A Large Scale Comprehensive Evaluation of Single-Slice Ring Oscillator and PicoPUF Bit Cells on 28nm Xilinx FPGAs

Chongyan Gu
Queen's University Belfast
Belfast, U.K.
cgu01@qub.ac.uk

Chip Hong Chang
Nanyang Technological University
Singapore
echchang@ntu.edu.sg

Weiqiang Liu
Nanjing University of Aeronautics and Astronautics
Nanjing, China
liuweiqiang@nuaa.edu.cn

Neil Hanley
Queen's University Belfast
Belfast, U.K.
n.hanley@qub.ac.uk

Jack Miskelly
Queen's University Belfast
Belfast, U.K.
jmiskelly08@qub.ac.uk

Máire O'Neill
Queen's University Belfast
Belfast, U.K.
m.oneill@ecit.qub.ac.uk

ABSTRACT

Many field programmable gate array (FPGA)-based security primitives have been developed, *e.g.*, physical unclonable functions (PUFs) and true random number generator (TRNG). To accurately evaluate the performance of a PUF or other security designs, data from a large number of devices are required. A slice is the smallest reconfigurable logic block in an FPGA. The maximum or minimum entropy, exploitable from each slice of an FPGA, is an important factor for the design of a single-bit disorder-based security primitive. 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 (PicoPUF) cells that can each be fully fitted into a single slice are evaluated. 217 Xilinx Artix-7 FPGAs has been employed to provide a large-scale comprehensive analysis for the two designs. This is the first time two different single slice based security entities have been investigated and compared on 28nm Xilinx FPGA. Experimental results, including uniqueness, uniformity, correlation, reliability, bit-aliasing and min-entropy, based on 4 different floorplan locations are presented. The experimental results demonstrate that the lower the correlation between devices, the higher the min-entropy and uniqueness for both designs on the FPGAs. While the implementation location of both designs on the FPGA affects their performances, the overall min-entropy, correlation and uniqueness of PicoPUF are slightly higher than those of RO. All other metrics, including uniformity, bit-aliasing and reliability of the PicoPUF are slightly lower than those of the RO. The raw data for the PicoPUF design is made publicly available to enable the research community to use them for benchmarking and/or validation.

ASHES'19, November 15, 2019, London, United Kingdom

ACM ISBN 978-1-4503-6839-1/19/11…$15.00
https://doi.org/10.1145/3338508.3359570

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

KEYWORDS

FPGA, entropy, single slice, PUF

ACM Reference Format:
Chongyan Gu, Chip Hong Chang, Weiqiang Liu, Neil Hanley, Jack Miskelly, and Máire O'Neill. 2019. A Large Scale Comprehensive Evaluation of Single-Slice Ring Oscillator and PicoPUF Bit Cells on 28nm Xilinx FPGAs. In *3rd Attacks and Solutions in Hardware Security Workshop (ASHES'19), November 15, 2019, London, United Kingdom.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3338508.3359570

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 the same monolithically integrated chip is 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 unforgeable identity of an integrated circuit (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, anti-counterfeiting and supply chain security. Some PUFs can also be used or doubled as TRNGs or hybrid TRNGs. TRNG is another widely used hardware security primitive that makes use of noise and non-systematic variations of 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 application-specific integrated circuit (ASIC) and FPGA based PUFs is that individual devices of a ASIC PUF are not manufactured until the design has been physically placed and routed whereas the hardware resources of a FPGA PUF

have already been manufactured prior to physical design. Consequently, the maximum and minimum entropy that can be extracted from an FPGA chip for PUF becomes more dependent on the size and locality of its bit cells even though every bit cell is identically designed. Specifically, the entropy of each slice, which is the minimum reconfigurable unit of a FPGA, is an essential factor that contributes to the quality of the security primitive, particularly when it is to be extracted for the purpose of random number generation. Investigating and evaluating the entropy contributed by each slice of a 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 a FPGA device, which introduces inaccuracy and inconsistency in such 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 basic repetitively used core elements can be implemented on a single FPGA slice for this evaluation. The oscillation frequency generated by the same smallest 3-stage RO are different from slice to slice and from device to device. RO is 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 RO have been proposed in the literature, such as the original ring oscillator PUF (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 a 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. More specifically, our research contributions are summarized as follows.

- The testbed comprising 217 FPGA devices with 8,000 PicoPUF instances and 6,592 RO instances is, to the best of the authors' knowledge, the largest reported to date.
- A fair and comprehensive large-scale experimental analysis of uniqueness, reliability, uniformity, bit-aliasing, correlation, and min-entropy for both RO and PicoPUF.
- The impact of floorplan implementation location on min-entropy of RO populations and quality metrics of PicoPUF evaluated over a large-scale testbed.
- A detailed analysis and comparison of the two single slice entropy sources for design consideration and application of security primitives made of these components on FPGA platform.
- The raw data is made publicly available to the research community as a reference for further research into the design and implementation of security primitives on FPGA.

2 RELATED WORKS

A number of previous works have examined ROs on FPGA in the context of PUFs [5, 8–10], as well as process variations [11, 12]. However, there are only two existing large-scale RO PUF datasets on FPGAs. In [13], the testbed comprises 193 FPGA devices and 512 ROs, which is smaller in size than this work. Moreover, their data is generated from Xilinx Spartan-3 FPGAs, which are somewhat outdated compared to the current FPGA families. Recently, another dataset based on 100 Xilinx 28nm Artix-7 FPGAs is provided by [14] 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, $16 \times 80 = 1,280$, 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 [15]. This is acceptable for some types of analysis, but it is also restrictive wtih regards to inter-device variation analysis. The UNIQUE project also investigated RO PUFs on a large number of devices (96) [16], but it targeted 65nm ASICs instead of FPGAs. The impact of floorplan location for RO PUFs has also been investigated previously [17] based on a small scale testbed of FPGAs. A large scale RO PUF analysis in terms of slice type, evaluation time and temperature on 28nm Xilinx FPGAs is provided in [18]. 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. For the PicoPUF design [3], this is the first evaluation of such a single slice delay-based PUF design on large scale testbed. It is also the first investigation and comparison of two different known single slice based PUF cell designs.

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. 1. An *enable* input activates or deactivates the oscillator and an output buffered by a toggle flip flop is used to generate a signal. It can be compactly fitted in a single Xilinx Artix-7 slice. We fix the physical placement and routing paths of the ROs consistently over all the FPGAs.

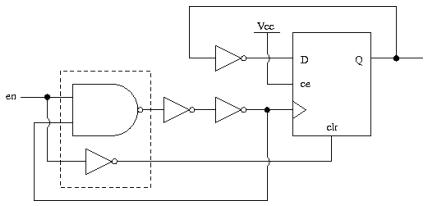


Figure 1: Single-slice RO cell [18].

3.2 PicoPUF Design

The PicoPUF design [3, 4] under test is shown in Fig. 2a. It is based on a cross-coupled NAND construction, with the input signals provided by two D flip flops (DFFs), which have synchronous enable and asynchronous clear signals, with the inputs D connected to 1. To evaluate the PicoPUF, the DFFs are first cleared such that the outputs Q0 and Q1 are both 0. This places the circuit in a stable state where the inputs of each NAND gate are 0 and 1, and both outputs are 1. Once the clock enable signal is set, this creates a

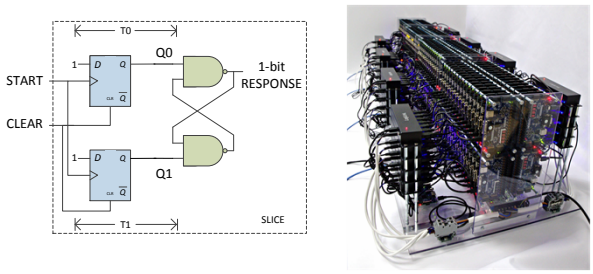


Figure 2: (a) Single slice PicoPUF cell [3] and (b) the hardware testing platform.

Table 1: The Number of ROs and PicoPUFs Based on Different Locations of An FPGA.

Location	#ROs	#PicoPUFs
LEFT-UPPER	1,600	1,952
LEFT-LOWER	1,600	1,952
RIGHT-UPPER	1,696	2,048
RIGHT-LOWER	1,696	2,048
Total	6,592	8,000

race condition between the NAND gates such that one gate will have a $1 \rightarrow 0$ transition, while keeping the second gate 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.

The architecture was designed to fit into a single slice of Xilinx Artix-7 FPGA. However, it can also be placed into a single slice on previous generation of FPGA devices such as Spartan-6 or Virtex-5/6 FPGAs.

4 FPGA IMPLEMENTATION

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

The names and numbers of the 4 different RO and PicoPUF types are listed in Table 1. The numbers of ROs and PicoPUFs implemented on the Artix-7 are 6, 592 and 8, 000, respectively.

The PicoPUF implemented on 4 different locations of a Xilinx Artix-7 FPGA is shown in Fig. 3. The routing for any path which contributes directly to the race condition was fixed using hook script in the Vivado design flow. The detailed implementations of both the RO and PicoPUF designs can be found in [18] and [3], respectively.

5 EXPERIMENTAL RESULTS

5.1 Experimental Setup

Fig. 2b shows the experimental setup, which consists of four modules in total, each of which holds 60 Basys-3 boards, 10 7-port

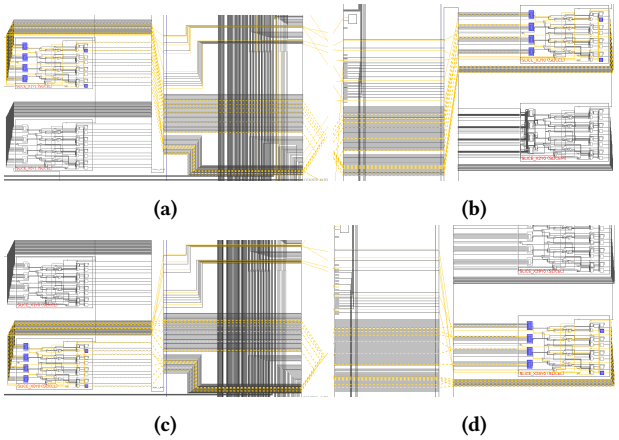


Figure 3: The fixed routings of the PicoPUF implementations at the 4 different locations, (a) *LEFT-UPPER*, (b) *RIGHT-UPPER*, (c) *LEFT-LOWER* and (d) *RIGHT-LOWER*.

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 program the FPGA with the design under test. A UART interface is used to communicate with the configured design and receive the measurement results. The Raspberry-Pi communicates over a local area network (LAN) with a global experiment control server, which also stores the measured data.

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

5.2 Overall Metrics

In order to evaluate and compare the designs from a security viewpoint, a number of metrics have been suggested [19]. In this paper, we will show the experimental results for each design for the following metrics: uniqueness, reliability, uniformity, bit-aliasing, correlation and min-entropy. Experimental results for both the RO and PicoPUF designs based on different implementation locations are shown in Table 2. 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 devices should have the same response, and learning the response from a (large) number of devices should not allow an adversary to infer any information about the response from a different device. It measures the fractional Hamming distance (HD) between each pair of responses from the devices, where 0 indicates none of the bits are the same, and 1 means that all the bits are identical. Ideally, the expected distance between any pair of responses is 0.5. The uniqueness experiment

Table 2: Experimental Results for Both the 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)		1,952	1,600	1,952	1,600	2,048	1,696	2,048	1,696	8,000	6,592	-
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%
Bit-Aliasing	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
	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	$\mu_{H_{min}}$	0.7585	0.7734	0.8826	0.7945	0.7706	0.7701	0.8781	0.7928	0.8225	0.7825	1.00
	$\sigma_{H_{min}}$	0.1278	0.0768	0.0861	0.0813	0.1261	0.0786	0.0897	0.0794	0.1236	0.0790	0.00

represents the best result in a row — μ is the mean value

represents the worst result in a row — σ is the standard deviation value

Note, all the values for RO PUF are the best results over all the evaluation time.

is taken from 217 devices to generate 217 responses in total with 6,592 bits generated by 6,592 independent single-slice bit cells for each response of a device. The average uniqueness result of the PicoPUF (0.4886) is almost the same as that of the RO (0.4805).

5.4 Correlation

Correlation indicates whether a relationship exists between devices in which the bit response from one device can be used to predict the response from another device. Correlations reduce the effort of an adversary to predict the secret from collected responses of other devices. In Table 2, the best correlation result (0.0102) is from the PicoPUF from the *RIGHT-UPPER* location and the worst one (0.0605) is from the RO from the *LEFT-LOWER* location. The best correlation result is from the PicoPUF, which is 0.0102, and the worst is from the RO at 0.0605. As a consequence, the PicoPUF provides a lower correlation between devices than the RO.

Fig. 4a shows the correlation results of the RO frequencies at the 4 different RO locations for 15 evaluation time. The longer the evaluation time for the ROs, the lower the correlation between the devices. The correlations from both the *LEFT-UPPER* and *RIGHT-UPPER* locations achieve a lower correlation than that of both the *LEFT-LOWER* and *RIGHT-LOWER* locations.

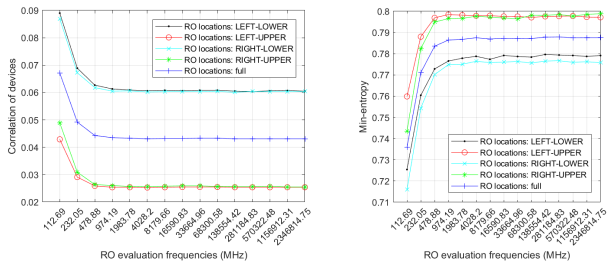
5.5 Min-entropy

The response is expected to be unpredictable such that an adversary can not efficiently predict the response of a device through an (un)limited observation of other devices which he/she may own or have access to. In order to verify this unpredictability, min-entropy is commonly used as a worst case analysis for how much uncertainty is present in a response [20]. It is calculated bit-wise across the available FPGA devices as shown in (1).

$$\mathrm{H}_{\mathrm{min},b} = -\log_2\left(p_{b\,\max}\right) \tag{1}$$

where

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



(a) (b)

Figure 4: The correlation result for (a) the RO frequency, and the min-entropy result for (b) the RO frequency over a varying evaluation time.

Table 2 presents the min-entropy results for the RO and PicoPUF at 4 different locations. The best and worst min-entropy results of both designs are derived from the *LEFT-UPPER* and *LEFT-LOWER* locations, respectively. In particular, the PicoPUF at the *LEFT-LOWER* location has the worst standard deviation (STD) of 0.1278. The average min-entropy over all locations for the PicoPUF, 0.8225, is higher than that of the RO, which is 0.7825.

5.5.1 Effect of Locations and Evaluation Time for the ROs. Fig. 4b presents the min-entropy results of the RO frequencies at different RO locations for 15 RO evaluation time. The longer the RO evaluation time, the greater the min-entropy value. Moreover, the min-entropy does not change significantly when the RO evaluation time is larger than 974.19 MHz. The RO frequencies from both the *LEFT-UPPER* and *RIGHT-UPPER* locations achieve higher min-entropy than at other locations.

5.5.2 Effect of the Number of Devices. Fig. 5a and Fig. 5b show the min-entropy results of RO and PicoPUF at the 4 different locations and over different numbers of devices. It indicates that the larger the number of the devices, the higher the min-entropy result. It can be seen that approximately 140 devices ($m \geqslant 140$) 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. 5b, and from 0.7701 to 0.7945 for the RO as shown in Fig. 5a. Hence, the PicoPUF has a greater spread of min-entropy results than the RO over the different locations.

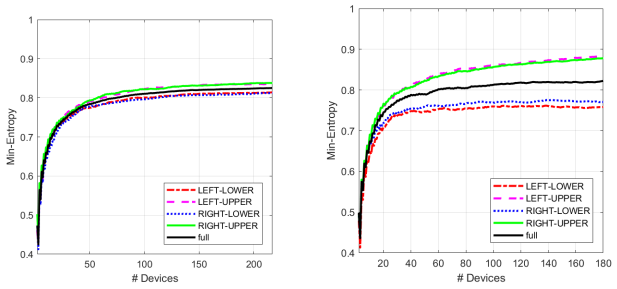


Figure 5: The min-entropy results for (a) the PicoPUF and (b) RO frequencies over a varying number of devices, respectively.

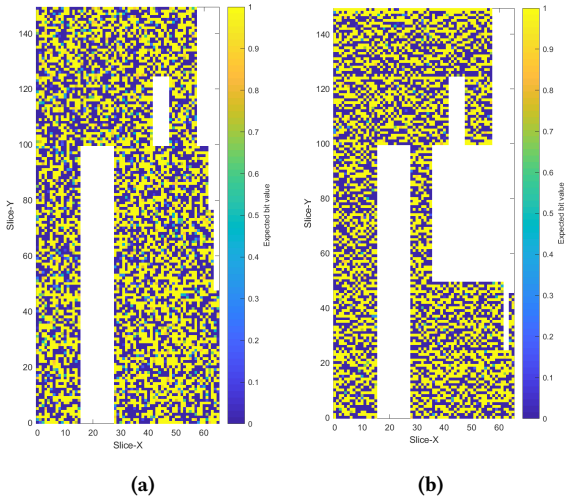


Figure 6: The heatmaps of the reliability results for (a) the PicoPUF and (b) RO frequencies, respectively.

5.6 Reliability

For practical use in PUF applications, the response generated by each bit cell must be reliable across repeated measurements. The greater the reliability, the less costly the error correction required. The 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, $r = 10,001$ repeated measurements were taken for every PicoPUF response bit, on each FPGA. Table 2 shows the result of the average response for each of the $m \times n = 180 \times 8,000 = 1.44M$ bits in the experimental dataset of the PicoPUF. For the PicoPUF, it can be seen that a significant portion of the bits, 44.87% (or 646,128 of 1440,000), returned 0 and 34.78% (or 500,832 of 1440,000) returned 1 for each of the r acquisitions. For the RO, the result of the average response for each of the $m \times n = 217 \times 6,592 = 1.43M$ bits is presented. It shows that 47.75% of the bits returned 0 (or 687,600 of 1440,000) and 48.11% returned 1 (or 692,782 of 1440,000) for each of the r acquisitions (Note, $r = 1,000$ for the RO). Hence, PicoPUF has approximately 10% difference between the number of stable 0's and 1's. On the other hand, RO bit cells has a smaller difference of approximately $1-3\%$, which is more reliable than the PicoPUF bit cells.

The heat-map in Fig. 6(a) 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 the appearance of 1 for each bit in r repetitions. It can be seen that the 1 or 0 bits are evenly and randomly distributed. A small number of bits are unreliable and they are also randomly distributed. The heat-map in Fig. 6(b) presents similar results for the RO. Note 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, *etc.*. The reliabilities of both PicoPUF and RO show no significant dependencies on the surrounding paths.

5.7 Uniformity

The uniformity metric depicts how the response from each device is split between [0,1]. It is essentially the expected "'weight"' of the response for a randomly chosen device calculated by taking the average of all the response bits. Ideally, this tends to 0.5. In Table 2, it can be seen that the best uniformity result, 0.5019, is from the RO at the *RIGHT-UPPER* location, and the worst, 0.4103, is from the PicoPUF at the *LEFT-LOWER* location. Additionally, the RO presents a better overall uniformity results than the PicoPUF.

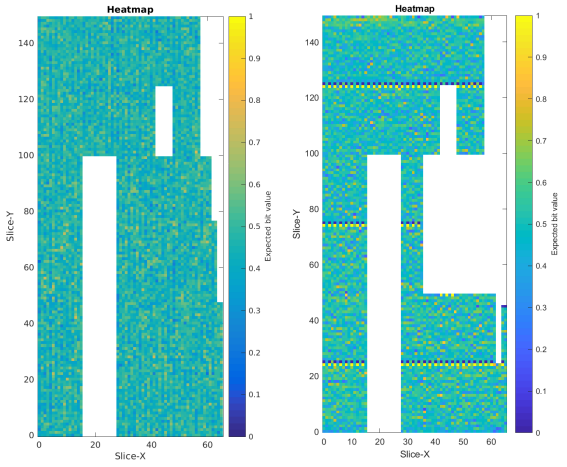
5.8 Bit-aliasing

Bit-aliasing investigates each of the response bits individually, to ensure that no physical locations of the FPGA are strongly biased towards [0,1]. This can be done by simply taking the average of each bit location across the number of available devices. This contains the expected bit response of each physical location of the target FPGA, which should be 0.5 for a well balanced design.

Heatmaps of the bit-aliasing results for both PicoPUF and RO are given in Fig. 7(a) and Fig. 7(b), respectively. In general, while no single slice location always returns the same value across different devices, a small number of cells are significantly biased. Biased values of either 1 or 0 are observed in the area adjacent to the clock distribution network for the clock tile as shown in Fig. 7(b) for the RO. As shown in Table 2, the best bit-aliasing result (0.5019) is from the RO at the *RIGHT-UPPER* location and the worst (0.4103) is from the PicoPUF at the *LEFT-LOWER* location.

5.9 Comparison and Discussion

The PicoPUF implementation at the *LEFT-UPPER* location derives the best uniqueness and min-entropy results but when implemented



(a) (b)

Figure 7: The heatmaps of the bit-aliasing results for (a) the RO frequencies and (b) the PicoPUF, respectively.

at the *LEFT-LOWER* location it provides the worst uniformity, reliability, bit-aliasing and min-entropy results. Hence, to achieve a better performance when designing a PicoPUF on an FPGA, the *RIGHT-UPPER* location should be the best choice. Interestingly, the RO implemented at the *RIGHT-LOWER* location achieves the best reliability result and in contrast, when implemented at both the *RIGHT-UPPER* and *LEFT-UPPER* locations, it achieves lower reliability but higher uniqueness results. Therefore, when designing a RO based security primitive, there is a trade-off between obtaining better uniqueness or better reliability depending on the implementation location choice. Considering the fact that RO-based design usually requires counters for calculating the RO frequencies, PicoPUF is a more lightweight choice than the RO for a design requiring better uniqueness and less hardware resources on FPGA.

6 CONCLUSION

In this work we presented a large scale analysis of two single-slice based bit cells on 217 Xilinx Artix-7 XC7A35T FPGAs. The entire fabric was covered by either 8,000 distinct PicoPUF cells or 6,592 RO instances. A fair and comprehensive experimental analysis of uniqueness, uniformity, correlation, reliability, bit-aliasing and min-entropy for two different types of designs, RO and PicoPUF, is presented for the first time. The experimental results show that the overall min-entropy, correlation and uniqueness of PicoPUF are slightly higher than those of RO, while the other metrics, including uniformity, bit-aliasing and reliability are slightly lower than those of RO. Moreover, the experimental results show that the lower the correlation between devices, the higher the min-entropy and uniqueness for both designs on the FPGA. Finally, it is shown that the implementation location 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 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 AVAILABILITY

The raw PicoPUF and RO frequency data is publicly available at QUB-CSIT-Raw-Picopuf-Data and EU-FP7-SPARKS-RO-DATA, respectively, for future research on PUFs designs.

ACKNOWLEDGMENTS

This work was partly supported 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 and National Natural Science Foundation of China (61771239).

REFERENCES

[1] K. H. Tsoi, K. Leung, and P. H. W. Leong, "Compact fpga-based true and pseudo random number generators," in *Proc. 11th Annual IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM)*. IEEE, 2003, pp. 51–61.
[2] V. Fischer, F. Bernard, N. Bochard, and M. Varchola, "Enhancing security of ring oscillator-based TRNG implemented in FPGA," in *Proc. International Conference on Field Programmable Logic and Applications (FCCM)*. IEEE, 2008, pp. 245–250.
[3] C. Gu, J. Murphy, and M. O'Neill, "A unique and robust single slice FPGA identification generator," in *Proc. Int. Symp. on Circuits and Syst. (ISCAS'14)*. Melbourne, Australia: IEEE, Jun. 2014, pp. 1223–1226.
[4] C. Gu and M. O'Neill, "Ultra-compact and robust FPGA-based PUF identification generator," in *Proc. Int. Symp. on Circuits and Syst. (ISCAS'15)*. Lisbon, Portugal: IEEE, May 2015, pp. 934–937.
[5] G. E. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in *Proc. 44th Design Automation Conference (DAC)*. ACM, 2007, pp. 9–14.
[6] M.-D. M. Yu and S. Devadas, "Recombination of physical unclonable functions," in *Proc. 35th Annual. GOMACTech Conference.* United States, Mar. 2010.
[7] C. Gu, W. Liu, N. Hanley, R. Hesselbarth, and M. O'Neill, "A theoretical model to link uniqueness and min-entropy for PUF evaluations," *IEEE Trans. Comput.*, pp. 1–1, 2018.
[8] A. Maiti and P. Schaumont, "Improved ring oscillator PUF: an FPGA-friendly secure primitive," *Journal of Cryptology*, vol. 24, no. 2, pp. 375–397, 2011.
[9] D. Merli, F. Stumpf, and C. Eckert, "Improving the quality of ring oscillator PUFs on FPGAs," in *Proc. the 5th ACM workshop on embedded system security*, 2010, p. 9.
[10] F. Kodỳtek and R. Lórencz, "A design of ring oscillator based PUF on FPGA," in *Proc. IEEE 18th Int. Symp. on Des. and Diagnostics of Electronic Circuits & Syst. (DDECS'15)*, 2015, pp. 37–42.
[11] H. Onodera, "Variability: Modeling and its impact on design," *IEICE Transactions on Electronics*, vol. 89, no. 3, pp. 342–348, 2006.
[12] L.-T. Pang and B. Nikolic, "Measurements and analysis of process variability in 90 nm CMOS," *IEEE J. Solid-State Circuits*, vol. 44, no. 5, pp. 1655–1663, 2009.
[13] A. Maiti, J. Casarona, L. McHale, and P. Schaumont, "A large scale characterization of RO-PUF," in *Proc. IEEE Int. Symp. on Hardware Oriented Security and Trust (HOST'10)*, 2010, pp. 94–99.
[14] A. Wild, G. T. Becker, and T. Guneysu, "A fair and comprehensive large-scale analysis of oscillation-based PUFs for FPGAs," in *2017 27th International Conference on Field Programmable Logic and Applications (FPL)*, Sept 2017, pp. 1–7.
[15] W. Che, V. K. Kajuluri, M. Martin, F. Saqib, and J. Plusquellic, "Analysis of entropy in a hardware embedded delay PUF," *Cryptography*, vol. 1, no. 1, p. 8, Jun. 2017.
[16] S. Katzenbeisser, Ü. Kocabaş, V. Rožić, A.-R. Sadeghi, I. Verbauwhede, and C. Wachsmann, *PUFs: Myth, Fact or Busted? A Security Evaluation of Physically Unclonable Functions(PUFs) Cast in Silicon.* Berlin, Heidelberg: Springer Berlin Heidelberg, Sep. 2012, pp. 283–301.
[17] W. Liu, Y. Yu, C. Wang, Y. Cui, and M. O'Neill, "RO PUF design in FPGAs with new comparison strategies," in *Proc. IEEE Int. Symp. on Circuits and Syst. (ISCAS'15)*, May 2015, pp. 77–80.
[18] R. Hesselbarth, F. Wilde, C. Gu, and N. Hanley, "Large scale RO PUF analysis over slice type, evaluation time and temperature on 28nm xilinx FPGAs," in *Proc. IEEE Int. Symp. on Hardware Oriented Security and Trust (HOST'18)*, April 2018, pp. 126–133.
[19] A. Maiti, V. Gunreddy, and P. Schaumont, "A systematic method to evaluate and compare the performance of physical unclonable functions," Cryptology ePrint Archive, Report 2011/657, 2011.
[20] M. Claes, V. van der Leest, and A. Braeken, "Comparison of SRAM and FF PUF in 65nm technology," in *Proc. Nordic Conf. on Secure IT Syst.* Springer, 2011, pp. 47–64.