INTEGRATION, the VLSI journal 55 (2016) 426–437

Contents lists available at ScienceDirect

INTEGRATION, the VLSI journal

journal homepage: www.elsevier.com/locate/vlsi

A survey of hardware Trojan threat and defense

He Li [a,b], Qiang Liu [a], Jiliang Zhang [b,*]

[a] School of Electronic Information Engineering, Tianjin University, Tianjin 300072, China
[b] Software College, Northeastern University, Shenyang 110819, China

ARTICLE INFO

Available online 24 February 2016

Keywords:
Hardware Trojan detection
Hardware Trojan diagnosis
Hardware Trojan prevention
IC market model

ABSTRACT

Hardware Trojans (HTs) can be implanted in security-weak parts of a chip with various means to steal the internal sensitive data or modify original functionality, which may lead to huge economic losses and great harm to society. Therefore, it is very important to analyze the specific HT threats existing in the whole life cycle of integrated circuits (ICs), and perform protection against hardware Trojans. In this paper, we elaborate an IC market model to illustrate the potential HT threats faced by the parties involved in the model. Then we categorize the recent research advances in the countermeasures against HT attacks. Finally, the challenges and prospects for HT defense are illuminated.

1. Introduction

A new type of hardware attack for integrated circuits (ICs), called hardware Trojan (HT), has emerged as an important research topic in recent years. HTs can launch serious attacks such as disabling or altering the functionality of an IC, or leaking sensitive user information. For example, in 2007, a Syrian radar failed to warn of an incoming air strike, which is responsible for a backdoor potentially built into the system's chips [1]. In January of 2014, the New York Times [2] exposed that the Quantum program of US National Security Agency (NSA) directly implants HT circuitry into USB communication protocol or USB port, which is believed to have achieved much secret data from all over the world, not only about military's network in China and Russian, but also the Mexican drug cartels and the police computer system, and even the EU's trade agency.

Until now, the number of reported HTs is much less than the software Trojans. However, various HTs have been designed and their effects have been demonstrated [3–6]. The worries about HTs have been expressed globally and it is believed that more sophisticated HTs will be discovered in the foreseeable future [7]. Therefore, investigations on HT threats and defense solutions have been carried out worldwide. Advanced software and equipment to aid in the fight against counterfeit microelectronics in U.S. weapons and cybersecurity systems has been transitioned to military partners under DARPA's Integrity and Reliability of Integrated Circuits (IRIS) program [8]. European project COST Action "Trustworthy Manufacturing and Utilization of Secure Devices" [9] also aimed at creating a European network of competence and experts on all aspects of hardware security including design, manufacturing, testing, reliability, validation and utilization. The network will play a key role in developing solutions responding to the hardware security challenges, hence strengthening the position of Europe in the field. Recently, national natural science foundation of China (NSFC) also initiated a project in 2015 (2.9 million RMB) on the countermeasures for IC potential safety hazard [10].

With these supports, various HT countermeasure approaches have been developed. This paper aims to review the recent progress in this research area, and more importantly to see how far we are from solving the problem. Recently, several surveys on hardware Trojan attacks and countermeasures have been published. Wang et al. [11] elaborated both HT taxonomy and Trojan detection methods. Tehranipoor et al. [12] presented a more in-depth discussion and classification of HT attacks, covering three topics: Trojan design and taxonomy, Trojan detection methods and design for hardware trust. Later, a more comprehensive review [13] augmented the complex HT threat models and illustrated the feasible countermeasures in specific fields concerning HT attacks. Another survey [14] discussed the feasibility of Trojan insertion at each stage of IC development and production chain. The latest survey [15] discussed the detection techniques of various hardware Trojans and analyzed complexity and limitations of the techniques.

Compared with the existing surveys, this survey from a different perspective uses an IC market model to elaborate specific HT threats faced by the parties involved in the model. Such a model has, to a large extent, made IC designers lost the control over the design, manufacture and application of ICs, and left opportunities for HT attacks. This paper also reports a new progress in HT defense—HT diagnosis, which not just detects the existence of HTs

[*] Corresponding author.
E-mail address: zhangjl@mail.neu.edu.cn (J. Zhang).

http://dx.doi.org/10.1016/j.vlsi.2016.01.004
0167-9260/© 2016 Elsevier B.V. All rights reserved.

but also finds their locations. In addition, the up-to-date HT prevention and real-time monitoring approaches are also surveyed. The purpose of this paper is to meet the demand for a survey on the state-of-the-art of HT attack countermeasures, with an emphasis on the recent developments that have taken place within the past three years and a focus on the approaches that most likely will be expected to practice.

The remainder of this paper is organized as follows. In Section 2, we present preliminaries related to hardware Trojan. In Section 3, we describe the IC market model, as well as HT threats faced by the parties involved in the model. In Section 4, we survey the countermeasures for HT attacks, including HT detection, diagnosis and prevention approaches. Challenges and prospects are presented in Section 5. Section 6 concludes the paper.

2. Preliminaries

2.1. Hardware trojan structure

A number of possible Trojans can be implanted in the design with varying activation mechanisms (triggers) and effects (payloads). Fig. 1 shows a typical structure of HT, which contains trigger, Trojan circuit, and payload [16]. To be hidden in ICs, the HTs are usually designed to be silent in most of time, and their triggers are associated with rare signals or events [13]. When the specified signal or event appears, the payload circuit is activated and implements malicious functions. By cleverly designing, the rare signals or events are unlikely to arise during design simulation or testing, but can occur during long period of field operation [17].

2.2. Hardware Trojan taxonomy

In the Trust_hub website [18], a number of typical HT-inserted benchmark circuits are available. Generally, the HTs can be categorized into two types: combinational Trojan and sequential Trojan, which are shown in Fig. 2 [13]. The combinational Trojan shown in Fig. 2(a) depends on the simultaneous occurrence of a set of rare signals (i.e., $T_1 \sim T_n$) to trigger a malfunction. The sequential Trojan shown in Fig. 2(b) undergoes a sequence of rare events (i.e., $S_1 \sim S_n$), each triggered by different sets of rare signals, before activating the payload. Alternatively, Bhunia et al. elaborated the taxonomy of HT in terms of trigger and payload. Based on the trigger condition, the hardware Trojans can be classified into analog and digital Trojans. The former is activated by analog conditions such as temperature, delay or device aging effect, whereas the latter is triggered by some Boolean logic functions. From the perspective of HT payload, a Trojan can cause a breach of confidence via a transmitted radio signal or serial data port interface. The payload of the HT can also be a side-channel attack which the information is leaked through the power trace or thermal radiation. Another type of HT payload would be unauthorized alteration in circuit functionality such as denial-of-service (DoS) attack.

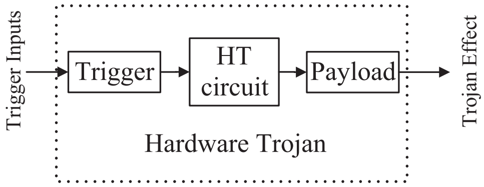


Fig. 1. A typical structure of HTs.

3. Hardware Trojan threats

In this section, we first introduce an IC market model in a way similar to the FPGA-based system market model proposed by Zhang and Qu [19], and then describe the potential threats from HTs in the model.

3.1. IC market model

As shown in Fig. 3, typically, there are five parties involved in the IC design, manufacture, and application flow. The role of each party is described below.

- *Foundries*: are the semiconductor manufacturers (e.g., TSMC, UMC, IBM) that contract with SoC designers to fabricate the ICs.
- *SoC designers*: design and create commercial products which contain various IPs.
- *IP vendors*: develop intellectual property cores (e.g., memory blocks, DSP cores) for SoC designers.
- *EDA tool vendors*: provide EDA tools for SoC designers and IP vendors to facilitate the design of large scale integrated circuits, e.g., Synopsys, Cadence, Xilinx and Altera.
- *IC end users*: companies or individuals purchase commercial products from SoC designers.

The interactions between these parties in the IC market model are shown in Fig. 3, where an arrow starts from the service supplier to the service receiver. Generally, as a party in this model, they will provide their competitive products to other parties. To be specific, SoC designers have connections with other parties in the IC market model. As a service receiver, they will purchase IPs from IP vendors to shorten the development cycle, acquire the licensed EDA tools to enhance the design toolkits from EDA tool vendors and contract with foundries to fabricate their chips. On the other hand, as a service supplier, they will provide their products to the end users who do not have a chip-level development team and require chips for a specific application. In addition, IP vendors will purchase the software tools from EDA tool vendors as well.

This model stems from the IC industry evolution and allows each party to focus on their expertise. However, the involvement

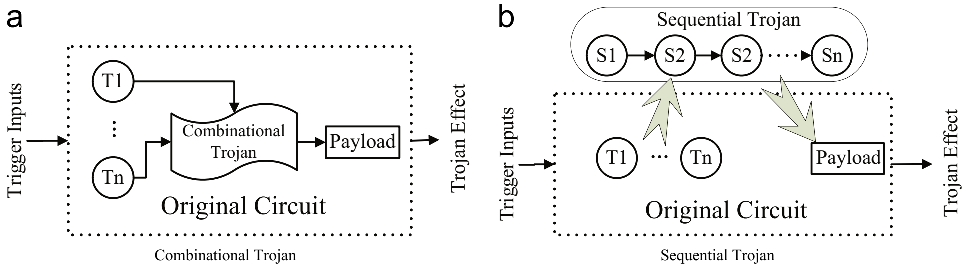


Fig. 2. Model for combinational and sequential Trojan circuits.

of multiple parties for producing a single product opens doors to adversaries for HT insertion.

3.2. HT threats in IC market model

HT can be implemented at different stages of IC life cycle. This section analyzes the existence of HT threats during the interactions between every two parties involved in the IC market model.

3.2.1. HT threat between SoC designers and foundries

During the fabrication process, there is no guarantee that foundries do not insert a certain type of HT in the chips. Chips fabricated in foundries may be threatened by untrusted staff or third parties to whom the fabrication process is accessible. For example, a Trojan can be implanted into the IC by intentionally/unintentionally modifying the dopant level [20] or the mask layout [21] during the sample or mass production [14]. In addition, foundries have their confidential instruments to manipulate the chip fabrication for some malicious purposes and may outsource the mask generation to a third party which has the opportunity to maliciously include mask macros in the GDSII.

3.2.2. HT threat between SoC designers and IP vendors

In the interaction between SoC designers and IP vendors, the former needs to ensure that the acquired IPs do not hide malicious function units which will be extremely difficult to be detected afterwards. Several different types of Trojan can be designed by a proficient adversary during the pre-silicon stage. To the best of our knowledge, the vast majority of HTs proposed so far in the literature are implemented and added to the design at the RTL level, before synthesis, placing and routing [22,23]. An untrusted insider in IP vendors can easily manipulate the RTL, insert malicious codes, modify macros during the design synthesis, and even alter the placement & route so as to make room for the Trojan circuitry [14]. Furthermore, an untrusted contractor who develops parts of the specification for IP vendors and SoC designers also has the opportunity to include malicious elements.

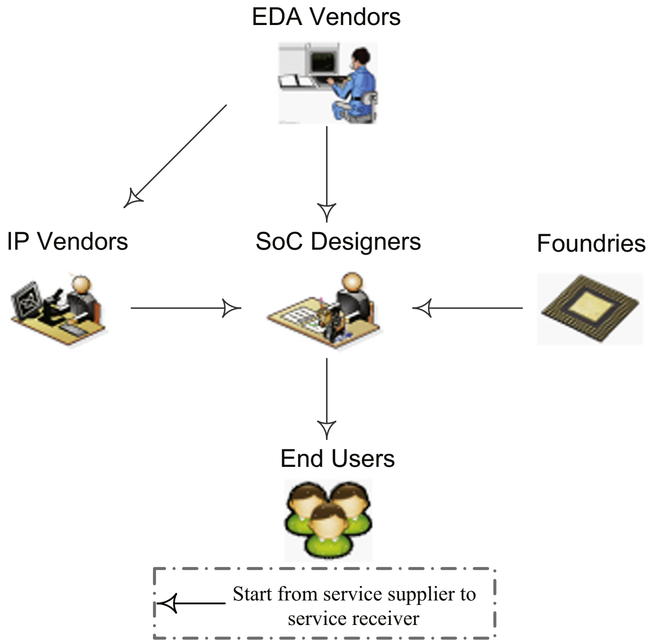


Fig. 3. IC market model.

3.2.3. HT threat between IP vendors (or SoC designers) and EDA vendors

EDA tools are widely used in many critical design stages. The software tools developed by EDA vendors may contain malicious codes which collect valuable data in IPs/SoCs. Recently, Qu and Yuan [24] analyzed the security vulnerabilities in EDA design tools and reported that logic implementations deduced by EDA tools may do more than required, regardless of the trustworthiness of the design team, the source of the EDA tools, and IP providers. This unexpected breaches could be exploited by adversaries to fulfill attacks.

3.2.4. HT threat between end users and SoC designers

End users who do not have a chip-level design team are concerned with HT attacks in products purchased from SoC designers. The HTs can be embedded into the chips during the SoC design step, to bypass the software security facilities and spy the users. The end users practically have no ability to detect this kind of hardware-level security threat, making them do not trust the traditionally reliable chips any more. The evidence of HTs/backdoors hidden in weapons control systems, nuclear power plants, and public transportation systems has been reported in [25].

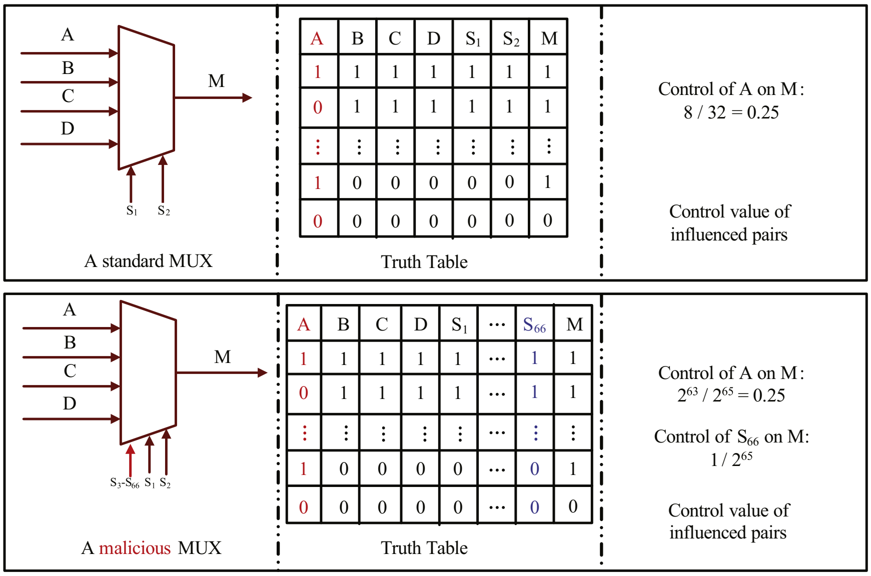
4. Countermeasures for hardware Trojan threat

Based on HT threats discussed in Section 3, we summarize current countermeasures against the threats existing in every two parties in the IC market model, as shown in Table 1. We can find that each party pair has corresponding techniques to deal with the HT threats and several defense techniques should work together to ensure trustworthiness in the whole IC market model.

Basically, countermeasures against HT attacks can be classified into three categories: (1) HT detection approaches, (2) HT diagnosis approaches and (3) HT prevention approaches. HT detection is the process that determines whether any HTs exist in the circuit. HT diagnosis is to orient the HTs in the ICs in terms of their types, locations, and input pins [26], so that one can either remove or mask the HTs from the circuit. Another kind of countermeasure, called HT prevention, is applied to the modern IC design stage to

Table 1
Countermeasures for HT threats existing during interactions between parties in the IC market model.

Party pairs	Stage & tech								
	Pre-silicon stage			Post-silicon Stage					Both stage
	IP trust verification	LO	DCI	Side channel	Logic test	Diagnosis	Split manufacture	Layout filler	Runtime monitor
SoC designers and foundries		√	√	√	√	√	√	√	√
SoC designers and IP vendors	√	√	√		√				√
IP vendors and EDA vendors	√								√
SoC designers and EDA vendors	√								√
End users and SoC designers									√



A	B	C	D	S_1	S_2	M
1	1	1	1	1	1	1
0	1	1	1	1	1	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮
1	0	0	0	0	0	1
0	0	0	0	0	0	0

A	B	C	D	S_1	⋯	S_{66}	M
1	1	1	1	1	⋯	1	1
0	1	1	1	1	⋯	1	1
⋮	⋮	⋮	⋮	⋮	⋯	⋮	⋮
1	0	0	0	0	⋯	0	1
0	0	0	0	0	⋯	0	0

Fig. 4. Control value comparison in 4-to-1 multiplexers. The output M takes on the value of one of the four data inputs (A, B, C, D) depending on the values of the two selection bits (S_1, S_2). In the malicious 4-to-1 multiplexer, there are also 64 extra selection bits $(\{S_3, \cdots, S_{66}\})$ that influence the MUX output if they match a specific input [36].

increase the difficulties of HT insertion, to enhance the efficiency of the HT detection and diagnosis methods, or to prevent the successful HT insertion during IC development. Actually, trustworthiness must be considered as an important measure in the design flow instead of relying on an passive protection afterwards. In Table 1, countermeasures against HT in pre-silicon and post-silicon stage are elaborated. Among them, IP trust verification [17,34,36,37,41], logic test [13,30,60–62] and side channel analysis [26,44–58] are HT detection approaches. Logic obfuscation (LO) [65,66,68–73,93], layout filler [95,96], dummy circuit insertion (DCI) [98,99] and split manufacture [100,102] belong to HT prevention approaches.

4.1. Approaches for hardware trojan detection

Early HT detection approaches typically rely on the golden model which represents the anticipated behavior of HT-free IC [27,28,30]. If the suspect IC deviated sufficiently from the golden model, it would be considered as being HT-infected. However, such model is difficult to acquire with the increasing complexity of ICs. One possible way is to use destructive reverse engineering [13]. Reverse-engineering is a very complicated, error-prone, and time-consuming process, usually consisting of five steps: decapsulation, delayering, imaging, annotation, and schematic creation [28,29]. Recently, Bao et al. proposed to use support vectors machine to automatically distinguish the features between HT-free and HT-inserted structures in the ICs based on the IC images [31]. The approach operates in the imaging step and thus simplifies the process of reverse engineering. However, state-of-the-art approaches would not allow destructive verification of ICs to be either cost effective or scalable [32]. Therefore, more and more researchers proposed advanced detection techniques to eliminate the requirement of golden model.

In what follows, we will introduce some golden-free HT detection approaches, which exploit pre-silicon verification and post-silicon testing techniques.

4.1.1. HT detection in pre-silicon stage

Pre-silicon trust verification techniques can be used to detect the HTs from the third party IP cores or inserted by the EDA tools. Adversaries always try to insert HTs in a way that the HTs are dormant during functional verification. Therefore, the HTs are resistant to the traditional functional verification approaches. Recently, several trust verification approaches have been proposed to flag suspicious circuits, which arise from manipulating RTL, inserting malicious codes and modifying macros at the design stage. These approaches are developed based on the formal verification and functional simulation methods.

The first set of approaches uses static functional verification techniques such as formal verification and assertion-based verification. The design comparison method [33] was proposed to resolve a question “How does one verify that a block does what it is expected to do, and nothing else?”. The basic idea was to make a comparison between two blocks from different sources with equivalent functionality. The full process involves wrapping designs and unrolling internal states to express each output entirely in terms of past and present inputs, completely removing state components such as flip-flops and latches, leaving only combinational logic and delayed inputs, and finally comparing the designs with a Boolean satisfiability (SAT) solver to find redundant logic.

Zhang et al. [34] defined all functions in the specification as properties, and the corresponding assertions were defined simultaneously. Then, coverage metrics (code coverage and functional coverage) were analyzed to identify uncovered parts, which can be considered as suspicious circuits. Moreover, a verification approach of system specification and implementation was also presented to identify extra functionality in hardware designs [35].

Nearly unused circuit identification (FANCI) approach was proposed in [36] to identify input signals with weak effect by static Boolean function analysis. The criterion for suspicious inputs is defined as the control value shown in Eq. (1). The control value of an input w_1 on an output w_2 quantifies what fraction of the rows in the truth table for w_2 are directly influenced by w_1:

$$\textit{Control value}(w_1, w_2) = \frac{\textit{counter}(w_1)}{\textit{size}(w_2)} \quad (1)$$

where $\textit{counter}(w_1)$ denotes the total number of rows of w_1 which determines the value of output w_2 in the truth table; $\textit{size}(w_2)$ denotes the total number of rows of w_2 in the truth table. Take a multiplexer shown in Fig. 4 as an example. Input A has a control value 0.25, which can be obtained by counting the number of rows with the same value in Column A and the total number of rows of Column M. For practical calculation, we only need to look through a half of the truth table, because the two halves represent the same property. For a malicious multiplexer, there are 64 additional select bits. When those 64 bits match a specific 64-bit key, M is changed to a malicious payload. However, each additional bit only affects a small fraction of Column M and their control value is 2^{-65}. After calculating the control value for all inputs, the approach derives a threshold for control value and those inputs with the value below the threshold are considered as suspicious inputs.

The second set of approaches combines static and dynamic verification techniques. Hicks et al. [17] first formulated the HT detection as unused circuit identification (UCI) problem which can be considered as suspicious circuits, whenever they did not affect outputs during simulation. The UCI algorithm could trace all signal pairs, and select those signals with equal properties as suspicious HT insertion targets. Afterwards, the suspicious circuit could be isolated and a exception notification logic is added to notify the abnormalities at run time. Fig. 5 shows an example. In subfigure (a) UCI identifies signal pair rout.su and rin.su in the MUX as a suspicious circuit for HT insertion. In subfigure (b) rin.su is assigned to rout.su and a comparison circuit is inserted to verify the inconsistencies of the output.

Later, an optimized approach VeriTrust [37] flagged suspicious circuits by identifying potential trigger inputs used in parasite-based HTs. A parasite-based HT exists along with the original circuit, and does not cause the original design to lose any normal functionalities. The authors used the example shown in Fig. 6 to illustrate the approach. The K-Map of the parasite-based HT-inserted circuit is shown in Fig. 6(b), where the third row represents the malicious function while other rows show the normal function. By comparing it with the K-Map of the original circuit in Fig. 6(a), we can see that the parasite-based HT enlarges the K-Map size with additional inputs so that it can keep the original function while embedding the malicious function. If we set all entries of the malicious function as don't cares, the trigger inputs (i.e., t_1 and t_2) become redundant. These redundant inputs are then flagged as potential HT trigger inputs for the further examination [38].

With increased IC complexity, computational effort required by verification methods increases dramatically [39]. It is not proper to carry out formal verification to the whole circuit. A metric for the assessment of Trojan inexistence called Trojan Assurance Levels (TALs) was introduced to locate the insecure area of chip designs [40]. This metric can be mathematically defined by evaluating the circuit functionality, structure and functional interactions at different levels of abstraction. It is expected that the HT detection process will be more efficient by focusing on the regions with high possibility of HT inserted according to TAL.

Most recently, feature analysis-based third party IP trust vitrification approaches have been presented to detect HT in the pre-silicon stage [42,43]. Oya et al. [42] proposed the first feature analysis-based method for identifying HT-free or HT-inserted gate-level netlists without using a Golden netlist. The proposed method does not directly detect HTs themselves but Trojan nets. First of all, features of Trojan nets are extracted from the randomly selected gate level netlist of HT-inserted benchmarks and they are assigned different scores based on the weakness in the circuit. Then a set of score thresholds of Trojan nets in terms of Max score, Max score count and Max constant cycles are used to classify HT-free and HT-inserted netlists. Yao et al. [43] presented another analysis-based third-party IP trust verification framework, which conducted HT feature analysis on the flip-flop level control-data flow graph (CDFG) of the circuit. Firstly, HT features are categorized based on different HT trigger conditions and an HT feature database is established by analyzing the CDFG. For each feature in the HT feature database, different feature matching algorithms are developed to detect nodes or node groups using the feature. Finally, all HT candidates that match HT features are reported for further manual examination.

4.1.2. HT detection in post-silicon stage

Post-silicon testing process can be used to find the HTs inserted at the design stage and the manufacturing stage. The detection approaches can be further divided into side channel analysis and logic test. The former can passively detect the HT's side-channel signal and the latter activate the HT by using appropriate test patterns. It is believed that both approaches are complementary with each other for IP/SoC designers to detect HTs.

Side channel analysis: Side channel (SC) analysis has been widely applied to HT detection, due to the fact that the inserted HTs would have effects on the circuit's power consumption [26,44–49,59], signal delay [50–53] and electromagnetic emanation (EM) [54–57]. SoC designers can utilize advanced test instruments to measure the power, delay and EM of ICs, and prove existence of HTs through the analysis techniques such as differential power analysis.

According to current analysis, HT embedded in the original circuit will add extra leakage current and power, but it is difficult to be detected directly due to the tiny impact on the whole circuit. Therefore, various design partition-based approaches have been proposed [26,49,47]. The basic idea is to augment the effect of the HTs. For instance, a scan cell distribution-based partition technique [49] is used to divide the circuit into regions. Then, activity-driven test pattern is generated to magnify the activity in the target region where the HT may be located. Finally, the localized transient current in each region is measured for HT detection.

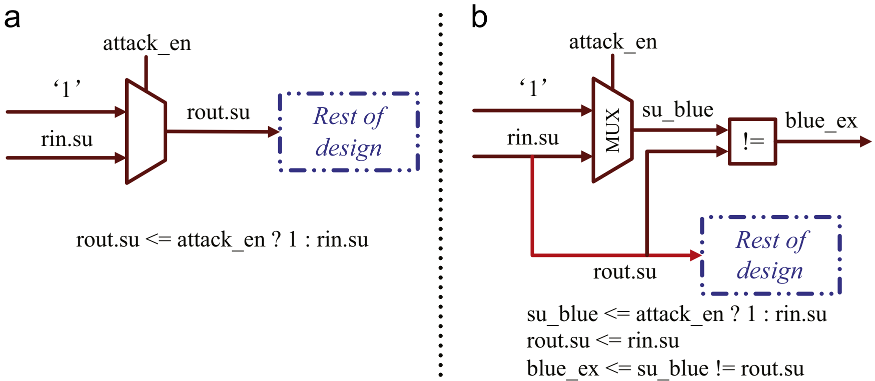


Fig. 5. A example of circuit/HDL transformation to remove an attack from a design based on UCI. In (a), UCI detects that unused circuits where the value for rout.su always equals rin.su, then identifying the mux as a candidate for removal. (b) shows the circuit modification by connecting rin.su to rout.su directly and adding exception notification logic to notify any inconsistencies at runtime [17].

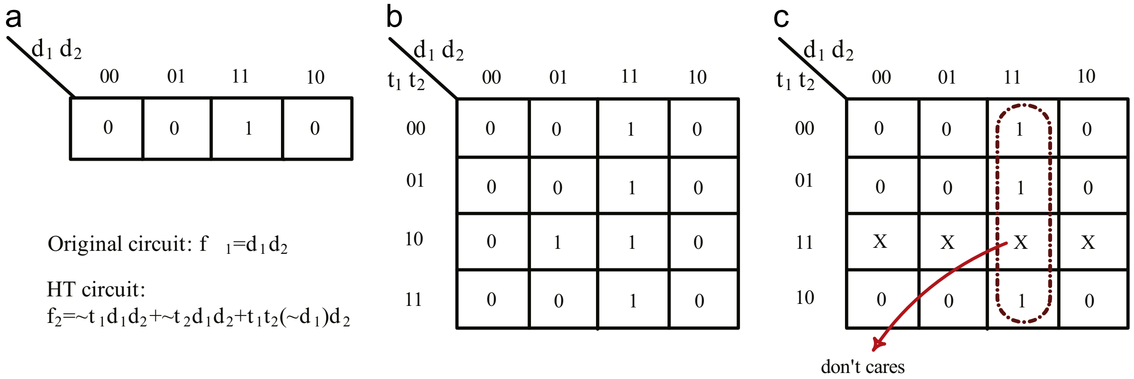


Fig. 6. K-map in VeriTrust [37]. (a) K-map of original circuit; (b) K-Map of parasite-based HT; (c) K-Map of parasite-based HT in (b) by setting entities of the malicious function as don't cares.

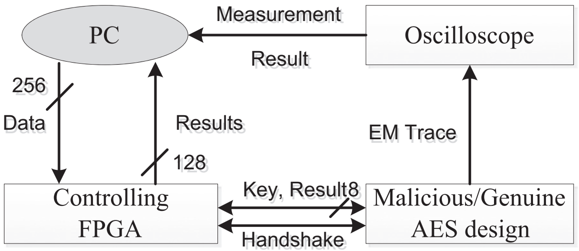


Fig. 7. Schematic view of EM-based the measurement setup [54].

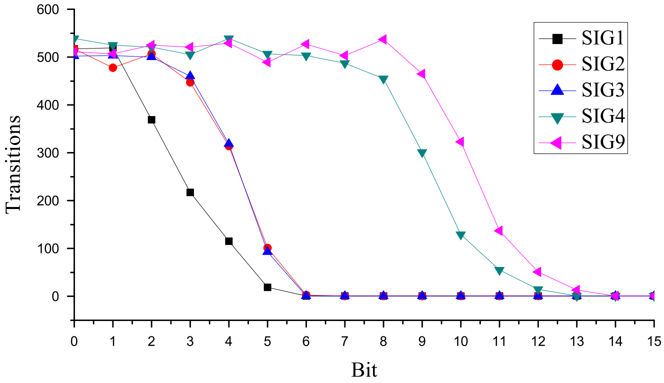


Fig. 8. Acceleration of hardware Trojan detection based on word-level statistical properties management [62].

Compare with other side-channel signals, path delay has its advantage: each path delay is independent with each other and they can be measured separately. However, the main challenge is that the tiny impact on HT delay cannot be effectively measured under the increasing level of process variation. Hence, Cha and Gupta [51] focus on reducing the process variation influence on Trojan delay. They calibrate the effect of process variation and built the delay model for each logic block, and then carry out HT detection in each block. In their further research, the additional path delay induced by Trojan is maximized by using a path selection scheme [52].

With the development of the testing instrument, a further parameter, electromagnetic emanation (EM) was proposed to carry out side-channel analysis [54–56]. Soll et al. [54] proposed an HT detection approach using localized EM measurements on FPGA. The basic flow of the EM-based HT measurement is shown in Fig. 7. It consists of two FPGAs, one FPGA can be used as a controller while the other FPGA is used to evaluate a cryptographic implementation. The Virtex-II Pro XC2VP30 is used as the control FPGA and the Virtex-II Pro XC2VP7 is configured with either the genuine or the Trojan-inserting design.

Logic test: As semiconductor technique advances, side channel analysis-based detection approaches alone become ineffective due to the significant impacts of process variation. It is suggested to combine side channel analysis with logic test approaches that focus on generating appropriate test patterns to activate HTs [13,30,60–62], which can help IP vendors and SoC designers to observe the payload of HTs.

Since an adversary can insert a number of different HT instances, generating deterministic test patterns to activate all of them is impractical. Therefore, statistical approach for test vector generation have been developed. A random sampling approach, multiple excitation of rare occurrence (MERO) [30] was proposed to generate effective input vectors. The basic concept is to detect low probability conditions at the internal nodes and then derive an optimal set of vectors to activate the rare nodes at least *N* times, in a similar way to N-detect test used in stuck-at ATPG. The probability of activating a Trojan is improved by increasing the transitions of nodes that are random-pattern resistant. The iterative nature of the process determine that the approach is time-consuming.

To improve the efficiency of HT detection, Li and Liu [62] proposed an HT detection acceleration approach to increase the probability of activating HTs and thus reduce the detection time. Basically, the bit-level transition activity can be enhanced by managing signal word-level statistical properties with mean (μ), standard deviation (σ) and autocorrelation (ρ). Signals list in Fig. 8, such as SIG1 and SIG2, have different statistical parameters, i.e., different standard deviation and temporal correlation. In [62], the author enhanced the rare nodes transition activity by increasing the standard deviation and reducing the temporal correlation. To be specific, the statistical statistics (μ,σ,ρ) of SIG1–SIG3 are (0, 10, 0.99), (1, 10. 0.40) and (0, 10, 0.10) respectively. Fig. 8 shows that the transition activity of the 6th bit can be enhanced. Alternatively, signals with larger standard deviation can also increase the transition activity of rare nodes dramatically. For example, we can observe the clear enhancement for transition activity of the 6th bit with SIG4 (0, 1000, 0.99) and SIG9 (0, 3000, 0.99). A traceback approach is then designed to determine the statistical properties of primary input signals to increase the transitions of an internal bits. As an example, the enhancement of rare node transition in a 5-tap FIR circuit is presented in Fig. 8. Transition activity of the 6th bit can be increased by statistical signals with different (μ,σ,ρ).

4.2. *Approach for hardware Trojan diagnosis*

As mentioned in Section 4.1, various HT detection approaches have been developed to determine whether HTs exist in a circuit. However, for the IP and SoC designers, they prefer to acquire the

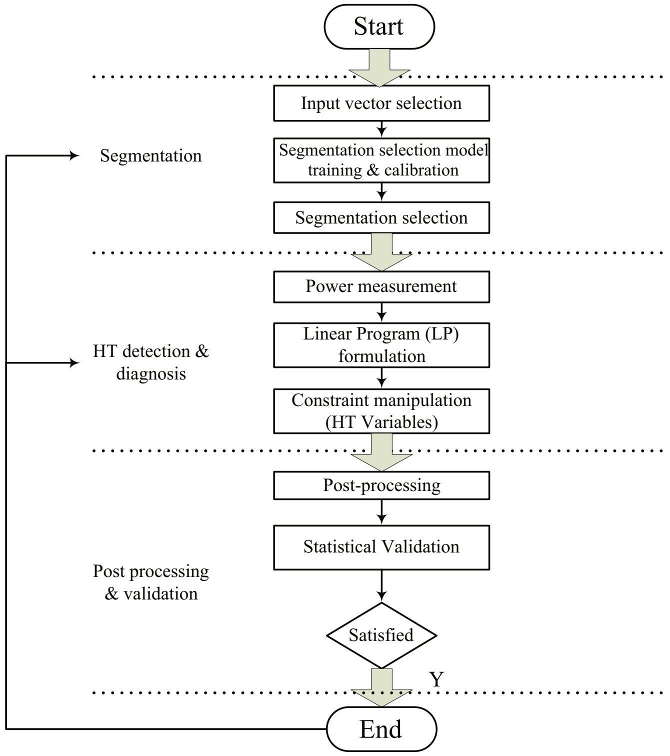


Fig. 9. Flow of the segmentation-based HT detection and diagnosis approach [47].

specific HT information in their product in order to remove the HT threats. The difference between HT detection and diagnosis is that HT detection is the process that determines whether any HTs exist in the circuit, whereas an HT diagnosis approach is to determine the locations, types and triggers of the HTs in the circuit in order to remove or mask the HTs from the circuit.

Wei and Potkonjak proposed HT diagnosis approach based on circuit segmentation and gate level characterization (GLC) [47]. The basic procedure, as shown in Fig. 9, contains three phases: segmentation, HT detection and diagnosis, and post-processing. A segmentation model is first trained by segment properties including controllability ratio, correlation ratio and GLC accuracy. The model is then used to divide large circuits into small sub-circuits. In sub-circuits with small number of gates it is easier and more accurate to detect and diagnose HTs by tracing leakage power. In the third phase, statistical methods are applied to validate the prediction results in the post-process. The whole process is repeated multiple times if necessary.

The same authors also proposed another HT diagnosis approach [26] based on segmentation and consistency analysis of gate-level properties. The approach detects the HTs by measuring the gate-level properties of two segments with overlapping gates when these gates exhibit inconsistent leakage power. However, the HT detection results do not indicate which segment the HTs may be embedded in, and thus it is difficult for the HT masking process to handle the HTs. To diagnose the HTs, a third segment with the same set or subset of overlapping gates is introduced to specify the HT locations, which can be used as an arbiter for HT diagnosis. Fig. 10 shows an example of the consistency-based HT diagnosis. There are three segments in the circuit. Segment 1, Segment 2 and Segment 3 with the same overlapping gate *X* have leakage power scaling factor S_1, S_2, and S_3, respectively. Given that S_3 has the same value with S_1, it indicates that Segment 2 potentially contains an HT. In the case where all three scaling factor values have large difference compared with the others, it concludes that multiple HTs are embedded in at least two segments and then more segments that cover the overlapping gates should be found to further diagnose the HTs.

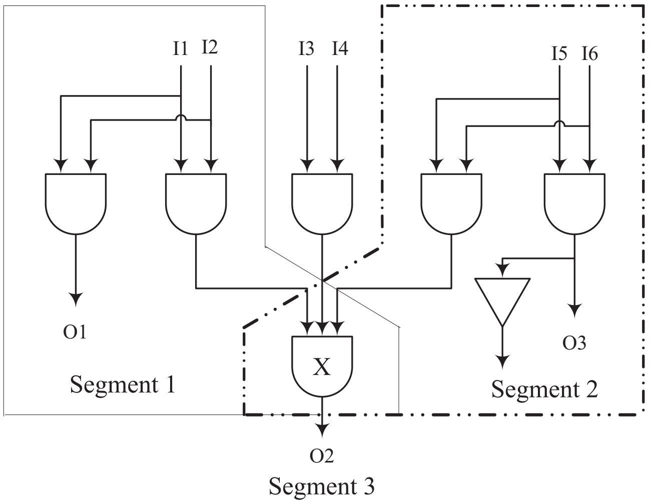


Fig. 10. Example of consistency-based HT diagnosis. We demonstrate the gate characterization in three segments with overlapping gates *X*. The consistency in Segments 1 and 3 exposes the possible HTs in Segment 2 [26].

4.3. *Approach for hardware Trojan prevention*

The HT detection and diagnosis approaches, though promising, still face some challenges such as rare node identification, process variations, and measurement deviation. To improve the effectiveness of these approaches, ICs must be designed with self-protection awareness. Currently, obfuscation, layout-filler, dummy circuit insertion and split manufacturing are the main techniques for IP vendors, SoC designers and foundries to prevent HT attacks.

4.3.1. *Logic obfuscation*

Logic obfuscation (LO) is a probabilistic transform *O* that converts a source circuit *F* into a new circuit *O*(*F*) that has the same functionality as *F* but less intelligible in some senses [65]. After this kind of transformation, the secrets in designs tend to be more difficult to comprehend. This feature will help to keep the original design and protect the functionality of commercial IC which is only known by IP vendors or SoC designers.

Barak et al. [66] initiated a theoretical study of obfuscation. They focused on black-box obfuscation which should have two strong requirements: (1) the obfuscated circuit operates the same function as the original circuit efficiently, and (2) the obfuscated circuit should leak no information except for its black-box (input–output) functionality. The main challenge is that this definition of obfuscation cannot always be achieved. Later Goldwasser and Rothblum [67] proposed a new notion of “best possible” obfuscation which relaxes the second requirement to that the obfuscated circuit leaks no more information than any circuit of the same functionality. Best-possible obfuscation guarantees that any information that is not hidden by the obfuscated circuit is also not hidden by any other circuit having the same functionality.

Generally, obfuscation techniques can be classed into combinatorial logic [68–70] and sequential logic obfuscation [65,71–73]. Combinatorial logic obfuscation is to obfuscate IC designs by randomly inserting additional key-gates (XOR/XNOR [68] or Multiplexer [69,70]). One of the inputs to a key-gate is the functional input in the design and the other is one bit key input. The correct key will be stored in a tamper-evident memory inside the design to prevent access to attackers or implemented by a physical unclonable function (PUF) (with the increasing demands of security and privacy in various areas [74–92], cryptographic key storage become one of the most challenging design concerns, while Silicon PUF has become a promising solution for the challenge [93]). However, key-gates are randomly inserted into the design, which does not necessarily ensure that wrong keys corrupt the outputs. Therefore, Rajendran et al. [94] proposed a fault analysis-based logic encryption technique which can ensure that wrong keys corrupt the outputs to maximize the ambiguity for an attacker. Sequential logic obfuscation is another way to obfuscate IC designs by adding new states and transformations in the finite state machine (FSM) [71–73]. Li and Zhou [65] proved that any best-possible obfuscation of a sequential circuit can be realized by a procedure of four operations: retiming, resynthesis, sweep, and conditional stuttering. Upon applying the correct key, the obfuscated design will exhibit a correct function.

4.3.2. *Layout filler*

Layout filling techniques are proposed to facilitate the HT detection and to reduce the likelihood of HT insertion by filling functional logic in the empty space of layout. This technique will prevent untrusted staff or third parties from inserting a Trojan by intentionally/unintentionally modifying the mask layout in foundries or prohibit an insider in IP vendors or SoC designers from altering the placement & route for HT circuits.

Built-in self-authentication (BISA) techniques were presented in [95,96] to prevent the insertion of additional Trojan gates in the layout and mask of circuits. BISA works by eliminating this spare space and filling it with functional standard cells (SCs), instead of nonfunctional filler cells, during layout design. The inserted SCs are then connected to form a circuitry called BISA circuit, which is independent of the original circuit. BISA provides protection against removal attacks where one or more of the filler standard cells are removed to place the Trojan cells by producing an incorrect signature during self-authentication. Fig. 11 shows the BISA design flow, where BISA can be embedded into the conventional ASIC design flow. The left rectangles in the figure show the basic conventional ASIC design flow, and the right ones in the figure are the additional steps for inserting BISA circuitry.

An HT prevention approach for FPGA was proposed in [97] by identifying unused resources at the layout-level within the FPGA device where low-level dummy logics (LLDLs) were filled. Basically, LLDLs are added to the original circuit after the placement and routing stage. The native circuit description (NCD) file of the original circuit could be converted to the layout-level hardware description language (LHDL) which can be obtained by certain FPGA vendors. Compared with high-level protection techniques, filling LLDLs may leave considerable amount of logic resources to be misused by attackers. Additionally, by employing the proposed low-level technique, the bitstream reverse engineering becomes much more difficult for the purpose of leaking design specifications.

4.3.3. *Transition probability enhancement*

Usually, adversaries exploit rare nodes with low transition probability (*TP*) in circuits to insert HTs. Therefore, reducing the number of rare nodes can reduce the possibility of HT insertion. Basically, there are two ways to increase the transition possibility. One is to insert the dummy scan flip-flops (dSFF), the other is to insert MUXs. In this paper we call the approach for increasing transition probability dummy circuit insertion (DCI). In what follows, we will describe these techniques in detail.

Salmani et al. [98] designed an approach to increase the transition probability of rare nodes by inserting dummy flip-flops. Firstly, the nodes with transition probability less than a specific threshold TP_{th} is identified. Then, dummy flip-flops are inserted to improve their transition probabilities. Fig. 12 shows the structure of dummy scan flip-flop (dSFF) in addition to an extra gate (AND or OR). If P_0 is less than P_1, a dDFF-AND block is placed to increase P_0, as depicted in Fig. 12(a). Conversely, if P_1 is less than P_0, an dSFF-OR gate is used to increase P_1, as in Fig. 12(b). Herein, P_0 and P_1 represent the signal probability being “0” and “1” of the input, respectively. According to the computation rule, the largest transition probability $TP = P_0 \times P_1$ for one signal net is 0.25 on the condition that $P_0 = P_1 = 0.5$.

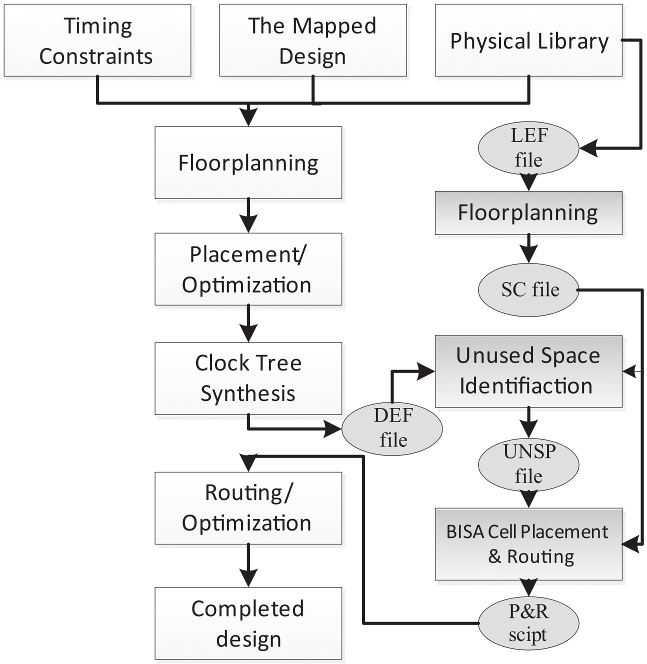


Fig. 11. BISA design flow [95].

Similarly, Zhou et al. [99] increased the node transition probability based on the insertion of 2-to-1 MUXs. Taking an AND gate to illustrate the improvement of the transition probability after inserting a 2-to-1 MUX, as shown in Fig. 13. The P_1 of the candidate net before MUX insertion can be computed in the following equation:

$$P_1 = \prod_{k=1}^{N} P_1^k \tag{2}$$

where P_1^k represents the signal probability being “1” of the kth input.

Assuming that the jth input of the candidate net Net_i has the smallest P_1^j, after applying insertion of 2-to-1 MUXs on the jth input, both $P_0^{j'}$ and $P_1^{j'}$ are equal to 0.5 because of importing random test patterns. In this way, the updated P_1 of the candidate net Net_i can be computed in the following equation:

$$P'_1 = P_1^{j'} \times \prod_{k \neq j}^{N} P_1^k \tag{3}$$

As we know, the candidate net Net_i with smaller transition probability TP depends on two cases: (1) $P_0^i \ll P_1^i$ and (2) $P_0^i \gg P_1^i$. Take case 1 as an example, suppose the jth input of Net_i with smallest P_1^j, then $\prod_{k \neq j}^{N} P_1^k$ is close to 1. By inserting a 2-to-1 MUX on jth input, P_1^j will be 0.5 and P'_1 tend to be 0.5 where we can get the highest transition probability at Net_i.

4.3.4. *Split manufacturing*

In split manufacturing, the front end of line (FEOL) layers (transistors and lower metal layers) is fabricated in advanced technology at an untrusted high-end foundry and the back end of

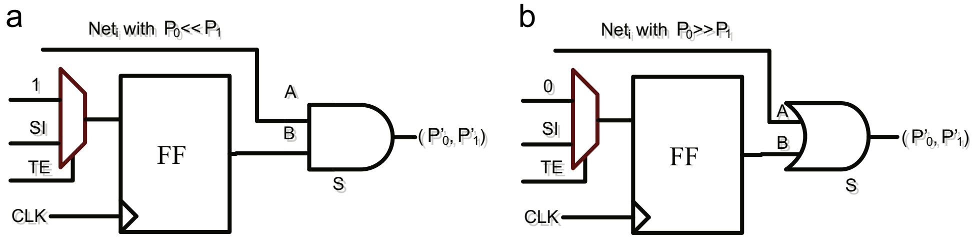


Fig. 12. The dummy flip–flop structures: (a) $P_0 \ll P_1$ and (b) $P_0 \gg P_1$ [97].

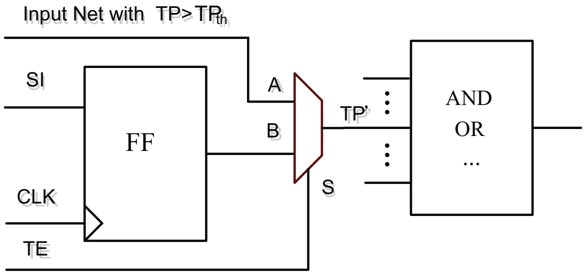


Fig. 13. The inserted 2-to-1 MUX circuit [98].

line (BEOL) layers (higher metal layers) is then fabricated at the design house's trusted low-end foundry [100]. Basically, split manufacturing tends to improve the security of an IC as FEOL and BEOL layers are fabricated separately in different foundries to hide the design intent and prevent malicious insertion [18]. This alleviates the HT attack in a single foundry from the whole process of the IC fabrication. For example, an attacker in the FEOL foundry cannot identify the suitable places in a circuit to insert Trojans without the BEOL layers. Split manufacturing is considered secure as it hides the BEOL connections from an attacker in the FEOL steps at an untrusted foundry [100]. However, Rajendran et al. exploited the heuristics used in typical floor planning, placement, and routing tools to bypass the security in the FEOL foundry where an attacker can connect 96% of the missing BEOL connections correctly. This is because floor planning & placement tools place the partitions as close as possible to reduce the delay between the two adjacent pins [101] so that attackers can find the connection information near the target pins. To overcome this vulnerability, a fault analysis-based defense was presented to rearrange the partition pins in order to deceive the FEOL attacker into making wrong BEOL connections [102].

4.4. *Runtime monitor*

Although lots of techniques have been developed for HT detection, diagnosis and prevention in the whole IC market model, they cannot guarantee to cover all potential HTs. It is still necessary to construct on-chip monitoring for HTs during run time [64]. Once the abnormal operation of the circuit happens, alert mechanism can shut the circuit function, and trigger other security measures to prevent further consequence caused by hardware Trojans. Analog sensors such as thermal sensor can also be exploited to detect deviations in power/thermal profiles caused by Trojan activation [63]. Runtime monitor has been comprehensively introduced in [13], where runtime monitor techniques are classified into three subclasses: configurable security monitors, variant-based parallel execution, and hardware-software approach. It is believed that the detection at the chip testing phase and run-time monitoring are complementary to detect Trojans.

5. Challenges and prospects

With advanced technologies, adversaries are likely to initiate new and unanticipated attacks which are difficult to be tackled by existing countermeasure approaches. Therefore, countermeasure techniques against HT attacks need further development, in order to win the race. To obtain the trustworthiness in the whole IC market model, several possible challenges and prospects for parties in the model are listed as follows.

- Efficient HT detection approaches for third party IPs at the pre-silicon stage are demanded by the SoC designers. Detecting HTs before integration will definitely reduce the complexity. The nature of external IPs, such as no golden reference model and many opportunities for HT insertion, makes the trust verification approaches face great challenges in terms of validity and efficiency. In addition, system level methods are also needed to identity and isolate Trojan IPs.
- Vulnerabilities from EDA tools are not widely concerned currently. Therefore, there is an underlying risk for the whole IC design cycle. The EDA tool vendors must provide the sufficient proof of the trustworthiness with their tools, and also the necessary design and verification tools to support cross trust verification.
- HT diagnosis approaches are a prospective research field, but accurate orientation of HTs is very difficult for large and complicated circuit designs. Efficient and clever diagnosis techniques are needed.
- With the increasing size of ICs, an adversary can exploit a large number of Trojan instances in various forms and sizes [103]. It can be extremely challenging to activate arbitrary Trojan instances and observe their effects in advanced technologies with process variations [13]. Therefore, design for trust is a more feasible solution for IC designers.
- It is suggested that combining HT detection, diagnosis, prevention and runtime monitoring will probably provide a complete solution to address the HT issues [13]. A systematic approach which supports hierarchical and synthetic application of various HT detection techniques is desired.
- In the future, it is necessary to build a trusted third party (TTP) with all necessary equipments and techniques to focus on the HT detection for end users.

6. Conclusion

In this survey, we elaborate an IC market model, and describe the HT threats at the interactions between parties involved in the model. We survey HT detection, diagnosis, prevention and runtime monitor approaches against the potential HT attacks. Finally, we discuss the challenges and the prospects for HT defense. In a word, tackling the threat of hardware Trojan will require long-term and tough endeavor. With proper approaches, we could gradually increase the difficulty and cost of HT attacks and even eliminate them, and leave HTs to the past.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grand nos. 61574099, 61232016, U1405254, 61303042, 61402162, 61402031 and the PAPD fund and Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education.

References

[1] S. Mitra, H.S.P. Wong, S. Wong, The Trojan-proof chip, Spectr. IEEE 52 (2) (2015) 46–51.
[2] The Quantum Program of NSA, Available on-line: ⟨http://www.nytimes.com/2014/01/15/us/nsa-effort-pries-open-computers-not-connected-to-internet.html⟩.
[3] R.S. Chakraborty, I. Saha, A. Palchaudhuri, G.K. Naik, Hardware Trojan insertion by direct modification of FPGA configuration bitstream, IEEE Des. Test 30 (2) (2013) 45–54.
[4] D.M. Shila, V. Venugopal, Design, implementation and security analysis of hardware Trojan threats in FPGA, In: IEEE International Conference on Communications (ICC), 2014, pp. 719–724.
[5] L.W. Kim, J.D. Villasenor, Dynamic function replacement for system-on-chip security in the presence of hardware-based attacks, IEEE Trans. Reliab. 63 (2) (2014) 661–675.
[6] M.R. Rudra, N.A. Daniel, V. Nagoorkar, D.H.K. Hoe, Designing stealthy Trojans with sequential logic: a stream cipher case study, In: 51st ACM/EDAC/IEEE Design Automation Conference (DAC), 2014, pp. 1–4.
[7] S. Bhasin, F. Regazzoni, A survey on hardware Trojan detection techniques, In: IEEE International Symposium on Circuits and Systems (ISCAS), 2015, pp. 2021–2024.
[8] DARPA, Technology Identifies Counterfeit Microelectronics, Available On-line: ⟨http://www.spacedaily.com/reports/DARPA_Technology_Identifies_Counterfeit_Microelectronics_999.html⟩.
[9] COST Action 1204, Trustworthy Manufacturing and Utilization of Secure Devices, Available On-line: ⟨http://www.cost.eu/COST_Actions/ict/Actions/IC1204?⟩.
[10] NSFC, Theory and Method of Potential Safety Hazard Detection in IC, Available On-line: ⟨http://isisn.nsfc.gov.cn/egrantindex/funcindex/prjsearch-list⟩.
[11] X. Wang, M. Tehranipoor, J. Plusquellic, Detecting malicious inclusions in secure hardware: challenges and solutions, In: IEEE International Workshop on Hardware-Oriented Security and Trust (HOST), 2008, pp. 15–19.
[12] M. Tehranipoor, F. Koushanfar, A survey of hardware Trojan taxonomy and detection, IEEE Des. Test 27 (1) (2010) 10–25.
[13] S. Bhunia, M.S. Hsiao, M. Banga, S. Narasimhan, Hardware Trojan attack: threat analysis and countermeasures, Proc. IEEE 102 (8) (2014) 1229–1247.
[14] N. Jacob, D. Merli, J. Heyszl, G. Sigl, Hardware Trojans: current challenges and approaches, Comput. Digit. Techn. IET 8 (6) (2014) 264–273.
[15] J. Francq, F. Frick, Introduction to hardware Trojan detection methods, In: Design, Automation and Test in Europe Conference and Exhibition (DATE), 2015, pp. 770–775.
[16] S. Dupuis, P.S. Ba, G. Di Natale, M.L. Flottes, B. Rouzeyre, A novel hardware logic encryption technique for thwarting illegal overproduction and hardware Trojans, In: IEEE 20th International On-Line Testing Symposium (IOLTS), 2014, pp. 49–54.
[17] M. Hicks, M. Finnicum, S.T. King, M. Martin, J.M. Smith, Overcoming an untrusted computing base: detecting and removing malicious hardware automatically, In: IEEE Symposium on Security and Privacy, 2010, pp. 159–172.
[18] M. Tehranipoor, R. Karri, F. Koushanfar, M. Potkonjak, TrustHub, Available On-line: ⟨https://www.trust-hub.org/⟩.
[19] J. Zhang, G. Qu, A survey on security and trust of FPGA-based systems, In: International Conference on Field-Programmable Technology (FPT), 2014, pp. 147–152.
[20] Y. Shiyanovskii, F. Wolff, A. Rajendran, C. Papachristou, D. Weyer, W. Clay, Process reliability based Trojans through NBTI and HCI effects, In: NASA/ESA Conference on Adaptive Hardware and Systems (AHS), 2010, pp. 215–222.
[21] G.T. Becker, F. Regazzoni, C. Paar, W.P. Burleson, Stealthy dopant-level hardware Trojans, In: Lecture Notes in Computer Science, vol. 4, no. 1, 2013, pp. 197-214.
[22] S.T. King, J. Tucek, A. Cozzie, C. Grier, W. Jiang, Y. Zhou, Designing and implementing malicious hardware, Leet (2008).
[23] L. Lin, M. Kasper, T. Gneysu, C. Paar, W. Burleson, Trojan side-channels: lightweight hardware Trojans through side-channel engineering, In: Lecture Notes in Computer Science, vol. 5747, 2009, pp. 382-395.
[24] G. Qu, Y. Yuan, Design things for the internet of things—an EDA perspective, In: IEEE/ACM International Conference on Computer-Aided Design (ICCAD), 2014, pp. 411–416.
[25] S. Skorobogatov, Hardware Assurance and Its Importance to National Security, Available On-line: ⟨http://www.cl.cam.ac.uk/sps32/secnews.html⟩.
[26] S. Wei, M. Potkonjak, Self-consistency and consistency-based detection and diagnosis of malicious circuitry, IEEE Trans. Very Large Scale Integr. Syst. 22 (9) (2014) 1845–1853.
[27] S. Narasimhan, S. Bhunia, Hardware Trojan detection, In: Introduction to Hardware Security and Trust, 2012, pp. 51–57.
[28] R. Torrance, D. James, The state-of-the-art in semiconductor reverse engineering, In: 48th ACM/EDAC/IEEE Design Automation Conference (DAC), 2011, pp. 333–338.
[29] F. Courbon, P. Loubet-Moundi, J.J.A. Fournier, A. Tria, A high efficiency hardware Trojan detection technique based on fast SEM imaging, In: Design, Automation and Test in Europe Conference and Exhibition (DATE), 2015, pp. 788–793.
[30] R.S. Chakraborty, F. Wolff, S. Paul, C. Papachristou, S. Bhunia, MERO: A Statistical Approach for Hardware Trojan Detection, Cryptographic Hardware and Embedded Systems (CHES), Springer, Berlin, Heidelberg, 2009.
[31] Bao Chongxi, D. Forte, A. Srivastava, On application of one-class SVM to reverse engineering-based hardware Trojan detection, In: 15th International Symposium on Quality Electronic Design (ISQED), 2014, pp. 47–54.
[32] Chipworks Inc, Semiconductor Manufacturing—Reverse Engineering of Semiconductor Components, Parts and Process, Available On-line: ⟨http://www.chipworks.com⟩.
[33] T. Reece, D.B. Limbrick, W.H. Robinson, Design comparison to identify malicious hardware in external intellectual property, In: 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, 2011, pp. 639–646.
[34] X. Zhang, M. Tehranipoor, Case study: detecting hardware Trojans in third-party digital IP cores, in: IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), 2011, pp. 67–70.
[35] C. Krieg, M. Rathmair, F. Schupfer, A process for the detection of design-level hardware trojans using verification methods, in: IEEE International Conference on High Performance Computing and Communications, 2014, pp. 729–734.
[36] A. Waksman, M. Suozzo, S. Sethumadhavan, FANCI: identification of stealthy malicious logic using boolean functional analysis, In: ACM SIGSAC Conference on Computer and Communications Security, 2013, pp. 697–708.
[37] J. Zhang, F. Yuan, L. Wei, Y. Liu, Q. Xu, VeriTrust: verification for hardware trust, IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. 34 (7) (2015) 1148–1161.
[38] J. Zhang, F. Yuan, Q. Xu, DeTrust: Defeating hardware trust verification with stealthy implicitly-triggered hardware trojans, In: ACM SIGSAC Conference on Computer and Communications Security, 2014, pp. 153–166.
[39] J. Xu, M. Williams, H. Mony, J. Baumgartner, Enhanced reachability analysis via automated dynamic netlist-based hint generation, In: Formal Methods in Computer-Aid Design (FMCAD), 2012, pp. 157–164.
[40] M. Rathmair, F. Schupfer, C. Krieg, Applied formal methods for hardware Trojan detection, In: IEEE International Symposium on Circuits and Systems (ISCAS), 2014, pp. 169–172.
[41] C. Bao, Y. Xie, A. Srivastava, A security-aware design scheme for better hardware Trojan detection sensitivity, In: IEEE International Symposium on Hardware Oriented Security and Trust (HOST), 2015, pp. 52–55.
[42] M. Oya, Y. Shi, M. Yanagisawa, N. Togawa, A score-based classification method for identifying Hardware-Trojans at gate-level netlists, In: Design, Automation and Test in Europe Conference and Exhibition (DATE), 2015, pp. 465–470.
[43] S. Yao, X. Chen, J. Zhang, Q. Liu, J. Wang, Q. Xu, y. Wang, H. Yang, FASTrust: feature analysis for third-party IP trust verification, In: IEEE International Test Conference (ITC), 2015, pp. 1–10.
[44] L. Ni, S. Li, J. Chen, P. Wei, Z. Zhao, The influence on sensitivity of hardware Trojans detection by test vector, In: IET Communications Security Conference (CSC), 2014, pp. 1–6.
[45] A.N. Nowroz, K. Hu, F. Koushanfar, S. Reda, Novel techniques for high-sensitivity hardware Trojan detection using thermal and power maps, IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. 33 (2014) 1792–1805.
[46] J. Zhang, G. Su, Y. Liu , L. Wei, F. Yuan, G. Bai, Q. Xu, On Trojan side channel design and identification, In: IEEE/ACM International Conference on Computer-Aided Design (ICCAD), 2014, pp. 278–285.
[47] S. Wei, M. Potkonjak, Scalable hardware Trojan diagnosis, IEEE Trans. Very Large Scale Integr. Syst. 20 (6) (2012) 1049–1057.
[48] Y. Cao, C.H. Chang, S. Chen, A. Cluster-Based, Distributed active current sensing circuit for hardware Trojan detection, IEEE Trans. Inf. Forens. Secur. 9 (2014) 2220–2231.
[49] M. Xue, A. Hu, G. Li, Detecting hardware Trojan through heuristic partition and activity driven test pattern generation, In: IET Communications Security Conference, 2014, pp. 1–6.
[50] N. Yoshimizu, Hardware Trojan detection by symmetry breaking in path delays, In: IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), 2014, pp. 107–111.
[51] B. Cha, S.K. Gupta, Efficient Trojan detection via calibration of process variations, In: IEEE 21st Asian Test Symposium, 2012, pp. 355–361.
[52] B. Cha, S.K. Gupta, Trojan detection via delay measurements: a new approach to select paths and vectors to maximize effectiveness and minimize cost, In: Design, Automation and Test in Europe Conference and Exhibition (DATE), 2013, pp. 1265–1270.
[53] S. Wei, K. Li, F. Koushanfar, M. Potkonjak , Provably complete hardware Trojan detection using test point insertion, In: IEEE/ACM International Conference on Computer-Aided Design (ICCAD), 2012, pp. 569–576.
[54] O. Soll, T. Korak, M. Muehlberghuber, M. Hutter, EM-based detection of hardware Trojans on FPGAs, In: IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), 2014, pp. 84–87.

[55] D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, B. Sunar, Trojan detection using IC fingerprinting, In: IEEE Symposium on Security and Privacy, 2007, pp. 296–310.

[56] F. Koushanfar, A. Mirhoseini, A unified framework for multimodal submodular integrated circuits Trojan detection, IEEE Trans. Inf. Forens. Secur. 6 (1) (2011) 162–174.

[57] C. Lavin, M. Padilla, J. Lamprecht, P. Lundrigan, B. Nelson, B. Hutchings, RapidSmith: do-it-yourself CAD tools for Xilinx FPGAs, In: 21st International Conference on Field Programmable Logic and Applications, 2011, pp. 349–355.

[58] X.T. Ngo, I. Exurville, S. Bhasin, J.L. Danger, S. Guilley, Z. Najm, J.B. Rigaud, B. Robisson, Hardware Trojan detection by delay and electromagnetic measurements, In: Design, Automation and Test in Europe Conference and Exhibition (DATE), 2015, pp. 782–787.

[59] I. Wilcox, F. Saqib, J. Plusquellic, GDS-II Trojan detection using multiple supply pad VDD and GND IDDQs in ASIC functional units, In: IEEE International Symposium on Hardware Oriented Security and Trust (HOST), 2015, pp. 144–150.

[60] S. Jha, S. Jha, Randomization based probabilistic approach to detect Trojan circuits, In: 11th High Assurance Systems Engineering Symposium, 2008, pp. 117–124.

[61] M. Banga, M.S. Hsiao, A novel sustained vector technique for the detection of hardware Trojan, In: 22nd International Conference on VLSI Design, 2009, pp. 327–332.

[62] H. Li, Q. Liu, Hardware Trojan detection acceleration based on word-level statistical properties management, In: International Conference on Field-Programmable Technology (FPT), 2014, pp. 153–160.

[63] C. Bao, D. Forte, A. Srivastava, Temperature tracking: toward robust run-time detection of hardware Trojans, IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. 34 (2015) 1577–1585.

[64] T.N. Xuan, J.L. Danger, S. Guilley, Z. Najm, O. Emery, Hardware property checker for run-time hardware Trojan detection, In: European Conference on Circuit Theory and Design (ECCTD), 2015, pp. 1–4.

[65] L. Li, H. Zhou, Structural transformation for best-possible obfuscation of sequential circuits, In: IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), 2013, pp. 55–60.

[66] B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S. Vadhan, K. Yang, On the (im)possibility of obfuscating programs, J. ACM 59 (2) (2012) 1–18.

[67] Shafi Goldwasser, Guy N. Rothblum, On best-possible obfuscation, J. Cryptol. 27 (3) (2013) 1–26.

[68] J.A. Roy, F. Koushanfar, I.L. Markov, EPIC: ending piracy of integrated circuits, In: Design, Automation and Test in Europe Conference and Exhibition, 2008, pp. 1069–1074.

[69] B. Liu, B. Wang, Reconfiguration-based VLSI design for security, IEEE J. Emerg. Select. Top. Circuits Syst. 5 (1) (2015) 1–11.

[70] J. Zhang, A practical logic obfuscation technique for hardware security, IEEE Trans Very Large Scale Integr. Syst. 99 (2015) 1–5.

[71] Y. Alkabani, F. Koushanfar, M. Potkonjak, Remote activation of ICs for piracy prevention and digital right management, In: IEEE/ACM International Conference on Computer-Aided Design, 2007, pp. 674–677.

[72] F. Koushanfar, Provably secure active IC metering techniques for piracy avoidance and digital rights management, IEEE Trans. Inf. Forens. Secur. 7 (1) (2012) 51–63.

[73] J. Zhang, Y. Lin, Y. Lyu, G. Qu, A PUF-FSM Binding scheme for FPGA IP protection and pay-per-device licensing, IEEE Trans. Inf. Forens. Secur. 10 (6) (2015) 1137–1150.

[74] Z. Xia, X. Wang, X. Sun, Q. Wang, A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data, IEEE Trans. Parallel Distrib. Syst. (2015), http://dx.doi.org/10.1109/TPDS.2015.2401003.

[75] Y. Ren, J. Shen, J. Wang, J. Han, S. Lee, Mutual verifiable provable data auditing in public cloud storage, J. Internet Technol. 16 (2) (2015) 317–323.

[76] J. Li, X. Li, B. Yang, X. Sun, Segmentation-based image copy-move forgery detection Scheme, IEEE Trans. Inf. Forens. Secur. 10 (3) (2015) 507–518.

[77] Z. Xia, X. Wang, X. Sun, Q. Liu, N. Xiong, Steganalysis of LSB matching using differences between nonadjacent pixels, Multimed. Tools Appl. (2014) 1–16.

[78] Z. Xia, X. Wang, X. Sun, B. Wang, Steganalysis of least significant bit matching using multi-order differences, Secur. Commun. Netw. 7 (8) (2014) 1283–1291.

[79] Z. Fu, X. Sun, Q. Liu, L. Zhou, J. Shu, Achieving efficient cloud search services: multi-keyword ranked search over encrypted cloud data supporting parallel computing, IEICE Trans. Commun. 98 (1) (2015) 190–200.

[80] P. Guo, J. Wang, B. Li, S. Lee, A variable threshold-value authentication architecture for wireless mesh networks, J. Internet Technol. 15 (6) (2014) 929–936.

[81] Z. Fu, K. Ren, J. Shu, X. Sun, F. Huang, Enabling Personalized Search over Encrypted Outsourced Data with Efficiency Improvement, IEEE Trans. Parallel Distrib. Syst. (2015), http://dx.doi.org/10.1109/TPDS.2015.2506573.

[82] B. Gu, V.S. Sheng, A Robust Regularization Path Algorithm for ν-Support Vector Classification, IEEE Transp Neural Networks and Learn. Syst. (2016), 10.1109/TNNLS.2016.2527796.

[83] T. Ma, J. Zhou, M. Tang, Y. Tian, A. AL-DHELAAN, M. AL-RODHAAN, S. Lee, Social network and tag sources based augmenting collaborative recommender system, IEICE Trans. Inf. Syst. 98 (4) (2015) 902–910.

[84] J. Shen, H. Tan, J. Wang, J. Wang, S. Lee, A. Novel Routing, Protocol providing good transmission reliability in underwater sensor networks, J. Internet Technol. 16 (1) (2015) 171–178.

[85] B. Gu, V.S. Sheng, K.Y. Tay, W. Romano, S. Li, Incremental support vector learning for ordinal regression, IEEE Trans. Neural Netw. Learn. Syst. 63 (2) (2015) 781–785.

[86] Z. Pan, Y. Zhang, S. Kwong, Efficient motion and disparity estimation optimization for low complexity multiview video coding, IEEE Trans. Broadcast. 61 (2) (2015) 166–176.

[87] B. Chen, H. Shu, G. Coatrieux, G. Chen, X. Sun, J. Coatrieux, Color image analysis by quaternion-type moments, J. Math. Imaging Vis. 51 (1) (2015) 124–144.

[88] X. Wen, L. Shao, Y. Xue, W. Fang, A rapid learning algorithm for vehicle classification, Inf. Sci. 295 (1) (2015) 395–406.

[89] Y. Zheng, B. Jeon, D. Xu, Q.M.J. Wu, H. Zhang, Image segmentation by generalized hierarchical fuzzy C-means algorithm, J. Intell. Fuzzy Syst. 28 (2) (2015) 961–973.

[90] B. Gu, X. Sun, V. Sheng, Structural Minimax Probability Machine, IEEE Transp Neural Networks and Learn. Syst. (2016), http://dx.doi.org/10.1109/TNNLS.2016.2544779.

[91] B. Gu, Victor, S. Sheng, W. Zhijie, H. Derek, O. Said, L. Shuo, Incremental learning for ν-Support Vector Regression, Neural Networks 67 (2015) 140–150.

[92] S. Xie, Y. Wang, Construction of Tree Network with Limited Delivery Latency in Homogeneous Wireless Sensor Networks, Wireless Pers. Commun. 78 (1) (2014) 231–246.

[93] J. Zhang, G. Qu, Y. Lv, Q. Zhou, A survey on silicon PUFs and recent advances in ring oscillator PUFs, J. Comput. Sci. Technol. 29 (4) (2014) 664–678.

[94] J. Rajendran, H. Zhang, C. Zhang, G.S. Rose, Youngok Pino, O. Sinanoglu, R. Karri, Fault analysis-based logic encryption, IEEE Trans. Comput. 64 (2) (2015) 410–424.

[95] E. Dubrova, M. Naslund, G. Carlsson, B. Smeets, Keyed logic BIST for Trojan detection in SoC, In: International Symposium on System-on-Chip (SoC), 2014, pp. 1–4.

[96] K. Xiao, D. Forte, M. Tehranipoor, A. Novel Built-In, Self-authentication technique to prevent inserting hardware Trojans, IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. 33 (12) (2014) 1778–1791.

[97] B. Khaleghi, A. Ahari, H. Asadi, S. Bayat-Sarmadi, FPGA-based protection scheme against hardware Trojan horse insertion using dummy logic, IEEE Embedded Syst. Lett. 7 (2015) 46–50.

[98] H. Salmani, M. Tehranipoor, J. Plusquellic, A. Novel, Technique for improving hardware Trojan detection and reducing trojan activation time, IEEE Trans. Very Large Scale Integr. Syst. 20 (1) (2012) 112–125.

[99] B. Zhou, W. Zhang, S. Thambipillai, J.K.J. Teo, A low cost acceleration method for hardware Trojan detection based on fan-out cone analysis, In: International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS), 2014, pp. 1–10.

[100] K. Vaidyanathan, R. Liu, E. Sumbul, Q. Zhu , F. Franchetti, L. Pileggi, Efficient and secure intellectual property (IP) design with split fabrication, In: IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), 2014, pp. 13–18.

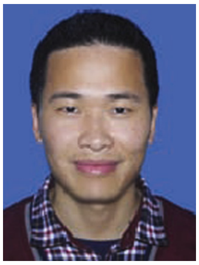
[101] N.A. Sherwani, Algorithms for VLSI Physical Design Automation, Kluwer Academic Publishers, 1995.

[102] J. Rajendran, O. Sinanoglu, R. Karri, Is split manufacturing secure?, In: Proceedings of the Conference on Design, Automation and Test in Europe EDA Consortium, 2013, pp. 1259–1264.

[103] S. Narasimhan, D. Du, R.S. Chakraborty, S. Paul, F.G. Wolff, C.A. Papachristou, K. Roy, S. Bhunia, Hardware Trojan detection by multiple-parameter side-channel analysis, IEEE Trans. Comput. 62 (11) (2013) 2183–2195.

He Li received the B.E. degree in Physics from Yangzhou University, Yangzhou, China, in 2013, and the master degree in the School of Electronic Information Engineering with Tianjin University, Tianjin, China, in 2015. His current research interests include hardware security, such as security for field-programmable gate arrays, hardware Trojan detection and IP verification techniques.

Qiang Liu received the Ph.D. degree from the Department of Electrical and Electronic Engineering at Imperial College London, London, UK, in 2008. From 2009 to 2011, he was a Research Associate in the Department of Computing at Imperial College London. He is currently an Associate Professor in School of Electronic Information Engineering at Tianjin University, with research interests in hardware security, VLSI design optimization, and reconfigurable computing.



Jiliang Zhang received the Ph.D. degree in Computer Science and Technology from Hunan University, Changsha, China in 2015. In 2013–2014, he worked as a research scholar at the Maryland Embedded Systems and Hardware Security Lab, University of Maryland, College Park. He is currently an Associate Professor in the Department of Information Security, Software College, Northeastern University, China. His research interests include hardware security, IOT/embedded security, hardware-assisted software security, and authentication.