Scanning the Issue

Special Issue on Hardware/Software Co-Design

System design is a very important engineering discipline. Most systems are either electronic in nature or contain an electronic subsystem for monitoring and control. In either case, they are usually designed with a predominant digital component consisting of a hardware platform which executes software application programs. *Hardware/software co-design* means meeting system-level objectives by exploiting the synergism of hardware and software through their concurrent design. Co-design problems have different flavors according to the application domain, implementation technology, and design methodology.

This special issue addresses hardware/software co-design from different angles. The practice of hardware and software design has changed rapidly in the recent years. Digital hardware design has become increasingly more similar to software design because of the use of specialized languages for modeling and synthesis. Current integrated circuits can incorporate one (or more) processor core(s) and memory array(s) on a single substrate. These *systems on silicon* exhibit a sizable amount of embedded software, which provides flexibility for product evolution and differentiation purposes. Therefore, the skills required by digital system designers have evolved and must include the ability of balancing hardware and software.

The first paper in this issue, "Hardware/Software Co-Design," by Guest Editor De Micheli and Gupta introduces the reader to various aspects of co-design. The authors provide a taxonomy of digital design with the purpose of putting different co-design problems in perspective. They distinguish between *embedded system* design and design of *self-contained* computing systems. The former are usually dedicated to performing data processing or control functions of larger systems in which they are embedded. The latter are exemplified by computers. Embedded systems differ radically from traditional computers in the way in which they are programmed, and this has important ramifications on hardware organization and software compiler development.

The article highlights the commonalities and points out the differences in various co-design problems, to help the reader develop a perspective on modern digital system design that relies on computer-aided design (CAD) tools and methods.

Embedded system design is addressed in the paper "Design of Embedded Systems: Formal Models, Validation, and Synthesis" by Edwards *et al.* As the title suggests, this paper reviews design tools and methodologies for the design of reactive real-time embedded systems. The paper stresses the heterogeneous nature of these systems and describes different models of computation that form the mathematical basis of system modeling. Then this paper reviews hardware/software validation and synthesis problems as well as their solutions.

The following three papers address co-design in the telecommunication domain. "Hardware/Software Co-Design of Digital Telecommunication Systems" by Bolsens *et al.* focuses on a vertical slice of problems, ranging from system co-specification to system implementation. This paper also presents a case study: a design environment for specifying, simulating, and synthesizing heterogeneous hardware/software systems. This environment encapsulates hardware and software compilers and supports the interactive synthesis of hardware/software and hardware/hardware interfaces.

The next paper presents instead an extensive survey of trends in the use of embedded processors in wireless communication, multimedia and general telecommunication. "Embedded Software in Real-Time Signal Processing Systems: Application and Architecture Trends," by Paulin *et al.* motivates the use of application-specific architectures in the telecommunication domain and points out the requirements for high-performance software compilation, by analyzing several design examples as well as hardware and software design tools and methods.

A companion paper, "Embedded Software in Real-Time Signal Processing Systems: Design Technologies," by Goossens *et al.* focuses instead on software compilation for application-specific architectures. Such architectures differ from standard processors because *ad hoc* storage and interconnection resources are designed to support specific instruction set mixes. As a result, the task of software compilation is more difficult. At the same time, the challenge of compilation is to provide high-performing compiled programs, thus enabling the programming of these architectures with high-level programming languages while being competitive with other implementation styles.

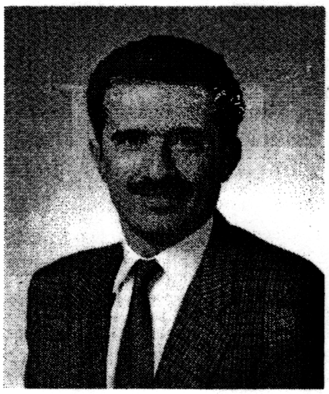
Publisher Item Identifier S 0018-9219(97)02204-4.

"Hardware/Software Co-Design of the Stanford FLASH Multiprocessor" by Heinrich *et al*. deals with multiprocessor design for solving general computing problems. The authors review the design methodology for general-purpose computing systems and stress the importance of concurrent development of hardware and software. They also present, as a case study, the major issues in the design of the memory and communication controller for the FLASH multiprocessor. Such a controller has been co-designed as a programmable processor with a specific architecture.

Overall, this issue presents a set of representative topics in the wide domain of hardware/software co-design. Cross fertilization in different scientific domains is shown by the similarities of approaches in both hardware and software design. Due to the scientific and commercial interest in this field, evolutionary and revolutionary changes in the way digital systems are designed are expected in the coming years.

The Guest Editor would to thank the patient and careful reviewers of the articles of this issue, as well as the Editorial Staff of the PROCEEDINGS OF THE IEEE for their support.

GIOVANNI DE MICHELI
Guest Editor
Stanford University
Stanford, CA



Giovanni De Micheli *(Fellow, IEEE) received the Nuclear Engineer degree from the Politecnico di Milano, Italy, in 1979. He received the M.S. and Ph.D. degrees in electrical engineering and computer science from the University of California at Berkeley in 1980 and 1983, respectively.*

He is Professor of Electrical Engineering and (by courtesy of) Computer Science at Stanford University, Stanford, CA. Prior to his present position, he was with the IBM T. J. Watson Research Center, Yorktown Heights, NY, the Department of Electronics at the Politecnico di Milano, Italy, and at Harris Semiconductor, Melbourne, FL.. His research interests include several aspects of computer-aided design of integrated circuits and systems, especially automated synthesis, optimization, and validation. He is the author of Synthesis and Optimization of Digital Circuits *(McGraw–Hill, 1994), co-author of* High-Level Synthesis of ASICs under Timing and Synchronization Constraints *(Kluwer, 1992), and co-editor of* Hardware/Software Co-design *(Kluwer, 1995), and of* Design Systems for VLSI Circuits: Logic Synthesis and Silicon Compilation *(Martinus Nijhoff, 1986). He was also co-director of the NATO Advanced Study Institutes on Hardware/Software Co-Design (Tremezzo, Italy, 1995) and Logic Synthesis and Silicon Compilation (L'Aquila, Italy, 1986).*

Dr. De Micheli received a Presidential Young Investigator Award in 1988. He received the 1987 Transactions on CAD/ICAS Best Paper Award and two Best Paper Awards at the Design Automation Conferences in 1983 and 1993. He is the Program Chair (for Design Tools) of the 1996–1997 Design Automation Conference. He was Program and General Chair of the International Conference on Computer Design (ICCD) in 1988 and 1989, respectively.