J. Cryptol. (2011) 24: 375–397
DOI: 10.1007/s00145-010-9088-4

Journal of
CRYPTOLOGY

Improved Ring Oscillator PUF: An FPGA-friendly Secure Primitive

Abhranil Maiti and Patrick Schaumont
Secure Embedded Systems Lab, Bradley Department of Electrical and Computer Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061, USA
abhranil@vt.edu; schaum@vt.edu

Received 1 September 2009
Online publication 14 October 2010

Abstract. In this paper, we analyze ring oscillator (RO) based physical unclonable function (PUF) on FPGAs. We show that the systematic process variation adversely affects the ability of the RO-PUF to generate unique chip-signatures, and propose a compensation method to mitigate it. Moreover, a configurable ring oscillator (CRO) technique is proposed to reduce noise in PUF responses. Our compensation method could improve the uniqueness of the PUF by an amount as high as 18%. The CRO technique could produce nearly 100% error-free PUF outputs over varying environmental conditions without post-processing while consuming minimum area.

Key words. Ring oscillator, Physical unclonable function, Challenge–response, Process variations (PV), Systematic process variation, Uniqueness, Reliability.

1. Introduction

An on-chip physical unclonable function, which is a die-unique challenge–response function, can generate a random response/signature that is available while the chip is powered-on. Moreover, its sensitivity to physical probing renders it tamper-resistant against invasive attacks. These properties of a PUF promises to solve issues such as intellectual property (IP) protection, chip authentication, cryptographic key generation and trusted computing.

There are practical reasons why FPGA-based PUF implementation is necessary. First, FPGAs are suitable for faster implementation of cryptographic algorithms on hardware because of their reconfigurable nature. Additionally, the regular structure of FPGAs prevents identification of the implemented circuit through an invasive attack. PUF-generated keys can be used on FPGA-based cryptographic implementation for enhanced security. On the other hand, due to the reconfigurable nature of FPGAs, intellectual properties (IPs) stored in a bitstream can be easily stolen. Therefore, protecting IPs on FPGA is an issue that needs a robust solution. PUF-based trustworthiness seems to be a promising answer to it.

However, implementing a PUF on FPGAs involves significant challenges. In this platform, neither is a designer able to exploit the layout-level design techniques, nor is any useful knowledge about the gate-level structure of the FPGA fabric available. Only the higher level design blocks such as the LUTs, the memory blocks, the connection matrices can be manipulated. In this constrained platform, significant variation information is lost 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 a careful routing symmetry that is difficult to implement on FPGAs [18]. Therefore, in spite of the availability of multiple PUF topologies, not all of them are suitable for FPGAs.

An RO-PUF, as proposed by Suh and Devadas [1], has several advantages in this respect. First, due to its sensitivity to process variations, RO has been used widely in modeling process variations with good results [2–4]. Second, implementing several identical ROs on FPGAs for a PUF is simplified by using the hard-macro design technique. However, besides these advantages, the factors like the systematic or correlated process variation and the environmental noise caused by the voltage and temperature variations degrade the uniqueness and the reliability of PUF responses. In this paper, we analyze these negative factors, and propose easily implementable yet effective solutions to mitigate them. The main contributions of this paper are:

- We show that the systematic process variation negatively affects a PUF's ability to generate unique chip-signature. We show this effect using probability, and propose a simple design methodology to alleviate it.
- A novel configurable ring-oscillator (CRO) design technique is proposed that can drastically reduce the effect of noise on PUF responses. This technique is highly area-efficient as well as effective to resolve PUF reliability issues on FPGAs.

In summary, the main goal of this paper is to optimize the use of ring oscillators in characterizing process variations present in FPGAs in order to build a robust PUF. The benefits of our optimization can be utilized at a higher level of PUF design in enhancing the security as well as in reducing the complexity of error correction. The remainder of this paper is organized as follows. In Sect. 2, a comparative analysis is provided among different PUF techniques along with the working principle of an RO-based PUF. In Sect. 3, we analyze the effect of the systematic process variation on the uniqueness of PUF along with the new reliability-improvement technique. Experimental results are presented in Sect. 4. We conclude the paper in Sect. 5.

2. Related Works

There are several PUF techniques that have been proposed for on-chip implementation. However, we will limit our discussion within those techniques that have been implemented on FPGAs.

An intrinsic PUF based on random start-up states of SRAM cells in FPGAs has been proposed by Guajardo et al. [5]. In another technique, the initial states of the flip-flops present in the reconfigurable logic of a commercial FPGA are captured, and the random nature of these states is used to build a PUF [7]. The Butterfly PUF (BPUF) emulates the behavior of an SRAM-PUF on FPGAs using two cross-coupled latches [10]. Gassend et al. proposed a PUF circuit, called silicon random function, using several configurable delay paths in a self-oscillating loop in order to measure the oscillation frequency which is a function of the delay variation of the configurable paths [8]. The Arbiter PUF (APUF) employs a pair of configurable delay paths stimulated by an input signal, and the skew in propagation delay between the two paths due to process variation is detected by an edge-sensitive latch or arbiter [9].

All the above mentioned PUFs have been shown to be implemented on FPGAs (the SRAM-PUF and the FF-PUF are intrinsic parts of an FPGA). We discuss how our contribution differs from these works.

First, none of the works discusses the effect of the systematic process variation on PUF responses. Even the work proposing the RO-PUF technique [1] does not mention the effect of the systematic process variation. In our work, we show how the systematic process variation affects the functionality of the RO-PUF.

Second, except the silicon random function (no specific quantization is mentioned), all other PUFs readily produce quantized responses in binary form. As a result, the techniques of improving responses in these PUFs are implemented in the post-quantization phase.

In contrast, our PUF improvement techniques are implemented in the pre-quantization phase as we will describe in the next section. Yu et al. proposed a reliability-improvement technique in pre-quantization phase of the RO-PUF [17]. However, we will show in Sect. 3.2 that our technique differs from it in terms of the method of reliability improvement as well as resource utilization.

Third, The SRAM-PUF employs BCH code for error correction (reliability) and hash function for privacy amplification (uniqueness) [5]. The FF-PUF [7] and the BPUF [10] use Fuzzy Extractor for error correction (reliability) and randomness extraction (uniqueness). Random physical function also proposes the use of hash function and error correcting codes to improve PUF responses. All these post-processing methods require redundant PUF information (e.g. response bits) to be saved as public data. Additionally, they require extra resources in hardware or software, depending on the implementation.

Our proposed techniques are based on circuit-level design decision, and do not require any redundant PUF information to be stored publicly. Though we use redundancy in hardware for our reliability-enhancement technique, we show that it does not require additional circuit resources compared to the existing RO-PUF design on FPGAs.

Recently, Yu et al. proposed a method of error correction for RO-PUF based on a soft-decision coding called index-based syndrome (IBS) coding [11]. This is also a post-processing technique, and makes use of BCH coding. Unlike our technique, it does not involve any circuit-level design decision.

2.1. *Ring Oscillator PUF (RO-PUF)*

An RO-PUF circuit is composed of n identically laid-out ROs, RO_1 to RO_n, with frequencies, f_1 to f_n, respectively (Fig. 1(a)). A pair of frequencies, f_a and f_b $(a \neq b)$, out of n ring oscillator outputs, are selected using a pair of multiplexers with the PUF challenge as the select bits of the multiplexers. Due to process variations, f_a and f_b tend to differ from each other. A response bit r_{ab} is produced by the quantization of two real-valued quantities, f_a and f_b, using a simple comparison method as follows:

$$r_{ab} = \begin{cases} 1 & \text{if } f_a > f_b, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

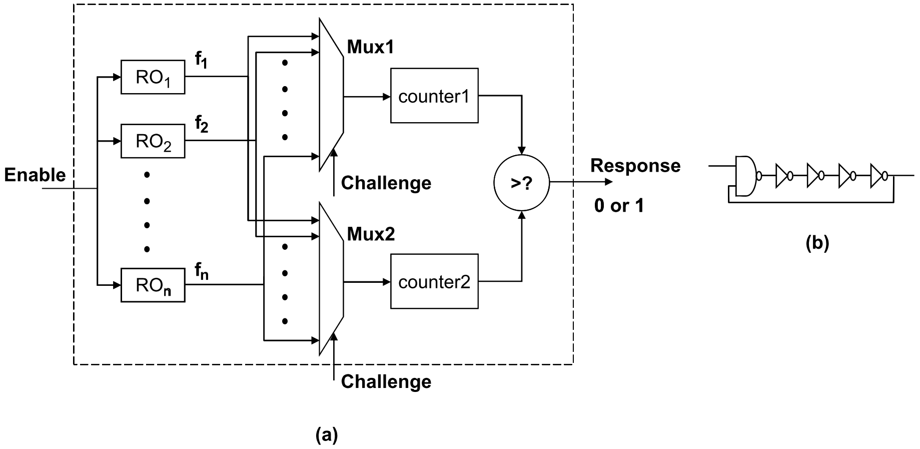

Fig. 1. (**a**) Ring oscillator PUF scheme. (**b**) A basic five-stage ring oscillator loop.

There are other possible ways of quantization that can convert a pair of real-valued frequencies into a binary bit. For example, Yu et al. proposed a method of quantization called index-based syndrome coding to convert RO frequencies into binary strings [11]. In this paper, however, we stick to the simple comparison method as the basis of our subsequent analysis. Figure 1(b) shows an example of a basic ring oscillator with five inverting delay stages. During a process called enrollment, an RO-PUF is characterized to extract all possible challenge–response pairs (CRPs) at normal environmental condition, which are stored in a secure database.

Here we explain why we think RO-PUF is FPGA-friendly. The reason is mainly related to the implementation issues on FPGAs. Nowadays, most of the FPGA manufacturers reset the start-up state of the SRAM cells to a known value rendering the SRAM-PUF [5] difficult to be used. Though the FF-PUF does not face any such problem, its outputs are highly biased, and require heavy post-processing (with the help of the post-processing, it can produce 147 error-free bits using 765 flip-flop states) [7].

On the other hand, the silicon random function [8], the APUF [9] and the BPUF [10] need structurally symmetric logic and routing so that the PV-induced mismatch can be harvested. However, on FPGAs, achieving the structural symmetry is not trivial. From a higher level of abstraction, the arrays of the logic blocks in FPGAs can be assumed to be identical. These assumptions are based on the high level data provided by the vendors about their product architecture, and are not validated by any layout-level evidence. However, these assumptions may not be valid for the interconnect. For example, two identical components inside a logic block might be connected to the routing matrix using routes with different lengths depending on the individual placements. Also, the routing tracks near the corner of an FPGA may be different from that in the middle part. The existing FPGA design tools can minimize the delay-skew between a pair of routes, but they do not guarantee the structural symmetry. Even the fabric-level manipulation tools (e.g. Xilinx FPGA Editor) allow a designer to modify the placement and the routing of a design with only a higher level of abstraction. Therefore, it often happens that the implemented PUF circuits have delay-skew introduced by the design. In this case, if the manufacturing process variation is not sufficient to offset it, the PUF output will be highly biased. From our implementation experience we have observed that even a tight timing constraint on routings, and a constrained placement of logics, produces a highly skewed output in a 90-nm commercial FPGA [18]. However, an RO-PUF has the advantage of the fact that identical ROs can be implemented in several logic blocks using the hard-macro technique provided by the standard design tools.

3. Analysis of the Uniqueness and the Reliability of RO-PUF

The trustworthiness of a PUF can be fairly estimated using three principal factors. We call them the quality factors of a PUF. They are:

- Uniqueness—This factor estimates how uniquely a PUF can distinguish different chips based on the generated response.
- Reliability—Reliability gives an estimate about how consistent/stable PUF responses are when the environment (such as ambient temperature, supply voltage) varies.
- Security/Attack resiliency—Ability of the PUF to prevent an adversary from revealing the PUF secrets, i.e. the PUF CRPs.

In this paper, we mainly focus on the first two factors: uniqueness and reliability.

3.1. *Uniqueness*

We estimate the uniqueness of a PUF by the average inter-die Hamming distance (HD) over a group of chips. With a pair of chips, i and j $(i \neq j)$, both having n-bit response, R_i and R_j respectively, the average inter-die HD among a group of k chips is defined as

$$\frac{2}{k(k-1)} \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \frac{HD(R_i, R_j)}{n} \times 100\%. \tag{2}$$

The above equation is an estimate of the inter-die variation in terms of PUF responses, and not the actual probability of the inter-die process variation. If the PUF responses are truly random in nature, i.e. if each binary response bit (r) of a PUF has an equal probability of producing a ‘0’ or a ‘1’, the uniqueness should converge to 50%. Truly random response bits also produce a uniformly distributed response string R. This can be estimated by the Hamming weight (*HW*) of R. For uniformly distributed R, the value of $HW(R) = \sum_{t=1}^{n} r_t$ should also converge to 50% of the length n of R.

Truly random bits are produced if only the random process variation exists. However, in practice, the systematic or correlated process variation exists besides the random variation. With continuous scaling down of silicon device feature size, the systematic variation is becoming more prominent. We show that the systematic variation results in bit-aliasing among different chips. In bit-aliasing, multiple chips produce significantly similar R if not completely alike. This reduces the inter-die HD resulting in false positives or ID-collision in device authentication. Su et al. showed a theoretical estimate of the probability of ID-collision that increases linearly with the size of the chip population for a fixed n [6]. Therefore, there is a motivation to mitigate the effect of the systematic variation in order to maximize the uniqueness of the PUF. In a simple comparison method of quantization of the RO-PUF, the relative physical locations of an RO-pair have a direct relationship with the systematic process variation. We show that a placement strategy of RO-pairs can compensate for the systematic variation preventing bit-aliasing. In order to do that, we analyze the PUF response generation using probability.

3.1.1. *Analysis of the Systematic Process Variation in an RO-PUF*

The total delay in a ring oscillator loop can be modeled as follows:

$$d_{\text{LOOP}} = d_{\text{AVG}} + d_{\text{RAND}} + d_{\text{SYST}} \tag{3}$$

where d_{AVG} = the nominal delay that is same for all identical ROs, d_{RAND} = delay variation due to the random process variation, d_{SYST} = delay variation due to the systematic variation; d_{RAND} and d_{SYST} could be positive or negative. We ignore the noise component in delay for this analysis.

The difference in d_{LOOP} between a pair of ROs, a and b, is determined as follows:

$$\begin{aligned}\Delta d_{\text{LOOP}} &= (d_{\text{AVG}} + d_{\text{RAND}a} + d_{\text{SYST}a}) - (d_{\text{AVG}} + d_{\text{RAND}b} + d_{\text{SYST}b}) \\ &= \Delta d_{\text{RAND}} + \Delta d_{\text{SYST}}. \end{aligned} \tag{4}$$

In the simple comparison method of quantization, a single response bit r_{ab} between a pair of ROs, a and b, is decided using (1) as follows:

$$r_{ab} = \begin{cases} 1 & \text{if } \Delta d_{\text{LOOP}} < 0, \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

The order in which a and b are selected is important because interchanging the order will generate complementary response bits. The computation of r_{ab} can be modeled as a Bernoulli trial with the probability of success, p as $\text{Prob}(r_{ab} = 1)$, and the probability of failure, q as $\text{Prob}(r_{ab} = 0)$, $(p + q = 1)$. Assignment of p and q can be interchanged without loss of generality. For an n-bit PUF response, we have n such Bernoulli trials. From (4) and (5), it is evident that the value of p is determined by the random variation and the systematic variation. We assume that the outcome of the Bernoulli trial will be equally likely between ‘0’ and ‘1’ ($p = 0.5$) if it is solely influenced by the random variation, i.e. in the absence of the systematic variation. This is because we assume no prior information about the outcome will be known when only the random variation is present. Therefore, the relation between p and the systematic variation can be summarized as follows:

$$\begin{aligned} &p = 0.5 \quad \text{if } \Delta d_{\text{SYST}} = 0, \\ &p = 0.5 \pm \Delta p \quad \text{if } \Delta d_{\text{SYST}} \neq 0, \end{aligned} \tag{6}$$

where Δp is the bias introduced in the value of p due to the systematic variation ($0 \leq \Delta p \leq 0.5$).

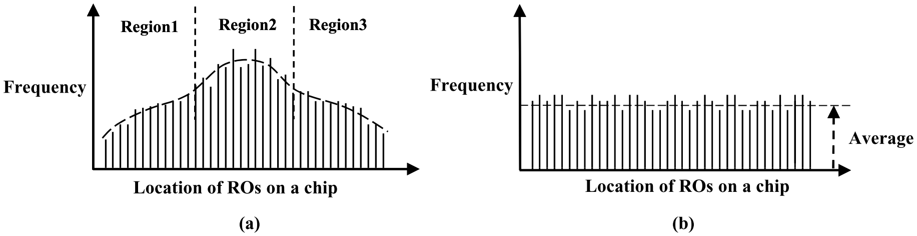

Fig. 2. (**a**) Case with the systematic variation. (**b**) Case without the systematic variation.

The systematic process variation can create a regular pattern in the delay values of ROs. This type of pattern shows a gradual change in the delay as a function of the physical location of ROs. Sedcole and Cheung mentioned such a pattern in the RO frequencies on a 90-nm FPGA due to the systematic process variation [4]. We use an arbitrary hypothetical example of such a pattern in RO frequencies as shown in Fig. 2(a) to explain how p can have bias based on the systematic variation. In Fig. 2(a), the frequencies follow a pattern caused by the systematic variation with respect to the locations of the RO. It is obvious that the value of p for the comparison between an RO in region 1 and another in region 2 is more likely to be less than $0.5(-ve\ \Delta p)$, i.e. more in favor of ‘0’. The probability becomes more than $0.5(+ve\ \Delta p)$ if an RO from region 2 is compared with another one in region 3. In the example of Fig. 2(b), no systematic variation induced pattern is found. As a result, the value of p for any pair of ROs is 0.5 in this case according to our assumption. (These two figures are representation of a conceptual idea, and not based on any real data.)

The value of p is the deciding factor in the evaluation of the inter-chip HD, i.e. the uniqueness of the PUF. The HD between two n-bit responses R_i and R_j from a pair of chips, i and j, is calculated as follows:

$$HD(R_i, R_j) = \sum_{t=1}^{n} (r_{i,t} \oplus r_{j,t}), \tag{7}$$

where $r_{i,t}$ is the tth response bit of the n-bit response string R_i of the chip i.

We can model each bitwise XOR operation to be a Bernoulli trial with the following parameters:

$$\begin{aligned} p_{\text{XOR}} &= \text{Prob}(\textit{no bit-aliasing}) = \text{Prob}\big((r_{i,t} \oplus r_{j,t}) = 1\big), \\ q_{\text{XOR}} &= \text{Prob}(\textit{bit-aliasing}) = \text{Prob}\big((r_{i,t} \oplus r_{j,t}) = 0\big), \end{aligned} \tag{8}$$

where $(p_{\text{XOR}} + q_{\text{XOR}} = 1)$. The p_{XOR} can be found as follows:

$$p_{\text{XOR}} = \text{Prob}(r_{i,t} = 1 \text{ and } r_{j,t} = 0) + \text{Prob}(r_{i,t} = 0 \text{ and } r_{j,t} = 1) = p_i + p_j - 2p_i p_j, \tag{9}$$

where $\text{Prob}(r_{i,t} = 1) = p_i$ and $\text{Prob}(r_{j,t} = 1) = p_j$, $q_i = 1 - p_i$ and $q_j = 1 - p_j$. We ignore the index t in p and q for the simplicity of representation.

If no systematic variation exists, using $p_i = p_j = 0.5$, p_{XOR} becomes 0.5, i.e. equally likely between ‘1’ and ‘0’. In the presence of the systematic variation, if Δp_i and Δp_j

Table 1. The limits of the PUF uniqueness.

U_{MIN}	0	when $l = k$ or $m = k$
U_{MAX}	$k/(2(k-1))$	when $l = m$

are the deviations from the unbiased value of 0.5 in i and j, respectively, then $p_i = 0.5 \pm \Delta p_i$ and $p_j = 0.5 \pm \Delta p_j$. Therefore, p_{XOR} can be rewritten by using (9) as follows.

For opposite signs of Δp_i and Δp_j,

$$p_{\text{XOR}} = 0.5 + 2\Delta p_i \Delta p_j. \tag{10}$$

For same signs of Δp_i and Δp_j,

$$p_{\text{XOR}} = 0.5 - 2\Delta p_i \Delta p_j. \tag{11}$$

In the case of opposite bias as in (10), the XOR operation is more likely to generate a ‘1’ increasing the value of uniqueness. In case of same bias as in (11), it is more likely to produce a ‘0’ resulting in bit-aliasing thus reducing the uniqueness. This establishes how the systematic variation causes bit-aliasing. Now, we will estimate the maximum and the minimum limits of the PUF uniqueness.

Limits of the PUF Uniqueness Suppose there are k chips, each producing an n-bit PUF response. We consider any arbitrary bit position t out of the possible n positions. For k chips, there are k such ‘tth’ response bits. For a pair-wise comparison, there will be total of ${}^kC_2 = k(k-1)/2$ XOR operations for the ‘tth’ response bit. Out of k bits, let us assume an arbitrary composition of different biases:

$$\begin{aligned} l &= \textit{number of response bits with } p = 0.5 + \Delta p, \quad (0 \le l \le k), \\ m &= \textit{number of response bits with } p = 0.5 - \Delta p, \quad (0 \le m \le k), \\ k - l - m &= \textit{number of unbiased response bits, i.e. } p = 0.5, \quad (0 \le l + m \le k). \end{aligned} \tag{12}$$

We assume that half of the unbiased bits are likely to be ‘1’ and the other half are likely to be ‘0’. Therefore, the probable total number of ‘1’ response bits is $N_1 = l + (k - l - m)/2 = k/2 + (l - m)/2$. Similarly, the probable total number of ‘0’ response bits is $N_0 = k/2 - (l - m)/2$. Therefore, total number of mismatches in the XOR operation for the tth bit can be written as:

$$N_0 \times N_1 = \bigl(k/2 + (l-m)/2\bigr) \times \bigl(k/2 - (l-m)/2\bigr) = k^2/4 - (l-m)^2/4.$$

The uniqueness of PUF as the average inter-die HD for the tth bit is:

$$U = (N_0 \times N_1)/\bigl({}^kC_2\bigr) = \bigl(k^2/4 - (l-m)^2/4\bigr)\big/\bigl(k(k-1)/2\bigr). \tag{13}$$

Table 1 shows the limits of U.

Figure 3 shows that for any large value of k, U_{MAX} converges to 50%. Essentially, the proportion of ‘1’ bits and ‘0’ bits in the tth bit position determine the uniqueness

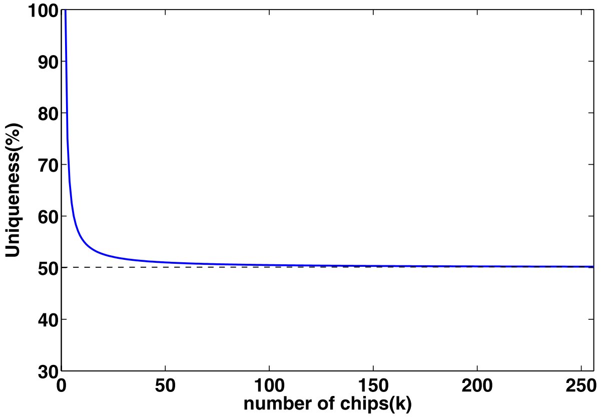

Fig. 3. The maximum PUF uniqueness for a set of k chips.

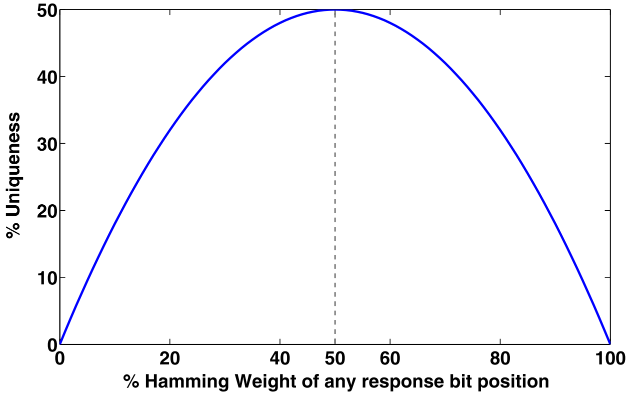

Fig. 4. PUF uniqueness vs. HW of any bit position across sample chips.

between 0 and 0.5. To show how the uniqueness varies with the ratio of '0's and '1's in the tth bit position out of k chips, we assume the number of '1' bits $= x.k$ and the number of '0' bits $= (1-x).k$, $0 \leq x \leq 1$. Note that $x.k (= \sum_{i=1}^{k} r_{i,t})$ is nothing but the Hamming weight (HW) for the tth bit among k chips. Therefore, the uniqueness becomes

$$U = k^2.x(1-x)/\big(k(k-1)/2\big) = \big(2k/(k-1)\big).\big(x(1-x)\big) = U_{\text{MAX}}.4x(1-x). \quad (14)$$

In Fig. 3, it is shown that U_{MAX} converges to 50% for a large k. Therefore, we deduce $U = 2x(1-x)$. U is plotted against the HW of the tth bit in Fig. 4.

From the graph in the figure, it can be observed that when $x = 0.5$, the HD reaches the maximum of 50%. As regards Table 1, this corresponds to the case when $l = m$. The uniqueness approaches zero when either l or m approaches k. Therefore, an even proportion of '0' and '1' bits achieves the maximum uniqueness. This can be achieved using two strategies:

1. Keeping the proportion of unbiased bits high, thus keeping $(l+m)$ low [refer to (12)];
2. Keeping $l=m$ [refer to (12)].

This analysis for the tth bit can be extrapolated for all n bits of the PUF response.

Now we discuss how l and m in (12) are dependent on the systematic variation. The systematic variation in a die may occur for several reasons. One of the main causes is die-pattern or layout dependency [2,12]. The wafer scale variation such as variation in wafer thickness can also produce this type of variation. Though this variation is systematic inside a die, it may be randomly distributed over a group of chips [12]. This case is more likely to create an even proportion of l and m for any bit position among k chips. This will result in higher uniqueness according to the strategy (2). However, bit-aliasing will occur if the systematic variation creates a similar trend across a group of dice resulting in higher proportion of either l or m. Experimental results on 0.18 μm technology reported by Onodera shows that a similar systematic pattern of variation does exist across different dice [2]. In our experimental result section, we show that similar pattern of systematic variation is visible in the sample 90-nm FPGAs used. We describe a method to compensate for the systematic variation in the next section.

3.1.2. *The Compensation Method*

Since the modification of the die is not possible, it is more practical to follow a method that will minimize the effect of the systematic variation. The systematic process variation has a spatial characteristic. The logic and interconnect in close proximity will be affected by process variations in a similar way. Therefore, two closely located ROs will have similar d_{SYST} in (3) resulting in a very low value of Δd_{SYST} in (4). As a result, Δd_{RAND} will be more likely to offset the effect of Δd_{SYST}. The end result is that the value of p, as described in (6), will converge towards 0.5, i.e. the individual response bits will be unbiased. This, according to the strategy (1) discussed in Sect. 3.1.1, will improve the uniqueness. Thus, the location of the RO-pair can improve the PUF uniqueness. To implement this strategy, we propose two steps involving the placement of ROs, and deciding an RO-pair based on their relative spatial location. The steps are:

- Place the group of ring oscillators as close as possible to each other, e.g. in a 2-dimensional array of adjacent logic blocks on the FPGA.
- While evaluating a responses bit, pick the physically adjacent pair of ROs.

These two steps essentially ensure maximum proximity of a pair of ROs used in evaluating a PUF CRP. Ideally, if one has the frequency distribution of all the ROs in a PUF, then it is possible to avoid the effect of the systematic variation by analyzing the distribution, 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 more efficient in that respect as it is very easy to implement, and it works always even if no information about the nature of the correlated variation is available. An additional advantage is that the scheme helps in reducing the environmental noise in the frequency difference of an RO-pair assuming closely located ROs will be subjected to similar environmental noise.

A disadvantage of this method is that it restricts the maximum number of independent response bits that can be extracted from a PUF with n ring oscillators to $(n-1)$. This is, however, a pessimistic estimate assuming maximum correlation. In practice, additional independent response bits can be extracted by analyzing the frequencies of all possible neighboring ROs.

3.2. *Reliability*

Though the PUF responses are expected to be static, factors such as temperature variation, supply voltage fluctuation, thermal noise introduce errors in them, and thus affect the reproducibility of the PUF responses. Reliability quantifies the variation in PUF responses over varying operating conditions. To estimate the reliability of RO-PUFs, we extract an n-bit reference response (R_i) from the chip i at the normal operating condition. The same n-bit response is extracted at a different operating condition (different ambient temperature or different supply voltage) with a value R'_i. For each operating condition, x samples of R'_i are taken. The PUF reliability is estimated as the average intra-die HD, i.e. $HD(R_i, R'_i)$ over x samples. For the chip i, it is defined as

$$S = \frac{1}{x}\sum_{y=1}^{x}\frac{HD(R_i, R'_{i,y})}{n} \times 100\%, \tag{15}$$

where $R'_{i,y}$ is the yth sample of R'_i. The responses being compared are produced from the same chip. Hence, we call it intra-die HD. A lower value of the average intra-die HD results in more reliable PUF responses. We also estimate the total number of distinct response bits that flipped at least once in the sample measurements. This gives us an estimate about the worst-case reliability value. In our Results section, we present the reliability data for the RO-PUF for a range of varying temperature and supply voltage.

3.2.1. *Existing Solutions*

A noise reduction method was proposed by Suh and Devadas exploiting the fact that the stability of a response bit is directly proportional to the frequency difference between a pair of ROs [1]. In this technique, to use an RO-pair with relatively higher frequency difference, a group of k ROs are selected, and the pair with the highest frequency difference in the group is picked for response evaluation. Though this method is a simple and efficient solution, for an n-bit response string, $k \times n$ ROs are required resulting in large area footprint which is critical for a resource-constrained platform like FPGAs.

A fuzzy extractor with universal hash function to reconstruct a key from noisy PUF responses has been discussed [13]. The additional functionality of the fuzzy extractor is the privacy amplification which converts the PUF responses into a uniformly distributed set of bits. Bosch et al. presented an implementation technique for the fuzzy extractor on FPGA [14]. This error correction method, though an effective solution, is a post-processing step requiring additional resources with redundant PUF bits as the helper data. Yin and Qu proposed a temperature-aware cooperative (TAC) method [15]. It restores the stability of an unstable RO-pair exploiting the stability of other stable pairs over a range of varying temperature. However, this method is limited to variation in temperature only. It also requires partial information about the response bits to be stored publicly that makes the PUF vulnerable to attack. Moreover, careful temperature-dependent characterization is necessary for this method. Vivekraja and Nazhandali

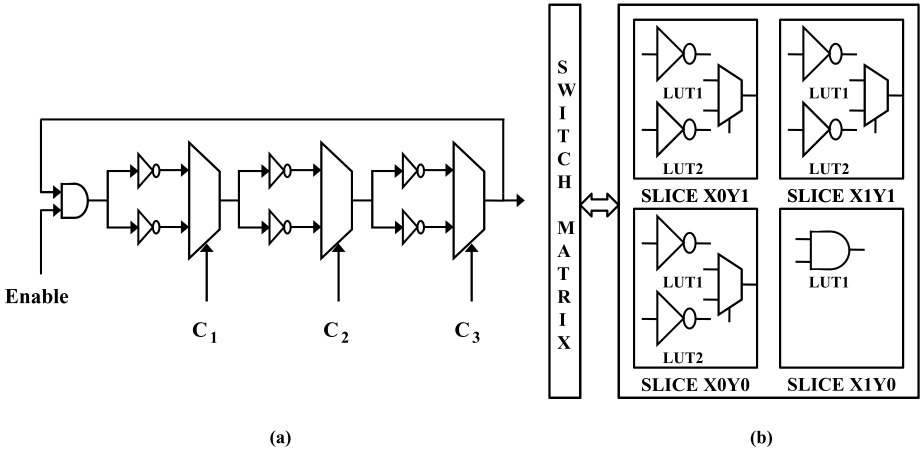

Fig. 5. (**a**) Configurable RO. (**b**) The mapping of a CRO in a single CLB on a Xilinx Spartan 3E platform.

claim that the sub-threshold operation of a chip, as well as the reverse body biasing of the CMOS transistors, improves the PUF reliability [16]. Though this technique is novel, manipulating transistor level parameter like reverse body biasing is not trivial in FPGAs. Yu et al. proposed a technique to reduce noise in RO-PUF responses [17]. They use averaging of RO frequencies along with a thresholding mechanism to select a stable response bit. This solution requires four ROs to generate one response bit resulting in significant area cost.

3.2.2. *Configurable RO (CRO) Technique*

In this section, we propose the configurable ring oscillator technique. A redundancy scheme such as majority vote or 1-out-of-k methods is suitable for noise removal. However, implementing the redundancy scheme comes with the price of additional circuit resources. For example, assuming an RO hard macro consumes an entire configurable logic block (CLB) on Xilinx FPGA, $k \times n$ CLBs will be occupied for generating an n-bit PUF key using 1-out-of-k method. The resource utilization factor for implementing the ROs is $(2/k) \times 100\%$. For $k = 8$, the utilization factor is only 25%. However, we show that redundancy can still be achieved with 100% resource utilization by modifying the RO structure. In a configurable RO structure, a 2:1 multiplexer is inserted at each delay stage of the RO to select one delay element out of the two. In such an RO with m stages, 2^m distinct oscillator rings can be configured. This technique enables a designer to create multiple instantiations of ROs inside a single CLB.

In the circuit in Fig. 5(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 Fig. 5(b)). It is created as a hard macro inside of a single CLB consisting of four slices in a Xilinx Spartan 3E 500 FPGA. 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

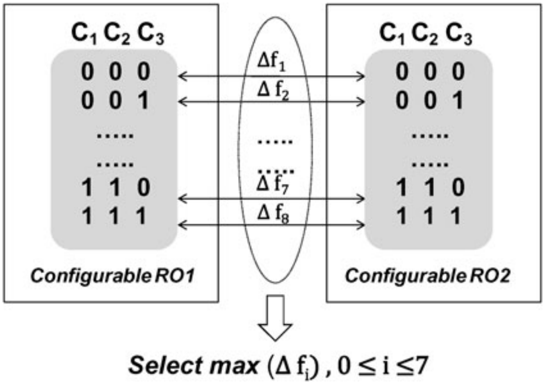

Fig. 6. Selection of the RO-pair with the maximum frequency difference between two configurable ROs.

is possible to evaluate eight pairs of frequency comparisons between two CLBs instead of just a single pair in a simple RO (without multiple configurable stages in the loop). Due to process variations, these eight pairs are expected to have varying frequency differences. To achieve the maximum reliability, we can select the pair which has the maximum difference in frequency. We call it the most stable pair. This is illustrated in Fig. 6. The comparison is done between equal configurations among different CLB's: only those ring oscillators will be structurally identical. The RO-pair, for which Δf is maximum, is stored as the challenge during the PUF enrollment. For a pair of configurable ROs, CRO_a and CRO_b, creating response bit r_{ab} with the multiplexer select input s (a 3-bit binary number with the MSB as C1, the second bit as C2 and the LSB as C3) for configuring the most stable pair ($0 \leq s \leq 7$), the challenge–response pair will be stored as : $r_{ab} \longleftarrow (CRO_a, CRO_b, s)$.

Some of the advantages of this method as compared to other solutions are:

- It is area-efficient. It exhaustively explores all possible circuit 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. In this way redundancy is achieved with a 100% resource utilization compared to the 1-out-of-k method [1] and the method of averaging [17].
- Unlike the TAC method [15], it does not require complex characterization process. Selection of maximum frequency difference is the only additional step required. Moreover, it addresses variation due to any noise source.
- Both fuzzy extractor and TAC method [13,15] require public storage of CRP information; our method requires no such data for error correction.
- Existing FPGA design technique can be used to implement our method. It does not involve very low level circuit manipulation like the sub-threshold technique [16].
- This method requires one-time characterization of the PUF to find out the most stable pair based on average RO frequency. During evaluation of a response bit, it requires only one measurement. However, the method proposed by Yu et al. requires multiple readings of RO frequencies [17].

3.2.3. *Environmental Adaptive Reliability Method*

We have observed from the experimental results that the CRO technique produces extremely low error or no error at all at many different temperatures and supply voltages. This can be utilized to calibrate the PUF response according to different operating conditions. The entire operating range of temperature and voltage can be divided into few regions. During the enrollment, the PUF will be characterized in each of these regions separately to find out the most stable RO-pair for a response bit. This environment-specific information can be stored in the CRP database. During a subsequent CRP evaluation, based on the current operating condition, the appropriate challenge will be used. If a pair of configurable ROs, CRO_a and CRO_b, create a response bit r_{ab} at temperature T_{REF} and supply voltage V_{REF} with the index s for the most stable pair $(0 \leq s \leq 7)$, then the challenge–response pair will be stored as $r_{ab} \longleftarrow (CRO_a, CRO_b, s, T_{\mathrm{REF}}, V_{\mathrm{REF}})$.

A configurable RO can produce very highly stable response at varying conditions within the allowable operating region of an FPGA, because it can dynamically adjust itself by choosing the most stable RO-pair depending on the environment. However, a simple RO (without the configurable property) only has a single static configuration, and hence does not have the capability of dynamically enhancing the reliability of the response bits.

Though a PUF with simple ROs can be registered at different environmental conditions, the stability may not be guaranteed because there may be some RO-pairs that would reach a very low frequency difference resulting in unstable response bits. On the other hand, the CRO technique can still select an RO-pair to maximize the frequency difference at a specific condition out of the multiple available configurations. Thus, it has a higher probability of rectifying the instability. This is also supported by the experimental results in Sect. 4.2 where the static configurations produce significantly larger number of unstable response bits compared to the CRO technique at different temperature and the supply voltage.

4. Results

All the experimental results have been obtained from implementations on Xilinx Spartan XC3S500E FPGAs. All uniqueness analyses related to the systematic variation effect are based on simple RO structure with no configurable stages; the configurable RO is used for the reliability assessment.

4.1. *Uniqueness*

In order to demonstrate the proposed method to compensate for the systematic variation, we designed the PUF circuit in two ways. In the *first* design, the placements of ROs were decided by the place and route tool without any user constraint. After the placement and routing process, it was confirmed that the ROs are spread randomly over the FPGA. In the *second* design, with a placements constraint, ROs were aligned closely in a 2-D array fashion as shown in Fig. 7. The black boxes on the FPGA fabric are the RO hard macros. We follow three methods of extracting response to incrementally show the validity of the compensation method.

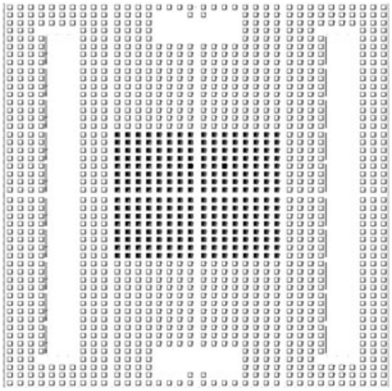

Fig. 7. Sample array configuration of a PUF with 256 ROs on Spartan XC3S500E FPGA.

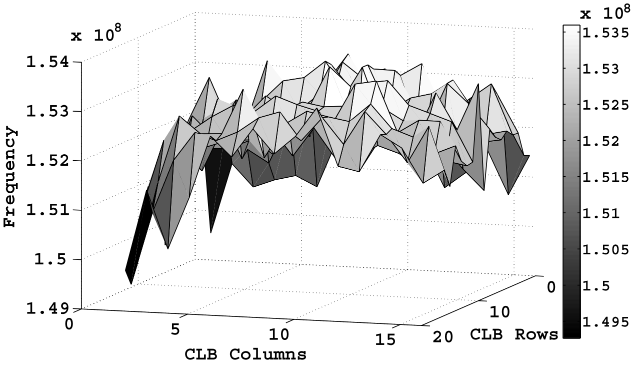

Fig. 8. Distribution of the average frequency of ROs in a PUF with 256 ROs with controlled placement.

Method 1 (*M*1)—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. We select the RO-pairs with maximized physical distance to show the worst-case systematic variation.

Method 2 (*M*2)—The extraction was done from the second design based on the average frequency distribution of the RO array across five FPGAs as shown in Fig. 8. It can be observed from Fig. 8 that there is a trend starting at a relatively lower value on the right, and then it increases slowly towards the middle, and it drops again on the left. This pattern represents the systematic intra-die variation. The uneven peaks stand for the random variation. The objective of this method is to implement the group of ROs in proximity, but selected RO-pairs for CRP evaluation are kept as far as possible within the array. For example, one RO is chosen from the middle to be compared with one that is at one of the sides. This implements only the first part of the compensation method. The RO-pair could have been chosen randomly within the array, but we deliberately choose this configuration in order to show the possible worst-case scenario.

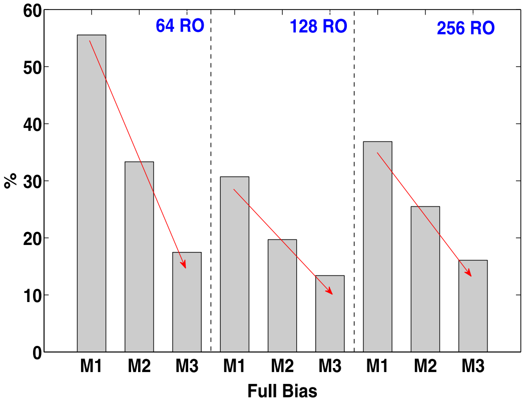

Fig. 9. Full bias.

Method 3 ($M3$)—Physically adjacent RO-pairs are chosen inside the array for extracting responses from the second design to implement both the steps of our proposed method.

We validate our proposed technique by measuring the following parameters:

(a) HW_t, the Hamming weight (HW) for each tth bit position ($1 \leq t \leq n$) across k FPGAs $= \sum_{i=1}^{k} r_{i,t}$—This is an estimate of the bias Δp in (6) introduced in the response bits due to the systematic variation. In our experiment, for $k = 5$, we designate a bit position as follows:

full bias—all 5 *bits are either* '1' *or* '0',

medium bias—4 *bits are* '1' *and* 1 *bit is* '0' *or vice versa*,

no bias—3 *bits are* '1' *and* '2' *bits are* '0' *or vice versa*.

Figures 9, 10 and 11 show the distribution of fully biased, medium biased and unbiased bit positions respectively for three different sizes of the PUF for each of the experimental method. In the figures, it is evident that our proposed method improves the proportion of the unbiased bit positions consistently over all the three PUF configurations (64, 128 and 256 ROs) while reducing the proportion of the fully biased bit positions. The increases in unbiased bit positions are 14, 14 and 22% respectively for 64, 128 and 256 ROs, whereas the respective decreases in fully biased bit positions are 38, 17 and 20%.

(b) HW_n, the Hamming weight (HW) of n-bit response from FPGA $i = \sum_{t=1}^{n} r_{i,t}$—This is the estimate of the uniformity of PUF responses. For truly random bits, the percentage of HW_n should be close to 50% of n. The results in Figs. 12, 13 and 14 show that our technique with controlled placement and adjacent RO-pair yields fairly consistent value close to 50%. The other two methods produce fluctuating results deviating from 50%.

(c) Average inter-die HD—Finally, we show how the inter-die HD varies for the three methods as an estimate of uniqueness. The uniqueness graph for three different PUF settings is shown in Fig. 15 for each of the above methods: $M1$, $M2$ and $M3$. For all three

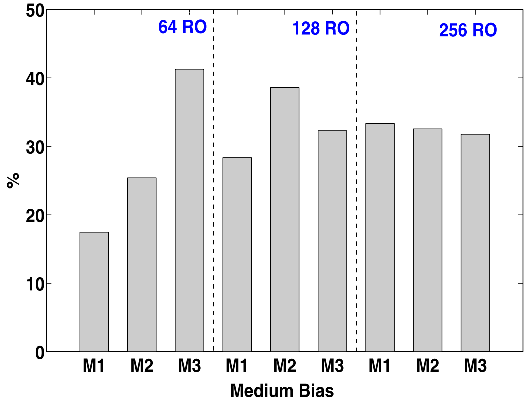

Fig. 10. Medium bias.

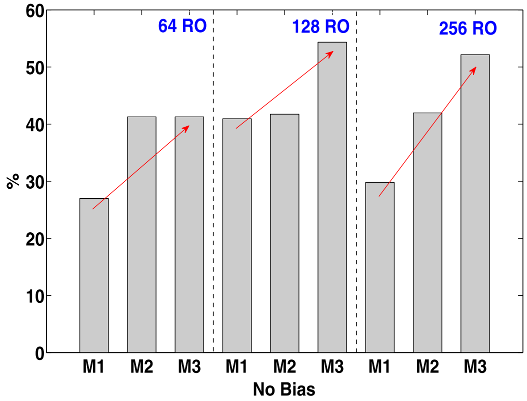

Fig. 11. No bias.

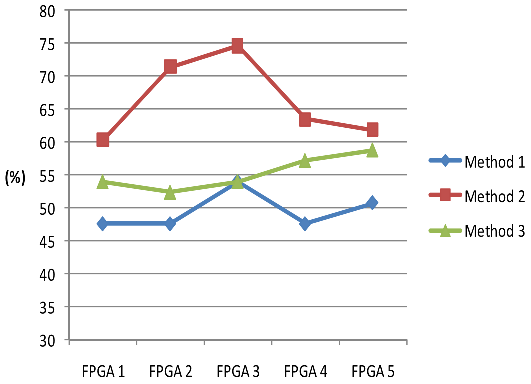

Fig. 12. HW of the response from a PUF with 64 ROs.

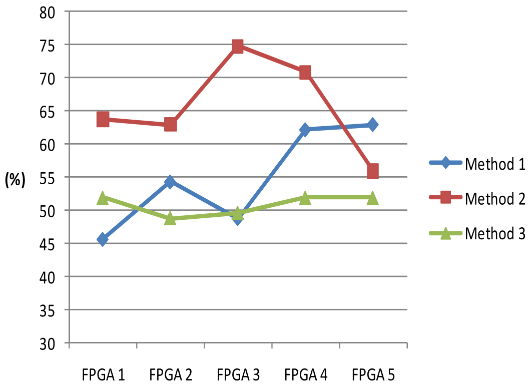

Fig. 13. HW of the response from a PUF with 128 ROs.

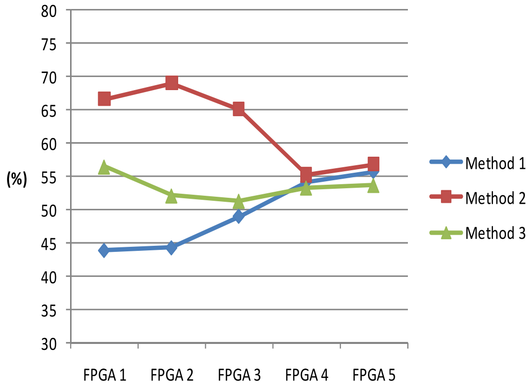

Fig. 14. HW of the response from a PUF with 256 ROs.

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.

Validation using a large population of chips—We tested a bigger PUF circuit with 512 ROs using our proposes technique. We used a population of 125 Spartan 3E S500 FPGAs for this experiment. A large-scale experiment was conducted to study PUF characterization using these chips [19]. A summary of the results is shown in Table 2. Figure 16 shows the distribution of uniqueness with an average of 47.31% with the minimum of 38.98% and the maximum of 56.36%.

4.2. *Reliability*

In Sect. 3.2, we defined reliability as the percentage of average intra-die HD (refer to (15)). However, for clarity, instead of the percentage figure, we present the absolute number of unstable bits that are different from the reference response bits. Experiments

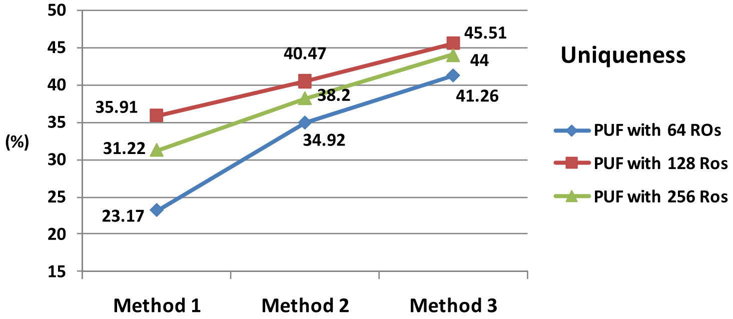

Fig. 15. Improvement of the PUF uniqueness due to the proposed compensation method.

Table 2. Experimental results for the large-scale experiment.

HW_t	HW_n	*Average inter-die HD*
50.72%	50.72%	47.31%

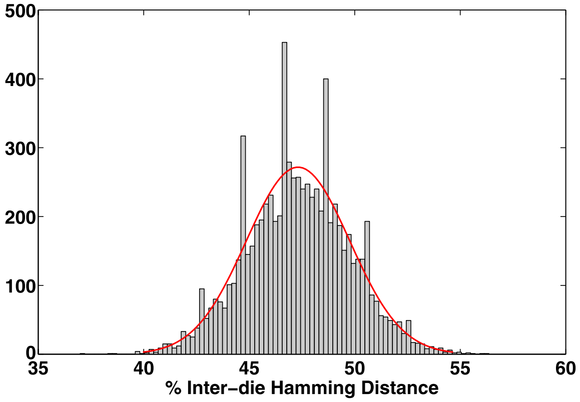

Fig. 16. Uniqueness distribution among 125 FPGA chips.

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 the CRO technique compared to the situation with no configurable RO. Figure 17 shows the number of unstable bits for both voltage and temperature variation for a PUF with 64 ROs. We show nine cases. The 'CRO' shows reliability using the proposed CRO technique that finds a pair of ROs that has the maximum frequency difference out of the possible eight configurations. Each of the other results (shown by 3-bit indexes) show reliability corresponding to a fixed configuration for all ROs specified by the 3-bit index. For example, the line corresponding to '001' shows the reliability of PUF when all the n CROs are configured with $C1 = 0$, $C2 = 0$ and $C3 = 1$. C1, C2 and C3 are the

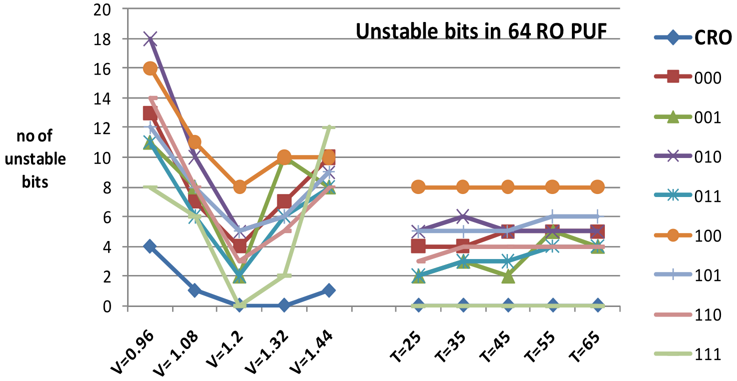

Fig. 17. Reliability with varying voltage and temperature for PUF with 64 ROs.

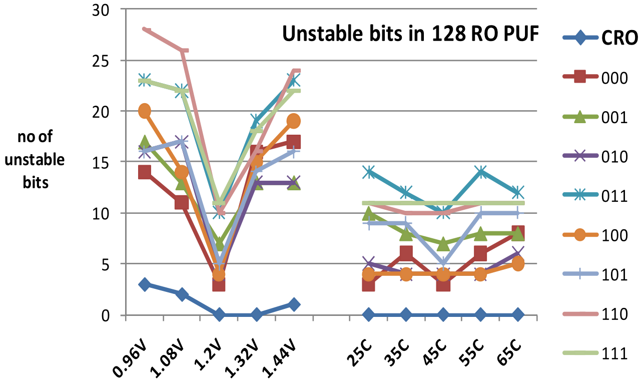

Fig. 18. Reliability with varying voltage and temperature for PUF with 128 ROs.

select inputs of the three 2:1 multiplexers in the CRO loop as shown in Fig. 5(a). This is done in order to compare the proposed technique with the scenario when all the ROs are identically configured. With varying voltage, our proposed method has the lowest number of unstable bits in all the cases with no unstable bits at the normal operating voltage at 1.2 V. For varying temperature, the CRO technique produces 100% reliable PUF outputs in all the cases. Figures 18 and 19 show similar reliability results for PUFs with 128 and 256 ROs respectively. Figure 20 shows the number of distinct unstable bits for all the PUF settings. It is clear that the proposed method yields a result that is a distinct outlier with a consistently lower number of unstable bits. The experimental result shows that our technique selected all eight RO configurations with even proportion. This shows that the maximum frequency difference between a pair of CROs is not specific to a particular RO configuration; instead, it depends on the 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, 128 and 256 RO PUF respectively. For the environment adaptive technique, we enrolled the PUF at three different reference voltages

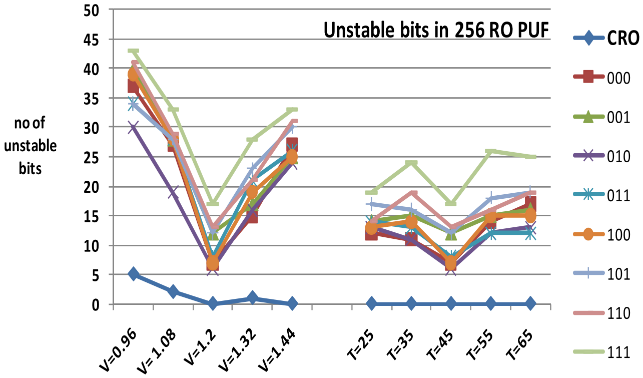

Fig. 19. Reliability with varying voltage and temperature for PUF with 256 ROs.

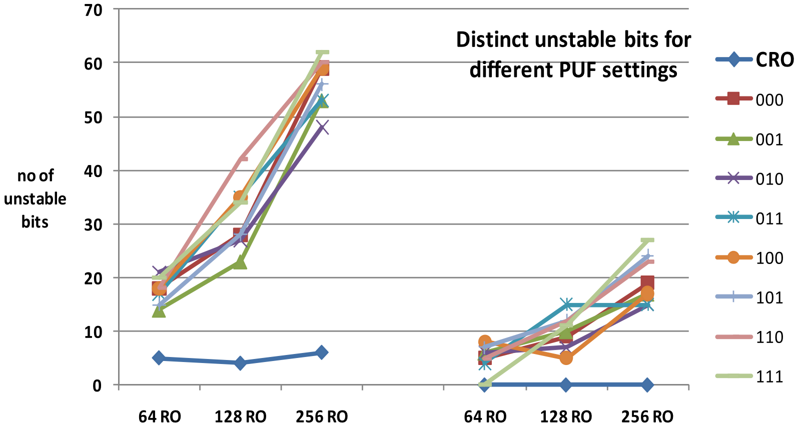

Fig. 20. Distinct unstable bits for different PUF settings.

of 0.96, 1.2 and 1.44 V. Since our method yielded 100% error-free outputs as shown in the previous section, we do not enroll for different temperatures. This measurement is done on a PUF with 128 ROs. Figure 21 shows that when the PUF is enrolled at a reference voltage of 0.96 V, it is error-free at 0.96 V and 1.08 V. Enrollment at 1.2 V produces error-free output at 1.2 V and 1.32 V while the same PUF enrolled at 1.44 V produces error-free output at 1.32 V and a single-bit error at 1.44 V. This indicates that characterizing the PUF at different operating conditions can significantly eliminate noise.

5. Conclusion

We have shown that an RO PUF requires careful design decisions to avoid the systematic process variation effect. The compensation method using the placement strategy and the selection of RO-pairs significantly improves the uniqueness of the PUF. On the

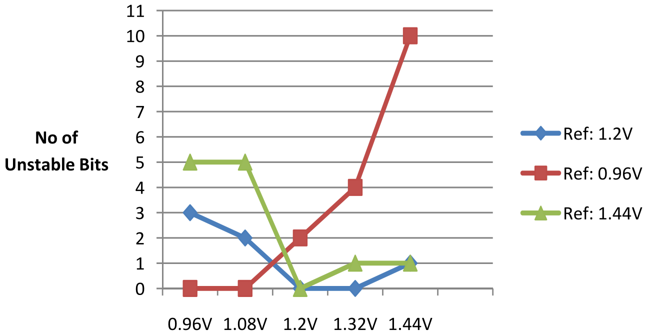

Fig. 21. Environment-adaptive reliability method using the CRO technique.

other hand, the configurable RO technique offers a very compact yet highly efficient solution to the PUF reliability issue. The on-chip experimental results work as a strong proof of concept. We validated our idea on a large chip population. The proposed technique would eventually result in practical benefits like precise device identification and simplification of higher level error correction techniques.

On the other hand, though RO PUF has certain implementation advantages on an FPGA, it suffers from low number of CRPs. In the future, one main area of our research would be to address this issue.

Acknowledgements

This work has been supported in part by National Science Foundation (NSF) Grant No. 0855095. We also thank the reviewers for their valuable feedbacks in improving this paper. We are also thankful to the guest editors for their efforts.

References

[1] G.E. Suh, S. Devadas, Physical unclonable functions for device authentication and secret key generation, in *Proceedings of the 44th Annual Design Automation Conference* (ACM, New York, 2007), pp. 9–14

[2] H. Onodera, Variability: Modeling and its impact on design. *IEICE Trans. Electron.* **E89-C**(3), 342–348 (2006)

[3] L.-T. Pang, B. Nikolic, Measurements and analysis of process variability in 90 nm CMOS. *IEEE J. Solid-State Circuits* **44**(5), 1655–1663 (2009)

[4] P. Sedcole, P.Y.K. Cheung, Within-die delay variability in 90-nm FPGAs and beyond, in *Proceedings of International Conference on Field Programmable Technology* (IEEE, New York, 2006), pp. 97–104

[5] J. Guajardo, S.S. Kumar, G.-J. Schrijen, P. Tuyls, FPGA intrinsic PUFs and their use for IP protection, in *Proceedings of the 9th International Workshop on Cryptographic Hardware and Embedded Systems*. LNCS, vol. 4727 (2007), pp. 63–80

[6] Y. Su, J. Holleman, B. Otis, A 1.6 pJ/bit 96% stable chip ID generating circuit using process variations, in *International Solid-State Circuits Conference. Digest of Technical Papers* (IEEE, New York, 2007), pp. 406–411

[7] R. Maes, P. Tuyls, I. Verbauwhede, Intrinsic PUFs from flip-flops on reconfigurable devices, in *3rd Benelux Workshop on Information and System Security*, 2008

[8] B. Gassend, D.E. Clarke, M. van Dijk, S. Devadas, Silicon physical random functions, in *Conference on Computer and Communications Security*, ed. by V. Atluri (ACM, New York, 2002), pp. 148–160

[9] D. Lim, J.W. Lee, B. Gassend, G.E. Suh, M. van Dijk, S. Devadas, Extracting secret keys from integrated circuits. *IEEE Trans. Very Large Scale Integr.* **13**(10), 1200–1205 (2005)

[10] S.S. Kumar, J. Guajardo, R. Maes, G.J. Schrijen, P. Tuyls, The butterfly PUF: Protecting IP on every FPGA, in *International Workshop on Hardware-Oriented Security and Trust* (IEEE, New York, 2008), pp. 67–70

[11] M.-D. Yu, S. Devadas, Secure and robust error correction for physical unclonable functions. *IEEE Des. Test Comput.* **27**(1), 48–65 (2010)

[12] D.S. Boning, S. Nassif, Models of process variations in device and interconnect, in *Design of High Performance Microprocessor Circuits*, ed. by A. Chandrakasan, W. Bowhill, F. Fox (IEEE Press, New York, 2000), Chap. 6

[13] B. Skoric, P. Tuyls, An efficient fuzzy extractor for limited noise, Cryptology ePrint Archive, Publication Number 030, 2009

[14] C. Bosch, J. Guajardo, A. Sadeghi, J. Shokrollahi, P. Tuyls, Efficient helper data key extractor on FPGAs, in *Proceedings of the 10th International Workshop on Cryptographic Hardware and Embedded Systems*. LNCS, vol. 5154 (2008), pp. 181–197

[15] C. En Yin, G. Qu, Temperature-aware cooperative ring oscillator PUF, in *International Workshop on Hardware-Oriented Security and Trust* (IEEE Press, New York, 2009), pp. 36–42

[16] V. Vivekraja, L. Nazhandali, Circuit-level techniques for reliable physically unclonable functions, in *International Workshop on Hardware-Oriented Security and Trust* (IEEE Press, New York, 2009), pp. 30–35

[17] H. Yu, P.H.W. Leong, H. Hinkelmann, L. Moller, M. Glesner, Towards a unique FPGA-based identification circuit using process variations, in *International Conference on Field Programmable Logic and Applications* (IEEE Press, New York, 2009), pp. 397–402

[18] S. Morozov, A. Maiti, P. Schaumont, An analysis of delay based PUF implementations on FPGA, in *6th International Symposium on Applied Reconfigurable Computing*. LNCS, vol. 5992 (2010), pp. 382–387

[19] A. Maiti, J. Casarona, L. McHale, P. Schaumont, A large scale characterization of RO-PUF, in *Proceedings of the International Workshop on Hardware-Oriented Security and Trust* (HOST) (IEEE, New York, 2010), pp. 94–99