Hybrid Obfuscation of Chiplet-Based Systems

Yousef Safari[1], Pooya Aghanoury[2], Subramanian S. Iyer[3], Nader Sehatbakhsh[2], and Boris Vaisband[1]
[1]THInK Team, ECE Department, McGill University, Montreal, QC, Canada
[2]SsysArch Lab, ECE Department, University of California, Los Angeles, CA, USA
[3]CHIPS, ECE Department, University of California, Los Angeles, CA, USA

2023 60th ACM/IEEE Design Automation Conference (DAC) | 979-8-3503-2348-1/23/$31.00 ©2023 IEEE | DOI: 10.1109/DAC56929.2023.10247738

***Abstract*—The growing concern about offshore chip manufacturing has created considerable interest in solutions that can ensure the integrity and security of chips. Among various solutions, split manufacturing has received a lot of attention due to its security guarantees. With the recent emergence of new heterogeneous manufacturing technologies, including chiplet-based systems, there is a new opportunity for revisiting the design considerations for split manufacturing to fully exploit the opportunities presented by chiplet-based systems and improve various metrics, such as security, performance, and overhead.**

This work improves the state-of-the-art in secure chip manufacturing by proposing a new split manufacturing scheme. The key idea is to exploit the capabilities provided by chiplet integration technology for designing a new *hybrid* split manufacturing scheme that includes both vertical and horizontal splitting. Unlike existing vertical-only split manufacturing mechanisms, that target obfuscation of interconnections by splitting the design at a specific metallization layer into two portions, the proposed hybrid method increases trust by exploiting the chiplet paradigm shift, specifically, breaking the design into sub-designs, each represented by chiplets (independently fabricated), and obfuscating interconnections among them. The proposed obfuscation mechanism targets systems that exploit the chiplet technology to obtain important performance advantages, thus any chiplet-related overhead is not due to obfuscation. We evaluate our method using several experiments and compare it with the state-of-the-art using standard metrics, including area, power, delay, wirelength, and trust. Compared to conventional split manufacturing, our hybrid method achieves up to 245× higher trust, while exhibiting negligible overhead.

***Index Terms*—Hardware security, split manufacturing, chiplet-based integration, Split-Fabric.**

I. Introduction

To combat the growing concern about secure chip manufacturing [1], methods based on obfuscating the manufacturing process, specifically based on splitting the manufacturing steps, have been recently proposed [1], [2], [3], [4], [5], [6].

In split-manufacturing [2], the most mature *macro-level* hardware security approach, designs are obfuscated by splitting the layout *vertically* into two portions, front-end of line (FEOL) and back-end of line (BEOL). The FEOL includes devices and low metal layers, and the BEOL consists of the remaining (higher) metal layers. The FEOL and BEOL are fabricated as two separate ICs by two different foundries and then bonded at a trusted facility. In this approach, the complete design is exposed only to the trusted integration facility. Thus, split-manufacturing provides fabless industry with a trusted access to advanced technology nodes. This improved level of trust, however, comes at the cost of a significantly high performance overhead since each terminated node on the FEOL chip must be connected to the corresponding node on the BEOL chip through a highly parasitic path (packaging layers of the BEOL and FEOL chips, and solder-based bonding).

The emergence of new integrations technologies, particularly chiplet-based systems, has created a new opportunity to revisit the existing assumptions and strategies for split manufacturing. Chiplet integration can be particularly helpful because it inherently *splits* a large monolithic system-on-chip (SoC) into smaller chiplets that can be independently manufactured.

Exploiting the chiplet paradigm-shift, a *hybrid* split manufacturing strategy, is proposed in this paper. Unlike state-of-the-art split-based

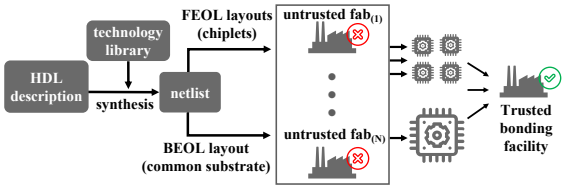


Figure 1: A schematic of the *hybrid* split manufacturing ecosystem for fabricating trusted hardware.

obfuscation (including vertical-only splitting of conventional CMOS ICs [7] and, more recently, using chiplets [3]), this work presents a comprehensive split strategy that includes *both* vertical *AND* horizontal splitting, resulting in significant security improvement while exhibiting minimal overhead.

Assuming the threat model that is depicted in Figure 1, the *hybrid* split manufacturing method ensures integrity and security of the manufacturing process by leveraging a three-step procedure: (1) fabrication of FEOL chiplets (small packaged dies) or dielets (small unpackaged dies), (2) fabrication of a BEOL chip, *i.e.,* a common substrate (interconnect fabric) that includes the BEOL metallization layers of chiplets and inter-chiplet interconnect, and (3) integration, *i.e.,* assembly of the FEOL chiplets on the BEOL substrate.

Effective and efficient employment of horizontal and vertical obfuscation is a twofold challenge. ***First***, previously proposed metrics (*e.g.,* K-isomorphism [8] and graph similarity [9]) for evaluating the trust level in split-manufacturing, are practical in a conventional packaging setting but do not support the chiplet-based systems. A chiplet-based platform features unique physical design attributes which can be exploited to cipher the original design. Therefore, new metrics for measuring trust need to be developed.

Second, achieving a horizontal split is challenging since in practice, the number of accessible foundries is limited (typically less than five for high-end ICs). Note that a chiplet-based platform can potentially accommodate thousands of chiplets in a waferscale assembly [10]. Establishing design guidelines is, therefore, crucial requirement to intelligently distribute the functionality of the system across chiplets. Trust-oriented distribution of functionality among chiplets will prevent a resourceful attacker (*i.e.,* with access to the layout of multiple FEOL chiplets) from accomplishing malicious activity.

This paper addresses the above challenges by introducing a new *metric* and a new *algorithm* for implementing a *hybrid* split manufacturing method for a given (monolithic) design.

To address the metric concern, we introduce *SplitScore*, a robust graph similarity-based metric that considers the unique features of chiplet-based platforms. We address the second challenge by designing an algorithm based on two new design guidelines called *InterSim* and *IntraSim*. These two design guidelines are used to maximize the benefit of horizontal and vertical obfuscation when available foundries are limited.

The key contributions of this paper are as follows.

- We propose a new *hybrid* method for split manufacturing by

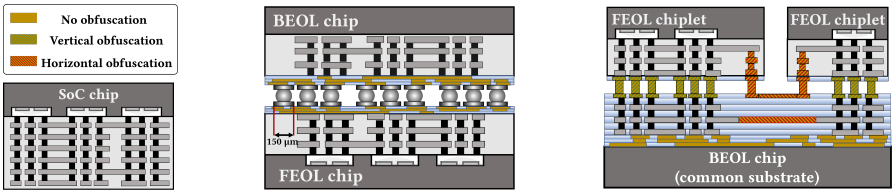


Figure 2: Illustration of an example reference SoC secured by utilizing different obfuscation approaches. (Left) Reference SoC prior to obfuscation, (Center) SoC secured by split-manufacturing, (Right) SoC secured by *hybrid* split manufacturing methodology.

employing both horizontal and vertical obfuscation mechanisms in chiplet-based systems, *i.e.*, systems that already rely on splitting the function into smaller ICs.
- We develop *SplitScore*, a robust graph similarity-based figure of merit to quantify levels of trust in chiplet-based systems obfuscated when two security mechanisms are employed in a *hybrid* scheme.
- We design and implement a new algorithm for *hybrid* splitting for a given design to maximize trust while minimizing overhead.
- We evaluate our design and implementation using two popular benchmark suites. Area, power, wirelength, and delay overheads are compared to state-of-the-art split manufacturing techniques.

The rest of the paper is composed of the following sections. The proposed *hybrid* obfuscation scheme is introduced in Section II. The methodology for designing *hybrid* split manufacturing scheme is described in Section III. The evaluation setup, simulation results, and discussion related to the scalability of the proposed approach, are described in Section IV. Finally, some concluding remarks are offered in Section V.

II. Hybrid Obfuscation Scheme

To prevent vulnerabilities, such as hardware Trojans [4], split manufacturing is proposed [7], where the circuit layout is split into two layers. In the front-end-of-line (FEOL) layer, there exist all the cells and interconnections in lower metal layers, while all the interconnections in higher metal layers are assigned to back-end-of-line (BEOL) layer. Because the fabrication of BEOL layers usually requires less advanced technologies, it is affordable to maintain such trusted foundries for the BEOL layer fabrication, by which important circuit information can be hidden to prevent Trojan insertions by untrusted foundries.

Since its introduction, several different split manufacturing frameworks have been proposed to improve various aspects of the design including security guarantees and overhead [1], [2], [3], [4], [5], [6].

Very recently, a method called Split-Fabric [3] has been proposed to extend the concept of split manufacturing to chiplet systems. Leveraging the advantages of chiplet integration, including smaller bonding size and faster interconnect, Split-Fabric was able to achieve similar security while achieving superior performance and lower overhead compared to prior split manufacturing techniques.

Observing that chiplet technology could potentially improve both security and performance, our key insight is that the security can be further improved by leveraging a *hybrid* split manufacturing scheme on chiplets instead of *only* relying on vertical obfuscation as proposed by previous work including Split-Fabric [3].

The proposed *hybrid* obfuscation scheme for a chiplet-based system utilizes both horizontal and vertical obfuscations, which are defined as follows.

Vertical obfuscation – an existing security mechanism that improves security by vertically dividing the layout of each chiplet (and/or conventional CMOS ICs) into two portions: (1) devices and lower metallization layers up to layer $\boldsymbol{M^s}$ (referred to as the FEOL chiplet/chip), and (2) remaining metallization layers (*i.e.,* above $\boldsymbol{M^s}$) as part of the layout of the common interconnect substrate (referred to as the BEOL chiplet/chip).

The level of trust in the vertical obfuscation mechanism is a function of $\boldsymbol{M^s}$. Smaller $\boldsymbol{M^s}$ means higher security, due to a greater number of open connections on the BEOL chip/chiplet (*i.e.,* greater solution space for the attacker). This also means higher performance overhead of the integrated system.

Horizontal obfuscation – Our key observation in this work is that the security can be further improved by dividing the layout of an initial design horizontally into several (at least two) sub-designs (i.e., several chiplets). Each sub-design includes a portion of the devices from the initial design, and all associated metallization layers for the realization of the internal connections within that sub-design. Each sub-design is fabricated as a separate FEOL chiplet. All other metallization layers, required for inter-chiplet connections, are embedded within the layout of the common substrate (BEOL chip). In other words, ***the horizontal obfuscation mechanism enables a trust-aware division of the system-on-chip (SoC) functions across a chiplet-based platform***.

The number of FEOL chiplets (sub-designs) affects the level of obfuscation in the system, but it is not a reliable metric for quantifying the level of trust of the horizontal obfuscation mechanism (*i.e.,* higher number of FEOL chiplets does not guarantee higher level of trust). Quantification of the level of trust in horizontal obfuscation is addressed in Section III.

An illustration of the existing split-manufacturing methodology versus our proposed *hybrid* scheme is provided in Figure 2. An example reference SoC with six metallization layers is shown in Figure 2(left). Utilization of the split-manufacturing and *hybrid* methodology to secure the reference SoC is illustrated in, respectively, Figures 2(middle) and 2(right).

Realization of an effective *hybrid* split manufacturing scheme requires two major new components: first, since existing metrics for measuring trust are not directly applicable to a *hybrid* scheme, new systematic methods for measuring security should be developed. Second, using this new metric, a systematic step-by-step algorithm should be designed to effectively split a given design to maximize trust/security while minimizing the overhead.

III. System Design

A. SplitScore: A Metric for Measuring Trust in Split Manufacturing

To measure trust in our proposed *hybrid* scheme, we develop a graph similarity measurement approach. Specifically, a given design is modeled as a colored directed graph G, according to the notations provided in Table I. Note that *SplitScore* can be employed at any level of design abstraction by customizing the definition of wires and nodes. (*e.g.,* for physical design, each global or local interconnect is a wire – edge in G, and each active or passive device is a node – vertice

Table I: Design parameters and corresponding graph notations.

Design parameter	Graph notation
Node v in netlist	Vertice $v \in V$
Wire e in netlist	Edge $e \in E$, $E \subseteq V \times V$
Incoming wires to node v_x	E_x^{in}
Outgoing wires from node v_x	E_x^{out}
Nodes connected to wire e_x	V_x^{con}
Type of wire e	$\zeta(e) : E \rightarrow \{1, \cdots, t_n^e\}$
Type of node v	$\zeta(v) : V \rightarrow \{1, \cdots, t_n^v\}$
Length of wire e	$\ell(e)$
Complete netlist	Colored directed graph $G = \langle V, E, \zeta \rangle$

in G). The equivalent graphs of the original (*i.e.,* prior to obfuscation) and obfuscated designs are denoted by, respectively, G^{org} and G^{obf}. Before introducing *SplitScore*, two previously proposed state-of-the-art metrics are reviewed.

- *k-security*: a design is *k-secure* if for every vertice u from G^{obf}, there exist k distinct vertices $v_1, \cdots, v_k$ in G^{org}, and mappings $\phi_1, \cdots, \phi_k$ from V^{org} to V^{obf} such that every ϕ_i is a *subgraph isomorphism* from G^{org} to G^{obf}, and for all $i \in [1, k]$, $\phi_i(u) = v_i$. The assumption is that G^{obf} is a *subgraph* of G^{org}, where $V^{obf} = V^{org}$ and $E^{obf} \subseteq E^{org}$ [8].
- *Graph edit distance (GED)*: calculates the minimum number of edit operations to transform G^{org} to G^{obf}, where edit operations include adding/deleting an isolated vertex, adding/deleting an edge, and relabeling a vertex, while the cost of each edit operation can be customized [9].

As shown in the example in Figure 3, the existing metrics do not support chiplet systems due to the following two issues. **First,** while the available metrics are compatible with vertical split-manufacturing, the threat model of *hybrid* split manufacturing assumes a fully-untrusted environment (*i.e.,* BEOL chip may be fabricated by an untrusted foundry too). As such, the metric must be robust for evaluating trust level under the constraint that the attacker may have access to any subgraph(s) of FEOL chiplets *AND* the BEOL chip.

Alternatively, a robust metric must support a case that some of the FEOL chiplets or the BEOL chip is fabricated with trusted foundries. Considering the example shown in Figure 3(b), assume that FEOL chiplet$_1$ is fabricated in a trusted foundry, and the rest of the FEOL chiplets and the BEOL chip are fabricated with another untrusted foundry. The *k-security* metric cannot support the evaluation of trust level, since $V^{obf} \neq V^{org}$.

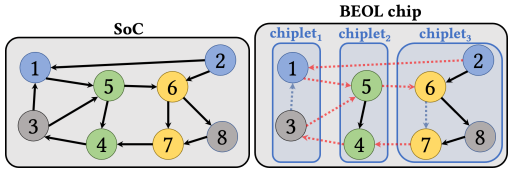
Second, vertical and horizontal obfuscation mechanisms feature unique physical design attributes, such as the length of obfuscated wires, which can be exploited by the attacker to decipher the original design. Both *k-security* and *graph edit distance* assume that all obfuscated wires are similar in terms of length. However, in our setting, using the example shown in Figure 3(b), the difference in length of the vertically obfuscated wire between nodes 1 and 2, and the horizontally obfuscated wire between nodes 6 and 7, could help the attacker to distinguish the correct connection between the FEOL chiplets and the BEOL chip.

To address these issues, we propose *SplitScore*. We derive this metric based on the definitions described in the following.

Definition 1 (mapping)**.** *For two sets of edges E and E', mapping is a surjective function* $M(E \mapsto E') = \{(e_i, e_g) | e_i \in E; e_j \in E'\}$.

Definition 2 (weighted mapping)**.** A mapping M, where each pair (e_i, e_j) is weighted by w_{ij}, and the total weight of the mapping is

$$W_M = \sum_{\forall (e_i, e_j) \in M} w_{ij}$$



(a) (b)

Figure 3: Example of the need for a new metric of obfuscation. (a) Original netlist of SoC before obfuscation, and (b) obfuscated netlist using *hybrid* split manufacturing, including three FEOL chiplets and a BEOL chip (dotted red and blue arrows represent obfuscated wires using, respectively, horizontal and vertical obfuscation mechanisms, which are located within the BEOL chip).

where,

$$w_{ij} = \begin{cases} \frac{min\{\ell(e_i), \ell(e_j)\}}{max\{\ell(e_i), \ell(e_j)\}} \times \frac{|V_i^{con} \cap V_j^{con}|}{|V_i^{con} \cup V_j^{con}|} & , \quad \zeta(e_i) = \zeta(e_j) \\ 0 & , \quad \zeta(e_i) \neq \zeta(e_j) \end{cases}$$

Definition 3 (optimal mapping)**.** *A mapping with the highest total weight among all existing weighted mappings ($M^\star$).*

Definition 4 (NodeSim)**.** For $v_i \in V$ & $v_j \in V'$, NodeSim(v_i,v_j) =

$$\begin{cases} \frac{W_{M^\star(E_i^{in} \mapsto E_j^{in})} + W_{M^\star(E_i^{out} \mapsto E_j^{out})}}{2} & , \zeta(v_i) = \zeta(v_j) \\ 0 & , \zeta(v_i) \neq \zeta(v_j) \end{cases}$$

Definition 5 (SplitScore)**.** For two graphs $G = \langle V, E, \zeta \rangle$ and $G' = \langle V', E', \zeta' \rangle$,

$$SplitScore(G, G') = \frac{1}{|V||V'|} \times \sum_{i=1}^{|V|} \sum_{j=1}^{|V'|} NodeSim(v_i, v_j)$$

To summarize, *SplitScore* computes graph similarity by averaging the node-to-node similarity (*i.e., NodeSim*) for the optimal mapping of their connected edges. Note that *SplitScore* is backward-compatible with methodologies that employ conventional packaging, such as split-manufacturing.

B. An Algorithm for Hybrid Splitting

The ultimate goal of our design is to maximize security while reducing the overheads associated with horizontally and/or vertically splitting a given design.

To find an algorithm for this, we first define two new concepts called *InterSim* and *IntraSim*. These two concepts are design guidelines that support effective hardware obfuscation using *hybrid* split manufacturing. Note that our similarity metric, *SplitScore*, is used in these guidelines, as shown in the following definitions.

Definition 6 (InterSim)**.** For original netlist (before obfuscation) of complete design $G^{org} = \langle V^{org}, E^{org}, \zeta^{org} \rangle$ and corresponding netlist obfuscated using *hybrid* splitting $G^{org} = \langle V^{obf}, E^{obf}, \zeta^{obf} \rangle$,

$$InterSim(G^{org}, G^{obf}) = 1 - SplitScore(G^{org}, G^{obf}).$$

Definition 7 (IntraSim)**.** For FEOL chiplets modeled as $G_i^{ch} = \langle V_i^{ch}, E_i^{ch}, \zeta_i^{ch} \rangle$, where $i \in [1, \cdots, \boldsymbol{N}]$

$$IntraSim(G_1^{ch}, \cdots, G_{\boldsymbol{N}}^{ch}) = \frac{1}{\boldsymbol{N}^2} \sum_{i=1}^{\boldsymbol{N}} \sum_{\substack{j=0 \\ i \neq j}}^{\boldsymbol{N}} SplitScore(G_i^{ch}, G_j^{ch}).$$

Similar to *SplitScore*, *InterSim* and *IntraSim* return a real-value score in the range from 0 to 1. *InterSim* indicates the lack of similarity between the entire original and obfuscated designs. A higher value of *InterSim*, therefore, is better in terms of security (ideally 1).

Additionally, *IntraSim* is a parameter that indicates the average similarity among FEOL chiplets. A higher value of *IntraSim* is, therefore, better in terms of security (ideal value of 1), meaning that distinguishing a targeted node will be more challenging for the attacker. In other words, similar FEOL chiplets (*i.e.,* higher *IntraSim*) reduce the chances of a sucessful attack.

Furthermore, *InterSim* demonstrates the worst-case trust scenario against the most resourceful attacker, where the entire obfuscated design is exposed to a single untrusted environment. Alternatively, *IntraSim* evaluates the trust in a practical setting, where the design house has access to several untrusted foundries.

In an ideal scenario, in terms of trust, the design house would send each FEOL chiplet for fabrication at a dedicated foundry. In practice, each untrusted foundry has access to multiple FEOL chiplets, and the design house should attempt to obfuscate the design such that the FEOL chiplets delivered to each untrusted foundry exhibit high *IntraSim*. It should be emphasized that *IntraSim* is defined per foundry, and in a multi-foundry scenario, the objective is to maximize the average *IntraSim* across the foundries.

Given the independent perspective of *InterSim* and *IntraSim* to the level of trust, defining a single metric of trust is helpful to unify the trust information that *InterSim* and *IntraSim* provides. To this end, the *Level-of-Trust* (LoS) is defined as follows:

Definition 8 (Level-of-Trust (LoS))**.**

$$LoS = \alpha \times InterSim + (1 - \alpha) \times IntraSim,$$

where α is a real-value in the range from 0 to 1, to give different weigths to *InterSim* and *IntraSim* for calculating *Level-of-Trust*. Selecting a proper α depends on the heterogeneity of the whole design. The more homogeneous the design, the less information *IntraSim* gives about the level of trust. For a fully homogeneous design (zero heterogeneity), the value of $\alpha = 1$ is appropriate since when all FEOL chiplets are similar, there is no preference for sending chiplets to each foundry.

Two different approaches can be taken towards obfuscation with *hybrid* split manufacturing, including **performance- and trust-oriented** approaches.

Our performance-oriented approach obfuscates the design at a fixed relative placement of nodes across the system, which has the *maximum* performance of the original SoC system. The proposed framework evaluates the overheads of obfuscation at different obfuscation depths, starting from the lowest depth and gradually increasing it until reaching the maximum tolerable overhead. Note that the depth (extent) of obfuscation in vertical and horizontal security mechanisms is a function of, respectively, $\boldsymbol{M^s}$ (the splitting metallization layer) and $\boldsymbol{N}$ (the number of BEOL chiplets). At the maximum tolerable obfuscation depth, the algorithm performs multiple permutations (different scenarios for splitting the design layout) and selects a permutation with the largest *LoS*.

Alternatively, in our trust-oriented approach, specifically in a multi-foundry scenario ($\boldsymbol{N_F} \geq 2$), the objective is to maximize the similarity among FEOL chiplets that each foundry receives, which is realized by manipulating the relative placement of nodes across the chiplets, as *IntraSim* guideline suggests. The proposed framework maximizes *LoS* at each obfuscation depth, starting from the lowest depth and gradually increasing the depth until targeted *LoS* is achieved.

In the next section, we extensively study both approaches and report the tradeoffs in various benchmark circuits.

Algorithm 1: Simulation flow

```
1  Input: benchmark ciruit, N, M^s, N_F, N_P, α
   Data: electrical model of the package, PDK
   Result: trust and overhead of hybrid obfuscation scheme
   // Step 1: Initialization
2  initialize AreaDict, DelayDict, and MetalDict ;
3  parse verilog code to V^org and E^org ;
   // Step 2: Evaluation
4  for obfuscation depth of (N, M^s) do
       // Step 2.1: Obfuscation
5      generate V^obf and E^obf;
       // Step 2.2: Trust Evaluation
6      for p = 1 to N_P do
7          update V^obf and E^obf;
8          calculate and store Level-of-Trust;
9      end
10     sort permutations; // maximum Level-of-Trust
11     select the most-secure permutation;
       // Step 2.3: Overhead Calculation
12     calculate and return overhead of most-secure permutation;
13     // area, wirelength, power, delay
14     compare results to split-manufacturing and Split-Fabric;
15 end
```

IV. Evaluation Setup and Results

A. Evaluation Setup

Two RTL benchmark suites in Verilog hardware description language have been used in the simulations, including ISCAS'85 (17 designs, largest design includes 3.51k gates) and ISCAS'89 (41 designs, largest design includes 11.45k gates).

Horizontal and vertical splitting can be implemented on any chiplet-based platform. In this paper, the Si-IF [11] is chosen as the chiplet-based platform. Si-IF is an advanced chiplet-based heterogeneous integration platform that promotes a paradigm shift in system integration and packaging methodologies [11], [12]. The complex and high-cost hierarchy of conventional packages is replaced by a passive wafer of silicon as a substrate that includes several BEOL interconnect layers reducing the performance overhead of vertical obfuscation. Moreover, integrating dielets without bulky packages enables dense integration of components (dielets) at a horizontal spacing that is comparable to SoCs enabling horizontal obfuscation as an effective security mechanism. The electrical parameters of the Si-IF (used to model the evaluated obfuscation schemes) and conventional packages (used for split-manufacturing), are derived from two recent works [3], [13].

A framework for design space exploration (DSE) is implemented in Python. The simulation flow of the DSE framework is described in Algorithm 1. It receives the following input specifications (see line ❶ of Algorithm 1): (i) benchmark circuit (Verilog netlist), (ii) $\boldsymbol{N}$, number of FEOL chiplets for the horizontal obfuscation mechanism, (iii) $\boldsymbol{M^s}$, split metallization layer for the vertical obfuscation mechanism, (iv) $\boldsymbol{N_F}$, number of foundries, and (v) $\boldsymbol{N_P}$, number of permutations of the DSE. Furthermore, the electrical model of the package (*e.g.,* lumped model of the bonding and interconnect) and technology node files are also provided to the framework.

In the first step, the required parameters (area and delay of gates, and metallization resources using TSMC 65 nm PDK) are initialized (❷), and the input netlist is transformed into graph representation (❸). The evaluation process comprises three steps. First, for each obfuscation depth of ($\boldsymbol{N}$, $\boldsymbol{M^s}$), the corresponding graphs are initialized (❺). Second, for multiple permutations of the netlist across the

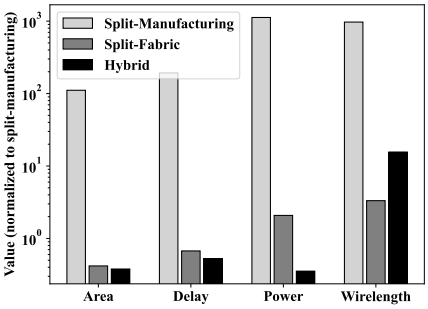


Figure 4: Overheads of *hybrid* scheme compared to split-manufacturing and Split-Fabric, for a constant LoS of 0.8.

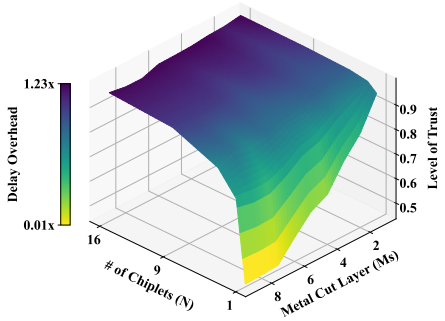


Figure 5: Level of trust (LoS) vs. delay as the function of vertical (metal layer) and horizontal (number of chiplets) splitting.

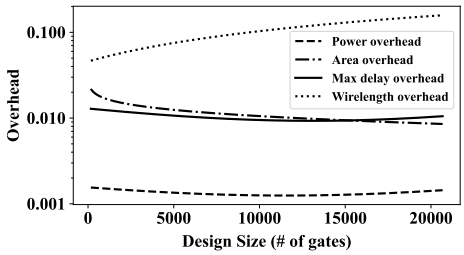


Figure 6: Overheads of *hybrid* scheme as a function of design size (values are normalized to split-manufacturing).

FEOL chiplets, the corresponding *Level-of-Trust* (LoS) parameters are stored, sorted, and the most-secure permutation (with the largest LoS is selected (❻:⓫). Finally, four different overhead metrics for the most-secure permutation are calculated and compared to the split-manufacturing approach (⓬:⓯).

Area overhead is estimated based on the additional area required to accommodate the bonding inputs/outputs (I/Os) used for obfuscation, assuming a chip utilization percentage (on-chip area dedicated to placement of cells) of 60% for all FEOL chiplets. Delay is calculated based on the static timing analysis implemented within the framework. Power overhead is computed assuming an operating frequency of 2 GHz for the calculation of dynamic power. Finally, wirelength overhead is estimated, assuming that each vertical and horizontal obfuscated connection imposes an overhead equivalent of, respectively, the minimum spacing of bonding I/Os and Manhattan distance between first and last BEOL chiplets (center-to-center).

B. Results

Three experiments have been conducted to evaluate the effect of different design and fabrication parameters (*i.e.,* input specifications in Algorithm 1) on the tradeoff between the level of trust and associated overhead for different combinations of horizontal and vertical security mechanisms.

Experiment 1: Obfuscation Depth. We first analyze the impact of changing vertical and horizontal splitting on important metrics including security, power, and area.

The effect of vertical obfuscation is a function of the metal split layer ($\boldsymbol{M}^s$), and the effect of horizontal splitting is measured by the number of FEOL chiplets ($\boldsymbol{N}$).

In a standard setup, the lowest metal layer is 8 (i.e., the minimal amount of overhead but lowest amount of obfuscation/security) while the maximum could be 1. Further, in vertical-only setups (i.e., all prior work), $\boldsymbol{N} = 1$, indicating that no horizontal splitting is used.

Using ISCAS'89 benchmark suite, Table II shows the impact of various configurations for vertical and horizontal splitting on LoS, area, power, wirelength, and delay. To provide more insights, we compare our results with conventional split manufacturing (vertical-only) and its chiplet version (Split-Fabric). Utilizing fine-grained low-parasitic bondings in *hybrid* scheme and Split-Fabric significantly reduces overheads compared to conventional Split-manufacturing.

To provide an overview and for a constant LoS, the overhead of the *hybrid* scheme is compared to two other approaches in Figure 4. The *hybrid* method exhibits two to three orders of magnitude lower overheads compared to split manufacturing. Compared to Split-Fabric, *hybrid* scheme reduces power, area, and delay overheads. Note that the data in Figure 4 includes overhead originating from the initial split into chiplets (*e.g.*, slightly larger wirelength is due to the effect of minimum spacing among chiplets, which is dominant in small designs.) The impact of benchmark circuit size on overheads is studied in experiment 2. To further investigate this, we plot LoS versus delay overhead as a function of vertical and horizontal splitting (*i.e.,* $\boldsymbol{M}^s$ and $\boldsymbol{N}$), in Figure 5.

The results indicate that after a certain depth ($\boldsymbol{N}$ = 4 and $\boldsymbol{M}^s$ = 5), the improvement in the level of trust saturates while the overhead continues to increase at a higher rate. Therefore, a vertical and horizontal splitting point that makes the best tradeoff between the level of trust and design overhead exists. Overall, experiment 1 confirms that the *hybrid* obfuscation is an effective scheme by leveraging horizontal and vertical security mechanisms to significantly increase the level of trust as compared to existing approaches.

Experiment 2: Design Complexity. To further investigate the impact of circuit complexity on the LoS vs. overhead tradeoffs, we conduct a new experiment.

In this experiment, for a constant configuration of $\boldsymbol{N} = 4$ and $\boldsymbol{M}^s = 5$ (best configuration in Experiment 1), the effect of design complexity on trust and performance overhead have been studied using designs with various sizes from two benchmark suites (with a minimum and maximum of, respectively, 13 and 11.45k gates).

Results show that the level of trust is constant regardless of the design size, while overhead metrics exhibit different pattern as shown in Figure 6. Furthermore, increasing the design complexity will increase the power and delay overheads as expected. However, increasing design complexity leads to a decrease in area and wirelength overheads. This pattern is due to a manufacturability limitation in chiplet-based integration for the minimum spacing among chiplets (*e.g.,* for the Si-IF platform, the minimum spacing is ∼100 µm). For extremely small designs, where the spacing among chiplets constitutes the majority of the system area, the area and wirelength overhead due to this manufacturability limitation is too large. Alternatively, for large designs, the area of FEOL chiplets will be dominant, and the effect of spacing among chiplets on area and wirelength will be negligible.

Experiment 3: Design Orientation. We compare our two strate-

Table II: Simulated results of trust level versus corresponding performance overhead for different depths of obfuscation.

Metric	Split-manufacturing			Split-Fabric			*Hybrid* Scheme					
	M^s=8	M^s = 4	M^s = 1	M^s = 8	M^s = 4	M^s = 1	(N=9,M^s=8)	(N=9,M^s=4)	(N=9,M^s=1)	(N=16,M^s=8)	(N=16,M^s=4)	(N=16,M^s=1)
Level-of-Trust	0.0056	0.4276	0.7700	0.0047	0.3501	0.9092	0.8896	0.9447	0.9968	0.9118	0.9611	1
Area (μm^2)	259,200 (1×)	1,5616,800 (60×)	33,242,400 (128×)	1,100 (0.004×)	56,800 (0.219×)	132,900 (0.512×)	301,084 (1.161×)	310,016 (1.196×)	3,187,21 (1.229×)	424,170 (1.636×)	434,132 (1.674×)	441,632 (1.703×)
Power (W)	0.71 (1×)	319.70 (446.38×)	992.82 (1,386.21×)	0.03 (0.047×)	0.55 (0.78×)	1.90 (2.6×)	0.32 (0.45×)	0.52 (0.73×)	0.75 (1.05×)	0.43 (0.60×)	0.62 (0.87×)	0.80 (1.11×)
Wirelength (μm)	14,849 (1×)	5,755,604 (387×)	1,7870,587 (1,203×)	2,595 (0.174×)	19,214 (1.2×)	62,774 (4.2×)	217,002 (14.6×)	220,707 (14.8×)	235,002 (15.825×)	262,360 (17.6×)	273,142 (18.3×)	275,217 (18.5×)
Delay (ns)	231 (1×)	16,014 (69×)	44,124 (190×)	3 (0.014×)	50 (0.217×)	158 (0.681×)	236 (1.018×)	239 (1.033×)	243 (1.047×)	279 (1.206×)	283 (1.220×)	285 (1.230×)

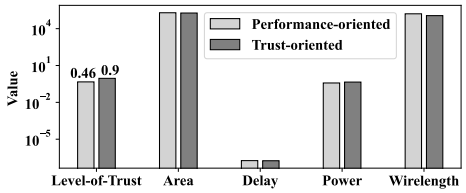


Figure 7: The level of trust versus overheads as a function of obfuscation depth.

gies for splitting, trust- and performance-oriented in this experiment.

For the best configuration from Experiment 1 (*i.e.,* N = 4 and M^s = 5), we assume having two foundries (*i.e.,* N_F = 2), and compare LoS versus overheads. Results are shown in Figure 7.

For the performance-oriented design, the original placement of the gates in the benchmark has been used, while for the trust-oriented, all gates with similar type are forced to be placed close to each other. Results in Figure 7 confirm that the trust-oriented design significantly increases the effectiveness of obfuscation, as the *Level-of-Trust* significantly increased (∼2×) with almost the same overhead. This experiment confirms that an active approach towards trust at early design stages significantly increases the effectiveness of *hybrid* split manufacturing methodology, rather than passively obfuscating a placed and routed design with a standard EDA tool.

C. Robustness and Scalability

A comparison between this work and state-of-the-art in terms of performance and computation scalability is provided in Table III.

Performance Scalability. The proposed *hybrid* methodology supports chiplet-based, ensuring continuous performance improvement benefits from the advancement of the packaging industry, as minimum spacing between chiplets and parasitics of the bondings scales down. Furthermore, in a large design, not only the obfuscation depth can be customized for targeted performance threshold, M^s can be customized for each chiplet (depth of vertical mechanism), trading performance off in critical chiplets only rather than in the entire design.

Computation Scalability. Calculating LoS using Inter and Intra similarity for each configuration is in the order of $\mathcal{O}(|V| \times |V|)$. For all configurations, this means the total worst case time complexity is $\mathcal{O}(N_p N M^s |V|^2)$.

Calculating metrics and overheads is a deterministic and fundamentally P-type problem. However, finding a layout which results in an optimal *LoS* and overhead trade-off is NP-hard. The overall complexity of analysis can be reduced with a combination of reasonable parameters and multi-threading. Given that the product of $N_P N M^s$ is in the order of several tens to hundreds, such scenarios can be executed concurrently on machines capable of supporting that number of threads, placing the complexity in the order of $\mathcal{O}(|V|^2)$.

Table III: Key contributions and comparison with previous work.

	[8]	[9]	**This work**
Methodology	Split-manufacturing	Split-manufacturing	*hybrid* scheme
Metric	*k-security*	*GED*	*SplitScore*
Abstraction layer	RTL	RTL	Multi-layer
Complexity	NP	NP-hard	P
Packaging (SoC , chiplets)	(✓ , ✗)	(✓ , ✗)	(✓ , ✓)

V. Conclusions

The *hybrid* split manufacturing methodology that employs horizontal and vertical obfuscation mechanisms is introduced. Horizontal obfuscation adds a second dimension to design-for-trust by security-aware functionality division across chiplets. *SplitScore*, a robust metric of trust is proposed, which, unlike available metrics, supports horizontal chiplet-based obfuscation. Furthermore, *InterSim* and *IntraSim*, two design-for-trust guidelines, are introduced for the effective use of the horizontal and vertical security mechanisms, particularly for fabless design houses with limited access to foundries.

Various experiments studied the impact of vertical and horizontal splitting on security vs. overheads. Results demonstrate that the *hybrid* scheme can significantly increase the level of trust, with up to three orders of magnitude lower performance overhead compared to split-manufacturing.

References

[1] J. Rajendran *et al.*, "Is Split Manufacturing Secure?" *DATE*, 2013.
[2] K. Xiao *et al.*, "Efficient and secure split manufacturing via obfuscated built-in self-authentication," *HOST*, 2015.
[3] Y. Safari *et al.*, "Split-Fabric: A Novel Wafer-Scale Hardware Obfuscation Methodology using Silicon Interconnect Fabric," *ECTC*, 2022.
[4] M. Li *et al.*, "A Practical Split Manufacturing Framework for Trojan Prevention via Simultaneous Wire Lifting and Cell Insertion," *TCAD*, 2019.
[5] Y. Wang *et al.*, "The Cat and Mouse in Split Manufacturing," *TVLSI*, 2018.
[6] K. Vaidyanathan *et al.*, "Detecting Reliability Attacks During Split Fabrication using Test-Only BEOL Stack," *DAC*, 2014.
[7] "IARPA Trusted Integrated Circuits (TIC) Program," https://www.iarpa.gov/research-programs/tic, 2011, online.
[8] F. Imeson *et al.*, "Securing Computer Hardware Using 3D Integrated Circuit (IC) Technology and Split Manufacturing for Obfuscation," *USENIX*, 2013.
[9] M. Fyrbiak *et al.*, "Graph Similarity and its Applications to Hardware Security," *IEEE Transactions on Computers*, 2020.
[10] S. Pal *et al.*, "Designing a 2048-Chiplet, 14336-Core Waferscale Processor," *DAC*, 2021.
[11] S. S. Iyer *et al.*, "Silicon Interconnect Fabric: A Versatile Heterogeneous Integration Platform for AI Systems," *IBM Journal of Research and Development*, 2019.
[12] Y. Safari *et al.*, "Wafer-Scale Integration," *Wiley EEEE*, 2022.
[13] ——, "Power Delivery for Silicon Interconnect Fabric," *ISCAS*, 2021.