Hardware constitutes the foundation of any computer system. Ensuring its integrity throughout the entirety of the hardware supply chain poses a significant challenge in establishing a secure computer system. The involvement of numerous untrusted parties in the process opens the door to vulnerabilities.

Integrity at Every Link: A Roadmap to Trustworthy Hardware Supply Chains

by Lennart M. Reimann, Dominik Sisejkovic and Rainer Leupers

The Integrated Circuits (ICs) supply chain uses a horizontal model, where Intellectual Property (IP) owners rely on external partners for competitiveness and cost reduction. However, this reliance raises significant trust concerns, including IP theft, IC counterfeiting, and the introduction of malicious circuit alterations (Hardware Trojans (HTs)) [1].

In the last decade, Hardware Trojans have emerged as a significant security concern [1][2]. These circuit alterations pose a threat by allowing unauthorized access, manipulation, and control of electronic systems. While practical instances of hardware Trojans are not conclusively documented, the ability to make subtle modifications with basic tools [3] emphasizes the need for robust security measures.

Over the past decades, numerous research programs have explored different methodologies aimed at ensuring trustworthiness throughout the IC supply chain. For instance, the United States Defense Advanced Research Project Agency (DARPA) has initiated multiple funding programs to advance R&D in the domain of reliable electronics. These programs include IRIS [4], TRUST [5], and SHIELD [6], among others. The significance of this issue has also been acknowledged in Germany, with the German Federal Ministry of Education and Research (BMBF) launching a funding program spanning 2021 to 2024, specifically aimed at addressing the challenges of dependable microelectronics for Germany and Europe [7] [8]. Regrettably, as of now, there is no established formal process for ensuring the trustworthiness of hardware across the IC supply chain. Consequently, it is required to assess existing protective measures and determine the necessary focus areas for future research, with the ultimate goal of guaranteeing the security of hardware throughout the supply chain. This article is based on the findings provided in [11].

Key Insights

- Design-dependent hardware Trojans are a fundamental security issue.
- Standard detection mechanisms only allow the identification of known hardware Trojans.
- Newer approaches focus on identifying malicious modifications by comparing the IP with a golden reference throughout the complete supply chain.
- Automated and complete reverse engineering is crucial for achieving formal security guarantees for the entire supply chain.

- Best-effort security measures implemented through active protection mechanisms remain an important pillar in protecting against malicious hardware modifications.

Key Recommendations

To holistically protect against (1) malicious hardware modifications (Trojans), we need to invest in (2) long-term research to formally secure the hardware supply chain and (3) short-term, best-effort security.

(1) Invest in a tangible estimation of the effort, tools, and skills required to design and insert design-specific hardware Trojans.

(2) Formal security guarantees:

- ○ Develop end-to-end, automatic, zero-fault, and non-destructive reverse engineering methods as key enabler for end-to-end equivalence checking.
- ○ Develop end-to-end and complete equivalence checking (EQ) methods from the abstract design specifications to the final physical device.

(3) Best-effort active protection mechanisms: advance the research on best-effort security with active protection mechanisms that aim at disallowing malicious modifications to the design throughout the hardware supply chain.

From Specification to Silicon

The hardware IP necessitates protection across the entire supply chain, spanning from initial specifications to the final device. The supply chain involves the IP owner, design house, foundry, assembly facility, OEMs, and users. The asset undergoes format changes via automated EDA tools or manual processes by hardware designers [9][10].

1. IP Creation and Ownership

- The process starts with the creation of hardware IP, which includes specifications, virtual prototypes, high-level descriptions, Register Transfer Level (RTL) designs, and gate-level netlists.
- This IP is initially owned by a development team, company, or individual who holds the rights to its use and distribution.

2. External Design House

- The IP owner may collaborate with an external design house for specialized expertise or additional resources.
- The IP owner provides either the RTL design or gate-level netlist to the external design house.

3. Design and Layout

- The external design house takes the provided RTL design or gate-level netlist and utilizes it to generate the layout in GDSII format.
- This layout serves as a blueprint for the physical components of the hardware.

4. Foundry and Manufacturing

- The GDSII layout is forwarded to a foundry, which is a specialized facility equipped for semiconductor fabrication.
- The foundry utilizes the layout to develop a mask of the chip design.
- The mask is used to manufacture the final chip using advanced semiconductor fabrication processes.

5. Assembly and Integration

- Once the chips have been produced, they are sent to an assembly facility.
- At this stage, the chips are combined with other electronic components to create either intermediate devices or the final product.

6. Packaging and Distribution

- The assembled devices undergo packaging to protect them from environmental factors and facilitate handling during transportation and use.
- Packaged devices are then distributed to original equipment manufacturers or directly to end users.

7. OEMs

- OEMs may further incorporate the hardware into larger systems or products.

8. End Users

- End users receive the final packaged devices, which they utilize for their intended purposes.

Throughout this journey, the original hardware IP undergoes a series of transformations from high-level descriptions to tangible physical components. It is crucial to ensure the integrity and security of the IP at each stage, as it may be exposed to vulnerabilities when handled by external parties (see Figure 1).

Hardware Trojans: A Fundamental Threat in the Hardware Supply Chain

The electronics supply chain, geared towards minimizing time-to-market and cutting costs, involves various external entities and closed-source third-party Electronic Design Automation (EDA) tools. This decentralized structure introduces security vulnerabilities due to inherent trust uncertainties among involved parties. A significant consequence is the potential introduction of malicious alterations known as hardware Trojans (HTs), an ongoing concern in security research for over a decade [2]. Unfortunately, the issue of HTs remains largely unresolved.

Hardware Trojans: A Subtle Change with Disastrous Consequences

Over ten years ago, a radar system in Syria failed to provide advance warning of an approaching airstrike, purportedly due to the presence of HTs in the defense systems [46]. While confirming the presence of HTs in such instances is challenging, the mere possibility of these subtle yet malicious design alterations has garnered significant attention in both research and industry. The US military and intelligence executives have identified HTs as one of the most significant threats the nation could encounter during times of war.

An HT is characterized by intentional, malicious, and covert alterations made to integrated circuits throughout the entire hardware supply chain [12]. This malicious behavior can take different forms, including information leakage, power dissipation, denial of service, performance degradation, or unintended behaviors. The intentional nature of the modification distinguishes it from random faults. Trojans are implemented stealthily to evade detection during tests and security checks.

Before we take a look at different classes of HTs, it is important to understand the process of Reverse Engineering (RE) - a term that is tightly coupled to HTs. Hardware RE is defined as the process of extracting a set of specifications for a hardware design by an entity other than the original design owner [42]. Consequently, RE has traditionally been associated with potentially malicious activities, such as IP theft, that are of significant

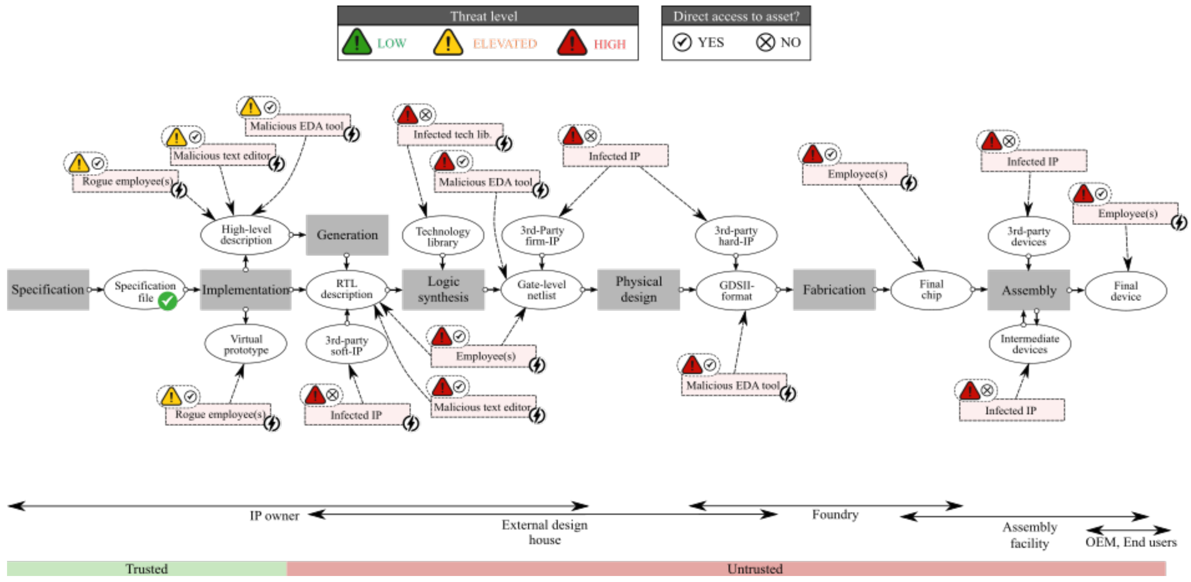

Figure 1 Assets and vulnerabilities within the hardware supply chain [11]

concern to governments, the military, and industry.

Within the context of RE, HTs can be categorized by addressing a fundamental question: does the design and insertion of the Trojan necessitate RE? This classification framework, as outlined in [14], divides all Trojans in two groups: Class-1 HTs (C1HTs) and Class-2 HTs (C2HTs). C1HTs encompass Trojans that rely on RE. As a result, the attacker needs to invest into comprehending the design specifications of the asset at various levels to create an HT tailored to that specific design. Consequently, a Class-1 HT enables a controllable trigger, paving the way for a high-impact attack. C2HTs consist of less hidden Trojans that do not rely on RE. Consequently, an attacker can insert these HTs into a design at any stage or level of abstraction without possessing any knowledge about the asset. Thus, C2HTs remain resilient against potential protection mechanisms.

The Untrusted Design House and Foundry

One of the most common threat scenarios within the hardware supply chain involves the malicious actions of external design houses and foundries [1] [13] [40]. Since these third-party entities are often located at remote sites around the world and lack a verifiable level of trust in the design and production process, they present a potential vulnerability for malicious alterations, such as HTs. Both external design houses and foundries receive the asset in a form that remains modifiable before it is permanently coded into silicon. What are our assumptions about the adversaries' capabilities? In general, the following is true for both an external design house and foundry: (1) The entity is granted complete access to the design. The external design house is provided with either the RTL or gate-level design, while the foundry receives the final layout. (2) The entity functions without any limitations or oversight by the legitimate IP owner. (3) The insertion of a design-specific (class 1) HT demands a certain level of RE effort.

A crucial element influencing the characteristics of the introduced Trojan is RE. The primary aim of RE is to attain an abstraction level of the asset that allows for subsequent analysis and manipulation. As the asset's abstraction level decreases, the need for a more extensive effort becomes apparent to gain a comprehensive understanding of the design and, potentially, to insert class-1 HTs. The RE process contains a multitude of manual, semi-automatic, and automatic steps which all paint only part of a still error-prone, lengthy picture. Thus, a completely automated, non-destructive, and flawless RE process still remains elusive despite being the key factor in determining the effectiveness of an active protection against HTs.

Open Challenges: Reverse Engineering

- How to quantify the cost, required effort, and complexity of reverse engineering?
- How to quantify the success criteria and the amount of retrieved information of the reverse engineering process?
- How to implement non-destructive, automatic, and zero-fault RE for every abstraction level of the asset?

Fighting Hardware Trojans

In the pursuit of combating HTs, research takes two distinct approaches. While some design methodologies focus on detecting these insidious elements for subsequent removal, achieving foolproof detection faces formidable challenges along the hardware supply chain. Consequently, many researchers are now delving into proactive measures aimed at safeguarding against malicious modifications in the first place.

Detecting Hardware Trojans

A high number of methodologies aim to prevent HTs by identifying malicious modifications in the asset and removing them. Detection mechanisms fall into two classes: pre-silicon and post-silicon. Pre-silicon methods focus on design analysis before manufacturing, while post-silicon mechanisms target the manufactured design.

Pre-Silicon Detection

Pre-silicon detection mechanisms can be summarized with the following four major techniques: Code coverage analysis, formal verification techniques, structural analysis, and functional analysis.

In this context, formal verification stands out as the most promising approach for thoroughly proving the absence of hardware Trojans. Formal verification techniques like equivalence checking can mathematically prove properties and equivalence between two descriptions of the asset. This allows detecting discrepancies indicating potential

Trojans. Existing verification methods like security assertions can also be reused for Trojan detection [17] [18] [45]. However, limited abstraction details and a lack of formal models for manufactured chips constrain formal techniques. A common formal method used in this context is equivalence checking. Equivalence checking in hardware security verifies that two different representations of a design perform the same operations and produce the same results [15] [16]. It ensures the integrity and trustworthiness of a design, identifying any discrepancies or potential malicious modifications. Therefore, it allows comparing two descriptions of the same IP between any two steps in the supply chain.

Moreover, approaches like code coverage analysis, structural analysis, and functional tests are employed to identify hardware Trojans [19][20]. However, due to their limitations in providing comprehensive security assurance or necessitating specific Trojan structure information, there is an increasing research emphasis on formal methods.

Post-Silicon Detection

Functional tests execute application tests on the chip to check for incorrect behaviors indicating potential Trojans. But tests may not trigger all Trojans, and some may not corrupt functionality.

After the chips are manufactured every device needs to be tested for modifications instead of analyzing a single hardware description. Therefore, researchers work on reverse-engineering to yield a formal description of the produced hardware. Reverse engineering involves delayering and imaging the manufactured chip layer-by-layer to reconstruct a gate-level netlist [1] and apply pre-silicon analysis. However, RE is expensive, time-consuming, and destructive.

Overall, post-silicon detection has limited coverage compared to pre-silicon techniques. RE provides a gate-level netlist but is expensive and destructive. More research is needed to enable post-silicon security guarantees.

Open Challenges: Passive Detection Mechanisms

- Does equivalence checking offer a complete assurance of the absence of malicious modifications?
- Is there a detection scheme capable of identifying any potential hardware Trojan?

Protection Against Malicious Design Modifications

Many Design-for-Trust (DfTr) methodologies have been introduced in the last decades to protect hardware against malicious modifications, including functional filler cells [37] [39], split manufacturing [40] [41], and layout camouflaging [42] [43]. In the following, we will, however, only take a closer look at logic locking – a premier technique to circumvent the insertion of class 1 HTs, and the only active protection mechanism capable of protecting against untrusted entities throughout the microelectronics supply chain

Logic Locking (LL) aims to protect the integrity of hardware designs at different supply-chain stages and design levels [38]. LL modifies the hardware design through the incorporation of logic alterations that link the proper functioning of the chip to a confidential activation key. This alteration carries two primary consequences. First, the functional behavior of the HW design is contingent upon the correctness of the key. When the correct key is applied, the design operates as intended. Second, the inclusion of key-dependent logic brings about structural modifications in the design, essentially "obfuscating" the hardware.

How is LL applied? Let’s assume that logic locking is implemented at the gate level. The IP owner – the trusted entity - aims to develop a legitimate chip. During this stage, the RTL description of the hardware design is logically synthesized into a gate-level netlist. At that point, logic locking is applied, resulting in a locked netlist and a secret key. The secret key remains exclusively with the legitimate IP owner. Note that the key is not needed for any subsequent steps. The locked netlist is then provided to external parties for layout generation, fabrication, and assembly. Once the final chip is prepared, the IP owner performs activation. The secret key is incorporated into the chip using a non-volatile memory, such as flash, e-fuse, or EEPROM [23]. This process has been successfully implemented by HENSOLDT Cyber GmbH through the production of the "Made in Germany RISC-V" (MiG-V) processor—a groundbreaking example of a fully logic-locked commercial processor [14] [24] [25] [32].

The Interplay of Reverse Engineering and Logic Locking

A lot of research has popped up around the topic of logic locking, including resilient schemes design and novel key recovery attacks [44]. Unfortunately, one mistake is still being repeated: the security of LL is only seen through the recoverability of the key. Why is this a problem? The key itself should evidently not be recoverable before production; otherwise, an attacker could simply remove the locking-induced structures and dissolve the impact of the obfuscation. However, focusing on "how difficult is it to retrieve the key" as a measure of security is somewhat misleading as it completely ignores the main objective of LL: making reverse engineering harder to perform. As a result, the concept of the key's "retrievability" has often been employed as an indicator of security. However, the question of "how much more challenging RE becomes" because of logic locking has remained unanswered.

Secret Key, Unsecure Storage

A fundamental issue in logic locking hides in the availability of a secure key storage. Unfortunately, a growing number of physical attacks have successfully shown that the correct key can be extracted from an activated chip through probing and fault-injection attacks [27] [28] [29]. Moreover, it is possible to design a design-independent HT that leaks the key value after the chip is activated simply by forwarding the key inputs to an observable output [30]. Hence, without a secure key storage, logic locking will have a very limited effect in a high-volume production setting which allows the availability of activated chips on the market.

Universal Circuits: A Way Out?

A promising approach to addressing many challenges in logic locking is rooted in the concept of universal circuits [26] [31]. Drawing from a cryptographic primitive introduced by Valiant [33], universal circuits can be programmed to emulate any circuit within a specified size limit. From a security perspective, a universal circuit can represent a wide range of hardware functionalities while consistently maintaining the same underlying structure. In fact, universal circuits could be seen as "the ultimate" obfuscation. Why is this interesting? When all the circuits entrusted to a potentially untrustworthy foundry or external design house share the same physical structure, irrespective of the functionality programmed by a secret key, an attacker has only one avenue for introducing modifications: random, design-independent, and most likely low-impact class 2 HTs. Regrettably, the expenses associated with the implementation of this approach far exceed acceptable levels. As a compromise, an alternative solution has been examined using Embedded Field-Programmable Gate Arrays (eFPGAs). In this setup, specific security-critical design modules are substituted with fully reconfigurable soft eFPGA or pre-designed eFPGA hard macros [22] [34]. However, additional research is imperative to ascertain the security and cost-effectiveness of FPGA-based obfuscation [22] [35] [36].

Open Challenges: Active Protection Mechanisms

- How to protect the activation key in logic locking from physical attacks?
- How to measure the impact of logic locking on the required reverse-engineering effort?
- How to design cost-effective and generic indistinguishable circuits?

The Silver Bullet: Formal Guarantees

The ultimate goal in security is to achieve formal guarantees for the absence of malicious modifications. Formal verification is the major approach to achieving this goal. However, although a mathematical proof for the analysis is given, there are a few points that are still open for research to achieve this complete guarantee.

1. As the IP changes its level of abstraction throughout the hardware supply chain, the discrepancy between the levels of abstraction needs to be considered.
2. Manufactured chips do not offer a formal description of the underlying hardware so that reverse engineering needs to be advanced further to offer a non-destructible, fast and complete solutions to generate a description for every manufactured device
3. A complete chain of formal verification tools is required to cover the entire hardware supply chain. This needs to be standardized and further development is required.

Research Needs: The Way Forward

To establish a secure microelectronics supply chain, it is essential to address two fundamental challenges. (1) The first challenge revolves around formally securing the entire supply chain and achieving mathematically proven security assurances across the entirety of the hardware design and manufacturing process. Undoubtedly, pursuing this objective embodies the highest level of security, although it may involve high risks and long-term projects. (2) Consequently, it is equally vital to promote research endeavors focusing on lower-risk, short-term projects that aim to provide best-effort security. These two overarching goals are further elucidated below, following the visualization in Figure 2.

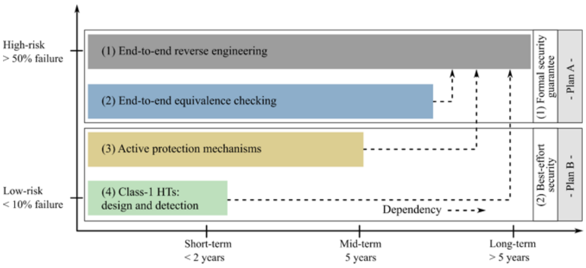

Figure 2 Research goals for achieving a secure hardware supply chain [11].

Reaching Formal Security Guarantees

To enable a formally secure hardware design and fabrication flow, the following must be achieved:

Need 1: A fully automated, error-free, non-destructive, and seamless reverse engineering process spanning from the physical device to high-level abstractions is of paramount importance. The establishment of a comprehensive RE workflow, starting from the final chip, serves as a critical facilitator for comprehensive Equivalence Checking (EQ), HT detection, and the evaluation of active protection approaches. This research gap includes the following tasks:

a. Estimation of the complexity, cost, and time effort of reverse engineering;

b. Design and implementation of fully automatic reverse engineering methodologies.

Need 2: Achieving end-to-end and comprehensive EQ spanning from high-level abstractions down to the ultimate physical device is imperative. This EQ continuum guarantees that the final, manufactured, packaged, and embedded device maintains complete equivalence with its initial design specifications. This research gap includes the following tasks:

a. Evaluation of the influence of the abstraction level and the design details it provides on the effectiveness of equivalence checking;

b. Introducing formats and standards that enable equivalence checking at high abstraction levels;

c. Introducing formats, standards, and methods to enable post-fabrication equivalence checking;

d. Offering open-source, verifiable, and trustworthy equivalence-checking tools.

Supporting Best-Effort Security

It is crucial to back research endeavors addressing lower-risk gaps that may not lead to formal and all-encompassing security assurances but contribute to best-effort security. These research objectives encompass:

Need 3: Active protection mechanisms may not offer formal security assurances, yet they represent a crucial and currently the sole line of defense against malevolent alterations in the course of external design and manufacturing stages. Nonetheless, the efficiency of active protection methods is tethered to several unresolved inquiries, such as:

a. Enabling secure key storage solutions that are resilient against physical attacks;

b. Supporting the development of cost-efficient universal circuits or approximations thereof in the form of reconfigurable circuits;

c. Evaluate the impact of active protection mechanisms on the reverse engineering effort;

d. Evaluate the possibility of formally secure active protection mechanisms.

Need 4: The design and detection of class-1 HTs remains a focal point in security research. To facilitate this goal, the following must be considered:

a. A concrete assessment of the resources, tools, and expertise needed for the creation and insertion of Class-1 HTs;

b. Evaluation of the design of HTs that could potentially circumvent equivalence checking;

c. Support post-fabrication, non-equivalence-checking-based HT detection methods.

References

[1] Swarup Bhunia and Mark Tehranipoor. Hardware Security: A Hands-on Learning Approach. 1st. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2018.

[2] Swarup Bhunia and M Tehranipoor. "The Hardware Trojan War: Attacks, Myths, and Defenses". In: Springer International Publishing (2018). doi: 10.1007/978-3-319- 68511-3.

[3] Tiago Perez and Samuel Pagliarini. "Hardware Trojan Insertion in Finalized Layouts: a Silicon Demonstration". In: arXiv, 2021. doi: 10.48550/ARXIV.2112.02972

[4] Defense Advanced Research Project Agency (DARP). Integrity and Reliability of Integrated Circuits (IRIS). https://www.darpa.mil/program/integrity-and-reliability-of-integrated-circuits. accessed: July 2021.

[5] Defense Advanced Research Project Agency (DARP). Trusted Integrated Circuits (TRUST). https://www.darpa.mil/program/trusted-integrated-circuits. accessed: July 2021

[6] Defense Advanced Research Project Agency (DARP). Supply Chain Hardware Integrity for Electronics Defense (SHIELD). https://www.darpa.mil/program/supply-chainhardware-integrity-for-electronics-defense. accessed: July 2021.

[7] Bundesministerium für Bildung und Forschung (BMBF). Mikroelektronik. Vertrauenswürdig und nachhaltig. Für Deutschland und Europa. Rahmenprogramm der Bundesregierung für Forschung und Innovation 2021-2024. https://www.elektronikforschung.de/rahmenprogramm. accessed: July 2021.

[8] Bundesministerium für Bildung und Forschung (BMBF). Vertrauenswürdige Elektronik. Forschung und Innovation für technologische Souveränität. https://www.elektronikforschung.de/service/publikationen/vertrauenswuerdige-elektronik. accessed: July 2021.

[9] Bicky Shakya et al. "Benchmarking of hardware trojans and maliciously affected circuits". In: Journal of Hardware and Systems Security 1.1 (2017), pp. 85–102. d o i: 10.1007/s41635-017-0001-6.

[10] Imran Abbasi et al. "TrojanZero: Switching Activity-Aware Design of Undetectable Hardware Trojans with Zero Power and Area Footprint". In: Mar. 2019, pp. 914–919. d o i:10.23919/DATE.2019.8714829.

[11] Dominik Sisejkovic et al. Agentur für Innovation in der Cybersicherheit GmbH. Ecosystem for Trustworthy IT; Los 4: Formal Security Guarantees for Trustworthy Hardware Supply Chains. https://www.cyberagentur.de/wp-content/uploads/2023/07/OevIT-Vorstudien-Los-4.pdf

[12] K. Xiao et al. "Hardware Trojans: Lessons Learned after One Decade of Research". In: ACM Trans. Des. Autom. Electron. Syst. 22.1 (May 2016). doi: 10.1145/2906147.

[13] Jan-Peter Kleinhans and Nurzat Baisakova. The global semiconductor value chain. https://www.stiftung-nv.de/sites/default/files/the_global_semiconductor_value_chain.pdf. Accessed: Oct. 2020.

[14] Dominik Sisejkovic and Rainer Leupers. Logic Locking: A Practical Approach to Secure Hardware. In: Springer Cham (2022). doi: https://doi.org/10.1007/978-3-031-19123-7[42] M. G. Rekoff. "On reverse engineering". In: IEEE Transactions on Systems, Man, and Cybernetics SMC-15.2 (1985), pp. 244–252. doi: 10.1109/TSMC.1985.6313354.

[15] Synopsys Inc. Formality. https://www.synopsys.com/implementation-and-signoff/signoff/formality-equivalence-checking.html. Oct. 2022.

[16] Claire Wolf. Yosys. http://bygone.clairexen.net/yosys/documentation.html. Oct. 2022.

[17] Lennart M. Reimann et al. "QFlow: Quantitative Information Flow for Security-Aware Hardware Design in Verilog". In: 2021 IEEE 39th International Conference on Computer Design (ICCD). 2021, pp. 603–607. d o i: 10.1109/ICCD53106.2021.00097

[18] L. M. Reimann, S. Erdönmez, D. Sisejkovic and R. Leupers, "Quantitative Information Flow for Hardware: Advancing the Attack Landscape," *2023 IEEE 14th Latin America Symposium on Circuits and Systems (LASCAS)*, Quito, Ecuador, 2023, pp. 1-4, doi: 10.1109/LASCAS56464.2023.10108235.

[19] Huili Chen et al. AdaTest: Reinforcement Learning and Adaptive Sampling for On-chip Hardware Trojan Detection. 2022. d o i: 10.48550/ARXIV.2204.06117.

[20] Susmit Jha and Sumit Kumar Jha. "Randomization Based Probabilistic Approach to Detect Trojan Circuits". In: 2008 11th IEEE High Assurance Systems Engineering Symposium. 2008, pp. 117–124. d o i: 10.1109/HASE.2008.37.

[21] Synopsys Inc. TestMAX. https://www.synopsys.com/implementation-and-signoff/test-automation/testmax-atpg.html. Oct. 2022.

[22] Hadi Mardani Kamali et al. "Advances in Logic Locking: Past, Present, and Prospects". In: Cryptology ePrint Archive (2022)

[23] M. Tanjidur Rahman et al. "Defense-in-Depth: A Recipe for Logic Locking to Prevail". In: Integr. VLSI J. 72.C (May 2020), pp. 39–57. doi: 10.1016/j.vlsi.2019.12.007.

[24] HENSOLDT Cyber GmbH. Press Release: HENSOLDT Cyber presents MiG-V, the first RISC-V Processor "Made in Germany" for Security Applications. https://hensoldt-cyber.com/wp-content/uploads/2020/05/20200515-HENSOLDT-Cyber-PM-MiG-V-is-ready-1.pdf. Accessed: May 2020.

[25] Dominik Sisejkovic et al. "A Secure Hardware-Software Solution Based on RISC-V, Logic Locking and Microkernel". In: Proceedings of the 23th International Workshop on Software and Compilers for Embedded Systems. SCOPES '20. St. Goar, Germany: Association for Computing Machinery, 2020, pp. 62–65. doi: 10.1145/3378678.3391886.

[26] Kaveh Shamsi, David Z. Pan, and Yier Jin. "On the Impossibility of Approximation Resilient Circuit Locking". In: 2019 IEEE International Symposium on Hardware Oriented Security and Trust (HOST). 2019, pp. 161–170. doi: 10.1109/HST.2019. 8741035.

[27] Susanne Engels, Max Hoffmann, and Christof Paar. The End of Logic Locking? A Critical View on the Security of Logic Locking. Cryptology ePrint Archive, Report 2019/796. https://eprint.iacr.org/2019/796. 2019.

[28] M. T. Rahman et al. "The Key is Left under the Mat: On the Inappropriate Security Assumption of Logic Locking Schemes". In: 2020 IEEE HOST. 2020, pp. 262–272. doi: 10.1109/ HOST45689.2020.9300258.

[29] A. Jain, M. T. Rahman, and U. Guin. "ATPG-Guided Fault Injection Attacks on Logic Locking". In: 2020 IEEE Physical Assurance and Inspection of Electronics (PAINE). 2020, pp. 1–6. doi: 10.1109/PAINE49178.2020.9337734.

[30] Ayush Jain, Ziqi Zhou, and Ujjwal Guin. "TAAL: Tampering Attack on Any Key-Based Logic Locked Circuits". In: 26.4 (Mar. 2021). doi: 10.1145/3442379.

[31] Elisaweta Masserova et al. "Logic Locking-Connecting Theory and Practice". In: Cryptology ePrint Archive (2022).

[32] Šišejković, D., Merchant, F., Reimann, L.M., Leupers, R., Kegreiß, S. (2020). Scaling Logic Locking Schemes to Multi-module Hardware Designs. In: Brinkmann, A., Karl, W., Lankes, S., Tomforde, S., Pionteck, T., Trinitis, C. (eds) Architecture of Computing Systems – ARCS 2020. ARCS 2020. Lecture Notes in Computer Science(), vol 12155. Springer, Cham. https://doi.org/10.1007/978-3-030-52794-5_11

[33] Leslie G. Valiant. "Universal Circuits (Preliminary Report)". In: Proceedings of the Eighth Annual ACM Symposium on Theory of Computing. STOC '76. Hershey, Pennsylvania, USA: Association for Computing Machinery, 1976, pp. 196–203. doi: 10.1145/800113. 803649.

[34] Jitendra Bhandari et al. Not All Fabrics Are Created Equal: Exploring eFPGA Parameters For IP Redaction. 2021. doi: 10.48550/ARXIV.2111.04222.

[35] Gaurav Kolhe et al. "Breaking the Design and Security Trade-off of Look-up-Table–Based Obfuscation". In: ACM Trans. Des. Autom. Electron. Syst. 27.6 (June 2022). doi: 10.1145/3510421.

[36] Gaurav Kolhe et al. "Security and Complexity Analysis of LUT-based Obfuscation: From Blueprint to Reality". In: 2019 IEEE/ACM ICCAD. 2019, pp. 1–8. doi: 10.1109/ICCAD45719.2019.8942100.

[37] K. Xiao and M. Tehranipoor. "BISA: Built-in self-authentication for preventing hardware Trojan insertion". In: 2013 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST). 2013, pp. 45–50. doi: 10.1109/HST.2013.6581564.

[38] D. Sisejkovic and R. Leupers, "Trustworthy Hardware Design with Logic Locking," 2021 IFIP/IEEE 29th VLSI-SoC, Singapore, Singapore, 2021, pp. 1-2, doi: 10.1109/VLSI-SoC53125.2021.9606998.

[39] K. Xiao, D. Forte, and M. Tehranipoor. "A Novel Built-In Self-Authentication Technique to Prevent Inserting Hardware Trojans". In: IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 33.12 (2014), pp. 1778–1791. doi: 10.1109/ TCAD.2014.2356453.

[40] Richard Jarvis and Michael McIntyre. Split manufacturing method for advanced semiconductor circuits. U.S. Patent no 20040102019A1, May 2004.

[41] Yajun Yang et al. "How Secure Is Split Manufacturing in Preventing Hardware Trojan?" In: 25.2 (Mar. 2020). doi: 10.1145/3378163

[42] Jeyavijayan Rajendran et al. "Security Analysis of Integrated Circuit Camouflaging". In: Proceedings of the 2013 ACM SIGSAC CCS '13. Berlin, Germany: Association for Computing Machinery, 2013, pp. 709–720. doi: 10.1145/2508859.2516656.

[43] Satwik Patnaik et al. "Obfuscating the Interconnects: Low-Cost and Resilient FullChip Layout Camouflaging". In: Proceedings of the 36th ICCAD '17. Irvine, California: IEEE Press, 2017, pp. 41– 48.

[44] Dominik Sisejkovic. "Designing trustworthy hardware with logic locking". Veröffentlicht auf dem Publikationsserver der RWTH Aachen University; Dissertation, Rheinisch-Westfälische Technische Hochschule Aachen, 2022. Online–Ressource : Illustrationen. doi: 10.18154/RWTH-2022-02625.

[45] M. R. Fadiheh *et al.*, "An Exhaustive Approach to Detecting Transient Execution Side Channels in RTL Designs of Processors," in *IEEE Transactions on Computers*, vol. 72, no. 1, pp. 222-235, 1 Jan. 2023, doi: 10.1109/TC.2022.3152666.

[46] IEEE Spectrum, "Stopping Hardware Trojans in Their Attacks", https://spectrum.ieee.org/stopping-hardware-trojans-in-their-tracks Accessed: Nov, 2023.

Lennart M. Reimann is the chief engineer at the Institute for Communication Technologies and Embedded Systems at RWTH Aachen University.

Dominik Sisejkovic is a research activity manager for cybersecurity automation at Corporate Research, Robert Bosch GmbH, Hildesheim, Germany.

Rainer Leupers is a professor at the Institute for Communication Technologies and Embedded Systems at RWTH Aachen University.