Experience of using ORFS in ibex design using ASAP7 platform

Geetima Kachari
er.geetimakachari@gmail.com*

***Abstract*—In this paper, a brief overview on study of ORFS on a RISCV based ibex design is presented. ORFS is an opensource RTLtoGDS flow developed with the intention of providing an automatic, no human loop 24hr digital design implementation. This study is done as a part of 7nm design contest for obtaining the best possible design specification of ibex on a given system configuration. Here, the discussion highlights overall experience of usage of ORFS on a complex ibex design.**

***Index Terms*—ORFS, RTLtoGDS, ASAP7, ibex**

I. Introduction

ORFS is a collection of a number of opensource tool integrated together to provide a complete rtltogds flow on a digital design without any manual intervention in the process. This automatic ORFS flow targets at providing a cost effective, run time effective solution for a complete rtltogds flow part of chip designing with a 24 hour runtime in vision. The usage of opensource tools makes it easily available to all and provides a soultion in understanding and usage in digital design forms including academic purpose. This automatic flow can also be implemented at individual stage for understanding the PnR flow. ORFS also provides an interactive gui for viewing the output (db files) and analysing the different parameter eg drc violations and rectifying the errors. ORFS has another advantage of different platforms or PDK for exploration on the different designs.

II. Design flow experience

The OpenROAD flow scripts or ORFS provides a complete solution for the rtl to GDSII part of a digital design. The ORFS integrates the different tools using scripts and provide self driving, open-source digital design implementation tool chain. This tool set generates the different db output files, report file, log file, constraints file for each stage of the design flow and it also provides visualization of the layouts at each stage via its gui which contains all the necessary buttons and editor for analysing the design visually. A no human in the loop provides an easy pathway to the complete rtltogds flow part of a SOC design.

III. Experience on ORFS flow

To understand the flow initially I have used the gcd design with ASAP7 platform which executed hasselfree within 5min18sec generating allthe output files. However as contest required to use one of the provided RISCV based design I tried to execute ibex using the same platform I encountered slow execution during the detailed route stage. At this point I want to mention that I am using Ubuntu 18.04 with 4GB RAM, 4 Cpu core and 6GB swap area. Now, inorder to understand the behaviour of the system during the flow execution I monitored the system behaviour. Until the CTS stage the maximum cpu utilzation was 100% sometimes distributed among all the cores and vice versa. During the fast route stage the cpu usage reached its maximum around 398% and the flow till this point have taken around 30minutes to execute. As the flow proceed to the detailed routing stage the %cpu usage dropped to less than 10% with the %memory usage of approx 70% - 78% and it resulted in slow exectuion with almost 11 hrs taken for 0th optimzation iterations till 20% of the 1st optimization iteration.

IV. Conclusion and recommendation

Although the recommendation states requirement of 8GB RAM system the ORFS flow can be used for a 4GB system with certain adjustment done during the installation of the flow including increasing the swap area and shifting to xfce4 desktop environment. Irrespective of the slow execution of the detailed routing stage the flow continues and the major hiccup occured as the system shuts down, mostly due to power cut in the region. The execution next starts from the detailed routing stage which means another 11 hrs for the 0th optimization iteration stage and if the flow can be modified for execution from the exact point before the system shut downs it can reduce the repeative part runtime

References

[1] T. Ajayi et al., *INVITED: Toward an Open-Source Digital Flow: First Learnings from the OpenROAD Project*, 2019 56th ACM/IEEE Design Automation Conference (DAC), Las Vegas, NV, USA, 2019, pp. 1-4.
[2] *https://github.com/The-OpenROAD-Project/OpenROAD-flow-scripts*
[3] *https://openroad-flow-scripts.readthedocs.io/en/latest/tutorials/FlowTutorial.html*