

OpenROAD in 2024: Expanding adoption, fostering ecosystems, improving PPA

2024 was a big year for OpenROAD! We focussed on improving tool quality and robustness, supporting educational and outreach efforts, and enabling our customers to gain significant advantages with OpenROAD in early stages of chip design. With continued focus on improving QoR and efficiency in its tools and its native flow OpenROAD-Flow-scripts (ORFS), OpenROAD updated existing tools across the flow through better algorithms, integration and an overall enhancements in the tool optimization capabilities—these resulted in significant improvements in quality of results and runtimes. This review looks back on the year's highlights, from growing the community, supporting adjacent open-source chip design ecosystems and achieving scale in chip-design education.

Sustainable development model with ecosystem support

OpenROAD's core mission is to drive silicon innovation by making chip design fast – RTL-GDSII in < 24 hrs, with tools that are easy to deploy, train and apply, without license and computational barriers across a wide range of applications. While the DARPA funding for OpenROAD ended on December 31, 2023, ongoing tool development has been on track targeting critical project goals of PPA improvement and feature support for a rapidly growing user community and ecosystem partners. [Precision Innovations Inc.](#) has expanded its team of experts to execute on these goals offering flexible paid-support models to ensure long-term sustainability and success of OpenROAD and its users.

Open-source based conferences such as [GoIT](#), [ORConf](#) and [LatchUp](#) and [Birds-of-Feather](#) at DAC foster collaborative thought leadership that identify gaps, trends, challenges and advances in open-source technologies thought by domain experts and leaders that are aligned with regional goals for semiconductors and workforce development. One such outcome led to a concerted effort to drive recommendations and a [roadmap](#) based on Europe's key strategy based on a vision for semiconductor and research innovation and capacity building for long-term economic success.

Strategic Government Initiatives and funding across Europe, India and Latin America led to a targeted focus on building educational and research programs in chip design combining open-source and commercial tools and flows. The European commission's core mission for semiconductor expansion hinges on open-source based innovation: *"Chips are strategic assets for key industrial value chains. With the digital transformation, new markets for the chip industry are emerging such as highly automated cars, cloud, Internet of Things, connectivity, space, defence and supercomputers."*

The project's commitment to support our partners globally is critical to the vision and successful execution of these roadmaps. Precision Innovations is a committed collaborator and maintainer of OpenROAD, fostering a growing and diverse community by delivering software, addressing user inquiries, and providing pre-built binaries for easier deployment.

Expanding educational outreach and workforce development

IHP announced a free [MPW program](#) using OpenROAD as one of the tools for its chip-design flow. This marks another milestone in enabling a new foundry that provides a lab-fab pathway for students, educators and industry prototyping applications using the IHP-SG13G2 BicMOS technology supported by the OpenROAD flow as an openPDK. This technology enables digital, analog as well as mixed-signal applications at 130nm for a wider range of applications including RF.

ETH Zurich standardized a course based on the OpenROAD flow for its chip-design courses. They are building a [certificate course](#) funded by the German Federal Ministry of Education and Research to help participants build chips using OpenROAD with IHP. These OS-EDA courses under development at IHP target researchers and career development programs.

In India, IITG announced a strategic education initiative as part of a 5-year skilling program targeted at students and industry innovators using OpenROAD for chip design— an average of >5000 students are expected to train annually as part of a PAN India program supported by the Government of India through NSDC and in partnership with Precision Innovations. In Jan 2024, an introductory workshop for students and faculty members flagged off this important initiative.

“India’s push for self-reliance in Electronics System Design and Manufacturing (ESDM) faces a critical challenge: an acute shortage of skilled manpower that is ready for the workforce. While commercial EDA tools may be available, their high costs and complexity limit access, support and training at scale. OpenROAD offers a cost-effective alternative, enabling advanced VLSI design research and educational opportunities for a wide spectrum of students, researchers, educators and industry innovators. IIT Guwahati conducted a workshop in Jan 2024 supported by Precision Innovations Inc. that introduced OpenROAD to > 200 participants including college students and faculty members. In Dec 2024, around 3300 students from VTU, Belagavi, Karnataka, India were trained as part of a 4-week extensive training program based on OpenROAD for digital design. Additionally, a faculty development program was also conducted on OpenROAD in November 2024, which was attended by over 50 faculty members. Supported by certification programs from India’s Ministry of Skill Development, this initiative strengthens India’s path to self-sufficiency and highlights the potential of global collaborations in advancing research and innovation. Our future programs involve open-source based design flows including OpenROAD to achieve the strategic goal of Viksit Bharat 2047 (Developed India 2047).”, states Prof. Gaurav Trivedi, Professor, EEE Dept., IIT Guwahati who is a key champion of using OpenROAD for education and research in India.

The impact of such strategic programs leveraging the unique advantages of open-source tools ensures a progressive and a sustainable form of education and workforce programs in the burgeoning semiconductor industry worldwide.

Tapeout-centric design flow for education and prototyping

An estimated total of 5000+ students completed training in OpenROAD through multiple educational forums like university programs and the popular TinyTapeout program powered by open-source tools like OpenROAD. These programs have accelerated education in chip design by enabling students to build designs and tape out on publicly supported foundries for a wide range of applications at a fraction of a time it would take using commercial tools. Product based courses such as RISC-V and open-source based workshops have expanded pathways into previously non-existing learning modalities.

[TinyTapeout](#), the popular educational program for chip-design, hosted several onsite workshops in the U.S.—we were proud to host one in San Diego!



Key Technology Updates

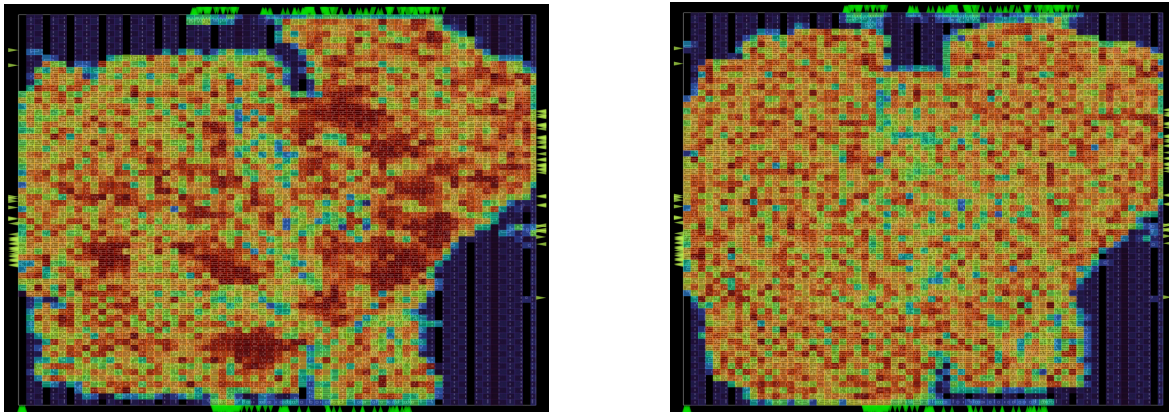
Several new features and algorithmic changes to core tools were made along with tighter integration across analyses and optimization which continuously improved PPA and runtime.

Resizer integration into the Global Placer

Integrating the **Resizer** into the **Global Placer (GPL)** in OpenROAD-Flow-Scripts (ORFS) streamlines the design flow by enabling timing-aware placement, optimizing cell resizing and buffering during placement itself. This reduces iterations for optimizing timing between placement and global routing, improves wirelength and congestion management, and enhances power, performance, and area (PPA) trade-offs. The integration also results in a more seamless and efficient placement-based timing and downstream congestion management resulting in better QoR and runtimes.

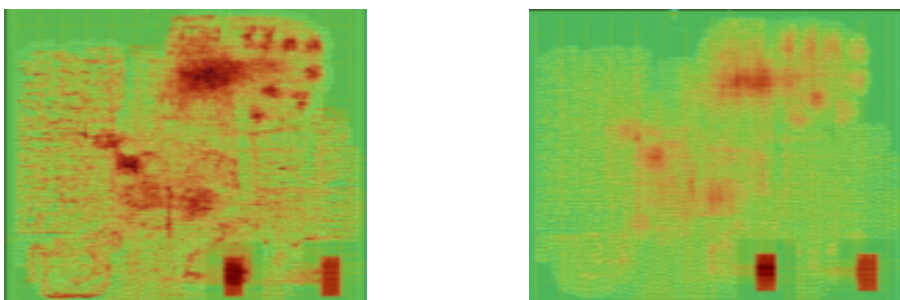
The integration of the Resizer into GPL eliminates redundant virtual iterations of costly repair_design optimizations and subsequent design database updates that were present in

the previous GPL timing-driven version. This enhancement resolves the placement density hotspots that occurred after GPL execution in earlier versions. Furthermore, the implementation was meticulously designed to ensure that integrating the resizer with GPL introduces no runtime overhead. Figure below shows placement densities before and after the integration showing a more uniform distribution of hotspots for ibex, skywater 130HS.



Routability-driven Global Placement using RUDY

The global placer now includes [RUDY](#) (Rectangular Uniform wire DensitY) as a replacement to Fastroute-based routability estimation, providing faster congestion analyses during placement. RUDY assesses the routing demand of nets and the net bounding box area to estimate routing congestion in placement regions to guide the global placer and improve subsequent optimizations. This significantly improves runtime in GPL for routability analysis to identify hotspots. This further removes a bottleneck problem of GPL resulting in faster global placement with fewer local congestions. RUDY runtimes are faster as compared to FastRoute with lower accuracy level but sufficient for predicting hotspots and thereby guiding subsequent routing. For instance, in the Nangate45/Swerv design, FastRoute takes approximately 8 seconds to provide a routing congestion estimation, whereas RUDY requires only 0.2 seconds. Moreover, QoR experiments demonstrate equivalent results during detailed routing, indicating that RUDY significantly reduces runtime without any impact on QoR. Figure below shows similarity in the heatmaps using FastRoute and RUDY for hotspot detection—RUDY is faster and operates at a higher level of abstraction without loss of accuracy to effectively guide the global router.



For details on Global Placement enhancements in both timing and routability-driven placement refer to the [README](#). Note that `-routability_use_grt` allows the mode as RUDY

(default) or FastRoute. To choose the overflow value to retain resizer changes past virtual iterations use ***-keep_resize_below_overflow***.

Improvements to Pin placement

Improved constraint support includes enhanced handling of mirrored pins for abutment. The annealing-based pin placer which is more effective at managing constraints, will become the default in 2025.

The pin placement tool now provides meaningful warnings and error messages with actionable feedback to help users address issues. Features like pin constraints for specific regions benefit from this enhancement, as the tool assists in determining optimal region sizes and guides users accordingly.

Pin constraints are now stored in the database, making them accessible to other tools in the flow. This design is especially beneficial for macro placement, which can utilize constraint regions instead of fixed pin positions, offering greater flexibility and improving integration with other tools.

The mirrored pins feature has seen significant improvements. It ensures not only mirrored (x, y) locations but also matching routing layers. The tool also supports mirrored pins within region constraints, providing users with more customization options for pin placement.

Lastly, a new solver has been added to the pin placement tool. Alongside the Hungarian Matching technique, a Simulated Annealing solver is now available via the ***-annealing*** flag. This solver supports region constraints, mirrored pins, and other features of the Hungarian Matching method and will become the default solver in 2025.

Macro Placement (MPL2) Enhancements

The tool has been significantly improved to enhance flexibility and performance. Key updates include improvements to macro placement across the chip periphery, better partitioning using PAR and pin access resulting in a more optimized floorplan. Support for interconnected macro arrays allow compact placement due to improvements in the clustering engine, floorplan centralization and the management of simulated annealing to improve QoR.

Memory management has significantly improved due to automation, fixes to memory leak cases and also data structure enhancements for greater efficiency. Collectively, these features now improve placement of complex macros (such as arrays) resulting in better QoR for hierarchical design.

MPL2 will become the default macroplacer (MPL) in 2025.

Support for multi-height cells

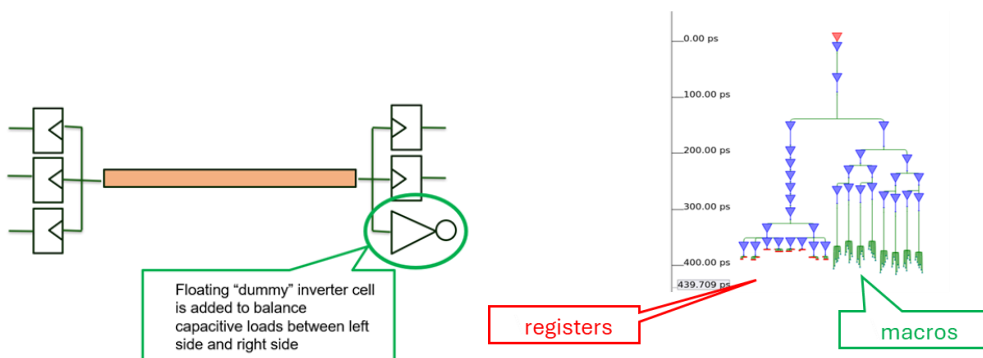
Multi-height cells are critical to enable designers to achieve high performance while trading off power and area. The OpenROAD flow now supports multi-height cell rows for variable drive strength as needed for complex functionalities like clock buffers and large multiplexers.

Support for multi-bit Flip-flops (MBFF)

As **MBFF** combines multiple single-bit flip-flops into a single, larger cell to enable sharing of common resources like the clock driver and enable signal, which makes them more efficient than using individual single-bit flip-flops. The ORFS flow now supports MBFFs to support energy efficient designs with optimal resource utilization while minimizing clock tree power consumption and reducing area.

CTS Enhancements

CTS was significantly enhanced to improve performance through specific algorithmic, methodology updates and integration to Yosys. This includes arithmetic operator selection support using the Kogge-Stone algorithm and advanced ABC9 recipes, leading to better performance and timing quality of results. Optimization enhancements have resulted in up to a 45% improvement in final setup total negative slack (TNS). Key optimizations include the addition of a new buffer removal transform in addition to buffering, pin swapping and gate cloning to repair_timing and an overhauled hill-climbing algorithm that leverages ODB save and restore techniques that also eliminates logic errors during optimization. Other improvements focused on reducing clock skew and latency by enabling max fanout fixing and dummy load insertion by default. Notably, enhanced clock latency computation has reduced the need for delay buffers while improving clock skew. Separate clock trees are built for macros and registers to improve skew and latency.



Designers now have greater flexibility with the ability to specify LVT or SVT libraries for clock buffers to further manage performance and power tradeoffs. Overall, these improvements have resulted in approximately 45% improvement in TNS on tested designs.

Finally, preserving topology-preserving route guides further improves post-optimization convergence.

Improved Antenna repair

Current enhancements to antenna violations repair support both diode insertion along with the newly added layer assignment changes resulting in fewer diodes and better repair results.

In addition to post-global routing fixing, where five iterations of antenna repair are performed, a post-detailed routing fixing stage was added, also running five iterations. Antenna violations are evaluated after detailed routing and repaired through diode insertion near the instance with the violation followed by incremental global routing and detailed routing on the nets showing violation. This approach gives near zero antenna violations for almost every design in the OpenROAD flow.

Layer assignment changes are applied after global routing by modifying the routing and adding jumpers above the violation segment by taking into account routing resources and any pre-existing vias at the applied locations. This approach reduces both the number of diodes and total antenna violations across all designs. For sky130hd, diode use decreased by 38%; for sky130hs, by 58%; and for ihp-sg13g2, by 63%, with a 53% reduction in total violations for the latter.

Post grt/drt timing repair with incremental routing

Timing optimization used to occur only after CTS. This meant that a lot of uncertainty around routing would have to be accommodated by adding extra guardband to the repair. OpenROAD can now repair timing again after global and detailed routing. This allows for guard band reduction and improved results. Both routers support incremental routing updates.

Improved convergence in detailed routing

The detailed router has a number of enhancements that combine to reduce routing time. The router can converge in fewer iterations as a result. Detailed routing is typically the most runtime intensive step in the flow. We were able to reduce detailed routing runtime significantly by cutting down unnecessary computations during routing and DRC checking. This had a reasonably huge impact on the detailed routing runtime of the gf12 designs. Moreover, we incorporated a new strategy for tackling the remaining small number of design rule violations at late iterations by trying out different routing parameters in parallel for each cluster of DRVs and choosing the best results. This reduced the number of detailed routing iterations across our test cases significantly.

Improved RDL routing

RDL routing between pad drivers and bumps is an important part of chip-level IO. This requires a specialty router that can handle 45° segments. Peter Gadfort rewrote this router from a simple pattern based router to a more capable router.

GUI Improvements

Improved performance of the slack histogram. Users can now click a bar and see the corresponding paths.

The clock tree viewer now distinguishes between flops and macros. The insertion delay of a macro is also displayed.

A repaint in-progress indicator was added. The repaint is interruptible on zoom/pan making it more responsive. Implant layers can now be drawn which helps with multi Vt designs.

Large design flow enhancements

The ability to rapidly explore ASIC microarchitectures for key design components early in the design flow is essential for achieving success, including fast design turnaround, good QoR, and a predictable path to production-quality designs. Moreover, the capability to build and test large designs with sensitive PPA components remains a critical objective. These advanced capabilities are now enabled through the integration of ORFS and Bazel, through the [bazel-orfs](#) workflow which delivers essential features of bazel such as artifact sharing, parallel, reproducible builds, and remote execution to ORFS. [bazel-orfs](#) complements the AutoTuner by enabling detailed analysis of specific datapoints and automating graph generation for result analysis, while the AutoTuner focuses on automatically finding optimal parameters for a target function. Together, they provide a wider range and scope for design exploration in the OpenROAD flow.

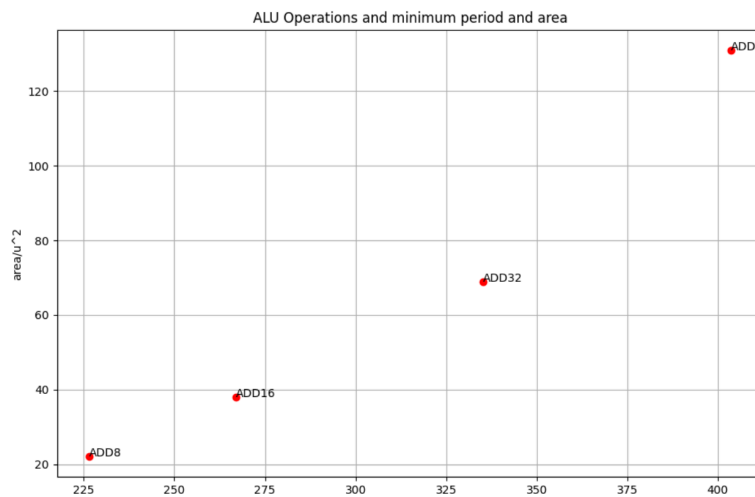
Efficient Design Exploration

[Ascenium](#), a pioneering user of OpenROAD, leverages this efficient workflow to push the boundaries of ASIC design exploration. By combining an exhaustive RTL-based design exploration with A/B analysis, they are able to make decisive architectural choices for implementation to achieve good QoR. This innovative approach enables Ascenium to build high-performance, energy-efficient general-purpose accelerators at reduced cost and with faster turnaround times. For example, Ascenium can easily model and study how PPA of its CPU ALU bypass networks and register file scales compared to traditional out-of-order CPUs and thereby validate earlier architecture choices.

Also, Ascenium used a large design test case, [Megaboom](#) –a RISC-V based SoC from Chipyard, which serves as an ideal test case for their needs. Megaboom’s robust testbed supports PPA trade-off analysis and provides insights into optimizing out-of-order (OoO) microarchitectures for performance and energy-efficiency consistent with final production goals. Ascenium used flow variables to sweep and explore the potential design space along with the recently enhanced metrics reporting to effectively analyze impact on design metrics like placement density and routing layer adjustment to optimize the physical design.

As an example, the following graph illustrates the analysis results showing area and performance impact of various ALU operations. Such operations can be efficiently automated and analyzed for good decision-making using bazel-orfs.

Fig below shows 8,16,32 and 64 ADD operations examined on Sky130HD and ASAP7.



For additional examples on design exploration such as algorithmic exploration refer to <https://github.com/The-OpenROAD-Project/OpenROAD/discussions/3881>. Such exploration can be far more effectively automated with Bazel-ORFS than with ad-hoc makefiles and Python scripts.

Streamlined collaboration for efficient development and test

Artifact sharing further streamlined collaboration within the OpenROAD development community. The workflow improves design management efficiency by providing access to design artifacts—like ODB files, logs, and scripts—without requiring time-consuming ORFS rebuilds or reruns. Developers can focus on specific tasks, while collaborators can rerun only the relevant sections of the flow—this significantly improves testing, debugging, and overall development processes.

Øyvind Harboe, V.P of Engineering at Ascenium, affirms: "*OpenROAD produces results that agree with the SOTA innovations to managing OoO CPU scaling requirements for PPA when doing A/B comparisons of RTL design choices. We are confident that the tool is giving us an excellent basis for making decisions leading in from the architecture as we move into production. We are definitely having success using bazel-orfs and proceeding with*

architectural exploration of RTL beyond tweaking flow parameters and design sweeps. When exploring the design space we run hundreds of variants of RTL and ORFS builds --this is time-consuming and also impossible for manual results analyses. ORFS and OpenROAD are perfect, because there is no concern about scaling up the exploration due to licensing or computational barriers. bazel-orfs also avoids unnecessary rebuilds during development of the architectural exploration cases thanks to the excellent artifact sharing features which makes our workflows highly efficient. OpenROAD today is easily able to scale to 1mm² designs in ASAP7 and bazel-orfs has a built-in flow to easily create mock, aka fake, SRAMs needed in architectural exploration. Recently OpenSTA combined with Verilator also allows detailed power reporting for real workloads from simulation. The OpenROAD QoR is accurate enough at a higher level of design abstraction such that negative results can be trusted: if it can't be made to work in OpenROAD, then the RTL needs work.”

While ongoing strategies to improve QoR in flow stages like macro placement, clock tree synthesis, and global placement continue to yield ever improving results, OpenROAD and open source PDKs today are already effective and successful in doing architectural exploration and comparing design choices.

Looking Ahead: 2025 and Beyond

As OpenROAD advances into 2025, we remain committed to expanding our capabilities and ecosystem support. We will enhance support for hierarchical design through the Verilog hierarchy and SDC constraints on hierarchical ports for large designs. To meet the growing demands of enterprise-scale chip design, we will add **support for RHEL and OpenSUSE**, ensuring broader compatibility and seamless deployment.

We are also closely tracking advances in **AI, Generative AI (GenAI), and ML-driven methodologies** that leverage open-source tools to drive productivity, innovation, and automation in chip design. Additionally, our roadmap includes **support for 3DIC integration** and other cutting-edge **physical design and packaging technologies**, positioning OpenROAD at the forefront of modern semiconductor design.

As we move forward, we extend our gratitude to our project team members, users, and ecosystem partners for their invaluable support. With renewed momentum, we look forward to an exciting and transformative year ahead!