Security Closure of Physical Layouts

ICCAD Special Session Paper

Johann Knechtel
NYU Abu Dhabi
johann@nyu.edu

Jayanth Gopinath
NYU
jg6476@nyu.edu

Jitendra Bhandari
NYU
jb7410@nyu.edu

Mohammed Ashraf
NYU Abu Dhabi
ma199@nyu.edu

Hussam Amrouch
University of Stuttgart
amrouch@iti.uni-stuttgart.de

Shekhar Borkar
Qualcomm
shekhar.y.borkar@gmail.com

Sung-Kyu Lim
Georgia Tech
limsk@ece.gatech.edu

Ozgur Sinanoglu
NYU Abu Dhabi
ozgursin@nyu.edu

Ramesh Karri
NYU
rkarri@nyu.edu

***Abstract*—Computer-aided design (CAD) tools traditionally optimize for power, performance, and area (PPA). However, given a vast number of hardware security threats, we call for *secure-by-design* CAD flows, to adopt principles of secure hardware design and streamline *security closure* throughout the flow. The stakes are high for integrated circuit (IC) vendors and design companies, as security risks that are not addressed during design will inevitably be exploited in the field, where vulnerabilities are almost impossible to fix. This paper highlights the need for *security closure of physical layouts* because efforts taken toward securing ICs at higher abstraction layers may be futile without support for securing the tape-out ready layouts.**

***Index Terms*—Hardware Security, Security Closure, Physical Layouts, CAD, Advanced Nodes**

I. Introduction

An ever-growing body of security threats can compromise information systems in general and ICs in particular (Fig. 1). The three pillars of security—confidentiality, integrity, and availability of data and assets—must be considered during design and manufacturing of ICs. Importantly, this must also be done in ways that the desired resilience is maintained throughout the use of ICs in the field [1]. Such *secure-by-design* efforts are challenging for four reasons:

1) ICs are subject to threats throughout their life-cycle, from design to manufacturing, to packaging/testing, to deployment, and to use in the field (Fig. 1). The fact that most stages of the IC supply-chain are outsourced to third parties across the world exacerbates these threats.
2) Unlike software, ICs cannot be "patched" in the field. Vulnerabilities overlooked or even introduced during design cannot be fixed after manufacturing. Hence, secure handling of sensitive data cannot be guaranteed anymore, rendering critical ICs even unusable altogether.
3) Traditionally, the cryptography community led the research and development of secure ICs. Still, security-focused IC engineering is essential, given that any security scheme may be undermined by physical attacks. Such attacks are highly effective and sometimes simple to launch, e.g., see [2]–[5].
4) Commercial CAD tools and best practices do not consider security. While ad-hoc efforts address some threats, holistic notions of *secure-by-design* and *security closure of physical layouts* are missing. Security-focused IC engineering is timely, especially for the CAD community [1].

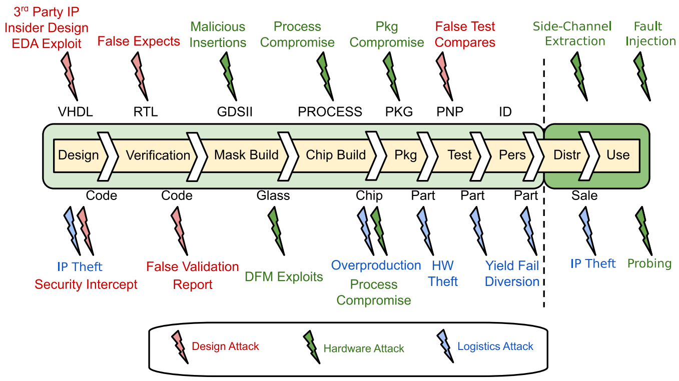


Fig. 1. Security threats throughout the life-cycle of ICs. Adopted from [6].

This paper presents a vision for *security closure of physical layouts*, i.e., assessment and defense measures tailored for physical layouts and integrated into CAD flows. Thus, the scope of threats are physical attacks. Without loss of generality, we focus on sensitive assets and data processed within ICs, not the intellectual property of IC design.

This paper is part of the *Security Closure of Physical Layouts* special session at ICCAD'21.[1] The structure and contributions of this paper are as follows. First, we introduce and motivate security closure of physical layouts in Sec. II. Second, we review physical attacks in Sec. III. Third, we frame security challenges and objectives for physical layouts, also for advanced technology nodes, in Sec. IV. Next, we elaborate how security can become a first-order consideration for CAD flows in Sec. V. As an example, we present *DEFense*, an extensible framework that uses commercial tools for establishing security closure at the DEF level. Finally, we provide concluding remarks in Sec. VI.

[1] We invite the reader to a companion paper, *Toward Security Closure in the Face of Reliability Effects* [7], found also in this session. It provides insights into reliability effects in interconnects and aging effects in transistors, and their role in security closure of physical layouts.

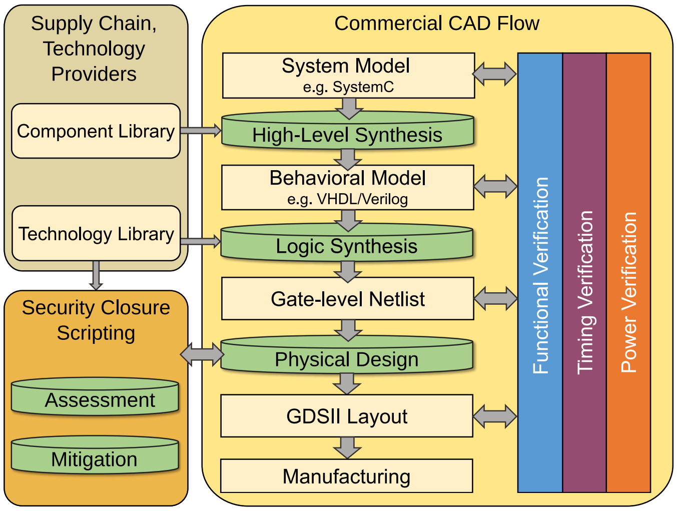


Fig. 2. Overview on CAD flow with means for security closure of physical layouts integrated through scripting. See also Fig. 5 and Fig. 7 for more details on security closure techniques proposed in this work.

II. Security Closure of Layouts: What and Why?

A. Terminology

Secure-by-design is an emerging CAD paradigm to tackle 1) top-down propagation and translation of security requirements and specifications and 2) bottom-up verification and validation of defenses against attacker's technical capabilities and limitations [1]. This must be realized throughout the CAD flow, starting from the specification and behavioral design, all the way down to physical layouts, along with feedback loops for the different CAD stages. *Security closure of physical layouts* assesses and hardens the physical layout against layout-level threats. Such security closure has to apply on physical layouts, with millions of polygons. Besides, security closure of physical layouts will need extensible and scalable techniques for layout verification, validation, and ECO modifications.

For secure-by-design CAD flows in general, important objectives are to 1) truthfully carry over security schemes introduced in earlier design stages without having them "optimized out" by CAD tools, and 2) enable independent verification and validation (IV&V) of physical layouts obtained from third-party design houses.

In short, secure-by-design is an overarching paradigm for security-centric CAD and security closure of physical layouts is the specific paradigm for the sign-off stage. The latter is essential since vulnerabilities introduced or missed during sign-off cannot be fixed later. See Fig. 2 for an overview on security closure means integrated in CAD flows.

B. Motivation

Security closure of physical layouts is essential for (at least) two reasons as follows:

1) Design efforts are more and more being outsourced to third parties. Aside from economic benefits (especially for small companies and government entities), this opens up threat vectors, as the outsourced design process can no longer be considered trustworthy.[2] In such scenarios, security-centric IV&V of the final layouts is not only prudent but essential.
2) One cannot assume that security measures taken at higher abstraction levels and earlier design stages will be correctly carried over into the final layouts. This is because many security schemes incur PPA overheads and may thus be "optimized out" of the layout. This is especially true in the absence of holistic, secure-by-design CAD flows. Even once such flows become available, IV&V for security closure of the final layouts is prudent.

Regarding 1), consider the following arguments. IC technology advances have slowed down; the gap between advanced and not-so-advanced IC technologies is narrowing and will diminish even further over the years. This fact is supporting a new IC ecosystem, providing low-cost design and fabrication for almost anyone, by offering design support, access to legacy technologies, and leverage of heterogeneous system-level integration. Although this shift in business models provides opportunity to new players, it will also pose several challenges in reliability and security of fabricated ICs. More specifically, although it is less of a concern today, the back-end design may become vulnerable to security threats in the future.

Today, the back-end design is performed by trusted employees in major design houses and, thus, threats of malicious tampering are limited and manageable—security does not represent a concern yet for major fab-less design houses and foundries. However, with the advent of inexpensive fabrication in the future, there will also be third-party services to make designs affordable. This is where potential breaches of security could go unnoticed; addition of unwanted spy-logic, or Trojans, in the back-end design becomes a major concern.

Security breaches in the front-end design can be detected by, e.g., register-transfer level (RTL) verification against tests or formal verification. In the back-end, however, breaches would go undetected, since there are little means to compare a final layout to the intended logic, unless it is reverse-engineered and meticulously compared with the original GDSII design file. Therefore, back-end design in the future will be much more vulnerable to security threats.

Regarding 2), consider the example of *private circuits* [9]. This scheme guarantees confidentiality in the face of side-channel attacks in a controlled and quantifiable manner. Without loss of generality, a bit a of sensitive data can be encoded as a vector (a_1, a_2, a_3), where $a = a_1 \oplus a_2 \oplus a_3$ and $\oplus$ denotes bitwise XOR. All operations can be implemented in encoded form while incorporating random bits. The security promise is that all components of one such vector are never processed at the same time. Thus, an adversary cannot learn the secret from power or other side-channels.

One security challenge for *private circuits* passing through regular CAD flows is as follows. The order of computation, as indicated by parentheses, is critical for *private circuits*

[2] The current assumption of trustworthy designers and CAD flows seems overly optimistic in any case [8].

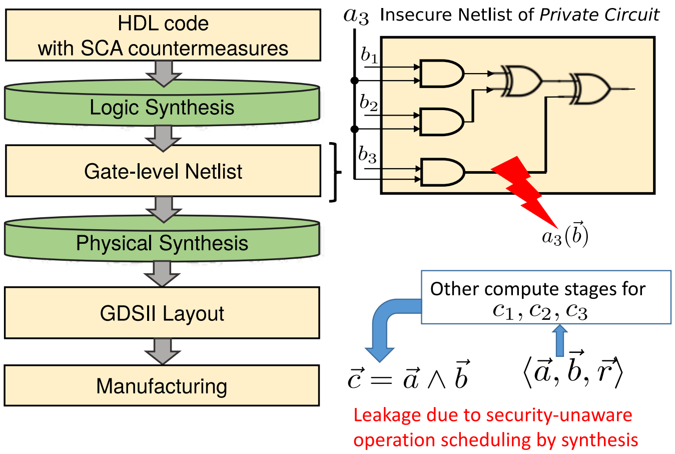

HDL code
with SCA countermeasures
Logic Synthesis
Gate-level Netlist
Physical Synthesis
GDSII Layout
Manufacturing
a_3 Insecure Netlist of *Private Circuit*
b_1
b_2
b_3
$a_3(\vec{b})$
Other compute stages for c_1, c_2, c_3
$\vec{c} = \vec{a} \wedge \vec{b}$
$\langle \vec{a}, \vec{b}, \vec{r} \rangle$
Leakage due to security-unaware operation scheduling by synthesis

Fig. 3. Motivational example for "security-blind" CAD flows. Here, the problem lies with logic synthesis, which optimizes out functionally redundant structures that are added-in by the *private circuits* scheme to prevent information leakage [9]. The resulting netlist cannot guarantee the security promise offered by *private circuits*. Adopted from [1].

to prevent information leakage. However, this order is irrelevant for correctness, as $\oplus$ is commutative. In Fig. 3, let us assume that logic synthesis implements c_3 for the AND operation such that the expression $a_3b_1 \oplus a_3b_2 \oplus a_3b_3 = a_3(b)$ is covered first and random bits r_{ij} are added later on—such an implementation will leak b. Regular, "security-blind" CAD tools can easily take such decisions, e.g., to improve timing, thereby undermining security promises. Furthermore, information leakage can occur for *private circuits* even when the netlist is synthesized in a security-aware manner, namely due to glitches [10]. Given that glitches manifest themselves directly at the level of physical layouts and transistors, such practical challenges for formally secure schemes are another motivation for security closure of physical layouts.[3]

III. Background on Physical Attacks

Physical attacks target the IC to probe, tamper, glitch, or even modify the underlying circuitry. Next, we briefly review such attacks. We outline prior, traditional art for countermeasures in Appendix A.

Side-channel attacks (SCAs) exploit information leakage that emanates from ICs in the field. SCAs monitor the operation of the circuitry to extract sensitive data. They exploit different channels, like power consumption [2], photon emission [13], timing behavior [14], and thermal emissions [15].

Fault-injection attacks (FIAs) induce faults mainly to aid subsequent attacks to infer sensitive data. That is, FIAs can be combined with SCAs [14] and/or statistical sampling [4]. FIAs can be direct, invasive fault injections (e.g., laser light [16] or electro-magnetic waves [17]) or indirect fault injections (e.g., repetitively writing to memory locations [18] or misusing frequency and voltage scaling features [19]).

Probing attacks extract data from standard cells or wires by probing the layout through the frontside metal stack or the backside substrate. Such attacks are enabled by different means, like direct micro-probing, electro-magnetic probing, and electro-optical probing [13], [20]–[22]. Probing attacks have their roots in failure analysis (FA), hence apply also in advanced nodes. For example, electro-optical probing has been demonstrated in 10nm nodes [23]. Some attacks like micro-probing require direct access and line of sight to the cell/wire of interest. Such attacks must be complemented by FA techniques like focused ion beam milling [24].

Hardware Trojans are malicious hardware modifications. Since IC supply-chains are largely outsourced, adversaries at various entities could introduce such Trojans [25], [26]. More specifically, Trojans can be introduced via untrustworthy third-party IP, by adversarial designers, during mask generation or manufacturing, or even during distribution or deployment of ICs. The notion of Trojans is versatile, covering malicious modifications that are: (i) targeting at the system level, RTL, gate/transistor level, and/or the physical level; (ii) seeking to leak information from an IC, reduce the IC's performance, or disrupt an IC's working altogether; (iii) are always on, triggered internally, or triggered externally [25].

IV. Security Challenges for Physical Layouts and Closure Strategies for CAD Flows

To make security closure of physical layouts a success, one has to consider a complex set of challenges, outlined next along with our envisioned strategies. We argue that tackling those challenges requires cross-disciplinary team efforts, driven by security researchers and practitioners, physical-design engineers, and CAD experts. Hence, we call the security, design, and CAD communities to such joint efforts.

A. Challenge: Limitations for Layout-Level Security Closure

Physical layouts are the final outcome of sophisticated CAD flows. Thus, they represent IC designs in their entirety and complexity, abstracted only by polygons. Now, security closure of physical layouts cannot mitigate all possible threats working only at this low level. In the same way as regular sign-off measures like ECO are limited in their efforts, security closure of physical layouts will also be limited—fixing for an architectural security flaw at the layout level is impractical.

Closure Strategy: Security closure of physical layouts will benefit from integration with secure-by-design CAD flows to truly harden IC layouts. For IV&V, stand-alone means for security closure can be used to assess the resilience of IC layouts, especially those procured from third parties.

B. Challenge: Complexities of Physical Layouts

As indicated, security closure of physical layouts would work directly on the layout level as needed. Accordingly, security closure of physical layouts has to comprehend various low-level circuit complexities, while handling millions or more of polygons. The complexities imposed by modern large-scale

[3] *Private circuits* has been extended to consider glitches in [11]. Still, this formalism cannot capture the continuous nature of physical information leakage and, thus, the guarantees it provides are only qualitative [12].

designs, advanced technology nodes, and physical effects[4] are considerable, especially given their ever-more tight interaction.

Closure Strategy: Security closure of physical layouts will benefit from scalable and extensible implementations. CAD flows enable such means by offering scripting interfaces. In addition to regular scripting, such interfaces also support system calls to custom, high-performance tools as needed. Whenever practical, security closure of physical layouts should use some abstractions, without sacrificing understanding of the relevant physical details. For example, security closure measures can be implemented by scripts that work on the DEF layout format and leverage integration with the regular CAD flow. We demonstrate such an approach in Sec. V-B.

C. Challenge: Lack of Holistic Layout-Level Approach

Various attacks exploit fallacies arising from physical layouts, as outlined in Sec. III and in [27], but there are only few efforts to harden the IC directly at the layout level. While there are some defenses that individually target physical-design stages, their composition may counteract security closure if not considered in a holistic manner [1].

Closure Strategy: Cross-disciplinary teams of security researchers and practitioners, physical-design engineers, and CAD experts should study how security threats can apply at the physical layout. Prior art that defends against particular threats at specific CAD stages are helpful starting points, but should be revisited toward holistic security closure.

D. Challenge: Security Metrics, A Different Kind of Metrics

CAD flows are driven by heuristics and metrics. However, there is neither an "all-in" heuristic nor a single metric to achieve security closure of physical layouts. Security metrics scale and behave differently than classic CAD metrics. For example, a transient fault in a critical component may be ignored during reliability verification, in case it occurs rarely in regular operation. However, when it comes to security closure, an attacker may put extra effort to inject exactly that fault. Thus security metrics are more like step functions, where certain effort must be taken to reach a particular security level. While spending less will break security guarantees, spending more may not provide extra benefits. This is fundamentally different from PPA metrics and has to be considered accordingly for security closure in secure-by-design CAD flows.

Closure Strategy: Metrics for security closure of physical layouts have to be revisited. While there are metrics for individual threats, tailored for particular design stages, it is important to study the interaction of metrics, to avoid counterproductive guidance in secure-by-design CAD flows. Also, heuristics and optimization strategies for security closure need to be devised. The emergence of machine learning (ML) for parameter tuning in CAD flows [28], [29] seems promising here as well. ML models could learn 1) the different scaling/behaviour of security metrics versus PPA and 2) strategies from training and evaluation in red team versus blue team.

E. Challenge: Advanced Nodes

As we continue to march into advanced nodes such as 5nm, 3nm, and below, challenges in securing physical design rapidly escalate. Some imminent ones include the following. Note that these challenges do not lend themselves directly to closure strategies, but are rather to be considered as what they are—challenges—for both sides, attackers and defenders.

1) Vanishingly Small Cell Size: In standard cells below 5nm, the number of fins in each transistor reached one. The transistors and their routing inside a cell are so tightly packed that there is no empty space for anything other than the logic. Moreover, the fins are becoming extremely small, thin, and short. This raises an interesting question on how applicable the probing mechanisms are to target these extreme geometries. Thus, how vulnerable are the cells to probing threats. Also, the whitespace available to insert Trojans becomes less. This means that, unless the size of Trojan circuits shrinks as well, it is not a stretch to assume that the opportunities for Trojan insertion diminishes in advanced nodes.

2) Super-Narrow Local Interconnects: Local interconnects are extremely narrow and tall in advanced nodes. This is to reduce the metal pitch while keeping resistance more manageable. Here, the same question arises as to how capable wire tapping and probing equipment today is to handle such small interconnects. On the other hand, the distance between the Trojan's trigger and payload reduces, possibly easing attacks.

3) Increasing Physical Design Complexity: A major driving force behind advanced technology development is the thirst for ever-more functionalities and capabilities from modern ICs. This puts tremendous pressure to physical-design tools, to deliver aggressive PPA that the customers need, while ensuring secure design, manufacturing, and operation. Thus, the PPA vs. security balancing act between these two often competing objectives becomes even more challenging in advanced nodes.

4) Reliability Effects: The reliable operation of ICs is subject to effects like electromigration, negative bias temperature instability, self-heating, etc. While these effects have been managed well for legacy nodes, this becomes a considerable challenge for advanced nodes as these nodes are pushing at the physical limits. Threats arising from subtle exploitation of reliability effects are largely overlooked so far, opening up a considerable attack surface. For more detailed security examples, see 1) Sec V-C3 for self-heating and 2) [7] for migration effects in interconnects and aging effects in transistors.

V. Discussion and Case Studies

A. Scanning and Defending Against Trojans in GDSII Layouts

IC Attack Surface (ICAS) is prior work for a Trojan-vulnerability assessment tool [30]. It proposes three metrics to measure the difficulty of inserting Trojans in a layout: 1) trigger space measures the amount of contiguous space available for Trojan insertion, 2) net blockage measures the routing blockages around security-critical nets, and 3) route distance measures how close the trigger spaces are from security-critical nets. ICAS proposes scripts to scan layouts in

[4] Among others, reliability effects are also directly exploitable [7].

GDSII format to evaluate these metrics. In terms of defense, ICAS tunes parameters for placement and routing tools (e.g., placement density, target clock frequency, and signal slew requirement). For example, to protect security-critical nets from probing, ICAS tightens the slew constraint so that all—not some—nets become shorter and create local routing congestion everywhere. Such routing is done in hopes of creating “probing obstacles” for security-critical nets. The ICAS study shows that: (i) trigger space reduces (= more secure) if the target placement density increases, (ii) net blockage increases (= more secure) if the slew constraint decreases, and (iii) route distance increases (= more secure) if the target placement density increases, or the slew constraint decreases.

We argue that ICAS is a good first step but has the following shortcomings that must be addressed to make such security closure means a part of daily routines for physical designers.

- PPA overhead of parameter tuning is not well demonstrated. Placement density, target frequency, and slew constraints will affect PPA in a complicated fashion.
- The impact of parameter tuning on routing congestion and coupling noise is not clear. Increasing placement density can inevitably worsen these two metrics.
- ICAS focuses on Trojan threats alone. More threats for physical layouts and metrics to evaluate defenses against those threats must be developed.

B. DEFense Framework

We propose an extensible layout-level CAD framework, called *DEFense*, to assess and mitigate layout-level threats throughout the physical-design flow. Recall Fig. 2 for an overview on such a CAD framework.

Unlike prior art, we consider security as key objective during design, not as an afterthought. We base our framework on commercial, scriptable CAD tools. This way, we aim for an secure-by-design CAD flow, as motivated earlier on.

We target at the DEF level, serving well for an integrated, automated, and easy-to-use approach. Given a designer’s input of security assets (e.g., key registers), one can assess vulnerabilities of layout resources (e.g., for Trojan insertion, placement and routing resources near those key registers). The assessment is visualized for designer feedback and quantified for guidance throughout the design flow. By guidance of our scripts, we utilize the CAD tools’ engines for joint consideration of security closure of physical layouts and PPA.

1) Scanning and Defending Against Trojans Throughout Physical Design: We developed scripts to scan and defend against additive Trojans. The scripts run after the different stages of the physical-design flow. Fig. 4 demonstrates the visualization of the scanning. After scanning for vulnerabilities, we apply defense scripts to reduce the sites where Trojan cells could be placed and routed. For example, the number of sites can be reduced by selectively increasing the density in the vulnerable regions; such defense can be integrated before and after the placement stage.

In general, the CAD optimizations running after and in-between the various levels of defense scripting help to minimize the PPA impact while maintaining security. The integration of scan and defense means is outlined in Fig. 5. Next, we provide more implementation details and discussion.

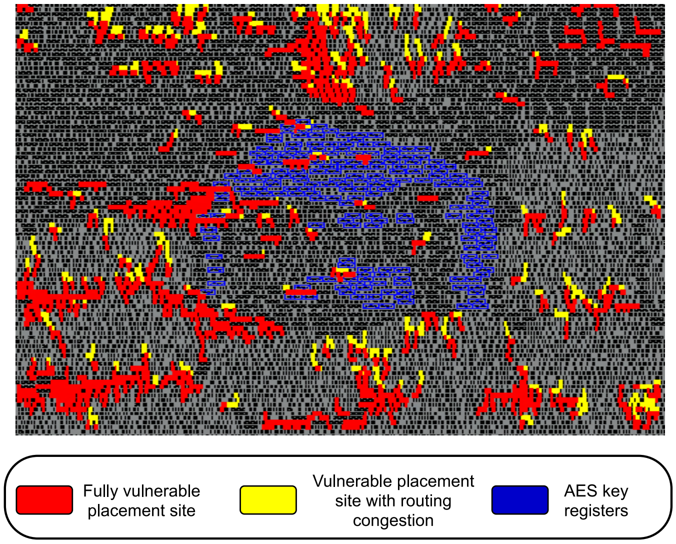

Fully vulnerable placement site
Vulnerable placement site with routing congestion
AES key registers

Fig. 4. Visualization of scanning for vulnerabilities to additive Trojans, for an AES design.

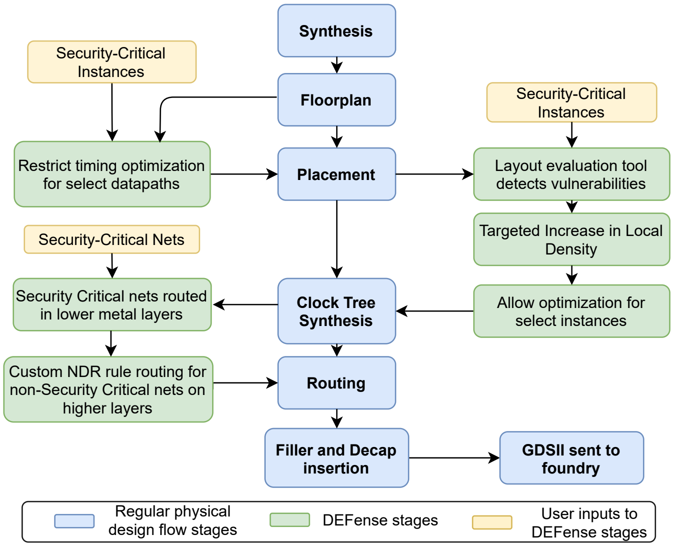

Synthesis
Security-Critical Instances
Floorplan
Security-Critical Instances
Restrict timing optimization for select datapaths
Placement
Layout evaluation tool detects vulnerabilities
Targeted Increase in Local Density
Security-Critical Nets
Security Critical nets routed in lower metal layers
Clock Tree Synthesis
Allow optimization for select instances
Custom NDR rule routing for non-Security Critical nets on higher layers
Routing
Filler and Decap insertion
GDSII sent to foundry
Regular physical design flow stages
DEFense stages
User inputs to DEFense stages

Fig. 5. Integration of layout-level security closure techniques into physical-design flow – protection against Trojan insertion.

a) Trojan-Resistant Placement: Instead of forcing the global placement density below a given target, we can manage whitespace and low-density regions selectively. A popular approach for placement is to remove overlaps during non-linear global placement. Here, we modify the placer by dividing the placement into tiles and assigning high target density values to tiles with security-critical components (Fig. 6). Moreover, this per-tile density assignment can be adapted as the global placement progresses. This will ensure that whitespaces and low-density regions are pushed closer to the areas of less sensitive assets, without hurting wirelength and PPA too much.

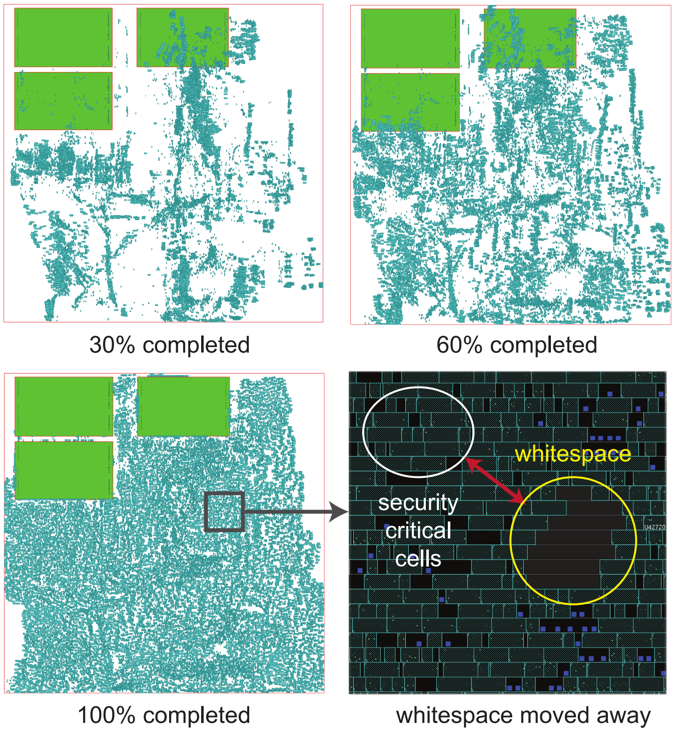


Fig. 6. Tile-based global placement with Trojan-resistant whitespace management.

b) Trojan-Resistant Routing: To hinder Trojans from tapping into/connecting with sensitive wires, instead of the "blind" approach by ICAS [30]—increase routing congestion everywhere—we propose to modify the router so that these wires are surrounded by short neighboring nets on all four sides. This can be done, e.g., as a post-route step, where we 1) rip up neighboring nets and 2) re-route them around the security-critical net by using routing fences or confinements. Such an approach will reduce undesirable routing congestion and coupling noise on timing-critical nets. Another strategy is to impose long distances between whitespaces and nearby routes, thereby improving the route distance metric by ICAS.

c) Trojan-Resistant Timing Closure: Timing closure is a critical task during physical design. Buffering and gate sizing—the two major operations conducted during timing closure—are done at multiple stages of physical design, including post placement, post clock-routing, and post signal-routing. While PPA should remain the top priority for optimizing sizes and locations of buffers and gates, some flexibility can be easily granted to co-optimize for PPA and security. For example, there might be multiple candidate sizes and locations for buffer insertion for a given net that result in the same worst negative slack. Then, one can prefer candidates that also optimize security metrics: reducing space for Trojan insertion, minimizing tapping, and increasing the distance between possible trigger and payload regions.

2) Scanning and Defending Against Frontside Probing Attacks: We observe that prior solutions such as layout filling, net shielding, or self-testing filler logic come with considerable PPA cost. Also, implementing such solutions while having to avoid violations for sign-off metrics seems time-consuming, thus potentially delaying tight tape-out schedules.

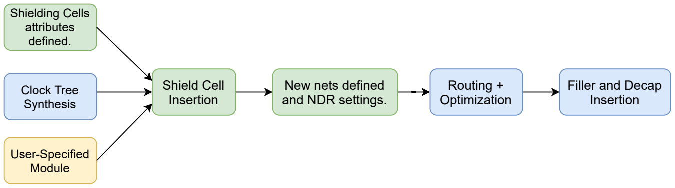

Shielding Cells attributes defined.
Clock Tree Synthesis
User-Specified Module
Shield Cell Insertion
New nets defined and NDR settings.
Routing + Optimization
Filler and Decap Insertion

Fig. 7. Integration of layout-level security closure techniques into physical-design flow – protection against frontside probing.

Thus, we have developed different scripts to scan as well as defend against frontside probing attacks, all within the physical-design stages. For scanning, we work directly at the polygon level. Given some sensitive cell or net, we extract all the wire polygons that are blocking the line of sight for an attacker seeking to lock-in on that asset. Using trigonometry, we ensure that probing attacks targeting at any angle is evaluated, all while considering the varying dimensions of different metal layers across the stack. For defense, we study two solutions conducted after clock-tree synthesis. The first solution, focused on defending nets, moves sensitive routes to lower metal layers while increasing the width of other, non-sensitive nets routed above, to reduce exposure/block the line of sight. The second solution, suitable for defending nets and/or cells, creates custom shielding cells in upper metal layers. The steps are outlined in Fig. 7. An example for the *MIT-LL CEP* benchmark is shown in Fig. 8.

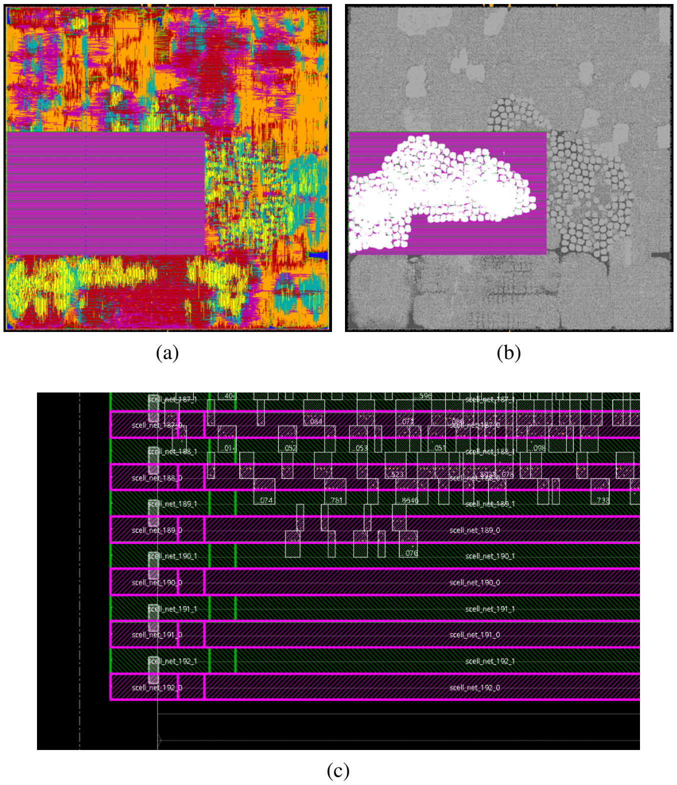
3) Scanning and Defending Against Crosstalk Attacks: With ever-more dense routing, especially for advanced nodes, parasitics and crosstalk (i.e., capacitive coupling) come into the picture. Signal transitions in some aggressor wires may trigger glitches (i.e., unintended signal flipping) in nearby victim wires, especially for long unbuffered wires.

In [31], this effect was exploited to launch a privilege-escalation attack on a microprocessor. However, they had to enable the attack by re-routing the victim net and some aggressor nets as extremely long parallel wires. To scan for such attacks, we developed scripts for sign-off metrics including noise and crosstalk. We found that (i) this attack [31] is easy to detect and (ii) our defenses integrated into *DEFense* do not support such attacks. This is expected, as physical-design tools handle crosstalk well, offering a built-in defense.

C. Advanced Attacks

Next, we outline some advanced attacks, targeting on vulnerabilities directly arising at the physical layouts. In future work, we will extend *DEFense* to counter such attacks as well.

1) Attacks Exploiting on Sign-Off Corners and Clock Skew: Among the various corners provided for multi-mode, multi-corner based timing analysis, physical designers often choose a subset of corners for sign-off checks such that most of the timing issues are covered and design closure is sped up. Even though most violations occurring for the corners not


(a)

(b)

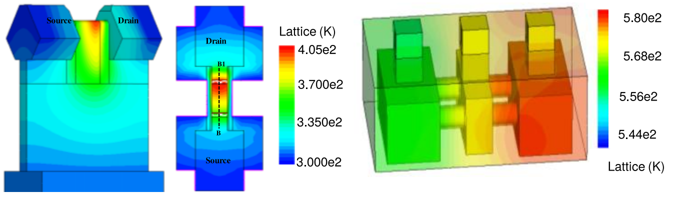
(c)

Fig. 8. Defense of *MIT-LL CEP* benchmark against frontside probing. Custom shielding cells are implemented in higher metal layers, blocking the access to the AES module below. (a) Top view routing, (b) Top view placement, AES module covered by shield routing highlighted. (c) Zoom-in for custom shielding cells.

covered are detected by the other sign-off checks, a few outlier violations may be missed, even during post-silicon testing. Such missed cases can represent a powerful threat.

For an attack demonstration, we impose short detours for the clock tree, thereby decreasing the hold margin and causing hold-timing violations. Note that the endpoint flop's clock latency can be targeted to cause hold violations only for a particular corner. Such increase in skew can be achieved either by adding additional buffers or logic in the clock path.

Such malicious modifications would be realized during physical design, e.g., using route guides. The attacker can study the libraries in detail to identify cases where the delay of cells scale with increase in the load across different corners. Using this insight, the routing length required to increase the delay at the output of a given cell can be determined and implemented in the layout. Detection of such modifications can be very difficult if the targeted corners are not tested for during sign-off analysis and post-silicon testing. As long as the additional clock logic does not change the functionality of the design, these modifications can remain stealthy. Finally, these modifications would cause violations and faults when the IC is operating in the particular corners in the field. Note that the concept is similar to the *CLKscrew* attack [32], but the implementation is more stealthy.



(a) 14nm FinFET

(b) 14nm Nanowire

Fig. 9. Impact of self-heating effects in transistors for 14nm FinFET [33] (a) and 14nm nanowire technology [34] (b), respectively.

2) Attacks on Power Grid and Decap Cells: In a high-performance and dense layout, standard cells are packed tightly. Cells of frequently-used modules draw large amounts of current from the power grid. To prevent IR drop violations, i.e., to build a robust power grid for these areas, physical-design engineers increase the number of power stripes or increase the width of existing stripes. Decap cells can also be added into the layout.

Such IR-hardened regions are easily identified in the final layouts. A foundry adversary can then, e.g., decrease the width of the power stripes across the different metal layers such that the overall resistance of the power grid increases. Such reduction in width can be minimal in the higher metal layers and yet have a significant impact on the IR drop. Similarly, decap cells can be replaced by filler cells. As decap cells have no functional purpose, replacement of such cells is easy for an attacker. Such attack can be stealthy and easily degrade the reliability of the fabricated ICs. A potential defense is to close the design at much lower IR-drop threshold. However, doing so still cannot guarantee to stop such attacks entirely. Future work is to study the attack and defense in more detail.

3) Attacks Exploiting Transistor Self-Heating: Transistor self-heating is emerging as serious concern in advanced technology nodes because of its harmful impact on performance, variability, and reliability [35]. To ensure electrostatic control in the face of technology scaling, the gate-all-around (GAA) structure with nanowires is important. However, the higher current provided by GAA structures, confined within a small geometry, results in significant power density that increases the channel temperature (due to Joule heating). In Fig. 9, we showcase self-heating effects on different transistor structures.

Self-heating is difficult if not impossible to capture during testing [36]. This is because even if transistors experience excessive self-heating, the temperature of the silicon die, e.g., obtained from high-precision infrared cameras, will exhibit lower temperatures—the heat generated by self-heating is often trapped within the transistors, especially in advanced geometries. Excessive heat inside the transistors significantly accelerates aging and reduces the lifetime of circuits; this can be exploited for disruptive Trojans or permanent-fault attacks. In addition, high temperatures can adversely impact IC operation by inducing glitches, faults, and timing errors, i.e., transient-fault attacks.

Thus, self-heating is a stealthy and versatile threat for fault attacks. Different factors contribute to self-heating and its varied scope of threat vectors:

1) The thermal characteristics of the underlying transistors (i.e., R_{th} and C_{th} of transistors), which are subject to the proprieties used during fabrication. Hence, there is a foundry threat vector.
2) The physical layout and its overall thermal characteristic, including active devices as well as interconnects. These characteristics are results of all the various trade-offs and optimization considered during physical design. Hence, there is, more or less explicit threat vector by designers.
3) The workload run by end-users dictates the switching activities of transistors and, thus, the degree of self-heating. Hence, there is also an end-user threat vector.

We argue that the most severe threat arises from a malicious foundry. During fabrication, changes in the doping profile of the transistor's channel, thickness of buried oxide, materials used to form the high-κ dielectric, source/drain contacts, and percentage of germanium in the channel impact the thermal characteristics of the transistors (R_{th} and C_{th}). All these aspects/knobs determine how excessive the self-heating effect will be during IC operation. Note that a malicious foundry can impose self-heating effects on sensitive layout regions of interest (e.g., to trigger faults in cipher modules) or across the layout (e.g., to make the IC fail in shorter time).

Approaches for security closure of physical layouts against self-heating-based attacks can be devised. For example, [37] shows that increasing the via-volume fraction from 3% to 9% nearby of self-heated transistor can lower the temperature by 10°C. While such a reduction might seem small, it has a large impact on mitigating aging effects, especially electromigration effects in Metal-1. Furthermore, routing patterns and the length of interconnects impact how the generated heat will propagate and impact other devices. Still, more research is required toward security closure against this stealthy, emerging threat.

VI. Conclusion

In this paper, we introduce the notion of *security closure of physical layouts*. There is a clear need for such, given that CAD flows are not focused on security yet, design efforts are more and more outsourced to third-party providers and, most importantly, any security flaw introduced or missed during layout sign-off cannot be fixed later on.

Given its importance and complexity, the notion of security closure of physical layouts will have to be driven by interdisciplinary teams of security researchers and practitioners, physical-design engineers, and CAD experts. In this paper, we took such joint efforts. We have compiled an extensive set of challenges toward security closure of physical layouts, along with strategies to tackle those. We conducted cases studies on selected threats, demonstrating that security closure of physical layouts is practical when integrated with commercial, sign-off-grade CAD flows. We covered a wide range of physical attacks within our case studies and our discussion, including advanced attacks that are overlooked so far. A summary matrix of threats and prospects for security closure of physical layouts against each threat is provided in Table I.

TABLE I
Physical-Design Stages Where Security Closure Applies (Green) or Not (Red)

Threats (↓)	Floor-Planning	Place	Clock-Tree Synth.	Route	Filler Cells	GDSII
Trojans	🟥	🟩	🟩	🟩	🟩	🟩
Probing	🟥	🟥	🟥	🟩	🟩	🟩
Clock Skew	🟥	🟥	🟥	🟥	🟩	🟩
Power Grid, Decap	🟥	🟥	🟥	🟥	🟩	🟩
Self-Heating	🟥	🟩	🟩	🟩	🟥	🟥
Reliability	🟥	🟩	🟩	🟩	🟩	🟩

Outreach via red-team-blue-team competitions can build a community focused on secure-by-design CAD flows. Toward this end, we are introducing the *IC Layout Security competition*, new for NYU CSAW 2021. For more details, please visit https://www.csaw.io/ic-layout-security

Appendix

A. Countermeasures Against Physical Attacks

For completeness, here we review prior art to defend against physical attacks. Note that most of the prior art does not follow any notion of security closure of physical layouts.

Countermeasures against SCAs are manifold and are tailored to the physical channel that is protected against. SCA-hardened implementations span device level to the software/application level. For example, prior art obfuscates the power consumption at the level of standard cells [38], [39], circuits [40], and systems [41].

FIA countermeasures include detection at runtime [42] and mitigation at design-time. Detection schemes may use sensors [43] and shielding structures [44]. FIA mitigation at design-time hardens the layout against FIA [45]–[47].

For probing countermeasures, e.g., Ishai *et al.* [48] leverage formal methods to establish guarantees on how many probing points are needed to leak data. Such schemes depend on 1) the formal methods being correctly implemented in the IC design—see counter-example in Sec. II—and 2) the attackers' capabilities. Both aspects are practical challenges [1], [22], rendering this scheme [48] secure in theory but insecure in practice. Other countermeasures are implemented directly with the devices or in the layout, e.g., shielding structures at the frontside [49], [50] and capacitive sensing [51].

Trojan countermeasures can be classified into detection during design, manufacturing, and testing versus mitigation at runtime. The former relies on testing, verification, and inspection [52]–[54], whereas the latter proposes security features for testability and self-authentication [55] or for monitoring of malicious activities [26], [56].

Acknowledgement

H. Amrouch would like to thank Om Prakash for transistor self-heating analysis presented in Fig. 9.

References

[1] J. Knechtel *et al.*, "Towards secure composition of integrated circuits and electronic systems: On the role of EDA," in *Proc. Des. Autom. Test Europe*, 2020, pp. 508–513.

[2] M. Randolph and W. Diehl, "Power side-channel attack analysis: A review of 20 years of study for the layman," *Crypt.*, vol. 4, no. 2, 2020.

[3] A. Barenghi *et al.*, "Fault injection attacks on cryptographic devices: Theory, practice, and countermeasures," *Proc. IEEE*, vol. 100, no. 11, pp. 3056–3076, 2012.

[4] J. Pan *et al.*, "One fault is all it needs: Breaking higher-order masking with persistent fault analysis," in *Proc. Des. Autom. Test Europe*, 2019.

[5] T. Krachenfels *et al.*, "Evaluation of low-cost thermal laser stimulation for data extraction and key readout," *Journal of Hardware and Systems Security*, vol. 4, no. 1, pp. 24–33, 2020.

[6] K. Bernstein, "Rapid authentication through verification, validation, and marking," DARPA, Tech. Rep., 2016.

[7] J. Lienig *et al.*, "Toward security closure in the face of reliability effects," in *Proc. Int. Conf. Comp.-Aided Des.*, 2021.

[8] K. Basu *et al.*, "CAD-Base: An attack vector into the electronics supply chain," *Trans. Des. Autom. Elec. Sys.*, vol. 24, no. 4, 2019.

[9] D. B. Roy *et al.*, "From Theory to Practice of Private Circuit: A Cautionary Note," in *Proc. Int. Conf. Comp. Des.*, 2015, pp. 296–303.

[10] S. Mangard, T. Popp, and B. M. Gammel, "Side-channel leakage of masked cmos gates," in *Topics in Cryptology – CT-RSA 2005*, Berlin, Heidelberg, 2005, pp. 351–365.

[11] S. Faust *et al.*, "Composable masking schemes in the presence of physical defaults & the robust probing model," *Proc. Cryptogr. Hardw. Embed. Sys.*, vol. 2018, no. 3, p. 89–120, Aug. 2018.

[12] G. Cassiers *et al.*, "Towards tight random probing security," in *CRYPTO 2021*, Cham, 2021, pp. 185–214.

[13] H. Lohrke *et al.*, "No place to hide: Contactless probing of secret data on FPGAs," in *Proc. Cryptogr. Hardw. Embed. Sys.*, 2016.

[14] P. Kocher *et al.*, "Spectre attacks: Exploiting speculative execution," in *Proc. Symp. Sec. Priv.*, 2019, pp. 19–37.

[15] R. J. Masti *et al.*, "Thermal covert channels on multi-core platforms," in *Proc. USENIX Sec. Symp.*, 2015, pp. 865–880.

[16] B. Selmke, J. Heyszl, and G. Sigl, "Attack on a DFA protected AES by simultaneous laser fault injections," in *Proc. Worksh. Fault Diag. Tol. Cryptogr.*, 2016, pp. 36–46.

[17] A. Dehbaoui *et al.*, "Injection of transient faults using electromagnetic pulses practical results on a cryptographic system," in *ePrint-123*, 2012.

[18] V. van der Veen *et al.*, "Drammer: Deterministic rowhammer attacks on mobile platforms," in *Proc. Comp. Comm. Sec.*, 2016, pp. 1675–1689.

[19] P. Qiu *et al.*, "VoltJockey: Breaching TrustZone by software-controlled voltage manipulation over multi-core frequencies," in *Proc. Comp. Comm. Sec.*, 2019, pp. 195–209.

[20] S. Tajik *et al.*, "Photonic side-channel analysis of arbiter PUFs," *J. Cryptol.*, vol. 30, no. 2, pp. 550–571, 2017.

[21] ——, "On the power of optical contactless probing: Attacking bitstream encryption of FPGAs," in *Proc. Comp. Comm. Sec.*, 2017, pp. 1661–1674.

[22] T. Krachenfels *et al.*, "Real-world snapshots vs. theory: Questioning the t-probing security model," 2020.

[23] M. von Haartman *et al.*, "Optical fault isolation and nanoprobing techniques for the 10 nm technology node and beyond," in *Proc. Int. Symp. Test. Failure Analys.*, 2015, pp. 52–56.

[24] C. Helfmeier *et al.*, "Breaking and entering through the silicon," in *Proc. Comp. Comm. Sec.*, 2013, pp. 733–744.

[25] R. Karri *et al.*, "Trustworthy hardware: Identifying and classifying hardware trojans," *Computer*, vol. 43, no. 10, pp. 39–46, 2010.

[26] K. Xiao *et al.*, "Hardware trojans: Lessons learned after one decade of research," *Trans. Des. Autom. Elec. Sys.*, vol. 22, no. 1, 2016.

[27] D. Fujimoto *et al.*, "Correlation power analysis using bit-level biased activity plaintexts against AES cores with countermeasures," in *Proc. Int. Symp. Electromag. Comp.*, 2014, pp. 306–309.

[28] Y.-C. Lu *et al.*, "GAN-CTS: a generative adversarial framework for clock tree prediction and optimization," in *Proc. Int. Conf. Comp.-Aided Des.*, 2019, pp. 1–8.

[29] A. Agnesina, K. Chang, and S. K. Lim, "VLSI placement parameter optimization using deep reinforcement learning," in *Proc. Int. Conf. Comp.-Aided Des.*, 2020.

[30] T. Trippel *et al.*, "ICAS: an Extensible Framework for Estimating the Susceptibility of IC Layouts to Additive Trojans," in *Proc. Symp. Sec. Priv.*, 2020, pp. 1742–1759.

[31] C. Kison *et al.*, "Security implications of intentional capacitive crosstalk," *Trans. Inf. Forens. Sec.*, vol. 14, no. 12, pp. 3246–3258, 2019.

[32] A. Tang, S. Sethumadhavan, and S. Stolfo, "CLKSCREW: Exposing the perils of security-oblivious energy management." Proc. USENIX Sec. Symp., 2017, p. 1057–1074.

[33] V. M. van Santen *et al.*, "Impact of self-heating on performance, power and reliability in finfet technology," in *Proc. Asia South Pac. Des. Autom. Conf.*, 2020, pp. 68–73.

[34] O. Prakash *et al.*, "Impact of NBTI aging on self-heating in nanowire FET," in *Proc. Des. Autom. Test Europe*, 2020, pp. 1514–1519.

[35] S. Shin *et al.*, "Substrate and layout engineering to suppress self-heating in floating body transistors," in *Proc. Int. Elec. Devices Meeting*, 2016.

[36] O. Prakash *et al.*, "Transistor self-heating: The rising challenge for semiconductor testing," in *VLSI Test Symp.*, 2021, pp. 1–7.

[37] W. Ahn *et al.*, "Integrated modeling of self-heating of confined geometry (FinFET, NWFET, and NSHFET) transistors and its implications for the reliability of sub-20 nm modern integrated circuits," *Microelectronics Reliability*, vol. 81, pp. 262–273, 2018.

[38] D. Bellizia *et al.*, "Secure double rate registers as an RTL countermeasure against power analysis attacks," *Trans. VLSI Syst.*, vol. 26, no. 7, pp. 1368–1376, 2018.

[39] F. Zhang *et al.*, "Design and evaluation of fluctuating power logic to mitigate power analysis at the cell level," *Trans. Comp.-Aided Des. Integ. Circ. Sys.*, 2020.

[40] S. Das *et al.*, "Abetting planned obsolescence by aging 3D networks-on-chip," in *Proc. NoC Symp.*, 2018, pp. 1–8.

[41] J. Knechtel and O. Sinanoglu, "On mitigation of side-channel attacks in 3D ICs: Decorrelating thermal patterns from power and activity," in *Proc. Des. Autom. Conf.*, 2017, pp. 1–6.

[42] X. Guo *et al.*, "Security analysis of concurrent error detection against differential fault analysis," *Journal of Cryptographic Engineering*, vol. 5, pp. 153–169, 2014.

[43] N. Homma *et al.*, "EM attack is non-invasive? - design methodology and validity verification of EM attack sensor," in *Proc. Cryptogr. Hardw. Embed. Sys.*, 2014, pp. 1–16.

[44] X. T. Ngo *et al.*, "Cryptographically secure shield for security IPs protection," *Trans. Comp.*, vol. 66, no. 2, pp. 354–360, 2017.

[45] M. Li *et al.*, "Cross-level monte carlo framework for system vulnerability evaluation against fault attack," in *Proc. Des. Autom. Conf.*, 2017.

[46] M. Khairallah *et al.*, "DFARPA: differential fault attack resistant physical design automation," in *Proc. Des. Autom. Test Europe*, 2018, pp. 1171–1174.

[47] R. A. Viera *et al.*, "Standard CAD tool-based method for simulation of laser-induced faults in large-scale circuits," in *Proc. Int. Symp. Phys. Des.*, 2018, pp. 160–167.

[48] Y. Ishai, A. Sahai, and D. Wagner, "Private circuits: Securing hardware against probing attacks," in *Advances in Cryptology*, 2003, pp. 463–481.

[49] H. Wang *et al.*, "Probing assessment framework and evaluation of antiprobing solutions," *Trans. VLSI Syst.*, vol. 27, no. 6, 2019.

[50] K. Yi, M. Park, and S. Kim, "Practical silicon-surface-protection method using metal layer," *J. Semicond. Tech. Sci.*, vol. 16, no. 4, 2016.

[51] M. Weiner *et al.*, "The low area probing detector as a countermeasure against invasive attacks," *Trans. VLSI Syst.*, vol. 26, no. 2, 2018.

[52] Y. Jin and Y. Makris, "Hardware trojan detection using path delay fingerprint," in *Proc. Int. Symp. Hardw.-Orient. Sec. Trust*, 2008.

[53] E. Love, Y. Jin, and Y. Makris, "Proof-Carrying Hardware Intellectual Property: A Pathway to Trusted Module Acquisition," *Trans. Inf. Forens. Sec.*, vol. 7, no. 1, 2012.

[54] X. Guo *et al.*, "QIF-Verilog: Quantitative information-flow based hardware description languages for pre-silicon security assessment," in *Proc. Int. Symp. Hardw.-Orient. Sec. Trust*, 2019, pp. 91–100.

[55] K. Xiao, D. Forte, and M. Tehranipoor, "A novel built-in self-authentication technique to prevent inserting hardware trojans," *Trans. Comp.-Aided Des. Integ. Circ. Sys.*, vol. 33, no. 12, 2014.

[56] M. Nabeel *et al.*, "2.5D root of trust: Secure system-level integration of untrusted chiplets," *Trans. Comp.*, vol. 69, pp. 1611–1625, 2020.