Compact FPGA Ring Oscillator Physical Unclonable Functions Circuits Based on Intertwined Programmable Delay Paths

Yangpingqing Hu, Yuqiu Jiang, and Weizhong Wang

Abstract—The Physical Unclonable Functions (PUFs) provide a strong and secure root source for identification and authentication applications. It is especially valuable for FPGA-based systems, as FPGA designs are vulnerable to IP thefts and cloning. Ideally, the randomness of PUFs should come from the random variation in the manufacturing process. It should be free of deterministic variation coming from systematic bias among all chips of the same model. Correspondingly, one of the major challenges for FPGA-based PUFs is the difficulty of avoiding systematic bias between the nominally matched delays in the competing paths. In this paper, a deep investigation into the LUT structure on Xilinx FPGA is conducted. Based on the investigation, a compact PUF circuit based on programmable Look-Up-Tables (LUTs) paths is reported. The proposed intertwined structure and the novel 2-phase 2-pass operation significantly reduced the systematic bias in Xilinx FPGA LUT. The proposed PUFs circuits are based on random variations between the same delay paths implemented in the adjacent LUT cells, thus showing very strong uniformity and uniqueness.

Index Terms—Field Programmable Gate Array, Physical Unclonable Features, Identification and Authentications, Secure root sources

I. INTRODUCTION

One of the key requirements for securing communications through public open networks is the ability to authenticate the recipients. In order to block malicious network elements, a network node must validate the identity of the recipient. Besides, IP (Intellectual Property) vendors require anti-counterfeiting methods to prevent loss in revenue due to the counterfeits of their IP creations. One of such methods is to use unique hardware-dependent keys provided by physical unclonable functions (PUF) [1, 2]. The basic idea is to exploit non-reproducible manufacturing variations to provide a device-specific signature. However, such manufacturing variations are effectively impossible to be predicted or replicated. Therefore, a PUF that can be implemented using general-purpose, reconfigurable hardware is extremely attractive.

The most fundamental challenge for all PUFs is that they must exhibit extreme sensitivity to manufacturing variations, yet it must be deterministic to provide a consistent response over the life of the operation. Therefore, the ideal PUF

structures should be free of systematic bias so that it is able to maximize the entropy due to manufacturing variations among different chips. In the meantime, consistency in the responses under various operation conditions is desired, which mandates that the ideal PUF design should have a mechanism to identify and screen out the challenge-response pairs (CRPs), whose responses fall within the noise margin from the 0/1 output threshold.

G. E. Suh and S. Devadas proposed RO (Ring Oscillator) PUF in [10]. In general, RO PUF is easy to be implemented on both ASIC and FPGA. However, FPGA RO PUF architectures often occupy ample fabric resources. Aiming to improve FPGA RO PUFs' area efficiency, a RO PUF architecture based on programmable LUT delays was proposed [2]. The basic idea is to explore the delay differences among various delay paths inside LUT cells. As the delay paths inside LUT cells can be programmed by LUT inputs, the delay of a single LUT may have many variations depending on the LUT inputs. For a 4-input LUT cell, there are 16 programmable delay paths available. However, since the RO delay consists of both delay inside LUTs and the interconnect delay between LUTs, this approach requires all the interconnects between adjacent LUTs to be better matched than the programmable delay paths inside LUTs. Such requirements set the limit of how many LUTs can be cascaded. So far, there has not been published experimental work on programmable LUT delay PUFs with more than 4 LUTs cascaded. Therefore, the potential benefit of programmable LUT delay paths based on FPGA RO PUF architecture has not been fully revealed.

As more LUTs are cascaded, the number of unique programmable delay paths grows exponentially. In [18], when eight 3-input LUTs are cascaded, the total number of programmable delay paths grows to $(2^3)^8=16,777,216$. The idea is to run FPGA RO PUF twice sequentially with different programmable delay paths inside LUT but the same interconnects between LUTs. In this way, the impacts from interconnects between LUTs are eliminated, as both runs use the same set of interconnects. The PUF whose response bit is determined based on two runs is referred to as 2-pass PUF. Based on Altera/Intel Cyclone IV, an 8 LUTs cascaded RO PUF architecture was proposed to overcome the aforementioned difficulties. But the design was not implemented in real devices [18]. There are systematic biases among programmable delay paths inside LUT. To avoid the impact of such systematic bias, the careful challenge selection was discussed in [18]. But it concluded that such extra selection operations introduce too

much burden in real operation.

This work experimentally investigates the biases in LUT structure in Xilinx Artix-7 FPGA. Major bias between two LUT5s and biases within LUT5 are presented. Furthermore, a modified 2-pass PUF is proposed and experimentally demonstrated. The measurement results demonstrated that the structure and operation could mitigate the two types of biases mentioned above.

Various attack approaches on PUF are discussed by recent literature, including side-channel attacks [22][28], and modeling attacks [20]. Especially, there is an increasing interest in the attack using machine learning (ML). Some methods have been proposed against ML attacks. These methods can be roughly classified into structural non-linearization and CRP obfuscation [27]. This paper will focus on the designs to mitigate systematic bias in FPGA and corresponding PUF performance of the improved PUF circuit designs. The study on the resilience to machine learning attacks is ongoing and the results will be presented in our future works. The contributions of this paper are as follows:

- For the first time, systematic biases among programmable delay paths inside Xilinx Artix-7 LUTs are thoroughly investigated. Experimental results show that there is major systematic bias between two LUT5s in each LUT6. Besides, systematic bias among different programmable delay paths is also found within the LUT5. The systematic biases in Xilinx LUTs are different and more complex than the ones in Intel LUTs that were briefly discussed in previous literature [18].
- A novel intertwined programmable delay (IPD) structure is proposed. It is inspired by the theoretical loop PUF delay model presented in [21]. In that model, the challenge selects one of the two intertwined paths consists of fixed delay elements. This work proposed IPD in which the challenge configures the programmable delay paths inside LUTs. A dedicated control bit manages the use of the two intertwined programmable delays. The abundant circuit resources inside LUT can be fully used. RO based on IPD is implemented on Xilinx Artix-7 FPGA. The experimental results indicate that the major systematic bias between LUT5s is eliminated.
- A modified 2-pass operation is proposed and experimentally demonstrated to mitigate the biases in the structure of Xilinx LUT6. The new 2-pass operation consists of 2-phase in each pass. IPD-RO configured by the same challenge in the two intertwined loops is sampled in the two phases, which mitigates the deterministic bias in the LUT5 layout. Compared with the one in [18], our design does not need any CRP pairing. In meanwhile, the measured correlation is still lower.

The rest of this paper is constructed as follows. Section II gives the background information on programmable LUT delay RO PUF architecture. Section III focuses on the experimental investigation and demonstration of systematic bias among programmable LUT delay paths in LUT6 of Xilinx Artix-7 FPGA. Section IV presents the implementation of the proposed IPD-RO structure on Xilinx 7 series FPGA. Section V shows the experimental results along with a comprehensive analysis

of the measurement data. Finally, it is concluded in Section VI.

II. PRELIMINARIES

A. Programmable LUT delay-based FPGA RO PUF

Traditional RO PUFs are based on the “symmetrical” paths formed by identically designed inverters and interconnect wires. The difference caused by manufacture variation in these paths is the source of randomness [2][10]. In FPGA implementations, such “symmetrical” paths are instantiated by design software, which are often opaque to circuit designers [29]. In addition, FPGA chip designers and manufacturers usually do not focus on the matching among the symmetrical interconnect wires. Therefore, those “symmetrical” paths often carry systematic bias, i.e., specific paths are always faster than others. Such systematic bias is due to (a) predictable wire delay differences; (b) systematic differences among driving transistors strength. The systematic bias reduces the randomness or entropy of the traditional RO-based PUFs designs.

Fig. 1. FPGA LUT cell programmed into 8 inverters selectable by using input pins.

FPGAs’ LUT cells can be programmed into many inverters and associated delay paths. Such configuration can be achieved by using the input bits of each LUT. For an m -input LUT, there are 2^{m-1} delay paths. These paths are referred to as PDL (programmable delay line) [29]. As shown in Fig. 1, pin A, the LSB of inputs is used as the inverter’s input. The other three inputs can program the 4-input LUT into 8 PDLs. Such abundant circuit resources inside LUT cells have been explored to design PUFs with significantly improved area efficiency [2][15][30]. These types of designs have been demonstrated on various FPGA platforms. Due to the systematic bias, these designs may show poor uniformity or uniqueness. An example of the bias in LUT is that some specific wires are systematically longer than the other ones that are controlled by the same input. As shown in Fig. 1, input D makes a selection between the red and blue wires. However, the delay in the red wire is systematically larger than the blue wire, which causes a systematic bias. To mitigate the impact of bias, a few methods were proposed. One approach is to use multiple PDLs to determine only 1-bit response [2]. Another approach is to use the average RO frequency of ROs at the same locations of all devices to adjust the measured frequency from each RO so the

comparison of two RO frequencies would be fair [4].

B. 2-pass FPGA RO PUF to eliminate impacts from the interconnect between two adjacent LUTs

In an attempt to eliminate the impact from interconnect mismatch between two ROs, PDL (programmable delay line)-RO-PUF, a 2-pass RO PUF architecture was proposed but not implemented in [18]. Unlike traditional RO-based PUF, competing paths in this design are different PDLs residing in the same LUT chains. These paths were called VRO (virtual RO) in [18]. VROs share the same interconnects so that it is free of the biases due to mismatched interconnects. The structure of PDL-RO-PUF includes multiple cascaded LUTs that are configured as inverters. For an m -input LUT, there are 2^{m-1} PDLs. When N stages of m -input LUTs are cascaded to form RO, the number of PDLs in this RO is $2^{N*(m-1)}$. Two VROs should be sampled at two different times because only one can be activated at a time. Two different configurable bits config the paths inside LUTs to form two different VROs. The competition between a pair of VRO determines 1-bit PUF response.

1) Systematic bias among programmable LUT delay paths

It has been demonstrated that there are systematical biases among programmable LUT delay paths in Intel FPGAs [18]. To achieve the best result, it was claimed that VRO pairs with equal nominal delays should be selected. However, by doing this, a large number of CRPs cannot be used.

So far, there is no published investigation about systematic bias among programmable delay paths inside Xilinx LUTs. The LUT structures in Intel FPGA and Xilinx FPGA are quite different [16][19]. In Section III, an experimental investigation will demonstrate that systemic bias similar to the one reported in Intel FPGA exists in Xilinx FPGA as well. Furthermore, some systemic bias pattern never reported before is found in Xilinx FPGA.

With such systematic bias confirmed in both Intel and Xilinx FPGAs, the PUF delays corresponding to certain challenges can be guessed based on the measurement results obtained from another chip of the same model. Consequently, only a sub-set of the challenges can be used to maintain secure PUF operation. Such limitation prevented the first attempt on 2-pass FPGA RO PUF architecture from being implemented [18]. To overcome such difficulty, an improved FPGA RO PUF design will be presented in Section IV.

2) Variation between 2 passes

For 2-pass PUF, the potential system clock variation between 2 passes becomes a new concern because the system clock is not steady all the time. It may impact the reliability of the PUF when two passes have two different running times. Besides, voltages and temperatures are two main factors impacting the frequency of RO [32][33]. To mitigate variation caused by these factors, a reference RO is introduced. To distinguish the two ROs, the RO that gives the PUF response is referred to as target RO in the remainder of this paper. During operation, the reference RO runs simultaneously with the target ROs, so that the timing variation, voltage, and temperature variation are common between the two ROs. The counter reading from the targeted ROs will be calibrated by using the counter reading from the reference ROs. The impacts from system clock,

voltage and temperature variations are removed in the calibrated targeted RO oscillation counts.

III. EXPERIMENTAL INVESTIGATION ON THE SYSTEMATIC BIASES ON PROGRAMMABLE LUT DELAYS IN XILINX FPGAS

Fig. 2. Investigation RO Structure, based on single Xilinx LUT6 4-bit RO.

According to the Xilinx 7 series FPGA datasheet, each LUT6 consists of two physical LUT5 [16]. This LUT6 structure is shown in Fig. 2. Input I5 of the LUT6 selects the LUT5 that in effect renders the output of LUT6. As Xilinx's datasheet does not provide any details in LUT6, the knowledge about LUT6 layout is absent. It was claimed that LUT6 in Xilinx Virtex 5 has the smallest delay when I1~I5 are 00₁₆ and the biggest delay when 1f₁₆ [29]. However, there is no published experimental data for detail. Thus, the biases between the two physical LUT5 are investigated experimentally in this paper.

ROs oscillation counts are used to measure the delay of the programmable LUT delay paths. The structure of the experimented RO is shown in Fig. 2. Each RO consists of an AND gate and a LUT6. Input I0 of the LUT6 is connected to the output of the AND gate so the loop is formed. Input I1~I5 are fed with 5-bit challenges which pick one of the 32 possible LUT delay paths. The MSB of the challenge is used to determine which LUT5 is used in the specific run.

The CLB (configurable logic block) on Artix-7 has two SLICES. Each SLICE has 4 LUT6s. Therefore, there are $2*4=8$ LUT6s in each CLB. On each device, 96 CLBs are tested. These CLBs are on different parts of the FPGA. In total, $96*8=768$ LUT6s are tested on each device.

A. Systematic bias between two LUT5 in a LUT6

Fig. 3(a) shows the result of one sample RO. The 5-bit challenges ranged from 00₁₆ to 1f₁₆ are all covered. 32 unique challenges correspond to 32 PDLs in LUT6. The number of oscillations that happen in each path is presented in Fig. 3(a). Due to operation condition variation, each cluster has some variations. Different LUT6s are subject to different manufacturing variations. One LUT6 may have a very different situation from another one. It should be noted that the blue clusters and the red clusters are the results from two LUT5, respectively. It is obvious that all points in the blue clusters are higher than their counterparts in the red ones. In fact, for all the $768*2(\text{chip})=1536$ tested LUT6s, the blue clusters are always

higher than the red ones. Fig. 5(b) shows the results from 40 LUT6s that are randomly picked from 1536 LUT6s. It is clearly shown that systematic bias between corresponding programmable delay paths located in two LUT5 within the same LUT6 exists in Xilinx 7 series FPGAs. In the 1536 tested LUT6s, the percentage of the systematic bias in the number of oscillations is a normal variable $N(0.3\%, 0.096\%^2)$. This systematic bias is similar to the one reported in Cyclone IV, which has a big bias between the two LUT3 in a LUT4 [18][19].

Fig. 3. (a) Number of oscillation cycles distribution for single LUT6 with 5-bit challenges (b) Number of oscillation cycles of sample ROs picked from two devices. Index 1~20 from device 1 and 21~40 from device 2. For all ROs at different locations and different devices, the blue clusters are always higher than the red ones.

B. Systematic bias within LUT5

After observing the existence of the bias between the two LUT5 within the same LUT6, the next question is whether there are systematic biases among the programmable delay paths within each LUT5. If not, the PUF can be designed based on two competing programmable paths within the LUT5 cell. Otherwise, a new design is needed to mitigate the systematic bias among all programmable delay paths within the LUT. The delays of the programmable delay paths are still measured by using ROs. To enhance the signal-to-noise ratio in the measurements, ROs consist of an AND gate and 8 LUTs cascaded together. All 8 LUTs are programmed by the same 5 challenge bits so that identical delay paths are chosen for each LUT. Fig. 4 shows the oscillation counts corresponding to the $2^5=32$ different challenge patterns. While the biases between the top and bottom LUT5 are still noticeable, the pattern within the group of blue clusters and the group of red clusters is clearly

visible. It is obvious that there is a repeating pattern for every 4 clusters. It appears that the 2 least significant bits of challenge determine the pattern within the cluster of 4. Although there is some variation in each cluster, the intrinsic systematic bias is not negligible. To mitigate the systematic bias in the LUT5, a modified 2-pass operation is proposed in Section IV.

Fig. 4. Boxplot of 32 instances of 32-bit RO configured by 4-bit repeated challenge. Biases between each instance of RO are neglected here.

C. PUF performance impacted by the bias inside LUT6.

Fig. 5 Structure of investigation PUF. N is an even number.

Building PUF without eliminating the bias would result in a loss of uniqueness. The systematic biases become the major contributor to determine the PUF output bit. All ROs at different parts of the FPGA chip tend to give the same PUF response responding to the same challenge. And all FPGA chips would give the same PUF responses. To quantify the impacts, an investigation was done on a Xilinx FPGA with ROs configured in the way mentioned in [18]. Fig. 5 shows the structure of the investigation PUF. N LUT6s plus a NAND gate are used in each RO, and each LUT6 takes 5 challenge bits as configurable bits. In total, $5N$ challenge bits configure the RO. For the demonstration in this section, $N=8$.

The sequential sampling method presented in [18] is used to generate the 1-bit PUF response. Two measurements are performed on the RO configured by a pair of randomly chosen challenges. Two measurements are compared to produce 1-bit outcome as the PUF response. In the experiments, 4000 PUF responses were collected from each of 32 ROs. [18] discussed how the competing pairs should be selected to achieve the best results. Their approach filtered CRPs by delay threshold and disparity threshold, which would discard many challenges. To understand how biases affect the PUF, all challenges are kept for analysis in this investigation.

Bit-aliasing is found to be a straightforward metric demonstrating biases' effects. First, bit-aliasing is analyzed

across different devices [7]. PUF response collected from the RO at location p on device i with challenge l is denoted as $r_{i,l,p}$. With k devices, the bit-aliasing is defined as,

$$\text{Bit-aliasing}_{l,p} = \frac{1}{k} \sum_{i=1}^k r_{i,l,p} * 100\% \quad (1)$$

On 2 devices, 32 different locations are randomly selected to implement RO. 2000 challenge pairs are tested, and the corresponding PUF responses are analyzed. The result in Fig. 6 shows that the two tested devices have very poor bit-aliasing. For only two devices, the outcome of $\text{Bit-aliasing}_{l,p}$ can be 0%, 50%, and 100% (x-axis in Fig. 6). Ideally, the percentage for each case should be 25%, 50%, and 25%, respectively. However, as shown in Fig. 6, the numbers from experiments are about 36%, 25.5%, and 38.5%.

Fig. 6 Bit-aliasing for investigation PUF

The PDL-RO-PUF has the advantage that a single RO could be the entropy source for PUF. Whether there is a correlation between ROs in the same device attracted our interest. The correlation within a single device has attracted growing interest as shown in recent literature. [24] proposed a method to measure the correlation in the PUF responses. Spatial autocorrelation analysis was proposed in [25], which considers the proximity of PUF circuits on a single device. These methods were proposed to analyze the correlation that cannot be detected by the canonical PUF evaluation methods [7]. At this step, bit-aliasing is computed intra-device, whose result already indicates the correlation issue in the RO PUFs in a single device [25]. $r_{i,p,l}$ is the PUF response from RO at location p in device i . P is the number of tested RO locations in the same device. The intra-device bit-aliasing is defined as,

$$(\text{Bit-aliasing}_{\text{intra}})_{i,l} = \frac{1}{P} \sum_{p=1}^P r_{i,p,l} \quad (2)$$

As shown in Fig. 7, there is a large probability that all the 32 ROs in the same device give the same PUF responses, i.e., either output 1 or 0 from all 32 ROs located at different parts of the chip. The results on both devices are very similar. Ideally, different ROs are expected to give different PUF responses to the same challenge. In that case, the peak of the distribution should be at the center. A curve of the distribution from 32 simulated ideal ROs is plotted in Fig. 7. Based on the simulated data from the random number generator, the ideal ROs have independent random responses. Unlike the ideal distribution, the probability of having split between 1 and 0 evenly among measurement results from 32 ROs is around 2%. In comparison, 10.35% and 7.81% challenges would yield identical PUF output

bits in all 32 ROs. And the probability that over 90% ROs (that is at least 28 out of 32 ROs) have the same PUF responses to the same challenges is 20.90% and 21.48% for 0's and 1's respectively. However, as shown by the ideal distribution, in truly random PUFs, the probability that 90% of ROs give the same PUF output bits should be very small. Systematic biases are common for all LUTs, and the amount of the bias is quite significant. Such systematic bias shadows the disparity due to manufacturing randomness, leaving strong bit-aliasing. For comparison, in Section V, the result from the proposed IPD-RO-PUF will be presented. Besides, the uniqueness of such investigation PUF is presented in Table I.

Fig. 7. Intra-device bit-aliasing distribution. 100% and 0% mean all 32 ROs in the same device yield the same PUF response, while 50% means PUF responses to the same challenge are random on these ROs.

IV. IMPLEMENTATION OF IPD-RO-PUF

A. From single LUT stage to intertwined LUT stage

To study LPUF (loop-PUF), a delay model was used but not implemented in [21]. Controlled by the 1-bit challenge, each stage of the delay model incorporates two twisted paths. Each path is a combination of a top delay and a bottom delay. For each stage, the 1-bit challenge selects one from two paths in the 1st pass. The challenge flips in the 2nd pass so the other path is selected. PUF response is determined based on the competition of the two passes. LPUF was implemented on Xilinx Artix-7 with LUT6 in [22]. However, each delay stage used in [22] is based on a single LUT stage. In each delay stage, only one input of LUT6 was controlled by the challenge to select one of the only two delay paths. It was not clarified how the other inputs are used.

Fig. 8. (a) Single LUT stage (b) Intertwined LUT stage in adjacent LUT cells. The top inverters are systematically faster than the bottom ones.

Fig. 8 shows the single LUT based stage (left) and the

intertwined LUT stage (right). The single LUT stage is based on only one LUT and the intertwined LUT stage is based on two LUTs. The LUT mentioned here is the structure accessible on a CLB. Different vendors have different LUT designs on a CLB. In Xilinx Artix-7, a single LUT stage is based on LUT6, and the two inverters are two LUT5s functioning as inverters (as shown in Fig. 2). From the investigation experiments results, we could derive the delay model for LUT6 in Xilinx Artix-7. D^0 is the nominal delay of the inverter (bottom inverter in Fig. 8) when the LUT6 input I5 is 0, and D^1 is the nominal delay when I5 is 1. From experimental data, it was found D^1 is from a fast inverter (smaller delay, larger number of oscillations). In this paper RO frequency (number of RO oscillations within a certain period) is used to represent the delay of the chosen path. The difference in the delay is measured as the difference in the frequency. We have the major systematic delay difference between the two LUT5s as,

$$D^\Delta = D^0 - D^1 \quad (3)$$

In the single LUT stage, the challenges are used to select whether the top or the bottom inverter is used at a time. Challenges flip in two passes in [22]. The nominal difference between the two passes is either D^Δ or $-D^\Delta$ depending on whether D^0 or D^1 is used first. When such LUT stages are cascaded to build a RO, whether more top inverters or bottom inverters are selected becomes the dominant factor. In this case, the selection of challenges becomes a critical issue. It was concluded that to maximize the entropy from LPUF, the challenges should be selected based on Hadamard codewords, except the all-zero and all-one cases [21]. In this scenario, at a time, the same number of top inverters and bottom inverters are used. However, the number of usable challenges is limited, making the entropy deteriorated. Furthermore, although matching the fast and slow delay does nominally mitigate the bias, there is still some remaining bias in an implementation, which leads to degraded uniformity. The investigation PUF experimented in Section III.C and the PDL-RO-PUF in [18] are also based on a single LUT stage. However, different from [22], they use two different challenges two passes. Therefore, due to the major bias between LUT5s, the outcome is severely affected. Besides, the competing elements are also affected by the bias within LUT.

For one of the 16 paths in a LUT5, the delay is composed of its LUT5 nominal delay D_{I5} and its deviation from the LUT5 nominal delay $\delta_{I1\sim4}$. In Fig. 3(b), it is shown that the delay of a path is heavily dependent on the LUT5 nominal delay. Being controlled by LUT6 inputs I1~4, $\delta_{I0\sim4}$ has smaller impact on the delay of the path d_i . Therefore,

$$d_{I1\sim5} = D_{I5} + \delta_{I1\sim4} \quad (D_{I5} \gg \delta_{I1\sim4}) \quad (4)$$

where I5 is the MSB of the 5-bit configurable input and I1~4 is the other 4-bits. When two paths are used in two passes, the difference in delay is,

$$\begin{aligned} d_{I1\sim5} - d_{I1\sim5'} &= (D_{I5} + \delta_{I1\sim4}) - (D_{I5'} + \delta_{I1\sim4'}) \\ &= (D_{I5} - D_{I5'}) - (\delta_{I1\sim4} + \delta_{I1\sim4'}) \end{aligned} \quad (5)$$

If the bias between D_{I5} and $D_{I5'}$ is not considered, the bias between $\delta_{I1\sim4}$ and $\delta_{I1\sim4'}$ determines the outcome of $d_{I1\sim5} - d_{I1\sim5'}$. For example, in Fig. 4, the cluster for challenge index 1 ($\delta_{I1\sim4=1}$) is always higher than the one for challenge index 4 ($\delta_{I1\sim4=4}$). Therefore, on any RO, the PUF response based on

these two challenges is deterministic.

In the intertwined LUT structure (Fig. 8(b)), the top inverter in the first LUT and the bottom inverter in the second LUT are connected as a delay path (blue). The other delay path is based on the other pair of top and bottom inverters. Both paths include a fast and a slow inverter. The challenge is also used to select which path is used in the RO [21]. Without considering $\delta_{I1\sim4}$, the nominal delays for the two passes are, $D_0^a + D_1^b$ and $D_1^a + D_0^b$ respectively. The intertwined LUT stage can be considered a physical implementation of the single LUT stage that is configured by challenges based on the Hadamard codeword. Within one intertwined LUT stage, the delay is always the sum of a delay in the fast inverter and one in the slow inverter. Within one intertwined LUT stage, the delay difference between the two passes is $(D_0^a - D_1^a) - (D_0^b - D_1^b) = D_a^\Delta - D_b^\Delta$. Since FPGA manufacture often use the same LUT cell design throughout in the same device, the nominal value for D_a^Δ and D_b^Δ are essentially equal. That gives $D_0 - D_1 = 0$. Therefore, the systematic bias due to inverter layout differences within the LUT cells is mitigated within each intertwined LUT stage.

However, it should be noted that, despite the nominal delays for two paths are the same, for an instance of intertwined LUT stage, the actual delays are not necessarily identical due to the variation of D^Δ , which also applies to the single LUT stage using Hadamard codeword. When multiple intertwined LUT stages cascaded, the remaining unbalanced bias can be further reduced. The effect is determined by how many stages are cascaded. [4] claimed that average bias over all devices for each RO can be known by prior testing. The systemic relative differences between them can be nullified by using the average bias. This method can be helpful here. However, helper data is needed by this method. In this work, a 2-phase 2-pass operation (Fig. 12) is proposed to solve this.

B. The intertwined programmable delay (IPD) RO

Fig. 9 The intertwined programmable delay (IPD) structure

In Fig. 8(b), based on the intertwined LUT stage, the intertwined programmable delay (IPD) is developed. On one side, as stated before, IPD stage gives the two paths the same nominal delays. On the other side, the configurable paths in LUT6 are fully utilized. Fig. 9 shows the implementation of IPD stage. Two LUT6 are configured as two cascade inverters to form the IPD stage. When IPD stage is used to build IPD-RO, Input I0 of LUT6(a) is used as the oscillation signal inputs fed from the previous stage. The output of LUT6(b) acts as the output signal of the stage and it is connected to the input of the next stage. In LPUF, each bit of the challenge is used to select one between the top and bottom delay [21]. In IPD-RO-PUF,

we propose to use a specific signal (*Switch-bit*) to control this. *Switch-bit* is independent of challenge. It is 0 in the 1st pass and 1 in the 2nd pass. Input I5 of LUT6(a) takes the *Switch-bit* while I5 of LUT6(b) takes the inversion of *Switch-bit*. The delay path inside the LUT5 is programmed by the 4-bit input I1~4. For one IPD stage, 8-bit challenge is used, whose lower 4-bit configure LUT6(a) and higher 4-bit configure LUT6(b). The delay deviates by $\delta_{I1\sim4}$ from LUT5 nominal delay D_{I5} . They are directly configured by the challenge. Therefore, we have the delay model for the intertwined LUT structure as,

$$d_{chl,0} = (D_0^a + \delta_{0,I1\sim4}^a) + (D_1^b + \delta_{1,I1\sim4}^b) \quad (6)$$

$$d_{chl,1} = (D_1^a + \delta_{1,I1\sim4}^a) + (D_0^b + \delta_{0,I1\sim4}^b) \quad (7)$$

C. PUF structures and operation

Fig. 10. Block diagram of the testbench of IPD-RO-PUF.

The IPD-RO-PUF testbench built on Xilinx Artix-7 FPGA is as shown in Fig. 10. PUF is implemented solely in PL (programmable fabric). BRAM stores challenge to ROs. The operation starts with Zynq PS (Processing system) sending the challenge to BRAM. BRAM then passes the challenge to target RO. The ROs are turned on for a pre-determined period measured by the number of system clock cycles. The counters count the number of oscillation cycles of both target RO and reference RO. BRAMs capture the readings from two counters and PS takes over all the readings and data post-processing which can also be done off FPGA chips.

Fig. 11. Structure of the IPD-RO

Fig. 11 presents the structure of IPD-RO. In addition to the even number of IPD stages, a NAND gate implemented in LUT5 is included. One of NAND's inputs is connected to a control signal from PS to start and stop the oscillation in IPD-RO. The other input of NAND is connected to the output of the last IPD stage to complete a loop. The output of the last inverter stage is also connected to a buffer, whose output goes to counter. Target RO and reference RO are both based on this structure. While target RO is configured by the challenge, the reference

RO is always configured by a fixed bit string, 00...0, for example. Therefore, a fixed path in reference RO is always used and any fluctuation in its number of oscillations can reflect the fluctuation in FPGA.

Fig. 12. The proposed 2-phase 2-pass operation. Each pass involves two phases in which the same challenge configures the IPD-RO.

Fig. 12 shows the proposed 2-phase 2-pass operation of the IPD-RO-PUF. The 1-bit response is determined by calibration in two passes. Two different challenges are used in two passes. Each pass includes two phases, in which two measurements are taken. In each pass, the dedicated *Switch-bit* is set 0 and 1 for the first and second phases, respectively. Before each pass, the challenge is sent from BRAM to ROs. Control circuits set *Switch-bit* to 0 for the 1st phase. At this point, paths inside LUTs are determined. Then, the control signal on the NAND gate is set to 1 to start the oscillations. Once the clock counter reaches the pre-set target value, the counters stop counting RO oscillations, and then control circuits stop the oscillation. At this time, the 1st phase is done. The numbers of oscillations cycles of target RO are read from counters. While the challenge being the same, the same operations repeat for the 2nd phase after the *Switch-bit* is set to 1. After two phases are done, the calibration corresponding to the challenge *chl* is calculated as

$$\text{calibrate}(chl) = N_{\text{target}}(chl, 0) - N_{\text{target}}(chl, 1) \quad (8)$$

$N_{\text{target}}(chl, 0)$ is the number of oscillation cycles that happened in target RO that is configured with challenge *chl* and *Switch-bit* 0.

For the 2nd pass, the aforementioned process is repeated with another challenge *chl'*. The second calibration number is acquired as $\text{calibrate}(chl')$.

Finally, $\text{calibrate}(chl)$ and $\text{calibrate}(chl')$ are compared to generate $\text{diff}(chl, chl')$, which is the final calibrated result. $\text{diff}(chl, chl')$ determines *r*, which is the 1-bit PUF response. They are described as,

$$\text{diff}(chl, chl') = \text{calibrate}(chl) - \text{calibrate}(chl') \quad (9)$$

$$r = \text{sign}(\text{diff}(chl, chl')) \quad (10)$$

In the rest of the paper, *r* denotes the 1-bit PUF response bit based on $\text{diff}(chl)$ and *R* for multiple PUF response bits.

How the proposed 2-phase 2-pass operation differentiates from other literature is explained as follows. Same as [18], two different challenges are used in two passes. However, in this work, each pass comprises two samplings in 2-phases with the same challenge. The 2-phase operation can mitigate the bias within LUT5, making the calibration in each pass suffer much less bias. For this reason, $\text{calibrate}(chl)$, $\text{calibrate}(chl')$ and $\text{diff}(chl, chl')$ are mostly dependent on manufacture variation. Similar to LPUF [21][22], the same challenge is used for two samplings in one pass. But the 1-bit response is determined by the operations with two different challenges. The proposed operation is not applicable to LPUF as the correlation issue

would arise.

Below, how the biases are mitigated in IPD-RO-PUF will be shown. To avoid excessive use of expressions, the procedure of how the systematic biases between two LUT5s in a LUT6 and the one within LUT5 are mitigated will be presented below respectively.

Bias between LUT5s

When $\text{calibrate}(\text{chl})$ is captured, the delays listed below are compared.

$$d_{\text{chl},0} = D_a^0 + D_b^1, \quad d_{\text{chl},1} = D_a^1 + D_b^0$$

The difference between these two is,

$$d_{\text{chl},0} - d_{\text{chl},1} = (D_a^0 - D_a^1) - (D_b^0 - D_b^1) = D_a^A - D_b^A \quad (11)$$

As pointed before, although D_a^A is nominally the same as D_b^A , (11) is not sufficient to cancel the major bias between LUT5s due to the huge variation of D^A . If PUF response is simply determined by this delay difference, uniformity would be degraded. In the final calibration $\text{diff}(\text{chl}, \text{chl}')$, the computed delay difference is $(D_a^A - D_b^A) - (D_a^A - D_b^A)$, which must be 0 regardless of how different D_a^A and D_b^A are. Therefore, $\text{diff}(\text{chl}, \text{chl}')$ is free of the bias between LUT5s.

Bias within LUT5

Here, only the biases within LUT5 are considered for the delay difference. To get $\text{calibrate}(\text{chl})$, the intertwined delay lines configured by the same challenge chl are compared as follows,

$$\begin{aligned} d_{\text{chl},0} - d_{\text{chl},1} &= (\delta_{0,\text{chl}}^a + \delta_{1,\text{chl}}^b) - (\delta_{1,\text{chl}}^a + \delta_{0,\text{chl}}^b) \\ &= (\delta_{0,\text{chl}}^a - \delta_{1,\text{chl}}^a) - (\delta_{0,\text{chl}}^b - \delta_{1,\text{chl}}^b) \end{aligned} \quad (12)$$

Fig. 3(b) demonstrated that the biases within LUT5 are quite similar in the two LUT5s. Therefore, $\delta_{0,\text{chl}}^a - \delta_{1,\text{chl}}^a$ is very close to 0. With both $\text{calibrate}(\text{chl})$ and $\text{calibrate}(\text{chl})$ having very low biases, $\text{diff}(\text{chl}, \text{chl}')$ is only dependent on the manufacturing variation.

Remark on the use of reference RO

During the 2-phase 2-pass operation, the reference RO is active for the same periods and under the same operating environment as the target RO. However, the reference RO is always configured with a fixed bit string 00...0, regardless of the value of challenge and *Switch-bit*. In one pass, two counter measurements, N_{ref}^0 and N_{ref}^1 , are taken for reference RO. There are two ways to take the calibration with the help of reference RO:

$$\begin{aligned} \text{calibrate}_{\text{wref}}(\text{chl}) &= (N_{\text{target}}(\text{chl}, 0) - N_{\text{ref}}^0) \\ &\quad - (N_{\text{target}}(\text{chl}, 1) - N_{\text{ref}}^1) \end{aligned} \quad (13)$$

$$\text{calibrate}_{\text{wref}}(\text{chl}) = \frac{N_{\text{target}}(\text{chl}, 0)}{N_{\text{ref}}^0} - \frac{N_{\text{target}}(\text{chl}, 1)}{N_{\text{ref}}^1} \quad (14)$$

N_{ref}^0 and N_{ref}^1 reflect the variation caused by the system clock, voltage, and temperature of the two phases. (13) gives the additive comparison between target RO and reference RO, while (14) gives the multiplicative comparison. The oscillation frequency is expected to be subjective to variations by multiplicative factors. Taking the ratio of the measurement from target RO and reference RO may better reflect the variation. But our data-processing shows that the difference between (13) and (14) is minor. In Section V, the improved

reliability in two ways will be shown. Simply subtracting N_{ref} from N_{target} is beneficial due to its easiness of implementation. The division in (14) requires more hardware resources. Therefore, (13) is used in the proposed IPD-RO-PUF. Reference RO is used to improve the reliability of CRP. It does not contribute much to the randomness. In this paper, to avoid any minor impact, $\text{calibrate}(\text{chl})$ is used as the calibration for PUF analysis in the rest of the paper unless otherwise indicated.

V. IPD-RO-PUF EXPERIMENTAL RESULTS

A. Systematic bias elimination.

Fig. 13. (a) Challenge to number of oscillations map (b) Challenge to $N_{\text{target}}(\text{chl}, 0) - N_{\text{target}}(\text{chl}, 1)$ map

The huge bias between the two LUT5s in one LUT6 is mitigated by the intertwined structure. Each IPD stage suffers much less from the huge bias. Experiments similar to the one in Fig. 4 are done with IPD-RO-PUF. ROs is $N=8$ stages. All 8 stages take the same 4-bit challenges so that they are always configured in the same way. Identical paths in LUT5 are always used at a time. Fig. 13(a) shows the boxplots for the 32 paths configured by 4-bit challenge chl and *Switch-bit*. When *Switch-bit* is 0, the quartiles corresponding to the 16 challenges index are shown in red. When *Switch-bit* is 1, quartiles are shown in blue. In Fig. 13(a), in contrast to Fig. 4, there is no big gap

between $N_{target}(chl,0)$ and $N_{target}(chl,1)$. Furthermore, the pattern of $N_{target}(chl,0)$ and $N_{target}(chl,1)$ are about the same. When chl keeping the same, the position of the quartiles of $N_{target}(chl,0)$ and $N_{target}(chl,1)$ are at a similar level. So, the bias within LUT5 can be mitigated by taking their difference, which is $calibrate(chl)$ in (8) and $calibrate_{wref}(chl)$ in (12).

In Fig. 13(b), the boxplot for $calibrate(chl)$ is shown. It can be found that the distribution of quartiles does not have any pattern. This means the biases within the internal structure of LUT5 are mitigated. The middle values of all clusters distribute randomly around 0 with tight deviation. In the implementation, reference RO just has a very minor effect on the distribution. $calibrate_{wref}(chl)$ has very similar distribution as $calibrate(chl)$.

B. Bit-aliasing analysis

Fig. 14. (a) Bit-aliasing (b) Intra-device bit-aliasing of IPD-RO-PUF and the ideal distribution.

To show how the quality of PUF responses is improved, experiments are conducted with 4 stages of IPD stage. Therefore, RO is configured by 32-bit challenge. 4000 challenges are selected randomly and used as inputs to the PUF. 32 instances of such ROs are implemented on each device. Because two challenges are used to give 1-bit PUF response, 2000 PUF responses are collected from each RO. The bit-aliasing measured inter and intra-device is shown in Fig. 14. The result is compared with the ideal distribution, as well as the investigation experiment shown in Fig. 6 and Fig. 7. Regarding the inter-device bit-aliasing, as shown in Fig. 14(a), for the 32 pairs of ROs at the same location on two devices, there is a 49.27% chance that ROs at the same location but different devices yield different PUF responses. And the chance is 25.20% and 25.52% for two 0's and two 1's respectively. As for the intra-device bit-aliasing, the experimental data distribution is very close to the ideal one. The quality of PUF responses is further proved by the strong uniqueness and uniformity in Table

I.

C. Mitigation against operation condition variation and noise reduction

To measure the effectiveness of the reference RO, an experiment is conducted with $N=8$ stages IPD-RO-PUF. The method in (13) is used here for demonstration. Same as reference RO, the target RO is also configured by 0000_h. Therefore, the variation due to temperature and voltage can be better seen. *Switch-bit* switches between 0 and 1 from one measurement to another. For simplicity, $N_{target}(chl,0)$ and $N_{target}(chl,1)$ together are referred to as $N_{target}(chl,\sim)$. N_{ref}^0 and N_{ref}^1 are referred to as N_{ref} . 20480 measurements samples of $N_{target}(chl,\sim)$ and 20480 sample measurements of N_{ref} are taken consecutively. $N_{target}(chl,\sim)$ fluctuates from time to time. That indicates $N_{target}(chl,\sim)$ varies along with the operation condition variations. N_{ref} has a similar fluctuation trend. N_{ref} can be used to calibrate $N_{target}(chl,\sim)$ against the operation condition variations. As shown in Fig. 15(b), compared to $N_{target}(chl,\sim)$, the trend of $N_{target}(chl,\sim) - N_{ref}$ is flattened.

Fig. 15. (a) The variation in $N_{target}(chl,\sim)$ (blue) and N_{ref} (green). (b) The variation in $N_{target}(chl,\sim) - N_{ref}$ is smaller.

As shown in Fig. 15(a), the fluctuation of $N_{target}(chl,0)$ and $N_{target}(chl,1)$ have a similar trend. However, simply subtracting $N_{target}(chl,1)$ from $N_{target}(chl,0)$ does not necessarily cancel the variation within the trend. $N_{target}(chl,0)$ and $N_{target}(chl,1)$ are taken at different time. If there is large time interval between their measurements, the system clock and environmental variables maybe be quite different. Only the reference RO that

runs at the same time as target RO suffers from the same system clock and environmental variables.

In Fig. 15, besides the changing trend, it is noticeable that both $N_{target}(chl, \sim)$ and N_{ref} have some ‘thickness’. The ‘thickness’ comes from the noise in the FPGA. Fig. 15(a) and (b) are on the same scale. It can be observed that the noise in $N_{target}(chl, \sim) - N_{ref}$ is smaller than the one in $N_{target}(chl, \sim)$ and N_{ref} themselves.

D. Experiment result analysis

32 PUFs are implemented on each of the two Xilinx Artix-7 FPGAs. Experiments are conducted with randomly selected challenges for $N=2$ and $N=4$. For $N=2$, IPD-RO can be configured by $2^{8*N}=65,536$ unique challenges. All 65,536 challenges are tested so all the cases are covered. For $N=4$, it is infeasible to have such full coverage for $2^{8*N}=4,294,967,296$ possible challenges. Instead, 80,000 challenges are tested. Since IPD-RO-PUF gives 1-bit PUF response per 2 challenges, 32,768 and 40,000 PUF responses are generated by random pairing, respectively. Each PUF response has 20 samples.

Each measurement is collected after RO running for 15.729ms. The selection of RO running time is based on the consideration of reliability and feasibility. Short RO running time would deteriorate reliability because discrete counter counts would not distinguish the difference in compared calibrations. RO running time cannot be too long as that will not be feasible for practical application. Error-correcting code (ECC) can be used to reduce either the RO running time or the error probability [23]. Experiments are conducted at room temperature and normal voltage.

To evaluate the performance of the proposed PUF, the canonical metrics described in [7] are used.

1) Uniqueness

In the tests, uniqueness indicates how different a PUF is from another PUF. Hamming distance (HD) is evaluated for two n -bit PUF responses. Ideally, uniqueness would be 50%. For traditional RO PUF, each device has only one PUF. Such PUF are implemented in the same way on all the devices. For this work, a single IRD-RO can be used as a source of PUF. The flexibility where the RO is being implemented gives extra entropy to the device. Uniqueness is evaluated in two ways:

a) *Inter-device uniqueness*: Uniqueness is measured on the ROs that are placed on two different devices i and j . On device i and j , ROs at the same locations p are considered. This uniqueness is more like the one for the traditional RO PUF. Inter-devices uniqueness is measured by the following equation, where $R_{i,p}$ is n -bit PUF responses from RO at location p on device i .

$$Uniqueness_p = \frac{2}{k(k-1)} \sum_{i=1}^{k-1} \sum_{j=i+1}^k \frac{HD(R_{i,p}, R_{j,p})}{n} * 100\% \quad (15)$$

b) *Intra-device uniqueness*: In the same device, the PUF responses from RO at different locations p and q are considered. The resulted uniqueness is intra-device uniqueness. This metric could better determine the possible entropy given by a single device i .

$$Uniqueness_i^{intra} = \frac{2}{P(P-1)} \sum_{p=1}^{P-1} \sum_{q=p+1}^P \frac{HD(R_{i,p}, R_{i,q})}{n} * 100\% \quad (16)$$

As shown in Table I, for the investigation PUF, the uniqueness and intra-device uniqueness are only 26.62% and 26.35%, respectively. It suffers from the systematic biases that were investigated in Section III. By canceling the bias between LUT5s and mitigating the bias within LUT5, the systematic bias is substantially reduced. Taking advantage of this, IPD-RO-PUF ($N=2$) has very good uniqueness. 49.18% for inter-device uniqueness and 49.72% for intra-device uniqueness are both very close to the ideal value of 50%.

2) Uniformity

Any PUF should have the same probability of giving a PUF response as 0 or 1. Otherwise, an attacker is more likely to be successful by predicting the response with a larger probability. In the PUF i , the percentage of 1's out of n PUF responses defines the uniformity, whose ideal value is 50%.

$$Uniformity_i = \frac{1}{n} \sum_{l=1}^n r_{i,l} * 100\% \quad (17)$$

Uniformity is not expected to be an issue as the two competing elements *calibrate(chl)* and *calibrate(chl')* come from the same RO. As a result, the uniformity of IPD-RO-PUF ($N=2$) is 49.99%, which is very close to the ideal value.

3) Reliability

PUF should produce consistent responses whenever the ROs are given the same challenge. Reliability indicates how likely a PUF could reproduce the same PUF response to the same challenge. On PUF i , with the challenge same as getting R_i , $R_{i,t}'$ are recorded. HD is calculated regarding R_i and $R_{i,t}'$. m is the number of samples of $R_{i,t}'$. The ideal value for reliability is 100%.

$$Reliability_i = \left(1 - \frac{1}{m} \sum_{t=1}^m \frac{HD(R_i, R_{i,t}')}{n} \right) * 100\% \quad (18)$$

For many other reported works, there was a tradeoff between randomness and reliability. In this work, the manufacturing variation is successfully extracted without compromising reliability. Reliability is further improved by introducing the reference RO. The reliability in IPD-RO-PUF is increased from 97.83% to 97.94% by using reference RO as proposed in (13). The improved reliability by (14) is 97.91%, which is quite similar. The data in Table I are based on room temperature. If the ambient temperature and supplied voltage have bigger variations, the improvement is expected to be more significant.

Table I IPD-RO-PUF experiment results, with randomly selected challenges. The test for IPD-RO-PUF ($N=2$) has covered all the challenges. $N=4$ is not fully covered, but it gives similar results.

	N	# of LUTs	Challenge length	Number of tested challenges	Uniformity	Uniqueness	Intra-Uniqueness	Reliability (wo Ref)	Reliability (w Ref using (13))	Reliability (w Ref using (14))
Investigation PUF	8	8	40	40k	50.64%	26.62%	26.35%	98.79%	N/A	N/A
IPD-RO-PUF	2	4	16	65k	49.99%	49.18%	49.72%	97.83%	97.94%	97.91%
	4	8	32	80k	50.09%	49.27%	49.77%	97.55%	97.84%	97.83%

Previous works based on 2-pass PDL lack the data for PUF performance. Implementation is not found in [18][21], and [22] focused on the side-channel analysis of the LPUF. Compared with other RO PUF designs, it is found that IPD-RO-PUF is more advantaged in both uniformity and uniqueness, as shown in Table II. The careful removal of the systematic bias helps to make a significant improvement in these two metrics. Uniformity and uniqueness in the proposed work are very close to the ideal value. With less deterministic biases, the variation in PUFs is largely based on the random variation during the manufacturing process. Besides, this work's reliability is very close to the highest level.

Table II Compare IPD-RO-PUF ($N=2$) with the previous RO PUFs.

	Uniformity	Uniqueness	Reliability
[2] Habib	50.75%	47.67%	98.1%
[15] Zhou	48.96%	47.57%	100%
[30] Anandakumar	50.61%	47.13%	99.16%
IPD-RO-PUF	49.99%	49.18%	97.94%

E. PUF challenge correlation

The metric for uniqueness only regards the 1-bit response. As pointed in [18][25], besides canonical metrics like uniqueness, the correlation in CRPs must also be investigated to evaluate the guessing complexity of PUF. To better determine the PUF correlation, a metric that evaluates the correlation between two PUF responses was proposed in [24] and applied to PDL-RO-PUF in [18]. With different challenges, one would have two 1-bit PUF responses $r_{i,j}$ and $r_{i,k}$ from PUF i . The correlation $cor_{j,k}^i$ is defined as follows,

$$cor_{j,k}^i = \begin{cases} 1, & \text{if } r_{i,j} = r_{i,k} \\ -1, & \text{otherwise} \end{cases} \quad (18)$$

Then, the correlation is summed up over all t PUFs,

$$cor_{j,k} = \frac{1}{t} \sum_{i=1}^t cor_{j,k}^i \quad (19)$$

For all the possible pair of challenges, the correlation could be determined, whose histogram is plotted in Fig. 16. IPD-RO-PUF ($N=2$) is tested with random challenges. Due to the limit of our data processing computer, 6000 PUF responses are randomly selected for the correlation calculation. Thus, $6000 \times 5999 / 2 = 17,997,000$ PUF responses pairs are considered. The fitted curve is from the distribution of the simulated uncorrelated PUF responses based on a random number generator. Without having any CRP selection, the achieved

correlation is close to the ideal distribution. In [18], VROs were carefully selected with delay threshold criteria and disparity criteria. With these criteria, some CRPs are discarded. The result in Fig. 16 is above 9% when $cor_{j,k}$ is 0, which is better than the one in [18] that was achieved with a disparity threshold of 6 (less than 8%). Furthermore, by a disparity threshold of 6, only 113 out of 8188 PUF responses were kept, i.e., more than 95% CRPs were discarded. Although it was claimed the correlation became very close to the ideal distribution when the disparity threshold is increased to 7. However, such a high threshold leads to very few CRPs (19 out of 8188) left, which would not be capable of producing convincing distribution. In contrast, the correlation shown in Fig. 16 is achieved without any CRPs selection.

Fig. 16 PUF responses correlation in IPD-RO-PUF (histogram), compared with ideal distribution (fitted curve).

VI. CONCLUSION

In this paper, a configurable RO-based PUF design is proposed. The starting point is a comprehensive experimental investigation into the systematic biases in Xilinx Artix-7 FPGA LUT6 structure. One type of systematic biases is between the two LUT5s in a LUT6. This is overcome by pairing two LUT6s as an intertwined LUT stage. The other type of systematic biases is within the LUT5 structure. The proposed 2-phase 2-pass operation mitigates that bias. With both types of biases being eliminated and mitigated, the proposed IPD-RO-PUF is solely based on random variation in LUT cells due to the manufacturing process. The timing variation between 2-pass is mitigated using reference RO, which reflects the change of system clock, voltage, etc. Performance is carefully considered, and the removal of systematic bias has a significant effect on bit-aliasing and uniqueness. The canonical metrics measured on the proposed IPD-RO-PUF are very close to the ideal values. Besides, the correlation in PUF responses is considered. These tests show that our RO would be supportive of a strong RO-

based PUF design.

REFERENCES

- [1] A. Maiti, I. Kim, and P. Schaumont, "A Robust Physical Unclonable Function with Enhanced Challenge-Response Set," *IEEE Trans. Information Forensics and Security*, Vol. 7, No. 1, Feb. 2012.
- [2] B. Habib, K. Gaj, J. Kaps, "FPGA PUF based on Programmable LUT Delays," *16th Euromicro Conf. Digital System Design*, Sep. 2013.
- [3] F. Kodytek, R. Lorencz, "A design of ring oscillator based PUF on FPGA," *IEEE 18th Int. Symposium on Design and Diagnostics of Electronic Circuits & Systems*, Apr. 2015.
- [4] L. Feiten, J. Oesterle, T. Martin, M. Sauer, and B. Becker, "Systemic Frequency Biases in Ring Oscillator PUFs on FPGAs," *IEEE Trans. Multi-Scale Computing Systems*, Vol. 2, No. 3, Jul-Sep 2016.
- [5] U. Rührmair and J. Sölter, "PUF modeling attacks: An introduction and overview," *2014 Design, Automation & Test in Euro. Conf. & Exhib.*, Dresden, Germany, 2014, pp. 1-6, doi: 10.7873/DATE.2014.361.
- [6] W. Liu, Y. Yu, C. Wang, Y. Cui and M. O'Neill, "RO PUF design in FPGAs with new comparison strategies," in *2015 IEEE Int. Symposium on Circuits and Systems (ISCAS)*, Lisbon, Portugal, 2015, pp. 77-80, doi: 10.1109/ISCAS.2015.7168574.
- [7] A. Maiti, V. Gunreddy, and P. Schaumont, "A Systematic Method to Evaluate and Compare the Performance of Physical Unclonable Functions," in *Embedded Systems Design with FPGAs*. New York, NY, USA: Springer-Verlag, 2013, pp. 245-267.
- [8] C. Herder, M. Yu, F. Koushanfar and S. Devadas, "Physical Unclonable Functions and Applications: A Tutorial," in *Proc. of the IEEE*, vol. 102, no. 8, pp. 1126-1141, Aug. 2014.
- [9] B. Gassend, D. E. Clarke, M. van Dijk, and S. Devadas, "Silicon physical random functions," in *Proc. of the 9th ACM Conf. Computer and Communications Security*, CCS 2002, Washington, DC, USA, November 18-22, 2002. ACM, 2002, pp. 148-160.
- [10] G. E. Suh and S. Devadas, "Physical Unclonable Functions for Device Authentication and Secret Key Generation," in *44th ACM/IEEE Design Automation Conf.*, San Diego, CA, 2007, pp. 9-14.
- [11] I. Kim, A. Maiti L. Nazhandali, P. Schaumont, V. Vivekraj, H. Zhang "From Statistics to Circuits: Foundations for Future Physical Unclonable Functions," in *Towards Hardware-Intrinsic Security. Information Security and Cryptography*, Springer, Berlin, Heidelberg, 2010. https://doi.org/10.1007/978-3-642-14452-3_3
- [12] A. Maiti, J. Casarona, L. McHale and P. Schaumont, "A large scale characterization of RO-PUF," in *2010 IEEE Int. Symposium on Hardware-Oriented Security and Trust (HOST)*, Anaheim, CA, 2010, pp. 94-99, doi: 10.1109/HST.2010.5513108.
- [13] D. Lim, J. W. Lee, B. Gassend, G. E. Suh, M. van Dijk and S. Devadas, "Extracting secret keys from integrated circuits," *IEEE Trans. on VLSI Systems*, vol. 13, no. 10, pp. 1200-1205, Oct. 2005, doi: 10.1109/TVLSI.2005.859470.
- [14] A. Maiti and P. Schaumont, "Improving the quality of a Physical Unclonable Function using configurable Ring Oscillators," in *2009 Int. Conf. on Field Programmable Logic and Applications*, Prague, 2009, pp. 703-707, doi: 10.1109/FPL.2009.5272361.
- [15] K. Zhou, H. Liang, Y. Jiang, Z. Huang, C. Jiang and Y. Lu, "FPGA-based RO PUF with low overhead and high stability," *Electronics Lett.*, vol. 55, no. 9, pp. 510-513, 2 5 2019, doi: 10.1049/el.2019.0451.
- [16] *Vivado Design Suite 7 Series FPGA Libraries Guide*, Xilinx, San Jose, CA, USA 2012, pp. 250-263.
- [17] A. Schaub, J. Danger, S. Guilley and O. Rioul, "An Improved Analysis of Reliability and Entropy for Delay PUFs," in *2018 21st Euromicro Conf. on Digital System Design (DSD)*, Prague, 2018, pp. 553-560, doi: 10.1109/DSD.2018.00096.
- [18] L. Feiten, K. Scheibler, B. Becker, M. Sauer, "Using different LUT paths to increase area efficiency of RO-PUFs on Altera FPGAs," *TRUDEVICE Workshop*, Dresden, Mar 2018.
- [19] *Cyclone IV Device Handbook*, Vol. 1, Altera, San Jose, CA, USA, 2009, pp. 29-36.
- [20] U. Rührmair, F. Sehnke, J. Solter, G. Dror, S. Devadas, and J. Schmidhuber, "Modeling attacks on physical unclonable functions," in *Proc. 17th ACM Conf. Comput. Commun. Security*, 2010, pp. 237-249.
- [21] O. Rioul, P. Solé, S. Guilley and J. Danger, "On the entropy of Physically Unclonable Functions," *2016 IEEE Int. Symposium on Information Theory (ISIT)*, 2016, pp. 2928-2932, doi: 10.1109/ISIT.2016.7541835.
- [22] Lars Tebelmann and Jean-Luc Danger and Michael Pehl, "Self-Secured PUF: Protecting the Loop PUF by Masking," Available: <https://eprint.iacr.org/2020/145>
- [23] Z. Cherif, J. Danger, S. Guilley and L. Bossuet, "An Easy-to-Design PUF Based on a Single Oscillator: The Loop PUF," *2012 15th Euromicro Conf. on Digital System Design*, 2012, pp. 156-162, doi: 10.1109/DSD.2012.22.
- [24] L. Feiten, M. Sauer, and B. Becker, "On metrics to quantify the inter-device uniqueness of PUFs," in *TRUDEVICE Workshop*, Dresden, March 2016, Available: <https://eprint.iacr.org/2016/320>
- [25] F. Wilde, B. M. Gammel and M. Pehl, "Spatial Correlation Analysis on Physical Unclonable Functions," in *IEEE Trans. on Information Forensics and Security*, vol. 13, no. 6, pp. 1468-1480, June 2018, doi: 10.1109/TIFS.2018.2791341.
- [26] L. Yu, X. Wang, F. Rahman and M. Tehranipoor, "Interconnect-Based PUF with Signature Uniqueness Enhancement," in *IEEE Trans on Very Large Scale Integration (VLSI) Systems*, vol. 28, no. 2, pp. 339-352, Feb. 2020, doi: 10.1109/TVLSI.2019.2943686.
- [27] J. Zhang and C. Shen, "Set-Based Obfuscation for Strong PUFs Against Machine Learning Attacks," in *IEEE Trans on Circuits and Systems I: Regular Papers*, vol. 68, no. 1, pp. 288-300, Jan. 2021, doi: 10.1109/TCSI.2020.3028508.
- [28] Y. Liu, Y. Xie, C. Bao and A. Srivastava, "A Combined Optimization-Theoretic and Side-Channel Approach for Attacking Strong Physical Unclonable Functions," in *IEEE Trans on VLSI Systems*, vol. 26, no. 1, pp. 73-81, Jan. 2018, doi: 10.1109/TVLSI.2017.2759731.
- [29] M. Majzoobi, F. Koushanfar and S. Devadas, "FPGA PUF using programmable delay lines," *2010 IEEE Int. Workshop on Information Forensics and Security*, 2010, pp. 1-6, doi: 10.1109/WIFS.2010.5711471.
- [30] N. N. Anandakumar, M. S. Hashmi and S. K. Sanadhya, "Compact Implementations of FPGA-based PUFs with Enhanced Performance," *2017 30th International Conf. on VLSI Design and 2017 16th International Conference on Embedded Systems (VLSID)*, 2017, pp. 161-166, doi: 10.1109/VLSID.2017.7.
- [31] D. Merli, F. Stumpf, and C. Eckert, "Improving the quality of ring oscillator PUFs on FPGAs," *2010 Proc. of the 5th Workshop on Embedded Systems Security (WESS)*, 2010.
- [32] R. Maes, V. Rozic, I. Verbauwhe, P. Koeberl, E. van der Sluis and V. van der Leest, "Experimental evaluation of Physically Unclonable Functions in 65 nm CMOS," *2012 Proc. of the ESSCIRC (ESSCIRC)*, 2012, pp. 486-489, doi: 10.1109/ESSCIRC.2012.6341361.
- [33] S. Tao and E. Dubrova, "Temperature aware phase/frequency detector-based RO-PUFs exploiting bulk-controlled oscillators," *Design, Automation & Test in Europe Conf. & Exhibition*, 2017, 2017, pp. 686-691, doi: 10.23919/DATE.2017.7927077.

Yangqing Hu received the B.S. degree in electrical engineering from Hefei University of Technology, Hefei, China, in 2012, and the M.S. degree in electrical engineering from University of Wisconsin-Milwaukee, Milwaukee, USA, in 2015. He is currently pursuing for PhD degree in electrical engineering at University of Wisconsin, Milwaukee. His research interests include FGPA security, embedded system design and VLSI circuits.

Yuqiu Jiang was born in Beijing, China. He received the B.S. degree in Electrical Engineering from the University of Wisconsin-Milwaukee, Wisconsin, Milwaukee and North China Electric Power University, Beijing, China in 2015. He is currently a Ph.D. candidate at the University of Wisconsin-Milwaukee. His research interests include hardware

security such as security for FPGA, PUF and PUF related applications, IP protection and machine learning modelling analysis.

Weizhong Wang received the B.S. in Electrical Engineering from the Southeast University in Nanjing China in 1990, M.S. and Ph.D. degrees in Electrical Engineering from the University of Maryland, College Park, in 1996 and 1998 respectively.

From 1998 to 2000, he was a senior device engineer with Advanced Micro Device Inc, in Sunnyvale, California. From 2000 to 2002, he was a member of technical staff at Sun Microsystems Inc. in Sunnyvale, California. Since 2002, he has been an Assistant Professor with the Electrical Engineering Department, University of Wisconsin – Milwaukee. He is the author of more than 30 articles, and 7 inventions. His research interests include high performance circuits, FPGA related security circuits, and innovative spintronics devices.