Proceedings of the 35th European Safety and Reliability & the 33rd Society for Risk Analysis Europe Conference
Edited by Eirik Bjorheim Abrahamsen, Terje Aven, Frederic Bouder, Roger Flage, Marja Ylönen
©2025 ESREL SRA-E 2025 Organizers. *Published by* Research Publishing, Singapore.
doi: 10.3850/978-981-94-3281-3_ESREL-SRA-E2025-P1748-cd

Trojan-Free FPGA Hardware? The Challenge of End-User Verification

Andre Waltoft-Olsen
Norwegian Institute of Science and Technology, Norway

Lasse Øverlier
Norwegian Institute of Science and Technology, Norway

Geir Olav Dyrkolbotn
Norwegian Institute of Science and Technology, Norway

Ensuring the trustworthiness of Field Programmable Gate Arrays is essential for critical infrastructures' safe and reliable operation. However, the globalized nature of IC manufacturing introduces hardware trojan risks, creating challenges for end users tasked with ensuring hardware trustworthiness. This work explores the nature of the Field Programmable Gate Array hardware trojan threat across the IC supply chain. Detection and prevention methods are evaluated, focusing on their feasibility for end users. Although current research offers effective approaches, these methods are infeasible for end users. Our findings highlight a disconnect between vendor assurances, certification practices, and the methods for verifying hardware trustworthiness. We emphasize the need to bridge these gaps by developing end-user-feasible solutions to help verify the trustworthiness of FPGA-based systems.

Keywords: FPGA Trustworthiness, Hardware Trojans, Supply Chain Vulnerability

1. Introduction

Field-programmable gate Arrays (FPGAs) are being used more widely due to their flexibility, high-performance parallel processing, and reconfigurability. They are particularly effective for real-time tasks that require high reliability and low latency. Adoption across sectors, such as energy, underscores the growing need to secure and ensure trust in these devices.

A common approach to assuring trustworthy hardware involves setting security requirements during procurement. Achieving trustworthiness requires more than procurement stipulations. It begins with domain expertise to translate general risk descriptions into actionable assessments of complex threats, such as hardware trojans (HT). Feasible verification methods are then needed to address these threats effectively.

Security certifications in procurement requirements enhance the credibility and transparency of security measures implemented. While security certifications establish trust, their scope is inherently focused and cannot guarantee that hardware is free from vulnerabilities [11]. Moreover, demanding certifications in procurement processes may inadvertently restrict the pool of viable vendors. Manufacturers with significant market power might choose not to invest in obtaining certifications, thereby limiting procurement options. Consequently, trust is often placed in vendors to assure the security and integrity of FPGA hardware. However, this reliance is challenged by the focused scope of certifications and the fragmented nature of the global integrated circuit (IC) supply chain.

Bitstream and FPGA's physical hardware are foundational to achieving FPGA trustworthiness. The bitstream defines the routing and configuration of programmable logic, precisely representing the hardware's operational state. Countermeasures such as encryption, secure boot, and physical access are feasible methods to help safeguard the bitstream. Nevertheless, these alone cannot guarantee trustworthiness, as malicious actors can embed stealthy HTs into the FPGA IC. Consequently, this work focuses on HT threats within the FPGA IC, reviewing the feasibility of end-user methods to verify trustworthiness. It asks the question: What feasible methods can end users adopt to independently verify the trustworthiness of FPGA hardware? Our contribution is to examine current methods and assess their feasibility for end users seeking to verify that their FPGA hardware is trojan-free.

Understanding HTs' characteristics and operational contexts is crucial for assessing their risks. A strong foundational understanding enables end users to effectively evaluate the feasibility of existing detection and prevention techniques. The following sections delve into this foundation and review methods to evaluate end-user feasibility in ensuring trustworthy FPGA hardware. Section 2 analyzes the HT threat, focusing on the HT taxonomy development to provide a foundational understanding of its nature. Section 3 introduces our threat model. We evaluate end users' challenges in ensuring hardware trustworthiness with current methods. Finally, section 4 summarizes the findings and concludes the review.

2. FPGA Hardware Trojan Analysis

Wang et al. (2008) [14] presented an early HT taxonomy classifying them by physical, activation, and action characteristics. The physical characteristics describe their hardware manifestations, while the activation characteristics detail the criteria or conditions required to trigger the HT. The action characteristics describe the HT's impact.

Chakraborty et al. (2009) [5] enhanced the understanding of HT activation characteristics and provided a taxonomy for HT Triggers and Payloads leveraging the work by Wolff et al. [15]. Triggers are divided into Digital and Analog types. Digital triggers include, but are not limited to, logic changes in discrete states, specific patterns, or sequences of changes. Analog triggers are HT activations due to environmental conditions such as temperature changes, voltage levels, etc., and can occur gradually and naturally. Similarly, the Payload is divided into Digital and Analog categories. According to Chakraborty et al., the digital payload manipulates binary data such as gates, registers, memory content, state machines, etc. Analog Payload affects circuit parameters such as power and noise margin.

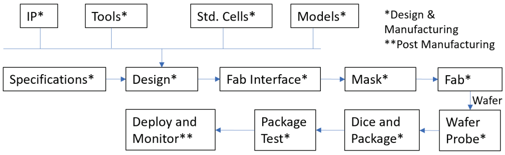

Fig. 1. IC life cycle provided by Chakraborty et al. [5]

Figure 1 illustrates the IC life cycle modeled by Chakraborty et al., within which detection methods are categorized as destructive and non-destructive. Destructive approaches encompass techniques like hardware reverse engineering (HRE), where physical hardware is analyzed through processes such as de-metallizing the chip and reconstructing its structure using advanced microscopy imaging. Non-destructive approaches are divided into Invasive and Non-invasive techniques. Invasive techniques modify the hardware design to incorporate features aimed explicitly at trojan detection or prevention. For example, hardware obfuscation obfuscates the design and challenges attackers from reverse engineering the circuit's functionality, making it difficult for them to identify optimal locations to implement the HT. Unlike Invasive techniques, Non-invasive techniques preserve the original design, making no alterations to the design or hardware. Examples are run-time monitoring, logic testing, and Side Channel Analysis (SCA). Non-invasive techniques often leverage Invasive techniques that implement design alterations for optimal HT detection.

Karri et al. (2010) [8] integrated the IC life cycle model from [5] with previous classifications and proposed a new HT taxonomy. The result is an enhanced HT Taxonomy that adds 1) Insertion phase, 2) Abstraction level, and 3) Location. 1) helps understand the relation between the IC life cycle stages and what vulnerabilities can be exploited. 2) The abstraction level identifies where the potential payload can be situated within the design hierarchy, as shown in Table 1. 3) The Location categorizes the locations in the hardware architecture at the system level where an HT can be inserted, such as the Processor, Memory, I/O, Power Supply, and Clock grid. This taxonomy reaches beyond the single IC level and can be applied to more complex designs such as System-on-Chip (SoC), System-in-Package (SiP), etc.

Table 1. Abstraction Level and potential Payload

Abstraction Level		HT Payload
System level	≻	Change protocol Modify design
Development environment	≻	Subvert synthesis, simulation, verification, validation
Register-transfer level	≻	Change logic
Gate Level	≻	Add/Modify gates
Transistor Level	≻	Modify parameters
Physical Level	≻	Modify Layout Modify wiring

Source: Karri et al.[8].

Furthermore, their work highlights the challenges encountered by various methods to ensure hardware trust. According to Karri et al., while reliability and fault tolerance methods are effective against natural failures, they are less effective at detecting malicious modifications like HTs. Design for Test seeks to improve the controllability and observability of manufacturing flaws, but achieving sufficient test coverage to detect malicious HTs, which fall outside traditional fault models, remains challenging. Tamper resistance and tamper evidence can raise the cost for attackers but cannot entirely prevent tampering. Logic verification is focused on identifying accidental errors, while design techniques aimed at preventing side-channel attacks depend heavily on manufacturers adhering to strict fabrication standards.

Fault attack countermeasures similarly rely on design-level techniques and trustworthy fabrication processes. Watermarking helps deter counterfeiting but struggles to detect small changes, such as HT insertion. HRE can determine whether a chip is HT-free, but it is not feasible on a large scale due to time and cost constraints. Physical Unclonable Functions (PUFs) are helpful for authentication and key generation, but detecting a maliciously replaced PUF in black-box testing remains difficult [8].

These limitations suggest that no single method is sufficient to counter the HT threat on its own. Moreover, the discussed countermeasures hinge on IC design and manufacturing. As a result, they leave critical gaps in the post-manufacturing phases. These gaps include the absence of end-user feasible detection methods, independent verification tools, and alternatives to reliance on manufacturer assurances.

Bhunia et al. (2014) [4] describe pre-silicon[a] and post-silicon as two phases during which any unauthorized modification of an IC should ideally be detectable. Pre-silicon detection relies on verification/simulation of the IC design, offering early detection of HTs, cost efficiency, design integrity, and reducing the likelihood of HT insertion during the subsequent phase. However, acquiring a golden model (GM)[b] for pre-silicon verification or simulation presents significant challenges due to the distributed and globalized IC supply chain, where IPs are often sourced from third-party vendors (3PiP). Furthermore, exhaustive verification is generally infeasible in chip systems with numerous interconnected components or modules. The complexity and sheer number of possible interactions in large designs make testing every potential scenario impractical as the number of test vectors grows exponentially.

Post-silicon detection involves specialized testing, such as SCA, where it is important to distinguish between accidental manufacturing faults and intentional modifications. Manufacturing imperfections are inherently non-deterministic, whereas HTs are deterministic and deliberately designed to evade detection. This distinction highlights the limitations of traditional testing during the design and manufacturing stages in detecting malicious HT insertions.

The threat posed by HTs is further amplified when compared to software trojans. Unlike software, which can often be patched or removed after deployment, IC HTs are embedded in hardware, making their in-field removal highly difficult.

A significant contribution of their work is the categorization of current HT countermeasures into the following groups:

[a] We understand pre-silicon as the phase before the IC's physical fabrication and entails the design process and the verification of the design functionality.

[b] A golden model is a trusted reference design created during the pre-silicon phase to validate functionality and detect deviations in testing.

(i) Trojan Detection Approaches.
(ii) Design for Security (DfS).
(iii) Run-time Monitoring.

i) Detection approaches involve Destructive methods such as HRE to unveil HTs post-silicon. However, HRE is again argued to be nonviable due to cost and time constraints. Non-destructive methods, like post-silicon logic testing and SCA, are deemed more viable. While pre-silicon versions of these methods exist, they still face the golden model challenge.

ii) DfS techniques aim to prevent trojan insertion or facilitate detection for post-fabrication validation methods. DfS is primarily implemented during the design and manufacturing stages and is argued to strengthen i) e.g., post-silicon trojan detection approaches such as logic testing and SCA.

iii) Run-time Monitoring operates beyond the design and manufacturing phases to detect and mitigate trojan activation by, for example, containing the threat by activating fail-safe mechanisms. However, these mechanisms hinge on the prerequisite of DfS techniques, which loop back into ii) and trusting the product supplier.

The categorization of countermeasures, along with the division of design and manufacturing into pre- and post-silicon phases, is a significant contribution for risk owners in strategizing against the HT threat. It highlights the importance of combining the three categories of countermeasures and addressing pre- and post-silicon phases to ensure their effectiveness. However, the efforts are still hinged in the IC design and manufacturing.

Xiao et al. (2016) [16] summarized a decade of research, expanding on earlier efforts to deepen the understanding of HT threats [14, 2, 5, 8, 3, 4]. Building upon the previous life cycle model, they introduced an enhanced semiconductor supply chain model, illustrated in Figure 2. While the model was designed for the semiconductor supply chain, we believe it adequately represents the FPGA IC supply chain when evaluating feasible methods for end users to ensure trustworthy FPGA hardware. The model outlines seven stages in producing semiconductor-based chips. Leveraging this semiconductor supply chain model, a comprehensive attack evaluation identifies the threats the various stakeholders face within the supply chain, as illustrated in Table 2.

Furthermore, Xiao et al. integrate new HT countermeasure research and enhance previous HT taxonomies. Functional Validation, Formal Verification, and Code/Circuit Coverage are pre-silicon detection techniques. Functional Validation applies simulation compared to test vectors used in Functional Tests but shares the same limitations related to triggering rare states in large and complex designs. Formal Verification deploys mathematical methods to prove the predefined security properties of the design but can fail to detect functionality outside the defined security logic. Code analysis checks the HDL code exercised during simulations attempting to cover all functional scenarios. On the other hand, Circuit Coverage evaluates how much of the physical circuit, including its structural elements like gates, nodes, and paths, is tested to detect faults and verify functionality. The main drawback of code and circuit coverage is that achieving high coverage does not guarantee functional correctness or the absence of defects. While they measure how thoroughly the code and hardware structure have been exercised, they may fail to verify the correctness of behavior and detect subtle errors, adding manual postprocessing to identify if unknown signals and gates are related to HTs.

For the DfS/DfT approach, prevention methods are expanded. In addition to Obfuscation and Functional Filler Cells, Camouflage is added. The main goal of the Camouflage technique is to protect from HT insertion by concealing functionality, altering layouts, and replacing unused spaces with testable circuitry.

Lastly, Split Manufacturing (SM) for Trust has been added as an anti-HT approach. SM secures IC designs by dividing fabrication between the untrusted Front-End-of-Line and trusted Back-End-of-Line foundries. This promotes a need-to-know strategy and hides critical interconnections, reducing the risk of trojan insertion or design theft. Techniques like 2D, 2.5D, and 3D manufacturing [12] have been researched to enhance feasibility.

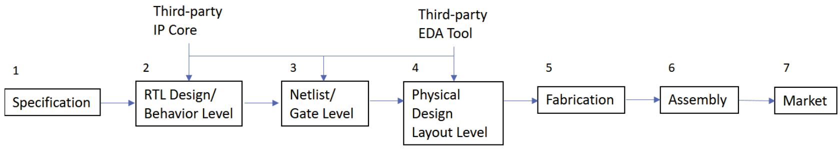

Fig. 2. The Semiconductor Supply Chain, as provided by Xiao et al. [16]

Table 2. Comprehensive Attack Models

Model	Description	3PIP Vendor	SoC Developer	Foundry
A	Untrusted 3PIP vendor	Untrusted	Trusted	Trusted
B	Untrusted foundry	Trusted	Trusted	Untrusted
C	Untrusted EDA tool or rogue employee	Trusted	Untrusted	Trusted
D	Commercial off-the-shelf component	Untrusted	Untrusted	Untrusted
E	Untrusted design house	Untrusted	Untrusted	Trusted
F	Fabless SoC design house	Untrusted	Trusted	Untrusted
G	Untrusted SoC developer with trusted IPs	Trusted	Untrusted	Untrusted

Source: Xiao et al.[16]

However, SM is not a universal solution for countering HT threats [10]. Moreover, performance trade-offs and costs have significant impacts.

2.1. *Emerging detection methods*

Emerging research shows promise for end users by advancing HT detection methods.

To enhance the coverage for formal verification, Wang et al.[13] proposed an SAT-based bounded model checking approach using Propositional Projection Temporal Logic (PPTL). The approach mitigates the state explosion problem. It relies on an accurate RTL design and correctly defined PPTL properties; flaws in either may cause critical issues to be overlooked. Additionally, it goes undetected if an HT does not violate the predefined properties. The finite exploration depth also means that dormant HTs may be missed, and the approach is limited to RTL-level anomalies. As a design-time verification method, it does not address post-synthesis or fabrication-time HTs.

At the RTL level, He et al. [7] propose a novel EM side-channel fingerprinting method for HT detection that eliminates the need for a GM and reduces the end user's involvement in the design and manufacturing phases. Instead, the method uses the genuine RTL code to generate EM signatures as a reference. However, the method predominantly focuses on detecting sequential HTs and may not effectively identify combinational HTs or payloads that do not significantly alter EM emissions. Although this approach removes the requirement for a GM, it still relies on an initial trusted state at the RTL level for reference. Challenges remain, including the need for specialized lab conditions to shield the environment from EM and noise interference, access to advanced measurement equipment, and expert knowledge and skills.

Krachenfels et al. [9] propose a novel approach for combating dormant HTs using Laser Logic State Imaging (LLSI) without requiring triggering or a GM. LLSI employs laser voltage imaging to extract logic states from transistors on a trusted chip. Modulating the supply voltage forces dormant logic to become visible, enabling the detection of parametric HTs, particularly those affecting routing, logic state transitions, and doping-level modifications. However, it is less effective against passive or highly subtle parametric HTs that do not significantly alter logic states. For effective analysis, the chip must be physically removed from the PCB and, in many cases, depackaged to expose the silicon die for direct laser probing. Additionally, power supply modulation is necessary to stimulate logic transitions. If the onboard voltage regulators do not support this, they may need to be bypassed or modified. LLSI is expensive and complex, requiring specialized laser microscopes, precise lab conditions, and expert handling.

Cheng et al. [6] introduce an innovative run-time monitoring technique for detecting HTs in FPGAs deployed in IoT devices. This approach leverages embedded temperature sensors to collect real-time data, using a predictive model to estimate expected frequency distributions. The method dynamically builds its reference from FPGA sensor readings without requiring a GM. However, it still relies on an initial trusted state for learning, making it vulnerable if an HT is present from the start. Additionally, its dependence on dynamically adjusting detection thresholds introduces the risk of miscalibration, potentially reducing accuracy. Since it detects HTs based on thermal behavior, logic-based attacks without notable power changes may evade detection.

The above advances significantly contribute to verifying trojan-free ICs but remain heavily dependent on the design and manufacturing stages. Moreover, effectively addressing the HT threat often necessitates a combinatorial deployment of countermeasures, drastically increasing cost and complexity and reducing feasibility.

3. FPGA Supply Chain: Stakeholders, Roles, and Threats

The globalized FPGA supply chain introduces vulnerabilities at multiple stages of the design and manufacturing process. These vulnerabilities create opportunities for HT insertion, posing significant risks. Chip suppliers within the semiconductor supply chain operate across various roles and business models. Foundries specialize in fabricating chips for other companies, and fabless firms focus solely on chip design and outsourcing production. Integrated Device Manufacturers combine design, manufacturing, and sales within a single organization, maintaining full control over the process. 3PiP providers contribute building blocks for chip designs, while EDA tool providers enable efficient design and optimization. Original Equipment Manufacturers integrate these chips into consumer products, relying on component distributors and resellers to ensure supply.

The threat model detailed in Table 3 assumes a fabless FPGA supplier and that the end user is past the design and manufacturing phases, concentrating on the FPGA IC design process from Specification to Assembly. Additionally, we include the Deployment stage to illustrate how HT exploitation can be facilitated by malicious hardware manipulation in earlier design stages.

3.1. *End users challenges*

Section 2 analyzed the HT threat, and according to [4], combining post-silicon validation with DfS techniques can achieve high levels of trust against HTs. However, this level of security is typically for extreme cases and is unlikely to be feasible for most defenders post-manufacturing. Such a security strategy requires comprehensive insight and control over all design and manufacturing stages. The globalization of the IC supply chain further complicates this, making it infeasible for end users to oversee all contributors involved.

Even with complete oversight and control, persuading a chip manufacturer to adjust ongoing production to accommodate tailored DfS solutions is highly unlikely. For instance, integrating real-time run-time monitoring to detect abnormal behavior would need to be planned during the design phase. While end users can request such features, they inevitably lead to increased production and testing costs and performance overhead. Additionally, the complexity of sourcing components from multiple providers makes uniform enforcement of DfS strategies across the supply chain exceptionally challenging. Ensuring transparency from all parties involved is further hindered by concerns over intellectual property and the risk of exposing proprietary technologies valued at millions.

Regardless of the FPGA supplier's business model (fabless, 3PiP, etc.), it is unrealistic for end users to demand added requirements in the supplier's design or manufacturing stages for security verification. Only a select few end users may have the leverage and resources to achieve the oversight and control required to ensure complete trust in these stages. Even if the design and manufacturing stages are trustworthy, the post-manufacturing ecosystem faces challenges from unauthorized or counterfeit suppliers, potentially jeopardizing the FPGA's trustworthiness. Given these constraints and looking at the attack models detailed in Table 2, we believe Attack Model D best represents end-

Table 3. Design stages, Stakeholders, and Threats.

Design Stage	Entity	Role	Threat Examples
Specification Stage	Fabless Company	Defines chip specifications and high-level design requirements	Manipulated specifications can introduce vulnerabilities that propagate throughout the design process.
RTL Design/Behavioral Level	Fabless Company, IP Core Provider	Develops RTL code; Provides reusable IP blocks for design integration	Trojan insertion or design flaws due to rogue designers, untrusted IP cores, or third-party design elements.
Netlist/Gate Level	EDA Tool Vendor	Converts RTL design into netlist or gate-level representation through synthesis	Malicious modifications during synthesis from untrusted EDA tools, potentially inserting Trojans or altering gate-level designs.
Physical Design/Layout Level	EDA Tool Vendor	Translates netlist into physical design layout for manufacturing	Trojans or security vulnerabilities introduced during layout generation via untrusted EDA tools, affecting the final chip design.
Fabrication and Post-Fabrication Testing	Foundry, Test Facility	The Foundry fabricates ICs on silicon wafers, and the Test Facility conducts functionality tests on fabricated chips	Hardware Trojans may be inserted during fabrication, and some vulnerabilities could go undetected during testing, allowing compromised chips to proceed to assembly. Compromised test tools could cloak HTs
Assembly	Assembly Facility	Assembles and packages the chip after testing	HTs may be introduced during chip assembly and packaging at untrusted facilities.
Deployment	OEM, Distributor	Integrates ICs into products and distributes them to the market	Physical or side-channel attacks can exploit vulnerabilities introduced during earlier stages, impacting the deployed products in the market. A dishonest integrator could counterfeit ICs

users' FPGA trust challenges. As a result, with current methods for ensuring trustworthiness, end users must rely on vendor trust when procuring FPGA hardware.

The HT threat area remains challenging to comprehend due to the limited number of publicly disseminated attacks. In this context, examining the threat from an attacker's perspective provides valuable insights. Xue et al. [17] shift the perspective to evaluate the feasibility of inserting HTs in real-world scenarios. Their research reveals that while HT insertion is achievable, it involves significant complexity and demands substantial expertise and resources, relying on a highly skilled and well-equipped attacker to compromise the IC supply chain. This perceived improbability can make it difficult for end users to justify the considerable costs and efforts required to mitigate HT threats. Nevertheless, research has underscored the potentially severe consequences for mission-critical systems if such an attacker were to exploit the HT threat successfully [1].

4. Conclusion

This work underscores the pressing need for feasible methods that enable end users to ensure trojan-free hardware. Our evaluation of current methods reveals significant limitations in their feasibility for end-user applications. While existing research predominantly focuses on countermeasures during the design and manufacturing stages, it leaves substantial gaps for end users operating beyond these phases. By addressing these gaps, we hope our work provides valuable insights into the barriers to achieving hardware trustworthiness and highlights the importance of future research to develop feasible, end-user-oriented solutions.

Artificial intelligence-based tools and automation are already enhancing the effectiveness of techniques like side-channel analysis and run-time monitoring. However, for end users who cannot influence design and manufacturing, there remains a critical need for methods that can be applied post-manufacturing. Advancing these efforts is essential for strengthening trust in FPGA-based systems and safeguarding future critical infrastructures.

References

1. Adee, S. (2008, May). The Hunt For The Kill Switch. *IEEE Spectrum 45*(5), 34–39.
2. Alkabani, Y. and F. Koushanfar (2008, June). Extended abstract: Designer's hardware Trojan horse. In *2008 IEEE International Workshop on Hardware-Oriented Security and Trust*, pp. 82–83.
3. Baumgarten, A., M. Steffen, M. Clausman, and J. Zambreno (2011, February). A case study in hardware Trojan design and implementation. *International Journal of Information Security 10*(1), 1–14.
4. Bhunia, S., M. S. Hsiao, M. Banga, and

S. Narasimhan (2014, August). Hardware Trojan Attacks: Threat Analysis and Countermeasures. *Proceedings of the IEEE 102*(8), 1229–1247. Publisher: Institute of Electrical and Electronics Engineers (IEEE).

5. Chakraborty, R. S., S. Narasimhan, and S. Bhunia (2009, November). Hardware Trojan: Threats and emerging solutions. In *2009 IEEE International High Level Design Validation and Test Workshop*, pp. 166–171. ISSN: 1552-6674.
6. Cheng, J., Q. Feng, C. Li, and W. Yang (2024, August). Securing FPGAs in IoT: a new run-time monitoring technique against hardware Trojan. *Wireless Networks 30*(6), 5455–5463.
7. He, J., H. Ma, Y. Liu, and Y. Zhao (2021, January). Golden Chip-Free Trojan Detection Leveraging Trojan Trigger's Side-Channel Fingerprinting. *ACM Transactions on Embedded Computing Systems 20*(1), 1–18.
8. Karri, R., J. Rajendran, K. Rosenfeld, and M. Tehranipoor (2010). Trustworthy Hardware: Identifying and Classifying Hardware Trojans. *Computer (Long Beach, Calif.) 43*(10), 39–46. Place: LOS ALAMITOS Publisher: IEEE.
9. Krachenfels, T., J.-P. Seifert, and S. Tajik (2021, November). Trojan Awakener: Detecting Dormant Malicious Hardware Using Laser Logic State Imaging. In *Proceedings of the 5th Workshop on Attacks and Solutions in Hardware Security*, Virtual Event Republic of Korea, pp. 17–27. ACM.
10. Rajendran, J., O. Sinanoglu, and R. Karri (2013, March). Is split manufacturing secure? In *2013 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1259–1264. ISSN: 1530-1591.
11. Ray, S. (2018). Hardware Trust in Industrial SoC Designs: Practice and Challenges. In S. Bhunia and M. M. Tehranipoor (Eds.), *The Hardware Trojan War: Attacks, Myths, and Defenses*, pp. 371–383. Cham: Springer International Publishing.
12. Vemuri, R. and S. Chen (2021). Split Manufacturing Methods. In R. Vemuri and S. Chen (Eds.), *Split Manufacturing of Integrated Circuits for Hardware Security and Trust: Methods, Attacks and Defenses*, pp. 1–29. Cham: Springer International Publishing.
13. Wang, W., N. Zhang, C. Tian, Z. Duan, Z. Xu, and C. Yu (2023). Verifying Chips Design at RTL Level. In C. David and M. Sun (Eds.), *Theoretical Aspects of Software Engineering*, Cham, pp. 146–163. Springer Nature Switzerland.
14. Wang, X., M. Tehranipoor, and J. Plusquellic (2008, June). Detecting malicious inclusions in secure hardware: Challenges and solutions. In *2008 IEEE International Workshop on Hardware-Oriented Security and Trust*, pp. 15–19.
15. Wolff, F., C. Papachristou, S. Bhunia, and R. S. Chakraborty (2008, March). Towards trojan-free trusted ICs: problem analysis and detection scheme. In *Proceedings of the conference on Design, automation and test in Europe*, DATE '08, New York, NY, USA, pp. 1362–1365. Association for Computing Machinery.
16. Xiao, K., D. Forte, Y. Jin, R. Karri, S. Bhunia, and M. Tehranipoor (2016, December). Hardware Trojans: Lessons Learned after One Decade of Research. *ACM Transactions on Design Automation of Electronic Systems 22*(1), 1–23. Publisher: Association for Computing Machinery (ACM).
17. Xue, M., C. Gu, W. Liu, S. Yu, and M. O'Neill (2020). Ten years of hardware Trojans: a survey from the attacker's perspective. *IET Computers & Digital Techniques 14*(6), 231–246.