Tuning design performance with OpenRoad Flow Scripts

Bittu N
admin@diadatp.com

***Abstract*—This document presents my experience with obtaining and using OpenROAD Flow Scripts (ORFS) to generate a GDS from the ibex design RTL.**

I. Introduction

OpenROAD Flow Scripts (ORFS) is an ongoing effort to create a comprehensive, fully automated and open source RTL-to-GDS flow with quick turn around times as little as a day.

The flow is based on the OpenROAD unified application and a few other open source tools such as yosys and LSOracle. The OpenROAD unified application is itself based on over a dozen tools all developed under the DARPA IDEA program.

II. Experience of Using ORFS

The documentation for getting started with the flow was sufficient when combined with the videos released on the **vsdiat** platform. A showstopping bug with the lemon dependency had to be worked around. The website hosting the software had gone down and an alternative source had to be used instead.

I chose to use the Docker method as my OS while Linux based would not be compatible with either the apt or yum package managers used in **DependencyInstaller.sh**. Unfortunately the structure of the Dockerfile did not allow for cached build steps and every bug encountered in the bash scripts would result in starting the whole build from scratch.

In order to gain experience with the flow, the **ibex** sample design and asap7 PDK were used to generate a GDS. The flow was run on a machine with the following specifications:

CPU: Intel i3-5005U @ 1.900GHz
Memory: 12GiB
OS: NixOS 22.11

The ORFS flow successfully generated a GDS for the ibex design in a little under 1 hour. The synthesized design had 22373 standard cells, a die area of $6166.33\mu m$ and a utilization of 45%. A significant portion of the flow was a single threaded workload with load averages around 1.0.

III. Conclusion and Recommendations

The ease with which I could obtain the tools and employ them to generate a GDS was refreshing. ORFS is a step in the right direction. The open source nature of the flow allowed for easy debugging of a showstopper bug and gave deep visibility into its execution. My next step will be to explore the AutoTuner. The single threaded nature of some parts of the flow can be used to our advantage by launching multiple exploratory runs, one on each core. The best fit can then be sent through the rest of the flow.

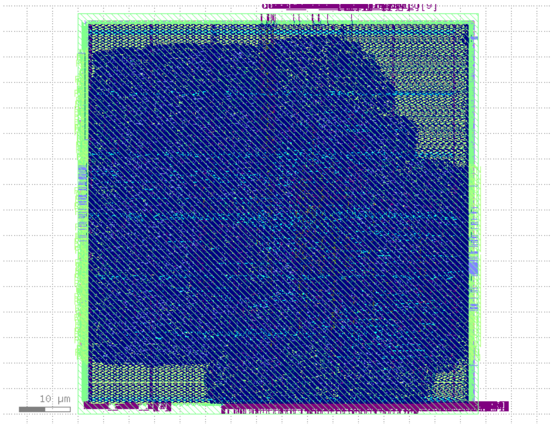

Fig. 1. ibex design GDS output using ORFS.

References

[1] "Welcome to OpenROADx2019;s documentation! x2014; OpenROAD documentation — openroad.readthedocs.io," https://openroad.readthedocs.io/en/latest/, [Accessed 19-Mar-2023].

[2] "Background on DARPA Electronics Resurgence Initiative x2013; The OpenROAD Project — theopenroadproject.org," https://theopenroadproject.org/news/background-on-darpa-electronics-resurgence-initiative/, [Accessed 19-Mar-2023].

[3] "GitHub - The-OpenROAD-Project/OpenROAD-flow-scripts: OpenROAD's scripts implementing an RTL-to-GDS Flow. Documentation at https://openroad-flow-scripts.readthedocs.io/en/latest/ — github.com," https://github.com/The-OpenROAD-Project/OpenROAD-flow-scripts, [Accessed 19-Mar-2023].