Invited Paper: IEEE CEDA DATC Emerging Foundations in IC Physical Design and MLCAD Research

Jinwook Jung*, Andrew B. Kahng, Sayak Kundu, Zhiang Wang, and Dooseok Yoon
jinwookjung@ibm.com, {abk,sakundu,zhw033,d3yoon}@ucsd.edu

UC San Diego, La Jolla, CA, USA
*IBM T. J. Watson Research Center, Yorktown Heights, NY, USA

***Abstract*—Recent activities of the IEEE CEDA DATC strengthen the DATC Robust Design Flow (RDF) and broadly support research on machine learning for CAD/EDA (MLCAD). The RDF-2023 version of the RDF adds standalone and integrated netlist partitioners, a detailed placement optimizer, dynamic power analysis, and enablement of new directions (design-technology co-optimization and 3D layout). Advancement of benchmarking practices and strong baselines has continued – e.g., the MacroPlacement effort introduced in RDF-2022 now has new benchmarks, integration of the AutoDMP macro placer, and baseline solutions generated by Simulated Annealing and human experts. Other DATC efforts have focused on *proxies* and other elements of MLCAD research enablement. These include real and synthetic benchmarks tailored for IR drop analysis, a calibration methodology for research PDKs, and artificial netlist generation for data augmentation and design space coverage of netlists used in model training. We conclude with directions for future DATC efforts.**

***Index Terms*—VLSI CAD, open source, EDA, machine learning**

I. Introduction

The Design Automation Technical Committee (DATC) is a technical committee of the IEEE Council on EDA (CEDA) [36]. It provides a forum for discussing strategies and issues in design automation. Since 2016, the DATC has maintained the Robust Design Flow (RDF), an academic reference RTL-to-GDS design flow that incorporates numerous contest-winning point tools and the OpenROAD tool chain [32]. The RDF was initiated with two overarching goals: (i) to preserve and integrate leading research codes, and (ii) to trigger design flow and cross-stage optimization research via tools developed in academia. A series of invited papers [7]–[9], [16]–[20] has provided updates on RDF enhancements as well as DATC's strategic directions and development activities. Table I lists the tools currently used in RDF; the new 2023 additions are in bold.

With RDF as a foundation, the IEEE CEDA DATC has been extending its activities beyond flow enablement, with the goal of establishing and expanding research foundations for IC physical design and for machine learning in CAD/EDA (MLCAD). Some of the main directions of DATC efforts in the past year include the following.

- **Recent improvements of RDF.** RDF-2023 has added the *TritonPart* netlist partitioner, as well as the *DPO* detailed placement optimizer and *OpenSTA*-based dynamic power analysis. RDF-2023 also opens up two new directions: design-technology co-optimization (DTCO) via the *PROBE3.0* platform [15], and enablement of fine-grain 3D layout generation. The RDF improvements are available as open source through OpenROAD integrations.
- **Advancement of benchmarking practices and baselines.** The MacroPlacement effort, introduced in RDF-2022, now has additional modern benchmarks, integration of the *AutoDMP* macro placer [1], and baseline solutions produced by multiple methods including simulated annealing (SA) and human experts. Separately, a benchmarking and leaderboard repository for hypergraph partitioning [34] now includes results for K-way partitioning instances and the *K-SpecPart* partitioner.
- **MLCAD research enablement.** The need for open data for MLCAD research has been highlighted in many forums; [45] gives a recent summary of the associated challenges. DATC efforts have focused on *proxies* and other elements of MLCAD research enablement. Developments in 2023 include real and synthetic benchmarks for IR drop analysis and prediction [38] [43], and a calibration methodology to close the gap between open research PDKs and closed commercial PDKs. Artificial netlist generation [23] [37], for data augmentation and design space coverage of netlists used in model training, has also been integrated. Key nascent directions include (i) connecting cell library synthesis and design enablement flows of *PROBE3.0* to data generation for MLCAD, and (ii) making the RDF tool chain more ML-friendly and connecting to recent initiatives such as NVIDIA's *CircuitOps* [42].

The remainder of this paper is organized as follows. Section II discusses the recent advancements in RDF. Section III reviews developments toward benchmarks and stronger baselines. Section IV discusses proxies (optimizers, design enablements), synthetic data generation, and related topics. We conclude in Section V with directions for future DATC efforts.

II. Recent Improvements of RDF

We highlight four main improvements made in RDF-2023.

A. TritonPart: Constraints-Driven Partitioning Framework

Netlist partitioning has been a prominent missing component in RDF; this has limited the ability to effectively handle large designs with millions of instances. For physical

TABLE I
RDF-2023 COMPONENTS.

Component	Tools
RTL generator	Chisel/FIRRTL
RTL obfuscation	ASSURE
Logic synthesis	Yosys, ABC
Hypergraph Partitioner	**SpecPart**, **TritonPart**
DFT insertion	Fault
Floorplanning	TritonFP
Macro Placement	TritonMP, RTL-MP, Hier-RTLMP, **AutoDMP**
Global placement	RePlAce, FZUplace, NTUPlace3, ComPLx, Capo, Eh?Placer, FastPlace3-GP, mPL5/6, DREAMPlace
Detailed placement	OpenDP, MCHL, FastPlace3-DP, **DPO**
Flip-flop clustering	Mean-shift, FlopTray
Clock tree synthesis	TritonCTS
Global routing	FastRoute4-lefdef, NCTUgr, CUGR
Detailed routing	TritonRoute, NCTUdr, DrCU
Layout finishing	KLayout, Magic
Gate sizing	Resizer, TritonSizer
Parasitic extraction	OpenRCX
STA	OpenSTA, iTimerC
Database	OpenDB
Libraries/PDK	**GF180MCU**, NanGate45, SKY130, ASAP7, NCTUcell, **ASAP5**
Integrated app	OpenROAD
Benchmark conversion	RosettaStone
DTCO	**PROBE3.0**

estimation and implementation, large system designs must be partitioned into multiple blocks, tiers and devices that achieve timing closure when implemented in parallel or concurrently. *TritonPart* has been added to RDF-2023 as a general-purpose constraints-driven partitioning engine, integrated into OpenROAD. In contrast to other publicly available partitioners such as *SpecPart* [34], *hMETIS* [22] and *KaHyPar* [31], *TritonPart* provides a modern *"partitioning multi-tool"* that is:

- Applicable to both classical balanced hypergraph partitioning (*triton_part_hypergraph* in Figure 1) and timing-aware VLSI netlist partitioning (*triton_part_design* in Figure 1);
- Able to handle rich constraint types including fixed vertex constraints, multi-dimensional balance constraints, grouping constraints, embedding constraints, and timing constraints;
- Permissively open-sourced and scalable in order to accommodate future problem instances.

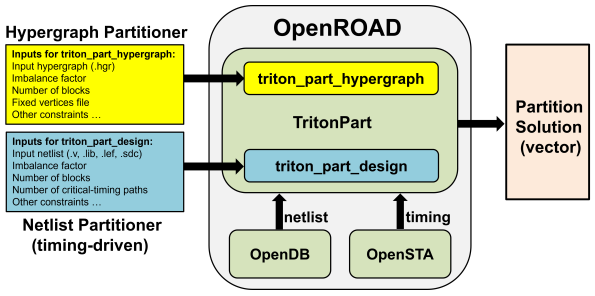
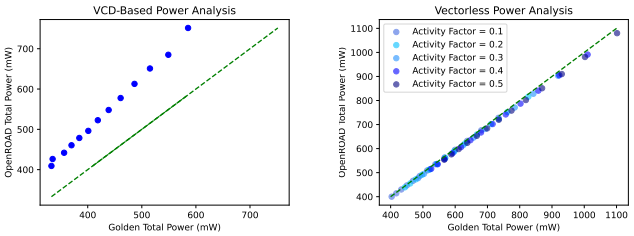


Fig. 1. Applications of the *TritonPart* engine.

B. Dynamic Power Analysis in OpenSTA

The OpenSTA static timing analysis engine now supports vectored and vectorless dynamic power analysis. Given a specific *value change dump* (VCD), or an activity factor, OpenROAD can calculate the total dynamic power consumption of a design. The command for this analysis is displayed in Figure 3. Users have reported differences in the power analysis results from OpenSTA and unnamed signoff evaluators; an example (Ariane RISC-V core in NanGate45, with bsg_fakeram SRAM models [39] [44] and evaluation at multiple clock periods) is shown in Figure 2. While the current implementation is reported to reasonably align with signoff evaluators, improvement of the dynamic power calculation is ongoing.



(a) (b)

Fig. 2. (a) Vectored and (b) vectorless dynamic power correlation between OpenSTA and an unnamed signoff tool for the Ariane design on NanGate45 with bsg_fakeram SRAM models.

```
# VCD-Based Power Analysis
read_power_activities -scope <scope> -vcd <vcd file>
report_power
# Vectorless Power Analysis
set_power_activity -global -activity <activity factor>
report_power
```

Fig. 3. OpenROAD (OpenSTA) command for dynamic power analysis.

C. Detailed Placement Optimization

A new *detailed placement optimizer* (DPO) has been added to RDF-2023, via integration into the OpenROAD app. The code was developed by Professor Andrew Kennings at the University of Waterloo. DPO is invoked using the *improve_placement* command in OpenROAD; it performs a variety of operations including global swaps, vertical swaps, random cell swaps, and cell reordering. These operations can be conducted with varying or combined goals, such as reducing half-perimeter wirelength (HPWL), maximizing placement utilization, and/or minimizing overall cell displacement. Table II shows DPO results for aes, ibex, and jpeg designs across ASAP7, NanGate45, and SKY130HD enablements, indicating a 1% to 5.5% routed wirelength (rWL) benefit from DPO.

TABLE II
POST-PLACEMENT HPWL AND POST-ROUTE WIRELENGTH (RWL) FOR AES, IBEX AND JPEG DESIGNS ON ASAP7 (A), NANGATE45 (N) AND SKY130HD (S) ENABLEMENTS WITH AND WITHOUT DPO.

Design-Enable.	Baseline		With DPO		HPWL Δ %	rWL Δ %
	HPWL	rWL	HPWL	rWL		
aes-A	74320.3	112370.0	71780.8	109239.0	-3.42	-2.79
aes-N	210407.7	329984.0	199669.0	312472.0	-5.10	-5.31
aes-S	553464.0	838329.0	519277.4	793545.0	-6.18	-5.34
ibex-A	62157.6	89161.0	61114.0	88154.0	-1.68	-1.13
ibex-N	192411.9	269948.0	186931.2	263464.0	-2.85	-2.40
ibex-S	446403.5	664436.0	435362.8	647759.0	-2.47	-2.51
jpeg-A	138651.1	162624.0	134809.9	159228.0	-2.77	-2.09
jpeg-N	475773.5	568330.0	456259.9	548508.0	-4.10	-3.49
jpeg-S	972223.3	1185920.0	927963.1	1136801.0	-4.55	-4.14

D. PROBE3.0 DTCO Pathfinding Framework

Design-technology co-optimization (DTCO) is increasingly required to extract IC product benefit from advanced manufacturing technology [28]. The DTCO process can enable exploration of today's rich space of scaling booster, integration technology, and system and architecture options. However, this requires scalability and automation of standard-cell library and PDK (tool enablement) generation processes, which have traditionally been the bottleneck for DTCO. Figure 4 shows the flow of *PROBE3.0* standard-cell and PDK generation [15]. Technology and design parameters are inputs to the flow.

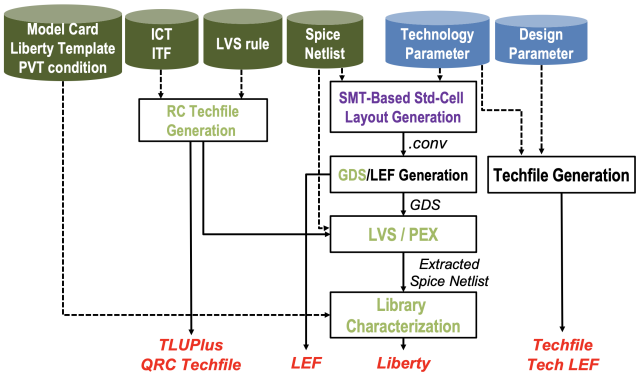


Fig. 4. Automatic generation of standard-cell library and PDK (*Design Enablements*) in the PROBE3.0 framework. In addition to technology and design parameters, other technology-related inputs are required: (i) device model cards, (ii) Liberty templates, (iii) process/voltage/temperature (PVT) conditions, (iv) interconnect technology files (ICT or ITF formats), (v) LVS rules, and (vi) SPICE netlists [15].

Three salient aspects of PROBE3.0 are as follows.

- *Realistic artificial netlist generation using ML.* The capability to generate artificial yet realistic designs enables more robust evaluation of PPAC impacts from various technology options, across a wider space of designs. PROBE3.0 leverages the *Artificial Netlist Generator* (ANG) [23] to generate artificial designs, and applies machine learning (AutoML) to improve the similarity of artificially-generated netlists to targeted real netlists.
- *Improved 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. PROBE3.0 and its extensions improve the SMT-based layout generation used in PROBE2.0 [10] to support arbitrary "gear ratios" between contacted poly pitch and M1 pitch.
- *PPAC exploration of scaling boosters.* Advanced-node scaling has been accomplished through steady targeting and realization of scaling boosters (contact over active gate, single diffusion break, etc.). To overcome challenges of high BEOL resistance and routing congestion overheads, multiple foundries have begun implementing backside power delivery networks (BSPDN) and buried power rails (BPR) as scaling boosters in their sub-5nm technologies [29] [30]. PROBE3.0 enables detailed study of PDN scaling boosters for "end to end" DTCO and pathfinding that spans from patterning ground rules and device models all the way to IR drop- and routability-constrained PPAC assessment of real design blocks.

Figure 5 shows Energy-Delay Product (EDP) versus Area data from a PROBE3.0 exploration of power delivery scaling boosters. The data show that compared to P_{FS} (iso-area), (i) for $2Fin$ (*Lib1* and *Lib2*), EDP with P_{FB}, P_{BS} and P_{BB} decreases by 0.2, 0.2 and 0.4 $mW \cdot ns^2$, respectively; and (ii) for $3Fin$ (*Lib3* and *Lib4*), EDP with P_{FB}, P_{BB} decreases by 0, 0.3 $mW \cdot ns^2$ respectively, while P_{BS} increases by 0.1 $mW \cdot ns^2$. Further details are available in [15].

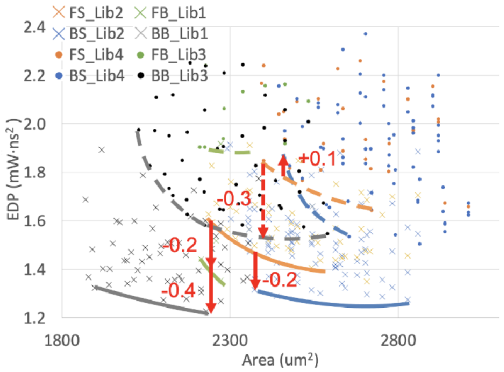


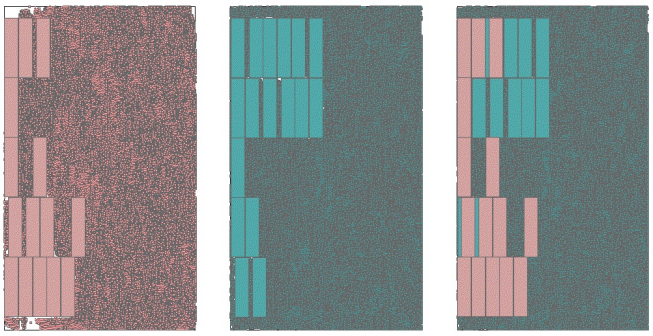
Fig. 5. Energy-Delay Product (EDP) vs. Area tradeoffs for JPEG with four PDN options in the PROBE3.0 framework: (i) Frontside PDN (FS), (ii) Frontside PDN with BPR (FB), (iii) Backside PDN (BS), and (iv) Backside PDN with BPR (BB). For details, see [15].

E. Extensions for 3D Physical Design

With continued headwinds for traditional 2D scaling, along with clear opportunities for 3D heterogeneous integration-based scaling, there is growing interest in EDA tooling and MLCAD support for multi-tier physical design [48]. OpenDB has been extended to support multiple dies (tiers) and hybrid bonds, enabling flexible exploration and development of new fine-grain 3D place-and-route and multi-tier optimization capabilities in OpenROAD. For example, the RePlAce global placer has been updated to assist with the global placement of 3D multi-tier systems. Exploration of netlist partitioning, clustering and tier assignment can be done using OpenROAD infrastructure and engines. For 3D global placement, RePlAce can cycle through the tiers, making a single iteration for each tier followed by an update to hybrid bond locations. The RePlAce iterations for each tier explicitly account for hybrid bond connections. The OpenDP legalizer has been enhanced to support the legalization of hybrid bonds. Figure 6 shows an example 3D global placement for Case 4 of ICCAD-2023 Contest Problem B, generated using OpenROAD.

III. Advancing Benchmarking and Baselines

DATC efforts have sought changes in culture and academia-industry interaction models, to improve both the velocity and the accessibility of EDA and MLCAD research. These efforts bridge the gap caused by restrictions on benchmarking, sharing of design and tool data, etc. in the commercial sphere. The topic of benchmarking is closely tied to the fact that the core of EDA (and IC design itself) is optimization: benchmarking clarifies the leading edge (i.e., via *baseline* methods and solutions to benchmark problem instances) and drives progress in the field. Benchmarking is also closely associated with availability of open enablements and data for research, since



(a) (b) (c)

Fig. 6. OpenROAD 3D global placement of ICCAD-2023 Contest Problem B, Case 4: (a) top die, (b) bottom die, and (c) both dies together.

these open the door to communication and sharing. We note that benefits for applied optimization in EDA are similarly seen for MLCAD, where benchmarks and baselines promote transparency, reproducibility, and continual refinement of ML models. In this section, we review advances in benchmarking and baselines in two areas, hypergraph partitioning and macro placement.

A. Balanced Hypergraph Partitioning

Balanced hypergraph partitioning is a well-studied, fundamental combinatorial optimization problem with applications throughout VLSI CAD. Many partitioners, such as *hMETIS* [22] and *KaHyPar* [31], have been proposed over the past decades. However, in that state-of-the-art hypergraph partitioners follow the multilevel paradigm, they may become stuck in local optima. To address this, *SpecPart* at ICCAD-2022 [5] introduced a supervised spectral framework for balanced hypergraph partitioning solution improvement, resulting in a stronger baseline for research in both partitioning optimizers and optimization proxies.

SpecPart is an instantiation of a general framework for improving a given solution to a partitioning instance. It takes a cut obtained by a multilevel partitioner as a *hint* toward a better solution, and uses a spectral algorithm that encodes into a generalized eigenvalue problem the supervision information from the *hint*. Figure 7 illustrates how supervision incentivizes the computation of an embedding that in general respects (spatially) the given *hint* solution, but also identifies vertices of contention where improving the solution may be possible. *SpecPart* substantially improves the known leaderboard of minimum hyperedge cutsize values for the classic ISPD98 [3] and Titan23 benchmarks [27].

The *HypergraphPartitioning* repo [34] exemplifies a potential "papers with code" culture change to attain the sharing, reproducibility and transparency that is seen in AI/ML, machine vision and other fields. All runscripts and related partitioning solutions are provided in the repo, to ensure full transparency and reproducibility. The repo also maintains a leaderboard of best known K-way partitioning [6] results for the ISPD98 and Titan23 benchmarks.

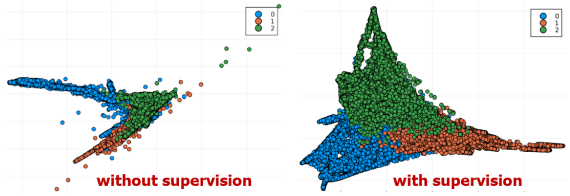


Fig. 7. Vertex embeddings of the Titan23 *gsm_switch* benchmark. Point colors indicate block membership in a 3-way partitioning solution with imbalance factor $\epsilon = 5\%$ computed by *hMETIS* [22]. The embedding on the right uses as a hint the same *hMETIS* solution, while the embedding on the left is unsupervised.

B. Macro Placement

Macro placement is another classic problem in the VLSI CAD literature that has received heightened attention due to its impact on physical implementation QOR and the rapid scaling of real-world instance complexities.

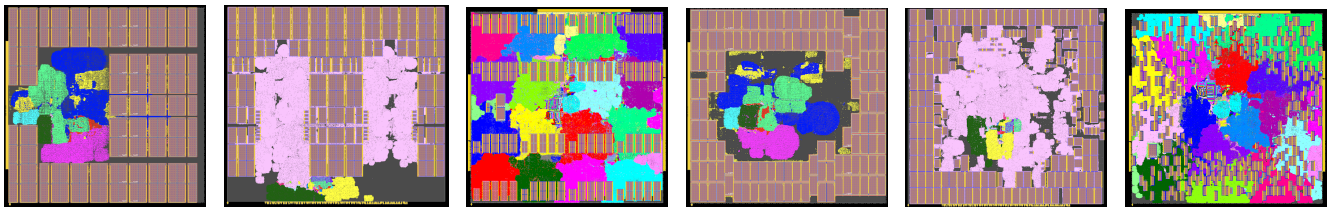
New Benchmarks. New, highly scaled benchmarks in open enablements have been added to the *MacroPlacement* effort [39] [11] that was introduced in RDF-2022. The BlackParrot Quad-core and MemPool Group benchmarks respectively have macro counts of 220 and 324, and instance counts of approximately 800K and 2.8M, in the NanGate45 enablement. New TCL flow scripts have been made public in [39] to facilitate physical synthesis using Synopsys Design Compiler Topographical and post-P&R evaluation of macro placement solutions using Cadence Innovus; headers seen in the TCL scripts reflect the recently-granted permissions from major EDA suppliers, which provide a major step forward for the research community.

New Baselines. Along with the testcases, we introduce two sets of baseline solutions, one using Simulated Annealing (SA) and the other from human experts. SA is implemented according to the description given in [26], [40], with code open-sourced in the MacroPlacement repository. The results presented in [11] demonstrate that these benchmarks serve as a competitive point of reference for ML-based macro placers. Figure 8 shows example macro placement solutions generated by human experts and by SA.

AutoDMP Baseline. The Automated DreamPlace-based Macro Placer (AutoDMP) [1] has been developed and open-sourced by NVIDIA Research; it has been added as another baseline macro placement optimizer. AutoDMP fine-tunes various DREAMPlace hyperparameters, such as target density, density weight, the number of horizontal and vertical bins, learning rate (LR), LR decay, HPWL model, and more, using a multi-objective Parzen tree estimator to identify the Pareto front for rectilinear Steiner minimum tree (RSMT) wirelength, density, and Rudy-based congestion. It then selects the configuration from the Pareto front that yields the lowest cost based on wirelength, density and congestion. Figure 9 displays example macro placement solutions generated by AutoDMP in NanGate45 enablement.

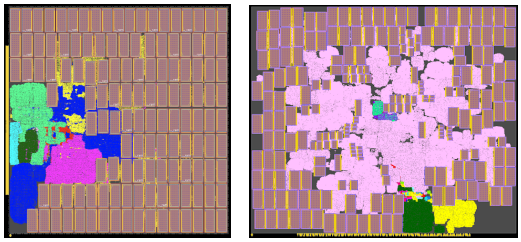
IV. MLCAD Research Enablement

Research in MLCAD has long-standing challenges that include scarcity of training data, confidentiality (PDKs, de-



(a) (b) (c) (d) (e) (f)

Fig. 8. Baseline macro placement solutions generated by human experts: (a) Ariane, (b) BlackParrot and (c) MemPool Group; and using SA: (d) Ariane, (e) BlackParrot and (f) MemPool Group.



(a) (b)

Fig. 9. Macro placement results generated by AutoDMP for: (a) Ariane and (b) BlackParrot.

signs, EDA tool reports, etc.) which demands the creation of *proxies*, and the lack of standardized data formats [45]. These challenges have come under the spotlight as education and workforce development needs of the semiconductor design and EDA industries grow more urgent. In this section, we present several DATC efforts toward MLCAD research enablement, including artificial netlist generation, benchmark generation for IR drop prediction, calibration of proxy PDKs, and nascent connections of OpenROAD to NVIDIA *CircuitOps* and ML platforms.

A. Artificial Netlist Generation

To address the lack of shareable place-and-route (P&R) data, Artificial Netlist Generator (ANG) [23] [37] offers a structured approach to creating realistic P&R benchmarks for ML applications. The framework uses a number of standard topological and layout parameters, including number of instances, number of IOs, average net degree, average net half-perimeter wirelength, average depth of timing paths, and ratio of sequential:combinational instances. Figure 10, reproduced from [23], shows the P&R data generation flow using a netlist generated by ANG; the figure shows how different topological parameters are sampled to yield an artificial but realistic training dataset for MLCAD.

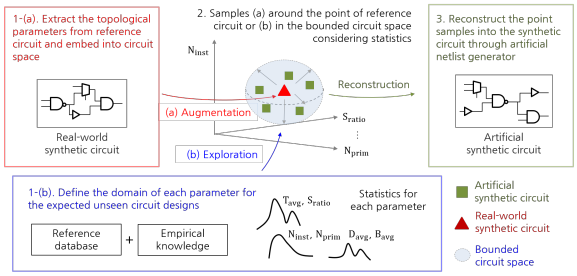


Fig. 10. Flow used by ANG to generate an artificial netlist [23] along with the netlist sampling process using ANG. Figure reproduced from [23].

B. Calibration and ML Benchmarks for IR Drop Prediction

The ICCAD-2023 Contest Problem C focuses on utilizing machine learning to predict static IR drops. The contest provides twenty real [43] circuit testcases on NanGate45, created using OpenROAD-flow-scripts [33] and PDNSim for SPICE netlist representation. These testcases vary with respect to design areas, power maps, grid density, and voltage pad locations. Additionally, a GAN-based [13], [38] framework is utilized to generate thousands of synthetic netlists featuring varied power distributions and grid densities, which are publicly accessible. Golden IR drop data produced by SPICE simulation accompanies each data point in both real and synthetic datasets, providing support for ML EDA, calibration and power integrity research.

C. Calibration of Proxy Design Enablement

Recent work has explored the possibility of scaling public PDK and design enablement (Liberty, LEF, etc.) data, so as to achieve a *proxy* enablement that mimics analogous commercial data. The goal is to match or "bound" the commercial design enablement, particular in terms of design PPA outcomes. This serves not only EDA and MLCAD researchers, but also designers who wish to perform early design exploration with reduced barriers to access.

Our working hypothesis is that to narrow the gap between open-source and closed-source design enablements, it may be sufficient to apply scaling (eventually, aided by machine learning) of standard-cell delays and slews, internal power, switching power and other SP&R-relevant attributes. Figure 11 shows the outcome when the ASAP7 [47] enablement is calibrated (scaled) to match a corresponding Foundry 7nm enablement. The figure shows block-level power, performance and area for the JPEG design at 70% utilization, based on a sweep of target clock period (Fmax). In this example, scaling factors (undisclosed to protect the foundry's IP) for internal power, switching power and cell delays/slews in Liberty are applied separately for each threshold voltage (VT). This small set of nine scaling factors can expand to encompass pin capacitance, leakage, setup and hold timing, gate type, area, and BEOL resistance and capacitance. As the number of scaling factors grows, techniques such as autotuning can help to search for best-possible combinations of scaling factors. In this experiment, Ray Tune [25] is applied in the search of scaling factors. Figure 11 shows that autotuning can find a combination of the nine scaling factors that closely matches the foundry 7nm target *from the design implementation PPA point of view*. Ongoing work pursues automated, ML-boosted

search for proxy enablements using an expanded space of scaling parameters (e.g., including area and BEOL).

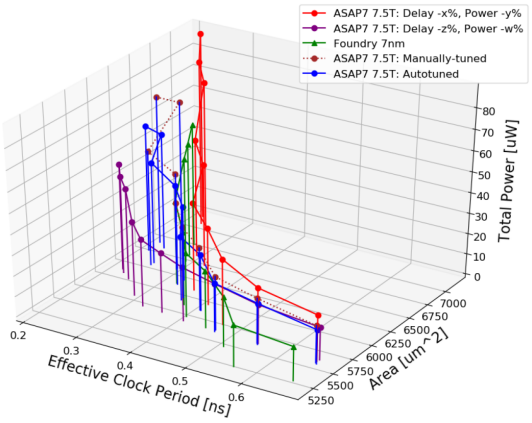


Fig. 11. 3D PPA plot for JPEG encoder implementation based on reference Foundry 7nm and scaled ASAP7 proxy design enablements.

D. MLCAD Data Generation and Accessibility

Toward MLCAD data generation and accessibility, a significant recent development is *CircuitOps* [24], which has created an ML-friendly circuit data representation format in the form of a labeled property graph (LPG) for easy operation with ML algorithms. CircuitOps uses OpenROAD to parse design and technology files (DEF/LEF/Verilog/Liberty) to generate an intermediate representation of the circuit data in the form of relational tables through TCL APIs. The relational tables with the properties and connectivity information are then used by CircuitOps to create LPGs. The CircuitOps GitHub repository [42] contains sample intermediate representation and LPGs for various designs in NanGate45, ASAP7, and SKY130HD technology nodes. These LPGs can serve graph-based ML applications in physical design, e.g., enabling easy application of GNN/GCN based algorithms, easy query of node features, and batch parallel processing with PyTorch.

Figure 12 depicts a roadmap for Python and OpenROAD. While OpenROAD has a Python interpreter that runs all engines except for OpenSTA, this capability has to date been underused. The vision is that OpenROAD in RDF can become a "playground" for EDA researchers and the chip design community, where ML/RL algorithms integrate seamlessly and the research community can explore the potential of ML both within and around traditional EDA algorithms [12]. While some of the python APIs exist, an ML for EDA playground requires three elements: (i) data representations/formats for LEF/DEF/Verilog that integrate with ML libraries; (ii) additional python APIs that return information from the database in formats ML algorithms can interpret; and (iii) APIs that can have callbacks from ML predictions back to the database. For example, graph-based ML algorithms that perform node and edge transformations can map back to updates to the netlist and database. Realizing this roadmap will be a major goal for the next edition of the RDF.

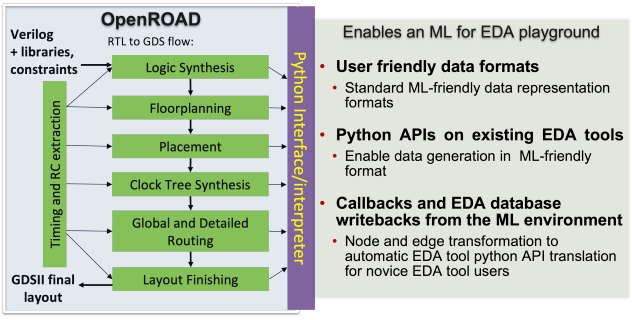


Fig. 12. Python and OpenROAD roadmap. Figure courtesy of V. A. Chhabria [12].

V. Conclusion and Future Directions

In this paper, we have summarized key DATC efforts from the past year.

Improvements in the RDF-2023 tool chain include integration of the *TritonPart* constraint-driven netlist partitioner, dynamic power analysis capability, a new detailed placement optimizer, enablement of 3D layout generation within OpenROAD, integration of the new *PROBE3.0* DTCO framework, and extensions for 3D physical design.

Advancement of benchmarking practices and baselines saw expansion of the MacroPlacement initiative to incorporate new benchmarks along with baseline solutions from *AutoDMP*, Simulated Annealing and human experts – as well as commercial tool flow scripts that enable full reproduction of academic research results. For the latter, recent permissions from major EDA suppliers are a major step forward for the research community, and are gratefully acknowledged. A benchmarking and leaderboard repository [34] for K-way balancecd hypergraph partitioning gives an example of potential future "papers with code" practice in the EDA research literature.

MLCAD research enablement has continued to address two primary challenges: the shortage of data, and the need for baselines to assess both ML models and ML-based optimization and analysis outcomes. Development of *proxies* can unblock availability of open data (i.e., tool enablements, design testcases, and all forms of derived data from tools and flows). One potential direction is seen in autotuned scaling of the ASAP7 enablement to align with a commercial 7nm enablement. A second direction is the use of an artificial netlist generator (ANG) that is designed for ML model training in physical design tasks. Additionally, new real and synthetic IR drop benchmarks have been made available for IR drop prediction; this builds on the Calibrations effort initiated in RDF-2020 [8] [46] and is the basis for "Contest C" at this conference [43]. As noted above, RDF-2023 also provides baselines for ML EDA optimizations, including hypergraph partitioning and macro placement.

Future DATC and RDF efforts aim to connect the cell library synthesis and design enablement flows of *PROBE3.0* to data generation for MLCAD. Another important direction is to make the RDF tool chain more ML-friendly.

Acknowledgments

We thank Vidya Chhabria, Bodhisatta Pramanik, Minsoo Kim and Haoxing (Mark) Ren for many helpful discussions and contributions. This work is supported in part by the IEEE Council on Electronic Design Automation (CEDA), by DARPA HR0011-18-2-0032, and by NSF CCF-2112665.

REFERENCES

[1] A. Agnesina, P. Rajvanshi, T. Yang et al., "AutoDMP: Automated DREAMPlace-based Macro Placement", *Proc. ISPD*, 2023, pp. 149–157.

[2] T. Ajayi, V. A. Chhabria, M. Fogaça, S. Hashemi, A. Hosny et al., "Toward an Open-Source Digital Flow: First Learnings from the OpenROAD Project", *Proc. ACM/IEEE DAC*, 2019, pp. 76:1–76:4.

[3] C. J. Alpert, "The ISPD98 circuit benchmark suite", *Proc. ISPD*, 1998, pp. 80–85.

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

[5] I. Bustany, A. B. Kahng, Y. Koutis, B. Pramanik and Z. Wang, "SpecPart: A Supervised Spectral Framework for Hypergraph Partitioning Solution Improvement", *Proc. ICCAD*, 2022, pp. 1–9.

[6] I. Bustany, A. B. Kahng, Y. Koutis, B. Pramanik and Z. Wang, "K-SpecPart: Supervised Embedding Algorithms and Cut Overlay for Improved Hypergraph Partitioning", *arXiv*, 2305.06167, 2023. https://arxiv.org/abs/2305.06167

[7] J. Chen, I. H.-R. Jiang, J. Jung, A. B. Kahng, V. N. Kravets, Y.-L. Li, S.-T. Lin and M. Woo, "DATC RDF-2019: Towards a Complete Academic Reference Design Flow", *Proc. ICCAD*, 2019, pp. 1–6.

[8] J. Chen, I. H.-R. Jiang, J. Jung, A. B. Kahng, V. N. Kravets, Y.-L. Li, S.-T. Lin and M. Woo, "DATC RDF-2020: Strengthening the Foundation for Academic Research in IC Physical Design", *Proc. ICCAD*, 2020, pp. 1–6.

[9] J. Chen, I. H.-R. Jiang, J. Jung, A. B. Kahng, S. Kim, V. N. Kravets, Y.-L. Li, R. Varadarajan and M. Woo, "DATC RDF-2021: Design Flow and Beyond", *Proc. ICCAD*, 2021, pp. 1–6.

[10] C.-K. Cheng, A. B. Kahng, H. Kim, M. Kim, D. Lee, D. Park and M. Woo, "PROBE2.0: A Systematic Framework for Routability Assessment from Technology to Design in Advanced Nodes", *IEEE Trans. on CAD* 41(5) (2022), pp. 1495–1508.

[11] C.-K. Cheng, A. B. Kahng, S. Kundu, Y. Wang and Z. Wang, "Assessment of Reinforcement Learning for Macro Placement", *Proc. ACM/IEEE Intl. Symp. on Physical Design*, 2023, pp. 158–166. [See also: https://arxiv.org/abs/2302.11014.]

[12] V. A. Chhabria, *personal communication*, September 2023.

[13] V. A. Chhabria, K. Kunal, M. Zabihi and S. S. Sapatnekar, "BeGAN: Power Grid Benchmark Generation Using a Process-portable GAN-based Methodology", *Proc. ICCAD*, 2021, pp. 1–8.

[14] 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", *Proc. IEEE Symposium on VLSI Technology*, 2021, pp. 1–2.

[15] S. Choi, J. Jung, A. B. Kahng, M. Kim, C.-H. Park, B. Pramanik and D. Yoon, "PROBE3.0: A Systematic Framework for Design-Technology Pathfinding with Improved Design Enablement", arXiv:2304.13215 [cs.AR], 2023.

[16] J. Jung, I. H.-R. Jiang, J. Chen, S.-T. Lin, Y.-L. Li et al., "DATC RDF: an academic flow from logic synthesis to detailed routing", *Proc. ICCAD*, 2018, pp. 1–4.

[17] J. Jung, I. H.-R. Jiang, J. Chen, S.-T. Lin, Y.-L. Li et al., "DATC RDF: an open design flow from logic synthesis to detailed routing", *Proc. Workshop on Open-Source EDA Technology*, 2018, pp. 1–4.

[18] J. Jung, I. H.-R. Jiang, G.-J. Nam, V. N. Kravets, L. Behjat and Y.-L. Li, "OpenDesign Flow Database: The infrastructure for VLSI design and design automation research", *Proc. ICCAD*, 2016, pp. 42:1–42:6.

[19] J. Jung, A. B. Kahng, R. Varadarajan and Z. Wang, "IEEE CEDA DATC: Expanding Research Foundations for IC Physical Design and ML-Enabled EDA", *Proc. ICCAD*, 2022, pp. 1–8.

[20] J. Jung, P.-Y. Lee, Y. Wu, N. K. Darav, I. H. Jiang, V. N. Kravets et al., "DATC RDF: Robust design flow database", *Proc. ICCAD*, 2017, pp. 872–873.

[21] A. Kahng, A. B. Kahng, H. Lee and J. Li, "PROBE: Placement, Routing, Back-End-of-Line Measurement Utility", *IEEE Trans. on CAD* 37(7) (2018), pp. 1459–1472.

[22] G. Karypis, R. Aggarwal, V. Kumar and S. Shekhar, "Multilevel hypergraph partitioning: applications in VLSI domain", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 7(1) (1999), pp. 69–79.

[23] D. Kim, S. Y. Lee, K. Min and S. Kang, "Construction of Realistic Place-and-Route Benchmarks for Machine Learning Applications", *IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems* 42(6) (2022), pp. 2030–2042.

[24] R. Liang, A. Agnesina, G. Pradipta, V. A. Chhabria and H. Ren, "CircuitOps: An ML Infrastructure Enabling Generative AI for VLSI Circuit Optimization", *Proc. ICCAD*, 2023.

[25] R. Liaw, E. Liang, R. Nishihara, P. Moritz, J. E. Gonzalez and I. Stoica, "Tune: A Research Platform for Distributed Model Selection and Training", *arXiv preprint*, 2018.

[26] A. Mirhoseini, A. Goldie, M. Yazgan et al., "A Graph Placement Methodology for Fast Chip Design", *Nature* 594 (2021), pp. 207-212.

[27] K. E. Murray, S. Whitty, S. Liu, J. Luu and V. Betz, "Titan: Enabling large and complex benchmarks in academic CAD", *Proc. Intl. Conf. on Field Programmable Logic and Applications*, 2013, pp. 1–8.

[28] M. Papermaster, "Advancing EDA Through the Power of AI and High-Performance Computing", *opening keynote* at ACM/IEEE Design Automation Conference, July 2022.

[29] D. Prasad, S. S. T. Nibhanupudi, S. Das, O. Zografos, B. Chehab, S. Sarkar, R. Baert, A. Robinson, A. Gupta, A. Spessot, P. Debacker, D. Verkest, J. Kulkarni, B. Cline and S. Sinha, "Buried Power Rails and Back-side Power Grids: Arm® CPU Power Delivery Network Design Beyond 5nm", *Proc. IEEE International Electron Devices Meeting*, 2019, pp. 19.1.1–19.1.4.

[30] J. Ryckaert et al., "Extending the Roadmap Beyond 3nm through System Scaling Boosters: A Case Study on Buried Power Rail and Backside Power Delivery", *Proc. Electron Devices Technology and Manufacturing Conference*, 2019, pp. 50–52.

[31] S. Sebastian, H. Tobias, G. Lars, A. Yaroslav, S. Christian and S. Peter, "High-quality Hypergraph Partitioning", *ACM Journal of Experimental Algorithmics* 27(1.9) (2023), pp. 1–39.

[32] The OpenROAD Project (GitHub). https://theopenroadproject.org and https://github.com/The-OpenROAD-Project/OpenROAD

[33] OpenROAD-Flow-Scripts, https://github.com/The-OpenROAD-Project/OpenROAD-flow-scripts

[34] Hypergraph Partitioning for VLSI: Benchmarks, Code and Leaderboard, https://github.com/TILOS-AI-Institute/HypergraphPartitioning.

[35] TritonPart: an Open-Source Constrainyts-Driven Partitioner, https://github.com/The-OpenROAD-Project/OpenROAD/tree/master/src/par.

[36] IEEE CEDA Design Automation Technical Committee. https://ieee-ceda.org/node/2591

[37] Artificial Netlist Generator. https://github.com/daeyeon22/artificial_netlist_generator

[38] BeGAN Power Grid Benchmakrs. https://github.com/UMN-EDA/BeGAN-benchmarks

[39] MacroPlacement. www.github.com/TILOS-AI-Institute/MacroPlacement

[40] "Stronger Baselines for Evaluating Deep Reinforcement Learning in Chip Placement", August, 2022. https://statmodeling.stat.columbia.edu/wp-content/uploads/2022/05/MLcontra.pdf

[41] The PROBE3.0 Framework. https://github.com/ABKGroup/PROBE3.0

[42] CircuitOps. https://github.com/NVlabs/CircuitOps

[43] ML for IR drop. https://github.com/ASU-VDA-Lab/ML-for-IR-drop

[44] Bespoke-silicon-group bsg_fakeram. https://github.com/jjcherry56/bsg_fakeram

[45] NSF Workshop on Shared Infrastructure for Machine Learning EDA, March 2023. https://sites.google.com/view/ml4eda/home

[46] IEEE CEDA DATC RDF Calibrations. https://github.com/ieee-ceda-datc/datc-rdf-calibrations

[47] ASAP7 PDK and Cell Libraries. https://github.com/The-OpenROAD-Project/asap7

[48] "Statement of Interest in Fine-grained Physical Design Automation Tools for Digital 3D Integrated Circuits", DARPA-SN-23-74 *Special Notice*. https://sam.gov/opp/9ab902cff6854afb996ee8ca4c5ffc36/view