TUTORIAL 3 LOW-POWER/LOW-ENERGY EMBEDDED SOFTWARE: WHAT, WHY AND HOW?

Speakers:

Luca Benini - DEIS Univ. di Bologna, Bologna, Italy *Nikil Dutt* - Univ. of California, Irvine, CO *Peter Marwedel* - Univ. of Dortmund, Dortmund, Germany

Background: Power and energy issues have become important, if not primary constraints for many embedded systems. With the increasing importance and content of software in contemporary embedded systems, designers need to understand how software issues can affect power dissipation. Recently several research efforts have addressed software energy and power issues, and there is growing industrial interest in understanding the interaction between software and power/energy.

Description: With a focus on programmable embedded systems, this tutorial will:

- survey the interaction of architecture, operating systems, compilers and memories from a power/energy focus,
- present specific contributors of each part to power and energy, and
- outline software techniques for minimization of power/energy.

In the first section, an introduction to embedded processors and energy consumption of embedded software is provided.

Next, architectures of embedded systems are presented. The focus is on programmable embedded systems and their memory organization. Differences between different types of memories are explained and their impact on the resultant energy consumption are highlighted.

In the next section, energy-saving compiler optimizations are considered. These are based on models of the energy consumption of processor-based systems. For most systems, a major portion of the energy is consumed by memory references. This energy can be reduced by exploiting memory hierarchies. The potential of standard compiler optimizations for saving energy is also considered in this section of the tutorial.

The final section describes system software and real-time operating system (RTOS) issues. This will include hardware for RTOS-based power management, software support for power management, power-aware process scheduling and power-aware device management. Exploitation of application-specific information and power management of distributed systems will also be covered.