Guest Editors' Introduction: Multicore SoC Validation with Transaction-Level Models

Prabhat Mishra
University of Florida

Sandeep Shukla
Virginia Tech

Zeljko Zilic
McGill University

■ **FUNCTIONAL VERIFICATION IS** widely acknowledged as a major bottleneck in today's SoC design methodology. In spite of such extensive efforts, many SoC designs fail at the very first use (silicon failures) as a result of functional errors. The complexity of functional verification is expected to increase further because of the combined effects of increasing design complexity and the recent paradigm shift from single-processor SoC designs to heterogeneous multicore SoC architectures. A significant bottleneck in the verification of such architectures is the lack of high-level modeling and validation techniques. Clearly, system-level modeling and validation is a promising approach to capture as many bugs as possible in the early stages of a design and thereby drastically reduce the overall validation effort, accelerating the time to market. Various transaction-level models (TLMs) are widely used for high-level modeling, evaluation, and exploration of SoC architectures.

This issue of *IEEE Design and Test* presents four special-theme articles that highlight challenges and recent trends of multicore architecture validation using transaction-level models. The articles cover theoretical as well as practical aspects related to high-level validation including transaction-level modeling of multicore architectures, validation and debug of TLM models, and industrial case studies. These articles were selected through a rigorous review process with each article receiving at least three reviews from industry and academic experts.

The first article, "Automatic TLM Generation for Early Validation of Multicore Systems" by Samar Abdi et al., presents efficient techniques for automatically generating transaction-level models that can be used for high-level validation of multicore systems. The authors' framework accepts applications (C tasks) as inputs mapped to processing units in the platform. Based on the mapping, the functional TLM is generated by instantiating the application tasks inside a SystemC model of the platform. For timed-TLM generation, the basic blocks in the application tasks are analyzed and annotated with estimated delays. The delay-annotated C code is then linked with a SystemC model of the hardware or software platform. Both TLMs can be natively compiled and executed on the host machine, making them much faster than conventional cycle-accurate models. TLMs of industrial-scale multicore designs—such as JPEG encoder, MP3 decoder, and H.264 decoder—are rapidly generated and simulated. Estimation error compared to board measurements has been shown to be small, making the TLMs ideal for early validation of multicore systems.

Traditional cooperative discrete-event simulation engines cannot effectively use the parallelism in multicore CPU hosts while simulating designs described using C-based system-level languages. The next article, "Multicore Simulation of Transaction-Level Models Using the SoC Environment" by Weiwei Chen et al., presents an extension of the discrete-event simulation engine to support multicore simulation of SpecC-based designs. It describes two important extensions—communication protection and optimized implementation. The utility of the approach is demonstrated using a case study on an

0740-7475/11/$26.00 © 2011 IEEE

Copublished by the IEEE CS and the IEEE CASS

H.264 video decoder and a JPEG encoder application. This approach can significantly reduce the simulation time of large transaction-level models at several abstraction levels.

The third article, "On MPSoC Software Execution at the Transaction Level" by Frédéric Pétrot et al., studies the challenges associated with software execution in multiprocessor SoC (MPSoC) environments. It describes a wide variety of solutions that can be used to perform transaction-level, hardware/software system simulation.

Many SoCs are designed using a globally asynchronous, locally synchronous (GALS) paradigm to solve timing and scalability issues. This design style introduces two fundamental challenges for debug: first, how to obtain a consistent state, and second, how to force the SoC to reach an erroneous state. The final article, "Interactive Debug of SoCs with Multiple Clocks" by Bart Vermeulen and Kees Goosens, presents an efficient debug approach that addresses these challenges. It uses temporal abstractions from clock cycles to communication handshakes and transactions as its key ingredient, complemented by scan-based state access, and guided replay. Experimental results demonstrate that this approach improves stability of the state bits captured via scan and the repeatability of the execution trace in a GALS SoC.

These articles provide a glimpse into various transaction-level modeling, simulation, and debug techniques practiced or proposed recently for multicore architectures. We sincerely hope that the four special-theme articles in this issue will provide interesting and useful insights to readers, and that it will also motivate readers to pursue further research in high-level design and validation of heterogeneous multicore architectures. ■

Acknowledgments

We thank all the authors and reviewers for their tremendous efforts and contribution in producing these high-quality articles. We also take this opportunity to thank *Design & Test* Editor-in-Chief Krishnendu Chakrabarty for his continued support and encouragement. Finally, the editorial staff of the IEEE Computer Society deserves special thanks for their wonderful job in editing and organizing these articles.

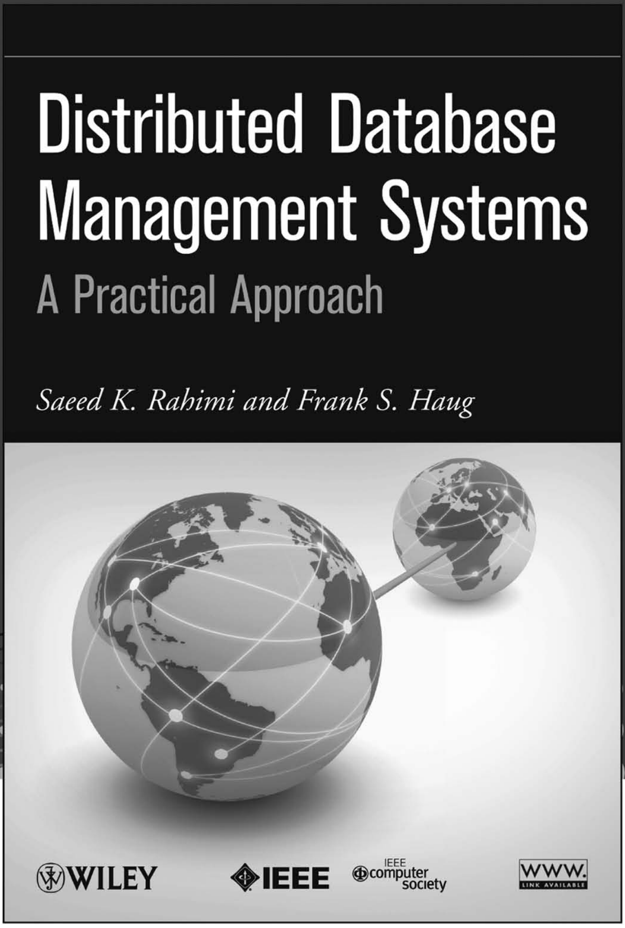
Prabhat Mishra is an associate professor of computer and information science and engineering at the University of Florida. His research interests include high-level validation, design automation of embedded systems, and low-power reconfigurable architectures. He has a PhD in computer science from the University of California, Irvine. He is a senior member of both ACM and IEEE.

Zeljko Zilic is an associate professor of electrical and computer engineering at McGill University. He conducts research on aspects of the quality-driven design of embedded heterogeneous microsystems. He has a PhD in electrical and computer engineering from the University of Toronto. He is a senior member of both ACM and IEEE.

Sandeep Shukla is an associate professor of electrical and computer engineering at Virginia Tech. His research interests include system-level design, formal verification, defect-tolerant computing, low-power design of systems, and embedded software synthesis. He has a PhD in computer science from the State University of New York at Albany. He is a senior member of both ACM and IEEE.

■ Direct questions and comments about this article to Prabhat Mishra, University of Florida, P.O. Box 116120, Gainesville, FL 32611; prabhat@cise.ufl.edu; or Zeljko Zilic, McGill University, 546 McConnell Engineering Bldg., 3480 University Street, Montreal, Quebec, Canada H3A-2A7; zeljko.zilic@mcgill.ca; or Sandeep Shukla, Virginia Tech, 302 Whittemore Hall, Blacksburg, VA 24061; shukla@vt.edu.

NEW TITLE FROM WILEY & IEEE CSPress



Distributed Database Management Systems

A Practical Approach

Saeed K. Rahimi and Frank S. Haug

Unique coverage of traditional database theory and current research for building easier-to-manage distributed database systems

The book covers three main units: theoretical foundations, the "state of the practice," and DDBMS implementation.

Supplemental resources for instructors and students are also available.

978-0-470-40745-5 • August 2010
Hardcover • 728 pages • $135
A Wiley-IEEE Computer Society Press Publication

To Order

North America
1-877-762-2971

Rest of the World
+ 44 (0) 1243
843291

Order Online
computer.org/store

wiley.com/ieeecs

15% Off for CS Members

WILEY
Publishers Since 1807

IEEE

IEEE computer society
www.computer.org