

TUTORIAL 3

SPECIFICATION AND DESIGN OF MULTIMILLION GATE SOCS

Speakers:

Ramesh Chandra – ST Microelectronics, San Diego, CA

Joerg Henkel – NEC USA, Inc., Princeton, NJ

Preeti Ranjan Panda - Indian Institute of Tech., Delhi, India

Sridevan Parameswaran – Univ. of New South Wales, New South Wales, Australia

Loganath Ramachandran – Synopsys, Inc., Mountain View, CA

Background: Advances in technology have made it possible to integrate several multi-million transistors on a single chip. However, design and verification teams face several challenges not seen before: modeling and verifying entire system functionality, closing early on the system level architecture, extensive simulations of the hardware and software components and the final path to implementation of the entire SOC. New specification languages, new modeling and design paradigms, and new verification methods are evolving to meet these challenges.

Description: This tutorial will cover the state-of-the-art in specification and design methodologies. It will benefit design engineers, managers, researchers and students who are interested in understanding these new modeling, design and verification paradigms. This full-day tutorial is structured into five major parts.

- (a) System level design challenges from a designer's perspective
- (b) Specification languages for system level design
- (c) Functional and architectural modeling and verification
- (d) Platform based design concepts
- (e) Path from specification to implementation.

An overview of the real challenges faced by system level designers will be presented with examples. The weaknesses of traditional design methods will be highlighted, thereby motivating the need for newer system level design tools and methodologies.

Several modeling languages that focus on system level have been proposed in the recent years. Languages such as SystemC, SpecC, and Superlog represent different approaches for specifying the entire system. In this section of the tutorial, we will present an overview of these approaches, explain the specific properties of these languages and discuss in detail the application of various system modeling principles in SystemC.

In the next section of the tutorial we will focus on system level verification, which includes both functional and architectural verification. Functional verification addresses the modeling of the entire system functionality in order to understand system level performance issues. In general this requires heterogeneous modeling involving several models of computation. Architectural verification addresses architectural tradeoff issues that enable designers to close on the architecture early in the design cycle. This is achieved using transaction level models (TLMs) of the system components.

One popular method to manage SOC design complexity is to define a platform, which is an abstraction that embodies the lower level details. Such a platform along with an associated RTOS, forms the backbone for the SOC design. The next part of the tutorial will deal with the underlying concepts of platform-based design methodologies and their advantages compared to traditional system design concepts. Examples of commercial platforms will be presented.

The path to implementation from system level specification can be in any one of three domains: software; hardware; or hardware/software. In the software realization path, we will present modern compiler tools and techniques specifically targeted towards application specific systems. In the hardware section, we will present techniques for synthesis and discuss quality of results for designs created from system level specifications. Finally in the hardware/software implementation we will show the procedures that can lead to single/multi processor systems and ASIPs.