2021 31st International Conference on Field-Programmable Logic and Applications (FPL) | 978-1-6654-3759-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/FPL53798.2021.00085

Taping out an FPGA in 24 hours with OpenFPGA: The SOFA Project

Xifan Tang, Ganesh Gore, Grant Brown and Pierre-Emmanuel Gaillardon
Laboratory for NanoIntegrated Systems (LNIS), University of Utah, U.S.A.
Email: xifan.tang@utah.edu

***Abstract*—This paper highlights the *Skywater Open-source embedded FpgAs* (SOFA) project, which is a series of open-source embedded FPGA IPs built with the Skywater 130nm technology. The SOFA project showcases an agile prototyping methodology for FPGAs, enabled by the OpenFPGA framework, whose fabrication-ready layouts are generated in 24 hours. We also present the associated Verilog-to-Bitstream toolchain for end users.**

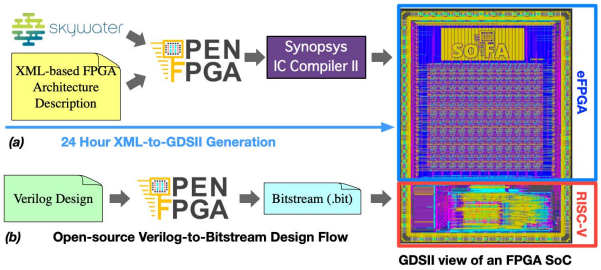

Fig. 1: Design flows for SOFA eFPGAs: (a) production flow and (b) end-user flow.

Introduction: FPGAs are notoriously difficult to prototype due to the massive design effort required by manual layouts, as well as the complexity in building the associated *Electrical Design Automation* (EDA) toolchain. The technical barriers result in not only a long development cycle for FPGA vendors when producing an FPGA chip, but also a unaffordable cost for researchers when attempting to develop a silicon proof for their architectural innovations. As a result, open-source FPGAs are never presented. However, recent years have seen the rising needs in *embedded FPGAs* (eFPGAs), as well as research on how to automate the design process of FPGA layouts. Open source tools have been developed to translate the FPGA architecture description to synthesizable/technology-mapped Verilog netlists, after which FPGA layouts can be produced by standard ASIC design flows. Empowered by one of those tools, namely, OpenFPGA [1], this paper presents the first open-source eFPGA IPs, namely, the *Skywater Open-source embedded FpgAs* (SOFA) [2]. FPGAs are built not only by open-source tools but also by an open-source *Process Design Kit* (PDK), offering easy access to users without any *Non-Disclosure Agreement* (NDA) [3]. The open-source nature brings a high level of transparency to end users, from the FPGA architecture details, down to the physical layouts, as well as the bitstream generation toolchain. More than just fabrication-ready layouts, SOFA is also a foundational project for researchers to perform realistic architecture exploration based on fabricable PDKs. The rest of the paper consists of tutorials demonstrating various aspects of the eFPGAs.

XML-to-GDSII Generation: This section focuses on a tutorial presenting how to generate an eFPGA layout from an XML-based FPGA architecture description file, using an automated design flow based on OpenFPGA, as depicted in Fig. 1(a). The FPGA architecture decription files are based on the well-known University of Toronto FPGA Architecture Language (UTFAL) [4] and the OpenFPGA architecture description language [5]. OpenFPGA is used to convert the FPGA architecture description to technology mapped Verilog netlists that model a complete FPGA fabric. In this tutorial, we focus on generating a standard cell FPGA fabric based on the Skywater 130nm High-Density cell library. Synopsys IC Compiler II performs the physical design of the FPGA fabric, where floorplanning, detailed placement, and routing are conducted. By the end of the design flow, timing sign-offs are applied using Synopsys PrimeTime and post-layout HDL simulations are performed to validate the correctness of the eFPGA IP. Note that the methodology is general purpose and hence portable to other technology nodes.

Verilog-to-Bitstream: All the SOFA eFPGAs are long-term supported by the open-source OpenFPGA framework, with which end users can implement HDL designs. As depicted in Fig. 1(b), OpenFPGA offers push-button design flows for end users to generate bitstreams for Verilog designs. Inside OpenFPGA, a custom version of Yosys [6] synthesizes the HDL codes to technology-mapped netlists, and VPR8 [7] performs physical implementation by packing, placement, and routing. OpenFPGA's bitstream generator analyzes VPR results and outputs a bitstream file. In addition, we present how to use the bitstream files to run HDL simulations before downloading them to FPGAs. Several example of HDL designs are available on the SOFA github repository, covering representative applications, such as counters and binary-to-BCD converters [2].

ACKNOWLEDGMENT

This material is based on research sponsored by Air Force Research Laboratory (AFRL) and Defense Advanced Research Projects Agency (DARPA) under agreement number FA8650-18-2-7855. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Air Force Research Laboratory (AFRL) and Defense Advanced Research Projects Agency (DARPA) or the U.S. Government. This project is also supported by a research gift from Google.

REFERENCES

[1] *The OpenFPGA project*, Github: https://github.com/lnis-uofu/OpenFPGA.

[2] *Skywater Open-source eFpgA*, Github: https://github.com/lnis-uofu/SOFA.

[3] *FOSS 130nm Production PDK*, Github: https://github.com/google/skywater-pdk.

[4] J. Luu, *A Hierarchical Description Language and Packing Algorithm for Heterogenous FPGAs*, University of Toronto, Master Thesis, July 2010 https://tspace.library.utoronto.ca/handle/1807/24601.

[5] *The OpenFPGA Architecture Description*, Github: https://openfpga.readthedocs.io/en/master/manual/arch_lang/.

[6] *Yosys*, Github: https://github.com/YosysHQ/yosys.

[7] *Verilog-to-Routing*, Github: https://github.com/verilog-to-routing/vtr-verilog-to-routing.

978-1-6654-3759-2/21/$31.00 ©2021 IEEE
DOI 10.1109/FPL53798.2021.00085