IMPROVING THE QUALITY OF A PHYSICAL UNCLONABLE FUNCTION USING CONFIGURABLE RING OSCILLATORS

Abhranil Maiti, Patrick Schaumont

Electrical and Computer Engineering Department
Virginia Tech
Blacksburg, VA – 24061
email: { abhranil, schaum }@vt.edu

ABSTRACT

A silicon Physical Unclonable Function (PUF), which is a die-unique challenge-response function, is an emerging hardware primitive for secure applications. It exploits manufacturing process variations in a die to generate unique signatures out of a chip. This enables chip authentication and cryptographic key generation.

A Ring Oscillator (RO) based PUF is a promising solution for FPGA platforms. However, the quality factors of this PUF, which include uniqueness, reliability and attack resiliency, are negatively affected by environmental noise and systematic variations in the die. This paper proposes two methods to address these negative effects, and to achieve a higher reliability in an RO-based PUF. Both methods are empirically verified on a population of five FPGAs over varying environmental conditions, and demonstrate how practically useful RO-based PUF can be achieved.

1. INTRODUCTION

A PUF has the ability to create non-volatile chip-unique signature exploiting manufacturing process variation of integrated circuits. Lack of manufacturing control over sub-micron process variation makes a PUF unclonable. Hence, PUF can be used to protect private data and Intellectual Property (IP).

On an FPGA platform, implementing a PUF is a challenging task because a designer neither has the ability to exploit layout level design techniques, nor has the knowledge about the gate-level structure of an FPGA fabric. In this constrained platform, it is expected that we lose significant variation information upfront, due to the averaging effect of individual component-level variations over larger composite structures such as LUTs and other vendor-specific structures. Moreover, many PUF designs require careful routing symmetry, and this is difficult to implement on FPGA.

A Ring-oscillator-based (RO-based) PUF, proposed in [1], has several advantages in this respect. First, RO have been used widely in modeling process variations on FPGAs [5, 7] with good results. Second, implementing several identical ROs on FPGA for PUF is simplified by using hard-macro design techniques.

However, along with all these advantages, factors like correlated process variation and environmental noise caused by voltage and temperature variations, are detrimental for PUF qualities. In particular these factors degrade the uniqueness of the PUF signatures, the reliability of the signatures over varying environmental conditions, and the resiliency to external attacks. In this paper, we analyze the effects of these negative factors on PUF qualities. We also propose solutions to minimize them. The main contributions of this paper are as follows.

- We show that correlated process variation negatively affects the uniqueness, and we propose a simple design methodology to improve it.
- We also propose a new area-efficient technique, based on a configurable ring-oscillator design, to drastically improve the reliability of the PUF.
- Our analysis also shows that our design is more secure against possible attacks.

The remainder of this paper is organized as follows. Section 2 briefly describes the working principle of an RO-based PUF with background information about other types of PUF. In Section 3, we discuss the method to compensate for the effect of correlated variation on PUF uniqueness along with the new reliability-improvement technique and security analysis. Experimental results are presented in section 4. We conclude the paper in section 5.

2. BACKGROUND

A RO-based PUF exploits the fact that manufacturing process variations cause random but static variations in the frequency of identically laid-out ring oscillators. The PUF output is created by pair-wise comparison of the ring oscillator frequencies (Figure 1). These comparisons can be represented as a challenge/response function, where the chosen RO pair is the challenge, and the comparison result is the response. The term ‘response’ and ‘PUF output’ are used interchangeably throughout this paper. During an enrollment process, reference challenge-response pairs

978-1-4244-3892-1/09/$25.00 ©2009 IEEE

(CRPs) are read out from a PUF and are stored in a secure database for subsequent use.

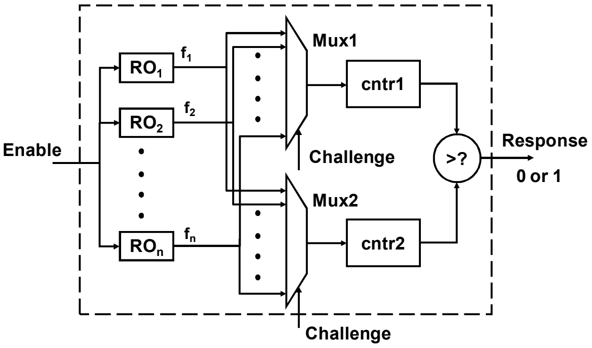

Fig 1. Ring-oscillator based PUF

There are several other implementations of PUF on FPGAs. SRAM- based PUF [2] use the random startup state of memory bits. A similar PUF was proposed in [6] as well. Lim proposed arbiter-based PUF in [3] using variability of two on-chip configurable delay paths. A Butterfly PUF [4] is based on a cross coupled latch.

In general, a PUF depends on random process variation in logic and interconnect on a chip. To our knowledge, no analysis has been done yet to analyze the effect of correlated or systematic variation on PUF. We address this issue and propose a method to counteract it.

So far, the PUF reliability has been addressed using fairly complex error correction and post-processing schemes. In this work, we make an effort to obtain reliable PUF outputs at the circuit level to avoid extra resource.

3. ANALYSIS OF PUF QUALITY FACTORS

3.1. Uniqueness

Uniqueness is an estimate of how distinctly a PUF can identify an FPGA among a group of FPGAs. The Hamming distance between two n-bit responses, R_1 and R_2, generated by a PUF from a pair of FPGAs F_1 and F_2 respectively, is a good estimate of the uniqueness of the PUF. Moreover, it is necessary to estimate the collision of response when F_1 and F_2 have the same or nearly same response for a challenge.

In a RO PUF, designer's flexibility to select RO pair, allows creation of several combinations of RO pairs producing a large number of response bits. However, the resulting set of response bits should be uncorrelated. One case of correlation is shown in [1] where a simple comparison of RO frequencies would be used (namely, if A < B and B < C, then it must be that A < C). We show that also circuit level effects can introduce correlation among PUF responses.

In a RO PUF, first, a response bit is created by comparing frequencies of a pair of ROs used as a challenge to an FPGA. This is, therefore, based on intra-die variation. Second, the resulting bit is compared against another bit generated with the same challenge from another FPGA to evaluate the Hamming distance. If the RO frequency variation is purely dependent on random variability then the average Hamming distance among a group of several chips should be around 50%. The probability of response collision among K chips, each producing an X bit response, is given in [6] as

$$P_{collision} = 1 - \prod_{n=1}^{K} (1 - \frac{n-1}{2^X}) \qquad (1)$$

This equation assumes that all the X bits are equally likely to have a value '0' or '1'. However, any bias of these bits towards a particular value will lead to a higher value of P_{collison}. Correlated or systematic variation is a factor that introduces this bias in PUF output.

The propagation delay (d_{LOOP}) around a ring oscillator loop can be expressed as a sum of two components, one representing correlated or spatial intra-die variation (d_{CORR}) and another representing stochastic intra-die variation (d_{RAND}). Ideally, for a PUF to have maximum uniqueness, the variation in d_{LOOP} should be determined by d_{RAND} alone, independent of the selected RO pair locations. However, the presence of correlated variation imposes restrictions on the physical locations of the selected RO pair.

Correlated intra-die variation can create a systematic pattern of components delays in a die e.g. a spatially systematic variation in the frequencies of several ring oscillators on an FPGA was observed in [5]. In a hypothetical example shown in figure 2, the dashed curved line represents an arbitrary pattern of spatially correlated intra-die variation in the RO frequencies.

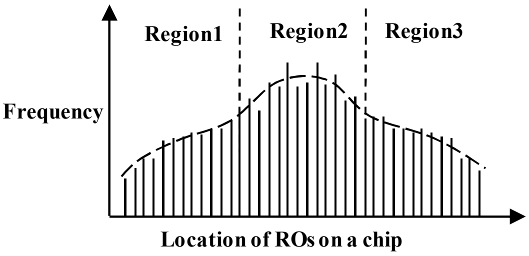

Fig 2. Scenario with correlated intra-die variation

There are approximately three regions with different average frequencies (region1, region2, region3). The frequency difference between a pair of ROs, selected from two different regions, will be influenced by the systematic pattern of RO frequencies. For example, an RO from region1 is more likely to have a frequency lower than that of an RO in region 2. If another FPGA has a similar correlated pattern in it, then PUF will produce identical responses from both the FPGAs for a given challenge which selects the RO pair from two different regions shown in Figure 2. As a result, uniqueness of PUF reduces and probability of collision increases.

One of the main causes of correlated intra-die variation is die-pattern or layout dependency [7] which creates a systematic pattern across many dies. Experimental results from [7] show that a similar systematic pattern of variation does exist across different dies.

One way to compensate for the correlated variation would be to follow a strategy to minimize its effect. Following the observation that the physical proximity reduces the magnitude of correlated variation, we propose a method to minimize the effect of correlated variations on RO-based PUF designs using the following two steps.

- Place the group of ring oscillators as close as possible to each other e.g. in a 2D array formation on the FPGA.
- While selecting the RO pair read out the responses,pick the pair of ROs such that they are located adjacent.

Ideally, it is possible to avoid the effect of correlation by analyzing the distribution of RO frequency, but in bigger PUF with large number of ROs this is a time-consuming and costly process. Moreover, it is impractical for fabrication, since each PUF would need to be measured and calibrated. Our proposed method is easier to implement, and it works always even if the nature of the correlated variation is unknown. As a limitation, this method restricts the maximum number of independent response bits from a PUF with n ring oscillators to (n - 1). This is, however, a pessimistic estimate assuming maximum correlation.

3.2. Reliability

Reliability of a PUF expresses how consistently a response R is reproduced by a PUF from an FPGA for a challenge C over several PUF read outs. It is affected by varying temperature, fluctuating supply voltage, and so on.

It is not trivial to reproduce the response from a PUF without errors because the magnitude of process variation is not high enough to safely offset environmental noise. Solving the reliability problem using error correction is a topic of ongoing research. Most of the error correction schemes are implemented as a post-processing step requiring significant resources resulting in overhead of the overall system. Instead, addressing this issue, while designing the PUF, can save a lot of resources.

In a RO-based PUF, the reliability of a response bit is solely dependent on the difference in frequencies of the RO pair used as the challenge. A higher frequency difference will ensure a higher reliability. Redundancy is one way of achieving higher reliability e.g. a method has been proposed in which a group of k 5-stage ROs are first selected, and a response bit is derived by selecting the pair of ROs in the group that has the maximum frequency difference [1]. However, this method has a large area footprint requiring k × n ring oscillators for an n-bit response, while a 5-stage ring oscillator uses almost a full configurable logic block (CLB) on Xilinx platform.

We propose a configurable RO design which enables a designer to create multiple instantiations of ROs inside a single CLB using the FPGA design techniques (Figure 3).

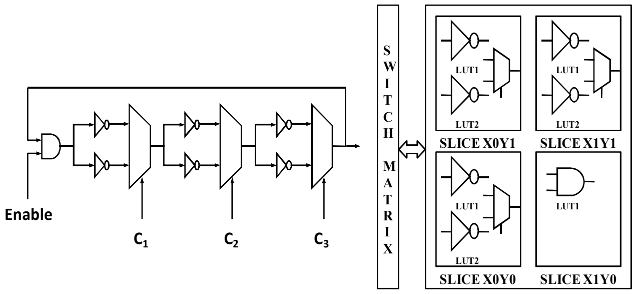

Fig. 3(a)

Fig. 3(b)

Fig 3. (a) Configurable RO (b) The configurable RO fits in a single CLB on a Xilinx Spartan 3E platform.

In the above circuit in figure 3(a), we can configure eight different ROs using the control inputs c1, c2 and c3 of the three 2:1 multiplexers. This design consumes 7 LUTs (6 for inverters, 1 for and gate) and three dedicated multiplexers (refer to figure 3(b)). It is created as a hard macro inside of a single CLB consisting of four slices. Restricting the hard macro into a CLB ensures that all the configurable ROs use only the local routing of the FPGA. Applying the same control inputs to two different CLBs will configure identical ring oscillators in both of the CLBs. Hence, it is possible to form 8 RO pairs for frequency comparisons between two CLBs instead of just a single. Due to random process variations, these 8 pairs are expected to have varying frequency differences. To achieve maximum reliability, we can select the pair which has the maximum difference in frequency. Table 1 below show how different RO pairs can be formed using the new scheme. The RO pair, for which Δf is maximum, is stored as the challenge during PUF enrollment.

Table 1. Frequency differences in a configurable RO pair

$c_1c_2c_3$	Frequency of ROs in CLB i	Frequency of ROs in CLB j	Δf
000	f_0	f'_0	$\|f_0 - f'_0\|$
001	f_1	f'_1	$\|f_1 - f'_1\|$
010	f_2	f'_2	$\|f_2 - f'_2\|$
011	f_3	f'_3	$\|f_3 - f'_3\|$
100	f_4	f'_4	$\|f_4 - f'_4\|$
101	f_5	f'_5	$\|f_5 - f'_5\|$
110	f_6	f'_6	$\|f_6 - f'_6\|$
111	f_7	f'_7	$\|f_7 - f'_7\|$

This scheme exhaustively explores all possible RO configurations within a fixed resource to find the most stable output. It effectively utilizes the available circuit resources which otherwise would have been unused because the implementation of a simple RO with five to seven delay elements will still occupy a complete CLB. Implementing smaller ROs to save area results in a higher RO frequency, which is harder to measure. The proposed method offers an efficient solution to the reliability issue as will be demonstrated in the result section. Additionally, the scheme of selecting adjacent ROs, as described in section 3.1, helps in reducing the environmental noise by relative measurement assuming closely located ROs will be subjected to similar environmental noise.

3.3. Attack Resiliency

An attacker should not be able to predict challenge-response pairs (CRPs) of a PUF given any information. Hence, correlated response bits from a PUF are discarded. Assume that on an FPGA, we have a trend of correlated variation e.g. as shown in Figure 2. Also assume that an attacker has information about this trend and a challenge is known to him. This could allow the attacker to predict the response given a challenge, for example, when the attacker can determine that ROs are located in different chip regions as shown in Figure 2. Therefore, correlated variations are a weakness from a security perspective. Although the challenge may be obfuscated to make the attack more difficult, it is better to compensate for the correlated variations. Our method of implementing ROs in an array and selecting physically adjacent RO pair for response evaluation solves this problem.

4. EXPERIMENTAL RESULTS

The RO PUF was implemented on five different Spartan XC3S500E FPGAs. It was built as a coprocessor and added to a Microblaze core using a fast simplex link (FSL) for data collection. Following the method proposed in section 3.1, we extract a (n-1) bit response from a PUF with n ROs.

4.1. Uniqueness with compensation method

Given a PUF is implemented on k FPGA chips, we define its uniqueness U as the average of the percentage Hamming distance between the responses from every pair of implementations.

$$U = \frac{2}{k(k-1)} \sum_{i=1}^{i=k-1} \sum_{j=i+1}^{j=k} \frac{h_{ij}}{n} \times 100\% \qquad (2)$$

where h_{ij} is the Hamming distance between two n-bit responses from two different FPGAs, i and j respectively for a challenge C. This formula is an estimate of inter-die variation. To demonstrate the method to compensate for correlated variation, we designed the PUF circuit in two ways. In the first design, the placements of the ROs were decided by the place and route tool. In the second one, a placement constraint on the ROs was used to align them closely in a 2-D array. Simple 5-stage ROs were used in these designs. We follow three methods of extracting the response to incrementally show the validity of our idea.

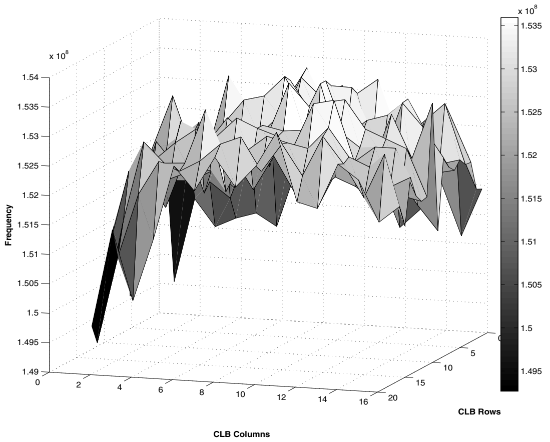

Fig 4. Distribution of average frequency of the ROs in a 256 RO based PUF with controlled placement

First, The PUF responses were extracted from the first design by deliberately selecting RO pairs that are physically distant from each other. This shows the PUF uniqueness *without* our method.

Second, Extraction was done from the second design based on the average frequency distribution of the RO array across five FPGAs as shown in figure 4 above. It has a trend with relatively lower value on the sides and a higher value in the middle. This pattern represents the correlated intra-die variation. The peaks stand for random variation. In this method, the RO pairs were selected in such a way that they are physically distant from each other as well as in different slopes of the distribution curve. This implements only the step a of our proposed method.

Third, physically adjacent RO pairs are chosen for extracting responses from the second design to implement both the steps of our proposed method.

The uniqueness graph for three different PUF settings is shown below in Figure 5 for each of the above methods. For all three PUFs, uncontrolled placement has the least uniqueness whereas it gradually increases with controlled placement. Our proposed method yields the highest uniqueness for all three settings. The net improvements are 18.09%, 9.6% and 12.78% for 64, 128 and 256 RO settings respectively where an ideal uniqueness value is 50%. Experimental results also show that close placement of ROs does not affect the reliability of the PUF.

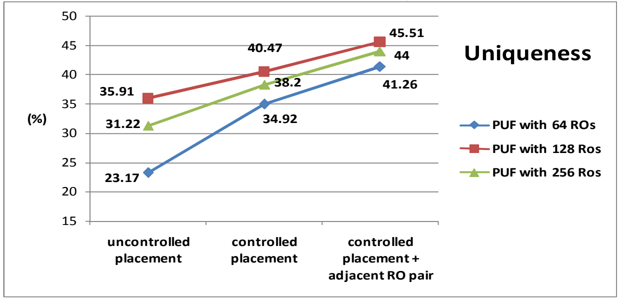

Fig 5. Effect of correlated variation on PUF uniqueness

4.2. Reliabilty with configurable RO

Reliability can be estimated as the number of stable bits out of the total response bits. However, for clarity, we present the figures for unstable bits in this section. Experiments were carried out for temperature variation from 25°C to 65°C using a temperature controlled chamber. The core voltage of the Spartan XC3S500E FPGA was varied ±20% using a controllable power supply. We compare our proposed method against all eight individual RO configurations. This shows how much improvement we can achieve using our method compared to the situation with no configurable RO. The ROs are placed in 2-D array following our method proposed in section 3.1.

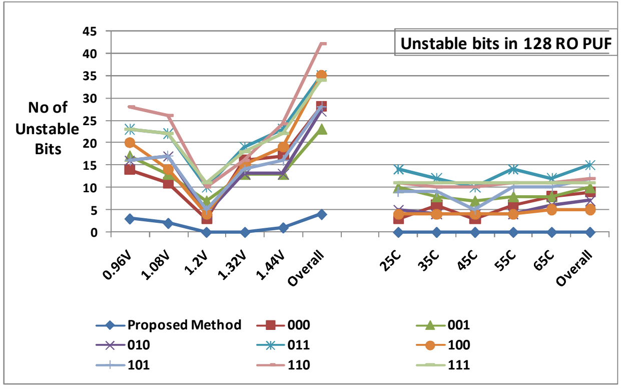

Fig 6. Unstable bits with varying voltage and temperature

Figure 6 shows the number of unstable bits for both voltage and temperature variation for a PUF with 128 ROs with 127 bit response. The three bit binary indexes stand for the RO configurations. Overall figure stands for the total number of distinct bits that have flipped at least once over the full range of varying voltage or temperature. With varying voltage, our proposed method has lowest number of unstable bits in all the cases with no unstable bits at the normal operating voltage at 1.2V. On average, individual RO configurations have 30 overall unstable bits where our method yields only 4 unstable bits. For the given range of temperature, our method produces no unstable bits, while other configurations have an average of 10 unstable bits.

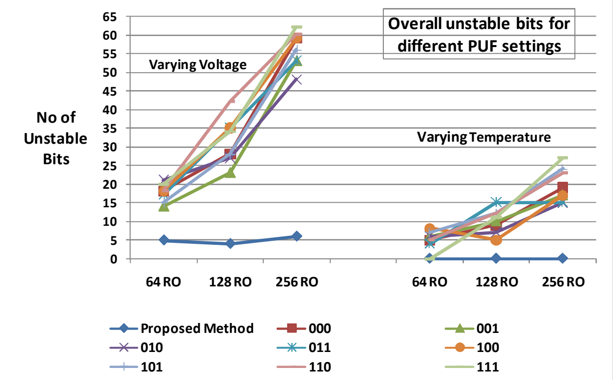

Fig 7. Overall unstable bits for different PUF settings

Figure 7 shows the number of overall unstable bits for all the PUF settings. It is clear that the proposed method yields a result that is a distinct outlier with consistently minimum number of unstable bits. The experimental result shows that the selection of the most stable RO pair is nearly equally distributed among all eight RO configurations. This shows that the maximum frequency difference between a pair of ROs is not specific to a particular RO configuration; instead, it depends on random variation. The PUF has a high value of the uniqueness with this reliability method. We found 45.9%, 43.5% and 44.1% of uniqueness for 64 RO PUF, 128 RO PUF and 256 RO PUF respectively.

5. CONCLUSION

In this paper, we proposed a method to compensate for correlated process variation. A new compact reliability method is also proposed to make the PUF more robust to environmental noises. Moreover, this PUF has improved security. Though the FPGA sample size is small, the experimental result is a good proof of concept.

6. ACKNOWLEDGEMENT

This work was supported in part by the Institute for Critical Technology and Applied Science (ICTAS).

7. REFERENCES

[1] G. E. Suh and S. Devadas. Physical Unclonable Functions for Device Authentication and Secret Key Generation. *In Proceedings of Design Automation Conference, June 2007.*

[2] J. Guajardo, S. S. Kumar, G.-J. Schrijen and P. Tuyls, FPGA intrinsic PUFs and their use for IP protection. In *Cryptographic Hardware and Embedded Systems*, 2007.

[3] D. Lim, J.W. Lee, B. Gassend, G. E. Suh, M. van Dijk and S. Devadas. Extracting secret keys from integrated circuits. In *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2005.

[4] S. S. Kumar, J. Guajardo, R. Maes G.-J. Schrijen, and P. Tuyls. Extended abstract: The butterfly PUF protecting IP on every FPGA. *In IEEE International Workshop on Hardware-Oriented Security and Trust, June,2008, 2008.*

[5] P. Sedcole and P. Y. K. Cheung. Within-die delay variability in 90nm FPGAs and beyond. *In Proceedings of IEEE International Conference on Field Programmable Technology*, 2006.

[6] Y. Su, J. Holleman, B. Otis, ”A 1.6pJ/bit 96% stable chip ID generating circuit using process variations”. *In: Digest of Technical Papers, 2007 IEEE International Solid-State Circuits Conference.*

[7] H. Onodera, “Variability: Modeling and Its Impact on Design,” *IEICE Trans. Electron.*, March 2006.