Tutorials

A — Compilers for Embedded Processors

Peter Marwedel, Dortmund University, Germany

This talk responds to the rapidly increasing use of embedded processors for implementing systems. Such processors come in the form of discrete processors as well as in the form of core processors. They are available both from vendors and within system companies. Applications can be found in most segments of the embedded system market, such as automotive electronics and telecommunications. These applications require extremely efficient processor architectures, optimised for a certain application domain or even a certain application.

We will present techniques for retargeting compilers to new architectures easily. These techniques are motivated by the need for domain- or application-dependent optimisations of processor architectures. The scope for such optimisations should not be restricted to hardware architectures but has to include the corresponding work on compilers as well. We will show how compilers can be generated from descriptions of processor architectures. Presented techniques aim at bridging the gap between electronic CAD and compiler generation.

B — Design, Validation & Test of Core-Based System Chips

Yervant Zorian, LogicVision, USA
Sujit Dey, University of California at San Diego, USA

A major key to the success of the new embedded system-on-a-chip design methodology lies in the development and use of pre-designed, pre-characterised, and pre-verified functional blocks called cores or Intellectual Property. A wide range of industrially available cores, including processor, microcontrollers, DSP, interface, multimedia, and communications/networking cores will be described. System-on-a-chip design methodologies using cores will be described, including required software and hardware development support, software and hardware estimation tools, and issues in system-level integration. Validation methodologies currently used to verify the correctness of such systems will be described, including emulation, in-circuit emulation, compliance test environments, instruction-set simulation, and hardware-software co-simulation. The tutorial will also cover the test related aspects of systems-on-a-chip containing embedded cores. Finally it will discuss the recent challenges and adopted strategies to implement an integrated test strategy from embedded cores to a system-on-a-chip.

C — Design Techniques for Low-Power Systems

Christian Piguet, CSEM, Neuchatel, Switzerland

Low-voltage and low-power digital design has to be performed at several levels such as architecture, logic and basic cell levels, while considering activity, capacitance, frequency and supply voltage reduction. Comparison of energy-efficient architectures will be performed while using energy/operation and throughput. Examples of activity and capacitance reduction will be provided for a low-power digital cell library as well for logic modules. Various structures of Finite State Machines including gated clock and asynchronous structures are analysed from the activity point of view. Logic design at low supply voltage will be demonstrated while considering complex gate decomposition and parallelised logic circuits such as parallelised memories, synchronous counters and shift registers that present the same throughput than the non parallelised structures. Low-power embedded microcontrollers for portable applications will be presented. The basic idea behind low-power microprocessor architectures is to reduce the number of basic steps and clock cycles for the execution of a given task. Power reduction can be addressed at the software level as well as at the architecture level. Various architectures of 8-bit and 16-bit microcontrollers will be presented and compared. Evolution towards low-power devices results in new pipelined, multithreaded or parallel architectures for microcontrollers and DSP processors.

D — Formal Verification

Hans Eveking, T.U. Darmstadt, Germany

In the first part of the tutorial, an introduction to different types of decision diagrams (OBDD's, FDD's, MTBDD's, BMD's) and their applications is given. In the second part, a number of advanced techniques for the verification of large systems like processors are discussed (Burch/Dills' methods for pipelined designs, STE - symbolic state evaluation, etc.).

E — ILP Compilation Techniques for Embedded Systems

Alex Nicolau, University of California at Irvine, USA
Nikil Dutt, University of California at Irvine, USA

Software is already a significant component in today's embedded systems, and it is expected to greatly dominate the design of future embedded systems. Compilation technology is vital for the effective use of software in embedded systems. This half-day tutorial will review the basic issues and advances in Instruction-Level Parallelism (ILP) compilation techniques for embedded systems. We begin with a brief historical perspective describing experiments and analysis for ILP. We will then describe a number of compiler techniques for ILP, including trace scheduling, superblocks, percolation scheduling, software pipelining and memory disambiguation. In the context of designing embedded systems, we will describe a number of memory-related optimisations, including register allocation, instruction and data caching, data reorganisation and alignment. Next, the intrinsic phase coupling and ordering problem between the compiler subtasks of instruction selection, scheduling and register allocation will be presented. We will describe approaches to code generation that combines these subtasks into a unified framework that allows design tradeoffs to be made "on-the-fly" during compilation. The tutorial will conclude with a discussion of embedded processor characteristics and compiler opportunities for processor-core based designs.

This tutorial is designed for researchers, system designers, hardware engineers and managers who wish to understand the capabilities, limitation and status of ILP compilation techniques for embedded systems. No specialised knowledge of compilers or embedded systems is required; the tutorial will illustrate concepts through extensive use of examples.

F — Microsystems Technology and Applications

J. Malcolm Wilkinson, Technology for Industry (TFI), UK

Microsystems are intelligent miniaturised systems comprising sensing, processing and/or actuating functions, normally combining two or more electrical, mechanical or other properties on a single chip or multi-chip hybrid. They provide increased functionality, improved performance and reduced system cost to a large number of products. This tutorial will provide an overview of the evolution, types, fabrication and application of modern microsystems, including examples of technological strategies suitable for direct utilisation by small and medium size high-tech companies. The tutorial includes specialised and low-cost processes; 3D/2D structures; novel materials for optical, magnetic, biological and thermal sensors; special packaging techniques including flip-chip and anodic bonding (direct or indirect contact to a liquid or gas, intimate thermal or mechanical coupling, magnetic coupling, EMI shielding or conversely RF coupling); microsystems design and manufacture flow, CAD tools; packaging and test issues; overcoming factors limiting growth; microsystems strategies for small and medium companies.

Audience: Engineers, design managers and small/medium company technologists interested in the development and utilisation of silicon microsystems technology in numerous industrial applications. Materials engineers, process engineers, packaging engineers, production engineers, microsystems CAD tool developers.