# Enhancing EDA Physical Synthesis Workflows with najaeda for the RISC-V Ecosystem

Christophe Alexandre[1] and Noam Cohen[1]

[1]keplertech.io

**Abstract**

*The **najaeda** project provides a robust framework for developing post-synthesis Electronic Design Automation (EDA) algorithms. This presentation will explore the capabilities of **najaeda**, focusing on its data structures, APIs, and practical applications in improving EDA workflows within the RISC-V ecosystem. We will demonstrate how **najaeda** can be utilized to enhance netlist exploration, perform Engineering Change Order (ECO) transformations, and prototype EDA ideas efficiently, addressing the unique needs of the RISC-V community.*

## Introduction

The rise of RISC-V open source hardware has opened new opportunities for small and medium-sized businesses, enabling them to enter an industry traditionally dominated by large semiconductor companies. However, one major barrier remains: the high cost of proprietary EDA tools. To fully realize the potential of open-source hardware, the RISC-V ecosystem needs open-source EDA solutions.

Recent developments like Yosys and OpenROAD have pushed open-source EDA forward, but they lack the capacity to handle large-scale industrial-grade designs efficiently. The **najaeda** project represents an effort in this direction, focusing on high-capacity hierarchical netlist processing. **najaeda** has already delivered promising results in large-scale netlist cleaning and optimization.

Beyond scalability, **najaeda** also aims to modernize the EDA experience by promoting a software-driven approach. Unlike traditional Tcl-based workflows, **najaeda** uses Python, enabling a more flexible and accessible environment for software engineers and hardware designers alike. By integrating with modern data science and AI tools like Pandas and PyTorch, **najaeda** is paving the way for next-generation, open-source EDA workflows tailored for industrial RISC-V designs.

## Closing the Gap with najaeda: Open-Source EDA for Industrial-Scale RISC-V Designs

Over the past decade, open-source EDA has made remarkable progress in breaking into a traditionally closed and proprietary semiconductor industry. Projects like Yosys, OpenROAD, and KLayout have opened new opportunities, particularly for small and emerging players, enabling them to access and explore IC design flows without the financial barriers of proprietary tools.

However, for open-source EDA to truly compete with commercial alternatives, it must overcome two critical challenges:

1. Access to Modern, Open PDKs: Although efforts such as IHP (130nm) and GlobalFoundries GF180MCU (180nm) have expanded availability, there is still a significant gap in support for advanced process nodes. Without access to more modern PDKs, open-source EDA will struggle to support cutting-edge designs.
2. Scalability for Large, Industrial-Scale Designs: Most existing OSS EDA tools focus on smaller designs, and, as a result, scalability has not been a priority. Some key building blocks in the OSS EDA ecosystem cannot handle modern high-capacity designs, limiting their viability for large-scale RISC-V architectures.

To be a credible alternative to proprietary EDA solutions, open-source tools must evolve beyond small-scale projects and embrace industrial-level capabilities.

The **najaeda** project is designed to help close this gap, providing high-capacity industrial-grade data structures and tools that can efficiently support current and future RISC-V architectures.

**najaeda** is built on three core principles:

- Ease of Installation & Use: Traditional EDA tools are notoriously complex to install and use. **najaeda** simplifies this with a Python-based API, avoiding the traditional Tcl-based experience and making EDA more accessible to students, software engineers, and new contributors. Installation is straightforward with a simple `pip install najaeda`, eliminating the need for complex setup procedures.
- High Capacity & Performance: **najaeda** introduces collaborative data structures that separate

responsibilities. One layer stores structured user data, while another provides a simplified, flat data representation optimized for high-speed, multi-threaded algorithms. Efficient communication between these layers ensures fast processing without breaking the hierarchy or discarding critical data.

- Data Fidelity & Precision: EDA algorithms and tools should only modify the data they need, ensuring that hierarchies remain intact and that no unnecessary data is discarded simply because it is irrelevant to a particular algorithm. The original design structure is preserved while enabling powerful transformations.

By focusing on scalability, usability, and data integrity, **najaeda** aims to become a key enabler of industrial-scale open-source EDA - empowering the RISC-V community to develop competitive, high-performance designs without the constraints of proprietary tools.

## Experiment: Hierarchical Design Optimization with najaeda

In open-source flows, such as Yosys, applying hierarchical synthesis often disables cross-boundary optimizations. To address this limitation, **najaeda** can perform optimizations like Dead Logic Elimination and Constant Propagation without destroying the hierarchy, utilizing optimized uniquification techniques.

In this experiment, we interleave **najaeda** between Yosys and OpenROAD. In Yosys, we apply the option to keep the complete hierarchy.

The following RISC-V designs were utilized in this study:

1. BlackParrot [1]: A Linux-capable, cache-coherent, 64-bit RISC-V multicore processor designed to be the default open-source accelerator host.
2. Basilisk [2]: An end-to-end open-source, Linux-capable RISC-V System-on-Chip (SoC) developed by ETH Zürich and fabricated using IHP's open 130 nm technology.
3. MegaBOOM [3]: One of the largest open-source RISC-V chips.

The results presented in Table 1 demonstrate the effectiveness of **najaeda**'s Dead Logic Elimination (DLE) and constant propagation optimizations in netlist cleaning. For the Basilisk and MegaBOOM designs, synthesis was performed using Yosys with the 'keep hierarchy' option enabled, preserving the design's hierarchical structure. In the case of BlackParrot, we utilized the synthesis script maintained by the OpenROAD team, which selectively determines instances to flatten or retain based on specific criteria.

**Table 1:** *najaeda optimization results*

| Design | Initial Primitives | After najaeda Primitives | Logic Reduction |
|---|---|---|---|
| Basilisk | 950256 | 811302 | -14,62% |
| blackparrot | 302530 | 250181 | -17,30% |
| megaboom | 3871705 | 3042874 | -21,40% |

## Conclusion

The initial results highlight the potential of netlist optimization within hierarchical design flows. While flattening the hierarchy during synthesis is a common workaround, it becomes impractical for large-scale, complex designs. To support scalability and efficiency, EDA flows must evolve to maintain design data, ensuring the preservation of RTL structure and hierarchy, avoiding unnecessary data loss. Additionally, retaining structural information is critical for downstream tools such as placement and routing, enabling more efficient physical design implementation. Integrating these flows with AI-driven frameworks further enhances design optimization by allowing data-guided decision-making.

By addressing these challenges, **najaeda** contributes to the advancement of open-source EDA, offering a scalable, high-performance alternative for handling modern RISC-V architectures efficiently. Future work will focus on enhancing AI-assisted optimizations and expanding hierarchical flow capabilities to further bridge the gap between open-source and industrial EDA solutions.

## References


[1] D. Petrisko, F. Gilani, M. Wyse, D. C. Jung, S. Davidson, P. Gao, C. Zhao, Z. Azad, S. Canakci, B. Veluri, T. Guarino, A. J. Joshi, M. Oskin and M. B. Taylor. "BlackParrot: An Agile Open Source RISC-V Multicore for Accelerator SoCs". In: *IEEE Micro Special Issue on Agile and Open-Source Hardware* (July 2020).

[2] Paul Scheffler et al. *Basilisk: An End-to-End Open-Source Linux-Capable RISC-V SoC in 130nm CMOS*. 2024. arXiv: 2406.15107 [cs.AR]. URL: https://arxiv.org/abs/2406.15107.

[3] Jerry Zhao et al. "SonicBOOM: The 3rd Generation Berkeley Out-of-Order Machine". In: *Fourth Workshop on Computer Architecture Research with RISC-V* (May 2020).