Guest Editors' Introduction: The Verification and Test of Complex Digital ICs

Magdy S. Abadir
Motorola

Li-C. Wang
University of California, Santa Barbara

To compete in the marketplace, all semiconductor products have tight time-to-market requirements. With design complexity exploding, functional verification is now on the critical path to RTL signoff and relies mainly on extensive vector simulation. A typical microprocessor requires billions of simulation cycles to tape out. But microprocessor manufacturers aren't the only ones concerned about the increasingly long simulation time. More and more, SoC and ASIC manufacturers face the same challenge. This special issue focuses on the important problem of functional verification and testbench generation for verifying complex digital ICs—a synergistic area that has significant impact on both the test and verification communities. The articles in this issue cover many different aspects of functional verification, including verification methodology, test program generation, case studies, design specification, and model checking.

The first article, by Allon Adir et al. (IBM Research Lab, Haifa), presents Genesys-Pro, a second-generation model-based testbench generator for functional verification of uniprocessors and multicore processors. Genesys-Pro, an improved version of its predecessor, Genesys, is currently the main functional verification tool for all IBM processors. In the 1980s, IBM was among the first to develop an automatic test program generator using a biased, random, and dynamic generation scheme, coupled with a traditional expert-system paradigm. This early test program generator was architecture dependent; thus, the need for a generic solution inspired the development of a model-based test generation scheme. A model-based test generator consists of two major components:

- a model describing the target architecture, and
- a generic constraint-solving engine for test generation.

Genesys-Pro introduces significant advancements in both areas. First, its grammar-based test template language has the same expressive power as a generic programming language, therefore maximizing the productivity of verification engineers. Second, its improved ability to translate test generation into a constraint-solving problem allows for a powerful and universal constraint-solving engine. The article summarizes the development of Genesys-Pro and provides an excellent overview of how this testbench generation tool evolved at IBM.

Next, Carl Scafidi of Intel, and Doug Gibson and Rohit Bhatia of Hewlett-Packard, describe a unit-level approach for verifying the Intel Itanium 2 processor's exception control unit, XPN. In full-chip functional verification, it's common for certain units, buried deep inside a chip, to receive less coverage than other units. To alleviate this problem, unit-level verification adopts a divide-and-conquer approach consisting of three components:

- a test generator to emulate all possible inputs from the surrounding units,
- a golden model to predict the unit's behavior, and
- a monitor program to check for possible errors.

In this article, the authors summarize their effort to verify XPN as an independent unit. They demonstrate that, even after the full-chip model neared tape-out quality, the unit-level verification approach uncovered 44 hardware bugs.

The third article, by Fulvio Corno et al. (Politecnico di Torino, Italy), presents a novel functional verification methodology based on test program evolution. The authors propose a scheme that involves creating a small initial set of test programs. The user can apply these test programs, and feedback from the simulation results feeds into an improved set of test programs. The feed-

0740-7475/04/$20.00 © 2004 IEEE Copublished by the IEEE CS and the IEEE CASS

back information comes from coverage measurements. In a typical functional verification environment, the verification engineer often carries out this feedback process manually. A mechanism that automates the feedback process represents a key advancement in functional verification.

The difficulty in implementing such a mechanism lies in the composition of the smaller, initial test programs. The authors use a genetic-algorithm-based evolution engine for this purpose. The flexibility of this approach allows for effective searching of the best tests for reaching a particular target. Basing their evaluation of the proposed methodology on a SPARC-V8-compliant Leon2 microprocessor, the authors demonstrate the superiority of the approach by comparing it to random test program generation. A detailed discussion of three interesting corner cases shows the effectiveness of code optimization and the capability of the evolutionary approach.

In the fourth article, Chia-Chih Yen and Jing-Yang Jou of National Chiao-Tung University, Taiwan, and Kuang-Chien Chen of Verplex propose another approach to automatic functional-test generation using state-of-the-art hybrid solvers. The fundamental idea is to divide a design into loosely coupled components that have limited interactions with each other, thus facilitating independent searches of each component.

The authors describe two key steps that support the proposed idea. First, they propose an automatic partitioning scheme to decompose a design based on state-holding elements. Second, the authors present a method that uses test generation results obtained from individual components to form the test for the entire design. They demonstrate that this new divide-and-conquer scheme can improve a solver's capacity for functional-test generation in every test case under study.

The fifth article, by Prabhat Mishra and Nikil Dutt (University of California, Irvine), and Narayanan Krishnamurthy and Magdy S. Abadir (Motorola), summarizes the effort to develop a top-down functional verification methodology. One key challenge that today's designers face is the lack of a formal specification for verifying the RTL model. Design intention is usually specified in workbooks written in a natural language such as English.

The authors propose using an architecture description language (ADL) to describe design intention at a higher level. An ADL model encapsulates all design properties that a corresponding RTL model must satisfy. During verification, designers can generate an assertion model from an ADL model and use it with a symbolic verifier to check the RTL design. Moreover, designers can produce a complete reference model to use in equivalence checking against the RTL model. The authors demonstrate the feasibility of this approach by verifying a memory management unit design in Motorola's high-performance microprocessor and in a 32-bit DLX microprocessor.

The final article, by Ganapathy Parthasarathy et al. (University of California, Santa Barbara), discusses the potential in applying sequential satisfiability (SSAT) to improve the capacity of bounded model checking. Model checking provides an ultimate guarantee in functional verification by exploring all relevant state spaces to prove or disprove a design property. It is commonly viewed as a supplementary methodology to testbench simulation for improving verification quality. Traditional bounded model checking relies on binary-decision-diagram- and combinational-SAT-based techniques to efficiently search the relevant state space. The authors suggest that SSAT is an alternative that provides a unique strength in its ability to support model checking. Introducing an SSAT solver can potentially improve a hybrid solver's capabilities for test generation as well as for model checking. The authors provide a detailed analysis of the strengths and weaknesses of using SSAT as opposed to using BDD and combinational SAT solvers.

THESE SIX ARTICLES provide an excellent overview of state-of-the-art approaches to functional verification and testbench generation. We hope you enjoy this special issue. ■

Magdy S. Abadir manages the High Performance Tools and Methodology Group at Motorola's PowerPC Design Center in Austin, Texas. He is also an adjunct faculty member of the Computer Engineering Department at the University of Texas at Austin. His research interests include microprocessor test and verification, test economics, EDA tools, and DFT. Abadir has a BS in computer science from the University of Alexandria, Egypt; an MS in computer science from the University of Saskatchewan, Saskatoon, Canada; and a PhD in electrical engineering from the University of Southern California. He has cofounded and chaired a series of international workshops on the economics of design, test, and manufacturing; and on microprocessor test and verification. A coeditor of several books on those

subjects, Abadir has published more than 120 conference and technical papers on test economics, DFT, computer-aided design, high-level test generation, and design verification and economics. He is a senior member of the IEEE.

Li-C. Wang is an assistant professor in the Department of Electrical and Computer Engineering at the University of California, Santa Barbara. His research interests include microprocessor test and verification, defect-oriented testing, ATPG SAT solver development, and statistical methods for speed test and validation. Wang has a BS in computer engineering from National Chiao Tung University, Taiwan, and an MS in computer science and a PhD in electrical and computer engineering, both from the University of Texas at Austin. He received best-paper awards from the 1998 Design Automation and Test in Europe (DATE) Conference, the 1999 IEEE VLSI Test Symposium, and DATE 2003 for his work on PowerPC verification, commercial experiments on novel ATPG methods, and delay defect diagnosis, respectively. He is cofounder and program chair of the IEEE Microprocessor Test and Verification (MTV) Workshop and is a member of the IEEE.

■ Direct questions and comments about this special issue to Li-C. Wang (University of California, Santa Barbara, Dept. of Electrical and Computer Engineering, Santa Barbara, CA 93106-9560; licwang@ece.ucsb.edu) or Magdy Abadir (Motorola, 7700 W. Parmer Ln. PL30, Austin, TX 78729; m.abadir@motorola.com).

For further information on this or any other computing topic, visit our Digital Library at http://www.computer.org/publications/dlib.

IEEE Design & Test Call for Papers

IEEE Design & Test, a bimonthly publication of the IEEE Computer Society and the IEEE Circuits and Systems Society, seeks original manuscripts for publication. *D&T* publishes articles on current and near-future practice in the design and test of electronic-products hardware and supportive software. Tutorials, how-to articles, and real-world case studies are also welcome. Readers include users, developers, and researchers concerned with the design and test of chips, assemblies, and integrated systems. Topics of interest include

- Analog and RF design,
- Board and system test,
- Circuit testing,
- Deep-submicron technology,
- Design verification and validation,
- Electronic design automation,
- Embedded systems,
- Fault diagnosis,
- Hardware/software codesign,
- IC design and test,
- Logic design and test,
- Microprocessor chips,
- Power consumption,
- Reconfigurable systems,
- Systems on chips (SoCs),
- VLSI; and
- Related areas.

To submit a manuscript to *D&T*, access Manuscript Central, http://cs-ieee.manuscriptcentral.com. Acceptable file formats include MS Word, PDF, ASCII or plain text, and PostScript. Manuscripts should not exceed 5,000 words (with each average-size figure counting as 150 words toward this limit), including references and biographies; this amounts to about 4,200 words of text and five figures. Manuscripts must be doubled-spaced, on A4 or 8.5-by-11 inch pages, and type size must be at least 11 points. Please include all figures and tables, as well as a cover page with author contact information (name, postal address, phone, fax, and e-mail address) and a 150-word abstract. Submitted manuscripts must not have been previously published or currently submitted for publication elsewhere, and all manuscripts must be cleared for publication.

To ensure that articles maintain technical accuracy and reflect current practice, *D&T* places each manuscript in a peer-review process. At least three reviewers, each with expertise on the given topic, will review your manuscript. Reviewers may recommend modifications or suggest additional areas for discussion. Accepted articles will be edited for structure, style, clarity, and readability. Please read our author guidelines (including important style information) at http://www.computer.org/dt/author.htm.

Submit your manuscript to *IEEE Design & Test* today!

D&T will strive to reach decisions on all manuscripts within six months of submission.

IEEE Design&Test of Computers