Secure Physical Design

Sukanta Dey, Jungmin Park, Nitin Pundir, Dipayan Saha, Amit Mazumder Shuvo, Dhwani Mehta, Navid Asadi, Fahim Rahman, Farimah Farahmandi, and Mark Tehranipoor

FICS Research Institute, Dept of Electrical and Computer Engineering,
University of Florida, Gainesville, FL, US

***Abstract*—An integrated circuit is subject to a number of attacks including information leakage, side-channel attacks, fault-injection, malicious change, reverse engineering, and piracy. Majority of these attacks take advantage of physical placement and routing of cells and interconnects. Several measures have already been proposed to deal with security issues of the high level functional design and logic synthesis. However, to ensure end-to-end trustworthy IC design flow, it is necessary to have security sign-off during physical design flow. This paper presents a secure physical design roadmap to enable end-to-end trustworthy IC design flow. The paper also discusses utilization of AI/ML to establish security at the layout level. Major research challenges in obtaining a secure physical design are also discussed.**

***Index Terms*—Attack, EDA Security, Physical Layout, Physical Design, RTL to GDS-II flow, Security, Trust, Vulnerability**

I. Introduction

The worldwide semiconductor market has been growing due to the increase in demand for smartphones, 5G wireless devices, wearable devices, gaming, autonomous systems, servers, and artificial intelligence (AI)-based computing. Research firm International Data Corporation (IDC) expects the market to grow by 17.3% in 2021 versus 10.8% in 2020 [1]. To satisfy time-to-market in the semiconductor industry as well as the complex specification of an SoC, the reuse of intellectual property (IPs) and the global semiconductor supply chain are inevitable. Due to globalization, multiple parties are involved in the IC design. As a result, adversary can introduce malicious IPs in the design flow. This can lead to unintentionally serious hardware security breaches [2], such as asset leakage [3], [4], hardware trojan induced confidentiality and integrity violations [5], [6], [7], [8], [9], [10], [11], [12], [13], side-channel leakage [14], [15], [16], [17], and fault-injection vulnerability [18], [19], [20], [21], [22], [23]. Apart from these, counterfeiting and recycling of integrated circuits have become a significant piracy threat in microelectronics supply chain [12], [24], [25], [26], [27], [28], [29], [30], [31], [32].

With the increase in complexity of the chip design and the involvement of multiple entities to create the design, it has become necessary to place security checks at all levels of the design cycle. Much in the literature has tried to address hardware security issues and countermeasures, starting from high-level synthesis to gate-level synthesis [33], [34], [35], [36]. However, little work has been invested in security verification at the physical design stage to ensure the design is protected against vulnerabilities that take advantage at cell placement and wire routing. A secure gate-level netlist may become insecure after the generation of its physical layout. This is because the physical information (i.e., cell placement, power vias, power distribution network, placement of decaps, timing criticality of paths, obscurity of cells by wires in metal lines, etc.) is not available during the logic synthesis phase. Hence, making security sign-off during and after physical design[1] a necessity.

Over the past several years, hardware security community has concentrated mostly on securing logic designs at a higher level of abstraction. Suppose we have obtained a secure gate-level netlist after applying all security measures in a synthesized netlist. Subsequently, when the placement and routing (P&R) tool generates the layout of the chip, many physical information becomes available. The concern is: can an adversary utilize this physical information to extract vital information from the packaged chip i.e., cause confidentiality violation? Or can an adversary manipulate logical values in the circuit to cause integrity violations? We have demonstrated a few such vulnerabilities in this paper. That shows such questions are of major concerns that need to be carefully addressed. Therefore, we need to have a secure physical layout of the chip, free from any physical vulnerabilities, from which any information leakage is not possible; This is called *secure physical design* (SEPHYD). Overall, objective of this paper is to demonstrate a perspective to identify physical characteristics that are related to security vulnerabilities and use those physical characteristics to guide the physical design process to ensure security sign-off.

This paper will answer the following questions,

- What vulnerabilities are possible at the physical layout?
- How to identify vulnerabilities at the layout level?
- What countermeasure can be applied to establish security-aware physical design flow?
- What are the challenges concerning secure physical design?

The organization of the paper is as follows. In Section III, we have provided basic introduction to physical design and some recent works related discussion. In Section IV, we have described possible vulnerabilities that can occur in physical design level[2]. In Section V, proposed flow for secure physical design and verification sign-off is described. In Section VI, we have demonstrated an AI/ML roadmap to obtain the secure physical design. Open challenges for establishing a secure physical design are listed in Section VII. We conclude the paper in Section VIII.

[1] In this paper, we have used the term “physical design” and “layout” interchangeably.

[2] We also use the term “physical design level” and “physical level” interchangeably throughout the paper.

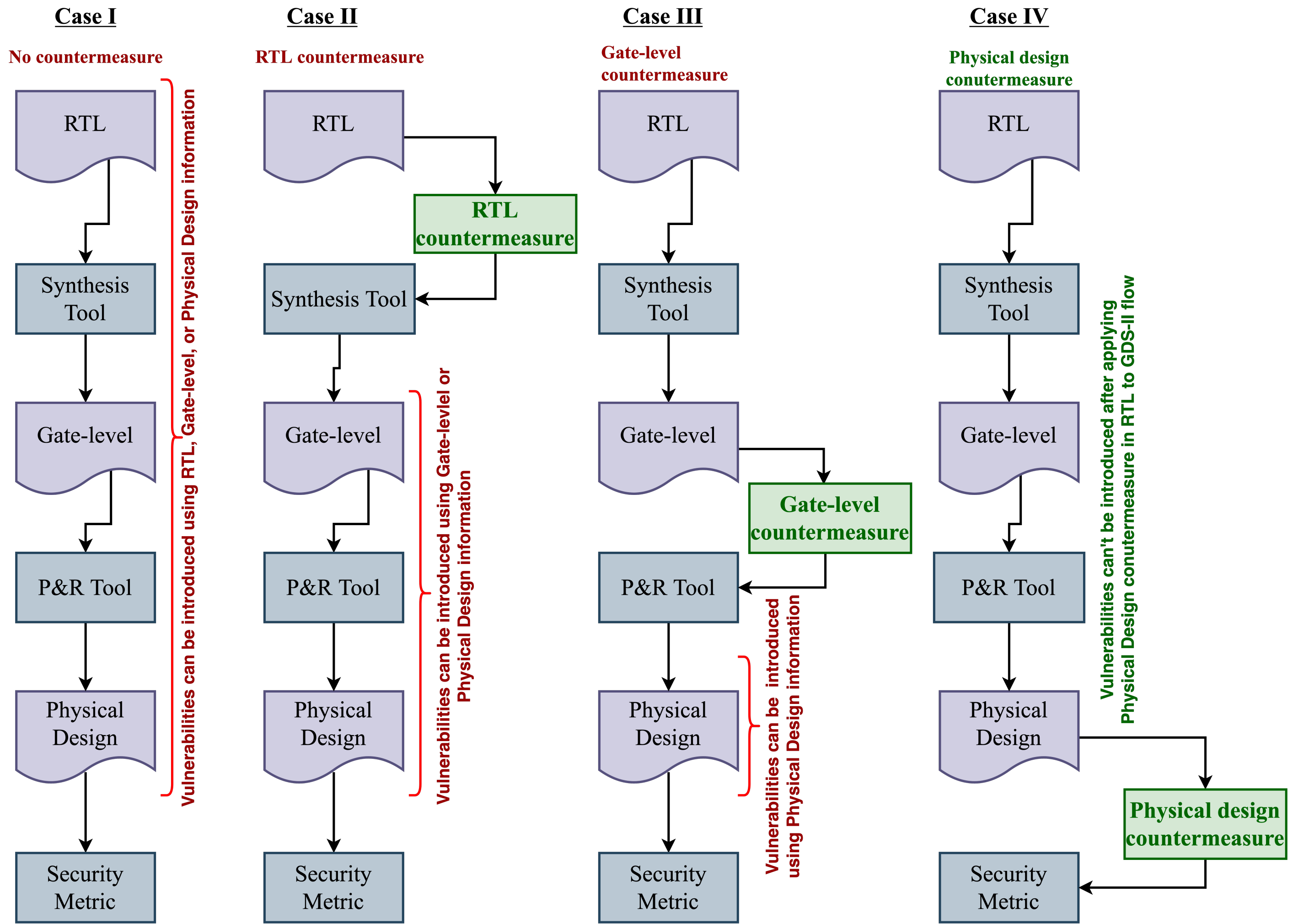

Fig. 1. Security metric evaluations in physical design level with various levels of countermeasures applied in RTL to GDS-II flow

II. Motivating Example

An integrated circuit can be made secure by applying countermeasures at different levels of abstractions i.e.,: RTL, gate, and physical design levels, as shown in Fig. 1. The extent of security can be ensured if we have a security metric to evaluate the vulnerabilities of the final physical design, as shown in Case IV of Fig. 1. The usual practice of making secure integrated circuit possesses to apply the countermeasures at RTL level or gate-level. We already discussed in the previous section that applying only RTL level/gate-level countermeasure doesn't make the packaged chip secure, as many physical information is not available at the high-level functional design. Further, sometime situations may arise that make it difficult to apply RTL level or gate-level countermeasures. It is also possible that logic designers have already transferred the design (after RTL and gate-level sign-off) to physical design team without applying the RTL/gate-level countermeasures. In such a situation, having a physical design level countermeasure becomes very essential. Therefore, irrespective of the countermeasures applied at RTL level/gate-level or not, it is necessary to apply physical design countermeasures in order to obtain a secure chip package. Considering the above situation as our motivation, we discuss various aspects of realizing secure physical design in this paper.

III. Preliminaries

A. Physical Design

The physical design phase starts with a synthesized RTL netlist as input, also known as a gate-level netlist. The physical design steps majorly consist of *Floorplanning, Placement,* and *Routing*. In the floorplanning step, estimated positions of macros and standard cells are determined. In this step, an on-chip power grid network is also generated. Once the initial positions of the blocks are determined, then it goes to the placement stage. Again, placement stage follows three sub-steps; *Global Placement, Legalization,* and *Detailed Placement*. In global placement, blocks are loosely placed. In the legalization stage, checks for overlapping blocks are performed. Finally, we obtain the final placement of macros and standard cells in the detailed placement stage. The primary objective of the physical design stage is to build the geometrical and physical layout of the design with the exact specifications as defined in the RTL phase. Simultaneously,

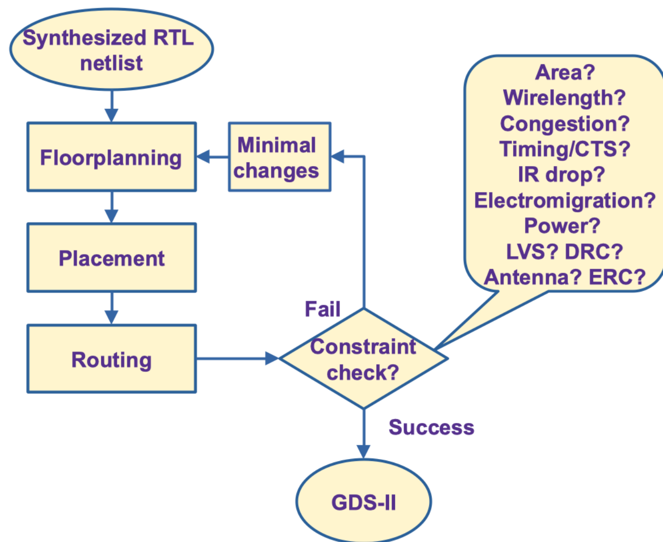

Fig. 2. Traditional physical design flow.

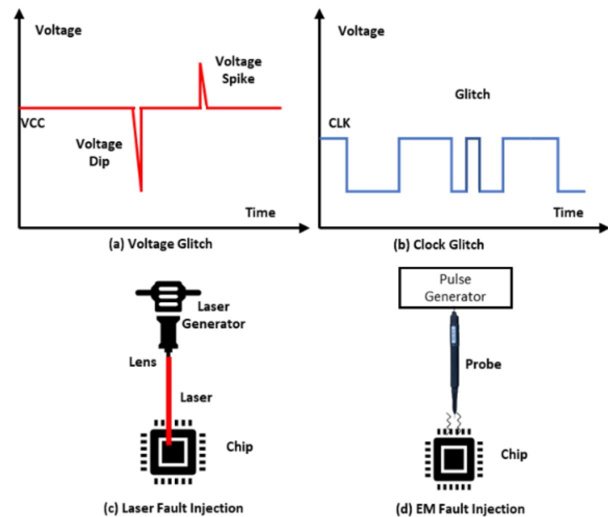

Fig. 3. Different fault-injection techniques.

optimizing the area, power, and other objectives or constraints as shown in Fig. 2 are objectives of the physical design stage. There exist many other sub-steps of physical design stage, which include *Power Planning, Clock Tree Synthesis* (CTS), *Equivalence Check, Parasitic Extraction*, and several sign-off stages. These include *IR drop sign-off, timing sign-off, and physical sign-off*. In the power planning phase, the power grid is designed and initial IR/Electromigration sign-off is conducted. In CTS step, it is ensured that the clock reaches evenly in all the sequential elements of the design. The objective of the CTS step is to minimize clock skew and insertion delay. *Layout Versus Schematic* (LVS) is the process where equivalence logical connections and physical layout connections are checked. *Design Rule Checking* (DRC) is the process that verifies if the design is compatible with the foundry fabrication rules. In *Electrical Rule Checking* (ERC), a design is checked for substrate areas to verify proper contacts and spacing, which ensures correct power and ground connections. We have shown each of these substeps in Fig. 2 as a constraint check step. However, in practice, these checks are performed chronologically at a specific defined stage. Finally, when each check is successful, we receive the final layout in *Graphic Design System* (GDS)-II format. This GDS-II is sent to foundry for fabricating the layout into silicon.

B. Related Work

In the last two decades, there have been several works to devise efficient physical design algorithms that can produce cost-effective and fast time-to-market design solutions [37]. However, it has been observed that some vulnerabilities still exist even after the security sign-off in the RTL phase of the design, as many physical parameters are not present in the RTL design steps. Therefore, evaluating the security metrics in the physical layout phase has become more critical to produce trustworthy physical designs of an integrated circuit, free from any vulnerabilities. In this section, previous works related to vulnerability and security in the physical design are discussed. There are few works in literature that explores side-channel and fault injection vulnerabilities in the physical design stage. Firstly, we describe works related to side-channel vulnerability assessment and fault injection vulnerabilities at the physical design level. Later on, we also present a brief literature survey on secure physical design-related works.

1) Works Related to Side-channel Vulnerability Assessment in Physical Design: Side-channel vulnerability assessment at the physical design level is crucial because there is more information in the layout then there is at the RTL and gate-level. For example, the capacitance and resistance of the wires between blocks and standard cells are exploited as additional sources of side-channel leakage at the physical layout level. Cross-talk and IR drop, which are the potential of such security leakages, are not available at the RTL. This is why leakage analysis at the layout level offers the highest accuracy among the levels of abstraction in the pre-silicon stage. Very few works have been published regarding side-channel leakage at the layout level. Authors in [38] investigated the impact of the physical layout on side-channel security. They co-simulated the analog power delivery network (PDN) with a digital logic core. They quantified the impact of different layout parasitics such as parasitic resistors, inductors, capacitors, and power supply buffers. By examining the impact of these layout parasitics, they provided a deeper insight into potential layout sources. However, one major limitation of this work is the scalability of the approach. The authors applied their analysis to a minimal design (AND2, XOR2) because of the lack of computational ability to analyze the larger design. Cnudde *et al.* [39] claimed that placement and routing cause information leakage in FPGA. Some works [40] [41] used fast SPICE-based simulations to evaluate side-channel leakage analysis at back-end stages. However, these works also suffer from scalability problems for a larger design. Unlike other works, Lin *et al.* [42] developed fast simulation methodology for the layout-based power-noise side-channel leakage analysis. Their tool can improve the simulation time by 110 times for a SoC design compared to VCD-based analysis.

2) Works Related to Fault Injection Attack: Several fault injection attacks in security-critical applications are shown in recent research. These applications include an embedded system [43], [44], microprocessor-based implementations [45], RFID tags [46], TRNG, SRAM, PLL, oscillators and so on [47], [48], [49], [50]. Among the global fault injection techniques, clock glitching, voltage glitching, and thermal glitching are widely used to inject exploitable faults in an Application-Specific Integrated Circuit (ASIC). Extensive research is performed to prove the feasibility and effectiveness of these global techniques [51], [52], [53]. In addition, local fault injection techniques are well researched, like laser/optical fault injection and EM fault injection [54], [55], [56]. Apart from non-invasive and semi-invasive attacks, faults can be injected using invasive techniques (e.g., FIB probing) proposed in literature [57], [58]. Clock glitching involves inserting glitches in the clock supply or disturbing the normal behavior to shorten the effective period of the clock, as shown in Figure 3(b). These clock glitches can cause setup and hold time violations [59], [60], create metastability in the sequential elements, and inject timing faults. Since voltage starvation can also impact the clock and circuit timing, voltage and clock glitching attacks are combined to strengthen timing fault injection attacks. Voltage glitching involves a momentary power drop or spike in the supply voltage during the operation, as shown in Figure 3(a). These voltage spikes or drops increase the logic propagation delay, thus, causing timing violations [61]. A voltage glitch can be caused by either disturbing the main power supply, creating a global effect, or by running a power-hungry circuit, like ring oscillators, or ROs, creating a localized voltage drop [62]. In recent years, remote fault injection attacks have been caused by disturbing the supply voltage remotely by using either software-based registers to control the supply voltage or by using ROs to cause a voltage drop in remote FPGAs [63], [64], [65]. Moreover, timing faults can be injected by inserting glitches in operating temperature. Overheating increases the scattering of charge carriers, decreasing mobility and eventually affecting the logic propagation delay [61]. Optical/laser fault injection attacks use different wavelength laser/optics to inject transient faults from either the front or backside of the chip, as shown in Figure 3(c). To attack from the front-side, a higher wavelength laser (1300μm) is used, whereas from the backside a near-infrared laser (1064μm) is used (due to its lower absorption coefficient). The injected laser generates electron-hole pairs in the active region, which drift apart under the electric field's influence, resulting in transitory currents. These transient currents cause the transistors to conduct and thus cause the charging/discharging of the capacitive loads. Laser/optical fault injection generally requires de-packaging of the chip to expose the die and thus qualifies as a semi-invasive attack. However, with the laser/optics, an attacker can target the precise locations and time of the laser injection, thus making it a powerful attack [54]. Similarly, EM fault injection attacks use electric or magnetic field flux to influence the normal functioning of the device, as shown in Figure 3(d). An electromagnetic field causes voltage and current fluctuations inside the device, leading to the faults [66].

3) Works Related to Security of Physical Design: Many works of literature address securing physical layout fabrication by split manufacturing [78], [69], [71], where front end of line (FEOL) interconnects are manufactured in untrusted foundries and back end of line (BEOL) interconnects are manufactured in trusted foundries to manufacture secure layouts and prevent counterfeit. However, split manufacturing does not ensure inherent layout security, as security is achieved by splitting the physical layout of an integrated circuit. There are few works which tried to obtain layout-level security assessment for hardware trojan-based and microprobing-based vulnerability [79], [80], [81], [82]. There is a need to obtain security by design for physical layouts. There have been very few works that deal with this problem of obtaining secure physical layouts. We have listed these works in Table I. In [67], authors have proposed intellectual property protection for VLSI physical design. In [68], authors have proposed a netlist scrambling-based technique to prevent reverse engineering. In [70], authors have proposed a physical design level hardware obfuscation technique to prevent reverse engineering. In [72], authors have proposed a differential fault analysis attack preventive physical design flow using floorplan heuristics. In [57], authors have proposed a physical design flow considering anti-probing attacks. The flow utilizes internal shielding for protection. However, these approaches mentioned in Table I are not scalable due to the difficulty in implementing them separately, which makes the design flow costly. It is necessary to have a scalable IC design flow, where all security vulnerabilities in the physical design level are checked for a successful sign-off. Recent work [77] demonstrates a top-level overview of security closure of physical layouts. The work [77] mainly focuses on a case study of scanning and defending against Trojans and frontside probing attack. However, it does not provide a comprehensive roadmap for obtaining a secure physical design. This work, however, presents a roadmap to obtain secure physical designs and considers various vulnerabilities at the physical design level.

IV. Possible Vulnerabilities in Physical Design

Even after the security closure of gate-level synthesis, several vulnerabilities may arise utilizing physical design parameters at different steps. It has been observed that some vulnerabilities may arise due to the poor floorplanning stage. For example, while designing an SoC, an AES crypto module should be placed far away from the power supply pins in order to become less vulnerable to power side-channel attacks. Similarly, every step of physical design plays a crucial role in terms of security vulnerability. Any poor choices of floorplanning, placement, routing or post-routing steps in physical design may increase the vulnerability level of the design. Based on the dependency of the vulnerabilities on various stages of physical design, we have broadly classified all the trust-hub physical vulnerabilities [83] into five categories which are floorplanning, placement (within a module), placement (SoC level), routing, and post-routing and listed it in Table II and Table III. Further, the dependency of some of these vulnerabilities on the physical design stages are not well-defined. Most of

TABLE I
PHYSICAL DESIGN SECURITY RELATED PREVIOUS WORKS

Work	Year	Physical Vulnerabilities	Proposed Countermeasure	Remarks
[67]	2007	IP protetion	Robust IP scheme	Theoretically proved; MCNC benchmarks
[68]	2013	Reverse Engineering	Netlist Scrambling	Random scrambling is employed; IWLS 2005 benchmarks
[69]	2015	Proximity Attack	Partitioning-based heuristic	Solution for 2.5D IC; ISCAS'85 and ITC'99 benchmarks
[70]	2016	Reverse Engineering	Hardware Obfuscation	Device to Logic-level investigation
[71]	2017	Enhanced Security Split Manufactiring	Routing Perturbation	Split fabrication approach; ISCAS'85 and ITC'99 benchmarks
[72]	2018	Differential Fault Attack	floorplan heuristic	Deals with localized faults; AES and Plantlet benchmarks
[73]	2019	Power Side-Channel	Gate reconfiguration	Divide & Conquer Approach; AES, SIMON, PRESENT benchmarks
[57]	2019	Probing Attack	Internal Shielding	Probing attack countermeasure; AES and DES benchmarks
[74], [75]	2021	Power/EM Side-Channel	Backside Power Grid	Distributed decap-based mitigation ; AES and ECC benchmarks
[76]	2021	EM Side-Channel	Modified Power Grid and Decap	Decap-based mitigation; AES and DES benchmarks
[77]	2021	Trojan and Frontside Probing	Scanning and Defending	DEFense countermeasure; AES, MIT-LL CEP benchmark

the active invasive attacks can be prevented if we do security-aware floorplanning, placement, routing. Similarly, for other vulnerabilities we can establish security-aware physical design steps. Therefore, the problem of obtaining secure physical design lies in secure floorplanning, secure placement, secure routing, and secure post-routing measures. Out of these vulnerabilities mentioned in Table II and III, physical designs are most vulnerable to the following categories of attacks, which are described in detail in subsequent subsections.

A. Power and EM Side-channel Vulnerabilities

A side-channel attack is considered one of the most crucial threats and most studied security vulnerabilities. In 1996, Kocher drew the attention of the security community both from academia and industry through the first timing attack on different cryptography algorithms [84]. Later, through the invention of simple power analysis (SPA) and differential power analysis (DPA), he showed that there is a relationship between power consumption and the data being processed [85]. In the aftermath, researchers concentrated on the implementation, rather than the weakness in the algorithm, to recover the key. Since then, quality research has been conducted in the side-channel domain. Along with power consumption, other side-channel information, such as electromagnetic (EM) radiations [86], timing [84], sound [87], photonic [88], and micro-architectural elements [89] have been studied. It is found that there is a connection between these physical signals and the properties of the computational stack. In this work, we mainly discuss the power/EM side-channel attack.

The data dependency of power consumption [90] is prevalent at every level of abstraction. For example, different instructions might consume different amounts of power depending on the number of cycles and switching activity the instruction requires [91], [92], which is defined at the system specification level and can be a source of side-channel leakage. Data-dependent switching activities and state transitions in the RTL cause side-channel leakage [17]. The data dependency can be eliminated by randomizing all intermediate results that occur during computations, called *masking* in the RTL [93]. However, the masked implementation can be vulnerable to power/EM side-channel attacks through *the effect of glitches* caused by timing properties of gates and interconnection delays at the gate level and the transistor level [94], [95]. Nikova *et al.* developed a side-channel resistant masking technique, called *threshold implementation*, even in the presence of glitches based on secret sharing, threshold cryptography, and multi-party computations [96]. Nevertheless, the threshold implementations also leak side-channel information to reveal secret keys caused by tightly placing logically independent shares, called *the effect of coupling capacitance*, and *IR drop* [97], and by *parasitic capacitance, resistance, and inductance* [38] as byproducts of the placement and routing in the layout level. Consequently, even though side-channel leakage cannot be detected at the higher level of abstraction, it can be determined at the advanced levels of abstraction.

To identify the side-channel leakage at various levels of abstraction, *post-silicon* and *pre-silicon side-channel leakage simulators* have been developed. The post-silicon simulator first generates a set of estimated power/EM traces and then performs a leakage detection test (e.g., TVLA [98]). The estimated power/EM traces are based on the stochastic modeling (e.g., a linear regression model) depending on consecutive instructions and operands in a target device, which requires *a prior* knowledge about the instruction set architecture and power/EM measurements of the target device, such as ARM Cortex-M0 or RISC-V [99], [91], [100], [101]. These post-silicon simulators can detect side-channel vulnerable instructions and then rewrite codes to mitigate side-channel leakage [100], [101]. The post-silicon simulator can be used for developing side-channel resistant software in a typical device without side-channel measurements. On the other hand, the pre-silicon simulator can help hardware designers identify side-channel leakages in the design stages, which are the RTL, gate level, and layout level. For example, RTL-PSC [17] can detect vulnerable modules in a complete design at the RTL in such a way that it profiles power consumption by counting the number of transitions in each module during computations with randomly generated plaintext and fixed-key inputs in an AES design and then calculates the statistical distance between two different sets, which correspond to two different keys. SCRIPT [102] identifies target registers that leak side-channel information by utilizing information flow tracking at the gate level. It can estimate signal-to-noise (SNR) by dividing the power difference generated by a pair of specific patterns to make the maximum Hamming distance of target registers by the power consumption of the rest of the design based on vectorless power analysis. Karna [103] searches for vulnerable gates in the layout level and then changes the gate parameter

TABLE II
CLASSIFICATION OF ACTIVE ATTACKS OF TRUST-HUB PHYSICAL VULNERABILITIES-DB [83] BASED ON ITS DEPENDENCY TO PHYSICAL DESIGN STEPS.

Active Attacks			Floorplanning	Placement (Within module)	Placement (SoC level)	Routing	Post-Routing
Invasive	Die Analysis	Delayering, Netlist Reconstruction	✓	✗	✓	✓	✗
		Grind	✓	✗	✓	✓	✗
		Section	✓	✗	✓	✓	✗
		Dimple Down	✓	✗	✓	✓	✗
		Photon(Laser) Induced Current	✓	✓	✓	✓	✗
		Focused Ion Beam Deposition	✓	✓	✓	✓	✗
		Focused Ion Beam Removal	✓	✓	✓	✓	✗
		Ion Milling	✓	✓	✓	✓	✗
		Direct Metal or Contact Probing	✓	✓	✓	✓	✗
		Light Sensing	✓	✓	✓	✓	✗
		Circuit Parameter Sensing	✓	✓	✓	✓	✗
	Board Analysis	Delayering, Netlist Reconstruction	✓	✓	✓	✓	✗
	Design or FAB Injection	HW Trojan	✓	✓	✓	✓	✓
Non-Invasive	Timing	Delay Analysis	✓	✓	✓	✓	✓
		Clock Glitching Injection	✓	✓	✓	✓	✓
		Overclocking	✓	✓	✓	✓	✓
		Underclocking	✓	✓	✓	✓	✓
	Fault Injection	Photon(Laser) Induced current	✗	✓	✓	✗	✗
		Ambient / Ultra - violet	✗	✓	✓	✗	✗
		Ionizing Radiation	✗	✓	✓	✗	✗
		E and M Field	✓	✓	✓	✗	✗
		Voltage Spike	✓	✓	✓	✓	✓
		Temperature	✓	✓	✓	✗	✗
		Over / Under Voltage	✓	✓	✓	✓	✓

TABLE III
CLASSIFICATION OF PASSIVE ATTACKS OF TRUST-HUB PHYSICAL VULNERABILITIES-DB [83] BASED ON ITS DEPENDENCY TO PHYSICAL DESIGN STEPS.

Passive Attacks			Floorplanning	Placement (Within module)	Placement (SoC level)	Routing	Post-Routing
Non-Invasive	Side-Channel Observation	Acoustic	✓	✗	✓	✗	✗
		Photoemission	✓	✓	✓	✓	✓
		Voltage, Charge contrast	✓	✓	✓	✓	✓
		SEM Inspection	✓	✗	✓	✗	✗
		IREM Inspection	✓	✓	✓	✓	✓
		Temperature Imaging	✓	✗	✓	✗	✗
		E or M Fields	✓	✓	✓	✓	✓
		Current & Power Measurement	✓	✓	✓	✓	✓
		Voltage Measurement	✓	✓	✓	✓	✓
		Indirect Voltage Measurement	✓	✓	✓	✓	✓
		Data Remanence	✓	✗	✓	✓	✓
		Black Box I / O	✓	✗	✓	✗	✗
	Logical Attacks	Brute Force Algorithm	✓	✗	✓	✗	✗
		Protocol Attacks	✓	✗	✓	✗	✗

through threshold voltage, supply voltage, or the size of gates to mitigate side-channel vulnerability without the cost of performance and area. Recent work [104] also demonstrates a side-channel evaluation platform for post-quantum algorithms.

Table IV summarizes the evaluation time and accuracy of power/EM side-channel leakage estimation, the flexibility to make design changes at various pre-silicon design phases, and the post-fabricated device, and available tools. There is a trade-off between time/accuracy and flexibility at various design stages when assessing side-channel leakage. Although the post-silicon simulator offers the highest accuracy and fastest processing time, it does not allow design modifications to address potential vulnerabilities. Higher levels of pre-silicon abstraction, on the other hand, provide more flexibility at the expense of accuracy. Manufacturers prefer to discover side-channel vulnerabilities early in the product development process. Early detection of leakage is better because, according to the rule of 10, the cost associated with leakage detection, identification, and mitigation will increase by ten folds if detection is delayed by one stage.

TABLE IV
COMPARISON: PRE AND POST-SILICON SIDE-CHANNEL LEAKAGE SIMULATOR.

	Pre-silicon Simulator			**Post-silicon Simulator**
	RTL	**Gate-level**	**Layout**	
Time	Medium	High	Very High	Low
Accuracy	Low	Medium	High	Very High
Flexibility	High	Medium	Low	Not Feasible; Only Software
Tool	RTL-PSC [17], RTL-PAT [105], RTL-TG [106]	SCRIPT [102], EMSIM [107] Coco [108]	Karna [103]	ELMO [99], ROSITA [100] ROSITA++ [101]

B. *Fault-injection Vulnerabilities*

An attacker intentionally injects a fault in the system to change its runtime behavior. The attacker exploits this change to leak sensitive information or gain access to privileged functionality. Fault injection attacks are primarily physical attacks. However, recent software-assisted hardware attacks have shown that fault injection could also be performed in remote systems [63], [109]. Based on the means of fault injection, the attacks could be classified into voltage glitching [52], clock glitching [51], optical/laser injection [54], [110], and EM injection [111].

1) Optical/Laser Fault Injection: A laser is one efficient and precise method to inject faults into the ICs. A laser that interacts with layers that are silicon or metal creates photoelectric or thermoelectric effects, which can be exploited to inject faults.

From the backside of IC, when a laser beam with a high energy wavelength passes through silicon, it creates electron-hole pairs (EHPs) along the path. Most of these EHPs recombine together, having no impact on the IC. However, under the influence of a strong electric field of reverse-biased PN junction, these EHPs drift in opposite directions, causing a current pulse. These current pulses are transient and exist for a few nanoseconds after stopping laser injection. These transient currents can turn on the reverse-biased junctions, causing the charging/discharging of the capacitive load. The voltage change in the capacitive load causes faults, also known as *single-event transient* (SET). A laser can also induce faults directly in the memory elements (e.g., RAM and registers), leading to a *single-event upset* (SEU).

When the IC is exposed to the laser from the front side, it can hit the metal layers or find gaps to reach the active region. If the laser penetrates the active region, the observed effect is equivalent to backside exposure. However, modern ICs have dense metal layers, therefore, the laser most exposed from the front side interacts with the metal layers. A laser interacting with the metal layers is either reflected or absorbed and converted to heat. This heat can diffuse into the PN junction creating a thermoelectric effect. However, depending on the number of metal layers in an IC, an indirect heating effect from top layers towards bottom layers can be observed. Furthermore, if there are many metal layers, the heat generated may not be sufficient to diffuse in the PN junction and have any noticeable effect.

If the IC's backside is exposed, an attacker can target specific regions on the chip to expose to the laser [112], [113]. Thus, an attacker can inject faults in precise spatial locations and control the laser exposure time to have temporal controllability on the attack. These combinations make the LFI attacks lethal and can break crypto and secure systems to violate integrity and confidentiality. LFI can inject faults during regular crypto operations, causing faulty outputs. Faulty outputs can be used with differential fault analysis to guess the secret key used during the operations. Similarly, LFI can be used to inject fault to gain access to unauthorized functionality of the design or leak secrets about the machine learning/AI models. LFI attacks on crypto designs, such as AES and smart cards, have been demonstrated predominantly in the literature [114], [115], [116], [117]. [118] and [119] demonstrated that LFI could be used to gain unauthorized voice-control access to the system and bypass secure-boot on smartphones. Similarly, fault assessments on neural networks have been presented to reverse engineer the model architecture or leak weights or biases [120].

2) Timing Fault Injection: Fault injection through glitch (e.g., clock, voltage, or temperature) is a global attack that is non-invasive and less expensive to perform. Inserting a glitch can cause timing violations at the target registers and can inject faults.

If a glitch is introduced at a specific clock cycle, it reduces the effective period of that particular clock cycle. Data propagating through a combinational logic must be stable inside the setup-time or hold-time margin to prevent a flop from going into a meta-stable state while latching the data at its output. Therefore, data must be latched without any timing violation, and the data transition must occur with sufficient positive slack from both the setup-time and hold-time boundary. However, shortening the time period through glitch injection reduces the time for data to propagate through combinational logic and can violate the setup-time or hold-time constraint (shown in Fig. 4). Injecting spikes in power signals to reduce the supply voltage temporarily increases the propagation delay of a conbinational logic. It causes the datapath delay to increase and can violate the setup-time constraints as well (shown in Fig. 5). Overheating can also increase the datapath delay and cause timing violations.These timing violations are responsible for causing bit flip at the output of the flop (single-event upset), which can be termed as a *timing fault injection* (TFI) [59], [60].

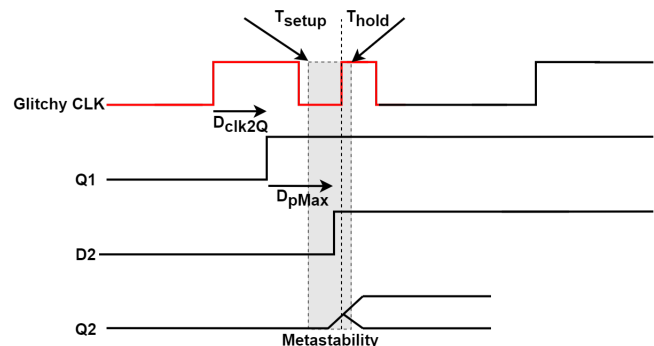

Fig. 4. Timing violation through clock glitch.

An attacker can tamper a clock port or power distribution network of a design to inject timing faults at some specific registers. As this is a global fault injection technique, the attacker cannot control the location of the fault injection. Nevertheless, the fault injection time can be controlled precisely. If a TFI can be successfully propagated to an observable output (exploitable fault injection), it can break crypto modules and secure designs by leaking secrets. The faulty outputs can be exploited along with fault analyses (e.g., differential fault analysis [121], fault sensitivity analysis [122], differential fault intensity analysis [123], etc.) to extract the secrets. Addition-

ally, a TFI glitch can be used to violate the security properties related to the confidentiality, integrity, and availability of the assets of a design. Plenty of successful TFI attacks through glitches have been well demonstrated in various research works [50], [47], [48], [49], [44], [45], [52], [53].

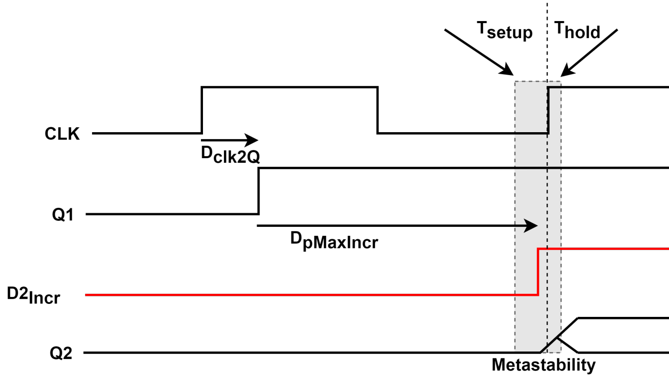

Fig. 5. Timing violation through voltage glitch/overheating.

3) EM Fault Injection: In 1831, Michael Faraday found that a current-carrying coil can induce current in a nearby coil. Similarly, strong electric/magnetic fields can induce currents in loops within an IC chip. In modern chips, interconnect wires on one end are connected to the high-resistance CMOS gates, creating no impact due to EMFI [124]. On the other hand, routing power grids and rails form a large network of low resistive loops in the chip. Therefore, one can observe voltage drops and ground bounces on power rails in the presence of electric/magnetic fields. These power noises can impact the signal propagation to inject timing faults. The observed impact is similar to that of voltage glitching, however, due to the feasibility of generating targeted EM waves, EMFI can be local to achieve single-bit faults.

V. Secure Physical Design Flow for Possible Vulnerability Prevention

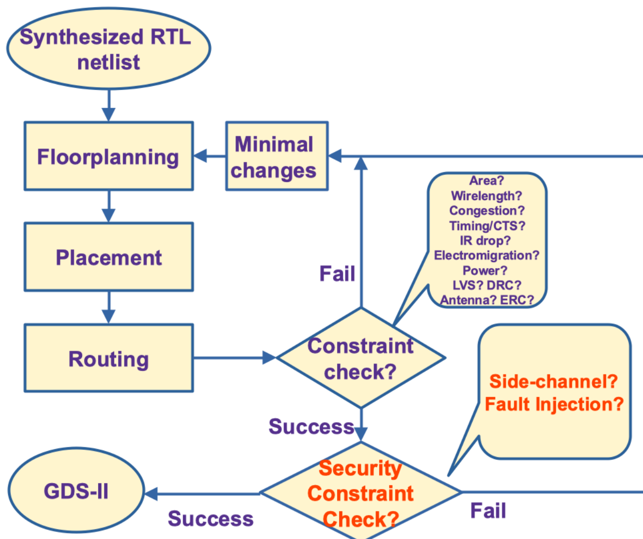

Fig. 6. Traditional physical design flow with security constraint check.

A. Security Rules for Physical Design

One of the important steps to establish a secure physical design is to have a database of rules that defines various vulnerabilities and its countermeasure. Based on the physical design level countermeasures and after performing experiments, several security rules are created. These security rules need to be implemented in P&R tools in order to ensure security, and checks need to be performed, similar to design rule checks (DRC) for physical layouts. Therefore, our proposed flow will have an extra level of security verification compared to the traditional physical design flow, as shown in Fig. 6. For example, to prevent the effect of coupling capacitance caused by tightly placing logically independent shares (mentioned in Section IV-A), which can violate an independent condition required in a side-channel resistant masking scheme, the shares should be placed separately not to influence each other. In this case, we can define a *Security Rule: the logically independent shares in a masking scheme should be placed far apart*. Suppose that the security checker finds the rule violation. In that case, the placement tool will change the location of the share logic cells without the violation and with little cost of performance and area. Table V shows the summary of the security rules according to power/EM side-channel vulnerabilities and fault-injection vulnerabilities at the physical design phases.

B. Implementation of Secure Physical Design Flow

In this section, we describe two ways of implementing secure physical design and formulate the problem of secure physical design. Security rules are discussed in the previous section. Traditional P&R-based CAD tools do not have those security rules. Security rules must be incorporated in P&R-based CAD tools to obtain secure physical design flow. Based on the source code availability of CAD tools, we can adapt the following approaches for realizing the secure physical design flow.

1) With Help of Open-source CAD Tools: There are various open-source CAD tools and frameworks that deal with RTL to GDSII flows, like OpenROAD [126]. Also, some open-source tools perform the internal steps of physical design quite well, which include NTUPlace [127], RePlace [128], and DREAMPlace [129] for global placement, and NTUGr[130] for Global Routing. Since the source codes of the open-source tools are available to the public, the source code can be modified, and the security rules can be applied to establish a secure physical design flow using the open-source tools.

2) With Help of Commercial CAD Tools: The source codes of commercial CAD tools are not available. Therefore, we have to implement security rules as design constraints to restrict the layout designs to obtain desired security closure in commercial CAD tools.

Based on the discussion above, we can formally define our problem statement of a secure physical design as the following:

Problem 1: Suppose there is a regular layout $\boldsymbol{L}$ generated by the P&R tool. The regular layout may have some vulnerabilities. In that case, a security verification engine needs to

TABLE V
SECURITY RULES AND COUNTERMEASURES FOR SECURE PHYSICAL DESIGN.

Security Vulnerability	Physical Characteristic	Security Rule	Implementation Stage
Power-side-channel	Minimize the IR drop of critical instances (s-box cells)	Place critical instances closer to decaps or power stripes	Placement
	Balance power consumption of shares	Place shares at equidistant from decaps and power stripes	Placement & Routing
	Interference from share placed to each other	Place shares far away from each other	Placement
EM-side-channel	EM signature related to the secret asset should not reach the topmost metal layer.	Local low-level metal routing with a SAH	Routing
Laser Fault Injection (Frontside)	A laser should not be able to penetrate to active region	Shield cells with metal layers	Routing
Laser Fault Injection (Backside)	Critical cell placed nearby high switching & bulky cells, are more susceptible	Metal shielding critical cells from nearby high switching & bulky cells	Routing
	Protecting active region from direct intensity of laser	Backside Buried Metal meander[125]	Placement and Routing
	Bulky cells are less susceptible to laser compared to small cells	Replace critical cells with bulky cells	Placement
Timing Fault Injection (clock/power/voltage glitch, voltage)	Faults injected and propagated by clock glitch, voltage glitch, power glitch can impact path delays	Placement & Routing of cells should be done in such a way to minimize path delays and clock skew	Placement and Routing
EM fault injection	Mitigate induced eddy currents	Connected loop structures should be avoided near critical cells	Placement and Routing

be run to find the vulnerabilities. If a vulnerability exists, then changes need to make to this layout to minimize the security constraint violations and obtain a new secure layout L_{secure}.

We believe that AI/ML will help us to solve this problem and to achieve a secure physical layout, which can be best described as shown in Fig. 9. We have presented a detailed AI/ML roadmap for a secure physical design in Section VI. Before going to the AI/ML roadmap, let us discuss our proposed flow for secure physical design and secure physical verification sign-off.

C. Proposed Flow for Secure Physical Design (SEPHYD) and Secure Physical Design Verification (SPDV) Sign-off

Our proposed flow for secure physical design and secure physical design verification sign-off is shown in Figure 7. The main components of our proposed flow are listed below,

- **Input**, the input to our flow can be either gate-level netlist or a physical design.
- **P&R tool**, if the input is in gate-level netlist, generate corresponding regular layout from the netlist. We use commercial P & R Tool such as Cadence Innovus [131] or Synopsys ICC2 [132] for generating layout. Or we can also employ OpenROAD [126] for generating the regular layout from synthesized netlist.
- **Secure Physical Design Verification (SPDV) tool**, which evaluates several physical vulnerabilities based on the various security metrics described in Section V-D.
- **Security Metrics**, these are same security metrics mentioned in Section V-D.
- **Security Rule Check database**, which basically contains several secure P&R recommendation as mentioned in Section V-A. Some of the preliminary Security Rules are listed in Table V.
- **SEPHYD tool**, takes a gate-level netlist as its input (for the first time) and generates a secure physical design based on several security rules. These security rules are developed from several physical design level countermeasures (mentioned in Section V-E and V-F), which in turn are constructed to ensure security from any physical vulnerabilities. For applying countermeasures in terms of security rules and to bring automation, it employs AI/ML approaches. A detailed roadmap of the AI/ML approach for realizing SEPHYD tool is described in Section VI.
- **Output**, a secure physical design free from any physical vulnerabilities.

In the proposed flow, we consider the input gate-level netlist is secure where all high-level security measures are implemented. However, when the physical design is created from this secure gate-level netlist using the P&R tool, some security vulnerabilities are unintentionally injected, which some adversaries can exploit to compromise the secure design. Such threats can occur only at the physical design level, as much information is not available in the logic synthesis stage. For example, power grid network and clock tree synthesis information can only be obtained in the physical design stage.

Once the layout is generated using the P&R tool, we employ our security verification engine (SPDV tool) in order to verify the security of the regular layout. The security verification is performed by evaluating several security metrics depending on the vulnerabilities mentioned in Section V-D. If the security metrics report that there is any vulnerability in the layout, then it goes through the *SEPHYD* tool, which performs minimal change in the layout and reconstructs the layout considering the security rules discussed in Section V-A. In this way, we obtain a secure physical design. To ensure that generated physical design from the SEPHYD tool is actually secure, we again verify our generated physical design using our SPDV tool. If the verification engine reports that no physical vulnerability is present in the layout, we sign off our secure physical design and verification flow. Therefore, our proposed flow goes into a loop of several iterations between the security verification engine (SPDV tool) and SEPHYD tool, until the security metrics suggest that the desired level of security has been achieved. Once the security metrics suggest an acceptable level of security of physical layouts, we say that we have obtained the secure physical design.

For our initial experiments, SEPHYD tool is being implemented using a heuristic-based AI/ML approach. More about this is discussed in Section VI. We are also exploring data-driven AI/ML techniques to implement SPDV tool and SEPHYD tool.

D. Security Metrics for Evaluating Physical Vulnerabilities

The metrics used for evaluating the security vulnerability is described here. At present, we have defined preliminary metrics for evaluating Side-channel vulnerability and Timing Fault-injection metrics.

1) Side-channel vulnerability metrics: SEPHYD framework performs the side-channel vulnerability analysis based

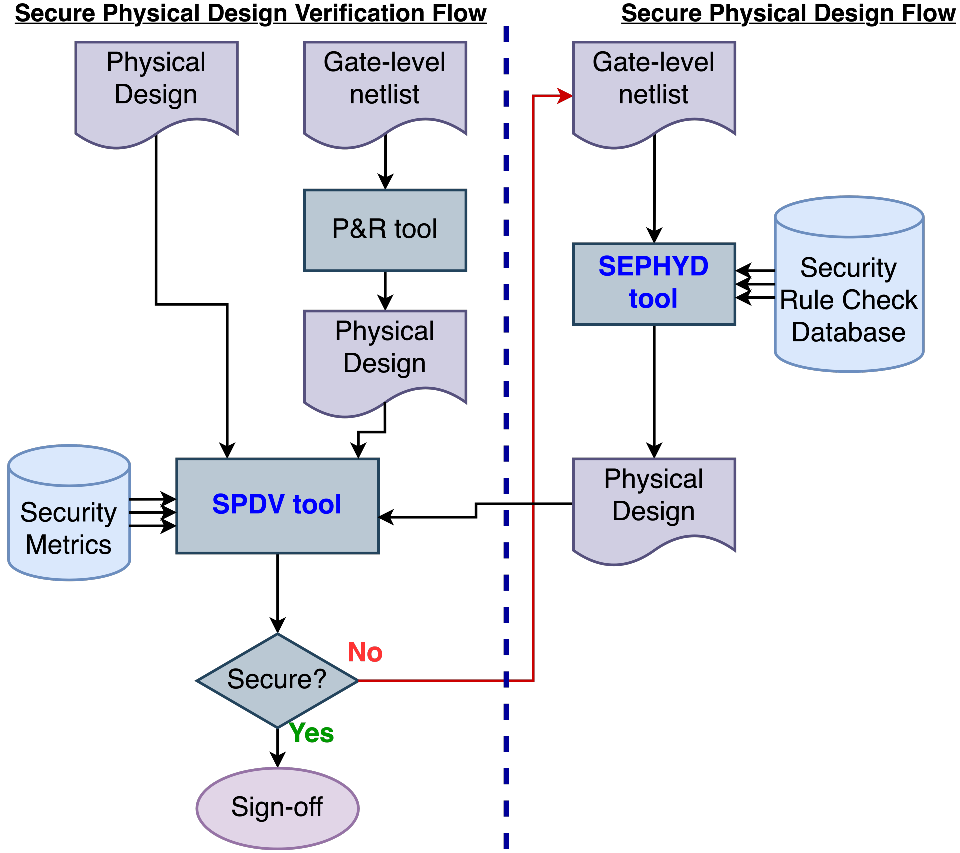

Fig. 7. Proposed flow for secure physical design (SEPHYD) and secure physical design verification (SPDV) sign-off.

TABLE VI
KL DIVERGENCE THRESHOLD FOR DIFFERENT FAILURE PROBABILITIES ($\mathbf{Pr}$).

Pr	KL	Pr	KL
>0.96	<0.01	>0.53	<0.78
>0.90	<0.03	>0.45	<1.12
>0.80	<0.12	>0.38	<1.53
>0.71	<0.28	>0.32	<2.00
>0.61	<0.50	>0.26	<2.53

on the Kullback-Leibler (KL) divergence metric. Details on the KL divergence metric and its relation with the attacker's success rate (SR) based on maximum likelihood estimation are described as follows.

Let $f_x(z)$ and $f_y(z)$ be the probability density functions of random variables X and Y. KL divergence is defined as the following equation:

$$D_{KL}(X||Y) = \int f_x(z) log \frac{f_x(z)}{f_y(z)} dz. \quad (1)$$

If X and Y are of normal distributions with means (μ_x, μ_y) and variances (σ_x^2, σ_y^2), then the KL divergence equation can be simplified as follows:

$$D_{KL}(X||Y) = \frac{(\mu_x - \mu_y)^2 + \sigma_x^2 - \sigma_y^2}{2\sigma_y^2} + ln(\frac{\sigma_y}{\sigma_x}). \quad (2)$$

The KL divergence is expected to be high if power leakage probability distributions for two different keys are distinguishable, meaning an adversary can easily correlate the power consumption between the keys. Hence, the maximum KL divergence for allowable failure probability (Pr) can be obtained, where failure probability is the adversary's probability of an incorrect inference based on PSC attacks. The higher the failure probability is, the more secure the design is against PSC attacks. For example, if we want the failure probability of more than 0.9, the KL divergence should be less than 0.03 [133]. Table VI provides the required KL divergence threshold for different failure probabilities (Pr).

Also, to assert with a high confidence level of $100(1-\alpha)\%$ that the two Gaussian distributions X and Y differ, it is necessary to account for the number of traces N in the KL divergence metric. The number of traces N contributes significantly to quantifying a lower constraint on side-channel attack complexity in terms of the required number of power traces. The smallest number of traces to satisfy that $\Pr[|\bar{X} - \bar{Y} - (\mu_X - \mu_Y)| < \epsilon] = (1-\alpha)$ is

$$N \geq \frac{(\sigma_X + \sigma_Y)^2}{\epsilon^2(\mu_X - \mu_Y)^2} \dot{z}_{1-\alpha/2}^2, \quad (3)$$

where the quantile $z_{1-\alpha/2}$ of the standard normal distribution has the property that $\Pr[Z \leq z_{1-\alpha/2}] = 1 - \alpha/2$.

One can correlate the KL divergence analysis with the probability of an attacker's success in leaking the key. Assuming for a given key K, the probability density function of the

switching activity T follows a Gaussian distribution, then SR could be derived as follows:

$$f_{T|K}(t) = \frac{1}{\sqrt{2\pi}\sigma_k} e^{-\frac{(t-\mu_k)^2}{2\sigma_k^2}} \tag{4}$$

where μ_k and σ_k^2 are the mean and variance of T, respectively. The likelihood function is defined as $\mathcal{L}(k;t) = \frac{1}{n}\sum_{i=1}^{n} \ln f_{T|K}(t_i)$. Based on the maximum likelihood estimation, an adversary typically selects a guess key $\hat{k}$ as follows:

$$\hat{k} = \arg\max_{k \in K} \mathcal{L}(k;t) = \arg\max_{k \in K} \frac{1}{n}\sum_{i=1}^{n} \ln f_{T|K}(t_i) \tag{5}$$

If the guess key ($\hat{k}$) is equal to the correct key (k^*), the side-channel attack is successful. Thus, the attacker's success rate can be defined as follows:

$$SR = \Pr[k_g = k^*] = \Pr[\mathcal{L}(k^*;t) - \mathcal{L}(\langle \bar{k^*} \rangle;t) > 0] \tag{6}$$

where $\langle \bar{k^*} \rangle$ denotes all wrong keys, i.e., the correct key k^* is excluded from $\mathcal{K} = \{k_0, k_1, \ldots, k_{n_k-1}\}$, where n_k is the number of all possible keys.

The mathematical expectation of $\mathcal{L}(k^*;t) - \mathcal{L}(k_i;t)$ in Equation (6) is equal to KL divergence between $T|k^*$ and $T|k_i$ [134]. It shows that both SR and KL divergence are closely related and follow a similar trend. Therefore, leakage obtained through KL analysis can be associated with an attacker's success in leaking the key.

2) Timing Fault-injection security metric: We define a security metric, V_{TFI}, that quantifies the susceptibility of a design against TFI attacks [135]. It is proportional to the probability of vulnerability to DFA attacks introduced by timing faults, P_{sp} for all the security properties of a crypto-design against DFA. It is also inversely proportional to the number of security properties defined in the database, $N_{DB_{sp}}$ since the attacker has to deal with more options. Here P_{sp} depends on the probability of finding feasible fault locations and the probability of the dispersion of the security critical paths within the timing margin. The following equation represents the mathematical expression of the security metric.

$$V_{TFI} = \frac{1}{N_{DB_{sp}}} \sum_{p=1}^{p=N_{DB_{sp}}} P_{sp} \tag{7}$$

A larger value of P_{sp} of Eq. 7 means a higher probability of security property violations due to DFA attacks, and ultimately a higher susceptibility to DFA attacks.

E. Possible Countermeasures for Side-channel Vulnerabilities

Based on the violation of the defined security rules (shown in Table V), suitable countermeasures can be applied at the physical design phases as follows:

- *Minimizing IR drop*: IR drop or power supply noise is caused by the finite resistivity of each metal layer in the PDN. Instantaneous power consumption of each share in a masking scheme can be affected by adjacent shares significantly if not considering the IR drop effect. The IR drop should be minimized by inserting a sufficient number of de-cap cells or reconstructing the PDN to remove power dependency between logically independent shares. Interconnect width optimization-based approaches can also be implemented to minimize IR drop [136], [137], [138].
- *Separation of independent shares*: Due to capacitive coupling between two adjacent wires, when a wire switches a value, another wire can be influenced by the inter-wire capacitance, called crosstalk. Logically independent shares should be placed with enough distance to remove the capacitance coupling between them [97].
- *Local low-level metal routing with a signature attenuating hardware (SAH)*: Das *et al.* [139] proposed a signature attenuation method with local low-level metal routing and a signature attenuating hardware (SAH), called *STELLAR*. The SAH significantly suppresses the EM/power signature with low overhead before it reaches the top metal layer of the chip that has the most contribution to measurable EM side-channel leakage. This method protects the AES-128 encryption against EM/power SCAs and achieves $Measurements\ to\ disclosure\ (MTD) > 1M$ with a low-overhead physical countermeasure ($1.5\times$ power and $1.23\times$ area overhead).
- *Supply isolation*: The critical power signature of cryptographic modules can be isolated from the external supplies, which eliminates power side-channel information measured by adversaries. For example, a switched capacitor current equalizer isolates critical activities by equalizing currents into the cryptographic module [140]. The capacitor current equalizer consists of an array of capacitor circuits, and each circuit has three different switching states: S1-charge the capacitor from the supply, S2-provide charge to the cryptographic module, and S3-discharge the capacitor to a pre-programmed value. The charged capacitor serves as a voltage source for the cryptographic module. Since the capacitor is disconnected from the external supply during cryptographic operations, power signatures of the cryptographic module can not be measured. Three independent capacitor modules do not overlap switching states for uninterrupted operations of the cryptographic module.

The first and second countermeasures require a masking technique at the gate level, consisting of multiple logically independent shares. In contrast, the third and fourth approaches do not require any approaches at the higher level of design abstract. However, there is a need for a specific attenuating circuit with a routing method and a current equalizer at the physical level. Fig. 8 shows the possible countermeasures against side-channel vulnerabilities at the physical design phase.

F. Possible Countermeasures for Fault Injection Vulnerabilities

For protecting against fault injection vulnerabilities, various countermeasures could be applied based on logical or physical

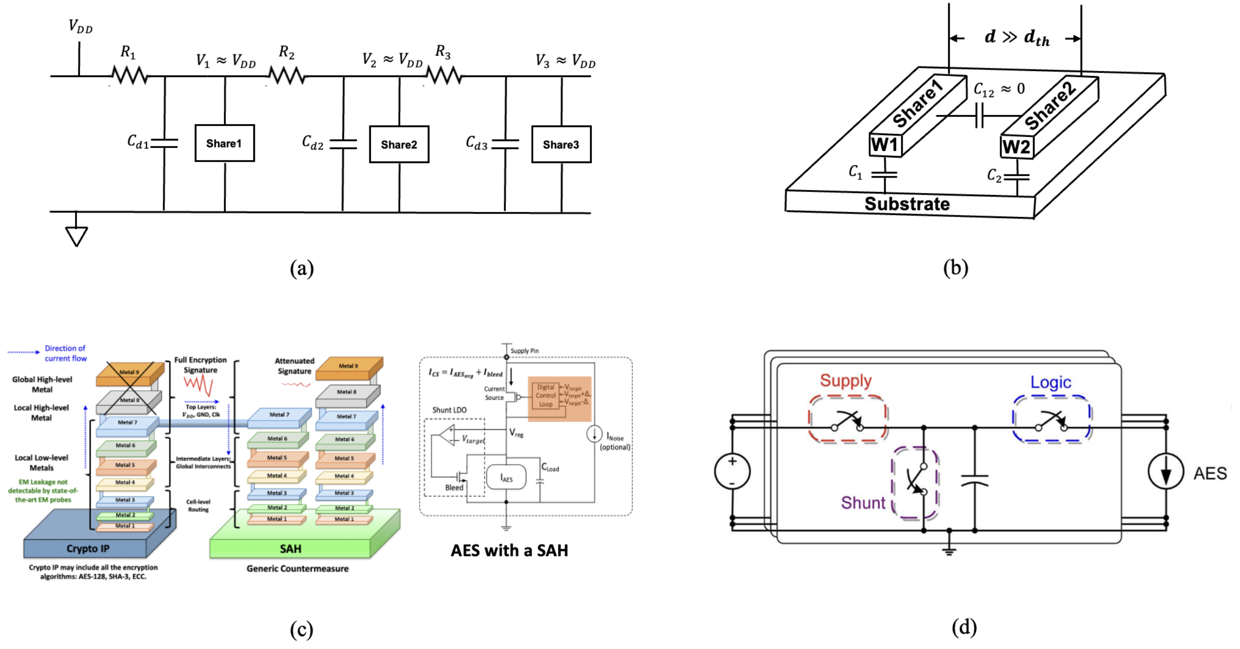

Fig. 8. SCA Countermeasures at the physical level: (a) Minimizing IR drop using de-cap cells; (b) Separation of independent shares; (c) Local low-level metal routing with a SAH [139]; (d) Supply isolation [140].

parameters. Some of these countermeasures or parameters are discussed below.

- *Spatial or temporal redundancy:* Most prominent FI countermeasures include creating redundancies in the design to exploit the fact that it is difficult to precisely inject faults in multiple redundant circuits at the same location and time.
- *FI standard cell library:* Different cell parameters, such as size, threshold, cell layout, etc., can play a significant role in determining the design's resiliency against FI vulnerabilities. For example, changing the gate size impacts the path delays and the cell's driving strength. A high threshold of the cells may require higher laser energy to flip the output. Similarly, the layout of a cell (sharing of p/n-well regions, poly widths, etc.) can impact how a cell behaves under different fault injection methods.
- *FI-aware layout design:* Standard cells in addition to different physical parameters can also impact the design's resiliency against fault injection methods. For example, power distribution networks, clock tree synthesis, placement, routing, and other factors can be customized to account for fault injection security.

VI. AI/ML Roadmap for Secure Physical Design and Verification Sign-off

Recently, AI/ML has attracted the attention of chip design community [141]. Google has also claimed successful deployment of reinforcement learning (RL) for macro placement [142] in its AI processor design. Although RL-based solution of Google has not provided good results for standard-cell placement yet [143]. There are several other works on machine learning (ML) in applications of CAD domain [144], [145], [146], [147], [148], [149]. We also believe ML can be implemented in order to obtain a secure physical design. Here, we present the AI/ML roadmap for obtaining secure physical design. Initially, we describe AI/ML approach for developing our SEPHYD tool. Later on, we discuss AI/ML approach for developing our SPDV tool. Overall, the final goal is to implement the proposed flow of Fig. 7 using AI/ML automation.

A. AI/ML Roadmap for SEPHYD tool

Here, we describe AI/ML roadmap to implement the proposed SEPHYD tool. When we say SEPHYD, it basically means secure floorplanning, secure placement and secure routing. Here, our discussion is only limited to floorplannning and placment. However, similar approach can be adapted for secure routing. We have already observed one such vulnerability in floorplanning and placement stage using the timing fault injection parameters. The delay distributions of critical datapath vary widely in the gate and physical layout levels. Adversaries can utilize these vulnerabilities to inject fault and extract secret information from the chip. Initially, we propose a gate sizing-based solution to mitigate delay variation-based vulnerabilities [135]. However, we observe that this vulnerability can be mitigated using the AI/ML approach described below.

Suppose, we have n critical paths. Let the path delays of critical paths in the gate-level be $t_{g_i}\ \forall\ i \in \{1, 2, \cdots, n\}$ and path delays of critical paths in physical design-level be $t_{pd_i}\ \forall\ i \in \{1, 2, \cdots, n\}$. The absolute path delay difference between gate-level and physical design level can be represented by

$$\Delta t_i = |t_{g_i} - t_{pd_i}|\ \ \forall\ i \in \{1, 2, \cdots, n\}. \qquad (8)$$

We have the values of t_{g_i} during the gate-level simulations, which can't be changed during the physical design stage.

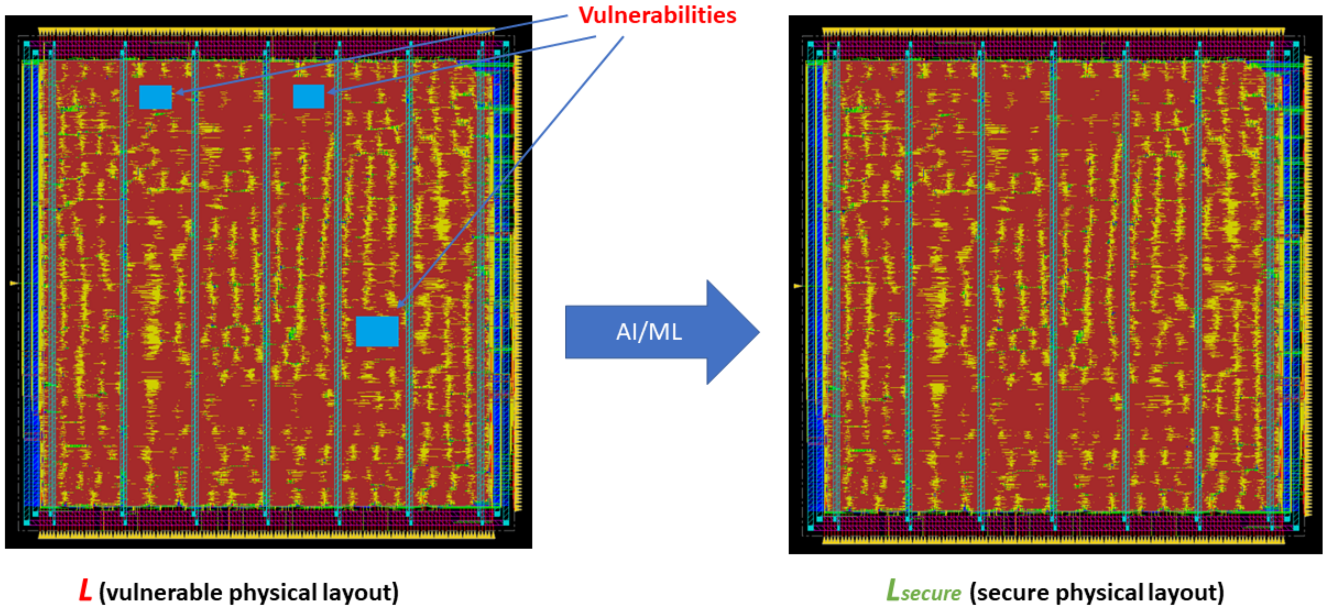

Fig. 9. Objective of AI/ML roadmap to generate secure physical layout from regular layout.

Therefore, the objective is to place the standard cells connecting critical paths in such a way during the physical design stage that Δt_i becomes zero or minimum. However, if consider Δt_i as the only objective, and perform minimization, then the total wirelength and congestion may increase. Therefore, along with Δt_i, we can also consider the traditional placement metrics, such as total wirelength and congestion in our cost function. By considering all of these objectives, our cost function for Timing Fault Injection-Aware secure placement can be written as,

$$min \ \ (\sum_{e \in E} WL(e; x, y)) + \lambda D(x, y) + \Delta t_i, \tag{9}$$

where $(\sum_{e \in E} WL(e; x, y))$ represents total wirelength, $D(x, y)$ represents density [142] and Δt_i represents delay differences between gate-level and physical design level $\forall i \in \{1, 2, \cdots, n\}$ critical paths. The challenge is to obtain the locations of the standard cells, where Δt_i becomes zero or minimum, which is a NP-hard problem, as solution space of this problem is of exponential order. That is where we feel several AI/ML techniques can be implemented in order to obtain a near-optimal solutions for our TFI-Aware secure placement problem. Similarly, for other kinds of physical design level vulnerabilities, the cost function becomes as,

$$min \ \ (\sum_{e \in E} WL(e; x, y)) + \lambda D(x, y) + \Phi_i, \tag{10}$$

where Φ_i represents the security metrics for the physical vulnerabilities in concern. Therefore, it is necessary to define the security metrics to evaluate the vulnerabilities. Subsequently, optimize the cost function that also includes the security metrics. To find solutions of such cost functions, we can apply two classes of AI/ML approaches. The first class is a heuristic-based approach and the second class is a data-driven approach. Since we have already mentioned about TFI vulnerability in (9), rest of our discussion is limited only for TFI vulnerability. However, the same approaches can be extended to any other kinds of physical vulnerabilities.

1) Heuristic-based Approach: In heuristic-based approach, based on some predefined cost functions, agents perform search in the solution space to find near-optimal solutions, satisfying the cost functions and associated constraints. For our problem, we change the floorplanning and placement of standard cells to reduce the vulnerability by employing heuristics. We employ (9) as a cost function and subsequently apply well-known heuristic approaches, (e.g., simulated annealing [150], genetic algorithm [151], or gradient-descent search [152]). These heuristics interact with each other to produce near-optimal solutions. The heuristics start from random solutions and gradually improve the quality of solutions to converge in a desired level of tolerance for the cost function. These heuristics are slow, as it takes a considerable time to converge. However, we anticipate to obtain TFI-aware secure macro and standard cell locations using these heuristics.

2) Data-Driven Approach: When we talk about data-driven approaches it basically means traditional machine learning approaches, recent deep learning approaches, and deep reinforcement learning approaches. Each of these techniques employ datasets to find near-optimal solutions. One important aspect of data-driven approaches is to have a proper dataset. However, unlike the computer vision domain, there is no public open-source dataset available as of now, for developing machine learning models for electronic design automation (EDA). This leads us to generate our own dataset and perform proper feature engineering to find suitable features to develop machine learning models. Currently, we are creating datasets following the format mentioned in Table VII and Table VIII. Basically, using the dataset of Table VII and Table VIII, it is possible to define a floorplan with connections among macros

and standard cells. Therefore, we engineer these datasets to obtain a suitable machine learning model. Now we will describe various data-driven approaches, which can be viable options for solving the problem of security-aware floorplanning and placement.

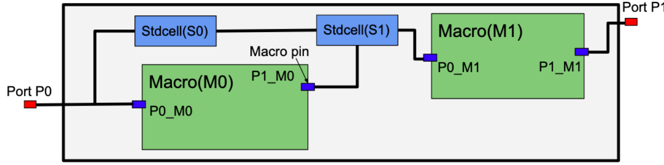

Fig. 10. Example of a chip floorplanning with macros and standard cells.

TABLE VII
POSITION DATASET FORMAT FOR THE SAMPLE FIG. 10

Item name	Location	Length	Height
P0	(xP0,yP0)	–	–
S0	(xS0,yS0)	lS0	hS0
M0	(xM0,yM0)	lM0	hM0
S1	(xS1,yS1)	lS1	hS1
M1	(xM1,yM1)	lM1	hM1
P1	(xP1,yP1)	–	–
P0_M0	(xP0_M0, yP0_M0)	–	–
P1_M0	(xP1_M0, yP1_M0)	–	–
P0_M1	(xP0_M1, yP0_M1)	–	–
P1_M1	(xP1_M1, yP1_M1)	–	–

TABLE VIII
EMPTY TABLE ENTRIES REPRESENT THE MANHATTAN DISTANCES BETWEEN ITEMS PRESENT IN THAT ROW AND COLUMN. FOR EXAMPLE, $md(i, j)$ REPRESENTS MANHATTAN DISTANCE BETWEEN S0 AND P1. THE DIAGONAL ENTRIES ARE ZEROES.

Item name	P0	S0	M0	S1	M1	P1	P0_M0	P1_M0	P0_M1	P1_M1
P0	0									
S0		0				$md(i, j)$				
M0			0							
S1				0						
M1					0					
P1						0				
P0_M0							0			
P1_M0								0		
P0_M1									0	
P1_M1										0

a) Supervised Learning: For supervised learning, relations between dependent and independent variables need to be established. For the delay variance as a timing fault-injection parameter, we can label the locations of macros and standard cells to distances/datapath delays to create a dataset. After that, we can train a neural network and generate a model to deal with vulnerabilities. This is the supervised regression problem, and the prediction accuracy needs to be measured using Mean-Squared Error (MSE). However, from the understanding of the TFI-aware secure placement, MSE can be very high, as this approach may not produce the refined level of placement desired.

b) Sequential Supervised Learning: The task of TFI-aware secure placement is sequential in nature. Because the position of second macro depends on the first macro's position, in order to reduce the vulnerability and optimize other placement objectives. Therefore, we can employ several sequential machine learning techniques, such as Sliding Window, Recurring Sliding Window, Hidden Markov Model, Maximum Entropy Markov Model, Input-Output Markov Model, Conditional Random Field, and Graph Transformer Networks [153]. Accordingly, we perform engineering using the datasets mentioned in Table VII and Table VIII.

c) Unsupervised Learning: Unsupervised learning works by finding patterns in an unlabelled dataset. This ML approach may not be a good candidate for this particular task, as TFI-aware secure macro or standard cell placement task doesn't rely on finding any pattern. However, more research and development is required before giving any final verdict about this approach.

d) Semi-supervised Learning: In this case, we must have a mix of labelled and unlabelled data, with which predictions are performed. Subsequently, this technique may present viable options to predict secure placement or to solve the cost function of concern. However, more research and development is required before giving any final judgement about this approach.

e) Reinforcement Learning Approach: The reinforcement learning approach would be a suitable choice for the practical implementation of the TFI-aware secure macro and cell placement task, as this approach depends solely on the reward of agents. The reward is high for more secure positions and the reward is less for less secure positions. Already, Google has received success in their macros placement task [142]. However, Google's work doesn't consider the security aspect while performing the placement. The cost function of (9) needs to be addressed using deep reinforcement learning with appropriate reward, state definition, and policy function to have TFI-aware secure placement. Again, we need the datasets mentioned in Table VII and Table VIII if we want to develop deep reinforcement learning-based solutions.

Overall, the goal is to develop a single security metric (Φ) which can evaluate the security measures of all the vulnerabilities using a single cost function, instead of having separate cost functions for each vulnerability. Subsequently, we can solve the cost functions using the AI/ML approaches mentioned above. Likewise, we can also employ these AI/ML approaches to generate secure routing in the physical design. We discuss the AI/ML roadmap for side-channel assessment as part of our SPDV tool in the next section.

B. AI/ML Roadmap for SPDV tool

Here, we describe a roadmap for developing SPDV tool using AI/ML approach. Our description mostly focused on adopting AI/ML approach for side-channel vulnerability assessment and verification as a case study. Similarly, AI/ML approach can be implemented for other types of physical vulnerabilities.

Side-channel analysis has seen remarkable growth in the last five years because of the inclusion of machine learning-based approaches for side-channel vulnerability assessment. The practice has been noticed only in post-silicon side-channel analysis, though the pre-silicon analysis is essential, especially at the physical design level. As a roadmap for side-channel analysis at the physical layout level, we propose

Fig. 11. Roadmap for AI-based side-channel assessment

two approaches: estimation and measurement. To elaborate, we define the estimation approach as a quick way to assess the side-channel vulnerability at the physical layout. The estimation approach will completely avoid any simulation, i.e., no EDA tool will be used in this approach for calculation. Tools can only give physical information. All these constraints are applied to make the process fast for deciding whether a physical design is vulnerable or not.

The measurement approach will be based on simulation. However, the measurement approach will give a more robust decision about side-channel vulnerability. No constraint is imposed upon this method like estimation. Since EDA tools will be used for power calculation, this approach will be time-consuming. The measurement approach starts with collecting power side-channel traces at the layout level of a full-blown SoC design through dynamic power analysis using EDA tools. A corresponding power trace is collected for a random plaintext and random key in each simulation. The vulnerability of the physical design can be assessed by performing a side-channel attack on these collected traces. A portion of the collected traces is used for training the neural network, and the attack is performed on the rest. The quality of the attack is evaluated through performance metrics like guessing entropy, success rate, etc. If the required number of traces to recover a key is less than a predefined threshold value, the design is vulnerable.

In this work, we provide a roadmap for AI-based side-channel assessment dor measurement approach. The roadmap is shown in Figure 11. At first, the collected power trace is processed through the data pre-processing step. The most common practice of data pre-processing is to apply the typical normalization or standardization technique. Additionally, signal decomposition techniques such as empirical mode decomposition (EMD) [154], Hilbert vibration decomposition (HVD) [155], variational mode decomposition (VMD) [156], and other techniques can be applied. Such decomposition techniques depict the intrinsic features unseen in raw signals. Notably, EMD can be used as a denoising step to remove unnecessary information from the power traces. In the case of a hiding countermeasure, dynamic time warping (DTW) is an effective way to bring alignment into the misaligned traces. Data augmentation is another data processing technique that can improve the quality of an attack by including newer observations of data in the training set and preventing overfitting.

The roadmap suggests several ways to perform appropriate feature selection with the help of a feature checker. The purpose of a feature checker is to ensure the choice of selected feature set is correct. The feature checker can be formed in different ways. One way is to calculate feature importance using a Random Forest classifier. In this case, the relative feature importance of each type of feature can be calculated in terms of Gini impurity. The features also show that higher feature importance can be kept, and other features can be discarded. Correlation analysis can be also a viable way to find out efficacy of a set of features. After selecting the appropriate features, one must choose a neural network. Convolutional neural networks (CNN) and multilayer perceptron networks are the most widely used neural architectures for side-channel analysis. Zaid *et. al* [157] discovered efficient but shallow convolutional neural network (CNN) architectures through weight visualization, gradient visualization, and heatmaps. However, with such techniques, achieving appropriate hyperparameter tuning can be challenging. Wu *et. al* [158] showed how automated hyperparameter tuning using Bayesian optimization assists in performing superior side-channel attacks. Perin *et. al* [159] preferred ensemble models to the single best performance for improving the generalization of the attack. Integration of automated hyperparameter tuning and model ensembling to the framework can boast the performance of the neural network. Using proper evaluation metrics to assess the performance of the neural network architecture is another important step. Guessing entropy and ranking loss are the top evaluation metrics for such task.

VII. Open Challenges

In the previous sections, we have proposed several possible countermeasures to mitigate physical design level vulnerabilities, including an AI/ML roadmap. However, many challenges exist and it may be difficult to create countermeasures efficiently. These challenges are described in the following sections.

A. Challenges in Developing Physical Design Security Metrics

At present, there are no properly defined metrics that can evaluate the security of a physical design for any specific kind of vulnerability. To progress in secure physical design research, it is necessary to define secure physical design metrics so that we can evaluate how secure a physical design is. Once we have well-defined security metric, the objective of the physical design steps would be to optimize the layout in order to minimize the security metric and optimize other physical design constraints at the same time.

B. Challenges in Vulnerability Realization

1) Challenges in Side-Channel Vulnerability Assessment in Physical Level: For side-channel assessment, we have

already proposed an AI/ML roadmap in section VI-B. Although AI/ML-based approaches have a high potential for the successful assessment of side-channel vulnerability at the physical layout level, there are multiple critical challenges for performing AI/ML approaches, as explained below:

- Lack of data is one of the main challenges for the ML-based solution for side-channel vulnerability assessment at the physical level. The growth of deep learning-based approaches for side-channel attacks at post-silicon traces has been around since 2018 when the open-source large dataset ASCAD [160] was introduced. However, no AI/ML-based approach is performed for the pre-silicon side-channel analysis at layout because datasets of side-channel traces are not collected at the physical design level. The main reason for such unavailability of data at the physical layout is the high simulation time required for data collection. Trace collection at the layout level is a time-consuming and tedious task, as shown in Table IV. Lin *et. al* [42] shows that for a chip with a node count of 3.465M, the traditional VCD-based analysis takes around 671 hours run time to generate 10000 power traces. ML-based approaches are expected to require millions of traces to train the neural network, which indicates that it may take many months to collect adequate data for an ML-based solution of side-channel assessment at the layout level. The challenge remains to introduce a well defined and complete dataset of power traces (collected from the physical layout at the platform level) which may pave the way to new research direction for side-channel analysis at the physical design level (similar to what happened with the ASCAD dataset at the post-silicon stage).
- As mentioned before, the profiled DL-based side-channel attack requires the following steps: data pre-processing, feature engineering, algorithm selection, and attack evaluation. There is still no proper guideline for a post-silicon or pre-silicon side-channel attack for any of these steps. It is unclear what type of pre-processing and feature extraction approaches would work efficiently. For AI-based side-channel assessment, security metrics used for performance evaluation are different from typically used ML metrics. Still, many recent studies suggest new security metrics for side-channel attack evaluation mentioning possible drawbacks of existing metrics.
- Lack of explainability is another open challenge for the ML-based approach to side-channel vulnerability assessment. Neural networks have shown great success, even in robust countermeasures. However, how these networks deal with masking countermeasures is unclear. If an unsuccessful attack, it is tough to say whether it happens because of a weak countermeasure or an ineffective AI approach. It is unclear why the attack did not succeed if it was not successful. Similarly, in case of a successful attack, no one has been able to point out the exact weakness of the design. Such a lack of explainability makes it challenging to propose effective countermeasures.

2) Challenges in Laser Fault Injection in Physical Level: Pre-silicon assessment for LFI poses many challenges at the physical level, as discussed below.

- Precise modeling of laser effects during pre-silicon conditions is challenging. Various SPICE models have been proposed to emulate the laser's transient current effect, yet fail to account for physical design parameters (power distribution network, nearby elements, etc.).
- Most models are presented on small components (like an inverter or a D-flip flop). Modeling the laser impact on a large SoC is still a challenge.
- With the reduction in technology nodes, the smallest laser spot size can impact multiple cells in the region, causing multiple faults. Laser modeling for the multi-faults scenario in a large SoC is also challenging.

3) Challenges in Clock-Glitch Fault Injection Assessment in Physical Level:

- Timing margin analysis: Defining the exact time range within which an attacker can inject a feasible and controllable fault is challenging. The upper and lower margin of this time range is also dependent of the input stimuli. An exhaustive simulation is required with random inputs to figure out the exact margin of feasible fault injection.
- Feasibility analysis: Timing fault feasibility analysis requires several factors to be considered simultaneously. A feasible fault location depends on input stimuli, RC-delay corner of static timing analysis, pulse width of glitch signal and delay distribution of the datapath delays of critical registers. Finding a feasible location where an attacker can inject a feasible fault through clock glitching is a very difficult task.

4) Challenges in AI model realization for Secure Physical Design: As discussed in the sections VI, designing AI/ML-based models for secure physical designs is challenging, which are a must to ensure trust in every stage of the hardware flow. Many challenges need to be addressed in order to move ahead. A few of the challenges are listed below:

- Data Requirement: To create a model that trains and produces accurate results, a model needs data. Acquiring circuit and layout data and labeling them is a critical stage. The most significant drawbacks of acquiring such data are specialized equipment and the time required to collect and label them. It will also be a challenge to collect data that is scalable from one design to another.
- Feature Extraction: This would be another challenge based on the data that is collected (1D and/or 2D) and how each feature or set of features can be utilized to make the physical design secure.
- Developing Models: Training supervised models has a huge bottleneck since the data required for training such models is very limited. Some research has shown progress when utilizing heuristic-based approaches. However, such approaches need to analyze the vulnerabilities. It also depends on the formulation of cost functions based on our observed vulnerability and how it scales.

C. Challenges in Reaching Optimum Design Point

Suppose we have the physical design security metrics, with which we can obtain the level of vulnerability of any physical design. Now, when we optimize the physical design to mitigate the physical vulnerabilities, what will be the Power-Performance-Area (PPA) overhead of the physical design. With the advancement of technology nodes, it is already very challenging to obtain a PPA trade-off. With security as an added objective, it will be more difficult for designers to obtain an acceptable PPA trade-off.

D. Challenges in Achieving Competitive Time-to-Market

The authors believe that defining physical design security metrics involves lots of dynamic data and statistical computations at all levels of metal stacks and throughout the design. For example, power side-channel assessment may require dynamic power traces at all levels of metal stacks and obtaining some kind of correlation from these dynamic power traces, which is a time-consuming and costly affair. Thus, several additional assessments need to be done. Subsequently, mitigating those security metrics (or vulnerabilities) will increase the time-to-market of such secure physical design-based chips. The need is a smart way of assessing the vulnerability which reduces time and resources needed in the process.

VIII. Concluding Remarks

To ensure trust at every level of the hardware design flow, having a secure physical design is crucial. In this paper, we have introduced the concept of obtaining a *secure physical design* that is free from any physical vulnerability. We demonstrated how vulnerabilities can occur in the physical layout and utilize physical vulnerabilities despite having a secure synthesized gate-level netlist. Also, we have listed a brief overview of possible countermeasures to combat physical vulnerabilities. Moreover, we described our proposed framework for secure physical design and verification. We also presented an AI/ML roadmap to obtain a secure physical layout. Several challenges for obtaining a secure physical layout are also listed in the paper. Overall, this paper presents various aspects of obtaining a secure physical design.

Acknowledgment

This work was supported in part by research gifts from Synopsys and Ansys Corporations.

References

[1] IDC, "Semiconductor market to grow by 17.3 % in 2021 and reach potential overcapacity by 2023, idc reports," https://www.idc.com/getdoc.jsp?containerId=prAP48247621, accessed: 2021-12-9.

[2] B. Ahmed, M. K. Bepary, N. Pundir, M. Borza, O. Raikhman, A. Garg, D. Donchin, A. Cron, M. A. Abdel-moneum, F. Farahmandi *et al.*, "Quantifiable assurance: From ips to platforms," *arXiv preprint arXiv:2204.07909*, 2022.

[3] N. Farzana, A. Ayalasomayajula, F. Rahman, F. Farahmandi, and M. Tehranipoor, "Saif: Automated asset identification for security verification at the register transfer level," in *2021 IEEE 39th VLSI Test Symposium (VTS)*. IEEE, 2021, pp. 1–7.

[4] G. K. Contreras, A. Nahiyan, S. Bhunia, D. Forte, and M. Tehranipoor, "Security vulnerability analysis of design-for-test exploits for asset protection in socs," in *2017 22nd Asia and South Pacific Design Automation Conference (ASP-DAC)*. IEEE, 2017, pp. 617–622.

[5] M. Tehranipoor and F. Koushanfar, "A survey of hardware trojan taxonomy and detection," *IEEE design & test of computers*, vol. 27, no. 1, pp. 10–25, 2010.

[6] H. Salmani, M. Tehranipoor, and J. Plusquellic, "A novel technique for improving hardware trojan detection and reducing trojan activation time," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 20, no. 1, pp. 112–125, 2011.

[7] H. Salmani, M. Tehranipoor, and J. Plusquellic, "New design strategy for improving hardware trojan detection and reducing trojan activation time," in *2009 IEEE International Workshop on Hardware-Oriented Security and Trust*. IEEE, 2009, pp. 66–73.

[8] S. Bhunia and M. Tehranipoor, "The hardware trojan war," *Cham,, Switzerland: Springer*, 2018.

[9] X. Wang, H. Salmani, M. Tehranipoor, and J. Plusquellic, "Hardware trojan detection and isolation using current integration and localized current analysis," in *2008 IEEE international symposium on defect and fault tolerance of VLSI systems*. IEEE, 2008, pp. 87–95.

[10] M. Tehranipoor, H. Salmani, X. Zhang, M. Wang, R. Karri, J. Rajendran, and K. Rosenfeld, "Trustworthy hardware: Trojan detection and design-for-trust challenges," *Computer*, vol. 44, no. 7, pp. 66–74, 2010.

[11] H. Salmani and M. Tehranipoor, "Analyzing circuit vulnerability to hardware trojan insertion at the behavioral level," in *2013 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFTS)*. IEEE, 2013, pp. 190–195.

[12] M. Tehranipoor, H. Salmani, and X. Zhang, "Integrated circuit authentication: Hardware trojans and counterfeit detection," 2016.

[13] M. Li, A. Davoodi, and M. Tehranipoor, "A sensor-assisted self-authentication framework for hardware trojan detection," in *2012 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2012, pp. 1331–1336.

[14] G. T. Becker, F. Regazzoni, C. Paar, and W. P. Burleson, "Stealthy dopant-level hardware trojans," in *CHES*. Springer, 2013, pp. 197–214. [Online]. Available: https://www.iacr.org/archive/ches2013/80860203/80860203.pdf

[15] T. Zhang, J. Park, M. Tehranipoor, and F. Farahmandi, "Psc-tg: Rtl power side-channel leakage assessment with test pattern generation," in *2021 58th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2021, pp. 709–714.

[16] A. Nahiyan, J. Park, M. He, Y. Iskander, F. Farahmandi, D. Forte, and M. Tehranipoor, "Script: A cad framework for power side-channel vulnerability assessment using information flow tracking and pattern generation," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 25, no. 3, pp. 1–27, 2020.

[17] M. He, J. Park, A. Nahiyan, A. Vassilev, Y. Jin, and M. Tehranipoor, "Rtl-psc: Automated power side-channel leakage assessment at register-transfer level," in *2019 IEEE 37th VLSI Test Symposium (VTS)*. IEEE, 2019, pp. 1–6.

[18] C. Momin, O. Bronchain, and F. Standaert, "A stealthy hardware trojan based on a statistical fault attack," *Cryptogr. Commun.*, vol. 13, no. 4, pp. 587–600, 2021. [Online]. Available: https://doi.org/10.1007/s12095-021-00480-4

[19] N. Ahmed, M. Tehranipoor, and V. Jayaram, "Transition delay fault test pattern generation considering supply voltage noise in a soc design," in *Proceedings of the 44th annual Design Automation Conference*, 2007, pp. 533–538.

[20] H. Wang, H. Li, F. Rahman, M. M. Tehranipoor, and F. Farahmandi, "Sofi: Security property-driven vulnerability assessments of ics against fault-injection attacks," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2021.

[21] A. Nahiyan, F. Farahmandi, P. Mishra, D. Forte, and M. Tehranipoor, "Security-aware fsm design flow for identifying and mitigating vulnerabilities to fault attacks," *IEEE Transactions on Computer-aided design of integrated circuits and systems*, vol. 38, no. 6, pp. 1003–1016, 2018.

[22] F. Bao, K. Peng, M. Yilmaz, K. Chakrabarty, L. Winemberg, and M. Tehranipoor, "Efficient pattern generation for small-delay defects using selection of critical faults," *Journal of Electronic Testing*, vol. 29, no. 1, pp. 35–48, 2013.

[23] M. Tehranipoor, K. Peng, and K. Chakrabarty, *Test and diagnosis for small-delay defects*. Springer, 2011.

[24] U. Guin, X. Zhang, D. Forte, and M. Tehranipoor, "Low-cost on-chip structures for combating die and ic recycling," in *2014 51st ACM/EDAC/IEEE Design Automation Conference (DAC)*. IEEE, 2014, pp. 1–6.

[25] H. Dogan, D. Forte, and M. M. Tehranipoor, “Aging analysis for recycled fpga detection,” in *2014 IEEE international symposium on defect and fault tolerance in VLSI and nanotechnology systems (DFT)*. IEEE, 2014, pp. 171–176.

[26] U. Guin, D. DiMase, and M. Tehranipoor, “A comprehensive framework for counterfeit defect coverage analysis and detection assessment,” *Journal of Electronic Testing*, vol. 30, no. 1, pp. 25–40, 2014.

[27] M. M. Alam, M. Tehranipoor, and D. Forte, “Recycled fpga detection using exhaustive lut path delay characterization and voltage scaling,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 12, pp. 2897–2910, 2019.

[28] M. Alam, S. Chowdhury, M. M. Tehranipoor, and U. Guin, “Robust, low-cost, and accurate detection of recycled ics using digital signatures,” in *2018 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*. IEEE, 2018, pp. 209–214.

[29] Z. Guo, M. T. Rahman, M. M. Tehranipoor, and D. Forte, “A zero-cost approach to detect recycled soc chips using embedded sram,” in *2016 IEEE international symposium on hardware oriented security and trust (HOST)*. IEEE, 2016, pp. 191–196.

[30] J. Wurm, Y. Jin, Y. Liu, S. Hu, K. Heffner, F. Rahman, and M. Tehranipoor, “Introduction to cyber-physical system security: A cross-layer perspective,” *IEEE Transactions on Multi-Scale Computing Systems*, vol. 3, no. 3, pp. 215–227, 2016.

[31] K. Ahi, N. Asadizanjani, S. Shahbazmohamadi, M. Tehranipoor, and M. Anwar, “Terahertz characterization of electronic components and comparison of terahertz imaging with x-ray imaging techniques,” in *Terahertz Physics, Devices, and Systems IX: Advanced Applications in Industry and Defense*, vol. 9483. SPIE, 2015, pp. 82–96.

[32] U. Guin, K. Huang, D. DiMase, J. M. Carulli, M. Tehranipoor, and Y. Makris, “Counterfeit integrated circuits: A rising threat in the global semiconductor supply chain,” *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1207–1228, 2014.

[33] W. Hu, C. Chang, A. Sengupta, S. Bhunia, R. Kastner, and H. Li, “An overview of hardware security and trust: Threats, countermeasures, and design tools,” *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, vol. 40, no. 6, pp. 1010–1038, 2021. [Online]. Available: https://doi.org/10.1109/TCAD.2020.3047976

[34] N. Pundir, F. Farahmandi, and M. Tehranipoor, “Secure high-level synthesis: Challenges and solutions,” in *2021 22nd International Symposium on Quality Electronic Design (ISQED)*. IEEE, 2021, pp. 164–171.

[35] M. R. Muttaki, R. Mohammadivojdan, M. Tehranipoor, and F. Farahmandi, “Hlock: Locking ips at the high-level language,” in *2021 58th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2021, pp. 79–84.

[36] H. M. Kamali, K. Z. Azar, F. Farahmandi, and M. Tehranipoor, “Advances in logic locking: Past, present, and prospects,” *Cryptology ePrint Archive*, 2022.

[37] A. B. Kahng, J. Lienig, I. L. Markov, and J. Hu, *VLSI physical design: from graph partitioning to timing closure*. Springer Science & Business Media, 2011.

[38] D. Šijačić, J. Balasch, and I. Verbauwhede, “Sweeping for leakage in masked circuit layouts,” in *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2020, pp. 915–920.

[39] T. De Cnudde, B. Bilgin, B. Gierlichs, V. Nikov, S. Nikova, and V. Rijmen, “Does coupling affect the security of masked implementations?” in *International Workshop on Constructive Side-Channel Analysis and Secure Design*. Springer, 2017, pp. 1–18.

[40] S. Bhasin, J.-L. Danger, T. Graba, Y. Mathieu, D. Fujimoto, and M. Nagata, “Physical security evaluation at an early design-phase: A side-channel aware simulation methodology,” in *Proceedings of International Workshop on Engineering Simulations for Cyber-Physical Systems*, 2013, pp. 13–20.

[41] F. Regazzoni, S. Badel, T. Eisenbarth, J. Grobschadl, A. Poschmann, Z. Toprak, M. Macchetti, L. Pozzi, C. Paar, Y. Leblebici *et al.*, “A simulation-based methodology for evaluating the dpa-resistance of cryptographic functional units with application to cmos and mcml technologies,” in *2007 International Conference on Embedded Computer Systems: Architectures, Modeling and Simulation*. IEEE, 2007, pp. 209–214.

[42] L. Lin, D. Selvakumaran, D. Zhu, N. Chang, C. Chow, M. Nagata, and K. Monta, “Fast and comprehensive simulation methodology for layout-based power-noise side-channel leakage analysis,” in *2020 IEEE International Symposium on Smart Electronic Systems (iSES)(Formerly iNiS)*. IEEE, 2020, pp. 133–138.

[43] C. O'Flynn, “Fault injection using crowbars on embedded systems.” *IACR Cryptol. ePrint Arch.*, vol. 2016, p. 810, 2016.

[44] J. Balasch, B. Gierlichs, and I. Verbauwhede, “An in-depth and black-box characterization of the effects of clock glitches on 8-bit mcus,” in *2011 Workshop on Fault Diagnosis and Tolerance in Cryptography*. IEEE, 2011, pp. 105–114.

[45] A. Barenghi, G. M. Bertoni, L. Breveglieri, M. Pelliccioli, and G. Pelosi, “Injection technologies for fault attacks on microprocessors,” *Fault Analysis in Cryptography*, pp. 275–293, 2012.

[46] M. Hutter, J.-M. Schmidt, and T. Plos, “Contact-based fault injections and power analysis on rfid tags,” in *2009 European Conference on Circuit Theory and Design*. IEEE, 2009, pp. 409–412.

[47] H. Martin, T. Korak, E. San Millán, and M. Hutter, “Fault attacks on strngs: Impact of glitches, temperature, and underpowering on randomness,” *IEEE transactions on information forensics and security*, vol. 10, no. 2, pp. 266–277, 2014.

[48] G. Canivet, P. Maistri, R. Leveugle, J. Clédière, F. Valette, and M. Renaudin, “Glitch and laser fault attacks onto a secure aes implementation on a sram-based fpga,” *Journal of cryptology*, vol. 24, no. 2, pp. 247–268, 2011.

[49] B. Selmke, F. Hauschild, and J. Obermaier, “Peak clock: Fault injection into pll-based systems via clock manipulation,” in *Proceedings of the 3rd ACM Workshop on Attacks and Solutions in Hardware Security Workshop*, 2019, pp. 85–94.

[50] T. Bonny and Q. Nasir, “Clock glitch fault injection attack on an fpga-based non-autonomous chaotic oscillator,” *Nonlinear Dynamics*, vol. 96, no. 3, pp. 2087–2101, 2019.

[51] B. Ning and Q. Liu, “Modeling and efficiency analysis of clock glitch fault injection attack,” in *2018 Asian Hardware Oriented Security and Trust Symposium (AsianHOST)*. IEEE, 2018, pp. 13–18.

[52] N. Timmers and C. Mune, “Escalating privileges in linux using voltage fault injection,” in *2017 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*. IEEE, 2017, pp. 1–8.

[53] D. Ha, K. Woo, S. Meninger, T. Xanthopoulos, E. Crain, and D. Ham, “Time-domain cmos temperature sensors with dual delay-locked loops for microprocessor thermal monitoring,” *IEEE transactions on very large scale integration (VLSI) systems*, vol. 20, no. 9, pp. 1590–1601, 2011.

[54] J. G. Van Woudenberg, M. F. Witteman, and F. Menarini, “Practical optical fault injection on secure microcontrollers,” in *2011 Workshop on Fault Diagnosis and Tolerance in Cryptography*. IEEE, 2011, pp. 91–99.

[55] M. Yilmaz, K. Chakrabarty, and M. Tehranipoor, “Interconnect-aware and layout-oriented test-pattern selection for small-delay defects,” in *2008 IEEE International Test Conference*. IEEE, 2008, pp. 1–10.

[56] H. Ziade, R. A. Ayoubi, R. Velazco *et al.*, “A survey on fault injection techniques,” *Int. Arab J. Inf. Technol.*, vol. 1, no. 2, pp. 171–186, 2004.

[57] H. Wang, Q. Shi, A. Nahiyan, D. Forte, and M. M. Tehranipoor, “A physical design flow against front-side probing attacks by internal shielding,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 10, pp. 2152–2165, 2019.

[58] H. Wang, Q. Shi, D. Forte, and M. M. Tehranipoor, “Probing assessment framework and evaluation of antiprobing solutions,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 6, pp. 1239–1252, 2019.

[59] M. Agoyan, J.-M. Dutertre, D. Naccache, B. Robisson, and A. Tria, “When clocks fail: On critical paths and clock faults,” in *International conference on smart card research and advanced applications*. Springer, 2010, pp. 182–193.

[60] L. Zussa, J.-M. Dutertre, J. Clédiere, B. Robisson, A. Tria *et al.*, “Investigation of timing constraints violation as a fault injection means,” in *27th Conference on Design of Circuits and Integrated Systems (DCIS), Avignon, France*. Citeseer, 2012, pp. 1–6.

[61] B. Razavi, *Fundamentals of microelectronics*. John Wiley & Sons, 2021.

[62] K. M. Zick, M. Srivastav, W. Zhang, and M. French, “Sensing nanosecond-scale voltage attacks and natural transients in fpgas,” in *Proceedings of the ACM/SIGDA international symposium on Field programmable gate arrays*, 2013, pp. 101–104.

[63] K. Murdock, D. Oswald, F. D. Garcia, J. Van Bulck, D. Gruss, and F. Piessens, “Plundervolt: Software-based fault injection attacks against intel sgx,” in *2020 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2020, pp. 1466–1482.

[64] J. Krautter, D. R. Gnad, and M. B. Tahoori, “Fpgahammer: Remote voltage fault attacks on shared fpgas, suitable for dfa on aes,” *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pp. 44–68, 2018.

[65] N. N. Anandakumar, S. K. Sanadhya, and M. S. Hashmi, "Fpga-based true random number generation using programmable delays in oscillator-rings," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 3, pp. 570–574, 2019.

[66] M. Dumont, M. Lisart, and P. Maurine, "Electromagnetic fault injection: how faults occur," in *2019 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*. IEEE, 2019, pp. 9–16.

[67] D. Saha and S. Sur-Kolay, "Fast robust intellectual property protection for vlsi physical design," in *10th International Conference on Information Technology (ICIT 2007)*. IEEE, 2007, pp. 1–6.

[68] S. Zamanzadeh and A. Jahanian, "Automatic netlist scrambling methodology in asic design flow to hinder the reverse engineering," in *2013 IFIP/IEEE 21st International Conference on Very Large Scale Integration (VLSI-SoC)*. IEEE, 2013, pp. 52–53.

[69] Y. Xie, C. Bao, and A. Srivastava, "Security-aware design flow for 2.5 d ic technology," in *Proceedings of the 5th International Workshop on Trustworthy Embedded Devices*, 2015, pp. 31–38.

[70] A. Vijayakumar, V. C. Patil, D. E. Holcomb, C. Paar, and S. Kundu, "Physical design obfuscation of hardware: A comprehensive investigation of device and logic-level techniques," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 1, pp. 64–77, 2016.

[71] Y. Wang, P. Chen, J. Hu, and J. J. Rajendran, "Routing perturbation for enhanced security in split manufacturing," in *2017 22nd Asia and South Pacific Design Automation Conference (ASP-DAC)*. IEEE, 2017, pp. 605–510.

[72] M. Khairallah, R. Sadhukhan, R. Samanta, J. Breier, S. Bhasin, R. S. Chakraborty, A. Chattopadhyay, and D. Mukhopadhyay, "Dfarpa: Differential fault attack resistant physical design automation," in *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2018, pp. 1171–1174.

[73] P. Slpsk, P. K. Vairam, C. Rebeiro, and V. Kamakoti, "Karna: A gate-sizing based security aware eda flow for improved power side-channel attack protection," in *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 2019, pp. 1–8.

[74] H. Sonoda, K. Monta, T. Okidono, Y. Araga, N. Watanabe, H. Shimamoto, K. Kikuchi, N. Miura, T. Miki, and M. Nagata, "Secure 3d cmos chip stacks with backside buried metal power delivery networks for distributed decoupling capacitance," in *2020 IEEE International Electron Devices Meeting (IEDM)*. IEEE, 2020, pp. 31–5.

[75] K. Monta, H. Sonoda, T. Okidono, Y. Araga, N. Watanabe, H. Shimamoto, K. Kikuchi, N. Miura, T. Miki, and M. Nagata, "3-d cmos chip stacking for security ics featuring backside buried metal power delivery networks with distributed capacitance," *IEEE Transactions on Electron Devices*, vol. 68, no. 4, pp. 2077–2082, 2021.

[76] M. Wang, V. V. Iyer, S. Xie, G. Li, S. K. Mathew, R. Kumar, M. Orshansky, A. E. Yilmaz, and J. P. Kulkarni, "Physical design strategies for mitigating fine-grained electromagnetic side-channel attacks," in *2021 IEEE Custom Integrated Circuits Conference (CICC)*. IEEE, 2021, pp. 1–2.

[77] J. Knechtel *et al.*, "Security closure of physical layouts," in *2021 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 2021.

[78] T. D. Perez and S. Pagliarini, "A survey on split manufacturing: Attacks, defenses, and challenges," *IEEE Access*, vol. 8, pp. 184 013–184 035, 2020.

[79] H. Salmani and M. M. Tehranipoor, "Vulnerability analysis of a circuit layout to hardware trojan insertion," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 6, pp. 1214–1225, 2016.

[80] Q. Shi, N. Asadizanjani, D. Forte, and M. M. Tehranipoor, "A layout-driven framework to assess vulnerability of ics to microprobing attacks," in *2016 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*. IEEE, 2016, pp. 155–160.

[81] H. Salmani and M. Tehranipoor, "Layout-aware switching activity localization to enhance hardware trojan detection," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 1, pp. 76–87, 2011.

[82] H. Salmani, M. Tehranipoor, and J. Plusquellic, "A layout-aware approach for improving localized switching to detect hardware trojans in integrated circuits," in *2010 IEEE International Workshop on Information Forensics and Security*. IEEE, 2010, pp. 1–6.

[83] S. Brown, S. Aftabjahani, and M. Tehranipoor, "Trust-hub physical vulnerabilities-db." [Online]. Available: https://trust-hub.org/#/vulnerability-db/physical-vulnerabilities

[84] P. C. Kocher, "Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems," in *Annual International Cryptology Conference*. Springer, 1996, pp. 104–113.

[85] P. C. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology*, ser. CRYPTO '99. Berlin, Heidelberg: Springer-Verlag, 1999, p. 388–397.

[86] D. Agrawal, B. Archambeault, J. R. Rao, and P. Rohatgi, "The em side-channel(s)," in *Revised Papers from the 4th International Workshop on Cryptographic Hardware and Embedded Systems*, ser. CHES '02. Berlin, Heidelberg: Springer-Verlag, 2002, p. 29–45.

[87] D. Genkin, A. Shamir, and E. Tromer, "Rsa key extraction via low-bandwidth acoustic cryptanalysis," in *Advances in Cryptology – CRYPTO 2014*, J. A. Garay and R. Gennaro, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 444–461.

[88] A. Schlösser, D. Nedospasov, J. Krämer, S. Orlic, and J.-P. Seifert, "Simple photonic emission analysis of aes," in *Cryptographic Hardware and Embedded Systems – CHES 2012*, E. Prouff and P. Schaumont, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 41–57.

[89] Y. Yarom and K. Falkner, "Flush+reload: A high resolution, low noise, l3 cache side-channel attack," in *Proceedings of the 23rd USENIX Conference on Security Symposium*, ser. SEC'14. USA: USENIX Association, 2014, p. 719–732.

[90] N. Ahmed, M. H. Tehranipour, and M. Nourani, "Low power pattern generation for bist architecture," in *2004 IEEE International Symposium on Circuits and Systems (IEEE Cat. No. 04CH37512)*, vol. 2. IEEE, 2004, pp. II–689.

[91] D. McCann, E. Oswald, and C. Whitnall, "Towards practical tools for side channel aware software engineering: 'grey box' modelling for instruction leakages," in *26th USENIX Security Symposium (USENIX Security 17)*. Vancouver, BC: USENIX Association, Aug. 2017, pp. 199–216. [Online]. Available: https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/mccann

[92] J. Park, X. Xu, Y. Jin, D. Forte, and M. Tehranipoor, "Power-based side-channel instruction-level disassembler," in *2018 55th ACM/ESDA/IEEE Design Automation Conference (DAC)*, 2018, pp. 1–6.

[93] M.-L. Akkar and C. Giraud, "An implementation of des and aes, secure against some attacks," in *Cryptographic Hardware and Embedded Systems — CHES 2001*, Ç. K. Koç, D. Naccache, and C. Paar, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 309–318.

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

[95] M. H. Tehranipour, N. Ahmed, and M. Nourani, "Testing soc interconnects for signal integrity using boundary scan," in *Proceedings. 21st VLSI Test Symposium, 2003.* IEEE, 2003, pp. 158–163.

[96] S. Nikova, C. Rechberger, and V. Rijmen, "Threshold implementations against side-channel attacks and glitches," in *Information and Communications Security*, P. Ning, S. Qing, and N. Li, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 529–545.

[97] T. D. Cnudde, B. Bilgin, B. Gierlichs, V. Nikov, S. Nikova, and V. Rijmen, "Does coupling affect the security of masked implementations?" Cryptology ePrint Archive, Report 2016/1080, 2016, https://ia.cr/2016/1080.

[98] G. Goodwill, B. Jun, J. Jaffe, and P. Rohatgi, "A testing methodology for side channel resistance," 2011.

[99] N. Sehatbakhsh, B. B. Yilmaz, A. Zajic, and M. Prvulovic, "Emsim: A microarchitecture-level simulation tool for modeling electromagnetic side-channel signals," in *2020 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, 2020, pp. 71–85.

[100] M. A. Shelton, N. Samwel, L. Batina, F. Regazzoni, M. Wagner, and Y. Yarom, "Rosita: Towards automatic elimination of power-analysis leakage in ciphers," in *28th Annual Network and Distributed System Security Symposium, NDSS 2021, virtually, February 21-25, 2021*. The Internet Society, 2021.

[101] M. A. Shelton, L. Chmielewski, N. Samwel, M. Wagner, L. Batina, and Y. Yarom, "Rosita++: Automatic higher-order leakage elimination from cryptographic code," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 685–699. [Online]. Available: https://doi.org/10.1145/3460120.3485380

[102] A. Nahiyan, J. Park, M. He, Y. Iskander, F. Farahmandi, D. Forte, and M. Tehranipoor, "Script: A cad framework for power side-channel vulnerability assessment using information flow tracking and pattern generation," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 25, no. 3, may 2020. [Online]. Available: https://doi.org/10.1145/3383445

[103] P. SLPSK, P. K. Vairam, C. Rebeiro, and V. Kamakoti, "Karna: A gate-sizing based security aware EDA flow for improved power side-channel attack protection," in *Proceedings of the International Conference on Computer-Aided Design, ICCAD 2019, Westminster, CO, USA, November 4-7, 2019*, D. Z. Pan, Ed. ACM, 2019, pp. 1–8. [Online]. Available: https://doi.org/10.1109/ICCAD45719.2019.8942173

[104] J. Park, N. N. Anandakumar, D. Saha, D. Mehta, N. Pundir, F. Rahman, F. Farahmandi, and M. M. Tehranipoor, "Pqc-sep: Power side-channel evaluation platform for post-quantum cryptography algorithms." *IACR Cryptol. ePrint Arch.*, vol. 2022, p. 527, 2022.

[105] N. Pundir, J. Park, F. Farahmandi, and M. Tehranipoor, "Power side-channel leakage assessment framework at register-transfer level," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2022.

[106] T. Zhang, J. Park, M. M. Tehranipoor, and F. Farahmandi, "PSC-TG: RTL power side-channel leakage assessment with test pattern generation," in *58th ACM/IEEE Design Automation Conference, DAC 2021, San Francisco, CA, USA, December 5-9, 2021*. IEEE, 2021, pp. 709–714. [Online]. Available: https://doi.org/10.1109/DAC18074.2021.9586210

[107] N. Sehatbakhsh, B. B. Yilmaz, A. G. Zajic, and M. Prvulovic, "Emsim: A microarchitecture-level simulation tool for modeling electromagnetic side-channel signals," in *IEEE International Symposium on High Performance Computer Architecture, HPCA 2020, San Diego, CA, USA, February 22-26, 2020*. IEEE, 2020, pp. 71–85. [Online]. Available: https://doi.org/10.1109/HPCA47549.2020.00016

[108] B. Gigerl, V. Hadzic, R. Primas, S. Mangard, and R. Bloem, "Coco: Co-Design and Co-Verification of masked software implementations on CPUs," in *30th USENIX Security Symposium (USENIX Security 21)*. USENIX Association, Aug. 2021, pp. 1469–1468. [Online]. Available: https://www.usenix.org/conference/usenixsecurity21/presentation/gigerl

[109] M. M. Alam, S. Tajik, F. Ganji, M. Tehranipoor, and D. Forte, "Ram-jam: Remote temperature and voltage fault attack on fpgas using memory collisions," in *2019 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*. IEEE, 2019, pp. 48–55.

[110] C. Roscian, J.-M. Dutertre, and A. Tria, "Frontside laser fault injection on cryptosystems-application to the aes'last round," in *2013 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*. IEEE, 2013, pp. 119–124.

[111] P. Maurine, "Techniques for em fault injection: equipments and experimental results," in *2012 Workshop on Fault Diagnosis and Tolerance in Cryptography*. IEEE, 2012, pp. 3–4.

[112] N. Pundir, H. Li, L. Lin, N. Chang, F. Farahmandi, and M. Tehranipoor, "Security properties driven pre-silicon laser fault injection assessment," *2022 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, 2022.

[113] N. Pundir, H. Li, L. Lin, N. Chang, F. Farahmandi, and M. Tehranipoor, "Spili: Security properties and machine learning assisted pre-silicon laser fault injection assessment," *International Symposium for Testing and Failure Analysis (ISTFA)*, 2022.

[114] A. Tria, "Frontside laser fault injection on cryptosystems-application to the aes'last round," in *2013 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*. IEEE Computer Society, 2013, pp. 119–124.

[115] M. Agoyan, J.-M. Dutertre, A.-P. Mirbaha, D. Naccache, A.-L. Ribotta, and A. Tria, "Single-bit dfa using multiple-byte laser fault injection," in *2010 IEEE International Conference on Technologies for Homeland Security (HST)*. IEEE, 2010, pp. 113–119.

[116] B. Selmke, J. Heyszl, and G. Sigl, "Attack on a dfa protected aes by simultaneous laser fault injections," in *2016 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*. IEEE, 2016, pp. 36–46.

[117] F. Cai, G. Bai, H. Liu, and X. Hu, "Optical fault injection attacks for flash memory of smartcards," in *2016 6th International Conference on Electronics Information and Emergency Communication (ICEIEC)*. IEEE, 2016, pp. 46–50.

[118] T. Sugawara, B. Cyr, S. Rampazzi, D. Genkin, and K. Fu, "Light commands: laser-based audio injection attacks on voice-controllable systems," in *29th {USENIX} Security Symposium ({USENIX} Security 20)*, 2020, pp. 2631–2648.

[119] A. Vasselle, H. Thiebeauld, Q. Maouhoub, A. Morisset, and S. Ermeneux, "Laser-induced fault injection on smartphone bypassing the secure boot," in *2017 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*. IEEE, 2017, pp. 41–48.

[120] J. Breier, D. Jap, X. Hou, S. Bhasin, and Y. Liu, "Sniff: reverse engineering of neural networks with fault attacks," *IEEE Transactions on Reliability*, 2021.

[121] E. Biham and A. Shamir, "Differential fault analysis of secret key cryptosystems," in *Annual international cryptology conference*. Springer, 1997, pp. 513–525.

[122] Y. Li, K. Sakiyama, S. Gomisawa, T. Fukunaga, J. Takahashi, and K. Ohta, "Fault sensitivity analysis," in *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 2010, pp. 320–334.

[123] N. F. Ghalaty, B. Yuce, M. Taha, and P. Schaumont, "Differential fault intensity analysis," in *2014 Workshop on Fault Diagnosis and Tolerance in Cryptography*. IEEE, 2014, pp. 49–58.

[124] S. Ordas, L. Guillaume-Sage, K. Tobich, J.-M. Dutertre, and P. Maurine, "Evidence of a larger em-induced fault model," in *International Conference on Smart Card Research and Advanced Applications*. Springer, 2014, pp. 245–259.

[125] T. Miki, M. Nagata, H. Sonoda, N. Miura, T. Okidono, Y. Araga, N. Watanabe, H. Shimamoto, and K. Kikuchi, "A si-backside protection circuits against physical security attacks on flip-chip devices," in *2019 IEEE Asian Solid-State Circuits Conference (A-SSCC)*. IEEE, 2019, pp. 25–28.

[126] T. Ajayi, V. A. Chhabria, M. Fogaça, S. Hashemi, A. Hosny, A. B. Kahng, M. Kim, J. Lee, U. Mallappa, M. Neseem *et al.*, "Toward an open-source digital flow: First learnings from the openroad project," in *Proceedings of the 56th Annual Design Automation Conference 2019*, 2019, pp. 1–4.

[127] T.-C. Chen, Z.-W. Jiang, T.-C. Hsu, H.-C. Chen, and Y.-W. Chang, "Ntuplace3: An analytical placer for large-scale mixed-size designs with preplaced blocks and density constraints," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 7, pp. 1228–1240, 2008.

[128] C.-K. Cheng, A. B. Kahng, I. Kang, and L. Wang, "Replace: Advancing solution quality and routability validation in global placement," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 38, no. 9, pp. 1717–1730, 2018.

[129] Y. Lin, Z. Jiang, J. Gu, W. Li, S. Dhar, H. Ren, B. Khailany, and D. Z. Pan, "Dreamplace: Deep learning toolkit-enabled gpu acceleration for modern vlsi placement," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 40, no. 4, pp. 748–761, 2020.

[130] C.-H. Hsu, H.-Y. Chen, and Y.-W. Chang, "Multi-layer global routing considering via and wire capacities," in *2008 IEEE/ACM International Conference on Computer-Aided Design*. IEEE, 2008, pp. 350–355.

[131] "Innovus implementation system," Cadence, 2022. [Online]. Available: https://www.cadence.com/en_US/home/tools/digital-design-and-signoff/soc-implementation-and-floorplanning/innovus-implementation-system.html

[132] "Ic compiler ii," Synopsys, 2022. [Online]. Available: https://www.synopsys.com/implementation-and-signoff/physical-implementation/ic-compiler.html

[133] J. Park and A. Tyagi, "Security metrics for power based sca resistant hardware implementation," in *2016 29th International Conference on VLSI Design and 2016 15th International Conference on Embedded Systems (VLSID)*, 2016, pp. 541–546.

[134] Y. Fei, A. A. Ding, J. Lao, and L. Zhang, "A statistics-based fundamental model for side-channel attack analysis," Cryptology ePrint Archive, Report 2014/152, 2014, https://eprint.iacr.org/2014/152.

[135] A. Mazumder Shuvo, N. Pundir, J. Park, F. Farahmandi, and M. Tehranipoor, "Ldtfi: Layout-aware timing fault-injection attack assessment against differential fault analysis," *2022 IEEE Computer Society Annual Symposium on VLSI*, 2022.

[136] S. Dey, S. Dash, S. Nandi, and G. Trivedi, "PGIREM: reliability-constrained IR drop minimization and electromigration assessment of VLSI power grid networks using cooperative coevolution," in *2018 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*. IEEE, 2018, pp. 40–45.

[137] S. Dey, S. Nandi, and G. Trivedi, "PGOpt: Multi-objective design space exploration framework for large-Scale on-chip power grid design in VLSI SoC using evolutionary computing technique," *Microprocessors and Microsystems*, vol. 81, p. 103440, 2021.

[138] S. Dey, S. Nandi, and G. Trivedi, "PGRDP: reliability, delay, and power-aware area minimization of large-scale VLSI power grid network using cooperative coevolution," in *Intelligent Computing Paradigm: Recent Trends*. Springer, 2020, pp. 69–84.

[139] D. Das, M. Nath, B. Chatterjee, S. Ghosh, and S. Sen, "Stellar: A generic em side-channel attack protection through ground-up root-cause analysis," Cryptology ePrint Archive, Report 2018/620, 2018, https://ia.cr/2018/620.

[140] C. Tokunaga and D. Blaauw, "Securing encryption systems with a switched capacitor current equalizer," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 1, pp. 23–31, 2010.

[141] "4th acm/ieee workshop on machine learning for cad," 2022. [Online]. Available: https://mlcad-workshop.org/

[142] A. Mirhoseini, A. Goldie, M. Yazgan, J. W. Jiang, E. Songhori, S. Wang, Y.-J. Lee, E. Johnson, O. Pathak, A. Nazi *et al.*, "A graph placement methodology for fast chip design," *Nature*, vol. 594, no. 7862, pp. 207–212, 2021.

[143] Y.-J. Lee *et al.*, "Learning to play the game of macro placement with the help of deep reinforcement learning," 2021. [Online]. Available: https://youtu.be/EKjlr2k_wBM

[144] G. Huang, J. Hu, Y. He, J. Liu, M. Ma, Z. Shen, J. Wu, Y. Xu, H. Zhang, K. Zhong *et al.*, "Machine learning for electronic design automation: A survey," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 26, no. 5, pp. 1–46, 2021.

[145] M. Rapp, H. Amrouch, Y. Lin, B. Yu, D. Z. Pan, M. Wolf, and J. Henkel, "Mlcad: A survey of research in machine learning for cad keynote paper," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2021.

[146] S. Dey, S. Nandi, and G. Trivedi, "Machine Learning for VLSI CAD: A Case Study in On-Chip Power Grid Design," in *2021 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*. IEEE, 2021, pp. 378–383.

[147] S. Dey, S. Nandi, and G. Trivedi, "Machine learning approach for fast electromigration aware aging prediction in incremental design of large scale on-chip power grid network," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 25, no. 5, pp. 1–29, 2020.

[148] S. Dey, S. Nandi, and G. Trivedi, "PowerPlanningDL: Reliability-aware framework for on-chip power grid design using deep learning," in *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2020, pp. 1520–1525.

[149] S. Dey, "Design Methodology for On-Chip Power Grid Interconnect: AI/ML Perspective," IIT Guwahati, 2021.

[150] P. J. Van Laarhoven and E. H. Aarts, "Simulated annealing," in *Simulated annealing: Theory and applications*. Springer, 1987, pp. 7–15.

[151] J. H. Holland, "Genetic algorithms," *Scientific american*, vol. 267, no. 1, pp. 66–73, 1992.

[152] H. B. Curry, "The method of steepest descent for non-linear minimization problems," *Quarterly of Applied Mathematics*, vol. 2, no. 3, pp. 258–261, 1944.

[153] R. Maclin, "Machine learning for sequential data," University of Minnesota.

[154] N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N.-C. Yen, C. C. Tung, and H. H. Liu, "The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis," *in Proc. the Royal Society of London. Series A: mathematical, physical and engineering sciences*, vol. 454, no. 1971, pp. 903–995, 1998.

[155] M. Feldman, "Time-varying vibration decomposition and analysis based on the hilbert transform," *Journal of Sound and Vibration*, vol. 295, no. 3-5, pp. 518–530, 2006.

[156] K. Dragomiretskiy and D. Zosso, "Variational mode decomposition," *IEEE Trans. signal processing*, vol. 62, no. 3, pp. 531–544, 2013.

[157] G. Zaid, L. Bossuet, A. Habrard, and A. Venelli, "Methodology for efficient cnn architectures in profiling attacks," *IACR Trans. Cryptographic Hardware and Embedded Systems*, vol. 2020, no. 1, pp. 1–36, 2020.

[158] L. Wu, G. Perin, and S. Picek, "I choose you: Automated hyperparameter tuning for deep learning-based side-channel analysis." *IACR Cryptol. ePrint Arch.*, vol. 2020, p. 1293, 2020.

[159] G. Perin, Ł. Chmielewski, and S. Picek, "Strength in numbers: Improving generalization with ensembles in machine learning-based profiled side-channel analysis," *IACR Trans. Cryptographic Hardware and Embedded Systems*, pp. 337–364, 2020.

[160] E. Prouff, R. Strullu, R. Benadjila, E. Cagli, and C. Dumas, "Study of deep learning techniques for side-channel analysis and introduction to ascad database," *Cryptology ePrint Archive*, 2018.