Abstract

Ubiquitous embedded systems are revolutionizing our daily lives. Whole systems on a chip deliver unprecedented computation power at ever decreasing costs. However, their complexity makes their design with traditional RTL-based flows extremely challenging. Complexity in such systems arises not only from the diversity of the technologies, from RF front-ends to baseband DSP software, that must be integrated on-chip, but also from the fact that such systems must be increasingly built from parts that have been designed separately and using different tools and flows. High abstraction levels and component re-use are essential. Often such systems are highly networked and rely on sophisticated communication mechanisms. Architectural design and performance analysis of such networked system-chips is a crucial part of the embedded system design process.

Two basic methods for tackling the growing complexity of system-on-chip design are emerging. Formal specification and design methods, and platform-based design. Formal methods are essential to capture precisely the designer intent at the highest possible level of abstraction, and to guide him down towards an implementation. Design errors are reduced by the use of synthesis techniques, as well as by the availability of a variety of validation techniques. Moreover, the orthogonalization of concerns (e.g., computation vs. communication, function vs. architecture) promotes extensive reuse of both parts of an application (derivative designs) and of previously used architectures.

A platform in this context is the precise specification of a high-level implementation target that supports both a variety of lower level implementations, and a variety of higher-level applications. A classical example is the PC, that has supported an explosive growth of both applications and implementations. In embedded systems-on-chip platform clearly need to be application-specific. A typical platform includes one or more processors, memory, one or more bus hierarchy levels, and peripherals devoted to application-specific functions (e.g., MPEG decoding, filtering, A/D conversion, ...). Mapping to a platform involves identifying which portions of the application will be implemented on each computation and communication resource. Communication must be refined from high-level primitives (e.g., FIFOs) down to platform primitives (e.g., interrupts, memory buffers, ...). Functionality must be synthesized and scheduled based on platform-specific cost, memory and performance constraints.

In this tutorial we will cover both aspects in detail, illustrating first the languages and Models of Computation available to the system-level designer to capture precisely and unambiguously the
requirements. We will discuss for what application domain and platform each language is most appropriate, focusing mostly on platform-independent languages and MOCs, since they support the greatest freedom in mapping choice.

We will then discuss how the architecture of the platform, and the services it offers to the application designer, can also be formally and compactly captured and specified. We will show how the mapping paradigm can be used to select an implementation for the functional blocks and their communication, and how simulation and implementation methods can be derived automatically. In particular, we will describe how software estimation and synthesis for reactive real-time systems can be competitive with hand design, while retaining the ease of re-use typical of high-level specifications.

In the next part of the tutorial, we will establish the importance of on-chip communication architectures in determining the performance of System-on-Chips (SoCs). We will track several on-chip communication architectures that are in use today, including different types of bus architectures, token ring architectures, and crossbar switches. We will analyze the performance of the existing communication architectures under different classes of on-chip communication traffic. We will also describe novel on-chip communication architectures that are being developed to satisfy the growing needs of fast and concurrent on-chip communication in very high-performance systems like network processors.

Next, we will describe efficient methodologies that can be used to analyze and design customized on-chip communication architectures for new application-specific platforms, as well as optimally map an application's communication requirements to the communication architecture present in an existing platform. We will also discuss reconfigurable on-chip communication architectures that allow runtime adaptation of the communication protocols to changing communication demands of the platform. Besides performance, we will analyze the energy consumed by on-chip communication architectures, and describe techniques to minimize such energy consumption. Lastly, we will analyze the impact of deep sub-micron technologies on SoC communication, and discuss the needs for development of noise-aware on-chip communication.

Luciano Lavagno graduated in Electrical Engineering from Politecnico di Torino (Italy) in 1983. From 1984 to 1988 he was with CSELT Laboratories (Torino, Italy). In 1992 he received his Ph.D. in Electrical Engineering and Computer Science from the University of California at Berkeley. Between 1993 and 1998 he was an Assistant Professor with the Department of Electronics of Politecnico di Torino. Since 1993 he has been the architect of the POLIS project developing a complete hardware/software co-design environment for control-dominated embedded systems. Since 1998 he has been an Associate Professor with the Department of Electrical, Management and Mechanical Engineering (DIEGM) of the University of Udine, Italy. His research interests include the synthesis of asynchronous circuits, and the concurrent design of mixed hardware and software systems.

Dr. Lavagno is the author of a book on asynchronous circuit design, the co-author of a book on hardware/software co-design of embedded systems, and has published over 100 journal and conference papers. He has served as technical committee member of several international conferences in his field (namely the Design Automation Conference, the International Conference on Computer Aided Design, the conference on Design Automation and Test in Europe) and as technical committee member or chair of several workshops and symposia.

Sujit Dey received the Ph.D. degree in Computer Science from Duke University in 1991. From 1991 to 1997, he was at the NEC C&C Research Laboratories, Princeton, NJ, where he was a Senior Research Staff Member. While at NEC, he developed several design methods and tools for the design of high performance and low power system-on-chips, leading to several technology transfers both within NEC as well as outside. In 1998, he joined the University of California, San Diego, where he is an Associate
Professor in the Electrical and Computer Engineering Department. His research group at UCSD is developing configurable platform architectures, consisting of adaptive wireless applications, protocols, and hardware-software architectures for next-generation wireless network appliances. He is affiliated with the California Institute of Telecommunications and Information Technology, the UCSD Center for Wireless Communications, and the DARPA/MARCO Gigascale Silicon Research Center. Dr. Dey has co-authored more than 100 publications, including journal and conference papers, a book and several book chapters. He received Best Paper awards at the Design Automation Conferences in 1994, 1999, and 2000, and the 11th VLSI Design Conference in 1998, and several best paper nominations. He is a co-inventor of 8 U.S. patents, and has several others pending. He has presented numerous full-day and embedded tutorials, and participated in panels, in the topics of low-power wireless systems design, hardware-software embedded systems, and deep sub-micron system-on-chip design and test. He has been the General Chair and Program Chair, and member of organizing and program committees, of several IEEE conferences and workshops.

Rajesh Gupta (Ph. D. Stanford, M.S. UC Berkeley) is an associate professor in Information and Computer Science at University of California, Irvine. His research interests are in system-level design for embedded and portable systems, VLSI design, and adaptive system architectures. He worked as an assistant professor at University of Illinois, Urbana-Champaign from 1994 through 1996. Prior to that he was at Intel Corporation in Santa Clara, California where he worked as a member on a number of processor design teams. He is co-author of three patents on PLL-based clock circuit; synthesis with regularity and system-on-chip modeling and a patent (filed) on data-path synthesis. He is author of a book on Co-synthesis of Hardware and Software for Digital Embedded Systems published by Kluwer Academic in 1995. At UCI, he leads an effort on Adaptive Memory System Architectures and co-leads an effort on Compiler-Controlled Power/Performance Management both sponsored by DARPA programs. Gupta serves as Chair of the CANDE technical committee and as a board of governor of the IEEE Circuits and Systems Society. He also serves as associate editor-in-chief of IEEE Design and Test and on the editorial board of IEEE Transactions on CAD.