Invited: Toward Sustainable and Transparent Benchmarking for Academic Physical Design Research

Liwen Jiang
Fudan University
Shanghai, PRC
lwjiang24@m.fudan.edu.cn

Andrew B. Kahng
UC San Diego
La Jolla, CA, USA
abk@ucsd.edu

Zhiang Wang*
Fudan University
Shanghai, PRC
zhiangwang@fudan.edu.cn

Zhiyu Zheng
Fudan University
Shanghai, PRC
zyzheng24@m.fudan.edu.cn

Abstract

This paper presents RosettaStone 2.0, an open benchmark translation and evaluation framework built on OpenROAD-Research [1]. RosettaStone 2.0 provides complete RTL-to-GDS reference flows for both conventional 2D designs and Pin-3D-style face-to-face (F2F) hybrid-bonded 3D designs, enabling rigorous *apples-to-apples* comparison across planar and three-dimensional implementation settings. The framework is integrated within OpenROAD-flow-scripts (ORFS)-Research [2]; it incorporates continuous integration (CI)-based regression testing and provides a standardized evaluation pipeline based on the METRICS2.1 convention, with structured logs and reports generated by ORFS-Research. To support transparent and reproducible research, RosettaStone 2.0 further provides a community-facing leaderboard, which is governed by verified pull requests and enforced through Developer Certificate of Origin (DCO) compliance.

CCS Concepts

• **Hardware → Electronic design automation**.

Keywords

VLSI physical design, Heterogeneous Integration, Benchmarking

ACM Reference Format:
Liwen Jiang, Andrew B. Kahng, Zhiang Wang*, and Zhiyu Zheng. 2026. Invited: Toward Sustainable and Transparent Benchmarking for Academic Physical Design Research. 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/10.1145/3764386.3779611

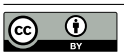
1 Introduction

The advancement of 2D and 3D VLSI physical design (PD) research is inherently dependent on rigorous benchmarking frameworks built upon complete flows, realistic benchmarks, and reproducible methodologies. Academic contests have historically provided benchmarks that drive innovation in physical design [3, 4]. However, these benchmarks usually lack the full set of inputs required by modern RTL-to-GDS flows; even Liberty, technology LEFs, parasitic models, and signoff constraints are often missing or only partially specified. Moreover, different publications may use different libraries, timing corners, or routing rules under the same testcase name, and scripts are frequently tied to specific tools or versions. As a result, reported quality-of-results (QoR) and runtime improvements are difficult to compare in an *apples-to-apples* way, especially across independent works.

Recent perspectives argue that the next physical design roadmap inflection is driven by fine-grain heterogeneous 3D integration [5], where bonding and backside processing make it practical to stack multiple active tiers and tailor per-tier device and interconnect choices to the function being implemented [6]. Despite strong interest in fine-grain 3D integration, there is still no widely adopted open-source RTL-to-GDS reference flow for Pin-3D-style face-to-face (F2F) hybrid bonding [7, 8], with maintainable 3D enablements and reproducible stage-wise checkpoints. Hence, the integration, fair comparison, and reproducibility of emerging 3D physical design algorithms across different tool versions and Process Design Kit (PDK) variants remain challenging. Reproducible evaluation must increasingly encompass both planar and 3D settings, while enabling more rigorous control not only of algorithmic choices, but of tool enablements and measurement protocols as well.

Several 3D integration styles have been explored. Through-silicon via (TSV) based stacking is well-established in commercial products such as high-bandwidth memory, but TSVs with 10 μm-scale pitch consume area and add large parasitics [9, 10]. Monolithic 3D (M3D) uses nanoscale monolithic inter-tier vias and sequential low-temperature processing [11] to achieve very high integration density, but remains costly and less mature [12]. By contrast, face-to-face hybrid bonding bonds two pre-fabricated dies using hybrid bonding terminals (HBTs) in the back-end-of-line (BEOL), with an approximately 1 μm-scale pitch and strong manufacturing support [13, 14]. Recent products such as AMD's Zen 4 and Meta's AR SoC adopt F2F hybrid bonding [15, 16], which makes F2F-based 3D integration a realistic and attractive target for academic 3D PD flows.

Recent academic research works have started to address challenges of reproducibility and standardization for 3D physical design. Open-3DBench [17] provides an open 3D-IC backend benchmarking framework built on OpenROAD-flow-scripts [18], covering partitioning, placement, routing, extraction, and thermal analysis across 3D flows. Meanwhile, the IEEE CEDA DATC Robust Design Flow (RDF) [19] and the ML EDA Commons [20] seek to advance shared benchmarks, metrics, and evaluation infrastructures based on OpenROAD [21] and ORFS, with a strong focus on artifact evaluation and sustainable research backplanes. Despite such efforts, a durable and open-source RTL-to-GDS reference flow for Pin-3D-style F2F hybrid bonding – with maintainable enablements and standardized evaluation checkpoints – remains unavailable. To advance the availability of fair and reproducible assessments for 3D PD flows, we make the following contributions.



ISPD '26, Bonn, Germany

ACM ISBN 979-8-4007-2314-8/2026/03
https://doi.org/10.1145/3764386.3779611

- **Sustainable benchmarking backplane.** We establish a maintained infrastructure within OpenROAD- and ORFS-Research, featuring co-versioned artifacts, CI-based regression, METRICS2.1-compatible reporting, and community-driven governance.
- **Pin-3D reference flow.** We release an end-to-end F2F 3D RTL-to-GDS reference flow, complete with 3D enablements (e.g., HBT modeling, tier libraries) and standardized checkpoints for reproducible evaluation.
- **RosettaStone 2.0 roadmap.** We outline the extension of this backplane to support systematic benchmark translation, synthetic netlist generation, and unified 2D and 3D validation under explicit evaluation contracts.

In the following, Section 2 describes prospects for OpenROAD-Research and ORFS-Research, and introduces RosettaStone 2.0 as a sustainable benchmarking backplane supporting reproducible 2D and Pin-3D physical design evaluation. Section 3 presents our initial Pin-3D enablement and RTL-to-GDS reference flow for F2F hybrid bonding. Section 4 reports example validation results and sensitivity analyses under a consistent evaluation contract. Section 5 outlines the roadmap toward the complete RosettaStone 2.0 framework, and we conclude in Section 6.

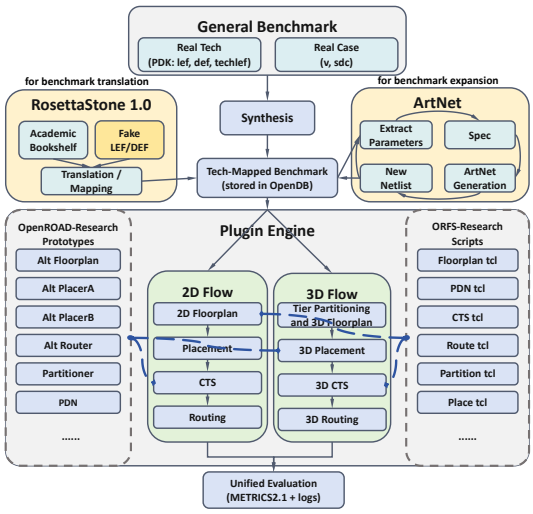


Figure 1: RosettaStone 2.0 roadmap (vision) to sustain and expand the open-source benchmarking backplane. Not all planned components are fully validated in this paper.

2 Sustainable Benchmarking Backplane Based on OpenROAD-Research and ORFS-Research

OpenROAD-Research [1] and ORFS-Research [2] are designed as open and collaborative platforms to guide future efforts that will strengthen open EDA infrastructure and community collaboration. They preserve algorithmic diversity by hosting academically validated but potentially non-production code, thus lowering the barrier to experimentation in emerging areas such as ML-assisted optimization and agentic flows; supporting education through reproducible implementations; and fostering global collaboration via contrib-style repositories. In contrast to the mainline repositories [21] [18], which focus on industrial-grade stability, the research backplanes prioritize transparency, reproducibility, and extensibility, making them a natural substrate for sustainable benchmarking.

Building on this backplane, we release the Pin-3D RTL-to-GDS reference flow as a set of versioned sub-tools within ORFS-Research. The Pin-3D flow provides an open, reproducible instantiation of fine-grained 3D physical design for face-to-face hybrid bonding, including curated flow scripts, maintainable 3D enablements, and METRICS2.1-compatible structured logging. The structured evaluation kernel enables consistent measurement of wirelength, routing congestion, total negative slack, design-rule violations, and runtime across all stages, allowing direct end-to-end comparison between alternative placement engines and the native OpenROAD-Research flow under identical downstream conditions.

While OpenROAD-Research and ORFS-Research provide the execution substrate, systematic benchmarking further requires robust mechanisms for benchmark translation, normalization, and coverage expansion. To this end, RosettaStone 2.0 provides an extended backplane that enables systematic translation of legacy academic benchmarks, integration of synthetic benchmark generation, and unified validation of both 2D and 3D flows under explicit evaluation contracts. These contracts define stage boundaries, exported artifacts, and measurement semantics, ensuring that results remain comparable and reproducible across tool versions, PDKs, and stack variants.

We frame RosettaStone 2.0 (Figure 1) as an advance over RosettaStone 1.0 [22]. Put together, OpenROAD-Research, ORFS-Research, our end-to-end Pin-3D flow and RosettaStone 2.0 form a concrete instance of a sustainable benchmarking backplane. Our goal is for this backplane to enable new algorithms and tool prototypes to be integrated, evaluated, and compared under shared conditions, supporting credible and cumulative progress in both 2D and 3D physical design research.

3 Pin-3D Enablement and Flow

This section presents Pin-3D as an initial, concrete realization of RosettaStone 2.0 within the IEEE CEDA DATC *ORFS-Research* repository. Our goal is not to claim best-possible QoR, but to demonstrate that RosettaStone 2.0 can deliver a maintained, reproducible RTL-to-GDS reference flow for F2F hybrid-bonded 3D designs, including versioned 3D enablements, stage-wise checkpoints, and METRICS2.1-compatible reporting. By instantiating Pin-3D on top of OpenROAD-Research and ORFS-Research, we show how new 3D methodologies can be packaged, validated, and compared under shared inputs and measurement semantics.

3.1 Overview

Figure 2 presents an overview of our homogeneous and heterogeneous Pin-3D flow, implemented on top of OpenROAD-Research and ORFS-Research. The flow comprises the following key stages:

- **Synthesis and 2D abstraction:** RTL is synthesized into a flat gate-level netlist using a common logical library, which is later mapped to tier-specific physical libraries for heterogeneous designs.
- **Partitioning and 3D floorplanning.** An initial 2D floorplan undergoes timing-driven bipartitioning [23] to assign cells to

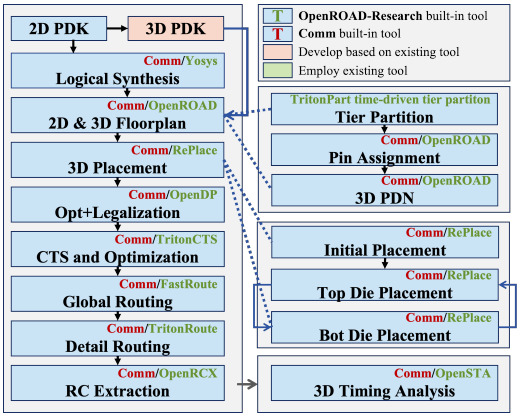


Figure 2: Overview of the Pin-3D flow built on OpenROAD-Research and ORFS-Research.

tiers. The design is then translated into tier-aware 3D views with independent power delivery networks.

- **Iterative alternating-tier 3D placement.** Placement refinement begins from a global 2D placement and proceeds by alternating optimization between tiers: one tier is fixed (loaded as **COVER**) while the other is optimized for timing/congestion, followed by tier-aware legalization; the roles are then swapped and iterated.
- **3D clock tree synthesis (CTS).** The clock tree is built on the bottom tier. Top-tier sinks connect to this tree through inter-tier vias, leveraging the unified 2D representation for vertical connectivity.
- **3D routing and optimization.** Routing utilizes the full metal stack, modeling hybrid bonding terminals as special vias. Parasitics are then extracted to drive post-route timing and power closure.
- **Metrics collection:** The flow records runtime, memory, timing, power, wirelength, congestion, and violations (DRVs/FEPs) in a structured format for both open-source and commercial reference flows.

We highlight three key features of our Pin-3D-style framework:

- **Unified 2D technology abstraction for F2F 3D.** We model HBTs as special vias in an extended 2D metal stack, enabling unmodified 2D routers to realize cross-tier connections.
- **Iterative tier-by-tier optimization and legalization with 2D placers.** We alternate placement optimization between tiers using **COVER** views, with tier-aware legalization to enforce tier-specific sites and masters for homogeneous and heterogeneous stacks.
- **End-to-end 3D reference flow with hybrid toolchain support.** A unified scripting interface runs the RTL-to-GDS flow using OpenROAD, commercial tools, or mixed toolchains. Stage-wise checkpoints and METRICS2.1-compatible reports are exported to enable consistent, reproducible measurement.

Throughout the flow, constraints and 3D enablements are aligned under an explicit evaluation contract. All metrics are reported using a consistent schema to eliminate ambiguity in metric definitions and measurement points.

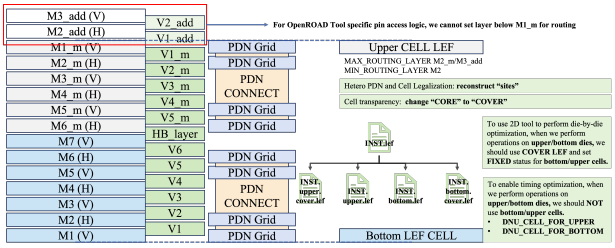


Figure 3: Metal stack, PDN strategy, and tier strategy in the 3D ASAP7 PDK.

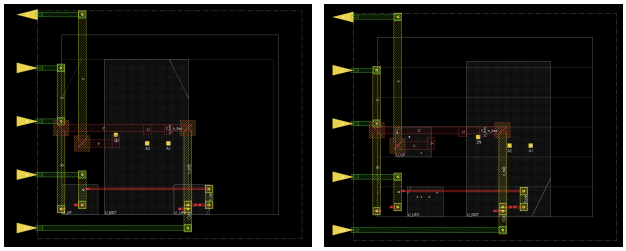


Figure 4: Rebuilding of rows for heterogeneous legalization.

3.2 PDK Preparation

Process Design Kit (PDK) preparation is a foundational component of a robust 3D physical design flow. A 3D PDK must support stable and automated execution, while still capturing the essential integration effects needed for meaningful tool and algorithm evaluation.

Figure 3 summarizes our 3D ASAP7 enablement, which is derived from the standard ASAP7 PDK [24, 25]. We construct a face-to-face stack by replicating the 2D metal stack per tier and introducing a dedicated cut layer to model hybrid bonding terminals (HBTs) as vertical vias. `M2_add` and `M3_add` are compatibility layers required by the legacy ORFS-Research router's pin-access limitations (i.e., upper-only pin access); this will be removed once the router is updated to support lower-metal pin access, consistent with the COMM TechLEF. Since the two dies in an F2F configuration are fabricated independently prior to bonding [13], we implement isolated power delivery networks (PDNs) on the two tiers. This design enables independent supplies and voltage domains, which is important for supporting heterogeneous stacks.

To support 3D physical implementation, we construct 3D standard-cell libraries by deriving tier-specific physical views from a base 2D library. For each logical cell, we generate distinct LEF masters for the bottom and top tiers (denoted by **_bottom** and **_upper** suffixes) by reassigning the cell's metal layers to the appropriate tier. We additionally provide **COVER** LEF views for both tiers, which serve as physical abstractions to exclude the inactive tier from overlap and density calculations while preserving connectivity [8].

For homogeneous stacks, logic synthesis targets the standard 2D library; the resulting netlist is later remapped to tier-specific masters following the partitioning stage. For heterogeneous stacks, we synthesize against a unified library that represents the intersection of available cells across the two technologies. To ensure compatibility, any physical pins not shared between the tiers are hidden as internal pins in this logical view, allowing the synthesized netlist to

be flexibly mapped to either tier. Although this intersection method keeps the flow simple, hiding of pins that are not shared requires careful checking that circuit function is preserved. It might also limit optimization scope, compared to natively utilizing multiple libraries. During the physical design stage, logical cells are assigned to specific tiers, and legality is ensured by reconstructing the rows with tier-specific sites (Figure 4).

3.3 Physical Design Flow Details

This subsection details the script-based implementation of each stage in Figure 2. Two design choices enable a standard 2D toolchain to operate on 3D designs: (i) a unified technology representation that models inter-tier connections as vias on an HBT cut layer, and (ii) a **tier strategy** that alternates optimization between tiers by loading **COVER** views for the inactive tier and restricting cell usage to the active tier.

Stage 1: Synthesis and technology-neutral netlist. We synthesize (using Yosys or a commercial tool) a flat gate-level netlist that is input to timing-driven bipartitioning. Homogeneous designs use the native 2D library, while heterogeneous designs use the unified logical library described in Section 3.2, producing a single netlist that is later mapped to tier-specific physical masters.

Stage 2: Floorplan, partitioning, and tier view generation. We first generate an initial 2D floorplan based on target utilization and aspect ratio. The floorplan and netlist are then passed to TritonPart for timing-driven bipartitioning. For heterogeneous stacks (ASAP7+NanGate45_3D in this paper), we additionally enable a capacity-aware baseline split to reflect technology-dependent standard-cell area differences. More precisely: we invoke TritonPart with a partition balance constraint, denoted as `UBfactor` (unbalance factor). After sweeping this constraint from `PAR_BAL_LO` to `PAR_BAL_HI`, we adopt the minimum cutsize solution found. This cutsize serves as an early estimate for HBTs.[1]

We then generate tier-specific DEF/Verilog views through a conversion step: (i) remapping logical masters to tier-specific physical masters (**_bottom**/**_upper**), (ii) applying pin remapping for heterogeneous designs, and (iii) tying off pins that are not present in the logical abstraction.

For homogeneous stacks (7+7 and 45+45), we assign IO pins on the bottom tier. For the heterogeneous stack (7+45), we assign IO pins on the top tier, since it hosts more cells and this choice empirically reduces cross-tier connections.

Finally, we build electrically isolated PDNs for each tier using symmetric grid topologies (Figure 6).

Stage 3: Iterative 3D placement with adaptive strategy. In addition to loading **COVER** views, we evaluate two tier strategies for placement and legalization in both homogeneous and heterogeneous stacks: (i) a **Restricted** strategy, which limits available masters to the active tier and keeps the inactive tier fixed; and (ii) a **Flexible** strategy, which does not enforce tier constraints and allows the placer to choose between tier-specific masters during optimization.

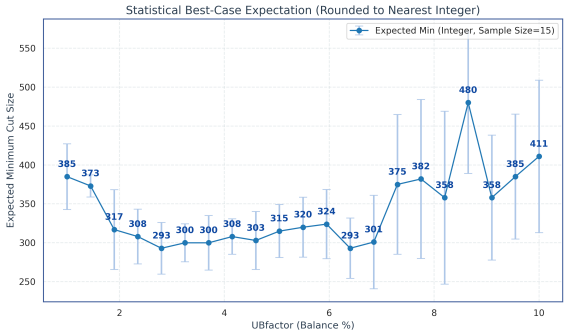


Figure 5: Cross-tier net count vs. `UBfactor` based on 50 random seeds per point. The trace shows the Expected Minimum Cutsize (bootstrapped, $N = 15$), representing the typical best result achievable.

Our reference flow uses the **Flexible** strategy, while we also study the **Restricted** strategy and report WNS and total power. The subsequent resizing and legalization stage is always **Restricted**, to prevent buffers from being inserted onto the inactive tier and creating tier-illegal solutions.

Stage 4: 3D clock tree synthesis. We perform CTS on a specific active tier (**bottom** for homogeneous stacks, **top** for the 7+45 stack) while holding the other tier fixed and utilize the **Restricted** strategy according to the selected tier. Clock sinks on the other tier are connected through inter-tier vias, leveraging the unified cut-layer abstraction. For the ORD flow, we run an additional legalization step after buffer insertion. Figure 7a shows the resulting clock tree and routed clock net.

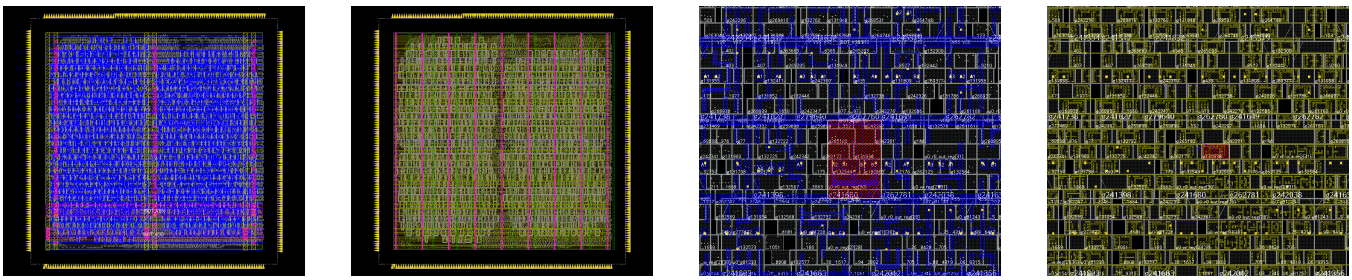
Stage 5: 3D routing and parasitic extraction. Global and detailed routing are executed with standard 2D routers. Cross-tier connections are realized by inserting HBT vias defined in the unified tech LEF. After routing, we run parasitic extraction using the same unified tech LEF, producing SPEF that includes inter-tier RC for final timing and verification (Figure 7).

Stage 6: Metrics collection and reporting. We collect QoR metrics at multiple checkpoints. For the OpenROAD tool, ORFS-Research reporting commands emit runtime, memory, wirelength, timing, and violations in METRICS2.1-style structured JSON. For COMM, we run an equivalent set of built-in reports and parse them into CSV with consistent metric definitions. All scripts, inputs, and outputs are version-controlled to ensure reproducible evaluation.

4 Pin-3D Validations

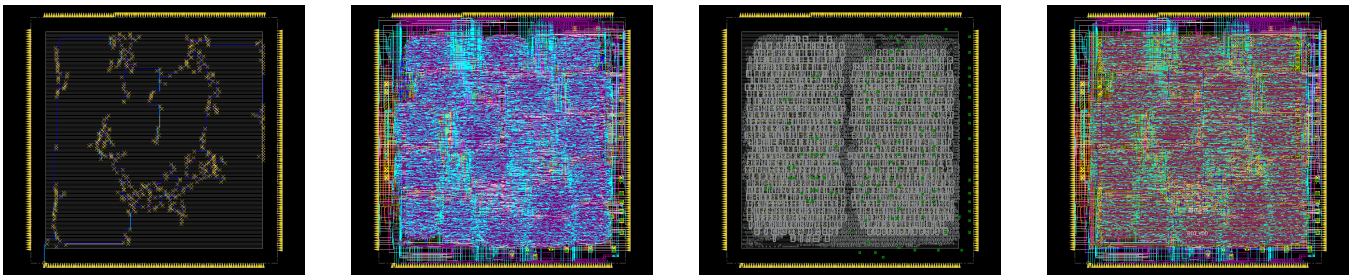
This section reports experimental studies that showcase how the Pin-3D flow and ORFS-Research can enable fair and reproducible physical-design (PD) comparisons across tools and algorithms. We report a range of results from both the open-source flow (ORD) and

[1]We implement two independent sweep modes, each evaluated over multiple `PAR_BAL_ITERATION` points (11 by default), and select the minimum-cut solution in timing-driven mode. *(A) UBfactor sweep.* We sweep the imbalance constraint (`UBfactor`) uniformly from `PAR_BAL_LO` to `PAR_BAL_HI`, run timing-driven bipartitioning at each point, and select the solution with the minimum cutsize. *(B) Base-balance sweep.* To account for unequal effective placement capacities across tiers, we sweep a technology-aware `base_balance` while keeping timing-driven partitioning enabled. For ASAP7+NanGate45, we start from `base_balance`$=(0.06, 0.94)$ (reflecting $\sim$16$\times$ standard-cell area ratio) and linearly reduce the imbalance over 11 points toward a less skewed split; at each point, we run TritonPart and pick the minimum-cut solution. In both modes, the selected cutsize provides an early estimate of cross-tier connectivity (hybrid bonding terminals).



(a) Bottom PDN (M1 stripe in blue) (b) Top PDN (M1_m stripe in yellow) (c) Bottom cells (in blue) (d) Top cells (in yellow)

Figure 6: Heterogeneous PDN grid and cell placement (*aes*, 7+45).



(a) Top die clock tree & routed clock net (b) Post-route optimized layout (c) HBTs assigned by the router (green) (d) Heterogeneous 3D *aes*

Figure 7: HBTs and routed signal nets (*aes*, 7+45).

a commercial reference flow (COMM) to demonstrate portability, flexibility, and end-to-end execution; absolute quality-of-results (QoR) measurement is not a primary objective of this work. All experiments are run on a Linux server with an Intel Xeon Gold 6230N CPU and 503 GB RAM.

We conduct the following experiments:

- **3D/2D baseline studies:** three designs across multiple enablements (Tables 2 and 3).
- **Tier strategy comparison: Restricted** vs. **Flexible** (Table 4).
- **Mixed-toolchain ablations:** swapping synthesis and backend components to localize QoR gaps (Table 5).
- **Runtime studies:** stage-wise breakdowns (Figure 8).
- **Sensitivity studies:** (i) HBT pitch (Figures 9 and 10); and (ii) clock period (Figure 11).

We use OpenROAD-Research (v2025.11) as the open-source implementation flow (**ORD**), and a commercial RTL-to-GDS implementation flow as an external reference baseline (**COMM**). To minimize evaluator-induced discrepancies in reported outcomes, we use a leading commercial P&R tool, **Cadence Innovus** (v21.39), as a common *evaluator* to report timing, power, and physical violations (DRC/DRV/FEP) for *all* experiments at matched checkpoints. By contrast, the commercial synthesis engine and the commercial P&R engine used to *produce* COMM solutions are left unnamed. Aside from this tool anonymity, we emphasize that COMM is used strictly as a reference to contextualize trends observed with the open infrastructure, rather than for any head-to-head tool comparison.

We evaluate three benchmark designs (*aes*, *ibex* and *jpeg*) under five technology enablements: two 2D baselines (*ASAP7* and *NanGate45*) and three 3D stacks (*ASAP7_3D*, *NanGate45_3D*, and a heterogeneous stack denoted as *ASAP7+NanGate45_3D*) [25, 26]. In our naming convention, the first technology corresponds to the *top* tier and the second corresponds to the *bottom* tier; for brevity, we also denote the three 3D stacks as **7+7**, **45+45**, and **7+45**.

Table 1: Hybrid Bonding Terminal (HBT) geometry settings.

Parameter	Value	Description
Width	0.5 μm	Side length of the square cut
Spacing	0.5 μm	Minimum edge-to-edge distance
Pitch	1.0 μm	Effective center-to-center pitch
Resistance	0.02 Ω	Parasitic resistance per HBT

Table 1 lists the nominal geometry for Hybrid Bonding Terminals (HBTs) used in our main experiments. We model HBTs as square inter-tier vias with a 1.0 μm pitch. Since this size is fixed, its impact varies across technology nodes. In 45+45, the blockage caused by HBTs is manageable. In 7+7, however, an HBT spans more routing tracks, which increases congestion and makes pin access more difficult. To illuminate sensitivities to scaling, Figure 9 shows results from varying HBT pitch while maintaining WIDTH = SPACING and PITCH = WIDTH + SPACING.

In all experiments, we target a fixed 60% core utilization and derive the die outline from the synthesized design size.[2] Both toolchains are evaluated at the typical–typical (TT) corner. Parasitics are extracted using a unified 3D technology LEF that captures the 3D stack definition. Power is estimated in vectorless mode with a default switching activity, so the reported differences primarily reflect physical implementation effects rather than stimulus

[2]In Table 2, *Core* is the footprint area of a single tier. We assume symmetric floorplans, so both tiers share the same outline. *StdCell* is the total standard-cell area summed over the top and bottom tiers, including logic, buffers, and physical-only cells. For the 2D results in Table 3, *Core* and *StdCell* follow standard single-die definitions.

variations. For routing, we limit the detailed-routing engine to 20 iterations to generate the final routed DEF.

We collect consistent metrics, including runtime, memory, routed wirelength (rWL), timing (WNS/TNS), and physical/timing violations (Design Rule Violations (DRVs) and Failing Endpoints (FEPs)).

Table 2: 3D implementation results across stacks, designs, and flows. Results are obtained using the Flexible tier strategy.

Enablement	Design	Flow	Clock	Area (μm^2)		Power	rWL	Timing (ns)		Violations		HBT
			(ns)	Core	StdCell	(mW)	(mm)	WNS	TNS	DRVs	FEPs	Cnt
45+45	aes	COMM	0.820	13,342.6	16,446.5	20.34	232.2	-0.016	-0.043	3	6	904
		ORD	0.820	15,262.0	18,100.8	46.34	193.4	-0.064	-1.092	8	55	650
	ibex	COMM	2.200	18,295.3	22,244.2	11.35	228.8	0.147	0.000	8	0	1,072
		ORD	2.200	24,022.5	30,307.2	25.80	291.6	-0.508	-255.542	1	1,175	2,041
	jpeg	COMM	1.200	60,049.5	68,321.0	75.05	386.0	0.086	0.000	13	0	1,492
		ORD	1.200	86,059.5	98,157.5	122.10	614.3	0.008	0.000	2	0	2,880
7+7	aes	COMM	0.380	892.8	1,260.2	7.20	57.6	-0.020	-0.305	5,415	64	1,111
		ORD	0.380	1,410.5	1,866.8	25.53	70.8	-0.087	-4.957	16,779	156	2,261
	ibex	COMM	1.000	1,340.4	1,650.1	5.71	72.4	-0.009	-0.064	3,119	27	1,027
		ORD	1.000	1,965.5	2,655.7	10.99	128.4	-0.449	-260.812	39,311	1,371	4,466
	jpeg	COMM	0.680	3,532.6	4,064.9	26.20	103.6	-0.002	-0.002	2,632	1	1,445
		ORD	0.680	5,445.1	6,788.3	47.68	158.2	0.061	0.000	14,270	0	3,328
7+45	aes	COMM	0.820	1,958.5	2,311.8	5.96	80.4	-0.607	-60.910	19	227	122
		ORD	0.820	1,946.4	2,443.6	7.84	64.7	0.052	0.000	217	0	88
	ibex	COMM	2.200	2,611.7	3,147.0	7.69	78.8	-1.264	-438.321	124	539	257
		ORD	2.200	4,501.7	4,823.4	5.23	110.6	-0.043	-0.043	1,351	1	671
	jpeg	COMM	1.200	7,976.3	9,342.2	20.3	133.9	0.006	0.000	11	0	491
		ORD	1.200	11,927.9	13,820.0	29.92	236.4	0.241	0.000	3,686	0	1,946

Table 2 and Table 3 respectively summarize representative 3D and 2D results under a common evaluation contract.[3] The purpose of these tables is to demonstrate the *flexibility* of the ORFS- and RosettaStone 2.0-based implementation and the Pin-3D enablements: the infrastructure enables controlled studies across stacks, benchmarks, and toolchains with consistent checkpoints and metric semantics. Notably, the 3D results (Table 2) are produced using ORFS-Research to exercise the Pin-3D enablements, whereas the 2D baselines (Table 3) use the default 2D ORFS flow [18]. This split is intentional: it highlights that RosettaStone 2.0 can accommodate both stable mainline flows and research-enablements within a unified benchmarking and reporting framework.

Table 2 reports relatively high HBT counts, especially for the 7+7 stack. A primary contributor to this is cross-tier buffer insertion, which can typically add up to two extra HBTs (and increase routing complexity) when both incident nets become cross-tier. Prior work [8] shows that strictly forbidding cross-tier buffering can reduce HBT usage, but may also degrade overall QoR. In our reference flow, we keep this optimization enabled for robustness and report the resulting HBT counts for transparency.[4]

We also clarify the main DRC contributors in the 7+7 stack. Because our HBT geometry is large relative to the 7+7 routing pitch, the DRCs of COMM flow are dominated by HBT-induced `SHORT/CUTSPACING` violations originating from the relatively oversized HBT features under 7+7 rules. To keep the evaluation contract consistent across all settings, we cap detailed routing to only 20 iterations in all flows, which can leave residual violations.

[3]For the 7+45 case, **COMM** is evaluated by importing a saved design database at the reporting checkpoint (instead of direct DEF readback) to avoid import-related errors. For **ORD**, metrics are extracted by reading DEF, Verilog, and SDC at the matched checkpoint. Numbers are shown only to demonstrate end-to-end flow completeness and are not intended for any direct comparison.

[4]We have separately examined **ORD** flow outcomes when the 0.380 ns target clock period for *aes* in Table 2 is relaxed to 5 ns. TritonPart reports a cutsize of 147 (IO excluded). With the 5 ns clock period, the final layout contains 784 HBTs (reduced from 2,261) and 651 cross-tier nets (229 IO, 422 internal); the final DRC count is 96 (reduced from 16,779). The remaining cross-tier connectivity and the gap between HBT count and physical cross-tier net count mainly come from cross-tier buffer insertion, CTS clock buffering, and subsequent optimization.

Table 3: 2D implementation results using the default 2D ORFS flow and an unnamed 2D commercial flow.

Enablement	Design	Flow	Clock	Area (μm^2)		Power	rWL	Timing (ns)		Violations	
			(ns)	Core	StdCell	(mW)	(mm)	WNS	TNS	DRVs	FEPs
NanGate45	aes	COMM	0.820	28,870.8	16,157.1	21.6	249.3	-0.013	-0.074	0	17
		ORFS	0.820	28,376.9	20,142.6	48.9	212.9	-0.037	-0.681	0	47
	ibex	COMM	2.200	36,842.9	22,094.0	12.1	229.0	0.019	0.000	0	0
		ORFS	2.200	47,332.0	29,529.5	20.6	257.7	-0.094	-5.401	0	247
	jpeg	COMM	1.200	114,557.4	68,179.0	78.4	397.4	0.011	0.000	0	0
		ORFS	1.200	147,735.9	93,869.5	143.6	552.5	-0.189	-45.772	0	702
ASAP7	aes	COMM	0.380	1,691.7	1,229.6	6.6	52.6	-0.005	-0.020	0	9
		ORFS	0.380	2,786.2	1,950.5	30.9	62.3	-0.041	-2.498	161	148
	ibex	COMM	1.000	2,515.7	1,579.2	5.2	65.6	-0.016	-0.891	0	152
		ORFS	1.000	3,512.3	2,445.3	10.9	78.7	-0.078	-38.599	564	977
	jpeg	COMM	0.680	6,505.7	4,016.0	24.7	99.6	0.001	0.000	3	0
		ORFS	0.680	10,207.9	6,410.3	46.4	158.0	0.008	0.000	1,249	0

Impact of tier strategy. Table 4 gives example comparisons between the **Restricted** and **Flexible** tier strategies across stacks and designs. The observed effects vary by design and stack, and changes in timing, area/power, and HBT usage do not always move in the same direction.

Table 4: Effect of tier strategy on flow outcomes.

Enablement	Design	Strategy	StdCell Area (μm^2)	Power (mW)	rWL (mm)	WNS (ns)	TNS (ns)	HBT Count
45+45 ORD	aes	Restricted	18 109.5	46.4	197.7	−0.132	−2.110	749
		Flexible	18 100.8	46.3	193.4	−0.064	−1.092	650
	ibex	Restricted	30 423.2	26.4	311.3	−1.175	−755.922	2432
		Flexible	30 307.2	25.8	291.6	−0.508	−255.542	2041
	jpeg	Restricted	98 168.6	122.6	629.5	−0.159	−1.106	3055
		Flexible	98 157.5	122.1	614.3	0.008	0.000	2880
7+7 COMM	aes	Restricted	1420.2	7.6	58.4	−0.014	−0.669	1097
		Flexible	1260.2	7.2	57.6	−0.020	−0.305	1111
	ibex	Restricted	1659.8	5.7	71.2	−0.022	−4.206	993
		Flexible	1650.1	5.7	72.4	−0.009	−0.064	1027
	jpeg	Restricted	4084.9	26.5	102.9	0.003	0.000	1364
		Flexible	4064.9	26.2	103.6	−0.002	−0.002	1445
7+45 COMM	aes	Restricted	2295.5	4.4	80.4	−0.004	−0.005	43
		Flexible	2311.8	6.0	80.4	−0.607	−60.910	122
	ibex	Restricted	3107.6	2.6	77.6	0.110	0.000	260
		Flexible	3147.0	7.7	78.8	−1.264	−438.321	257
	jpeg	Restricted	9343.8	20.3	137.5	0.010	0.000	501
		Flexible	9342.2	20.3	133.9	0.006	0.000	491

Analysis of QoR gap contributors. To explore differences between ORD and COMM flow outcomes, we run *aes* in the 45+45 enablement under six mixed toolchain configurations: (1) **COMM**: commercial synthesis + commercial backend; (2) **Mixed1**: commercial synthesis + OpenROAD placement + commercial CTS/routing; (3) **Mixed2**: commercial synthesis + OpenROAD backend; (4) **Mixed3**: Yosys synthesis + commercial backend; (5) **Mixed4**: Yosys synthesis + commercial placement + OpenROAD CTS/routing; (6) **ORD**: Yosys synthesis + OpenROAD backend.

As shown in Table 5, **Mixed1** achieves outcomes similar to those of **COMM**, suggesting that OpenROAD placement is not a primary bottleneck. In contrast, replacing commercial synthesis with Yosys (**Mixed3**) noticeably increases power; this points to synthesis as an area for improvement. Degradation also occurs in the backend: switching to OpenROAD CTS/routing (**Mixed1→Mixed2**,

Mixed3→Mixed4) leads to a modest increase in dynamic power and implementation violations.

Table 5: QoR comparison across mixed toolchain configurations (*aes*, 45+45).

Config	StdCell Area (μm^2)	Power (mW)	rWL (mm)	WNS (ns)	DRVs	FEPs	HBT
COMM	16 446.5	20.3358	232.2	−0.016	3	6	904
Mixed1	16510.1	22.5075	237.5	-0.040	0	54	1031
Mixed2	16193.3	24.5887	209.8	-0.135	153	144	522
Mixed3	17604.7	38.7790	195.2	0.012	0	0	982
Mixed4	17664.3	45.2540	189.7	-0.094	0	87	915
ORD	18100.8	46.3385	193.4	-0.064	8	55	650

Runtime analysis. We also examine stage-wise runtimes of the Pin-3D flow under OpenROAD-Research and the commercial reference flow, across toolchains and technologies. Figure 8 shows the runtime breakdown for the *jpeg* design across three 3D configurations. Routing dominates the total runtime in all cases. While OpenROAD-Research shows efficiency in pre-routing steps such as PDN, placement, and legalization, its routing is much slower, leading to longer overall runtimes. This appears to be mainly driven by the high number of metal layers and persistent DRVs, and presents a clear opportunity for tool improvement.

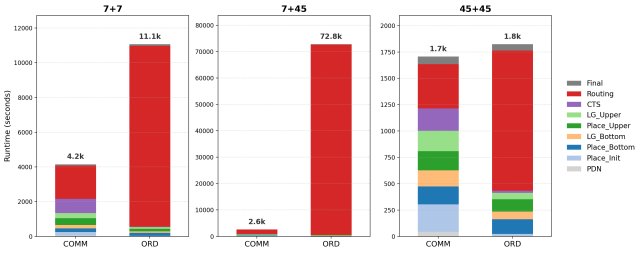


Figure 8: Runtime breakdowns for jpeg. Numbers above the bars indicate total runtime in seconds.

Hybrid Bonding Terminal pitch sweep. Figure 9 sweeps the hybrid bonding terminal (HBT) pitch for the testcase *aes*, 7+7. (As noted earlier, the HBT geometry is defined such that WIDTH = SPACING and the pitch equals WIDTH + SPACING.) As HBT pitch shrinks beyond a certain point, the total DRV count drops in both ORD and COMM. Notably, when HBT feature size is reduced to be comparable to a typical upper-metal via (`V6_m`-level pitch size), the DRV count for *aes*, 7+7 drops to a near-zero level (51 and 88, respectively) in the ORD and COMM flows.

Figure 10 breaks down DRVs by layer. At larger HBT pitch, violations concentrate on the HBT layer and its adjacent metal layers `M6_m` and `M7`. As HBT pitch decreases, the HBT layer-related portion shrinks and the HBT-related `SHORT` and `CUTSPACING` types largely vanish, consistent with the drop in total DRVs.

Clock-period sweep. Figure 11 studies the impact of clock constraints on implementation closure for *aes*, 45+45. We sweep the target clock period in 0.05 ns steps. For each target clock period, we re-run logic synthesis and then keep a fixed 60% target utilization

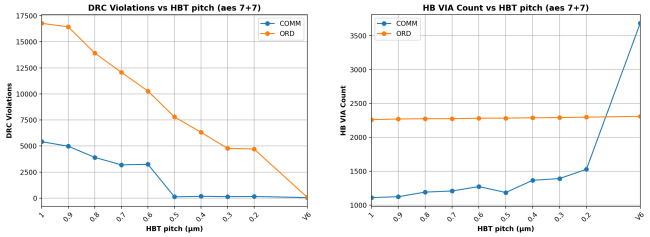


Figure 9: Impact of HBT pitch scaling on routing closure: DRV count (left) and HBT via count (right) versus HBT pitch.

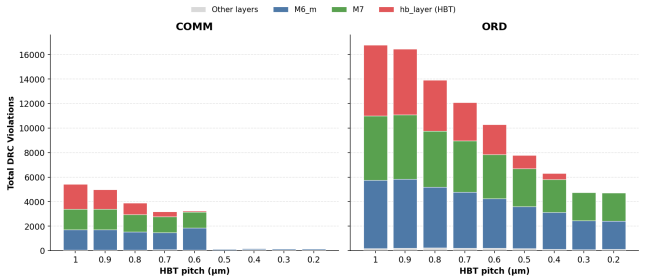


Figure 10: Impact of HBT pitch on DRVs. As the HBT pitch decreases, HBT-related errors (colored) vanish.

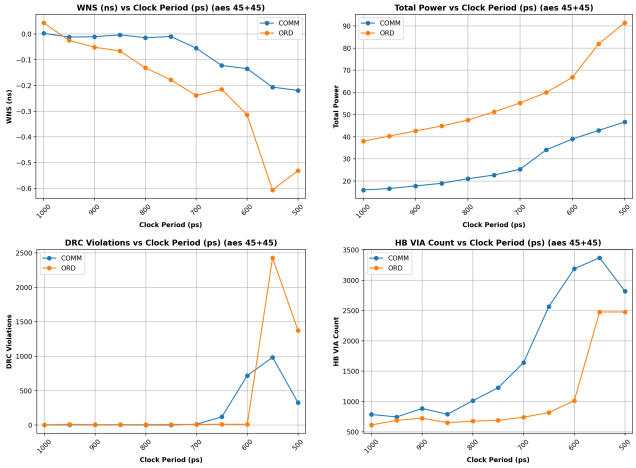


Figure 11: Impact of clock constraint on QoR (*aes*, 45+45).

for floorplanning, so that the observed QoR trends primarily reflect timing pressure induced by the constraint.

To reduce run-to-run noise, we apply a simple local denoising procedure. For each target clock period T, we also run the two neighboring settings at $T - 0.01$ ns and $T + 0.01$ ns, and report each metric as the average over $\{T - 0.01\,\text{ns}, T, T + 0.01\,\text{ns}\}$ outcomes.

Higher target frequency typically increases both timing and routability pressure. As the clock period decreases, meeting timing requires more aggressive optimization, typically increasing buffering and total cell count and raising routing demand. Accordingly,

rWL and total power tend to increase under tighter clocks, and endpoint repair becomes more difficult.

In the 3D setting, tighter clocks can also increase cross-tier connectivity. More cross-tier connections imply more HBTs, which add blockage and routing constraints in the unified-technology abstraction. As a result, DRVs can rise from near-zero levels as the clock period shrinks, indicating increased routing difficulty around HBT features under higher timing pressure.

5 Roadmap to RosettaStone 2.0

RosettaStone 2.0 is designed as a maintained component of the OpenROAD-Research project, enabling seamless integration between legacy academic benchmarks, modern physical design data models, and end-to-end evaluation flows. In contrast to RosettaStone 1.0 [22], which relied on external and loosely coupled conversion utilities, RosettaStone 2.0 is tightly coupled with OpenROAD-Research and ORFS-Research to ensure long-term sustainability.

Beyond simple format conversion, RosettaStone 2.0 introduces benchmark normalization and repair mechanisms to sanitize legacy inputs. Notably, it automatically filters or fixes common inconsistencies found in academic benchmarks. Examples of this include removing ill-formed nets (nets with all input or output pin) and splitting overly-large standard-cell instances into multiple smaller ones. Such repairs preserve the original design's intent while making the benchmarks valid for modern physical design tools, thereby ensuring physical realizability and fair evaluation of results. The framework also supports a wide range of benchmark formats, including traditional Bookshelf contest benchmarks from past academic competitions, as well as simplified or incomplete "fake" LEF/DEF design representations used in placement research. This broad compatibility means that RosettaStone 2.0 can ingest legacy benchmarks and bring them into a consistent, up-to-date RTL-to-GDS flow.

To address the limited availability of real-world designs for stress-testing algorithms, RosettaStone 2.0 also integrates with ArtNet, a hierarchical clustering-based synthetic netlist generator. ArtNet can produce scalable artificial circuits that mimic realistic design characteristics under user-controlled parameters (size, hierarchy, etc.). This allows researchers to generate new testcases on demand, complementing the fixed set of public benchmarks with structurally representative synthetic designs. All such cases are handled through the same unified RosettaStone interface and benefit from the framework's normalization and logging features. Together, these enhancements make RosettaStone 2.0 a robust and extensible benchmarking platform that lowers entry barriers and enables fair comparisons across both 2D and emerging 3D physical design flows, including the Pin-3D flow.

We now summarize the supported input classes, translation paths, and evaluation methodology that together instantiate the RosettaStone 2.0 flow.

A. Input classes. RosettaStone 2.0 supports multiple classes of academic benchmarks: traditional Bookshelf-based contest benchmarks, simplified or "fake" LEF/DEF representations common in placement research, and synthetic designs. To scale the number of evaluation instances beyond finite public benchmarks, ArtNet [27, 28] can be used to generate structurally representative, configurable, and reproducible netlists. Each synthetic design is packaged with standardized enablements, ensuring that synthetic and real designs share the same flow stages and measurement methodology.

B. Translation paths. For Bookshelf benchmarks, RosettaStone 2.0 directly translates the input into the OpenROAD internal database format (.odb) via OpenDB [29] APIs, preserving placement and netlist data while resolving mismatches in scale, grid alignment, and technology abstraction. For benchmarks provided in incomplete or "fake" LEF/DEF formats, which often lack sufficient PDK information, RosettaStone 2.0 follows a complementary path: it first reconstructs an equivalent Bookshelf representation, then remaps this form to target PDKs to generate technology-consistent LEF/DEF files compliant with modern design rules. This ensures that even academic benchmarks can be evaluated in realistic, technology-aware settings.

C. Evaluation and metrics. Once the .odb is generated, OpenROAD-Research executes an end-to-end RTL-to-GDS-style backend flow, while ORFS-Research provides structured flow control. RosettaStone 2.0 aligns with the METRICS2.1 convention and ORFS-Research's structured logging infrastructure, enabling transparent reporting of runtime, quality-of-results (QoR), and intermediate-stage metrics. By embedding this framework into the OpenROAD-Research continuous integration (CI) pipeline, we ensure that results remain reproducible and that the evaluation backplane evolves consistently with the underlying tools.

6 Conclusion

We contribute toward the goal of sustainable and transparent benchmarking for academic physical design research, by establishing RosettaStone 2.0 as a maintained backplane within the OpenROAD-Research, ORFS-Research ecosystem. The backplane co-versions artifacts with the toolchain, continuously validates regressions via CI, and standardizes evaluation through METRICS2.1-style structured reporting with community-facing governance. As an end-to-end example, we release and validate a Pin-3D-style F2F hybrid-bonded 3D RTL-to-GDS reference flow with 3D enablements and stage-wise checkpoints. Finally, we outline a roadmap toward a complete RosettaStone 2.0 framework that systematically supports benchmark generation/expansion and unified 2D/3D flow validation.

Acknowledgment

The work of Liwen Jiang, Zhiang Wang and Zhiyu Zheng is supported partly by AI for Science Program, Shanghai Municipal Commission of Economy and Informatization (2025-GZL-RGZN-BTBX-02038), and partly by Fudan Kunpeng&Ascend Center of Cultivation. Experimental studies are performed at the Fudan Kunpeng&Ascend Center of Cultivation, Fudan University. We thank Sayak Kundu and Matt Liberty for helpful discussions of the Pin-3D enablement and flow.

References

[1] "OpenROAD-Research," https://github.com/ieee-ceda-datc/OpenROAD-Research, Accessed: 2025-11-27.

[2] "ORFS-Research," https://github.com/ieee-ceda-datc/ORFS-Research, Accessed: 2025-11-27.

[3] "ISPD 2005 Placement Contest Benchmark Suite," https://www.ispd.cc/contests/05/contest.htm, Accessed: 2025-11-27.

[4] "ICCAD 2015 Contest Benchmarks," https://www.iccad-contest.org/2015/problem_C/default.html, Accessed: 2025-11-27.

[5] Y. Zhao, P. Liao, S. Liu, J. Jiang, Y. Lin, and B. Yu, "Analytical Heterogeneous Die-to-Die 3D Placement with Macros," *IEEE Trans. CAD*, 44(2) (2025), pp. 402–415.

[6] M. Brunion, N. K. Purayil, F. Dell'Atti, S. Lam, R. Bilgic, M. Tahoori, L. Benini, and J. Ryckaert, "CMOS 2.0: Redefining the Future of Scaling," *Proc. ICCAD*, 2025, pp. 1–8.

[7] S. S. K. Pentapati, K. Chang, V. Gerousis, R. Sengupta, and S. K. Lim, "Pin-3D: A Physical Synthesis and Post-Layout Optimization Flow for Heterogeneous Monolithic 3D ICs," *Proc. ICCAD*, 2020, pp. 1–9.

[8] S. Pentapati, K. Chang, and S. K. Lim, "Pin-3D: Effective Physical Design Methodology for Multidie Co-Optimization in Monolithic 3D ICs," *IEEE Trans. CAD*, 43(4) (2024), pp. 1009–1022.

[9] A. C. Fischer, M. Grange, N. Roxhed, R. Weerasekera, D. Pamunuwa, G. Stemme, and F. Niklaus, "Wire-Bonded Through-Silicon Vias with Low Capacitive Substrate Coupling," *J. Micromech. Microeng.*, 21(8) (2011), art. 085035.

[10] V. Vartanian, L. Smith, K. Hummler, S. Olson, S. Sapp, T. Barbera, S. Golovato, K.-H. Yu, T. Hasegawa, S. Hu, G. Leusink, K. Maekawa, J. Enloe, A. Gracias, G. Pattanaik, and F. Wafula, "Cost Analysis of TSV Process and Scaling Options," *International Symposium on Microelectronics*, 2014(1) (2014), pp. 1–7.

[11] S. Jadhav, "Architecture of 3D Integrated Circuits," Medium blog post, Feb. 2021. https://medium.com/3d-ics/architecture-of-3-dimensional-integrated-circuits-602f5d9a7b58, Accessed: 2025-08-14.

[12] X. Zhao, W. Li, Z. Zeng, Z. Huang, B. Xie, X. Li, and Y. Bao, "Toward Advancing 3D-ICs Physical Design: Challenges and Opportunities," *Proc. ASP-DAC*, 2025, pp. 294–301.

[13] S. Liu, J. Jiang, Z. He, Z. Wang, Y. Lin, B. Yu, and M. Wong, "Routing-Aware Legal Hybrid Bonding Terminal Assignment for 3D Face-to-Face Stacked ICs," *Proc. ISPD*, 2024, pp. 75–82.

[14] C. Netzband, K. Ryan, Y. Mimura, I. Son, H. Aizawa, N. Ip, X. Chen, H. Fukushima, and S. Tan, "0.5 μm Pitch Next Generation Hybrid Bonding with High Alignment Accuracy for 3D Integration," *Proc. ECTC*, 2023, pp. 1100–1104.

[15] B. Munger, K. Wilcox, J. Sniderman, C. Tung, B. Johnson, R. Schreiber, C. Henrion, K. Gillespie, T. Burd, H. Fair, et al., "Zen 4: The AMD 5 nm 5.7 GHz x86-64 Microprocessor Core," *Proc. ISSCC*, 2023, pp. 38–39.

[16] T. F. Wu, H. Liu, H. E. Sumbul, L. Yang, D. Baheti, J. Coriell, W. Koven, A. Krishnan, M. Mittal, M. T. Moreira, M. Waugaman, L. Ye, and E. Beigné, "A 3D Integrated Prototype System-on-Chip for Augmented Reality Applications Using Face-to-Face Wafer Bonded 7 nm Logic at <2 μm Pitch with up to 40% Energy Reduction at Iso-Area Footprint," *Proc. ISSCC*, 2024, pp. 210–212.

[17] Y. Shi, C. Gao, W. Ren, S. Xu, K. Xue, M. Yuan, C. Qian, and Z.-H. Zhou, "Open3DBench: Open-Source Benchmark for 3D-IC Backend Implementation and PPA Evaluation," *arXiv preprint*, 2025, arXiv:2503.12946.

[18] "OpenROAD-flow-scripts," https://github.com/The-OpenROAD-Project/OpenROAD-flow-scripts, Accessed: 2025-11-27.

[19] V. A. Chhabria, A. Ghose, V. Gopalakrishnan, A. B. Kahng, S. Kundu, Y. Liu, Z. Wang, and B.-Y. Wu, "Invited: IEEE DATC RDF-2025: Enabling an EDA Research Ecosystem," *Proc. ICCAD*, 2025, pp. 1–9.

[20] V. A. Chhabria, J. Hu, A. B. Kahng, and S. S. Sapatnekar, "Invited: Toward an ML EDA Commons: Establishing Standards, Accessibility, and Reproducibility in ML-Driven EDA Research," *Proc. ISPD*, 2025, pp. 93–101.

[21] "OpenROAD," https://github.com/The-OpenROAD-Project/OpenROAD, Accessed: 2025-11-27.

[22] A. B. Kahng, M. Kim, S. Kim, and M. Woo, "RosettaStone: Connecting the Past, Present, and Future of Physical Design Research," *IEEE Design & Test*, 39(5) (2022), pp. 70–78.

[23] I. Bustany, G. Gasparyan, A. B. Kahng, I. Koutis, B. Pramanik, and Z. Wang, "An Open-Source Constraints-Driven General Partitioning Multi-Tool for VLSI Physical Design," *Proc. ICCAD*, 2023, pp. 1–9.

[24] L. T. Clark, V. Vashishtha, L. Shifren, A. Gujja, S. Sinha, B. Cline, C. Ramamurthy, and G. Yeric, "ASAP7: A 7-nm FinFET Predictive Process Design Kit," *Microelectronics Journal*, vol. 53, July 2016, pp. 105–115.

[25] "ASAP7," https://github.com/The-OpenROAD-Project/asap7, Accessed: 2025-11-27.

[26] NCSU EDA, "NanGate45," https://eda.ncsu.edu/freepdk/freepdk45/, Accessed: 2025-11-27.

[27] A. B. Kahng, S. Kang, S. Park, and D. Yoon, "ArtNet: Hierarchical Clustering-Based Artificial Netlist Generator for ML and DTCO Application," *arXiv preprint*, 2025, arXiv:2510.13582.

[28] "ArtNet," https://github.com/shypark98/ArtNet, Accessed: 2025-11-27.

[29] "OpenDB," https://github.com/The-OpenROAD-Project/OpenDB, Accessed: 2025-11-27.