Infrastructure suggestions for OpenROAD Flow Scripts for faster user and developer debug experience

Mohit Sharma
Dept. of ECE
University of Windsor
Windsor, ON, Canada
mohit12562@gmail.com

***Abstract*—In this paper, I present infrastructure suggestions for improving debug experience of users of OpenROAD Flow Scripts (ORFS). Infrastructure suggestions include (i) splitting the top-level Makefile into individual config.mk files such as synthesis.mk, placement.mk, routing.mk, setup.mk; (ii) Creating separate folders for the OpenROAD flow steps containing the related config.mk file as well as related scripts; (iii) Add ability to dump custom flow script which contains only the relevant tool calls based on user provided options and default configs; (iv) Script-based (TCL) run environment where a top-level TCL script can be sourced to run the flow.**

***Keywords*—OpenROAD Flow Scripts (ORFS), ORFS Infrastructure, Debug experience, 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. In this paper, I provide suggestions for improving debug experience of using ORFS on a specific design project and highlight the benefits and challenges of this approach.

II. Current Flow Script Organization Overview

Currently, the ORFS scripts are organized as a single top-level Makefile and a single scripts folder. This makes management of different scripts easier with respect to version control as the number of directories and files are less. However, with a humongous top-level Makefile with only distinction between flow steps and configuration variables as text headers, it makes the debug of a failure, a time-consuming process as the user needs to parse a larger file. Also, it hinders with quick-learning of the infrastructure for a new user. Further, since the flow scripts are in a single folder with no naming pattern or organization also makes quick-understanding of the infrastructure a challenge.

III. Suggested Infra updates

1) I suggest that the ORFS scripts and Makefile be organized as shown in Figure **??**. This will ensure that users are easily able to debug a flow-step as well as quickly get adept with the flow scripts.

(a) (b) (c)

Fig. 1: Figure reference for suggested infra updates

2) A feature should be added in the form of a script whose functionality would be to dump a custom script that would only include the commands required for the specific run based on the config parameters provided by the user and hence wouldn't contain any if conditions. For eg. a partial tcl script for synthesis is presented in Figure 1c.

Conclusion

In conclusion, my experience of using OpenROAD Flow Scripts on a specific design project has been largely positive. I recommend the use of suggested ORFS directory structure and custom tcl script generation for every run for future projects.

References

[1] Experience of Using OpenROAD Flow Scripts on a Specific Design, by K. Ghosh, I. Iyer, P. Ghuring [Online]. Available: https://www.openroaddesigncontest.org/wp-content/uploads/2023/03/SamplePaper1_ORFSExperienceOnUserDesign_IEEETemplat

[2] ORFS github repo, [Online]. Available: https://github.com/The-OpenROAD-Project/OpenROAD-flow-scripts/tree/master/flow