

# PROBE3.0: A Systematic Framework for Design-Technology Pathfinding With Improved Design Enablement

Suhyeong Choi, *Graduate Student Member, IEEE*, Jinwook Jung<sup>ID</sup>, *Member, IEEE*,

Andrew B. Kahng<sup>ID</sup>, *Fellow, IEEE*, Minsoo Kim<sup>ID</sup>, *Member, IEEE*, Chul-Hong Park, *Member, IEEE*,

Bodhisatta Pramanik, *Graduate Student Member, IEEE*, and Dooseok Yoon, *Graduate Student Member, IEEE*

**Abstract**—We propose a systematic framework to conduct design-technology pathfinding for power, performance, area, and cost (PPAC) in advanced nodes. Our goal is to provide a configurable, scalable generation of process design kit (PDK) and standard-cell library, spanning key scaling boosters (backside PDN and buried power rail), to explore PPAC across given technology and design parameters. We build on Cheng et al. (2022), which addressed only area and cost (AC), to include power and performance (PP) evaluations through automated generation of full design enablements. We also improve the use of artificial designs in the PPAC assessment of technology and design configurations. We generate more realistic artificial designs by applying a machine learning-based parameter tuning flow to Kim et al. (2022). We further employ *clustering-based cell width-regularized placements* at the core of routability assessment, enabling more realistic placement utilization and improved experimental efficiency. We evaluate PPAC across scaling boosters and artificial designs in a predictive technology node.

**Index Terms**—Design-technology co-optimization (DTCO), place-and-route (P&R), physical design, routability, standard cell, technology pathfinding, VLSI CAD.

## I. INTRODUCTION

DUCE TO the slowdown of dimension scaling relative to the trend of the traditional Moore's law, scaling boosters, such as backside power delivery networks (BSPDNs) and buried power rails (BPRs), are introduced at advanced technology nodes. Since scaling boosters are important for optimizing power, performance, area, and cost (PPAC) of advanced technologies, accurate and fast evaluations and

Manuscript received 25 April 2023; revised 2 October 2023; accepted 8 November 2023. Date of publication 20 November 2023; date of current version 21 March 2024. This work was supported in part by DARPA under Grant HR0011-18-2-0032; in part by NSF under Grant CCF-1564302 and Grant CCF-2112665; in part by Qualcomm; in part by Samsung Electronics; and in part by the C-DEN Center. This article was recommended by Associate Editor I. H.-R. Jiang. (*Corresponding author: Minsoo Kim*.)

Suhyeong Choi was with the Semiconductor R&D Center, Samsung Electronics, Hwaseong 18448, South Korea. He is now with the Department of Electrical Engineering, Stanford University, Stanford, CA 94305 USA.

Jinwook Jung is with the Physical Design Department, IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598 USA.

Andrew B. Kahng, Bodhisatta Pramanik and Dooseok Yoon are with the Department of Electrical and Computer Engineering, University of California at San Diego, La Jolla, CA 92093 USA.

Minsoo Kim is with the Advanced Technology Group, NVIDIA Corporation, Austin, TX 78717 USA (e-mail: mik226@ucsd.edu).

Chul-Hong Park is with the Semiconductor Business, Hyundai MOBIS, Seoul 06232, South Korea.

Digital Object Identifier 10.1109/TCAD.2023.3334591

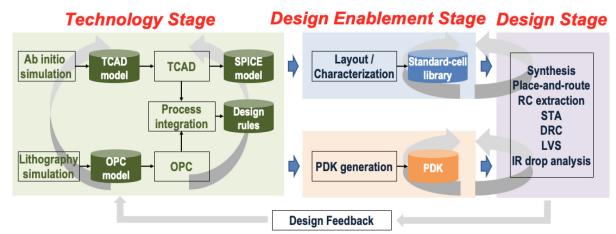


Fig. 1. Overview of the DTCO process. Figure is redrawn from [43].

predictions of PPAC are critical at an early stage of technology development. However, the use of scaling boosters comes with increased complexity in evaluations and predictions owing to the large number of design parameters they introduce.

Design-technology co-optimization (DTCO) is a key element in development of advanced technology nodes and designs in modern VLSI. Today's DTCO spans assessment and co-optimization across almost all components of semiconductor technology and design enablement. As described in Fig. 1, the DTCO process comprises three stages: 1) *Technology*; 2) *Design Enablement*; and 3) *Design*. First, the technology stage includes modeling and simulation methodologies for process and device technology. Second, the design enablement stage includes creation of process design kits (PDKs) needed in the ensuing design stage, with device models, standard-cell libraries, routing technology files, and interconnect parasitic (RC) models. Finally, the design stage includes logic synthesis and place-and-route (P&R) based on the PDKs generated in the design enablement stage.

To evaluate and predict technology and design at advanced nodes, all three stages must be correctly performed, and PDKs must be generated from technology and design enablement stages. However, the DTCO process is not simple: feedback from the design stage to the technology stage takes weeks to months of turnaround time, along with immense engineering efforts. Also, based on the design feedback, additional PDKs may need to be generated at the design enablement stage, which requires additional time. To reduce the turnaround time and maximize the benefit of the DTCO process, a fast and accurate DTCO methodology is needed to assess PPAC with reasonable turnaround time, and to more precisely guide multimillion dollar decisions for technology development.

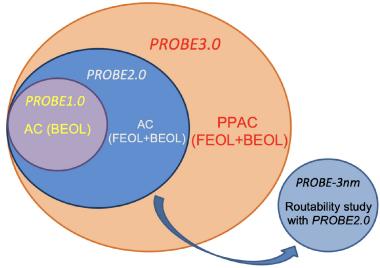


Fig. 2. Scope of PROBE-related works. PROBE1.0 [10] and PROBE2.0 [4] address AC given BEOL and FEOL/BEOL, respectively. PROBE-3nm [5] studies routability (AC) with sub-3-nm technology configurations. PROBE3.0 provides true full-stack PPAC pathfinding with automatic generation of EDA tool enablements.

*Contributions of Our Work:* Compared to the previous works PROBE1.0 [10] and PROBE2.0 [4], our new framework provides three main technical achievements.

1) *We Establish the First Comprehensive End-to-End Design and Technology Pathfinding Framework:*

Cheng et al. [4] and Kahng et al. [10] focused on area and cost (AC) without considering power and performance (PP) resulting in a significant deviation from the actual DTCO process in the industry. We propose a more complete and systematic PROBE3.0 framework, which incorporates PP aspects for design-technology pathfinding in early technology development. PROBE3.0 enables fast and accurate PPAC evaluation by generating configurable PDKs, including standard-cell libraries.

2) *We Improve Our Designs for PPAC Explorations:* For enhanced PPAC explorations, we utilize artificially generated designs from [13] to broaden our solution space.

To create more realistic artificial designs, we develop a machine learning (ML)-based parameter tuning flow (Section V) to find the best input parameters. Further, we propose a *clustering-based cell width-regularization* in Section VI to achieve more realistic utilization (and faster routability assessment).

3) *We Demonstrate the PPAC Exploration of Scaling Boosters:* We incorporate scaling boosters (BSPDN and BPR) to support P&R and IR drop analysis flows within the framework (Section IV). We show that incorporating BSPDN and BPR leads up to 8% and 24% reduction in power and area respectively, based on our predictive 3 nm technology. This is consistent with those reported in previous industry works [7], [18], [28], [30], which have demonstrated area reductions of 25%–30% through the use of BSPDN and BPR techniques.

## II. RELATED WORK

In this section, we divide the relevant previous works into the three categories of 1) advanced-technology research PDKs, 2) design-technology co-optimization and 3) scaling boosters, along with 4) “PROBE” frameworks.

*Advanced Technology Research PDKs:* PDKs of advanced node technologies are highly confidential. Academic research can be blocked by limited access to relevant information. To unblock academic research, predictive advanced-node PDKs have been published. ASAP7 [6] is a predictive PDK for

7 nm FinFET technology that includes standard cells which support commercial logic synthesis and P&R. FreePDK3 [19], [33] and FreePDK15 [1] are open-source PDKs for 3- and 15-nm technology. Kim et al. [12] proposed a 3-nm predictive technology called NS3K with nanosheet FETs (NSFETs). Kim et al. [12] also created 5-nm FinFET and 3-nm NSFET libraries to compare power, performance, and area.

*DTCO:* Previous DTCO works evaluate block-level PPAC and optimize design and technology simultaneously. Song et al. [21] proposed UTOPIA to evaluate block-level PPAC with thermally limited performance, and to optimize device and technology parameters. UTOPIA uses closed-source TSMC N10 technology, while our work’s PDK is fully open-sourced. Liebmann et al. [14] proposed a fast pathfinding DTCO flow for FinFET and complementary FET (CFET). Kahng et al. [9] described power delivery network (PDN) pathfinding for 3-D IC technology to study tradeoffs between IR drop and routability. In contrast, our work focuses on backside PDN and BPR. Cheng et al. [3] used ML to predict sensitivities to changes for DTCO based on 2.5–4.5T cells, while we evaluate technologies based on a more realistic range of 5–7T.

*Scaling Boosters:* As described in Section I, scaling boosters are used in advanced nodes to maximize the benefit of new technology. BSPDN and BPR are among the most promising scaling boosters in sub-5-nm nodes. Prasad et al. [17] carry out a CPU implementation with BSPDN and BPR in their 3-nm technology, demonstrating a reduction of up to 7× in worst IR drop. Similarly, [18] investigates BSPDN and BPR at sub-3-nm nodes and finds that they can lead to a 30% reduction in area based on IR drop mitigation. Chava et al. [2] also explore the impact of BSPDN and BPR on design, concluding that their use can lead to 43% area reduction with 4× less IR drop. Hossen et al. [7] study BSPDN configurations with  $\mu$ TSVs and observe 25%–30% reduction in the area using BSPDN and BPR. Additionally, [20] investigates BSPDN with nTSVs and  $\mu$ TSVs and finds that the average IR drop with BSPDN improves by 69% compared to traditional frontside PDN (FSPDN). Finally, [23] conducts holistic evaluations for BSPDN and BPR, demonstrating that FSPDN with BPR achieves a 25% lower on-chip IR drop, while BSPDN with BPR achieves an 85% lower on-chip IR drop with iso-performance and iso-area. In contrast to these previous DTCO works, we propose a highly *configurable* framework that enables more efficient investigation of scaling boosters in advanced nodes.

*“PROBE” Frameworks:* Prior “PROBE” [4], [10] works propose systematic frameworks for assessing routability with different FEOL and BEOL configurations. Specifically, [10] begins with an easily routable placement and increases the routing difficulty by random neighbor-swaps until the routing fails with greater than a threshold number of design rule violations (DRCs). On the other hand, [4] introduces an automatic standard-cell layout generation using satisfiability modulo theory (SMT) to support explorations of both FEOL and BEOL configurations. Additionally, [5] employs PROBE2.0 in a routability study with sub-3-nm technology configurations.

## III. STANDARD-CELL LIBRARY AND PDK GENERATION

Expediting the DTCO process requires automation of the standard-cell library and PDK generation flows. Therefore, the PROBE2.0 framework [4] introduces standard-cell layout

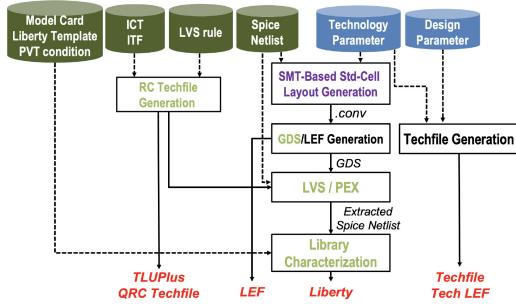


Fig. 3. Automatic standard-cell library and PDK generation (*Design Enablements*) in the PROBE3.0 framework. In addition to technology and design parameters, other technology-related inputs are required: 1) device model cards; 2) Liberty templates; 3) PVT conditions; 4) interconnect technology files (ICT or ITF formats); 5) LVS rules; and 6) SPICE netlists.

and PDK generation flows and utilizes them for routability assessments. In this work, we extend the PROBE2.0 framework to include proper electrical models of standard-cell libraries and interconnect layers for design-technology pathfinding. Additionally, we enhance the PDK generation flow to support advanced nodes. While the PROBE2.0 framework solely focuses on the physical layout of standard cells, the PROBE3.0 framework enables true full-stack PPAC pathfinding through automated, configurable standard-cell and PDK generation flows for advanced nodes. To demonstrate the use of PROBE3.0 for advanced-node PPAC pathfinding, we use a technology that incorporates cutting-edge (3-nm FinFET) technology predictions based on the works of [6] and [36].

#### A. Overall Flow

Fig. 3 describes our overall flow of standard-cell and PDK generation. Technology and design parameters are defined as input parameters for the flow. Beyond these input parameters, there are additional inputs required to generate standard-cell libraries and PDKs, as follows: 1) SPICE model cards; 2) Liberty template and process/voltage/temperature (PVT) conditions; 3) interconnect technology files (ICT/ITF); 4) LVS rule deck; and 5) SPICE netlists. Given the inputs, our SMT-based standard-cell layout generation and GDS/LEF generation are executed sequentially. Generation of timing and power models (Liberty) requires additional steps, including LVS, parasitic extraction, and library characterization flow. Aside from the library library generation, we also generate interconnect models from ICT/ITF, and P&R routing technology files from technology and design parameters. The PDK elements that we generate feed seamlessly into commercial logic synthesis and P&R tools. Further, to the best of our knowledge, ours is the first-ever work that is able to disseminate all associated EDA tool scripts for research purposes.

#### B. PROBE3.0 Technology

We build our own predictive 3-nm technology node, called the *PROBE3.0 technology*. We define FEOL and BEOL layers based on [6], which is the most complete and latest open-source PDK. Layer names and descriptions are as in [44]. We assume that all BEOL layers are unidirectional routing layers. Hence, we first change M1 to a unidirectional routing layer with a vertical preferred direction, since the work of [6] has

TABLE I  
LIST OF 41 STANDARD CELLS PER GENERATED LIBRARY

Cell List	Size
Inverter (INV), Buffer (BUF)	X1, X2, X4, X8
2-input AND/OR/NAND/NOR (AND2/OR2/NAND2/NOR2)	X1, X2
3-input AND/OR/NAND/NOR (AND3/OR3/NAND3/NOR3)	X1, X2
4-input NAND/NOR (NAND4/NOR4)	X1, X2
2-1 AND-OR-Inverter (AOI21), 2-2 AND-OR-Inverter (AOI22)	X1, X2
2-1 OR-AND-Inverter (OAI21), 2-2 OR-AND-Inverter (OAI22)	X1, X2
D flip-flop (DFFHQN), D flip-flop with reset (DFFRNQ)	X1
2-input MUX/XOR (MUX2/XOR2), Latch (LHQ)	X1

a bidirectional M1 routing layer. We add an M0 layer with horizontal preferred direction below the modified M1 layer and add contact layers V0 and CA which, respectively, connect between M1 and M0, and between gate/source-drain and M0.

Also, electrical features of technologies are critical to explore “PP” aspects. Therefore, parasitic extractions of standard cells and BEOL metal stacks are important steps. To extract parasitic elements, interconnect technology files are required to use commercial RC extraction, P&R, and IR drop analysis tools. In this work, we use commercial tools [31], [39], [42] for extractions, and each tool has its own technology file format.<sup>1</sup> Interconnect technology files include layer structures of technology and electrical parameters, such as thickness, width, resistance, and dielectric constant. We refer to the values of physical features in the 3-nm FinFET technology of [36], such as fin pitch/width, gate pitch/width, metal pitch (MP), and aspect ratio to emulate 3-nm technology characteristics. We also refer to [36] for the values for electrical parameters, such as via resistance and dielectric constant.

#### C. Improved Standard-Cell Library Generation

We generate standard-cell libraries via several steps illustrated in Fig. 3: 1) SMT-based standard-cell layout generation; 2) generation of GDS and LEF files; 3) LVS and PEX flow; and 4) library characterization flow.

*SMT-Based Standard-Cell Layout Generation:* In recent technology nodes, standard-cell architectures use a variety of pitch values for different layers in order to optimize PPAC. To accommodate this, PROBE3.0 improves the SMT-based layout generation used in PROBE2.0 to support nonunit gear ratios for M1 pitch (M1P) and contacted poly pitch (CPP).

Our standard-cell layouts are generated using SPICE netlists, technology and design parameters from [4]. However, in PROBE3.0 we change two key parameters: MP and PDN. Instead of using MP, we define parameters for the pitch values of each layer. Since M0, M1, and M2 layers are used for standard-cell layouts, we define M0P, M1P, and M2P as pitches of M0, M1, and M2 layers, respectively. Fig. 4 shows four layouts of AND2\_X1 cells with four parameter settings (*Lib1*, *Lib2*, *Lib3*, and *Lib4*) that are detailed in Table IX of Section VII. For our PPAC exploration, we generate 41 cells for each standard-cell library as shown in Table I.

*GDS/LEF Generation and LVS/PEX Flow:* While [4] only supports LEF generation for P&R, PROBE3.0 generates

<sup>1</sup>The MIPT file format is for *Siemens Calibre* [39] for extraction and is converted to an RC rule file for standard-cell layout extractions. Conversely, the ICT and ITF file formats are for *Cadence* and *Synopsys* extraction tools, respectively. We convert ICT to QRC techfile, and ITF to TLUPplus file, to enable P&R tools and IR drop analysis.

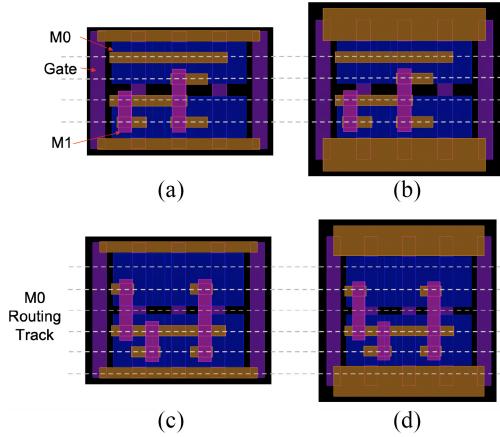


Fig. 4. Example standard cells (AND2\_X1) in this work. The cells are generated by our SMT-based standard-cell layout generation with the four parameter sets: (a) *Lib1*, (b) *Lib2*, (c) *Lib3*, and (d) *Lib4*.

standard-cell layouts in both GDS and LEF formats. The GDS files are used to extract parasitics from standard-cell layouts and check LVS between layouts and schematics. We use *Calibre* [39] to check LVS and generate extracted netlists for standard cells with intracell RC parasitics. All scripts are open-sourced in [44].

*Library Characterization Flow:* We perform library characterization to generate standard-cell libraries in the Liberty format. The inputs to the flow are model cards for FinFET devices obtained from [33], Liberty template including PVT conditions, and interconnect technology files. For the Liberty template, we define the PVT conditions, and the capacitance and transition time indices of  $(7 \times 7)$  tables for electrical models (delay, output transition time, and power). We use 5, 10, 20, 40, 80, 160, and 320 ps as the transition time indices. For the input capacitance, we obtain the input pin capacitance  $C_{inv}$  of an X1 inverter, then multiply this value by predefined multipliers (i.e., 2, 4, 8, 16, 24, 32, and 64). For characterization, we use the PVT corner ( $TT, 0.7\text{ V}, 25^\circ\text{C}$ ).

#### IV. POWER DELIVERY NETWORK

We study PDN scaling boosters to showcase the DTCO and pathfinding capability of PROBE3.0. There are two key challenges of traditional PDNs at advanced technologies.

- 1) *High Resistance of BEOL* [15]: Elevated resistance in BEOL layers exacerbates IR drop issues, necessitating denser PDN topologies.
- 2) *Routing Overheads (Routability)* [22]: PDN occupies routing resources that are shared with signal and clock wiring. The routability and area density impact of PDN become more severe with denser PDN.

To overcome these challenges, multiple foundries have started implementing BSPDN and BPR as scaling boosters in their sub-5-nm technologies. Similarly, we use BSPDN and BPR, to demonstrate the use of PROBE3.0. We establish four options for *PDN* parameter: 1) FSPDN without BPR ( $P_{FS}$ ); 2) FSPDN with BPR ( $P_{FB}$ ); 3) backside PDN without BPR ( $P_{BS}$ ); and 4) backside PDN with BPR ( $P_{BB}$ ). Fig. 5 illustrates the four PDN configurations in the PROBE3.0 framework.

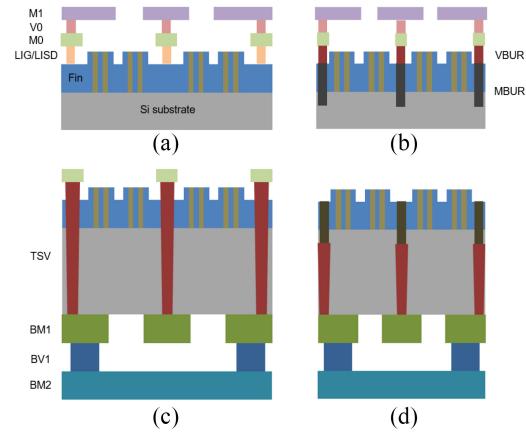


Fig. 5. Cross section view of four PDN options in the PROBE3.0 framework: (a) FSPDN ( $P_{FS}$ ); (b) FSPDN with BPR ( $P_{FB}$ ); (c) backside PDN ( $P_{BS}$ ); and (d) backside PDN with BPR ( $P_{BB}$ )

TABLE II  
PDN CONFIGURATIONS FOR FSPDN AND BSPDN. A PAIR OF POWER (VDD) AND GROUND (VSS) STRIPES ARE PLACED EVERY PITCH, WHILE MAINTAINING THE SPACING BETWEEN VDD AND VSS.  
*Density* DENOTES THE PERCENTAGE OF ROUTING TRACKS OCCUPIED BY PDN PER LAYER

PDN	Layer	Pitch (um)	Width (um)	Spacing (um)	Density (%)
FSPDN	M3	1.08	0.012	0.508	4
	M4	1.152	0.032	0.544	11
	M5-M11	5.0	1.0	1.5	20
	M12-M13	4.32	1.8	0.36	100
BSPDN	BM1-BM2	4.32	1.8	0.36	100

#### A. Frontside and Backside Power Delivery Network

We have defined realistic structures for both FSPDNs and BSPDNs and enabled IR drop analysis within our framework. Table II shows the configurations for FSPDN and BSPDN. Note that *Density* in Table II means the track occupancy of PDN routing divided by the total track resource in each given layer; it is not an area density. For example, in the case of M3 in Table II, 96% of routing tracks can be used for signal (or clock) while 4% of routing tracks are preoccupied by PDN. Since BEOL layers with smaller pitches (e.g., 24-nm-pitch layer) have high resistance, we add power stripes for every layer. While the work of [4] has multiple options for FSPDN, the PROBE3.0 framework has one PDN structure for FSPDN. Instead, we add the options of  $P_{FB}$ ,  $P_{BS}$ , and  $P_{BB}$ . Through this process, PDN exploration space is shrunk from 12 (three density options and four metal options) to 4. Furthermore, while the *Backside* option in [4] assumes no PDN at the frontside for the BSPDN option, we add power stripes at the backside for BSPDN to enable IR drop analysis.

Fig. 5(a) and (c), respectively, shows cross section views of  $P_{FS}$  and  $P_{BS}$  options. The  $P_{FS}$  option has M0 power and ground pins for standard cells, which connect to power stripes at the frontside of the die. The  $P_{BS}$  option uses the same M0 power and ground pins for standard cells but connects to power stripes at the backside of the die. For the  $P_{BS}$  option, we employ two backside metal layers (BM1 and BM2) and one via layer (BV1) between the backside metal layers. The layer characteristics (width, pitch, and spacing) are identical to the top two layers (M12 and M13) of FSPDN. Additionally, the M0 pins of standard cells and BSPDN are connected using

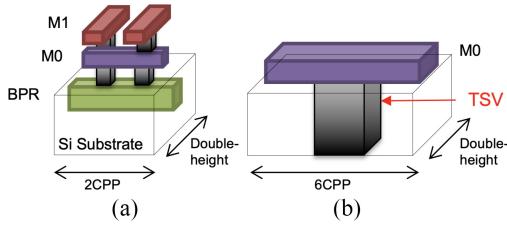


Fig. 6. Power tap cells for (a)  $P_{FB}$  and (b)  $P_{BS}$ .

through-silicon vias (TSVs). We assume nano-TSVs with 90 nm [20] width for the  $P_{BS}$  option and 1:10 width-to-height aspect ratio. For the  $P_{BS}$  option, TSV insertions necessitate reserved spaces in front-end-of-line (FEOL) layers, including keepout margins surrounding the TSVs. To accommodate this, we insert *power tap cells* prior to standard-cell placement.

### B. Frontside and Backside PDN With Buried Power Rail

In advanced nodes, power rails on BEOL metal layers can be “buried” into FEOL levels with shallow-trench isolation (STI). Using a deep trench and creating space between devices lowers the resistance of power rails. In addition to the resistance benefits, standard-cell height (area) can be further reduced with deep and narrow widths of power and ground pins. Fig. 5(b) and (d), respectively, show cross section views of FSPDN with BPR ( $P_{FB}$ ) and BSPDN with BPR ( $P_{BB}$ ) options. In the case of  $P_{FB}$ , connections between FSPDN and BPR are made through nano-TSVs with the same 90-nm width as in the  $P_{BS}$  option (but, with 1:7 aspect ratio). These nano-TSVs also necessitate the insertion of reserved spaces.

### C. Power Tap Cell Insertion

Although the use of BSPDN and BPR can reduce area and mitigate IR drop problems, connecting frontside layers to BSPDN and/or BPR remains a critical challenge. To establish “tap” connections from frontside metals to BPR, or from backside to frontside metals, space must be reserved on device layers—e.g., [17] proposes power tap cells for the connection between BPR to MINT (M0) layers. More frequent “taps” will mitigate IR drop problems, but occupy more placement area. In PROBE3.0, we define two types of power tap cells for the  $P_{FB}$  and  $P_{BS}$  options. Tap cells for  $P_{FB}$  connect BPR to M1, and tap cells for  $P_{BS}$  connect BM1 to M0. In contrast,  $P_{FS}$  and  $P_{BB}$  do not require power tap cells.

**Power Tap Cell Structure:** Fig. 6(a) shows a structure of power tap cells for  $P_{FB}$ . Double-height power tap cells for  $P_{FB}$  have 2CPP cell width. The connection between BPR and M0 is through a  $1 \times 2$  via array, and the two M1 metals are aligned with M1 vertical routing tracks. There are also two types of power tap cells for  $P_{FB}$  according to starting power and ground pins: power/ground pins on the double-height power tap cells are ordered as Power–Ground–Power (VDD–VSS–VDD) or Ground–Power–Ground (VSS–VDD–VSS). On the other hand, Fig. 6(b) shows a structure of power tap cells for  $P_{BS}$ . While power tap cells for  $P_{FB}$  have 2CPP width, double-height power tap cells for  $P_{BS}$  have 6CPP width due to the  $\sim 90$  nm width of nano-TSVs [20]. We also assume a 50-nm keepout spacing around nano-TSVs. Similar to power tap cells for  $P_{FB}$ , there are two types of double-height power tap cells for  $P_{FB}$ , Power–Ground–Power and Ground–Power–Ground.

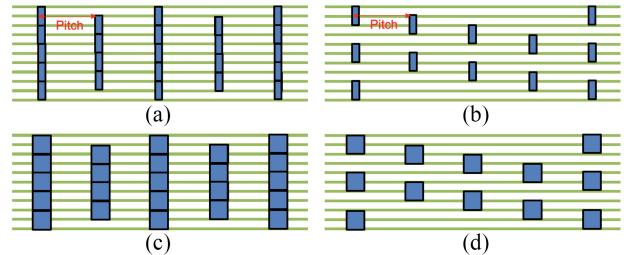


Fig. 7. Four power tap cell insertion results for: (a)  $P_{FB}$  (2CPP width) with *Column*; (b)  $P_{FB}$  with *Staggered*; (c)  $P_{BS}$  (6CPP width) with *Column*; and (d)  $P_{BS}$  with *Staggered*.

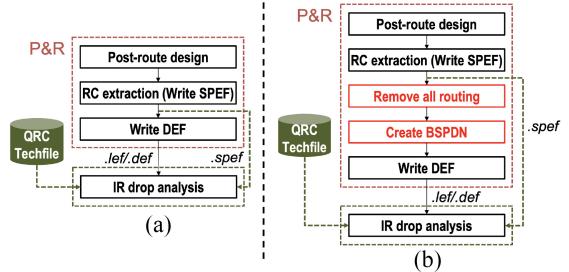


Fig. 8. IR drop analysis flow for (a) FSPDN and (b) BSPDN. For the IR drop flow for BSPDN, we delete all the signal and clock routing after P&R and build power stripes for BSPDN.

**Power Tap Cell Insertion Scheme:** Power tap cell insertion affects routability and IR drop, and hence affects PPAC of designs. In this work, we define five tap cell insertion pitches ( $I_{pitch}$ : 24, 32, 48, 96, and 128CPP) and two power tap insertion schemes ( $I_{scheme}$ : *Column* and *Staggered*).  $I_{pitch}$  and  $I_{scheme}$  denote tap cell insertion pitch and tap cell insertion scheme, respectively. Tap cell insertion scheme *Column* places double-height power tap cells on every two placement rows with the given tap cell pitch. Conversely, tap cell insertion scheme *Staggered* places double-height power tap cells on every four placement rows with the given tap cell pitch. Fig. 7 shows four power tap cell insertion results for  $P_{FB}$  and  $P_{BS}$  with *Column* and *Staggered* insertion schemes.

### D. IR Drop Analysis Flow

We develop two IR drop analysis flows for FSPDN and BSPDN. Fig. 8(a) presents our IR drop analysis flow for FSPDN. After P&R, we generate DEF and SPEF files for routed designs using a commercial P&R tool to perform standalone vectorless dynamic IR drop analysis. Additionally, an interconnect technology file (QRC techfile) is needed for RC extraction as input for the IR drop analysis flow. In contrast, Fig. 8(b) depicts our IR drop analysis flow for BSPDN. After P&R, we only create an SPEF file from routed designs. We then remove all routed signals and clocks from the P&R database and construct new power stripes for BSPDN. Since the standalone IR drop analysis tool obtains power stripe information from a DEF file, we generate a DEF file after creating power stripes on the backside. There are two backside metal layers, BM1 and BM2. When creating PDN on backside metal layers, we consider M1 as BM1 and M2 as BM2, respectively. For RC extraction with BSPDN, the QRC techfile must be scaled for backside metals since we assume BM1 and BM2 have the same pitches as M12 and M13. Full details are

TABLE III  
DEFINITION OF TOPOLOGICAL PARAMETERS IN ANG [13], [29]

Parameter	Definition
$N_{inst}(T_1)$	#instances.
$N_{prim}(T_2)$	#primary inputs/outputs.
$D_{avg}(T_3)$	Average net degree (average #terminals of a net).
$B_{avg}(T_4)$	Average size of net bounding box.
$T_{avg}(T_5)$	Average logic depth of timing paths.
$S_{ratio}(T_6)$	Ratio of #sequential cells to the total #cells.

visible in open-source scripts at [44] (specifically, the Design Enablement folder).

## V. ENHANCED ARTIFICIAL DESIGNS FOR PPAC EXPLORATION

Using specific real designs in DTCO and PPAC exploration can introduce biases and faulty decisions in technology configurations (e.g., cell architecture or BEOL stack). To avoid such biases, PROBE1.0 [10] bases its routability assessment on a mesh-like netlist topology, and PROBE2.0 [4] similarly uses a knight’s tour-based topology. However, these artificial topologies have two main limitations when we consider the “PP” aspects of PPAC. First, they are highly regular and cannot capture a wide range of circuit types. Second, they do not mimic the timing and power properties of real netlists, as they target routability assessment without regard to timing path structure. PROBE3.0 overcomes these limitations by generating artificial but realistic netlists with the artificial netlist generator (ANG) of [13] and [29], for use in PPAC studies. We use the six topological parameters of ANG (see Table III) to generate and explore circuits with various sizes, interconnect complexity, routed wirelengths and timing. Moreover, we apply ML (AutoML) to improve the match of generated artificial netlists to targeted (real) netlists.

### A. Comparison of ANG and Real Designs

We study four real designs from OpenCores [37] and the corresponding artificial netlists generated by ANG [13]. Each design is taken through commercial logic synthesis and P&R tools [40], [41] in the PROBE3.0 technology, to obtain a final-routed layout. For AES, JPEG, LDPC, and VGA, we, respectively, use target clock periods of 0.2, 0.2, 0.6, and 0.2 ns, and utilizations of 0.7, 0.7, 0.2, and 0.7. We then extract the six topological parameters from the routed designs and use these parameters to generate artificial netlists with ANG.

We introduce a *Score* metric to quantify similarity between artificial and real netlists, as defined in

$$\text{Score} = \prod_{i=1}^N \max\left(\frac{T_i^{\text{target}}}{T_i^{\text{out}}}, \frac{T_i^{\text{out}}}{T_i^{\text{target}}}\right) \quad (1)$$

where

$T_i^{\text{target}}$  =  $T_i$  in target parameter set;

$T_i^{\text{out}}$  =  $T_i$  of output parameter set;

$N$  = number of parameters ( $N = 6$ ).

In (1), target and output parameters are elements  $T_i^{\text{target}}$  and  $T_i^{\text{out}}$  of the target and output parameter sets. For each parameter, we calculate the discrepancy (ratio) between target and output values. The *Score* value is the product of these ratios. Ideally, if output parameters are exactly the same as target parameters, *Score* is 1. Larger values of *Score* indicate

TABLE IV  
TOPOLOGICAL PARAMETERS FOR REAL NETLISTS FROM OPENCORES [37] AND ARTIFICIAL NETLISTS GENERATED BY [13] (DENOTED BY \*)

Design	Parameters						Score
	$N_{inst}$	$N_{prim}$	$D_{avg}$	$B_{avg}$	$T_{avg}$	$S_{ratio}$	
AES	12318	394	3.28	0.55	7.98	0.04	-
JPEG	70031	47	3.09	0.21	10.36	0.07	-
LDPC	77379	4102	2.85	1.00	12.94	0.03	-
VGA	60921	185	3.71	0.42	8.25	0.28	-
AES*	10371	394	3.28	0.79	5.19	0.13	8.53
JPEG*	63185	47	3.16	0.70	6.97	0.15	12.03
LDPC*	58699	4106	3.10	0.78	6.96	0.13	14.8
VGA*	64412	188	3.32	0.26	6.39	0.25	2.8

TABLE V  
PARAMETER SETS FOR TRAINING AND TESTING. TESTING IS PERFORMED IN THE RANGES AROUND GIVEN TARGET PARAMETERS, ACCORDING TO THE STEP SIZES

Parameter	Training Value	Testing Value	
		Range	Step
$N_{inst}(T_1^{in})$	10000, 20000, 40000, 80000	$T_1^{\text{target}} \pm 500$	100
$N_{prim}(T_2^{in})$	100, 200, 500, 1000, 2000, 4000	$T_2^{\text{target}} \pm 5$	1
$D_{avg}(T_3^{in})$	1.8, 2.0, 2.2, 2.4, 2.6	$T_3^{\text{target}} \pm 0.2$	0.02
$B_{avg}(T_4^{in})$	0.70, 0.75, 0.80, 0.85, 0.90, 0.95	$T_4^{\text{target}} \pm 0.2$	0.02
$T_{avg}(T_5^{in})$	6, 8, 10, 12, 14, 16	$T_5^{\text{target}} \pm 10$	2
$S_{ratio}(T_6^{in})$	0.2, 0.4, 0.6, 0.8, 1.0	$T_6^{\text{target}} \pm 0.2$	0.02

a greater discrepancy between ANG-generated netlists and the target netlists.

Table IV shows the input parameters, extracted parameters, and *Score* metric in our comparison of real and artificial designs. The causes of discrepancy are complex, e.g., [13] has steps that heuristically adjust average depths of timing paths  $T_{avg}$  and the ratio of sequential cells  $S_{ratio}$ . Also, performing P&R will change the number of instances  $N_{inst}$ , the average net degree  $D_{avg}$ , and the routing which determines  $B_{avg}$ . Hence, it is difficult to identify the input parameterization of ANG that will yield artificial netlists whose post-route properties match those of (target) real netlists. We use ML to address this challenge.

### B. Machine Learning-Based ANG Parameter Tuning

We improve the realism of generated artificial netlists with ML-based parameter tuning for ANG. Fig. 9(a) shows the training flow in the parameter tuning. First, to generate training data, we sweep the six ANG input parameters to generate 21 600 combinations of input parameters, as described in Table V. Second, we use ANG with these input parameter combinations to generate artificial gate-level netlists. Third, we perform P&R with the (21 600) artificial netlists and extract the output parameters. The extracted output parameters are used as output labels for the ML model training. We use the open-source H2O AutoML package [35] (version 3.30.0.6) to predict the output parameters; the *StackedEnsemble\_AllModels* model consistently returns the best model.<sup>2</sup> Note that the trained ML model is at most weakly (via design enablement and SP&R tooling) technology-dependent, since it fits six *topological* parameters that are not directly related to technology information. The model training is a one-time overhead which took 4 h using an Intel Xeon Gold 6148 2.40-GHz

<sup>2</sup>The H2O AutoML package [35] includes an automatic parameter tuning process and recommends the best model based on the training data.

TABLE VI

COMPARISON OF ARTIFICIAL NETLISTS GENERATED BY ANG (DENOTED BY \*) AND ARTIFICIAL NETLISTS GENERATED WITH OUR PARAMETER TUNING FLOW (DENOTED BY \*\*)

Design	Similarity measures			
	Hamming	Centrality	Spectral	Reconvergence
AES*	0.00072	0.9920	0.9804	0.7356
JPEG*	0.00023	0.9940	0.9722	0.7762
LDPC*	0.00010	0.9916	0.9986	0.7787
VGA*	0.00016	0.9924	0.9792	0.8121
AES**	<b>0.00069</b>	<b>0.9936</b>	<b>0.9822</b>	<b>0.7478</b>
JPEG**	<b>0.00018</b>	<b>0.9932</b>	<b>0.9743</b>	<b>0.7871</b>
LDPC**	<b>0.00010</b>	<b>0.9938</b>	<b>0.9995</b>	<b>0.7952</b>
VGA**	<b>0.00015</b>	<b>0.9939</b>	<b>0.9822</b>	<b>0.8126</b>

server. Executing P&R required just over seven days in our academic lab setting, and is again a one-time overhead.<sup>3</sup>

Fig. 9(b) shows our inference flow. First, we define ranges around the target parameter and sweep the parameters to generate multiple combinations of input parameters as candidates, which are shown in Table V. Second, we use our trained model to predict the output parameters from each input parameter combination. Note that although there are 12.3M combinations as specified in the rightmost two columns of Table V, this step requires less than 10 min on an Intel Xeon Gold 6148 2.40-GHz server.<sup>4</sup> Third, we calculate a predicted *Score* per each input parameter combination, and then choose the parameter combination with the lowest predicted *Score*. Finally, we use ANG and the chosen parameter combination to generate an artificial netlist for PPAC explorations.

Table VI presents a comparison between the artificial and the real netlists using hypergraph similarity methods [26]: 1) Hamming distance; 2) closeness centrality; and 3) spectral distance. In addition, we use *reconvergence* from [25] to further compare structural similarities between the netlists. Scores near 0 for Hamming distance and near 1 for other metrics suggest high similarity. Table VI shows that the netlists from ANG and those generated with our parameter tuning flow show high similarity to the original netlists, with the latter having stronger similarity. Table VII shows the further benefit from ML-based ANG parameter tuning. Columns 2–5 show parameters from real netlists, which we use as target parameters. The trained ML model and the inference flow produce the tuned parameters for ANG shown in Columns 6–9, and corresponding results are shown in Columns 10–13. The average *Score* decreases to 4.89 from the original value of 8.87 for ANG without parameter tuning (Table IV).

The ML-enabled improvement of realism in ANG netlists can be seen using t-SNE visualization [16] from P&R results. We perform P&R for the four real designs by sweeping initial utilization from 0.6 to 0.8 with a 0.01 step size, and a target clock period from 0.15 to 0.25 ns with a 0.01-ns step size; this results in  $21 \times 11 = 231$  P&R runs. (For LDPC, we sweep utilization from 0.1 to 0.3 with a 0.01 step size, and

<sup>3</sup>The average P&R runtime on our 21 600 ANG netlists is 0.4 h on an Intel Xeon Gold 6148 2.40-GHz server. The data generation used 50 concurrently running licenses of the P&R tool, with each job running single-threaded.  $(21\,600 \times 0.4 / 50 / 24 \approx 7.2$  days. With multithreaded runs, we estimate that data generation would have taken 3–4 days.)

<sup>4</sup> $11 \times 11 \times 21 \times 21 \times 11 \times 21 = 12\,326\,391$ . We apply simple filtering based on lower and upper bounds, to avoid parameter values for which ANG does not work properly. Specifically, parameter values are restricted to be within:  $0 < B_{avg} \leq 1.0$ ;  $0 < S_{ratio} \leq 1.0$ ;  $1 < D_{avg} \leq 2.6$ ; and  $3 < T_{avg}$ . For example, the AES testcase then has  $\sim 3$ M input parameter combinations, and predicting output parameters for all of these takes 441 s of runtime.

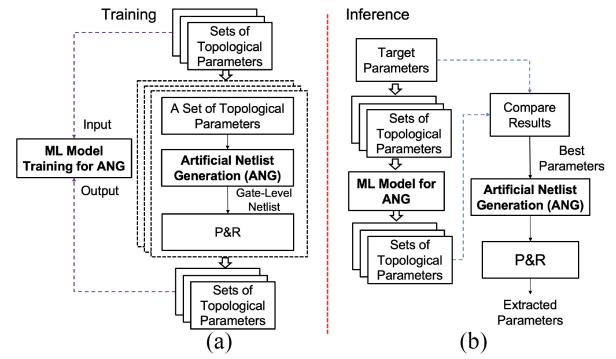


Fig. 9. ML-based parameter tuning for ANG: (a) Training and (b) inference flows. In that AutoML [35] returns the best model it finds based on given training data, the ML model in this flow would change with training data.

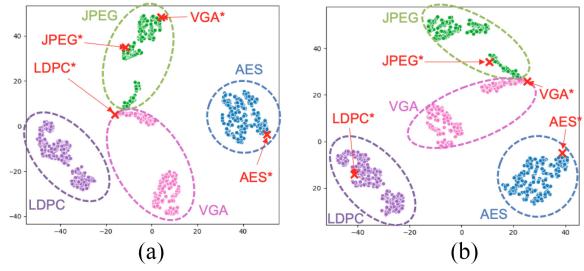


Fig. 10. Comparison between real and artificial designs by t-SNE [16]. (a) t-SNE visualization for real and artificial (ANG) designs **without** our parameter tuning flow. (b) Real and artificial designs **with** our ML-based parameter tuning flow. Design names followed by \* indicate artificial designs.

clock period from 0.55 to 0.65 ns with a 0.01-ns step size.) We then perform P&R for artificial netlists with and without our parameter tuning flow, with 0.7 utilization (0.2 for LDPC) and 0.2 ns (0.6-ns LDPC) target clock period. Fig. 10 shows t-SNE visualization<sup>5</sup> of the real and artificial designs. The 231 real datapoints per design form well-defined clusters. In Fig. 10(a), the datapoints of the artificial AES and JPEG designs are located in the corresponding designs’ clusters. However, the artificial LDPC and VGA designs are not close to the corresponding clusters of real designs. In contrast, Fig. 10(b) shows that with our ML-based ANG parameter tuning, datapoints of all four artificial designs are located within the corresponding clusters of real designs. This suggests that the ML-based ANG parameter tuning helps create artificial netlists that better match targeted design parameters—including parameters that are relevant to PPAC exploration.

## VI. IMPROVED ROUTABILITY ASSESSMENT

Recall that in the PROBE approach, routability (“AC”) is evaluated using the  $K$ -threshold ( $K_{th}$ ) metric [10]. That is, given a placed netlist, routing difficulty is gradually increased by iteratively swapping random pairs of neighboring instances. The cell-swaps progressively “tangle” the placement until it becomes unrouteable ( $> 500$  DRCs post-detailed routing). The number of swaps  $K$ —expressed as a multiple of the instance count—at which routing fails is the  $K_{th}$  metric. Larger  $K_{th}$  implies greater routing capacity or intrinsic routability.

<sup>5</sup>For t-SNE visualization, we collect 11 features from P&R results: #instances, #nets, #primary I/O pins, average fanout, #sequential cells, wirelength, area, #DRCs, WNS, TNS, and #failing endpoints.

TABLE VII

TOPOLOGICAL PARAMETERS FOR TARGET, INPUT, AND OUTPUT NETLISTS. THE DESIGN NAMES FOLLOWED BY \*\* INDICATE ANG-GENERATED ARTIFICIAL NETLISTS WITH ML-BASED ANG PARAMETER TUNING

Parameter	Parameters of Target Netlists				ANG Input Parameters (ML Inference)				Parameters from Artificial Netlists			
	AES	JPEG	LDPC	VGA	AES	JPEG	LDPC	VGA	AES**	JPEG**	LDPC**	VGA**
$N_{inst}$	12318	70031	77379	60921	12718	69531	76979	60421	10200	64296	64796	65113
$N_{prim}$	394	47	4102	185	390	42	4106	199	394	46	4110	202
$D_{avg}$	3.28	3.09	2.85	3.71	3.40	3.10	3.03	3.53	3.26	3.13	3.18	3.30
$B_{avg}$	0.55	0.21	1.00	0.42	0.49	0.31	1.98	0.28	0.72	0.21	0.73	0.36
$T_{avg}$	7.98	10.36	12.94	8.25	13.98	18.36	20.94	12.25	8.01	9.29	11.64	8.54
$S_{ratio}$	0.04	0.07	0.03	0.28	0.01	0.27	0.01	0.16	0.11	0.20	0.13	0.16
$Score$	-	-	-	-	-	-	-	-	4.39	3.59	2.77	8.81

Both PROBE1.0 [10] and PROBE2.0 [4] enable the study of real netlists through *cell width-regularized* placements. In this approach, combinational cells are inflated (by LEF modification) to the maximum combinational cell width in the library [a process termed as *cell width-regularization (CWR)*] to prevent cell overlaps during neighbor-swaps in the  $K_{th}$  evaluation. While this approach prevents illegal placements (i.e., cell overlaps due to varying widths), it often results in low utilizations that harm the realism of the study. (Moreover, high whitespace leads to high  $K_{th}$  values that require more P&R runs to determine.) We now describe a *clustering-based CWR* methodology that reduces whitespaces and generates placements with realistic utilization, based on real designs.

#### A. Clustering-Based CWR

We propose a *clustering-based CWR* using bottom-up hypergraph clustering, as detailed in Algorithm 1. The key intuition is to allow FC to 1) find clusters that optimize vertex connectivity (i.e., clustering vertices that are strongly connected) so as to not significantly degrade the wirelength during routability assessment and 2) generate clusters that are width-regularized, so as to minimize whitespaces and achieve realistic placement utilizations. In the following, we refer to standard cells of the original netlist as  $\text{cells}_{\text{orig}}$ , and (clustered) cells of the clustered netlist as  $\text{cells}_{\text{clustered}}$ .

*Clustered Hypergraph Creation:* For a given design, we first obtain a netlist hypergraph using OpenDB [38]. We perform *CWR clustering*, where cells  $\text{cells}_{\text{orig}}$  (vertices) in the original netlist hypergraph are clustered such that *clustered cell width*<sup>6</sup> does not exceed  $w_{\max}$ , the maximum cell width in the netlist. The inputs to *CWR* clustering are 1) a hypergraph  $H(V, E, W)$  with vertices  $V$ , hyperedges  $E$ , and cell widths  $W$ ; 2)  $w_{\max}$ ; and 3) the number of clustering iterations,  $N_{\text{iter}}$ .<sup>7</sup> The output is a clustered hypergraph ( $H_{\text{out}}$ ). We adapt First-Choice (FC) clustering that is used in state-of-the-art partitioners [11], [24]. We modify the cluster score to accommodate width regularization as a clustering objective, and refer to our clustering method as *cell width-regularized clustering with FC (CWR-FC)*.

*CWR-FC* first sorts vertices in increasing order of cell widths (line 4) and initializes *cluster assignments*  $cmap$  (line 6).  $cmap$  is a mapping of vertices to clusters, i.e.,  $V_k$  to  $V_{k+1}$ . Next, vertices are traversed to perform pairwise clustering; note that only combinational cells are considered

<sup>6</sup>Given vertex set  $V$  with cell widths  $W$ , clustering vertices  $v_i, v_j \in V$  yields a clustered cell with width  $W[v_i] + W[v_j]$ .

<sup>7</sup>In our experiments, we set  $N_{\text{iter}} = 20$ . However, the cell width-regularized clustering is strongly constrained by  $w_{\max}$ , and we observe on our testcases that clustering stops after  $\sim 3$  iterations.

#### Algorithm 1 CWR by Clustering

---

Inputs: Hypergraph  $H(V, E, W)$ , Maximum cell width  $w_{\max}$ , #iterations  $N_{\text{iter}}$   
Outputs: Clustered hypergraph  $H_{\text{out}}(V_{\text{out}}, E_{\text{out}}, W_{\text{out}})$

```

1:  $N_{\text{cluster}} \leftarrow |V|$ 
2: Hypergraph at iteration 0,  $H_0(V_0, E_0, W_0) \leftarrow H(V, E, W)$ 
3: for  $k \leftarrow 0$ ;  $k < N_{\text{iter}}$ ;  $k + +$  do
4:    $V_{\text{ordered}} \leftarrow$  Sorted  $V_k$  in increasing order of  $W_k$ 
5:    $\text{visited}[v] \leftarrow \text{false} \forall v \in V_k$ 
6:   Cluster assignments,  $cmap[v] \leftarrow v \forall v \in V_k$ 
7:   Clustered cell widths,  $W_{k+1} \leftarrow W_k$ 
8:   for  $v_i \in V_{\text{ordered}}$  do
9:     if  $\text{visited}[v_i] == \text{true}$  or  $v_i$  is a sequential cell then
10:      continue
11:    end if
12:     $V_{\text{neighbor}} \leftarrow$  Find adjacent vertices of  $v_i$ 
13:    Best cluster score,  $\phi_{\text{best}} \leftarrow 0$ ; Best cluster candidate,  $v_{\text{best}} \leftarrow -1$ 
14:    for  $v_j$  in  $V_{\text{neighbor}}$  do
15:      if  $W_k[v_i] + W_{k+1}[cmap[v_j]] \leq w_{\max}$  then
16:         $\phi(v_i, v_j) \leftarrow \frac{\sum_{v_i \in e, v_j \in e} \text{weight}_e}{|e|-1}$  // Cluster Score
17:        if  $\phi(v_i, v_j) > \phi_{\text{best}}$  then  $v_{\text{best}} \leftarrow v_j$ 
18:        end if
19:      end if
20:    end for
21:    if  $v_{\text{best}} == -1$  then
22:       $W_{k+1}[v_i] \leftarrow W_k[v_i]$ 
23:       $\text{visited}[v_i] \leftarrow \text{true}$ 
24:    else
25:       $cmap[v_i] \leftarrow cmap[v_{\text{best}}]$ 
26:       $W_{k+1}[v_{\text{best}}] \leftarrow W_k[v_i] + W_{k+1}[cmap[v_{\text{best}}]]$ 
27:       $\text{visited}[v_i] \leftarrow \text{true}; \text{visited}[v_{\text{best}}] \leftarrow \text{true}$ 
28:       $N_{\text{cluster}} \leftarrow N_{\text{cluster}} - 1$ 
29:    end if
30:  end for
31:  if  $N_{\text{cluster}} == |V_{k-1}|$  then
32:    break
33:  else
34:     $H_{k+1}(V_{k+1}, E_{k+1}, W_{k+1}) \leftarrow$  Build clustered hypergraph using  $cmap$ 
35:  end if
36: end for
37:  $H_c \leftarrow$  Clustered hypergraph generated at last iteration
38:  $H_{\text{out}} \leftarrow$  Best-fit bin packing on  $H_c$ 
39: Return  $H_{\text{out}}$ 

```

---

(line 8). For each vertex  $v_i$ , we find its neighbors  $v_j$  (line 11); each  $v_j$  is considered only if  $w_{\max}$  is not violated (line 14). A cluster score  $\phi(v_i, v_j)$  is calculated in line 15, where  $\text{weight}_e$  is the weight of hyperedge  $e$ , and  $W_k[v_i]$  is the width of  $v_i$ . In all our experiments, the hypergraphs have unit vertex and hyperedge weights. The vertex with the highest score is picked for clustering (lines 21–24). Finally, we construct the clustered hypergraph and proceed with subsequent iterations (line 28).

Note that *CWR-FC* strictly picks adjacent pairs of vertices for clustering. If all pairings exceed  $w_{\max}$ , the algorithm can stall (line 25). To address this and improve the uniformity of cluster contents, we perform best-fit bin-packing [8] with bins having capacity  $w_{\max}$  (line 30).<sup>8</sup> Finally, the output is the clustered hypergraph  $H_{\text{out}}$ .

<sup>8</sup>We choose best-fit for its simplicity and intuitiveness. Best-fit also enjoys a better approximation ratio compared to first-fit or next-fit alternatives [8].

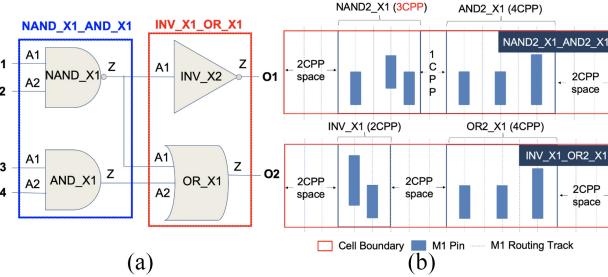


Fig. 11. Two example clustered cells NAND\_X1\_AND\_X1, and INV\_X1\_OR\_X1 in clustering-based CWR. (a) Schematic view, and (b) physical layout view assuming Lib2 with  $w_{\max} = 12\text{CPP}$ .

**Clustered Netlist Creation:** We convert the clustered hypergraph  $H_{\text{out}}$  into Verilog using OpenDB. To run P&R we require a new LEF file that captures the clustered netlist, i.e., we require a new netlist over the clusters,  $\text{cells}_{\text{clustered}}$ . Fig. 11(a) provides a schematic view of two clustered cells, NAND\_X1\_AND\_X1 and INV\_X1\_OR\_X1. These correspond to two clusters of original cells: NAND\_X1 and AND\_X1, and INV\_X1 and OR\_X1. Fig. 11(b) presents the physical layout of the clusters. In this case, a noninteger gear ratio between M1P (30 nm) and CPP (45 nm) forces cells in  $\text{cells}_{\text{clustered}}$  to be positioned at even CPP sites, to avoid M1 pin misalignment. In the first cluster, NAND\_X1 width (3CPP) is an odd number of CPPs, necessitating the addition of 1CPP padding between the two cells. In the second cluster, the total cell width is less than  $w_{\max}$ , so whitespace is added. We distribute whitespace uniformly, 1) at the sides of  $\text{cells}_{\text{clustered}}$  and 2) between consecutive cells in each cluster, as illustrated in Fig. 11(b). Note that we first allocate whitespace at junctions (between consecutive original cells) where no extra padding was previously allocated.

### B. Performance of CWR Clustering

We now discuss the advantages of our proposed CWR clustering in subsequent sections.

**Comparison to PROBE2.0:** Fig. 12 compares cell width distributions for instances in the clustered netlist and instances in the original netlist. The blue lines show the distribution of cell widths in the original netlist, where smaller cell widths predominate. The red lines indicate that CWR-FC increases the prevalence of cells with larger widths through the creation of the merged cells<sub>clustered</sub>. The maximum cell width  $w_{\max}$  is netlist-dependent. The blue lines show the distribution of cell widths in the original netlist, where smaller cell widths predominate. The red lines indicate that CWR-FC increases the prevalence of cells with larger widths through the creation of the merged cells<sub>clustered</sub>. We also see that for all four testcases in Fig. 12 no cells in cells<sub>clustered</sub> have width greater than  $w_{\max}$ . The larger cell widths in cells<sub>clustered</sub> lead to smaller amounts of added whitespace needed to regularize cell widths.

As anticipated, CWR clustering significantly reduces whitespace in the placed designs. With Lib2 and FSPDN for P&R, placing cell width-regularized instances used in PROBE2.0 at 90% density achieves actual utilizations of 0.21, 0.21, and 0.40 for AES, JPEG, and VGA, respectively. For LDPC, placing cell width-regularized instances at 30% density achieves actual utilization of 0.08. In contrast, CWR-FC achieves actual

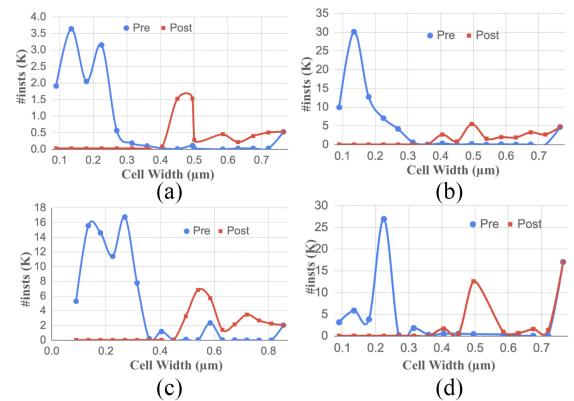


Fig. 12. Cell width distributions preclustering (i.e., original netlist) and post-clustering (i.e., by CWR-FC) for (a) AES, (b) JPEG, (c) LDPC, and (d) VGA.

TABLE VIII  
COMPARISON OF THE CWR CLUSTERED NETLIST PRODUCED BY CWR-FC ([C]) WITH THE ORIGINAL FLAT NETLIST ([A]) AND A CWR CLUSTERED NETLIST INDUCED FROM A PLACEMENT OF THE FLAT NETLIST ([B])

Stage	Design	#Insts	Area ( $\mu\text{m}^2$ )	Util	WL ( $\mu\text{m}$ )	Avg. FO
[A]	AES	12318	426.254	0.83	30849	2.32
	JPEG	70031	2781.981	0.73	112605	2.15
	LDPC	77379	6250.563	0.43	567630	1.85
	VGA	60921	4238.205	0.76	208845	2.71
[B]	AES	4275	426.254	0.83	32632	1.96
	JPEG	23281	2781.981	0.73	111241	1.86
	LDPC	42383	6250.563	0.43	585923	1.43
	VGA	40084	4238.205	0.76	189612	2.14
[C]	AES	4661	426.254	0.83	32679	2.08
	JPEG	25961	2781.981	0.73	143693	1.76
	LDPC	30636	6250.563	0.43	637417	1.29
	VGA	32768	4238.205	0.76	220915	2.02

utilizations of 0.71, 0.74, 0.71, and 0.23 for AES, JPEG, VGA, and LDPC respectively.

**Topological and Wirelength Comparisons to Real Designs:** Table VIII compares characteristics of our clustering-based cell width-regularized netlists and placements ([C]), versus analogous characteristics of real netlists and placements ([A]). We also implement another plausible clustering methodology, which is to induce clusters from a placement of the original design ([B]). In [B], clusters from the placement are induced by 1) traversing combinational cells left-to-right in each standard cell row and 2) clustering maximal contiguous sets of cells without exceeding  $w_{\max}$ .

We run P&R using Lib2 and  $P_{\text{FS}}$  for PDN, maintaining the same core area and utilizations.<sup>9</sup> Clustering decreases the number of instances and average fanouts for [B] and [C], relative to [A]. However, wirelengths exhibit no significant changes. The similarities between [A], [B], and [C] suggest that our CWR-FC methodology can preserve netlist properties relevant to P&R outcomes, with more realistic utilizations.

## VII. EXPERIMENTAL SETUP AND RESULTS

We have extensively studied the design-technology pathfinding capability of the PROBE3.0 framework using the PROBE3.0 technology. In this section, we report three main experiments. Expts 1 and 2 show PROBE3.0's capability to

<sup>9</sup>Use of other libraries and PDN options does not shed further light on comparisons between original and generated netlists since they lead to only marginal changes in the wirelength and area. We thus omit them and only use Lib2 and  $P_{\text{FS}}$  options for P&R.

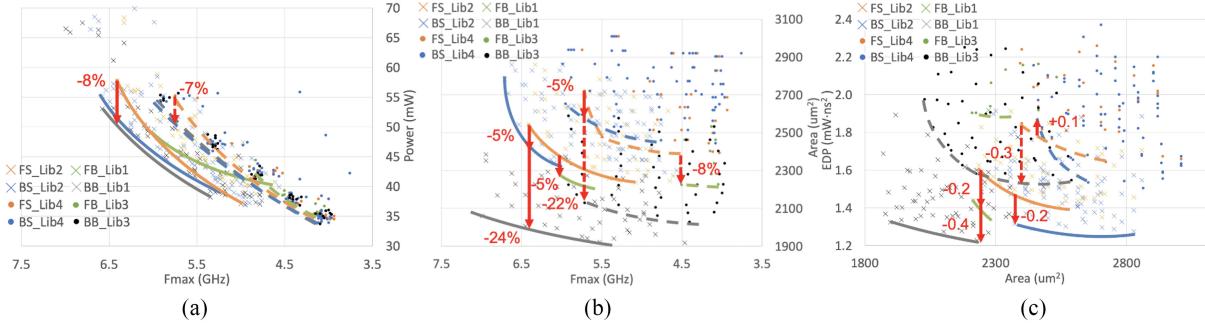


Fig. 13. PPAC tradeoffs for JPEG with four libraries (*Lib1–Lib4*): (a) *Performance–Power*, (b) *Performance–Area*, and (c) *EDP–Area*. We compare four PDNs in terms of performance and area and measure improvements relative to traditional FSPDN (P<sub>FS</sub>), to show the benefits of BSPDN and BPR.

assess PPAC trends and tradeoffs, using real and artificial designs, respectively. Expt 3 performs assessments of routability and achievable utilization.

In Expts 1 and 2, we analyze four tradeoffs.

- 1) We present *Performance–Power* plots that quantify tradeoffs between performance (maximum frequency) and power.
- 2) We present *Performance–Area* plots to quantify the tradeoffs between performance and area.
- 3) To address PP aspects, we use the energy–delay product (EDP) [14] as a single metric for PP. *EDP–Area* plots depict tradeoffs between performance/power and area.
- 4) We present *IR drop–Area* plots to demonstrate tradeoffs between IR drop and area. We also compare results obtained using artificial designs with those obtained using real designs. Expt 3 assesses routability and achievable utilization using our clustering-based cell width-regularized placements. We perform experiments on four designs (AES, JPEG, VGA, and LDPC); results are similar for all designs except LDPC.<sup>10</sup> Owing to space limitations, we only present results for JPEG since its size is between those of AES and VGA. Results of the other three designs are available in [44].

#### A. Experimental Setup

Based on the definition of technology and design parameters in [4], we define ten technology parameters and eight design parameters as the input parameters for the PROBE3.0 framework. Table X describes the definitions of these parameters and the options used in our experiments. Also, we use commercial tools for PDK generation, logic synthesis, P&R, and IR drop analysis. We use open-source tools for GDT-to-GDS translation [34] and SMT solver [45]. Tools and versions used in our experiments are summarized in [44].

*Criteria for Valid Result:* In our experiments, for given *Design*, *PDN*, and technology parameters, we perform logic synthesis, P&R, and IR drop analysis with multiple sets of parameters, including  $I_{\text{pitch}}$ ,  $I_{\text{scheme}}$ ,  $Util$ , and  $Clkp$ . We use 24, 32, 48, 96, and 128 *CPP* for  $I_{\text{pitch}}$ , and *Column* and *Staggered* for  $I_{\text{scheme}}$ . For  $Util$ , we use values ranging from 0.70 to 0.94 with a step size of 0.02, and for  $Clkp$ , we use values ranging from 0.12 to 0.24 ns with a step size of 0.02 ns. After the

<sup>10</sup>LDPC is a routing-dominant design, which magnifies the observed area gap (i.e., across PDN options) compared to the other three designs.

TABLE IX  
STANDARD CELL PARAMETER SETTINGS

Lib#	Fin	Route Track	Power-Ground Pin	Cell Height
Lib1	2Fin	4RT	BPR	5T
Lib2			M0	6T
Lib3	3Fin	5RT	BPR	6T
Lib4			M0	7T

implementation and the analysis steps, we filter out results that are likely to fail signoff criteria even if followed up with additional human engineering efforts.

To be precise, a “valid” result must satisfy three conditions: 1) the worst negative slack is larger than  $-50$  ps; 2) the number of post-route DRCs is less than 500; and 3) the 99.7 percentile of the effective instance voltage (EIV) is greater than 80% of the operating voltage ( $V_{\text{op}}$ ). To assess 3), we use an IR drop analysis tool [32] to measure vectorless dynamic IR drop, and calculate the EIV as  $V_{\text{op}} - V_{\text{drop}}$  per each instance, where  $V_{\text{op}}$  is an operating voltage (0.7 V) and  $V_{\text{drop}}$  is the worst voltage drop per instance. We take the 99.7 percentile of EIV as representative of IR drop after P&R, as it is within three standard deviations from the mean per the empirical rule [27].

#### B. Expt 1 (PPAC Exploration With Real Designs)

*Performance Versus Power:* We first present PPAC explorations that show tradeoffs between performance and power. (We assume that area is proportional to cost, since chip area is closely related to cost.) In this study, we show results for JPEG with four standard-cell libraries. Table IX gives standard-cell parameter settings for *Lib1–Lib4*. 2Fin-6T is selected to represent low-power libraries, and 3Fin-7T for high-performance libraries, considering the latest technology settings. We do not select 1Fin due to reliability concerns. We furthermore use four PDN structures ( $P_{\text{FS}}$ ,  $P_{\text{FB}}$ ,  $P_{\text{BS}}$ , and  $P_{\text{BB}}$ ), and measure improvements due to scaling boosters relative to the traditional FSPDN ( $P_{\text{FS}}$ ).

Fig. 13(a) gives *Performance–Power* plots that show tradeoffs between performance and power for JPEG, and improvements from the traditional FSPDN. We calculate the maximum achievable frequency ( $f_{\text{max}}$ ) as  $1/(Clkp - WNS)$  where  $Clkp$  is the target clock period and  $WNS$  is the worst negative slack. Also, we add up leakage and dynamic power to obtain the total power. To measure the improvement from  $P_{\text{FS}}$ , we compare the second-largest value (on the  $x$ -axis) attained with each PDN configuration. From the result, we make two main observations: 1) power consumption with

TABLE X  
TECHNOLOGY AND DESIGN PARAMETERS IN OUR EXPERIMENTS

Type	Parameter	Description	Option
Technology	$F_{in}$	The number of fins for devices of standard cells.	2, 3
	$CPP$	Contacted poly pitch for standard cells in nm.	45
	$M0P$	$M0$ (horizontal) layer pitch in nm.	24
	$M1P$	$M1$ (vertical) layer pitch in nm.	30
	$M2P$	$M2$ (horizontal) layer pitch in nm.	24
	$RT$	The number of available $M0$ routing tracks in standard cells.	4, 5
	$PGpin$	Power/ground pin layer for standard cells.	$BPR, M0$
	$CH$	Cell height of standard cells, expressed as a multiple of $M0P$ . For example, when the cell height in nm is $120nm$ and $M0P$ is $24nm$ , the cell height ( $CH$ ) is 5. The cell height value is calculated as $RT + 2$ for $M0$ $PGpin$ and $RT + 1$ for $BPR$ $PGpin$ .	5, 6, 7
	$MPO$	The number of minimum pin openings (access points).	2
	$DR$	Design rules. We define the same grid-based design rules, minimum area rule ( $DR\text{-MAR}$ ), end-of-line spacing rule ( $DR\text{-EOL}$ ) and via spacing rule ( $DR\text{-VR}$ ) as [6]. We use the <i>EUV-tight</i> ( <i>ET</i> ) design rule set, which includes $DR\text{-MAR} = 1$ , $DR\text{-EOL} = 2$ and $DR\text{-VR} = 1$ .	<i>EUV-Tight</i>
Design	$BEOL$	Metal stack options. We define 14M metal option which contains 14 metal layers ( $M0$ to $M13$ ). We define 1.2X, 2.6X, 3.2X and 30X layer pitches based on 24nm as the 1X pitch.	14M
	$PDN$	Power delivery network options.	$P_{FS}, P_{FB}, P_{BS}, P_{BB}$
	$I_{pitch}$	Power tap cell pitch in CPP.	24, 32, 48, 96, 128
	$I_{scheme}$	Power tap cell insertion scheme.	<i>Column</i> , <i>Staggered</i>
	$Tool$	Commercial P&R tools.	Synopsys IC Compiler II
	$Util$	Initial placement utilization.	0.70 to 0.94 with a 0.02 step size
	$Design$	Designs studied in our experiments. We conduct experiments with four open-source designs from OpenCores [44] and artificial netlists generated by ANG with our ML-based parameter tuning.	AES, JPEG
	$Clkp$	Target clock periods that reflect maximum achievable frequencies for logic synthesis and P&R.	0.12 to 0.24ns with a 0.02ns step size

$P_{BS}$  and  $P_{BB}$  decreases by 7%–8% compared to  $P_{FS}$  and 2) power consumption with  $P_{FB}$  is similar to  $P_{FS}$ , with the same performance. We observe power reductions from use of scaling boosters, BSPDN and BPR. However, use of BPR without BSPDN does not reduce power consumption.

**Performance Versus Area:** Performance-area tradeoffs for JPEG are shown in Fig. 13(b). We make two main observations: 1) area with  $P_{FB}$ ,  $P_{BS}$ , and  $P_{BB}$  decreases by up to 8%, 5%, and 24%, respectively, as compared to  $P_{FS}$ , while maintaining the same level of performance and 2) the use of scaling boosters results in area reductions across all four standard-cell libraries. The area reduction results obtained using the PROBE3.0 framework are consistent with previous industry works [7], [18], [28], [30], which shows that the use of BSPDN and BPR can result in area reductions of 25%–30%.

**EDP Versus Area:** Given the tradeoffs among PPAC criteria, a simpler metric is useful to comprehend multiple aspects simultaneously. The EDP is adopted by, e.g., [14] as a single-value metric that captures both power efficiency and maximum achievable frequency (performance). EDP is calculated as  $P \times f_{max}^2$ , where  $P$  denotes power consumption and  $f_{max}$  denotes maximum achievable frequency. Lower EDP means more energy-efficient operations for the chip. We draw *EDP-Area* plots to show PPAC tradeoffs of various PDN structures. We again use four libraries (*Lib1*–*Lib4*).

From Fig. 13(c), we make four main observations.

- 1) For 4RT (*Lib1* and *Lib2*), EDP with  $P_{FB}$ ,  $P_{BS}$ , and  $P_{BB}$  decreases by 0.2, 0.2, and 0.4 mW · ns<sup>2</sup>, respectively, compared to  $P_{FS}$  with the same area.
- 2) For 5RT (*Lib3* and *Lib4*), EDP with  $P_{BB}$  decreases by 0.3 mW · ns<sup>2</sup>, compared to  $P_{FS}$  with the same area.
- 3) For 5RT, EDP with  $P_{FB}$  shows no improvements, and EDP with  $P_{BS}$  increases by 0.1 mW · ns<sup>2</sup>, as compared to  $P_{FS}$  with the same area.
- 4) Use of  $P_{BB}$  better optimizes area than other PDN structures with the same EDP.

**Supply Voltage (IR) Drop Versus Area:** With recent advanced technologies and designs, denser PDN structures are required due to the large resistance seen in tight-pitch

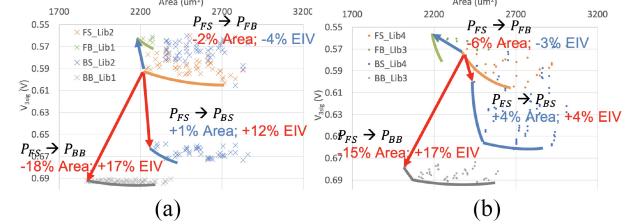


Fig. 14. *IR drop-Area* plots for JPEG with four libraries (*Lib1*–*Lib4*). (a) JPEG with 4RT (*Lib1*/*Lib2*). (b) JPEG with 5RT (*Lib3*/*Lib4*).

BEOL metal layers. The denser PDN structures bring added routability challenges which critically impact area density. In light of this, we measure IR drop and area from valid runs, and plot *IR drop-Area* tradeoffs in Fig. 14. In the plots, we compare the points with the minimum area for each PDN configuration in terms of area and 99.7 percentile (three-sigma) of EIV. Note that larger EIV means better IR drop mitigation. Fig. 14(a) and (b) shows *IR drop-Area* tradeoffs for JPEG with 4RT (*Lib1* and *Lib2*) and 5RT (*Lib3* and *Lib4*), respectively. We make four main observations.

- 1) Area with  $P_{FB}$  decreases by 2%–6% compared to  $P_{FS}$ , while the EIV increases by 3%–4%.
- 2) Area with  $P_{BS}$  increases by 1%–4% compared to  $P_{FS}$ , while EIV decreases by 4%–12%.
- 3) Area with  $P_{BB}$  decreases by 15%–18% compared to  $P_{FS}$ , while EIV decreases by 17%.
- 4) Backside PDN offers IR drop mitigation, but BPR ( $P_{FB}$ ) worsens it. Thus, more power tap cells are essential for IR drop mitigation, though the added area overhead might degrade the IR drop quality achieved by BPR.

### C. Expt 2 (PPAC Exploration With Artificial Design)

Expt 2 is similar to Expt 1 and uses the *artificial* JPEG design generated with our ML-based parameter tuning.

**Performance Versus Power:** Fig. 15(a) shows the tradeoffs between performance and power with the *artificial* JPEG design. We make three main observations.

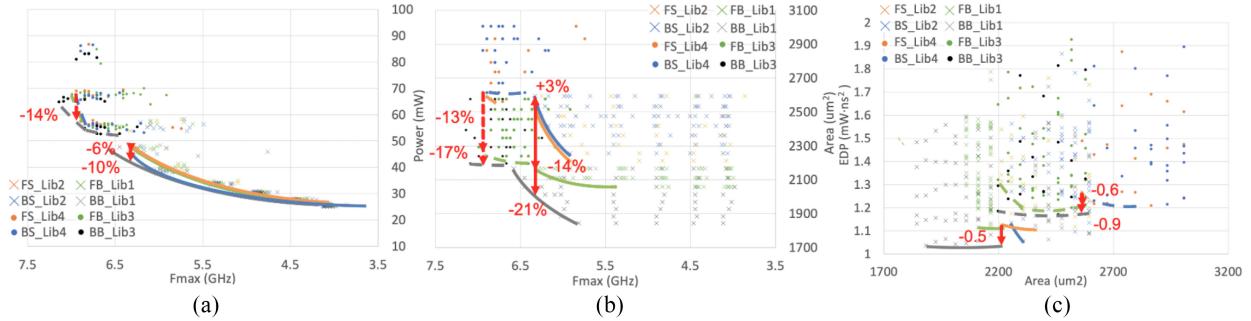


Fig. 15. PPAC tradeoffs for “artificial” JPEG with four libraries (*Lib1–Lib4*). Shown: (a) *Performance–Power*, (b) *Performance–Area*, and (c) *EDP–Area*.

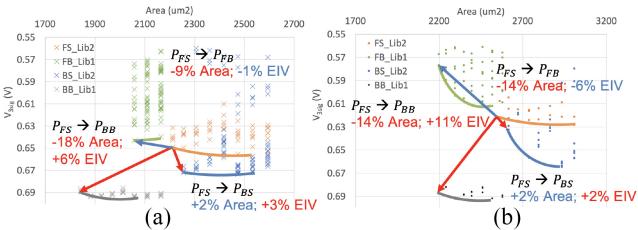


Fig. 16. *IR drop–Area* plots for artificial JPEG with four libraries (*Lib1–Lib4*). (a) JPEG with 4RT (*Lib1/2*). (b) JPEG with 5RT (*Lib3/4*).

- 1) Power consumption with  $P_{BS}$  and  $P_{BB}$  decreases by 6%–14%, compared to  $P_{FS}$  with the same performance.
- 2) Power consumption with  $P_{FB}$  is similar to  $P_{FS}$  with the same performance.
- 3) Results with the artificial JPEG show up to 7% differences, but with similar trends, compared to the results obtained with the real JPEG design.

*Performance Versus Area:* Fig. 15(b) shows tradeoffs between performance and area with the artificial JPEG design. We make three main observations.

- 1) Area with  $P_{FB}$  and  $P_{BB}$  decreases up to 14%–21% compared to  $P_{FS}$  with the same performance.
- 2) Area with  $P_{BS}$  increases by 0%–3% compared to  $P_{FS}$  with the same performance. This area penalty is caused by power tap cell insertion for  $P_{BS}$ .
- 3) We observe that the results with the artificial JPEG show up to 9% differences, but with similar trends, compared to the results obtained with the real JPEG design. However, area for  $P_{BS}$  shows opposite trends to what we observe with the real design, although the discrepancy is not too large.

*EDP Versus Area:* Fig. 15(c) yields three main observations.

- 1) For 4RT (*Lib1* and *Lib2*), EDP with  $P_{BB}$  decreases by 0.5 mW·ns<sup>2</sup>, compared to  $P_{FS}$  with the same area while EDP with  $P_{FB}$  and  $P_{BS}$  remains unchanged.
- 2) For 5RT (*Lib3* and *Lib4*), EDP with  $P_{FB}$  and  $P_{BB}$  decreases by 0.6 and 0.9 mW·ns<sup>2</sup>, compared to  $P_{FS}$  with the same area. Yet, EDP with  $P_{BS}$  shows no improvements.
- 3) We observe that results with the artificial JPEG show similar trends as the real JPEG design.

*Supply Voltage (IR) Drop Versus Area:* Fig. 16(a) and (b) shows tradeoffs between IR drop and area for the artificial JPEG design with 4RT (*Lib1* and *Lib2*) and 5RT (*Lib3* and *Lib4*), respectively. We make four main observations.

- 1) Area with  $P_{FB}$  decreases by 9%–14%, compared to  $P_{FS}$ , while the EIV increases by 1%–6%.

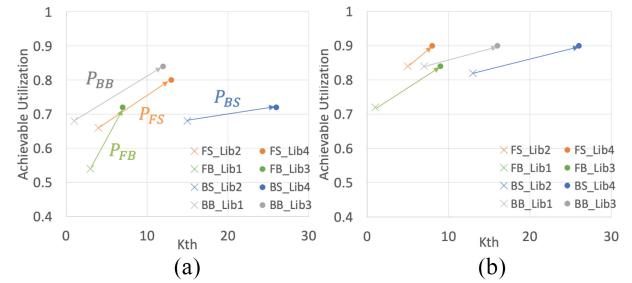


Fig. 17.  $K_{th}$  and achievable utilization for (a) AES and (b) JPEG, with various libraries and power delivery methodologies.

- 2) Area with  $P_{BS}$  increases by 2%, compared to  $P_{FS}$ , while EIV decreases by 2%–3%.
- 3) Area with  $P_{BB}$  decreases by 14%–18%, compared to  $P_{FS}$ , while EIV decreases by 6%–11%.
- 4) We observe that results with the artificial JPEG show similar trends as results obtained with the real JPEG design, and that discrepancies are reasonably small.

#### D. Expt 3 (Routability Assessment and Achievable Utilization)

We measure  $K_{th}$  using our *clustering-based cell width-regularized placements* (Section VI) and explore the relationship between  $K_{th}$  and achievable utilization. We note that [4] introduced *Achievable Utilization* as the maximum utilization for which the number of DRCs is less than a predefined threshold of 500 DRCs. Here, we include all three criteria for a valid result (Section VII-A) and define *Achievable Utilization* as the maximum utilization among all valid runs seen.

Fig. 17 shows the experimental results. We conduct our experiments with artificial JPEG and four cell width-regularized libraries (*Lib1–Lib4*). We make two observations.

- 1) On comparing the results with 2Fin/4RT standard-cell libraries (*Lib1/2*) to those with 3Fin/5RT standard-cell libraries (*Lib3/4*) we find that a larger number of M0 routing tracks brings better routability.
- 2) On comparing to  $P_{FS}$ ,  $P_{FB}$ , and  $P_{BB}$ , the plots for  $P_{BS}$  are skewed to the right for each design, showing better routability than the other PDN configurations. We observe that the routability improvement of  $P_{BS}$  comes from regularly placed power tap cells: the power tap cell placement eases routing congestion caused by high cell and/or pin density.

Finally, we compare the  $K_{th}$  results obtained with the previous *cell width-regularized placements* used in the

TABLE XI

$K_{th}$  COMPARISON FOR THE JPEG DESIGN WITH CELL WIDTH-REGULARIZED PLACEMENTS ([A]) AND CLUSTERING-BASED CELL WIDTH-REGULARIZED PLACEMENTS ([C]). UTIL DENOTES REAL UTILIZATION WITH 0.6 INITIAL UTILIZATION

Rank	PDN	RT	Library	[A]		[C]	
				$K_{th}$	Util	$K_{th}$	Util
1	$P_{FB}$	4	$Lib1$	6	0.14	3	0.50
2	$P_{FS}$	4	$Lib2$	9	0.14	5	0.49
3	$P_{BB}$	4	$Lib1$	12	0.14	7	0.50
4	$P_{FS}$	5	$Lib4$	15	0.14	8	0.50
5	$P_{FB}$	5	$Lib3$	16	0.14	9	0.50
6	$P_{BS}$	4	$Lib2$	17	0.14	13	0.49
7	$P_{BB}$	5	$Lib3$	18	0.14	16	0.50
8	$P_{BS}$	5	$Lib4$	23	0.14	26	0.50

PROBE2.0 work ([A]) and *clustering-based cell width-regularized placements* obtained using the *CWR-FC* algorithm of Section VI-A ([C]). We perform routability assessments as summarized in Table XI. We rank-order  $K_{th}$  across the eight combinations of four PDN and two RT with the JPEG design. The main observation from this comparison is that the ordering of enablements based on  $K_{th}$  is the same for both placements, even as the area utilization of the *clustering-based cell width-regularized placements* is closer to the initial utilization (0.6). We conclude that our *clustering-based cell width-regularized placement* methodology successfully provides more realistic placements without disrupting the  $K_{th}$ -based rank-ordering of enablements. Moreover, the generally smaller  $K_{th}$  values seen in the rightmost two columns of Table XI imply fewer P&R trials needed to evaluate the  $K_{th}$  metric.

## VIII. CONCLUSION

We have presented PROBE3.0, a systematic and configurable framework for “full-stack” PPAC exploration and pathfinding in advanced technology nodes. We introduce automated PDK and standard-cell library generation, along with the enablement of scaling boosters in a predictive 3-nm technology. Our work is permissively open-sourced in GitHub [44] and includes open-sourceable PDKs and EDA tool scripts that incorporate PP considerations into the framework. We employ ANG with an ML-based parameter tuning to mimic properties of arbitrary real designs. Along with a new clustering-based width-regularized netlist and placement methodology, this enables PPAC exploration of a wider space of technology, design enablement, and design options. Experimental results indicate up to 8% power reduction and 24% area reduction using our predictive 3-nm technology, aligning with prior research [7], [18], [28], [30] estimating 25%–30% area reduction using BSPDN and BPR.

Future directions include the following.

- 1) Improving PROBE3.0’s software architecture will enhance user accessibility for PPAC explorations. Allowing user-defined variables can further facilitate the study of diverse technology and design assumptions. For example, power-related parameters such as power scenarios and switching activities can be implemented in the framework for more detailed and design context-specific studies.
- 2) Enhancing the framework’s robustness and real-world relevance requires better device models, parasitic extraction models, signoff corner definitions, and pertinent

design examples. Incorporating DRC rule decks for commercial tools will further bolster the framework.

- 3) Our GitHub repository publicly shares scripts for commercial tools, but using them requires valid licenses. Integrating open-source tools into the PROBE3.0 framework could enable wider deployments, reduce turnaround times, and cater to a broader audience.
- 4) The use of the PROBE3.0 framework still entails the execution of the physical design enablement and flow, from PDK generation to IR drop analysis, which can take significant time. Future works may seek to provide the best parameters for a given technology based on implementation in a greatly reduced set of parameter combinations.

## ACKNOWLEDGMENT

The authors thank Dr. Mustafa Badaroglu at Qualcomm, Dr. Gi-Joon Nam at IBM, Dr. S. C. Song at Google, and Prof. Taigon Song at Kyungpook National University for valuable discussions.

## REFERENCES

- [1] K. Bhanushali and W. R. Davis, “FreePDK15: An open-source predictive process design kit for 15nm FinFET technology,” in *Proc. ACM Int. Symp. Phys. Design*, 2015, pp. 165–170.
- [2] B. Chava et al., “Backside power delivery as a scaling knob for future systems,” in *Proc. SPIE Design Process Technol. Optim. Manuf. XIII* 2019, pp. 1–6.
- [3] C.-K. Cheng, C.-T. Ho, C. Holtz, and B. Lin, “Design and system technology co-optimization sensitivity prediction for VLSI technology development using machine learning,” in *Proc. ACM/IEEE Int. Workshop Syst. Level Interconnect Prediction*, 2021, pp. 1–8.
- [4] C.-K. Cheng et al., “PROBE2.0: A systematic framework for routability assessment from technology to design in advanced nodes,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.* vol. 41, no. 5, pp. 1495–1508, May 2022.
- [5] C. Chidambaram, A. B. Kahng, M. Kim, G. Nallapati, S. C. Song, and M. Woo, “A novel framework for DTCO: Fast and automatic routability assessment with machine learning for sub-3nm technology options,” in *Proc. IEEE Symp. VLSI Technol.*, 2021, pp. 1–2.
- [6] L. T. Clark et al., “ASAP7: A 7-nm FinFET predictive process design kit,” *Microelectron. J.*, vol. 53, pp. 105–115, Jul. 2016.
- [7] M. O. Hossen, B. Chava, G. Van der Plas, E. Beyne, and M. S. Bakir, “Power delivery network (PDN) modeling for backside-PDN configurations with buried power rails and  $\mu$  TSVs,” *IEEE Trans. Electron Devices*, vol. 67, no. 1, pp. 11–17, Jan. 2020.
- [8] D. Johnson, “Near-optimal bin packing algorithms,” Ph.D. dissertation, Dept. Comput. Sci., Massachusetts Inst. Technol., Cambridge, MA, USA, 1973.
- [9] A. B. Kahng, S. Kang, S. Kim, and B. Xu, “Enhanced power delivery pathfinding for emerging 3D integration technology,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 29, no. 4, pp. 591–604, Apr. 2021.
- [10] A. Kahng, A. B. Kahng, H. Lee, and J. Li, “PROBE: Placement, routing, back-end-of-line measurement utility,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.* vol. 37, no. 7, pp. 1459–1472, Jul. 2018.
- [11] G. Karypis, R. Aggarwal, V. Kumar, and S. Shekhar, “Multilevel hypergraph partitioning: Applications in VLSI domain,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 7, no. 1, pp. 69–79, Mar. 1999.
- [12] T. Kim et al., “NS3K: A 3-nm nanosheet FET standard cell library development and its impact,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 31, no. 2, pp. 163–176, Feb. 2023.
- [13] D. Kim, S.-Y. Lee, K. Min, and S. Kang, “Construction of realistic place-and-route benchmarks for machine learning applications,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 42, no. 6, pp. 2030–2042, Jun. 2023.
- [14] L. Liebmann et al., “DTCO acceleration to fight scaling stagnation,” in *Proc. SPIE Design Process-Techol. Manuf. XIV*, 2020, pp. 1–15.
- [15] L.-C. Lu, “Physical design challenges and innovations to meet power, speed, and area scaling trend,” in *Proc. ACM Int. Symp. Phys. Design*, 2017, pp. 63.

- [16] L. Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.
- [17] D. Prasad et al., "Buried power rails and back-side power grids: Arm CPU power delivery network design beyond 5nm," in *Proc. IEEE Int. Electron Devices Meeting*, 2019, pp. 19.1.1–19.1.4.
- [18] J. Ryckaert et al., "Extending the roadmap beyond 3nm through system scaling boosters: A case study on buried power rail and backside power delivery," in *Proc. Electron Devices Technol. Manuf. Conf.*, 2019, pp. 50–52.
- [19] S. Sadangi, "FreePDK3: A novel PDK for physical verification at the 3nm node," M.S. Thesis, Comput. Eng., North Carolina State Univ., Raleigh, NC, USA, 2021.
- [20] G. Sista et al., "IR-drop analysis of hybrid bonded 3D-ICs with backside power delivery and  $\mu$ - & n-TSVs," in *Proc. IEEE Int. Interconnect Technol. Conf.*, 2021, pp. 1–3.
- [21] S. C. Song et al., "Unified technology optimization platform using integrated analysis (UTOPIA) for holistic technology, design and system co-optimization at  $\leq 7$  nm nodes," in *Proc. IEEE Symp. VLSI Circuits*, 2016, pp. 1–2.
- [22] H. Su, J. Hu, S. S. Sapatnekar, and S. R. Nassif, "Congestion-driven codesign of power and signal networks," in *Proc. ACM/ESDA/IEEE Design Autom. Conf.*, 2002, pp. 64–69.
- [23] S. S. T. Nibhanupudi et al., "A holistic evaluation of buried power rails and back-side power for sub-5nm technology nodes," *IEEE Trans. Electron Devices*, vol. 69, no. 8, pp. 4453–4459, Aug. 2022.
- [24] 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," in *Proc. Int. Conf. Comput.-Aided Design*, 2023, pp. 1–9.
- [25] M. D. Hutton, J. Rose, J. P. Grossman, and D. G. Corneil, "Characterization and parameterized generation of synthetic combinational benchmark circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 10, no. 17, pp. 985–996, Oct. 1998.
- [26] A. Surana, C. Chen, and I. Rajapakse, "Hypergraph similarity measures," *IEEE Trans. Netw. Sci. Eng.*, vol. 10, no. 2, pp. 658–674, 2023.
- [27] "68–95–99.7 rule." Accessed: Dec. 10, 2023. [Online]. Available: [https://en.wikipedia.org/wiki/68-95-99.7\\_rule](https://en.wikipedia.org/wiki/68-95-99.7_rule)
- [28] "Applied materials logic master class." Accessed: Dec. 10, 2023. [Online]. Available: <https://ir.appliedmaterials.com/static-files/acba6be3-4778-41eb-9183-5c8e52884dea>
- [29] "Artificial netlist generator." Accessed: Dec. 10, 2023. [Online]. Available: [https://github.com/daeyeon22/artificial\\_netlist\\_generator](https://github.com/daeyeon22/artificial_netlist_generator)
- [30] D. O'Laughlin, "Backside power delivery and bold bets at Intel," Accessed: Jun. 2022. [Online]. Available: <https://www.fabricatedknowledge.com/p/backside-power-delivery-and-bold>
- [31] "Cadence QRC extraction user guide." Accessed: Dec. 10, 2023. [Online]. Available: <http://www.cadence.com>
- [32] "Cadence voltus user guide." Accessed: Dec. 10, 2023. [Online]. Available: <http://www.cadence.com>
- [33] "FreePDK3 predictive process design kit." Accessed: Dec. 10, 2023. [Online]. Available: <https://github.com/ncsu-eda/FreePDK3>
- [34] "GDT to GDS format translator." Accessed: Dec. 10, 2023. [Online]. Available: <https://sourceforge.net/projects/gds2>
- [35] "H2O AutoML." Accessed: Dec. 10, 2023. [Online]. Available: <https://www.h2o.ai>
- [36] "IEEE international roadmap for devices and systems (IRDS)." 2020. Accessed: Dec. 10, 2023. [Online]. Available: <https://irds.ieee.org/editions/2020>
- [37] "OpenCores: Open source IP-cores." Accessed: Dec. 10, 2023. [Online]. Available: <http://www.opencores.org>
- [38] "OpenDB." Accessed: Dec. 10, 2023. [Online]. Available: <https://github.com/The-OpenROAD-Project/OpenDB>
- [39] "Siemens EDA calibre user guide." Accessed: Dec. 10, 2023. [Online]. Available: <https://eda.sw.siemens.com/en-US/ic/calibre-design/>
- [40] "Synopsys design compiler user guide." Accessed: Dec. 10, 2023. [Online]. Available: <http://www.synopsys.com>
- [41] "Synopsys IC compiler II user guide." Accessed: Dec. 10, 2023. [Online]. Available: <http://www.synopsys.com>
- [42] "Synopsys StarRC user guide." Accessed: Dec. 10, 2023. [Online]. Available: <http://www.synopsys.com>
- [43] "Synopsys DTCO flow: Technology development." Accessed: Apr. 25, 2023. [Online]. Available: <https://www.synopsys.com/silicon/resources/articles/dtco-flow.html>
- [44] "The PROBE3.0 framework." Accessed: Dec. 10, 2023. [Online]. Available: <https://github.com/ABKGroup/PROBE3.0>
- [45] "Z3 SMT solver." [Online]. Available: <https://github.com/Z3Prover/z3>



**Suhyeong Choi** (Graduate Student Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology, Daejeon, South Korea, in 2016 and 2018, respectively. He is currently pursuing the Ph.D. degree in electrical engineering with Stanford University, Stanford, CA, USA.

His research interests include monolithic 3-D IC physical design and its architecture.



**Jinwook Jung** (Member, IEEE) received the Ph.D. degree in electrical engineering from the Korea Advanced Institute of Science and Technology, Daejeon, South Korea, in 2018.

He is currently a Senior Research Scientist with IBM T. J. Watson Research Center, Yorktown Heights, NY, USA. His current research interests include VLSI physical design methodology and AI hardware accelerator designs.



**Andrew B. Kahng** (Fellow, IEEE) received the Ph.D. degree in computer science from the University of California at San Diego, La Jolla, CA, USA, in 1989.

He is currently with the University of California at San Diego. His research interests include IC physical design, design-manufacturing interface, large-scale combinatorial optimization, AI/ML for EDA and IC design, and technology roadmapping.



**Minsoo Kim** (Member, IEEE) received the Ph.D. degree in electrical and computer engineering from the University of California at San Diego, La Jolla, CA, USA, in 2023.

He is currently with NVIDIA Corporation, Austin, TX, USA. His research interests include VLSI physical design and design-technology co-optimization.



**Chul-Hong Park** (Member, IEEE) received the Ph.D. degree in electrical and computer engineering from the University of California at San Diego, La Jolla, CA, USA, in 2008.

He is currently with Hyundai MOBIS, Seoul, South Korea. His research interests include nanometer physical design, automotive circuit and power-device design for manufacturing, and 2-D/3-D package for emerging technologies.



**Bodhisattva Pramanik** (Graduate Student Member, IEEE) received the M.S. degree in computer engineering from Iowa State University, Ames, IA, USA, in 2022. He is currently pursuing the Ph.D. degree with the University of California at San Diego, La Jolla, CA, USA.

His research interests include hypergraph partitioning, graph clustering, placement methodology, and optimization algorithms.



**Dooseok Yoon** (Graduate Student Member, IEEE) received the B.S. degree in electrical engineering from Ajou University, Suwon, South Korea, in 2008. He is currently pursuing the Ph.D. degree with the University of California at San Diego, La Jolla, CA, USA.

His research interests include design-technology co-optimization and VLSI physical design.