More and more products we buy contain electronics, including phones, cars, entertainment equipment, and toys. Unlike in computers, the electronics used in these applications are deeply embedded and must interact with the user and the real world through sensors and actuators. Embedded electronic systems are often highly distributed, and the parts must collaborate to implement a complete application. Because of performance and cost pressures, embedded systems are built using a wide variety of techniques, including software, firmware, application-specific integrated circuits (ASICs), general-purpose or domain-specific processors, core-based ASICs, Application-Specific Instruction-Set Processors, memory, field-programmable gate arrays, analog circuits, and sensors. The design of complex embedded systems is a difficult problem, requiring designers with skills and experience to identify the best solution.

Designers have had little assistance from electronic design automation tools to perform such tasks. Most system designers must work in an ad hoc manner, at least partly because many products are enhancements of existing systems. There are no widely accepted methodologies or tools available to support the designer in the definition of a functional specification and the subsequent mapping phase onto a target architecture. Therefore, designers usually rely on manual techniques, mainly driven by experience, allowing them to explore only a limited set of alternative solution architectures. An exception is represented by digital signal processing systems, for which there are a longer design tradition and better tools supporting their development.

The rapid evolution of most markets in telecommunications, multimedia, and consumer electronics has added to the pressure on design time. Furthermore, as huge consumer volumes drive down prices, embedded system designers look to the semiconductor industry to integrate functionality up to the system level through new multiprocessor architectures. Design aspects such as low power, architecture exploration, and high-level specification will become essential for producing application-targeted systems-on-chip. Market forces are forcing system design time to be reduced and the design made rapidly adaptable, while always improving the cost, performance, and functionality. Embedded system designers have therefore shifted most of the system functionality to software, with dedicated hardware sparingly used for only the highest performance fixed functionality.

This evolution requires a careful choice of target architecture in terms of size, functionality, and performance and a careful management of the entire design process. A concurrent design process is necessary to try to balance the conflicting requirements that complicate embedded system design. Such a concurrent design process requires collaboration between several development teams: the system architects, the software developers, and the hardware designers. A good methodology is required to shorten overall design time by overlapping activities in all these areas and, most
This unified view allows an easier exploration of alternative solutions that can be evaluated, verified, and tested before the hardware and software components are designed.

This difficult process must be supported by specific electronic design automation tools, which can provide the coordination and communication methods between the system, hardware designers, and software designers during the entire design flow. Academic research in this area started just a few years ago, while commercial tools have been put on the market only recently, covering a few specific parts of the complete design flow. This research area is known as hardware/software codesign, providing a global view of the design of (primarily digital) embedded systems.

The codesign of hardware and software aims at providing a unified view of the system through all phases of the design. This unified view allows an easier exploration of alternative solutions that can be evaluated, verified, and tested before the hardware and software components are designed. System integration in this case is improved with respect to the traditional separate design flows, because the codesign flow maintains an alignment between software and hardware developments. Furthermore, a global framework allows management of the documentation in a global manner, thus keeping track of all modifications that could occur in later design stages in a unified format, after the specification phase.

The goal of this issue is to provide an overview of the research and advances in this codesign process, showing the characteristics of the main tasks and the main issues involved. The articles presented in this issue analyze different aspects of system design methodologies and experiences. Each article presents a methodology and then its application on real industrial examples, mostly in the telecommunications and multimedia areas.

In particular, Marco Sgroi et al. deal with one of the most important problems of specification, by analyzing different models of computation. The authors argue that a formal model is an essential element for the definition of a system-level design methodology. They advocate the necessity of separating functionality, performance, and communication aspects of the system behavior in order to better explore the different architecture alternatives.

Michael Eisenring et al. focus on the problem of coping with the complexity of system design, considering the heterogeneity among components, behaviors, requirements, and constraints that make the entire process difficult to manage effectively. The authors define a methodology that allows multiobjective design space exploration and optimization.

Frank Slomka et al. introduce a methodology for rapid prototyping with configurable and programmable hardware/software systems, allowing early, low-cost validation of the final system in the real environment. Their proposed framework is based on the formal specification languages SDL and Message Sequence Charts.

Alberto Allara et al. present a system design experience performed by a telecommunications company, using a formal system-level specification language and a methodology for hardware/software space exploration supported by prototype tools. They describe the design of a telecommunications controller unit, identifying the hardware and software partitions that met all design constraints and goals. Their results show that such a methodology allows the analysis of different partitioning solutions in a fraction of the time required by previous manual approaches, with a small effort to learn the methodology.

Margarida Jacome and Gustavo de Veciana deal with a specific problem: the design of application-specific core processors suitable for given applications. This is one of the main trends the semiconductor companies follow to provide performance and flexibility to their customers. In fact, very large instruction word
application-specific instruction-set processor cores provide cost/efficiency/flexibility trade-offs that can be very attractive for the portable multimedia consumer products market. Their article addresses the issues of the concurrent synthesis of the cores and the development of the associated retargetable compilers.

**Finally,** Luca Benini et al. discuss the issue of designing low-power embedded systems, whose importance is increasing with the growth of the electronic portable systems market. The design of such systems requires a careful balance of power consumption between computation, communication, and storage. Memory organization and communication can represent a major contribution to the overall power budget, therefore careful design of these components, tailored on the application and the specific core, can provide a relevant power consumption reduction of the overall system. Their article presents a novel methodology for the automatic generation of a memory architecture that is energy efficient and specific for the given application.

**Donatella Sciuto** is currently a full professor at the Dipartimento di Elettronica e Informazione of the Politecnico di Milano, Italy. Her research interests cover methodologies for testability analysis and testable design of complex circuits and methodologies for hardware/software codesign of embedded systems. She received her Laurea in electronic engineering in 1984. She received her PhD in electrical and computer engineering in 1988 from the University of Colorado, Boulder. She is a member of IEEE, IFIP 10.5, SIG-VHDL, and EDAA. She is a member of program committees of EDA conferences: DAC, ICCAD, DATE, CODES/CASHE, DFT, and FDL. She is associate editor of the journal *Design Automation of Embedded Systems.*

Send comments and questions to Donatella Sciuto, Dipartimento di Elettronica e Informazione, Politecnico di Milano, P.zza L. da Vinci 32, 20133 Milano, Italy; sciuto@elet.polimi.it.