IDDA-3D: Inter-Die Delay Aware Timing-Driven Placement on Face-to-Face bonded 3D ICs

Zixian Yang
zxy.zixianyang@gmail.com
National Yang Ming Chiao Tung University
Hsinchu, Taiwan, R.O.C.

Shanyi Li
syli@link.cuhk.edu.hk
The Chinese University of Hong Kong
Shatin, Hong Kong SAR

Leilei Jin
leileijin@cuhk.edu.hk
The Chinese University of Hong Kong
Shatin, Hong Kong SAR

Tsung-Yi Ho
tyho@cse.cuhk.edu.hk
The Chinese University of Hong Kong
Shatin, Hong Kong SAR

Chien-Nan Jimmy Liu
jimmyliu@nycu.edu.tw
National Yang Ming Chiao Tung University
Hsinchu, Taiwan, R.O.C.

Abstract

3D ICs extend integration freedom beyond post-Moore limits and can improve performance. Yet, existing true-3D placers remain primarily wirelength-driven, and partition-based 3D flows struggle to incorporate timing during design-space exploration. Prior 2D timing-driven approaches often rely on RSMT-based routing lookahead, which is unstable under z-moves and lacks a smooth objective for gradient-based optimization; simple net weighting further fails to capture path-level timing. We present **IDDA-3D**, the first timing-driven placement framework for face-to-face (F2F) bonded 3D ICs. IDDA-3D introduces a quadratic RC formulation that models intra-/inter-die driver-sink delay as a differentiable timing cost for analytical placement. The RC parameters are derived directly from the technology library, ensuring that the model reflects physical delay accurately and remains applicable across diverse technology nodes without manual tuning. To handle the discrete nature of die assignment, we employ a finite-difference approximation (FDA)-based gradient computation with preconditioning, which integrates seamlessly with the analytical placement engine. Experimental results show that **IDDA-3D** improves total negative slack (TNS) by up to 44% and worst negative slack (WNS) by 22%, while maintaining competitive wirelength and runtime compared with state-of-the-art true-3D placers.

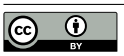
CCS Concepts

• **Hardware → Placement**; **3D integrated circuits**.

Keywords

Timing-driven Placement, 3D Integrated Circuits, Physical Design, Timing Optimization

ACM Reference Format:
Zixian Yang, Shanyi Li, Leilei Jin, Tsung-Yi Ho, and Chien-Nan Jimmy Liu. 2026. IDDA-3D: Inter-Die Delay Aware Timing-Driven Placement on Face-to-Face bonded 3D ICs . In *Proceedings of the 2026 International Symposium on Physical Design (ISPD '26), March 15–18, 2026, Bonn, Germany*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3764386.3779590



ISPD '26, Bonn, Germany

ACM ISBN 979-8-4007-2314-8/2026/03
https://doi.org/10.1145/3764386.3779590

1 Introduction

3D ICs have emerged as a promising solution to continue performance scaling beyond traditional 2D limitations. Face-to-face (F2F) bonded 3D ICs, which stack prefabricated dies connected through hybrid bonding terminals (HBTs), represent a particularly attractive approach due to their high integration density and manufacturing feasibility. However, F2F architectures introduce unique timing challenges that fundamentally differ from 2D designs [1][2]. Signal paths traverse multiple dies with heterogeneous delay characteristics, while HBTs exhibit distinct parasitic effects compared to conventional interconnects, making timing closure substantially more complex [3].

Timing-driven placement becomes critical in F2F 3D IC design due to the complex inter-die timing dependencies. Unlike 2D designs where timing optimization relies on post-placement routing adjustments, 3DIC architectures require timing-aware placement decisions. The distinct delay characteristics across different dies and the limited density of HBT connections make it essential to optimize timing during placement rather than deferring it to subsequent design stages. This makes timing-driven placement a fundamental requirement for successful F2F 3D IC implementation.

1.1 Previous Work

Current timing-driven placement for 2D designs employs two primary approaches: net-weighting methods that adjust timing-critical net weights based on STA results [4–6], and path-weighting methods that directly optimize critical path delays through mathematical formulations [7, 8]. While these 2D technologies achieve significant TNS and WNS improvements, they cannot handle inter-die connections in 3D architectures.

3D placement methods can be categorized into pseudo-3D and true-3D approaches. Pseudo-3D methods emulate 3D placement using conventional 2D placers by performing die partitioning on projected placements. Representative works include Compact2D [9] with partitioning-last design flows, Snap3D [10] using row-based transformations, and specialized approaches like Macro-3D [11] for F2F Memory-on-Logic systems and Pin-3D [12] for monolithic

designs. While benefiting from mature 2D tool integration, these methods lack awareness of inter-layer connectivity and cannot accurately model 3D timing characteristics such as vertical delays.

True-3D placers treat the entire 3D volume as a continuous optimization space using analytical frameworks. Early works like NTUPlace3-3D [13] and ePlace-3D [14] established foundational formulations with TSV considerations and electrostatic density constraints. Recent advances [15–17] extend to F2F architectures with adaptive density models and bistratal wirelength formulations for heterogeneous die connectivity. However, both pseudo-3D and true-3D approaches remain fundamentally wirelength-driven without explicit timing optimization, creating a significant capability gap for timing-driven F2F 3D designs.

1.2 Our Contributions

To address the limitations of existing 3D placement approaches in timing optimization, we propose IDDA-3D, an inter-die delay aware timing-driven placement framework specifically designed for F2F bonded 3D ICs. Our approach integrates timing considerations directly into the analytical placement optimization process, enabling effective timing closure for multi-die architectures. The main contributions of this work are summarized as follows:

- We introduce a quadratic RC-based delay model that directly captures inter-die signal propagation characteristics using standard library parameters, enabling accurate timing estimation without complex SPICE simulations or manual parameter tuning.
- We develop a path-aware timing optimization approach that integrates criticality-weighted timing objectives into the analytical placement framework, allowing gradient-based optimization to prioritize timing-critical connections spanning multiple dies.
- We propose a finite-difference approximation technique with adaptive gradient preconditioning that handles discrete die assignments within continuous 3D optimization, ensuring numerical stability and robust convergence across all spatial dimensions.
- Experimental results demonstrate that IDDA-3D achieves up to 44% TNS reduction and 22% WNS improvement over state-of-the-art 3D placers [16] while maintaining competitive wirelength and runtime efficiency.

2 Preliminary

2.1 Analytical Placement

Given a netlist $G = (V, E)$, where V is the set of instances and E is the set of nets, the placement objective is to minimize the total half-perimeter wirelength (HPWL) under density constraints [14]. The 3D HPWL for a net $e \in E$ is defined as a peak-to-peak function in three dimensions:

$$W_e(\mathbf{x}, \mathbf{y}, \mathbf{z}) = p_e(\mathbf{x}) + p_e(\mathbf{y}) + \alpha \cdot p_e(\mathbf{z}), \tag{1}$$

where $p_e(u) = \max_{i \in e} u_i - \min_{i \in e} u_i$, with $\alpha \geq 0$ as a scaling factor that controls the relative weight of vertical extent.

Since the max/min operators are non-differentiable, a smooth function usually approximates $p_e(u)$. A common choice is the *weighted-average* (WA) model:

$$p_{e,\mathrm{WA}}(\mathbf{u}) = \frac{\sum_{i \in e} u_i \exp(u_i/\gamma)}{\sum_{i \in e} \exp(u_i/\gamma)} - \frac{\sum_{i \in e} u_i \exp(-u_i/\gamma)}{\sum_{i \in e} \exp(-u_i/\gamma)}, \tag{2}$$

where γ is a smoothing parameter. As $\gamma \to 0$, $p_{e,\mathrm{WA}}(u) \to p_e(u)$.

Formally, analytical global placement can be formulated as the following optimization:

$$\min_{\mathbf{x},\mathbf{y},\mathbf{z}} \; W(\mathbf{x}, \mathbf{y}, \mathbf{z}) \; + \; \lambda \, U(\mathbf{x}, \mathbf{y}, \mathbf{z}), \tag{3}$$

where $W(\cdot)$ denotes the total (smoothed) wirelength, $U(\cdot)$ is the density penalty, and λ is a Lagrange multiplier that balances wirelength and density.

2.2 Static Timing Analysis

Static Timing Analysis (STA) is a method used to verify the timing performance of a digital circuit. The timing graph is modeled as a directed acyclic graph (DAG), where each *startpoint* (e.g., register or input port) propagates delays through combinational arcs to an *endpoint* (e.g., register or output port).

Each pin v has a required arrival time $R(v)$ and an arrival time $A(v)$. The slack is defined as

$$\mathrm{slack}(v) = R(v) - A(v). \tag{4}$$

A negative slack at an endpoint signifies a timing violation. To quantify the severity of these violations, we employ Worst Negative Slack (WNS) and Total Negative Slack (TNS):

$$\begin{aligned} \mathrm{WNS} &= \min_{v \in endpoints} \mathrm{slack}(v), \\ \mathrm{TNS} &= \sum_{v \in endpoints} \min(\mathrm{slack}(v), 0), \end{aligned} \tag{5}$$

Zero WNS and TNS indicate that all timing constraints are met.

2.3 Problem Formulation

Let $G = (V, E)$ denote the circuit netlist, where $V = \{v_1, \ldots, v_n\}$ is the set of instances (movable cells and I/O ports) and E is the set of nets. All movable cells must be placed in a 3D cuboid region

$$\Omega = [0, R_x] \times [0, R_y] \times [0, R_z] \tag{6}$$

The partition is represented by discretized z-coordinates $\hat{\mathbf{z}} \in \{0, 1\}^{|V|}$, with $\hat{z}_i = 0$ if instance i is placed on the bottom die while $\hat{z}_i = 1$ if on the top die.

A net e is identified as a cross-die net (net with pins on both dies) if its net-cut indicator $p_e(\hat{\mathbf{z}}) = 1$. For such nets, we insert hybrid bonding terminals (HBTs) at planar coordinates (x_{t_e}, y_{t_e}) to provide vertical connectivity between dies. Each cross-die net e is partitioned into two subsets based on die assignment:

$$e^+ = \{v_i \in e \mid \hat{z}_i = 1\}, \quad e^- = \{v_i \in e \mid \hat{z}_i = 0\}, \tag{7}$$

where $\hat{z}_i$ indicates the discrete die assignment of cell i. The HBT serves as a shared terminal for both die subsets:

$$\hat{e}^+ = e^+ \cup \{t_e\}, \quad \hat{e}^- = e^- \cup \{t_e\}. \tag{8}$$

Based on this decomposition, the die-to-die wirelength (D2D WL) of a cross-die net e is defined as

$$W_{e,\mathrm{D2D}}(\mathbf{x}, \mathbf{y}, \mathbf{z}, x_{t_e}, y_{t_e}) = \mathrm{HPWL}(\hat{e}^+) + \mathrm{HPWL}(\hat{e}^-), \tag{9}$$

where $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ denote the coordinates of all instances in net e.

In IDDA-3D, the primary design goal is to jointly minimize the D2D wirelength and the absolute value of TNS:

$$\min_{\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{x}_t,\mathbf{y}_t} \left(|\mathrm{TNS}|, \sum_{e\in E} W_{e,\mathrm{D2D}}(\mathbf{x},\mathbf{y},\mathbf{z},x_{t_e},y_{t_e}) \right). \tag{10}$$

However, directly optimizing Equation (10) is intractable because TNS is non-differentiable and computationally expensive to update. Therefore, instead of optimizing TNS directly, we introduce a differentiable timing cost, denoted as C_{tim}, to serve as a surrogate objective that effectively correlates with TNS reduction (as formulated later in Equation (26)). The placement result must also satisfy the following constraints:

(1) **Die assignment:** Every cell must be assigned to either the top or bottom die.
(2) **Legality:** All movable cells must be non-overlapping and aligned to rows/sites. Each cut net must be assigned exactly one legal HBT, and HBT spacing rules must be satisfied.

3 Placement Flow

In this section, we describe the IDDA-3D placement flow, as illustrated in Figure 1. The design flow presents a comprehensive overview of our proposed 3D placement approach, which begins with the input circuit netlist and PDK, followed by driver-sink pair initialization. The core component is the global placement stage (Section 3.1), while the timing optimization procedure is detailed separately in Section 4.

3.1 Global placement

The global placement stage optimizes in a 3D space, allowing cells to move along the z-axis before final die assignment. This added flexibility over 2D demands robust objectives whose behavior remains stable under true 3D motion, ensuring convergence and compatibility with the timing objective introduced next. To capture these requirements in a unified framework, we formalize the global placement problem as the minimization of a composite objective:

$$\begin{aligned}\min_{\mathbf{x},\mathbf{y},\mathbf{z}} \quad & \sum_{e\in E} W_{e,\mathrm{Bi}}(\mathbf{x},\mathbf{y},\mathbf{z}) + \lambda\, U(\mathbf{x},\mathbf{y},\mathbf{z}) \\ & +\alpha \sum_{e\in E} p_e(\mathbf{z}) + \beta\, C_{\mathrm{tim}}(\mathbf{x},\mathbf{y},\mathbf{z}),\end{aligned} \tag{11}$$

where $W_{e,\mathrm{Bi}}$ denotes the bistratal wirelength of net e, $U(\cdot)$ is the density penalty, $p_e(z)$ represents the vertical extent used in the cutsize term, and C_{tim} is the timing cost. The coefficients λ, α, β control the relative weights among these objectives. We detail each term below.

Electrostatics-based 3D density. State-of-the-art 3D analytical placement frameworks adopt the eDensity-3D, in which each cell is modeled as an electric charge equal to its volume, and density overflow is resolved through electrostatic repulsion. This analogy yields a smooth and differentiable density function that can be integrated into the gradient-based analytical solver. In our setting, the library cells provided in the PDK are inherently planar and lack physical depth. To embed them into the 3D density field, we assign both cells and the two dies a nominal thickness of $R_z/2$, where R_z is the vertical extent of the placement region Ω, as illustrated in Figure 2.

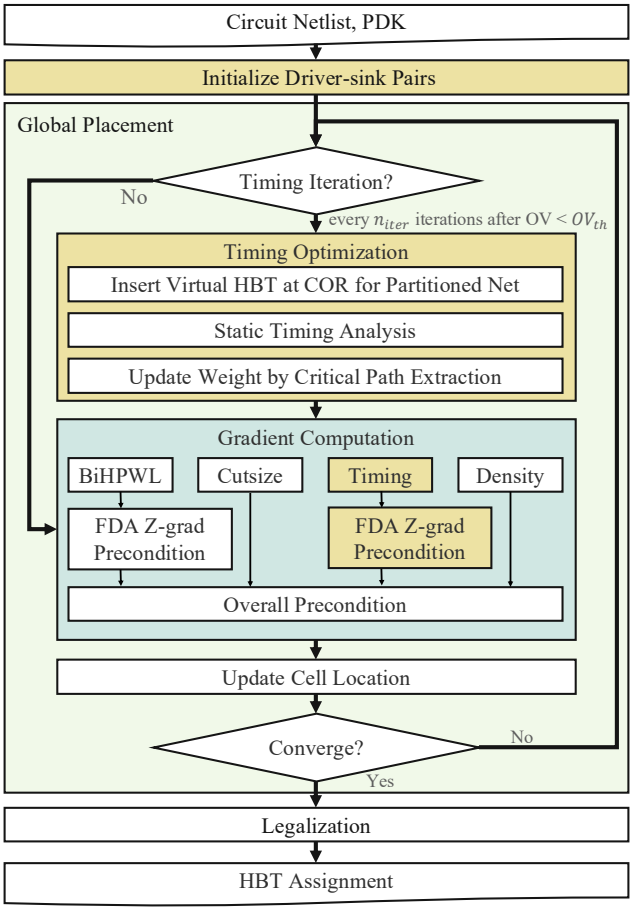


Figure 1: Overall placement flow of IDDA-3D. Blocks highlighted with yellow denote timing-driven steps.

Consequently, the placement cuboid Ω can be separated into two sub-cuboids:

$$\begin{aligned}\Omega^- &= [0,R_x]\times[0,R_y]\times[0,R_z/2], \\ \Omega^+ &= [0,R_x]\times[0,R_y]\times[R_z/2,R_z].\end{aligned} \tag{12}$$

We then define the z-coordinates of the center of two sub-cuboids as

$$Z_{\mathrm{bot}} = \tfrac{R_z}{4}, \qquad Z_{\mathrm{top}} = \tfrac{3R_z}{4}. \tag{13}$$

and hence allow each cell to move freely along the z-axis within the interval

$$z_i \in \left[Z_{\mathrm{bot}},\ Z_{\mathrm{top}}\right]. \tag{14}$$

Then, the 3D density cost is modeled as the total electrostatic potential energy of the system:

$$U(\mathbf{x},\mathbf{y},\mathbf{z}) \ = \ \sum_{i\in V} q_i\, \Phi_i(x_i,y_i,z_i), \tag{15}$$

where q_i is the electric quantity of volume of instance i and $\Phi_i(\cdot)$ is the electrostatic potential of positive charge i located in (x_i, y_i, z_i). Cells placed in overfull regions correspond to higher potential values and are pushed outward by the gradient of Φ, thus alleviating density overflow and ensuring an even distribution throughout the 3D placement region.

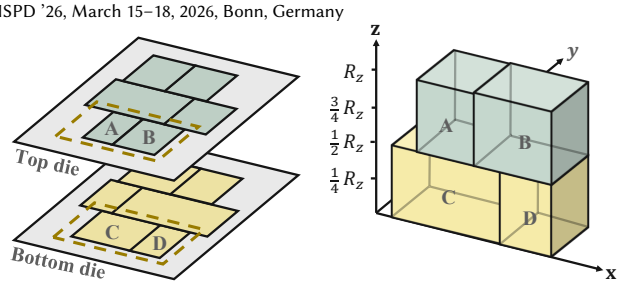


Figure 2: Illustration of true-3D cell depth configuration. Cells modeled as cuboids with depth $R_z/2$; their center z-coordinates are positioned at $3R_z/4$ and $R_z/4$, respectively.

Bistratal wirelength model (BiHPWL). During the global placement stage, no physical HBTs are inserted, and the circuit netlist remains unchanged; yet, we still need to estimate the D2D WL, as shown in Equation (9). To this end, we define the optimal region of an HBT t_e in a cross-die net. Along each axis, the region is given by

$$\begin{aligned} x'_{\min} &= \min\left\{\max(x^+_{\min}, x^-_{\min}),\ \min(x^+_{\max}, x^-_{\max})\right\}, \\ x'_{\max} &= \max\left\{\max(x^+_{\min}, x^-_{\min}),\ \min(x^+_{\max}, x^-_{\max})\right\}, \end{aligned} \tag{16}$$

and similarly for $y'_{\min}$ and $y'_{\max}$.

Geometrically, x'min and x'max are the two medians among the four bounding-box edges $x^+_{\min}, x^+_{\max}, x^-_{\min}, x^-_{\max}$ (and similarly for y). The rectangle $[x'_{\min}, x'_{\max}] \times [y'_{\min}, y'_{\max}]$ is the optimal region of the HBT: inserting the HBT anywhere inside this region does not increase the D2D WL. Two representative cases are illustrated in Figure 3.

We assume that each HBT will eventually be placed within its corresponding optimal region during the HBT assignment stage (in section 3.2). Therefore, in the global placement stage, a bistratal wirelength model [15, 16] is required to explicitly evaluate the D2D WL of cross-die nets without determining the exact HBT location beforehand. Formally, the BiHPWL in the x-axis is defined as

$$W_{e_x}(\mathbf{x}, \hat{\mathbf{z}}) \ = \ \max\left\{p_e(\mathbf{x}),\ p_{e^+}(\mathbf{x}) + p_{e^-}(\mathbf{x})\right\}, \tag{17}$$

and the overall BiHPWL of net e is

$$W_{e,\mathrm{Bi}}(\mathbf{x}, \mathbf{y}, \mathbf{z}) \ = \ W_{e_x}(\mathbf{x}, \hat{\mathbf{z}}) \ + \ W_{e_y}(\mathbf{y}, \hat{\mathbf{z}}). \tag{18}$$

Here, $p_e(x)$ denotes the peak-to-peak span of all pins of net e along the x-axis, while $p_{e^+}(x)$ and $p_{e^-}(x)$ denote the corresponding peak-to-peak spans of its top- and bottom-die pin sets, respectively(similarly for y). This formulation also depends on the z-dimension, which requires a binary discretized $\hat{z}_i$ to be used in determining the top-/bottom-die pin sets p_{e^+} and p_{e^-}. To this end, $\hat{z}_i$ is defined as

$$\hat{z}_i = \begin{cases} 1, & \text{if } z_i > \frac{1}{2}R_z, \\ 0, & \text{otherwise}, \end{cases} \tag{19}$$

where z_i is the continuous vertical coordinate of instance i. This discretization maps continuous z_i into a binary $\hat{z}_i$, which serves as a tentative die assignment during global placement.

Intuitively, the BiHPWL definition reflects two cases of optimization for a cross-die net. When the bounding boxes of the top and bottom subnets overlap, as shown in the left case of Figure 3, the BiHPWL can be minimized by independently reducing the HPWL on each die. Otherwise, when the two bounding boxes are disjoint, as shown in the right case of Figure 3, the optimization focuses on reducing the overall bounding box of net e.

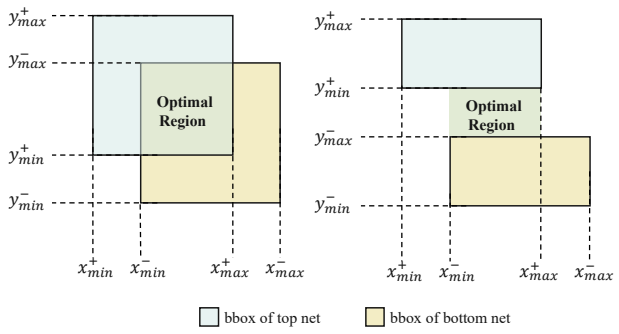


Figure 3: Illustration of the optimal region for inserting an HBT, as defined in Equation (16). In the left case, the bounding boxes of the top net e^+ and the bottom net e^- overlap. In the right case, the two bounding boxes are disjoint.

For gradient computation, a common practice is to replace the peak-to-peak function p_e with a smooth wirelength approximation such as the weighted-average (WA) model. The planar gradients can then be derived directly under the two cases of cross-die nets. However, the z-axis gradient remains discontinuous because partitioning is inherently discrete. To address this, the bistratal wirelength model employs a finite-difference approximation (FDA) [15] to obtain vertical gradients reliably.

FDA-based z-axis gradient. Let $\mathbf{z}^{(\mathrm{top})} = \mathbf{z} + (Z_{\mathrm{top}} - z_i)\,\mathbf{e}_i$ and $\mathbf{z}^{(\mathrm{bot})} = \mathbf{z} + (Z_{\mathrm{bot}} - z_i)\,\mathbf{e}_i$, where $\mathbf{e}_i$ is the unit vector along z for cell i. The FDA-based z-gradient for cell i is

$$\begin{aligned} \left(\widetilde{\nabla}_z W_{e,\mathrm{Bi}}\right)_i &= \Delta_{\frac{R_z}{4}} W_{e,\mathrm{Bi}}(\mathbf{x}, \mathbf{y}, \mathbf{z}) \\ &= \frac{W_{e,\mathrm{Bi}}(\mathbf{x}, \mathbf{y}, \mathbf{z}^{(\mathrm{top})}) - W_{e,\mathrm{Bi}}(\mathbf{x}, \mathbf{y}, \mathbf{z}^{(\mathrm{bot})})}{R_z/4}, \end{aligned} \tag{20}$$

where $R_z/4$ is the perturbation step along the vertical axis. However, the scale of the planar gradients is not consistent with the FDA-based vertical gradient $\widetilde{\nabla} z W_{e,\mathrm{Bi}}$. Therefore, we precondition the vertical gradient as

$$\nabla_z W_{e,\mathrm{Bi}} \ = \ \frac{\left\|\nabla_x W_{e,\mathrm{Bi}}\right\| + \left\|\nabla_y W_{e,\mathrm{Bi}}\right\|}{2\left\|\widetilde{\nabla}_z W_{e,\mathrm{Bi}}\right\|} \ \cdot \ \widetilde{\nabla}_z W_{e,\mathrm{Bi}}, \tag{21}$$

where $\|\cdot\|$ denotes the Euclidean norm. As illustrated in the flow graph in Figure 1, the gradient computation of BiHPWL includes a special step, *FDA z-gradient preconditioning*, which is required to ensure stable optimization. Here, we overload the notation and use $\nabla_z W_{e,\mathrm{Bi}}$ to denote the preconditioned FDA-based gradient, even though it is not the exact partial derivative. This allows the gradient descent updates in the (x, y, z) dimensions to share the same form.

Cutsize: In the true-3D context, it is crucial to discourage excessive HBT usage, since wirelength may increase or even become illegal if too many HBTs exist. We define the cutsize cost based on the z-dimension peak-to-peak function $p_e(\mathbf{z})$ of each net e, which measures the vertical extent of the net.

Overall Precondition: In timing-driven placement, the gradients must remain numerically stable. Without proper scaling, the optimization can suffer from ill-conditioned updates and oscillatory behavior. Following the philosophy of ePlace3D [14], the preconditioner is defined as

$$P(v_i) \;=\; \max\left(1,\ \lambda\, q_i\right) \;=\; \max\left(1,\ \lambda\, \mathrm{vol}(v_i)\right), \tag{22}$$

where q_i (or equivalently $\mathrm{vol}(v_i)$) denotes the electric quantity of cell v_i and λ is a Lagrange multiplier. This final step in the gradient computation shown in Figure 1: once the preconditioned gradient is obtained, it is used to update the cell locations.

3.2 Legalization and HBT Assignment

As global placement converges, each cell is already close to either the top or bottom die. The discretized vertical coordinate $\hat{z}$ then determines the final die assignment: $\hat{z}_i = 0$ corresponds to the bottom die, while $\hat{z}_i = 1$ corresponds to the top die. Accordingly, the z-coordinate of each cell is fixed to Z_{bot} or Z_{top}.

Subsequently, we perform Abacus legalization [18] independently within each die to resolve overlaps while minimizing the displacement of the global placement solution. For every cross-die net whose net-cut indicator is $p_e(\hat{\mathbf{z}}) = 1$, a corresponding HBT is inserted, and the circuit is then changed. HBTs are treated as standard-cell-like objects on an HBT layer. The initial location of the HBT is placed at the center of its optimal region (COR):

$$(x_{t_e}, y_{t_e}) \;=\; \left(\tfrac{x'_{\min}+x'_{\max}}{2},\ \tfrac{y'_{\min}+y'_{\max}}{2}\right), \tag{23}$$

where the bounds $x'_{\min}, x'_{\max}, y'_{\min}, y'_{\max}$ represent the optimal region defined in equation 16. If the HBT layer exhibits more than 10% overflow after initial assignment to COR, we invoke a 2D global placement on the HBT layer. A legalization on the HBT layer is then performed. This refinement ensures that spacing constraints of HBTs are satisfied while minimizing wirelength degradation relative to the global placement result. Our work mainly targets the global placement stage. After legalization and HBT assignment, the layout can then proceed to a standard detailed placement step as in a conventional placement flow.

4 Timing Optimization

Timing remains the most challenging to optimize in the 3D setting. Metrics such as TNS and WNS are highly sensitive to small perturbations in placement and exhibit significant nonlinearity. The difficulty is further amplified in F2F 3D ICs, where delay originates not only from intra-die routing but also from cross-die interconnects through HBTs. In the following, we present the timing optimization of IDDA-3D, beginning with the proposed intra-/inter-die delay model.

4.1 Intra-/Inter-Die Delay Modeling

Traditional timing-driven placement approaches often rely on net-based weighting [4], where nets on critical paths are assigned larger weights to guide placement. However, such methods only indirectly approximate timing objectives and fail to capture the delay along driver-sink connections. A more accurate direction is to incorporate the routing-lookahead delay estimation. Rectilinear Steiner minimum trees (RSMTs) are a state-of-the-art approach for constructing

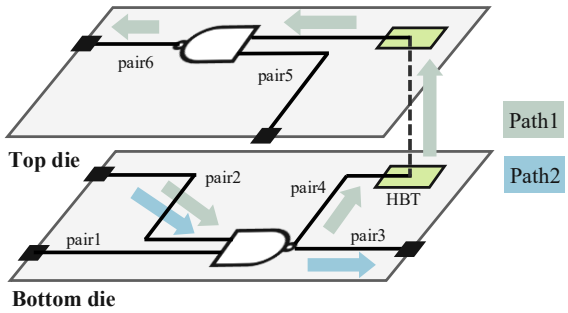


Figure 4: Schematic of circuit across two dies, showing intra-die connections, inter-die connections through HBTs, and two unique critical paths (Path1 and Path2) reported by STA.

RC trees and estimating parasitics at the placement stage, since their topology is consistent with the eventual routing.

However, the combinatorial complexity and sensitivity to geometric changes of the steiner topologies make them difficult to explicitly integrate into analytical placement engines. Moreover, in the true-3D placement context, movements along the z-axis can fundamentally alter the tree topology and invalidate the RC tree structure, making RSMT-based approaches impractical.

Recent works have proposed simplified tree-based models to strike a balance between accuracy and efficiency. For example, XPlace3.0 [19] introduces a degenerated RC tree model based on spanning trees to estimate parasitic delay effects. Motivated by this trade-off, we adopt a further simplified delay model derived from an RC delay formulation. For a distributed RC wire of length L with unit resistance r and capacitance c, its end-point delay under Elmore approximation [20] is:

$$d(L) = \int_0^L r\cdot x\cdot (c\,dx) = rc\int_0^L x\,dx = \tfrac{1}{2}rcL^2. \tag{24}$$

This shows that the delay is quadratic in the wirelength, which naturally motivates the use of the quadratic distance cost. Prior work, such as Efficient-TDP [8] has empirically demonstrated that the quadratic Euclidean distance modeling correlates well with timing metrics (e.g., WNS/TNS), further validating its effectiveness. Accordingly, for every driver-sink pair (i, j), we define the intra-/inter-die delay model as

$$D(i,j) = \begin{cases} \frac{1}{2}rc\,\|(x_i,y_i)-(x_j,y_j)\|^2, & \hat{z}_i = \hat{z}_j \\ \frac{1}{2}rc\,\|(x_i,y_i)-(x_{t_e},y_{t_e})\|^2 & \hat{z}_i \neq \hat{z}_j. \\ \quad +\,\|(x_j,y_j)-(x_{t_e},y_{t_e})\|^2, & \end{cases} \tag{25}$$

where $\|\cdot\|$ denotes the Euclidean norm, and (x_{t_e}, y_{t_e}) are the coordinates of the HBT assigned to the cross-die net e when $\hat{z}_i \neq \hat{z}_j$. Figure 4 illustrates an example circuit, showing both an intra-die driver-sink connection (e.g., pair2) and an inter-die connection through an HBT (e.g., pair4).

Our quadratic formulation implements an RC-based delay model for 3D integration that seamlessly incorporates HBTs for inter-die connections. This approach provides a tractable approximation capturing essential RC delays for both intra- and inter-die nets while avoiding the tree-based methods' instability. For STA purposes,

we place virtual HBTs at their respective COR on cross-die nets, deferring overlap and spacing constraint considerations to later design stages.

4.2 Path-based Weighting

The proposed timing cost is formulated as a weighted delay model:

$$C_{\text{tim}}(\mathbf{x},\mathbf{y},\mathbf{z}) \;=\; \sum_{(i,j)\in C} w(i,j)\, D(i,j), \tag{26}$$

where $w(i,j)$ is the weight of driver-sink pair(i,j), and $D(i,j)$ is the intra-/inter-die delay defined in Section 4.1.

To emphasize timing-critical connections in the delay model, we derive the weights $w(i,j)$ from STA during placement. We use OpenSTA in OpenROAD [21] to compute slacks and to extract a single representative path for each failing endpoint, following the efficient critical-path extraction strategy of Efficient-TDP [8]. In Efficient-TDP, this is realized by the `report_timing(n,1)` command in OpenTimer [22], which enumerates n endpoints with negative slack and returns one path for each. In OpenSTA, an equivalent set of paths is obtained via `report_checks -unique -slack_max 0 -endpoint n`, which selects n endpoints with slack < 0 and reports exactly one path per endpoint.

Let C denote the set of all driver-sink pairs, and each pair is initialized with a base weight. After a STA call, we obtain a set of unique critical paths $\mathcal{P}_-$. We then traverse all paths $p \in \mathcal{P}_-$ and accumulate additional weights for the driver-sink pairs on these paths. Pairs that are shared by multiple violating paths naturally receive larger weights. Pairs not on any violating path retain the base weight. The following equations summarize the weighting procedure:

$$\begin{aligned} &w(i,j) \leftarrow w_{\text{base}}, && \forall (i,j)\in C,\\ &w(i,j) \leftarrow w(i,j) \;+\; w_{\text{cum}}\cdot\frac{\text{slack}(p)}{\text{WNS}}, && \forall p\in\mathcal{P}_-,\;\; \forall (i,j)\in p. \end{aligned} \tag{27}$$

where w_{base} is the base weight and w_{cum} controls the accumulation magnitude. Since WNS < 0, the ratio $\frac{\text{slack}(p)}{\text{WNS}} \in (0,1]$ for violating endpoints, ensuring that the increment is always positive and larger for more critical connections. As illustrated in Figure 4, the example circuit contains six driver-sink pairs and two unique failing paths. We assume the two paths have slacks of Path1 $= -10$, Path2 $= -7$, with the WNS determined by Path1 at -10.

Pair1 and pair5 don't lie on any failing path and therefore retain only the base weight w_{base}. Pair3, pair4, and pair6 each lie on a single failing path, resulting in weights of $w_{\text{base}} + w_{\text{cum}}\cdot(-7/-10)$, $w_{\text{base}} + w_{\text{cum}}\cdot(-10/-10)$, and $w_{\text{base}} + w_{\text{cum}}\cdot(-10/-10)$, respectively.

Pair2 lies on two failing paths, which is the key feature of our path-based weighting: a shared pair accumulates slack weights from multiple paths. Specifically, the weight of pair2 becomes $w_{\text{base}} + w_{\text{cum}}\cdot(-10/-10) + w_{\text{cum}}\cdot(-7/-10)$. This accumulation mechanism strengthens the pulling force on pairs shared by multiple critical paths, ensuring that these timing-critical connections are effectively shortened.

In the global placement stage, the base weight remains active throughout the entire process, ensuring that the timing cost is always present in the objective. As shown in Figure 1, timing iterations are triggered only after the density overflow drops below OV_{th}. At this point, the cumulative term is enabled, and we periodically invoke STA and update the weights according to the reported critical paths every n_{iter} iterations to control runtime overhead.

4.3 Gradient Computation

The proposed timing cost C_{tim} must be optimized jointly with wirelength, density, and cutsize in an analytical framework, which requires reliable gradient computation. For planar gradients (i.e., derivative w.r.t x or y), the quadratic delay model is smooth and admits closed-form derivatives, which can be directly integrated into the optimization engine.

In contrast, the timing cost is non-differentiable in the z-dimension, because delay propagation changes fundamentally when a connection switches between intra-die and inter-die. This discrete transition makes direct gradient computation with respect to the z-axis infeasible. We leverage a finite-difference approximation (FDA) similar to that in BiHPWL to overcome this issue. To construct a stable FDA gradient, we perturb the continuous z_i by a step of $R_z/4$, which is large enough to flip the discretized $\hat{z}_i$ between the top and bottom die while keeping the cell within the placement region boundaries. The following illustrates two cases:

- **Case 1 (bottom → top):** If $z_i \in \left[\frac{R_z}{4}, \frac{R_z}{2}\right]$, then $\hat{z}_i = 0$. Perturb cell i upward by $R_z/4$, so that its $\hat{z}_i$ flips from 0 to 1. The FDA z-gradient is

$$\begin{aligned} \Delta_{\frac{R_z}{4}} C_{\text{tim}} &= \frac{C_{\text{tim}}(\mathbf{x},\mathbf{y},\mathbf{z}+\frac{R_z}{4}\mathbf{e}_i) - C_{\text{tim}}(\mathbf{x},\mathbf{y},\mathbf{z})}{R_z/4} \\ &= \frac{C_{\text{tim}}(\mathbf{x},\mathbf{y},\mathbf{z}^{(\text{top})}) - C_{\text{tim}}(\mathbf{x},\mathbf{y},\mathbf{z}^{(\text{bot})})}{R_z/4}. \end{aligned} \tag{28}$$

- **Case 2 (top → bottom):** If $z_i \in \left(\frac{R_z}{2}, \frac{3R_z}{4}\right]$, then $\hat{z}_i = 1$. Perturb cell i downward by $R_z/4$, so that its $\hat{z}_i$ flips from 1 to 0. The FDA z-gradient is

$$\begin{aligned} \Delta_{-\frac{R_z}{4}} C_{\text{tim}} &= \frac{C_{\text{tim}}(\mathbf{x},\mathbf{y},\mathbf{z}-\frac{R_z}{4}\mathbf{e}_i) - C_{\text{tim}}(\mathbf{x},\mathbf{y},\mathbf{z})}{-R_z/4} \\ &= \frac{C_{\text{tim}}(\mathbf{x},\mathbf{y},\mathbf{z}^{(\text{bot})}) - C_{\text{tim}}(\mathbf{x},\mathbf{y},\mathbf{z}^{(\text{top})})}{-R_z/4}. \end{aligned} \tag{29}$$

Both cases lead to the same formulation of the FDA z-axis gradient. We therefore define

$$\begin{aligned} \left(\widetilde{\nabla}_z C_{\text{tim}}\right)_i &= \Delta_{\frac{R_z}{4}} C_{\text{tim}}(\mathbf{x},\mathbf{y},\mathbf{z}) \\ &= \frac{C_{\text{tim}}(\mathbf{x},\mathbf{y},\mathbf{z}^{(\text{top})}) - C_{\text{tim}}(\mathbf{x},\mathbf{y},\mathbf{z}^{(\text{bot})})}{R_z/4}. \end{aligned} \tag{30}$$

where $\mathbf{z}^{(\text{top})} = \mathbf{z} + (Z_{\text{top}} - z_i)\,\mathbf{e}_i$ and $\mathbf{z}^{(\text{bot})} = \mathbf{z} + (Z_{\text{bot}} - z_i)\,\mathbf{e}_i$, with $\mathbf{e}_i$ denoting the unit vector along the z-axis for cell i. These vectors represent the perturbed coordinates of cell i when it is forced onto the top or bottom die, respectively.

However, the FDA-based vertical gradient scale is inconsistent with the planar gradients. If left unnormalized, this disparity may dominate the update when overly large, or conversely, have a negligible impact when too small, thereby destabilizing the optimization

Table 1: Benchmark statistics used in our experiments.

Tech	Benchmark	#Cells	#Nets	T_{cycle} (ns)
ASAP7	NV_NVDLA_partition_m	24,324	24,504	0.40
	NV_NVDLA_partition_p	76,352	76,783	0.50
	aes_256	278,265	278,663	0.30
	hidden1	38,090	38,304	0.35
	hidden4	259,511	260,457	0.65
	hidden5	246,797	247,118	0.30
Nangate45	ibex	15,874	18,537	2.80
	jpeg_encoder	60,745	73,102	2.00

process. To ensure balanced updates, we introduce a preconditioned z-gradient. Specifically, we rescale the FDA-based $\widetilde{\nabla}_z C_{\text{tim}}$ such that its magnitude aligns with the average norm of the planar gradients

$$\nabla_z C_{\text{tim}} = \frac{\|\nabla_x C_{\text{tim}}\| + \|\nabla_y C_{\text{tim}}\|}{2\,\|\widetilde{\nabla}_z C_{\text{tim}}\|} \cdot \widetilde{\nabla}_z C_{\text{tim}}, \tag{31}$$

where $\|\cdot\|$ denotes the Euclidean norm. As illustrated in the flow graph in Figure 1, the gradient computation of the timing cost requires an additional step, *FDA z-gradient preconditioning*, which is designed to ensure that the vertical gradients remain stable and compatible with the planar ones. By defining the preconditioned vector as $\nabla_z C_{\text{tim}}$, we facilitate a unified gradient descent update form for all (x, y, z) dimensions, despite $\nabla_z C_{\text{tim}}$ not being a strict partial derivative.

5 Experimental Results

We evaluate our IDDA-3D by comparing wirelength, TNS, and WNS against wirelength-driven 2D and 3D placers. To demonstrate that our RC-based delay cost works without manual parameter tuning and adapts to different technology platforms, we conduct experiments on eight benchmarks: Six designs are in the ASAP7 [23] library, and two designs are in the Nangate45 [24] library. All benchmarks are purely logic circuits without macros.

5.1 Experimental Setup

The six ASAP7 designs are benchmarks from the ICCAD 2024 Contest Problem C [25]. We obtain the gate-level netlists in Verilog (.v) format. Since this contest problem focuses on gate sizing, the provided netlists contain several initial slew and capacitance violations, necessitating preprocessing in OpenROAD [21]. Specifically, we (1) remove all buffers (as recommended by OpenROAD, since buffer insertion will be handled later by `repair_design`), (2) randomly place I/O pins, (3) perform global placement with a target utilization of 60%, (4) apply the `repair_design` command to resolve slew and capacitance violations, and (5) after detailed placement, output the final DEF file for our 3D placer. For the Nangate45 library, two designs are included in our experiments. `ibex` and `jpeg_encoder` are obtained in gate-level Verilog format from the OpenROAD Flow Scripts project [26]. Similar to the ASAP7 flow, we randomly place I/O pins and perform global placement with the core utilization ratio set to 60%, then detailed placement. The results of these 2D preprocessed benchmarks are reported in Table 2.

All preprocessed 2D DEF files are used as inputs to IDDA-3D. To construct the 3D design, we scale the die area by 0.5 and generate both a bottom and a top die, making the placement region a cuboid with $R_x \approx R_y \approx R_z$. Each HBT is modeled as a standard cell, with size, resistance, and capacitance set to 0.5 μm, 0.5 Ω, and 0.2 fF, respectively. The HBT cell is incorporated into the standard-cell library so that STA can properly account for inter-die delay during placement. The hyperparameters are configured as follows: the cutsize coefficient is set to $\alpha = 1$ to limit HBT usage slightly; regarding the timing update schedule, we set the overflow threshold $OV_{th} = 0.35$ and the update frequency $n_{iter} = 15$ to balance gradient fidelity with runtime efficiency; the timing cost coefficient β is 0.1; the path-based weighting parameters are $w_{\text{base}} = 1$ and $w_{\text{cum}} = 0.02$; the density bin count follows the default setting of ePlace-3D [14], determined by the ratio of the total cell volume to the placement cuboid volume; the Lagrange multiplier λ together with the smoothing parameter γ in the WA function are updated according to the schedule of Xplace1.0 [27]. We implement our IDDA-3D in C++. All experiments are conducted on a Linux workstation running Ubuntu 20.04.6. The machine is equipped with an Intel® Core™ Ultra 7 265K processor (20 cores, 3.9 GHz) and 64 GB of memory.

5.2 Timing Optimization Analysis

Table 2 reports the results of 2D, 3D, and our timing-driven placer IDDA-3D. Here, 2D denotes the OpenROAD baseline [21], 3D refers to our reimplementation of the SOTA true-3D placer [16].

IDDA-3D delivers consistently superior timing. Relative to the 2D baseline, it improves TNS by 75% on average and WNS by 23%. Compared to the 3D baseline, IDDA-3D matches wirelength while substantially enhancing timing, reducing TNS and WNS by 44% and 22%, respectively. These gains arise from an RC-based intra-/inter-die delay model coupled with path-based weighting and a smooth, differentiable gradient tightly integrated with the analytical placer. The timing cost systematically shortens critical driver-sink connections, adapts weights based on path slack, and preserves stability during optimization. By explicitly balancing wirelength and timing, the formulation avoids overemphasis on timing, preventing detours or density violations, and thereby sustaining overall placement quality across heterogeneous designs and operating corners.

In terms of runtime, IDDA-3D incurs approximately 2× overhead compared with 3D baseline, which is mainly due to additional STA and timing cost gradient updates. Nevertheless, the overall runtime is practical given the timing improvements.

5.3 Ablation Study on FDA z-axis Gradients

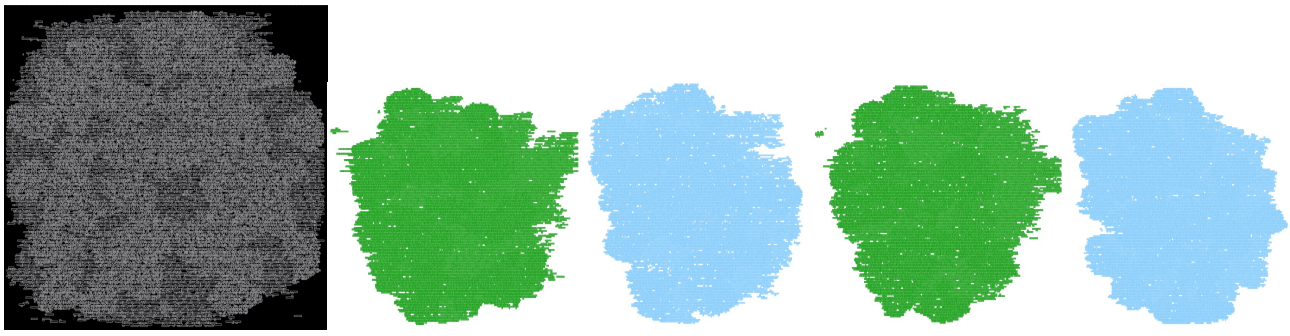
We conducted an ablation study to evaluate the critical role of FDA-based z-axis gradients in our timing optimization framework. As summarized in Table 3, removing FDA gradients severely hampered the solver's ability to navigate the discrete vertical dimension. Without the guidance of FDA, the optimizer failed to effectively assess the timing benefits of inter-die cell movement, leading to convergence failures in two specific benchmarks, **aes_256** and **hidden5**. For the converged designs, TNS and WNS degraded by 35% and 18% on average, respectively. Meanwhile, wirelength exhibits a slight reduction after applying FDA z-axis gradients. These results confirmed that FDA z-axis gradients were indispensable for escaping local optima and ensuring robust 3D timing optimization.

Table 2: Results on eight benchmarks. The 2D baseline corresponds to OpenROAD [21], and the 3D baseline refers to our reimplementation of [16]. Wirelength (WL) is reported in μm, total negative slack (TNS) in ns, worst negative slack (WNS) in ps, and runtime (RT) in seconds.

	2D [21]				Analytical-3D [16]					Ours				
Benchmark	WL	TNS	WNS	RT	WL	TNS	WNS	#HBTs	RT	WL	TNS	WNS	#HBTs	RT
NV_NVDLA_partition_m	**60,290**	-17.72	-367	4	60,700	-17.48	**-347**	599	60	61,779	**-16.85**	-351	690	283
NV_NVDLA_partition_p	**285,748**	-293.70	-384	17	319,814	-133.87	**-354**	632	129	333,531	**-97.46**	-412	786	406
aes_256	1,166,061	-66.54	189	61	1,109,336	-36.67	-156	2,325	1128	**1,065,214**	**-35.84**	**-71**	1571	2623
hidden1	129,622	-45.44	-218	6	**121,609**	-35.68	-183	542	77	124,836	**-25.80**	**-144**	533	214
hidden4	**1,198,507**	-792.40	-373	72	1,435,713	-332.88	**-330**	1,331	846	1,402,333	**-175.95**	-383	1363	1326
hidden5	996,534	-41.83	-207	48	905,401	-23.43	-188	1,839	973	**880,923**	**-6.52**	**-64**	1297	1542
ibex	**263,318**	-46.84	-255	3	274,959	-3.02	-60	454	28	275,012	**-1.79**	**-50**	414	78
jpeg_encoder	816,389	-153.40	-340	14	643,738	-63.00	-300	1605	128	**641,722**	**-0.11**	**-30**	1435	215
Ratio	1.03	4.05	1.30	0.03	1.02	1.79	1.27	1.15	0.50	1.00	1.00	1.00	1.00	1.00

Table 3: Ablation study of FDA z-axis gradients of C_{tim} on eight benchmarks. WL in μm, TNS in ns, WNS in ps, RT (runtime) in seconds.NA indicates benchmarks that failed to converge.

	Our w/o C_{tim} FDA z-gradient					Our w/ C_{tim} FDA z-gradient				
Benchmark	WL	TNS	WNS	#HBTs	RT	WL	TNS	WNS	#HBTs	RT
NV_NVDLA_partition_m	63,370	-23.78	-453	446	287	**61,779**	**-16.85**	**-351**	690	283
NV_NVDLA_partition_p	**330,843**	-129.54	**-405**	815	386	333,531	**-97.46**	-412	786	406
aes_256	NA	NA	NA	NA	NA	**1,065,214**	**-35.84**	**-71**	1571	2623
hidden1	**121,609**	**-23.88**	-150	542	189	124,836	-25.80	**-144**	533	214
hidden4	1,433,505	-259.70	-386	1568	1133	**1,402,333**	**-175.95**	**-383**	1363	1326
hidden5	NA	NA	NA	NA	NA	**880,923**	**-6.52**	**-64**	1297	1542
ibex	**273,338**	**-1.53**	**-40**	503	75	275,012	-1.79	-50	414	78
jpeg_encoder	659,478	-48.08	-220	1580	194	**641,722**	**-0.11**	**-30**	1435	215
Ratio	1.02	1.53	1.21	1.04	0.90	1.00	1.00	1.00	1.00	1.00



(a) 2D (b) 3D Bottom (c) 3D Top (d) 3D-TDP Bottom (e) 3D-TDP Top

Figure 5: Comparison across placement results of NV_NVDLA_partition_m: (a) 2D, (b) 3D bottom die, (c) 3D top die, (d) Our IDDA-3D bottom die, and (e) IDDA-3D top die. Note that the area of the 3D implementation is reduced by 50% compared to the 2D baseline.

6 Conclusion

This work presented IDDA-3D, a timing-driven placement framework for face-to-face bonded 3D ICs. The framework introduced an RC-based intra-/inter-die delay model that reflected parasitic information derived from the cell library, making it physically grounded and applicable across different technology nodes. In addition, a preconditioned FDA-based z-axis gradient was employed to provide stable gradients, ensuring robust optimization in the discrete 3D placement setting. On eight benchmarks, IDDA-3D achieved up to 44% TNS reduction and 22% WNS improvement, with negligible impact on wirelength and acceptable runtime. These results validated that our approach delivered significant timing improvements in 3D IC placement, making timing closure achievable with IDDA-3D.

References

[1] Anthony Agnesina, Moritz Brunion, Jinwoo Kim, Alberto Garcia-Ortiz, Dragomir Milojevic, Francky Catthoor, Gioele Mirabelli, Manu Komalan, and Sung Kyu Lim. Power, performance, area, and cost analysis of face-to-face-bonded 3-d ics. *IEEE Transactions on Components, Packaging and Manufacturing Technology*, 13(3):300–314, 2023.

[2] Xueyan Zhao, Weiguo Li, Zhisheng Zeng, Zhipeng Huang, Biwei Xie, Xingquan Li, and Yungang Bao. Toward Advancing 3D-ICs Physical Design: Challenges and Opportunities. In *Proc. ASPDAC*, pages 294–301, 2025.

[3] Jinwoo Kim, Lingjun Zhu, Hakki Mert Torun, Madhavan Swaminathan, and Sung Kyu Lim. A PPA Study for Heterogeneous 3-D IC options: Monolithic, Hybrid Bonding, and Microbumping. *IEEE TVLSI*, 32(3):401–412, 2023.

[4] Peiyu Liao, Siting Liu, Zhitang Chen, Wenlong Lv, Yibo Lin, and Bei Yu. DREAMPlace 4.0: Timing-driven Global Placement with Momentum-based Net Weighting. In *Proc. DATE*, pages 939–944, 2022.

[5] Jai-Ming Lin, You-Yu Chang, and Wei-Lun Huang. Timing-Driven Analytical Placement According to Expected Cell Distribution Range. In *Proc. ISPD*, pages 177–184, 2024.

[6] Linhao Lu, Wenxin Yu, Hongwei Tian, Chenjin Li, Xinmiao Li, Zhaoqi Fu, Zhengjie Zhao, and Jingwei Lu. Timing-Driven Global Placement With Hybrid Heuristics and Nadam-Based Net Weighting. In *Proc. DATE*, pages 1–7, 2025.

[7] Zizheng Guo and Yibo Lin. Differentiable-Timing-Driven Global Placement. In *Proc. DAC*, pages 1315–1320, 2022.

[8] Yunqi Shi, Siyuan Xu, Shixiong Kai, Xi Lin, Ke Xue, Mingxuan Yuan, and Chao Qian. Timing-driven Global Placement by Efficient Critical Path Extraction. In *Proc. DATE*, pages 1–7, 2025.

[9] Bon Woong Ku, Kyungwook Chang, and Sung Kyu Lim. Compact-2D: A Physical Design Methodology to Build Two-Tier Gate-Level 3-D ICs. *IEEE TCAD*, 39(6):1151–1164, 2019.

[10] Pruek Vanna-Iampikul, Chengjia Shao, Yi-Chen Lu, Sai Pentapati, and Sung Kyu Lim. Snap-3D: A Constrained Placement-Driven Physical Design Methodology for Face-to-Face-Bonded 3D ICs. In *Proc. ISPD*, pages 39–46, 2021.

[11] Lennart Bamberg, Alberto Garcia Ortiz, Lingjun Zhu, Sai Pentapati, Da Eun Shim, and Sung Kyu Lim. Macro-3D: A Physical Design Methodology for Face-to-Face-Stacked Heterogeneous 3D ICs. In *Proc. DATE*, pages 37–42, 2020.

[12] Sai Surya Kiran Pentapati, Kyungwook Chang, Vassilios Gerousis, Rwik Sengupta, and Sung Kyu Lim. Pin-3D: A Physical Synthesis and Post-Layout Optimization Flow for Heterogeneous Monolithic 3D ICs. In *Proc. ICCAD*, pages 1–9, 2020.

[13] Tung-Chieh Chen, Zhe-Wei Jiang, Tien-Chang Hsu, Hsin-Chen Chen, and Yao-Wen Chang. NTUplace3: An Analytical Placer for Large-Scale Mixed-Size Designs With Preplaced Blocks and Density Constraints. *IEEE TCAD*, 27(7):1228–1240, 2008.

[14] Jingwei Lu, Hao Zhuang, Ilgweon Kang, Pengwen Chen, and Chung-Kuan Cheng. ePlace-3D: Electrostatics based Placement for 3D-ICs. In *Proc. ISPD*, pages 11–18, 2016.

[15] Peiyu Liao, Yuxuan Zhao, Dawei Guo, Yibo Lin, and Bei Yu. Analytical Die-to-Die 3-D Placement with Bistratal Wirelength Model and GPU Acceleration. *IEEE TCAD*, 43(6):1624–1637, 2023.

[16] Yuxuan Zhao, Peiyu Liao, Siting Liu, Jiaxi Jiang, Yibo Lin, and Bei Yu. Analytical Heterogeneous Die-to-Die 3D Placement with Macros. *IEEE TCAD*, 2024.

[17] Yan-Jen Chen, Cheng-Hsiu Hsieh, Po-Han Su, Shao-Hsiang Chen, and Yao-Wen Chang. Mixed-Size 3D Analytical Placement with Heterogeneous Technology Nodes. In *Proc. DAC*, pages 1–6, 2024.

[18] Peter Spindler, Ulf Schlichtmann, and Frank M. Johannes. Abacus: Fast Legalization of Standard Cell Circuits with Minimal Movement. In *Proc. ISPD*, pages 47–53, 2008.

[19] Bangqi Fu, Lixin Liu, Martin DF Wong, and Evangeline FY Young. Hybrid Modeling and Weighting for Timing-driven Placement with Efficient Calibration. In *Proc. ICCAD*, pages 1–9, 2024.

[20] W. C. Elmore. The Transient Response of Damped Linear Networks with Particular Regard to Wideband Amplifiers. *Journal of Applied Physics (JAP)*, 19(1):55–63, 1948.

[21] Tutu Ajayi, Vidya A. Chhabria, Mateus Fogaça, Soheil Hashemi, Abdelrahman Hosny, Andrew B. Kahng, Minsoo Kim, Jeongsup Lee, Uday Mallappa, Marina Neseem, Geraldo Pradipta, Sherief Reda, Mehdi Saligane, Sachin S. Sapatnekar, Carl Sechen, Mohamed Shalan, William Swartz, Lutong Wang, Zhehong Wang, Mingyu Woo, and Bangqi Xu. Toward an Open-Source Digital Flow: First Learnings from the OpenROAD Project. In *Proc. DAC*, 2019.

[22] Tsung-Wei Huang and Martin D. F. Wong. OpenTimer: A High-Performance Timing Analysis Tool. In *Proc. ICCAD*, pages 895–902, 2015.

[23] Lawrence T. Clark, Vinay Vashishtha, Lucian Shifren, Aditya Gujja, Saurabh Sinha, Brian Cline, Chandarasekaran Ramamurthy, and Greg Yeric. ASAP7: A 7-nm finFET predictive process design kit. *Microelectronics Journal*, 53:105–115, 2016.

[24] Nangate45 Libraries.

[25] Bing-Yue Wu, Rongjian Liang, Geraldo Pradipta, Anthony Agnesina, Haoxing Ren, and Vidya A. Chhabria. 2024 ICCAD CAD Contest Problem C: Scalable Logic Gate Sizing Using ML Techniques and GPU Acceleration. In *Proc. ICCAD*, 2025.

[26] Openroad-flow-scripts.

[27] Lixin Liu, Bangqi Fu, Martin D. F. Wong, and Evangeline F. Y. Young. Xplace: An Extremely Fast and Extensible Global Placement Framework. In *Proc. DAC*, pages 1309–1314, 2022.