TUTORIAL 1: November 10, 2008  1:30-5:00pm  
Reliable System Design: Models, Metrics and Design Techniques

Organizer: Subhasish Mitra - Stanford Univ., Stanford, CA

Speakers: Ravishankar K. Iyer - Univ. of Illinois, Urbana-Champaign, IL  
Kishor Trivedi - Duke Univ., Durham, NC  
James W. Tschanz - Intel Corp., Hillsboro, OR

Design of reliable systems meeting stringent quality, reliability, and availability requirements is becoming increasingly difficult in advanced technologies. The current design paradigm, which assumes that no gate or interconnect will ever operate incorrectly within the lifetime of a product, must change to cope with this situation. Future systems must be designed with built-in mechanisms for failure tolerance, prediction, detection and recovery during normal system operation. This tutorial will focus on models and metrics for designing reliable systems, algorithms and tools for modeling and evaluating such systems, will discuss a broad spectrum of techniques for building such systems with support for concurrent error detection, failure prediction, error correction, recovery, and self-repair. Complex interplay between power, performance and reliability requirements in future systems, and associated constraints will also be discussed.

TUTORIAL 2: November 11, 2008  8:30am-12:00pm  
Architecting Parallel Programs

Organizer: Joel Phillips - Cadence Research Labs, Berkeley, CA

Speakers: Kurt Keutzer - Univ. of California, Berkeley, CA  
Michael Wrinn - Intel Corp., Hillsboro, OR

The current shift from sequential to multicore and manycore processors presents serious challenges to software developers. A significant part of the industrial and research communities believes that either a) they can squeak by or b) the right compiler, parallel language etc will save them. Such ad hoc responses are likely to prove neither correct nor sustainable. To systematically find and exploit parallelism, and to achieve forward scalability – that is, designs which efficiently scale to much larger numbers of cores -- will require re-architecting software applications such as EDA.

We believe that the key to re-architecting software is the use of design patterns and a pattern language. Furthermore, structural patterns (aka architectural styles) and computational patterns (aka the thirteen dwarfs) are the key high-level design patterns. The patterns are then used to create programming frameworks that can be used to facilitate implementation of the software architecture.

This tutorial presents the most recent research results by UC Berkeley and Intel. We will present our working pattern language and give examples on its use in EDA and other application areas.
TUTORIAL 3: November 11, 2008  1:30-5:00pm
Embedded Software Verification: Challenges and Solutions

Organizers:  Chao Wang - NEC Labs America, Princeton, NJ
             Malay Ganai - NEC Labs America, Princeton, NJ

Speakers:  Chao Wang - NEC Labs America, Princeton, NJ
           Shuvendu Lahiri - Microsoft Corp., Redmond, WA
           Daniel Kroening - Oxford Univ., Oxford, United Kingdom

Embedded software are becoming more and more pervasive in our lives, and many application domains have
very high reliability requirements. Ensuring high software quality while still maintaining software productivity
is a challenging task. In order to address this challenge, more formal analysis and automated verification
techniques are needed in addition to standard software testing.

In this tutorial, we will showcase the important ideas and techniques of software formal verification, including
static program analysis, program modeling and (bounded) model checking, and predicate abstraction
refinement. We will emphasize some of the key techniques that have been successfully adopted by recent,
industrial-strength software verification tools. In this tutorial, we will focus on detecting bugs in sequential
programs written in C/C++ for portable devices as well as for general purpose platforms.

The tutorial will be tailored to the ICCAD audience by emphasizing the use of decision procedures such as
BDDs, Presburger arithmetic, bit-vector arithmetic, SAT and SMT solvers. Many of these techniques have been
used in the context of analyzing, optimizing, and verifying IC designs.

By attending this tutorial, the audience will get a better understanding of the challenges and potential solutions
of embedded software verification.

TUTORIAL 4: November 12, 2008  8:30am-12:00pm
Nanolithography and CAD Challenges for 32nm/22nm and Beyond

Organizer:  David Z. Pan - Univ. of Texas, Austin, TX

Speakers:  Stephen Renwick - Nikon Precision, Inc., Belmont, CA
           Vivek Singh - Intel Corp., San Jose, CA
           Judy Huckabay - Cadence Design Systems, Inc., San Jose, CA

The semiconductor industry is stuck at 193nm lithography as the main workhorse for manufacturing integrated
circuits of 45nm and most likely 32nm nodes. On one hand, many novel approaches are being developed to
extend the 193nm lithography, including immersion, double patterning, and exotic resolution enhancement
techniques. On the other hand, next generation lithography, in particular, extreme ultra violet lithography
(EUVL) is projected by ITRS as the main contender for technology nodes at or below 22nm, though significant
challenges still exist from both technology and economy aspects. This tutorial will cover key nanolithography
and CAD challenges with possible solutions for 32nm/22nm (and beyond?), from the underlying
hardware/equipment perspectives (for double patterning, EUV, and so on), to the computational lithography
aspects (extreme RET, inverse lithography, pixelated mask, etc.), and to the key EDA issues on nanolitho-
friendly layouts (e.g., double patterning compliance layout, and so on).