On-Chip Trust Evaluation Utilizing TDC-Based Parameter-Adjustable Security Primitive

Haocheng Ma, Jiaji He, Yanjiang Liu, Jun Kuai, He Li, *Member, IEEE*,
Leibo Liu, *Senior Member, IEEE*, and Yiqiang Zhao, *Member, IEEE*

***Abstract*—Field-programmable gate arrays (FPGAs) are integrated circuits (ICs) that can be reconfigured to the desired functionalities, without manufacturing dedicated chips. Due to their programmable nature, FPGAs have been prevalent in the large majority of modern systems. This raises high demands for verifying the security of circuit implementations on FPGAs, since they are vulnerable to hardware trojans (HTs) that can be inserted through modified configuration files. In this article, we propose an on-chip security framework to ensure the trustworthiness of circuit implementations on FPGAs at runtime. The core of the framework is a time-to-digital converter (TDC)-based hardware security primitive that can be predeployed on FPGAs to verify whether the FPGA-based designs are tampered with or corrupted by HTs. The parameter-adjustable TDC sensor, which is the primary component of the primitive, is carefully designed, adjusted, and implemented, thus the TDC sensor can monitor the transient voltage fluctuations within FPGAs with a high resolution. Versus statistical data analysis, tiny abnormal variations introduced by the Trojan insertion and activation are distinguished. Experimental results on Xilinx Spartan-6 FPGAs demonstrate the effectiveness of the proposed TDC-based on-chip trust evaluation framework and HT detection method.**

***Index Terms*—Field-programmable gate array (FPGA), hardware trojan (HT), on-chip detection, principal component analysis (PCA), time-to-digital converter (TDC).**

I. Introduction

FIELD-PROGRAMMABLE gate arrays (FPGAs) can perform various user-defined functions flexibly leveraging abundant programmable resources and available intellectual property (IP) cores. Combined with other features, such as high computing efficiency and data throughout, FPGAs are becoming increasingly attractive in many applications, including military, aerospace, data center, over-the-air updating, the Internet of Things (IoT), etc. Therefore, FPGAs have become attractive targets for adversaries to jeopardize the entire physical system. Many severe attacks have been demonstrated in the literature. For instance, hardware trojans (HTs) inserted in Actel/Microsemi ProASIC3 Flash FPGAs help gain access to the configuration data [1]. In addition, a design flaw in the Xilinx 7-Series FPGAs, named StarBleed, can be exploited to break the bitstream encryption [2]. Recently, remote voltage attacks are proposed to extract the secret key and induce stealthy faults [3]–[6].

Among the security threats, HT is a type of threat characterized by malicious modifications to the original designs. An adversary can achieve this by changing designs programmed on FPGA in the post-fabrication phase [7]. An elaborate HT consists of two parts: 1) trigger and 2) payload. The trigger part is the activation mechanism that monitors either internal or external signals. In general, preset conditions are extremely difficult to reach to assure stealth, and once the conditions are satisfied, the payload part is activated and then delivers malicious effects to the original design, resulting in altering intended functions, reducing chip performance, leaking secret information or denial of service, etc.

To assure the trustworthiness of circuits' implementations on FPGAs, various HT detection methods have been proposed covering all phases of the circuit lifecycle. The fundamental idea is to identify extra influences (e.g., circuit functionalities, security properties or side-channel behaviors) introduced by HTs. Logic testing methods aim to detect functional modifications caused by HTs [8]. Formal verification-based methods aim to check whether the formalized specification of the design violates certain security properties [9]. Among them, the most heavily investigated is side-channel-based methods. Since HTs will inevitably make an impact on side-channel behaviors of the FPGA design, side-channel-based approaches identify HTs by comparing side-channel parameters of designs under test and those of golden references [10]. These approaches are non-destructive, cost effective, and have no need for complete HT activation, making them stand out among various HT detection methods. Starting from the power consumption [11], multiple side-channel parameters have been exploited in side-channel-based approaches, such as timing, electromagnetic radiation, thermal, etc.

A successful side-channel-based HT detection requires collecting adequate side-channel traces with a high signal-to-noise ratio (SNR). In this process, the noise from the external environment and internal circuits will significantly

Manuscript received July 10, 2020; revised September 25, 2020; accepted October 24, 2020. Date of publication November 3, 2020; date of current version September 20, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61832018; in part by the National Science and Technology Major Project of the Ministry of Science and Technology of China under Grant 2018ZX01027101-002; and in part by the China Postdoctoral Science Foundation under Grant 2019TQ0167. This article was recommended by Associate Editor S. Ghosh. *(Corresponding author: Jiaji He.)*

Haocheng Ma, Yanjiang Liu, Jun Kuai, and Yiqiang Zhao are with the School of Microelectronics, Tianjin University, Tianjin 300072, China.

Jiaji He and Leibo Liu are with the Institute of Microelectronics, Tsinghua University, Beijing 100084, China (e-mail: jiaji_he@mail.tsinghua.edu.cn).

He Li was with the Department of Computer Science and Technology, University of Cambridge, Cambridge CB3 0FD, U.K. He is now with the Department of Engineering, University of Cambridge, Cambridge CB2 1PZ, U.K.

Digital Object Identifier 10.1109/TCAD.2020.3035346

affect the final HT detection efficacy. As a result, how to collect side-channel data is of great importance. Taking the mostly exploited power side-channel-based methods for example, off-chip measurements are realized by measuring the voltage over a shunt resistor [12]. The environmental noise will disturb the traces collected by the external probe inevitably. Although multiple preprocessing approaches can be leveraged for data denoising, environmental noise cannot be filtered thoroughly, and a part of the useful information may also be erased together. Furthermore, there are some situations where monitoring the voltage variations using external equipment is infeasible, such as distributed IoT nodes. With abundant on-chip resources on FPGAs, it is more flexible to implement on-chip sensors that can typically achieve a higher SNR for on-chip measurements since they are less affected by external noise.

On-chip sensors deployed on the FPGA platform is important for accurately monitoring of voltage fluctuations. Researchers have proposed to use on-chip sensors realized by reconfigurable logic, e.g., ring oscillators (ROs) [13] and time-to-digital converters (TDCs) [14], instead of dedicated on-chip sensors integrated on FPGAs. In this article, we propose an on-chip security framework to monitor the anomalous voltage fluctuations on the FPGA power distribution network (PDN) caused by HTs. A parameter-adjustable TDC security primitive is designed, adjusted, and predeployed on target FPGAs. To the best of our knowledge, this is the first TDC-based HT detection method on FPGAs. Overall, the main contributions of this article are listed as follows.

1) An on-chip security framework is established to ensure the trustworthiness of circuit implementations on FPGAs at runtime, and the core of the framework is a parameter-adjustable TDC security primitive that is predeployed on FPGAs to accurately monitor the transient voltage fluctuations.
2) Sensitivity analysis of the TDC and on-chip variations' influence on the TDC are investigated to improve the detection accuracy. Furthermore, constraints of TDC configurations are optimized to be packaged into an opensourced hard IP core[1] that can be incorporated among different FPGA platforms within the same bracket.
3) Onboard experiments are performed to validate the effect of the on-chip framework under two fundamental HT detection scenarios. It is validated that the TDC-based sensor can capture voltage fluctuations, regardless of whether the HTs are activated.

The remainder of this article is organized as follows. Section II introduces the relevant background. Section III discusses the overall framework of proposed on-chip HT detection using TDC-based sensor. The effectiveness of our method is validated through experiments in Section IV. The overall article is concluded in Section V.

II. Background

A. Attack Model

HTs can be implanted at different phases for FPGA-based system designs, including prefabrication, fabrication, and post-fabrication [15], [16]. We assume that the FPGA fabrication and corresponding tool chains are trusted and the adversary exists in the post-fabrication stage. In terms of trigger conditions, HTs can be classified into combinational and sequential types. Combinational HTs will remain dormant unless a specific set of rare nodes occurs. While for sequential HTs, the condition of activation is generally a specific sequence of logic states. Compared to combinational ones, sequential HTs are more difficult to detect and can closely monitor the internal logic status of the original circuit's implementations on FPGAs. Therefore, the proposed method primarily focuses on the detection of sequential HTs.

According to the FPGA design flow, the adversary can complete HT insertion by modifying RTL codes, netlist files, bitstream files, etc. Modifying RTL codes is the easiest way for HT insertion because the logic units and the corresponding connections are explicit in the behavioral or structural space. However, HTs may be trimmed during the synthesis process and thus loses their partial or complete functionalities [17]. Meanwhile, the modifications of the bitstream are extremely complex for HT insertion since the basic information of the bitstream organization is not publicly available. Therefore, we assume that the adversary prefers inserting HTs in the post-synthesized netlist like netlist circuit description (NCD) files. The NCD file can be converted to a readable xilinx design language (XDL) file that contains specific FPGA resources and corresponding connections. After inserting HTs, the XDL file can then be reverted back to the NCD file for further implementing. For Vivado-based design flow, the HTs can be inserted by modifying the design checkpoint (DCP) file following similar routines. Finally, the HT-contaminated bitstream file is generated and configured onto the FPGA.

During HT detection, the acquisition of the golden model is of great importance. In this framework, the trusted design (HT-free circuit) can be implemented on the identical FPGAs, in the form of the bitstream files. The trusted data can be collected from large numbers of FPGA implementations, and will serve as a golden model through post-processing like on-chip variations reduction. Also, the above process should be performed in a trusted (controlled) environment. Therefore, the adversary could not interfere with the process to obtain the golden model.

B. On-Chip HT Detection Methods

The principle of on-chip detection is to monitor side-channel parameters inside the circuit utilizing embedded on-chip sensors. With minimal area and power overheads, this type of detection method can achieve more accurate results than off-chip methods and can be used along the entire lifetime of the circuit. The verification processes are completed by comparing the response of on-chip sensors with reference designs. Furthermore, with the well-designed framework and

[1] The IP core is available online at: https://github.com/AndrewMzZ/TDC-based-Security-Primitive.

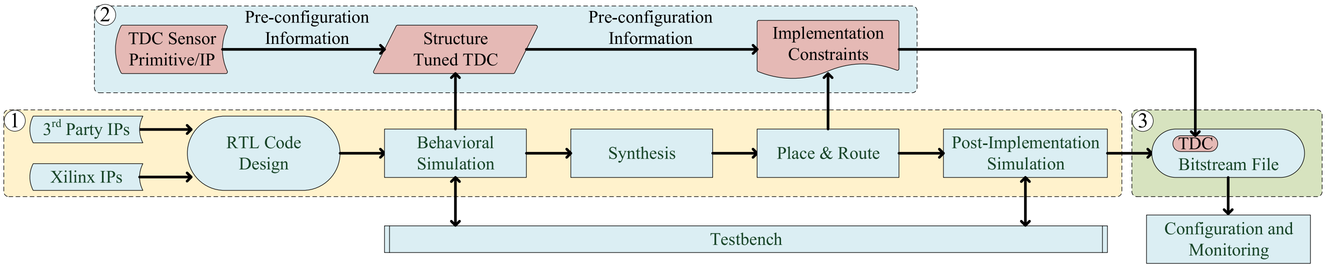


Fig. 1. On-chip security framework.

specific peripherals, the on-chip methods can achieve runtime detection.

ROs are more generally used for on-chip HT detection heretofore, and ROs are composed of an odd number of inverters that are chained together in a ring formation (i.e., the output of the last inverter is fed back as input to the first). When the HT consumes extra power, the voltage supplied to the oscillator decreases, resulting in a lower frequency of ROs' oscillations. Zhang and Tehranipoor [18] allocated an RO network (RON) across the entire chip and combined with the principal component analysis (PCA) to identify HT-inserted integrated circuits (ICs). The authors exploit an RON to monitor the static distribution of the supply voltage over an IC's surface, and an adaptive t-test is applied to distinguish the static impact of HTs insertion [19]. An RON can also serve as thermal sensors to detect temperature differences introduced by HTs [20]. The sensitivity of RON in terms of HT's switching activities was analyzed in earlier work [21]. Moreover, the detection results subject to the spatial distribution, where ROs close to the HT show more efficient responses.

Since the restrained length of a single oscillation ultimately determines the resolution of ROs, the path for the ring formation must be as short as possible. For FPGAs, the ROs can only be realized using lookup tables (LUTs) as inverters, and its output is routed through a switch matrix back to the inputs of the configurable logic blocks (CLBs). These relatively long inverter connecting paths have irreducible lengthy delays, thus the resolution of ROs is limited. Furthermore, previous works have proved that RO-based sensors are better suited for sensing static effects, while power transients are undetectable [14], [22]. With the consideration that carry-chain primitive on FPGAs has minimal delay, TDC-based sensors for monitoring transient voltage fluctuations on local FPGA have already been studied [4], [23]. In terms of the remote key recovery attack, the efficacy of TDC-based sensors is further validated on cloud FPGAs such as Amazon EC2 F1, as reported in [24]. However, there lacks work on the analysis of TDC-based on-chip HT detection for FPGA implementations.

C. HT Detection Principles

In the proposed on-chip HT detection framework, the TDC-based on-chip sensor is predeployed in FPGAs, and the sensor primitive is logically separated from other circuit parts to serve as a standalone verification module for HT detection. The TDC sensor can be packaged as a hard IP core that is transferable among certain ranges of different FPGA platforms. Whether implemented as a hard IP core or not, we assume that the TDC-based sensor itself can not be tampered by attackers, and the noninterference feature of the sensor primitive can be guaranteed leveraging *Pblock* [25] or other techniques alike. In this article, two fundamental HT detection principles are considered corresponding to different scenarios that a security examiner can handle. Here, the security examiner represents the person who will examine whether ICs are tampered by HTs.

The first one is the gray-box model that assumes the security examiner is able to trigger HTs through particular stimuli, such as activation generation [26] and test generation [27]. In this scenario, the activated payload will result in extra power variations due to its switching activities. The second one is the black-box model that assumes the security examiner does not have the capability to fully trigger the unknown HTs, which is more common in real applications. Although the security examiner cannot fully trigger the HTs, the HTs can still be partially triggered because the HTs' trigger parts keep monitoring the internal logic status. Thus, in this scenario, the power variations mostly come from the alterations of the original circuit design itself and the activities of the HTs' partial trigger parts.

III. On-Chip Security Evaluation Framework

In this section, the proposed on-chip security framework with a TDC-based hardware security primitive as the core component is introduced. The whole framework is illustrated in Fig. 1. It mainly includes three parts: 1) the FPGA development flow starts from the RTL code design as shown in ①; 2) the TDC sensor security primitive or IP is designed and adjusted as shown in ②; and 3) the circuits together with the preimplemented TDC security primitive are converted into bitstream files for FPGA configuration as shown in ③. More specifically, the TDC's basic operating principles are first analyzed, and then we investigate the voltage fluctuations on FPGA PDN due to the existence of HTs. Furthermore, the sensitivity analysis of the TDC and the influences of on-chip variations on the TDC are also demonstrated.

A. TDC Basics

In this section, the basics about the TDC structure, working mechanism, and implementation details are introduced from the security-oriented perspectives. As shown in Fig. 2, the basic TDC design exploits a delay-chain structure that can be utilized to measure tiny voltage fluctuations on FPGAs. Its

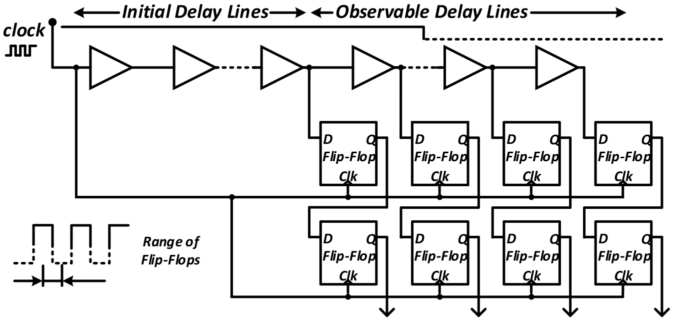


Fig. 2. Delay-chain-based TDC.

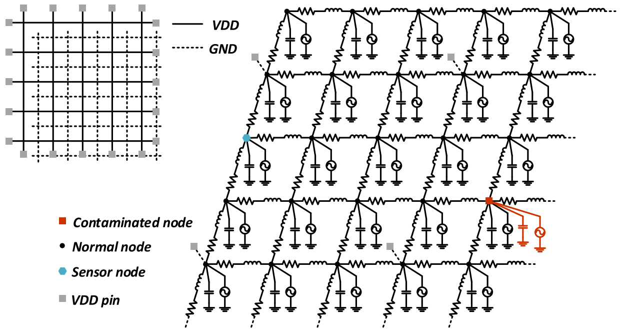


Fig. 3. FPGA power grid model [29].

core component is a tapped delay line consisting of a series of delay elements, and these lines are separated into initial and observable parts. For the observable part, signals in between the delay elements are tapped to memory elements, i.e., two-stage flip-flops (FFs). Since these delay elements are voltage sensitive, voltage fluctuations can be represented as the propagation distance of an input signal, i.e., the clock signal, through the dedicated tapped delay lines. More specifically, the output values of the TDC is the number of memory elements with state "1" within the measuring range T_r.

The TDC sensors on Xilinx FPGAs are realized utilizing CLBs [4]. A CLB contains a pair of slices, in which each slice consists of four LUTs, eight FFs, a network of carry-chain logic, and three types of multiplexers. These fundamental elements allow the designers to implement sequential as well as combinational logic. For the initial part of the TDC, its implementation is based on the elements with a less-area overhead but higher delay, i.e., multiple LUT primitives. For the observable part of the TDC, it is implemented using elements that provide the smallest granularity per bit, i.e., carry-chain logic, which is CARRY4 logic. Each CARRY4 consists of four configurable carry logic stages, where intermediate bits can be read out. This means that one CARRY4 can quantify the voltage fluctuations on the FPGA as four levels. For the memory part, two-stage FFs are exploited to store the propagation distance of the input clock signals, meanwhile, metastability is also reduced. More specifically, the clock inputs of FFs are driven by global clocks, which belong to the dedicated networks of interconnects for synchronous elements across the FPGA. These networks are designed to maintain low skew, low duty cycle distortion, low power, and improved jitter tolerance [28]. Thus, the clock signal integrity is guaranteed by the clock path that starts from the clock root and ends at the final clock sink.

To improve the practical applicability, the structural adjustability is integrated into the generation for the TDC-based hardware security primitive. We develop a script IPGEN[2] to carry out this process. IPGEN will import a preconfiguration file and then create the TDC-based IP core and the corresponding implementation constraints. The preconfiguration file contains user-defined parameters that are referenced to adjust the TDC structure. These parameters include the instance name, the numbers of LUT, and CARRY4 elements, respectively. The measuring range of the TDC-based sensor is determined by the adjustment of the relative number of LUTs and CARRY4s. The TDC-based IP core is specified in an electronic data interchange format (EDIF) netlist used as input to the Xilinx implementation tools. The implementation constraints are exploited to maintain the instantiated order, in which relative location constraints are applied to the above groups of sensor elements. Using these files, the TDC-based hardware security primitive can be deployed on the FPGA platform without impacting the design under test.

B. Voltage Fluctuations Analysis-Based HT Detection

From the analysis in Section III-A, the TDC sensor will monitor the voltage fluctuations on the FPGA caused by HTs, thus it is necessary to know how the PDN of the FPGA behaves under two fundamental HT detection principles. The PDN includes a network from the voltage regulation module on board down to internal power grids and every single logic cell on the FPGA. During the FPGA operation, logic cells draw current from the on-chip power grids under transition conditions (rise, fall, or stable). In general, on-chip power grids can be modeled as a mesh of resistive, inductive, and capacitive elements, as illustrated in Fig. 3. Hence, generated branch currents flow within the power grids will lead to voltage fluctuations on the FPGA. Here, we regard the currents draw by logic cells as independent current sinks in the power grids. Therefore, the behavior of power grids can be formulated by the modified nodal analysis technique [30], exactly as the following first-order differential:

$$G \cdot v_i(t) + C\frac{\mathrm{d}v_i(t)}{\mathrm{d}t} = A_j \cdot i_j(t) \tag{1}$$

where $v_i(t), i = 1, 2, \ldots, n$ is the node voltages over time and n is the node amount. G and C are the conductance and capacitance matrix, respectively. $i_j(t), j = 1, 2, \ldots, m$ is the time-varying current sinks, m denotes the sink amount, and A_j specifies where the current sink locates. Note that the inductive elements of power grids have been neglected at low frequency [31]. By solving (1), the node voltages $v_i(t)$ can be written as (2), where v_0 is the nominal supply voltage. From these equations, it can be concluded that power grids have a significant impact on the node voltages due to the nonideal

[2] The corresponding script and IP core is available online at: https://github.com/AndrewMzZ/TDC-based-Security-Primitive-with-Tunable-Parameters

impedance characteristics

$$v_i(t) = \left[v_0 - A_j \cdot i_j/G\right] \cdot e^{-\frac{G}{C}t} + A_j \cdot i_j/G. \tag{2}$$

Considering HTs inserted in the post-synthesized netlist, malicious changes, such as adding/modifying the resources and redundant interconnections, will affect the impedance of FPGA power grids, thus affecting the FPGA voltage fluctuations. This process will be explained starting from the logic cells inserted by the adversary. These logic cells serve as either the trigger part or the payload part of the HT. In order to supply power, the VDD and VSS pins of logic cells are connected to the on-chip power grids by adding metal wires (see the red part in Fig. 3). This means that extra current sinks, i.e., $i_j(t)$ and A_j, and impedance elements, i.e., C and G, are introduced into the original design. The current sinks consist of static and dynamic current components. Note that the characteristics of dynamic current components are different for the two detection principles. In the black-box model, the dynamic current components come from both the trigger and payload parts. While in the gray-box model, the dynamic current components mostly derive from the HT's trigger parts. According to (1) and (2), the HTs will always change the normal voltage fluctuations of FPGA PDN in both detection principles, whether the HTs are activated or not.

C. Sensor Sensitivity Analysis

In this section, we will characterize the output behavior of the TDC-based sensor as the voltage fluctuations vary. For the delay chain in TDC-based sensor, t_{typical} denotes its propagation delay at the typical corner, i.e., nominal supply voltage, room temperature and typical process. The values of t_{typical} are provided by the foundry and serve as the baseline of the propagation delay. When these conditions change, the propagation delay t_{pd} of delay elements will vary as follows:

$$t_{\text{pd}} = K_{\text{process}} \cdot [1 + (K_{\text{volt}} \cdot \Delta V)] \cdot \left[1 + (K_{\text{temp}} \cdot \Delta T)\right] \cdot t_{\text{typical}} \tag{3}$$

where K_{process}, K_{volt}, and K_{temp} are the process derating factor, voltage derating factor, and temperature derating factor, respectively. The voltage fluctuation ΔV is the deviation between the current supply voltage and nominal supply voltage, and temperature fluctuation ΔT is the deviation between the current temperature and room temperature. Under various operating states, different instantaneous current demands of the circuit lead to diverse voltage fluctuations across the FPGA PDN. These voltage fluctuations can be quantified by delay-chain-based TDC in the form of varying propagation distance, which is expressed as (Section III-C)

$$S = \frac{T_r}{t_{\text{typical}}} \left\{ 1 - \frac{1}{K_{\text{process}} \cdot [1 + (K_{\text{volt}} \cdot \Delta V)] \cdot \left[1 + (K_{\text{temp}} \cdot \Delta T)\right]} \right\} \tag{4}$$

Under both the black-box model and the gray-box model, the HTs that are inserted in the design can modify the normal voltage variations. In the proposed on-chip security framework, these anomalies are the monitoring objects of the TDC-based sensor embedded in FPGAs.

D. On-Chip Variations Analysis

The on-chip variations, such as process, voltage, and temperature variations have a strong impact on the output values of the TDC-based sensor. These effects are briefly expressed in (3) and analyzed in detail in this section.

The first type of on-chip variation is the voltage variation. Due to the capacity and noise of the voltage regulator, the voltage supplied to power pins of FPGAs may change within the proportion of up to 10%. This will alter the power consumption and delay of the design. To guarantee good stability of the supply voltage, we power FPGAs utilizing a stabilized direct current (DC) supply source with an accuracy of 0.05%.

The working temperature is the second type of on-chip variation that affects the power and delay for a circuit [32]. For most cases, increased temperature will reduce transistor speed and increase transistor leakage current. However, when coming to technologies below 65 nm, especially in 28 nm and under, the transistor switching speed increases as the temperature increases. During the experiments, we keep the ambient temperature as constant (room temperature 25°C) for the FPGA used in the experiments.

Due to variations of physical parameters and fabrication process parameters, transistors may have different threshold voltages and channel lengths. In addition, interconnect lines may have different widths, thicknesses, and contact resistances. All of these bring about the third type of on-chip variation, i.e., the process variations, which can be classified into two categories: 1) interdie variations and 2) intradie variations. Due to their influence, the output values of the TDC-based sensor will vary correspondingly, as shown in the following:

$$S_i = S + \Delta S_{\text{inter}} + \Delta S_{\text{intra}} \tag{5}$$

where $\Delta S_{\text{inter}} \sim N(0, \sigma^2_{\text{inter}})$ and $\Delta S_{\text{intra}} \sim N(0, \sigma^2_{\text{intra}})$ represent the effect of interdie and intradie process variations. They both obey normal distribution with the corresponding standard deviation [33]. As reported in [34], the intradie variation is only 0.75 % while the interdie variation is 6.61 %. Hence, the process variations appear mainly as interdie differences and should be eliminated in the HT detection.

Hou *et al.* [35] exploited the intrinsic relationship between transient current I_{DDT} and quiescent current I_{DDQ} to reduce the effect of the process variations. Cha and Gupta [36] used test structures to calibrate the delay influenced by process variations. In this article, we adopt the chip averaging technique proposed in [37] to reduce the effects of process variations on the TDC-based sensor. The chip-averaging technique averages the output values of the TDC-based sensor measured from all chips. Since process variations follow the Gaussian distribution, the averaging operations will eliminate the majority influence of process variations and reveal systematic anomalies caused by HTs. The output values of the TDC-based sensor

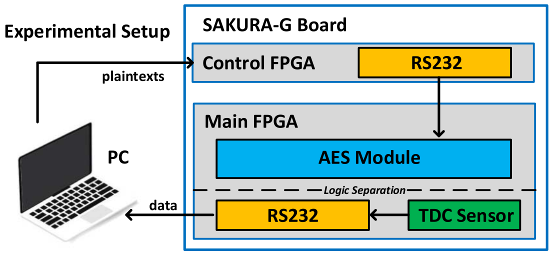


Fig. 4. Overview of the experimental setup.

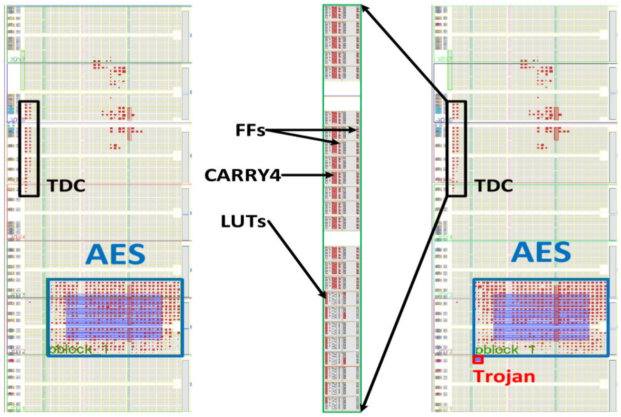


Fig. 5. FPGA layout of the HT-free (left), enlarged layout of TDC structure (middle) and HT-inserted circuit (right).

using the chip-averaging method are calculated as follows:

$$\bar{S} = \frac{1}{\#\text{chips}} \sum_{i=1}^{\#\text{chips}} S_i. \tag{6}$$

E. HT Detection Methodology

Theoretically, any analysis techniques can be leveraged for the HT detection, since our framework concentrates on the exploitation of the TDC-based on-chip sensor design and optimization. In the following, we first exploit a multivariate analysis technique, named PCA, to extract the statistical differences among the sensor data [38]. Then, a simple one-class classifier, i.e., minimum volume enclosing ellipsoid (MVEE) is performed on the first three principal components to distinguish the differences introduced by HTs [39]. As shown in (7) and (8), two security metrics are defined to quantify the efficacy of HT detection. One is the minimum distance between the spatial data set S^j_{PCA} of the testing circuit and the MVEE $\bar{S}_{PCA}$ of the reference circuit, denoted as $B_{\min}$. The other is the probability that the data of S^j_{PCA} falls into the ellipsoid $\bar{S}_{PCA}$ of the reference circuit, denoted as the false detection rate R_{false}. A value of $B_{\min} \geq 0$ signifies that the HT inserted in the testing circuit is identified successfully without false recognition. Otherwise, the false recognition happens with a possibility of R_{false} that classifies the HT-free circuits into HT-inserted circuits by mistake

$$B_{\min} = \min_{a\in S^j_{PCA}} \left\{ \min_{b\in \bar{S}_{PCA}} \{d(a, b)\} \right\} \tag{7}$$

$$R_{\text{false}} = \left(S^j_{PCA} \bigcap \bar{S}_{PCA}\right)/S^j_{PCA} \times 100\%. \tag{8}$$

IV. Experimentation Validation

A. Experimental Setup

Fig. 4 shows the overview of the experimental setup. Our experiment runs on the widely used hardware security evaluation platform SAKURA-G [40], featuring a *main* and a *control* Xilinx Spartan-6 FPGA. The main FPGA is a Spartan-6 XC6SLX75 FPGA for security implementations, controlled by an auxiliary Spartan-6 XC6SLX9 FPGA. Note that to better demonstrate the feasibility of the framework, three SAKURA-G boards are utilized in the experiment. For HT detection, the golden reference is obtained from these three boards by the chip-averaging technique. As a proof of concept, we select a 128-b AES module [41] as the golden circuit, implemented in the main FPGA at 48 MHz. This module occupies 1889 FFs and 1713 LUTs in terms of resource utilization. We use the script IPGEN to generate the TDC-based sensor, which consists of 28 LUTs and 16 CARRY4 elements. This sensor is implemented together with the 128-b AES design, serving as a hardware security primitive for detecting abnormal voltage fluctuations. A serial communication block, i.e., RS232, is used to handle the communication between the PC and the FPGA platform. The plaintexts required by the AES design are transmitted by the RS232 in the control FPGA. Meanwhile, the data in the TDC-based sensor are collected by the RS232 in the main FPGA linking the sensor with the PC.

Fig. 5 illustrates the entire layout of the main FPGA part. The AES module is placed and fixed in the left lower part of the main FPGA. Based on implementation constraints, the TDC-based sensor is fixed on the left side of the center region. Its overall location ranges from *SLICE_X*0*Y*75 to *SLICE_X*1*Y*98 in the FPGA layout. According to the user-defined parameters, the location will be adjusted in the y direction due to the number of used elements. Note that these elements of this part are logically separated from the AES module.

B. HT Insertion

To best determine how the proposed methods perform in detecting HTs, a total of 21 HTs targeting the tampering of the AES design are chosen from Trust-HUB [42]. Among them, 18 HTs perform some sort of data-leak to compromise the integrity of the circuit. The other three HTs (AES-T500, AES-T1800, and AES-T1900) directly attack the battery life of a power source attached to the host circuit. Besides, most of the HTs are time based, getting activated after observing a predefined sequence of input plaintext or the predefined number of encryptions. The other two HTs (AES-T100 and AES-T200) are always-on type. In our experiments, the *Pblock* technique is applied to constrain the HTs into certain regions (see Fig. 5) of the FPGA outside the AES module. Their respective resource utilization is listed in the Table I.

TABLE I
HT DETECTION RESULTS ON SPARTAN-6 FPGAs

Benchmarks	FFs	LUTs	Board 1				Board 2				Board 3			
			Grey-box model		Black-box model		Grey-box model		Black-box model		Grey-box model		Black-box model	
			B_{min}	R_{false}	B_{min}	R_{false}	B_{min}	R_{false}	B_{min}	R_{false}	B_{min}	R_{false}	B_{min}	R_{false}
AES-T100	1974	1808	0.21	0%	N/A	N/A	0.56	0%	N/A	N/A	0.81	0%	N/A	N/A
AES-T200	1918	1749	0.71	0%	N/A	N/A	2.14	0%	N/A	N/A	0.99	0%	N/A	N/A
AES-T300	1947	1793	0.50	0%	0.98	0%	0.56	0%	0.58	0%	1.01	0%	0.96	0%
AES-T400	2389	2022	0.56	0%	1.13	0%	2.16	0%	2.16	0%	0.90	0%	0.78	0%
AES-T500	2021	1805	3.99	0%	0.23	0%	0.79	0%	0.76	0%	3.25	0%	1.52	0%
AES-T600	2364	2159	0.17	0%	0.06	0%	1.63	0%	1.64	0%	0.83	0%	0.74	0%
AES-T700	1975	1784	0.07	0%	0.59	0%	0.55	0%	0.56	0%	0.95	0%	0.77	0%
AES-T800	1977	1792	0.20	0%	0.06	0%	2.13	0%	2.18	0%	1.23	0%	1.01	0%
AES-T900	1974	1879	2.24	0%	0.72	0%	0.13	0%	0.16	0%	1.71	0%	1.55	0%
AES-T1000	1975	1766	0.05	0%	0.33	0%	2.10	0%	2.13	0%	0.91	0%	0.89	0%
AES-T1100	1978	1795	0.05	0%	0.06	0%	2.12	0%	2.14	0%	0.96	0%	0.93	0%
AES-T1200	1975	1883	0.73	0%	0.34	0%	0.10	0%	0.13	0%	1.68	0%	1.66	0%
AES-T1300	1948	1805	0.32	0%	0.21	0%	0.09	0%	0.14	0%	4.47	0%	4.36	0%
AES-T1400	1951	1863	0.30	0%	0.20	0%	0.12	0%	0.12	0%	2.60	0%	2.50	0%
AES-T1500	1933	1954	1.54	0%	0.62	0%	0.08	0%	0.09	0%	1.52	0%	1.41	0%
AES-T1600	2393	2144	0.10	0%	0.12	0%	2.25	0%	2.27	0%	0.83	0%	0.80	0%
AES-T1700	2388	2182	0.11	0%	0.08	0%	2.23	0%	2.27	0%	0.97	0%	0.92	0%
AES-T1800	1891	1776	0.12	0%	0.08	0%	0.56	0%	0.55	0%	0.95	0%	0.93	0%
AES-T1900	1891	1837	1.94	0%	1.41	0%	0.12	0%	0.11	0%	1.80	0%	1.79	0%
AES-T2000	2411	2286	0.70	0%	0.96	0%	2.18	0%	2.16	0%	1.14	0%	1.13	0%
AES-T2100	2403	2396	2.42	0%	2.35	0%	0.56	0%	0.55	0%	2.46	0%	2.43	0%

N/A: AES-T100 and AES-T200 are always-on type HTs which do not require to be triggered.

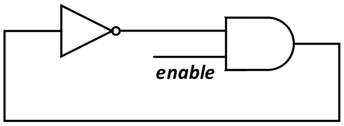


Fig. 6. One instance of the power virus.

C. Sensitivity Validation of the TDC-Based Sensor

We first analyze the sensitivity of the TDC-based sensor. This is achieved by associating its output values to the known voltage changes in the FPGA. To do so, we exploit the RON as the power viruses that create gradient voltage drops. Each power virus consists of an inverter followed by an AND gate and back into its own input port. As shown in Fig. 6, the switching activity of the power virus is controlled by setting the enable signal as 0 or 1. In this way, each power virus can create two power levels according to its maximal and minimal voltage fluctuations.

Altogether, there are 96 power viruses implemented on the FPGA in the form of LUT primitive. Every three 3 continuous distributed power viruses are grouped together to produce 33 gradient voltage fluctuation levels. As mentioned above, these levels are adjusted by enabling the specific amounts of the 32 power virus groups. For each voltage level, 50 samples of the TDC output are collected during the 300 sampling periods at 48 MHz. These samples are averaged to obtain the steady output values of the TDC-based sensor, which minimizes the effect caused by internal noise. Meanwhile, we record the actual voltage variations caused by power viruses using an oscilloscope.

Based on the results shown in Fig. 7, the relation between voltage fluctuations $f(x)$ and TDC output values x can be built as the linear regression equation: $f(x) = 0.00094x + 1.042$. We use the coefficient of determination R^2 [43] to evaluate the correlation between equation predicts $f(x)$ and actual data $y(x)$, given as (9)

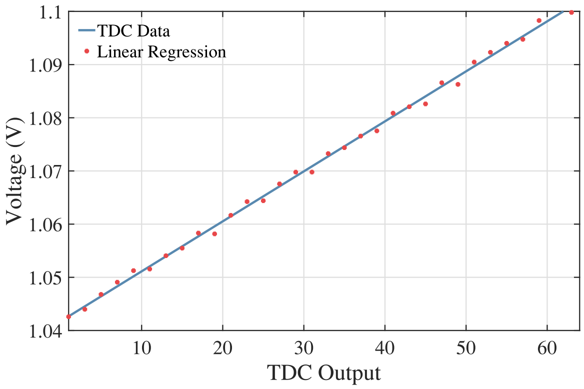


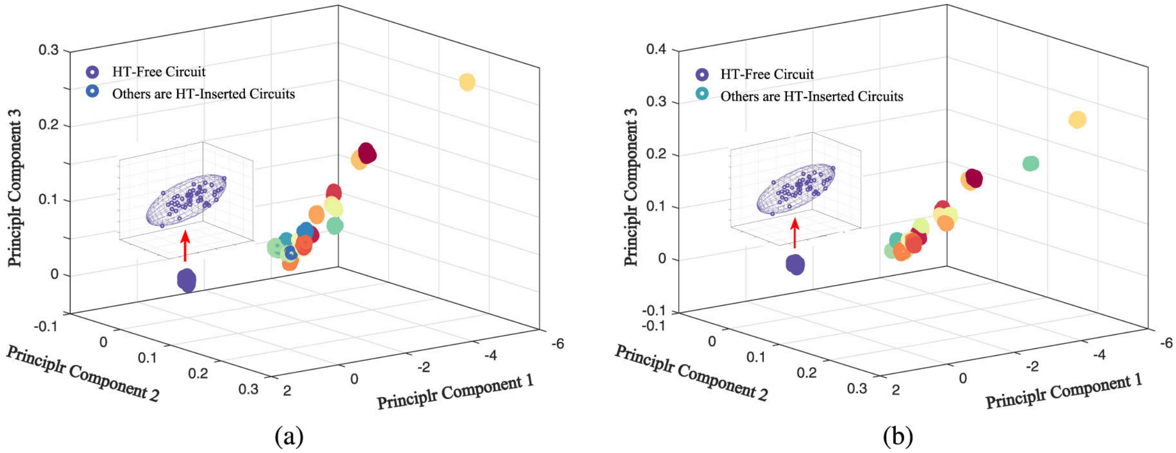
Fig. 7. TDC sensitivity analysis.

$$R^2 = 1 - \frac{\sum_x \left[y(x) - f(x)\right]}{\sum_x \left[y(x) - \hat{y}(x)\right]} \tag{9}$$

where $\hat{y}(x)$ is the mean of actual data. Note that an R^2 of 1 indicates that the equation-based predictions perfectly fit the data. The value of R^2 is 0.966 in our experiment, showing that the linear regression equation is consistent with the measured data. Hence, the sensor resolution can be obtained by computing the slope of the linear regression equation. Results show that the TDC-based senor can provide approx 0.94 mV of voltage resolution on average in overall experiments.

D. HT Detection Under Grey-Box Model

In this section, we will study the effectiveness of our on-chip security framework under the gray-box model. Here, 50 groups of sensor data are collected and each data is averaged with 16 replicate samples. Then, the chip-averaging method is applied to the data measured from three HT-free boards to minimize the effect of on-chip variations. In the gray-box



(a)

(b)

Fig. 8. Three top principal components of PCA results (board 1) under the gray-box model and black-box model. (a) HT-free and HT-inserted circuits under the gray-box model. (b) HT-free and HT-inserted circuits under the black-box model.

model, we assume that the security examiner will be able to trigger the HTs before detection, thus the payloads will be executing the designed malicious functions during detection. Table I shows the results obtained by applying the HT detection method proposed in this article. All HTs inserted in the AES design are identified successfully without occurring false recognition. Take circuits implemented on the board 1 as an instance, we project the first three principal components obtained by PCA onto the 3-D space. The MVEE encloses the populations of the HT-free circuit and serves the purpose of differentiating the circuit under testing into HT-free (i.e., inside the MVEE) and HT-inserted (i.e., outside the MVEE). As shown in Fig. 8(a), these populations of the circuits tampered by HTs are spatially separable with HT-free populations enclosed by MVEE and no data intersection exists in the 3-D space. Correspondingly, the minimum $B_{\min}$ value is 0.05 measured by AES-T1000 and AES-T1100 circuits, respectively. These results show that the TDC-based security primitive can efficiently perceive voltage anomalies from HTs.

E. HT Detection Under Black-Box Model

In this section, we validate the efficacy of the on-chip security framework under the black-box model. Similarly, 50 groups of sensor data with an average of 16 times are collected. Then, the chip-averaging method is applied to the data measured from three HT-free boards to minimize the effect of on-chip variations. In the black-box model, we assume that the security examiner has no capability to trigger the unknown HTs. This is a more realistic scenario in which payloads keep dormant during detection. Also, results obtained by applying the HT detection analysis are listed in Table I. It can be seen that all HTs inserted in the AES design are identified successfully without occurring false recognition. Take circuits implemented on the board 1 as an instance, we project the first three principal components obtained by PCA onto the 3-D space. The MVEE encloses the populations of the HT-free circuits and serves the purpose of differentiating the circuit under testing into HT-free (i.e., inside the MVEE) and HT-inserted (i.e., outside the MVEE). As shown in Fig. 8(b), these populations of the HT-inserted circuits all fall outside the MVEE that encloses the HT-free populations. It can be seen that no data intersection exists between these two type populations in the 3-D space. Correspondingly, the minimum $B_{\min}$ value is 0.06 detected from AES-T600, AES-T800, and AES-T1100, respectively. These results validate the on-chip security framework, validating that the TDC-based security primitive can efficiently perceive voltage anomalies from HTs. Overall, HT detection under the above two models can achieve 82% accuracy with a false-positive rate of 36%.

TABLE II
HT DETECTION RESULTS ON ARTIX-7 FPGA

Benchmarks	Grey-box model		Black-box model	
	B_{min}	R_{false}	B_{min}	R_{false}
AES-T300	0.59	0%	0.19	0%
AES-T1800	1.54	0%	1.59	0%

F. Scalability Analysis

To validate the scalability of the proposed TDC-based HT detection, we duplicate the experiments on the AX7035 platform featuring an Artix-7 XC7A35T FPGA. The same AES module is selected as the golden circuit and implemented on the Artix-7 FPGA at 50 MHz. Using the script IPGEN, the TDC-based sensor is obtained composed of 21 LUTs and 16 CARRY4 elements. As a proof of principle, we select the AES-T300 and AES-T1800 for HT insertion and detection, which leaks sensitive data and attacks the battery life, respectively. As discussed in Sections IV-D and IV-E, 50 groups of sensor data are collected and each data is averaged with 16 replicate samples. The Gaussian noise are added to mimic the effects the 0.75 % interdie variation and 6.61 % interdie variation. Under both gray-box model and black-box model, the results obtained by applying the chip-averaging and HT detection analysis are listed in Table II. It is validated that both HTs inserted in the AES design are identified successfully without occurring false recognition.

V. Conclusion

In this article, we proposed a novel on-chip security framework to detect whether FPGA-based designs are contaminated by HTs. The key idea is the exploitation of the TDC security primitive for monitoring voltage fluctuations of FPGA PDN. This TDC-based sensor is parameter-adjustable and packaged into a hard IP core. In two main detection scenarios, actual experiments demonstrated the efficacy of this method on detecting HTs under on-chip variations. For future work, we will design TDC-based security primitive with more precision to detect possible HTs with smaller sizes. More advanced classification algorithms will be incorporated into our on-chip HT recognition framework. Also, future work will decouple dependency on the pre-existing golden model and realize complete runtime HT detection.

References

[1] S. Skorobogatov and C. Woods, "Breakthrough silicon scanning discovers backdoor in military chip," in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst.*, 2012, pp. 23–40.

[2] M. Ender, A. Moradi, and C. Paar, "The unpatchable silicon: A full break of the bitstream encryption of Xilinx 7-Series FPGAs," in *Proc. 29th USENIX Security Symp. (USENIX Security)*, 2020, pp. 1803–1819.

[3] M. Zhao and G. E. Suh, "FPGA-based remote power side-channel attacks," in *Proc. IEEE Symp. Security Privacy (SP)*, San Francisco, CA, USA, 2018, pp. 229–244.

[4] F. Schellenberg, D. R. Gnad, A. Moradi, and M. B. Tahoori, "An inside job: Remote power analysis attacks on FPGAs," in *Proc. Design Autom. Test Eur. Conf. Exhibit. (DATE)*, Dresden, Germany, 2018, pp. 1111–1116.

[5] J. Krautter, D. R. Gnad, and M. B. Tahoori, "FPGAhammer: Remote voltage fault attacks on shared FPGAs, suitable for DFA on AES," *IACR Trans. Cryptograph. Hardw. Embedded Syst.*, vol. 2018, pp. 44–68, Aug. 2018.

[6] D. Mahmoud and M. Stojilović, "Timing violation induced faults in multi-tenant FPGAs," in *Proc. Design Autom. Test Eur. Conf. Exhibiti. (DATE)*, Florence, Italy, 2019, pp. 1745–1750.

[7] F. Schellenberg, D. R. Gnad, A. Moradi, and M. B. Tahoori, "Remote inter-chip power analysis side-channel attacks at board-level," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, 2018, pp. 1–7.

[8] A. Bazzazi, M. T. M. Shalmani, and A. M. A. Hemmatyar, "Hardware Trojan detection based on logical testing," *J. Electron. Test.*, vol. 33, no. 4, pp. 381–395, 2017.

[9] J. He, X. Guo, T. Meade, R. G. Dutta, Y. Zhao, and Y. Jin, "SoC interconnection protection through formal verification," *Integration*, vol. 64, pp. 143–151, Jan. 2019.

[10] J. He, Y. Zhao, X. Guo, and Y. Jin, "Hardware trojan detection through chip-free electromagnetic side-channel statistical analysis," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 10, pp. 2939–2948, Oct. 2017.

[11] D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, and B. Sunar, "Trojan detection using IC fingerprinting," in *Proc. IEEE Symp. Security Privacy (SP)*, Berkeley, CA, USA, 2007, pp. 296–310.

[12] Y. Liu, J. He, H. Ma, and Y. Zhao, "Hardware trojan detection leveraging a novel golden layout model towards practical applications," *J. Electron. Test.*, vol. 35, no. 4, pp. 529–541, 2019.

[13] A. Maiti and P. Schaumont, "Improved ring oscillator PUF: An FPGA-friendly secure primitive," *J. Cryptol.*, vol. 24, no. 2, pp. 375–397, 2011.

[14] K. M. Zick, M. Srivastav, W. Zhang, and M. French, "Sensing nanosecond-scale voltage attacks and natural transients in FPGAs," in *Proc. ACM/SIGDA Int. Symp. Field Program. Gate Arrays*, 2013, pp. 101–104.

[15] R. Karri, J. Rajendran, K. Rosenfeld, and M. Tehranipoor, "Trustworthy hardware: Identifying and classifying hardware trojans," *Computer*, vol. 43, no. 10, pp. 39–46, Oct. 2010.

[16] M. Tehranipoor and F. Koushanfar, "A survey of hardware Trojan taxonomy and detection," *IEEE Design Test Comput.*, vol. 27, no. 1, pp. 10–25, Jan./Feb. 2010.

[17] T. Reece and W. H. Robinson, "Analysis of data-leak hardware Trojans in AES cryptographic circuits," in *Proc. IEEE Int. Conf. Technol. Homeland Security (HST)*, Waltham, MA, USA, 2013, pp. 467–472.

[18] X. Zhang and M. Tehranipoor, "RON: An on-chip ring oscillator network for hardware Trojan detection," in *Proc. Design Autom. Test Eur.*, Grenoble, France, 2011, pp. 1–6.

[19] M. Lecomte, J. Fournier, and P. Maurine, "An on-chip technique to detect hardware Trojans and assist counterfeit identification," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 12, pp. 3317–3330, Dec. 2017.

[20] L. Pyrgas, F. Pirpilidis, A. Panayiotarou, and P. Kitsos, "Thermal sensor based hardware Trojan detection in FPGAs," in *Proc. Euromicro Conf. Digital Syst. Design (DSD)*, Vienna, Austria, 2017, pp. 268–273.

[21] Y. Qin and T. Xia, "Sensitivity analysis of ring oscillator based hardware Trojan detection," in *Proc. IEEE 17th Int. Conf. Commun. Technol. (ICCT)*, Chengdu, China, 2017, pp. 1979–1983.

[22] A. Le Masle and W. Luk, "Detecting power attacks on reconfigurable hardware," in *Proc. 22nd Int. Conf. Field Program. Logic Appl. (FPL)*, Oslo, Norway, 2012, pp. 14–19.

[23] D. R. Gnad, F. Oboril, S. Kiamehr, and M. B. Tahoori, "Analysis of transient voltage fluctuations in FPGAs," in *Proc. Int. Conf. Field Program. Technol. (FPT)*, Xi'an, China, 2016, pp. 12–19.

[24] O. Glamočanin, L. Coulon, F. Regazzoni, and M. Stojilović, "Are cloud FPGAs really vulnerable to power analysis attacks?" in *Proc. Design Autom. Test Eur. Conf. Exhibit. (DATE)*, Grenoble, France, 2020, pp. 1007–1010.

[25] *Pblock*. Accessed: Mar. 20, 2020. [Online]. Available: https://www.xilinx.com/support/answers/59424.html

[26] M. Yoshimura, T. Bouyashiki, and T. Hosokawa, "A hardware Trojan circuit detection method using activation sequence generations," in *Proc. IEEE 22nd Pac. Rim Int. Symp. Depend. Comput. (PRDC)*, Christchurch, New Zealand, 2017, pp. 221–222.

[27] Y. Liu, Y. Zhao, J. He, and R. Xin, "A statistical test generation based on mutation analysis for improving the hardware Trojan detection," *J. Circuits Syst. Comput.*, vol. 29, no. 03, 2020, Art. no. 2050049.

[28] *7 Series FPGAs Clocking Resources*. Accessed: Mar. 20, 2020. [Online]. Available: https://www.xilinx.com/support/documentation/user_guides/ug472_7Series_Clocking.pdf

[29] C. Wang, Y. Cai, H. Wang, and Q. Zhou, "Electromagnetic equalizer: An active countermeasure against EM side-channel attack," in *Proc. Int. Conf. Comput.-Aided Design*, San Diego, CA, USA, 2018, p. 112.

[30] C. J. Alpert, D. P. Mehta, and S. S. Sapatnekar, *Handbook of Algorithms for Physical Design Automation*. Milton, U.K.: Auerbach Publ., 2008.

[31] I. P. Vaisband *et al.*, *On-Chip Power Delivery and Management*. Cham, Switzerland: Springer, 2016.

[32] M. Barbareschi, G. Di Natale, and L. Torres, "Implementation and analysis of ring oscillator circuits on Xilinx FPGAs," in *Hardware Security and Trust*. Cham, Switzerland: Springer, 2017, pp. 237–251.

[33] B. Cha and S. K. Gupta, "Trojan detection via delay measurements: A new approach to select paths and vectors to maximize effectiveness and minimize cost," in *Proc. Design Autom. Test Eur. Conf. Exhibit. (DATE)*, Grenoble, France, 2013, pp. 1265–1270.

[34] A. Maiti, J. Casarona, L. McHale, and P. Schaumont, "A large scale characterization of RO-PUF," in *Proc. IEEE Int. Symp. Hardw. Orient. Security Trust (HOST)*, Anaheim, CA, USA, 2010, pp. 94–99.

[35] B. Hou, C. He, L. Wang, Y. En, and S. Xie, "Hardware Trojan detection via current measurement: A method immune to process variation effects," in *Proc. 10th Int. Conf. Rel. Maintainability Safety (ICRMS)*, Guangzhou, China, 2014, pp. 1039–1042.

[36] B. Cha and S. K. Gupta, "Efficient Trojan detection via calibration of process variations," in *Proc. IEEE 21st Asian Test Symp.*, Niigata, Japan, 2012, pp. 355–361.

[37] D. Ismari, J. Plusquellic, C. Lamech, S. Bhunia, and F. Saqib, "On detecting delay anomalies introduced by hardware Trojans," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Austin, TX, USA, 2016, pp. 1–7.

[38] Y. Liu, Y. Jin, A. Nosratinia, and Y. Makris, "Silicon demonstration of hardware Trojan design and detection in wireless cryptographic ICs," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 4, pp. 1506–1519, Apr. 2017.

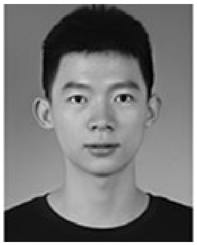
[39] P. Kumar and E. A. Yildirim, "Minimum-volume enclosing ellipsoids and core sets," *J. Optim. Theory Appl.*, vol. 126, no. 1, pp. 1–21, 2005.

[40] *SAKURA*. Accessed: Mar. 20, 2020. [Online]. Available: http://satoh.cs.uec.ac.jp/SAKURA/index.html

[41] *Announcing the Advanced Encryption Standard (AES)*, Federal Information Processing Standard 197, 2001.

[42] B. Shakya, T. He, H. Salmani, D. Forte, S. Bhunia, and M. Tehranipoor, "Benchmarking of hardware trojans and maliciously affected circuits," *J. Hardw. Syst. Security*, vol. 1, no. 1, pp. 85–102, 2017.

[43] D. Zhang, "A coefficient of determination for generalized linear models," *Amer. Stat.*, vol. 71, no. 4, pp. 310–316, 2017.



Haocheng Ma received the B.S. degree in microelectronics from Tianjin University, Tianjin, China, in 2017, where he is currently pursuing the Ph.D. degree with the School of Microelectronics.

His current research interests include digital circuit design, hardware security, and EDA for security.

Jiaji He received the B.S. degree in electronic science and technology and the M.S. and Ph.D. degree in microelectronics from Tianjin University, Tianjin, China, in 2013, 2015, and 2019, respectively.

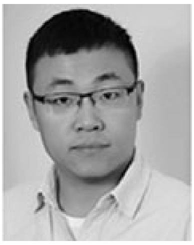
He was a visiting scholar with UCF, Orlando, FL, USA, and UF, Gainesville, FL, USA, from 2016 to 2018. He is currently a Postdoctoral Research Fellow with the Institute of Microelectronics, Tsinghua University, Beijing, China. His research interests are digital circuit design, hardware security, and EDA for security.

Yanjiang Liu received the M.S. degree in circuits and systems from the Guangdong University of Technology, Guangzhou, China, in 2016, and the Ph.D. degree from the School of Microelectronics, Tianjin University, Tianjin, China, in 2020.

His current research interests include digital circuit design and hardware security.

Jun Kuai received the B.S. degree in electronic science and technology from Tianjin University, Tianjin, China, in 2019, where he is currently pursuing the M.S. degree with the School of Microelectronics.

His current research interests include digital circuit design, hardware security, and EDA for security.



He Li (Member, IEEE) received the Ph.D. degree with the Department of Electrical and Electronic Engineering, Imperial College London, London, U.K.

His research work has been published in leading journals and conferences, such as IEEE TRANSACTIONS ON COMPUTERS, IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, DAC, ARITH, and FPT.

Dr. Li was a recipient of the best paper presentation award at FPT 2017. He serves on the technical program committees of the top-tier EDA and reconfigurable computing conferences, such as DAC, ICCAD, ASP-DAC, SOCC, and FPT, and severed as a Review Editor for *Frontiers in Electronics*. He is serving as the Publicity Co-Chair at IEEE FPT 2020.

Leibo Liu (Senior Member, IEEE) received the B.S. degree in electronic engineering and the Ph.D. degree from the Institute of Microelectronics, Tsinghua University, Beijing, China, in 1999 and 2004, respectively.

He is currently a Full Professor with the Institute of Microelectronics, Tsinghua University. His current research interests include reconfigurable computing, mobile computing, and very large-scale integration digital signal processing.

Yiqiang Zhao (Member, IEEE) received the B.S. degree in semiconductor physics and device, the M.S. degree in microelectronics, and the Ph.D. degree in microelectronics and solid-state electronics from Tianjin University, Tianjin, China, in 1988, 1991, and 2006, respectively.

In 1991, he joined Jinhang Technical Physics Institute, Tianjin, where he was responsible for analog and mixed signal circuit design. Since 2001, he has been with the School of Electronic Information Engineering and the School of Microelectronics, Tianjin University, where he is currently a Professor. His research interests include mixed-signal-integrated circuits, security chips, and hardware security.