Extension and Evaluation of a Python-based High-level Synthesis Tool Flow

Department of Computer Science
Paderborn University

February 7, 2020

Supervisors: Prof. Marco Platzner
Lennart Clausing
Student: Vivek Jaganath, Master Computer Science

1 Background

Using a hardware/software co-design approach for accelerating applications is becoming increasingly popular, but the development of the hardware part is a challenging task in particular for a software developer. Therefore, high-level synthesis tools such as the Xilinx Vivado HLS [1] have been made available. Such tools improve design productivity by taking code written in a software programing language and generating hardware designs out of it.
The majority of current high-level synthesis tools support as programming language only C/C++. While this is an appropriate language for many hardware/software co-designs, in the fields of data analytics and machine learning Python is, however, more widely used. Generating hardware designs out of Python programs is a much more involved task than C/C++ high-level synthesis, since Python provides a multitude of language concepts and features which are difficult to realize in hardware.
A recently published attempt towards Python-based high-level synthesis is *Hot & Spicy* [1]. The presented tool flow first translates Python code for a subset of Python to a syntax tree, then creates HLS-ready C code from the syntax tree and, finally, runs a C-based high-level synthesis tool on the C code to generate the hardware accelerator.

2 Problem Definition

This thesis project has two main goals: The first goal is to gain an understanding of the state-of-the-art in Python-based high-level synthesis. This requires one to get an overview over Python-based synthesis approaches, including the recently presented *Hot & Spicy* tool flow, and to experimentally prove the functionality of such tools. The second goal is to extend the *Hot & Spicy* tool flow to include a larger subset of Python language concepts and syntax, in particular those needed for typical applications in data analytics and machine learning. After implementing the additional functionality in the *Hot & Spicy*

[1] https://www.xilinx.com/support/documentation-navigation/design-hubs/dh0012-vivado-high-level-synthesis-hub.html

tool flow, a demo application should be developed. The practical software/hardware implementation should use a Xilinx Zynq 7020 System-on-Chip on a Xilinx PYNQ board.

3 Subtasks

- Conduct a literature study on Python-based high-level synthesis approaches and available tool flows and compare the approaches with respect to their project status, supported subset of Python language concepts and syntax, etc.
- Familiarize yourself with the internal workings of the *Hot & Spicy* tool flow and how to use it, which includes familiarization of the required Xilinx tools. Evaluate the tool flow based on provided test functions.
- Propose extensions to the current state of the *Hot & Spicy* project in order increase the supported subset of language concepts, data types, etc. Focus your work on functionality typically required to implement applications in the fields of data analytics and machine learning.
- Implement the new functionality and develop a demo application for a hardware/-software co-design using Python-based high-level synthesis, showcasing the extended *Hot & Spicy* tool flow.
- Compare the extended *Hot & Spicy* tool flow to the commercial Xilinx Vivado HLS tool flow using the test functions and the demo application. Evaluate parameters such as the synthesis time and the resulting hardware area and speed.

4 Organizational Matters

- **Schedule:** At the beginning of your work, prepare a time schedule together with your advisor. Keep a continuous record of the progress of your work.
- **Regular meetings:** To keep track of your work's progress and to discuss difficulties or next steps, you should meet regularly (i.e., every one or two weeks) with your advisor.
- **Introductory presentation:** After you have familiarized yourself with the work and its objectives, you are expected to give a short introductory presentation of your task, the proposed solution and approaches, and your time schedule.
- **Final presentation:** At the end of your work, you are expected to give a final presentation covering the task and the objectives, the taken approaches and the achieved results.

References

[1] Sam Skalicky, Joshua Monson, Andrew Schmidt, and Matthew French. Hot & spicy: Improving productivity with python and hls for fpgas. In *International Symposium on Field-Programmable Custom Computing Machines*, Apr 2018.