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 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 manufacture 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 the investigation, a compact PUFs circuits based on programmable Look-Up-Tables (LUTs) paths is reported. The proposed intertwined structure and the 2-phase operation mitigate 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. One of such authentication 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 query. Such manufacturing variations are effectively impossible to be predicted or replicated. A PUF that can be implemented using general-purpose, re-configurable hardware is extremely attractive.

The most fundamental challenge for all PUFs is that it must exhibit extreme sensitivity to manufacturing variations, yet it must be deterministic to provide a consistent query response. 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, the desired consistency in the outcomes under

various operation conditions mandates that the ideal PUF design should have a mechanism to identify challenge-response pairs which have sufficient margins to withstand changes due to long-term device aging and temporal random noise in measurement system.

G. E. Suh and S. Devadas proposed RO PUF in [10]. In general, RO PUF is easier to be implemented on FPGA. However, FPGA RO PUF architectures often occupy large fabric resources. In order to improve FPGA RO PUFs' area efficiency, a new RO PUF architecture based on programmable LUT delays were proposed [2]. The basic idea is to explore the delay differences among variable delay paths inside LUT cells. As the delay paths inside LUT cells can be programmed by LUT inputs, the delay of single LUT may have many variations depending on the LUT inputs. For a 4 input LUT cell, there will be 16 programmable delay paths available. As more LUTs are cascaded, the number of unique programmable delay paths grows exponentially. For example, when two 4 input LUTs are cascaded, the total number of programmable delay paths grows to 256. The difference between two programmable delay paths can be used to form RO PUF. It was referred as virtual RO PUF in [18].

Practically, there are two different implementations to form FPGA RO PUFs based on programmable LUT delay paths. The first one is to extract random delay difference between two identical programmable delay paths from two different LUT cells. [1][10]. This approach requires all the interconnects between adjacent LUTs are 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 work with more than 4 LUTs cascaded. Therefore, the potential benefit of programmable delay paths based FPGA RO PUF architecture hasn't been fully revealed. In order to overcome this limit, 2-pass programmable delay paths based FPGA RO PUF architecture was proposed [18]. The idea is to run FPGA RO PUF twice with different programmable delay paths inside LUT with the same interconnects between LUTs. In this way, the impact from interconnects between LUTs are self-eliminated. Unfortunately, there are systematic bias among programmable LUT delay paths. Such systematic bias was experimentally demonstrated in Intel FPGAs [4]. Such systematic bias among programmable delay paths inside LUTs would lead to vulnerability due to information leaks [18].

In this paper, systematic bias among programmable delay paths in Xilinx LUTs were investigated and demonstrated first.

Then, a new programmable LUT delay paths-based FPGA RO PUF architecture is proposed to eliminate the systematic bias among different delay paths inside LUT cell. The proposed architecture falls into the 2-pass RO PUF architecture category so that the interconnections between two adjacent LUTs are the same for the competing passes. Therefore, only the programmable LUT delay matters.

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

II. PROPOSED ENTROPY SOURCE AND HARVEST MECHANISM

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. In FPGA implementations, such “symmetrical” paths are instantiated by design software, which are often opaque to circuit designers. In addition, FPGA chip designers and manufactures usually do not focus on the matching among the symmetrical interconnect wires. Therefore, those “symmetrical” paths often carry systematic bias, i.e., certain paths are faster than others. Such systematic bias is due to (1) predictable wire delay differences; and (2) systematic differences among driving transistors layouts. The systematic bias reduces the randomness or entropy of traditional RO based PUFs designs.

FPGAs’ LUT cell can be programmed into many inverters and associated delay paths. Such configuration can be achieved by using input bits. As shown in Fig. 1, the 4-input LUT is programmed into 8 inverters/delay paths configurable by using input pins B, C and D, while pin A is used as the input pin. Such abundant circuit resources inside LUT cells have been explored to design PUFs with significantly improved area efficiency [2][15][18].

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

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

Traditionally, RO based PUF compares results from two separated ROs running simultaneously. The delay difference between two ROs are due to mis-match between interconnects as well as the delay path inside LUTs. The mis-match between interconnects are in general much greater than the mis-matches between programmable delay paths inside LUTs. In order to eliminate the impact from interconnect mismatch between two ROs, a 2-pass RO PUF architecture was proposed but not implemented in [18].

Fig. 2 shows the concept of the 2-pass FPGA RO PUF architecture. The Ring Oscillator portion consists of even number of inverter pairs and one NAND gate. Each inverter pair (red and blue) is programmed inside one LUT cell. It should be noted that only two inverters are shown inside LUT in Fig. 2. There should be more inverters implemented in each LUT cell. For example, 8 inverters can be formed in a single 4 input LUT cell, as shown in Fig. 1. The NAND gate is used to control the start and stop of the oscillation. The output of NAND gate is fed into a counter and data processing unit.

The operation of the proposed RO based PUF has two passes for one bit outcome generation. Two challenge bit strings are used for the two passes. The two challenge bit strings are substrings of the challenge that generates one bit outcome. In the 1st pass, the red inverters are selected by the 1st challenge bit string to form the ring oscillator. The control signal let the RO run for a pre-determined amount of time or certain number of clock cycles. The counter records how many oscillation cycles happened during that period. Then the PUF is switched to the 2nd pass. In the 2nd pass, the blue inverters are selected by the 2nd challenge bit string to form another ring oscillator. The ring oscillator runs for the same amount of time as in the 1st pass. The number of the oscillation cycles recorded by the counter is then compared against its counterpart from the 1st pass. Depending on which pass has higher number of oscillation cycles, the result is one bit of the PUF outcome.

Fig. 2. Concept of the proposed RO PUF architecture

This design requires less hardware resources than the traditional design. However, the potential system clock timing variation between 2 passes becomes a new concern because 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. To mitigate variation caused by these factors, a reference RO is introduced. To distinguish the reference RO from PUF RO, the PUF RO is referred as target RO in the remaining 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.

C. Systematic bias among programmable LUT delay paths

While the programmable LUT delay paths offered opportunity to greatly enhance the FPGA RO PUF area efficiency, it has been demonstrated that there is systematical bias among programmable LUT delay paths in Intel FPGAs. [18] So far, there hasn't been published investigation about systematic bias among programmable delay paths inside Xilinx FPGAs. The LUT structure in Intel FPGA and Xilinx FPGA are quite different. [16][19] In section III, experimental investigation will demonstrate that systemic bias similar to the one 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 ranked and/or guessed. Consequently, only 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 challenge, an improved FPGA RO PUF design will be presented in Section II.D.

D. The proposed FPGA RO PUF architecture to eliminate the systematic biases among programmable LUT delay paths.

After eliminating the systematic bias due to imperfectly matched metal interconnects, the systematic bias due to layout difference among inverters within each LUT emerges as the major source of systematic bias. In order to overcome this bias, the proposed design replaces the single inverter stage with a pair of intertwined inverters in two adjacent LUTs.

Fig. 3. Single inverter stage in the left is replaced by intertwined inverter pair in adjacent LUT cells.

Fig. 3 shows the change from the single LUT based stage (left) to the stage formed by two adjacent LUTs. In the single LUT stage design, the challenge bit strings are used to select which inverter is used in one of the two competing passes. In an instance, the top inverter is selected in the 1st pass and the bottom one selected in the 2nd pass. If the top inverter is stronger than the bottom one due to the layout difference, the systematic bias will be the dominant factor determining the output. In the two-LUT stage design (right), the challenge bit string is also used to select which path is used in the RO. Since both paths include both top and bottom inverters in LUT, any layout difference induced bias between the top and the bottom inverters are cancelled. Therefore, the systematic bias due to inverter layout differences within the LUT cells is cancelled in the proposed design. Section IV shows how intertwined LUTs eliminate systematic bias.

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

Fig. 4. Structure of 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. 4. Input I6 of the LUT6 controls which LUT5 in effect renders its output as the output of LUT6. As Xilinx datasheet does not provide any details in LUT6, the knowledge about LUT6 layout is absent. Thus, the biases between the two

physical LUT5 are investigated experimentally.

Firstly, the corresponding programmable delay paths inside two LUT5 in single LUT6 are compared. ROs oscillation counts are used to measure the delay of the programmable LUT delay paths. As shown in Fig. 4, each RO consists of an AND gate and an inverter shown in Fig. 3(a). The inverter is implemented with the LUT delay path programmed by 4 input challenge bits. One of the 16 possible LUT delay paths is chosen by 5 challenge bits. The last challenge bit is used to determine which LUT5 is used in the specific run. Fig. 5(a) shows how the 32 unique challenges map to the number of oscillation cycles on a RO. It should be noted that the blue clusters and the red clusters are the results from two LUT5 respectively. It is clearly shown in Fig. 5(a) that corresponding programmable delay paths in two LUT5 have systematic bias. This systematic is similar to the one in Cyclone IV, which has big bias between the two LUT3 in a LUT4. [18][19] While Fig. 5(a) shows the results from one RO, Fig. 5(b) shows the results from 32 ROs located at different part of the FPGA chip. It is clearly shown that systematic bias between corresponding programmable LUT delay paths located in two LUT5 within the same LUT6 exists in Xilinx 7 series FPGAs. Using the intertwined inverter pair is expected to neutralize this systematic bias.

Fig. 5. (a) Number of oscillation cycles distribution for single LUT6 with 5-bit challenge (b) Number of oscillation cycles grouped by 16 of LUT6.

After verifying the existence of the bias between the two LUT5 within the same LUT6, the next question is whether there

is systematic bias 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. If the systematic bias within each LUT5 block do exist, the differences are expected to be small and hard to detect. The delays of the programmable delay paths are still measured by using ROs. In order to enhance the signal to noise ratio in the measurements, ROs consist of an AND gate and 8 LUTs cascaded together. Each of the 8 LUTs is programmed by the same 5 challenge bits so that the identical delay paths are chosen for each LUT. Fig. 6 shows the oscillation counts corresponding to different challenge bit strings. 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's pattern for every 4 challenge bit string clusters. It appears that the last 2 bits of challenge determine the pattern within the cluster of 4. Although there's some variation in each cluster, the intrinsic systematic bias is not negligible. To mitigate the systematic bias in the LUT5, a 2-phase operation is proposed in Section IV.

Fig. 6. Boxplot of 32 instances of 32-bit RO configured by 4-bit repeated challenge. Biases between each RO is neglected here.

Building RO PUF without eliminating the bias would result in the loss in uniqueness. The systematic biases become the major contributor to determine the PUF output bit. All RO PUF at different part of FPGA chip tend to give the same PUF bit responding to the same challenge. To understand how that happens, investigation was done on a Xilinx FPGA with ROs configured in the way mentioned in [18]. 8 LUT stages plus an AND gate is used in each RO, and each PUF takes 5 challenge bits as configurable bits. Therefore, 40 challenge bits configure the RO. Two passes compete to produce 1-bit outcome as PUF bit. The competing pair is chosen randomly. In the experiments, 4000 PUF bits were collected from each of 32 ROs. [18] discussed how the competing pairs should be selected to achieve best results. Some portion of challenges are discarded by that approach. In this investigation experiment, all challenges are kept for analysis. All ROs are fed with same challenges. PUF bits collected from RO i with challenge l is denoted as R_{il} . To challenge l , the mean of PUF bits from all 32 ROs is,

$$R_l = \frac{1}{k} \sum_{i=1}^k r_{i,l} \quad (1)$$

k is the number of RO ($k=32$). 4000 samples of R_l are analyzed. As shown in Fig. 7, there is very large probability that all the 32 ROs give the same PUF bits, i.e., either output 1 or 0 from all 32 ROs located at different part of the chip. Ideally, to the same challenge, different ROs are expected to give different PUF bits. In that case, the peak of the distribution of should lie around 0.5. An ideal curve of the distribution from 32 different ROs is plotted in Fig. 7. The ideal distribution is based on the simulated data from the random number generator. Unlike the ideal distribution, the probability of having evenly split between 1 and 0 among measurement results from 32 ROs is around 2%. In comparison, 8.62% and 7.84% 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 32ROs) have the same PUF bits to the same challenges is 20.67% and 19.88% 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. This result effectively demonstrated that these RO are not unique. Systematic biases are common for all LUTs and the amount of the bias are quite significant. Such systematic bias shadows the disparity due to manufacturing randomness. For comparison, in section V, the result from proposed PUF will be presented. Besides, the uniqueness of such investigation PUF is presented in Table I.

Fig. 7. The probability distribution of average outputs from 32 ROs, Average of 1 or 0 means all 32 ROs yield the same output, while 0.5 means evenly split between 1 and 0 from 32 ROs located at different part of the chip.

IV. IMPLEMENTATION OF THE PROPOSED RO PUF

A. Structures and operation

Fig. 8. Block diagram of proposed PUF architecture.

The proposed architecture was implemented on Xilinx Artix-7 FPGA, as shown in Fig. 8. Zynq core is used to communicate with C code in PC. PUF is implemented solely in PL (programmable fabric). BRAM stores and sends challenge bit strings to ROs. The operation starts with Zynq sending the challenge bit strings to BRAM. BRAM then passes challenge bit string to target RO. The ROs are turned on for a pre-determined period which is 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 Zynq takes over all the readings and data post-processing which can also be done off FPGA chips.

Fig. 9. (a) Structure of intertwined LUT stage (b) Structure of RO

Fig. 9 shows the structure of ROs and the RO stages formed by intertwined LUT cells. As shown in Fig. 9(a), two LUT6's operating as inverters are included in one stage. Input I1 is used as the oscillation signal inputs fed from previous LUT. Inputs I2~I5 are used for 4 challenge bits to choose one out of 16 programmable delay paths inside LUT. Input I6 of LUT6_a

takes the *switch-bit* while I6 of LUT6_b takes the inversion of *switch-bit*. The *switch-bit* determines which of two LUT5 is used in the programmable delay paths. The output of LUT6_b acts as output signal of intertwined LUT stage and is connected to the input of the next intertwined LUT stage.

Fig. 9(b) presents the structure of RO. In addition to the even number stages of inverters, a NAND gate implemented in LUT5 is included. One of its inputs to NAND is connected to a control signal from Zynq to start and stop the oscillation in RO. The other input to NAND is connected to the output of the last inverter stage to complete a loop. The output of the last inverter stage is also connected to a buffer, whose output goes to counter.

As mentioned in Section II. B, the proposed RO PUF architecture has 2-pass for each operation on any given challenge. The 2-phase operation mentioned in Section III is conducted for each pass. The dedicated *switch-bit* is set 0 and 1 for two phases, respectively. The pattern in each LUT5 could mitigate each other. Before each pass, challenge bit string is sent from BRAM to ROs. Control circuits set *switch-bit* to 0 for the 1st phase. At this point, RO paths inside LUTs are determined. Then, control signal on 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 of the 1st pass is done. The numbers of oscillations cycles of target and reference RO are read from counters. While the challenge bit string 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 for the 1st pass corresponding to the 1st challenge bit string chl_1 is calculated as

$$\begin{aligned} \text{calibrate}(chl_1) = & (N_{\text{target}}(chl_1, 0) - N_{\text{ref}}) \\ & - (N_{\text{target}}(chl_1, 1) - N_{\text{ref}}') \end{aligned} \quad (2)$$

In (2), $N_{\text{target}}(chl_1, 0)$ is the number of oscillation cycles happened in target RO that is configured with challenge bit string chl_1 and *switch-bit* 0. N_{ref} is the number of oscillations cycles recorded from the reference RO running simultaneously with the target RO. N_{ref} and N_{ref}' reflect the difference caused by the system clock, voltage and temperature variation between the time when $N_{\text{target}}(chl_1, 0)$ and $N_{\text{target}}(chl_1, 1)$ are measured.

At this point, the calibrated number of oscillations corresponding to challenge bit string chl_1 is recorded. Above mentioned process is repeated with another challenge bit string chl_2 for the 2nd pass. The second calibration number is acquired as $\text{calibrate}(chl_2)$.

Finally, $\text{calibrate}(chl_1)$ and $\text{calibrate}(chl_2)$ are compared to generate $\text{diff}(chl)$, which is the final calibrated result for challenge chl . This determines 1-bit PUF response bit r . They are described as,

$$\text{diff}(chl) = \text{calibrate}(chl_1) - \text{calibrate}(chl_2) \quad (2)$$

$$r = \text{sign}(\text{diff}) \quad (3)$$

In the rest of paper, $\text{diff}(chl)$ denotes the calibration giving 1-bit PUF response bit with challenge chl , and chl is (chl_1, chl_2) . DIFF denotes the one giving multiple PUF response bits with multiple challenges. r denotes the 1-bit PUF response bit based on $\text{diff}(chl)$ and R for multiple PUF response bits based on DIFF .

V. EXPERIMENTAL RESULTS

A. Systematic bias elimination

The huge bias between the two LUT5 in one LUT6 is corrected by the intertwined structure. Experiments similar to the one in Fig. 6 is done with the intertwined structure. Fig. 10(a) shows the boxplots for the 32 paths configured by 4-bit challenge bit string chl_1 and *switch-bit*. In Fig. 10(a), contrast to Fig. 6, there is no big gap between $N_{\text{target}}(chl_1, 0)$ and $N_{\text{target}}(chl_1, 1)$. Implementation with mixture of the top and bottom LUT5s neutralize the biases between the two LUT5. Furthermore, the pattern of $N_{\text{target}}(chl_1, 0)$ and $N_{\text{target}}(chl_1, 1)$ are about the same. When chl_1 keeping the same, the position of the quartiles of $N_{\text{target}}(chl_1, 0)$ and $N_{\text{target}}(chl_1, 1)$ are at the similar level. So, the bias in LUT5 can be further eliminated by taking their difference, which is done by (2).

Fig. 10. (a) Challenge to number of oscillations map (b) Challenge to $N_{\text{target}}(chl_1, 0) - N_{\text{target}}(chl_1, 1)$ map

In Fig. 10(b), it can be found that the distribution of quartiles for $N_{\text{target}}(chl_1, 0) - N_{\text{target}}(chl_1, 1)$ does not have any pattern. The biases in the internal structure of LUT5 are canceled. All the quartiles' mean are around 0. In (2), N_{ref} is involved in the calculation of the calibration of one pass, $\text{calibrate}(chl_1)$. N_{ref} just helps to improve the stability to the timing difference. It has very minor effect on the distribution. $\text{calibrate}(chl_1)$ has very similar distribution as $N_{\text{target}}(chl_1, 0) - N_{\text{target}}(chl_1, 1)$. Therefore, with the help of 2-phase operation, the two competing passes'

calibration are free of any systematic biases.

Fig. 11. PUF bits mean extracted from proposed PUF structure.

Fig. 9 and Fig. 10 demonstrated how systematic biases are eliminated in proposed RO PUF configured by 4-bit challenges. For showing how the quality of PUF bits is improved, experiments are conducted with 32-bit configurable bits. 32 instances of such ROs are implemented and 4000 PUF bits are collected from each RO. The experimental result is shown in Fig. 11. The result is compared with the ideal distribution, as well as the investigation experiment shown in Fig. 7. Here, the experimental data's distribution is very close to the ideal one. It indicates good randomness, which provides strong resilience to attacker. The quality of PUF bits is further proved by the strong uniqueness and uniformity in Table I.

B. Mitigation against operation condition variation and noise reduction

The purpose of using reference RO is to mitigate the variation caused by system clock and environmental variables, i.e., voltage, temperature etc. Fig. 12 shows the effectiveness of the reference RO to reduce the impact from operation condition variations. Here, for simplicity, $N_{target}(chl_1, 0)$ and $N_{target}(chl_1, 1)$ are referred as $N_{target}(chl_1, \sim)$. 20480 measurements samples of $N_{target}(chl_1, \sim)$ from a target RO and 20480 sample measurements of N_{ref} from a reference RO are taken consecutively with a fixed challenge bit string given to target RO. It is clear that $N_{target}(chl_1, \sim)$ fluctuates from time to time. That indicates $N_{target}(chl_1, \sim)$ is sensitive to operation condition variations. N_{ref} has the similar fluctuation patterns. Therefore, N_{ref} can be used to calibrate $N_{target}(chl_1, \sim)$ against the operation condition variations. As shown in Fig. 12(b), compared to $N_{target}(chl_1, \sim)$, the trend of $N_{target}(chl_1, \sim) - N_{ref}$ is flattened.

While $N_{target}(chl_1, 0)$ and $N_{target}(chl_1, 1)$ have similar trend, subtracting them does not necessarily cancel the variation within the trend. $N_{target}(chl_1, 0)$ and $N_{target}(chl_1, 1)$ are taken at different time. If there is large time interval between their measurements, system clock and environmental variables maybe be quite different. Only the reference RO that runs at the same time with target RO suffer common interference.

Fig. 12. Target (blue) and Reference (red) number oscillation cycles: (a) Target RO and reference RO in the same clock region (b) In different clock regions

In Fig. 12, besides the changing trend, it is noticeable that both $N_{target}(chl_1, \sim)$ and N_{ref} have some 'thickness'. This comes from the noise in FPGA. Fig. 12(a) and (b) are in the same scale. It can be observed that the noise in $N_{target}(chl_1, \sim) - N_{ref}$ is smaller than the one in $N_{target}(chl_1, \sim)$ and N_{ref} . This noise reduction can also be seen in the PUF output $diff(chl)$. Data processing is done for the scenario without reference RO. In this scenario, as reference RO is not involved in the process to get PUF response bit, $calibrate(chl_1)$ is defined as $N_{target}(chl_1, 0) - N_{target}(chl_1, 1)$. This scenario is compared with the one using reference RO in Fig. 13. Here, $\sigma_{specific}$ is the deviation in $diff(chl)$. If a $diff(chl)$ is near decision boundary, the smaller the $\sigma_{specific}$ is, the less likely that it yields opposite 1-bit PUF bit. It is found that reference RO helps to push the deviation down for RO64 and RO128. However, when width of RO is small, 32-bit for instance, reference RO has negative impact on $\sigma_{specific}$. In this case, reference RO is less correlated to target RO. They both suffer from large random variations. Rather than reducing variations, taking the difference between N_{target} and N_{ref} adds these random variations up. Increasing the number of LUTs reduces the random variations. So, the variations in target RO and reference RO are more in common so that it can be cancelled.

Fig. 13. $\sigma_{specific}$ for the scenario w/wo reference RO. The smaller the σ is, the more stable the response is.

C. PUF Performance

To evaluate the performance of proposed PUF, the metrics described in [7] are used. One physical RO in this work acts as a PUF. Experiments are carried out on two Xilinx Artix-7 FPGAs and 32 PUFs on each FPGA ($m=64$). Each PUF is fed with 2000 challenges to get a 2000 PUF bits ($n=2000$). Each response has 20 samples ($t=19$) on each PUF.

1. Uniqueness

In the tests, uniqueness indicates how different a PUF is from another PUF. R_i is n -bit PUF bits of PUF i . Hamming distance (HD) evaluates the uniqueness between the PUF bits of PUF i and j . HD is calculated for all possible PUFs pairs selected from k PUFs. Ideally, uniqueness would be 50.

$$Uniqueness = \frac{2}{k(k-1)} \sum_{i=1}^{k-1} \sum_{j=i+1}^k \frac{HD(R_i R_j)}{n} * 100\% \quad (4)$$

2. Uniformity

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

$$Uniformity_i = \frac{1}{n} \sum_{l=1}^n r_{i,l} * 100\% \quad (5)$$

3. Reliability

Reliability is important as PUF should produce consistent responses whenever the ROs are given the same challenge. It indicates how likely a PUF could reproduce the same PUF bit 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%.

$$Reliability_i = \left(1 - \frac{1}{m} \sum_{t=1}^m \frac{HD(R_i, R_{i,t}')}{n} \right) * 100\% \quad (6)$$

To cover a wide range of compact FPGA PUF design, ROs are implemented with 32, 48, 64, and 128-bit challenge, and each width is experimented with 32 samples of RO ($k=32$) on

each of two instances of board. Each measurement is collected after RO running for 15.729ms. Experiments are conducted at room temperature and normal voltage.

For comparison purpose, the LUT programmable delay path based ROs are implemented both with and without intertwined stages. The one without intertwined designs are referred as investigation PUF. The results from those ROs have very poor uniqueness. As stated in previous sections, the systematic biases are the major contributor to such disparity.

When compared with other RO PUF designs, it is found that the proposed RO PUF is in advantage of previous works in terms of both uniformity and uniqueness. The careful removal of the systematic bias helps to make significant improvement in these two metrics. Uniformity and uniqueness in the proposed work are very close to ideal value. With less deterministic biases, the variation in PUFs is largely based the random variation during manufacturing process. Besides, this work's reliability is very close to the highest level.

		Uniformity	Uniqueness	Reliability
[14] Maiti		-	43.50%	94%
[2] Habib		50.75%	47.67%	98.1%
[15] Zhou		48.96%	47.57%	100%
	Challenge width			
Investigation PUF	80	50.6395%	37.9401%	98.9123%
	120	49.8419%	38.4202%	98.9123%
Proposed PUF	32	50.0773%	49.8745%	98.6295%
	48	50.1314%	49.5939%	98.6582%
	64	49.9760%	49.7558%	98.6329%
	128	50.0368%	49.7490%	98.6999%

Table I Compare this work with previous RO PUF.

VI. CONCLUSION

In this paper, a configurable RO based PUF design is proposed. It eliminates the systematic bias from imperfectly matched metal interconnects and LUTs cell inter structure. The systematic bias between two LUT5 in one LUT6 in Xilinx FPGA is overcome by pairing two LUTs as intertwined LUT pairs in a stage. The systematic in LUT5 is in pattern. The proposed 2-phase operation eliminate that pattern, making the PUF solely based on random variation during 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 significant effect on uniformity and uniqueness. Both uniformity and uniqueness are very close to ideal value. These tests show that our RO would be supportive for 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.

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.