Guest Editor's Introduction

Design Automation

Hillel Ofek,* Silvar-Lisco

As a term, "design automation" denotes a discipline that involves the use of computers to aid in the design process for a variety of products. It is often used in association with computer-aided engineering, computer-aided design, computer-aided testing, and computer-aided manufacturing. In general, design automation includes a broad spectrum of application fields, ranging from electronic (analog and digital) systems to mechanical systems and many more. In this premiere issue of IEEE Design & Test of Computers, we examine design aids used in the development of digital electronic products.

Specifically, the topics covered in these pages were chosen to represent technical activities that are growing in importance as we enter the VLSI era. All six feature articles are based on papers originally presented at the 20th Design Automation Conference.

Our lead article, "Improving Color CAD Systems for Users," by Francine Frome, is destined to become a landmark piece of work. It deals with some very important aspects of engineering workstation design and human-machine interaction.

Frome is reporting on one of only a few attempts to actually use a scientific approach to a subject in which goodness is difficult to measure. Her article reports on studies based on concepts drawn from cognitive psychology. In it she looks at the representation of various kinds of visual information and the application of human factors strategies to the design of effective and productive color CAD systems.

The results of these studies are used by Frome to generate guidelines that can be useful in the optimization of color displays of printed wiring boards, menus, messages, and design rule violations. These results are also used by the author to help her design new workstations oriented to increasing the CAD user's productivity, and to derive abbreviations that are easy for users to learn and remember.

A major trend in the evolution of CAE is an increasing emphasis on checking and verification functions that are particularly suitable for use in the early stages of the hardware design cycle. We now know that the earlier one detects and removes design errors, the less costly it will be in the long run. This concept has been the cornerstone and motivation behind the definition of the project described in Larry Dunn's article, "IBM's Engineering Design System Support for VLSI Design and Verification."

This article addresses a complete design and verification process that emphasizes the use of high-level languages and symbology. It discusses a structured approach to design and verification and includes state-of-the-art components such as static (Boolean) verification, the concept of separation between functional simulation and timing verification, synthesis through the use of local transformations, and the utilization of a

* Ofek's work on this issue was done while he worked for IBM at its corporate headquarters.
hardware simulation engine. Some of the material Dunn presents has been previously published and other parts are still in development, but this is the first time that the whole design and verification (DAV) system has been discussed as a comprehensive design tool.

The third article describes a successful technology transfer of the results from a university project to industry and, as such, contains a lesson in fruitful cooperation between industry and academia. Charles Rose and his coauthors describe an interesting mix of technical leading-edge approaches to design and verification of microprocessor-based systems and an account of the process involved in adapting the results of a university project to the needs of industry.

Their article, “N.mPc: A Study in University-Industry Technology Transfer,” highlights a design support which is useful to those engineering personnel who design complex systems by using off-the-shelf components of varying sizes and degrees of complexity. The use of high-level hardware description methods is another significant aspect of the work reported here.

The next article, “The VLSI Design Automation Assistant: An IBM System/370 Design,” by T. J. Kowalski and D. E. Thomas, reports on the progress of one of the leading university projects in the area of expert systems application to logic synthesis. For some time now, it has been suggested that design automation is a good candidate for a technique that allows the utilization of expert knowledge once it is captured and represented in the system.

The Design Automation Assistant is a knowledge-based expert system for synthesizing computer architecture from algorithmic descriptions. It eliminates the need for search by using a large amount of expert knowledge.

The logic functions chosen for experimental synthesis are specific part numbers previously implemented in industry. This choice of a subject made it possible to compare the results with actual industrial designs executed by experienced engineers. At the present time, only the first part of synthesis is addressed, leading to the generation of logic without a complete mapping to a technology realization. However, the output of this system could become a good input to the various silicon compilers and other physical design systems now in use or under development.

One of the most significant technological developments in the design field in recent years has been the emergence of special-purpose hardware engines designed specifically for design automation applications. One such engine is IBM’s Yorktown Simulation Engine (YSE), which is basically a functional logic simulator executing at extremely fast speeds. The fifth article in this issue deals with extending the use of the YSE to pass-transistor simulation.

“Fast Pass-Transistor Simulation for Custom MOS Circuits,” by Z. Barzilai et al., contains an algorithm that provides an efficient tool for custom VLSI circuit design verification and fault simulation. The article shows how to transform a transistor circuit model to a discrete model acceptable to the YSE and how a fault simulation methodology can be defined for the engine. This subject is extremely important in light of the performance issues which face VLSI designers, who need to simulate and test large and complex circuit and logic networks.

The final article in this issue addresses one of the most important areas in the test technology field. Like Barzilai’s article, it, too, seeks to improve the fault simulation process and reduce its high demand on computer resources. To do this, one can use a fast, special-purpose engine, or one may consider novel alternatives, such as the one described in “Critical Path Tracing—An Alternative to Fault Simulation,” by M. Abramovic et al.

The method discussed in this article determines the faults detected by a set of tests using a backtracing algorithm starting at the primary outputs of a circuit. Critical path tracing is an approximate method, but the approximation introduced seldom occurs and its effect is negligible. Early experimental results show that this method is more efficient than conventional fault simulation.

In summary, the articles included in this issue represent recent work in areas ranging from high-level and top-down design to design verification and testing. The omission of papers in the physical design applications area is not intended to imply any lack of importance. Quite the contrary. New algorithms dealing with layout, compaction, floor planning, placement and routing, and others, continue to be the subject of intense development. Such articles will most certainly be published in later issues of IEEE Design & Test.

Acknowledgments

I thank the authors for contributing their time and effort and for their understanding and patience while dealing with me. I also thank the referees, whose helpful comments ensured the quality of the articles in the issue. Finally, I want to express my deep appreciation to Roy Russo, editor-in-chief of this magazine, for his guidance and assistance.

Hillel Ofek’s photo and biography appear on p. 10.