MMPack: Multi-Mask Co-Design for Ultra-Large Wafer-Scale Package Integration

Shanyi Li, Zhen Zhuang*, Siyuan Liang, Bei Yu, Tsung-Yi Ho
The Chinese University of Hong Kong

***Abstract*—Interposer-based packaging has emerged as a pivotal technology for integrating advanced logic and memory chiplets in artificial intelligence (AI) and high-performance computing (HPC) systems. To accommodate growing system complexity, ultra-large wafer-scale integration employs expanded silicon interposers to support more chiplets. However, manufacturing such interposers exceeds the limits of single-mask lithography, requiring mask stitching, a technique that introduces unique physical design constraints and structural discontinuities. Additionally, thermo-mechanical stress, particularly near through-silicon vias (TSVs) and stitching regions, poses critical reliability challenges that conventional floorplanning methods fail to address. This paper presents MMPack, a hierarchical analytical framework for multi-mask chiplet-package co-design. Our approach integrates three key innovations: (1) a performance-driven partitioning algorithm that minimizes inter-chiplet and inter-mask communication overhead; (2) a stitching-aware hierarchical floorplanning strategy based on alternating optimization to address mask boundary constraints; and (3) a stress-aware post-processing step that employs an analytical model to reduce critical stress concentrations while preserving floorplanning quality. Experimental results demonstrate that MMPack significantly enhances both architectural performance and mechanical reliability while maintaining efficient layout and runtime scalability. These results highlight the practicality of our framework for enabling robust, high-performance designs in next-generation wafer-scale integration systems.**

I. Introduction

Interposer-based packaging has become essential for integrating advanced logic and memory chiplets in artificial intelligence (AI) and high-performance computing (HPC) systems. As illustrated in Fig. 1, this 2.5D wafer-scale integration employs a silicon interposer to host high-bandwidth memories (HBMs) and application-specific integrated circuits (ASICs), providing high-performance, low-latency interconnects between chiplets and the package substrate. A major trend in enhancing the performance is the continuous expansion of interposer area, which enables the integration of a greater number of heterogeneous and high-performance chiplets [1]. This progression is clearly reflected in TSMC's CoWoS development roadmap, where the interposer area nearly doubled between 2016 and 2022 [2]–[4].

However, manufacturing such ultra-large silicon interposers faces fundamental lithography constraints. A single reticle exposure is limited to approximately $830mm^2$ [5], [6], making it infeasible to manufacture large-scale interposers using a single mask. To address this limitation, the mask stitching technique has become essential for 2.5D wafer-scale packaging. As shown in Fig. 2, this approach partitions the interposer into multiple sub-regions, each manufactured using a separate mask. Adjacent regions incorporate overlapping stitching zones to ensure proper alignment and pattern continuity during fabrication. These stitching regions introduce unique physical design constraints [7]. During floorplanning, designers must treat stitching boundaries as keep-out zones while carefully allocating macros within each mask-specific region.

As ultra-large interposer designs continue to scale, thermal mechanical stress has emerged as an important reliability concern in advanced packaging. The inherent size scaling of interposers induces

*: Corresponding author

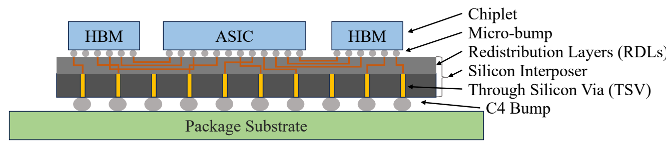

Fig. 1 The illustration of interposer-based 2.5D package.

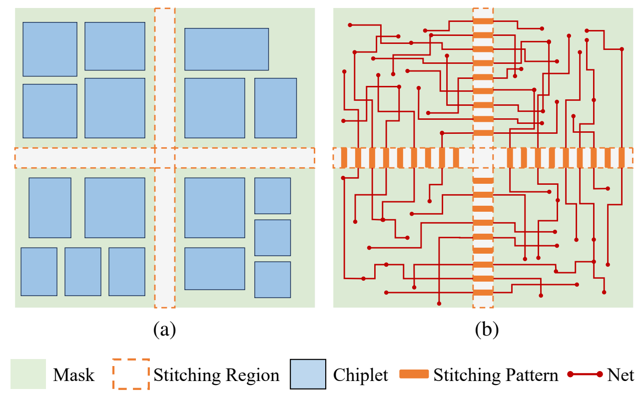

Fig. 2 Schematic of the mask stitching technique.

non-uniform stress distributions that significantly impact yield and long-term reliability [8], [9]. These stress concentrations primarily originate from coefficient of thermal expansion (CTE) mismatches between constituent materials—including silicon substrates, copper redistribution layers (RDLs), and heterogeneous chiplets—leading to potential cracking and interfacial delamination during thermal cycling. Through-silicon vias (TSVs) are especially vulnerable, with stress-induced failures degrading system reliability. Moreover, mask stitching regions introduce structural discontinuities, further concentrating stress. Thus, stress-aware design is essential for reliable wafer-scale integration.

A. Previous Work

Physical design methodologies originally developed for on-chip integration, such as those proposed in [11], [12], face significant limitations when extended to advanced packaging scenarios. These techniques are not directly applicable at the package level due to fundamental differences in design constraints, interconnect characteristics, and thermo-mechanical behavior. Among the various challenges in advanced packaging, chiplet floorplanning optimization has emerged as a key problem, aiming to maximize system performance while ensuring mechanical and thermal reliability [13]. This optimization task becomes increasingly complex as packaging scales toward wafer-level integration, where design spaces are significantly larger and more constrained.

Current research on package-level floorplanning can be broadly categorized into two main categories of approaches. The first category

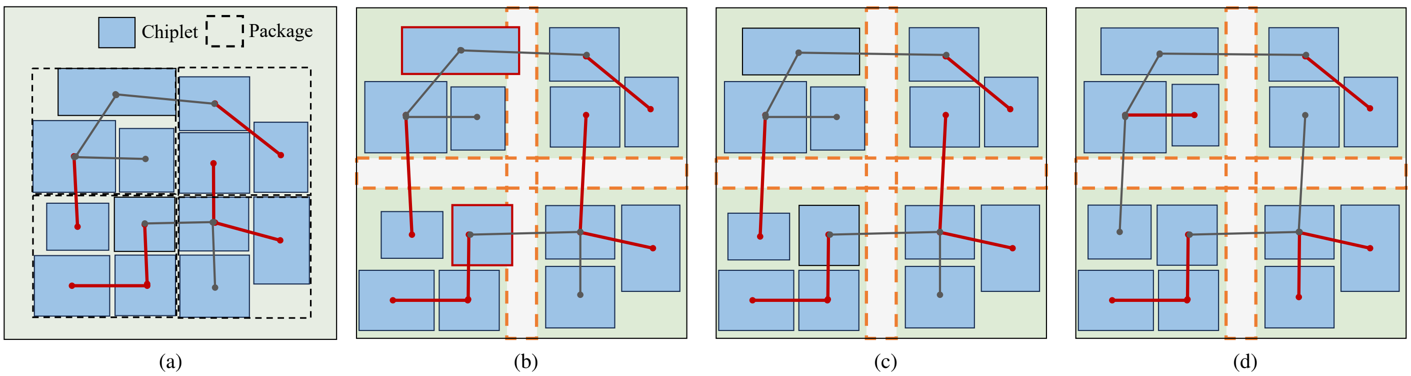

Fig. 3 Comparison of different multi-package designs. (a) A baseline MCMP design from [10] without consideration of multi-mask constraints. (b) A multi-mask co-design that violates the chiplet overlap constraint at stitching boundaries. (c) A stitching-aware multi-mask co-design that neglects interconnect performance. (d) The proposed MMPack design jointly considers both overlap constraints and interconnect performance.

involves simulated annealing (SA)-based methods [14]–[16], which often utilize data structures such as B^*-trees and Transitive Closure Graphs (TCGs) to represent physical layouts. These approaches offer flexibility in optimizing multiple design metrics beyond wirelength, including thermal dissipation and communication latency. However, they typically suffer from converging to suboptimal solutions, especially under the multi-domain constraints of large-scale heterogeneous designs.

The second category comprises analytical floorplanning algorithms [17], [18], which formulate layout optimization as mathematical programming problems. These approaches facilitate accurate modeling of complex physical constraints and performance objectives, which is critical for enabling scalable and manufacturable design in advanced packaging systems. Several notable efforts have demonstrated the effectiveness of integrating physical and architectural considerations into floorplanning frameworks. Additionally, [10] introduces a hierarchical co-optimization framework for multi-chiplet multi-package architectures, laying the foundation for scalable system integration, as illustrated in Fig. 3(a). While analytical methods exhibit improved scalability and optimization performance compared to heuristic approaches, there is still a notable limitation. Existing floorplanning frameworks fail to incorporate mask-specific manufacturability constraints, such as stitching-aware keep-out zones, or to model thermo-mechanical stress as part of the optimization. This gap becomes critical as interposer size scales beyond single-mask limits.

B. Our Contributions

To address the limitations of existing floorplanning methodologies in ultra-large wafer-scale integration, we propose a hierarchical analytical framework that explicitly incorporates mask stitching constraints throughout the optimization process. Unlike the multi-package floorplanning approach in [10], which primarily targets to minimize area, our framework focuses on optimizing system-wide communication latency, an essential metric for high-performance computing systems. Moreover, our framework also enhances the reliability of the final layout by integrating analytical stress modeling into the floorplanning process. Our key contributions are summarized as follows:

- To the best of our knowledge, this is the first work on hierarchical chiplet-package co-design, which explicitly considers mask stitching constraints for ultra-large wafer-level packaging. The proposed method demonstrates high efficiency in multi-mask scenarios.
- We propose a performance-driven partitioning algorithm that clusters macros into mask regions, and groups them into chiplets, effectively reducing both inter-mask and inter-chiplet communication overhead.
- We adopt an alternating optimization-based hierarchical floorplanning framework, which jointly optimizes both intra- and inter-chiplet layout, enabling efficient optimization under mask stitching constraints.
- A stress-aware post-processing stage is introduced to improve mechanical reliability by minimizing critical stress concentrations around through silicon vias (TSVs).
- Experimental results demonstrate that our method enhances performance and reliability metrics while significantly reducing runtime.

II. Preliminaries

A. Design Rules of Mask Stitching Regions

In ultra-large wafer-scale integration, mask stitching technology enables the fabrication of silicon interposers that exceed the dimensional constraints of a single lithography mask. However, this technique imposes specialized design rules during chiplet floorplanning to ensure both functional reliability and system performance. This section introduces the critical design constraints associated with mask stitching regions.

To ensure manufacturability and mitigate physical risks, each component must be fully contained within a single mask region, maintaining a safe clearance from stitching boundaries. As illustrated by the red-bordered components in Fig. 3(b), any overlap with stitching margins violates essential floorplanning constraints and increases the likelihood of functional failures. This is due to the intrinsic alignment limitations of multi-mask lithography, which makes stitching regions highly susceptible to pattern distortion, line width variation, and circuit discontinuities. Therefore, enforcing strict spatial constraints around stitching interfaces is critical for maintaining design integrity and process reliability.

Beyond component floorplanning, interconnect integrity is another key consideration in mask-aware floorplanning. High-performance interconnects should be confined within a single mask region whenever possible, as stitching boundaries physically separate chiplets, leading to increased interconnect lengths, signal delay, and degraded communication efficiency. Following the methodology in [17], our framework employs gem5-based simulation tools [19] to evaluate data transfer performance across diverse chiplet workloads. These simulations not only estimate inter-chiplet latency but also help identify performance-critical interconnects. Interconnects with high data transfer rates are classified as system-critical and are prioritized during both floorplanning and mask assignment stages.

As shown in Fig. 3, we categorize interconnects based on their performance sensitivity. Fig. 3(d) presents an optimized floorplanning result where critical interconnects are successfully confined within a single mask region. This strategy minimizes cross-mask communication overhead, reduces routing congestion near stitching boundaries, and shields high-activity signal paths from manufacturing-induced variability at mask interfaces.

B. Stress Analysis

The stress within packages can lead to warpage, cracking, and delamination, causing reliability issues and affecting performance. Due to the coefficient of thermal expansion (CTE) mismatch with the silicon interposer, through-silicon via (TSV) plays an important role in inducing thermomechanical stress. The governing equation of the thermomechanical stress is as follows [20]:

$$\begin{aligned} -\nabla \cdot \sigma(\boldsymbol{u}) &= f \text{ in } \Omega, \\ \sigma(\boldsymbol{u}) &= \lambda \mathrm{tr}(\epsilon(\boldsymbol{u})) + 2\mu\epsilon(\boldsymbol{u}) - \alpha(3\lambda + 2\mu) \cdot \Delta T, \\ \epsilon(\boldsymbol{u}) &= \frac{1}{2}(\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^\top) \end{aligned} \tag{1}$$

where Ω is the computational domain, σ is the stress tensor field, $\boldsymbol{u}$ is the three-dimensional displacement vector field, f is the body force vector, ϵ is the strain tensor field, λ and μ are the Lamé parameters, α is the CTE, and ΔT is the thermal load.

To accurately analyze the stress, finite element method (FEM) is widely applied [21]. However, due to its time- and memory-consuming nature, it is impractical to utilize FEM-based stress analysis in design automation flows. To efficiently conduct stress analysis while maintaining relatively high accuracy, analytical models are developed to simplify and accelerate the calculation [22], [23].

Specifically, in [23], a semi-analytical TSV stress model is proposed:

$$\sigma_{rr} = -\sigma_{\theta\theta} = \begin{cases} \frac{c_0 \Delta T}{R^2} & \text{if } r < R, \\ \frac{c_0 \Delta T}{r^2} & \text{if } r \geq R. \end{cases} \tag{2}$$

$$\sigma_{zz} = \sigma_{rz} = \sigma_{\theta z} = \sigma_{r\theta} = 0 \tag{3}$$

$$c_0 = \frac{-E(\alpha_{Cu} - \alpha_{silicon})}{2(1-\nu)} R^2 \tag{4}$$

where σ is the thermomechanical stress value in the cylindrical coordinate system, E is the Young's modulus. α is the CTE, here Cu represents copper, the common TSV material, and Si represents silicon. ν is the Poisson's ratio, ΔT is the thermal load, R is the radius of TSV, and r is the distance from the measured point to the center of TSV.

As there are multiple TSVs in a package design, the stress at each point comes from all TSVs simultaneously. According to the stress superposition principle, the stress value at a certain point is the sum of the stress applied from different TSVs separately. In other words, this means that the stress value at each point is the accumulated TSV-induced stress.

C. Architectural Performance Simulation Method

To evaluate the impact of mask stitching on overall system performance, cycles per instruction (CPI) are simulated using the existing methods from [24]. This approach focuses on modeling how constraints imposed by stitching regions influence critical path delays and system-level performance.

For each candidate layout, we estimate the effective interconnect latency as:

$$lat = lat_{base} + \alpha \times Crossings_s, \tag{5}$$

where lat_{base} denotes the baseline latency without stitching effects, $Crossings_s$ represents the number of stitching region crossings encountered by critical communication paths, and α is a technology-dependent coefficient quantifying the delay penalty per stitching crossing.

Subsequently, the CPI for each design configuration is computed as:

$$CPI = staCPI + \beta \times (lat - stalat), \tag{6}$$

where $staCPI$ and $stalat$ refer to the standard CPI and latency in an ideal scenario, and β is a coefficient generated from the statistic. This simulation methodology enables quantitative evaluation of performance degradation due to multi-mask co-design. The resulting CPI estimates guide our layout optimization process toward configurations that minimize the performance impact of stitching constraints while adhering to all relevant design rules.

D. Problem Formulation

The floorplanning problem for mask stitching regions in ultra-large wafer-scale integration can be formulated as follows:

Let $\mathcal{M} = \{m_1, m_2, \ldots, m_{|\mathcal{M}|}\}$ represent the set of macros and $\mathcal{N} = \{n_1, n_2, \ldots, n_{|\mathcal{N}|}\}$ represent the set of nets connecting these macros. The wafer surface is divided into four lithography mask regions, denoted by $\mathcal{R} = \{r_1, r_2, r_3, r_4\}$, and the corresponding set of stitching boundaries between adjacent mask regions is denoted by $\mathcal{S} = \{s_1, s_2, s_3, s_4\}$.

The objective is to generate a hierarchical floorplanning solution that includes four key tasks: assigning macros to chiplets, assigning chiplets to specific mask regions, placing macros within each chiplet, and placing chiplets within their designated mask regions. They must satisfy all mask stitching constraints, including the avoidance of component overlap across stitching boundaries and the confinement of critical interconnects within single mask domains whenever possible. Furthermore, the solution should optimize multiple objectives. First, it minimizes the system-level communication latency, which is sensitive to the number of stitching region crossings along critical paths. Second, it should enhance mechanical reliability by reducing stress concentrations near stitching interfaces. Ultimately, the goal is to improve architectural performance, as measured by metrics such as CPI, while ensuring manufacturability of the floorplanning result.

III. Technical Details

A. Overview of the Design Flow

This section provides an overview of the proposed framework, outlining the key stages involved in our multi-mask co-design methodology. As shown in Fig. 4, the overall flow consists of three main components: (1) performance-driven partitioning, (2) hierarchical floorplanning, and (3) stress-aware post-processing. Each of these components is described in detail in the following subsections.

In the partitioning phase, a two-level strategy is employed. First, multi-mask macro partitioning divides the entire set of macros into four distinct masks. Then, each mask's macros are further decomposed into chiplets according to netlist performance requirements, thereby enhancing overall efficiency. After partitioning, the design is refined through hierarchical floorplanning. The intra-chiplet floorplanning step optimizes the placement of macros within each chiplet to minimize area while preserving connectivity. Subsequently, the inter-mask chiplet floorplanning step employs an alternating optimization

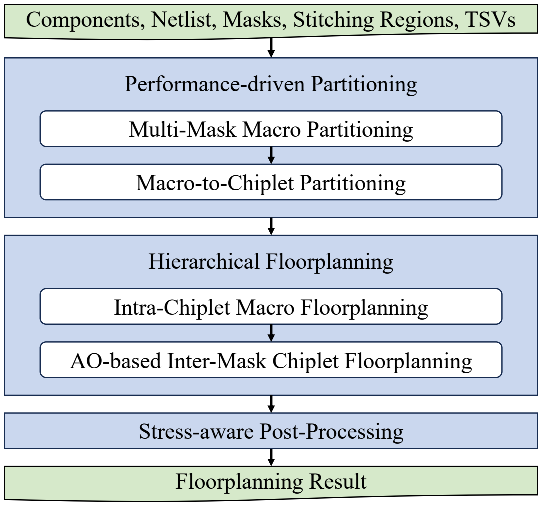

Fig. 4 The design flow of the proposed framework.

method to determine the chiplets' positions across the wafer-scale package, balancing local and global layout requirements. Finally, stress-aware post-processing uses an analytical stress model to adjust the layout, reducing thermal-mechanical stress induced by chiplet and TSV placements, thus enhancing reliability.

B. Performance-driven Partitioning

In multi-mask co-design, macros are physically distributed across multiple masks, and the interconnect distances between related macros have a significant impact on overall system performance. To address this, our partitioning algorithm strategically assigns macros with performance-critical connections to the same mask whenever feasible. This approach reduces inter-mask communication latency and helps to optimize critical path delays, thereby improving system-level performance. The partitioning problem is formulated as a mathematical programming (MP) model, which captures both connectivity and performance constraints. We solve this formulation using a commercial MP solver to obtain an optimized mask-level partitioning solution.

For each macro $m_i \in \mathcal{M}$, it should be assigned to a mask and thus can be constrained as:

$$\sum_{r_j \in \mathcal{R}} t_{i,j} = 1, \quad \forall m_i \in \mathcal{M}, \tag{7}$$

where $t_{i,j}$ is a 0-1 variable representing whether m_i is assigned to a mask region r_j.

To balance the sum of the macro area in each mask, the constraints are formulated as:

$$\sum_{m_i \in \mathcal{M}} (area_i \times t_{i,j}) \geq area_{r_j}^{lb}, \quad \forall r_j \in \mathcal{R}, \tag{8}$$

$$\sum_{m_i \in \mathcal{M}} (area_i \times t_{i,j}) \leq area_{r_j}^{ub}, \quad \forall r_j \in \mathcal{R}, \tag{9}$$

where $area_i$ represents the area of macro m_i, $area_{r_j}^{lb}$ and $area_{r_j}^{ub}$ represent the lower and upper bounds of the sum of the areas of the macros in mask j, respectively. It can be calculated as:

$$area_{r_j} = \frac{\sum_{m_i \in \mathcal{M}} area_i \times ar}{|\mathcal{M}|}, \tag{10}$$

where ar represents the user-defined area ratio coefficient. In our experiments, we set the lower and upper bound ratios as 0.9 and 1.1, respectively.

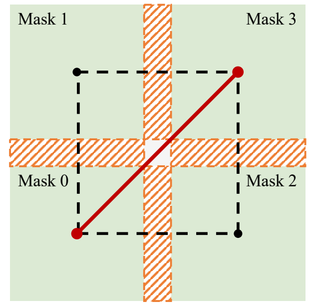

Fig. 5 Schematic of inter-mask net communication under mask stitching constraints.

In addition to maintaining performance balance across different masks, we quantify the communication overhead between masks based on a set of edges $\mathcal{E}$ extracted from the weighted netlist $\mathcal{N}$, which is generated by the chiplet simulator [17]. This overhead metric reflects the performance cost of inter-mask communication and is incorporated into our partitioning formulation to guide macro allocation. For each edge $e_{i,j} \in E$, the inter-mask overhead oh_e^{ij} can be formulated as:

$$oh_e^{ij} \geq t_{i,l} - t_{j,l}, \quad \forall r_j \in \mathcal{R}, \tag{11}$$

$$oh_e^{ij} \geq t_{j,l} - t_{i,l}, \quad \forall r_j \in \mathcal{R}. \tag{12}$$

If the two macros belong to the same mask, oh_e^{ij} is 0. Otherwise, the oh_e^{ij} is 1. Therefore, our objective can be formulated as:

$$inf \sum_{e_{i,j} \in E} (oh_e^{ij} \times W_e), \tag{13}$$

where W_e represents the performance of the net. The larger the W_e, the more critical it is. It helps our MP-based algorithm to better partitioning results and speed up the convergence of our algorithm.

Following the initial partitioning phase, we observe that macro communications between diagonally positioned masks pose additional performance challenges due to mask stitching constraints. As illustrated in Fig. 5, under the given mask arrangement, inter-mask connections are only permitted between physically adjacent masks. Consequently, when two macros are assigned to non-adjacent masks, such as the red line in the figure, they must traverse two stitching regions to complete the communication shown in dashed lines. This significantly increases both latency and routing complexity.

To address this issue, we introduce a mask permutation strategy as a post-partitioning optimization step. This technique preserves the assignment of macros to masks but systematically reorders the physical placement of the masks themselves. The goal is to minimize the number of critical connections between diagonally placed masks that exhibit high communication demand. The proposed approach effectively reduces inter-mask overhead and enhances system-level performance under stitching constraints by strategically relocating masks to encourage direct communication through adjacent masks.

After optimizing the physical placement of masks, we proceed to the macro-to-chiplet partitioning phase, which focuses on clustering functionally related macros into coherent chiplet groups within each mask. The objective of this step is to improve overall system performance by exploiting macro-level connectivity patterns to guide chiplet formation. Our partitioning strategy follows a methodology similar to the mask-level partitioning process, with one key distinction: instead

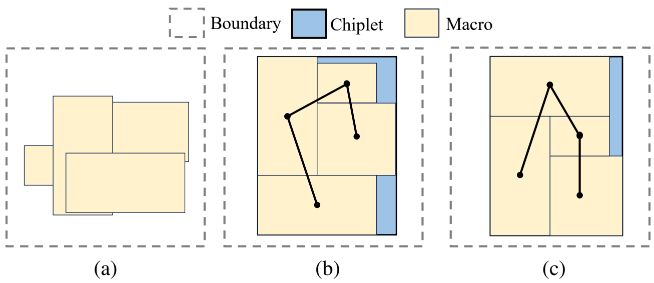

Fig. 6 Schematic of intra-chiplet macro floorplanning. (a) The input with several macros and a boundary. (b) A feasible floorplanning result satisfying non-overlap and boundary constraints. (c) An optimized floorplanning result with a smaller chiplet area.

of balancing physical area, we aim to evenly distribute the number of macros across chiplets. This design choice promotes a more uniform macro layout, which is advantageous for manufacturing regularity, thermal distribution, and scalability [17].

To control partition granularity, we introduce a user-defined parameter called the unit chiplet size, denoted by us, which specifies the target number of macros per chiplet. Given a mask $r_i \in \mathcal{R}$, the number of chiplets is computed as:

$$|Chiplet_{r_i}| = |M_{m_i}|/us, \tag{14}$$

where $|Chiplet_{r_i}|$ and $|M_{r_i}|$ represent the number of chiplets and the number of macros belonging to mask r_i. Similar to the bounds defined in Equations (11) and (12), we apply lower and upper bounds on the unit chiplet size to ensure balanced partitioning. By leveraging an optimization solver, our approach efficiently produces high-quality chiplet partitions that minimize inter-chiplet communication overhead while maintaining structural regularity and balance.

C. Hierarchical Floorplanning

Upon completing the partitioning stage, the design flow proceeds to the hierarchical floorplanning phase, which aims to optimize the placement of macros and chiplets across multiple masks. The first step in this hierarchy is the intra-chiplet macro floorplanning, which focuses on determining the spatial arrangement of macros within each chiplet region. This step optimizes both performance and area efficiency by arranging macros in a compact, non-overlapping manner while respecting chiplet boundaries.

As shown in Fig. 6, each chiplet consists of multiple interconnected macros. The floorplanning problem is formulated as a mathematical programming (MP) model, which is solved using an MP solver. Similar to previous MP-based floorplanning approaches, the solution must satisfy both non-overlap constraints and boundary constraints for all macros. In Figure 6(b), the geometric envelope of the placed macros defines the chiplet boundary, shown in yellow. To satisfy the boundary constraint, this chiplet boundary must not exceed the pre-specified outline. Our objective is to minimize the chiplet area, which can be formulated as:

$$\inf \quad w_j^{m_k} \times h_j^{m_k}, \tag{15}$$

where $w_j^{c_k}$ and $w_j^{c_k}$ represent the width and height of chiplet c_k, respectively. A small dead space ratio may result in an infeasible floorplanning result due to tight packing constraints. The fixed outline of each chiplet group is computed based on a specified dead space ratio. The fixed outline of each group is calculated based on the ratio

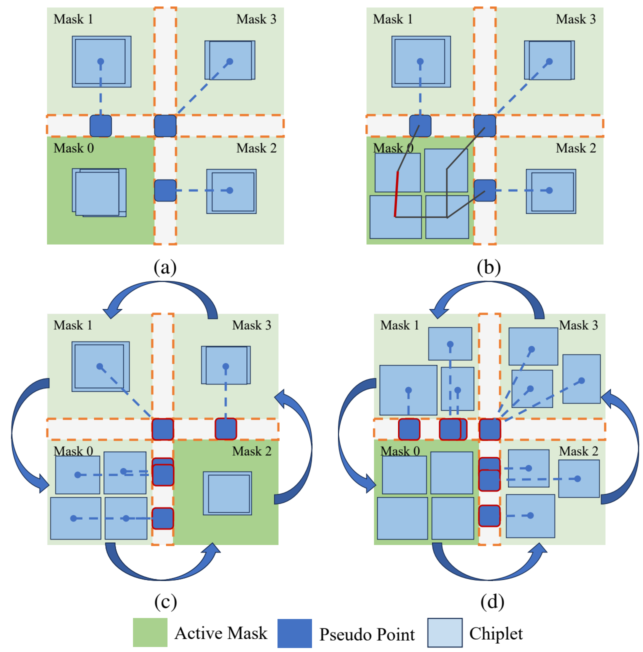

Fig. 7 Schematic of the chiplet-level alternating optimization process. (a) Chiplet floorplanning begins with Mask 0 as the active mask. (b) Chiplets from other masks are projected as pseudo points to guide inter-mask routing estimation. (c) The optimization proceeds sequentially to other masks and updates pseudo points. (d) Iterative updates continue across masks until convergence.

sr as:

$$w_b = h_b = \sqrt{area_M \times (1 + sr)}, \tag{16}$$

where $area_M$ is the total area of macros within the chiplet, and w_b, h_b are the width and height of the chiplet outline. The initial value of sr is set to 0.1 and is incrementally increased by 0.2 in each iteration until a feasible floorplan is obtained.

After completing intra-chiplet macro floorplanning, the design flow advances to the chiplet-level alternating optimization phase, which plays a critical role in our hierarchical floorplanning framework. As illustrated in Fig. 7, this stage adopts an iterative and mask-wise optimization strategy to coordinate chiplet floorplanning across multiple masks under mask stitching constraints. The directional arrows indicate the sequential optimization order among masks. During the optimization of a specific mask (e.g., Mask 0 in Fig. 7(a)), the chiplets from all non-active masks are projected into their corresponding stitching regions as pseudo points. These pseudo points act as spatial placeholders that approximate the positions of chiplets in other masks and are incorporated into the wirelength estimation process. This modeling approach enables each mask to optimize chiplet floorplanning locally, while still considering global inter-mask communication behavior. Within each active mask, we also employ an MP model similar in structure to the one used during intra-chiplet macro floorplanning. However, the objective function in this phase differs significantly: instead of minimizing area, we aim to reduce the total communication latency across the multi-chiplet system. The optimization objective is formulated as:

$$\min \sum_{n \in \mathcal{N}_{\text{intra}}} wl(n) \times \alpha_n + \sum_{n \in \mathcal{N}_{\text{inter}}} wl_{pseudo}(n) \times \alpha_n, \tag{17}$$

where $wl(n)$ and $wl_{\text{pseudo}}(n)$ denote the estimated wirelengths of net

Algorithm 1 : Alternating Chiplet Floorplanning

Input : A set of chiplets $\mathcal{C}$, a set of masks $\mathcal{M}$
Output : Final chiplet floorplan $\mathcal{C}_{\text{out}}$
Initialize displacement vector $\mathbf{D} \leftarrow$ `allOnes()` ;
Initialize iteration counter $i \leftarrow 0$;
repeat
  $m_i \leftarrow i \bmod |\mathcal{M}|$; `// Select current active mask`
  `MP_Floorplan(`m_i`)` ; `// Solve MP model for current mask`
  $\mathbf{D}[m_i] \leftarrow$ `calcDisplacement(`m_i`)` ; `// Check displacement`
  `UpdatePseudoPoint(`m_i`)` ;
  `Update(`$\mathbf{c}_{\text{out}}$`)` ;
  $i \leftarrow i + 1$;
until *allZeros*$(\mathbf{D})$ ***or*** $i \geq i_{th}$;
return $\mathcal{C}_{out}$

n for intra-mask and inter-mask connections, respectively. The term α_n represents a performance-driven weight derived from simulation or profiling, capturing the relative communication importance of each net.

As shown in Fig. 7(c), after optimizing the chiplet layout within mask 0, the corresponding pseudo points are updated, and the algorithm proceeds to mask two according to the predefined optimization sequence.

To provide a clearer understanding of the iterative optimization process, we present the pseudo-code of our algorithm in Algorithm 1. The algorithm is governed by a predefined maximum number of iterations, denoted as i_{th}, and terminates once this threshold is reached. Within each iteration, the function `MP_Floorplan` is invoked to perform MP-based chiplet floorplanning for a specific mask m_i. After completing the floorplanning, we evaluate the total displacement of chiplets within the current mask to assess the degree of convergence. The algorithm also includes an early stopping criterion: if all chiplets across all masks exhibit zero displacement between consecutive iterations, the process is deemed to have converged, and the optimization terminates early. This alternating optimization scheme enables coordinated floorplanning across masks while maintaining computational tractability, effectively balancing local layout quality and global communication efficiency.

D. Stress-aware Post-Processing

Following the hierarchical floorplanning process, we focused on stress issues that directly impact chip performance and reliability. While the preceding stages primarily focus on minimizing area and improving communication latency, they do not explicitly consider the stress induced by chiplet proximity and TSVs. If not properly mitigated, these stresses may lead to long-term reliability issues, especially in heterogeneous integration scenarios where materials with mismatched CTE are used.

To mitigate this issue, we introduce a stress-aware post-processing stage that refines the chiplet floorplan by reducing potential stress concentrations, while preserving the compactness and connectivity achieved in earlier optimization phases. This processing step enhances the mechanical robustness of the design and contributes to system reliability under thermal cycling and operational load. Since FEM-based mechanical models are typically non-linear and non-quadratic, directly integrating them into an MP-based framework is computationally infeasible. Instead, we adopt a lightweight analytical stress model, as shown in Equation 2, which approximates the radial stress experienced by each TSV. The model is based on the empirical observation that the stress induced by neighboring chiplets decreases inversely with the square of the distance:

$$\sigma_{rr}(t) \propto \sum_{c \in \mathcal{C}} \frac{1}{\text{dist}(t, c)^2} \tag{18}$$

The objective of the MP-based optimization in this post-processing stage is defined as:

$$\min \; (\sum_{t \in \mathcal{T}} \alpha \times \sigma_{rr}) + \beta \times area_M, \tag{19}$$

where σ_{rr} represents the radial stress experienced by TSV t, and $area_M$ represents the bounding rectangular area of all chiplets on the mask. α and β are user-defined coefficients that balance stress minimization and area compactness. This objective ensures that the refinement process reduces stress without unnecessarily spreading out chiplets, which could degrade communication efficiency.

To further enhance computational scalability, we deliberately avoid the use of binary (0-1) variables typically required to model pairwise non-overlapping constraints in floorplanning. Instead, we leverage the relative placement order derived from the existing coarse-grained floorplan and perform continuous optimization over chiplet coordinates. This approach preserves layout validity while significantly reducing the problem complexity. For instance, if chiplet C_{up} is already placed above chiplet C_{down} without spatial overlap, or chiplet C_{left} is positioned to the left of C_{right}, we enforce directional non-overlap via the following constraints:

$$y_{c_i} + h_{c_i} \leq y_{c_j}, \quad x_{c_i} + w_{c_i} \leq x_{c_j}, \tag{20}$$

where (x, y) denote the bottom-left coordinates and (w, h) represent the width and height of chiplet c_i, respectively. By fixing relative positions and eliminating combinatorial variables, the solution space is substantially reduced, resulting in faster convergence without sacrificing key layout properties such as connectivity, wirelength, or topological consistency. This relaxation serves as a critical enabler for scaling the framework to large design instances.

TABLE I Multi-Mask Co-Design Benchmark Statistics

Designs	$\|\mathcal{R}\|$	$\|\mathcal{M}\|$	$\|\mathcal{N}\|$	$\|\mathcal{T}\|$
$\mathcal{B}_1$	4	32	132	16
$\mathcal{B}_2$	4	54	313	16
$\mathcal{B}_3$	4	85	394	16
$\mathcal{B}_4$	4	104	589	32
$\mathcal{B}_5$	4	160	856	32
$\mathcal{B}_6$	4	200	1339	32

IV. Experimental Results

A. Experimental Setup

The proposed methods are implemented in C++ and executed on an Intel(R) Xeon(R) Gold 6226R 2.90GHz Linux machine with 256 GB RAM, ensuring sufficient computational resources for handling large-scale optimization tasks. To solve the MP-based models embedded in our framework, we utilize the Gurobi Optimizer [25], which is known for its efficiency and robustness in solving large-scale mixed-integer and continuous optimization problems. Since prior works such as [15] and [17] primarily focus on chiplet-level floorplanning without modeling internal macro structures or mask stitching constraints, they are not directly comparable to our hierarchical methodology. Instead, we adopt the MCMP framework proposed in [10] as our pri-

TABLE II Comparison of Floorplanning Results

Method	Benchmark	HPWL ($\times 10^3 \mu m$)	ComCost ($\times 10^5$)	PA ($\times 10^5 \mu m^2$)	MaxStress (MPA)	AveLat ($\times 10^9$ cycles)	Normalized CPI	Runtime (sec)
MCMP [10]	$\mathcal{B}_1$	49.12	6.50	4.21	57.63	69.49	0.18	14.77
	$\mathcal{B}_2$	99.15	12.58	3.55	61.17	158.71	0.22	1806.90
	$\mathcal{B}_3$	138.64	16.56	5.44	101.61	231.20	0.26	1835.63
	$\mathcal{B}_4$	171.37	21.00	3.05	172.40	325.32	0.31	3963.39
	$\mathcal{B}_5$	138.84	16.64	5.26	210.52	232.30	0.26	4309.52
	$\mathcal{B}_6$	356.76	42.81	2.93	337.85	988.47	0.64	11656.11
Ratio	N/A	1	1	1	1	1	1	1
Ours w./o. Stress Opt.	$\mathcal{B}_1$	46.27	5.64	4.15	59.76	58.92	0.17	2.64
	$\mathcal{B}_2$	86.32	11.41	3.33	64.91	139.46	0.21	9.39
	$\mathcal{B}_3$	121.96	13.20	5.39	109.78	169.20	0.23	24.14
	$\mathcal{B}_4$	161.83	19.03	3.00	180.4	281.71	0.28	62.41
	$\mathcal{B}_5$	142.18	15.96	5.22	222.86	219.48	0.25	318.74
	$\mathcal{B}_6$	325.82	35.77	2.82	351.68	737.66	0.51	817.71
Ratio	N/A	**0.93**	**0.88**	**0.99**	1.05	**0.81**	**0.88**	**0.05**
Ours with Stress Opt.	$\mathcal{B}_1$	49.12	6.06	4.41	56.55	63.98	0.18	2.83
	$\mathcal{B}_2$	90.88	12.04	3.65	59.72	149.62	0.22	9.50
	$\mathcal{B}_3$	130.31	15.46	5.54	98.80	210.03	0.25	25.01
	$\mathcal{B}_4$	169.84	20.04	3.34	156.32	303.54	0.30	63.81
	$\mathcal{B}_5$	149.67	16.62	5.41	202.80	232.68	0.26	319.86
	$\mathcal{B}_6$	330.90	39.74	3.19	298.45	874.70	0.58	819.55
Ratio	N/A	**0.96**	**0.94**	1.04	**0.94**	**0.91**	**0.95**	**0.05**

mary baseline, as it considers multi-package co-design and supports interconnect-aware optimization at a finer granularity.

Given the lack of publicly available real-world benchmarks for multi-mask wafer-scale systems, we conduct all evaluations using six synthetic benchmarks generated with the Floorplet framework [17]. These benchmarks are constructed to reflect realistic chiplet system configurations, capturing diverse macro connectivity patterns and communication demands. To emulate practical multi-mask packaging scenarios, we incorporate mask stitching constraints, including stitch line boundaries, non-overlap zones, and other relevant design rules during layout generation. TABLE I summarizes the key statistics of our benchmarks. We use $|\mathcal{R}|$, $|\mathcal{M}|$, $|\mathcal{N}|$, and $|\mathcal{T}|$ to represent the number of masks, macros, nets, and through-silicon vias (TSVs), respectively.

B. The comparison of floorplanning results

To demonstrate the effectiveness of our proposed multi-mask co-design framework, we perform a comprehensive evaluation using six standard benchmarks. The experimental results are summarized in TABLE II, which provides a detailed comparison against the multi-chiplet multi-package (MCMP) co-design methodology introduced in [10]. Additionally, we report results for our method in two configurations: with and without stress-aware post-processing, to examine the individual contributions of our post-processing stage. The comparison spans multiple key metrics that collectively capture both physical design quality and architecture-level system performance. In TABLE II, "HPWL", "ComCost", and "PA" represent the estimated wirelength based on the half-perimeter wirelength model, the inter-chiplet communication cost, and the total package area, respectively. "MaxStress" denotes the peak mechanical stress near TSVs, while "AveLat" and "CPI" measure average communication latency (in clock cycles) and normalized cycles per instruction, respectively. "Runtime" refers to the end-to-end execution time of the co-design flow.

Overall, our framework demonstrates consistent improvements across most metrics compared to the MCMP baseline. Specifically,

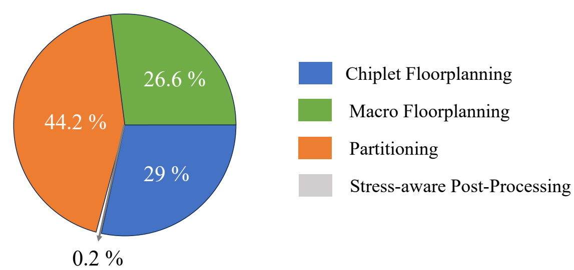

Fig. 8 Runtime distribution of our hierarchical framework on $\mathcal{B}_6$.

since the MP-based solver in [10] exhibits poor convergence behavior for large-scale problems, we follow their setup by imposing a runtime limit to ensure fair comparison using best-effort solutions. Even under this constraint, our framework delivers superior results with significantly faster runtime.

The results indicate that our framework consistently outperforms the MCMP baseline across most metrics, demonstrating its ability to effectively balance physical constraints with architectural performance objectives. Without stress-aware optimization, we observe average reductions of 7% in HPWL and 12% in communication cost. These improvements are primarily driven by our performance-aware partitioning and floorplanning strategy, which effectively minimizes long-range inter-chiplet connections and stitching crossings.

At the architectural level, as discussed in Section II-C, our simulation model captures the impact of stitching-induced delay by penalizing links crossing mask boundaries. By directly embedding stitching constraints into the floorplanning process, our method reduces critical interconnect length and avoids unnecessary mask transitions. As shown in TABLE II, our method achieves a notable reduction in average communication latency compared to the baseline MCMP approach. Without applying stress-aware post-processing, our framework lowers the average latency by up to 19%. After enabling stress-aware post-processing, the latency remains substantially lower than

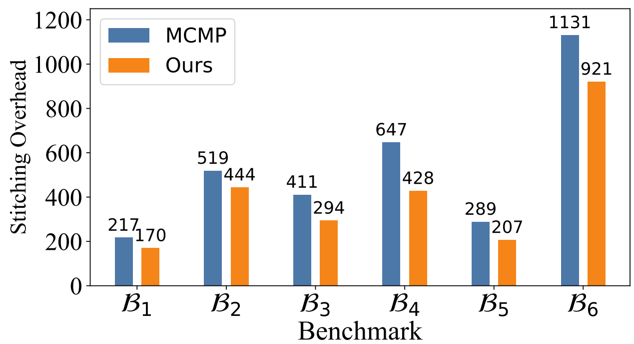

Fig. 9 Comparison of mask stitching overhead.

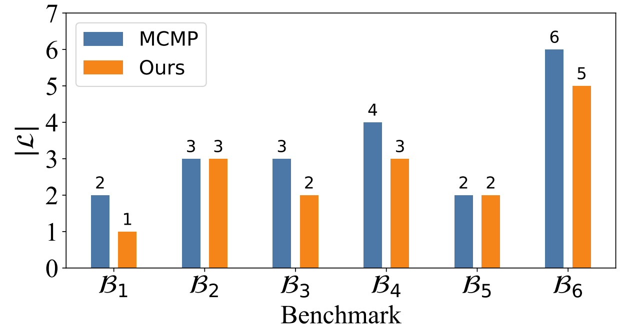

Fig. 10 Comparison of routing layer utilization.

MCMP, with a reduction of up to 9%. These latency improvements are further reflected in normalized CPI values, where we observe an average of 12% improvement without stress-aware post-processing and 5% improvement with stress-aware post-processing.

Although we have achieved good latency and CPI metrics before stress-aware post-processing, it suffers from elevated mechanical stress on the TSVs, even exceeding that of the baseline. It demonstrates the need for explicit stress-aware post-processing in multi-mask designs. Once stress-aware post-processing is enabled, the mechanical stress is significantly alleviated — achieving an average 6% reduction in maximum TSV stress — while maintaining superior architectural performance. Although this stress-aware post-processing introduces minor increases in floorplanning area and wirelength, the average latency and CPI remain consistently better than the baseline.

Furthermore, our proposed framework achieves substantial runtime improvements over the MCMP baseline, with an average reduction exceeding 95%. All benchmarks are completed within 15 minutes, demonstrating the practicality and scalability of our method for large-scale chiplet systems. This acceleration is especially significant for large-scale designs. For example, in benchmark $\mathcal{B}_6$, our method finishes in 820 seconds, while MCMP requires over 11,000 seconds to converge.

As illustrated in Fig. 8, the majority of runtime (44.2%) is spent in the partitioning stage, where macros are assigned to chiplets and masks while considering connectivity and stitching constraints. Chiplet-level floorplanning and intra-chiplet floorplanning contribute 29% and 26.6% of total runtime, respectively. The stress-aware post-processing accounts for only 0.2%, indicating that the core solver produces near-final quality solutions with minimal refinement. This balanced runtime distribution underscores the efficiency of our hierarchical optimization framework and highlights its suitability for future extensions to more complex design objectives.

In summary, our method achieves a well-balanced co-design solution for multi-chiplet multi-mask systems, simultaneously optimizing for latency, CPI, and reliability. Even with stress-aware constraints, it consistently outperforms the MCMP baseline in most key metrics, demonstrating its practicality and effectiveness in large-scale wafer-level integration.

C. Evaluation of Stitching Overhead and Routing Resources

We evaluated mask stitching overhead and routing layer utilization on multiple benchmarks, as illustrated in Fig. 9 and Fig. 10. Fig. 9 compares the total stitching overhead between the MCMP method and our proposed performance-driven partitioning strategy. In all cases, our method consistently achieves lower stitching overhead by clustering performance-critical macros within the same mask region, thus reducing the number of inter-mask connections. This optimization is especially impactful in the largest benchmark $\mathcal{B}_6$, where the overhead is reduced 19%. The result demonstrates improved scalability and routing efficiency when handling complex, large-scale chiplet systems.

In Fig. 10, we show the number of routing layers $|\mathcal{L}|$ used by each method, assuming each layer supports up to 50 nets. Our approach requires fewer layers in most benchmarks, indicating a more compact and efficient interconnect structure. This not only simplifies physical implementation and reduces manufacturing costs.

Overall, these results validate the effectiveness of our multi-mask co-design framework. By jointly optimizing partitioning, floorplanning, and stress-aware refinement, our method addresses both physical design complexity and architectural performance degradation. This holistic optimization not only improves traditional metrics (as shown in TABLE II), but also enhances manufacturability, reliability, and system efficiency, which are essential for next-generation wafer-scale chiplet integration.

V. Conclusion

This paper presents MMPack, a hierarchical co-design framework that addresses the critical challenges of ultra-large wafer-scale chiplet integration with multi-mask constraints. Our work makes significant advances by developing a performance-driven partitioning algorithm that optimizes macro assignments across masks and chiplets to minimize communication overhead, combined with a novel hierarchical floorplanning strategy that efficiently handles stitching constraints through intra-chiplet optimization and mask-level alternating optimization. The framework further incorporates a computationally efficient stress-aware post-processing technique that effectively reduces TSV-induced stress concentrations while preserving floorplan quality. Experimental results demonstrate that MMPack significantly improves wirelength, latency, and CPI, while reducing maximum TSV stress and runtime by over 95% compared to prior work. By simultaneously addressing performance, manufacturability, and reliability constraints, MMPack represents a significant improvement in enabling practical wafer-scale system design for advanced packaging technologies.

Acknowlegement

The research work described in this paper was conducted in the JC STEM Lab of Intelligent Design Automation funded by The Hong Kong Jockey Club Charities Trust. This work was jointly supported by the Research Grants Council of Hong Kong SAR (No. CUHK14211324 and No. CUHK14211824) and Science and Technology Innovation Key R&D Program of Chongqing (CSTB2023TIAD-STX0001).

References

[1] I. Lee, S. Nam, S. Kim, S. Shin, Y. Kim, S.-K. Seo, H. J. Yu, and D.-W. Kim, "Extremely Large 3.5 D Heterogeneous Integration for the Next-Generation Packaging Technology," in *Proc. ECTC*, 2023, pp. 893–898.

[2] "TSMC CoWoS," Available: https://3dfabric.tsmc.com/english/dedicatedFoundry/technology/cowos.htm.

[3] S. Hou, C. H. Lee, T.-D. Wang, H. C. Hou, and H.-P. Hu, "Supercarrier redistribution layers to realize ultra large 2.5 D wafer scale packaging by CoWoS," in *Proc. ECTC*, 2023, pp. 510–514.

[4] Y.-C. Hu, Y.-M. Liang, H.-P. Hu, C.-Y. Tan, C.-T. Shen, C.-H. Lee, and S. Hou, "CoWoS architecture evolution for next generation HPC on 2.5 D system in package," in *Proc. ECTC*, 2023, pp. 1022–1026.

[5] S. Nam, Y. Kim, A. Jang, I. Hwang, S. Park, S.-C. Lee, and D.-W. Kim, "The Extremely Large 2.5 D Molded Interposer on Substrate (MIoS) Package Integration-Warpage and Reliability," in *Proc. ECTC*, 2021, pp. 1998–2002.

[6] S. Hou, W. C. Chen, C. Hu, C. Chiu, K. Ting, T. Lin, W. Wei, W. Chiou, V. J. Lin, V. C. Chang *et al.*, "Wafer-level integration of an advanced logic-memory system through the second-generation CoWoS technology," *IEEE TED*, pp. 4071–4077, 2017.

[7] Z. Zhuang, Q. Chen, H. Yu, and T.-Y. Ho, "Fast Routing Algorithm for Mask Stitching Region of Ultra Large Wafer Scale Integration," in *Proc. ASPDAC*, 2025, pp. 279–284.

[8] S. Chen, H. Zhang, Z. Ling, J. Zhai, and B. Yu, "The Survey of Chiplet-based Integrated Architecture: An EDA perspective," in *Proc. ASPDAC*, 2025.

[9] H. Kim, J. Y. Hwang, S. E. Kim, Y.-C. Joo, and H. Jang, "Thermo-mechanical challenges of 2.5-D packaging: a review of warpage and interconnect reliability," *IEEE TCPMT*, pp. 1624–1641, 2023.

[10] Z. Zhuang, B. Yu, K.-Y. Chao, and T.-Y. Ho, "Multi-package co-design for chiplet integration," in *Proc. ICCAD*, 2022, pp. 1–9.

[11] Y.-K. Ho and Y.-W. Chang, "Multiple chip planning for chip-interposer codesign," in *Proc. DAC*, 2013, pp. 1–6.

[12] J.-M. Lin, T.-T. Chen, Y.-F. Chang, W.-Y. Chang, Y.-T. Shyu, Y.-J. Chang, and J.-M. Lu, "A fast thermal-aware fixed-outline floorplanning methodology based on analytical models," in *Proc. ICCAD*, 2018, pp. 1–8.

[13] Y.-W. Chang, "Physical design challenges in modern heterogeneous integration," in *Proc. ISPD*, 2024, pp. 125–134.

[14] Y. Hsu, M.-H. Chung, Y.-W. Chang, and C.-H. Lin, "Transitive Closure Graph-Based Warpage-Aware Floorplanning for Package Designs," in *Proc. ICCAD*, 2022, pp. 1–7.

[15] C.-C. Lee and Y.-W. Chang, "Floorplanning for embedded multi-die interconnect bridge packages," in *Proc. ICCAD*, 2023, pp. 1–8.

[16] H.-W. Chiou, J.-H. Jiang, Y.-T. Chang, Y.-M. Lee, and C.-W. Pan, "Chiplet placement for 2.5 D IC with sequence pair based tree and thermal consideration," in *Proc. ASPDAC*, 2023, pp. 7–12.

[17] S. Chen, S. Li, Z. Zhuang, S. Zheng, Z. Liang, T.-Y. Ho, B. Yu, and A. L. Sangiovanni-Vincentelli, "Floorplet: Performance-aware floorplan framework for chiplet integration," *IEEE TCAD*, pp. 1638–1649, 2023.

[18] Q. Wang, X. Li, T. Jia, Y. Lin, R. Wang, and R. Huang, "ATPlace2. 5D: Analytical thermal-aware chiplet placement framework for large-scale 2.5 D-IC," in *Proc. ICCAD*, 2024, pp. 1–9.

[19] S. Bharadwaj, J. Yin, B. Beckmann, and T. Krishna, "Kite: A family of heterogeneous interposer topologies enabled via accurate interconnect modeling," in *Proc. DAC*, 2020, pp. 1–6.

[20] R. B. Hetnarski, M. R. Eslami, and G. Gladwell, *Thermal stresses: advanced theory and applications*. Springer, 2009, vol. 41.

[21] J. H. Lau, M. Li, L. Yang, M. Li, I. Xu, T. Chen, S. Chen, Q. X. Yong, J. P. Madhukumar, W. Kai *et al.*, "Warpage measurements and characterizations of fan-out wafer-level packaging with large chips and multiple redistributed layers," *IEEE TCAD*, pp. 1729–1737, 2018.

[22] T. Zhu, Q. Wang, Y. Lin, R. Wang, and R. Huang, "MORE-Stress: Model Order Reduction based Efficient Numerical Algorithm for Thermal Stress Simulation of TSV Arrays in 2.5 D/3D IC," in *Proc. DATE*, 2025, pp. 1–8.

[23] T. Lu, Z. Yang, and A. Srivastava, "Post-placement optimization for thermal-induced mechanical stress reduction," in *2016 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*. IEEE, 2016, pp. 158–163.

[24] Z. Zhuang, K.-Y. Chao, B. Yu, T.-Y. Ho, and M. D. Wong, "Multi-Product Optimization for 3D Heterogeneous Integration with D2W Bonding," in *Proc. ICCAD*, 2023, pp. 1–9.

[25] "Gurobi Optimizer," Available: http://www.gurobi.com/.