In-depth Analysis and Enhancements of RO-PUFs with a Partial Reconfiguration Framework on Xilinx Zynq-7000 SoC FPGAs

Andreas Herkle, Holger Mandry, Joachim Becker, Maurits Ortmanns
University of Ulm, Institute of Microelectronics, Ulm, Germany
{andreas.herkle, holger.mandry, joachim.becker, maurits.ortmanns}@uni-ulm.de

***Abstract*—Physical unclonable functions (PUFs) are excellent candidates to generate secret information on-chip without the need for secure storage. Ring-oscillator (RO) based PUFs have been receiving great attention over the years due to their easy design and superior statistical characteristics on field programmable gate arrays (FPGAs). Although previous work has improved their statistical measures and provided deeper insights, there are still gaps to be filled. Therefore, this work presents an in-depth analysis of RO-PUFs on Xilinx Zynq-7000 FPGAs with a framework based on partial reconfiguration. This approach allows for full-chip characterization of 100% of the targeted area. Based on the measured data and beforehand estimated routing delay, we will show how to identify and avoid potential bias in the final PUF placement. By utilizing DSP48 slices, an enhanced counter was designed for high-frequency measurements. A second feedback path was added to the ring-oscillators in order to avoid glitches at the counters input. In combination with a reference normalization, the frequency standard deviation could be reduced to 0.0229% at a much shorter evaluation time of 10µs compared to the state-of-the-art, while maintaining the maximum inter-hamming distance. An investigation on the influence of spatial distribution on different RO pairings was performed. The chip variations were found to have a much larger effect on the statistical measures than the difference between logic elements. The measurement data and the framework will be made accessible to interested researchers to provide them with a data basis for further research.**

***Index Terms*—Physical Unclonable Function (PUF), Ring-oscillator (RO), Field Programmable Gate Array (FPGA), Zynq-7000, Partial Reconfiguration**

I. INTRODUCTION

In order to guarantee trust-worthy authentication in hardware-cryptography, a source of uniqueness or a key is involved, which has to be kept secret, non-accessible and tamper resistant to potential attackers. The demand for non-volatile secure memory usually means a large effort for intrusion detection and prevention in addition to a constant power supply, especially for low-power IoT devices. In comparison, PUFs fulfill many of these demands already by design. PUFs are electrical circuits, which utilize uncontrollable variations from manufacturing to generate device specific fingerprints. Delay-based PUFs [1] measure the small differences of identically routed delay lines and output the results as a quantized value. The circuit itself only has to be activated for generating the output, which minimizes the time frame for possible side-channel attacks. As the variations stay constant, no secure memory is needed but the response can be regenerated on demand. Due to their identical design, the output can not be predicted for specific devices and should not be predictable for specific positions on a chip.

For FPGAs, it has been shown that ring-oscillators are one of the most preferable implementation choices [2]: they can be kept very small [3], have very few design restrictions [4], achieve low bit-error rates and close-to-ideal uniqueness [5]. Many publications have targeted this specific PUF type, with the aim to maximize the generated key length [6], analyze frequency characteristics [7], [8] and minimize the influence of adjacent switching logic [9].

Our work contributes to that field with an in-depth analysis of Xilinx Zynq FPGAs with the aim to maximize the characterized chip area, analyze and minimize causes of bit-errors and counter failures and reduce the necessary evaluation time.

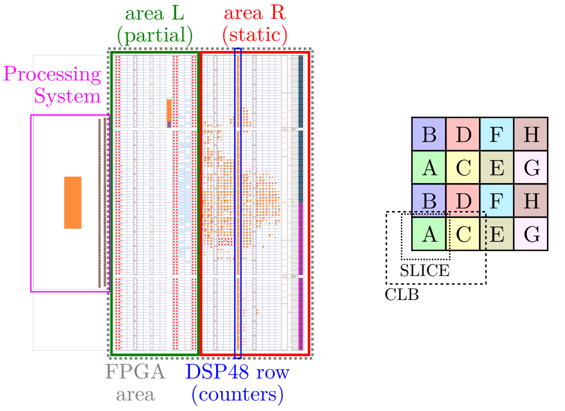
The rest of this work is organized as follows: Section II presents our partial reconfiguration based measurement framework, with additional details on enhanced counters, reference generation and a novel addition to the traditional RO-PUF design. In Section III, we present an analysis of our measurements and the implications, which come from chip characteristics extracted from the tool-chain in advance. In Section IV, we analyze the possible cause of counter failures, present an in-depth explanation of our novel RO design and the improvements due to reference normalization. The analysis is completed in Section V with an evaluation of PUF metrics and further testing of alternative RO pairing. Section VI concludes our work.

II. FRAMEWORK AND SYSTEM DESIGN

The presented measurement method takes full advantage of several design aspects Xilinx offers for their Zynq All-Programmable System-on-Chip (AP SoC) based chips. We will cover them in the following Subsections and explain how these aspects add improvement to a generic measurement system.

A. Partial Reconfiguration Framework

Xilinx FPGAs support a partial reconfiguration flow, which enables dynamic reconfiguration of a dedicated area of the FPGA at runtime by loading a partial bitstream over the



(a) (b)

Fig. 1: (a) Segmented chip overview, (b) RO placement grid for different bitstreams with a security margin of one CLB

configuration access port, while leaving the remaining area untouched. The reconfiguration can also be carried out by the dedicated CPU in the Processing System of the chip, which is an outstanding feature of Zynq SoCs. Such a design flow has already been used in [6] to extract as much entropy as possible from a small dedicated area. This reconfiguration method is utilized in order to fully characterize the whole chip area. The principle is depicted in Fig. 1 (a), which shows the complete chip area and the FPGA design split in half.

On the left side is the *partial area (L)*, in which PUFs can be placed at varying positions, by loading the respective partial bitstreams. The *static area (R)* on the right side contains the remaining logic for a complete measurement setup such as counters, timers, and data-access logic for the CPU. This approach guarantees the following non-trivial advantages.

First, the implementation runs in Xilinx Vivado are organized such that the static area is an unchanging single *parent* implementation with an empty partial area. The partial area allocations are built as *children* implementations. The children implementations inherit the static area from their respective parent implantation and thus, the static area and the measurement conditions stay the same for all PUFs. This approach allows fair and unbiased analysis.

Second, the whole *partial area* contains nothing but the PUFs themselves, eliminating possible switching activity from nearby logic [10]. This also ensures that the PUF instances can be placed with any spacing distance, which acts as additional security margin [9] to avoid crosstalk between the PUF instances. Most important, the PUF instances can be placed at any arbitrary position, which enables the possibility to successively cover the whole area over multiple implementation runs. The principle of placement is shown in Fig. 1 (b) in a logical view, meaning that two horizontal neighboring SLICEs form a CLB, e.g. SLICE A and C in the lower left corner. In one partial implementation, all PUF instances are placed into the SLICEs indexed as A, in another partial implementation into the SLICEs indexed as B and so on. We developed a TCL script, which automatically creates all runs needed to cover the complete partial area with a few given parameters (X/Y boundaries of the FPGA area, desired spacing, etc.).

TABLE I: Comparison of chip area coverage on Xilinx FPGAs

design	**chip**	**utilized**	**available**	**coverage**
[11]	Spartan-3E 500E	512	1164	44.0%
[10]	Spartan-3E 1200E	5425	8672	62.5%
[5]	Artix-7 A35T	1280	5200	24.6%
[6]	Zynq Z020	1424	26600	5.4%[a]
[8]	Artix-7 A35T	6592	8150	80.9%
This work	Zynq Z010	3800	4400	86.4%[b]

[a] partial reconfiguration with the aim of minimum size

[b] 100% of partial reconfigurable SLICEs (targeted area)

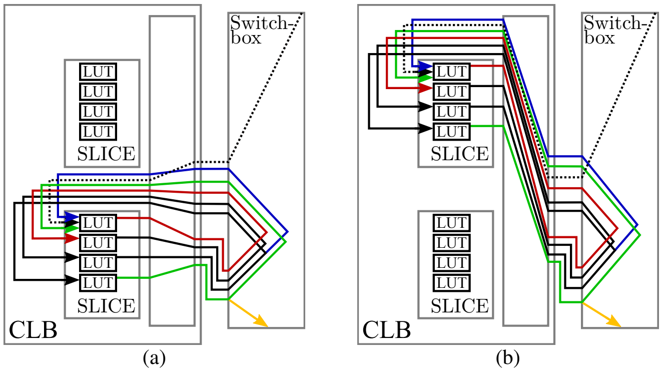
In a second static parent implementation, both areas are switched such that the *partial area* now covers the right side in Fig. 1 (a) and vice versa. Some columns in the left area of the chip are excluded, which are non-reconfigurable and dedicated for special purposes like AXI4 bus connections for the CPU. In conclusion, our design approach reaches the highest reported coverage on Xilinx FPGAs, 100% of the reconfigurable area, which corresponds to 86.4% percent of the whole chip area. A comparison to previous chip analysis is shown in Table I.

B. *Ring-Oscillator Design and Routing*

Logic cells in Xilinx FPGAs of the 7-series are organized as configurable logic blocks (CLB), in which the 6-input Lookup-Tables (LUT6) are organized in two stacked-on-top slices (SLICE) of different capabilities (logic only or additional memory) and the CLBs are placed left or right of their respective switchbox. We have chosen the naming scheme accordingly as follows: S for SLICE, the CLB type (CLB**ML** or CLB**LL**) as the first index, the orientation (**l**eft or **r**ight) as the second index, the SLICE type (SLICE**L** or SLICE**M**) as the third index and the relative position within the CLB (**t**op or **b**ottom) as the last index. E.g. the group of S_{LrLt} contains all logic SLICEs, located in the top of the CLBs of type CLBLL, located on the right side of their respective switchbox.

Our implementation of the ring-oscillator fits within a single SLICE, using four LUT6 elements in series, three of them as inverters with one preceding control stage. Short, local routing benefits the readout of ROs greatly [9]. Therefore, no chosen routing resource leaves the adjacent switchbox and out of multiple paths choices for each route, the one directly leading back to the SLICE without using any indirect bouncing resource was chosen. This reduces the number of involved buffering elements, thus further reduces electrical noise, and keeps the route delay to a minimum.

Both SLICEs of a CLB can be used to place our RO with a slightly different routing as shown in Fig. 2 (a) and (b), with a coloring scheme according to following figures. We will refer to these implementations according to their position within the CLB as the RO type bottom ro_b and top ro_t. Although the routing for ro_t looks longer, it actually has a lower mean delay and the illustration from Vivado is just misleading on



(a) (b)

Fig. 2: Ring-oscillator placement and routing in (a) the lower SLICE and (b) the upper SLICE of an individual CLB

the first sight. These two specific RO types are sufficient to cover all CLB and SLICE types and each orientation. The only notable difference is the varying naming of some intermediate routing resources, which are otherwise physically equivalent (e.g. `IMUX30` on the right side ↔ `IMUX_L30` on the left side of the switchbox; `CLBLM_L_C5` for memory CLBs ↔ `CLBLL_LL_C5` for logic CLBs).

C. Counter, Timer and Reference Ring-Oscillator

To the best knowledge of the authors, previous publications never investigated the importance of well-chosen measurement logic thoroughly. It is well known that the design (sub-elements, placement, routing, etc.) of ROs is the most crucial part of the PUF development cycle on FPGAs. Thus, one might think that counters are as trivial as a simple ”`count=count+1`” VHDL/Verilog statement. However, depending on the used Xilinx Design Constraints, this statement will either results in a *CARRY4* based counter or the instantiation of a hard-macro *DSP48* based counter. We will show in Section IV that the first topology is the worst synthesis result and present our optimized DSP48 based counter design now.

In our design, we omitted the usage of any frequency down-scaling [3], [6], [8] in order to avoid obscuring the interaction between the ROs and the counters. The DSP48 switching characteristics datasheet [12] reveals that, with all internal registers used, the maximum frequency for an adder topology is 464.25 MHz, yet the product specification datasheet [13] states that the maximum frequency for the internal accumulator is up to 741 MHz. The used concept is shown in Fig. 3, where the internal routing, set by unchanging control inputs (*OPMODE*, *ALUMODE*, *CARRYIN*, etc.), is indicated as the green lines. The clock input is the only active path from the outside and it is directly excited by the RO output. Thus, the only remaining critical path in the sense of synchronous logic is the feedback path from the register *P* output through the accumulator back to the register *P* input. In comparison, the *COUNTER_LOAD_MACRO* provided by Xilinx always instantiates counters as a feedback loop from the output *P* back to any of the inputs *A* to *D*, exemplary depicted in Fig. 3 as the dashed path in red. Obviously, this much longer external routing explains the lower maximum frequency. We couldn’t find any timing information in the datasheets about our chosen internal feedback path and suspect that this configuration originally was not intended to be used. Still, we can assume its superiority by design, which will be confirmed by measurements in Section IV.

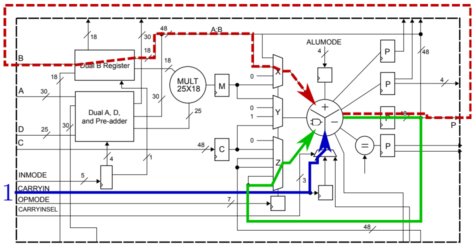


Fig. 3: DSP48 slice [14] instantiated as counter, the optimum routing indicated in blue and green, the standard macro routing indicated in dashed red

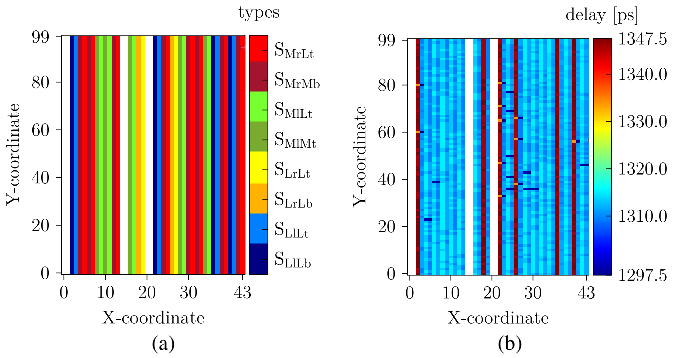
In addition to the design as counters, the dedicated DSP48 slices were also utilized as timers, driven by a 100 MHz clock generated in the processing system. As long as timers provide an enable signal of equal length for each measurement, the timers’ influence on the readout quality is marginal. However, clock jitter, voltage drop and crosstalk not only impact the ROs but also the timers, slightly altering the evaluation time. Therefore, a single timer instance was used for all ROs to ensure consistency at least on a per-evaluation base.

To further enhance the measurements quality, we instantiated a reference RO in the static area, similar to [15]. Its placement is not varied for the evaluation of the whole partial area and its reference measurements will be used in Section V to normalize the ROs readouts.

D. Software Control and Measurement Setup

The Zynq-7000 family are so-called All Programmable (AP) Systems-on-Chip (SoC), where the programmable logic of the FGPA is tightly interconnected with a Cortex-A9 CPU in the Processing System (PS). The PS was utilized to outsource as many controlling functionality as possible. This approach provides a more flexible control than multiple fixed bitstreams and speeds up prototyping. The CPU controls all parametric setup for each single measurement, e.g. the evaluation time and number of readouts, loads the *parent* toplevel bit-files and consecutively the associated *partial* bit-files from an SD-Card, reads the measurement results directly from the DSP48 counters and stores them back to the same SD-Card. This self-sufficient, plug-and-play method reduces the need for readout buffering on-chip and control signals from the outside to a minimum and increases the number of storable results.

For ease of comparing results, we have chosen the indices similar to [8]. The measurement indices are as follows: r indicates repetitive readouts, n indicates the different ROs on a chip and i indicates different chips. A complete data set contains $M_r = 1000$ readouts from each of the $M_n = 3800$ ring-oscillators per chip, with a total of $M_i = 20$ chips (XC7Z010-



(a) (b)

Fig. 4: Heatmap illustration of (a) the SLICEs type composition and (b) the estimated delay

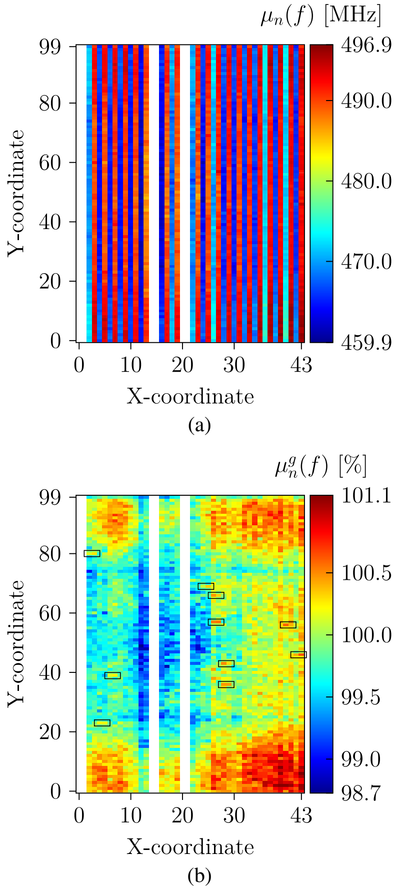
1CLG400C on the ZYBO) measured at room temperature. Each data set contains serial (1 RO at a time) and parallel (all 32 ROs of one partial bitstream at once) measurements for three evaluation times: 1 µs, 10 µs and 100 µs. Three different designs were investigated, a conventional RO design, a novel RO design presented in Section IV and a design with a conventional CARRY4 based counter.

III. FREQUENCY AND HEATMAP ANALYSIS

In this Section, we will analyze the measured frequencies, and their resemblance with the estimated delays. The Vivado toolchain allows for inspection of estimated delays of routed nets and logic elements, calculated by the Vivado routing algorithms based on specific routes, placement and timing models. We developed a post-implementation TCL script, which extracts all these estimated delays for each of the RO instances and the respective SLICE type properties. The spatial distribution of all different SLICE types existent on the chip is shown in Fig. 4 (a). Although the *physical* arrangement of SLICEs within the same CLB is actually stacked-on-top, our chosen *logical* arrangement resembles more of a quadratic shape, which is closer to the real physical dimensions of the chip. Additionally, each column of the matrix now contains only SLICEs of the very same type, which illustrates potential differences more clearly. By combining all SLICEs with the same name, column-wise distinguishable groups can be formed and it has already been proposed in the literature [9] that these groups should not be mixed to avoid the introduction of systematic bias.

Fig. 4 (b) shows the estimated delay for each RO, extracted with our TCL script from post-implementation results. This delay map provides some insightful knowledge before any measurement, with three outstanding observations leading to the following assumptions.

Assumption 1. ROs in SLICEs of the type S_{LlLb} and S_{LrLt} are expected to have a significantly higher delay than any other group of ROs.



(a)

(b)

Fig. 5: Heatmap illustration of (a) the mean frequencies $\mu_n(f)$ and (b) the group-normalized mean frequencies $\mu_n^g(f)$

Assumption 2. The delay offset due to a different routing between the two RO types $\mathrm{ro_b}$ and $\mathrm{ro_t}$ should be observable as an interleaved column pattern.

Assumption 3. At some scattered positions with no obvious regularity, the estimated delay strongly deviates from the respective group mean, which should be observable as large frequency deviations of single ROs during measurement.

Fig. 5 (a) shows the map of mean frequencies $\mu_n(f)$ for each $ro_n \in M_n$, calculated from M_i chip measurements, each measurement with $k_{i,n,r}$ counted edges in an evaluation time of Δt = 100µs and successive activation:

$$f := f_{i,n,r} = \frac{k_{i,n,r}}{\Delta t} \tag{1}$$

$$\mu_n(f) := \mathrm{mean}_{\forall i}(\mathrm{mean}_{\forall r}(f)) \tag{2}$$

We examined these assumptions from the estimated delay map and compared it to the averaged measured results.

Fig. 5 (a) not only disproves Assumption 1, but also shows that the mean frequency (471.2 MHz) of the group $S_{LlLb} \cup S_{LrLb}$ is even higher than that of any other group with RO type $\mathrm{ro_b}$ ($\leq$ 470.1 MHz). We found no explanation for why the routing algorithms of Vivado over-estimated these groups delay in such a large quantity. Within the group of RO type

$\mathrm{ro_b}$, memory slices (SLICEM) have a lower mean frequency, as it has been shown before [9].

Assumption 2 holds true as the routing difference between RO type $\mathrm{ro_b}$ and $\mathrm{ro_t}$ obviously has a much larger influence on the mean frequency than the different types of SLICEs (red and blue stripes). The RO types complete statistics are listed in Table II as absolute values and as percentage of the overall mean, listed as $\mathrm{ro_{b+t}}$. These numbers explain the visual pattern clearly, as the maximum frequency of $\mathrm{ro_b}$ is still lower than the minimum frequency of $\mathrm{ro_t}$.

With the raw measurement data, no more obvious observations can be depicted, because the large delay difference between $\mathrm{ro_b}$ and $\mathrm{ro_t}$ overshadows less prominent deviations. In order to analyze the frequencies on a type-independent level, a frequency normalization was applied, dividing the mean frequency $\mu_n(f)$ of each ro_n by the mean frequency of its affiliated group. Equation 3 describes the selection method for grouping, indicated as g, which is used for all following analysis in an equivalent way:

$$\mu_{n|g}(f) \subset \mu_n(f) | g \in S_{LlLe}||...||S_{MrLo} \tag{3}$$

$$\mu_n^g(f) = \frac{\mu_{n|g}(f)}{\mathrm{mean}_{\forall n|g}(\mu_{n|g}(f))} \tag{4}$$

This normalization removes any difference due to routing and SLICE type. By averaging over all measured chips, the PUF-defining local mismatch of transistor parameters is also eliminated, which only leaves the influence of chip-structure dependent sources. These, in general, can be non-uniform voltage drop, non-uniform temperature distribution, adjacent switching logic and many more. The map of this normalized distribution is depicted in Fig. 5 (b) and shows roughly six distinct regions. Four of them (mostly red), located in the corners of the chip, have a higher relative frequency with an increase of up to 1.23%. One area (mostly blue), located left to the middle of the chip, has a much lower relative frequency with a decrease in the same range of up to 1.23%. The area closest to the average (mostly green) can be found to the right side of the middle. This observations further restricts possible RO-PUF pairing such that the spatial distance of selected pairs of ROs, even within the same groups, should be kept to a minimum [10] to prevent unintended biasing, which will be further supported by the analysis in Section V.

A second observation can be made with respect to Assumption 3. Some of the expected outliers can be easily found again in Fig. 5 (b), while some blur with their surroundings and their effect seems to be vanished. The positions of the observably influential outliers in Fig. 5 (b) are marked with black boxes. In conclusion, both maps of Fig. 4 should be used to sort out any potentially biased position.

Further frequency analysis can be done with the standard deviation between chips and its group-wise normalization:

$$\sigma_n(f) = \mathrm{std}_{\forall i}(\mathrm{mean}_{\forall r}(f)) \tag{5}$$

$$\sigma_n^g(f) = \frac{\sigma_n(f)}{\mathrm{mean}_{\forall n|g}(\sigma_n(f))} \tag{6}$$

TABLE II: Statistics of both RO types, as measured frequencies and in percentages of $\mathrm{ro_{b+t}} = \mathrm{ro_b} \cup \mathrm{ro_t}$

		$\mathrm{ro_b}$		**$\mathrm{ro_t}$**		**$\mathrm{ro_{b+t}}$**
statistics		[MHz]	[%]	[MHz]	[%]	[MHz]
$\mu_n(f)$	min	459.9	95.7	485.3	100.9	480.8
	mean	485.3	97.2	491.2	102.2	
	max	476.3	99.1	496.9	103.4	
$\sigma_n(f)$	min	23.6	87.2	25.0	92.3	27.0
	mean	26.2	96.9	27.9	103.2	
	max	29.1	107.6	31.3	115.7	

$\sigma_n(f)$ is depicted in Fig. 6 (a) as raw values and the same stripe pattern as in Fig. 5 (a) can be observed, though the observable inter-column difference becomes smaller. The group of RO type $\mathrm{ro_t}$ shows a mean standard deviation about 1.7 MHz higher than RO type $\mathrm{ro_b}$, which corresponds to the higher mean frequency of $\mathrm{ro_t}$. In Fig. 6 (b), $\sigma_n^g(f)$ shows a very distinct distribution of frequency variability with areas more prone to frequency deviation, which is beneficial for PUFs in general. In our case of a Zynq-7000 chip, a larger variability can be found in the upper half of the chip area, which shows some similarity to [8], where the upper half left corner of Artix-7 chips had a larger standard deviation.

Table II also lists the $\sigma_n(f)$ statistics for both RO types separately and as percentages of $\mathrm{ro_{b+t}}$. In opposite to their respective mean frequencies, which were clearly separated, the distributions of both $\sigma_n(f)$ overlap frequency-wise as well as percentage-wise. Consequently, the influence of routing and group differences on the frequency standard deviation is less dominant compared to its influence on the mean frequency. Vice versa, the influence of spatial variations has a much larger effect on the standard deviation of the ROs, which is why its pattern in Fig. 6 (b) is already visible in the non-normalized plot in Fig. 6 (a).

Fig. 6 (c) shows the map of the group-normalized mean of the frequencies median absolute deviation, which was chosen to avoid distortion by very large outliers:

$$\tilde{\mu}_r(f) = \mathrm{mean}_{\forall i}(\mathrm{mad}_{\forall r}(f)) \tag{7}$$

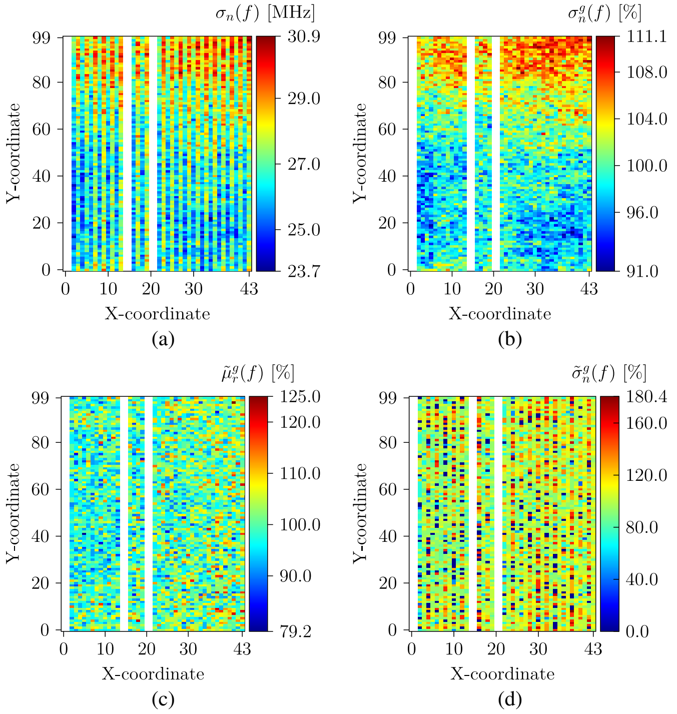
$$\tilde{\mu}_r^g(f) = \frac{\tilde{\mu}_r(f)}{\mathrm{mean}_{\forall n|g}(\tilde{\mu}_r(f))} \tag{8}$$

$\mu_r^g(f)$ shows how strong the different ROs deviate from their respective mean frequency over multiple readouts due to noise. This measure can be seen as a coarse indicator for the expected bit-error rate and the map should be used to mask specific error-prone locations. In our measured data set, no systematic spatial pattern is observable, which verifies our measurement approach such that the influence of nearby switching logic is highly reduced.

Fig. 6 (d) shows the map of the group-normalized standard deviation of the median absolute deviation:

$$\tilde{\sigma}_r(f) = \mathrm{std}_{\forall i}(\mathrm{mad}_{\forall r}(f)) \tag{9}$$

$$\tilde{\sigma}_r^g(f) = \frac{\tilde{\sigma}_r(f)}{\mathrm{mean}_{\forall n|g}(\tilde{\sigma}_r(f))} \tag{10}$$



(a) (b) (c) (d)

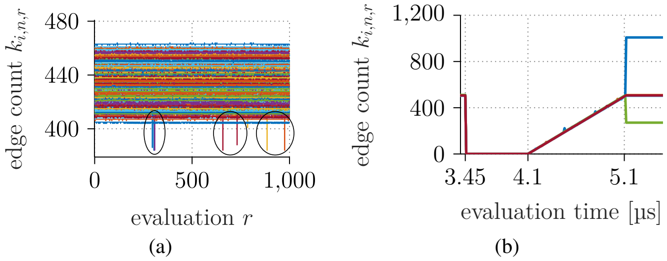
Fig. 6: Heatmap illustration of (a) the standard deviation of the mean frequencies $\sigma_n(f)$, (b) the group-normalized standard deviation of the mean frequencies $\sigma_n^g(f)$, (c) the group-normalized mean of the frequencies median absolute deviation $\tilde{\mu}_r^g(f)$, (d) the group-normalized standard deviation of the frequencies median absolute deviation $\tilde{\sigma}_r^g(f)$

$\tilde{\sigma}_r^g(f)$ indicates the consistency of the bit-error rates between different chips. Although no systematical spatial pattern can be observed, the minimum and maximum values are mostly found in the columns containing ROs of type $\mathrm{r_b}$, which is possibly linked to their higher mean delay. In opposite, the ROs of type $\mathrm{r_t}$ mostly have values close to the average, making them a better choice in some scenarios, e.g. for a post-processing error correction algorithm, which assumes a certain maximum error-rate.

IV. Error Analysis

In this Section, we will analyze our measurement data regarding error rates. This includes a comparison to the state-of-the-art and an investigation on counter failures, which led to a novel ring-oscillator design for RO-PUFs.

A. *Counter Failures with High Frequency Oscillators*



(a) (b)

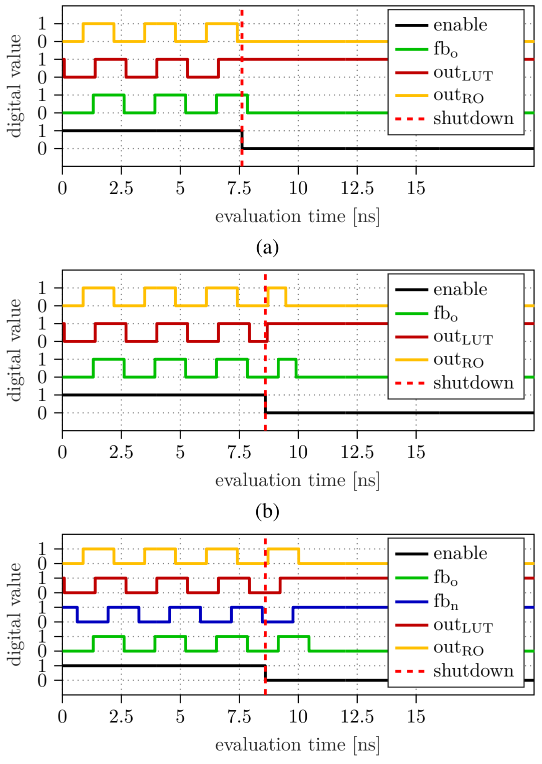
Fig. 7: (a) Measurements of a single chip with few counter failures marked, (b) on-chip sampling of counters for a 1 µs evaluation

Our first design of a state-of-the-art RO consists of an AND-gate as an input gate, which combines the ROs output feedback with an external enable signal, and the usual odd number of delay elements. Analyzing the measurement data for short evaluation times, we observed some very large deviations, similar to [9]. An example of a full chip readout is depicted in Fig. 7 (a), with 3800 ROs measured 1000 times for an evaluation period of 1 µs. Eight out of 3.8 million readout values dropped drastically by around 8%. The mean frequency of the respective ROs are most often identical (e.g. 415 counter value) and many of these false readouts drop to the same value (e.g. 384 counter value); this indicates a very consistent cause. If we examine the binary values for these examples closely, which are $\texttt{110011111}_2$ (415_{10}) and $\texttt{110}\boxed{\texttt{0}}\texttt{00000}_2$ (384_{10}), it seems to be a failed carry-over in a binary addition. This addition can only happen at the very end of the measurement, as the next value would have been an overshooting bit-flip to $\texttt{110}\boxed{\texttt{1}}\texttt{00000}_2$ (416_{10}), which strongly suggests that a timing problem might be the reason.

In order to confirm these theoretical observations, an Integrated-Logic-Analyzer (ILA) was included in the design, which subsampled the counter-values over the course of the active evaluation phase. Though the sampling frequency (100 MHz) was much lower than the actual RO frequencies, the hypothesis could be observed in the results. Fig. 7 (b) shows an exemplary observation of one measurement period (t = 1 µs). At time instance t = 3450 ns, the counters are reset, the ROs are activated at time instance t = 4100 ns and deactivated again at time instance t = 5100 ns. Two counter values jumped to outlying values right at the end, although their curve showed no unexpected behavior throughout the measurement.

The cause of this behavior was found with a post-implementation timing simulations of the RO design. Two simulations are depicted in Fig. 8 (a) and (b), where the evaluation period is stopped at two different time instances by setting the enable signal (black) to `0`. In both plots, the output of the NAND-gate (red) then changes to a constant `1`, which deactivates the oscillation and pulls the output signal of the RO (yellow) to a constant `0`.

However, depending on the current value of the feedback signal (green) at the disabling time, the NAND-gate was either (a) already outputting a `1` or (b) just recently started to output a `0` as the start of a new cycle and is abruptly pulled back to `1`. The last case results in a last glitch pulse with a shorter `1`-period than the previous pulses. It is obvious that this glitch pulse can become even shorter in unfortunate cases and violate hold-time constraints of any following digital logic like the accumulator of the counters and storage registers. Likewise, disabling the counters would also be asynchronous to their input signal, thus this approach could produce a similar glitch.



(a)

(b)

(c)

Fig. 8: Behavior of a RO at shutdown time with (a) no glitch, (b) a glitch, and (c) the glitch avoided by the novel RO design

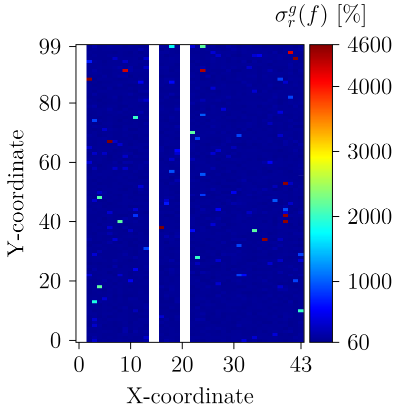


Fig. 9: Heatmap illustration of the standard deviation of the frequency standard deviation

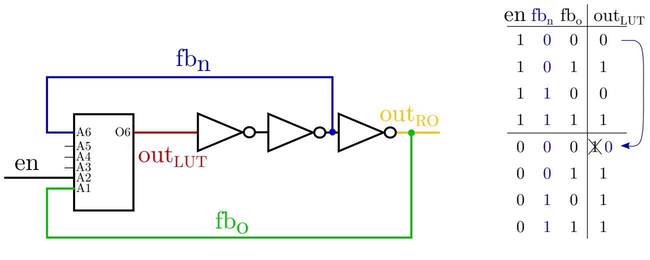
With the results of this investigation in mind, it is easy to explain why the effects of this glitch only affect very few ROs and even those not all of the time. First, as described with the example of the binary values of the counter values 415 and 384, a bit flip has to occur close to the MSB, while bit-flips close to the LSB are overshadowed by the present circuit noise. Consequently, the effect is reduced by a larger evaluation time. Additionally, other ROs with just a slightly higher or lower mean frequency will not be affected as their bit-flips at that time are mostly close to the LSB. Second, whether a hold-time violation really occurs depends on the time instance when the enable signal is deactivated, which also depends on the jitter of the timers. The deactivation not only has to happen in a 0-phase of the NAND-gate but also early enough to shorten the pulse significantly. Based on the simulation results, we calculated that the vulnerable phase of a full pulse cycle is around 50%. The min-period of the DSP48 reported in Vivado is 2.154ns, while our ROs have a mean period of 2.21ns, meaning that a decrease of the pulse width by 5% is still above the min-period. For very short pulses, CMOS based logic shows filtering effects, meaning that a very short pulse width is not long enough to load or unload follow-up CMOS gates. Even though the glitch effect occurs rarely, it is still observable, especially when outliers are not suppressed by the statistical measure, which was the median absolute deviation before. Fig. 9 shows the heatmap of a similar measure to Fig. 6 (d), but this time calculated as the standard deviation of the standard deviation instead:

$$\sigma_r(f) = \mathrm{std}_{\forall i}(\mathrm{std}_{\forall r}(f)) \tag{11}$$

$$\sigma_r^g(f) = \frac{\sigma_r(f)}{\mathrm{mean}_{\forall n|g}(\sigma_r(f))} \tag{12}$$

The positions of the ROs leading to counter failures show up as very large spikes (in red) and they are easily spotted against the blue background. However, while this makes their identification after a full characterization of the FPGA possible, it drastically worsens a qualitive assessment of the remaining ROs.

Our solution to the described problem of counter failures is a novel ring-oscillator design with a second feedback from the second last inverter, shown in Fig. 10 (a) as $\mathrm{fb_n}$ in blue color. The purpose of this path is to provide additional information to the controlling Lookup-Table (LUT6) about the current state of the oscillation cycle and thus the current length of the pulse generated. If the enable signal goes down shortly after a falling edge was produced (leading to a rising edge on the RO output), the output of this LUT6 will be set to a constant `1` only after the new feedback path changes again. Consequently, the content of the LUT was updated as shown in Fig. 10 (b). The upper half is an unchanged NAND-functionality in the active phase of the evaluation, the lower half is the inactivate or shutdown phase. Changes to the original LUT are depicted in blue; a second feedback path now determines whether the current phase already propagated through the inverter chain. A blue arrow shows the transition, which can cause the glitch. Therefore, the row `000` has been adapted to not output the former final shutdown value of `1` (red) but a `0` (blue) to avoid producing a short pulse. The final transitions then propagate through row `000` to row `001`, to row `011` and finally stays in row `010`. Fig. 8 (c) shows the influence of this change. Although the enable signal is deactivated at the same time as in (b), the signal $\mathrm{out_{LUT}}$ does not immediately react to the enable signal but is only set to a constant `1` after the new



en	$\mathrm{fb_n}$	$\mathrm{fb_o}$	$\mathrm{out_{LUT}}$
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1
0	0	0	$\cancel{1}$ 0
0	0	1	1
0	1	0	1
0	1	1	1

(a) (b)

Fig. 10: (a) Proposed novel RO design with additional feedback of an intermediate signal $\mathrm{fb_n}$ and (b) its respective Lookup-Table for the controlling LUT6

feedback path is again `1`, consequently forming the last pulse of the signal $\mathrm{out_{RO}}$ correctly.

B. Statistical Analysis and Reference Normalization

The results of our comparative analysis is shown in Table III as $\sigma_{f,RO}^{norm}$ [8], which is the mean standard deviation of the ROs frequencies. Our first design in the row *old* already outperforms [8] by a factor of almost 3 for 1 µs and a factor of 2.5 for 10 µs. This result confirms the improved measurement method, which includes the DSP48 based counters and separating the ROs from almost any negative influence by partial reconfiguration. For 100 µs, the measurements hit a slightly higher barrier of approximately 0.029%. This effect was to be expected as our boards embed the low cost chip of the Zynq-7000 family, while for [8], a middle class chip of the Artix-7 family was used.

Our RO implementation with the 2nd feedback for glitch-avoidance in the row *new* initially shows an improvement of about 16.4% at 1 µs, which shrinks to 4.9% at 100 µs, where the glitches in the *old* variant expectedly had a minor effect.

As previously mentioned, the *static area* always contained a reference RO to measure possible chip-wide influences like supply voltage drops. The influence of such disturbances are mostly independent of the placement and the type of the RO themselves, as shown in Fig. 11. Although the spatial distance between the *data* RO (solid blue) and the *ref* RO (solid red) is half of the chip size (in average), their deviation from the mean has a very similar trend. This observation holds true even if their RO type is different (dotted lines). Each RO was normalized by multiplying its readout values with the inverse factor that the reference oscillator on the same chip deviated from its mean in the same evaluation period:

$$\tilde{f}_{i,n,r} = f_{i,n,r} \cdot \frac{\mathrm{mean}_{\forall r}(f_{i,r}^{ref})}{f_{i,r}^{ref}} \qquad (13)$$

The respective row *normed* in Table III shows how strong chip-wide disturbances can be: for 1 µs evaluation time, $\sigma_{f,RO}^{norm}$ of the normalized $\tilde{f}_{i,n,r}$ is reduced by 6%. For larger evaluation times, this effect becomes even stronger, reducing $\sigma_{f,RO}^{norm}$ by 49% for 10 µs and still by 27.8% for 100 µs evaluation time, showing that the disturbances are integrated over time and average out very slowly.

TABLE III: $\sigma_{f,RO}^{norm}$ [%] for different designs, readout strategies and evaluation times

	serial			**parallel**		
type	**1µs**	**10µs**	**100µs**	**1µs**	**10µs**	**100µs**
[8]	0.3559	0.1124	0.0213	-	-	-
old	0.1126	0.0455	0.0291	0.1123	0.0455	0.0268
new	0.0942	0.0449	0.0277	0.0944	0.0454	0.0267
normed	0.0885	0.0229	0.0200	0.0882	0.0210	0.0179
carry4	0.1849	0.0609	0.0290	0.1860	0.0630	0.0286

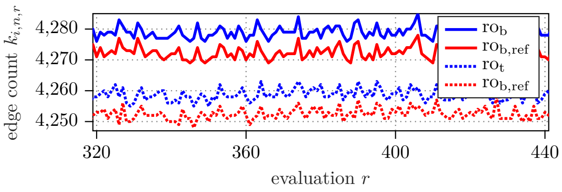


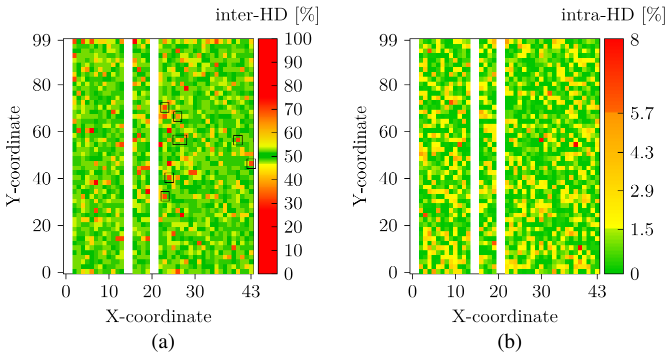
Fig. 11: Exemplary curves of RO measurements (blue) and the affiliated reference curves (red) for two different measurements (solid and dotted)

The last row *carry4* in Table III shows the results of a test design, in which we swapped our DSP48 based counter for a generic Verilog description, resulting in the synthesis of a CARRY4 based counter. As already mentioned in Section II-C, this led to the worst results for 1 µs and 10 µs, however approaches the same limit for 100 µs. We conclude that our DSP48 based counter design is superior for shorter evaluation times, which benefits measurements with a much larger number of readouts.

V. PUF STATISTICS AND PAIRING ANALYSIS

The last missing evaluation for completeness of our analysis is the PUF bit quality assessment. RO frequency analysis is sufficient for first order quantization [16] and the heatmaps of Section III give helpful insight on the chips area quality, allowing for early bit-masking. The impact of second order quantization by taking differential measurements needs to be evaluated by measured data. The most obvious choice for a RO-PUFs is to combine adjacent ROs vertically within the same column, which ensures that they are exactly of the same type and that the influence of on-chip process, voltage and temperature variations is as small as possible.

Fig. 12 (a) shows the inter-hamming distance (inter-HD) map of the same measurement data as in Section III for the mentioned RO pairing scheme. A very strict coloring scheme was applied to show that the average inter-HD is close to the ideal 50%, and to highlight that there are only few positions with a low entropy. These positions are sprinkled over the whole chip with no systematic pattern, with the exception that some of these correlate with the outstanding delay variations observed in Fig. 5 (b) from the Vivado extracted data. For comparison, some of these positions were also marked with



(a)

(b)

Fig. 12: PUF bit heatmaps of (a) the inter-HD distribution and (b) the intra-HD distribution

TABLE IV: RO-PUF statistics with vertical RO pairing

		evaluation	**intra-HD [%]**		**inter-HD [%]**	
data basis		**time [µs]**	**mean**	**std**	**mean**	**std**
raw	serial	1	5.1	2.4	48.9	2.8
		10	2.6	1.8	49.2	1.4
		100	1.6	1.4	49.2	1.3
	parallel	1	5.0	2.5	48.8	2.9
		10	2.5	1.7	49.2	1.5
		100	1.5	1.4	49.2	1.3
normed	serial	1	5.3	2.4	49.1	1.5
		10	1.3	1.3	49.2	1.3
		100	1.1	1.2	49.2	1.3
	parallel	1	5.0	2.4	49.0	1.6
		10	1.1	1.2	49.1	1.4
		100	1.0	1.2	49.2	1.3

black boxes in Fig. 12 (a). This further emphasizes our conclusion that the exposure of the estimated delay increases the predictability of RO-PUFs at some positions in advance. In addition, the most top row of the measured frequencies in Fig. 5 (b) shows an abrupt frequency drop compared to the row below, which again can be seen in Fig. 12 (a) as a largely biased row. One has to keep in mind that some of the outliers could also falsely appear biased due to the limited sample size of 20 chips.

Fig. 12 (b) shows the intra-HD distribution as the representation of the mean bit-error map of all devices. The coloring scheme reflects the statistical moments mean (green), below (yellow to orange) and above (red) 3-sigma. Again, no systematic pattern can be detected, which confirms the findings in Section III in Fig. 5 (a) and (b), that the expected bit-error rate will be randomly distributed. We conclude that for RO-PUF generation, the bit-error rates are mostly device and implementation specific. Bit-masks have to be extracted separately for each device, which can be done quickly with our presented framework with low measurement time per bit.

Table IV lists the statistical results as raw measurements and normalized to the reference RO, as mentioned in Section IV. Regarding the evaluation times, two observations can be concluded. First, there is no significant difference between the parallel and serial evaluation. This contradicts previous publications [10], yet can easily be explained by the approach of our framework. Although a parallel readout enables all ROs at the same time, they are spatially separated by a security margin of at least one CLB (with our TCL settings). Along with the fact that there is no other nearby switching logic, this diminishes the effect of crosstalk and other disturbances on a single RO. Second, while the inter-HD might theoretically improve with each order of magnitude of the evaluation time, it already reaches its maximum at $t = 10\,\mu s$. This means that, at the cost of a slightly increased bit-error rate, the evaluation time can be greatly decreased in comparison to previous work.

Regarding the normalized statistics, no improvement can be seen for the evaluation time of 1 µs, which is not surprising as these readout values are very small. For RO-PUFs a useful trade-off between evaluation time and precision no longer exists, when the evaluation time goes below a certain limit. Thus, for the used FPGA generation and our implementation, this lower limit is somewhere between $t = 1\,\mu s$ and $t = 10\,\mu s$.

For larger evaluation times, the reduction of the intra-HD becomes significant: at least 50% improvement compared to the *raw* results at 10 µs and still 31% compared to the *raw* results at 100 µs. The reason for such an improvement with normalization, and thus its necessity, becomes clear by looking again at the placement grid in Fig. 1 (b). In order to reduce the crosstalk between ROs, we separate their spatial placement, however when creating PUFs from their readouts, we combine adjacent ROs. This means that the RO instances are evaluated at different times, e.g. A and B in Fig. 1 (b), and therefore are affected by different on-chip disturbances like supply voltage drop. This influence of temporal separation is removed by normalizing each value to its respective reference value, and the only remaining deviation on the readout is inherent circuit noise of the RO instance itself. Thus the *normed* results represent a close approximation of the PUF statistics in a case, where all ROs are evaluated at the same time.

Additionally, we re-investigated the state-of-the-art approach that pairs of ROs should only be created within the same column, which ensures a restriction to the very same type (same CLB type and orientation, same SLICE and RO type). We also tested pairs created in two other ways. In the first ($horizontal_A$), a RO is paired with its next horizontal neighbor of the very same SLICE and RO type, but ignoring the equality of the CLB type and orientation. In the second ($horizontal_B$), the pairing was also horizontal but with real equality of the group definitions, which equals to the restrictions of the vertical pairing. Table V compares the results of this different pairings with the results from the original pairing. The intra-HD for both horizontal pairings were slightly lower than the vertical ones, for which we have found no explanation. However, the inter-HD binds a stronger restriction to the pairing selection and as can be seen, both the mean and the standard deviation are worse than for the vertical pairing.

To explain this effect, the last two rows additionally shows the minimum and mean distance of the pairing coordinates. Both the horizontal pairings have a larger minimum and a

TABLE V: Statistics of different RO-PUF pairings based on normalized measurements at $\mathrm{t} = 100\,\mu\mathrm{s}$ with serial readout

	intra-HD [%]		**inter-HD [%]**		**distance**	
pairing	**mean**	**std**	**mean**	**std**	**min**	**mean**
vertical	1.6	1.4	49.2	1.3	1	1.0
horizontal$_{\mathrm{A}}$	1.4	1.4	48.0	2.2	2	3.3
horizontal$_{\mathrm{B}}$	1.3	1.3	46.7	4.3	2	7.3

much larger mean distance than the vertical pairing, which results from the non-uniform column distribution shown in Fig. 1 (a). As shown in Fig. 4 (b), there exist strong variations in the ROs mean frequencies for such greater spatial distances. Obviously, these variations manifest as a strong bias, which reduces the inter-HD of both alternative pairings. However, the drop of inter-HD for the pairing *horizontal*$_A$ is less significant than that of *horizontal*$_B$, although SLICEs with different CLB types were paired. Complementing the state-of-the-art, we conclude that the influence of different CLB types on the inter-HD is observable, yet marginal compared to the much stronger influence of the spatial chip variations, as long as the SLICE types are kept consistent.

VI. CONCLUSION

In this work, we present an in-depth analysis of measurement data extracted from Xilinx Zynq-7000 FPGAs with the aid of a framework based on partial reconfiguration. We achieved a coverage of 100% of the targeted chip area by splitting the toplevel in half and measure each side separately. By repeatedly reprogramming the partial area, we extracted the statistical properties of each possible placement while maintaining a security margin between the instances, avoiding unnecessary nearby switching logic.

With the knowledge of the chip structural layout, we could reveal structural bias due to on-chip variations independent from the ring-oscillator type and underlying SLICE properties. Together with the estimated delays from the Vivado toolchain, we could further reveal areas with a potentially larger bias, which have to be avoided for PUF placement.

We found a particular glitching problem for ring-oscillators at short evaluation times, which leads to counter failures and further distorts the readouts. With a novel addition to ring-oscillators in the form of a second feedback path to suppress glitches combined with an enhanced DSP48 based counter, we could reduce the normalized frequency standard deviation to 0.0449% at $\mathrm{t} = 10\,\mu\mathrm{s}$, which is about a factor of two smaller compared to the state-of-the-art. By further normalizing the readouts to a known static reference RO, this measure was further reduced to 0.0229%. Both the normalized frequency standard deviation and the minimum evaluation time significantly outperform previous work and prove the efficiency of our combined efforts.

We additionally tested various RO pairings, which ignored the recommended vertical pairing and type equality. Although the inter-HD is visibly effected, the effect is not as strong as expected and overshadowed by the influence of the chip variations.

VII. AVAILABILITY

We will make the framework and the measurement data used in this work accessible on our institute homepage: *https://www.uni-ulm.de/en/in/institute-of-microelectronics/research/topics/mixed-signal-cmos-circuits/physical-unclonable-functions*

ACKNOWLEDGMENT

This work was funded by the German National Science Foundation DFG under Grant Number FI 982/15-1.

REFERENCES

[1] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, “Silicon physical random functions,” in *Proceedings of the 9th ACM Conference on Computer and Communications Security*, ser. CCS ’02. New York, NY, USA: ACM, 2002, pp. 148–160. [Online]. Available: http://doi.acm.org/10.1145/586110.586132

[2] S. Morozov, A. Maiti, and P. Schaumont, “An analysis of delay based puf implementations on fpga,” in *Proceedings of the 6th International Conference on Reconfigurable Computing: Architectures, Tools and Applications*, ser. ARC’10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 382–387.

[3] S. Gehrer and G. Sigl, “Reconfigurable pufs for fpga-based socs,” in *2014 International Symposium on Integrated Circuits (ISIC)*, Dec 2014, pp. 140–143.

[4] S. Morozov, A. Maiti, and P. Schaumont, “An analysis of delay based puf implementations on fpga,” in *Reconfigurable Computing: Architectures, Tools and Applications*, P. Sirisuk, F. Morgan, T. El-Ghazawi, and H. Amano, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 382–387.

[5] A. Wild, G. T. Becker, and T. Güneysu, “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.

[6] S. Gehrer and G. Sigl, “Using the reconfigurability of modern fpgas for highly efficient puf-based key generation,” in *2015 10th International Symposium on Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC)*, June 2015, pp. 1–6.

[7] L. Feiten, J. Oesterle, T. Martin, M. Sauer, and B. Becker, “Systemic frequency biases in ring oscillator pufs on fpgas,” *IEEE Transactions on Multi-Scale Computing Systems*, vol. 2, no. 3, pp. 174–185, July 2016.

[8] 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 *2018 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, April 2018, pp. 126–133.

[9] A. Wild, G. T. Becker, and T. Güneysu, “On the problems of realizing reliable and efficient ring oscillator pufs on fpgas,” in *2016 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, May 2016, pp. 103–108.

[10] D. Merli, F. Stumpf, and C. Eckert, “Improving the quality of ring oscillator pufs on fpgas,” in *Proceedings of the 5th Workshop on Embedded Systems Security*, ser. WESS ’10. New York, NY, USA: ACM, 2010, pp. 9:1–9:9. [Online]. Available: http://doi.acm.org/10.1145/1873548.1873557

[11] A. Maiti, J. Casarona, L. McHale, and P. Schaumont, “A large scale characterization of ro-puf,” in *2010 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, June 2010, pp. 94–99.

[12] *Zynq-7000 SoC DC and AC Switching Characteristics - DS187*, Xilinx, 7 2018, v1.20.1.

[13] *Zynq-7000 SoC Data Sheet: Overview - DS190*, Xilinx, 8 2018, v1.11.1.

[14] *7 Series DSP48E1 Slice User Guide - UG479*, Xilinx, 3 2018, v1.19.

[15] R. Maes, A. Van Herrewege, and I. Verbauwhede, “Pufky: A fully functional puf-based cryptographic key generator,” in *Proceedings of the 14th International Conference on Cryptographic Hardware and Embedded Systems*, ser. CHES’12. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 302–319. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-33027-8_18

[16] V. Immler, M. Hiller, J. Obermaier, and G. Sigl, “Take a moment and have some t: Hypothesis testing on raw puf data,” in *2017 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, May 2017, pp. 128–129.