Overview of Tutorials

Tutorial 1

Design Verification and Diagnosis

Dhiraj Pradhan, Texas A&M University
Jacob Abraham, University of Texas at Austin

Description:

This tutorial will cover both fundamentals and advances in design verification. With 100+ million transistor chips becoming a reality, traditional hardware verification methods using simulation have been proven inadequate. Part of the tutorial will discuss a unified approach using Binary Decision Diagram representations of Boolean functions and finite-state machines to formally verify the correctness of hardware designs and implementations from the transistor level up to the behavioral level. It will cover current practice in industry as well as recent research results in these areas, including the use of abstractions and partitioning to improve the problem of state space explosion. Recent developments in verification using ATPG based methods such as Recursive Learning will also be discussed. State-of-the-art university tools in this field will be described, and examples of verifying real chips from industry application of the tools will be included. Open problems and directions for research will also be pointed out. Attendees will receive copies of the notes and key publications.

Tutorial 2

New Trends in Designing and Testing VLSI Systems

Sujit Dey, NEC USA
Peter Marwedel, University of Dortmund

Description:

This tutorial provides a comprehensive overview of the new methodologies that are emerging for the design of electronic systems, including high level synthesis, system level synthesis, hardware-software co-design, and core-based design. Various design and analysis issues of each of the advanced methodologies will be discussed, with special emphasis on the testing challenges and opportunities that arise with each new design methodology. Existing test generation and design-for-testability techniques to generate testable non-scan, partial scan, and BIST designs from RT-level, behavioral, and system level specifications are reviewed. Core-based design will be motivated and its consequences will be analyzed. Consequences that will be discussed, will include, for example, the need for hardware-software co-design environments, requirements for compilers, and opportunities for testing. Attendees will receive copies of the notes.