits. An unbiased bit has a uniformity of 0.5. The results in Table 3 show that the best uniformity of 0.5019 is generated by the RO cells at the *RIGHT-UPPER* locations, and the worst uniformity of 0.4103 is generated by the PicoPUF cells at the *LEFT-LOWER* locations. Additionally, RO has better overall uniformity than PicoPUF. Similar to the results of uniqueness and correlation, the uniformity at the **-UPPER* locations for both the PicoPUF and single-slice RO is better than those at the **-LOWER* locations. Hence, it is recommended to avoid the placement of these cells at the **-LOWER* locations on Xilinx Artix-7 FPGA.

5.8 Bit-aliasing

Bit-aliasing investigates each of the response bits individually. This can be done by simply averaging the response bits generated by all cells at the same location across the number of available devices. To ensure that no physical locations of the FPGA are strongly biased towards [0, 1], the expected bit response of each physical location of the target FPGA should be 0.5 for a well-balanced design.

Heatmaps of the bit-aliasing results for PicoPUF and RO are shown in Fig. 12a, b, respectively. In general, although no single-slice location returns the same value across different devices, a small number of cells are significantly biased. Skews toward either 1 or 0 are observed in the area adjacent to the clock distribution network for the clock tile as shown in Fig. 12b for the RO. As shown in Table 3, the best bit-aliasing result (0.5019) is from the RO at the *RIGHT-UPPER*

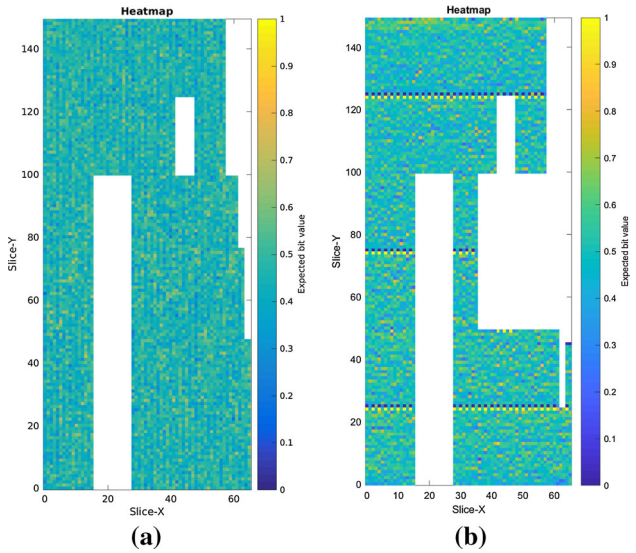


Fig. 12 Heatmaps of the bit-aliasing results for **a** the RO and **b** the PicoPUF responses, respectively

location and the worst (0.4103) is from the PicoPUF at the *LEFT-LOWER* location. It is therefore suggested to place the single-slice RO at the **-UPPER* locations and keep a distance from the clock distribution network if feasible.

5.9 Comparison and discussion

5.9.1 PicoPUF and RO

The PicoPUF implemented at the *LEFT-UPPER* location has the best uniqueness and min-entropy, but it gives the worst uniformity, reliability, bit-aliasing and min-entropy when implemented at the *LEFT-LOWER* location. Hence, the *RIGHT-UPPER* location is the best placement choice for implementing PicoPUF on FPGA. Interestingly, the RO achieves the best reliability when it is implemented at the *RIGHT-LOWER* location, but it achieves lower reliability and higher uniqueness when it is implemented at both the *RIGHT-UPPER* and *LEFT-UPPER* locations. Therefore, there is an inevitable trade-off between uniqueness and reliability depending on the placement of the RO on FPGA. Considering the fact that RO-based designs usually require counters for digitalizing the RO frequencies, PicoPUF is a more lightweight choice than the RO for a design that requires higher uniqueness and less hardware resources on FPGA. The RO presented in Fig. 1a and the PicoPUF shown in Fig. 1b have very different response bit generation mechanisms. This has led to the differences in their sensitivity to inter- and intra-slice wire delay variations. Unlike conventional RO PUF, the single-slice RO has only three inverter stages, which causes its RO frequency to be more susceptible to wire delay variations within and among slices. Thus, the intra- and inter-slice wire delays of the single-slice RO play a significant role in contributing to the frequency deviation and spatial correlation of RO frequencies, respectively, in our RO PUF construction. The latter can be averaged out by a longer evaluation time, as indicated in Fig. 6b. On the other hand, the response bit of PicoPUF is generated based on the race condition of cross-coupled NAND gates and the simultaneous switching of the two DFFs. The response is predominantly influenced by the intra- and inter-slice delay differences of the active elements instead of the wire delay. For this reason, the PicoPUF can achieve better min-entropy and uniqueness results by route balancing than the RO PUF constructed from single-slice ROs.

5.9.2 Other weak PUFs

A comparison of the resource usage and metrics of the PicoPUF, RO PUF and the previous work on PUF implementations is shown in Table 4. The SRAM PUF cell, proposed by Guajardo et al. [31], only generates a response upon resetting the memory array. The Latch PUF proposed by Su et al. [32] dissipates low power, but the results are only reported on ASIC implementation. The Flip-flop PUF proposed by Roel et al. [34] is similar to SRAM PUF in that it uses the power-up reset of flip-flops. However, its has limited entropy and requires post-processing to boost the randomness. The Butterfly PUF proposed by Kumar et al. [35] is suitable for FPGA implementation since it can be implemented using basic logic gates. It is reported to have 94% reliability over temperature variations, but its reliability over voltage changes is not evaluated. It consumes 130 slices of a Virtex-5 FPGA device for a 64-bit response. The RO PUF proposed by Suh et al. [5] has been implemented on different FPGAs, e.g. Virtex-4 and Spartan-3. The hardware resource consumption is at least 384 slices for a 64-bit response. To avoid the interdependent response bits of Suh's RO PUF [5], two independent single-slice ROs of Fig. 1a are used to generate one response bit. For a 128-bit response, it requires $2 \times 128 = 256$ slices. Additionally, counters and comparator are also required to compare the number of positive edges of the toggle flip-flops between two ROs. The length of each counter depends on the evaluation time, which has an impact on the min-entropy of RO frequencies, as evaluated in Fig. 6b. PicoPUF design [3] is a lightweight FPGA-based Weak PUF design compared to these Weak PUF designs. In [30], the reliability of the PicoPUF design [3] has been enhanced to almost 100% by a post-characterization process at the expense of a slight degradation of uniqueness. This post-characterized version of PicoPUF is denoted by PicoPUF* in Table 4.

Table 4 Comparison of hardware resource consumption and metrics of different PUF designs [30]

PUF design	Type	Uniqueness	Reliability	Hardware	Response (bit)	Resource consumption
SRAM PUF [31]	Weak	49.97%	$> 88\%^{t}$	FPGA	128	4600 SRAM memory bits
Latch PUF [32]	Weak	50.55%	96.96%	0.13μm CMOS	128	1 latch for each ID cell
Latch PUF [33]	Weak	46%	$> 87\%^{t}$	Spartan 3	128	2×128 slices
Flip-flop PUF [34]	Weak	$\approx 50\%$*	$> 95\%$*	Virtex 2	4096	4096 flip flops
Flip-flop PUF [24]	Weak	36%	$> 87\%^{t}$	ASIC	1024	1024 flip flops
Buskeeper PUF [25]	Weak	49%	$> 80\%^{t}, > 95\%^{v}$	TSMC 65-nm	192	$1\ GE$[1]
Butterfly PUF [35]	Weak	$\approx 50\%$	94%	Virtex 5	64	130 slices
RO PUF [5]	Weak	46.15%	99.52%	Virtex 4	128	$16 \times 64\ array$[2]
PicoPUF [3]	Weak	48.52%	93.00%	Spartan-6	128	128 slices
PicoPUF [30]	Weak	49.90%	94.53%	Artix-7	128	128 slices
PicoPUF* [30]	Weak	45.60%*	98.74%*	Artix-7	128	128 slices
RO PUF (this work)	Weak	48.05%	$99.30\%^{t}$	Artix-7	128	> 256 slices[a]

[1] GE represented gate equivalent. [3] COMB = 2NOR + 1MUX + 1DEMUX
[2] 16×64 array = 1024ROs, each RO consisting of 5 inverters and 1 AND.
^{t} is the under temperature variation. ^{v} is the under supply voltage variation. * required post-processing.
[a] required extra circuits, e.g., counter and comparator

6 Conclusion

In this work, we presented a large-scale analysis of two single-slice-based bit cells, RO and PicoPUF, for PUF implementation on 217 Xilinx Artix-7 XC7A35T FPGAs. The entire fabric was covered by either 8000 distinct PicoPUF cells or 6592 RO instances. The uniqueness, uniformity, correlation, reliability, bit-aliasing and min-entropy for these two designs in four different types of placement are rigorously evaluated for the first time. The experimental results show that the overall min-entropy, correlation and uniqueness of the PicoPUF are slightly higher than those of the RO, while the other metrics, including uniformity, bit-aliasing and reliability, are slightly lower. Moreover, the experimental results show that the lower the correlation between devices, the higher the min-entropy and uniqueness for both design implementations on FPGA. Finally, it is shown that the physical placement location of the cell for RO has a greater influence than for PicoPUF, specifically in the area adjacent to the clock distribution network. A PicoPUF can independently generate a 1-bit response per slice, whereas the RO-based PUF requires at least two ROs and extra post-processing, e.g., counter, to generate one response bit. From this perspective, PicoPUF is more efficient than RO PUF.

7 Raw data

The raw PicoPUF and RO frequency data can be publicly accessible at https://pure.qub.ac.uk/portal/en/datasets/picopuf-dataset(522efef3-eeac-4523-9be7-2c3f296d61ef).htmlQUB-CSIT-Raw-Picopuf-Data and https://s3.eu-central-1.amazonaws.com/aisecresearchdata/2018fpga-ro-data/index.htmlEU-FP7-SPARKS-RO-DATA, respectively.

Acknowledgements This work is supported by grants from the Engineering and Physical Sciences Research Council (EPSRC) (EP/N508664/-CSIT2), the Singapore Ministry of Education AcRF Tier 1 Grant No. 2018-T1-001-131, the National Natural Science Foundation of China (62022041 and 61871216), the Fundamental Research Funds for the Central Universities China (NE2019102) and the Six Talent Peaks Project in Jiangsu Province (2018XYDXX-009).

A formulae for computation of various metrics

$$\textbf{Uniqueness} = \frac{2}{m(m-1)} \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} \frac{HD(R_i, R_j)}{n} \times 100 \quad (1)$$

where m and n are number of devices and length of response, respectively.

$$\textbf{Reliability} = 100 - HD_{intra} \tag{2}$$

$$= 100 - \frac{1}{s}\sum_{t=1}^{s}\frac{HD(R_i, R_{i,t}^{'})}{n} \times 100 \tag{3}$$

where $R_{i,t}^{'}$ is the tth sample of $R_i^{'}$.

$$\textbf{Uniformity} = (HW)_l = \frac{1}{n}\sum_{l=1}^{n}(r_{i,l} = 1) \times 100 \tag{4}$$

where $r_{i,l}$ is the response bit at the lth cell location in the ith chip.

$$\textbf{Bit-aliasing} = (HW)_p = \frac{1}{m}\sum_{i=1}^{m}(r_{i,p} = 1) \times 100 \tag{5}$$

where $r_{i,p}$ is the response bit at the pth cell location in the ith chip.

$$\textbf{Min-entropy} = \mathsf{H}_{\mathsf{min},b} = -\log_2(p_{b\,\mathsf{max}}) \tag{6}$$

where

$$p_{b\ \mathsf{max}} = \begin{cases} \frac{\mathsf{HW}_b}{m} & \mathsf{HW}_b > \frac{m}{2} \\ 1 - \frac{\mathsf{HW}_b}{m} & \mathsf{HW}_b \leq \frac{m}{2} \end{cases} \tag{7}$$

$$\textbf{Correlation} = \rho\left(R_i, R_j\right) = \frac{\mathsf{cov}(R_i, R_j)}{\sigma_{R_i}, \sigma_{R_j}} \tag{8}$$

References

1. Tsoi, K.H., Leung, K., Leong, P.H.W.: Compact FPGA-based true and pseudo random number generators. In: Proceedings of 11th IEEE Annual Symposium on Field-Programmable Custom Computing Machines (FCCM), 2003, pp. 51–61
2. Fischer, V., Bernard, F., Bochard, N., Varchola, M.: Enhancing security of ring oscillator-based TRNG implemented in FPGA. In: Proceedings of IEEE International Conference on Field Programmable Logic and Applications, 2008, pp. 245–250
3. Gu, C., Murphy, J., O'Neill, M.: A unique and robust single slice FPGA identification generator. In: Proceedings of International Symposium on Circuits and System (ISCAS'14), pp. 1223–1226. IEEE, Melbourne, Australia (2014)
4. Gu, C., O'Neill, M.: Ultra-compact and robust FPGA-based PUF identification generator. In: Proceedings of IEEE International Symposium on Circuits and Systems. (ISCAS'15), Lisbon, Portugal, May 2015, pp. 934–937
5. Suh, G.E., Devadas, S.: Physical unclonable functions for device authentication and secret key generation. In: Proceedings of 44th ACM Design Automation Conference (DAC), pp. 9–14 (2007)
6. Yu, M.-D.M., Devadas, S.: Recombination of physical unclonable functions. In: Proceedings of 35th Annual GOMACTech Conference, USA March (2010)
7. Gu, C., Liu, W., Hanley, N., Hesselbarth, R., O'Neill, M.: A theoretical model to link uniqueness and min-entropy for PUF evaluations. IEEE Trans. Comput. **68**(2), 287–293 (2019)
8. Gu, C., Chang, C.H., Liu, W., Hanley, N., Miskelly, J., O'Neill, M.: A large scale comprehensive evaluation of single-slice ring oscillator and PicoPUF bit cells on 28nm Xilinx FPGAs. In: Proceedings of 3rd ACM Workshop on Attacks and Solutions in Hardware Security Workshop (ASHES'19), London, UK, pp. 101–106 (2019)
9. Maiti, A., Schaumont, P.: Improved ring oscillator PUF: an FPGA-friendly secure primitive. J. Cryptol. **24**(2), 375–397 (2011)
10. Merli, D., Stumpf, F., Eckert, C.: Improving the quality of ring oscillator PUFs on FPGAs. In: Proceedings of 5th ACM Workshop on Embedded System Security, p. 9 (2010)
11. Kodỳtek, F., Lórencz, R.: A design of ring oscillator based PUF on FPGA. In Proceedings of IEEE 18th International Symposium on Design and Diagnostics of Electronic Circuits and Systems (DDECS'15), pp. 37–42 (2015)
12. Onodera, H.: Variability: modeling and its impact on design. IEICE Trans. Electron. **89**(3), 342–348 (2006)
13. Pang, L.-T., Nikolic, B.: Measurements and analysis of process variability in 90 nm CMOS. IEEE J. Solid State Circuits **44**(5), 1655–1663 (2009)
14. Maiti, A., Casarona, J., McHale, L., Schaumont, P.: A large scale characterization of RO-PUF. In: Proceedings of IEEE International Symposium on Hardware Oriented Security and Trust (HOST'10), pp. 94–99 (2010)
15. Wild, A., Becker, G.T., Guneysu, T.: A fair and comprehensive large-scale analysis of oscillation-based PUFs for FPGAs. In: Proceedings of 27th International Conference on Field Programmable Logic and Applications (FPL), September 2017, pp. 1–7
16. Che, W., Kajuluri, V.K., Martin, M., Saqib, F., Plusquellic, J.: Analysis of entropy in a hardware embedded delay PUF. Cryptography **1**(1), 8 (2017)
17. Katzenbeisser, S., Kocabaş, Ü., Rožić, V., Sadeghi, A.-R., Verbauwhede, I., Wachsmann, C.: PUFs: Myth, fact or busted? a security evaluation of physically unclonable functions (PUFs) cast in silicon. In :Cryptographic Hardware and Embedded Systems (CHES'12), pp. 283–301. Berlin, Heidelberg (2012)
18. Liu, W., Yu, Y., Wang, C., Cui, Y., O'Neill, M.: RO PUF design in FPGAs with new comparison strategies. In: Proceedings of International Symposium on Circuits and Systems (ISCAS'15), May 2015, pp. 77–80
19. Hesselbarth, R., Wilde, F., Gu, C., Hanley, N.: Large scale RO PUF analysis over slice type, evaluation time and temperature on 28 nm Xilinx FPGAs. In: Proceedings of IEEE International Symposium on Hardware Oriented Security and Trust (HOST'18), April 2018, pp. 126–133
20. Willsch, B., Hauser, J., Dreiner, S., Goehlich, A., Vogt, H.: Statistical tests to determine spatial correlations in the response behavior of PUF. In: Proceedings of 12th Conference Ph.D. Research in Microelectronics and Electronics (PRIME'16), June 2016, pp. 1–4
21. Wilde, F., Gammel, B.M., Pehl, M.: Spatial correlation analysis on physical unclonable functions. IEEE Trans. Inf. Forensics Secur. **13**(6), 1468–1480 (2018)
22. Claes, M., van der Leest, V., Braeken, A.: Comparison of SRAM and FF PUF in 65 nm technology. In: Proceedings of Nordic Conference on Secure IT Systems, pp. 47–64. Springer (2011)
23. Ignatenko, T., Schrijen, G.J., Skoric, B., Tuyls, P., Willems, F.: Estimating the secrecy-rate of physical unclonable functions with the context-tree weighting method. In: Proceedings of Nordic Conference on Secure IT Systems, July 2006, pp. 499–503
24. van der Leest, V., Schrijen, G.-J., Handschuh, H., Tuyls, P.: Hardware intrinsic security from D flip–flops. In: Proceedings the 5th ACM Workshop on Scalable Trusted Computing (STC'10), Chicago, IL, USA, 2010, pp. 53–62
25. Simons, P., van der Sluis, E., van der Leest, V.:"Buskeeper PUFs, a promising alternative to D flip-flop PUFs. In: Proceedigs of International Symposium on Hardware Oriented Security and Trust (HOST'12), June 2012, pp. 7–12
26. Gu, C., Hanley, N., O'Neill, M.:"FPGA-based strong PUF with increased uniqueness and entropy properties. In: Proceedings of

International Symposium on Circuits and Systems (ISCAS'17), May 2017, pp. 1–4
27. Barker, E.B., Kelsey, J.M.: Recommendation for random number generation using deterministic random bit generators (revised). US Department of Commerce, Technology Administration, National Institute of Standards and Technology, Computer Security Division, Information Technology Laboratory (2007)
28. van den Berg, R., et al.: Entropy analysis of physical unclonable functions. MSc. thesis, Eindhoven Univ. Technol., Eindhoven (2012)
29. Maiti, A., Gunreddy, V., Schaumont, P.: A systematic method to evaluate and compare the performance of physical unclonable functions. Cryptology ePrint Archive, Report 2011/657 (2011)
30. Gu, C., Hanley, N., O'neill, M.: Improved reliability of FPGA-based PUF identification generator design. In: ACM Transactions on Reconfigurable Technology and Systems (TRETS), vol. 10, no. 3, pp. 1–23 (2017)
31. Guajardo, J., Kumar, S. S., Schrijen, G.-J., Tuyls, P.: FPGA intrinsic PUFs and their use for IP protection. In: Cryptographic Hardware and Embedded Systems (CHES'07), pp. 63–80. Berlin, Heidelberg (2007)
32. Ying, S., Jeremy, H., Brian, O.: A digital 1.6 pj/bit chip identification circuit using process variations. IEEE J. Solid State Circuits **43**, 69–77 (2008)
33. Yamamoto, D., Sakiyama, K., Iwamoto,M., Ohta, K., Ochiai, T., Takenaka, M., Itoh, K.: Uniqueness enhancement of PUF responses based on the locations of random outputting RS latches. In: Cryptographic Hardware and Embedded Systems (CHES'11), pp. 390–406. Berlin, Heidelberg (2011)
34. Maes, R., Tuyls, P., Verbauwhede, I.: Intrinsic PUFs from flip-flops on reconfigurable devices. In: Proceedings of 3rd Benelux Workshop on Information and System Security (WISSec'08), p. 17 (2008)
35. Kumar, S.S., Guajardo, J., Maes, R., Schrijen, G.-J., Tuyls, P.: The butterfly PUF protecting IP on every FPGA. In: Proceedings of International Symposium on Hardware Oriented Security and Trust (HOST'08), pp. 67–70 (2008)