Introduction to the Special Section on High-Level Synthesis for FPGA: Next-Generation Technologies and Applications

CHRISTIAN PILATO, Politecnico di Milano, Italy
ZHENMAN FANG, Simon Fraser University, Canada
YUKO HARA-AZUMI, Tokyo Institute of Technology, Japan
JIM HWANG, Xilinx, Inc., USA

Due to the end of Dennard scaling and Moore's law, heterogeneous System-on-Chip (SoC) architectures are replacing complex hyper-pipelined processors to achieve high performance and energy efficiency. Such architectures feature many specialized components that can be used to accelerate selected computational kernels by exploiting more intrinsic parallelism with custom logic. Among them, FPGA devices are becoming common targets for these systems since they allow fast turn-around time, field upgradability, and easy deployment of hardware/software solutions. However, co-designing FPGA systems still requires a combination of hardware and software design skills that are uncommon in most of the designers. To overcome these issues, designers need to raise the abstraction level from low-level manual designs to high-level approaches. High-level synthesis (HLS) is becoming a key enabling technology, especially for FPGA designs, since it allows designers to describe the functionality of a component at the software level and automatically generate the corresponding hardware description, enabling fast deployment of hardware/software solutions. HLS has been making tremendous progress in many application domains, ranging from Internet of Things and edge computing to data centers and cloud computing.

While HLS is becoming more and more popular, the other side of the coin is that it is pushing the application landscape for hardware acceleration towards unprecedented challenges. On one hand, modern applications must elaborate huge amounts of data, demanding efficient methods for managing memory accesses. On the other hand, HLS is a complex process that produces itself a huge amount of information that can be used to drive further optimizations. In both cases, machine learning is coming to the rescue to extract valuable knowledge and make accurate predictions.

In this special section, we have six articles covering both challenges (the first five articles) and application aspects (the last one). These articles show that HLS is a powerful but yet difficult-to-use solution. Indeed, many HLS tools offer *directives*, i.e., source code annotations that trigger specific optimizations, but understanding the optimal combination from a huge design space is still a manual and time-consuming effort. The articles in this special section provide interesting insights on how to automate this exploration process also with the help of machine learning. We hope you will enjoy them and find them interesting as we did.

The special section opens with an article on the compiler-level optimizations for optimizing the pointer synthesis within HLS. "A case for precise, fine-grained pointer synthesis in high-level synthesis" by N. Ramanathan et al. aims at reducing the gap between application designers, who could make heavy use of pointers to create compact and efficient software descriptions, and hardware designers, which demand precise memory information to implement the corresponding accesses to the storage resources. The proposed approach significantly reduces area resources and latency with precise and fine-grained pointer synthesis.

The next four articles discuss the use of machine learning for HLS optimization and design space exploration. "Correlated Multi-objective Multi-fidelity Optimization for HLS Directives Design" by Q. Sun et al. provides a

Authors' addresses: Christian Pilato, Politecnico di Milano, Milan, Italy, christian.pilato@polimi.it; Zhenman Fang, Simon Fraser University, Burnaby, Canada, zhenman@sfu.ca; Yuko Hara-Azumi, Tokyo Institute of Technology, Tokyo, Japan, hara@cad.ict.e.titech.ac.jp; Jim Hwang, Xilinx, Inc., San Jose, CA, USA, jhwang@xilinx.com.

1084-4309/2022/6-ART
https://doi.org/XXXXXXX.XXXXXXX

multi-objective optimization process that uses deep learning to determine the optimal HLS directives and their factors by correlating the *fidelities* of data between consecutive FPGA design stages. Indeed, the later stages provide more accurate reports but take longer running times. “AutoDSE: Enabling Software Programmers to Design Efficient FPGA Accelerators” by A. Sohrabizadeh et al. discusses the use of FPGA devices as accelerators in datacenters, showing that the optimization process is complex, application and tool dependent, and requires an intelligent process to guide the design space exploration and automatically apply code transformations. The authors use a bottleneck-guided coordinate optimizer on top of a source-to-source compiler to achieve up to almost 20× speedup over CPU implementations. The next two articles, “Sherlock: A Multi-Objective Design Space Exploration Framework” by Q. Gautier et al. and “Learning from the Past: Efficient High-Level Synthesis Design Space Exploration for FPGAs” by Z. Wang and B. Carrion Schafer, provide more efficient methods to perform design space exploration and select the proper optimization parameters with strategies based on the Multi-Armed Bandit problem and the identification of kernel substructures, respectively. Both approaches can optimize multiple objectives and provide better Pareto-optimal designs.

The last article, “High-Level Synthesis Implementation of an Embedded Real-Time HEVC Intra Encoder on FPGA for Media Applications” by P. Sjövall et al., is an application-oriented work. It describes how to use high-level synthesis for the implementation of video encoders in a cluster of network-attached FPGA cards, showing excellent design scalability. Also in this case, the efficient use of memory resources is one of the key factors for performance improvements.

We would like to thank both authors and reviewers for their valuable contributions. This special section would not be possible without their outstanding and timely work. We also hope that the achievements gathered in these articles will be valuable to the community.

Christian Pilato
Zhenman Fang
Yuko Hara-Azumi
Jim Hwang

Guest Editors

ACM Reference Format:
Christian Pilato, Zhenman Fang, Yuko Hara-Azumi, and Jim Hwang. 2022. Introduction to the Special Section on High-Level Synthesis for FPGA: Next-Generation Technologies and Applications. *ACM Trans. Des. Autom. Electron. Syst.* 1, 1 (June 2022), 2 pages. https://doi.org/XXXXXXX.XXXXXXX