Layout-level Hardware Trojan Prevention in the Context of Physical Design

Xingyu Tong[1,2], Guohao Chen[1,2], Min Wei[1,2], Zhijie Cai[1,2], Peng Zou[3], Zhifeng Lin[1,4]* and Jianli Chen[2]
[1]State Key Laboratory of Integrated Chips and Systems, Fudan University, Shanghai, China
[2]School of Microelectronics, Fudan University, Shanghai, China
[3]Shanghai LEDA Technology Co., Ltd., Shanghai, China
[4]School of Mathematics and Statistics, Fuzhou University, Fuzhou, China
xytong21@m.fudan.edu.cn; ghchen22@m.fudan.edu.cn; linzhifeng@fzu.edu.cn; chenjianli@fudan.edu.cn

ABSTRACT

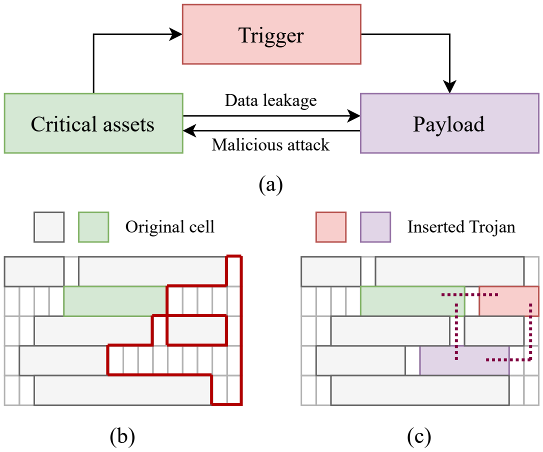
A growing recognition of potential vulnerabilities to layout-level Hardware Trojan (HT) attacks has spurred significant research efforts aimed at enhancing the resilience of ICs against such threats. However, traditional hardware security has been predominantly concerned with defensive measures, often overlooking the original key metrics in physical design evaluation: power, performance, and area (PPA). This study introduces an automated methodology incorporating HT considerations into the practical physical design process. Utilizing a Bayesian optimization framework, it effectively navigates the operation of commercial physical implementation tools in the solution space of hyper-parameter settings. Innovative strategies inspired by mosaic techniques, such as cell shifting and buffer insertion, realize additional improvements in layout-level trojan prevention. Comparative evaluations have shown that our approach outperforms leading entries from the ISPD 2023 Contest in terms of PPA and HT prevention metrics, thereby providing significant insights into the synergy between these critical factors.

KEYWORDS

Hardware Trojan, Hardware Security, Physical Design

1 INTRODUCTION

The paradigm shift towards the fabless model in integrated circuit (IC) design, where manufacturing is entrusted to external foundries, has traditionally featured production efficiency and cost reduction. This prioritization may have inadvertently exposed the design to potential security issues. Unlike software vulnerabilities, which can be easily fixed with patches afterward, Hardware Trojans (HTs) are physically embedded within a system's circuitry, irremovable after manufacture, thus causing irreversible malfunction. The burgeoning awareness of potential security breaches through layout-level HT insertions has instigated a surge in research endeavors dedicated to fortifying ICs against such malicious attacks[6].



(a)

(b)

(c)

Figure 1: (a) is a typical HT structure, including a trigger and a payload. The trigger activates the payload to perform data stealth or malicious attacks on critical assets. (b) and (c) are the physical layouts before and after Trojan insertion. Continuous unoccupied placement sites, framed in red lines, are exploitable for Trojan components. Exploitable regions also leave ample routing resources to connect triggers, payloads, and critical assets, as illustrated with dottle lines.

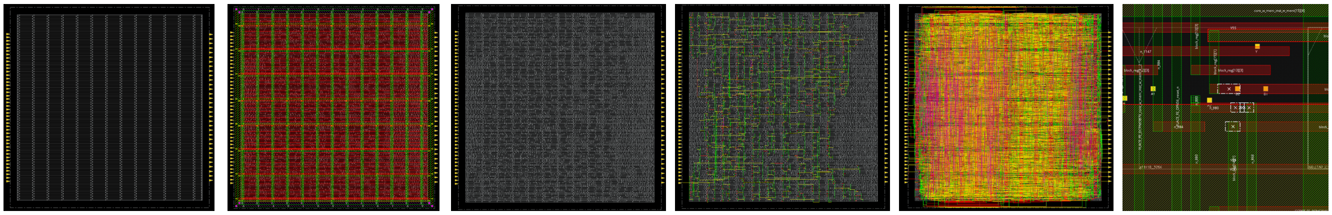
HTs execute malicious circuit modifications to compromise critical infrastructure, expropriate intellectual property, or initiate delayed attacks, resulting in significant financial losses[12]. Figure 1(a) depicts the typical two-part functionality of HTs. The *trigger* acts as an activation mechanism, often responding to infrequent and specific combinations of sequential and/or combinational circuit logic states. Upon encountering this pre-defined trigger condition, the HTs execute their malicious operation, referred to as the payload. The escalation of such corrupting attacks underscores the imperative for sophisticated defense strategies.

As both parts entail extra placement sites and routing resources to fit them into the packed physical layout of the victim IC, previous academic efforts have explored innovative design-time prevention strategies on exploitable resources. MUX (multiplexer)-based locking schemes[2, 20] work at physical synthesis stages, focusing on filling vulnerable open placement sites with additional MUX or gates without altering the original logic. Guo *et al.*[10] and Hsu *et al.*[11] have addressed security from a site-level perspective, partitioning graphs on vulnerable available sites and refined placement on cells. [21]

* Corresponding author.

This work was partially supported by National Natural Science Foundation of China under Grant 92373207, and the National Key Research and Development Program of China under Grants 2022YFB4400500 and 2021YFA1003602.

ICCAD '24, October 27–31, 2024, New York, NY, USA
© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-1077-3/24/10…$15.00
https://doi.org/10.1145/3676536.3676708



(a) Floorplanning (b) PDN (c) PlaceOpt (d) CTS (e) Routing (f) DRC refinement

Figure 2: Physical implementation flow of a crypto core SHA256

have customized anti-Trojan operators in Engineering Change Order (ECO) placement and routing, achieving minimum design quality impact compared with [3, 23]. These endeavors primarily focused on incremental modification added after the physical design flow to deal with HTs. The additional prevention overlooks or has to trade off with traditional power, performance, and area (PPA) metrics and technology rules for manufacturability, which are recognized to have a tangible impact on the yield, cost, and marketability[9, 22]. It is crucial to meticulously evaluate the impact of these countermeasures on the overall design quality, ensuring a balanced approach between layout-level HT defense and performance optimization.

This work starts from questioning the underlying assumption that security enhancements, such as layout-level HT prevention, inherently conflict with achieving optimal PPA metrics. In view of the emerging concept “security closure" [8] emphasizing the proactive integration of security considerations throughout the design flow, the motivation of this work focuses on proactively hardening layouts against the clandestine insertion of Trojans in the context of physical design. In essence, given the advancements in physical design space exploration, security should be integrated into the overall optimization process, rather than improving one metric at the detriment of others. This entails strategically manipulating the physical design flow to impede the integration of Trojan components.

In brief, this work has made the following major contributions:

- A holistic Bayesian design space exploration in view of standard physical design and layout-level HT prevention with hyper-parameters customization.
- A mosaic-inspired site-level physical layout ‘hardening’ strategy of timing-aware exploitable region decomposition by cell shifting and buffer insertion.
- A correlation analysis between PPA and pre-attack HT susceptibility prediction indicating possibilities of incorporating HT concern in the context of physical design.
- A robust result elevation compared with all top 3 winners of the ISPD 2023 Contest regarding HT prevention, design quality, and adherence to design rules.

The remainder of this paper is organized as follows. Section 2 provides background information and comprehensive modeling on metrics of physical design and HT susceptibility. Section 3 gives a detailed illustration of our physical design and security closure flow. Section 4 elaborates on the experiments on metric correlation and feasibility of the proposed algorithm. Section 5 concludes this work and makes expectations in potential future works.

2 PRELIMINARIES

2.1 HT susceptibility

As introduced beforehand, Trojan insertion detects critical assets, modifies the schematic with additional logical components, and finally finds available sites and routing resources to complete the implementation. Simulation of real-world HT ECO insertion is one credible way of the susceptibility evaluation. Especially in the form of Red Team vs. Blue Team [17], adversarial competitions will push the attack and defense techniques to the boundary, thus ensuring objectivity. This work utilized the Trojan insertion framework in the 2023 ISPD contest [7]. In detail, this framework takes a physical layout as input, makes multiple trials of Trojan insertion with multiple strategies, and returns a score based on the severity of violations encountered while inserting Trojans. Specific violation types and their point values are derived from industry best practices, such as timing violations, design rule checks (DRC) violations, or design failure[14]. Lower scores favor the defense as they indicate more significant challenges faced by attackers during insertion.

Due to the long period of black-box attack simulation, there are also acknowledged surrogate metrics to estimate the HT susceptibility are exploitable placement regions [18] and routing track utilization [15] An *exploitable region* is defined as a contiguous group of placement sites that are either empty or contain only fillers, decaps, or tap cells. The *threshold* of continuous placement sites in identifying exploitable regions is set to 20 and denoted by N_{th}, for consistency with the ISPD contest[7]. As shown in Figure 1(b) and (c), attackers would find these regions attractive for inserting additional Trojan components. To quantitatively assess the threat severity, we calculate three metrics across all exploitable regions: the maximum number of exploitable sites within a single region, the median number of sites per region, and the total number of sites summed across all regions. As for routing resources, tracks are the pre-defined parallel metal lines used to connect different circuit components, so a lower track utilization rate conduces to a secure defense.

2.2 Physical design closure

From a process perspective, as shown in Figure 2, physical design prepares a circuit for tape-out sequentially with multiple discrete point tools [13]. First, floorplanning defines the die area, the position of macro and IO pins, and regions for standard cells and routing tracks. Then, the power distribution network (PDN) is set to ensure the reliability and efficiency of power transmission. Placement assigns the location assignment of standard cells onto the layout.

Physical optimization enhances timing closure by gate sizing, buffering, and netlist modification. These two steps are usually performed iteratively in fine grain (PlaceOpt). CTS generates clock trees by balancing clock skew and minimizing insertion delay to satisfy the metrics of timing and power. The logical connections among macros, standard cells, clocks, and IOs are mapped to the physical wires and vias. Finally, Design Rule Check (DRC) refinement focuses on legalizing DRC violations with minimum displacement.

To ensure efficient and functional chip designs, the aforementioned flow is evaluated by fundamental metrics, namely physical design closure:

- Power (P) refers to the amount of electrical energy an IC consumes during operation. Minimizing power consumption is essential for battery-powered devices and heat malfunction prevention. This can be affected by PDN design, gate number, and sizing, and routed metal wire length. As PDN is usually pre-defined by power integrity engineers in real-world design flow, this work maintains the same PDN recipe provided in the 2023 ISPD Contest.
- Performance (P) metric encompasses speed and efficiency. With a preconfigured clock period, Worst Negative Slack (WNS) is a critical metric used in digital circuit design for analyzing timing constraints, specifically for setup time and hold time. It quantifies the margin of error, indicating how much "worse" the timing could become before a violation occurs in the time windows before and after the clock edge. A negative slack value is undesirable and needs to be addressed during design analysis.
- Area (A) refers to the physical size occupied by the circuit on the silicon wafer. A smaller area translates to more chips per wafer, reducing production costs. As introduced in the implementation flow, the die area can be shrunk in the floorplanning stage, but with a lower bound to ensure the margin density for PlaceOpt and routing,
- Design Rule Check (DRC), as introduced beforehand, is a fundamental and well-established real-world constraint. DRC is taken into serious consideration in this work, where the final physical layout is expected to clear all DRC violations. Automatic and manual DRC repairs are allowed.

2.3 Problem formulation

This section formally defines the layout-level HT prevention problem within the context of physical design:

Given the circuit netlist, design rules and specifications, and initial design configurations (including clock frequency and PDN), the objective is to generate an optimized physical design, denoted by P, that satisfies the following criteria:

- The design should minimize the traditional Power (P), Performance (P), and Area (A) metrics, collectively represented as a cost function $S_{PPA}(P)$.
- The design should be fortified against HT insertions. This is achieved by minimizing a cost function $S_{HT}(P)$, including surrogate metrics and simulated HT attack & defense score.
- The design must adhere to real-world manufacturing constraints. This translates to ensuring the layout is free of DRC violations, denoted by $C_{DRC}(P) = 0$.

In consistency with the ISPD 2023 Contest[7], the cost function $PPA(P)$ and $HT(P)$ are the weighted sum of relevant metrics normalized by ones of a baseline physical design $P_{baseline}$. The baseline is generated by a default full-flow without HT consideration, an example of which can be seen in Table 2. The normalization scheme ($S(P) = m(P)/m(P_{baseline})$, m stands for the raw value of a metric) discourages non-deterministic improvements with degradation in any other metrics.

The weighted cost function for physical design quality is:

$$\begin{aligned} S_{PPA}(P) &= \frac{1}{3}S_{Power}(P) + \frac{1}{3}S_{Performace}(P) + \frac{1}{3}S_{Area}(P), \\ S_{Performace}(P) &= \frac{1}{2}S_{HoldWNS}(P) + \frac{1}{2}S_{SetupWNS}(P). \end{aligned} \tag{1}$$

It is noted that the performance is evaluated by setup and hold WNS, which prefers positive values, and the larger the absolute value, the better. Hence, the normalized score for WNS will be different from other metrics ($S_{WNS}(P) = m(P_{baseline})/m(P)$).

The HT susceptibility comprises surrogate models (exploitable regions and track utilization) and simulated HT attack & defense scores:

$$\begin{aligned} S_{HT}(P) &= \frac{1}{3}S_{Surrogate}(P) + \frac{2}{3}S_{Attack}(P), \\ S_{Surrogate}(P) &= \frac{1}{2}S_{Exploitable}(P) + \frac{1}{2}S_{Track}(P). \end{aligned} \tag{2}$$

The maximum, median, and total number takes up $\frac{1}{3}$, $\frac{1}{6}$, and $\frac{1}{2}$ of the score of exploitable regions, respectively.

In sum, the holistic cost function of the problem in concern is

$$\begin{aligned} \min\ S(P) &= \frac{1}{2}S_{PPA}(P) + \frac{1}{2}S_{HT}(p), \\ \text{s.t. } C_{DRC}(P) &= 0, \\ S_{WNS}(P) &> 0. \end{aligned} \tag{3}$$

3 OUR METHODS

Figure 3 illustrates our proposed comprehensive optimization framework on layout-level HT-aware physical design. Holistic design space exploration is performed initially based on the configuration of extracted hyper-parameters in the implementation flow script. Bayesian optimization is employed to explore the design space automatically, with penalty terms added to the objective function to ensure the design's practicality. After the convergence of Bayesian optimization gives a feasible basic physical layout, we use a mosaic-inspired site-level strategy to perform advanced HT defense, which decomposes exploitable regions with cell shifting and buffer insertion.

3.1 Holistic Design Space Exploration

This section delves into how to achieve optimal design outcomes by efficiently navigating the vast design space defined by numerous hyper-parameters. Through a process of hyper-parameter customization, we seek to strike a Pareto improvement between achieving high-quality physical design and fortifying the design against potential HT insertions. This exploration leverages Bayesian optimization, a powerful technique ideally suited for navigating complex design spaces with intricate relationships between hyper-parameters. The following subsections will delve deeper into the specific hyper-parameters employed, the rationale behind employing Bayesian optimization,

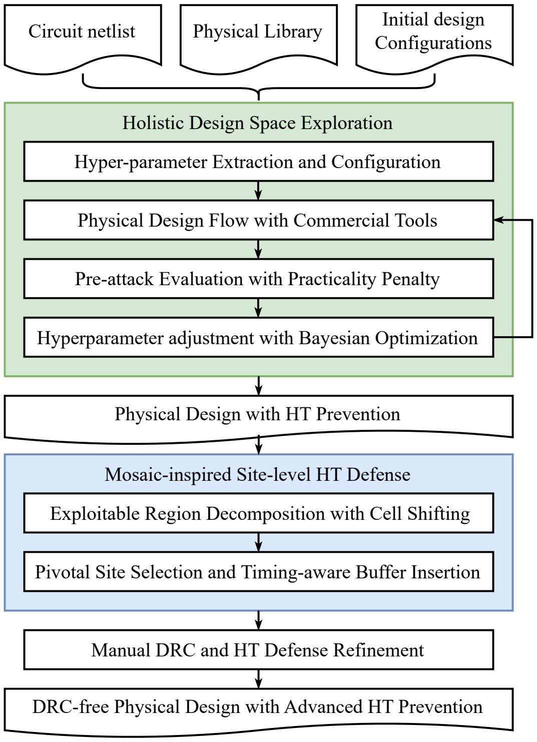

Circuit netlist
Physical Library
Initial design Configurations
Holistic Design Space Exploration
Hyper-parameter Extraction and Configuration
Physical Design Flow with Commercial Tools
Pre-attack Evaluation with Practicality Penalty
Hyperparameter adjustment with Bayesian Optimization
Physical Design with HT Prevention
Mosaic-inspired Site-level HT Defense
Exploitable Region Decomposition with Cell Shifting
Pivotal Site Selection and Timing-aware Buffer Insertion
Manual DRC and HT Defense Refinement
DRC-free Physical Design with Advanced HT Prevention

Figure 3: The framework of our automated physical design flow with layout-level HT prevention.

Table 1: Settings of Hyper-parameters extracted for Bayesian optimization

Parameters		Min	Max
setDesignMode	congEffort	low	high
	flowEffort	standard	extreme
	powerEffort	none	high
setOptMode	setupTargetSlack	0.02	0.04
	holdTargetSlack	0.02	0.04
Scaling	rowScale	0.85	1.1
	clkScale	0.6	1

and the practical implementation details of this approach in our work.

3.1.1 Hyper-parameter in physical design flow. As introduced in Section 2.2, the physical implementation of an integrated circuit is based on a flow with multiple discrete point tools, such as floorplanners, placers, routers, physical optimizers, and DRC legalizers [13]. Common commercial tools activate tools following a user-customized script, in TCL format, for instance. To fulfill varied combinations of objectives, interfaces are pre-designed out of algorithms of each step of the implementation flow. Hence, the problem is decomposed into what hyper-parameters can be customized and how to achieve a reasonable set.

Algorithm 1 Sequential Model-Based Optimization

```
Input: $f, \mathcal{X}, \mathcal{S}, \mathcal{M}, T$
Output: $\mathcal{H}$
Initiate $\mathcal{H}$ with $\mathbf{x} \subset \mathcal{X}$
for $t \leftarrow 1$ to $T$ do
    Update $p_{\mathcal{M}}(y|x, \mathcal{H})$ by $\mathcal{M}, \mathcal{H}$
    $x^* \leftarrow \arg\min_x S(x, p_{\mathcal{M}}(y|x, \mathcal{H}))$
    $y^* \leftarrow f(x^*)$
    $\mathcal{H} \leftarrow \mathcal{H} \cup (x^*, f(x^*))$
end for
```

The hyper-parameters concerned in this work are summarized in Table 1, which can be categorized into two groups: explicit options of commercial tools (`DesignMode` and `OptMode`) and the self-scaling factor of initial settings. On the one hand, explicit options of commercial tools such as `DesignMode` focus on the optimization effort balancing among metrics, whereas `OptMode` controls the algorithm objective of slack closure. On the other hand, the self-scaling factors of initial settings serve as supplemental prerequisites, such as reducing layout area and achieving timing closure. In general, rowScale squeezes layout area by adjusting the row length and row count; clkScale pushes the expected clock period to the extreme to leave abundant slacks for timing closure. It is noted that clkScale only affects the physical design stage; the clock period in WNS evaluation remains the same. Although these hyper-parameters seem irrelevant to HT resilience, it shall be seen through correlation analysis in Section 4.1 that fine-tuning of physical design parameters will also affect surrogate HT defense scores with a similar trend of variation.

3.1.2 Bayesian optimization. Given the dimension and scope of hyper-parameters, Bayesian optimization emerges as the ideal choice for hyper-parameter tuning/search. Bayesian optimization models the probability distributions of existing history explorations [4], and relevant algorithms have long been used in physical design exploration with proven efficiency and effectiveness[9, 16, 20, 22]. It uses Bayes' theorem to iteratively update its knowledge of the objective function and select the next point to evaluate [1]. Typically, Bayesian optimization is suitable for problems with the following characteristics:

- Unknown objective. Gradient-based methods are not applicable since the relationship between the hyper-parameters to be tuned and the final objective function of physical implementation is neither concave nor derivative.
- Expensive evaluation. A full flow of physical design usually takes hours or even days to get a feasible solution, leading to concerns about the runtime cost of result evaluation.
- Fair Continuity. A slight change in a single parameter might not cause bumpy turbulence in final metrics, meaning that a surrogate model can approximate the objective function.
- Moderate number of parameters. The dimension of inputs for Bayesian should not be too high (usually under 20), which is appropriate for our physical design exploration problem.

In conclusion, the task of physical design hyper-parameters tuning is compatible with Bayesian optimization.

The kernel mechanism is elaborated in Algorithm 1 on how to iteratively append new hyper-parameter sets based on history explorations and update the select criterion accordingly. For the mathematical symbols, f stands for the time-consuming precise objective function; $\mathcal{X}$ is the exploration space of the hyper-parameters; $\mathcal{S}$ is the acquisition function that determines the next exploration; $\mathcal{M}$ is the surrogate model approximating f yet easy to be calculated; T is the maximum number of inner iterations. Firstly, history explorations $\mathcal{H}$ are initiated with random attempts in $\mathcal{X}$ and corresponding evaluation of f. Upon $\mathcal{H}$, the probabilistic model $p(y|x, \mathcal{H})$ is updated. Then, we determine the subsequent point to be assessed according to the acquisition function, a mapping that transforms elements from the input space into a single scalar value. The expansion of $\mathcal{H}$ continues until the upper limit of iterations is met. A common acquisition function is expected improvement (EI):

$$\begin{aligned} I(x) &= \max(y^{'} - y, 0), \\ EI_{y^{'}}(x) &= \int_{-\infty}^{\infty} I(x) p_{\mathcal{M}}(y|x, \mathcal{H}) dy. \end{aligned} \tag{4}$$

The improvement $I(x)$ refers to the average potential new exploration that might bring, relative to the threshold $y^{'}$, whereas $EI_{y^{'}}(x)$ stands for the expectation of $I(x)$. The formulation of EI dictates a preference for regions with low mean and high variance, reflecting the exploration-exploitation trade-off. The surrogate model depicts the expensive objective function with the distribution, enabling an affordable understanding of the potential range of fluctuations. A common model Tree-structured Parzen Estimator (TPE), which converts the derivation $p_{\mathcal{M}}(y|x, \mathcal{H}))$ into that by Bayes' theorem:

$$\begin{aligned} p(y|x) &= \frac{p(x|y)p(y)}{p(x)}, \\ p(x|y) &= \begin{cases} l(x), & y < y^{'} \\ g(x), & y \geq y^{'} \end{cases}. \end{aligned} \tag{5}$$

TPE constructs different distributions for observations on either side of the threshold, that is, either a "good" score or a "bad" score. The threshold $y^{'}$ usually is set upon a quantile γ so that $p(y < y^{'}) = \gamma$. Particularly, the runtime of each iteration is scaled to $|\mathcal{H}|$ by sorting existing explorations in $\mathcal{H}$.

3.1.3 Framework implementation. Particularly in this work, the algorithms were implemented based on the open-source library `Hyperopt` [5]. The quantile threshold in Equation 4 is set to 0.25 as default in `Hyperopt`. First, we utilize commercial tools in a default physical design flow as introduced in Section 2.2 to initiate a solution. Quantified scores are given for each result based on analyzing the PPA and HT prevention as modeled in Section 2.3. It is noted that the simulated HT attack involves various types of HT, insertion methods, and failure detection; it has a high time cost. Besides, in the real world, the ECO Trojan insertion behavior should remain unknown in the preventive defense stage, Hence, at this stage, the HT susceptibility is evaluated using only the surrogate model ($S'_{HT}(P) = S'_{Surrogate}(P)$). Then, the holistic framework adjusts the hyper-parameters based on the metric feedback, makes corresponding modifications to TCL scripts, and launches another trial. The overall process ceases when the preset number of attempts reaches the upper limit so that a test case runs for no more than 2-3 days.

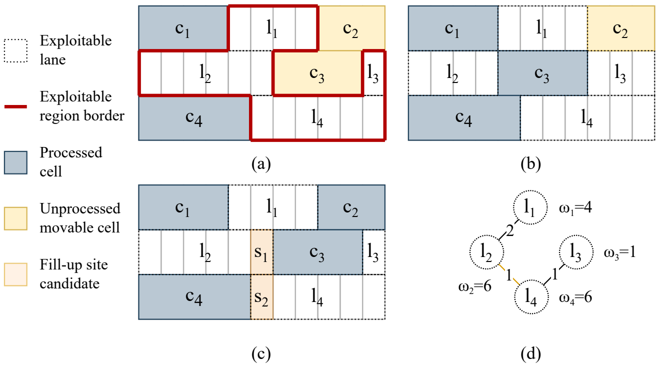


Figure 4: Mosaic site-level refinement (a) Original state. Four exploitable lanes $l_1 - l_4$ are connected to form an exploitable region. Cells c_1, c_2 have already been addressed and fixed. (b) After cell shifting. Cell c_3 has been moved to two sites to the left, cutting the connection between l_2 and other lanes. (c) Candidate sites for buffer insertion. Due to timing considerations, this works if cell shifting cannot break the exploitable region. (d) Clustered undirected graph of lane connection. The connection between l_2 and l_4 is chosen to be cut off on candidate sites, corresponding to s_1 and s_2 in (c).

In addition, penalty settings on DRCs and WNS ensure the validity of the final physical design. As introduced in Section 2.2, a layout at the sign-off level should be DRC-free with positive WNS. Hence, in the Bayesian optimization flow, we intentionally assign results a considerable penalty value with 1) a manually unfixable number of DRC violations (50 by experience) or 2) a negative WNS. The empirical DRC threshold results from the number of DRC violations left by current commercial tools at the detailed routing after multiple iterations of legalization. The remaining violations can addressed with disruptions of human experience to the site placement at the minimum cost of the already accomplished metrics. However, if there are too many, the manual labor will be not only cumbersome and time-consuming, but also more likely to introduce unintended consequences, leading to a cascading effect of generating new violations.

3.2 Mosaic-inspired site-level HT defense

After Bayesian optimization converges to a feasible basic physical layout, we employ a mosaic-inspired site-level strategy for advanced HT defense. The site-level refinement is named "mosaic-inspired" as the atomic operations mimic mosaic tiles addressing. Given an initial design with a fair overall score, the mosaic-inspired strategy series provides further defense against Trojans. We decompose an exploitable region by rows and denote the continuous unoccupied sites intra-row as *lanes* ($l_1 - l_4$) in Figure 4(a). To eliminate an exploitable region, we can either cut off the connection or shorten the length of each lane with cell shifting and buffer insertion. As the performance metric WNS is susceptible to cell disturbance, extra focus is on how to achieve exploitable region decomposition with minimum site-level operations.

3.2.1 Exploitable Region Decomposition via Standard Cell Shifting. As shown in Algorithm 2, we slide cell $c_i \in C$ from all cells C one by one in the adjacent exploitable lanes in its row $r_{c_i} \in \mathcal{R}, c_i \subset r_{c_i}$. The left x-coordinate x_i of cell c_i is optimized by walking through

Algorithm 2 Sequential Cell Shifting

```
Input: $N_{th}, N_t, \mathcal{R}, C, \mathcal{W}, \mathcal{X}$.
Output: Optimized x-coordinate $\mathcal{X}$.
 1: Sort $C$ by $\mathcal{X}$.
 2: Initialize $\mathcal{P}$ with layout left border x-coordinate.
 3: for $c_i \in C$ do
 4:     $c_j \leftarrow$ rightNeighbor($c_i$)
 5:     $x_i^{max} \leftarrow \min(p_{r_{c_i}-1}, p_{r_{c_i}+1}, p_{r_{c_i}} + N_{th} - 1, x_r - w_i)$
 6:     $x_i^{min} \leftarrow p_{r_{c_i}}$
 7:     if $x_i^{min} < x_i^{max}$ then
 8:         $x_i^{'} \leftarrow$ optPosition($x_i^{min}$, $x_i^{max}$)
 9:         if $|x_i^{'} - x_i| \leq N_t$ then
10:             $x_i \leftarrow x_i^{'}$
11:         end if
12:     end if
13:     $p_{r_{c_i}} \leftarrow x_i + w_i$
14: end for
```

a range $[x_i^{min}, x_i^{max}]$. x_i^{min} is set as the right x-coordinate of c_i's nearest cell in the left. x_i^{max} should ensure that c_i will not overlap with its right neighbor c_j, which is less than $x_j - w_i$, where w_i is the cell width. To save redundant coordinate calculations, cells are sorted by their x-coordinates in an ascending order. Then, a vector $\mathcal{P}$ is developed; each element $p_r \in \mathcal{P}$ stores the right x coordinate $x_i + w_i$ of the last processed cell c_i in row r.

Given the order and $\mathcal{P}$, $p_{r_{c_i}}$ can directly be used as the left end of the exploring range x_i^{min} as in line 6. As in line 5, x_i^{max} should be limited to $p_{r_{c_i}} + N_{th} - 1$, to prevent new exploitable regions on the left of c_i. Besides, cell shifting is expected to separate adjacent exploitable lanes by bordering on cells upper left and lower left in the adjacent rows, as c_3 and c_1 have corners attached in Figure 4(b). Hence, x_i^{max} should also consider the cell coordinates in the adjacent rows, that is $\min(p_{r_{c_i}-1}, p_{r_{c_i}+1})$. Suppose there are several candidate coordinates for the cell that cut off the same number of lane connections. In that case, we prefer the answer with the minor displacement from the original position to avoid potential WNS disturbance, which is the `optPosition` in line 8. In addition, the range $[x_i^{min}, x_i^{max}]$ deducted above is possibly less than x_i as cells are addressed in the x coordinates order, which might cause a left shifting on all cells. The overall shifting trend will lead to higher partial density and lower cell routability. A reference displacement criteria N_t is set to decide whether to accept a cell's shifting, Throughout the algorithm, the coordinates of each cell are optimized within their rows to decompose exploitable regions.

3.2.2 *Candidate Site Selection for Buffer Insertion.*

The displacement constraint in cell shifting may leave plenty of exploitable regions unresolved. Yet, they can cracked by filling buffers into critical sites as depicted in Figure 4(c). We shall introduce buffers in Section 3.2.3. This section will focus on selecting sites to be filled to decompose the exploitable regions by either 1) breaking long lanes into small segments or 2) breaking the connection of inter-row lanes.

As for intra-row operations, sites are scored and sorted in descending order in lanes longer than N_{th}. Then, we traverse the sites and label them as candidates to be filled until all remaining lane segments are shorter than N_{th}. The scoring considers the availability of vertical adjacent sites and periodic threshold intervals. The former part is similar to that used in cell shifting in work [10], where sites are favored that possibly cut off connections with adjacent lanes. In addition, when a candidate is chosen with each cut, the scoring is multiplied by a coefficient related to N_{th} and the remaining length of the lane. This periodic coefficient ensures that a cut segment can approach the limit N as closely as desired, and lanes can be cut with the fewest candidates.

As for inter-row operations, candidate site selection is based on the vertical connections among the lanes in the remaining exploitable regions. We abstract the connection relationship into a weighted undirected graph $G(V, E)$ as illustrated in Figure 4(d). Every vertex denotes a lane whose weight is the lane width ($\omega_i = w_i$). An edge represents vertical connections of lanes, with weight being the width of the borderline. The vertices are clustered to determine the critical edges to be cut; a better solution should have fewer edges with less edge weight cost. For example, vertices shall be clustered into (l_1, l_2) and (l_3, l_4) in Figure 4(d), so that the exploitable region can be decomposed by cutting only one connection between l_2 and l_4 at the weight of one.

Given the target connections, such as s_1 and s_2 both can split l_2 and l_4 in Figure 4(c), the next task is to choose sites of which lane as the candidate. This is determined by whether there are intra-row candidates in the connection zone: If only one lane has candidates on the borderline, choose this lane; If both lanes have, choose the lower lane; If no lanes have, choose the longer lane.

3.2.3 *Optimal Buffer Selection for Candidate Site Filling.*

The pivotal sites selected from exploitable regions shall be filled with minimum disturbance to WNS and without functional modification. Buffers, a type of logic gate, come in various widths, as documented in the standard cell library. Buffers temporarily store and regenerate a digital signal, improving the signal timing and integrity by isolating the signal from noise and distortion. As a result, proper insertion of buffer can meet the above requirement.

However, excessive buffer insertion can lead to a potential sudden disturbance in critical path timing or DRC violation repairing difficulty, which requires additional care mapping critical sites to buffers. For instance, sites occupied by newly added buffers can be freed if no additional exploitable regions are introduced. These sites are traversed with the breadth-first search from leaf vertices in the clustered undirected graph. Then, we merge adjacent available sites into the discrete candidate sites to meet the minimum width of buffers. In addition, closely placed buffers can be replaced by only one large buffer to reduce the number of influenced nets. Suppose some successions of sites were labeled for buffer insertion, but no buffer type in the library is of their size. In that case, we use a preprocessed combination of buffer types and map target sites to a bundle with minimum buffer number and size proximity. Finally, new buffers are intentionally inserted into non-critical paths to avoid timing degrades.

4 EXPERIMENTAL RESULTS

4.1 Metric correlation analysis

Here, we would like to share the correlation analysis between physical design and HT prevention metrics based on the trials in Bayesian

Table 2: Baseline metrics of surrogate HT prevention S_{HT} and physical design quality S_{PPA}

Test cases	Track Util.	Sites in exploitable region			Power	Performance		Area
		Max	Med	Sum		Setup WNS	Hold WNS	
AES128	63.89	460741	29	662065	396.56	34.286	33.247	676407.6
Camellia	58.76	1403	46	8820	20.45	22.233	20.818	25039.90
CAST	62.44	139439	30	147359	16.88	25.493	18.216	85989.70
MISTY	65.54	5932	33	26039	5.89	4.999	20.968	30429.31
SEED	65.77	3854	33	25478	9.89	34.22	29.35	42782.79
SHA256	68.26	2133	25	25964	11.13	20.8	22.631	36343.61

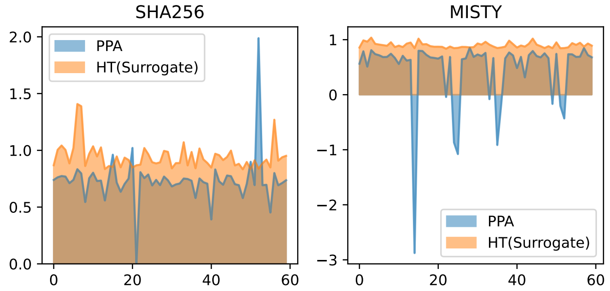


Figure 5: The variation of the physical design and surrogate HT prevention score throughout iterations

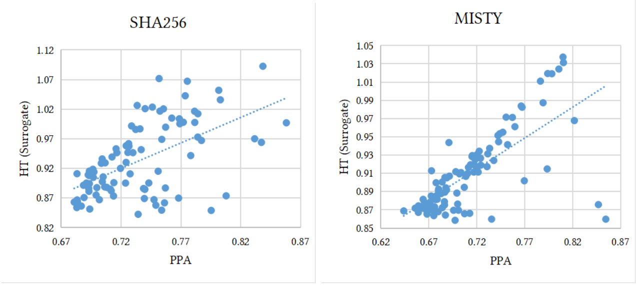


Figure 6: Correlation analysis between physical design and surrogate HT prevention metrics

optimization. As explained in Section 3.1.3, the surrogate HT prevention objective function $S_{Surrogate}(P)$ should consist of only track utilization and exploitable regions.

Figure 5 illustrates how metric scores vary with the iterations in the Bayesian optimization in the case SHA256 and MISTY. The optimization flow initiates with wild fluctuation, representing an extensive solution space exploration. It is noted that the negative value in the PPA score of MISTY is caused by illegal negative WNS. The absolute value of the negative value decreases with iteration, thanks to the practicality penalty. Besides, the PPA and HT prevention share the same variation tendency in each iteration. The exploitable regions and track utilization might be an inner correlation with regular physical design closure, leading to a potential comprehensive problem modeling of security-aware physical implementation.

Figure 6 depicts the outcome in pairs of PPA and surrogate HT scores of case SHA256 and MISTY with a scatter plot. Note that outliers are filtered for simplicity of presentation. The fair correlation between the two metrics confirms the hypothesis that both terms in the objective function can be optimized simultaneously without sacrificing either. Therefore, while the hyper-parameters may appear unrelated to hardware security, comprehensive optimization of the physical implementation can still provide a viable solution that holistically considers both PPA metrics and HT concerns.

4.2 Overall metrics

The discrete stages throughout the physical implementation process were performed with a scriptable commercial tool Cadence Innovus. The version of Innovus used in the following experiment was `v21.36-s078_1`. In addition, the mosaic-inspired site-level defense is implemented in C++, with an output Innovus TCL script to execute cell movement and buffer insertion. The experiment was conducted on a Linux workstation with Intel® Xeon® Gold 6240 CPU @ 2.60GHz; 8 CPU jobs allowed with the current license(s). The technology library for physical implementation is ASAP7 PDK[19], a predictive process design kit at the 7-nm finFET technology node. As a collaborative work from Arizona State and ARM, ASAP7 PDK's distinguished completeness won fair academic acknowledgment. The test cases remain the same as those in the ISPD 2023 Contest[7], including six crypto cores that require extra attention to counter backdoor attacks.

The baseline metrics are shown in Table 2, including surrogate HT prevention metrics and traditional PPA. The physical design results of the top 3 winners are downloaded from the ISPD 2023 contest website and evaluated locally with the official script and our version of Innovus. Table 3 presents the normalized score in major metrics concerned of our framework (*Ours*) and that of the contestants (*1st*, *2nd*, and *3rd*). As exploitable regions have all been cleared in most of the test cases, the score of track utilization and exploitable region sites merged into one block *Surrogate* for conciseness. However, manual work can only fix a limited portion of exploitable regions; automatic algorithms should address most exploitable regions in advance. We set the raw value of the remaining exploitable regions aside in Table 4 for reference.

In sum, our proposed algorithms demonstrably outperform the top 3 winners, achieving the highest overall score across most test cases. Additionally, our framework consistently delivers fair results when considering individual metrics for each case. It is also observed that eliminating exploitable sites does help in defending against

Table 3: Normalized metrics comparison with top 3 winners in ISPD 2023 contest

Metrics		AES128	Camellia	CAST	MISTY	SEED	SHA256	Average	Ratio
Surrogate	Ours	**0.471905**	0.468772	**0.444267**	0.477648	0.445644	0.421697	**0.454989**	1
	1st	0.558642	0.468687	0.445708	0.469561	0.445644	0.461471	0.474952	1.044
	2nd	0.535251	0.476090	0.532042	0.458347	0.459328	0.435248	0.482718	1.061
	3rd	1.000000	**0.338241**	0.633098	**0.338801**	**0.354113**	**0.342148**	0.501067	1.101
ECO insertion	Ours	**0.132715**	**0.098761**	**0.129625**	0.119336	**0.098761**	**0.098761**	**0.112993**	1
	1st	0.148146	0.119336	0.139912	**0.098761**	0.119336	**0.098761**	0.120709	1.068
	2nd	**0.132715**	**0.098761**	0.222217	**0.098761**	0.160489	**0.098761**	0.135284	1.197
	3rd	0.333330	**0.098761**	0.222217	**0.098761**	**0.098761**	0.181064	0.172149	1.524
HT prevention	Ours	**0.245778**	0.222097	**0.234505**	0.238773	0.214388	**0.206406**	**0.226991**	1
	1st	0.284978	0.235786	0.241844	0.222360	0.228105	0.219664	0.238790	1.052
	2nd	0.266893	0.224537	0.325491	0.218622	0.260101	0.210923	0.251095	1.106
	3rd	0.555553	**0.178587**	0.359176	**0.178774**	**0.183878**	0.234759	0.281788	1.241
Power	Ours	0.942345	**0.912287**	**1.013523**	**0.870173**	0.998168	**0.942345**	**0.946474**	1
	1st	**0.942343**	0.912291	1.099824	0.961550	0.998177	0.942422	0.976101	1.031
	2nd	1.000365	1.003390	1.096568	0.935398	**0.947368**	1.051632	1.005787	1.063
	3rd	1.000000	1.040791	1.067541	0.898705	0.992096	1.002733	1.000311	1.057
Performance	Ours	0.793184	**0.494661**	**0.250159**	0.257848	0.304492	0.275911	**0.396043**	1
	1st	0.793184	0.494665	0.297967	**0.254900**	**0.304233**	0.254518	0.399911	1.010
	2nd	**0.759775**	0.528075	0.573674	0.375165	1.210692	**0.246185**	0.615594	1.554
	3rd	1.000001	0.629130	0.646832	0.393500	1.102852	0.434175	0.701082	1.770
Area	Ours	**0.777597**	**0.893780**	0.765237	0.811693	0.888427	0.847666	0.830733	1
	1st	**0.777597**	**0.893780**	0.733358	0.822887	0.888427	0.837267	0.825553	0.994
	2nd	0.800929	0.932913	**0.727063**	**0.767682**	**0.802048**	**0.786232**	0.802811	**0.966**
	3rd	1.000000	1.000000	0.797794	0.789677	0.928235	0.889906	0.900935	1.085
PPA	Ours	**0.837707**	**0.766908**	**0.676305**	**0.646571**	0.730361	0.688639	**0.724415**	1
	1st	**0.837707**	0.766911	0.710382	0.679778	**0.730278**	**0.678068**	0.733854	1.013
	2nd	0.853688	0.821457	0.799101	0.692747	0.986702	0.694682	0.808063	1.115
	3rd	1.000000	0.889972	0.837388	0.693960	1.007726	0.775604	0.867442	1.197
Overall	Ours	**0.541743**	**0.494503**	**0.455405**	0.442672	**0.472375**	**0.447523**	**0.475704**	1
	1st	0.561343	0.501349	0.476113	0.451070	0.479192	0.448866	0.486322	1.022
	2nd	0.560291	0.522997	0.562296	0.455685	0.623402	0.452803	0.529579	1.113
	3rd	0.777777	0.534280	0.598283	**0.436367**	0.595802	0.505181	0.574615	1.208

Table 4: Remaining exploitable region statistics

Source	Test cases	Max	Med	Sum
1st	AES128	92	30	1320
2nd	AES128	96	24	14251
	CAST	2780	28	13422
3rd	AES128	460741	29	662065
	CAST	49642	29	76756

simulated HT attacks. However, over-minimizing surrogate models, especially track utilization, can lead to similar scores in ECO insertion, as seen in the 3rd place team's results for some cases. Overall, our approach strikes a holistic optimization in HT defense and design quality, resulting in a stable, practical, and high-quality layout-level HT defense within the physical design flow.

5 CONCLUSION

This work aims to defend hardware implementation against malicious threats while preserving and even enhancing essential physical design closure. Attempts have been made in automated design space exploration with hyper-parameter abstraction and Bayesian optimization. In addition, site-level exploitable region decomposition and reallocation refined the HT awareness. The feasibility of this framework is proven with test cases in the ISPD 2023 contest, where our result outperformed the top 3 winners in the overall metrics of PPA and layout-level HT defense. As previous experiments reveal the correlation between PPA and surrogate HT prevention, future research shall concentrate on meta-analysis into the interplay among all segmented metrics and real-world HT attack & defense, working as a foundation for a universal methodology ready for ever-changing challenges.

REFERENCES

[1] T. Agrawal and T. Agrawal. Bayesian Optimization. *Hyperparameter Optimization in Machine Learning: Make Your Machine Learning and Deep Learning Models More Efficient*, pages 81–108, 2021.

[2] A. Alaql, M. M. Rahman, and S. Bhunia. SCOPE: Synthesis-based constant propagation attack on logic locking. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 29(8):1529–1542, 2021.

[3] P.-S. Ba, S. Dupuis, M. Palanichamy, M.-L. Flottes, G. Di Natale, and B. Rouzeyre. Hardware Trust through Layout Filling: A hardware Trojan Prevention Technique. In *Proceedings of 2016 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pages 254–259, 2016.

[4] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl. Algorithms for Hyper-Parameter Optimization. In *Proceedings of the 24th International Conference on Neural Information Processing Systems*, page 2546–2554, 2011.

[5] J. Bergstra, D. Yamins, and D. Cox. Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28, pages 115–123, 2013.

[6] S. Bhunia, M. S. Hsiao, M. Banga, and S. Narasimhan. Hardware Trojan attacks: Threat analysis and countermeasures. *Proceedings of the IEEE*, 102(8):1229–1247, 2014.

[7] M. Eslami, J. Knechtel, O. Sinanoglu, R. Karri, and S. Pagliarini. Benchmarking Advanced Security Closure of Physical Layouts: ISPD 2023 Contest. In *Proceedings of the 2023 International Symposium on Physical Design*, page 256–264, 2023.

[8] M. Eslami, T. Perez, and S. Pagliarini. SALSy: Security-Aware Layout Synthesis, 2023.

[9] H. Geng, T. Chen, Y. Ma, B. Zhu, and B. Yu. PTPT: Physical Design tool Parameter Tuning via Multi-Objective Bayesian Optimization. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 42(1):178–189, 2023.

[10] G. Guo, H. You, Z. Tang, B. Li, C. Li, and X. Zhang. ASSURER: A PPA-Friendly Security Closure Framework for Physical Design. In *Proceedings of the 28th Asia and South Pacific Design Automation Conference (ASP-DAC)*, page 504–509, 2023.

[11] J.-W. Hsu, K.-C. Chen, Y.-S. Chen, Y.-H. Lo, and Y.-W. Chang. Security-Aware Physical Design against Trojan Insertion, Frontside Probing, and Fault Injection Attacks. In *Proceedings of the 2023 International Symposium on Physical Design (ISPD)*, page 220–228, 2023.

[12] D.-C. Huang, C.-F. Hsiao, T.-W. Chang, and Y.-Y. Chu. A security method of hardware Trojan detection using path tracking algorithm. *EURASIP Journal on Wireless Communications and Networking*, 2022(1):1–21, 2022.

[13] A. B. Kahng, J. Lienig, I. L. Markov, and J. Hu. *VLSI physical design: from graph partitioning to timing closure*, volume 312. Springer, 2011.

[14] J. Knechtel. Evaluation of Advanced Security Closure of Physical Layouts. https://wp.nyu.edu/ispd23_contest/evaluation/, 2023.

[15] J. Knechtel, J. Gopinath, M. Ashraf, J. Bhandari, O. Sinanoglu, and R. Karri. Benchmarking Security Closure of Physical Layouts: ISPD 2022 Contest. ISPD '22, page 221–228, New York, NY, USA, 2022. Association for Computing Machinery.

[16] Y. Ma, Z. Yu, and B. Yu. CAD Tool Design Space Exploration via Bayesian Optimization. In *2019 ACM/IEEE 1st Workshop on Machine Learning for CAD (MLCAD)*, pages 1–6, 2019.

[17] E. Puschner, T. Moos, S. Becker, C. Kison, A. Moradi, and C. Paar. Red team vs. blue team: A real-world hardware trojan detection case study across four modern cmos technology generations. In *Proceedings of 2023 IEEE Symposium on Security and Privacy (SP)*, pages 56–74, Los Alamitos, CA, USA, may 2023. IEEE Computer Society.

[18] T. Trippel, K. G. Shin, K. B. Bush, and M. Hicks. ICAS: an Extensible Framework for Estimating the Susceptibility of IC Layouts to Additive Trojans. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 1742–1759, 2020.

[19] V. Vashishtha, M. Vangala, and L. T. Clark. ASAP7 predictive design kit development and cell design technology co-optimization: Invited paper. In *Proceedings of the 2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 992–998, 2017.

[20] F. Wang, Q. Wang, B. Fu, S. Jiang, X. Zhang, L. Alrahis, O. Sinanoglu, J. Knechtel, T.-Y. Ho, and E. F. Young. Security Closure of IC Layouts Against Hardware Trojans. In *Proceedings of the 2023 International Symposium on Physical Design (ISPD)*, page 229–237, 2023.

[21] X. Wei, J. Zhang, and G. Luo. GDSII-Guard: ECO Anti-Trojan Optimization with Exploratory Timing-Security Trade-Offs. *Proceedings of 2023 60th ACM/IEEE Design Automation Conference (DAC)*, pages 1–6, 2023.

[22] H. Wöhrle, F. Schneider, F. Schlenke, D. Lebold, M. De Lucas Alvarez, F. Kirchner, and M. Karagounis. Multi-Objective Surrogate-Model-Based Neural Architecture and Physical Design Co-optimization of Energy Efficient Neural Network Hardware Accelerators. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 70(1):40–53, 2023.

[23] K. Xiao and M. Tehranipoor. BISA: Built-in self-authentication for preventing hardware trojan insertion. In *Proceedings of 2013 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, pages 45–50, 2013.