Decreasing the design area utilization in OpenROAD

*Kishan S Murthy*

*Electronics and Communication*

*RV College of Engineering*

*Bangalore, Karnataka, India*

kishansmurthy.ec20@rvce.edu.in

***Abstract*— OpenROAD is a new emerging technology which makes fab more easy to get the designs and it is much more advantageous to electronics engineers out there. Its main moto is no-human-in-the-loop. It automates each and every step (*RTL-to-GDSII) of asic design which humans used to do manually in the past. It has many PDK’s mainly asap7 (7nm pdk) which has many designs in it. Which can be used and we can create our own designs too.***

***Keywords— OpenROAD Flow Scripts, RTL-to-GDSII flow, no-human-in-the-loop.***

1. INTRODUCTION

In this paper I’ll explain my experience of using OpenROAD. Firstly I started with gcd design in asap7 PDK to understand the working flow and experimented gui and autotuner for this design also verified logs, reports and compared with metadata file. Then I tried to include some verilog design into Openroad and started off with MIPS processor which was very complex to give many errors. So, I moved on to some basic design and added 32 bit adder into OpenROAD which was successful.

1. DESIGN FLOW OVERVIEW

Ill try to explain the design flow of ORFS. The ORFS design flow includes 6 steps to be successful RTL to GDSII they are Synth, floorplan, place, CTS, route, finish. Which resembles the normal asic design flow. Tcl scripts are used to automate this flow. The platforms directory take care of the configuration files for the asap7 (7nm PDK) .Adding a design onto the designs directory and adding suitable design config on the makefile would lead to run a new design.

The Yosys tool synthesizes and tests the design if suitable testbench is provided, The Openroad tool take care of other flow steps which were mentioned above. At every step the detailed area utilization, power, timing, wns and other performance parameters can be noted from the logs. The metadata json file contains the information regarding the previous best PPA results obtained which had to be improved.

1. EXPERIENCE OF USING ORFS

Our Coming to my experience of using ORFS, It was a very good start to learn about vlsi and trying to improve its PPA would be a knowledgeable and a challenging work.

In my case I was trying to decrease the total area utilization of the design by changing some of the modelling styles in the design files. I included a 32 bit adder file into ORFS and started checking on it by using behavioral modelling and data flow modelling instead of structural modelling. As a result I found some important outputs which are mentioned in the conclusion and results section.

While in the process of exploring the tool I learnt many things infact I came to know the actual working of a ASIC flow, processor architecture including branch prediction(Ibex) which is my area of interest. After getting some of the useful results on 32 bit adder I tried to implement samething on riscv32i processor by changing some of the structural and dataflow modelling to behavioral modelling and yet to get the results.

I learnt the detailed ***RTL-to-GDSII*** flow, Gui in each step of the flow was very helpful to visualize and analyse the PPA, I got to know the use of Klayout, Autotuner, Jenkins and mainly Docker. Also I got to know the use of SSH keys in git. It would be helpful if I get access to cloud platform of openRoad for further studies.

1. CONCLUSIONS AND RESULTS

I referred to some of the papers mentioned below and openROAD webpage documentation was very helpful in knowing the actual tools available in openroad. It also had a mention of decreasing area utilization by making changes in design or tool would be fine. This is where I got an idea.

The area utilization when I used structural modelling was 18% and after changing it to behavioral modelling it was 16% with more of sequencial circuit rather than combinational circuit. This shows that using procedural assignment rather than continuous assignment statement would decrease the area occupied drastically in larger designs. I also verified the results in Autotuner which showed similar PPA other than area.

The tool really needs to be worked on the compilation time. I don’t know if this could be improved by using multicore processor involved desktops. The PPA available is okay as per my research in this field. but necessary changes should be made in decreasing the power by using techniques like clock gating/transition reduction and downsizing the non critical path circuits.

REFERENCES

1. Tutu Ajayi , Vidya A. Chhabria, “INVITED: Toward an Open-Source Digital Flow: First Learnings from the OpenROAD Project” DAC ’19, June 2–6, 2019, Las Vegas, NV, USA.
2. Andrew B. Kahng and Tom Spyrou, “The OpenROAD Project:  
   Unleashing Hardware Innovation” ..