Guest Editors' Introduction: Raising the Abstraction Level of Hardware Design

Philippe Coussy
Université de Bretagne-Sud

Andres Takach
Mentor Graphics

DESIGN COMPLEXITY is continually rising with the higher levels of integration implied by Moore's law. Functional complexity increases with our ability to incorporate more computation in SoCs and to create more complex applications. Additional complexity is also introduced in the design process by the need to control power consumption and to tackle challenges with respect to physical timing closure and process variations. Automation is required to manage all this complexity, allowing designers to focus their attention on high-level design decisions that have the most effect in the quality of the implementation. Raising the level of abstraction of hardware design is also required to enable an effective exploration of software and hardware architectures, making high-level synthesis (HLS) a cornerstone of *electronic system-level* (ESL) design.

High-level synthesis tools today take functional specifications in high-level languages and generate RTL or gate-level specifications that can be used in well-established RTL synthesis and place-and-route flows. The generated RTL is optimized to the specific target ASIC or FGPA technology. RTL specifications with a wide range of performance, area, and power characteristics may be generated from a single source specification making the input specification highly reusable. The behavior of the generated hardware can be verified against the high-level specification using either simulation techniques or sequential formal-equivalence checking. The productivity gain of using HLS is obtained not only by its speeding up hardware creation but also by its facilitating the verification of the generated hardware. The first article in this special issue, written by Philippe Coussy et al., presents an overview of basic HLS techniques and introduces the reader to industrial tools with more specific details on two representative leading commercial HLS tools.

High-level synthesis began almost three decades ago as an ambitious attempt to provide a path from an algorithm to a custom digital hardware implementation tuned for that algorithm. It is ambitious in that HLS needs to excel at optimizing the hardware architecture to be competitive with what a human designer is able to craft. Research interest in HLS peaked in the late 1980s and early 1990s, at which time much of the research attention shifted to hardware and software codesign and embedded system design. Nonetheless, research on HLS has continued at a steady pace with significant contributions made toward incorporating additional constraints such as power and process variability, optimizing communication between blocks, optimizing arithmetic expressions, and improving on traditional HLS optimizations.

Although the adoption of HLS has had a number of false starts, there are clear indications that HLS can no longer be dismissed as a "lab experiment." In fact, several hundred ASIC tape-outs have been completed with HLS being used for designing significant or differentiating portions of them. Many more FPGA designs have also been completed with the help of HLS. The second article in this issue, written by Grant Martin and Gary Smith, provides an excellent account of the history and evolution of HLS from research to its current level of industry adoption. There are many lessons to be learned from understanding the trajectory of HLS about the do's and don'ts of introducing design automation into real-world design flows. Their article also supplies market data that shows the significant growth that the HLS market has experienced over the past five years.

0740-7475/09/$26.00 © 2009 IEEE Copublished by the IEEE CS and the IEEE CASS

The third article presents a "roundtable" of industrial HLS users representing the various commercial HLS tools available on the market. They share their experiences with HLS including benefits they obtained from using HLS and areas of improvement they would like to see in the future. They provide a diversity of use scenarios including synthesis of complex wireless and multimedia applications, cryptographic applications, and buses and transactors.

The fourth article, written by Soujanna Sarkar et al., presents the authors' experiences with several anonymous commercial HLS tools. While they describe the benefits, they focus on some of the challenges including portability across tools, formal verification, and ECO flows.

The most common use of HLS is to generate hardware for applications that are highly complex and compute intensive in nature. Arithmetic optimization is a key aspect of HLS. The fifth article, written by Maciej Ciesielski et al., is focused on the optimization of initial specifications by using a canonical representation that enables exploring various factorizations. The authors demonstrate the applicability of their approach in the context of HLS. The sixth article, written by Sumit Ahuja et al., presents a case study for synthesizing co-processors for compute intensive algorithms. They start from an ANSI C specification and use their C2R methodology to explore architectures by restructuring the C-code in a form that expresses the desired architecture.

While performance and area remain primary optimization criteria, optimization for low power is increasingly important. High-level synthesis has a major role to play as it enables the rapid exploration of architectures to generate hardware that meets the performance requirements as well as minimize area and power. Also, HLS can be used to enhance clock gating opportunities and to evaluate the effect of many techniques used to minimize power such as the use of multiple V_{TH}/V_{DD} and power-gated technology libraries. The seventh article, written by Maria Molina et al., presents an original high-level synthesis approach that addresses the problem of dynamic power consumption in data-dominated applications. The proposed scheduling and binding algorithms deal with the switching activity information at the sub-word level.

A great challenge in deep-submicron SoC design is the increasing importance of process variability. Deterministic models become overly conservative and techniques that can optimize based on many corners and on statistical analysis are needed from HLS down to place-and-route. The additional complexity of taking into account process variability underscores the need for HLS, as traditional manual RTL methodologies may require many design iterations and lead to over design. The eighth article, written by Yuan Xie and Yibo Chen, surveys the recent progress made by virtue of process variability's being taken into account in HLS.

With the increasing cost of ASIC re-spins, verification is as much a design concern as ever. In fact, verification is the number one driver for using HLS, with the productivity gain for generating the hardware being a close second. By generating correct-by-construction hardware from a specification at a high-level of abstraction, HLS eliminates the many verification and debug cycles that are typical with manual RTL creation methodologies. The input specification serves as a golden model, and since it is expressed at a high-level of abstraction, it simulates much faster than an RTL specification, enabling a more thorough verification of its functionality. HLS also enables the rapid prototyping of a design on FPGAs to achieve a more comprehensive functional verification. Verifying the correctness of the HLS transformation can be done by comparing the functionality of the input specification to the generated RTL, with either simulation or by *sequential equivalence checking* (SEC). The ninth and final article in this special issue, written by Anmol Mathur et al., presents SEC and how it can be used in HLS flows. The article provides an overview of SEC techniques, its challenges, and successes in real-world designs.

IT IS OUR HOPE THAT the articles in this special issue will not only be informative but will generate interest in the use of HLS and its further development to tackle larger subsets of the design and of the design process. As the diversity of the articles indicates, there is a wide range of areas of research where additional work can provide enhanced capabilities: power optimization, verification, process variation and physical awareness, arithmetic optimizations, memory hierarchy optimizations, input specifications including models of computations, generation of transaction level model to facilitate system-level exploration and optimization, interface and protocol synthesis including interface to buses and software, complex IP-core reuse,

optimization of highly pipelined systolic hardware, generation of multimode architectures, processor-based or reconfigurable hardware, enhanced automated architectural exploration, generation of architecture with multiple clock domains, and synthesis of asynchronous hardware.

Generating optimized application-specific hardware is a critical component of system-level design to fulfill the performance and the power constraints of designs for years to come. High-level synthesis has, and will, increasingly play an important role in the design of SoCs. ■

Philippe Coussy is an associate professor in the Lab-STICC at the Université de Bretagne-Sud, France, where he leads the HLS research group. His research interests include system-level design and methodologies, high-level synthesis, computer-aided design for SoCs, embedded systems, and low-power design for FPGAs. He has a PhD in electrical and computer engineering from the Université de Bretagne-Sud. Coussy has organized several workshops and tutorials on HLS and is co-editor of *High-Level Synthesis: From Algorithm to Digital Circuit* (Springer, 2008). He is a member of the IEEE and the ACM.

Andres Takach is Chief Scientist at Mentor Graphics. His research interests include high-level synthesis, low-power design, and hardware-software codesign. He has a PhD from Princeton University in Electrical and Computer Engineering. He is chair of OSCI's Synthesis Working Group.

■ Direct questions and comments about this article to Andres Takach, 8005 S.W. Boeckman Road, Wilsonville, OR 97070-7777, andres_takach@mentor.com; and Philippe Coussy, Lab-STICC, Centre de recherche, rue de saint maude, BP 92116, Lorient 56321, France; philippe.coussy@univ-ubs.fr.

For further information about this or any other computing topic, please visit our Digital Library at http://www.computer.org/csdl.