IACR Transactions on Cryptographic Hardware and Embedded Systems
ISSN 2569-2925, Vol. 2025, No. 1, pp. 37–77. DOI:10.46586/tches.v2025.i1.37-77

Trojan Insertion versus Layout Defenses for Modern ICs: Red-versus-Blue Teaming in a Competitive Community Effort

Johann Knechtel[1], Mohammad Eslami[2], Peng Zou[3], Min Wei[3], Xingyu Tong[3], Binggang Qiu[3], Zhijie Cai[3], Guohao Chen[3], Benchao Zhu[3], Jiawei Li[3], Jun Yu[3], Jianli Chen[3], Chun-Wei Chiu[4], Min-Feng Hsieh[4], Chia-Hsiu Ou[4], Ting-Chi Wang[4], Bangqi Fu[5], Qijing Wang[5], Yang Sun[5], Qin Luo[5], Anthony W. H. Lau[5], Fangzhou Wang[5], Evangeline F. Y. Young[5], Shunyang Bi[6], Guangxin Guo[6], Haonan Wu[6], Zhengguang Tang[6], Hailong You[6], Cong Li[6], Ramesh Karri[7], Ozgur Sinanoglu[1] and Samuel Pagliarini[2,8]

[1] New York University Abu Dhabi, Abu Dhabi, UAE, johann@nyu.edu
[2] Tallinn University of Technology, Tallinn, Estonia, mohammad.eslami@taltech.ee
[3] Fudan University, Shanghai, China, chenjianli@fudan.edu.cn
[4] National Tsing Hua University, Hsinchu, Taiwan, tcwang@cs.nthu.edu.tw
[5] Chinese University of Hong Kong, Hong Kong, China, fyyoung@cse.cuhk.edu.hk
[6] Xidian University, Xi'an, China, hlyou@mail.xidian.edu.cn
[7] New York University, New York City, USA, rkarri@nyu.edu
[8] Carnegie Mellon University, Pittsburgh, USA, pagliarini@cmu.edu

Abstract. Hardware Trojans (HTs) are a longstanding threat to secure computation. Among different threat models, it is the fabrication-time insertion of additional malicious logic directly into the layout of integrated circuits (ICs) that constitutes the most versatile, yet challenging scenario, for both attackers and defenders.

Here, we present a large-scale, first-of-its-kind community effort through red-versus-blue teaming that thoroughly explores this threat. Four independently competing blue teams of 23 IC designers in total had to analyze and fix vulnerabilities of representative IC layouts at the pre-silicon stage, whereas a red team of 3 experts in hardware security and IC design continuously pushed the boundaries of these defense efforts through different HTs and novel insertion techniques. Importantly, we find that, despite the blue teams' commendable design efforts, even highly-optimized layouts retained at least some exploitable vulnerabilities.

Our effort follows a real-world setting for a modern 7nm technology node and industry-grade tooling for IC design, all embedded into a fully-automated and extensible benchmarking framework. To ensure the relevance of this work, strict rules that adhere to real-world requirements for IC design and manufacturing were postulated by the organizers. For example, not a single violation for timing and design-rule checks were allowed for defense techniques. Besides, in an advancement over prior art, neither red nor blue teams were allowed to use any so-called fillers and spares for trivial attack or defense approaches.

Finally, we release all methods and artifacts: the representative IC layouts and HTs, the devised attack and defense techniques, the evaluation metrics and setup, the technology setup and commercial-grade reference flow for IC design, the encompassing benchmarking framework, and all best results. This full release enables the community to continue exploring this important challenge for hardware security, in particular to focus on the urgent need for further advancements in defense strategies.

Keywords: Hardware Security · Trojans · IC Design · Red-versus-Blue Teaming

Received: 2024-07-15 Accepted: 2024-09-15 Published: 2024-12-09

1 Introduction

Remarkable technology advances, driven by Moore's law, have been fueling the electronics industry's success for many decades. However, with each new technology node, the process of fabricating integrated circuits (ICs) becomes ever-more complex and expensive. This is especially true in today's era of fin-shaped field-effect transistor (FinFET) ICs that only a few, multi-billion-USD foundries are able to produce [MCM+19]. Said foundries compete for the title of "best transistor" while leaving the need for design-related innovation to design houses to pursue. This is referred to as the *fabless* business model.

Several security concerns arise for the fabless model with its outsourced, potentially untrusted fabrication. For example, intellectual property piracy [LJM12] and IC overbuilding [RKM08] can undermine the business of design houses. Importantly, there is another threat that brings severe consequences beyond unrealized business profits: *hardware Trojans (HTs)* [BHBN14, RKK14, CNB09, XFJ+16, DXL+20, KRRT10, TK10]. Broadly, HTs refer to any malicious modifications of ICs. HTs can diminish the reliability of ICs [BRPB13], corrupt data or computation [YHD+16, ABK+07], leak privileged information [LKG+09, PIVP21], or even cause ICs to stop working [KRRT10, XFJ+16]. Consequently, HTs can break the fundamental assumption of hardware serving as a steadfast root-of-trust for sensitive data processing.

Scope. This work is a serious attempt to expand the understanding of HT defense versus attack efforts in a realistic setting for modern ICs. We focus on *additive HTs* where the adversary inserts at fabrication-time—or rather right before—some additional logic that was not part of the original design. Among different scenarios [KRRT10, TK10, CNB09, XFJ+16], the concept of additive HTs represents the most generic and versatile threat: such HTs are not limited to, e.g., denial-of-service attacks through removal or modification of existing logic, but can embed any malicious functionality at will. However, it remains difficult to prove that some particular additive HT can be inserted by rogue fabrication engineers without disturbing the tightly-scheduled and rigorously-checked IC manufacturing process. Similarly, it remains difficult to prove that some pre-silicon defense techniques are robust against a wide range of HT designs.

We tackle these challenges, and further research questions as outlined below, through an extensive red-versus-blue teaming effort. Notably, 23 designers participated across all blue teams, investing an estimated 3,240 man-hours; 3 designers participated as the red team, spending an estimated 1,080 man-hours. The effort spanned over several months.

Research Questions. The following research questions guided this community effort.

- From the defenders' perspective:
  - RQ-D1) Are IC layouts per se vulnerable to HT insertion?
  - RQ-D2) Can layout-level, pre-silicon defenses hinder the fabrication-time insertion of additive HTs? What specific techniques should be used?
  - RQ-D3) Are such techniques practical, i.e., can they be realized without undermining design quality?
  - RQ-D4) What practical challenges arise for a modern and real-world setting for IC design and manufacturing?
- From the attackers' perspective:
  - RQ-A1) What are effective ways for actual insertion of additive HTs into IC layouts? What specific techniques should be employed toward that end?
  - RQ-A2) How successful are such efforts in the presence of layout-level defenses?
  - RQ-A3) What practical challenges for HT insertion are to be handled for a modern technology node and real-world settings for IC design and manufacturing?

Contributions. The contributions of this work are as follows.

- A first-of-its-kind community effort on fabrication-time insertion of additive HTs versus layout-level, pre-silicon defenses. Our effort follows a real-world setting based on a 7nm technology node and commercial tooling for IC design. Our effort employed a strict *red-versus-blue teaming*: there was no collaborative interaction between any of the teams and the competing blue teams remained anonymous to each other.[1]
- Based on the real-world IC setup, an end-to-end framework for pre-silicon benchmarking of attack versus defense trials, with a fully automated and extensible back-end.
- Novel and important insights for both attackers and defenders, which are highlighting the real-world challenges of defending modern ICs against HT insertion and also serve to counter some long-standing yet overly simplistic beliefs about defenses. Toward that end, our effort follows a stringent threat model and practical constraints where, e.g., defense efforts must maintain functional and manufacturable layouts. In an advancement over prior art, neither attack nor defense efforts are allowed to trivially leverage so-called fillers and spares.
- Full release of all artifacts and methods. This includes the reference flow for commercial-grade IC design in a modern 7nm node, the end-to-end benchmarking framework, the representative IC layouts and HTs, multiple sets of attack and defense techniques, and representative best results.[2] This release significantly advances the state-of-the-art for related research and development (R&D) efforts.

Organization. Section 2 provides the background and general motivation for this work. Section 3 discusses related works and their limitations, providing further motivation. Section 4 describes the threat model. Section 5 presents the end-to-end framework for benchmarking of defense versus attack trials. Sections 6 and 7 describe the benchmarks and HTs devised for this work, respectively. Section 8 details the contest results from both red and blue teams' perspectives. It also discusses challenges and limitations faced by both sides. Section 9 follows-up from the red team's perspective, strengthening the main argument of this work that the considered threat is practical. Section 10 concludes this work. Appendix A discusses further considerations of this work. Appendix B describes the contest format. Appendix C introduces the release. Appendix D provides supplementary results. Appendices E and F describe further details for the framework, the benchmarks, and the HTs, respectively. Appendix 10 provides acknowledgments.

2 Background

Design-Time Efforts for Hardware Security. Conceptually, the "hardening" of physical layouts during IC design can protect against various threats that are executed post design-time, not only against HTs [HCS+20, KKR+20, KGB+21, LRT+21, BMN+24, KSS+18]. Such proactive efforts are important for two reasons. First, the success of such attacks is directly related to an IC's layout and its characteristics. Second, threats that are not mitigated during design-time are virtually impossible to fix later on—ICs are unlike patchable software. Note that commercial design tools and industrial workflows currently lack such capabilities for "layout hardening," leaving contemporary ICs largely vulnerable.

[1] The setting of anonymity among blue teams reflects a real-world scenario where design houses offering security-aware products and services would independently compete for their market share. From a research perspective, we argue that this setting also helps to avoid bias toward overly similar defense strategies.

[2] All techniques, the reference flow, and the benchmarking framework are released at [https://github.com/DfX-NYUAD/Trojan-Insertion-versus-Layout-Defenses] whereas the IC layouts, the HTs, best results, and the technology setup are released at [https://drive.google.com/drive/folders/10GJ5hXOBQupwqv1WMtitarsEuEE_Y-vV?usp=sharing].

Table 1: Comparison to Selected Related Works

	[MGK+13]	[PMB+23]	[ST16]	[WZL24]	[PP22]	[GMMP20]	[KGB+21]	[TSBH20]	[WWF+23]	[TSBH23]	[GYT+23]	[EPP23]	**[This]**
Competition	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
Red-versus-Blue	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
Automated HT Insertion	✓✗	✗	✗	✓✗	✓	✓✗	✗	✗	✗	✓✗	✗	✗	✓
Automated HT Detection	✓	✓✗	✓✗	✓✗	○	✓✗	○	✓✗	○	✓✗	○	○	○
Actual HTs	✓	✗	✓	✓	✓	✓	✗	✗	✗	✓	✗	✗	✓
Placement Def.	✗	✗	✗	✓	✗	✗	✓	✗	✓	✗	✓	✓	✓
Routing Def.	✗	✗	✗	✗	✗	✗	✓	✗	✗	✓	✓	✓	✓
Rule Out Spares, Fillers	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
Technology Nodes (nm)	180	90, 65, 40, 28	90	45	65	90, 65	45	45	45	45	45	65	**7**
IC Tape-Out	✓	✓	○	○	✓	✓	○	✗	○	✗	○	✓	○

Def. is short for defense. Symbols: ✓ means yes, ✗ means no, ✓✗ means to some degree, and ○ means not applicable / out of scope.

Hardware Trojans. HTs are malicious modifications of ICs that can be implemented in many different ways and for various attack scenarios [BHBN14, RKK14, CNB09, XFJ+16, DXL+20, KRRT10, TK10, Kne21]. Most HTs have two distinct parts: trigger and payload. The trigger is an activation mechanism, typically based on rare and specific combinational and/or sequential conditions. Once these conditions are met, the payload performs the HT's actual malicious operation.

Real-World Relevance of HTs. Ideally, hardware should serve as a root-of-trust on which sensitive software solutions can be built on. If that premise is broken, any application concerned with security and privacy is completely at risk. Thus, HTs may attempt to compromise crypto cores [PP22, GMMP20], which are found in mainstream CPUs for more than a decade by now, highlighting the significance of HTs for consumer electronics. In other domains such as military and avionics, naturally, the stakes are even higher; potential HT vulnerabilities on military-grade ICs are discussed in [Ade08, SW12].

3 Related Works and Limitations

A high-level comparison of selected works and ours is provided in Tab. 1. Importantly, ours distinguishes itself by thorough coverage of practical, real-world aspects for both attack and defense efforts in modern ICs, and by a first-of-its-kind red-versus-blue teaming community effort. Details and further works are discussed next. Note that all related works and their limitations are mainly discussed in the context of fabrication-time insertion of additive HTs. For a broader review, readers may also refer to [BHBN14, RKK14, CNB09, XFJ+16, DXL+20, KRRT10, TK10].

Attackers' Strategies. Several studies have demonstrated the threat of HTs in real silicon [MGK+13, YHD+16, GMMP20, PP22, PMB+23]. An example is shown in Fig. 1. Notably, the actual HT insertion there was done in an automated fashion, by having the adversaries rely on a technique known as *engineering change order (ECO)*. The use of ECO is an industry-wide standard for incremental edits of IC layouts, offering relatively short turn-around cycles. It was only recently explored for HT insertion [PIVP21, PP22, HPPS22], whereas prior art assumed—often only implicitly—manual efforts for insertion of HTs, which can become error-prone and/or time-consuming.

Defenders' Strategies. These can be divided into pre- versus post-manufacturing detection as well as design-time assessment and prevention. Pre-manufacturing detection is often based on the defender simulating the design to find anomalous behavior [ZT11, WSS13]. Notably, this is incompatible with fabrication-time insertion of HTs—such HTs cannot be covered by simulation because they do not exist yet at design-time. Post-manufacturing detection approaches like [GBF17, LKG+09, DXL+20] also face a number of significant hurdles as follows. First, HTs are difficult to identify using traditional post-manufacturing test [XT13, CWP+09, WSS13, SCN+15, GGPR22]. This is because such defect-oriented tests are not helpful to determine some additional, malicious but rare functionality brought on by HTs. Second, HTs can differ greatly in terms of structural and functional properties, making their identification difficult [GGPR22]. Machine learning-based schemes aim to tackle this challenge, but remain limited in accuracy [HYT17, LAKS23]. Third, especially for advanced technology nodes, process variations and measurement noise makes the observation of physical HT effects more challenging [JMHS14, DNCB10]. Finally, some detection approaches assume the existence of a "golden IC", i.e., an HT-clean reference chip [ABK+07, DNCB10, MMST23]. Given the lack of trust in the foundry in the first place, this assumption cannot be reliably made.

Frameworks for design-time assessment of IC layouts against additive HTs were proposed in [TSBH20, ST16], albeit without consideration of specific attack/defense techniques. Another framework evaluates prior art for defenses against HT insertion [WZL24].

Design-time prevention strategies include design obfuscation [WWF+23, WWA+24], placement filling (mainly using spares and/or fillers) [KGB+21, WWF+23, BDP+16], routing filling [KGB+21, TSBH23], split manufacturing [PASK19, SNA+22], and insertion of self-testing circuitry [XT13]. Approaches that promote security-aware physical-design flows are discussed in [KGB+21, TSBH23, HCS+20, GYT+23, HCC+23, WZL23, WWF+23, WWA+24, EPP23], with a corresponding IC tape-out also described in [EPP23].

Limitations. In addition to the limitations outlined above, there are further, even more significant limitations as follows. First, all prior art assumes the presence of *spares* and/or *fillers*.[3] While spares and fillers are easy to utilize for defense efforts, i.e., by filling up all open *placement sites* with them,[4] these components are straightforward to exploit for HT insertion—they can be removed without disrupting any core functionality [PMB+23]. Second, none of the prior art for attacks considers layout-level defense efforts. Third, aside from [ST16, TSBH23, WZL24], no prior art for defenses considers actual HT insertion for a thorough assessment of their techniques. Finally, prior art also lacks practical relevance, e.g., in [TSBH20], the role of timing is only approximated via routing distances and, in [ST16, TSBH20, WZL24], any layout changes other than trivial exploitation of open placement sites and/or fillers are erroneously ruled out.

These limitations highlight the disconnect between prior art and the real-world threat of HT insertion. In short, prior art (i) significantly underestimates the attackers' capabilities and (ii) largely ignores challenges imposed on attackers by layout-level defenses. In this work, we will fully address all these limitations.

Competitions. More loosely related, i.e., for design-time HT insertion versus detection, prior red-versus-blue competitions are described in [RJK11, WRSS14] and some follow-up works in [BLL+11, SF12]. Another competition [DGH+19] focused on vulnerabilities in the source codes of IC designs, albeit without explicit consideration of HTs. As indicated, ours is the only effort that covers fabrication-time HT insertion versus design-time defenses.

[3]Spares can be thought of as redundant functional cells that are not in use initially, i.e., they are placed but not fully routed yet. Spares support last-minute changes in design functionality via routing-only ECO procedures. Fillers are non-functional cells that serve different physical purposes, e.g., maintaining the so-called N-well continuity. There are various types of physical fillers like so-called tap cells, decaps, and actual fillers. Furthermore, there are metal fillers which are beneficial for manufacturability.

[4]The physical layouts of modern ICs are arranged into so-called placement sites and placement rows as well as so-called routing tracks. For more details, see [KLMH11].

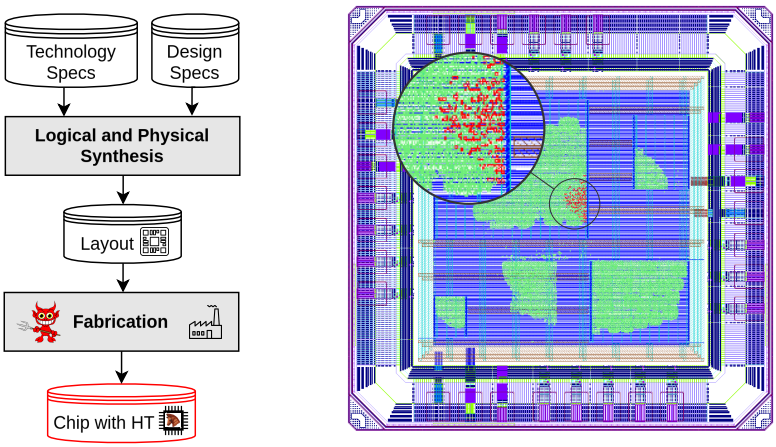


Figure 1: The threat of fabrication-time HT insertion. (Left) Simplified IC design flow and supply chain with the adversary and the attack highlighted. (Right) An exemplary HT attack on an AES core in an actual IC [PP22]. HT logic is shown in red, selected regular cells in green, routing in blue/brown, and peripherals in purple.

4 Threat Model

General Scope. We consider insertion of additive HTs into IC layouts conducted by foundry-based adversaries (Fig. 1). Our threat model focuses on a realistic and real-world scenario. It is largely in agreement with that of [MGK+13, YHD+16, GMMP20, PP22, PMB+23], i.e., the few related works that show actual fabricated ICs with HTs. Importantly, all assumptions and capabilities outlined below were communicated to the teams prior to the start of the community effort.

Capabilities for Red-Team Attackers. For any IC under attack, attackers have full access to: (1) the "IC blueprints," i.e., the files describing the layout-level design implementation, or *layout files* for short; (2) information on which design parts are security-sensitive and, thus, potential targets for HT insertion; (3) all technology details, including design and manufacturing rules as well as a set of standardized design components, i.e., the *standard-cell library*; and (4) the timing limits for correct operation, i.e., the *timing constraints*. Furthermore, we assume adversaries are knowledgeable in commercial IC design and manufacturing, and have access to contemporary design tools.

All assumptions are based on a pragmatic, real-world setting and reflect the key focus on layout-level attacks versus defenses. Assumptions (1) and (3) are due to the fact that adversaries reside with the foundry. Assumptions (2) and (4) would require that adversaries either obtain such knowledge directly from design specifications, or derive it from the layout files, e.g., following [HPPS22]. To avoid such reverse-engineering efforts, thereby ensuring a fair and focused contest, attackers and defenders were provided with timing constraints and so-called *cell assets*, i.e., registers that hold security-sensitive data (Sec. 6).

Capabilities for Blue-Team Defenders. We consider security-aware IC designers that are focused on proactive, pre-silicon defense efforts at the layout level. Defenders have the same capabilities as adversaries, with the exemption that defenders have direct access to all design details and, thus, do not need to recreate or derive any such information.

Note that defenders seeking to inspect their ICs against HT insertion have to pursue additional post-manufacturing efforts—in the fabless business model, there is no guarantee as to what the outsourced foundry has truly manufactured. For this focused community effort, we readily enable layout-level inspection capabilities, by returning HT-infested layouts back to the blue teams.

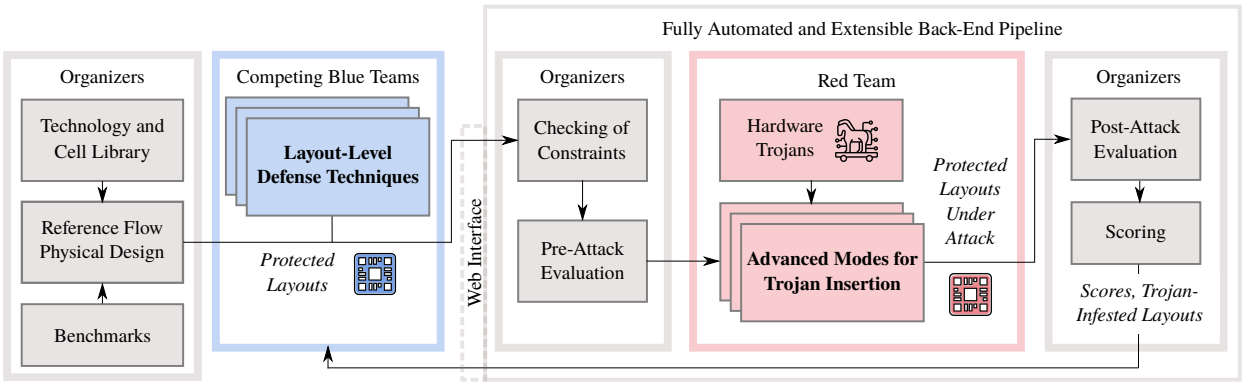


Figure 2: Overview of the end-to-end framework for benchmarking of layout-level defenses versus advanced HT insertion.

Advances. As indicated, prior art utilizes spares and/or fillers for trivial defense/attack efforts. In our threat model, we prohibit all types of spares and fillers, pushing both attackers and defenders to develop new, sophisticated strategies. This fundamentally alters the threat landscape, as defenders must devise more sophisticated measures to protect regular layout resources, while attackers must carefully seek out those regular resources that remain exploitable for HT insertion. Importantly, this constraint enhances the generalizability of our findings by preventing trivial reliance on spares and fillers. In other words, instead of limiting the efforts, this constraint rather advances the state-of-the-art for both attacks and defenses.

Limitations. First, while other types exist, e.g., zero-gate/non-additive HTs [BRPB13, SSF+14] or analog HTs [YHD+16], we focus on additive and digital HTs. This is particularly relevant for this community effort on layout-level defenses under realistic threats: additive HTs incur the most significant challenges on attackers for actual insertion into IC layouts, but also offer the most versatility in terms of malicious functionality. Second, post-manufacturing inspection can detect additive HTs [PMB+23, MMST23], with the help of sophisticated failure analysis equipment such as e-beam probing even for advanced technology nodes [LBL+22]. Third, as the HTs devised for this effort are specifically targeting on cell assets (Sec. 7), post-manufacturing testing might expose them as well.

In any case, it is essential to understand that the key objective of this work is to study the threat of fabrication-time HT insertion and the challenges it poses for layout-level defenses, regardless of whether HTs are designed with avoiding detection in mind or not. Also recall that we provide blue teams with straightforward inspection capabilities. We deliberately create such an optimistic scenario to ensure that any conclusion holds true even with these additional powers granted to the defenders.

5 Framework for Red-versus-Blue Benchmarking

Here, we present our end-to-end framework which is carefully developed for rigorous red-versus-blue benchmarking. More technical details are also provided in App. E, including a description of the fully automated and extensible back-end pipeline. Also note that the benchmark layouts and HTs are described separately in Sec. 6 and Sec. 7, respectively.

This framework, illustrated in Fig. 2, establishes the technological foundation and rules for our competitive community effort focused on layout-level defenses against HT insertion. The framework encompasses a real-world setup for a modern 7nm technology node and a reference physical-design flow replicating industry practices. It also includes diverse

defense and attack techniques as devised during this effort, along with a comprehensive evaluation system to assess both security risks and impacts on design quality.

5.1 Technology and Standard-Cell Library

For this community effort, the chosen process design kit (PDK) and standard-cell library are from *ASAP7* [CVS+16, VVC17]. It is important to note that the organizers selected *ASAP7* as an academic, freely accessible PDK on purpose: only this way it was possible to open up the contest and this research effort to the community at large.

The *ASAP7* PDK [CVS+16] describes an advanced 7nm FinFET technology originally developed by Arizona State University and ARM. The same team also provides a standard-cell library [VVC17] for their PDK, which utilizes FinFET transistors and is fully characterized. The files provided do resemble a commercial PDK quite well, including cells with different threshold voltages, parasitic extraction decks, *design rule checks (DRCs)* which are essential for ensuring manufacturability of the advanced layout geometries,[5] etc. *ASAP7* is the most complete and best established PDK by and for academia—numerous R&D studies are based on this PDK [CVH+17, CHL+21, LPH+21, CKSL17, KCU24].

For this community effort, some modifications were made by the organizers, mainly to ensure that participants could use different tools and versions with ease; technical details are given in App. E.2.

5.2 Reference Flow for Physical Design

Following best practices from industry, the organizers have implemented and thoroughly tested a reference flow for physical design (PD) of modern ICs based on the *ASAP7* PDK. Note that the organizers have a collective experience of designing and taping-out several ICs themselves [PP22, EPP23, AIRP22, YSN+17].

The PD flow was used for generating all the benchmark layouts (Sec. 6). It was also distributed early on to all blue teams, to ease their ramp-up for this advanced 7nm node and to enable them to develop their defense techniques into a robust baseline flow.

On a high-level, the flow works as follows: the behavioural register-transfer level (RTL) descriptions of the benchmarks are passed to the logic-synthesis tool (*Cadence Genus* in this work), followed by the physical-synthesis tool (*Cadence Innovus* in this work) which generates the layout files. Next, we outline the steps for the latter. See also, e.g., [KLMH11] for more background on modern PD in general and [KKR+20, HCS+20, KGB+21] for security-aware PD in particular.

1) Initialize. Variables are set, e.g., the paths for the benchmark's initial *netlist*,[6] the standard-cell library, and the timing constraints.

2) Floorplanning. The floorplan size is to be defined according to which design is being implemented. The blue teams are free to redefine the floorplan sizes. For a fair competition, however, the given power-planning strategy (see below) must be followed. Furthermore, input/output pins were placed and constrained to the left/right side of the IC, respectively. The spacing between pins as well as the metal layer of the pins is pre-defined and fixed.

3) Power Planning. Parameters for power planning are derived following the *ASAP7* PDK documentation and best practices. Once all parameters are set, the power distribution network (PDN) is routed and power vias are generated. See App. E.3 for more details.

4) Place and Route (PnR). First, all the needed instances of standard cells are placed. Second, clock-tree synthesis (CTS) is performed, i.e., all instances of sequential

[5] DRCs cover rules for arrangement of cells, considering well continuity, etc., and metal segments, considering minimum area, shorts, opens, etc. For more details, see [AGK+15, BPK+22].

[6] A netlist contains all the needed instances of standard cells and all the nets, i.e., the interconnectivity between these cell instances and toward primary inputs/outputs.

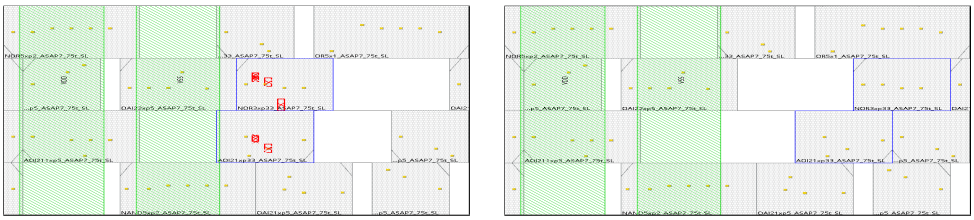


Figure 3: A simple example for DRC violations. Standard cells are shown in grey, and their pins as small, yellow squares. Power stripes are shown in green. (Left) DRC violations, marked by red polygons, arise due to congested pin access. For visual clarity, the actual routing congestion is not shown here. (Right) Fixing of those violations, by moving the affected cells into open placement sites to the right.

Table 2: Classes of Devised Defense Techniques

	Team			
	A	**B**	**C**	**D**
Shrink IC Outlines	✓	✓	✓	✓
Automated Parameter Tuning & Design-Space Exploration	✗	✓	✓	✗
Insertion of Functional Components	✓	✗	✗	✗
Insertion of Buffer Components	✓	✓	✗	✓
Insertion of Routing Detours	✓	✗	✗	✗
Re-Arrangement of Components	✓	✓	✓	✓

standard cells are interconnected within an optimized and dedicated tree network for clock delivery. Third, routing of the regular interconnects is conducted.

5) Finalize. The layout is exported as a design exchange format (DEF) file, along with its final post-route netlist. Post-route design and security metrics are gathered (Sec. 5.5).

Practical Challenges. It is important to note that routing is well-known to be one of the most complex steps in PD, especially for advanced nodes [AGK+15, BPK+22, LFL+24]. For example, routing access to cell pins around so-called power stripes, i.e., wide routing shapes of the PDN that run orthogonal to the placement rows, can become quickly congested and, thus, prone to DRC violations (Fig. 3).

It is expected that both defenders and attackers will face DRC violations, the latter even more so as they must work on the hardened layouts by defenders. Most DRC violations, even if just occurring once, could disqualify the whole layout from manufacturing; this is a hard and well-known constraint in the real-world of the IC industry. Thus, both defenders and attackers must carefully avoid DRC violations for their layout-level techniques. Importantly, the reference flow does not provide specific techniques for mitigating DRC violations. This is on purpose—blue (and red) teams are expected to devise their own mitigation techniques along with their defense (and attack) efforts.

5.3 Defense Techniques

Table 2 provides an overview of the blue teams' efforts, divided into classes of layout-level defense techniques. Note that the teams were free to implement prior art, devise similar strategies, and/or contribute novel methods. This led to this diverse range of efforts, also yielding different innovative contributions. Also note that, while some classes are covered by multiple teams, the related decisions were made independently; recall that blue teams were anonymous to each other. Thus, for the same class of techniques implemented by different teams, the efficiency and the results would still differ.

Next, the actual techniques are outlined. The scripts are also included in the release (App. C). For any step stated to be done *as much as possible*, the teams pushed those steps to their best ability while respecting timing constraints and zero DRC violations.

Shrink IC Outlines. All teams started by revising the floorplan, as in shrinking the IC outline, i.e., the actual silicon's dimensions, as much as possible. Team B incorporated this step into their exploration flow, whereas all other teams performed this step manually.

Automated Parameter Tuning & Design-Space Exploration. Team B devised a Bayesian optimization model for an automated design-space exploration flow. Hyper-parameters were extracted from the reference flow, including tool efforts for routing congestion and power consumption, and scaling of the clock periods as well as of the IC outline (via placement rows), all as much as possible. Team C utilized simple sweeping for tuning of timing parameters, i.e., target *slacks* for *setup* and *hold* timing.[7]

Insertion of Functional Components. Team A inserted additional functional components following the notion of logic locking [YSN+17], to fill up layout regions as much as possible. Toward this end, they first select paths where logic-locking structures should be inserted, based on timing slacks and coverage of cell assets. For actual physical integration, they devise an ECO-based PnR strategy.

Insertion of Buffer Components. Teams A, B, and D inserted buffers and repeaters to fill up open placement sites as much as possible. Team A integrated buffers only into specific paths related to their logic-locking structures. Team B inserted buffers specifically into open sites of so-called exploitable regions (Sec. 5.5). Team D inserted additional buffers during regular timing closure.

Insertion of Routing Detours. Team A extends the routing of selected nets into the upper metal layers which are typically less utilized. This is done to reduce the number of free routing tracks (Sec. 5.5). The selection of nets is based on timing slacks and existing routing patterns/shapes, to ease the insertion of the detours toward the upper layers.

Re-Arrangement of Components. For final tuning, all teams shifted components such that exploitable regions (Sec. 5.5) are reduced as much as possible. Teams A, B, and D perform customized cell-shifting techniques while considering other nearby cells, upper displacement thresholds, etc. Team C utilizes linear programming for each placement row, configured to (i) minimize component displacement and (ii) evenly distribute open sites in placement rows, thereby indirectly breaking up exploitable regions.

5.4 Attack Techniques

For each valid submission by a blue team, i.e., any submitted layout that passes all the constraint checks (Sec. 5.5), the red team conducts layout-level HT insertion. The red team devised an ECO-based procedure for implementing the attacks. These efforts resulted in novel techniques for advanced HT insertion.

The procedure is outlined next; see also Fig. 9 (Sec. 9) and App. C.

1) Initialize. First, the layout files are loaded into *Cadence Innovus*, along with all technology files as pre-defined for the corresponding benchmarks. Next, timing and power analysis is conducted.[8] Finally, all data and settings are stored in a design database.

[7] Slacks are margins by which timing constraints are (over-)fulfilled. Typically, some slacks are desired, e.g., to maintain correct operation of ICs even under the impact of aging [vSAMM+16], etc. Setup timing refers to the minimal time that some data should be stable before the clock's active edge. This is to ensure that the correct data can be properly latched/stored. Hold timing, in contrast, refers to the minimal time that some data should be stable after the clock's active edge. This is to ensure that some data launched at the current clock cycle does not get falsely captured at the same cycle already, which would lead to conflicts in the sequential behaviour.

[8] More specifically, timing and power analysis is configured for both setup and hold views at first. Second, toggling activities are set, without loss of generality, to a default value of 0.2 for all primary inputs. Third, clock propagation is set up with consideration of on-chip variations. Fourth, timing and power of the design is analyzed.

2) Reclaiming Area for HT Insertion. During this pre-attack stage, the red team manipulates the blue teams' protected layouts for easier handling during actual HT insertion later on. The idea is to revert any unnecessary buffer components, cloned logic structures, and/or oversized gates, i.e., cell instances with driver capacities larger than needed for reliable IC operation.

This stage was motivated by an anticipation that blue teams could employ such means for "bloating up" layouts, i.e., to make HT insertion more difficult without the need for more complex functional modifications of the design. Recall that most blue teams indeed inserted buffers and other components (Sec. 5.3). It was further motivated by the fact that such overheads exist in any layout, at least to some degree. Thus, attackers seeking to insert additive HTs would aim at such literal "room for improvement" in any case.

3) Actual HT Insertion. This stage covers the actual attack. The steps are to (i) integrate the HT at functional level, i.e., into the protected layout's netlist, and (ii) conduct timing-driven ECO PnR—an industry-wide standard for layout-level edits in general—for physical integration of the HT into the layout.

For functional integration, note the following. Depending on the blue teams' efforts, a submission's netlist can differ substantially from the corresponding benchmark's baseline netlist. Still, functional integration is guaranteed to succeed. This is because all HTs (Sec. 7) are designed such that they connect to cell assets (Sec. 6), which must be maintained by the blue teams (Sec. 5.5).

For physical integration, three different modes are devised by the red team: (a) *conservative*, (b) *moderate*, and (c) *aggressive*. The pre-attack stage and this step are orchestrated as follows. For the conservative mode, the pre-attack stage is skipped, and ECO PnR is configured to only work on the newly added HT components. For the moderate mode, the pre-attack stage is conducted once, and ECO PnR is configured as with the conservative mode. For the aggressive mode, the pre-attack stage is conducted twice, iteratively trying to reclaim more area, and ECO PnR is configured to work on the newly added HT components with higher priority but also on all other components as needed.

4) Finalize. Timing is re-analyzed and other checks are conducted, in particular for DRC violations. Depending on the attack setting, further efforts to resolve violations may be taken; see also Sec. 9 and App. D. Finally, the HT-infested layout is exported.

5.5 Evaluation

Constraints Checking. A number of constraints have to be respected by the blue teams' defense techniques to allow for a realistic and competitive, yet fair, community effort. Most importantly,[9] layouts must: (1) maintain all cell assets, (2) have zero DRC violations, (3) meet timing checks, (4) maintain functional equivalence to the benchmarks, and (5) cannot use spares or fillers. Note that (5) dictates that attackers cannot exploit spares or fillers either. Furthermore, layouts must pass additional design checks, like routing checks for dangling wires, etc. There are 12 such checks, carefully devised by the organizers, in anticipation of more or less trivial defense efforts by blue teams. For example, checks for dangling wires are implemented to catch the (mis-)use of regular functional cells as additional spares. While such a defense technique would be more thoughtful than trivially using fillers, it can still be easily undermined by real-world attackers.

Pre-Attack Evaluation: Design Overheads. Any layout-level defense is expected to incur some impact on design quality. To discourage efforts that come at excessive overheads and would be impractical, design quality is evaluated following industry best practices. More specifically, total power, worst negative slack (WNS) for both setup and hold timing,[10] and the IC outline are assessed.

[9] See Apps. E.4 and C for the full list of constraints and for more technical details.

[10] WNS quantifies the worst timing path, i.e., (i) the path with the least slack remaining, expressed as the smallest positive WNS value, or (ii), in case of timing being violated, the path with the largest

Pre-Attack Evaluation: Security Risks. Here, layouts are evaluated against the *generic* prospects for HT insertion, without consideration of specific HTs. Accordingly, this evaluation stage is focused on layout resources that would be relevant for any additive HT.

Inspired by Trippel et al. [TSBH20], the notion of *exploitable regions* is proposed. Exploitable regions serve to estimate an attacker's chances of inserting HTs with relative ease. More specifically, an exploitable region is defined as 20+ spatially continuous and free/open placement sites, within and/or across placement rows. Note that this particular size relates to an advanced implementation of the *A2* HT [TSBH20, YHD+16]; however, it is chosen without loss of generality here. Also note that illustrations of all exploitable regions for selected benchmarks are provided in Fig. 4.

Furthermore, *free routing tracks* are evaluated. Attackers have to exploit free tracks to connect the trigger and payload components of their HTs with each other and with the layout under attack. Here, free tracks across all metal layers and the whole layout are considered. This is done as, in the absence of actual HTs for this generic evaluation, not only tracks within exploitable regions but all tracks are of potential interest. This is because for actual HT insertion later on, the placement of trigger and payload components may have to be spread out across various exploitable regions, thereby also requiring free routing tracks across the layout.

Post-Attack Evaluation. Here, layouts are evaluated after the actual insertion of *specific* HTs (Sec. 5.4). This evaluation stage works as follows. First, for each separately inserted HT, the corresponding post-ECO reports are parsed. Second, a score-sheet is queried. Third, the scores are averaged across all relevant HT runs.

The score-sheet is defined as follows. In general, lower scores mean that the red team faces more challenges as in more severe violations for HT insertion, which means that the blue team's defense is more effective, and vice versa. The specific ordering and composition of violations is derived from best practices from industry. Also refer to App. E.4 for an example and for more details.

1. Design failures, like placement errors due to excessively high layout utilization,[11] for the aggressive, moderate, or conservative insertion mode, respectively: 0, 1, or 2 points. Note that points are assigned similarly across the three different insertion modes for the remainder of the score-sheet.
2. DRC violations: 5, 6, or 7 points.
3. Extensive timing violations, as in setup `AND` hold violations: 10, 11, or 12 points.
4. Less extensive timing violations, as in setup `XOR` hold violations: 15, 16, or 17 points.
5. Performance issues, as in violations for the clock tree `OR` so-called design rule violations (DRVs):[12] 20, 21, or 22 points.
6. No violations: 25, 26, or 27 points.

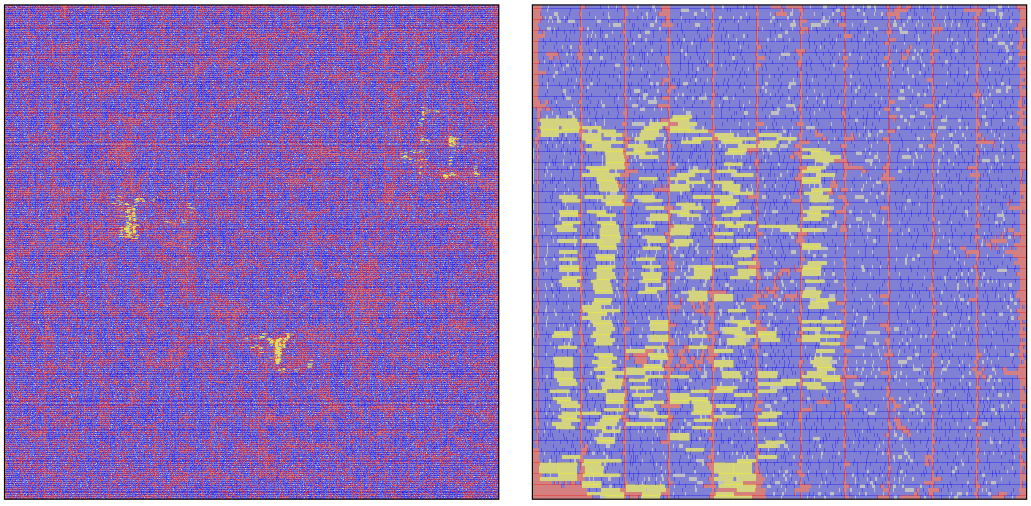
Scoring. Both pre-attack evaluation steps (i.e., design quality and generic prospects for HT insertion) are normalized to their nominal values for benchmark layouts as is, whereas the post-attack evaluation is normalized to the defenders' worst-case scenario.[13] In general, this normalization is more sensitive to degradations than to improvements. This is

violation, expressed as the largest negative WNS value. Since timing checks are a hard constraint for blue teams, only Case (i) applies here.

[11] Layout/placement utilization is the ratio of occupied over total placement sites. Recall that we rule out any kind of spares and fillers; thus, all utilization numbers quantify the regular use of standard cells.

[12] DRVs are not to be confused with DRCs. DRVs relate to detailed timing checks, whereas DRCs relate to technology rules for manufacturability.

[13] That scenario is "no violations occurred for HT insertion for the conservative mode." Such normalization is needed to put all attack efforts in their proper context, i.e., across benchmarks in general and when subject to different defenses in particular. This is important as already the baseline outcomes (i.e., for HT insertion into benchmark layouts as is) differ across benchmarks (Sec. 8). In other words, if we would *not* normalize to such a generally applicable scenario but rather to the different baseline outcomes, we would *not* be able to accurately quantify attack-versus-defense efforts across all benchmarks with their significant variety for their physical layouts (Sec. 6).



(a) AES128 **(b)** CAMELLIA

Figure 4: Layout illustrations for selected benchmarks. Shown are regular cell instances in blue, cell assets in dark-yellow, exploitable regions in red, and remaining open placement sites in grey, respectively. Different dimensions result in varying levels of detail visible in these fixed-size plots. All benchmarks are also illustrated in Fig. 13 (App. F).

on purpose—an important objective for this effort is to push blue teams for sophisticated "layout hardening" but *not* at an excessive cost for design quality.

After normalization, final scores are computed as:

$$score = (1/2 \times security\_risks) + (1/2 \times design\_overheads) \quad (1)$$

Note that lower final scores translate to better rankings for blue teams. Further details, including detailed score equations, are provided in Apps. E.4 and C.

6 Benchmark Layouts

For benchmarks, the organizers selected six different crypto cores, i.e., hardware accelerators for the following standardized and widely used cryptographic algorithms: AES128, CAMELLIA, CAST, MISTY, SEED, and SHA256 [Fre12, Cry07, sec13]. Naturally, such crypto cores are attractive targets for HTs, as their compromise could leak sensitive data or disrupt secure operations [PP22, GMMP20].

For generating the actual benchmarks, i.e., the unprotected baseline layouts of the crypto cores, the organizers employed the reference flow for PD (Sec. 5.2). While doing so, the organizers refrained from overly thorough optimization. This was done on purpose for two reasons, namely to (i) provide blue teams with some budget for their defense techniques, but also (ii) leave some exploitable resources for the red team. Note that this setting mirrors real-world practices where, due to tight schedules and the considerable complexity of PD, most layouts would not be "perfectly optimized."

Figure 4 illustrates the unprotected baseline layouts of selected benchmarks, including exploitable regions and cell assets. Recall that *cell assets* are registers holding critical data, such as encryption keys. For simplicity, the organizers have extracted cell assets manually such that selected sets of sensitive data and related potential targets are covered.

7 Hardware Trojans

The red team provided a range of novel HT implementations for this community effort. The following representative and realistic attack scenarios are covered: (1) leak/steal sensitive information, (2) induce faults, e.g., to corrupt the integrity of secure data processing, and (3) burn/over-consume power, e.g., to disrupt the operation of battery-powered ICs.

HTs for Scenario (1) are implemented by tapping into and maliciously propagating the outputs of selected cell assets, HTs for (2) by additional logic that is maliciously modifying the inputs/outputs of selected cell assets, and HTs for (3) by ring oscillators, i.e., dedicated circuitry which periodically oscillates and constantly consumes power. The latter are controlled/activated by the outputs of selected cell assets. Through various HT versions across all six benchmarks, 36 HTs were devised in total.

Further important considerations are discussed next. First, the organizers refrained from utilizing any published HT dataset, but rather tasked the red team to devise novel HT implementations. This was done for two important reasons: (i) to circumvent the lack of suitable datasets to begin with, and (ii) to avoid incentives for the blue teams to craft their defenses exclusively against specific existing HTs. For (i), note that, while the prominent *TrustHub* suite [STK13] contains few additive HTs for fabrication-time insertion, the validity of this suite has been put into question recently [Kri23]. For (ii), note that the blue teams were neither made aware in advance of the HT types to expect, nor had the actual HTs been released as such during the contest. However, following our threat model (Sec. 4), HT-infested layouts were returned to blue teams for post-manufacturing inspection. Still, this was done only later into the contest, after some blue teams had qualified for the final ranking (App. B). Again, the idea was to not incentivize blue teams to craft their defenses only against these specific HTs. This objective is further strengthened by maintaining both the generic pre-attack evaluation and the specific post-attack evaluation for the final ranking. Second, the red team was not constrained while devising their HTs; they only had to adhere to the real-world setup for IC design provided through the benchmarking framework. Finally, HTs similar to those devised for AES128 have been demonstrated through a real IC tape-out [PP22].

8 Results

The key findings from the competitive contest of this months-long community effort are the following.

1. *Defenses Are Both Needed and Practical:* All blue teams managed to successfully, yet to various degrees, identify and fix potential attack points for the benchmark layouts, all without undermining design quality (Sec. 8.1). The devised defenses represent competitive, practical, and state-of-the-art efforts.

2. *Attacks Remain a Threat:* Despite the demonstrated success of the various defenses, the red team still succeeded with their novel and advanced techniques for HT insertion as well, at least partially (Sec. 8.2). HT insertion represents a valid threat, which is difficult to fully defend against.

3. *Importance of Physical Design:* Both the blue teams and the red team faced similar challenges that are closely related to PD (Sec. 8.3).

These findings are consequential: they clearly underscore the need for carefully managing the layout-level intricacies of modern ICs, for both the defenders' and attackers' perspective. This is in contrast to prior art which stated that, past certain utilization thresholds, HTs cannot be inserted anymore due to lack of placement and routing resources [BDG+13, WWF+23, XT13, BDP+16].

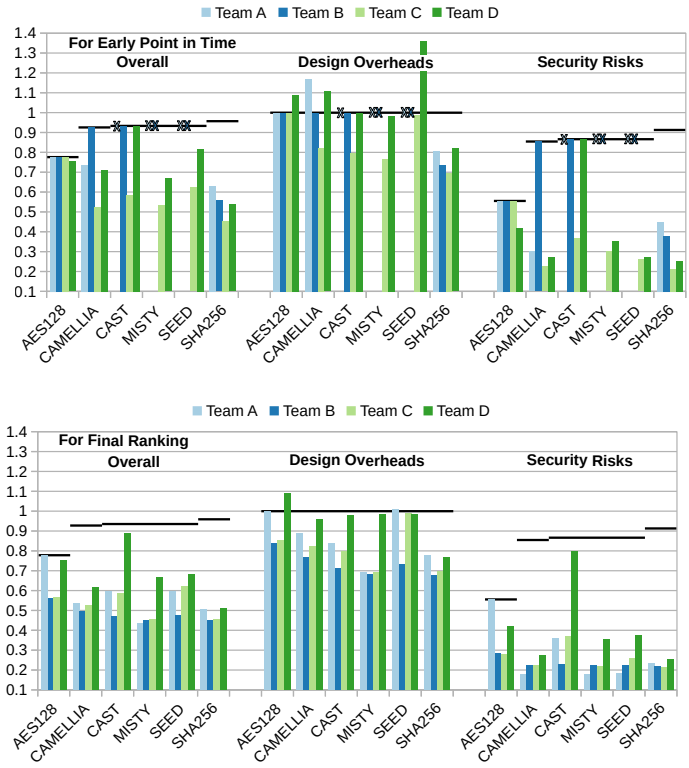


Figure 5: Scores for different points in time. For the early point, note that cases with no valid submission yet are represented by *X* labels. For overall scores and security-risk scores, note the varying positioning of the bold black line, i.e., the baseline reference does shift for the different benchmarks.

Figure 5 illustrates the overall scores versus the separate scores for design overheads and security risks, for two different points in time.[14] As expected, the blue teams performed differently well across benchmarks and points in time. The final ranking's outcome is: 1st place for Team B, 2nd place for Team C, 3rd place for Team A, and 4th place for Team D. More related details are provided in App. B.

8.1 Pre-Attack Evaluation

The following results analyze the blue teams' protected layouts in general, i.e., prior to the attack attempts by the red team.

Design Overheads. Related detailed scores are given in Fig. 6. Most blue teams succeeded in maintaining the design quality, even improving it to some degree. Recall that the organizers refrained, on purpose, from overly optimizing the baseline layouts (Sec. 6).

While the protected layouts exhibit marginal variations in power consumption, there are considerable variations in timing. This indicates on the different strategies pursued by the blue teams (Sec. 5.3): while Teams B and C automatically optimized for timing, Teams A and D did this manually. Regarding IC outlines, blue teams sought, most of

[14]For this and other score graphs, note the following. First, all data relates to the blue teams' best submissions in terms of overall scores. Second, the baseline reference points, i.e., the normalized scores obtained for benchmark layouts as is, illustrated as bold black lines, do shift across score categories and benchmarks. This is because of the fact that, even for the benchmark layouts as is, insertion of some HTs can be challenging for the red team. Importantly, this fact shows that the HTs devised for this effort are not optimized for ease of insertion. This is done on purpose, to enable a stringent assessment of layout-level efforts required even for advanced adversaries.

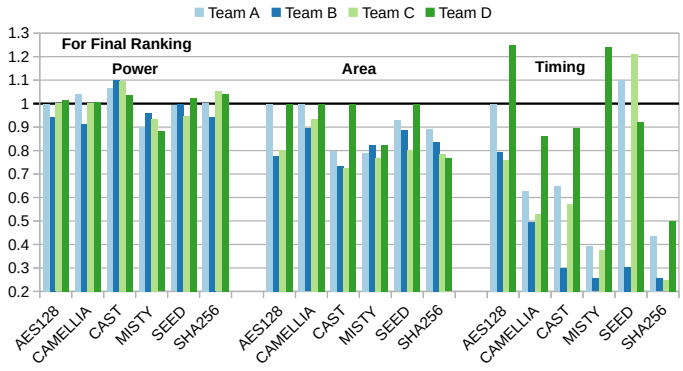


Figure 6: Detailed scores for design overheads, final ranking.

the time, to shrink layouts to some good degree. While this is beneficial for reducing IC manufacturing cost in general, it also hints at an important trade-off as discussed next.

Trade-Off for Design Quality and Efforts versus Security Risks. The smaller an IC's outline becomes, the higher the layout utilization becomes, and the fewer placement and routing resources remain available. As a result, the more difficult HT insertion *may* become, but, at the same time, the more difficult PD becomes in general.

Thus, the blue teams faced practical challenges when attempting to exhaustively harden the layouts by pushing for ever-higher utilization. This is especially true since the threat model for this effort (Sec. 4) did not allow for any spares or fillers. Still, the teams achieved remarkably high utilization numbers for their protected, violations-free layouts for the final ranking,[15] albeit at the cost of considerable R&D efforts (Sec. 1, App. A).

In short, there is a delicate balance defenders must strike between making layouts resilient against HT insertion and maintaining some relative ease of the design process itself. Furthermore, while there is some good promise, it requires further investigation, as below, to judge how resilient the protected layouts truly are.

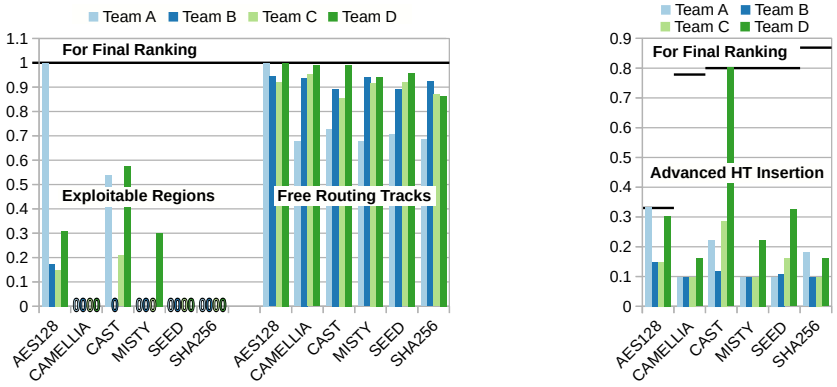
Assessment of Generic Prospects for HT Insertion. Related scores are given in Fig. 7(a). All blue teams managed, in most cases, to improve on security. At the same time, however, at least some layout resources remain potentially exploitable in all cases.

Consider the benchmark CAST as one of the most challenging to defend. This is because its baseline layout has the lowest placement utilization, and, as shown in Fig. 8(a), the largest number of exploitable regions (aside from the overall largest benchmark AES128). In fact, only the winning Team B managed to reduce exploitable regions down to zero for CAST (Fig. 8[b]). Note that, while there are no exploitable regions remaining as such (i.e., 20+ continuously arranged open placement sites), there are still various smaller regions and open sites remaining in their protected layout.

It is important to note that *none* of the blue teams managed to reduce open sites and free routing tracks down to zero for any of the benchmarks. This shortcoming is expected as, again, PD becomes ever-more challenging when pushing toward 100% utilization, which itself is impractical to achieve.

In short, despite the blue teams' best efforts, reaching for the absence of any exploitable resources remained elusive, underscoring the difficulty of completely eliminating HT attack surfaces at the layout-level. Thus, it remains to be seen—in the next subsection—whether these efforts can truly prevent HT insertion.

[15] More specifically, 67.34–90.21% for benchmark AES128, 84.70–93.44% for CAMELLIA, 53.76–87.57% for CAST, 77.89–91.61% for MISTY, 78.51–92.05% for SEED, and 91.24–94.61% for SHA256, respectively.



(a) Pre-attack evaluation. **(b)** Post-attack eval.

Figure 7: Detailed scores for security risks, final ranking. For exploitable regions in (a), note that score values of 0.0 are represented by *0* labels. For (b), note the varying positioning of the bold black line, i.e., the baseline reference does shift for the different benchmarks, as in Fig. 5.

8.2 Assessment of Red-versus-Blue Efforts

The following results analyze the blue teams' protected layouts under real-world attacks through the red team's HT insertion. Importantly, note that further results are also given in Sec. 9.2 and in App. D.1.

Scores. See Fig. 7(b). In most cases, the blue teams' defense efforts rendered HT insertion by the red team considerably more challenging, with the majority of attack trials falling into the categories of either design failures or DRC violations. However, none of the teams managed to fully prevent insertion of all the HTs considered for each benchmark. Note that this can be seen from the fact that none of the scores fall toward zero, i.e., where design failures would occur for all insertion modes and all HTs (Sec. 5.5).

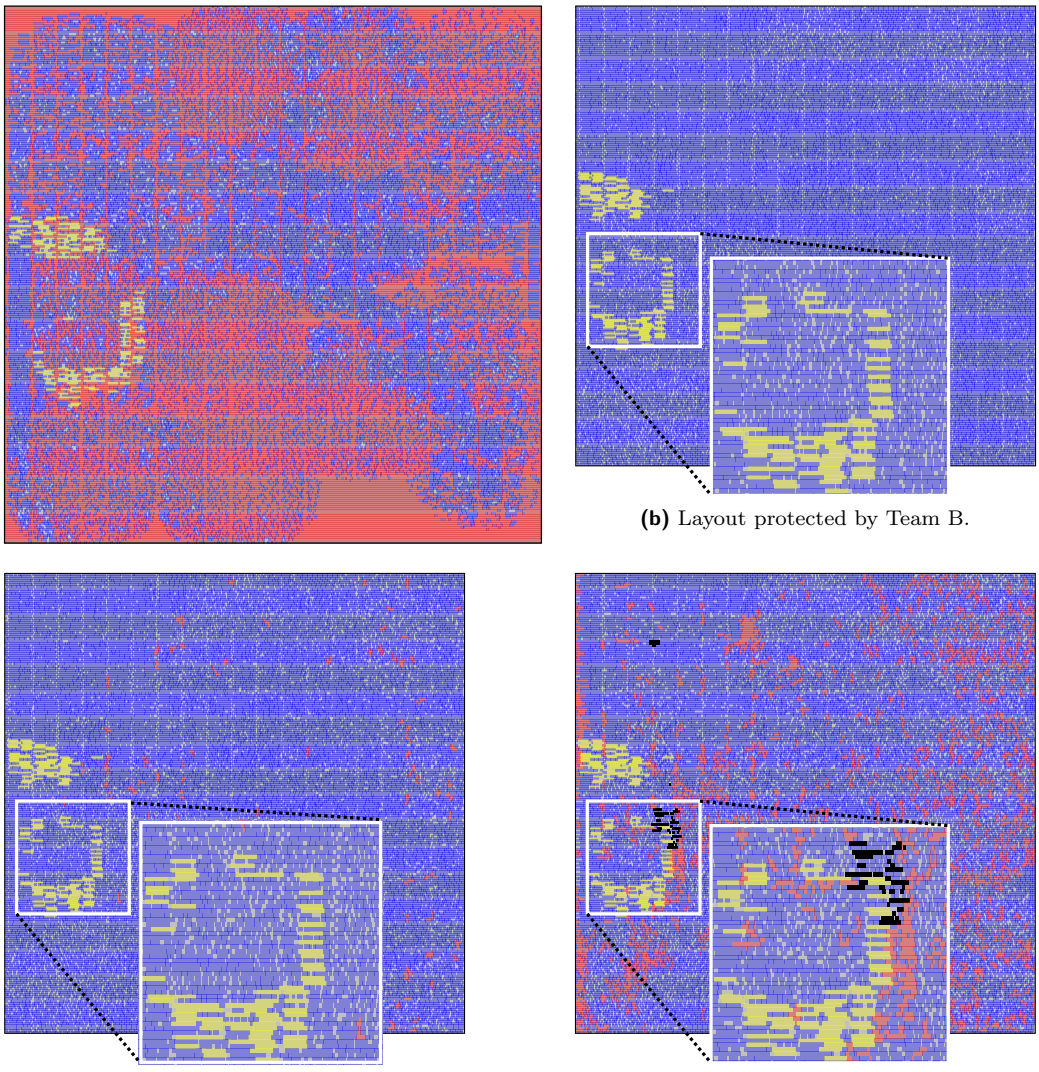
Full Example. Figure 8 illustrates a concrete and full example of defense versus attack efforts for the challenging benchmark CAST. Note the following here. First, although zero exploitable regions remain due to the defense efforts by the winning Team B (Fig. 8[b]), smaller regions and further open sites still remain. Recall that this situation is common to all benchmarks and defense efforts (Sec. 8.1). Second, the red team does face challenges for HT insertion: both the conservative and moderate insertion modes fail (Fig. 8[c]).[16] Third, for the aggressive mode, however, the red team succeeds (Fig. 8[d]). In this particular example for the HT *CAST-leak-targeted*, insertion readily succeeded even without inducing any DRC violations, clearly demonstrating the practicality of the attack.

Summary. While the various defense efforts did render HT insertion indeed more challenging in general, the attackers could still succeed to some degree. In fact, the aggressive mode enabled the red team to insert *all* different HTs into *all* protected layouts across *all* benchmarks, and in few cases even readily with zero timing and/or zero DRC violations. This reaffirms the severity of the threat: attackers may succeed even when no obvious layout vulnerabilities remain after the various, sophisticated defense efforts.

[16] The layout under attack in conservative mode is not shown here as it would be redundant to Fig. 8(b). Recall that this mode operates on the protected layout as is (Sec. 5.4), and since actual HT insertion fails, there would be no differences to Fig. 8(b).

8.3 Discussion

The preceding analysis shows that both the blue and red teams faced significant challenges related to the PD process. This is expected as the main focus of this effort was to thoroughly explore the layout-level prospects for attack versus defense efforts, all subject



(b) Layout protected by Team B.

(c) Protected layout under moderate attack.

(d) Protected layout under aggressive attack.

Figure 8: Layout illustrations for selected red-versus-blue efforts on benchmark CAST. HT cell instances are shown in black; all other components are shown as in Fig. 4. (Top-left) Baseline layout. Note the widespread nature of exploitable regions. Also note how the baseline layout is larger than the protected ones; the differences shown here are to scale of the actual layouts. (b) Layout protected by blue Team B, as obtained from final ranking. Note that zero exploitable regions remain. Also note that white insets are zooming-in 2× on the same region of interest across all protected layouts. (c) The protected layout under attack, in moderate mode for HT insertion. Note that there are no HT instances shown as HT insertion fails. (d) The protected layout under attack in aggressive mode. Insertion of the HT *CAST-leak-targeted* succeeded with zero DRC violations.

Table 3: Statistics for Blue Teams' Submissions

	Invalid Due to Violations For:						Valid
	Cell Assets	**Functional Equivalence**	**Timing**	**DRCs**	**Additional Design & Technology Checks**	**Others**	
Team A	0	1	6	10	97	7	54
Team B	0	0	4	15	48	1	37
Team C	3	1	0	5	25	12	94
Team D	0	0	3	6	3	3	62
Overall	3	2	13	36	173	23	247

Note the following for data organization. First, the statistics are combined across all benchmarks. Second, for simplicity, *Additional Design & Technology Checks* merges all additional design and technology checks (App. E.4 and C). Third, *Others* refers to any remaining, uncategorized failures, like syntax errors in the submission layout files.

to a modern and complex IC technology node. These challenges are discussed next in some detail. Limitations arising from the contest format are outlined as well. Additional considerations and a broader discussion of the overall effort is also provided in App. A.

Challenges and Limitations for Blue Teams. The common ground for all blue teams was to tune the PD process to harden layouts against HT insertion. Recall that the teams sought to increase both placement and routing utilization of layouts carefully, without inducing any violations, through different means (Sec. 5.3).

Despite their different approaches, all blue teams faced considerable challenges, as evidenced from the submission statistics in Tab. 3. First, the vast majority of invalid submissions were due to some violation of additional design and technology checks. Recall that these extensive checks were carefully devised by the organizers for detecting any trivial defense efforts. Thus, these numbers attest to the blue teams' efforts to push their techniques for the most competitive layout-level defenses in this stringent setting. Second, DRC violations represent the largest share across all remaining violation categories. This is because for an advanced technology node like that described by *ASAP7*, design rules are strict. Again, the key observation here is that "exhaustive hardening of modern IC layouts against HT insertion is extremely challenging," which makes this effort as a whole timely and practically relevant.

While the blue teams did seek to automatically resolve DRC violations, this turned out difficult to scale up for highly-utilized layouts. Eventually, all blue teams had to resort to manual fixing of any remaining DRC violations. This highlights the limitations of automated hardening techniques for modern IC layouts, also underscoring the need for more sophisticated defense strategies.

Challenges and Limitations for Red Team. The red team faced additional challenges since they had to be working on layouts already hardened by the blue teams. More details are as follows.

First, throughout the contest, the techniques for advanced HT insertion employed by the red team did, aside from some bug fixing, not progress. This was dictated by the organizers to avoid "moving targets" for the blue teams, to (i) avoid demotivating blue teams and (ii) enable a fair and consistent competition. Second, like the blue teams, the red team did not manage to employ techniques for automatic resolution of DRC violations. Third, since the red team's efforts had to be integrated into the fully automated back-end pipeline (App. E.1), the red team was, unlike the blue teams, however, *not* allowed to employ any manual efforts either. Importantly, these aspects emphasize the difficulty of fully automating HT insertion in such a complex and dynamic environment that is driven by a competition toward the most effective defenses.

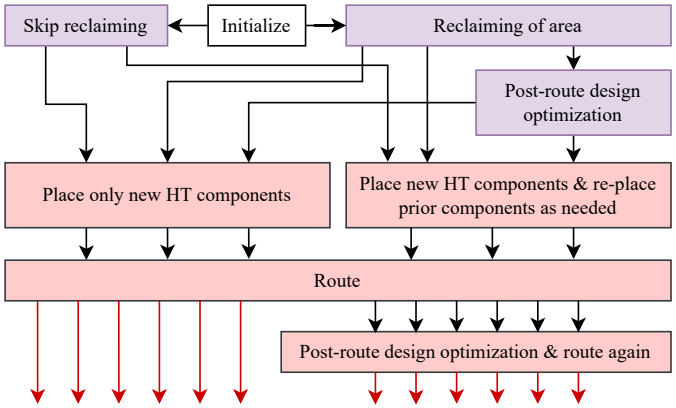


Figure 9: Extended attack techniques. The orchestration fans out into 12 different, fully automated modes for advanced HT insertion, as indicated by dark-red arrows. Pre-attack steps are highlighted in purple and attack steps for physical integration in light-red.

9 Further Assessment of Red-versus-Blue Efforts

Especially from the red team's perspective, given both the promising results (Sec. 8.2) as well as the significant limitations (Sec. 8.3) encountered during the contest, it is highly warranted to further explore the scope of red-versus-blue efforts. Thus, the post-contest analysis in this section investigates the persistence of the threat raised by HT insertion.

Here, the red team took further efforts to devise extended attack techniques and to study the practicality of manual intervention as needed. Such efforts are important to successfully embed HTs within the real-world constraints of the IC industry: attackers must fix at least all DRC violations; otherwise, the HT-infested layout would likely not be manufacturable at all. It is also important to note that, unlike prior art which often implicitly assumes manual HT insertion, the focus here remains on automated HT insertion by means of ECO techniques, with manual intervention limited to sparingly fixing any remaining violations after the automated insertion.

The key finding of this analysis is that advanced HT insertion is re-affirmed as a realistic, practical, and effective threat. This holds significant ramifications for security-aware design of modern ICs, such as the need to advance defenses further and the need for continuous red-versus-blue efforts toward realistic security assessments.

9.1 Methods

Case Study I: Extended Techniques for HT Insertion. This study explores whether advanced HT insertion can be made more effective, as in automatically achieving fewer violations, thereby requiring less manual intervention by attackers. Thus, this study deepens the understanding of the practicality for advanced HT insertion in real-world scenarios where attackers should not only fix violations but would be time-constrained while doing so, e.g., to avoid suspicions from colleagues and/or supervisors regarding some unusually long turn-around time for their activities.

Based on a thorough analysis of the contest results and further elaborate experiments, the red team devised a range of extended techniques for both the pre-attack stage (i.e., reclaiming of area, Sec. 5.4.2) and the attack stage for actual physical integration of HTs (i.e., ECO PnR, Sec. 5.4.3). The extended techniques are outlined in Fig. 9. Further technical details are provided in the release (App. C).

Table 4: Outcomes for Different Attack Settings

		Design Failures	DRC Vio.	Timing Vio. (Stp&Hld)	Timing Vio. (Stp⊕Hld)	DRV or Clock Vio.	No Vio.
Team A	**AIC**	0	28	21	11	6	0
	EXT	0	32	0	4	5	4
Team B	**AIC**	0	33	24	6	12	0
	EXT	0	27	0	2	0	9
Team C	**AIC**	0	35	20	12	0	1
	EXT	0	31	0	15	0	5
Overall	**AIC**	0	96	65	29	18	1
	EXT	0	90	0	21	5	18

Vio. is short for violations, *Stp&Hld* for setup `AND` hold, and *Stp⊕Hld* for setup `XOR` hold, respectively. For attack settings in rows, note that *AIC* is short for "as in contest" and *EXT* for "extended techniques." Overall best outcomes are highlighted in green. Note the following for data organization. First, the best outcomes across all 36 HTs and benchmarks are summarized here. Second, unlike with the scoring procedure, any less severe violations occurring at the same time are reported as well. For example, there are cases with DRC violations that also exhibit some timing violations.

Case Study II: Manual Fixing of Violations. This study explores what efforts are required to fix any remaining violations that occurred for the different techniques and settings for advanced HT insertion. For fixing of DRC violations as well as timing and DRV violations, two separate procedures are devised, following best practices from industry. See App. D.3 for related technical details.

9.2 Results

Note that the results discussed here concern the protected layouts as obtained for the final ranking from the top-three blue teams.

Case Study I: Extended Techniques for HT Insertion. Table 4 summarizes the outcomes for advanced HT insertion for the two different sets of techniques, or attack settings, namely techniques as in contest (AIC) versus extended techniques (EXT).

Across all the best protected layouts, the EXT setting succeeds to reduce the number of attack trials with violations by 44% (from 208 to 116). Remarkably, the largest improvements are observed for the layouts protected by the contest winners, Team B: trials with violations are reduced by 61% here (from 75 to 29).

In general, DRC violations are the most challenging to mitigate; with 78%, they represent the largest share of violations remaining also for the EXT setting. Thus, the scope of DRC violations is investigated in more detail next. Contrasting EXT to AIC, the average counts of DRC violations are reduced, approximately, by 33% for Team A (from 52 to 35), by 62% for Team B (from 21 to 8), and by 30% for Team C (from 10 to 7), respectively. The best gains are again observed on Team B's layouts. Thus, while their defense techniques were the most competitive during the contest, they lack in general robustness against more advanced modes for HT insertion. For all teams, there are also a few cases where DRC violations increase; see App. D.2 for details. Also see Fig. 11 in the same appendix for detailed plots of DRC violations.

In short, while the extended techniques clearly demonstrate that attackers can adapt and improve to overcome even the best defenses, case-by-case differences remain. The latter is expected, given the intricate interplay for red-versus-blue efforts.

Runtimes. It is also important to note that all attack techniques, i.e., for both the AIC and the EXT settings, incur rather short runtimes, namely only tens of minutes for any of the attack trials in this study. (In contrast, re-running PD in full, especially when facing these hardened layouts, could well take multiple hours for larger benchmarks.) This is thanks to the localized working of ECO algorithms toward minor design/layout changes. As indicated, short runtimes would be beneficial for adversaries to avoid raising suspicions

for their malicious activities. That very benefit of localized focus, however, can result in few violations remaining, as seen above. The prospects for fixing such violations are discussed next.

Case Study II: Manual Fixing of Violations. The red team conducted a number of trials for manual fixing of violations remaining for both the AIC and EXT attack settings. The related findings are summarized next, and note that an example for successful fixing of DRC violations is illustrated in Fig. 12, App. D.3

First, prospects for fixing are difficult to generalize; such efforts require case-by-case considerations. This is expected from the perspective of PD: there are specific reasons as to why violations arise in the first place, despite the well-established algorithms of contemporary design tools. Second, as a rule of thumb, layouts with more than 30 DRC violations and/or timing violations of multiple tens or even hundreds of ps are too difficult to fix completely. With that in mind, the red team finds that, for the AIC versus EXT settings, around 68% versus 81% of cases with DRC violations and 59% versus 100% of cases with timing violations could be fixed, respectively.

Summary. Overall, i.e., also considering practical best efforts for manual fixing of remaining violations, around 45% of the AIC trials versus 84% of the EXT trials could be pushed for zero violations, respectively. In other words, with the extended attack techniques, the red team has demonstrated a significant new milestone for state-of-the-art in HT insertion: for 84% of all attack trials, across all the commendable defenses, HT insertion would be fully successful.

10 Conclusions

This work provides a pioneering investigation of fabrication-time HT insertion versus layout-level defenses. Its unique aspects include a strict red-versus-blue setting, open-source release of all materials, and a strong emphasis on real-world relevance through the use of a modern technology node and commercial design tools.

Lessons Learned. Our effort yields a wide range of important findings, covering both the attackers' and defenders' perspectives, as follows. Note that the research questions (Sec. 1) for both defenders (RQ-D) and attackers (RQ-A) are also addressed here.

1. Challenges for HT insertion are closely related to IC design, in more complex ways than prior art like [MGK+13, GMMP20, PP22, PMB+23, TSBH20, BDG+13, WWF+23, XT13, BDP+16] had recognized (RQ-A3, RQ-D4). The interplay between defense and attack efforts is heavily influenced by real-world constraints and the intricate interactions between layout resources and design components, which ultimately determine the effectiveness of any effort.

2. Regular, security-unaware IC design leaves considerable layout resources exploitable, as evident from the evaluation of the benchmarks' baseline layouts (RQ-D1).

3. Layout-level defenses are practical in general and, when done carefully, even without undermining design quality (RQ-D3). Techniques shown to be effective across the board are (i) shrink IC outlines, (ii) parameter tuning for the design flow, and (iii) insertion of additional components (RQ-D2). For truly competitive efforts, however, the related push toward highly utilized layouts brings significant challenges for violations-free design closure (RQ-D4).

4. The (mis-)use of engineering change order (ECO) techniques by the red team, an industry-wide standard for minor design modifications, is demonstrated as an effective and efficient approach (RQ-A1). While most prior art assumed—often only implicitly—that HTs are manually inserted, we argue that doing so could become

error-prone and time-consuming, especially for modern ICs with their significant complexities and constraints for the design and manufacturing processes.

5. Despite layout defenses in place, ECO-based HT insertion remains effective (RQ-A2). The red team succeeds in their initial attempts, at least partially, across all benchmarks, all HTs, and all the defense efforts. Furthermore, attackers can employ extended ECO techniques as well as manual efforts toward violations-free integration of HTs (RQ-A3). Such efforts took, on average, only few minutes, and achieved zero violations after HT insertion for the vast majority of all attack trials. Importantly, this lesson highlights both the demonstrated difficulties and the urgent need for advanced defenses efforts to counter such powerful attack techniques. Clearly, prior art which stated that, past a certain utilization threshold, HTs cannot be inserted anymore [BDG+13, WWF+23, XT13, BDP+16], is refuted.

Takeaways and Ramifications for Defenders. Our extensive experimental results and the resulting lessons above show that there is no clear upper hand for either side: while defenders and attackers could significantly advance their capabilities during this months-long effort, neither of them could fully hinder their opponents' efforts. This is especially concerning for defenders. Ultimately, the threat model is confirmed as realistic—further advances for defenses are urgently needed. Such advances will be challenging to realize, due to the layout-level intricacies of modern ICs as well as the asymmetric nature for defense versus attack efforts. That is, while defenders need to proactively protect most if not all layout resources against various potential HTs, attackers can react by carefully designing ever-more advanced and smaller HTs. For example, the HTs considered in this work incur just $< 0.5\%$ additional standard cells over the baseline layouts (App. F).

Our large-scale community effort highlights the necessity and benefits of continuous red-versus-blue teaming for both the design of sensitive ICs in general and as an integral part for security-centric R&D efforts in particular. This is important because the success of defense strategies depends not only on their own techniques but also on the attackers' approach. Red-versus-blue teaming is also crucial for addressing the challenge of advancing defenses in the face of asymmetric defense versus attack efforts—defenders need to proactively explore various iterations of hardened layouts under realistic attack scenarios based on actual HT insertion. The proposed framework, which has demonstrated significant advances over prior art, enables such an integrated evaluation approach.

Future Work. While this work thoroughly covers proactive, pre-silicon defense efforts against HT insertion at the layout level, we acknowledge that a comprehensive defense strategy should also cover orthogonal efforts, e.g., toward post-silicon detection. Thus, future red-versus-blue teaming efforts could focus on attackers optimizing HTs to avoid post-silicon detection versus defenders hardening layouts to both resist HT insertion and assist detection. In that context, from the attacker's perspective, other types of HTs like non-additive, trigger-less, and/or analog HTs seem promising as they should require less complex layout-level modifications and should be more stealthy against detection. From the defender's perspective, high-level countermeasures that could assist detection, like self-testing circuitry or logic locking realized at the netlist level, should be further studied for their prospects of seamless integration during layout hardening.

Wider Implications. Beyond the specific context of HT insertion, we argue that design-time efforts for hardware security are needed at even wider scales. There are other well-demonstrated threats, like side-channel or fault-injection attacks to extract sensitive data from ICs at runtime, which might all be mitigated by layout-level defense efforts [HCS+20, KKR+20, KGB+21, LRT+21, BMN+24, KSS+18]. However, as industrial workflows currently lack such capabilities, contemporary ICs are left vulnerable. Thus, we also call for further interaction with practitioners, especially through red-versus-blue teaming efforts that carefully consider the real-world constraints and challenges for the design and manufacturing of modern and high-quality, yet secure ICs.

Acknowledgments

The authors thank the TCHES reviewers and editors for their time and service, and the ISPD committee and community for supporting the contest. This work was supported in part by the NYUAD Center for Cybersecurity (CCS) as well as the NYU CCS, the EU through the European Social Fund in the context of the project "ICT programme," and the National Science and Technology Council of Taiwan under Grant No. 113-2640-E-007-001. The work of Mohammad Eslami was also supported by the HARNO under Grant No. 11.4-1/23/1.

References

[ABK+07] Dakshi Agrawal, Selcuk Baktir, Deniz Karakoyunlu, Pankaj Rohatgi, and Berk Sunar. Trojan detection using IC fingerprinting. In *Proc. Symp. Sec. Priv.*, pages 296–310, 2007.

[Ade08] Sally Adee. The hunt for the kill switch. *IEEE Spectrum*, 45(5):34–39, 2008.

[AGK+15] Markus Ahrens, Michael Gester, Niko Klewinghaus, Dirk Müller, Sven Peyer, Christian Schulte, and Gustavo Téllez. Detailed routing algorithms for advanced technology nodes. *Trans. Comp.-Aided Des. Integ. Circ. Sys.*, 34(4):563–576, 2015.

[AIRP22] Felipe Almeida, Malik Imran, Jaan Raik, and Samuel Pagliarini. Ransomware attack as hardware Trojan: A feasibility and demonstration study. *IEEE Access*, 10:44827–44839, 2022.

[BDG+13] Shivam Bhasin, Jean-Luc Danger, Sylvain Guilley, Xuan Thuy Ngo, and Laurent Sauvage. Hardware Trojan horses in cryptographic IP cores. In *Proc. Worksh. Fault Diag. Tol. Cryptogr.*, pages 15–29, 2013.

[BDP+16] Papa-Sidy Ba, Sophie Dupuis, Manikandan Palanichamy, Marie-Lise Flottes, Giorgio Di Natale, and Bruno Rouzeyre. Hardware trust through layout filling: A hardware Trojan prevention technique. In *Proc. Comp. Soc. Symp. VLSI*, 2016.

[BHBN14] Swarup Bhunia, Michael S. Hsiao, Mainak Banga, and Seetharam Narasimhan. Hardware Trojan attacks: Threat analysis and countermeasures. *Proc. IEEE*, 102(8):1229–1247, 2014.

[BLL+11] Georg T. Becker, Ashwin Lakshminarasimhan, Lang Lin, Sudheendra Srivathsa, Vikram B. Suresh, and Wayne Burelson. Implementing hardware Trojans: Experiences from a hardware Trojan challenge. In *Proc. Int. Conf. Comp. Des.*, pages 301–304, 2011.

[BMN+24] Jitendra Bhandari, Likhitha Mankali, Mohammed Nabeel, Ozgur Sinanoglu, Ramesh Karri, and Johann Knechtel. Beware your standard cells! on their role in static power side-channel attacks. *Trans. Comp.-Aided Des. Integ. Circ. Sys.*, 2024.

[BPK+22] Kyeonghyeon Baek, Hyunbum Park, Suwan Kim, Kyumyung Choi, and Taewhan Kim. Pin accessibility and routing congestion aware DRC hotspot prediction using graph neural network and u-net. In *Proc. Int. Conf. Comp.-Aided Des.*, 2022.

[BRPB13] Georg T. Becker, Francesco Regazzoni, Christof Paar, and Wayne P. Burleson. Stealthy dopant-level hardware Trojans. In *Proc. Cryptogr. Hardw. Embed. Sys.*, pages 197–214, 2013.

[CHL+21] Chung-Kuan Cheng, Chia-Tung Ho, Daeyeal Lee, Bill Lin, and Dongwon Park. Complementary-FET (CFET) standard cell synthesis framework for design and system technology co-optimization using SMT. *Trans. VLSI Syst.*, 29(6):1178–1191, 2021.

[CKSL17] Kyungwook Chang, Bon Woong Ku, Saurabh Sinha, and Sung Kyu Lim. Full-chip monolithic 3D IC design and power performance analysis with ASAP7 library: (invited paper). In *Proc. Int. Conf. Comp.-Aided Des.*, pages 1005–1010, 2017.

[CNB09] Rajat Subhra Chakraborty, Seetharam Narasimhan, and Swarup Bhunia. Hardware Trojan: Threats and emerging solutions. In *Proc. Int. HL Des. Val. Test*, pages 166–171, 2009.

[Cry07] Cryptographic hardware project, 2007.

[CVH+17] Lawrence T. Clark, Vinay Vashishtha, David M. Harris, Samuel Dietrich, and Zunyan Wang. Design flows and collateral for the ASAP7 7nm FinFET predictive process design kit. In *Proc. Int. Conf. Microel. Sys. Edu.*, pages 1–4, 2017.

[CVS+16] Lawrence T. Clark, Vinay Vashishtha, Lucian Shifren, Aditya Gujja, Saurabh Sinha, Brian Cline, Chandarasekaran Ramamurthy, and Greg Yeric. ASAP7: A 7-nm FinFET predictive process design kit. *Microelectronics Journal*, 53:105–115, 2016.

[CWP+09] Rajat Subhra Chakraborty, Francis Wolff, Somnath Paul, Christos Papachristou, and Swarup Bhunia. MERO: A statistical approach for hardware Trojan detection. In *Proc. Cryptogr. Hardw. Embed. Sys.*, pages 396–410, 2009.

[DGH+19] Ghada Dessouky, David Gens, Patrick Haney, Garrett Persyn, Arun Kanuparthi, Hareesh Khattri, Jason M. Fung, Ahmad-Reza Sadeghi, and Jeyavijayan Rajendran. HardFails: Insights into software-exploitable hardware bugs. In *Proc. USENIX Sec. Symp.*, pages 213–230, 2019.

[DNCB10] Dongdong Du, Seetharam Narasimhan, Rajat Subhra Chakraborty, and Swarup Bhunia. Self-referencing: A scalable side-channel approach for hardware Trojan detection. In *Proc. Cryptogr. Hardw. Embed. Sys.*, pages 173–187, 2010.

[DXL+20] Chen Dong, Yi Xu, Ximeng Liu, Fan Zhang, Guorong He, and Yuzhong Chen. Hardware Trojans in chips: A survey for detection and prevention. *Sensors*, 20(18), 2020.

[EPP23] Mohammad Eslami, Tiago Perez, and Samuel Pagliarini. SALSy: Security-aware layout synthesis. 2023.

[Fre12] FreeCores: tiny_aes128, 2012.

[GBF17] Lonel Acunha Guimarães, Rodrigo Possamai Bastos, and Laurent Fesquet. Detection of layout-level Trojans by monitoring substrate with preexisting built-in sensors. In *Proc. Comp. Soc. Symp. VLSI*, pages 290–295, 2017.

[GGPR22] Vasudev Gohil, Hao Guo, Satwik Patnaik, and Jeyavijayan Rajendran. ATTRITION: Attacking static hardware Trojan detection techniques using reinforcement learning. In *Proc. Comp. Comm. Sec.*, pages 1275–1289, 2022.

[GMMP20] Samaneh Ghandali, Thorben Moos, Amir Moradi, and Christof Paar. Side-channel hardware Trojan for provably-secure SCA-protected implementations. *Trans. VLSI Syst.*, 28(6):1435–1448, 2020.

[GYT+23] Guangxin Guo, Hailong You, Zhengguang Tang, Benzheng Li, Cong Li, and Xiaojue Zhang. ASSURER: A PPA-friendly security closure framework for physical design. In *Proc. Asia South Pac. Des. Autom. Conf.*, pages 504–509, 2023.

[HCC+23] Jhih-Wei Hsu, Kuan-Cheng Chen, Yan-Syuan Chen, Yu-Hsiang Lo, and Yao-Wen Chang. Security-aware physical design against Trojan insertion, frontside probing, and fault injection attacks. In *Proc. Int. Symp. Phys. Des.*, pages 220–228, 2023.

[HCS+20] W. Hu, C. H. Chang, A. Sengupta, S. Bhunia, R. Kastner, and H. Li. An overview of hardware security and trust: Threats, countermeasures and design tools. *Trans. Comp.-Aided Des. Integ. Circ. Sys.*, 2020.

[HPPS22] Alexander Hepp, Tiago Perez, Samuel Pagliarini, and Georg Sigl. A pragmatic methodology for blind hardware Trojan insertion in finalized layouts. In *Proc. Int. Conf. Comp.-Aided Des.*, 2022.

[HYT17] Kento Hasegawa, Masao Yanagisawa, and Nozomu Togawa. Trojan-feature extraction at gate-level netlists and its application to hardware-Trojan detection using random forest classifier. In *Proc. Int. Symp. Circ. Sys.*, pages 1–4, 2017.

[JMHS14] Nisha Jacob, Dominik Merli, Johann Heyszl, and Georg Sigl. Hardware Trojans: Current challenges and approaches. *Comp. & Dig. Techs.*, 8(6):264–273, 2014.

[KCU24] Chaudhry Indra Kumar, Abhishek Chaudhary, and Shreyansh Upadhyaya. Design of high performance energy efficient CMOS voltage level shifter for mixed signal circuits applications. *Integration*, 95:102133, 2024.

[KGB+21] J. Knechtel, J. Gopinath, J. Bhandari, M. Ashraf, H. Amrouch, S. Borkar, S.-K. Lim, O. Sinanoglu, and R. Karri. Security closure of physical layouts. In *Proc. Int. Conf. Comp.-Aided Des.*, 2021.

[KKR+20] J. Knechtel, E. B. Kavun, F. Regazzoni, A. Heuser, A. Chattopadhyay, D. Mukhopadhyay, S. Dey, Y. Fei, Y. Belenky, I. Levi, T. Güneysu, P. Schaumont, and I. Polian. Towards secure composition of integrated circuits and electronic systems: On the role of EDA. In *Proc. Des. Autom. Test Europe*, 2020.

[KLMH11] Andrew B. Kahng, Jens Lienig, Igor L. Markov, and Jin Hu. *VLSI Physical Design: From Graph Partitioning to Timing Closure*. Springer, 2011.

[Kne21] J. Knechtel. Hardware security for and beyond CMOS technology. In *Proc. Int. Symp. Phys. Des.*, 2021.

[Kri23] Christian Krieg. Reflections on trusting TrustHUB. In *Proc. Int. Conf. Comp.-Aided Des.*, 2023.

[KRRT10] Ramesh Karri, Jeyavijayan Rajendran, Kurt Rosenfeld, and Mohammad Tehranipoor. Trustworthy hardware: Identifying and classifying hardware Trojans. *Computer*, 43(10):39–46, 2010.

[KSS+18] Mustafa Khairallah, Rajat Sadhukhan, Radhamanjari Samanta, Jakub Breier, Shivam Bhasin, Rajat Subhra Chakraborty, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. DFARPA: Differential fault attack resistant physical design automation. In *Proc. Des. Autom. Test Europe*, pages 1171–1174, 2018.

[KY13] Jian Kuang and Evangeline F. Y. Young. An efficient layout decomposition approach for triple patterning lithography. In *Proc. Des. Autom. Conf.*, 2013.

[LAKS23] Hazem Lashen, Lilas Alrahis, Johann Knechtel, and Ozgur Sinanoglu. TrojanSAINT: Gate-level netlist sampling-based inductive learning for hardware Trojan detection. In *Proc. Int. Symp. Circ. Sys.*, 2023.

[LBL+22] Bernhard Lippmann, Ann-Christin Bette, Matthias Ludwig, Johannes Mutter, Johanna Baehr, Alexander Hepp, Horst Gieser, Nicola Kovač, Tobias Zweifel, Martin Rasche, and Oliver Kellermann. Physical and functional reverse engineering challenges for advanced semiconductor solutions. In *Proc. Des. Autom. Test Europe*, pages 796–801, 2022.

[LCP+21] Nimisha Limaye, Animesh B. Chowdhury, Christian Pilato, Mohammed T. M. Nabeel, Ozgur Sinanoglu, Siddharth Garg, and Ramesh Karri. Fortifying RTL locking against oracle-less (untrusted foundry) and oracle-guided attacks. In *Proc. Des. Autom. Conf.*, pages 91–96, 2021.

[LFL+24] Lixin Liu, Bangqi Fu, Shiju Lin, Jinwei Liu, Evangeline F. Y. Young, and Martin D. F. Wong. Xplace: An extremely fast and extensible placement framework. *Trans. Comp.-Aided Des. Integ. Circ. Sys.*, 43(6):1872–1885, 2024.

[LJM12] Eric Love, Yier Jin, and Yiorgos Makris. Proof-carrying hardware intellectual property: A pathway to trusted module acquisition. *Trans. Inf. Forens. Sec.*, 7(1):25–40, 2012.

[LKG+09] Lang Lin, Markus Kasper, Tim Güneysu, Christof Paar, and Wayne Burleson. Trojan side-channels: Lightweight hardware Trojans through side-channel engineering. In *Proc. Cryptogr. Hardw. Embed. Sys.*, pages 382–395, 2009.

[LPH+21] Daeyeal Lee, Dongwon Park, Chia-Tung Ho, Ilgweon Kang, Hayoung Kim, Sicun Gao, Bill Lin, and Chung-Kuan Cheng. SP&R: SMT-based simultaneous place-and-route for standard cell synthesis of advanced nodes. *Trans. Comp.-Aided Des. Integ. Circ. Sys.*, 40(10):2142–2155, 2021.

[LRT+21] J. Lienig, S. Rothe, M. Thiele, N. Rangarajan, M. Ashraf, M. Nabeel, H. Amrouch, O. Sinanoglu, and J. Knechtel. Toward security closure in the face of reliability effects. In *Proc. Int. Conf. Comp.-Aided Des.*, 2021.

[LWU+19] Bernhard Lippmann, Michael Werner, Niklas Unverricht, Aayush Singla, Peter Egger, Anja Dübotzky, Horst Gieser, Martin Rasche, Oliver Kellermann, and Helmut Graeb. Integrated flow for reverse engineering of nanoscale technologies. In *Proc. Asia South Pac. Des. Autom. Conf.*, pages 82–89, 2019.

[MCM+19] Arkadiusz Malinowski, James Chen, Shiv Kumar Mishra, Srikanth Samavedam, and DK Sohn. What is killing Moore's law? challenges in advanced FinFET technology integration. In *Proc. Int. Conf. Mixed Des. ICs and Sys.*, pages 46–51, 2019.

[MGK+13] Michael Muehlberghuber, Frank K. Gürkaynak, Thomas Korak, Philipp Dunst, and Michael Hutter. Red team vs. blue team hardware Trojan analysis: Detection of a hardware Trojan on an actual ASIC. In *Proc. Int. Workshop Hardw. Arch. Supp. Sec. Priv.*, 2013.

[MMST23] Tahoura Mosavirik, Saleh Khalaj Monfared, Maryam Saadat Safa, and Shahin Tajik. Silicon echoes: Non-invasive Trojan and tamper detection using frequency-selective impedance analysis. In *Proc. Cryptogr. Hardw. Embed. Sys.*, pages 238–261, 2023.

[NAC+19] Mohammed Nabeel, Mohammed Ashraf, Eduardo Chielle, Nektarios G. Tsoutsos, and Michail Maniatakos. CoPHEE: Co-processor for partially homomorphic encrypted execution. In *Proc. Int. Symp. Hardw.-Orient. Sec. Trust*, pages 131–140, 2019.

[PASK19] Satwik Patnaik, Mohammed Ashraf, Ozgur Sinanoglu, and Johann Knechtel. A modern approach to IP protection and Trojan prevention: Split manufacturing for 3D ICs and obfuscation of vertical interconnects. *Trans. Emerg. Top. Comp.*, 9, 2019.

[PIVP21] Tiago Perez, Malik Imran, Pablo Vaz, and Samuel Pagliarini. Side-channel Trojan insertion - a practical foundry-side attack via ECO. In *Proc. Int. Symp. Circ. Sys.*, pages 1–5, 2021.

[PMB+23] E. Puschner, T. Moos, S. Becker, C. Kison, A. Moradi, and C. Paar. Red team vs. blue team: A real-world hardware Trojan detection case study across four modern CMOS technology generations. In *Proc. Symp. Sec. Priv.*, pages 763–781, 2023.

[PP22] Tiago Perez and Samuel Pagliarini. Hardware Trojan insertion in finalized layouts: From methodology to a silicon demonstration. *Trans. Comp.-Aided Des. Integ. Circ. Sys.*, 2022.

[RJK11] Jeyavijayan Rajendran, Vinayaka Jyothi, and Ramesh Karri. Blue team red team approach to hardware trust assessment. In *Proc. Int. Conf. Comp. Des.*, pages 285–288, 2011.

[RKK14] Masoud Rostami, Farinaz Koushanfar, and Ramesh Karri. A primer on hardware security: Models, methods, and metrics. *Proc. IEEE*, 102(8):1283–1295, 2014.

[RKM08] Jarrod A. Roy, Farinaz Koushanfar, and Igor L. Markov. EPIC: Ending piracy of integrated circuits. In *Proc. Des. Autom. Test Europe*, pages 1069–1074, 2008.

[SCN+15] Sayandeep Saha, Rajat Subhra Chakraborty, Srinivasa Shashank Nuthakki, Anshul, and Debdeep Mukhopadhyay. Improved test pattern generation for hardware Trojan detection using genetic algorithm and Boolean satisfiability. In *Proc. Cryptogr. Hardw. Embed. Sys.*, pages 577–596, 2015.

[sec13] Hardware implementation of SHA256, 2013.

[SF12] Juan Carlos Martinez Santos and Yunsi Fei. Designing and implementing a malicious 8051 processor. In *Proc. Int. Symp. Def. Fault Tol. in VLSI Nanotech. Sys.*, pages 63–66, 2012.

[SNA+22] Abhrajit Sengupta, Mohammed Nabeel, Mohammed Ashraf, Johann Knechtel, and Ozgur Sinanoglu. A new paradigm in split manufacturing: Lock the FEOL, unlock at the BEOL. *Cryptography*, 6(2), 2022.

[SSF+14] Takeshi Sugawara, Daisuke Suzuki, Ryoichi Fujii, Shigeaki Tawa, Ryohei Hori, Mitsuru Shiozaki, and Takeshi Fujino. Reversing stealthy dopant-level circuits. In *Proc. Cryptogr. Hardw. Embed. Sys.*, pages 112–126, 2014.

[ST16] Hassan Salmani and Mark M. Tehranipoor. Vulnerability analysis of a circuit layout to hardware Trojan insertion. *Trans. Inf. Forens. Sec.*, 11(6):1214–1225, 2016.

[STK13] Hassan Salmani, Mohammad Tehranipoor, and Ramesh Karri. On design vulnerability analysis and trust benchmarks development. In *Proc. Int. Conf. Comp. Des.*, 2013.

[SW12] Sergei Skorobogatov and Christopher Woods. Breakthrough silicon scanning discovers backdoor in military chip. In *Proc. Cryptogr. Hardw. Embed. Sys.*, pages 23–40, 2012.

[TK10] Mohammad Tehranipoor and Farinaz Koushanfar. A survey of hardware Trojan taxonomy and detection. *Des. Test*, 27(1):10–25, 2010.

[TSBH20] Timothy Trippel, Kang G. Shin, Kevin B. Bush, and Matthew Hicks. ICAS: an extensible framework for estimating the susceptibility of IC layouts to additive Trojans. In *Proc. Symp. Sec. Priv.*, pages 1742–1759, 2020.

[TSBH23] Timothy Trippel, Kang G. Shin, Kevin B. Bush, and Matthew Hicks. T-TER: Defeating A2 Trojans with targeted tamper-evident routing. In *Proc. Asia Comp. Comm. Sec.*, pages 746–759, 2023.

[vSAMM+16] Victor M. van Santen, Hussam Amrouch, Javier Martin-Martinez, Montserrat Nafria, and Jörg Henkel. Designing guardbands for instantaneous aging effects. In *Proc. Des. Autom. Conf.*, 2016.

[VVC17] Vinay Vashishtha, Manoj Vangala, and Lawrence T. Clark. ASAP7 predictive design kit development and cell design technology co-optimization: Invited paper. In *Proc. Int. Conf. Comp.-Aided Des.*, pages 992–998, 2017.

[WRSS14] Adam Waksman, Jeyavijayan Rajendran, Matthew Suozzo, and Simha Sethumadhavan. A red team/blue team assessment of functional analysis methods for malicious circuit identification. In *Proc. Des. Autom. Conf.*, 2014.

[WSS13] Adam Waksman, Matthew Suozzo, and Simha Sethumadhavan. FANCI: Identification of stealthy malicious logic using boolean functional analysis. In *Proc. Comp. Comm. Sec.*, pages 697–708, 2013.

[WWA+24] Fangzhou Wang, Qijing Wang, Lilas Alrahis, Bangqi Fu, Shui Jiang, Xiaopeng Zhang, Ozgur Sinanoglu, Tsung-Yi Ho, Evangeline F. Y. Young, and Johann Knechtel. TroLLoc: Logic locking and layout hardening for IC security closure against hardware Trojans. 2024.

[WWF+23] Fangzhou Wang, Qijing Wang, Bangqi Fu, Shui Jiang, Xiaopeng Zhang, Lilas Alrahis, Ozgur Sinanoglu, Johann Knechtel, Tsung-Yi Ho, and Evangeline F.Y. Young. Security closure of IC layouts against hardware Trojans. In *Proc. Int. Symp. Phys. Des.*, pages 229–237, 2023.

[WZL23] Xinming Wei, Jiaxi Zhang, and Guojie Luo. GDSII-Guard: ECO anti-Trojan optimization with exploratory timing-security trade-offs. In *Proc. Des. Autom. Conf.*, 2023.

[WZL24] X. Wei, J. Zhang, and G. Luo. Rethinking IC layout vulnerability: Simulation-based hardware Trojan threat assessment with high fidelity. In *Proc. Symp. Sec. Priv.*, pages 159–159, 2024.

[XFJ+16] K. Xiao, D. Forte, Y. Jin, R. Karri, S. Bhunia, and M. Tehranipoor. Hardware Trojans: Lessons learned after one decade of research. *Trans. Des. Autom. Elec. Sys.*, 22(1), 2016.

[XT13] Kan Xiao and Mohammed Tehranipoor. BISA: Built-in self-authentication for preventing hardware Trojan insertion. In *Proc. Int. Symp. Hardw.-Orient. Sec. Trust*, pages 45–50, 2013.

[YHD+16] K. Yang, M. Hicks, Q. Dong, T. Austin, and D. Sylvester. A2: Analog malicious hardware. In *Proc. Symp. Sec. Priv.*, pages 18–37, 2016.

[YSN+17] Muhammad Yasin, Abhrajit Sengupta, Mohammed Thari Nabeel, Mohammed Ashraf, Jeyavijayan (JV) Rajendran, and Ozgur Sinanoglu. Provably-secure logic locking: From theory to practice. In *Proc. Comp. Comm. Sec.*, pages 1601–1618, 2017.

[ZT11] Xuehui Zhang and Mohammad Tehranipoor. Case study: Detecting hardware Trojans in third-party digital IP cores. In *Proc. Int. Symp. Hardw.-Orient. Sec. Trust*, pages 67–70, 2011.

A Further Considerations

Practicality and Transfer of Findings. All provided findings cover real-world challenges for 7nm ICs. This is because all layouts are 1:1 representations of potentially real ICs. Thus, while actual IC manufacturing and post-silicon detection are out of scope—due to some technical as well as legal reasons—all the attacks, defenses, and conclusions are valid for such end-to-end assessment.

The presented techniques and addressed challenges are applicable to any IC design, not just the representative benchmarks used in this study. This is because the real-world IC tooling we use is agnostic to design functionality. Furthermore, for a given technology node, any design must use the same standard-cell library and design rules. Our release, especially the benchmarking framework, allows the community to target designs beyond our investigation, e.g., fostering further industry-driven case studies.

Scope in General. First, this effort meticulously examines and fixes vulnerabilities arising from the real-world complexities of modern IC layouts, especially the interplay of placement, routing, timing, manufacturability, etc. This also ensures practical basis and relevance for all assumptions and constraints of the threat model. Second, the presented first-of-its-kind framework, standardized benchmarks, metrics, and unified scoring guaranteed a systematic evaluation of all the participants' efforts. Third, while prior red-versus-blue HT competitions exist (Sec. 3), none match our effort in scope, detail, practical relevance, and open-sourcing of all artifacts and methods to the community. Our

effort is also complementary to prior competitions, as it is focused on the final frontier for assessment of HT threats: actual IC layouts.

Scope and Novelty of Techniques. All attack and defense efforts provide state-of-the-art techniques, developed over this months-long effort. Some techniques boast preliminary versions published in top-tier venues and practical validation. For example, some HTs that are targeting on AES are similar to those shown in Fig. 1, with the latter conclusively proven through a real IC tape-out [PP22].

Teams were free to replicate prior art, independently devise similar strategies, and contribute entirely novel methods. This led to a diverse array of approaches, yielding innovative contributions. For example, the red team significantly advanced ECO-assisted HT insertion, pushing the boundaries of this complex challenge.

Recommendations for Defenders. As indicated, a crucial lesson learned is that defenders do face a "dynamic game of cat-and-mouse." The interplay between attacks and defenses is heavily influenced by the intricate layout-level nuances of the 7nm technology, ultimately determining the effectiveness of any effort.

We recommend that defenders evaluate any layout-level technique using two key metrics: (1) security gains and (2) implementation costs. Security gains can be quantified through the proposed metrics, including the degree of violations imposed on attackers. Costs can be assessed via standard power-performance-area (PPA) metrics, one-time R&D costs, and recurring application costs. More specifically, while recurring application costs are marginal, e.g., only tens of minutes for any post-defense assessment run using our framework, one-time R&D costs can be substantial. Here, 23 designers participated across all blue teams, investing an estimated 3,240 man-hours, and 3 further designers participated as the red team, spending an estimated 1,080 man-hours, all over several months. Our release of demonstrated attack and defense techniques, and the potential for outsourcing such efforts, can significantly reduce the R&D costs in practice.

We emphasize the role of red-versus-blue teaming during the design-time of any sensitive IC. That is essential as the real-world feasibility of imposing failures on attackers not only depends on the defenses but also on the attack techniques. The proposed framework, which has demonstrated significant advances over prior art, enables such integrated evaluations. The framework is fully automated and extensible, thus open to further development. In general, the release of all artifacts and methods shall be valuable to the community for real-world assessment and benchmarking of their own defense/attack efforts.

Aside from proactive design-time efforts like ours, we also emphasize the need for orthogonal, reactive post-design measures for a comprehensive defense strategy, e.g., chip-level [MMST23], layout-level [PMB+23], or netlist-level [HYT17, LAKS23] inspection/detection as well as testing [SCN+15, XT13, CWP+09, WSS13]. However, further advances are also needed for these approaches [GGPR22, JMHS14, DNCB10, LWU+19].

Experience, Potential Bias, and Constraints for Teams. All teams possessed relevant experience and proven track records for state-of-the-art R&D efforts for IC design, notably including the in-house, end-to-end design toward successful tape-out of several ICs [PP22, EPP23, AIRP22, YSN+17, NAC+19, LCP+21, SNA+22].

The organizers took proactive measures to prevent bias, including an open call for blue-team participation and maintaining the teams' anonymity throughout the competition.

The red team was not constrained while devising their Trojans; they only had to adhere to our real-world setup. This yielded a diverse set of 36 HTs implemented by them. Similarly, the blue teams were not constrained either.

Summary. This large-scale, first-of-its-kind community effort rigorously investigates fabrication-time HT insertion versus layout-level defenses, setting itself apart from prior art in terms of scope, detail, practical relevance, and an full open-source release. We meticulously examined real-world complexities of modern 7nm ICs and employed a strict red-versus-blue setting to ensure unbiased results. Practicality is a core focus, with all

findings directly applicable and transferable to other real-world IC designs, due to the use of industry-standard tools and a realistic threat model. Despite prohibiting trivial strategies based on spares and fillers, both the attack and defense teams demonstrated innovative techniques.

Key takeaways for hardware-security practitioners include:

- *Proactive, Layout-Level Defenses are Essential:* While reactive post-silicon measures are important, this effort highlights the value of proactive layout-level defenses as part of a comprehensive security strategy.
- *Continuous Red-versus-Blue Teaming is Crucial:* The dynamic interplay between attacks and defenses necessitates an ongoing, security-centric evaluation throughout the IC design process. Our framework enables such evaluation.
- *Assess Techniques Holistically:* Defenders must evaluate techniques based on both security gains and implementation costs (PPA, R&D). Our release of all devised techniques and the benchmarking framework aids in such efforts.

Beyond these takeaways, this effort underscores the practicality of sophisticated HT attacks at the layout level, even against advanced defenses. It reinforces the urgent need for continued research and collaboration in the hardware-security community.

B Contest Format

General Logistics. The contest was open to students of all levels and/or practitioners from industry. The organizers recruited the red team separately, asking for extensive experience in real-world IC design and manufacturing with an emphasis on hardware security in general and HTs in particular. Recall that there was no collaborative interaction between the red and blue teams in general and all blue teams remained anonymous to each other during the contest.

Timeline. The contest was publicly announced end of October 2022. Around mid-December 2022, a sample benchmark, SHA256, was released publicly as well, along with the *ASAP7* technology setup. The benchmarking framework was opened up at the same time, to registered teams. On February 1st 2023, the registration was closed; 14 blue teams from all over the world had registered.

The deadline for the qualifying round was mid-February 2023. To pass this round, blue teams had to submit at least one valid layout with at least some improvement in scores over the baseline layout, and that for all benchmarks. Only 4 teams passed this round; the other teams indicated various challenges, but mainly lack of manpower and/or bandwidth. This was somewhat expected, as the contest followed a modern and real-world setting, utilizing an advanced 7nm technology node and commercial tooling for IC design. Despite the fact that the reference flow for PD, the sample benchmark, and the benchmarking framework were all made available early on, this setting still required solid expertise up-front.

The deadline for the final ranking was the end of March 2023. The winning blue teams were announced shortly after. The community effort continued beyond the contest, up until November 2023, with a focus on further advancing the red team's capabilities (Sec. 9).

Contest Winners. Considering the overall scores for the final ranking (Fig. 5), the contest winners were determined by the organizers by (i) ranking the scores for each team separately for each benchmark, (ii) deriving the average ranks across all benchmarks, and (iii) ranking again across these benchmark-specific ranks. This was done for fairness: given the widely varying complexities of benchmark layouts (as in varying baseline utilization numbers, timing constraints, etc.; see Tab. 5, App. F), a simple approach of averaging across all scores would have been inappropriate.

With the outlined ranking strategy, the following places were awarded: 1st place for Team B, 2nd place for Team C, 3rd place for Team A, and 4th place for Team D.

Operational Logistics. The contest interaction was conducted fully online, as explained next. All registered blue teams had been provided access to a dedicated and private *Google Drive* folder, which was acting as web interface for both submission and retrieval of results (Fig. 2). Teams could upload layout files at any time, upon which the files were automatically downloaded and queued for batch processing (App. E.1). Once done, the following result artifacts were returned to the blue teams' respective Google Drive folders: the overall scores, all technical reports (including any warnings and/or violations induced by their defense efforts), and the HT-infested versions of their protected layouts (the latter only for successful runs for HT insertion).

Blue teams interacted via e-mail with the organizers. The organizers summarized and shared related Q&As to all blue teams through a one-way e-mailing list. Note that Q&As were limited to high-level understanding, logistics, and selected technical details, e.g., for the *ASAP7* technology setup and the reference flow for PD; questions on HTs and the red team's techniques were not allowed. Also note that the organizers shared best scores and intermediate rankings regularly with those registered blue teams that opted in.

C Release

We publicly release all artifacts and methods:

1. the benchmarks, i.e., representative IC layouts,
2. the devised HTs,
3. different sets of attack and defense techniques,
4. the evaluation and scoring techniques,
5. different sets of best results obtained during the contest and beyond,
6. the reference flow for PD,
7. the *ASAP7* technology setup, and
8. the fully-automated, extensible benchmarking framework.

This release significantly advances the state-of-the-art for related R&D efforts, and it enables the community to continue exploring this important challenge for hardware security.

All techniques, the reference flow for PD, and the framework are released at [https://github.com/DfX-NYUAD/Trojan-Insertion-versus-Layout-Defenses]. The benchmarks, the HTs, best results, and the technology setup are released at [https://drive.google.com/drive/folders/10GJ5hXOBQupwqv1WMtitarsEuEE_Y-vV?usp=sharing].

D Supplementary Results

D.1 Assessment of Red-versus-Blue Efforts

Another detailed view on the red-versus-blue teams' performance during the contest is provided in Fig. 10. Note the following for data organization. First, the early point in time refers to the same as in Fig. 5. Second, overall sums for outcomes differ across time. This is because, at the early point in time, not all blue teams managed yet to make a valid submission for each benchmark (Fig. 5). Third, outcomes are combined across all benchmarks, but gathered separately for each of the six HTs considered per benchmark. In other words, the data shown is "raw," as the related scores are not averaged yet across the multiple HT runs. Fourth, unlike with the scoring procedure, any other, less severe violations occurring at the same time are reported as well. For example, there are cases with DRC violations that also exhibit some timing violations. Overall, this arrangement serves well to study the prospects of the different attack versus defense efforts.

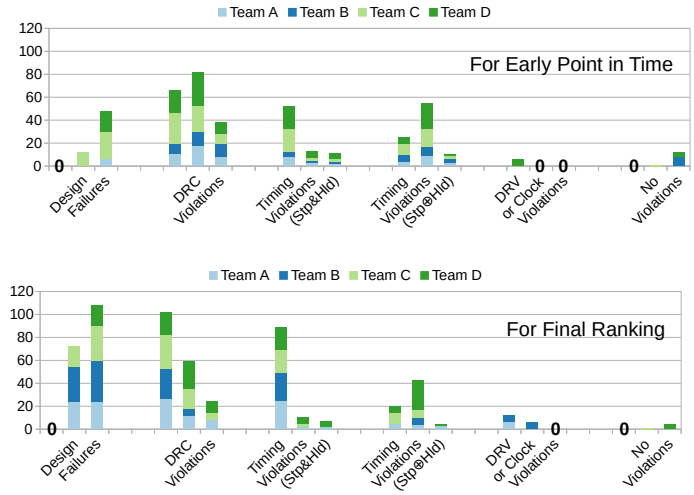


Figure 10: Stacked histograms of outcomes for the advanced HT insertion during the contest, for different points in time. Note that outcome categories are worded from the attackers' perspective, and that each category has three bars: from left to right, these correspond to the aggressive, moderate, and conservative insertion modes, respectively. Further note that bars/modes with zero occurrences are represented by *0* labels. Finally, *Stp&Hld* is short for setup `AND` hold, whereas *Stp⊕Hld* is short for setup `XOR` hold.

For interpretation of this figure, consider the following. First, the majority of outcomes fall into the categories of DRC violations or design failures. That is, the blue teams' defense efforts hindered the red team's attack trials considerably. Second, the blue teams also improved their defense efforts during the contest. This can be seen from the fact that outcomes are shifted further into the more challenging regime for the red team for the final ranking, i.e., (i) more design failures in general, (ii) more often design failures than DRC failures for both the conservative and moderate modes for HT insertion in particular, and (iii) more cases for joint setup and hold timing violations. Third, most important from the red team's perspective is the fact that there is not a single design failure for the aggressive mode. Thus, this mode enabled partially successful HT insertion, i.e., with DRC violations or better outcomes, across all the different HTs, benchmarks, and blue teams' defense efforts.

D.2 Extended Techniques for Trojan Insertion

See Fig. 11 for detailed plots of DRC violation counts across all 36 HTs, for all teams and for both attack settings.

For cases where DRC violations rather increase while using the extended techniques, note the following. Since the extended techniques encompass the contest techniques, yet in a more comprehensive manner for orchestration of attack steps, this can only be explained by differences in technical details. More specifically, for the contest techniques, ECO PnR was configured to consider all PDN routing shapes as obstacles. The red team initially identified this measure as helpful for limiting DRC violations, given that violations can quickly arise around those regions (Fig. 3). However, this setting also limits the built-in optimization algorithms for ECO PnR. Especially for the highly-utilized layouts achieved by blue teams for the final ranking of the contest, the red team found that, on average, this setting became counterproductive again. Thus, for the extended techniques, this setting was deprecated. Still, as seen here, this setting remains beneficial in specific cases.

D.3 Manual Fixing of Violations

For fixing of DRC violations, the following procedure is devised. First, inspect the types and locations of violations. Second, shift cells that are directly or indirectly impacted (as in placed nearby) into adjacent regions with lower placement and/or routing utilization. Third, run ECO refine/legalize placement and ECO route.

An example of successful fixing of DRC violations is illustrated in Fig. 12. Note that all violations could be resolved after a 3rd round of fixing (not illustrated).

For fixing of timing and DRV violations, the following procedure is devised. First, inspect violating timing paths. Second, revise paths with setup and/or corresponding DRV violations by (a) inserting or replacing repeaters such that wire delays are reduced, (b) increasing driver strengths or selecting lower-threshold-voltage cells such that cell delays are reduced, and/or (c) shifting of cells such that wire delays are more balanced and reduced. Third, revise paths with hold and/or corresponding DRV violations by

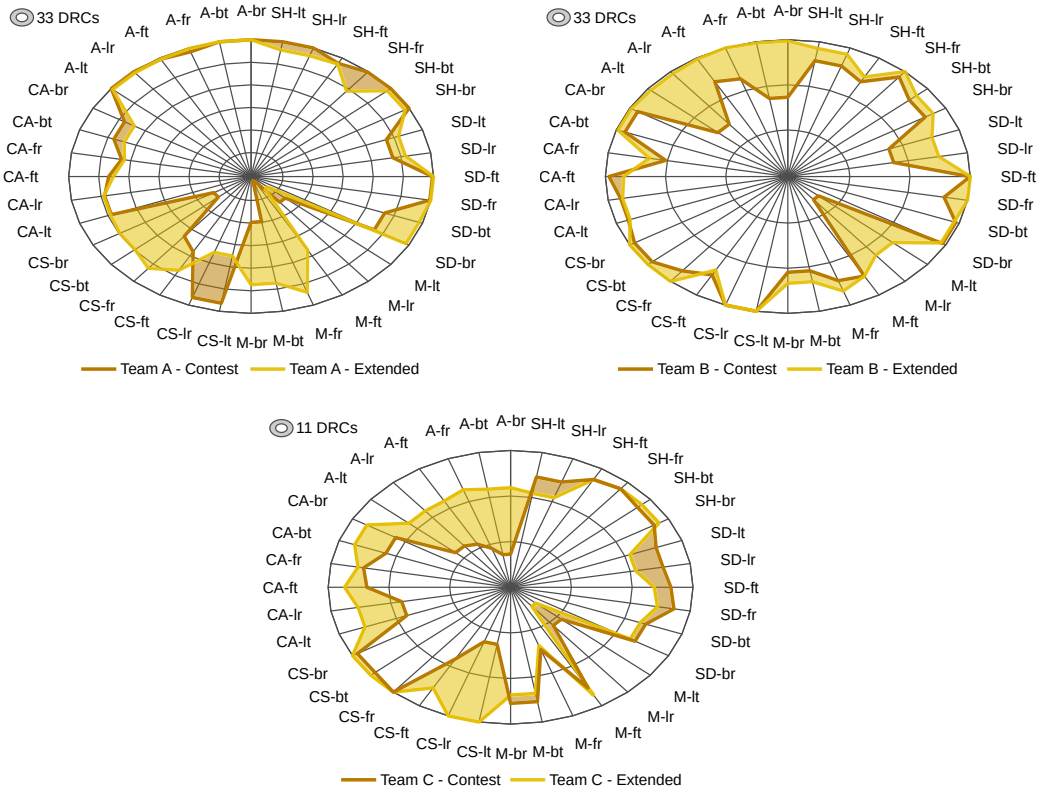


Figure 11: DRC violations incurred by the red team for the two different attack settings. Violations are shown separately for all 36 HTs while attacking the top-three blue teams' protected layouts as obtained from the final ranking. Note that violation counts increase toward the plots' center points; zero violations are represented by the outermost ellipse. Thus, cases where the extended techniques outperform the contest techniques—as in achieving fewer violations—are highlighted in light-yellow-filled regions and vice versa. Also note that the scales for violation counts between two ellipses are shown in the upper-left corners. For the corresponding HT labels at the periphery of the plots, *A-** is short for the benchmark AES128, *CA-** for CAMELLIA, *CS-** for CAST, *M-** for MISTY, *SD-** for SEED, and *SH-** for SHA256, respectively, while *br* is short for the corresponding, benchmark-specific version of the HT *burn-random*, *bt* for *burn-targeted*, *fr* for *fault-random*, *ft* for *fault-targeted*, *lr* for *leak-random*, and *lt* for *leak-targeted*, respectively.

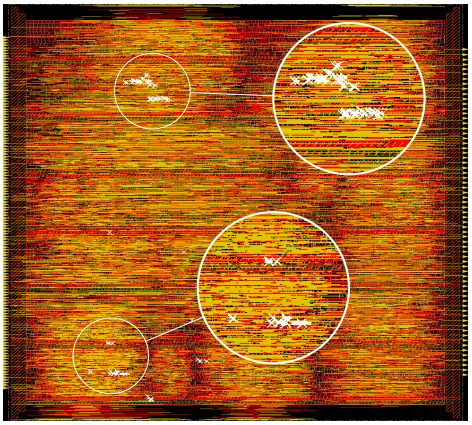
(a) inserting repeaters to the data path such that overall delays are increased, while also cross-checking that setup violations in other related paths are avoided, and/or (b) decreasing driver strengths or selecting higher-threshold-voltage cells such that cell delays are increased. Fourth, run ECO refine/legalize placement and ECO route.

Since fixing of violations typically incurs an iterative process, i.e., fixing some violations in one place / of one type might lead to new (but hopefully fewer) violations in another place, all procedures are repeated as needed but also stopped once not further converging.

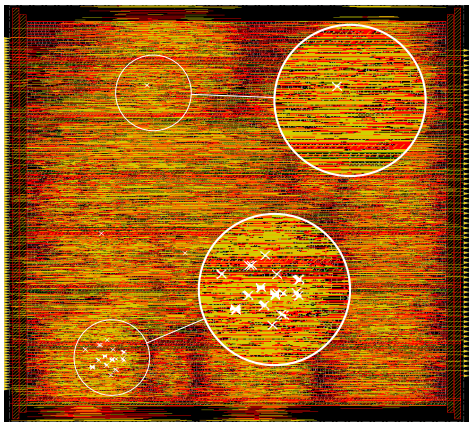
E Further Details for Benchmarking Framework

E.1 Automated, Extensible Back-End Pipeline

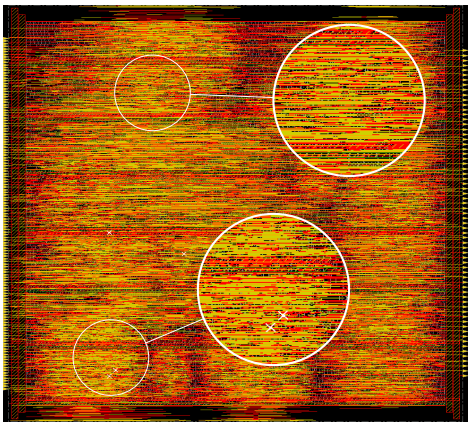
The back-end pipeline is an essential part of the benchmarking framework; it covers all post-defense tasks. Thus, its role is to orchestrate all the different procedures for constraints



(a) Before fixing.



(b) After 1st round of fixing.



(c) After 2nd round of fixing.

Figure 12: Top view on routing for benchmark MISTY, obtained from *Cadence Innovus*. The layout is protected by the winning blue Team B and under attack by the red team (aggressive mode), as obtained from the final ranking. The HT is *MISTY-leak-targeted*. Shown are all metal layers affected by DRC violations: M1–M6, except M4. Aside from routing (multiple colors), violations are pointed out by white *X* labels. Note the changes in violations for the exemplarily highlighted regions.

checking, advanced HT insertion, and evaluation, along with other important tasks such as submissions download and results upload, process monitoring, and logging.

The back-end itself is implemented in *bash* scripts, whereas all methods of the benchmarking framework are implemented in *tcl* scripts and *C++* code, tailored for interaction with the different commercial IC-design tools. The back-end is implemented as daemon, i.e., as a set of fully automated background procedures. All procedures are modularized, thereby enabling further extensions. The back-end is fully parallelized: the daemon itself supports parallel processing of multiple submissions, the daemon also manages multiple independent calls to the commercial tools as needed, and all benchmarking methods and procedures are devised for multi-threading. Such an implementation was important to maintain short turn-around times for all blue teams, especially around deadlines.

The back-end workflow is outlined next. Note that Steps 2)–4) are repeated periodically in that order.

1) Initialize. Global runtime variables are set, e.g., direct-access URLs for all the teams' *Google Drive* folders. (Recall that the web interface, based on *Google Drive*, was introduced in App. B.) The operation mode, "testing" or "production," is set. Note that the whole daemon can be started twice (or more), thereby providing a stable production instance that is fully independent from testing/debugging instances. Local work and backup folders are initialized for all teams, also accounting for the operation mode.

2) Submissions Download. All new submissions, if any, are downloaded and queued for processing. Before passing submissions on to actual processing, the queue procedure checks both the overall workload of the back-end server as well as how many runs are currently ongoing for the respective blue team(s). For fairness, all teams are allowed the same upper limits for parallel processing.

3) Actual Processing. Once some submissions pass the queue, actual processing is conducted: constraints checking, pre-attack evaluation, advanced HT insertion, post-attack evaluation, and scoring (in that order). At the same time when processing is started, e-mail notifications are sent out to the respective blue team(s).

4) Results Upload. Once processing is done, all result data (App. B) are uploaded. Follow-up e-mail notifications are sent out, and all processed files are backed up.

E.2 Technology and Standard-Cell Library

Without loss of generality, the organizers selected the 7.5-track version of the *ASAP7* standard-cell library [VVC17]. As indicated, a few modifications were made to the PDK by the organizers to ensure that participants could use different PD tools and versions with ease. These modifications are outlined next; see the so-called technology LEF file provided in the release (App. C) for full details.

Some complex via rules were dropped, while maintaining the major features of the technology. In tandem, new design rules had been added; this was done to create interesting, more realistic, and challenging scenarios for the participants to work with. The most significant addition is the notion of colored metals, i.e., metal layers that would be fabricated using more than one lithography mask [KY13]. This departure from the original setup of the *ASAP7* PDK introduces some challenges that are common in the first generation of FinFET technologies (e.g., TSMC 16nm, GF 14nm, or Intel 22nm). Further, maximum density rules for all metal layers have been introduced, to prevent participants from adopting trivial metal-filling solutions to protect their layouts against HT routing.

E.3 Reference Flow for Physical Design

Power Planning. The *ASAP7* PDK and library are rather restrictive regarding how power stripes can be implemented; there are only a few combinations of metal layers, width, spacing, and offset that can generate a coherent power network with adequate via

arrays. Taking this into account, the core rings are specified to be routed using the metal layers M6 and M7. For the standard-cell rails, the follow-pins appear in both metal layers M1 and M2 in what is called *stapled style*. Finally, the vertical and horizontal stripes are specified to be routed in the metal layers M3 and M4, respectively.

Compatibility. The organizers have evaluated this flow for different versions of *Cadence Innovus*, namely for 17.10, 18.10, 19.11, 20.11, 21.11, and 21.13. They found that results can vary somewhat across these versions. Thus, the flow has been tuned accordingly, providing the participants with a robust setup that requires as little ramp-up as possible and is largely independent of the version available at their ends. In any case, for consistency in the official evaluation, the organizers have used version 21.13 throughout the contest. This fact was communicated to all teams early on.

E.4 Evaluation

Note that all evaluation steps start from "bare" layout files. More specifically, the same initialization steps as for advanced HT insertion (Sec. 5.4) are also conducted here.

Thus, the organizers did not accept any design databases as submission, on purpose, for the following reasons. First, this is dictated by the threat model (Sec. 4): only layout files are accessible to the red team. Again, this aligns with the real-world setting for commercial IC manufacturing where original databases from design houses are not available to foundry-based attackers. (However, similar databases can be recreated by attackers, as needed for technical reasons for the advanced HT insertion, Sec. 5.4.) Second, this approach is also important to prevent any "hacking" attempts by blue teams to undermine the rigorous evaluation setting, including timing constraints, pre-arranged standard-cell library files, etc. Also note that such "hacking" is not applicable in practice; in a regular real-world setting, i.e., outside of such a competition, defenders are bound to the original technology setup and constraints in any case.

Constraints Checking. For a submission to be considered valid, various constraints must be met. This is to enable a realistic and competitive, yet fair, effort. The complete list is given next.

- Layouts must fully comply with the provided *ASAP7* technology library setup. Among other aspects, this means that layouts cannot incorporate custom cells and cannot revise the metal stack.
- Layouts cannot incorporate trivial defenses. Specifically, any spare cells, metal fillers, and physical fillers are prohibited.
- Layouts must meet setup and hold timing checks using the provided, so-called SDC files for timing analysis.
- Layouts must have zero DRC violations.
- Layouts must maintain functional equivalence of the underlying design.
- Layouts must maintain all cell assets declared along with each benchmark. However, blue teams are free to revise the physical implementation of assets, e.g., revise the cell-type or the location of assets.
- Layouts must include a fully functional clock tree, but blue teams are free to revise its implementation.
- Layouts must comply with the PDN recipe provided in the reference flow. The PDN's routing shapes are checked for dimensions, area, and locations.

- Layouts must maintain the arrangement of primary input/output (IO) pins. That is, IO pins must remain placed on the left/right side, as assigned in the baseline layout.

- Layouts must stay within a margin of +10 reported issues for all additional design checks. See the release (App. C) for technical details on these checks.

Constraint checks are implemented wherever suitable. For example, timing checks are integrated with design-quality evaluation, functional equivalence is checked separately (using *Cadence Conformal*), compliance with the technology setup is covered by checks for related errors during initialization (using *Cadence Innovus*), etc.

Pre-Attack Evaluation: Security Risks. The evaluation of exploitable regions is implemented as follows. First, the row-based placement information is exported from *Cadence Innovus* using *tcl* scripting. Second, switching over to more effective and efficient *C++* coding, an adjacency list of all rows with their sites and occupancy is built up. Third, from that list, exploitable regions are derived for each row. Fourth, regions are iteratively merged across adjacent rows. Note that regions extend across as many free/open sites as nearby available (in a transitive manner), into any polygonal shape. However, single open sites acting as "bridges" are rejected; in other words, two adjacent regions that would be connected only by one single site are kept separate.

For scoring, weighted metrics for sites across all the exploitable regions are computed: the sum of sites (weighted 1/2), the maximum of sites (weighted 1/3), and the median of sites (weighted 1/6). The metrics and weights are devised based on the following findings.

Exploratory experiments showed that, without any defense technique in place, layouts typically exhibit (i) few very large but localized exploitable regions and (ii) many smaller regions spread throughout. An attacker targeting on specific cell assets would be more interested in exploiting close-nearby regions (of some minimum size), and not necessarily in very large, possibly far-away regions. Now, unlike the mean, the median is not skewed by outliers introduced by such very large regions. Still, for evaluating a layout's overall resilience against HT insertion in general, the sum and maximum count of sites across all regions are considered even more relevant, hence the higher weights for these metrics.

Post-Attack Evaluation. Note the following aspects for the score-sheet defined in Sec. 5.5. First, the score-sheet is worded from the attacker's perspective, but applies to both attackers and defenders. Second, lower scores mean that the red team faces more difficulties with HT insertion which also means that the blue team's defense is more effective, and vice versa for higher scores. Third, the gap of 3 points between violation categories is on purpose: for attackers it is more important that an HT has, e.g., no DRC violations at all, versus what effort is required to reach that outcome. Thus, points for different insertion modes within the same category of violations are closer/more similar than points across categories.

For an example for the score-sheet, assume that some protected layout has, during insertion of some HT A, incurred (i) design failures for the conservative insertion mode, (ii) DRC violations for the moderate mode, and (iii) no violations for the aggressive mode, respectively. The corresponding individual scores are 2, 6, and 25 points, which are averaged to 11 points for the overall efforts to insert HT A. Further assume that the same protected layout has, during insertion of another HT B, incurred setup timing violations for the conservative mode, and no violations for the moderate mode. Note that, due to the latter violations-free outcome, the aggressive mode is said to be not relevant here and is skipped. The individual scores are 17 and 26 points then, which are averaged to 21.5 points. The overall score for insertion of HTs A and B would be $(11 + 21.5)/2 = 16.25$. However, note that this is still only a partial example—the true overall score for the post-attack evaluation is obtained by averaging across all relevant HT insertion runs for all the 6 HTs corresponding to the design/benchmark under attack.

Scoring. All evaluation metrics are normalized as follows.

Table 5: Selected Details for Benchmark Layouts

	Dimensions (µm)	# Cells	Util. (%)	# Cell Assets	CP: (ns)	WNS (ps): Setup / Hold	# Sites Across ERs: Sum / Max / Med	Routing Tracks: F. (%) / T.(#)
AES128	822.44 × 822.44	263,618	67.34	384	0.17	34.29 / 33.25	662,065 / 460,741 / 29	63.89 / 29,331
CAMELLIA	158.24 × 158.24	10,101	84.02	396	0.27	22.23 / 20.82	8,820 / 1,403 / 46	58.76 / 5,634
CAST	293.24 × 293.24	24,450	52.52	143	0.66	25.49 / 18.22	147,359 / 139,439 / 30	62.44 / 10,452
MISTY	174.44 × 174.44	10,558	68.77	332	1.85	05.00 / 20.97	26,039 / 5,932 / 33	65.54 / 6,215
SEED	206.84 × 206.84	17,334	76.65	127	1.02	34.22 / 29.35	25,478 / 3,854 / 33	65.77 / 7,369
SHA256	190.64 × 190.64	9,708	71.09	125	0.60	20.80 / 22.63	25,964 / 2,133 / 25	68.26 / 6,792

Util. is short for layout/placement utilization; *CP* is short for clock period, i.e., the global timing constraint; *ERs* is short for exploitable regions; *Med* is short for median; *F.* is short for free; *T.* is short for total.

- Points for the assessment of red-versus-blue efforts are normalized to the defenders' worst-case scenario, i.e., no violations for the conservative mode, equating to 27 points. Recall Footnote 13 for an explanation.
- All other metrics m are normalized to their respective baseline, i.e., the nominal values obtained for the benchmark layouts. For all metrics except timing, normalization is applied as $submission\_m/baseline\_m$.
- Free routing tracks are first normalized over total available tracks, and then normalized as above.
- Given that timing violations are ruled out as a hard constraint, only positive values can arise for WNS components. Hence, these are normalized as $baseline\_WNS/submission\_WNS$.

Recall that, after normalization of metrics and points, final scores are computed as:

$$score = (1/2 \times security\_risks) + (1/2 \times design\_overheads)$$

More specifically,

$$\begin{aligned} score = {} & (1/2 \times (1/3 \times pre\_attack + 2/3 \times post\_attack)) \\ & + (1/2 \times (1/3 \times power + 1/3 \times timing + 1/3 \times area)) \end{aligned} \tag{2}$$

where

$$timing = 1/2 \times WNS\_setup + 1/2 \times WNS\_hold \tag{3}$$

and

$$pre\_attack = 1/2 \times ERs + 1/2 \times FRTs \tag{4}$$

with ERs being short for exploitable regions and $FRTs$ for free routing tracks. Further,

$$ERs = 1/2 \times \sum_{ERs} sites + 1/3 \times max_{ERs}(sites) + 1/6 \times med_{ERs}(sites) \tag{5}$$

and

$$post\_attack = avg_{HTs}(avg_{relevant\_insertion\_runs}(points)) \tag{6}$$

Also recall that lower final scores translate to better rankings for blue teams.

F Further Details for Benchmarks and Hardware Trojans

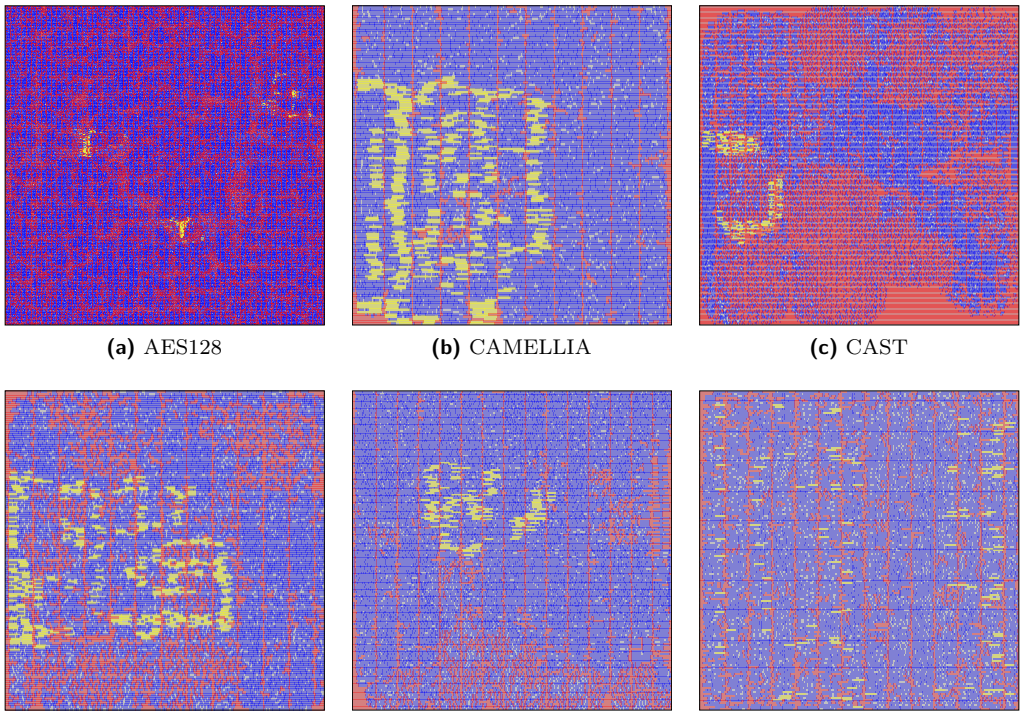
Benchmarks. As indicated, the process for logical and physical synthesis of benchmarks is conducted by the organizers, following the reference flow for PD (Sec. 5.2). For each benchmark, specific timing constraints and floorplan sizes are configured. These parameters are determined such that the resulting layouts maintain varying margins of layout resources, thereby imposing a varying degree of challenges, as shown in Tab. 5.

Fig. 13 provides layout illustrations for all benchmarks. Note how exploitable regions are often found around the vertically arranged power stripes (not illustrated as such).

This is because the layout regions around these stripes are particularly difficult to utilize without inducing DRC violations; also recall Fig. 3. Importantly, this challenge applies equally to defenders and attackers.

Hardware Trojans. Recall that the devised HTs cover three representative and realistic attack scenarios: (1) leak sensitive information, (2) induce faults, and (3) burn power. For each of the three scenarios, HTs are implemented in two versions: *random* versus *targeted*. (Thus, with 3×2 HT scenarios and versions per benchmark, there are 36 HTs in total.) While the random version connects trigger and payload components to randomly selected cell assets, the targeted version connects to assets that are placed close to each other in the benchmarks' baseline layouts. Also recall that cell assets must be maintained by blue teams (Sec. 5.5); thus, any asset that HTs seek to connect to is guaranteed to be present. These different versions are motivated by the hypothesis that targeted HTs would be easier to insert for attackers. On average, this holds true indeed; see Fig. 11 as well as the release of best results (App. C) for more insights.

In general, HTs are implemented (i) for varying sizes of trigger and payload signals, (ii) along with clock-divider logic, which is devised as needed for timing closure, and (iii) through 25–46 additional cell instances in total, including clock-divider logic. Note that these few instances represent just $\approx$ 0.01–0.47% of the benchmarks' baseline cell counts—this clearly shows that HTs can be extremely small, reemphasizing the fundamental challenge of fully protecting IC layouts against HTs.



(a) AES128 **(b)** CAMELLIA **(c)** CAST

(d) MISTY **(e)** SEED **(f)** SHA256

Figure 13: Layout illustrations for all benchmarks. Shown are regular cell instances in blue, cell assets in dark-yellow, exploitable regions in red, and remaining open sites in grey, respectively. Different dimensions result in varying levels of detail visible in these fixed-size plots.