Towards a More Intelligent OpenROAD: Proposals for Advancing Workflow

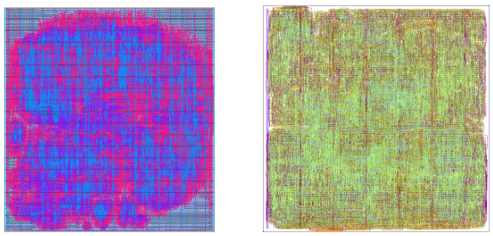
Yifei Zhu, Xinze Wang, Guohua Yin
RIOS Lab, Tsinghua-Berkeley Shenzhen Institute, Tsinghua University
yifei.z@rioslab.org

***Abstract*—This proposal compares the OpenROAD-flow-scripts (ORFS) with proprietary EDA softwares, identifying its limitations and potential future optimizations. Our focuses are improving chip area consumption (performance) through algorithm integration and speeding up the chip convergence process with user experience enhancement (runtime). Besides, we also plan to conduct a comparative study of the 7nm *CNFET7* PDK with the *ASAP7*. We aim to continue offering novel approaches to OpenEDA field prioritizing agility and innovation.**

***Index Terms*—OpenROAD-flow-scripts, FlowTune, Agile Development, EDA, OpenRoad contest**

I. BACKGROUND

The adoption of public toolchains has significant implications for the future of semiconductor innovation. Just as the use of open source compilers such as *GCC/LLVM* has revolutionized the software development industry, the optimization capabilities in OpenEDA have the potential to transform the hardware landscape. By leveraging the advantages of open source projects, EDA licenses no longer serve as a bottleneck to design space exploration (DSE), allowing for greater flexibility and collaboration among designers. The ORFS further enhances this process by integrating a comprehensive suite of tools, allowing for the efficient execution of the entire backend design process[1]. In our previous work, we have harden our 64-bit RISC-V CPU "GreenRio"[2] with both OpenROAD and proprietary EDA tools. Figure 1 displays the GDS generated by the two respectively. Specific statistics is shown in table I, we can generally conclude that open EDA tools have higher potential. Because under the same clock frequency, the area optimization performance has reached 60% of that of proprietary ones.



(a) Layout generated by OpenLane (b) Layout generated by Commercial EDA

Fig. 1. GDS generated by OpenEDA and proprietary softwares

Moreover, we delved into other functionalities for comparative analysis. ORFS can be improved in the following aspects: bolstered SystemVerilog syntax support, optimized Placement and Routing (PnR) algorithms; enhanced Logic Equivalence Check (LEC), efficient Antenna Violation fixing; power, performance, and area(PPA) optimization strategy, and multi-thread acceleration.

In the following section, we will elaborate our optimization focuses in this OpenROAD contest.

TABLE I
COMPARISON BETWEEN OPENROAD AND PROPRIETARY EDA TOOLS

Timing Corner: sky130_fd_sc_hd_tt_025C_1v80
Timing Closure: No hold & setup violation

	OpenROAD	proprietary EDA	Gap
synthesis run Time	6m12s	4m04s	1.5X
gate count	53K	33K	1.6X
placement & routing time	1h58m	43m	2.7X
die area(mm^2)	2.02	1.24	1.6X
leakage power	209nW	152nW	1.4X
placement density	32%	45%	1.4X
best clock period	80MHz	110MHz	1.4X

II. IMPROVEMENTS

A. Area Reduction

As mentioned above, due to the differences of synthesis strategies, *Yosys* and proprietary tools produced netlists with varying gate counts. To mitigate this issue, we plan to integrate *FlowTune*[3] to ORFS for area reduction. This framework takes advantage of domain-specific knowledge of DAG-aware synthesis algorithms to optimize standard-cell technology mapping and end-to-end PnR assessment using various backend tools, which will provide the netlist with at least 5% area reduction. We aim to add each mode of *FlowTune* in ORFS, including its AIG nodes/level minizaiton, area minimization before/after technology mapping, STA optimization etc. However, the challenges are: *FlowTune* may not support relatively complex design; and open-source synthesis tools like Yosys/ABC in ORFS may not be compatible with this structure; even the library file of *ASAP7* in ORFS is not complete enough. Nevertheless, We are trying to handle these problems, and we will continue to focus on this project.

B. Solution-inspiring Error Log System

Secondly, the emergence of OpenEDA tools such as ORFS, along with platforms like *OpenMPW*, has enabled the development of an efficient "lab2fab" research process for the open-source silicon community. We have observed that the "24 hours, no human" strategy in ORFS has expedited the chip hardening process. Nevertheless, due to the extensive design exploration space, multiple iterations and

trials are required, which present challenges to beginners and software engineers.

To alleviate these issues, we gather a comprehensive list of common user problems that arise during OpenROAD program execution on the Github issue page. We will embed potential solution hints in the error logs, which serve to inspire users and provide more direct debugging assistance, as well as a better understanding of the tool.

As part of our ongoing efforts, we are modifying the log generation mechanism in OpenROAD source code, adding solutions for each errors. We hope this solution-inspiring error log generation system will enable users to achieve a quick convergence in silicon hardening.

C. PDK porting

Lastly, we aim to make more open-source PDK libraries compatible with OpenROAD so that the open-source mode can be adapted to a wider range of scenarios. However, we encountered some challenges in using OpenROAD with CNFET-OCL, which is also a 7nm PDK.

While Yosys can synthesize circuits with CNFET-OCL[4], but OpenROAD requires extra files such as *.rc and *.track for placement and routing and a *.rcx file for static timing analysis. To address this challenge, we are working with the CNFET-OCL team to port *CNFET7* to OpenROAD. We believe that this effort will enable more open-source PDK libraries to be compatible with OpenROAD, and ultimately enhance the accessibility and usefulness of the tool set for digital circuit designers

III. CONCLUSION

In conclusion, the use of open-source EDA tools such as ORFS has the potential to transform the semiconductor industry by providing greater flexibility and collaboration among designers. In this proposal, we have identified the limitations and potential future optimizations of ORFS, with a focus on improving chip area consumption and speeding up the chip convergence process. By integrating tools such as FlowTune and developing a solution-inspiring error log system, we aim to enhance the user experience and agility of ORFS. Additionally, we plan to conduct a comparative study of the 7nm CNFET7 PDK with the ASAP7 to further explore the capabilities of ORFS. As the open-source silicon community continues to grow, we believe that the development of novel approaches and innovations in OpenEDA will be crucial to advancing the industry as a whole.

REFERENCES

[1] Ajayi T, Blaauw D. OpenROAD: Toward a self-driving, open-source digital layout implementation tool chain[C]//Proceedings of Government Microcircuit Applications and Critical Technology Conference. 2019.

[2] Zhu Y, Yin G, Wang X, et al. GreenRio: A Modern RISC-V Microprocessor Completely Designed with A Open-source EDA Flow[J].

[3] Neto, Walter Lau, et al. "FlowTune: End-to-end Automatic Logic Optimization Exploration via Domain-specific Multi-armed Bandit." IEEE

[4] Shi C, Miwa S, Yang T, et al. CNFET7: An Open Source Cell Library for 7-nm CNFET Technology[J]. 2023.