The Softening of Hardware
pp. 27-34
Frank Vahid

In the 1940s, when modern computing began, engineers tended to view computers and the programs running on them as unified entities. Now, after decades in which software and hardware developed along separate paths, we seem to have come full circle. The hardware on which our programs run is softening, thanks to embedded systems.

These systems force designers to work under incredibly tight constraints. To understand the technologies developed to satisfy these constraints, we must first distinguish the underlying embedded systems elements.

A Decade of Hardware/Software Codesign
pp. 38-43
Wayne Wolf

The term hardware/software codesign, coined about 10 years ago, describes a confluence of problems in integrated circuit design. By the 1990s, it became clear that microprocessor-based systems would be an important design discipline for IC designers as well. Large 16- and 32-bit microprocessors had already been used in board-level designs, and Moore’s law ensured that chips would soon be large enough to include both a CPU and other subsystems.

Multiple disciplines inform hardware/software codesign. Computer architecture tells us about the performance and energy consumption of single CPUs and multiprocessors. Real-time system theory helps analyze the deadline-driven performance of embedded systems. Computer-aided design assists hardware cost evaluation and design-space exploration.

Metropolis: An Integrated Electronic System Design Environment
pp. 45-52
Felice Balarin, Yosinori Watanabe, Harry Hsieh, Luciano Lavagno, Claudio Passerone, and Alberto Sangiovanni-Vincentelli

Today, the design chain lacks adequate support, with most system-level designers using a collection of unlinked tools. The implementation then proceeds with informal techniques involving numerous human-language interactions that create unnecessary and unwanted iterations among groups of designers in different companies or different divisions.

The move toward programmable platforms shifts the design implementation task toward embedded software design. When embedded software reaches the complexity typical of today’s designs, the risk that the software will not function correctly increases exponentially. The Metropolis project seeks to develop a unified framework that can cope with this challenge.

SystemC Cosimulation and Emulation of Multiprocessor SoC Designs
pp. 53-59
Luca Benini, Davide Bertozzi, Davide Bruni, Nicola Drago, Franco Fummi, and Massimo Poncino

SystemC is an open source C/C++ simulation environment that provides several class packages for specifying hardware blocks and communication channels. The design environment specifies software algorithmically as a set of functions embedded in abstract modules that communicate with one another and with hardware components via abstract communication channels.

The authors describe a simulation environment that targets heterogeneous multiprocessor systems. They are currently working to extend their methodology to more complex on-chip architectures.

A Formal Approach to MpSoC Performance Verification
pp. 60-67
Kai Richter, Marek Jersak, and Rolf Ernst

Multiprocessor system on chip designs use complex on-chip networks to integrate different programmable processor cores, specialized memories, and other components on a single chip.

MpSoCs have become the architecture of choice in many industries. Their heterogeneity inevitably increases with intellectual-property integration and component specialization. System integration is becoming a major challenge in their design.

Simulation is state of the art in MpSoC performance verification, but it has conceptual disadvantages that become disabling as complexity increases. Formal approaches offer a systematic alternative.

Domain-Specific Codesign for Embedded Security
pp. 68-74
Patrick Schaumont and Ingrid Verbauwhede

Systems with multiple design domains require codesign of application domains. Dedicated hardware processors implement the application domains and software integrates them.

The authors use ThumbPod, a prototype embedded security application, for remote identification applications such as intelligent keys or electronic payments. Sun’s K virtual machine offers an infrastructure for secure code download and execution. The virtual machine also integrates the cryptoprocessor through a native interface.