Received July 31, 2020, accepted August 27, 2020, date of publication September 2, 2020, date of current version September 16, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3021205

Configurable Ring Oscillator PUF Using Hybrid Logic Gates

DING DENG, SHEN HOU, ZHENYU WANG, AND YANG GUO
School of Computer, National University of Defense Technology, Changsha 410073, China

Corresponding author: Ding Deng (dengding15@nudt.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61832018.

ABSTRACT Physical unclonable function (PUF), a hardware that can extract the differences of the same implementations and provide unique secret keys without the utilization of non-volatile memory, is regarded as a promising security primitive in the near future. Ring Oscillator (RO) PUF is one of its easy silicon implementations, which exploits the frequency difference between a pair of structurally identical ring oscillators. However, a large number of ROs must be constructed if multiple stable output bits are required, which means unacceptable area overhead for lightweight applications. To solve this problem, configurable ROs using multiplexers and different delay units were proposed in previous papers. Unfortunately, most of them take advantage of the specific structure of a certain type of field-programmable gate arrays (FPGAs), thus not cost-saving for application-specific integrated circuit (ASIC). In this paper, we propose a configurable RO using only two hybrid logic gates in each stage for ASIC, which costs less area and power compared with previous proposals. Experiment on 50 FPGAs and one self-designed printed circuit board demonstrates satisfactory uniformity and uniqueness of this novel RO PUF. Furthermore, our proposal is proved to be reliable in a wide variety of environment conditions.

INDEX TERMS Configurable ring oscillator, hardware security, physical unclonable function.

I. INTRODUCTION

With more and more semiconductor devices are interconnected to form a huge network (aka Things Of Internet, IOT), hardware security has become an increasingly important concern in the past decade. Considering that if counterfeit or malicious chips are extensively employed in critical system or personal mobile devices, enormous threat may be brought to public health and personal privacy. It was reported that even the U.S. Department of Defense has been deceived into purchasing more than one million counterfeit electronic devices [1], not to mention common consumers. In this light, device authentication is a key issue worth studying.

Conventional authentication methods rely on secret keys stored in the non-volatile memory (NVM). However, it has been proved that the NVM can be easily tampered and cloned [2]. Although tamper-proof and tamper-resistant package can alleviate this harm to a certain extent, their formidable cost limits their popularity in lightweight applications. To address these drawbacks, physical unclonable function (PUF), a hardware that can extract the differences of the same implementations and provide unique secret keys without the utilization of NVM [3], becomes increasingly popular. PUF can be considered as a function that maps a m-bit input (i.e., Challenge) to a n-bit output (i.e., Response) in a device-specific manner. Because this function varies with the uncontrollable manufacturing variations, it cannot be cloned. And owing to the randomness and unpredictability of the manufacturing variations, the challenge-response pairs (CRPs) vary from chip to chip, which enables each device to be uniquely authenticated [4]. With all these merits, PUF has been widely used in security-related applications, such as key generation, intellectual property (IP) protection and counterfeit prevention [5].

The conception of PUF was first introduced by Pappu in 2001 [3]. Since then, several PUF designs have been proposed. According to the randomness sources, PUF can be classified as extrinsic PUF and intrinsic PUF. The Optical PUF [3] and the Coating PUF [6] belong to the former class, whose variations are introduced manually and explicitly during manufacturing. Whereas, more implementations

The associate editor coordinating the review of this manuscript and approving it for publication was Christian Pilato.

belong to the latter class [7]–[16], which make use of the natural randomness coming from the parameter deviation and mechanical mismatch [8].

Among intrinsic PUF, Arbiter PUF (APUF) is the first silicon PUF that uses an arbiter to compare the delay of two identical paths [7]. However, it is hard to achieve a symmetry route to guarantee good uniqueness, especially in the field programmable gate arrays (FPGAs). To relieve the routing stress, Ring Oscillator (RO) PUF came up as an FPGA-friendly PUF design that compares the oscillating periods instead of a single path delay [9]. Nevertheless, RO PUF offers much fewer response bits than the APUF of the same area. More response bits (i.e., CRPs) means longer service lifetime in terms of authentication times. That is to say RO PUF have to cost more area than the APUF to collect the same number of CRPs. Chen *et al.* proposed an even-stage RO PUF, named Bistable Ring PUF, which compares the rising path delay with the falling one. Unfortunately, it may take a long time before the responses become stable [10].

Besides the delay-based PUF mentioned above, memory-based PUF is another important PUF category that utilizes the existing memory on chip to generate identifier. Static random access memory (SRAM) PUF is the most typical one that uses the uncertain initial state when powering up as the randomness resource [11]. However, it is costly to power down and up when authentication is required during work mode. To this end, Butterfly PUF [12] and Buskeeper PUF [13] replace the memory unit with the D flip-flop and buskeeper respectively. Besides, Memristor PUF [14], Magnetic RAM PUF [15] and Resistance RAM PUF [16] were proposed successively. However, the number of CRPs for most memory-based PUF is only linear with the amount of memory units [11], [12].

Although so many PUF structures have been presented, all of them are troubled with two issues: unreliability and predictability. The unreliability is ascribed to the fact that all device parameters are sensitive to the environment more or less. If the environment (e.g., temperature and voltage supply) changes seriously, it is hard to guarantee all CRPs unchanged. Error correction code (ECC) techniques (e.g., BCH code [17] or fuzzy extractor [18]) and fault-tolerant techniques (e.g., majority voting [19]) are widely used to improve the reliability. The predictability is mainly attributed to the correlations between the CRPs, which accompanies with the specific hardware structure or unrealized measurement dependence. For example, the path delay of APUF is found to be perfectly fitted by linear additive model [20], thus is easily attacked by machine learning techniques [21]–[23]. Among all silicon intrinsic PUFs, RO PUF is expected to be a good choice that can achieve a good tradeoff between various metrics [9].

II. BACKGROUND

A. RO PUF

As Fig. 1 shows, the traditional RO PUF consists of N ROs, two N-to-1 multiplexers, two counters and one comparator.

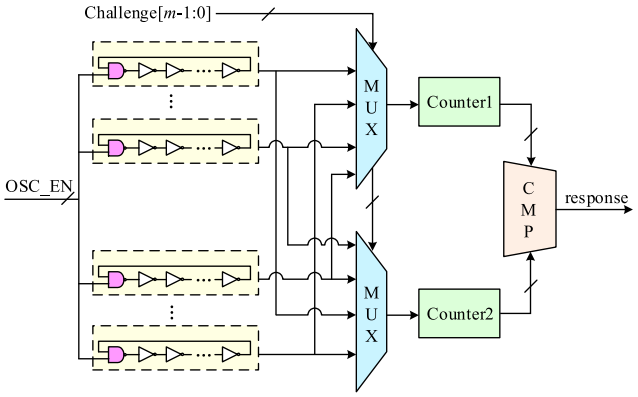


FIGURE 1. Structure of traditional RO PUF.

Each RO is composed of one NAND gate and even number of inverters. The m-bit challenge chooses two different ROs through the multiplexers. Once the oscillation enable (OSC_EN) signal rises up, the selected ROs start to oscillate and drive the subsequent counter to calculate the number of oscillation cycles. After a period of t (aka measuring period), pull down the OSC_EN, thus making the ROs stop running simultaneously. The comparator compares the count values of the two counters. Although all ROs are structurally identical, their frequencies are different because of the manufacturing variation. Hence, if the top RO is faster, the comparator outputs 1 as the response bit, otherwise it outputs 0. With n different challenges applied in the same manner, a n-bit response can be generated.

B. MOTIVATION

To strengthen the reliability of the RO PUF, Suh proposed a 1-out-of-8 mask scheme that only chooses the two with the largest frequency difference out of eight ROs to generate one response bit [9]. Obviously, this method leads to a low hardware utilization rate. To defeat against the prediction of CRPs, several frequencies provision of the same RO can be helpful to some extent. Both of these two observations promote the researchers to expand the frequency choices with the same number of ROs. As a result, configurable RO was widely developed and used in RO PUF.

Previous configurable RO are constructed by replacing every stage logic (i.e., the single inverter) with a configurable stage unit (CSU) that possesses two or more delay options, just as Fig. 2 shows. Maiti took a full advantage of the configurable logic block (CLB) in FPGA to build a RO with at most 8 frequency configurations [24]. Inspired by that, Xin extended the number of configurations to 256 for each RO with the same area overhead [25]. Moreover, Habib exploited the delay configurability of the look-up table (LUT) to change the frequency [26]. However, all of these three designs are based on the unique structure of a specific type of FPGA, which is not efficient enough to be applied to application-specific integrated circuit (ASIC). Sharif proposed a configurable RO PUF for ASIC, which can change frequency by

altering different voltage supplies for each stage [27]. However, the power network of this structure needs specialized design which makes it hard to be integrated into a single chip with other functional parts. Gao and Cao also employed the idea of frequency configurability in their proposals [28], [29].

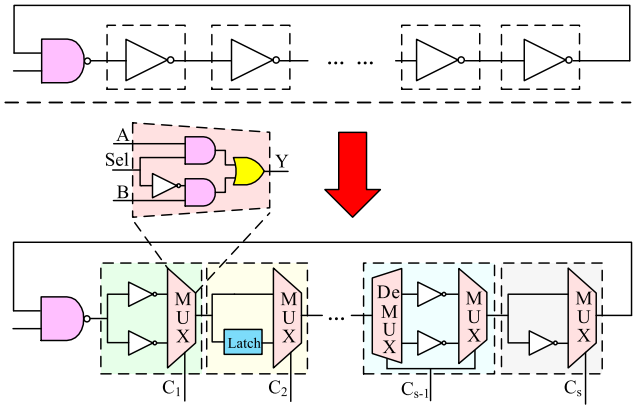


FIGURE 2. Traditional configurable ROs.

Fig. 3 presents the four kinds of CSUs that can be applied to ASIC in foregoing literatures. For all these CSUs, the configurable input (CI) can choose whether the top path or the bottom path is used as a stage in the RO. As we can see, the CSU-3 has an extra demultiplexer compared to CSU-1, which can prevent infructuous transitions in the unselected gate, thus saving power. However, it increases the area overhead. The difference between CSU-1/CSU-3 and CSU-2/CSU-4 is that one path of the CSU-2 and CSU-4 is pure wire. This kind of design saves area but also reduces the total number of configurations. It is because that at least two stages should be changed simultaneously to guarantee odd number of reverse logic gates in the whole chain. It is easy to find that all these CSUs at least requires one multiplexer and two alternative parts. Generally, merely a multiplexer consists of four logic gates, as shown in Fig. 2. Thus, there still seems to be some room for area saving.

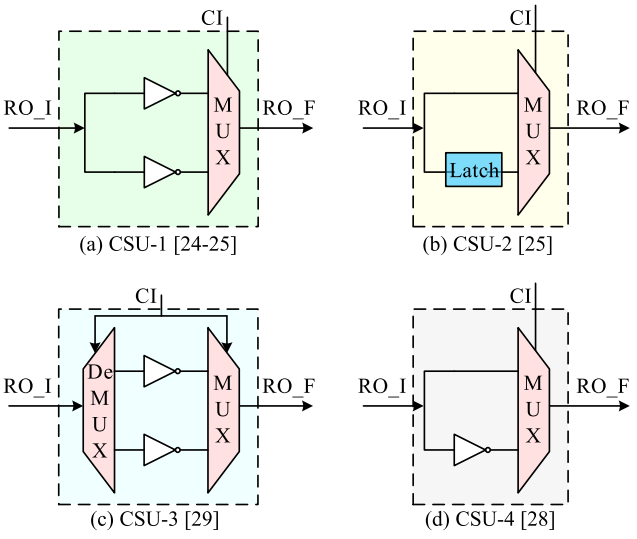


FIGURE 3. Structure of traditional configurable stage unit.

C. OUTLINE

The major contributions of this paper can be concluded as follows:

- We propose four types of delay configurable units (DCUs) that consists of only two combinational logic gates, without the utilization of multiplexer.
- We make a comprehensive characterization of the configuration RO PUF that composed of the proposed DCUs on 50 FPGAs to demonstrate its validity.
- We give some potential applications of these DCUs.

The rest of this paper is organized as follows. In Section III, we present the configurable RO PUF with hybrid logic gates in detail. In Section IV, we demonstrate the qualities of our proposal with sufficient experimental results. In Section V, we discuss two other occasions where the proposed DCUs can be used. Finally, we conclude the whole paper in Section VI.

III. HYBRID CONFIGURABLE RO

A. HYBRID DELAY CONFIGURABLE UNIT

In our previous paper [30], we proposed four types of stimulating units to detect the hardware Trojan. In this paper, we explain how they can also be used as delay configurable unit (DCU) to form a configurable RO. As shown in Fig. 4, each type of DCU has two input ports (i.e., RO_I and CI) and one output port: RO_F. CI provides access to configure input while RO_I and RO_F are used for oscillator connection.

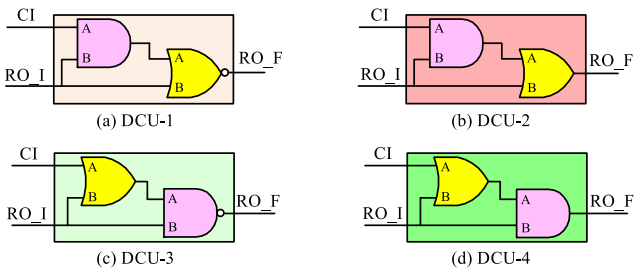


FIGURE 4. Structure of proposed delay configurable unit.

The CI signal can configure the rising time or falling time of the path from RO_I to RO_F, notated with D_r and D_f respectively. Let us define the propagation time (D_p) of one DCU as the mean of the rising time and the falling time (i.e., $D_p = (D_r + D_f)/2$), then we can draw the following conclusions:

- For DCU-1 and DCU-4, CI can adjust the D_r.
- For DCU-2 and DCU-3, CI can adjust the D_f.
- For DCU-1 and DCU-2, the D_p when CI=1 is larger than the one when CI=0.
- For DCU-3 and DCU-4, the D_p when CI=0 is larger than the one when CI=1.

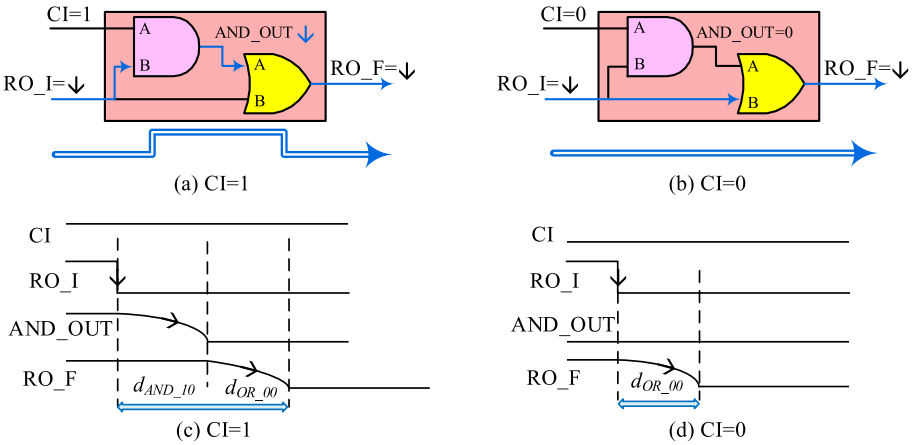


FIGURE 5. Influence of CI on the falling time of DCU-2.

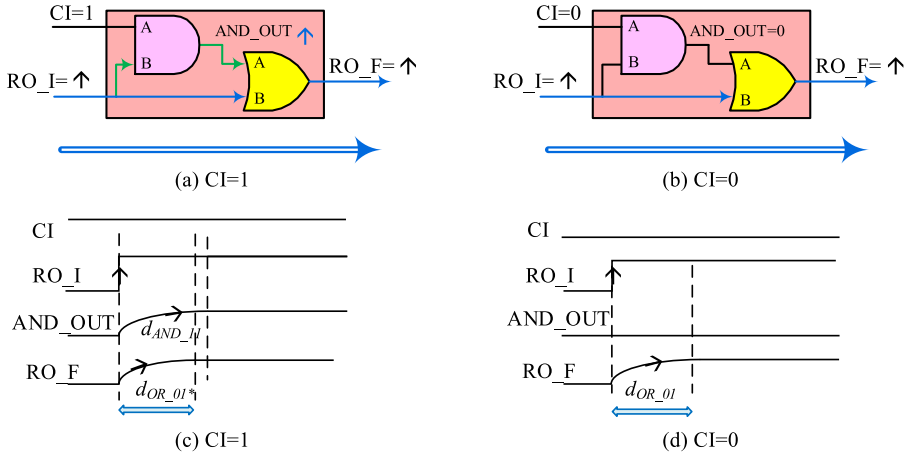


FIGURE 6. Influence of CI on the rising time of DCU-2.

To explain the influence of CI on the D_r/D_f of the DCU, let us firstly use d_{AND_10} to denote the falling time of an AND gate when the first input (referred as 'A' input) of it is 1 while the second input (referred as 'B' input) is 0. In the same manner, we can use d_{OR_01} to denote the rising time of an OR gate when its 'A' input is 0 while 'B' input is 1. With these denotations, we can take the influence of CI value on the propagation time of DCU-2 as an example.

For the falling time, let us assume RO_I is 1 at the beginning, thus leading RO_F to be 1 no matter what value CI holds. When RO_I pulls down, the 'B' input of the OR gate will pull down. Also, the output of the AND gate (i.e., AND_OUT) will pull down, no matter what value CI holds. That is to say the 'A' input of the OR gate will also pull down. Hence, the RO_F will always follow the RO_I to pull down. However, the falling time from the RO_I to RO_F is not the same for different CI values:

1) When CI = 1 (as shown in Fig. 5(a), Fig. 5(c)), the initial value of the AND_OUT is 1. So when RO_I pulls down, it must take a time of d_{AND_10} to pull down the AND_OUT firstly. After that, it will take another time of d_{OR_00} to pull down the RO_F. In other words, it takes a total time of $(d_{AND\_10} + d_{OR\_00})$ for RO_F to become 0 after RO_I pulls down.
2) When CI = 0 (as shown in Fig. 5(b), Fig. 5(d)), because the initial value of AND_OUT is already 0, it only takes a time of d_{OR_00} for RO_F to become 0 after RO_I pulls down.

In conclusion, if the DCU-2 is used as a stage of the RO, the value of CI can decide whether an AND gate is included in the path during falling phases.

For the rising time, let us assume RO_I is 0 at the beginning, thus leading RO_F to be 0 no matter what value CI holds. when RO_I pulls up, the 'B' input of the OR gate will also pull up. Hence, the RO_F will certainly follow RO_I to pull up, no matter what value AND_OUT is. However, the rising time from the RO_I to RO_F varies a little for different CI values:

1) When CI = 1 (as shown in Fig. 6(a), Fig. 6(c)), the initial value of the AND_OUT is 0. So when RO_I pulls up, the AND_OUT (i.e., the 'A' input of the OR gate) is rising during the period of d_{AND_11}.

TABLE 1. Influence of CI on the rising/falling time of DCUs.

DCU	CI = 0		CI = 1	
	D_r	D_f	D_r	D_f
DCU-1	d_{NOR_00}	d_{NOR_01}	d_{AND_10} $+d_{NOR\_00}$	$*d_{NOR\_01}$
DCU-2	d_{OR_01}	d_{OR_00}	$*d_{OR\_01}$	d_{AND_10} $+d_{OR\_00}$
DCU-3	$*d_{NAND\_10}$	d_{OR_01} $+d_{NAND\_11}$	d_{NAND_10}	d_{NAND_11}
DCU-4	d_{OR_01} $+d_{AND\_11}$	$*d_{AND\_10}$	d_{AND_11}	d_{AND_10}

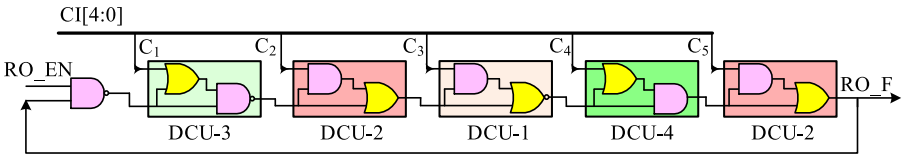


FIGURE 7. An example of hybrid configurable RO.

2) When CI = 0 (as shown in Fig. 6(b), Fig. 6(d)), the AND_OUT always holds 0.

Hence, it will take an exact time of d_{OR_01} for RO_F to pull up when CI = 0 while a little shorter time than d_{OR_01} (denoted as $*d_{OR\_01}$) when CI = 1.

In summary, the propagation time of DCU-2 is $(*d_{OR\_01} + d_{AND\_10} + d_{OR\_00})/2$ when CI = 1 while $(d_{OR\_01} + d_{OR\_00})/2$ when CI = 0. Similar analysis can be applied to the other three DCUs. The delay configurability of CI for all these four DCUs are listed in Table 1.

B. CONFIGURABLE RO

Based on the proposed DCUs, we can construct a hybrid configurable RO (HC-RO) with a NAND gate as the leading unit. Meanwhile, the following two rules must be abided by:

- The RO_I of current DCU should connect to the RO_F of the previous DCU and the RO_F of current DCU should connect to the RO_I of the next DCU.
- The total number of reverse logic units in a HC-RO should be odd, which includes the leading NAND gate, DCU-1, DCU-3 and inverter if exists.

Fig. 7 shows an example of our hybrid configurable RO. As we can see, a NAND gate is placed at the beginning of the RO and the total number of reverse logic units is three (i.e., one NAND gate, one DCU-3 and one DCU-1), exactly an odd number. When RO_EN is low, RO_F holds high all the time. Once RO_EN turns to high, the whole RO can oscillate at $2^5 = 32$ kinds of frequencies determined by the CI[4:0]. Of course, from the perspective of area cost, it would be better to use DCU-1 or DCU-3 as the stage unit.

IV. EXPERIMENTAL RESULTS

A. HARDWARE OVERHEAD

Note that the target of our paper is to design a cost-saving configurable RO for ASIC, so we think it would be convincing only by comparing with the previous configurable RO. However, the configurable ROs in [24] and [25] are implemented in FPGA, the configurable ROs in [27], [29] and [31] are implemented in different process library (90*nm*, 65*nm* and 40*nm* respectively) and have been full-customized in the voltage supply, inverter, or temperature compensation circuit elaborately. [10] even does not give the hardware overhead. Worse still, the number of ROs, the number of stages for each RO, the number of configurations for each stage, the multiplexer and the counter are also different in previous literatures. Thus, we decide to implement some of them by ourselves, using the same process library, the same multiplexer and counter without any other special design or customized optimization. Obviously, the stage unit in [10] which needs two NOR gates, one de-multiplexer and one multiplexer or the stage unit in [25] which needs a latch and a multiplexer occupies more area than our proposal. Hence, we only compare our proposal with the previous configurable ROs that use the CSU-1 [24], [25] or CSU-4 [28] as the stage unit.

Elaborated in TSMC 28*nm* high performance library with the Design Compiler of the Synopsys company, our proposal (referred as HC) is compared with the traditional unconfigurable ROs (referred as ALI) and other two configurable ROs (referred as DPI and SPI respectively) in terms of area and power consumption. All the ROs in these four PUFs consist of one NAND gate and 14 stage units. Every stage unit for ALI, DPI and HC is inverter, CSU-1 [24], [25] and DCU-3 respectively. For SPI, half of the 14 stage units are CSU-1s while the other half are CSU-4s [28]. The reason why we do not construct a RO whose stage units are all CSU-4s but instead only half stages are CSU-4s and another half are CSU-1s is that if most stages choose the pure wire path of CSU-4, the oscillating frequency may surpass the maximum allowable speed of the subsequent counters. The overall architecture for each estimated PUF, which does not include the controlling part, is shown in Fig. 8.

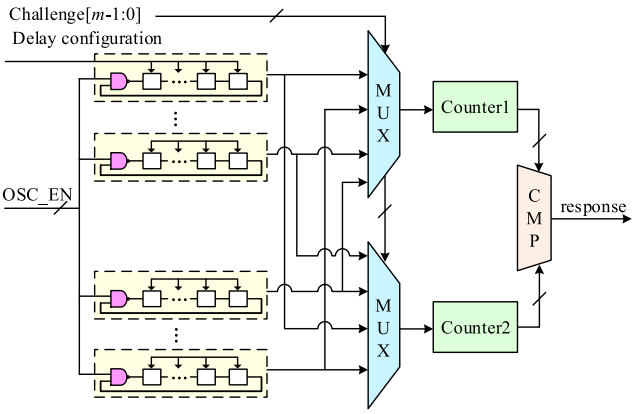
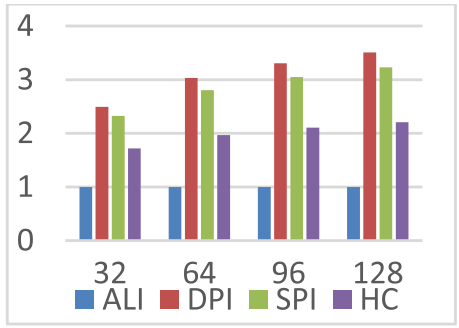
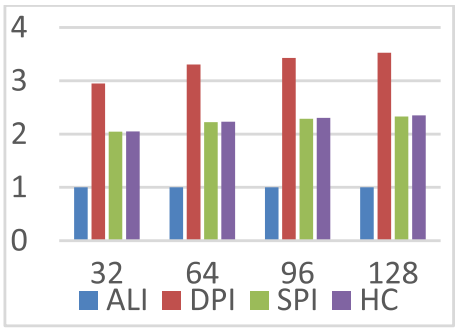


FIGURE 8. Architecture of the estimated PUF.



(a) Area



(b) Power

FIGURE 9. The relationship between the number of ROs and the hardware overhead normalized by the unconfigurable RO whose stage units are all inverters (referred as **ALI**). **DPI** represents the configurable RO whose stage units are all CSU-1s [24], [25]. **SPI** represents the configurable RO whose half stage units are CSU-1s [24], [25] and the other half are CSU-4s [28]. **HC** represents our proposal whose all stage units are DCU-3s.

Fig. 9 shows the relationship between the number of ROs and the overhead of these four PUF designs normalized by the traditional unconfigurable RO PUF (i.e., ALI). As Fig. 9(a) shows, our proposal (i.e., HC) occupies the smallest area among the three configurable RO PUFs, which is only about 2 times larger than the traditional RO PUF on average. Our proposal saves about 35.19% and 29.90% area on average compared with DPI and SPI respectively. As for the power consumption shown in Fig. 9(b), our proposal can save about 32.34% power on average over the DPI but consumes a little more power than the SPI. The reason why SPI can consume less power than our proposal is just because it can have no gates lying on the selected path for some stages. However, the SPI only has $2^7 \times (\binom{7}{1} + \binom{7}{3} + \binom{7}{5} + \binom{7}{7})$ configurations, rather than 2^{14} configurations as the DPI or HC possesses.

To make the comparison of hardware overhead more comprehensive, we also compare our proposal with some other works in Table 2, which are also implemented in ASIC but using different process technologies with different number of ROs, stages, configurations, counters or full-customized circuits. The data of our proposal in this table is from a design containing 32 HC-ROs, each of which has one NAND as the header and 14 DCU-3s. As we can see, the hardware overhead of our proposal is in the middle level.

B. PUF QUALITY

The quality of PUF is usually estimated by three widely used indicators:

- Uniformity: It represents the ratio of 1s in all responses of a device, which can be calculated as (1):

$$Ufm = \frac{1}{p}\sum_{l=1}^{p} r_l \times 100\% \tag{1}$$

  Here, p is the total number of responses that one PUF can generate.
- Uniqueness: It represents the ability of a PUF to distinguish different devices, which can be estimated by the inter-die hamming distance (HMD) as shown in (2):

$$Unq = \frac{2}{b(b-1)} \times \sum_{i=1}^{b-1}\sum_{j=i+1}^{b} \frac{\mathrm{HMD}(R_i, R_j)}{n} \tag{2}$$

  Here b is number of measured chips. R_i, R_j represent the n-bit response of a same challenge for chip i and chip j respectively.
- Reliability: It represents the ability of a PUF to reproduce a response under different conditions. It can be estimated by intra-die HMD as calculated in (3):

$$Rlb = \frac{1}{k}\sum_{i=1}^{k} \frac{\mathrm{HMD}(R_i, R_g)}{n} \tag{3}$$

  Here k is the total number of measured times for a same challenge applied to a same chip. R_g is the golden n-bit response and R_i is the i^{th} measured n-bit response of the same challenge for the same chip under different conditions.

For a perfect PUF design, the idea value of uniformity, uniqueness and reliability is 50%, 50% and 0%, respectively.

To demonstrate the delay configurability of our proposal, we constructed a 13-stage HC-RO, which consists of 1 AND as the header, 3 DCU-1s, 3 DCU-2s, 4 DCU-3s and 3 DCU-4s with the TSMC 28*nm* digital cell library. With all delay configurable inputs choosing the long path (i.e., $CI = 1$ for

TABLE 2. Hardware comparison of our proposal with other PUF designs.

	ISSCC'07 [32]	VLSI'10 [33]	JSSC'11 [34]	ICCD'12 [27]	TCAD'15 [29]	ISSCC'15 [31]	VLSI'17 [35]	**This work**
Process(nm)	130	65	90	90	65	40	130	**28**
Area(μm^2)	15288	1242	35000	289	250	845	44700	**1128**
Power(μW)	0.93	212.5	38	81.1	32.3	28.4	0.068	**119.4**
#CRPs	1	1	10^{25}	2^{22}	256	5.5×10^{28}	3.7×10^{19}	**8126464**

DCU-1/DCU-2 and $CI = 0$ for DCU-3/DCU-4), the oscillating frequency is 1.745GHz. While when all delay configurable inputs choose the short path (i.e., $CI = 0$ for DCU-1/DCU-2 and $CI = 1$ for DCU-3/DCU-4), the oscillating frequency becomes 2.568GHz, meaning that the oscillating period becomes 183.65ps larger, exactly about 13 delay of AND/OR gate as expected. Hence, we can say that our HC-RO is valid.

To simulate the performance of our HC-RO when employed in RO PUF, we firstly tried Monte Carlo simulation, but it was extremely time-consuming and we failed to save the unique hardware model in each run for different configurations. Thus, we have to build up a coarse model by ourselves and do simulation in MATLAB, just as the authors of [22], [36] did. As analyzed in Section III-A, our configurable RO can also be constructed as a linear additive model according to the delay information listed in Table 1. As TSMC 28nm library shows, the rising time of each logic gate is almost equal to its falling time, with only a little difference when its inputs hold different values (e.g., $d_{OR\_01} \approx *d_{OR\_01} \approx d_{OR\_00} = d_{OR}$). Besides, we find the delay of one NAND gate is nearly equal to that of one NOR gate, so for the case of AND gate and OR gate (i.e., $d_{NAND} \approx d_{NOR}$, $d_{AND} \approx d_{OR}$). Hence, we can approximately model the delay of one DCU with only two parameters (i.e., the delay of the first gate t_1 and the delay of the second gate t_2).

To simulate the uniformity and uniqueness, we build 50 different chip models, each of which has 128 HC-ROs of the same structure (i.e., 1 NAND and 64 DCU-3s). The delay for all gates of the same class is independent and identically distributed and follows a normal distribution with $\sigma = \alpha \cdot \mu$, here σ is the manufacturing variation and μ is the expected delay of corresponding gate. α is called the manufacturing variation coefficient. In this experiment, we set $\alpha = 0.1$ (for different α, the simulation result is similar) and randomly selected 20000 delay configurations to achieve $20000 \times 128/2 = 1.28 \times 10^6$ CRPs for each chip. To simulate the reliability, considering that it is hard to mimic the local hotspot (i.e., different temperature coefficient for some transistors) or the IR-drop effect as well as the measuring noise, for simplicity we introduce a parameter β to represent the environmental mismatch σ_e, where $\sigma_e = \beta \cdot \sigma_m$, ($\sigma_m = \sigma$ is the manufacturing variation). We randomly choose one out of the 50 chips and add the random environmental mismatch which follows a normal distribution $N(0, \sigma_e)$ to the existing model. 30 evaluations are repeated and the average results are listed in Table 3. From this table, we can see that the uniformity and uniqueness of our proposal are 49.61% and 49.95% respectively, very close to the ideal value (i.e., 50%). The reliability deteriorates from 0.87% to 8.6% as the environmental mismatch coefficient β increases from 0.1 to 1, which is also acceptable.

TABLE 3. Simulation of the PUF quality.

Uniformity (%)	Uniqueness (%)	Reliability (%)		
		$\beta = 0.1$	$\beta = 0.5$	$\beta = 1.0$
49.61	49.95	0.87	4.29	8.60

Due to the formidable cost of manufacturing ASIC, we further implement our proposal on FPGAs to get a more trustworthy result. To evaluate the uniformity and uniqueness, we implemented our proposal in 50 Basys3 boards each of which is equipped with a Xilinx Artix-7 FPGA (xc7a35t1cpg236c, 28nm). Each PUF consists of 128 16-stage HC-ROs, which are identically placed and routed with TCL script. We recorded the responses of 120 pairs of ROs under 4096 randomly chosen delay configurations, each of which was conducted in room temperature, about 26°C. Fig. 10 shows the uniformity for each FPGA. As this figure displays, the ratio of the response 1s is 50.36% on average, which is extremely close to the ideal value (i.e., 50%).

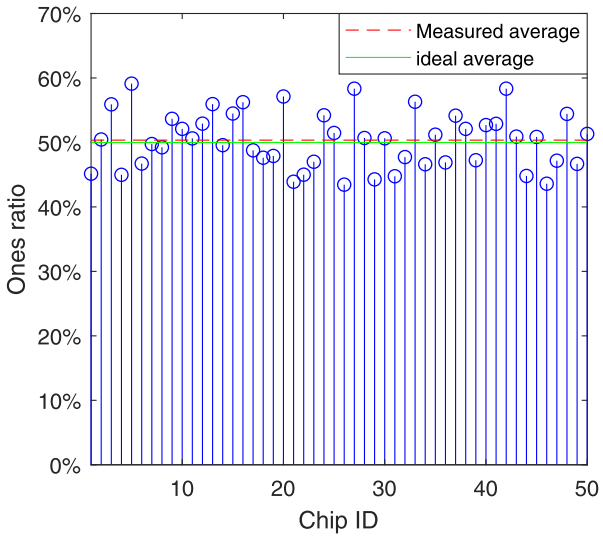


FIGURE 10. Uniformity for each chip.

Fig. 11 shows the inter-die HMD for the 120-bit responses over all the 4096 delay configurations across the 50 chips. As we can see, the occurrence histogram of the inter-

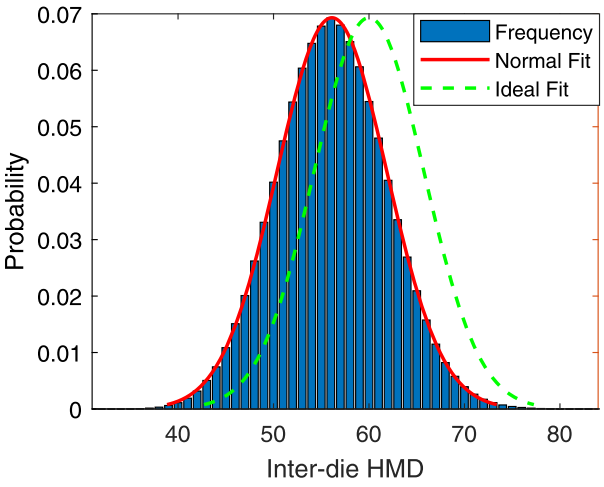


FIGURE 11. Inter-die HMD for 120-bit response among 50 FPGAs.

die HMD is in a perfect bell shape. Fitted with a normal distribution, we can get the probability density curve (red line) characterized with parameter $\mu = 56.1168$, which is very close to the counterpart of the ideal curve (i.e., $\mu = 120 \times 50\% = 60$) as the green line shows.

To estimate the reliability of temperature, we measured one of the 50 Basys3 boards in a thermostat (5°C, 15°C, 50°C, 60°C, 70°C) with the CRPs collected in the last experiment (i.e., 26°C) as the reference. Because the core voltage is fixed by a voltage regulator chip (LTC3633) in Basys3 board, we additionally designed a printed circuit board (PCB) embedded with a Xilinx spartan-3 FPGA (xc3s400pq208, 90*nm*) to estimate the reliability of voltage (1.16V, 1.18V, 1.20V, 1.22V, 1.24V, with 1.2V as the reference). Fig. 12(a) and Fig. 12(b) show the experimental setup for temperature and voltage variation respectively. Fig. 13 shows the intra-die HMD statistics for 128 delay configurations in different environments. As we can see, the average intra-die HMD for different temperatures and voltages are 1.78% and 0.44% respectively, which are both very close to the ideal value (i.e., 0%).

Fig. 14 gives the HMD distribution of all the 120-bit responses under the five temperatures (480 measurements for each delay configuration under every temperature). As we can see, the worst case has at most 9-bit difference with the golden one. Most cases only have one or two bit errors.

(a) Setup for temperature variation

(b) Setup for voltage variation

FIGURE 12. Measurement setup for reliability evaluation.

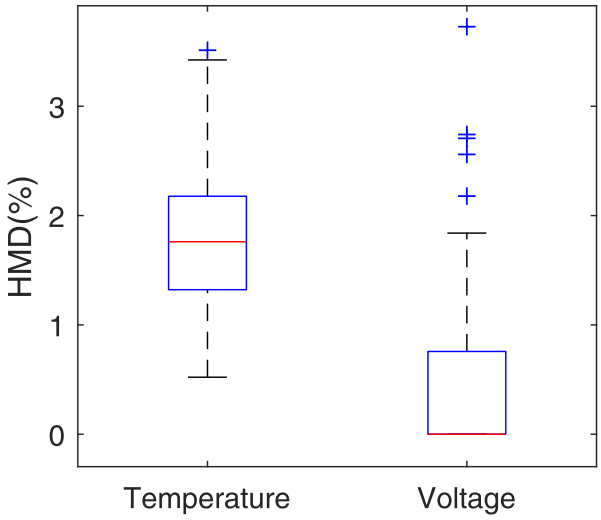


FIGURE 13. Intra-die HMD for different temperatures and voltages.

V. DISCUSSION

A. SECURITY

The security of PUF is usually estimated by the number of independent response bits. For the basic APUF, it can ideally at most generate s response bits for a pair of paths, where s is the number of stages for each path. For the traditional RO, it can ideally at most generate $\log_2(N!)$ bits, where N is the number of ROs [9]. For our proposal, if the top RO and the bottom RO of the chosen pair have the same delay configuration, each pair of ROs can at most generate s response bits, where s is the number of configurable stages for each RO. That is to say, a pair of our hybrid configurable ROs can generate comparable magnitude of response bits as the APUF. However, we have to admit that our configurable RO also fits the linear additive model, just like the APUF.

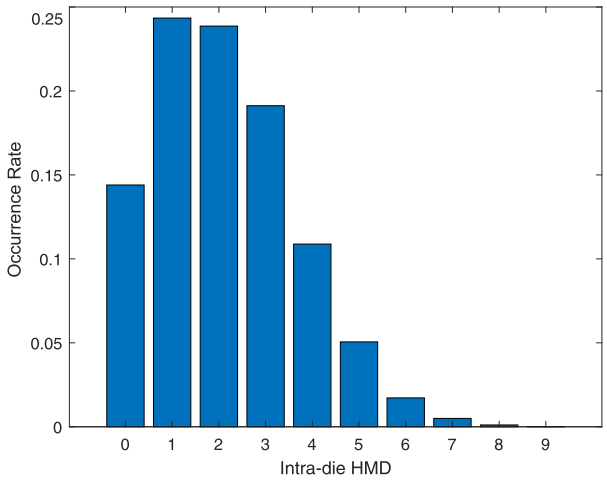


FIGURE 14. Intra-die HMD for 480 × 5 = 2400 measurements under different temperatures.

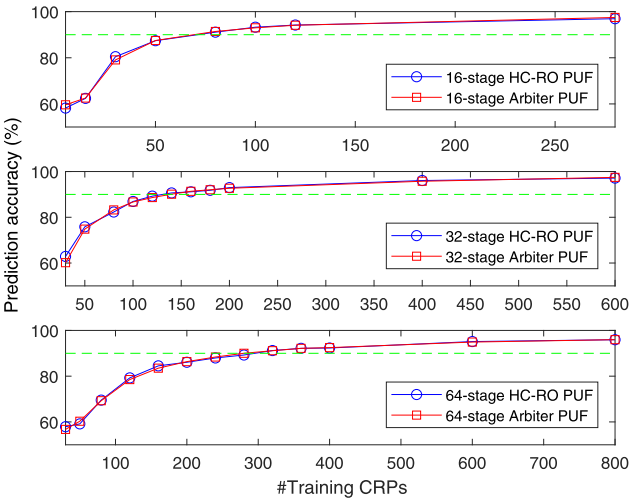


FIGURE 15. Security comparison between the APUF and our proposal.

Hence, this configurable RO PUF can also be attacked by the machine learning method [21]–[23].

To evaluate the security of our HC-RO PUF, we reuse the simulation model we build in Section IV-B and the Arbiter PUF model proposed in [22], [36]. For fairness, we adopt the same manufacturing variation coefficient $\alpha = 0.1$ for both APUF and our configurable RO PUF. (Other values for α have also been tried and the results are almost the same.) We construct 10 APUFs and 10 pairs of HC-ROs of 16, 32, 64 stages respectively. A specified number of CRPs are randomly chosen as the training set and 200 other CRPs are used as the testing set. Logistic regression (LR) is used as the attacking method since it has been proved to be one of the most efficient machine learning attacks for the linear additive model such as the APUF [36].

Fig. 15 shows the security comparison of an APUF and a pair of our HC-RO, each point of which represents the average prediction accuracy over 10 different runs. As we can see, with more CRPs used for training, the prediction accuracy becomes higher for both APUF and our HC-RO PUF. The prediction curves for the APUF and our HC-RO PUF are almost the same, which coincides with our conclusion that our proposal can match APUF in terms of security. From this figure, we can also find that only about 70, 140 and 280 CRPs are needed to obtain a prediction accuracy of 90% for the APUF or HC-RO PUF of 16, 32 and 64 stages respectively. Thus, nonlinearity and obfuscation improvement must be made on the circuit or protocol level to enhance the security [37], [38], but it is out of the scope of this paper. The major target of this paper is to design a cost-saving RO specially suitable for ASIC compared with previous configurable ROs.

B. POTENTIAL USE

Besides the use for the construction of configurable RO, our DCUs can also be used in other applications such as the path delay balance or the compensation of system variation. In [19], Majzoobi used the programmable delay lines (PDL) to balance the path bias of the APUF implemented in FPGA. PDL is a fine-grained adjuster benefitting from the special structure of the LUT. Similarly, we can use our DCUs as a coarse-grained adjuster to tune and remove the bias delay differences caused by asymmetries in net routing in ASIC, as shown in Fig. 16. Not limited to the APUF, we can also extend this idea to RO PUF. As Fig. 17 shows, we can construct ROs consisting of $(s+q)$ configurable stages, where s stages are applied to the same configurations that are used as challenges while q stages are assigned to different values to compensate the intra-die system variation. Furthermore, our DCUs can also be used in the crossover RO PUF to enhance its configurability [39]. This is also part of our future work.

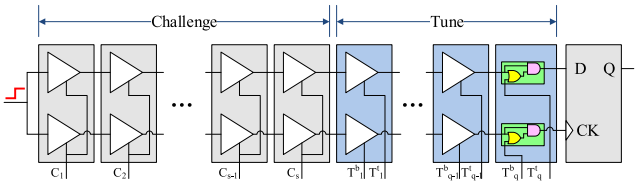


FIGURE 16. DCUs used for the path delay balance.

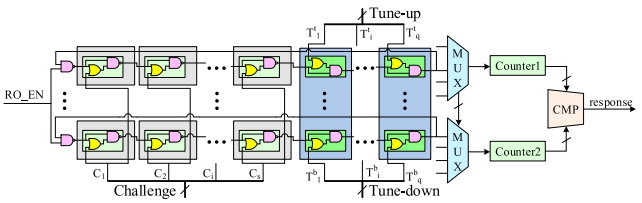


FIGURE 17. DCUs used for the compensation of system variation.

VI. CONCLUSION

In this paper, we propose four kinds of delay configurable units only using two logic gates, instead of inverters or multiplexers. With these units, a novel configurable RO PUF can

be constructed with more delay configurations at less area and power cost than previous counterparts. Experimental results on 50 FPGAs show the uniformity and uniqueness of this new RO PUF is 50.36% and 46.76% on average respectively. The intra-die HMD is 1.78% for 5°C∼70°C and 0.44% for 1.16V∼1.24V (i.e., 1.2V∓3.33%). More security improvements need to be proposed to strengthen the resistance to the advanced machine learning attack, such as the approximation attacks [40].

REFERENCES

[1] *Inquiry Into Counterfeit Electronic Parts in the Department of Defense Supply Chain*, USA Committee, Senate Rep., 2012, pp. 112–167.
[2] S. P. Skorobogatov, "Semi-invasive attacks-a new approach to hardware security analysis," Comput. Lab., Univ. Cambridge, Cambridge, U.K., Tech. Rep. 04 2005, 2005.
[3] P. Ravinkanth, "Physical one-way functions," Ph.D. dissertation, MIT, Cambridge, MA, USA, 2001.
[4] R. Maes, *Physically Unclonable Functions: Constructions, Properties and Applications*. Cham, Switzerland: Springer, 2016.
[5] Y. Cao, C. Q. Liu, and C. H. Chang, "A low power diode-clamped inverter-based strong physical unclonable function for robust and lightweight authentication," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 65, no. 11, pp. 3864–3873, Nov. 2018.
[6] P. Tuyls, G.-J. Schrijen, B. Škorić, J. Van Geloven, N. Verhaegh, and R. Wolters, "Read-proof hardware from protective coatings," in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst.* Cham, Switzerland: Springer, 2006, pp. 369–383.
[7] J. W. Lee, D. Lim, B. Gassend, G. E. Suh, M. van Dijk, and S. Devadas, "A technique to build a secret key in integrated circuits for identification and authentication applications," in *Symp. VLSI Circuits. Dig. Tech. Papers*, Jun. 2004, pp. 176–179.
[8] A. Chandrakasan, W. J. Bowhill, and F. Fox, "Models of process variations in device and interconnect," in *Design of High-Performance Microprocessor Circuits*. Piscataway, NJ, USA: IEEE Press, 2001, pp. 98–115, doi: 10.1109/9780470544365.ch6.
[9] G. E. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in *Proc. 44th ACM/IEEE Design Autom. Conf.*, Jun. 2007, pp. 9–14.
[10] Q. Chen, G. Csaba, P. Lugli, U. Schlichtmann, and U. Ruhrmair, "The bistable ring PUF: A new architecture for strong physical unclonable functions," in *Proc. IEEE Int. Symp. Hardw.-Oriented Secur. Trust*, Jun. 2011, pp. 134–141.
[11] 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—CHES*, P. Paillier and I. Verbauwhede, Eds. Berlin, Germany: Springer, 2007, pp. 63–80.
[12] S. S. Kumar, J. Guajardo, R. Maes, G.-J. Schrijen, and P. Tuyls, "Extended abstract: The butterfly PUF protecting IP on every FPGA," in *Proc. IEEE Int. Workshop Hardw.-Oriented Secur. Trust*, Jun. 2008, pp. 67–70.
[13] P. Simons, E. van der Sluis, and V. van der Leest, "Buskeeper PUFs, a promising alternative to d flip-flop PUFs," in *Proc. IEEE Int. Symp. Hardw.-Oriented Secur. Trust*, Jun. 2012, pp. 7–12.
[14] P. Koeberl, U. Kocabas, and A.-R. Sadeghi, "Memristor PUFs: A new generation of memory-based physically unclonable functions," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, 2013, pp. 428–431.
[15] J. Das, K. Scott, S. Rajaram, D. Burgett, and S. Bhanja, "MRAM PUF: A novel geometry based magnetic PUF with integrated CMOS," *IEEE Trans. Nanotechnol.*, vol. 14, no. 3, pp. 436–443, May 2015.
[16] A. Shrivastava, P.-Y. Chen, Y. Cao, S. Yu, and C. Chakrabarti, "Design of a reliable RRAM-based PUF for compact hardware security primitives," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2016, pp. 2326–2329.
[17] D. Merli, F. Stumpf, and C. Eckert, "Improving the quality of ring oscillator PUFs on FPGAs," in *Proc. 5th Workshop Embedded Syst. Secur. (WESS)*, New York, NY, USA: Association for Computing Machinery, 2010, pp. 1–9.
[18] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," in *Advances in Cryptology—EUROCRYPT*, C. Cachin and J. L. Camenisch, Eds. Berlin, Germany: Springer, 2004, pp. 523–540.
[19] S. Devadas *et al.*, "Automated design, implementation, and evaluation of arbiter-based PUF on FPGA using programmable delay lines," Rice Univ., Houston, TX, USA, Tech. Rep., 2014. [Online]. Available: https://hdl.handle.net/1911/96414
[20] U. R Ührmair, F. Sehnke, J. S Ölter, G. Dror, S. Devadas, and J. Ü. Schmidhuber, "Modeling attacks on physical unclonable functions," in *Proc. 17th ACM Conf. Comput. Commun. Secur. (CCS)*, New York, NY, USA, 2010, pp. 237–249.
[21] G. Hospodar, R. Maes, and I. Verbauwhede, "Machine learning attacks on 65nm arbiter PUFs: Accurate modeling poses strict bounds on usability," in *Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*, Dec. 2012, pp. 37–42.
[22] U. Ruhrmair, J. Solter, F. Sehnke, X. Xu, A. Mahmoud, V. Stoyanova, G. Dror, J. Schmidhuber, W. Burleson, and S. Devadas, "PUF modeling attacks on simulated and silicon data," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 11, pp. 1876–1891, Nov. 2013.
[23] G. T. Becker, "The gap between promise and reality: On the insecurity of XOR arbiter PUFs," in *Cryptographic Hardware and Embedded Systems—CHES*, T. Güneysu and H. Handschuh, Eds. Berlin, Germany: Springer, 2015, pp. 535–555.
[24] A. Maiti and P. Schaumont, "Improving the quality of a physical unclonable function using configurable ring oscillators," in *Proc. Int. Conf. Field Program. Log. Appl.*, Aug. 2009, pp. 703–707.
[25] X. Xin, J.-P. Kaps, and K. Gaj, "A configurable ring-oscillator-based PUF for xilinx FPGAs," in *Proc. 14th Euromicro Conf. Digit. Syst. Design*, Aug. 2011, pp. 651–657.
[26] B. Habib, K. Gaj, and J.-P. Kaps, "FPGA PUF based on programmable LUT delays," in *Proc. Euromicro Conf. Digit. Syst. Design*, Sep. 2013, pp. 697–704.
[27] S. S. Mansouri and E. Dubrova, "Ring oscillator physical unclonable function with multi level supply voltages," in *Proc. IEEE 30th Int. Conf. Comput. Design (ICCD)*, Sep. 2012, pp. 520–521.
[28] M. Gao, K. Lai, and G. Qu, "A highly flexible ring oscillator PUF," in *Proc. 51st Annu. Design Autom. Conf. Design Autom. Conf. (DAC)*, New York, NY, USA, 2014, pp. 89:1–89:6.
[29] Y. Cao, L. Zhang, C.-H. Chang, and S. Chen, "A low-power hybrid RO PUF with improved thermal stability for lightweight applications," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 34, no. 7, pp. 1143–1147, Jul. 2015.
[30] D. Deng, Y. Wang, and Y. Guo, "Novel design strategy towards A2 trojan detection based on built-in acceleration structure," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, early access, Feb. 28, 2020, doi: 10.1109/TCAD.2020.2977069.
[31] K. Yang, Q. Dong, D. Blaauw, and D. Sylvester, "A physically unclonable function with BER $<10^{-8}$ for robust chip authentication using oscillator collapse in 40nm CMOS," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2015, pp. 1–3.
[32] Y. Su, J. Holleman, and B. Otis, "A 1.6pJ/bit 96% stable chip-ID generating circuit using process variations," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2007, pp. 406–611.
[33] N. Liu, S. Hanson, D. Sylvester, and D. Blaauw, "OxID: On-chip one-time random ID generation using oxide breakdown," in *Proc. Symp. VLSI Circuits*, Jun. 2010, pp. 231–232.
[34] S. Stanzione, D. Puntin, and G. Iannaccone, "CMOS silicon physical unclonable functions based on intrinsic process variability," *IEEE J. Solid-State Circuits*, vol. 46, no. 6, pp. 1456–1463, Jun. 2011.
[35] X. Xi, H. Zhuang, N. Sun, and M. Orshansky, "Strong subthreshold current array PUF with 265 challenge-response pairs resilient to machine learning attacks in 130nm CMOS," in *Proc. Symp. VLSI Circuits*, Jun. 2017, pp. 268–269.
[36] P. H. Nguyen, D. P. Sahoo, C. Jin, K. Mahmood, U. Rührmair, and M. van Dijk, "The interpose PUF: Secure PUF design against state-of-the-art machine learning attacks," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2019, no. 4, pp. 243–290, 2019.
[37] M. Majzoobi, F. Koushanfar, and M. Potkonjak, "Lightweight secure PUFs," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design*, Nov. 2008, pp. 670–673.
[38] M. Majzoobi, M. Rostami, F. Koushanfar, D. S. Wallach, and S. Devadas, "Slender PUF protocol: A lightweight, robust, and secure authentication by substring matching," in *Proc. IEEE Symp. Secur. Privacy Workshops*, May 2012, pp. 33–44.
[39] Z. Pang, J. Zhang, Q. Zhou, S. Gong, X. Qian, and B. Tang, "Crossover ring oscillator PUF," in *Proc. 18th Int. Symp. Qual. Electron. Design (ISQED)*, Mar. 2017, pp. 237–243.

[40] J. Shi, Y. Lu, and J. Zhang, "Approximation attacks on strong PUFs," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, early access, Dec. 24, 2019, doi: 10.1109/TCAD.2019.2962115.

DING DENG received the B.S. degree from Beijing Jiaotong University and the M.S. degree from the National University of Defense Technology, Changsha, China, in 2017, where he is currently pursuing the Ph.D. degree. His research interests include hardware security, such as Trojan detection and PUF design, low-power design, and novel architecture of scan test for high-performance microprocessor in advanced nanometric technologies.

SHEN HOU received the B.E. degree in microelectronics from Nanjing University, Jiangsu, China, in 2005, and the M.S. degree in microelectronics from the National University of Defense Technology, Hunan, China, in 2008, where he is currently pursuing the Ph.D. degree in microelectronics. His main research interests include microprocessor design, hardware security, embedded systems, and the IoT application.

ZHENYU WANG received the M.S. degree from Hunan University, in 2019. He is currently pursuing the Ph.D. degree with the National University of Defense Technology. His research interests include security authentication protocol and trusted execution environment for embedded systems.

YANG GUO received the Ph.D. degree from the National University of Defense Technology, Changsha, China, in 1999. He is currently a Professor with the National University of Defense Technology, where he leads the Digital Signal Processor Group and the Director of the Integrated Circuits. He has authored or coauthored more than 50 publications on journals and conference proceedings. His primary research interests include low-power VLSI circuits, microprocessor design and verification, and electronic design automation (EDA) techniques for VLSI circuits.