

# Experience of Using OpenROAD Flow Scripts on a RISC-V core in 7 nm

1<sup>st</sup> Kevin Pizarro

*Electronic Department*

*Universidad Técnico Federico Santa María*

*Valparaíso, Chile*

kevin.pizarroa@sansano.usm.cl

**Abstract**—In this paper, I present my experience of using OpenROAD Flow Scripts (ORFS) on a RISC-V core. ORFS is a set of integrated scripts that allow for RTL-to GDSII flow using open-source tools. The OpenROAD Flow project aims for automated, no-human-in-the-loop digital circuit design with 24-hour turnaround time. I discuss the benefits and challenges of using ORFS and provide recommendations for future projects.

**Index Terms**—OpenROAD Flow Scripts, RTL-to-GDSII flow, open-source tools, automated design, no-human-in-the-loop.

## I. INTRODUCTION

OpenROAD Flow Scripts (ORFS) is a powerful toolset that enables full RTL-to-GDS flow using open-source tools. The OpenROAD Flow project aims to automate digital circuit design with no human intervention and achieve a 24-hour turnaround time. In this paper, I describe my experience of using ORFS on a RISC-V core project and highlight the benefits and challenges of this approach.

## II. DESIGN FLOW OVERVIEW

First of all, the OFRS repository [1] will be forked for the modifications, as a goal I will achieve the Best Possible Runtime throughout data structures and algorithms in C++, rewriting some scripts in the flow and analyzing the time as output of the ORFS logs.

One example of the optimization that I'm looking for is to restructure the counting and search algorithms. The digital design have a lot of components, so in that direction is really important to take care of the small improvements in the complexity of the classical algorithms, in that way will scale up to better performance.

Another topic that I will looking for improve will be the designs description. Modifying the verilog or tcl scripts taking care of the critical path in the design, using techniques like pipelining or using the resources efficiently with tools/system configurations as the iterations or multi-threading.

## III. EXPERIENCE USING ORFS

As a first experience with the ORFS I would like to say that the toolset is intimidating, you can modify a lot of parameters and play around it with no directions, hopefully with the guide of the documentation [2] and the material [3] it looks more easy to learn and contribute to the community. Nevertheless, I

have experience with analog open source tools like XSchem, Magic VLSI, Netgen, Laygo2 and also I have experience in the digital design flow of Xilinx tools.

## IV. CONCLUSIONS AND RECOMENDATIONS

Given my background in electronics, the knowlegde on IC design, open sorce tools, data structs and algorithms will be a contribute to the community of the Open-ROAD project. My personal recommendation as a newbie in the ORFS environmet is that you should see the VSD-IAT videos, read the documentation, ask as many questions you have in the Slack channel and put your hands-on. The best way to improve any skill is to put it in practice as much as you can.

## REFERENCES

- [1] OpenROAD Project Team, "GitHub ORFS." <https://github.com/The-OpenROAD-Project/OpenROAD-flow-scripts> (accessed Mar. 19, 2023).
- [2] OpenROAD Project Team, "Documentation ORFS." <https://openroad-flow-scripts.readthedocs.io/en/latest/> (accessed Mar. 19, 2023).
- [3] VLSI System Design Team, "VSD-IAT." <https://vsdiat.com/> (accessed Mar. 19, 2023).