Experience of Using OpenROAD Flow Scripts on Ibex Risc-V Core For best performance [f max]

*Note: Sub-titles are not captured in Xplore and should not be used

Anirudh Chokkasamudram
SOC Design Engineer
Intel Technology India Pvt Ltd,
anirudh.chokkasamudram@gmail.com

Abstract*— In this paper, I present my experience of using OpenROAD Flow Scripts (ORFS) to implement Ibex 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 digitalcircuit design with 24-hour turnaround time. I discuss the benefits and challenges of using ORFS and provide recommendations to carry out the , *RTL-to-GDSII flow.

Keywords— OpenROAD Flow Scripts, RTL-to-GDSII flow, open-source tools, automated design, no-human-in-the-loop,Ibex,risc-v

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, we present the implementation of the Ibex RISC-V core, which is a high-performance, low-power, and small-footprint implementation of the RISC-V instruction set The the Ibex risc-v core is implemented for best performance [fmax]

II. DESIGN FLOW OVERVIEW

The experience of using ORFS to implement the Ibex risc-v core has been positive overall. The use of open-source tools allows for flexibility and cost-effectiveness, while the automation of the design flow significantly reduces the time and effort required for the design process. I was able to achieve a design turnaround time of less than 24 hours, which is impressive considering the complexity of the design.

However, I also encountered some challenges during the design process. The goal is to implement the core with the best available performance, power and area targets.

This requires a proper consideration of design constraints, floor planning, macro-placement and clock tree synthesis to produce a DRC and LVS clean design. The timing reports are analyzed to identify the setup and hold violations and fixes are done based on available slack.

III. EXPERIENCE OF USING ORFS

The Ibex RISC-V core is a 32-bit pipelined processor that implements the RISC-V ISA. The Ibex core is highly configurable, allowing for customization of the pipeline depth, instruction set, and memory hierarchy. The Ibex core has a five-stage pipeline, which includes the following stages: instruction fetch (IF), instruction decode (ID), execution (EX), memory (MEM), and write-back (WB). The pipeline design of the Ibex core is optimized for high performance, low power, and small area footprint.

Proper design constraints are written, which enable optimization of the design for a given area. Since the turnaround time is about 24 hrs., multiple iterations can be carried out to identify the bottlenecks and issues in the design and arrive at an optimal recipe. OpenSTA is used to identify the timing violations and pipeline bottlenecks in the design. The OpenROAD GUI is used for floor-planning and analyzing heat map and congestion. The CTS GUI is used for visualizing and building a well-balanced clock tree.

The Ibex core was implemented using open-source tools - OpenROAD Flow Scripts (ORFS).The Ibex core was implemented using a 7nm ASAP Process design kit, and optimized for best achievable fmax.

The use of open-source tools requires careful consideration of tool compatibility and versioning, as well as potential issues with tool performance and stability.

IV. CONCLUSIONS AND RECOMMENDATIONS

In conclusion, OpenROAD Flow Scripts can be used to rapidly carry out RTL-to-GDSII flow, trying out different recipes to achieve best PPA for a given design. Having a very fast turnaround time of 24-hrs helps understand the design issues better and resolve issues proactively. Would certainly recommend ORFS and a silicon proven PDK to do quick implementations of the designs to achieve best performance

REFERENCES

[1] https://github.com/lowRISC/ibex
[2] https://ibex-core.readthedocs.io/en/latest/03_reference/index.html