High-Level Synthesis With LegUp: A Crash Course for Users and Researchers

Jason H. Anderson, Stephen D. Brown, Andrew Canis, and Jongsok Choi
Department of Electrical and Computer Engineering
University of Toronto
legup@eecg.toronto.edu

ABSTRACT

High-level synthesis (HLS) has been gaining traction recently as a design methodology for FPGAs, with the promise of raising the productivity of FPGA hardware designers, and ultimately, opening the door to the use of FPGAs as computing devices targetable by software engineers. In this tutorial, we introduce LegUp [1], an open-source HLS tool for FPGAs developed at the University of Toronto. With LegUp, a user can compile a `C` program completely to hardware, or alternately, he/she can choose to compile the program to a hybrid hardware/software system comprising a processor along with one or more accelerators. LegUp supports the synthesis of most of the `C` language to hardware, including loops, structs, multi-dimensional arrays, pointer arithmetic, and floating point operations. The LegUp distribution includes the CHStone HLS benchmark suite [2], as well as a test suite and associated infrastructure for measuring quality of results, and for verifying the functionality of LegUp-generated circuits. LegUp is freely downloadable at `www.legup.org`, providing a powerful platform that can be leveraged for new high-level synthesis research.

Categories and Subject Descriptors

B.7 [**Integrated Circuits**]: Design Aids

Keywords

High-level synthesis, FPGAs, hardware/software co-design

1. OVERVIEW

This tutorial is a crash course on LegUp, both as an HLS tool for designing hardware, and as a framework for new research on HLS algorithms and methodologies for hardware/software co-design. While understanding any new large software system can be a daunting task, this tutorial will lower the learning curve for LegUp and provide hands-on experience in using and modifying the tool.

We will begin with a general overview of the main HLS steps (allocation, scheduling, binding and HDL code generation) and then discuss the particular HLS algorithms used within LegUp. We will also overview the key compiler concepts needed to understand the LegUp implementation, such as the control dataflow graph. We then demonstrate the use of LegUp as a tool for the HLS of hardware-only and hybrid hardware/software systems, and describe the current capabilities and limitations of the tool. The HLS-generated hardware will be simulated with ModelSim and synthesized to an Altera FPGA. We will show how TCL parameters can be used to influence the HLS results, for example, by disabling or enabling resource sharing or the chaining together of computations in a single hardware clock cycle. TCL parameters that define the architecture of the hybrid processor/accelerator system will also be described.

The second part of the tutorial deals with the implementation of LegUp itself, and should be of interest to HLS researchers wishing to modify the tool. LegUp is implemented within the open-source LLVM compiler framework [3]. We will explain the key concepts of LLVM's intermediate representation (IR) of a software program and go over the basic LLVM programming idioms necessary for extending the LegUp codebase. We will describe the software architecture of LegUp, how it fits into LLVM, its internal data structures, and the software modules in which the key pieces of HLS functionality reside.

The last part of the tutorial will consist of demonstrations/labs that illustrate how LegUp can be modified on several fronts: 1) we will change the scheduling algorithm, which assigns operations from the `C` code to specific FSM states; 2) we will alter LegUp's approach to binding, which saves area by mapping several operations from the `C` to a single hardware functional unit; and 3) we will illustrate the steps needed to add support for new FPGA device architectures. Participants who have LegUp pre-installed on their laptops will be able to follow along with the demonstrations and try them out themselves during the tutorial.

2. REFERENCES

[1] A. Canis, J. Choi, M. Aldham, V. Zhang, A. Kammoona, J. Anderson, S. Brown, and T. Czajkowski. LegUp: High-level synthesis for FPGA-based processor/accelerator systems. In *ACM FPGA*, pages 33–36, 2011.

[2] Y. Hara, H. Tomiyama, S. Honda, and H. Takada. Proposal and quantitative analysis of the CHStone benchmark program suite for practical C-based high-level synthesis. *J. of Information Processing*, 17:242–254, 2009.

[3] LLVM. *The LLVM Compiler Infrastructure Project (http://www.llvm.org)*, 2013.

FPGA'13, February 11–13, 2013, Monterey, California, USA.
ACM 978-1-4503-1887-7/13/02.